

**Санкт-Петербургское государственное бюджетное профессиональное  
образовательное учреждение  
«Академия управления городской средой, градостроительства и печати»**

**УТВЕРЖДАЮ**  
Заместитель директора  
по учебно-методической работе  
\_\_\_\_\_ **О.В. Фомичёва**  
«\_\_\_» \_\_\_\_\_ 20\_\_\_ г.

**Методические рекомендации по организации и  
проведению практических занятий**

***ПМ.02 «АДМИНИСТРИРОВАНИЕ БАЗ ДАННЫХ»***

***МДК 02.01 Управление и автоматизация баз данных***

**для специальности  
специальности 09.02.13 Интеграция решений с применением технологий  
искусственного интеллекта**

**Форма обучения – очная**

**Санкт-Петербург  
2025**

Разработчик: Ипатова С.В./Оболенская Е.Г., методисты СПб ГБПОУ АУГСГиП

Одобрены на заседании цикловой комиссии

Общетехнических дисциплин и компьютерных технологий

Протокол № 4

09.12.2025 г.

Председатель цикловой комиссии:

Шурухина И.Е.

В результате изучения профессионального модуля обучающихся должен освоить основной вид деятельности ВД2. «Администрирование баз данных» и соответствующие ему общие компетенции и профессиональные компетенции:

#### 1.1.1. Перечень общих компетенций

Код	Наименование общих компетенций
<b>ОК 01</b>	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам
<b>ОК 02</b>	Использовать современные средства поиска, анализа и интерпретации информации, и информационные технологии для выполнения задач профессиональной деятельности
<b>ОК 03</b>	Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях
<b>ОК 04</b>	Эффективно взаимодействовать и работать в коллективе и команде
<b>ОК 05</b>	Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста
<b>ОК 06</b>	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения
<b>ОК 07</b>	Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях
<b>ОК 08</b>	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
<b>ОК 09</b>	Пользоваться профессиональной документацией на государственном и иностранном языках

#### 1.1.2. Перечень профессиональных компетенций

Код	Наименование видов деятельности и профессиональных компетенций
<b>ВД 2</b>	Администрирование баз данных
<b>ПК 2.1</b>	Выявлять проблемы, возникающие в процессе эксплуатации баз данных.
<b>ПК 2.2</b>	Осуществлять процедуры администрирования баз данных.
<b>ПК 2.3</b>	Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации.
<b>ПК 2.4</b>	Формировать требования хранилищ банка данных для обучения.
<b>ПК 2.5</b>	Подготавливать данные для базы знаний.

#### 1.1.3. В результате освоения профессионального модуля обучающийся должен:

<b>Иметь практический опыт</b>	<ul style="list-style-type: none"> <li>– Идентификации проблем, связанных с нормальным функционированием базы данных;</li> <li>– Восстановления системы.</li> <li>– Администрирования сервера баз данных;</li> <li>– Участия в администрировании отдельных компонент серверов;</li> <li>– Документирования результатов аудита безопасности информации;</li> <li>– Использования процедуры резервного копирования баз данных;</li> </ul>
--------------------------------	---

	<ul style="list-style-type: none"> <li>– Использования процедуры восстановления баз данных</li> <li>– Подготовки документации по формированию требований хранилищ банка данных</li> <li>– Проектирования, разработки и эксплуатации баз данных</li> </ul>
<b>Уметь</b>	<ul style="list-style-type: none"> <li>– Производить идентификацию проблем, связанных с нормальным функционированием базы данных;</li> <li>– Принимать решения по локализации проблем, связанных с нормальным функционированием базы данных;</li> <li>– Документировать внештатные ситуации связанные с нормальным функционированием базы данных;</li> <li>– Осуществлять основные функции по администрированию баз данных;</li> <li>– Настраивать политики безопасности при работе с сервером баз данных</li> <li>– Дать независимую оценку уровня безопасности</li> <li>– Производить регламентное обновление программного обеспечения</li> <li>– Разрабатывать перечень рекомендаций по дальнейшей эксплуатации БД с максимальной защитой хранящейся информации.</li> <li>– Производить формирование требований к обработке данных и их извлечению;</li> <li>– Добавлять, удалять и изменять данные в базе данных;</li> <li>– Производить операции по импорту и экспорту данных в различных форматах</li> </ul>
<b>Знать</b>	<ul style="list-style-type: none"> <li>– Основные коды ошибок при работе с базой данных;</li> <li>– Методы и средства устранения ошибок, возникающих при работе с базой данных;</li> <li>– Тенденции развития банков данных;</li> <li>– Технология установки и настройки сервера баз данных;</li> <li>Требования к безопасности сервера базы данных;</li> <li>– Протоколы безопасности при работе с базой данных;</li> <li>– Методы и средства защиты информации от несанкционированного доступа;</li> <li>– Уровни угроз безопасности информации</li> <li>– Формы документов, необходимых для формирования, ведения и использования банка данных</li> <li>– Типы данных хранения информации в базе данных</li> </ul>

## *Практические работы*

тема	название ПР	часы
МДК 02.01 Управление и автоматизация баз данных		
<b>Тема 1.1. Установка и настройка программного обеспечения для администрирования баз данных</b>	<b>Практическая работа 1</b> Установка СУБД MySQL и настройка службы на локальном сервере. <b>Практическая работа 2</b> Установка PostgreSQL и настройка параметров конфигурации (порт, логирование). <b>Практическая работа 3</b> Установка Oracle Database и настройка окружения (переменные PATH, ORACLE_HOME). <b>Практическая работа 4</b> Установка MongoDB и настройка репликации для отказоустойчивости. <b>Практическая работа 5</b> Установка Microsoft SQL Server и настройка параметров аутентификации.	6
	<b>Практическая работа 6</b> Создание и управление пользователями в установленной СУБД (назначение ролей и прав доступа). <b>Практическая работа 7</b> Настройка автоматического резервного копирования базы данных. <b>Практическая работа 8</b> Настройка параметров производительности СУБД (буферизация, размер кэша, максимальные соединения). <b>Практическая работа 9</b> Обновление версии установленной СУБД с сохранением данных. <b>Практическая работа 10</b> Настройка удалённого доступа к установленной СУБД через конфигурационные файлы.	6
<b>Тема 1.2. Установка и настройка программного обеспечения (ПО) для обеспечения работы пользователей с базами данных</b>	<b>Практическая работа 11</b> Установка и настройка клиента SQL Workbench для работы с базой данных MySQL. <b>Практическая работа 12</b> Установка и настройка pgAdmin для управления PostgreSQL. <b>Практическая работа 13</b> Установка и настройка Microsoft Management Studio (SSMS) для работы с SQL Server. <b>Практическая работа 14</b> Установка и настройка DBeaver для подключения к различным типам баз данных. <b>Практическая работа 15</b> Установка и настройка библиотек Python для взаимодействия с базами данных (pymysql, psycopg2).	6
	<b>Практическая работа 16</b> Настройка соединения клиента SQL Workbench с сервером MySQL (локально и удалённо). <b>Практическая работа 17</b> Настройка пользователей и прав доступа через pgAdmin для PostgreSQL. <b>Практическая работа 18</b> Создание и выполнение SQL-запросов с использованием DBeaver. <b>Практическая работа 19</b> Настройка интеграции баз данных с клиентским ПО через ODBC-драйверы. <b>Практическая работа 20</b> Проверка совместимости клиентских приложений с установленным ПО для взаимодействия с базами данных.	6

<p><b>Тема 1.3. Управление доступом к базам данных</b></p>	<p><b>Практическая работа 21</b> Создание пользователей и групп в MySQL и назначение прав доступа (GRANT, REVOKE).</p> <p><b>Практическая работа 22</b> Настройка ролей и прав доступа в PostgreSQL для различных пользователей.</p> <p><b>Практическая работа 23</b> Управление правами доступа в Microsoft SQL Server с использованием SQL Server Management Studio (SSMS).</p> <p><b>Практическая работа 24</b> Настройка аутентификации и шифрования соединения в MySQL.</p> <p><b>Практическая работа 25</b> Использование встроенных ролей в Oracle Database для управления доступом.</p>	<p>6</p>
	<p><b>Практическая работа 26</b> Конфигурация прав доступа для разных уровней пользователей (администратор, аналитик, пользователь) в PostgreSQL.</p> <p><b>Практическая работа 27</b> Создание политики безопасности в Microsoft SQL Server для ограничения действий пользователей.</p> <p><b>Практическая работа 28</b> Проверка и настройка доступа к базе данных через файл конфигурации в MySQL.</p> <p><b>Практическая работа 29</b> Реализация сценария управления доступом через роли и группы пользователей в Oracle Database.</p> <p><b>Практическая работа 30</b> Аудит действий пользователей в базе данных с помощью встроенных инструментов PostgreSQL.</p>	<p>6</p>
<p><b>Тема 1.4. Резервное копирование баз данных</b></p>	<p><b>Практическая работа 31</b> Создание резервной копии базы данных MySQL с использованием утилиты mysqldump.</p> <p><b>Практическая работа 32</b> Резервное копирование базы данных PostgreSQL с помощью pg_dump и pg_dumpall.</p> <p><b>Практическая работа 33</b> Настройка и выполнение резервного копирования в Microsoft SQL Server с использованием SSMS.</p> <p><b>Практическая работа 34</b> Автоматизация резервного копирования базы данных MongoDB с использованием скриптов.</p> <p><b>Практическая работа 35</b> Создание и управление резервными копиями Oracle Database с помощью RMAN (Recovery Manager).</p>	<p>6</p>
	<p><b>Практическая работа 36</b> Настройка расписания автоматического резервного копирования в MySQL с использованием CRON.</p> <p><b>Практическая работа 37</b> Проверка целостности и восстановления данных из резервной копии в PostgreSQL.</p> <p><b>Практическая работа 38</b> Выполнение дифференциального резервного копирования в Microsoft SQL Server.</p> <p><b>Практическая работа 39</b> Создание инкрементального резервного копирования в Oracle Database.</p> <p><b>Практическая работа 40</b> Разработка стратегии резервного копирования и восстановления для базы</p>	<p>6</p>

	данных предприятия.	
<b>Тема 1.5. Восстановление баз данных</b>	<p><b>Практическая работа 41</b> Восстановление базы данных MySQL из резервной копии, созданной с помощью mysqldump.</p> <p><b>Практическая работа 42</b> Восстановление PostgreSQL базы данных из дампа (pg_restore).</p> <p><b>Практическая работа 43</b> Восстановление базы данных Microsoft SQL Server из полной резервной копии с использованием SSMS.</p> <p><b>Практическая работа 44</b> Восстановление базы данных MongoDB из резервного архива.</p> <p><b>Практическая работа 45</b> Восстановление Oracle Database с использованием RMAN (Recovery Manager).</p>	8
	<p><b>Практическая работа 46</b> Восстановление данных из резервной копии MySQL с проверкой целостности данных.</p> <p><b>Практическая работа 47</b> Восстановление базы данных PostgreSQL на новый сервер с сохранением всех параметров.</p> <p><b>Практическая работа 48</b> Выполнение восстановления базы данных Microsoft SQL Server из дифференциальной резервной копии.</p> <p><b>Практическая работа 49</b> Настройка сценария аварийного восстановления базы данных MongoDB.</p> <p><b>Практическая работа 50</b> Разработка и тестирование сценария восстановления Oracle Database после сбоя.</p>	6
<b>Тема 1.6. Мониторинг событий, возникающих в процессе работы баз данных</b>	<p><b>Практическая работа 51</b> Настройка и использование утилиты MySQL Performance Schema для мониторинга работы базы данных.</p> <p><b>Практическая работа 52</b> Использование утилиты pg_stat_activity в PostgreSQL для отслеживания активных соединений и запросов.</p> <p><b>Практическая работа 53</b> Мониторинг событий в Microsoft SQL Server с помощью SQL Server Profiler.</p> <p><b>Практическая работа 54</b> Установка и настройка Prometheus для сбора метрик производительности базы данных MySQL.</p> <p><b>Практическая работа 55</b> Анализ журнала событий (log files) в Oracle Database для выявления ошибок и проблем.</p>	8
	<p><b>Практическая работа 56</b> Мониторинг запросов и идентификация "тяжёлых" операций в MySQL с использованием EXPLAIN.</p> <p><b>Практическая работа 57</b> Настройка алертинга (уведомлений) в PostgreSQL на основе событийных триггеров.</p> <p><b>Практическая работа 58</b> Анализ блокировок и ожиданий в Microsoft SQL Server с помощью DMVs (Dynamic Management Views).</p> <p><b>Практическая работа 59</b> Использование MongoDB Profiler для отслеживания производительности запросов.</p> <p><b>Практическая работа 60</b> Настройка и тестирование автоматизированного сбора метрик базы данных с</p>	6

	использованием Grafana.	
<b>Тема 1.7. Протоколирование событий, возникающих в процессе работы баз данных</b>	<b>Практическая работа 61</b> Настройка и анализ журнала ошибок (error log) в MySQL. <b>Практическая работа 62</b> Конфигурация и просмотр логов событий в PostgreSQL с использованием параметра logging_collector. <b>Практическая работа 63</b> Настройка протоколирования аудита в Microsoft SQL Server с использованием Extended Events. <b>Практическая работа 64</b> Включение и настройка логирования операций в MongoDB с использованием параметра profilingLevel. <b>Практическая работа 65</b> Настройка и просмотр журнала аудита (Audit Trail) в Oracle Database.	6
	<b>Практическая работа 66</b> Конфигурация параметров логирования запросов в MySQL (slow_query_log) и анализ записей. <b>Практическая работа 67</b> Создание и настройка собственного формата логов в PostgreSQL. <b>Практическая работа 68</b> Протоколирование событий доступа к данным в Microsoft SQL Server и анализ логов. <b>Практическая работа 69</b> Настройка ротации логов и очистки устаревших записей в MongoDB. <b>Практическая работа 70</b> Разработка политики протоколирования событий и настройка соответствующих параметров в Oracle Database.	6
		88

**Тема 1.1. Установка и настройка программного обеспечения для администрирования  
баз данных**

**Практическая работа 1 Установка СУБД MySQL и настройка службы на  
локальном сервере.**

**Цель работы:** выбрать «MySQL Installer for Windows» — инсталлятор, который позволяет установить сразу весь комплект инструментов.

**Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

**Ход практического занятия:**

• Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

Включать установку СУБД MySQL на локальный сервер и настройку службы MySQL. Для работы можно использовать официальный сайт MySQL, где доступны дистрибутивы для разных операционных систем. Рекомендуется выбрать «MySQL Installer for Windows» — инсталлятор, который позволяет установить сразу весь комплект инструментов. [metanit.comsecuritylab.ruserverspace.ru](http://metanit.comsecuritylab.ruserverspace.ru)

#### Установка

##### **Некоторые этапы установки:**

- **Загрузка дистрибутива** с официального сайта.
- **Запуск установщика** — дважды кликнуть по загруженному файлу, чтобы запустить мастер. При необходимости система может запросить права администратора.
- **Выбор типа установки** в мастере:
  - **Developer Default** (вариант по умолчанию) — устанавливает все необходимые компоненты для разработки.
  - **Server Only** — устанавливает только сервер и самые базовые инструменты.
  - **Custom** — позволяет самостоятельно выбрать нужные модули. Для начинающих чаще всего достаточно Developer Default, чтобы получить полный комплект (сервер, консоль, MySQL Workbench).
- **Проверка зависимостей** — установщик проверит наличие нужных компонентов, таких как Visual C++ Redistributable. Если какого-то из них не хватает, мастер предложит его доустановить.
- **Процесс установки** — нажать «Execute», чтобы начать установку. По завершении каждого шага мастер покажет статус и наличие ошибок или предупреждений.

[serverspace.rusecuritylab.ru](http://serverspace.rusecuritylab.ru)

#### Настройка

**В конце установки автоматически запускается мастер настройки (Configuration).** Он позволит выбрать тип конфигурации, задать порт (стандартный 3306), включить или отключить сетевой доступ и установить пароль для root-пользователя. Обычно при установке рекомендуют оставлять параметры по умолчанию, особенно на начальном этапе. Однако в разделе выбора порта можно указать другой номер порта, если уже используется 3306 для другого приложения. [securitylab.ru](http://securitylab.ru)

**После завершения установки и конфигурации MySQL сервер должен автоматически запуститься как служба Windows.** Убедиться в этом можно, выполнив следующие действия: [securitylab.ru](http://securitylab.ru)

- Зайти в «Службы» (Services) через меню «Пуск» или поисковую строку Windows.

- Найти службу с названием вроде «MySQL80» или «MySQL57» (зависит от версии).
- Убедиться, что её статус — «Работает» (Running).

[securitylab.ru](http://securitylab.ru)

**Также можно воспользоваться командной строкой:** открыть «Командную строку» (cmd) от имени администратора и ввести:

```
sc query MySQL80
```

. В выводе будет состояние запущенной службы. Если она остановлена, можно запустить вручную:

```
net start MySQL80
```

. [securitylab.ru](http://securitylab.ru)

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 2 Установка PostgreSQL и настройка параметров конфигурации (порт, логирование).**

***Цель работы*** включать установку PostgreSQL и настройку параметров конфигурации, включая порт и логирование. Для работы можно использовать установочные файлы для разных операционных систем и файл конфигурации postgresql.conf..

### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### ***Ход практического занятия:***

• Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

• Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.

- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

#### Установка

##### **Некоторые шаги для установки PostgreSQL:**

- Скачать инсталляционный файл, соответствующий системе (32-разрядной или 64-разрядной). Обычно это пакет, включающий сам сервер PostgreSQL и набор дополнительных инструментов.
- Запустить скачанный файл, нажать «Next» и следовать подсказкам установщика.
- При выборе компонентов убедиться, что отмечены PostgreSQL Server, pgAdmin (графическая консоль управления базами данных) и StackBuilder (по желанию, если планируется устанавливать дополнительные модули).
- Задать путь установки и указать пароль для суперпользователя (по умолчанию — пользователь postgres). Обязательно записать и сохранить пароль в надёжном месте.
- Выбрать порт для службы PostgreSQL — стандартное значение — 5432, но его можно изменить, если порт 5432 занят или по другим причинам безопасности.
- Подтвердить все настройки и дождаться окончания установки.

[securitylab.ru](http://securitylab.ru)

#### **Настройка параметров конфигурации**

##### **Пример изменения порта:**

- Открыть postgresql.conf в текстовом редакторе.
- Найти строку  
listen\_addresses  
и указать IP-адрес, по которому будет доступен сервер. По умолчанию используется localhost, но можно установить  
'\*'

для принятия соединений со всех адресов (учтите соображения безопасности).

- Перезапустить службу PostgreSQL, чтобы изменения вступили в силу.

[securitylab.ru](http://securitylab.ru)

##### **Настройка логирования:**

- **logging\_collector** — при включении (on) позволяет собирать логи в отдельные файлы.
- **log\_directory** — директория, куда будут сохраняться лог-файлы.
- **log\_filename** — шаблон имён файлов журналов.
- **log\_min\_error\_statement** — уровень ошибок (ERROR, WARNING и т. д.), которые должны попадать в журнал.

[securitylab.ru](http://securitylab.ru)

**Важно:** при возникновении ошибок в работе базы в первую очередь стоит проверить журналы, чтобы выяснить, нет ли проблем с подключением, правами доступа или неправильными SQL-запросами. [securitylab.ru](http://securitylab.ru)

#### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

**Практическая работа 3Установка Oracle Database и настройка окружения (переменные PATH, ORACLE\_HOME).**

***Цель работы*** изучить процесс установки Oracle Database и настроить переменные среды, связанные с работой с базой данных.

***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

***Теоретические и учебно-методические материалы по теме практической работы***

**Материалы для работы:** установочный дистрибутив Oracle Database, универсальный установщик Oracle (OUI), скрипты для настройки переменных окружения (oraenv, oraenv). [docs.oracle.com](https://docs.oracle.com/docs.oracle.com)

**Установка Oracle Database**

**Некоторые шаги для установки Oracle Database:**

- **Скачать установочный дистрибутив** с официального сайта Oracle.
- **Распаковать архив** и запустить программу установки (setup.exe).
- **Принять условия лицензионного соглашения.**
- **Указать каталог для установки** (можно оставить по умолчанию).
- **Задать пароль** для системных учётных записей: SYS, SYSTEM и PDBADMIN.

[info-comp.ru](http://info-comp.ru)

**Важно:** для установки Oracle Database необходимо обладать правами администратора. Если компьютер включён в домен, нужно войти в систему под локальным администратором. [trevog.net](http://trevog.net)

### **Настройка окружения**

#### **Некоторые особенности настройки окружения для работы с Oracle:**

- **ORACLE\_HOME** — задаёт путь к каталогу, где установлено программное обеспечение Oracle. Например, если базовый каталог Oracle установлен в C:\oracle, нужно задать значение:

ORACLE\_HOME=C:\oracle

. [testingdocs.comibm.com](http://testingdocs.comibm.com)

- **PATH** — необходимо настроить так, чтобы она включала каталог

ORACLE\_HOME/bin

. [docs.oracle.comtune-it.ru](http://docs.oracle.comtune-it.ru)

#### **Особенности настройки в зависимости от платформы:**

- **В Windows** универсальный установщик Oracle (OUI) автоматически присваивает значения ORACLE\_HOME и ORACLE\_SID в реестре Windows.

- **В UNIX и Linux** переменные среды нужно задать с помощью команд операционной системы.

- **При переключении между экземплярами** с разными Oracle homes необходимо изменить переменную среды ORACLE\_HOME. Если несколько экземпляров используют один и тот же Oracle home, то при переключении экземпляров необходимо изменить только ORACLE\_SID.

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 4 Установка MongoDB и настройка репликации для отказоустойчивости.**

***Цель работы*** создать группу серверов, которые вместе работают как единая база

данных, и настроить автоматический переход на другой ресурс при сбоях основного узла. .

#### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

#### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

#### **Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
  - Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
  - Подготовить ответы на контрольные вопросы.

#### **Теоретические и учебно-методические материалы по теме практической работы**

##### **Установка MongoDB**

Перед настройкой репликации необходимо: [serverspace.ru](http://serverspace.ru)

- **Установить MongoDB на все серверы.** Можно использовать установочные пакеты .msi или .exe, которые доступны на официальном сайте MongoDB. [serverspace.ru](http://serverspace.ru)[serverspace.rutimeweb.cloud](http://serverspace.rutimeweb.cloud)
- **Разрешить в правилах firewall подключение между серверами** для порта 27017. [serverspace.ru](http://serverspace.ru)
- **Настроить разрешение имён серверов:** если DNS-сервер отсутствует, добавить на каждом сервере записи в /etc/hosts. Имена должны быть одинаковыми на всех участниках. [serverspace.ru](http://serverspace.ru)[AdminVPS.ru](http://AdminVPS.ru)

Видео с инструкцией по установке MongoDB:

##### Установка MongoDB Windows 10

Rutube

##### **Настройка репликации**

##### **Некоторые шаги для настройки репликации в MongoDB:**

- **Настроить конфигурационный файл mongod.conf** на каждом узле. Найти или добавить блок replication и задать имя Replica Set (например, rs0). [AdminVPS.ru](http://AdminVPS.ru)
- **Запустить экземпляры MongoDB** на каждом сервере и убедиться, что они присоединяются к набору реплик, подключаясь друг к другу. [geeksforgeeks.org](http://geeksforgeeks.org)

- **Инициализировать набор реплик** в оболочке MongoDB: выполнить команду `rs.initiate()`  
— это назначит текущий узел в качестве основного. [geeksforgeeks.org](https://www.geeksforgeeks.org/)

#### **Особенности настройки:**

- **Нечётное число узлов** — это обеспечивает кворум во время выборов. Типичная конфигурация — один основной узел, два дополнительных узла и, при необходимости, один узел-арбитр.

- **Географическое распределение** — это расширяет возможности аварийного восстановления, так как данные останутся доступными, даже если в одном из центров обработки данных произойдёт сбой.

- **Распределение ресурсов** — нужно убедиться, что у каждого узла достаточно ресурсов (ЦП, памяти, дискового пространства) для выполнения рабочей нагрузки.

[openzeba.com](https://openzeba.com/)

**Проверка настройки:** после настройки кластера можно проверить, как он работает как кластер, а не просто набор независимых серверов. Например, выполнить команду

`rs.status()`

— вывод покажет список участников, их роли, приоритеты, состояние синхронизации и лаги. [AdminVPS.ru](https://www.adminvps.ru/)

#### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

##### **Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## **Практическая работа 5 Установка Microsoft SQL Server и настройка параметров аутентификации.**

**Цель работы** изучить процесс установки и выбрать модель безопасности (аутентификацию Windows или смешанный режим)..

#### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

#### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

#### ***Теоретические и учебно-методические материалы по теме практической работы***

##### **Установка**

##### **Некоторые шаги установки Microsoft SQL Server:**

- **Скачать дистрибутив** с официального сайта Microsoft. Можно выбрать нужную редакцию: Express, Developer или Evaluation. [softcomputers.org/infostart.ruhelp.sweb.ru](http://softcomputers.org/infostart.ruhelp.sweb.ru)
- **Запустить установщик** (файл setup). При старте установщик предлагает три варианта: «Базовая» — быстрая установка с настройками по умолчанию, «Пользовательский» — выбор компонентов и детальная конфигурация, «Скачать носитель» — загрузка установочных файлов для офлайн-развёртывания. [infostart.ruhelp.sweb.ru](http://infostart.ruhelp.sweb.ru)
- **Принять условия лицензии.** Выбрать язык интерфейса, ознакомиться с текстом соглашения и нажать «Принять», чтобы продолжить установку. [help.sweb.ru](http://help.sweb.ru)
- **Выбрать папку для установки SQL Server** (по умолчанию — путь C:\Program Files\Microsoft SQL Server). Можно указать другой диск или папку и нажать «Установить». [help.sweb.ru](http://help.sweb.ru)
- **После завершения установки** появится окно с уведомлением об успешной установке. [help.sweb.ru](http://help.sweb.ru)

Видео с инструкцией по установке Microsoft SQL Server:

##### Установка сервера Microsoft SQL Server 2019

Rutube

##### **Настройка параметров аутентификации**

**Можно выбрать модель аутентификации:** [selectel.rusqlservercentral.com](http://selectel.rusqlservercentral.com)

- **Смешанный режим** — сервер поддерживает оба типа аутентификации: SQL Server и Windows.
- **Режим Windows Only** — сервер поддерживает только Windows аутентификацию.

[sqlservercentral.com](http://sqlservercentral.com)

**Чтобы изменить режим аутентификации,** нужно: [itworkroom.com](http://itworkroom.com)

1. Зайти в систему от имени пользователя, учётная запись которого существует в SQL Server, и запустить SQL Management Studio.
2. Выбрать сервер и тип аутентификации.
3. После аутентификации войти в свойства сервера (properties).
4. На вкладке Security выбрать необходимый тип аутентификации.
5. Появится предупреждение о необходимости перезапуска служб SQL Server для применения изменений. Перезапустить сервер можно в оснастке SQL Management Studio, нажав правой кнопкой мыши и выбрав в меню «Restart».

[itworkroom.com](http://itworkroom.com)

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 6 Создание и управление пользователями в установленной СУБД (назначение ролей и прав доступа).**

***Цель работы*** создать нового пользователя через интерфейс администрирования

***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.

- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### ***Теоретические и учебно-методические материалы по теме практической работы***

Инструкция по созданию и управлению пользователями в СУБД, например в PostgreSQL или Microsoft SQL Server, с назначением ролей и прав доступа.

**В PostgreSQL** можно создать нового пользователя через интерфейс администрирования, например, в программе pgAdmin:

1. Открыть pgAdmin и подключиться к серверу PostgreSQL.
2. На левой панели навигации выбрать базу данных, в которой нужно создать пользователя.
3. Щёлкнуть правой кнопкой мыши на «Login/Group Roles».
4. Выбрать «Create», а затем — «Login/Group Role».
5. В появившемся окне заполнить следующую информацию:
  - В разделе General в поле Name ввести имя пользователя.
  - В разделе Definition в поле Password ввести пароль новой учётной записи. Здесь же можно настроить дополнительные параметры, например, дату истечения прав учётной записи.
  - В разделе Privileges выбрать необходимые привилегии. Например, включить опцию «Can login?», нажав на переключатель справа, чтобы разрешить пользователю вход в систему.
  - В разделе Membership можно назначить пользователю специфические роли.
  - Проверить сгенерированный SQL-код на крайней правой вкладке и затем нажать кнопку Save, чтобы создать пользователя с выбранными параметрами.

[help.sweb.ru](http://help.sweb.ru)

**В Microsoft SQL Server** организована двухуровневая настройка ограничения доступа к данным:

1. На первом уровне создать учётную запись пользователя (login), что позволяет ему подключиться к самому серверу, но не даёт автоматического доступа к базам данных.
2. На втором уровне для каждой базы данных MS SQL Server на основании учётной записи создать запись пользователя. На основе прав, выданных пользователю как пользователю базы данных (user), его регистрационное имя (login) получает доступ к соответствующей базе данных.

[multiurok.ru](http://multiurok.ru)

**Привилегии пользователей** назначаются им администратором базы данных и определяют, какие действия над данными и над объектами схемы являются разрешёнными. Существуют привилегии двух типов: системные, контролирующие общий доступ к базе данных, и объектные, контролирующие доступ к конкретным объектам базы данных. [intuit.ru](http://intuit.ru)

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

## ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 7 Настройка автоматического резервного копирования базы данных.**

***Цель работы*** изучить настройку автоматического резервного копирования базы данных в разных системах — MS SQL Server и программе «1С».

### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

## ***Теоретические и учебно-методические материалы по теме практической работы В MS SQL Server***

**Настройка с помощью SQL Server Management Studio (SSMS): [easeus.com](http://easeus.com)**

1. Запустить SSMS и подключиться к экземпляру SQL Server.
2. Открыть папку «Управление» и выбрать «Планы обслуживания».
3. Щёлкнуть по папке правой кнопкой мыши и выбрать «Новый план обслуживания».
4. В диалоговом окне ввести название плана.

5. На панели инструментов слева перетащить задачу резервного копирования базы данных в область конструктора справа. Дважды щёлкнуть задачу, чтобы отредактировать её: выбрать базы данных, указать расположение резервной копии и настроить другие параметры.

6. Перейти на вкладку «Расписание» и настроить расписание, по которому план обслуживания будет выполняться автоматически. Можно установить частоту, время и другие параметры.

[easeus.com](http://easeus.com)

Видео с инструкцией по настройке автоматического резервного копирования базы данных в MS SQL Server:

Настройка резервного копирования MS SQL Server

Rutube

В 1С

**Настройка автоматического резервного копирования в программе «1С»:** [smoff.rudzen.ru](http://smoff.rudzen.ru)

1. Зайти в меню «Администрирование» → «Обслуживание».
2. В пункте «Резервное копирование и восстановление» перейти в раздел «Настройка резервного копирования».
3. Поставить галочку «Выполнять автоматическое резервное копирование».
4. Настроить расписание резервного копирования: выбрать периодичность (ежедневно, еженедельно) и время запуска.

5. Указать место для резервного копирования (диск компьютера или внешний жёсткий диск).

6. Нажать «Готово».

[smoff.rupracticum.yandex.ru](http://smoff.rupracticum.yandex.ru)

**Важно:** автоматическое копирование требует регулярной проверки работоспособности: например, контроля наличия свободного места на диске и корректности созданных архивов. [practicum.yandex.ru](http://practicum.yandex.ru)

Видео с инструкцией по настройке автоматического резервного копирования базы данных в программе 1С:

Настройка автоматического резервного копирования базы 1С

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## Практическая работа 8 Настройка параметров производительности СУБД (буферизация, размер кэша, максимальные соединения).

**Цель работы** изучить настройку автоматического резервного копирования базы данных в разных системах — MS SQL Server и программе «1С».

### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### **Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### **Теоретические и учебно-методические материалы по теме практической работы**

#### **Буферизация**

**Параметр `shared_buffers`** указывает объём памяти, выделенной для кэша данных. **Рекомендация:** устанавливать значение около 25–40% от общего объёма оперативной памяти. [tproger.ru](http://tproger.ru)

#### **Пример настройки:**

- Для выделенного сервера с объёмом ОЗУ 1 ГБ и более разумное начальное значение — 25% от объёма памяти.
- В системах с объёмом ОЗУ меньше 1 ГБ стоит ограничиться меньшим процентом ОЗУ, чтобы оставить достаточно места операционной системе.

[postgrespro.ru](http://postgrespro.ru)

**Важно:** память под буфер выделяется при запуске сервера, и её объём при работе не изменяется. [opennet.ru](http://opennet.ru)

#### **Размер кэша**

**Параметр `effective_cache_size`** определяет представление планировщика об эффективном размере дискового кэша, доступного для одного запроса. **Рекомендация:** для OLTP-нагрузок — 75–80% от RAM. [dzen.ru](http://dzen.ru)

#### **Пример настройки:**

- Для конфигурации PostgreSQL 8 CPU, 8 ГБ RAM с shared\_buffers=2 ГБ под нагрузкой pgbench (экспоненциальный рост 5–100 соединений) оптимальное значение effective\_cache\_size — 6 ГБ. [dzen.ru](http://dzen.ru)

**Важно:** чрезмерное увеличение размера кэша обычно не даёт существенного эффекта. [infourok.ru](http://infourok.ru)

#### **Максимальные соединения**

**Параметр max\_connections** определяет максимальное число одновременных соединений с процессом СУБД. **Рекомендация:** для высоконагруженных систем с большим числом одновременных пользователей рекомендуется устанавливать значение больше, например, 4000. [docs.selectel.ru](http://docs.selectel.ru)

#### **Пример настройки:**

- В PostgreSQL по умолчанию установлено значение 100 соединений, можно изменить его в настройках СУБД на значение максимального количества соединений при нагрузке.

- 

#### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### **Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

### **Практическая работа 9 Обновление версии установленной СУБД с сохранением данных.**

**Цель работы** изучить настройку автоматического резервного копирования базы данных в разных системах — MS SQL Server и программе «1С».

#### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

#### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

#### **Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

Включать обновление версии установленной СУБД (например, PostgreSQL или SQL Server) с сохранением данных. Для этого используются специальные инструменты, которые создают системные таблицы заново, учитывая изменения последних версий, при этом старые файлы данных сохраняются. [timeweb.cloudosp.ru](http://timeweb.cloudosp.ru)[BlackArch.ru](http://BlackArch.ru)[postgrespro.ru](http://postgrespro.ru)

Ниже приведены примеры таких работ для PostgreSQL и SQL Server.

#### **PostgreSQL**

**Обновление с помощью программы pg\_upgrade.** Программа создаёт системные таблицы заново, учитывая изменения последних версий, и проверяет, что кластеры совместимы. **Некоторые особенности работы:** [timeweb.cloudpostgrespro.ru](http://timeweb.cloudpostgrespro.ru)

- Перед обновлением необходимо переименовать существующий каталог данных, чтобы избежать конфликтов имён. [timeweb.cloudBlackArch.ru](http://timeweb.cloudBlackArch.ru)
- Новый кластер баз данных необходимо инициализировать. [BlackArch.ru](http://BlackArch.ru)
- Если доступны обновления, pg\_upgrade автоматически создаёт скрипт для их установки. [help.sweb.ru](http://help.sweb.ru)

**Важно:** перед обновлением рекомендуется создать резервную копию базы данных, так как обновления могут повлиять на работу приложений. [yandex.cloud](http://yandex.cloud)

#### **SQL Server**

**Обновление с сохранением предыдущей версии.** Вместо того, чтобы напрямую заменять старый экземпляр SQL Server, данные базы данных и компонентов пересылаются из старого экземпляра в отдельный экземпляр новой версии. **Некоторые особенности:** [osp.ru](http://osp.ru)

- Можно пересылать данные и компоненты в экземпляр новой версии, размещённый на другом физическом сервере или на том же сервере.
- Придётся вручную переместить файлы данных и другие вспомогательные объекты из старого экземпляра SQL Server в экземпляр новой версии — программа установки SQL Server не выполнит эту задачу.

#### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**  
**Сформулировать выводы по результатам работы.**  
**Сдать и защитить работу.**

### **Практическая работа 10 Настройка удалённого доступа к установленной СУБД через конфигурационные файлы.**

**Цель работы** изучить настройку автоматического резервного копирования базы данных в разных системах — MS SQL Server и программе «1С».

**Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

**Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

**Теоретические и учебно-методические материалы по теме практической работы**

Включать настройку удалённого доступа к установленной СУБД через конфигурационные файлы на примере MySQL и PostgreSQL. Ниже приведены инструкции для обеих СУБД. [ZSComp.rutimeweb.cloudZSComp.ru](http://ZSComp.rutimeweb.cloudZSComp.ru)

**MySQL**

**По умолчанию сервер MySQL разрешает только локальные подключения** — с сервера, на котором он работает. Чтобы разрешить подключение извне, необходимо изменить конфигурационный файл **my.cnf**: [timeweb.cloudZSComp.ru](http://timeweb.cloudZSComp.ru)

- **Найти параметр bind-address** — по умолчанию он часто установлен в 127.0.0.1, что ограничивает подключение только с того же устройства. Чтобы разрешить подключение с другого сервера, нужно заменить этот адрес на 0.0.0.0 или указать конкретный внешний IP-адрес, с которого разрешено подключение.

- Если на сервере активен параметр **skip-networking**, его нужно закомментировать или удалить — иначе сетевые подключения к базе не будут приниматься.

- После изменений сохранить файл и перезапустить демон базы данных:

```
sudo systemctl restart mysql
```

**PostgreSQL**

По умолчанию PostgreSQL принимает подключения только с локального хоста.

Чтобы настроить удалённое подключение, нужно отредактировать конфигурационные файлы **pg\_hba.conf** и **postgresql.conf**: [1cloud.ruZSComp.ru](http://1cloud.ruZSComp.ru)

- Изменить файл **postgresql.conf** — найти параметр

```
listen_addresses
```

и заменить значение на '\*' или указать конкретный IP, с которого планируется подключение. Это позволит принимать обращения не только с локального интерфейса.

- Добавить строку в файл **pg\_hba.conf** — разрешить подключение с нужного IP-адреса или диапазона адресов. Например:

```
host your_database_name your_user_name 192.168.1.0/24 md5
```

.

- Перезапустить сервис PostgreSQL, чтобы изменения вступили в силу.

[ZSComp.ru](http://ZSComp.ru)

**Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

**Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## Тема 1.2. Установка и настройка программного обеспечения (ПО) для обеспечения работы пользователей с базами данных

### **Практическая работа 11 Установка и настройка клиента SQL Workbench для работы с базой данных MySQL.**

**Цель работы** изучить настройку автоматического резервного копирования базы данных в разных системах — MS SQL Server и программе «1С».

**Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### *Ход практического занятия:*

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

Включать установку и настройку клиента **MySQL Workbench** для работы с базой данных MySQL. Программа позволяет проектировать схемы баз данных, создавать и выполнять SQL-запросы, настраивать сервер и выполнять административные задачи. [securitylab.rugEEKSforgeeks.org/help.sweb.ru](http://securitylab.rugEEKSforgeeks.org/help.sweb.ru)

#### **Установка**

**MySQL Workbench** устанавливается вместе с MySQL Installer (если выбрать Developer Default) или скачивается отдельно с официального сайта MySQL. Некоторые шаги установки: [securitylab.ru](http://securitylab.ru)

- **Выбрать установщик** — например, «MySQL Installer for Windows».
- **Запустить установщик** — дважды кликнуть по загруженному файлу, чтобы запустить мастер. При необходимости система может запросить права администратора.
- **Выбрать тип установки** — в мастере есть варианты: Developer Default (по умолчанию), Server Only, Custom. Для начинающих чаще всего достаточно Developer Default, чтобы получить полный комплект (сервер, консоль, MySQL Workbench).
- **Проверить зависимости** — установщик проверит наличие нужных компонентов, если какого-то из них не хватает, мастер предложит его доустановить.
- **Нажать «Execute»**, чтобы начать установку. По завершении каждого шага мастер покажет статус и наличие ошибок или предупреждений.

[securitylab.ru](http://securitylab.ru)

#### **Настройка**

При первом запуске **MySQL Workbench** программа предложит настроить подключение к серверу. **Чтобы создать подключение**, нужно: [securitylab.ru](http://securitylab.ru)

- В окне «MySQL Connections» нажать «+» («Добавить новое соединение»).
- Ввести имя подключения (например, «Local MySQL»).
- Указать Hostname (обычно localhost), Port (3306, если не меняли), Username (root или ранее созданный пользователь).
- Если нужно, чтобы Workbench не спрашивал пароль каждый раз, нажать «Store in Vault...» и сохранить его.
- **Проверить соединение** — нажать «Test Connection». Если всё настроено верно, появится сообщение об успешном подключении.

[securitylab.ru](http://securitylab.ru)

**После подключения** в MySQL Workbench будет список баз данных. Можно нажать правой кнопкой мыши, чтобы создать новую схему или просматривать таблицы, просматривать данные и даже конструировать диаграммы связей. [securitylab.ru](http://securitylab.ru)

Видео с инструкцией по настройке программы MySQL Workbench для работы с базой данных MySQL:

Автоматическая настройка соединения в MySQL Workbench с сервером баз данны

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 12 Установка и настройка pgAdmin для управления PostgreSQL.**

***Цель работы*** изучить настройку автоматического резервного копирования базы данных в разных системах — MS SQL Server и программе «1С».

### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### ***Ход практического занятия:***

• Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

• Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.

- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

## *Теоретические и учебно-методические материалы по теме практической работы*

Включать установку и настройку утилиты **pgAdmin** для управления сервером PostgreSQL. pgAdmin — графический клиент для работы с сервером, через который можно создавать, удалять, изменять базы данных и управлять ими. [support.russianit.rumetanit.com](http://support.russianit.rumetanit.com)[tproger.ru](http://tproger.ru)

### **Установка**

На **Windows** pgAdmin входит в базовый комплект установки PostgreSQL. На **Linux** утилиту нужно установить отдельно. Некоторые шаги для установки: [metanit.com](http://metanit.com)

- **Скачать дистрибутив** pgAdmin с официального сайта. [info-comp.ru](http://info-comp.ru)
- **Запустить установку.** В мастере установки нужно выбрать директорию установки, отметить нужные компоненты (PostgreSQL Server, pgAdmin и другие). [info-comp.rutproger.ruself-learning.ru](http://info-comp.rutproger.ruself-learning.ru)
- **Указать данные** и задать пароль для суперпользователя (по умолчанию — postgres). [securitylab.rutproger.ru](http://securitylab.rutproger.ru)
- **Настроить порт** для сервера (по умолчанию — 5432, но его можно изменить, если порт 5432 занят или по другим причинам безопасности). [securitylab.rutproger.ru](http://securitylab.rutproger.ru)
- **Определить каталог** для хранения данных. [tproger.ru](http://tproger.ru)

Видео с инструкцией по установке утилиты pgAdmin для управления PostgreSQL:

Как установить и использовать программу pgAdmin — Видео от Postgres Professional  
Postgres Professional

### **Настройка**

После установки нужно открыть pgAdmin и подключиться к новому серверу. **Некоторые шаги начальной настройки:** [tproger.ru](http://tproger.ru)

- **Открыть pgAdmin**, введя пароль суперпользователя. После авторизации появится список баз данных, пользовательских ролей и других объектов.
- **Создать новую базу данных:** кликнуть правой кнопкой по «Databases», выбрать «Create», указать имя базы и при необходимости сменить владельца.
- **Открыть редактор запросов** и выполнить тестовый SQL-запрос, например: `CREATE TABLE test_table (id SERIAL PRIMARY KEY, description TEXT)`. Если таблица создавалась успешно, можно считать, что система функционирует корректно.

[securitylab.ru](http://securitylab.ru)

**Важно:** параметры настройки могут отличаться в зависимости от особенностей инфраструктуры и должны быть одобрены системным администратором. [support.russianit.ru](http://support.russianit.ru)

Видео с инструкцией по настройке утилиты pgAdmin для работы с сервером PostgreSQL:

### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### **Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и

кодированию.

**Ответить на контрольные вопросы.(устно)**  
**Сформулировать выводы по результатам работы.**  
**Сдать и защитить работу.**

### **Практическая работа 13 Установка и настройка Microsoft Management Studio (SSMS) для работы с SQL Server.**

**Цель работы** изучить, как установить SSMS и настроить её для работы с SQL Server.

**Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

**Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

**Теоретические и учебно-методические материалы по теме практической работы**

Включать установку и настройку **Microsoft Management Studio (SSMS)** — интегрированной среды от Microsoft для управления инфраструктурой SQL Server. SSMS объединяет графический интерфейс, редактор SQL-запросов и инструменты администрирования баз данных. [servercore.comgeeksforgEEKS.org](http://servercore.comgeeksforgEEKS.org)

**Цель работы** — [servercore.comlearn.microsoft.com](http://servercore.comlearn.microsoft.com)

**Установка**

**Скачать установщик SSMS** можно на официальном сайте Microsoft SQL Server. Перед установкой рекомендуется проверить: [servercore.com](http://servercore.com)

- системные требования для последней версии SSMS (по данным на май 2025 года — 21.1.3);
- последние обновления Windows;
- права администратора на компьютере.

[red9.comservercore.com](http://red9.comservercore.com)

После загрузки файла нужно запустить его, следовать инструкциям мастера установки и заполнить необходимые поля. Некоторые дополнительные параметры установки: [servercore.com](http://servercore.com)

- **Выбор рабочих нагрузок** — например, гибридная или миграционная рабочая нагрузка.
- **Указание модулей SSMS** — на вкладке «Отдельные компоненты» можно вручную указать, какие модули установить.
- **Изменение пути установки** — по умолчанию программа устанавливается на системный диск, но при необходимости можно указать другую папку.

[servercore.com](http://servercore.com)

По завершении установки SSMS появится в меню «Пуск». [servercore.com](http://servercore.com)

**Настройка**

При первом запуске SSMS откроется окно «Подключение к серверу». В нём нужно указать: [learn.microsoft.com](http://learn.microsoft.com)

- **Имя сервера** — имя SQL Server (можно использовать localhost, если подключение локально). Если не используется экземпляр по умолчанию (MSSQLSERVER), необходимо ввести имя сервера и имя экземпляра.
- **Аутентификацию** — можно выбрать проверку подлинности Windows или SQL Server.
- **Имя базы данных** (необязательно) — нужно ввести имя базы данных, к которой требуется подключиться (по умолчанию — master).
- **Дополнительные параметры подключения** — можно изменить значение времени ожидания подключения, намерение приложения и сетевой протокол.

[learn.microsoft.com](http://learn.microsoft.com)

После завершения всех полей нужно выбрать «Подключить». [learn.microsoft.com](http://learn.microsoft.com)

Чтобы убедиться, что подключение выполнено успешно, можно развернуть и изучить объекты в обозревателе объектов, где отображается имя сервера, версия SQL Server и имя пользователя. [learn.microsoft.com](http://learn.microsoft.com)

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 14 Установка и настройка DBeaver для подключения к различным типам баз данных.**

***Цель работы*** изучить, как создать новое соединение с базой данных в DBeaver, и настроить параметры подключения в зависимости от типа СУБД.

***Задачи:***

- Систематизировать подходы к изучению предмета.

- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

***Теоретические и учебно-методические материалы по теме практической работы***

Включать установку и настройку программы **DBeaver** для подключения к различным типам баз данных (СУБД). DBeaver поддерживает реляционные (MySQL, PostgreSQL, Oracle, SQL Server, SQLite и другие) и NoSQL-решения (MongoDB, Redis, Cassandra и др.). [support.servbay.com/practicum.yandex.rudbeaver.com](http://support.servbay.com/practicum.yandex.rudbeaver.com)

**Цель работы** — [gb.rupracticum.yandex.rudbeaver.com](http://gb.rupracticum.yandex.rudbeaver.com)

**Установка**

**DBeaver можно установить** на Windows, macOS и Linux. Дистрибутив приложения доступен на официальном сайте разработчиков. [practicum.yandex.ru](http://practicum.yandex.ru)

**Запустить DBeaver** можно через меню «Пуск» или ярлык на рабочем столе. В первый запуск нужно сразу создать тестовую базу для знакомства с доступными функциями. [practicum.yandex.ru](http://practicum.yandex.ru)

Видео с инструкцией по установке программы DBeaver:

02\_Установка DBeaver 7.2 и pgAdmin 4.23 на Windows 10

Rutube

**Настройка**

**Чтобы создать новое соединение с базой данных в DBeaver, нужно:** [practicum.yandex.ru](http://practicum.yandex.ru)

1. Запустить DBeaver.
2. Выбрать «Новое соединение» в меню «База данных» (находится в верхней панели инструментов).
3. Выбрать тип базы данных, с которой нужно создать соединение.
4. Указать параметры соединения — хост, порт, имя пользователя и пароль. Значения этих параметров зависят от типа базы данных. Можно добавить описание

соединения (это поле необязательно заполнять, но с описанием найти нужное соединение получится быстрее).

5. Нажать кнопку «Тест соединения» для проверки правильности ввода данных и работы соединения.

6. Нажать кнопку «Сохранить» для создания соединения.

[practicum.yandex.ru](http://practicum.yandex.ru)

**Созданное соединение** можно открыть, дважды щёлкнув по нему в списке соединений. [practicum.yandex.ru](http://practicum.yandex.ru)

**Важно:** количество и тип соединения зависят от драйвера БД. Например, встроенным драйверам (SQLite, Derby Embedded, HSQLDB, H2 Embedded) в отличие от внешних требуется только путь поиска БД. Более подробную информацию о настройках в зависимости от драйвера БД доступна в официальной документации каждого драйвера.

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 15 Установка и настройка библиотек Python для взаимодействия с базами данных (pymysql, psycopg2).**

***Цель работы*** изучить, как создать новое соединение с базой данных в DBeaver, и настроить параметры подключения в зависимости от типа СУБД.

### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### ***Ход практического занятия:***

• Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

• Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю

результаты работы.

- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

Включать установку и настройку библиотек Python для взаимодействия с базами данных (MySQL и PostgreSQL) —  **pymysql**  и  **psycopg2** . Эти библиотеки позволяют выполнять SQL-запросы, управлять соединениями, курсорами и транзакциями. [timeweb.cloudsky.prossl-team.com](http://timeweb.cloudsky.prossl-team.com)

Ниже приведены рекомендации по установке и настройке pymysql и psycopg2.

#### **pymysql**

##### **Установка:**

```
pip install pymysql
```

. Библиотека написана полностью на Python, не требует внешних зависимостей на C/C++. [proglib.iopythonlib.ru](http://proglib.iopythonlib.ru)

##### **Настройка:**

- **Подключение к базе данных:**

```
connection = pymysql.connect(host='localhost', port=3306, user='root', password='password', database='test_db')
```

- **Создание объекта cursor:**

```
cursor = connection.cursor()
```

. Он содержит методы для выполнения SQL-команд. Можно использовать контекстный менеджер

```
with
```

- **Выполнение запросов:**

```
cursor.execute("SELECT * FROM users")
```

.

[pythonlib.ruproglib.io](http://pythonlib.ruproglib.io)

#### **psycopg2**

##### **Установка:**

```
pip install psycopg2
```

. Библиотека — адаптер для PostgreSQL, обёртка для официальной клиентской библиотеки libpq. [timeweb.cloudPythonRu.compsycopg.org](http://timeweb.cloudPythonRu.compsycopg.org)

##### **Настройка:**

- **Подключение к базе данных:**

```
conn = psycopg2.connect(database="db_name", host="db_host", user="db_user", password="db_pass", port="db_port")
```

.

- **Создание объекта cursor:**

```
cursor = conn.cursor()
```

- **Выполнение запросов:**

```
cursor.execute("SELECT * FROM example_table")
```

.

• **Заккрытие подключения:**

```
conn.close()
```

.

[metanit.comdataquest.io](http://metanit.comdataquest.io)

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 16 Настройка соединения клиента SQL Workbench с сервером MySQL (локально и удалённо).**

***Цель работы*** изучить, как создать новое соединение с базой данных в DBeaver, и настроить параметры подключения в зависимости от типа СУБД.

#### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

#### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

#### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
  - Предъявить преподавателю результаты работы.
  - Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).

- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

Включать настройку соединения клиента MySQL Workbench с сервером MySQL — как локального, так и удалённого. Ниже приведены рекомендации для обоих случаев. [securitylab.ru/inmotionhosting.com/FirstVDS.ru](http://securitylab.ru/inmotionhosting.com/FirstVDS.ru)

#### **Локальное подключение**

**Чтобы настроить локальное подключение к серверу MySQL в MySQL Workbench, нужно:** [securitylab.ru](http://securitylab.ru)

1. Запустить программу. При первом запуске программа предложит настроить подключение к серверу.
2. В окне «MySQL Connections» нажать «+» («Добавить новое соединение»).
3. Ввести имя подключения (например, «Local MySQL»).
4. Указать **Hostname** (обычно localhost).
5. Ввести **Port** (по умолчанию 3306).
6. Указать **Username** (root или ранее созданный пользователь). Если нужно, чтобы Workbench не спрашивал пароль каждый раз, нажать «Store in Vault...» и сохранить его.
7. Проверить соединение — нажать «Test Connection». Если всё настроено верно, появится сообщение об успешном подключении.

[securitylab.ru](http://securitylab.ru)

#### **Удалённое подключение**

**Чтобы настроить удалённое подключение к серверу MySQL в MySQL Workbench, нужно:** [FirstVDS.ru](http://FirstVDS.ru)

1. При открытии программы выбрать **Database** → **Connect to Database**.
2. В окне настроек подключения указать **хост** (IP-адрес или домен).
3. Ввести **порт** (по умолчанию 3306).
4. Указать **имя пользователя** и **пароль**. Если не хранить пароль в MySQL Workbench, программа запросит его при подключении.

5. Опционально — указать **SSL-сертификаты** для шифрования соединения.

[FirstVDS.ru/8host.com](http://FirstVDS.ru/8host.com)

**Важно:** разрешение прямого подключения к MySQL извне может быть рискованным — рекомендуется использовать более безопасные альтернативы, например, SSH-туннелирование между приложением и базой данных. [FirstVDS.ru](http://FirstVDS.ru)

Видео с инструкцией по настройке удалённого подключения к серверу MySQL в программе SQL Workbench:

### *Оформление результатов работы*

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### *Содержание отчёта*

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**  
**Сформулировать выводы по результатам работы.**  
**Сдать и защитить работу.**

### **Практическая работа 17 Настройка пользователей и прав доступа через pgAdmin для PostgreSQL.**

**Цель работы** рассмотреть создание пользователя и назначение прав, а также использовать инструмент **Grant Wizard** для управления привилегиями объектов базы данных. .

#### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

#### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

#### **Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- 
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

#### **Теоретические и учебно-методические материалы по теме практической работы**

Включать настройку пользователей и прав доступа через графический интерфейс pgAdmin для PostgreSQL. В работе можно рассмотреть создание пользователя и назначение прав, а также использовать инструмент **Grant Wizard** для управления привилегиями объектов базы данных. [help.sweb.rudbserve.rucloud.rusupport.russianit.rupgadmin.org](http://help.sweb.rudbserve.rucloud.rusupport.russianit.rupgadmin.org)

#### **Создание пользователя**

**Алгоритм создания пользователя через pgAdmin: [dbserve.ru](http://dbserve.ru)**

1. Подключиться к базе данных.
2. Зайти в раздел **Login/Group Roles**, вызвать контекстное меню и выбрать **Create — Login/Group Role**.
3. В открывшемся окне:
  - **Во вкладке General** ввести имя пользователя и, по желанию, его описание в поле Comments.
  - **Во вкладке Definition** задать пароль и, по желанию, срок его действия.
  - **Во вкладке Privileges** с помощью тумблеров выдать пользователю необходимый набор прав: возможность логиниться, создавать другие роли и базы и т. д..
  - **Во вкладке Membership** присвоить пользователю нужные роли, например — дефолтную для PostgreSQL роль pg\_monitor, позволяющую просматривать разные таблицы и представления, связанные с мониторингом БД.
  - **Во вкладке SQL** привести SQL-команды, отвечающие за применение настроек, выполненных в предыдущих вкладках.
4. Нажать **Save**, после чего пользователь будет создан и доступен в разделе Login/Group Roles.

[dbserv.ru](http://dbserv.ru)

### Назначение прав

**Инструмент Grant Wizard** позволяет управлять привилегиями одного или нескольких объектов базы данных. **Чтобы запустить инструмент**, нужно выбрать объект базы данных в дереве pgAdmin и перейти в меню «Инструменты» — выбрать опцию **Grant Wizard**. [pgadmin.org](http://pgadmin.org)

**Процесс назначения прав:** [pgadmin.org](http://pgadmin.org)

1. В окне **Object Selection** выбрать объект или объекты, для которых будут назначены права. Можно использовать поле поиска по типу объекта или имени или прокрутку для просмотра списка всех доступных объектов.
2. В окне **Privileges Selection** назначить права. Для этого нужно:
  - Нажать значок «Добавить», чтобы назначить набор привилегий.
  - Выбрать имя роли из выпадающего списка в поле **Grantee**.
  - Нажать внутри поля **Privileges** и проверить флажки слева от одного или нескольких привилегий, чтобы назначить их указанному пользователю.
3. **Нажать кнопку «Next»**, чтобы продолжить, или «X» — закрыть инструмент без изменения привилегий.
4. **Нажать кнопку «Finish»**, чтобы сохранить настройки и выйти из инструмента.

### Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### Содержание отчёта

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

*Сформулировать выводы по результатам работы.  
Сдать и защитить работу.*

## **Практическая работа 18 Создание и выполнение SQL-запросов с использованием DBeaver.**

**Цель работы** рассмотреть создание пользователя и назначение прав, а также использовать инструмент **Grant Wizard** для управления привилегиями объектов базы данных. .

**Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

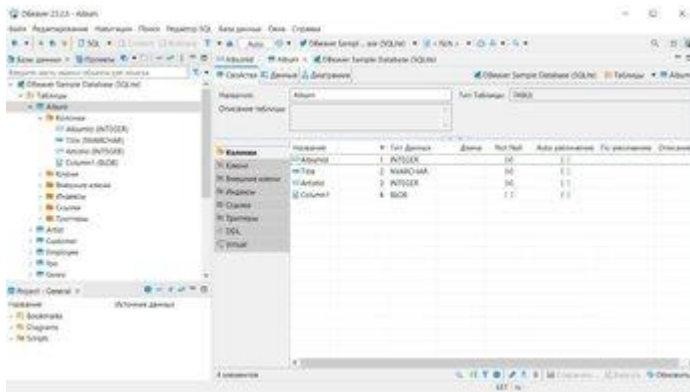
**Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

**Теоретические и учебно-методические материалы по теме практической работы**

Включать подключение к базе данных, написание SQL-запросов и их выполнение с помощью программы **DBeaver** — универсального клиента баз данных с открытым исходным кодом. DBeaver поддерживает более 80 различных СУБД, включая SQLite, MySQL, PostgreSQL, SQL Server, Oracle, MongoDB и другие. [forensicanvil.rukurshub.rublog.skillfactory.ru](http://forensicanvil.rukurshub.rublog.skillfactory.ru)

**Цель работы** — освоить интерфейс DBeaver, научиться работать с SQL-запросами и выполнять их. [forensicanvil.rukurshub.rublog.skillfactory.ru](http://forensicanvil.rukurshub.rublog.skillfactory.ru)



## Подключение

**DBeaver автоматически управляет драйверами JDBC для подключения к различным СУБД. Чтобы создать новое соединение, нужно:** [kurshub.ru](http://kurshub.ru)

1. Запустить DBeaver и в главном меню выбрать «База данных» → «Новое соединение».
2. В открывшемся мастере подключений выбрать нужную систему баз данных из списка (например, PostgreSQL, MySQL, Oracle).
3. Нажать «Далее» для перехода к настройке параметров соединения.

[kurshub.ru](http://kurshub.ru)

**Пример подключения к PostgreSQL:** указать хост (localhost или IP-адрес сервера), порт (5432), базу данных, имя пользователя и пароль для аутентификации. [kurshub.ru](http://kurshub.ru)

## Написание запросов

**В DBeaver есть встроенный редактор SQL. Открыть его можно несколькими способами:** [gb.rupracticum.yandex.rukurshub.ru](http://gb.rupracticum.yandex.rukurshub.ru)

- через контекстное меню — правый клик по соединению в навигаторе и выбор пункта «Редактор SQL»;
- на панели инструментов — кнопка «SQL-редактор» на верхней панели.

[kurshub.ru](http://kurshub.ru)

**Интерфейс редактора** разделён на две основные области: верхняя часть для написания запросов и нижняя — для отображения результатов выполнения. Дополнительные вкладки показывают планы выполнения, сообщения об ошибках и статистику производительности. [kurshub.ru](http://kurshub.ru)

**Система интеллектуальной поддержки кода:** контекстные подсказки (автодополнение имён таблиц, полей и функций на основе схемы базы данных), цветовая подсветка (различные цвета для ключевых слов SQL, строковых литералов, комментариев и идентификаторов), проверка синтаксиса (подчёркивание ошибок в реальном времени). [kurshub.ru](http://kurshub.ru)

## Выполнение

**Выполнение запросов в DBeaver** осуществляется через кнопку «Выполнить» или сочетание клавиш **Ctrl+Enter**. Для выполнения только выделенной части запроса — **Alt+X**. [kurshub.ru](http://kurshub.ru)

**Результат** выполненных запросов появится в соответствующем окне, расположенном изначально под окном редактора. Если в процессе выполнения возникли ошибки, они отобразятся в поле «Сообщения об ошибках». [gb.ru](http://gb.ru)

### *Оформление результатов работы*

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

## ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 19 Настройка интеграции баз данных с клиентским ПО через ODBC-драйверы.**

***Цель работы*** рассмотреть создание пользователя и назначение прав, а также использовать инструмент **Grant Wizard** для управления привилегиями объектов базы данных. .

***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями, студенты выполняют практическую работу.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

***Теоретические и учебно-методические материалы по теме практической работы***

Включать настройку интеграции баз данных с клиентским ПО через ODBC-драйверы, например, для работы с СУБД SQL Server или MySQL. Цель — обеспечить доступ к данным через ODBC-драйвер, который обрабатывает запросы специфично для конкретной СУБД. [infostart.rustorage.mstuca.ru](http://infostart.rustorage.mstuca.ru) [cved.rursdn.org/linter.ru](http://cved.rursdn.org/linter.ru)

**Теоретические основы**

- **ODBC (Open Database Connectivity)** — программный интерфейс для доступа к базам данных, использующий язык SQL. Приложения, использующие ODBC API, могут работать с различными источниками данных: реляционными, иерархическими и гетерогенными СУБД, файлами с данными. [infostart.rursdn.org/linter.ru](http://infostart.rursdn.org/linter.ru)

- **ODBC-драйверы** — для каждой СУБД используется собственный драйвер, который преобразует запросы от приложений в запросы на языке конкретной базы данных. [storage.mstuca.rulinter.ru](http://storage.mstuca.rulinter.ru)
- **Имя источника данных (Data Source Name, DSN)** — уникальный набор сведений, необходимых диспетчеру ODBC драйверов для подключения приложения к конкретной базе данных, поддерживающей ODBC. [storage.mstuca.ruintuit.ru](http://storage.mstuca.ruintuit.ru)

#### Пошаговая инструкция

- **Установить ODBC-драйвер** для СУБД. Например, для SQL Server — «ODBC Driver 17 for SQL Server». [storage.mstuca.ruhabr.comsky.pro](http://storage.mstuca.ruhabr.comsky.pro)
- **Настроить источник данных** с помощью утилиты **ODBC Data Source Administrator**. В Windows её можно найти на панели управления в разделе «Администрирование» или найти odbcad32 в меню «Пуск». [storage.mstuca.ruwiki.data-integration.ru](http://storage.mstuca.ruwiki.data-integration.ru)
- **Указать параметры подключения** для драйвера. Например, для подключения к SQL Server через ODBC необходимо задать: имя сервера (Server), имя базы данных (Database), пользователя (UID) и пароль (PWD). [habr.com1cved.ru](http://habr.com1cved.ru)
- **Протестировать подключение** — проверить, что соединение установлено успешно, с помощью простого запроса из клиентского приложения или инструмента командной строки. [wiki.data-integration.ru](http://wiki.data-integration.ru)

#### Возможные ошибки и их решение

- **Ошибка «Имя источника данных не найдено и не указан драйвер по умолчанию»** — проверить корректность настроек ODBC: убедиться, что имя DSN правильно указано в Администраторе источников данных ODBC.
- **Неправильная строка подключения** — исправить неверные параметры и проверить подключение.
- **Конфликты между драйверами** — проверить, соответствует ли версия драйвера в строке подключения установленной в системе.

#### *Оформление результатов работы*

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### *Содержание отчёта*

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

### **Практическая работа 20 Проверка совместимости клиентских приложений с установленным ПО для взаимодействия с базами данных.**

**Цель работы** убедиться, что программное обеспечение корректно работает в различных СУБД (например, MySQL, PostgreSQL, SQL Server).

**Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

***Ход практического занятия:***

• Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

• Руководствуясь методическими рекомендациями, студенты выполняют практическую работу.

- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

***Теоретические и учебно-методические материалы по теме практической работы***

• **Тестирование совместимости** — проверку, работает ли ПО должным образом в разных СУБД. Выделяют два типа тестирования:

○ **Обратная совместимость** — тестирование работы новых версий ПО со старым оборудованием или СУБД.

○ **Прямая совместимость** — установление совместимости ПО с существующими или будущими версиями СУБД.

[habr.comtquality.ru](http://habr.comtquality.ru)

**Методика**

Работа может включать:

• **Разработка планов тестирования** — создание четкой стратегии для проверки всех аспектов взаимодействия компонентов.

• **Определение приоритетных областей для проверки** — фокусировка на наиболее критичных участках системы.

• **Автоматизацию процессов** — использование инструментов автоматизации для повышения эффективности и скорости проведения тестов.

• **Документирование результатов** — подробная фиксация всех найденных ошибок и их последующее исправление.

[blog.skillfactory.ru](http://blog.skillfactory.ru)

**Инструменты**

Для проверки совместимости можно использовать:

• **Интеграционное тестирование** — проверку корректного взаимодействия между модулями приложения, валидации передачи данных, совместимости интерфейсов.

• **Тестовые фреймворки** — помогают отследить сетевые запросы, оценить корректность данных, передаваемых между frontend и backend.

- **Библиотеки для тестирования с реальными зависимостями** — например, Testcontainers, которые позволяют запускать тесты с базой данных того же типа, который используется для производственной среды.

[blog.skillfactory.ruhabr.com](http://blog.skillfactory.ruhabr.com)

#### **Примеры**

В практической работе можно провести:

- **Тестирование работы приложения на разных СУБД** — например, проверить, корректно ли оно работает на файловой версии СУБД и на одной или двух сторонних СУБД.

- **Проверку одновременного использования базы данных** — например, одновременного доступа и обновления записи.

- **Тестирование сложных запросов** — выполнение запросов на реальном количестве данных и при реальном состоянии статистики СУБД.

- **Оформление результатов работы**

- Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

- **Содержание отчёта**

- 1. Цель.

2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

- **Ответить на контрольные вопросы.(устно)**

- **Сформулировать выводы по результатам работы.**

- **Сдать и защитить работу.**

### **Тема 1.3. Управление доступом к базам данных**

#### **Практическая работа 21 Создание пользователей и групп в MySQL и назначение прав доступа (GRANT, REVOKE).**

**Цель работы** изучить, как управлять учётными записями пользователей и контролировать доступ к базам данных

#### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

#### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

#### **Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

- Руководствуясь методическими рекомендациями (инструкционная

карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.

- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

#### Создание пользователей

##### Команда

##### **CREATE USER**

позволяет создать нового пользователя в MySQL. Синтаксис:

```
CREATE USER 'username'@'host' IDENTIFIED BY 'password'
```

. [FirstVDS.ru](http://FirstVDS.ru)

##### Объяснение:

- username  
— имя создаваемого пользователя.
- host  
— хост, с которого пользователь может подключаться (localhost для локальных подключений или % для подключений с любых хостов). Также можно установить любой IP-адрес или доменное имя.

- password  
— пароль для нового пользователя.

. [FirstVDS.ru](http://FirstVDS.ru)

**Пример:** для создания пользователя user с возможностью локального подключения и заданным паролем password необходимо ввести:

```
CREATE USER 'user'@'localhost' IDENTIFIED BY 'password'
```

. [FirstVDS.ru](http://FirstVDS.ru)

Видео с инструкцией по созданию пользователя в MySQL:

MySQL создать пользователя для удаленного подключения  
Rutube

#### Назначение прав доступа

##### Команда

##### **GRANT**

позволяет назначить права (привилегии) для пользователя к базе данных или таблице.

Синтаксис:

```
GRANT privileges ON database.table TO 'username'@'host'
```

. [losst.pro](http://losst.pro)[FirstVDS.ru](http://FirstVDS.ru)

Если требуется предоставить привилегии для всех баз данных или таблиц, имена заменяют звёздочкой — «\*». [FirstVDS.ru](http://FirstVDS.ru)

**Пример:** чтобы выдать права на чтение, вставку, обновление и удаление данных в определённой таблице базы данных, нужно ввести:

```
GRANT SELECT, INSERT, UPDATE, DELETE ON user2db.user2table TO 'user2'@'localhost'
```

. [FirstVDS.ru](http://FirstVDS.ru)

**После изменения прав** необходимо обновить их:

```
FLUSH PRIVILEGES
```

. [fornex.com](http://fornex.com)

**Команда**

```
REVOKE
```

позволяет отозвать права у пользователя. Синтаксис:

```
REVOKE привилегии ON база_данных . таблица FROM 'имя_пользователя'@'хост'
```

. [losst.pro](http://losst.pro)

**Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

**Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## **Практическая работа 22 Настройка ролей и прав доступа в PostgreSQL для различных пользователей.**

**Цель работы** изучить, как управлять учётными записями пользователей и контролировать доступ к базам данных

**Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

**Ход практического занятия:**

• Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

• Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.

- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).

- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

Включать создание пользователей, назначение им привилегий и управление наследованием привилегий через группы ролей. Для этого можно использовать командную строку или графический интерфейс, например pgAdmin. [timeweb.cloudhelp.sweb.ruselectel.rudbserv.rutimeweb.cloud](http://timeweb.cloudhelp.sweb.ruselectel.rudbserv.rutimeweb.cloud)

#### Командная строка

##### Создание пользователей: [dbserv.ru](http://dbserv.ru)

- Для создания суперпользователя —  
CREATE USER name WITH PASSWORD 'pass' SUPERUSER
- .
- Для создания обычного пользователя —  
CREATE USER name WITH PASSWORD 'pass'
- .
- Для создания пользователя с дополнительными параметрами (например, лимит на число одновременных подключений к БД) —  
CREATE USER name WITH PASSWORD 'pass' CONNECTION LIMIT 5

[dbserv.rutimeweb.cloud](http://dbserv.rutimeweb.cloud)

##### Назначение прав: [timeweb.clouddbserve.ru](http://timeweb.clouddbserve.ru)

- Для подключения —  
GRANT CONNECT ON DATABASE database TO username
  - .
  - Для конкретных операций и таблиц — например,  
GRANT SELECT, UPDATE, INSERT ON ALL TABLES IN SCHEMA public TO 'username'
- разрешение на получение, обновление и добавление данных.

[dbserv.ru](http://dbserv.ru)

#### Графический интерфейс

##### Создание пользователя через pgAdmin: [dbserv.rutimeweb.cloud](http://dbserv.rutimeweb.cloud)

1. В pgAdmin подключиться к БД, зайти в раздел Login/Group Roles, вызвать контекстное меню и выбрать Create — Login/Group Role
- .
2. В открывшемся окне во вкладке General ввести имя пользователя и, по желанию, его описание в поле Comments
- .
3. Во вкладке Definition задать пароль и, по желанию, срок его действия.
4. Во вкладке Privileges выставить с помощью тумблеров необходимый набор прав: возможность логиниться, создавать другие роли и базы и т. д..
5. Во вкладке Membership присвоить пользователю нужные роли, например — дефолтную для PostgreSQL роль pg\_monitor

, позволяющую просматривать разные таблицы и представления, связанные с мониторингом БД.

6. Во вкладке

SQL

привести SQL-команды, отвечающие за применение настроек, выполненных в предыдущих вкладках.

7. Нажать

Save

, после чего пользователь будет создан и доступен в разделе

Login/Group Roles

.

[dbserv.rutimeweb.cloud](http://dbserv.rutimeweb.cloud)

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 23 Управление правами доступа в Microsoft SQL Server с использованием SQL Server Management Studio (SSMS).**

***Цель работы*** научиться назначать разрешения пользователям для объектов базы данных, используя SSMS, и учитывать баланс между безопасностью и разрешёнными действиями

### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю

результаты работы.

- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### ***Теоретические и учебно-методические материалы по теме практической работы***

Включать изучение теоретических основ, пошаговую инструкцию и рассмотрение ошибок. Цель — научиться назначать разрешения пользователям для объектов базы данных, используя SSMS, и учитывать баланс между безопасностью и разрешёнными действиями. [sky.promanage.accuwebhosting.combusinessstudio.ruosp.ru](http://sky.promanage.accuwebhosting.combusinessstudio.ruosp.ru)

#### **Теоретические основы**

В SQL Server организована двухуровневая настройка доступа: [multiurok.ru](http://multiurok.ru)

1. **Создание учётной записи пользователя (login)** — позволяет подключиться к серверу, но не даёт автоматического доступа к базам данных.
2. **Создание записи пользователя** для каждой базы данных на основании учётной записи. На основе прав, выданных пользователю как пользователю базы данных (user), его регистрационное имя (login) получает доступ к соответствующей базе данных.

[multiurok.ru](http://multiurok.ru)

Для управления доступом используются **роли** — они определяют уровень доступа к объектам SQL Server. Например: [multiurok.ru](http://multiurok.ru)

- **db\_datareader** — предоставляет право на чтение данных всех таблиц.
- **db\_datawriter** — даёт возможность изменять данные всех таблиц.
- **db\_owner** — делегирует все привилегии владельца базы данных.

[sky.pro](http://sky.pro)

#### **Пошаговая инструкция**

В SSMS управление доступом осуществляется через **графический интерфейс**. Например, чтобы назначить разрешения пользователю для конкретной таблицы, нужно: [sky.pro](http://sky.pro)

1. Кликнуть правой кнопкой мыши на пользователя.
2. Зайти в «Свойства» → «Сопоставление пользователей».
3. Установить нужные галочки напротив свойств.

[sky.pro](http://sky.pro)

**Важно:** перед предоставлением доступа для пользователя должен быть создан логин, а при назначении роли нужно действовать в контексте конкретной базы данных. [sky.pro](http://sky.pro)

#### **Ошибки и их решение**

Некоторые ошибки, которые могут возникать при управлении правами доступа в SSMS, и способы их решения:

- **Ошибка «Отказано в доступе»** — может быть связана с недостаточными правами у учётной записи SQL Server на файлы базы данных. Решение: запустить SSMS от имени администратора, проверить права учётной записи службы SQL Server и откорректировать настройки доступа в каталоге файлов базы данных.

- **Ошибка «Login failed for user sa»** — может быть связана с режимом аутентификации сервера (например, только Windows) — нужно изменить его на SQL Server. После изменения режима необходимо перезапустить службу SQL Server, чтобы изменения вступили в силу.

[sky.prodba.stackexchange.com](http://sky.prodba.stackexchange.com)

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 24 Настройка аутентификации и шифрования соединения в MySQL.**

***Цель работы*** обеспечить безопасность подключений, запретить нешифрованные соединения и настроить проверку подлинности сервера и имени хоста.

### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

***Теоретические и учебно-методические материалы по теме практической работы***

Работа может включать настройку на стороне сервера и клиента. [docs.oracle.com/docs.oracle.com](https://docs.oracle.com/docs.oracle.com)

#### **Настройка на стороне сервера**

- **Включить TLS** и запретить нешифрованные подключения. Для этого нужно:
  - Сгенерировать собственный центр сертификации (CA) и от него — серверный сертификат для mysqld. Если нет своего PKI, использовать корпоративный CA или приобрести публичные SSL-сертификаты.
  - Указать пути к ключам/сертификатам в конфигурации mysqld.
  - Ограничить версии TLS и при необходимости задать перечень шифров.
  - Включить требование защищённого транспорта глобально и/или на уровне пользователей.
- **Настроить проверку подлинности.** Клиент проверяет, что сертификат выпущен доверенным CA, и что имя хоста совпадает с полями Subject Alternative Name (SAN) сертификата. В современных клиентах MySQL это режим **VERIFY\_IDENTITY**.

[fastfox.pro/fastfox.pro](https://fastfox.pro/fastfox.pro)

#### **Настройка на стороне клиента**

- **Указать параметры для подключения с использованием шифрования.** По умолчанию клиентские программы MySQL пытаются установить зашифрованное соединение, если сервер поддерживает зашифрованные соединения. Можно использовать опцию `--ssl-mode`
  - `--ssl-mode=PREFERRED` — обеспечивает зашифрованное соединение, если другие настройки по умолчанию не изменились.
  - `--ssl-mode=REQUIRED` — клиентам требуется зашифрованное соединение, и они отказывают, если таковое не может быть установлено.
  - `--ssl-mode=VERIFY_CA` или `--ssl-mode=VERIFY_IDENTITY` — клиентам требуется зашифрованное соединение, а также выполняется проверка по сертификату CA сервера и (с помощью `VERIFY_IDENTITY`) по имени хоста сервера в его сертификате.

[docs.oracle.com/docs.oracle.com](https://docs.oracle.com/docs.oracle.com)

#### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

##### **Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## **Практическая работа 25 Использование встроенных ролей в Oracle Database для управления доступом.**

**Цель работы** обеспечить безопасность подключений, запретить нешифрованные соединения и настроить проверку подлинности сервера и имени хоста.

#### **Задачи:**

- Систематизировать подходы к изучению предмета.

- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

***Теоретические и учебно-методические материалы по теме практической работы***

- **Роли** — именованные группы привилегий, которые можно предоставлять пользователям с помощью команды GRANT. Можно создать заранее определённый набор ролей, с помощью которых пользователям предоставляются только те привилегии, которые им необходимы.
  - **Свойства ролей:**
    - динамичны, изменения роли автоматически затрагивают всех пользователей, которым была предоставлена эта роль;
    - роли могут быть включены или отключены во время сеанса;
    - можно защитить роли с помощью пароля, в этом случае только пользователи, знающие пароль роли, смогут включить роль.
  - Для администрирования привилегий и ролей используют инструкции DCL: GRANT — предоставить пользователям или ролям определённые привилегии или роли, REVOKE — отменить определённые привилегии или роли у пользователей или ролей.

[cyberleninka.rumlkv.ru](http://cyberleninka.rumlkv.ru)

**Примеры**

- **Создание роли** для группировки привилегий, например, для доступа к объектам конкретной схемы (таблицам, представлениям, процедурам). Можно назначить привилегии на роль, а затем предоставить её пользователю. [qna.habr.comsupport.dbagenesis.com](http://qna.habr.comsupport.dbagenesis.com)
- **Предоставление привилегий таблице на роль** с помощью оператора GRANT: GRANT privileges ON object TO role\_name  
 . Например, если нужно предоставить SELECT, INSERT, UPDATE и DELETE привилегии на таблицу suppliers к роли test\_role, нужно выполнить предложение GRANT: GRANT select, insert, update, delete ON suppliers TO test\_role

. [oracleplsql.ru](http://oracleplsql.ru)

• **Предоставление роли пользователю** с помощью оператора GRANT:

GRANT role\_name TO user\_name

. Например, если создана роль test\_role, её можно предоставить пользователю с именем jon:

GRANT test\_role TO jon

. [oracleplsql.ru](http://oracleplsql.ru)

**Ошибки**

• **Ошибка, что объект не найден (ORA-00942)** — возникает, если у разработчиков в коде не используется конструкция

SET ROLE my\_role IDENTIFIED BY password

. Решение — отключить аутентификацию у ролей, поскольку она фактически не использовалась в проекте.

• **Роль не удаляется при удалении схемы** — это может привести к неудаче при воссоздании схемы, так как роль уже существует. Решение — удалить роль перед созданием, чтобы не потерять доступ, который был предоставлен пользователям.

[habr.comstackoverflow.com](http://habr.comstackoverflow.com)

**Ресурсы**

• **Документация Oracle Database** — в разделе «Configuring Privilege and Role Authorization» описаны встроенные роли и привилегии, а также способы их назначения.

• **Статьи на тему «Управление доступом в СУБД Oracle»** — в них рассматриваются особенности организации управления доступом с помощью ролей.

[docs.oracle.comcyberleninka.ru](http://docs.oracle.comcyberleninka.ru)

***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

**Практическая работа 26 Конфигурация прав доступа для разных уровней пользователей (администратор, аналитик, пользователь) в PostgreSQL.**

***Цель работы*** обеспечить безопасность подключений, запретить нешифрованные соединения и настроить проверку подлинности сервера и имени хоста.

***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ

- Показать основные приемы эффективного использования возможностей
- ИИ

***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

***Теоретические и учебно-методические материалы по теме практической работы***

**В PostgreSQL права доступа для разных уровней пользователей (администратора, аналитика, обычного пользователя) настраиваются с помощью команд CREATE USER, GRANT и ALTER USER.** Права назначаются на различных уровнях — на уровне кластера/базы данных, таблиц, схем и объектов. [help.sweb.rudbserve.rucloud.ru](http://help.sweb.rudbserve.rucloud.ru)

Ниже приведены примеры настройки прав для разных уровней пользователей.

**Администратор**

Для администратора (суперпользователя) права назначаются с помощью команды **CREATE USER имя WITH PASSWORD 'пароль' SUPERUSER.** Это позволяет полностью управлять базой данных (БД). [help.sweb.rudbserve.ru](http://help.sweb.rudbserve.ru)

**Некоторые особенности настройки:**

- Можно назначить права для конкретных операций и таблиц, например, для получения, обновления и добавления данных —

**GRANT SELECT, UPDATE, INSERT ON ALL TABLES IN SCHEMA public TO username**  
.

- Можно использовать роль **SUPERUSER**, которая может обходить все проверки доступа, — для этого нужно добавить её в команду **CREATE USER**.

[dbserve.rucloud.ru](http://dbserve.rucloud.ru)

**Аналитик**

Для пользователя-аналитика (например, аналитика, который имеет доступ в систему, но не может создавать базы данных) права назначаются с помощью команды **CREATE USER analytics\_reader WITH PASSWORD 'ReaderSafePass123' LOGIN.** [cloud.ru](http://cloud.ru)

**Некоторые особенности настройки:**

- Можно дать право на вход (если нужно было NOLOGIN только временно) — **ALTER USER analytics\_reader LOGIN**

.

- Можно предоставить право только на чтение в конкретной базе данных —

```
GRANT CONNECT ON DATABASE data_warehouse TO analytics_reader
```

.  
[cloud.ru](http://cloud.ru)

#### **Пользователь**

Для **обычного пользователя** права назначаются с помощью команды **CREATE USER имя WITH PASSWORD 'пароль'**. Можно добавить дополнительные параметры, например, ограничения на подключение или права доступа к БД. [help.sweb.rudbserver.ru](http://help.sweb.rudbserver.ru)

#### **Некоторые особенности настройки:**

- Можно назначить права для работы с конкретной таблицей —  
GRANT ALL PRIVILEGES ON table\_users TO username  
. [timeweb.cloudmosk.ru](http://timeweb.cloudmosk.ru)
- При необходимости можно выдать пользователю новые права или изменить существующие с помощью команды **ALTER USER**. [cloud.ru](http://cloud.ru)

**Важно:** для обычных пользователей не используется роль **SUPERUSER** — для них применяются отдельные GRANT для нужных объектов. [cloud.ru](http://cloud.ru)

Видео с инструкцией по настройке прав доступа для обычного пользователя в PostgreSQL:

#### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

**Практическая работа 27 Создание политики безопасности в Microsoft SQL Server для ограничения действий пользователей.**

***Цель работы*** обеспечить безопасность подключений, запретить нешифрованные соединения и настроить проверку подлинности сервера и имени хоста.

#### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

#### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### *Ход практического занятия:*

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

В SQL Server для управления доступом пользователей к данным используется **ролевая модель**. Правила доступа описываются в виде троек: <субъект, объект, тип доступа>. Субъектами доступа могут быть учётные записи серверного уровня, пользователи баз данных, серверные роли и роли баз данных. [naukovedenie.ru](http://naukovedenie.ru)

Для ограничения доступа к данным RLS позволяет:

- **Определять предикат безопасности** — функцию, которая ограничивает строки на основе логики фильтрации.
- **Использовать предикаты разных типов:** FILTER (автоматически фильтруют строки, доступные для операций чтения) или BLOCK (явно блокируют операции записи, которые нарушают функцию предиката).

[newtechaudit.rulearn.microsoft.com](http://newtechaudit.rulearn.microsoft.com)

#### **Пошаговая инструкция**

**Создание политики безопасности** выполняется с помощью оператора

**CREATE SECURITY POLICY**

в Transact-SQL. Некоторые параметры, которые нужно указать: [learn.microsoft.com](http://learn.microsoft.com)[learn.microsoft.com](http://learn.microsoft.com)

- **Имя политики** — должно быть уникальным в пределах базы данных и схемы. [learn.microsoft.com](http://learn.microsoft.com)[github.com](http://github.com)

- **Имя схемы**, к которой принадлежит политика безопасности. [learn.microsoft.com](http://learn.microsoft.com)[github.com](http://github.com)

- **Целевую таблицу**, к которой будет применяться предикат безопасности. [learn.microsoft.com](http://learn.microsoft.com)[learn.microsoft.com](http://learn.microsoft.com)

- **Конкретную операцию DML**, для которой применяется предикат блокировки (например, INSERT или UPDATE). [learn.microsoft.com](http://learn.microsoft.com)

- **Параметр**

**STATE = { ON | OFF }**

— включает или отключает политику безопасности от применения предикатов к целевым таблицам. [learn.microsoft.com](http://learn.microsoft.com)

**Важно:** для одной таблицы для конкретной операции DML могут быть предназначены несколько отключённых политик безопасности, но только одна может быть включена в любой момент времени. [learn.microsoft.com](http://learn.microsoft.com)

#### **Ошибки и их решение**

**Некоторые ошибки, которые могут возникнуть при создании политики безопасности, и способы их решения:**

- **Ошибка «Неизвестный тип объекта „SECURITY“»** — политика безопасности доступна только в SQL Server 2016 (13.x) и более поздних версиях. В SQL Server 2014 SP3 оператор

```
CREATE SECURITY POLICY
```

не работает.

- **Слишком сложные предикаты** — ухудшают производительность базы данных, поскольку предикат будет проверяться каждый раз при выполнении доступа к данным. В этом случае нужно упростить предикаты.

[stackoverflow.com](http://stackoverflow.com)[newtechaudit.ru](http://newtechaudit.ru)

#### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

##### **Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## **Практическая работа 28 Проверка и настройка доступа к базе данных через файл конфигурации в MySQL.**

**Цель работы** изолировать пользователя от прямого доступа к данным и гарантировать применение установленных правил безопасности.

#### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

#### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### *Ход практического занятия:*

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

В SQL Server для управления доступом пользователей к данным используется **ролевая модель**. Правила доступа описываются в виде троек: <субъект, объект, тип доступа>. Субъектами доступа могут быть учётные записи серверного уровня, пользователи баз данных, серверные роли и роли баз данных. [naukovedenie.ru](http://naukovedenie.ru)

Для ограничения доступа к данным RLS позволяет:

- **Определять предикат безопасности** — функцию, которая ограничивает строки на основе логики фильтрации.
- **Использовать предикаты разных типов:** FILTER (автоматически фильтруют строки, доступные для операций чтения) или BLOCK (явно блокируют операции записи, которые нарушают функцию предиката).  
[newtechaudit.rulearn.microsoft.com](http://newtechaudit.rulearn.microsoft.com)

#### **Пошаговая инструкция**

**Создание политики безопасности** выполняется с помощью оператора CREATE SECURITY POLICY

в Transact-SQL. Некоторые параметры, которые нужно указать: [learn.microsoft.com](http://learn.microsoft.com)[learn.microsoft.com](http://learn.microsoft.com)

- **Имя политики** — должно быть уникальным в пределах базы данных и схемы. [learn.microsoft.com](http://learn.microsoft.com)[github.com](http://github.com)
- **Имя схемы**, к которой принадлежит политика безопасности. [learn.microsoft.com](http://learn.microsoft.com)[github.com](http://github.com)
- **Целевую таблицу**, к которой будет применяться предикат безопасности. [learn.microsoft.com](http://learn.microsoft.com)[learn.microsoft.com](http://learn.microsoft.com)
- **Конкретную операцию DML**, для которой применяется предикат блокировки (например, INSERT или UPDATE). [learn.microsoft.com](http://learn.microsoft.com)
- **Параметр**  
**STATE = { ON | OFF }**

— включает или отключает политику безопасности от применения предикатов к целевым таблицам. [learn.microsoft.com](http://learn.microsoft.com)

**Важно:** для одной таблицы для конкретной операции DML могут быть предназначены несколько отключённых политик безопасности, но только одна может быть включена в любой момент времени. [learn.microsoft.com](http://learn.microsoft.com)

## Ошибки и их решение

Некоторые ошибки, которые могут возникнуть при создании политики безопасности, и способы их решения:

- **Ошибка «Неизвестный тип объекта „SECURITY“»** — политика безопасности доступна только в SQL Server 2016 (13.x) и более поздних версиях. В SQL Server 2014 SP3 оператор

CREATE SECURITY POLICY

не работает.

- **Слишком сложные предикаты** — ухудшают производительность базы данных, поскольку предикат будет проверяться каждый раз при выполнении доступа к данным. В этом случае нужно упростить предикаты.

[stackoverflow.com/newtechaudit.ru](http://stackoverflow.com/newtechaudit.ru)

### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### **Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## **Практическая работа 29 Реализация сценария управления доступом через роли и группы пользователей в Oracle Database.**

**Цель работы** упростить управление привилегиями, например, предоставить права пользователям, исходя из их рабочих обязанностей

### **.Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### **Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю

результаты работы.

- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### ***Теоретические и учебно-методические материалы по теме практической работы***

Для реализации сценария можно использовать **механизм ролей** — набор заранее определённых привилегий, которые потом можно предоставить пользователю через GRANT. Также можно использовать **группы пользователей** — они предоставляют права, исходя из потребностей в доступе к данным. [cleverence.rudesktop.arcgis.com](http://cleverence.rudesktop.arcgis.com)

#### **Роли**

**Роли** — это логическая группировка прав. Вместо того чтобы вручную назначать каждому пользователю десятки привилегий, создаётся роль, например `read_only` или `developer`. Добавляя пользователя в роль, предоставляется ему полный набор соответствующих возможностей. [cleverence.ru](http://cleverence.ru)

#### **Некоторые особенности управления доступом через роли:**

- Роль может содержать как набор прав, так и другие роли.
- Есть предустановленные роли, например DBA, которая предоставляет системные права с возможностью делегирования другим пользователям.
- Можно включить или отключить роль для текущей сессии с помощью оператора SET ROLE.

[sqlsplus.comoracleplsql.ru](http://sqlsplus.comoracleplsql.ru)

#### **Группы пользователей**

**Группы пользователей** предоставляют права, исходя из потребностей в доступе к данным. Например, можно создать группы: те, кто просматривает данные, те, кто редактирует данные, и те, кто создаёт данные. [desktop.arcgis.com](http://desktop.arcgis.com)

#### **Некоторые особенности управления доступом через группы:**

- Предоставление прав группе не препятствует предоставлению прав отдельным пользователям. Например, можно предоставить минимальные права, необходимые для создания данных в базе, соответствующей группе, а затем присвоить администратору дополнительные права.
- Есть предустановленные группы, например PUBLIC, которая приравнивается к любому пользователю, подключившемуся к базе данных. Однако по соображениям безопасности предоставление привилегий для PUBLIC следует использовать только в случае крайней необходимости.

[desktop.arcgis.com](http://desktop.arcgis.com)

#### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**  
**Сформулировать выводы по результатам работы.**  
**Сдать и защитить работу.**

### **Практическая работа 30 Аудит действий пользователей в базе данных с помощью встроенных инструментов PostgreSQL.**

**Цель работы** — изучить, как отслеживать активность пользователей в базе данных, фиксировать события, связанные с безопасностью, и анализировать активность.

**.Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

**Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

**Теоретические и учебно-методические материалы по теме практической работы**

Включать анализ аудита действий пользователей в базе данных PostgreSQL с помощью встроенных инструментов и расширений. Для аудита используются SQL-запросы, представления, настройки и, в некоторых случаях, внешние инструменты. [labex.ioforum.hpc.namebytebase.com](http://labex.ioforum.hpc.namebytebase.com)

**Цель работы** — изучить, как отслеживать активность пользователей в базе данных, фиксировать события, связанные с безопасностью, и анализировать активность. [labex.ioforum.hpc.namesupabase.com](http://labex.ioforum.hpc.namesupabase.com)

**SQL-запросы**

- **Выявление активных пользователей** — запрос  
`SELECT username FROM pg_stat_activity WHERE state = 'active'`

. Можно узнать не только имена, но и адреса и порты активных пользователей:

```
SELECT username, client_addr, client_port FROM pg_stat_activity WHERE state = 'active'
```

.

• **Получение информации о состоянии пользователя и его текущем запросе** — запрос

```
SELECT username, state, query FROM pg_stat_activity WHERE state IN ('active', 'idle in transaction')
```

.

[sky.pro](http://sky.pro)

### Представления

• **pg\_stat\_activity** — показывает активные сессии и информацию о текущих запросах.

• **pg\_stat\_user\_tables** — собирает статистику по операциям (SELECT, INSERT, UPDATE, DELETE) для таблиц, помогает отслеживать частоту доступа к данным.

[forum.hpc.name](http://forum.hpc.name)

### Настройки

• **Конфигурация логирования** (в файле

postgresql.conf

) — позволяет включить запись команд, ошибок и подключений. Параметры:

○ log\_statement

— уровень логирования (может быть

all

,

ddl

,

mod

или

none

).

○ log\_connections

и

log\_disconnections

— запись подключений и отключений.

[forum.hpc.name](http://forum.hpc.name)

### Инструменты

• **Расширение pgAudit** — позволяет записывать команды, выполненные пользователями, в том числе SELECT, INSERT, UPDATE, DELETE, и обеспечивает контроль за доступом к данным. Можно настроить аудит событий: команд DDL (создание, изменение, удаление объектов), команд DML (вставка, обновление, удаление, выборка данных) и других.

• **Расширение pg\_proaudit** — позволяет регистрировать различные события, связанные с безопасностью, например, команды DDL и DML, команды управления доступом (GRANT, REVOKE). Журнал событий безопасности сохраняется отдельно от основного журнала сервера.

[forum.hpc.namedatafinder.ru](http://forum.hpc.namedatafinder.ru)

**Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

**Тема 1.4. Резервное копирование баз данных**

**Практическая работа 31 Создание резервной копии базы данных MySQL с использованием утилиты `mysqldump`.**

***Цель работы*** —попрактиковаться в создании резервной копии одной, нескольких баз данных или всех баз на сервере. Также можно создать резервную копию отдельной таблицы.

***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
  - Предъявить преподавателю результаты работы.
  - Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
  - Подготовить ответы на контрольные вопросы.

***Теоретические и учебно-методические материалы по теме практической работы***

Включать создание резервной копии базы данных MySQL с помощью утилиты **`mysqldump`**. Утилита создаёт текстовый дамп базы данных в виде SQL-скрипта с

командами CREATE, INSERT, DROP, LOCK, что позволяет воссоздать структуру и данные базы на другом сервере. [labex.iostatuser.cloudhelp.reg.ru](http://labex.iostatuser.cloudhelp.reg.ru)

### Синтаксис

**Общий синтаксис команды для создания резервной копии:**

```
mysqldump [опции] [имя_базы] > [путь_и_имя_файла].sql
```

. [help.reg.ru](http://help.reg.ru)

**Некоторые опции:**

- **-u** — имя пользователя для подключения к серверу MySQL.
- **-p** — запрашивает пароль от учётной записи указанного пользователя.
- **-database\_name** — название базы данных, которую нужно экспортировать.
- **-dump\_file.sql** — файл с расширением \*.sql, в который нужно сохранить

копию базы данных.

[blog.sedicomm.comacronis-backup.kz](http://blog.sedicomm.comacronis-backup.kz)

### Пример

**Для создания резервной копии одной базы данных:**

```
mysqldump -u username -p database_name > backup.sql
```

. [stater.cloud](http://stater.cloud)

**Для сохранения копии нескольких баз:**

```
mysqldump -u username -p --databases db1 db2 db3 > multi_backup.sql
```

. [stater.cloud](http://stater.cloud)

**Для создания резервной копии всех баз на сервере:**

```
mysqldump -u username -p --all-databases > full_backup.sql
```

. [stater.cloud](http://stater.cloud)

**Для создания резервной копии отдельной таблицы** — нужно добавить имя таблицы после названия базы данных:

```
mysqldump -u root -p db_name table_name > /db_table_backup.sql
```

. [help.reg.ru](http://help.reg.ru)

### Ошибки

**Некоторые ошибки, которые могут возникнуть при использовании mysqldump:**

- **Ошибка «Permission denied»** — может возникать, если файл резервной копии не имеет прав на запись. В этом случае нужно проверить права на файл и каталог, куда записывается дамп. [dev.tolinux.org.ru](http://dev.tolinux.org.ru)

- **Ошибка «MySQL server has gone away»** — как правило, причина — низкое значение параметра

`max_allowed_packet`

, который отвечает за ограничение выполнения команд из файла. Решение — увеличить значение параметра в конфигурационном файле MySQL.

### *Оформление результатов работы*

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

### *Содержание отчёта*

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**  
**Сформулировать выводы по результатам работы.**  
**Сдать и защитить работу.**

### **Практическая работа 32 Резервное копирование базы данных PostgreSQL с помощью pg\_dump и pg\_dumpall.**

**Цель работы** —попрактиковаться в создании резервной копии одной, нескольких баз данных или всех баз на сервере. Также можно создать резервную копию отдельной таблицы.

**.Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

**Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - • Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

**Теоретические и учебно-методические материалы по теме практической работы**

Для резервного копирования базы данных PostgreSQL используются утилиты **pg\_dump** и **pg\_dumpall**. Они решают разные задачи: **pg\_dump** создаёт резервную копию одной базы данных, а **pg\_dumpall** — всего кластера. [habr.com](http://habr.com)[beget.com](http://beget.com)[nic.rudocs.arenadata.io](http://nic.rudocs.arenadata.io)

**pg\_dump**

Создаёт резервную копию одной базы данных. Некоторые особенности: [docs.arenadata.ionic.ru](http://docs.arenadata.ionic.ru)

**• Синтаксис:**

`pg_dump <имя_базы_данных> > <имя_файла>`

.

- **Формат дампа:** по умолчанию создаётся SQL-файл, но `pg_dump` также поддерживает сжатый формат (-Fc).

- **Если нужно скопировать базу данных частично** (например, только структуру без данных), используют флаг

`--schema-only`

- **Для обратной задачи** — экспортировать только данные — подходит

`--data-only`

[docs.arenadata.ioslurm.io](https://docs.arenadata.ioslurm.io)

**Важно:** утилита `pg_dump` создаёт резервную копию только одной конкретной базы данных. Если в системе существуют глобальные объекты (например, роли или таблицы), которые принадлежат всей системе, а не конкретной базе данных, для их сохранения и восстановления необходимо использовать `pg_dumpall`. [nic.ru](https://nic.ru)

Видео с инструкцией по резервному копированию базы данных PostgreSQL с помощью утилиты `pg_dump`:

Как сделать бэкап баз 1С в Postgres с помощью `pg_dump` — инструкция по резервному копированию

Neosystemy Severo-Zapad LTD

`pg_dumpall`

**Создаёт резервную копию всего кластера.** Некоторые особенности: [docs.arenadata.ionic.ru](https://docs.arenadata.ionic.ru)

- **Выгружает глобальные объекты**, общие для всех баз данных, такие как роли и табличные пространства.

- **Для формирования файла резервной копии** `pg_dumpall` вызывает `pg_dump` для каждой базы данных внутри одного инстанса PostgreSQL.

- **Чтобы выгрузить только глобальные данные кластера**, используют опцию `pg_dumpall --globals-only`

.

[beget.com/docs.arenadata.io](https://beget.com/docs.arenadata.io)

**Важно:** утилита `pg_dumpall` может создать резервную копию только в формате SQL-скрипта. Для запуска `pg_dumpall` чаще всего необходимы права суперпользователя, так же как и для восстановления из созданной копии.

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## Практическая работа 33 Настройка и выполнение резервного копирования в Microsoft SQL Server с использованием SSMS.

**Цель работы** — изучить, как создавать резервные копии баз данных, настраивать регулярное резервное копирование с помощью планов обслуживания и решать ошибки при выполнении операций

### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### **Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы. Предъявить преподавателю результаты работы.
  - Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
  - Подготовить ответы на контрольные вопросы.

### **Теоретические и учебно-методические материалы по теме практической работы**

#### **Пошаговая инструкция**

#### **Настройка резервного копирования: [datanumen.com](http://datanumen.com)**

1. Запустить SSMS и подключиться к экземпляру SQL Server. [easeus.com](http://easeus.com)
2. В обозревателе объектов развернуть узел «Базы данных». [learn.microsoft.com/datanumen.com](http://learn.microsoft.com/datanumen.com)
3. Щёлкнуть правой кнопкой мыши по базе данных, для которой нужно создать резервную копию, и выбрать «Создать резервную копию...». [learn.microsoft.com/datanumen.com](http://learn.microsoft.com/datanumen.com)
4. В диалоговом окне резервного копирования выбрать тип резервной копии (по умолчанию — Full). [learn.microsoft.com](http://learn.microsoft.com)
5. В разделе «Компонент резервного копирования» выбрать «База данных». [learn.microsoft.com](http://learn.microsoft.com)
6. В разделе «Назначение» проверить расположение для файла резервной копии (по умолчанию — в папке ../mssql/data). Можно выбрать другое устройство или перераспределить данные между несколькими файлами. [learn.microsoft.com](http://learn.microsoft.com)
7. Настроить дополнительные параметры: сжатие резервной копии, шифрование, проверку целостности резервной копии после завершения. [datanumen.com](http://datanumen.com)
8. Нажать «ОК» для запуска процесса. [otus.ru/datanumen.com](http://otus.ru/datanumen.com)

**Настройка регулярного резервного копирования** (например, с помощью планов обслуживания): [winitpro.ru/aseus.com](http://winitpro.ru/aseus.com)

1. В SSMS перейти в раздел «Управление» → «Планы обслуживания» и запустить мастер создания плана обслуживания.
2. Указать имя плана и выбрать режим «Separate schedules for each task».
3. Выбрать операции для резервного копирования: «Back Up Database (Full)» и «Back Up Database (Transaction Log)».
4. Указать расписание для резервного копирования: например, полная резервная копия — каждые 24 часа, копия журнала транзакций — каждые 30 минут.

#### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

### **Практическая работа 34 Автоматизация резервного копирования базы данных MongoDB с использованием скриптов.**

***Цель работы*** — изучить, как создавать резервные копии баз данных, настраивать регулярное резервное копирование с помощью планов обслуживания и решать ошибки при выполнении операций

#### ***.Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

#### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

#### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы. Предъявить преподавателю результаты работы.

- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы* На Linux

#### **Пример скрипта для автоматического резервного копирования:**

```
#!/bin/bash
BACKUP_DIR="/backup/mongodb"
MONGO_USER="backupuser"
MONGO_PASSWORD="strong_password_here"
RETENTION_DAYS=7 [1](https://arenda-server.cloud/blog/kak-delat-rezervnoe-
kopirovanie-vosstanovlenie-i-migraciju-mongodb-na-ubuntu-24/)

# Создаём директорию с текущей датой
BACKUP_PATH="$BACKUP_DIR/$(date +%Y%m%d_%H%M%S)"
mkdir -p $BACKUP_PATH

# Создаём бэкап
mongodump --username $MONGO_USER --password $MONGO_PASSWORD --
authenticationDatabase admin --gzip --out $BACKUP_PATH

# Проверяем успешность операции
if [ $? -eq 0 ]; then
echo "Backup successful: $BACKUP_PATH"

# Удаляем старые бэкапы
find $BACKUP_DIR -type d -mtime +$RETENTION_DAYS -exec rm -rf { } \;

# Опционально: отправляем уведомление
echo "MongoDB backup completed successfully" | mail -s "Backup Success"
admin@example.com
else
echo "Backup failed"
echo "MongoDB backup failed" | mail -s "Backup Failed" admin@example.com
exit 1
fi
```

**Важно:** перед выполнением команд нужно создать директорию, в которой будут расположены резервные копии MongoDB. [passwork.ru](https://passwork.ru)

#### На Windows

**Пример скрипта для резервного копирования базы данных MongoDB с помощью PowerShell:**

#### *Оформление результатов работы*

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

**Практическая работа 35 Создание и управление резервными копиями Oracle Database с помощью RMAN (Recovery Manager).**

***Цель работы*** — изучить, как создавать резервные копии баз данных, настраивать регулярное резервное копирование с помощью планов обслуживания и решать ошибки при выполнении операций

***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы. Предъявить преподавателю результаты работы.
  - Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
  - Подготовить ответы на контрольные вопросы.

***Теоретические и учебно-методические материалы по теме практической работы***

Включать настройку параметров резервного копирования с помощью RMAN (Recovery Manager), создание резервной копии базы данных и восстановление из неё. [cals.ruvinchin.com](http://cals.ruvinchin.com)[vinchin.com](http://vinchin.com)

**RMAN** — специализированная утилита Oracle, предназначенная для архивации и восстановления файлов экземпляра базы данных. Позволяет создавать резервные копии как открытой (доступной для пользователей), так и закрытой (смонтированной и не доступной для пользователей) базы данных. [cals.ru](http://cals.ru)

### Настройка

Некоторые параметры, которые можно настроить с помощью RMAN:

- **Расположение резервных копий.** По умолчанию RMAN создаёт резервные копии на диске, но можно указать другое устройство. [docs.oracle.com](http://docs.oracle.com)[vinchin.com](http://vinchin.com)
- **Формат резервных копий.** Можно указать, что резервные копии — это наборы файлов, записанные в формате, доступном только RMAN, или копии изображений. [docs.oracle.com](http://docs.oracle.com)
- **Время хранения резервных копий.** Можно настроить политику хранения, например, указать, что резервные копии должны храниться в течение определённого периода. [vinchin.com](http://vinchin.com)
- **Сжатие резервных копий.** Можно настроить, чтобы резервные копии были сжаты, чтобы сэкономить место на диске. [vinchin.com/blog.data-alchemy.org](http://vinchin.com/blog/data-alchemy.org)

### Создание

Некоторые команды RMAN для создания резервной копии базы данных:

- **Полное резервное копирование без архивных журналов** —  
RMAN> BACKUP AS BACKUPSET DATABASE
- **Резервное копирование архивных журналов** (база данных открыта) —  
RMAN> BACKUP AS BACKUPSET DATABASE PLUS ARCHIVELOG
- **Резервное копирование только табличных пространств** —  
RMAN> BACKUP AS BACKUPSET TABLESPACE  
[vinchin.com](http://vinchin.com)

**Важно:** перед вводом команд RMAN база данных должна быть запущена. [vinchin.com](http://vinchin.com)

### Восстановление

Некоторые команды RMAN для восстановления базы данных из резервной копии:

- **Восстановление базы данных** —  
RMAN> RESTORE DATABASE  
. [vinchin.com](http://vinchin.com)
- **Восстановление архивных журналов** (база данных смонтирована или открыта) —  
RMAN> RESTORE ARCHIVELOG ALL  
. Можно указать, где следует искать резервную копию, —  
catalog start with '/path/to/backup'  
. [vinchin.com](http://vinchin.com)[iliachemodanov.ru](http://iliachemodanov.ru)
- **Восстановление конкретного табличного пространства или файла данных** — в зависимости от операции восстановления используется соответствующая команда  
RECOVER TABLESPACE  
или  
RECOVER DATAFILE  
. [thegeekstuff.com](http://thegeekstuff.com)

**Важно:** RMAN автоматически восстанавливает архивные журналы и инкрементные резервные копии, но после их применения RMAN автоматически удаляет их.

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 36 Настройка расписания автоматического резервного копирования в MySQL с использованием CRON.**

***Цель работы*** освоить методику настройки автоматического резервного копирования баз данных MySQL с помощью планировщика задач CRON в Linux-системах.

### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### ***Ход практического занятия:***

• Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

• Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы. Предъявить преподавателю результаты работы.

• Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).

- Подготовить ответы на контрольные вопросы.

### ***Теоретические и учебно-методические материалы по теме практической работы***

### **Теоретические сведения**

**CRON** — системный сервис в Unix-подобных ОС для планирования и автоматического выполнения задач по расписанию. Задачи задаются в специальных файлах — *crontab* (от *cron table*).

**mysqldump** — встроенная утилита MySQL для создания резервных копий (дампов) баз данных.

### Порядок выполнения

#### 1. Проверка наличия и установка CRON

Убедитесь, что CRON установлен:

```
bash
```

```
crontab -l
```

Если команда не найдена, установите CRON:

- для Debian/Ubuntu:

```
bash
```

```
sudo apt-get install cron
```

- для RedHat/CentOS:

```
bash
```

```
sudo yum install cronic
```

Запустите сервис:

```
bash
```

```
sudo systemctl start cron
```

```
sudo systemctl enable cron
```

#### 2. Создание скрипта резервного копирования

Создайте файл скрипта, например `/usr/local/bin/mysql_backup.sh`:

```
bash
```

```
#!/bin/bash
```

```
# Параметры подключения к MySQL
```

```
MYSQL_USER="your_username"
```

```
MYSQL_PASS="your_password"
```

```
MYSQL_HOST="localhost"
```

```
# Директория для бэкапов
```

```
BACKUP_DIR="/path/to/backup/directory"
```

```
# Дата для имени файла
```

```
DATE=$(date +"%Y%m%d_%H%M%S")
```

```
# Имя файла бэкапа
```

```
BACKUP_FILE="${BACKUP_DIR}/backup_${DATE}.sql.gz"
```

```
# Создание дампа и сжатие
```

```
mysqldump -u ${MYSQL_USER} -p${MYSQL_PASS} -h ${MYSQL_HOST} --all-databases | gzip > ${BACKUP_FILE}
```

```
# Удаление бэкапов старше 7 дней
```

```
find ${BACKUP_DIR} -name "backup_*.sql.gz" -mtime +7 -exec rm {} \;
```

**Важно:**

- замените `your_username`, `your_password` и `/path/to/backup/directory` на актуальные значения;

- убедитесь, что у пользователя есть права на запись в указанную директорию.

Сделайте скрипт исполняемым:

```
bash
chmod +x /usr/local/bin/mysql_backup.sh
```

### 3. Настройка CRON-задачи

Откройте пользовательский crontab:

```
bash
crontab -e
```

Добавьте строку для ежедневного бэкапа в 02:00:

```
crontab
0 2 * * * /usr/local/bin/mysql_backup.sh
```

**Формат записи CRON:**

мин час день\_месяца месяц день\_недели команда

**Примеры расписаний:**

- каждый понедельник в 3:00: `0 3 * * 1 /путь/к/скрипту`;
- каждые 6 часов: `0 */6 * * * /путь/к/скрипту`;
- раз в месяц 1-го числа в 4:00: `0 4 1 * * /путь/к/скрипту`.

### 4. Проверка работы

- Убедитесь, что задача добавлена:

```
bash
crontab -l
```

- Проверьте логи CRON (обычно `/var/log/cron` или `/var/log/syslog`) на наличие ошибок:

```
bash
grep CRON /var/log/syslog
```

- Вручную запустите скрипт для тестирования:

```
bash
/usr/local/bin/mysql_backup.sh
```

- Убедитесь, что файл бэкапа создан в указанной директории.

### 5. Дополнительные настройки (по желанию)

- **Отправка уведомлений по email:** добавьте в скрипт после команды `mysqldump`:

```
bash
echo "Бэкап выполнен успешно: ${BACKUP_FILE}" | mail -s "MySQL Backup"
your@email.com
```

- **Копирование на удалённый сервер:** используйте `rsync` или `scp` в скрипте:

```
bash
rsync -avz ${BACKUP_FILE} user@remote_server:/remote/backup/
```

- **Шифрование бэкапа:** добавьте `openssl` после `gzip`:

```
bash
| openssl enc -aes-256-cbc -pass pass:your_password
```

### **Контрольные вопросы**

1. Что такое CRON и для чего он используется?
2. Какая команда создаёт резервную копию всех баз данных MySQL?
3. Как задать расписание для CRON-задачи? Приведите пример.
4. Где хранятся файлы crontab?
5. Как проверить, что CRON-задача выполняется?

### **Отчёт по работе**

В отчёте укажите:

1. Команды, использованные для установки и настройки CRON.
2. Полный текст скрипта резервного копирования.
3. Строку CRON-задачи с расписанием.
4. Результаты тестирования (наличие файлов бэкапа, логи).
5. Ответы на контрольные вопросы.

### **Возможные ошибки и их устранение**

- **Ошибка «command not found» при запуске `crontab`:** установите CRON (см. шаг 1).

- **Скрипт не выполняется:** проверьте права на файл (`chmod +x`) и путь к скрипту в crontab.

- **Ошибки подключения к MySQL:** убедитесь в правильности логина/пароля и доступности сервера.

- **Нет файлов бэкапа:** проверьте директорию для сохранения и логи скрипта.

### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### **Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## **Практическая работа 37 Проверка целостности и восстановления данных из резервной копии в PostgreSQL.**

**Цель работы** изучить:освоить методы проверки целостности резервных копий PostgreSQL и отработать процедуру восстановления данных из бэкапа.

**Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

**Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

**Теоретические и учебно-методические материалы по теме практической работы**

**Теоретические сведения**

PostgreSQL поддерживает несколько стратегий резервного копирования:

- **физическое (холодное/горячее)** — копирование файлов данных СУБД;
- **логическое** — создание SQL-дампов базы.

Для проверки целостности используются:

- `pg_verifybackup` — проверка по манифесту (для `pg_basebackup`);
- тестовое восстановление — наиболее надёжный способ верификации.

**Инструменты**

- `pg_basebackup` — создание физических копий;
- `pg_dump`/`pg_restore` — логическое резервное копирование;
- `pg_verifybackup` — проверка целостности;
- `psql` — выполнение SQL-команд.

**Ход работы**

**1. Создание резервной копии**

**Вариант А. Физическое копирование (`pg_basebackup`)**

```
bash
```

```
pg_basebackup -D /backup/my_backup -F tar -z -P -v
```

**Параметры:**

- `-D` — каталог для бэкапа;
- `-F tar` — формат архива;
- `-z` — сжатие;
- `-P` — показ прогресса;
- `-v` — подробный вывод.

### Вариант Б. Логическое копирование (`pg_dump`)

```
bash
```

```
pg_dump -U username -d dbname -F c -b -v -f /backup/dbname.dump
```

#### Параметры:

- `-U` — пользователь;
- `-d` — имя БД;
- `-F c` — формат custom (сжатый);
- `-b` — включая большие объекты;
- `-v` — подробности;
- `-f` — файл вывода.

## 2. Проверка целостности

### Для физического бэкапа (`pg_basebackup`)

```
bash
```

```
pg_verifybackup /backup/my_backup
```

#### Ожидаемый результат:

```
backup successfully verified
```

### Для логического бэкапа (`pg_dump`)

1. Проверка заголовка дампа:

```
bash
```

```
pg_restore --list /backup/dbname.dump
```

2. Тестовый импорт в пустую БД (рекомендуется):

```
sql
```

```
CREATE DATABASE test_restore;
```

```
\c test_restore
```

```
pg_restore -U username -d test_restore /backup/dbname.dump
```

## 3. Восстановление данных

### Вариант А. Из физического бэкапа

1. Остановите сервер:

```
bash
```

```
sudo systemctl stop postgresql
```

2. Очистите каталог данных (осторожно!):

```
bash
```

```
rm -rf /var/lib/postgresql/14/main/*
```

3. Распакуйте бэкап:

```
bash
tar -xzf /backup/my_backup/base.tar.gz -C /var/lib/postgresql/14/main
```

4. Запустите сервер:

```
bash
sudo systemctl start postgresql
```

### Вариант Б. Из логического бэкапа

1. Создайте пустую БД:

```
sql
CREATE DATABASE restored_db;
```

2. Восстановите данные:

```
bash
pg_restore -U username -d restored_db /backup/dbname.dump
```

3. Проверьте содержимое:

```
sql
\c restored_db
SELECT count(*) FROM information_schema.tables;
```

---

## 4. Верификация восстановления

Выполните контрольные запросы:

```
sql
-- Проверка количества записей в ключевых таблицах
SELECT COUNT(*) FROM important_table;

-- Сравнение сумм (если есть числовые данные)
SELECT SUM(amount) FROM transactions;

-- Проверка целостности связей
SELECT COUNT(*)
FROM orders
WHERE customer_id NOT IN (SELECT id FROM customers);
```

Сравните результаты с исходной БД (если доступна).

---

## 5. Документирование

Заполните отчёт:

1. **Параметры бэкапа:**

- метод (`pg_basebackup`/`pg_dump`);
- дата и время;
- размер архива.

2. **Результаты проверки:**

- вывод (`pg_verifybackup`);
- ошибки (если были).

3. **Процесс восстановления:**

- время выполнения;
- возникшие проблемы;

- способы решения.
- 4. **Верификация:**
- сводная таблица сравнений;
- заключение о целостности данных.

---

### Контрольные вопросы

1. В чём отличие физического и логического резервного копирования?
2. Почему тестовое восстановление — лучший способ проверки бэкапа?
3. Какие параметры `pg_basebackup` влияют на целостность копии?
4. Как обработать ошибку «database already exists» при восстановлении?
5. Зачем проверять контрольные суммы после распаковки архива?

---

### Критерии оценки

- **Отлично:** все этапы выполнены без ошибок, отчёт содержит полные данные.
- **Хорошо:** 1–2 незначительные ошибки, исправленные самостоятельно.
- **Удовлетворительно:** 3–4 ошибки, требующие подсказки.
- **Неудовлетворительно:** более 4 ошибок, отсутствие верификации.

---

### Дополнительные задания (по желанию)

1. Настройте автоматическое резервное копирование через `cron`.
2. Реализуйте инкрементальное копирование с помощью `pg_probackup`.
3. Протестируйте восстановление на другой версии PostgreSQL.

□ □ □ □ □

### *Оформление результатов работы*

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### *Содержание отчёта*

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## Практическая работа 38 Выполнение дифференциального резервного копирования в Microsoft SQL Server.

**Цель работы:** освоить процедуру создания дифференциальных резервных копий базы данных в Microsoft SQL Server с использованием SQL Server Management Studio (SSMS) и Transact-SQL **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

**Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

**Теоретические и учебно-методические материалы по теме практической работы**

**Теоретические сведения**

**Дифференциальное резервное копирование** — это копирование только тех частей базы данных, которые изменились после последнего полного резервного копирования.

**Преимущества:**

- экономит место на диске (по сравнению с полным копированием);
- сокращает время создания резервной копии;
- позволяет быстро восстановить данные до актуального состояния.

**Важные условия:**

1. Перед созданием дифференциальной копии **обязательно** должна быть выполнена полная резервная копия базы данных.
2. Для восстановления потребуется:
  - последняя полная резервная копия;
  - последняя дифференциальная резервная копия.
3. Разрешения `BACKUP DATABASE` и `BACKUP LOG` по умолчанию назначены:

- членам роли сервера `sysadmin`;
- ролям базы данных `db_owner` и `db_backupoperator`.

**Порядок выполнения**

**Способ 1. Через SQL Server Management Studio (SSMS)**

1. Подключитесь к экземпляру SQL Server в SSMS.
2. В **Обозревателе объектов** раскройте узел *Базы данных* и выберите нужную базу.

3. Щёлкните правой кнопкой мыши по базе данных → *Задачи* → *Создать резервную копию...*
4. В диалоговом окне *Резервное копирование базы данных*:
  - В поле *База данных* проверьте имя базы (можно выбрать другую).
  - В списке *Тип резервной копии* выберите **Дифференциальная** (Differential).
  - **Убедитесь, что флажок *Копировать только резервную копию* снят.**
  - В разделе *Компонент резервного копирования* выберите *База данных*.
5. **Укажите место сохранения:**
  - В разделе *Назначение* выберите *Диск* и нажмите *Добавить...*
  - В появившемся окне укажите путь и имя файла с расширением `.bak` (например, `MyDB_diff.bak`).
6. **Настройте параметры срока действия** (опционально):
  - *После* — укажите количество дней до истечения срока (0 = бессрочно).
  - *На дату* — выберите конкретную дату истечения.
7. **Дополнительные параметры** (вкладка *Параметры*):
  - *Переписать носитель* — выберите:
    - *Добавить в существующий резервный набор данных* или
    - *Перезаписать все существующие резервные наборы данных*.
  - В разделе *Надежность* можно отметить:
    - *Проверить резервную копию после завершения;*
    - *Рассчитать контрольную сумму перед записью на носитель.*
8. Нажмите **ОК** для запуска резервного копирования.
9. После завершения проверьте наличие файла резервной копии в указанном месте.

### Способ 2. Через Transact-SQL (T-SQL)

1. Откройте **новый запрос** в SSMS (кнопка *New Query*).
2. Введите команду:

```
sql
USE master
GO
BACKUP DATABASE [Имя_базы_данных]
TO DISK = N'Путь_к_файлу.bak'
WITH DIFFERENTIAL, INIT
GO
```

#### Пример:

```
sql
USE master
GO
BACKUP DATABASE [MyAdvWorks]
TO DISK = N'C:\Backup\MyAdvWorks_diff.bak'
WITH DIFFERENTIAL, INIT
GO
```

#### Параметры:

- `[Имя_базы_данных]` — имя базы данных (в квадратных скобках).

- `N'Путь_к_файлу.bak'` — полный путь к файлу резервной копии (с префиксом `N` для Unicode).
  - `DIFFERENTIAL` — указывает на дифференциальный тип копии.
  - `INIT` — перезаписывает существующий файл (если есть).
3. Нажмите **Execute** (или F5) для выполнения команды.

#### Рекомендации по эксплуатации

1. **Регулярность:**
  - Выполняйте полное резервное копирование **раз в неделю**.
  - Создавайте дифференциальные копии **ежедневно** (или чаще, в зависимости от интенсивности изменений).
2. **Хранение:**
  - Сохраняйте полные и дифференциальные копии на **разных носителях**.
  - Используйте **сетевые хранилища** или облачные сервисы для надёжности.
3. **Проверка:**
  - Периодически **проверяйте целостность** резервных копий (опция *Проверить резервную копию после завершения*).
  - Проводите **тестовые восстановления** для уверенности в работоспособности копий.
4. **Автоматизация:**
  - Настройте **планы обслуживания** (Maintenance Plans) для регулярного резервного копирования.
  - Используйте **SQL Server Agent** для автоматизации задач по расписанию.

#### Контрольные вопросы

1. Что такое дифференциальное резервное копирование?
2. Какие предварительные условия необходимы для создания дифференциальной копии?
3. Какие разрешения требуются для выполнения резервного копирования?
4. В чём отличие полной и дифференциальной резервных копий?
5. Как восстановить базу данных из дифференциальной копии?
6. Какие инструменты SQL Server можно использовать для создания резервных копий?
7. Почему важно регулярно создавать полные резервные копии при использовании дифференциального копирования?

#### Отчёт по работе

Подготовьте отчёт, включающий:

1. Скриншоты этапов создания дифференциальной копии через SSMS.
2. Текст команды T-SQL с указанием параметров.
3. Описание места хранения резервной копии.
4. Ответы на контрольные вопросы.
5. Выводы о преимуществах и недостатках дифференциального резервного копирования.

#### Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

**Практическая работа 39 Создание инкрементального резервного копирования в Oracle Database.**

***Цель работы*** изучить, как управлять учётными записями пользователей и контролировать доступ к базам данных

***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.

- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

***Теоретические и учебно-методические материалы по теме практической работы***

Инкрементальное резервное копирование в Oracle Database позволяет экономить место и время за счёт копирования только изменённых данных с момента последнего полного или инкрементального бэкапа. Основным инструментом для этого — RMAN (Recovery Manager),

который поддерживает различные типы инкрементальных бэкапов. Также можно использовать сторонние решения, например, Vinchin Backup & Recovery. [vinchin.com](http://vinchin.com) +1

## Использование RMAN

### Типы инкрементальных бэкапов в RMAN:

1. **Уровень 0 (Level 0)** — полный бэкап, который служит основой для последующих инкрементальных бэкапов уровня 1. Команда:

```
text
RMAN> BACKUP INCREMENTAL LEVEL 0 DATABASE;
```

2. **Дифференциальный (по умолчанию для уровня 1)** — копирует блоки данных, изменённые после последнего инкрементального бэкапа (уровня 0 или 1). Команда:

```
text
RMAN> BACKUP INCREMENTAL LEVEL 1 DATABASE;
```

3. **Кумулятивный (уровень 1 с параметром CUMULATIVE)** — копирует все блоки, изменённые после последнего бэкапа уровня 0. Команда:

```
text
RMAN> BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE;
```

### Стратегия резервного копирования:

- Рекомендуется выполнять полный бэкап (уровень 0) еженедельно, а инкрементальные (уровень 1) — ежедневно. [vinchin.com](http://vinchin.com) +1
- Для восстановления необходимо сначала восстановить полный бэкап, а затем последовательно применить инкрементальные бэкапы.

### Оптимизация с помощью Block Change Tracking:

- Эта функция записывает физические местоположения изменённых блоков, что ускоряет инкрементальные бэкапы, так как RMAN проверяет только указанные блоки, а не все данные.
- Включить можно через SQL-команду:

```
text
SQL> ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE
'/u01/rman_bct.ctl';
```

## Использование Vinchin Backup & Recovery

Сторонние решения, такие как Vinchin Backup & Recovery, упрощают процесс за счёт графического интерфейса и автоматизации. Основные шаги:

1. Выбрать базу данных Oracle для бэкапа.
2. Указать место назначения для резервных копий.
3. Настроить стратегию: выбрать инкрементальный бэкап, а также обязательно включить полный бэкап и архивацию логов для гарантии целостности данных.
4. Задать расписание (например, еженедельный полный бэкап + ежедневный инкрементальный).

Vinchin поддерживает дополнительные функции: сжатие данных, дедупликацию, многопоточную передачу, шифрование. [vinchin.com](http://vinchin.com) +1

## Сравнение подходов

Критерий	RMAN	Vinchin Backup & Recovery
Управление	Требует написания скриптов	Графический интерфейс, упрощённая настройка
Гибкость	Широкий контроль над процессом	Автоматизация, меньше ручного вмешательства
Дополнительные функции	Базовые возможности	Сжатие, дедупликация, шифрование, облачное хранение

### **Рекомендации**

- Для сложных сред с высокими требованиями к контролю и настройке лучше подходит RMAN.
- Для упрощения процесса и автоматизации стоит рассмотреть сторонние решения вроде Vinchin.
- Регулярно проверяйте целостность резервных копий и тестируйте восстановление.

Если у вас есть конкретные вопросы по настройке RMAN или Vinchin, уточните их — помогу разобраться!

□ □ □ □ □

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 40 Разработка стратегии резервного копирования и восстановления для базы данных предприятия.**

***Цель работы*** изучить, как управлять учётными записями пользователей и контролировать доступ к базам данных

### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ

ИИ

- Показать основные приемы эффективного использования возможностей

***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

***Теоретические и учебно-методические материалы по теме практической работы***

**1. Анализ требований и угроз**

**Ключевые параметры:**

- **RPO (Recovery Point Objective)** — максимально допустимая потеря данных (например, 1 час, 4 часа, сутки).
- **RTO (Recovery Time Objective)** — допустимое время восстановления (например, 30 минут, 2 часа).
- **SLA (Service Level Agreement)** — согласованные уровни обслуживания.

**Модель угроз включает:**

- аппаратные сбои;
- программные ошибки;
- кибератаки (шифровальщики, DDoS);
- человеческий фактор (случайное удаление);
- стихийные бедствия (пожар, затопление).

**Критичность данных:**

- высококритичные (финансовые транзакции, клиентские данные);
- средней критичности (внутренняя отчетность);
- низкокритичные (архивные данные).

**2. Выбор стратегии резервного копирования**

**Основные типы:**

1. **Полное резервное копирование**
  - Создается полная копия БД.
  - Плюсы: простота восстановления.
  - Минусы: большой объем данных, долгая процедура.

- Применяется для небольших БД или как база для инкрементных копий.
- 2. **Инкрементное резервное копирование**
  - Копируются только изменения с момента последнего бэкапа любого типа.
  - Плюсы: экономия места и времени.
  - Минусы: сложное восстановление (нужна цепочка копий).
- 3. **Дифференциальное (разностное) резервное копирование**
  - Копируются изменения с момента последнего полного бэкапа.
  - Плюсы: быстрее полного копирования, проще восстановления, чем инкрементного.

- Минусы: растёт объём с течением времени.
- 4. **Синтетическое резервное копирование**
  - Создание полной копии из инкрементных бэкапов без нагрузки на основную БД.
  - Плюсы: снижение нагрузки на систему.
  - Минусы: требует дополнительных ресурсов для сборки.
- 5. **CDP (Continuous Data Protection)**
  - Непрерывная запись изменений в реальном времени.
  - Плюсы: минимальное RPO.
  - Минусы: высокие требования к хранилищу.

### 3. Правила «3-2-1» для надёжности

- **3 копии данных** (основная + 2 резервные).
- **2 разных типа носителей** (SSD, HDD, лента, облако).
- **1 копия вне площадки** (удалённый дата-центр, облако).

### 4. Планирование расписания

#### Пример для высококритичной БД:

- Полное копирование — раз в неделю (ночь воскресенья).
- Дифференциальное — ежедневно (ночь).
- Инкрементное — каждый час в рабочее время.
- CDP — для критически важных таблиц.

#### Для менее критичных данных:

- Полное — раз в месяц.
- Инкрементное — раз в день.

### 5. Хранение резервных копий

#### Носители:

- SSD/HDD — быстрое восстановление.
- Ленточные накопители — долгосрочное хранение.
- Облако (S3, Azure Blob) — географическая избыточность.

#### Уровни хранения:

- **Hot Tier** — активные копии (SSD, локальные диски).
- **Warm Tier** — резервные копии (HDD, облачные диски).
- **Cold Tier** — архив (лента, облачный архив).

### 6. Процедуры восстановления

#### Шаги:

1. Оценка инцидента (что повреждено, когда).
2. Выбор подходящей копии (по RPO).
3. Восстановление на тестовый сервер (проверка целостности).

4. Переключение на рабочую среду (по RTO).

5. Документирование процесса.

**Тестирование:**

- Ежемесячные пробные восстановления.
- Проверка контрольных сумм.
- Автоматизированные тесты (скрипты).

**7. Технические средства**

**ПО для резервного копирования:**

- Veeam Backup & Replication;
- Acronis Cyber Protect;
- Microsoft Data Protection Manager;
- Open-source решения (Vacula, Amanda).

**Аппаратные решения:**

- NAS/SAN для локальных копий.
- Ленты LTO-9 для архива.
- Гибридные хранилища (локально + облако).

**8. Безопасность данных**

- Шифрование (AES-256) при передаче и хранении.
- Разграничение доступа (RBAC).
- Журналирование операций.
- Защита от ransomware (изолированные копии).

**9. Документация и автоматизация**

**Обязательные документы:**

- План резервного копирования (расписание, ответственные).
- Инструкция по восстановлению (пошаговые сценарии).
- Журнал инцидентов.

**Автоматизация:**

- Скрипты для проверки целостности.
- Уведомления о сбоях (email, SMS, Telegram).
- Мониторинг свободного места.

**10. Бюджет и ресурсы**

**Статьи расходов:**

- ПО (лицензии, подписки).
- Оборудование (диски, ленты).
- Облачные сервисы.
- Обучение персонала.

**Оптимизация:**

- Дедупликация данных.
- Сжатие (компрессия).
- Политика хранения (удаление старых копий).

**11. Обучение персонала**

**Программы:**

- Курсы по работе с ПО резервного копирования.
- Тренировки по восстановлению (симуляции сбоев).
- Чек-листы для дежурных инженеров.

## 12. Мониторинг и аудит

### Метрики:

- Время выполнения бэкапа.
- Объём скопированных данных.
- Ошибки (количество, причины).
- Свободное место в хранилище.

### Инструменты:

- Zabbix, Nagios для мониторинга.
- Отчёты по SLA (соблюдение RPO/RTO).

### Пример итогового плана (сводка)

Параметр	Значение
<b>Стратегия</b>	Полное + дифференциальное + инкрементное
<b>RPO</b>	1 час
<b>RTO</b>	30 минут
<b>Носители</b>	SSD (локально), облако (S3)
<b>Расписание</b>	Полное — воскресенье, 02:00; дифференциальное — ежедневно, 01:00; инкрементное — каждый час
<b>Хранение</b>	3 копии (локальная, облачная, лента)
<b>Тестирование</b>	Ежемесячные восстановления

### Вывод:

Эффективная стратегия резервного копирования требует баланса между стоимостью, скоростью и надёжностью. Ключевое — регулярное тестирование и адаптация под изменяющиеся бизнес-требования.

### *Оформление результатов работы*

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

### *Содержание отчёта*

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## Тема 1.5. Восстановление баз данных

### Практическая работа 41 Восстановление базы данных MySQL из резервной копии, созданной с помощью mysqldump.

**Цель работы:** освоить процедуру восстановления базы данных MySQL из дампа, созданного утилитой

**Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

**Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

**Теоретические и учебно-методические материалы по теме практической работы**

**Теоретические сведения**

`mysqldump` — встроенная утилита MySQL для создания резервных копий (дампов) баз данных в виде SQL-скриптов. Результат работы `mysqldump` — файл `.sql`, содержащий команды `CREATE`, `INSERT` и др., необходимые для воссоздания структуры и данных БД.

Восстановление из дампа — обратный процесс: SQL-скрипт выполняется в СУБД, и база данных восстанавливается в состоянии на момент создания дампа.

**Подготовка к восстановлению**

1. **Убедитесь, что у вас есть:**
  - файл дампа (например, `backup.sql`), созданный через `mysqldump`;
  - доступ к серверу MySQL (логин и пароль пользователя с правами на восстановление).

2. **Проверьте версию MySQL.** Версия сервера, на который вы восстанавливаете данные, должна быть совместима с версией, на которой сделан дамп.

3. **Создайте целевую базу данных** (если её ещё нет):

```
sql
```

```
CREATE DATABASE target_db;
```

Замените `target_db` на имя вашей БД.

4. **Проверьте права пользователя.** У учётной записи, от имени которой идёт восстановление, должны быть права:

- `CREATE` (на создание объектов);
- `INSERT` (на загрузку данных);
- `ALTER` (на изменение структуры, если нужно);
- `DROP` (если требуется перезаписать существующие объекты).

Пример выдачи прав:

```
sql
```

```
GRANT ALL PRIVILEGES ON target_db.* TO 'your_user'@'localhost';  
FLUSH PRIVILEGES;
```

### Порядок выполнения восстановления

#### Способ 1. Восстановление в существующую БД (простой случай)

Если дамп содержит данные **одной** базы данных и вы хотите восстановить их в БД с тем же именем (или в новую с таким же именем), выполните в терминале:

```
bash
```

```
mysql -u your_user -p target_db < /path/to/backup.sql
```

- `-u your_user` — имя пользователя MySQL;
- `-p` — запрос пароля (введите его после нажатия Enter);
- `target_db` — имя целевой БД;
- `/path/to/backup.sql` — полный путь к файлу дампа.

После ввода команды система запросит пароль. Введите его и нажмите Enter — начнётся восстановление.

#### Способ 2. Восстановление одной БД из дампа с несколькими БД

Если дамп создан с опцией `--all-databases` и содержит несколько БД, а вам нужно восстановить только одну, используйте ключ `--one-database`:

```
bash
```

```
mysql -u your_user -p --one-database target_db < /path/to/all_databases.sql
```

Здесь `target_db` — имя той БД, которую вы хотите восстановить из многобазового дампа.

#### Способ 3. Извлечение нужной БД из многобазового дампа (с помощью `sed`)

Если нужно сначала выделить SQL-код для конкретной БД, используйте `sed`:

```
bash
```

```
sed -n '/^-- Current Database: `target_db`/,/^-- Current Database: `p`'  
/path/to/all_databases.sql > target_db.sql
```

Затем восстановите полученную часть:

```
bash
mysql -u your_user -p target_db < target_db.sql
```

### Проверка результата

1. Войдите в MySQL:

```
bash
mysql -u your_user -p
```

2. Выберите БД:

```
sql
USE target_db;
```

3. Проверьте таблицы:

```
sql
SHOW TABLES;
```

4. Убедитесь, что данные на месте (например, выберите несколько строк):

```
sql
SELECT * FROM some_table LIMIT 5;
```

### Типичные проблемы и их решение

1. **Ошибка «Access denied»**

- Проверьте логин и пароль.
- Убедитесь, что у пользователя есть права на целевую БД.

2. **Ошибка синтаксиса в SQL-файле**

- Возможно, дамп создан для другой версии MySQL. Попробуйте открыть файл в текстовом редакторе и поискать явные ошибки.
- Если дамп очень большой, проверьте его целостность (например, сравнив размер с исходным).

3. **Конфликт имён объектов**

- Если в целевой БД уже есть таблицы с такими же именами, может потребоваться предварительно удалить их:

```
sql
DROP TABLE IF EXISTS table_name;
```

- Либо восстановите в чистую БД.

4. **Проблемы с кодировкой**

- Убедитесь, что кодировка дампа совпадает с кодировкой целевой БД.

В `mysqldump` можно указать кодировку явно:

```
bash
mysqldump -u user -p --default-character-set=utf8mb4 db_name > backup.sql
```

- При восстановлении тоже укажите кодировку:

```
bash
mysql -u user -p --default-character-set=utf8mb4 target_db < backup.sql
```

5. **Большой размер дампа**

- Для очень больших файлов восстановление может занять много времени.
- Можно разбить дамп на части или использовать инструменты

типа `mysqlimport` для отдельных таблиц.

### Контрольные вопросы

1. Что такое `mysqldump` и для чего он используется?
2. В каком формате сохраняется резервная копия при использовании `mysqldump`?
3. Какая команда используется для восстановления БД из дампа?
4. Как восстановить одну БД из дампа, содержащего несколько БД?
5. Какие права MySQL должны быть у пользователя для успешного восстановления?

### **Отчёт по работе**

В отчёте укажите:

1. Цель работы.
2. Пошагово описанные действия (команды, SQL-запросы).
3. Скриншоты терминала с выполнением команд и результатами.
4. Ответы на контрольные вопросы.
5. Выводы (что получилось, с какими проблемами столкнулись, как их решали).

### **Дополнительные рекомендации**

- Всегда храните исходные дампы до успешного завершения восстановления и проверки данных.
- Тестируйте восстановление на копии, а не на продуктивной БД.
- Для автоматизации используйте скрипты (например, bash-скрипты с проверкой кодов возврата).
- Рассмотрите использование инструментов вроде MySQL Workbench для графического восстановления (если разрешено условиями работы).

### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### **Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## **Практическая работа 42 Восстановление PostgreSQL базы данных из дампа (pg\_restore).**

**Цель работы** освоить процесс восстановления базы данных PostgreSQL из резервной копии (дампа) с использованием утилиты

#### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ

- Показать основные приемы эффективного использования возможностей ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

**Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

**Теоретические и учебно-методические материалы по теме практической работы**

**Теоретические сведения**

`pg_restore` — утилита командной строки PostgreSQL для восстановления базы данных из архива, созданного `pg_dump` (в форматах `custom`, `directory`, `tar`).

**Ключевые возможности:**

- восстановление всей БД или выборочных объектов (таблицы, схемы);
- создание БД перед восстановлением;
- «чистая» перезапись (удаление существующих объектов);
- параллельное восстановление;
- генерация SQL-скрипта вместо прямого восстановления.

**Предварительные требования**

1. Установленный PostgreSQL (клиентские инструменты `pg_dump`/`pg_restore`).
2. Файл дампа БД (например, `backup.dump`).
3. Права доступа к серверу БД (логин/пароль).
4. Разрешённый доступ через брандмауэр.

**Порядок выполнения**

**1. Проверка файла дампа**

Узнайте формат и содержимое архива:

```
bash
```

```
pg_restore -l backup.dump > backup_list.txt
```

Файл `backup_list.txt` покажет список объектов (таблицы, индексы и т. п.) в дампе.

**2. Создание целевой БД (если требуется)**

Если БД ещё не существует:

```
bash
```

```
createdb -U username -T template0 target_dbname
```

- `-U username` — пользователь PostgreSQL;
- `-T template0` — создание пустой БД (без объектов из `template1`).

### 3. Базовое восстановление

Восстановите данные в существующую БД:

```
bash
```

```
pg_restore -U username -d target_dbname backup.dump
```

- `-U username` — пользователь для подключения;
- `-d target_dbname` — имя целевой БД;
- `backup.dump` — путь к файлу дампа.

### 4. Восстановление с созданием БД

Если БД отсутствует, используйте ключ `-C` (`--create`):

```
bash
```

```
pg_restore -U username -C -d postgres backup.dump
```

- `-C` — создаёт БД перед восстановлением (имя берётся из дампа);
- `-d postgres` — подключение к серверу (не к целевой БД).

### 5. «Чистая» перезапись (с удалением объектов)

Удаляет существующие объекты перед восстановлением:

```
bash
```

```
pg_restore -U username -C -c -d postgres backup.dump
```

- `-c` (`--clean`) — выполняет `DROP` перед `CREATE`.

### 6. Выборочное восстановление таблиц

1. Получите список объектов:

```
bash
```

```
pg_restore -l backup.dump > list.txt
```

2. Отредактируйте `list.txt`: оставьте только нужные таблицы (удалите лишние строки).

3. Восстановите выбранные объекты:

```
bash
```

```
pg_restore -U username -d target_dbname -L list.txt backup.dump
```

- `-L list.txt` — файл со списком восстанавливаемых объектов.

### 7. Параллельное восстановление

Ускоряет процесс за счёт многопоточности:

```
bash
```

```
pg_restore -U username -j 4 -d target_dbname backup.dump
```

- `-j 4` — 4 параллельных потока.

### 8. Генерация SQL-скрипта

Создайте скрипт вместо прямого восстановления:

```
bash
```

```
pg_restore -U username -s -f restore_script.sql backup.dump
```

- `-s` (`--schema-only`) — только структура (без данных);
- `-f restore_script.sql` — вывод в файл.

Для данных используйте `-a` (`--data-only`).

### 9. Дополнительные опции

- `--no-owner` — игнорирует владельцев объектов (назначает текущего пользователя);
- `--no-acl` — не восстанавливает права доступа;
- `-v` (`--verbose`) — подробный вывод;
- `--if-exists` — пропускает ошибки при наличии объектов;
- `-h host` — хост сервера;
- `-p port` — порт PostgreSQL.

### 10. Пример полной команды

Восстановление с созданием БД, «чистой» перезаписью и параллельной загрузкой:

```
bash
pg_restore \
-U admin \
-h localhost \
-p 5432 \
-C \
-c \
--if-exists \
-j 4 \
-d postgres \
/path/to/backup.dump
```

### Контрольные вопросы

1. В чём отличие `pg_restore` от `psql` при восстановлении из дампа?
2. Какие форматы архивов поддерживает `pg_restore`?
3. Как восстановить только определённые таблицы из дампа?
4. Для чего используется ключ `-C`?
5. Как ускорить восстановление больших БД?

### Отчёт по работе

Подготовьте отчёт, включающий:

1. Команды, выполненные в ходе работы.
2. Скриншоты вывода консоли.
3. Описание возникших ошибок и способов их устранения.
4. Ответы на контрольные вопросы.

### Возможные ошибки и решения

- **Ошибка подключения:** проверьте хост, порт, логин/пароль.
- **БД не существует:** используйте `-C` или создайте БД вручную.
- **Недостаточно прав:** запустите от имени суперпользователя PostgreSQL.

- **Конфликт объектов:** добавьте `-c` или `--if-exists`.
- **Проблемы с SSL:** установите `PGSSLMODE=require` (Linux: `export PGSSLMODE=require`; Windows: `SET PGSSLMODE=require`).

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 43 Восстановление базы данных Microsoft SQL Server из полной резервной копии с использованием SSMS.**

***Цель работы*** освоить процедуру восстановления базы данных SQL Server из полной резервной копии (\*.bak) с помощью SQL Server Management Studio (SSMS).

### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
    - Предъявить преподавателю результаты работы.
    - Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
    - Подготовить ответы на контрольные вопросы.

## *Теоретические и учебно-методические материалы по теме практической работы*

### **Предварительные требования**

- Установленный и настроенный сервер **Microsoft SQL Server**.
- Установленная **SQL Server Management Studio (SSMS)**.
- Наличие файла резервной копии базы данных (\*.bak), доступного на сервере

или в сетевой папке.

### **Порядок выполнения**

#### **1. Запуск SSMS и подключение к серверу**

1. Откройте **SQL Server Management Studio**.
2. В диалоговом окне **Connect to Server** укажите:
  - **Server type:** Database Engine;
  - **Server name:** имя вашего SQL-сервера (например, localhost или полное сетевое имя);
  - **Authentication:** выберите метод аутентификации (Windows Authentication или SQL Server Authentication);
  - При использовании SQL Server Authentication введите **Login** и **Password**.
3. Нажмите **Connect**.

#### **2. Создание пустой базы данных (опционально)**

Если вы восстанавливаете в **новую** базу (не заменяя существующую):

1. В **Object Explorer** щёлкните правой кнопкой по папке **Databases**.
2. Выберите **New Database....**
3. В окне **New Database:**
  - В поле **Database name** введите имя новой базы (например, RestoredDB).
  - При необходимости настройте файлы данных и логов на вкладке **Files**.
4. Нажмите **ОК**.

**Примечание:** если восстанавливаете в существующую базу, этот шаг пропускайте.

#### **3. Запуск процесса восстановления**

1. В **Object Explorer** щёлкните правой кнопкой по **Databases** (или по существующей базе, если заменяете её).
2. Выберите **Tasks** → **Restore** → **Database....**

#### **4. Настройка параметров на вкладке General**

1. В поле **To database** укажите имя базы для восстановления:
  - Если создавали новую базу — выберите её из списка.
  - Если заменяете существующую — выберите её имя (данные будут перезаписаны!).
2. В разделе **Source for restore** выберите **From device**.
3. Нажмите кнопку ... справа от поля **Backup media**.
4. В открывшемся окне:
  - Нажмите **Add**.
  - Укажите путь к файлу \*.bak (например, C:\Backups\MyBackup.bak).
  - Нажмите **ОК**.
5. В списке **Select the backup sets to restore** отметьте нужную резервную копию (обычно первая в списке).

6. Нажмите **ОК** для возврата в основное окно.

### 5. Настройка файлов на вкладке Files

1. В таблице **Restore the database files as** проверьте пути к файлам данных (**Data**) и логов (**Log**).

2. Если необходимо изменить пути (например, чтобы избежать конфликта с существующими файлами):

- Щёлкните по полю **Restore As** для нужного файла.
- Введите новый путь и имя файла (например, `C:\SQLData\RestoredDB.mdf` для данных и `C:\SQLLogs\RestoredDB_log.ldf` для логов).
- Убедитесь, что папки существуют и у SQL Server есть права на запись.

### 6. Настройка опций на вкладке Options

1. Установите флажок **Overwrite the existing database (WITH REPLACE)** — это позволит заменить существующую базу.

2. При необходимости настройте:
- **Take tail-log backup before restore** (если нужно сохранить последние транзакции).
  - **Restricted user access** (ограничить доступ после восстановления).
3. В разделе **Recovery state** выберите:
- **RESTORE WITH RECOVERY** (стандартный вариант — база будет доступна после восстановления).
  - **RESTORE WITH NORECOVERY** (если планируете применять дополнительные резервные копии).

### 7. Запуск восстановления

1. Проверьте все настройки.
2. Нажмите **ОК** для начала процесса.
3. Дождитесь завершения восстановления (появится сообщение **Restore succeeded**).

### 8. Проверка восстановленной базы

1. В **Object Explorer** обновите список баз (щёлкните правой кнопкой по **Databases** → **Refresh**).

2. Откройте новую/восстановленную базу и проверьте:
- Наличие таблиц и данных.
  - Целостность данных (например, выполните запрос `SELECT COUNT(*) FROM SomeTable`).

3. При необходимости измените **уровень совместимости** базы:

- Щёлкните правой кнопкой по базе → **Properties**.
- На вкладке **Options** выберите актуальный уровень совместимости (например, **150** для SQL Server 2019).

### Типичные ошибки и их устранение

- \**Ошибка доступа к файлу .bak*: проверьте права доступа к папке с резервной копией и наличие файла.
- **Конфликт путей файлов**: измените пути на вкладке **Files** на доступные директории.

- **Недостаточно места на диске:** убедитесь, что на целевом диске достаточно свободного пространства.

- **Ошибка совместимости версий:** восстановите базу на сервере той же или более новой версии SQL Server.

#### **Контрольные вопросы**

1. Какие предварительные условия необходимы для восстановления базы данных через SSMS?

2. В чём разница между восстановлением в новую базу и заменой существующей?

3. Зачем нужно изменять пути файлов данных и логов при восстановлении?

4. Что означает опция **WITH REPLACE**?

5. Как проверить успешность восстановления базы?

#### **Отчёт по работе**

Подготовьте отчёт, включающий:

1. Скриншоты ключевых этапов восстановления (выбор файла \*.bak, настройка путей, завершение процесса).

2. Описание возникших проблем и способов их решения (если были).

3. Вывод о работоспособности восстановленной базы (результаты проверок).

#### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### **Содержание отчёта**

1. Цель.

2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

### **Практическая работа 44 Восстановление базы данных MongoDB из резервного архива.**

**Цель работы** освоить процесс восстановления базы данных MongoDB из резервной копии с использованием утилиты

#### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

#### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### Ход практического занятия:

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### Теоретические и учебно-методические материалы по теме практической работы

`mongorestore`.

#### Необходимые инструменты

- Установленный MongoDB Server.
- Утилиты MongoDB Database Tools (включают `mongodump` и `mongorestore`).
- Резервная копия базы данных (созданная ранее через `mongodump`).

#### Пошаговая инструкция

##### 1. Подготовка среды

1. Убедитесь, что сервер MongoDB (`mongod`) запущен:

```
bash
sudo systemctl status mongod
```

2. Проверьте наличие утилит MongoDB Tools в системе. Они обычно располагаются в каталоге:

```
3. /usr/bin/mongorestore
4. /usr/bin/mongodump
```

##### 2. Проверка структуры резервной копии

Резервная копия, созданная `mongodump`, имеет следующую структуру:

```
/путь/к/резервной_копии/
├── database_name/
│   ├── collection1.bson
│   ├── collection1.metadata.json
│   ├── collection2.bson
│   └── collection2.metadata.json
```

Пример пути: `/var/backups/mongobackups/10-29-20/mydb/`

##### 3. Базовый сценарий восстановления

Чтобы восстановить базу данных из резервной копии:

```
bash
mongorestore --db <имя_базы_данных> <путь_к_резервной_копии>
```

#### Пример:

```
bash
```

```
mongorestore --db mydb /var/backups/mongobackups/10-29-20/mydb/
```

#### 4. Дополнительные опции `mongorestore`

- `--drop` — перед восстановлением удаляет существующую базу данных

(гарантирует чистоту восстановления):

```
bash
```

```
mongorestore --db mydb --drop /var/backups/mongobackups/10-29-20/mydb/
```

- `--nsInclude` — восстанавливает только указанные коллекции (например, все коллекции базы `mydb`):

```
bash
```

```
mongorestore --nsInclude=mydb.* /var/backups/mongobackups/10-29-20/mydb/
```

- Восстановление отдельной коллекции:

```
bash
```

```
mongorestore --nsInclude=mydb.users /var/backups/mongobackups/10-29-20/mydb/
```

- `--gzip` — если резервная копия сжата (создана с `--gzip`):

```
bash
```

```
mongorestore --db mydb --gzip /var/backups/mongobackups/mydb.gz
```

- `--uri` — восстановление на удалённый сервер:

```
bash
```

```
mongorestore --uri="mongodb://user:pass@remote-host:27017" --db mydb
```

```
/var/backups/mongobackups/10-29-20/my/
```

#### 5. Проверка результата

1. Зайдите в оболочку MongoDB:

```
bash
```

```
mongo
```

2. Переключитесь на восстановленную базу:

```
javascript
```

```
use mydb
```

3. Проверьте списки коллекций и количество документов:

```
javascript
```

```
show collections
```

```
db.users.count() // для конкретной коллекции
```

#### 6. Типичные ошибки и их устранение

- **Ошибка «permission denied»**

Убедитесь, что у вас есть права на чтение резервной копии. Используйте `sudo` при необходимости.

- **Ошибка «database already exists»**

Добавьте опцию `--drop` в команду `mongorestore`.

- **Не найдены файлы `.bson`**

Проверьте путь к резервной копии — он должен указывать на каталог с файлами `.bson` и `.metadata.json`.

- **Несовпадение версий MongoDB**

Убедитесь, что версия сервера MongoDB совместима с версией, использованной при создании дампа.

**Контрольные вопросы**

1. Какая утилита используется для восстановления базы данных MongoDB?
2. Что делает опция `--drop` в команде `mongorestore`?
3. Как восстановить только одну коллекцию из резервной копии?
4. Какой параметр позволяет восстановить данные на удалённый сервер?
5. Где хранятся файлы резервной копии после выполнения `mongodump`?

**Отчёт по работе**

В отчёте укажите:

1. Команды, которые вы выполняли.
2. Пути к резервным копиям.
3. Результаты выполнения команд (вывод в терминале).
4. Ответы на контрольные вопросы.
5. Описание возникших проблем и способов их решения (если были).

**Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

**Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

**Практическая работа 45 Восстановление Oracle Database с использованием RMAN (Recovery Manager).**

**Цель работы** освоить процедуру восстановления базы данных Oracle с помощью инструмента RMAN (Recovery Manager), включая подготовку среды, выполнение восстановления и проверку работоспособности БД после восстановления.

**Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### *Ход практического занятия:*

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

Освоить процедуру восстановления базы данных Oracle с помощью инструмента **RMAN** (Recovery Manager), включая подготовку среды, выполнение восстановления и проверку работоспособности БД после восстановления.

#### **Теоретические сведения**

**RMAN** — встроенный инструмент Oracle для резервного копирования и восстановления БД. Он позволяет:

- создавать полные и инкрементные резервные копии;
- восстанавливать отдельные файлы, табличные пространства, всю БД;
- автоматизировать процессы резервного копирования;
- управлять хранением резервных копий.

Ключевые компоненты:

- **контрольные файлы** (`controlfile`) — содержат метаданные о структуре БД;
- **архивные журналы повтора** (`archived redo logs`) — необходимы для восстановления до определённой точки времени;
- **файл параметров** (`SPFILE` или `PFILE`) — конфигурация экземпляра БД.

#### **Подготовка к восстановлению**

1. **Убедитесь в наличии резервных копий:**
  - проверьте расположение резервных файлов (обычно в `$ORACLE_HOME/flash_recovery_area` или указанном каталоге);
    - удостоверьтесь, что есть:
      - резервная копия контрольных файлов;
      - полные резервные копии данных;
      - архивные журналы повтора за нужный период.
2. **Подготовьте среду:**
  - установите переменные окружения Oracle (`ORACLE_HOME`, `ORACLE_SID`);

○ убедитесь, что экземпляр БД остановлен (если восстановление на другом сервере — настройте сетевое подключение).

### 3. Запустите RMAN:

```
bash  
rman target /
```

Или с указанием пользователя:

```
bash  
rman target sys/пароль@SID
```

## Пошаговая инструкция по восстановлению

### 1. Установите DBID (если требуется)

Если восстанавливаете на другом сервере, укажите идентификатор БД:

```
sql  
RMAN> SET DBID 12345;
```

*DBID можно узнать из имени файла автобэкапа контрольных файлов (например, `ctl_c-12345-...`).*

### 2. Запустите экземпляр в режиме NOMOUNT

```
sql  
RMAN> STARTUP NOMOUNT;
```

### 3. Восстановите контрольные файлы

Вариант 1. Из автобэкапа:

```
sql  
RMAN> RESTORE CONTROLFILE FROM AUTOBACKUP;
```

Вариант 2. Из указанного файла:

```
sql  
RMAN> RESTORE CONTROLFILE FROM '/backup/rman/ctl_c-12345-20251213-01';
```

### 4. Переведите БД в режим MOUNT

```
sql  
RMAN> ALTER DATABASE MOUNT;
```

### 5. Восстановите данные

```
sql  
RMAN> RESTORE DATABASE;
```

*RMAN автоматически определит нужные резервные файлы.*

### 6. Примените архивные журналы

Для восстановления до последней доступной точки:

```
sql  
RMAN> RECOVER DATABASE;
```

Для восстановления до определённой SCN или времени:

```
sql  
RMAN> RECOVER DATABASE UNTIL SCN 1234567;
```

-- или

```
RMAN> RECOVER DATABASE UNTIL TIME "TO_DATE('2025-12-13 10:00:00',  
'YYYY-MM-DD HH24:MI:SS')";
```

### 7. Откройте БД

После успешного восстановления:

```
sql
```

```
RMAN> ALTER DATABASE OPEN RESETLOGS;
```

Команда `RESETLOGS` создаёт новый поток журналов повтора.

### Дополнительные сценарии

#### Восстановление отдельного табличного пространства

```
sql
```

```
RMAN> RECOVER TABLESPACE users;
```

#### Восстановление архивных журналов в новый каталог

```
sql
```

```
RMAN> SET ARCHIVELOG DESTINATION TO '/new_arch/';
```

```
RMAN> RESTORE ARCHIVELOG ALL;
```

#### Просмотр информации о бэкапах

```
sql
```

```
RMAN> LIST BACKUP;
```

```
RMAN> LIST ARCHIVELOG ALL;
```

#### Проверка после восстановления

1. Убедитесь, что БД открыта:

```
sql
```

```
SQL> SELECT OPEN_MODE FROM V$DATABASE;
```

Ожидаемый результат: `READ WRITE`.

2. Проверьте целостность данных:

```
sql
```

```
SQL> SELECT COUNT(*) FROM важная_таблица;
```

3. Убедитесь, что архивные журналы применены:

```
sql
```

```
SQL> SELECT SEQUENCE#, APPLIED FROM V$ARCHIVED_LOG ORDER BY SEQUENCE#;
```

#### Типичные ошибки и решения

- **Ошибка «DBID not set»** — укажите DBID вручную (шаг 1).
- **Не найдены резервные файлы** — проверьте путь в конфигурации RMAN (`SHOW ALL;`).
- **Нехватка места в recovery area** — очистите старые бэкапы или увеличьте квоту.
- **Ошибка при `OPEN RESETLOGS`** — убедитесь, что все архивные журналы применены.

#### Контрольные вопросы

1. Для чего нужен параметр `RESETLOGS` при открытии БД?
2. Как восстановить только одно табличное пространство?
3. Где хранятся автобэкапы контрольных файлов по умолчанию?
4. Как проверить список доступных резервных копий в RMAN?
5. В чём отличие режимов `NOMOUNT`, `MOUNT` и `OPEN`?

#### Отчёт по работе

Подготовьте отчёт, включающий:

1. Скриншоты ключевых этапов восстановления (команды RMAN, вывод).

2. Логи ошибок (если были) и способы их устранения.
3. Результаты проверки работоспособности БД после восстановления.
4. Ответы на контрольные вопросы.

**Примечание:** перед выполнением восстановления на продуктивной среде протестируйте процедуру на тестовой БД.

#### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

### **Практическая работа 46 Восстановление данных из резервной копии MySQL с проверкой целостности данных.**

***Цель работы*** освоить процедуру восстановления базы данных MySQL из резервной копии и научиться проверять целостность восстановленных данных.

#### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

#### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

#### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

***Теоретические и учебно-методические материалы по теме практической работы***

## Необходимое оборудование и ПО

- Сервер с установленной СУБД MySQL (версия 5.7+ или 8.0+)
- Утилита `mysqldump` (встроена в MySQL)
- Доступ к командной строке (CLI) сервера
- Резервная копия базы данных (файл `.sql` или сжатый архив)

## Теоретические сведения

**Резервное копирование** в MySQL обычно выполняется через `mysqldump`, который создаёт SQL-скрипт с командами:

- `CREATE DATABASE` / `CREATE TABLE` — воссоздание структуры;
- `INSERT` — загрузка данных;
- `LOCK TABLES` — временная блокировка для консистентности.

**Восстановление** подразумевает выполнение этого скрипта в СУБД. Проверка целостности включает:

- сравнение количества записей;
- проверку контрольных сумм;
- тестирование ключевых запросов.

## Порядок выполнения

### 1. Подготовка резервной копии

Убедитесь, что у вас есть файл резервной копии (например, `backup_20251213.sql`).

Если копия сжата (`.sql.gz`), распакуйте:

```
bash
gunzip backup_20251213.sql.gz
```

### 2. Создание целевой базы данных

Подключитесь к MySQL и создайте пустую базу для восстановления:

```
sql
CREATE DATABASE restored_db DEFAULT CHARACTER SET utf8mb4 DEFAULT
COLLATE utf8mb4_unicode_ci;
```

*Примечание:* укажите нужную кодировку (обычно `utf8mb4` для поддержки Юникода).

### 3. Восстановление из резервной копии

Выполните импорт через командную строку:

```
bash
mysql -u ваш_пользователь -p restored_db < backup_20251213.sql
```

- `-u ваш_пользователь` — имя пользователя MySQL;
- `-p` — запрос пароля;
- `restored_db` — имя целевой базы;
- `backup_20251213.sql` — путь к файлу бэкапа.

*Важно:* если в файле бэкапа есть команда `USE имя_базы`, она переопределит указанную в команде. Проверьте файл:

```
bash
grep "USE" backup_20251213.sql
```

Если нужно, прокомментируйте `USE` в файле.

#### 4. Проверка структуры базы

Убедитесь, что все таблицы созданы:

```
sql
USE restored_db;
SHOW TABLES;
```

Сравните список таблиц с исходной базой (если доступна).

#### 5. Проверка количества записей

Для каждой ключевой таблицы сравните число строк:

```
sql
SELECT COUNT(*) FROM имя_таблицы;
```

Запишите результаты и сопоставьте с исходными данными (например, из отчётов или предыдущей проверки).

#### 6. Проверка целостности связей (для реляционных данных)

Если в базе есть внешние ключи, убедитесь, что ссылки не нарушены. Пример проверки:

```
sql
SELECT
  COUNT(*) AS orphaned_rows
FROM
  дочерняя_таблица
LEFT JOIN
  родительская_таблица
ON
  дочерняя_таблица.foreign_key = родительская_таблица.primary_key
WHERE
  родительская_таблица.primary_key IS NULL;
```

Результат должен быть `0` (нет «потерянных» строк).

#### 7. Проверка контрольных сумм

Для таблиц без сложных связей можно сравнить хеш-суммы данных. Пример для таблицы `users`:

```
sql
SELECT
  MD5(GROUP_CONCAT(CONCAT_WS('|', id, name, email) ORDER BY id))
AS data_hash
FROM users;
```

Сравните полученный хеш с эталонным (если он был зафиксирован до бэкапа).

#### 8. Тестирование ключевых запросов

Выполните типовые запросы, используемые в приложении:

```
sql
-- Пример 1: выборка активных пользователей
SELECT COUNT(*) FROM users WHERE status = 'active';
```

-- Пример 2: сумма заказов за период

```
SELECT SUM(amount) FROM orders WHERE order_date >= '2025-01-01';
```

Убедитесь, что результаты логичны и соответствуют ожиданиям.

## 9. Проверка индексов и производительности

Убедитесь, что индексы восстановлены:

```
sql
```

```
SHOW INDEX FROM имя_таблицы;
```

Выполните `EXPLAIN` для ключевых запросов, чтобы убедиться, что СУБД использует индексы:

```
sql
```

```
EXPLAIN SELECT * FROM users WHERE email = 'test@example.com';
```

## 10. Финальная верификация

Создайте отчёт с результатами:

1. Список восстановленных таблиц.
2. Количество записей в ключевых таблицах.
3. Результаты проверок внешних ключей.
4. Контрольные суммы (если использовались).
5. Примеры тестовых запросов и их результаты.

### Возможные ошибки и их устранение

1. **Ошибка доступа** (`Access denied`):
  - Проверьте логин и пароль MySQL.
  - Убедитесь, что у пользователя есть права на создание БД и загрузку данных.
2. **Конфликт кодировок**:
  - Укажите кодировку при создании БД (см. шаг 2).
  - Добавьте в команду восстановления параметр `--default-character-set=utf8mb4`.
3. **Ошибки синтаксиса в бэкапе**:
  - Откройте файл `.sql` в текстовом редакторе и найдите строки с ошибками.
  - Если бэкап создан для другой версии MySQL, могут потребоваться правки.
4. **Нехватка памяти при импорте**:
  - Разделите бэкап на части (по таблицам) и загружайте поочерёдно.
  - Увеличьте `max_allowed_packet` в конфигурации MySQL.

### Требования к отчёту

Включите в отчёт:

1. Описание шагов выполнения (кратко).
2. Команды, использованные для восстановления.
3. Результаты проверок (таблицы, запросы, хеши).
4. Выводы о целостности данных (успешно/неуспешно).
5. Список обнаруженных проблем и способов их решения.

### Контрольные вопросы

1. Какие утилиты MySQL используются для резервного копирования и восстановления?
2. Почему важно проверять целостность данных после восстановления?

3. Как убедиться, что внешние ключи не нарушены?
4. В каких случаях может потребоваться изменение кодировки при восстановлении?
5. Какие параметры `mysqldump` помогают создать консистентный бэкап?

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 47 Восстановление базы данных PostgreSQL на новый сервер с сохранением всех параметров.**

***Цель работы*** освоить процедуру полного восстановления базы данных PostgreSQL на новом сервере с сохранением структуры, данных, прав доступа и конфигурационных параметров.

### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.

• Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).

- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

#### **Инструменты**

- `pg_dump` — создание логического дампа;
- `pg_restore` — восстановление из архива;
- `psql` — выполнение SQL-скриптов;
- `pgAdmin 4` — графический интерфейс (опционально).

#### **Пошаговая инструкция**

##### **1. Создание резервной копии (дампа) на исходном сервере**

###### **Вариант А. Через командную строку (pg\_dump)**

```
bash
pg_dump -U <пользователь> -h <хост> -p <порт> -F c -b -v -f "<путь_к_файлу>.backup"
<имя_БД>
```

- `-U` — имя пользователя PostgreSQL;
- `-h` — адрес сервера (например, `localhost`);
- `-p` — порт (по умолчанию `5432`);
- `-F c` — формат `custom` (сжатый архив);
- `-b` — включать большие объекты;
- `-v` — подробный вывод;
- `-f` — путь к файлу дампа.

###### **Вариант В. Через pgAdmin 4**

1. Щёлкните правой кнопкой по БД → «Резервная копия».
2. Выберите формат «Специальный» (Custom).
3. Укажите путь и имя файла (например, `backup.backup`).
4. При необходимости настройте уровень сжатия (0–9).
5. Нажмите «Резервная копия».

##### **2. Передача дампа на целевой сервер**

Скопируйте файл дампа на новый сервер:

```
bash
scp <путь_к_файлу>.backup user@<новый_сервер>:<путь_на_новом_сервере>
```

Или используйте FTP/SFTP.

##### **3. Подготовка целевого сервера**

1. **Установите PostgreSQL** (если ещё не установлен):

```
bash
sudo apt install postgresql postgresql-contrib # для Debian/Ubuntu
```

2. **Создайте пользователя** (если нужен новый):

```
bash
```

```
sudo -u postgres createuser <пользователь>
```

3. **Создайте пустую БД для восстановления:**

```
bash
```

```
sudo -u postgres createdb -T template0 <имя_новой_БД>
```

- `-T template0` гарантирует, что БД будет пустой (без объектов из `template1`).

#### 4. Восстановление из дампа

##### Вариант А. Через командную строку (pg\_restore)

```
bash
```

```
pg_restore -U <пользователь> -h <хост> -p <порт> -d <имя_новой_БД> -v
```

```
<путь_к_файлу>.backup
```

- `-d` — имя целевой БД;
- `-v` — подробный вывод.

Для полного восстановления с пересозданием БД (если нужно заменить существующую):

```
bash
```

```
pg_restore -U <пользователь> -h <хост> -p <порт> -C -d postgres -v
```

```
<путь_к_файлу>.backup
```

- `-C` — пересоздаёт БД (имя берётся из архива);
- `-d postgres` — подключение к системной БД для выполнения `DROP/CREATE`.

##### Вариант В. Через pgAdmin 4

1. Подключитесь к целевому серверу.
2. Щёлкните по БД → «Восстановить».
3. Укажите путь к файлу `.backup`.
4. Нажмите «Восстановить».

#### 5. Проверка восстановления

1. **Подключитесь к БД:**

```
bash
```

```
psql -U <пользователь> -d <имя_новой_БД>
```

2. **Проверьте таблицы:**

```
sql
```

```
\dt
```

3. **Проверьте данные (пример):**

```
sql
```

```
SELECT count(*) FROM <название_таблицы>;
```

4. **Убедитесь в сохранности прав доступа:**

```
sql
```

```
SELECT grantee, privilege_type FROM information_schema.role_table_grants WHERE  
table_name = '<название_таблицы>;'
```

#### 6. Дополнительные настройки (при необходимости)

1. **Восстановите параметры конфигурации** (если они были изменены):

- Сравните `postgresql.conf` на исходном и целевом серверах.
- Примените нужные настройки и перезапустите PostgreSQL:

```
bash
```

```
sudo systemctl restart postgresql
```

2. **Проверьте табличные пространства** (если использовались):

```
sql
```

```
SELECT spcname, pg_tablespace_location(oid) FROM pg_tablespace;
```

### Типичные ошибки и их решение

1. **«Версия сервера не совпадает»**

→ Убедитесь, что версии PostgreSQL на исходном и целевом серверах совместимы.

2. **«Недостаточно прав»**

→ Используйте суперпользователя (`postgres`) или предоставьте нужные права:

```
sql
```

```
GRANT ALL PRIVILEGES ON DATABASE <имя_БД> TO <пользователь>;
```

3. **«Файл не найден»**

→ Проверьте путь к дампу и права доступа к файлу.

4. **«Конфликт имён»**

→ Удалите существующую БД перед восстановлением:

```
bash
```

```
dropdb <имя_БД>
```

### Контрольные вопросы

1. В чём отличие форматов дампа `custom` и `plain`?

2. Зачем использовать `template0` при создании БД?

3. Какой ключ `pg_restore` позволяет пересоздать БД?

4. Как проверить права доступа к таблицам после восстановления?

### Отчёт по работе

Предоставьте:

1. Скриншоты этапов создания и восстановления дампа.

2. Вывод команд `\dt` и `SELECT count(*)` из проверенной БД.

3. Описание возникших проблем и их решений (если были).

### Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### Содержание отчёта

1. Цель.

2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

**Практическая работа 48 Выполнение восстановления базы данных Microsoft SQL Server из дифференциальной резервной копии.**

**Цель работы:** освоить процедуру восстановления базы данных SQL Server из дифференциальной резервной копии с использованием SQL Server Management Studio (SSMS) и Transact-SQL.

**Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

**Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

**Теоретические и учебно-методические материалы по теме практической работы**

**Теоретические сведения**

**Дифференциальное резервное копирование** фиксирует только изменения, внесённые в базу данных с момента последнего полного резервного копирования.

**Ключевые особенности:**

- подходит для всех моделей восстановления (полная, с неполным протоколированием, простая);
- экономит время и место по сравнению с полным копированием;
- для восстановления требуется: последняя полная копия + последняя дифференциальная копия.

**Подготовка к восстановлению**

1. Убедитесь, что есть:
  - последняя **полная резервная копия** базы данных;
  - последняя **дифференциальная резервная копия**, созданная после полной.
2. Проверьте права доступа:
  - роль `sysadmin`;

○ или роли `db_owner` и `db_backupoperator` с разрешениями `BACKUP DATABASE` и `BACKUP LOG`.

3. Убедитесь в наличии прав на чтение/запись устройств хранения резервных копий.

### Способ 1. Восстановление через SQL Server Management Studio (SSMS)

**Шаг 1.** Подключитесь к экземпляру SQL Server в SSMS.

**Шаг 2.** В **Обозревателе объектов** разверните узел «**Базы данных**».

**Шаг 3.** Щёлкните правой кнопкой мыши по целевой базе данных → «**Задачи**» → «**Восстановить**» → «**База данных**».

**Шаг 4.** На странице «**Общие**» в разделе «**Источник**» выберите:

• «**Устройство**» → нажмите «**Обзор**» (...);

• в диалоговом окне «**Выбор устройств резервного копирования**» нажмите «**Добавить**»;

• укажите путь к файлам резервных копий (полной и дифференциальной);

• нажмите «**ОК**».

**Шаг 5.** В разделе «**Назначение**» проверьте или задайте имя восстанавливаемой базы данных.

**Шаг 6.** В сетке «**Резервные наборы данных для восстановления**» отметьте:

• последнюю **полную резервную копию**;

• последнюю **дифференциальную резервную копию** (должна идти после полной).

**Шаг 7.** На странице «**Параметры**» настройте:

• «**Перезаписать существующую базу данных**» (`WITH REPLACE`) — если нужно заменить текущую БД;

• «**Состояние восстановления**»:

○ `RESTORE WITH RECOVERY` — база готова к использованию (откат незафиксированных транзакций);

○ `RESTORE WITH NORECOVERY` — база в нерабочем состоянии (можно добавлять журналы транзакций);

○ `RESTORE WITH STANDBY` — режим «только для чтения» с возможностью отката.

• «**Закрыть существующие подключения**» — чтобы прервать активные сессии;

• «**Выдавать запрос перед восстановлением каждой резервной копии**» — для контроля процесса (опционально).

**Шаг 8.** Нажмите «**ОК**» для запуска восстановления.

### Способ 2. Восстановление через Transact-SQL

**Шаг 1.** Восстановите полную резервную копию в режиме `NORECOVERY`:

```
sql
RESTORE DATABASE [Имя_БД]
FROM DISK = N'Путь_к_полной_копии.bak'
WITH NORECOVERY, REPLACE;
```

GO

**Шаг 2.** Восстановите дифференциальную резервную копию:

```
sql
```

```
RESTORE DATABASE [Имя_БД]
```

```
FROM DISK = N'Путь_к_дифференциальной_копии.bak'
```

```
WITH RECOVERY;
```

```
GO
```

**Пояснения к параметрам:**

- `REPLACE` — перезаписывает существующую базу данных;
- `NORECOVERY` — оставляет БД в нерабочем состоянии (нужно для

последовательного восстановления);

- `RECOVERY` — завершает восстановление, делает БД доступной.

**Контроль результата**

1. Проверьте состояние базы данных:

```
sql
```

```
SELECT name, state_desc FROM sys.databases WHERE name = 'Имя_БД';
```

2. Убедитесь, что данные соответствуют ожидаемому состоянию на момент создания дифференциальной копии.

3. Протестируйте ключевые операции (выборки, обновления) в восстановленной БД.

**Типичные ошибки и их устранение**

1. «Резервная копия создана более новой версией SQL Server»

→ Используйте совместимую версию SQL Server.

2. «Активные соединения блокируют восстановление»

→ В SSMS включите опцию «Закрывать существующие подключения» или выполните:

```
sql
```

```
ALTER DATABASE [Имя_БД] SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
```

3. «Недостаточно прав для восстановления»

→ Проверьте членство в ролях `sysadmin` или `db_owner`.

4. «Файл резервной копии не найден»

→ Убедитесь в корректности пути к файлу и правах доступа к папке.

**Вывод**

В ходе работы освоены:

- алгоритм восстановления из дифференциальной копии через SSMS;
- синтаксис команд `RESTORE DATABASE` в Transact-SQL;
- методы контроля целостности восстановленной базы данных.

**Важно!** Для успешного восстановления всегда сохраняйте цепочку: **полная копия** → **дифференциальная копия**.

**Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

## ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 49 Настройка сценария аварийного восстановления базы данных MongoDB.**

***Цель работы:*** освоить методы резервного копирования и восстановления данных в MongoDB, сформировать устойчивый план аварийного восстановления (Disaster Recovery, DR).

### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### ***Теоретические и учебно-методические материалы по теме практической работы***

#### **Теоретические основы**

Аварийное восстановление в MongoDB строится на следующих ключевых механизмах:

- **репликация** (набор реплик с первичным, вторичными узлами и арбитром);

- **резервное копирование** (создание копий данных);
- **высокая доступность (HA)**;
- **автоматическое переключение** при сбоях;
- **мониторинг и оповещения** о нештатных ситуациях.

### Основные риски

- сбой ПО и оборудования;
- кибератаки и утечки данных;
- человеческие ошибки (случайное удаление, неверные настройки).

### Пошаговая инструкция

#### 1. Настройка репликации (набор реплик)

1. Запустите три экземпляра `mongod` с параметрами репликации:

```
bash
mongod --replSet rs0 --dbpath /data/db1 --port 27017 --bind_ip localhost
mongod --replSet rs0 --dbpath /data/db2 --port 27018 --bind_ip localhost
mongod --replSet rs0 --dbpath /data/db3 --port 27019 --bind_ip localhost
```

2. Инициализируйте набор реплик:

```
javascript
rs.initiate({
  _id: "rs0",
  members: [
    { _id: 0, host: "localhost:27017" },
    { _id: 1, host: "localhost:27018" },
    { _id: 2, host: "localhost:27019" }
  ]
})
```

3. Проверьте статус репликации:

```
javascript
rs.status()
```

#### 2. Резервное копирование данных (`mongodump`)

1. Создайте полную резервную копию всех баз данных:

```
bash
mongodump --out /backup/full-$(date +%Y%m%d)
```

2. Создайте резервную копию конкретной базы данных:

```
bash
mongodump --db myDatabase --out /backup/myDatabase-$(date +%Y%m%d)
```

3. Создайте резервную копию отдельной коллекции:

```
bash
mongodump --collection myCollection --db myDatabase --out /backup/collection-$(date +%Y%m%d)
```

#### 3. Восстановление данных (`mongorestore`)

1. Восстановите всю базу данных из резервной копии:

```
bash
mongorestore --db myDatabase /backup/myDatabase-20251213
```

2. Восстановите с перезаписью существующей базы (осторожно!):

```
bash
mongorestore --db myDatabase --drop /backup/myDatabase-20251213
```

3. Восстановите без индексов (быстрее):

```
bash
mongorestore --noIndexRestore --db myDatabase /backup/myDatabase-20251213
```

4. Восстановите на удалённый сервер:

```
bash
mongorestore --host remote-host --port 27017 --username admin --password secret
/backup/myDatabase-20251213
```

#### 4. Экспорт/импорт данных (`mongoexport` / `mongoimport`)

1. Экпортируйте коллекцию в JSON:

```
bash
mongoexport --db myDatabase --collection myCollection --out backup.json
```

2. Экпортируйте в CSV (с заголовками):

```
bash
mongoexport --db myDatabase --collection myCollection --type csv --fields field1,field2 --
out backup.csv
```

3. Импортируйте из JSON:

```
bash
mongoimport --db myDatabase --collection myCollection backup.json
```

4. Импортируйте из CSV:

```
bash
mongoimport --db myDatabase --collection myCollection --type csv --headerline backup.csv
```

#### 5. Настройка регулярного резервного копирования

1. Создайте скрипт резервного копирования (`backup.sh`):

```
bash
#!/bin/bash
BACKUP_DIR="/backup/$(date +%Y%m%d)"
mongodump --out $BACKUP_DIR
tar -czvf $BACKUP_DIR.tar.gz $BACKUP_DIR
rm -rf $BACKUP_DIR
```

2. Добавьте задание в `crontab` (ежедневно в 02:00):

```
bash
0 2 * * * /path/to/backup.sh
```

#### 6. Тестирование плана восстановления

1. Создайте тестовую среду (отдельный экземпляр MongoDB).
2. Восстановите данные из резервной копии в тестовую среду.
3. Проверьте целостность данных:
  - сравните количество документов;
  - проверьте ключевые коллекции;
  - выполните тестовые запросы.
4. Зафиксируйте время восстановления.

#### 7. Дополнительные меры безопасности

1. Шифрование резервных копий:

```
bash
```

```
tar -czvf - /backup/data | openssl enc -aes-256-cbc -out backup.tar.gz.enc
```

2. **Хранение копий вне площадки** (облако, внешний диск):

```
bash
```

```
rsync -avz /backup/ user@remote-server:/backup/
```

3. **Контроль доступа к резервным копиям** (права файловой системы, шифрование).

### **Рекомендации по плану DR**

1. **Правило 3-2-1:**

- 3 копии данных;
- 2 разных носителя;
- 1 копия вне площадки.

2. **Многорегиональные развёртывания** — размещайте реплики в разных географических зонах.

3. **Регулярное тестирование** — проводите учения по восстановлению не реже раза в квартал.

4. **Автоматизация** — используйте скрипты и CI/CD для резервного копирования и отработки отказа.

5. **Документация** — зафиксируйте процедуры восстановления и распределите роли в команде.

### **Контрольные вопросы**

1. Какие компоненты входят в систему аварийного восстановления MongoDB?

2. В чём отличие `mongodump` от `mongoexport`?

3. Как настроить набор реплик для высокой доступности?

4. Какие параметры `mongorestore` позволяют ускорить восстановление?

5. Почему важно хранить резервные копии вне площадки?

### **Отчёт по работе**

Подготовьте отчёт, включающий:

- описание настроенной инфраструктуры (репликация, бэкапы);
- скрипты резервного копирования и восстановления;
- результаты тестирования плана DR;
- рекомендации по улучшению устойчивости системы.

### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### **Содержание отчёта**

1. Цель.

2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## Практическая работа 50 Разработка и тестирование сценария восстановления Oracle Database после сбоя.

**Цель работы** отработать навыки планирования и выполнения восстановления базы данных Oracle в условиях имитации сбоя, освоить ключевые команды и сценарии резервного копирования/восстановления.

### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### **Ход практического занятия:**

• Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

• Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.

- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### **Теоретические и учебно-методические материалы по теме практической работы**

#### **Оборудование и ПО**

- Сервер с установленной Oracle Database (версия 12c/19c/21c).
- Клиентские инструменты: SQL \*Plus, RMAN (Recovery Manager).
- Доступ к файловой системе (резервные копии, архивные журналы).
- Среда для имитации сбоев (например, отключение дисков, удаление файлов).

#### **Теоретическая справка**

Основные механизмы восстановления Oracle:

- **RMAN** — встроенный инструмент для резервного копирования и восстановления.
- **Архивные журналы (Archive Logs)** — обеспечивают восстановление до конкретной точки времени.
- **Управляющие файлы (Control Files)** — содержат метаданные о структуре БД.
- **Файлы данных (Datafiles)** — хранят пользовательские данные.
- **SPFILE/PFILE** — файлы параметров инициализации.

Типы восстановления:

- Полное (до последней транзакции).
- Неполное (до определённой точки времени/SCN).
- Восстановление отдельных табличных пространств или файлов.

#### **Ход работы**

## 1. Подготовка резервных копий

**Шаг 1.1.** Включите режим архивирования:

```
sql
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT;
SQL> ALTER DATABASE ARCHIVELOG;
SQL> ALTER DATABASE OPEN;
```

**Шаг 1.2.** Создайте полную резервную копию через RMAN:

```
bash
rman target /
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;
Сохраните копию на отдельный носитель.
```

**Шаг 1.3.** Создайте резервную копию управляющего файла:

```
sql
SQL> ALTER DATABASE BACKUP CONTROLFILE TO TRACE AS
'/backup/control_file.trc';
```

## 2. Имитация сбоя

**Сценарий 1:** Удаление файлов данных табличного пространства `USERS`.

```
bash
rm /u01/app/oracle/oradata/ORCL/users01.dbf
```

**Сценарий 2:** Потеря управляющих файлов.

```
bash
rm /u01/app/oracle/fast_recovery_area/ORCL/control01.ctl
```

**Сценарий 3:** Сбой экземпляра (аварийное завершение работы).

```
sql
SQL> SHUTDOWN ABORT;
```

## 3. Восстановление по сценариям

**Сценарий 1. Восстановление файла данных**

1. Подключитесь как `SYSDBA`:

```
sql
SQL> CONNECT / AS SYSDBA;
```

2. Переведите БД в состояние `MOUNT`:

```
sql
SQL> SHUTDOWN IMMEDIATE;
SQL> STARTUP MOUNT;
```

3. Восстановите файл через RMAN:

```
bash
rman target /
RMAN> RESTORE DATAFILE '/u01/app/oracle/oradata/ORCL/users01.dbf';
RMAN> RECOVER DATAFILE '/u01/app/oracle/oradata/ORCL/users01.dbf';
```

4. Откройте БД:

```
sql
SQL> ALTER DATABASE OPEN;
```

**Сценарий 2. Восстановление управляющих файлов**

1. Получите скрипт создания управляющего файла из резервной копии (`control_file.trc`).

2. Отредактируйте скрипт, указав актуальные пути к файлам.

3. Выполните скрипт:

```
sql
SQL> @/backup/create_controlfile.sql
```

4. Восстановите БД:

```
sql
SQL> RECOVER DATABASE USING BACKUP CONTROLFILE;
SQL> ALTER DATABASE OPEN RESETLOGS;
```

### Сценарий 3. Восстановление после аварийного завершения

1. Запустите БД:

```
sql
SQL> STARTUP;
```

2. Oracle автоматически выполнит восстановление (instance recovery) при наличии архивных журналов.

#### 4. Тестирование целостности данных

**Шаг 4.1.** Проверьте статус БД:

```
sql
SQL> SELECT NAME, OPEN_MODE, LOG_MODE FROM V$DATABASE;
```

**Шаг 4.2.** Убедитесь, что все табличные пространства доступны:

```
sql
SQL> SELECT TABLESPACE_NAME, STATUS FROM DBA_TABLESPACES;
```

**Шаг 4.3.** Выполните тестовый запрос к ключевым таблицам:

```
sql
SQL> SELECT COUNT(*) FROM HR.EMPLOYEES;
```

**Шаг 4.4.** Проверьте отсутствие ошибок в журнале alert:

```
bash
tail -n 50 /u01/app/oracle/diag/rdbms/orcl/ORCL/trace/alert_ORCL.log
```

#### Отчёт по работе

Подготовьте отчёт, включающий:

1. Описание выполненных шагов по каждому сценарию.
2. Скриншоты команд и их вывода.
3. Анализ возникших ошибок (если были) и способы их устранения.
4. Время, затраченное на восстановление для каждого сценария.
5. Выводы о надёжности резервных копий и эффективности RMAN.

#### Контрольные вопросы

1. В чём разница между **RESTORE** и **RECOVER** в RMAN?
2. Зачем нужен режим **ARCHIVELOG**?
3. Что означает **RESETLOGS** при открытии БД?
4. Как восстановить отдельный табличный пространство без остановки БД?
5. Какие файлы критически важны для восстановления?

#### Критерии оценки

- Корректность выполнения всех сценариев.
- Полнота отчёта и анализ результатов.
- Умение интерпретировать сообщения об ошибках.
- Знание теоретических аспектов восстановления.

---

**Примечание:** Для реальной эксплуатации настройте автоматическое резервное копирование (например, через RMAN + cron) и храните копии на геораспределённых носителях.

#### Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### Содержание отчёта

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и

кодированию.

**Ответить на контрольные вопросы.(устно)**  
**Сформулировать выводы по результатам работы.**  
**Сдать и защитить работу.**

## **Тема 1.6. Мониторинг событий, возникающих в процессе работы баз данных**

### **Практическая работа 51 Настройка и использование утилиты MySQL Performance Schema для мониторинга работы базы данных.**

**Цель работы:** освоить настройку и практическое применение MySQL Performance Schema для комплексного мониторинга производительности СУБД: сбора статистики, анализа узких мест, оптимизации запросов и ресурсов.

#### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

#### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

#### **Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

#### **Теоретические и учебно-методические материалы по теме практической работы**

##### **Теоретические основы**

**Performance Schema** — встроенная в MySQL система инструментирования, которая:

- собирает данные о событиях выполнения сервера (ожидания, блокировки, ввод-вывод, память и др.);

- хранит данные в специальных таблицах базы `performance_schema`;
- позволяет динамически настраивать сбор данных через SQL-запросы;

- оказывает минимальное влияние на производительность при разумной конфигурации.

Ключевые особенности:

- таблицы `performance_schema` — временные (не используют дисковое хранилище);
- конфигурация меняется «на лету» без перезагрузки сервера;
- данные доступны через обычные `SELECT`-запросы.

### Шаг 1. Проверка доступности и включение Performance Schema

1. Убедитесь, что Performance Schema активна:

```
sql
SHOW VARIABLES LIKE 'performance_schema';
```

Ожидаемый результат: `Value = ON`.

2. Если отключена, добавьте в конфигурационный файл MySQL (`my.cnf` / `my.ini`):

```
ini
[mysqld]
performance_schema=ON
```

Перезапустите сервер.

3. Проверьте основные параметры:

```
sql
SHOW VARIABLES LIKE 'performance_schema%';
```

Обратите внимание на лимиты (например, `performance_schema_max_table_instances`).

### Шаг 2. Обзор ключевых таблиц Performance Schema

Выполните запросы, чтобы ознакомиться с основными группами таблиц:

1. События ожиданий (waits):

```
sql
SHOW TABLES FROM performance_schema LIKE 'events_waits%';
```

- `events_waits_current` — текущие ожидания;
- `events_waits_history` — история недавних ожиданий;
- `events_waits_summary_by_..._global` — сводные статистики.

2. События операторов (statements):

```
sql
SHOW TABLES FROM performance_schema LIKE 'events_statements%';
```

- `events_statements_current` — текущие SQL-запросы;
- `events_statements_summary_by_digest` — сводка по шаблонам запросов

(нормализованным).

3. Ресурсы и объекты:

- `table_io_waits_summary_by_table` — I/O по таблицам;
- `file_summary_by_event_name` — файловые операции;
- `memory_summary_by_event_name` — потребление памяти.

### Шаг 3. Настройка уровня детализации сбора данных

1. Управляйте включением инструментов (instruments):

```
sql
-- Включить сбор данных по мьютексам
UPDATE setup_instruments
SET ENABLED = 'YES', TIMED = 'YES'
WHERE NAME LIKE 'wait/synch/mutex/%';

-- Отключить редко нужные инструменты
UPDATE setup_instruments
SET ENABLED = 'NO'
WHERE NAME LIKE 'stage/%';
```

2. Настройте потребители (consumers) — кто получает данные:

```
sql
UPDATE setup_consumers
SET ENABLED = 'YES'
WHERE NAME LIKE '%history%'; -- Включить историю событий
```

3. Ограничьте размер буферов (чтобы не перегружать память):

```
sql
-- Например, уменьшить историю ожиданий
SET GLOBAL performance_schema_events_waits_history_size = 100;
```

### Шаг 4. Сбор и анализ данных о производительности

#### Пример 1. Топ-10 самых долгих запросов

```
sql
SELECT
  DIGEST_TEXT AS query,
  COUNT_STAR AS exec_count,
  SUM_TIMER_WAIT AS total_latency,
  MAX_TIMER_WAIT AS max_latency,
  AVG_TIMER_WAIT AS avg_latency
FROM performance_schema.events_statements_summary_by_digest
ORDER BY SUM_TIMER_WAIT DESC
LIMIT 10;
```

#### Пример 2. I/O по таблицам

```
sql
SELECT
  OBJECT_SCHEMA AS db,
  OBJECT_NAME AS table,
  COUNT_READ AS reads,
  SUM_NUMBER_OF_BYTES_READ AS read_bytes,
  COUNT_WRITE AS writes,
  SUM_NUMBER_OF_BYTES_WRITE AS write_bytes
FROM performance_schema.table_io_waits_summary_by_table
ORDER BY read_bytes + write_bytes DESC;
```

#### Пример 3. Ожидания по типам

```

sql
SELECT
    EVENT_NAME,
    COUNT_STAR AS total_waits,
    SUM_TIMER_WAIT AS total_latency,
    AVG_TIMER_WAIT AS avg_latency
FROM performance_schema.events_waits_summary_global_by_event_name
ORDER BY total_latency DESC;

```

### Шаг 5. Мониторинг в реальном времени

1. Текущие выполняющиеся запросы:

```

sql
SELECT * FROM performance_schema.events_statements_current;

```

2. Активные потоки и их состояние:

```

sql
SELECT * FROM performance_schema.threads;

```

3. Блокировки и взаимные ожидания:

```

sql
SELECT * FROM performance_schema.metadata_locks;

```

### Шаг 6. Автоматизация сбора метрик

Создайте скрипт (например, на Bash/Python), который:

1. Выполняет ключевые запросы из Шага 4.
2. Сохраняет результаты в CSV/JSON.
3. Запускает сбор каждые 5–15 минут.

Пример Bash-скрипта (`collect_perfddata.sh`):

```

bash
#!/bin/bash
mysql -u user -p -e "
SELECT NOW(), DIGEST_TEXT, COUNT_STAR, SUM_TIMER_WAIT
FROM performance_schema.events_statements_summary_by_digest
ORDER BY SUM_TIMER_WAIT DESC LIMIT 5
" > /var/log/mysql_perf_$(date +%Y%m%d_%H%M).csv

```

### Шаг 7. Оптимизация на основе данных

На основе собранных метрик:

1. Найдите запросы с высокой `SUM_TIMER_WAIT` — оптимизируйте их (индексы, перепись).
2. Выявите таблицы с большим I/O — проверьте шардирование, кэширование.
3. Проанализируйте ожидания (`events_waits`) — устраните узкие места (например, конфликты блокировок).

### Контрольные вопросы

1. Как проверить, включена ли Performance Schema?
2. Какие таблицы содержат сводную статистику по SQL-запросам?
3. Как временно отключить сбор данных по определённому типу событий?
4. Какой запрос покажет топ-5 самых долгих запросов за последний час?
5. Как ограничить объём памяти, используемый Performance Schema?

## **Отчёт по работе**

Подготовьте отчёт, включающий:

1. Скриншоты выполненных запросов и их результатов.
2. Анализ выявленных проблем (например, медленный запрос, высокий I/O).
3. Предложения по оптимизации на основе данных Performance Schema.
4. Вывод о полезности инструмента для мониторинга.

### **Дополнительные ресурсы**

- Официальная документация MySQL: раздел «Performance Schema».
- Утилиты: `sys schema` (дополняет Performance Schema удобными вьюхами).
- Инструменты визуализации: Grafana + MySQL Data Source для графиков.

### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы. (устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 52 Использование утилиты pg\_stat\_activity в PostgreSQL для отслеживания активных соединений и запросов.**

***Цель работы:*** освоить методы мониторинга активных сеансов и запросов в PostgreSQL с помощью системного представления

#### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

#### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

#### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.

- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

#### Теоретические основы

`pg_stat_activity` — это **системное представление** (view) в PostgreSQL, которое предоставляет актуальную информацию о текущих сессиях и выполняемых запросах на сервере БД.

#### Ключевые поля представления:

- `pid` — идентификатор процесса (PID) серверного процесса;
- `datid` — OID базы данных, с которой связана сессия;
- `datname` — имя базы данных;
- `username` — имя пользователя, подключённого к сессии;
- `client_addr` — IP-адрес клиента;
- `application_name` — название клиентского приложения;
- `query` — текст выполняемого запроса;
- `state` — состояние сессии (например, `active`, `idle`);
- `wait_event_type` и `wait_event` — тип и имя события, которого ожидает сессия;
- `xact_start` — время начала текущей транзакции;
- `query_start` — время начала выполнения текущего запроса.

#### Основные состояния сессий (`state`):

- `active` — выполнение запроса;
- `idle` — ожидание новой команды от клиента;
- `idle in transaction` — сессия внутри транзакции, но не выполняет запрос;
- `idle in transaction (aborted)` — транзакция содержит ошибку;
- `fastpath function call` — выполнение fast-path функции;
- `disabled` — отключён параметр `track_activities`.

#### Практические задания

##### 1. Получение списка всех активных запросов

```
sql
SELECT
  pid AS process_id,
  username AS username,
  datname AS database_name,
```

```
query AS active_query,  
state,  
now() - query_start AS query_duration  
FROM pg_stat_activity  
WHERE state = 'active'  
ORDER BY query_duration DESC;
```

**Что показывает:** PID, имя пользователя, базу данных, текст активного запроса, его состояние и длительность выполнения.

## 2. Просмотр всех текущих сессий

```
sql  
SELECT  
pid,  
username,  
datname,  
client_addr,  
application_name,  
state,  
query,  
now() - xact_start AS transaction_duration,  
now() - query_start AS query_duration  
FROM pg_stat_activity  
ORDER BY query_duration DESC;
```

**Что показывает:** полную информацию о всех подключениях, включая клиентские адреса и приложения.

## 3. Подсчёт сессий по состояниям

```
sql  
SELECT  
state,  
COUNT(*) AS session_count  
FROM pg_stat_activity  
GROUP BY state  
ORDER BY session_count DESC;
```

**Что показывает:** количество сессий в каждом состоянии (active, idle и др.).

## 4. Поиск долго выполняемых запросов

```
sql  
SELECT  
pid,  
username,  
datname,  
query,  
now() - query_start AS duration  
FROM pg_stat_activity  
WHERE state = 'active'  
AND now() - query_start > INTERVAL '5 minutes'  
ORDER BY duration DESC;
```

**Что показывает:** запросы, выполняющиеся дольше 5 минут.

### 5. Определение сессий с длительными транзакциями

```
sql
SELECT
    pid,
    username,
    datname,
    query,
    now() - xact_start AS transaction_duration
FROM pg_stat_activity
WHERE now() - xact_start > INTERVAL '10 minutes'
ORDER BY transaction_duration DESC;
```

**Что показывает:** сессии с транзакциями, открытыми более 10 минут.

### 6. Фильтрация по пользователю или базе данных

```
sql
SELECT *
FROM pg_stat_activity
WHERE username = 'your_username' -- или datname = 'your_database'
ORDER BY query_start;
```

**Что показывает:** все сессии указанного пользователя или для указанной базы данных.

#### Дополнительные возможности

#### 1. Прерывание сессии (осторожно!):

```
sql
SELECT pg_terminate_backend(pid);
```

Замените `pid` на идентификатор процесса из `pg_stat_activity`.

#### 2. Отмена запроса (менее радикально):

```
sql
SELECT pg_cancel_backend(pid);
```

#### 3. Мониторинг ожиданий:

```
sql
SELECT
    pid,
    username,
    wait_event_type,
    wait_event,
    query
FROM pg_stat_activity
WHERE wait_event IS NOT NULL;
```

#### Рекомендации по использованию

1. **Регулярный мониторинг:** запускайте запросы из разделов 1–5 периодически для контроля нагрузки.

2. **Анализ производительности:** ищите запросы с большим `query_duration` — они могут требовать оптимизации.

3. **Безопасность:** проверьте `client_addr` и `application_name` для выявления неавторизованных подключений.

4. **Резервное копирование:** перед прерыванием сессий (`pg_terminate_backend`) убедитесь, что это не нарушит работу критически важных процессов.

5. **Логирование:** сохраняйте результаты мониторинга для анализа трендов.

#### **Контрольные вопросы**

1. Что представляет собой `pg_stat_activity`?
2. Какие основные поля этого представления и что они означают?
3. Как отличить активную сессию от idle?
4. Как найти запросы, выполняющиеся более 5 минут?
5. В чём разница между `pg_cancel_backend` и `pg_terminate_backend`?

#### **Вывод**

В ходе работы освоены:

- базовые запросы к `pg_stat_activity`;
- методы фильтрации сессий по состоянию, длительности и другим параметрам;
- способы прерывания проблемных сессий.

`pg_stat_activity` — мощный инструмент для:

- диагностики производительности;
- выявления «зависших» запросов;
- контроля безопасности подключений;
- оптимизации работы сервера PostgreSQL.

#### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

##### **Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## **Практическая работа 53 Мониторинг событий в Microsoft SQL Server с помощью SQL Server Profiler.**

**Цель работы** освоить инструмент SQL Server Profiler для мониторинга и анализа событий в Microsoft SQL Server: научиться создавать трассировки, настраивать фильтры, собирать и анализировать данные о запросах и операциях.

### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
  - Предъявить преподавателю результаты работы.
  - Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
  - Подготовить ответы на контрольные вопросы.

### ***Теоретические и учебно-методические материалы по теме практической работы***

#### **Теоретические сведения**

**SQL Server Profiler** — графический инструмент для трассировки (записи) событий в SQL Server. Позволяет:

- отслеживать T-SQL-запросы, вызовы хранимых процедур, логины, ошибки и др.;
- анализировать производительность запросов (время выполнения, число чтений/записей);
- сохранять трассировки в файлы (`.trc`) или таблицы SQL;
- воспроизводить трассировки для отладки.

#### **Ключевые понятия:**

- **Событие (Event)** — действие в SQL Server (например, `SQL:BatchCompleted`, `RPC:Completed`, `Audit Login`).
- **Класс события (Event Class)** — тип отслеживаемого действия (например, выполнение запроса, подключение пользователя).
- **Трассировка (Trace)** — запись событий по заданным критериям.
- **Фильтр** — ограничение собираемых данных (по базе, пользователю, длительности и т. п.).

#### **Важные замечания:**

- Профилирование создаёт нагрузку на сервер. Не запускайте длительные трассировки на продуктивных серверах в рабочее время.

- Для долговременного мониторинга используйте серверные трассировки (**SQL Server Trace**) вместо Profiler.

### Порядок выполнения

#### 1. Запуск SQL Server Profiler

- Через меню «Пуск»:

Пуск → Все программы → Microsoft SQL Server → Средства обеспечения производительности → SQL Server Profiler.

- Из **SQL Server Management Studio (SSMS)**:

Меню **Инструменты** → SQL Server Profiler.

#### 2. Подключение к серверу

- В диалоговом окне «Подключение к серверу» укажите:
  - имя сервера SQL Server;
  - метод аутентификации (Windows/SQL Server);
  - логин и пароль (если требуется).
- Нажмите «Соединить».

#### 3. Настройка трассировки

1. В окне «Свойства трассировки» перейдите на вкладку «Общие»:

- Задайте **имя трассировки** (например, **Monitoring\_Trace**).

- Выберите шаблон:

- **Standard** — базовый набор событий;
- **T-SQL** — только T-SQL-запросы;
- **T-SQL Duration** — запросы с временем выполнения;
- **T-SQL SPs** — вызовы хранимых процедур.

- Определите способ сохранения:

- **Файл (.trc)** — для последующего анализа;
- **Таблица SQL** — для запросов к данным трассировки.

- Установите «**Время остановки трассировки**» (опционально).

2. Перейдите на вкладку «Выбор событий»:

- Отметьте нужные классы событий (например):

- **SQL:BatchStarting** — начало выполнения пакета запросов;
- **SQL:BatchCompleted** — завершение пакета запросов;
- **RPC:Completed** — вызов хранимой процедуры;
- **Audit Login** / **Audit Logout** — подключения/отключения

пользователей;

- **Exception** — ошибки;
- **Deadlock** — взаимоблокировки.

- Включите «**Показать все события**» и «**Показать все столбцы**» для полного списка.

- Снимите флажки с ненужных событий (например, `Audit Server`).
- 3. Настройте фильтры (кнопка «**Фильтры столбцов**»):
  - **DatabaseName** — фильтр по имени базы данных (например, `LIKE 'MyDB'`).
  - **Duration** — минимальное время выполнения (например, `>= 1000` мс).
  - **LoginName** — фильтр по пользователю.
  - **TextData** — поиск по тексту запроса (например, `LIKE '%SELECT%'`).

#### 4. Запуск и сбор данных

1. Нажмите «**Запустить**» (зелёная стрелка).
2. Выполните интересующие действия в приложении или SSMS (например, запустите запрос).
3. Нажмите «**Остановить трассировку**» (красный квадрат) после сбора данных.

#### 5. Анализ результатов

- Просмотрите события в таблице:
  - **StartTime** — время начала события.
  - **Duration** — длительность (в микросекундах; для мс разделите на 1000).
  - **Reads / Writes** — число операций чтения/записи.
  - **CPU** — загрузка процессора.
  - **TextData** — текст запроса или процедуры.
- Используйте поиск (Ctrl+F) для нахождения конкретных событий.
- Сортируйте столбцы (щёлочок по заголовку) для выявления «тяжёлых»

запросов.

#### 6. Сохранение трассировки

- **В файл:**

`Файл → Сохранить как → Файл трассировки (.trc)`.

- **В таблицу SQL:**

`Файл → Сохранить как → Таблица трассировки` → выберите сервер, базу и имя таблицы.

#### 7. Воспроизведение трассировки (опционально)

1. Откройте сохранённый файл `.trc`.
2. В меню `Воспроизведение` выберите:
  - **Выполнить до курсора** — пошаговая отладка.
  - **Установить точку останова** — остановка на определённом событии.
3. Анализируйте порядок выполнения и время операций.

#### Пример сценария: анализ медленного запроса

1. Запустите Profiler и подключитесь к серверу.
2. В «**Свойствах трассировки**»:
  - Шаблон: `T-SQL Duration`.
  - Фильтр: `DatabaseName LIKE 'SalesDB'`, `Duration >= 5000` (5 сек).
3. Запустите трассировку.
4. Выполните медленный запрос в SSMS.
5. Остановите трассировку и найдите событие с максимальным `Duration`.
6. Изучите `TextData` и `Reads / Writes` для оптимизации запроса.

### **Контрольные вопросы**

1. Для чего используется SQL Server Profiler?
2. Какие основные классы событий можно отслеживать?
3. Почему не рекомендуется запускать Profiler на продуктивном сервере?
4. Как отфильтровать трассировку по конкретной базе данных?
5. В каких форматах можно сохранить результаты трассировки?

### **Отчёт по работе**

Подготовьте отчёт, включающий:

1. Скриншоты настроек трассировки (вкладки «**Общие**» и «**Выбор событий**»).
2. Пример собранной трассировки с 5–10 событиями.
3. Анализ одного запроса (время выполнения, число операций чтения).
4. Ответы на контрольные вопросы.
5. Выводы о применимости Profiler для диагностики производительности.

### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### **Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## **Практическая работа 54 Установка и настройка Prometheus для сбора метрик производительности базы данных MySQL.**

**Цель работы:** для установки и настройки Prometheus для сбора метрик производительности MySQL необходимо выполнить несколько шагов: установить и настроить MySQL Exporter, настроить Prometheus для сбора метрик, а также (опционально) визуализировать данные в Grafana.

#### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

#### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### *Ход практического занятия:*

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

Для установки и настройки Prometheus для сбора метрик производительности MySQL необходимо выполнить несколько шагов: установить и настроить MySQL Exporter, настроить Prometheus для сбора метрик, а также (опционально) визуализировать данные в Grafana. Ниже представлена подробная инструкция.

#### **1. Установка и настройка MySQL Exporter**

MySQL Exporter — это инструмент, который собирает метрики из MySQL и предоставляет их в формате, понятном для Prometheus.

##### **Шаги:**

1. **Скачайте последнюю версию MySQL Exporter** с официального репозитория (например, через GitHub или официальный сайт Prometheus).
2. **Создайте пользователя в MySQL** с правами для сбора метрик:

```
sql
CREATE USER 'exporter'@'localhost' IDENTIFIED BY 'your_password';
GRANT PROCESS, REPLICATION CLIENT, SELECT ON *.* TO 'exporter'@'localhost';
FLUSH PRIVILEGES;
```

3. **Настройте конфигурационный файл MySQL**

**Exporter** (например, `/etc/mysql_exporter.cnf`):

```
ini
[client]
user=mysqlexporter
password=your_secure_password
host=127.0.0.1
```

4. **Создайте systemd-сервис для MySQL**

**Exporter** (файл `/etc/systemd/system/mysql_exporter.service`):

```
ini
[Unit]
Description=Prometheus MySQL Exporter
After=network.target

[Service]
```

```
Type=simple
Restart=always
ExecStart=/usr/local/bin/mysqld_exporter \
  --config.my-cnf /etc/mysql_exporter.cnf \
  --collect.global_status \
  --collect.info_schema.innodb_metrics
```

### [Install]

```
WantedBy=multi-user.target
```

#### 5. **Перезагрузите systemd и запустите сервис:**

```
bash
sudo systemctl daemon-reload
sudo systemctl start mysql_exporter
sudo systemctl enable mysql_exporter
```

#### 2. **Настройка Prometheus для сбора метрик**

Prometheus должен быть настроен для периодического сбора метрик с MySQL Exporter.

##### Шаги:

##### 1. **Откройте файл конфигурации**

**Prometheus** (обычно `/etc/prometheus/prometheus.yml`).

##### 2. **Добавьте конфигурацию для MySQL Exporter** в секцию `scrape_configs`:

```
yaml
scrape_configs:
  - job_name: 'mysql'
    static_configs:
      - targets: ['localhost:9104']
```

Здесь `localhost:9104` — адрес и порт, на котором работает MySQL Exporter. Если он запущен на другом сервере, укажите соответствующий IP-адрес.

##### 3. **Перезапустите Prometheus:**

```
bash
sudo systemctl restart prometheus
```

#### 3. **Проверка работы**

Чтобы убедиться, что Prometheus собирает метрики, выполните следующие действия:

1. Откройте веб-интерфейс Prometheus по адресу `http://ваш_ip_адрес:9090`.

2. В разделе **Targets** проверьте, что цель `mysql` имеет статус `UP`.

3. В разделе **Graph** можно выполнить запросы для проверки метрик, например:

```
promql
mysql_global_status_threads_connected
```

#### 4. **Визуализация в Grafana (опционально)**

Для визуализации метрик можно использовать Grafana:

1. **Добавьте Prometheus как источник данных** в Grafana:
  - Перейдите в **Configuration** → **Data Sources**.

- Нажмите **Add data source**, выберите **Prometheus** и укажите URL сервера Prometheus (например, `http://localhost:9090`).

2. **Импортируйте готовый дашборд для MySQL** (например, с ID 7362 из официального репозитория Grafana):

- Перейдите в **Dashboards** → **Manage**.
- Нажмите **Import**, введите ID дашборда и выберите Prometheus как источник данных.

#### **Рекомендации по безопасности**

- Ограничьте доступ к порту 9104 (MySQL Exporter) с помощью firewall, чтобы только Prometheus мог подключаться.
- Используйте надёжные пароли для пользователя мониторинга MySQL.
- Рассмотрите использование TLS для шифрования трафика между MySQL Exporter и Prometheus.

#### **Дополнительные ресурсы**

- Официальный репозиторий MySQL Exporter: .
- Документация Prometheus: .
- Примеры дашбордов для Grafana: .

#### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

### **Практическая работа 55 Анализ журнала событий (log files) в Oracle Database для выявления ошибок и проблем.**

***Цель работы*** изучить, как управлять учётными записями пользователей и контролировать доступ к базам данных

#### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

#### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий

- Компьютер с подключением к сети Интернет

### *Ход практического занятия:*

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

Для установки и настройки Prometheus для сбора метрик производительности MySQL необходимо выполнить несколько шагов: установить и настроить MySQL Exporter, настроить Prometheus для сбора метрик, а также (опционально) визуализировать данные в Grafana. Ниже представлена подробная инструкция.

#### **1. Установка и настройка MySQL Exporter**

##### **Шаг 1: Создание пользователя в MySQL**

Создайте пользователя с правами для сбора метрик:

```
sql
CREATE USER 'exporter'@'localhost' IDENTIFIED BY 'your_secure_password';
GRANT PROCESS, REPLICATION CLIENT, SELECT ON *.* TO 'exporter'@'localhost';
FLUSH PRIVILEGES;
```

##### **Шаг 2: Скачивание и установка MySQL Exporter**

Скачайте последнюю версию экспортера с официального репозитория:

```
bash
wget
https://github.com/prometheus/mysqld_exporter/releases/download/v<версия>/mysqld_exporter-
<версия>.linux-amd64.tar.gz
```

```
tar -xvzf mysqld_exporter-<версия>.linux-amd64.tar.gz
```

Скопируйте исполняемый файл в `/usr/local/bin`:

```
bash
cp mysqld_exporter-<версия>.linux-amd64/mysqld_exporter /usr/local/bin
```

##### **Шаг 3: Настройка конфигурации**

Создайте файл конфигурации (например, `/etc/mysqld_exporter.cnf`) с содержимым:

```
ini
[client]
user=exporter
password=your_secure_password
host=127.0.0.1
```

## Шаг 4: Запуск MySQL Exporter

Запустите экспортер с нужными параметрами:

```
bash
mysqld_exporter --config.my-cnf=/etc/mysqld_exporter.cnf --collect.global_status --
collect.info_schema.innodb_metrics
```

Для автоматического запуска создайте systemd-сервис:

```
bash
sudo nano /etc/systemd/system/mysqld_exporter.service
```

Добавьте в файл:

```
ini
[Unit]
Description=MySQL Exporter

[Service]
User=prometheus
Group=prometheus
ExecStart=/usr/local/bin/mysqld_exporter --config.my-cnf=/etc/mysqld_exporter.cnf --
collect.global_status --collect.info_schema.innodb_metrics
```

### [Install]

```
WantedBy=multi-user.target
```

Перезагрузите systemd и запустите сервис:

```
bash
sudo systemctl daemon-reload
sudo systemctl start mysqld_exporter
sudo systemctl enable mysqld_exporter
```

## 2. Настройка Prometheus

Откройте файл конфигурации Prometheus (`/etc/prometheus/prometheus.yml`) и добавьте секцию для сбора метрик MySQL:

```
yaml
scrape_configs:
  - job_name: 'mysql'
    static_configs:
      - targets: ['localhost:9104']
    labels:
      alias: 'mysql'
```

Перезапустите Prometheus:

```
bash
sudo systemctl restart prometheus
```

Проверьте, что Prometheus собирает метрики, перейдя в веб-интерфейс по адресу `http://ваш_ip_адрес:9090` и введя запрос `up` в разделе Graph.

## 3. Визуализация в Grafana (опционально)

### Шаг 1: Добавление источника данных

В Grafana перейдите в *Configuration* → *Data Sources*, нажмите *Add data source*, выберите *Prometheus* и укажите URL сервера Prometheus (например, `http://localhost:9090`).

### **Шаг 2: Импорт дашборда**

Импортируйте готовый дашборд для MySQL. Например, можно использовать дашборд с ID 7362 из официального репозитория Grafana:

1. Перейдите в *Dashboards* → *Manage*.
2. Нажмите *Import*.
3. Введите ID дашборда (7362) и выберите Prometheus как источник данных.

После импорта вы получите готовый дашборд с метриками, такими как количество запросов, активные соединения, состояние репликации и др.

### **Рекомендации по безопасности**

- Ограничьте доступ к порту 9104 (по умолчанию используется MySQL Exporter) с помощью firewall, разрешив доступ только с сервера Prometheus.

- Используйте аутентификацию (например, через Nginx с базовой аутентификацией) для защиты эндпоинта экспортера.

- Регулярно обновляйте Prometheus и MySQL Exporter до последних версий.

Для более детальной настройки и анализа метрик можно изучить документацию MySQL Exporter и Prometheus, а также примеры запросов в PromQL.

### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### **Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

### **Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## **Практическая работа 56 Мониторинг запросов и идентификация "тяжёлых" операций в MySQL с использованием EXPLAIN.**

**Цель работы:** освоить методы анализа производительности SQL-запросов в MySQL посредством команды `EXPLAIN` и научиться выявлять «тяжёлые» (ресурсоёмкие) операции, требующие оптимизации.

### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ

ИИ

- Показать основные приемы эффективного использования возможностей

#### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

#### **Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
  - Предъявить преподавателю результаты работы.
  - Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
  - Подготовить ответы на контрольные вопросы.

#### **Теоретические и учебно-методические материалы по теме практической работы**

#### **Теоретические основы**

##### **Что такое EXPLAIN**

`EXPLAIN` — это инструкция MySQL, которая показывает **план выполнения** запроса без его реального исполнения. Она позволяет:

- понять, как MySQL собирается обрабатывать запрос;
- увидеть порядок соединения таблиц;
- определить, какие индексы используются (или не используются);
- оценить количество обрабатываемых строк;
- выявить потенциальные узкие места (full table scan, отсутствие индексов и т.

п.).

##### **Синтаксис**

```
sql
EXPLAIN [EXTENDED | PARTITIONS]
SELECT ... ;
```

- `EXTENDED` — добавляет детализированную информацию (например, столбец `filtered`).
- `PARTITIONS` — показывает информацию о партициях (если таблица партиционирована).

В MySQL 8.0+ доступен `EXPLAIN ANALYZE`, который выполняет запрос и выдаёт **реальные метрики времени и количества строк**.

##### **Основные столбцы вывода EXPLAIN**

1. `id` — порядковый номер шага в плане выполнения.

2. `select_type` — тип выборки (например, `SIMPLE`, `SUBQUERY`, `UNION`).
3. `table` — имя таблицы, к которой обращается шаг.
4. `partitions` — используемые партиции (если есть).
5. `type` — метод доступа к данным:
  - `system` / `const` — одна строка (оптимально);
  - `eq_ref` — уникальное соединение по индексу;
  - `ref` — не уникальное соединение по индексу;
  - `range` — диапазонный поиск по индексу;
  - `index` — обход индекса (лучше, чем таблица);
  - `ALL` — полный скан таблицы (плохо!).
6. `possible_keys` — индексы, которые *могут* быть использованы.
7. `key` — реально используемый индекс.
8. `key_len` — длина используемого ключа.
9. `ref` — столбцы/константы, используемые для поиска по индексу.
10. `rows` — оценочное количество строк, которые MySQL прочтёт.
11. `filtered` — процент строк, отфильтрованных  
условием `WHERE` (при `EXTENDED`).
12. `Extra` — дополнительная информация (например, `Using index`, `Using temporary`, `Using filesort`).

## Практические задания

### Задание 1. Анализ простого запроса

1. Создайте таблицу `employees`:

```
sql
CREATE TABLE employees (
  id INT PRIMARY KEY,
  name VARCHAR(100),
  department VARCHAR(100),
  salary DECIMAL(10, 2)
);
```

2. Заполните её тестовыми данными:

```
sql
INSERT INTO employees (id, name, department, salary) VALUES
(1, 'John Doe', 'Engineering', 60000.00),
(2, 'Jane Smith', 'Marketing', 55000.00),
(3, 'Michael Johnson', 'HR', 50000.00);
```

3. Выполните `EXPLAIN` для запроса:

```
sql
```

```
EXPLAIN SELECT * FROM employees;
```

4. Проанализируйте вывод:

- Какой метод доступа (`type`) используется?
- Сколько строк оценивается (`rows`)?
- Есть ли использование индекса (`key`)?

### Задание 2. Анализ запроса с условием WHERE

1. Добавьте индекс на столбец `department`:

```
sql
```

```
CREATE INDEX idx_department ON employees(department);
```

2. Выполните `EXPLAIN` для запроса:

```
sql
```

```
EXPLAIN SELECT * FROM employees WHERE department = 'Engineering';
```

3. Сравните с предыдущим выводом:

- Появился ли используемый индекс (`key`)?
- Как изменилось оценочное количество строк (`rows`)?

### Задание 3. Анализ JOIN-запроса

1. Создайте таблицу `orders`:

```
sql
```

```
CREATE TABLE orders (  
  order_id INT PRIMARY KEY,  
  employee_id INT,  
  order_date DATE  
);
```

2. Добавьте индекс на `employee_id`:

```
sql
```

```
CREATE INDEX idx_employee_id ON orders(employee_id);
```

3. Выполните `EXPLAIN` для JOIN-запроса:

```
sql
```

```
EXPLAIN  
SELECT e.name, o.order_date  
FROM employees e  
JOIN orders o ON e.id = o.employee_id  
WHERE e.department = 'Engineering';
```

4. Проанализируйте:

- Порядок соединения таблиц (`table`).
- Методы доступа (`type`) для каждой таблицы.
- Используемые индексы (`key`).

### Задание 4. Выявление «тяжёлых» операций

1. Выполните запрос с `ORDER BY` без индекса:

```
sql
```

```
EXPLAIN SELECT * FROM employees ORDER BY salary;
```

2. Обратите внимание на столбец `Extra`:

○ Если есть `Using filesort`, это означает, что сортировка выполняется во временной таблице (ресурсоёмко).

3. Добавьте индекс на `salary`:

```
sql
```

```
CREATE INDEX idx_salary ON employees(salary);
```

4. Повторно выполните `EXPLAIN` и убедитесь, что `Using filesort` исчез.

### Задание 5. Использование EXPLAIN ANALYZE (MySQL 8.0+)

1. Выполните:

```
sql
```

```
EXPLAIN ANALYZE
```

```
SELECT COUNT(*) FROM employees WHERE salary > 50000;
```

2. Сравните вывод с обычным `EXPLAIN`:

○ Появились ли реальные времена выполнения (`actual time`)?

○ Совпадают ли оценочные (`rows`) и реальные (`actual rows`) количества строк?

### Критерии оценки «тяжёлости» запроса

1. **Полный скан таблицы** (`type: ALL`) — признак отсутствия индекса.

2. **Использование временной таблицы** (`Extra: Using temporary`) — часто возникает при `GROUP BY` / `ORDER BY` без индексов.

3. **Сортировка во временной таблице** (`Extra: Using filesort`) — замедляет выполнение.

4. **Большое оценочное количество строк** (`rows`) — чем больше, тем хуже.

5. **Отсутствие используемого индекса** (`key: NULL`) при наличии возможных (`possible_keys`).

### Рекомендации по оптимизации

1. **Добавляйте индексы** на столбцы, используемые в `WHERE`, `JOIN`, `ORDER BY`, `GROUP BY`.

2. **\*\*Избегайте SELECT \*\*\*** — указывайте только нужные столбцы (особенно для больших таблиц).

3. **Используйте LIMIT** для ограничения выборки.

4. **Переписывайте сложные запросы** — иногда разбиение на подзапросы эффективнее.

5. **Анализируйте реальные метрики** с `EXPLAIN ANALYZE` в продакшене.

### Контрольные вопросы

1. Что показывает столбец `type` в выводе `EXPLAIN`? Приведите примеры хороших и плохих значений.

2. Как понять, что запрос использует индекс?

3. Что означает `Using filesort` в столбце `Extra`?
4. В чём отличие `EXPLAIN` от `EXPLAIN ANALYZE`?
5. Какие действия вы предпримете, если `EXPLAIN` показывает `type: ALL` для большой таблицы?

### **Отчёт по работе**

Подготовьте отчёт, включающий:

1. Скрипты создания таблиц и заполнения данными.
2. Выводы `EXPLAIN` для всех выполненных запросов.
3. Анализ «тяжёлых» операций (если обнаружены).
4. Предложения по оптимизации (если требуются).
5. Ответы на контрольные вопросы.

### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### **Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## **Практическая работа 57 Настройка алертинга (уведомлений) в PostgreSQL на основе событийных триггеров.**

**Цель работы:** освоить механизм создания событийных триггеров в PostgreSQL для отправки уведомлений (алертинга) при изменении данных в таблицах.

### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### **Ход практического занятия:**

• Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

- Руководствуясь методическими рекомендациями (инструкционная

карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.

- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

#### **Теоретическая часть**

В PostgreSQL уведомления реализуются через:

- **Триггерные функции** — PL/pgSQL-процедуры, выполняемые при событиях `INSERT`, `UPDATE`, `DELETE`.
- **Механизм `NOTIFY/LISTEN`** — встроенная система асинхронных сообщений между сессиями.
- **Функцию `pg_notify()`** — отправляет сообщение в указанный канал.

#### **Пошаговая инструкция**

##### **1. Создание тестовой таблицы**

```
sql
CREATE TABLE table_user (
  id SERIAL PRIMARY KEY,
  username VARCHAR(100) NOT NULL,
  email VARCHAR(100)
);
```

##### **2. Создание функции триггера**

```
sql
CREATE OR REPLACE FUNCTION notify_trigger()
RETURNS trigger AS $trigger$
DECLARE
  rec table_user;
  dat table_user;
  payload TEXT;
BEGIN
  -- Определяем текущую и предыдущую запись в зависимости от типа операции
  CASE TG_OP
    WHEN 'UPDATE' THEN
      rec := NEW;
      dat := OLD;
    WHEN 'INSERT' THEN
      rec := NEW;
    WHEN 'DELETE' THEN
      rec := OLD;
  ELSE
    RAISE EXCEPTION 'Unknown TG_OP: "%". Should not occur!', TG_OP;
```

```

END CASE;

-- Формируем JSON-сообщение с данными события
payload := json_build_object(
  'timestamp', CURRENT_TIMESTAMP,
  'action', LOWER(TG_OP),
  'db_schema', TG_TABLE_SCHEMA,
  'table', TG_TABLE_NAME,
  'record', row_to_json(rec),
  'old', row_to_json(dat)
);

-- Отправляем уведомление в канал 'db_event'
PERFORM pg_notify('db_event', payload);

```

```

RETURN rec;
END;
$trigger$ LANGUAGE plpgsql;

```

### 3. Создание триггера на таблицу

```

sql
CREATE TRIGGER user_notify
AFTER INSERT OR UPDATE OR DELETE ON table_user
FOR EACH ROW
EXECUTE PROCEDURE notify_trigger();

```

### 4. Проверка работы триггера (тестовые операции)

```

sql
-- Вставка записи
INSERT INTO table_user (username, email) VALUES ('alice', 'alice@example.com');

-- Обновление записи
UPDATE table_user SET email = 'alice_new@example.com' WHERE id = 1;

-- Удаление записи
DELETE FROM table_user WHERE id = 1;

```

### 5. Приём уведомлений в текущей сессии (тест)

В отдельной сессии PostgreSQL выполните:

```

sql
-- Подписаться на канал
LISTEN db_event;

-- Теперь при выполнении операций выше вы увидите сообщения в выводе psql

```

## Интеграция с внешними системами (на примере Node.js)

### 1. Установка зависимостей

```

bash
npm init -y

```

```
npm install express pg pg-listen
```

## 2. Код слушателя (index.js)

```
javascript
```

```
const express = require("express");
```

```
const createSubscriber = require("pg-listen");
```

```
(async function () {
```

```
  const server = express();
```

```
  const port = process.env.PORT || 8080;
```

```
  const subscriber = createSubscriber({
```

```
    connectionString:
```

```
    `postgres://${process.env.DB_USER}:${process.env.DB_PASS}@${process.env.DB_HOST}:${process.env.DB_PORT}/${process.env.DB_NAME}`
```

```
  });
```

```
  await subscriber.connect();
```

```
  await subscriber.listenTo("db_event");
```

```
  subscriber.notifications.on("db_event", (data) => {
```

```
    console.log("Получено уведомление:", data);
```

```
    // Здесь можно добавить логику алерта: отправка email, Slack, Telegram и т.п.
```

```
  });
```

```
  server.listen(port, () => {
```

```
    console.log("Сервер слушает на порту", port);
```

```
  });
```

```
})();
```

## 3. Запуск слушателя

```
bash
```

```
export DB_USER=your_user
```

```
export DB_PASS=your_password
```

```
export DB_HOST=localhost
```

```
export DB_PORT=5432
```

```
export DB_NAME=your_db
```

```
node index.js
```

### Контроль выполнения

#### 1. Проверьте наличие триггера:

```
sql
```

```
SELECT tname FROM pg_trigger WHERE tgreid = 'table_user'::regclass;
```

#### 2. Убедитесь, что уведомления приходят:

- В psql-сессии после `LISTEN db_event` должны появляться сообщения при

операциях с `table_user`.

- В консоли Node.js-приложения должны выводиться JSON-объекты с деталями событий.

### **Возможные проблемы и их решение**

- **Нет уведомлений в psql:** убедитесь, что выполнена команда `LISTEN db_event` в той же сессии.
- **Ошибки подключения Node.js:** проверьте параметры подключения к БД в переменных окружения.
- **Триггер не срабатывает:** проверьте, что триггер привязан к правильной таблице и событиям.

#### **Вывод**

В ходе работы вы:

- создали таблицу и триггерную функцию для отправки уведомлений;
- настроили отправку событий в канал `db_event`;
- реализовали внешний слушатель на Node.js для обработки алёртов.

Этот механизм позволяет строить системы мониторинга изменений данных, интеграции с мессенджерами, логирования и т.п.

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 58 Анализ блокировок и ожиданий в Microsoft SQL Server с помощью DMVs (Dynamic Management Views).**

***Цель работы:*** освоить методы диагностики блокировок и ожиданий в SQL Server с использованием **Dynamic Management Views (DMVs)**, научиться выявлять проблемные сессии и анализировать причины замедления выполнения запросов.

#### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### *Ход практического занятия:*

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

Освоить методы диагностики блокировок и ожиданий в SQL Server с использованием **Dynamic Management Views (DMVs)**, научиться выявлять проблемные сессии и анализировать причины замедления выполнения запросов.

#### **Теоретические основы**

**Блокировки (locks)** — механизм СУБД для обеспечения целостности данных при параллельном доступе.

**Ожидания (waits)** — состояние сессии, когда запрос не может продолжить выполнение из-за недоступности ресурса.

#### **Ключевые DMVs для анализа:**

- `sys.dm_os_waiting_tasks` — информация о заблокированных и блокирующих процессах;
- `sys.dm_exec_sessions` — данные об активных сессиях;
- `sys.dm_exec_requests` — детали о текущих запросах;
- `sys.dm_tran_locks` — сведения о текущих блокировках;
- `sys.dm_exec_sql_text` — текст выполняемого SQL-запроса;
- `sys.dm_exec_query_plan` — план выполнения запроса.

#### **Ход работы**

##### **Шаг 1. Базовый запрос для выявления блокировок**

Выполните следующий T-SQL-запрос:

```
sql
WITH [Blocking] AS (
    SELECT
        w.[session_id],
        s.[original_login_name],
        s.[login_name],
        w.[wait_duration_ms],
```

```

w.[wait_type],
r.[status],
r.[wait_resource],
w.[resource_description],
s.[program_name],
w.[blocking_session_id],
s.[host_name],
r.[command],
r.[percent_complete],
r.[cpu_time],
r.[total_elapsed_time],
r.[reads],
r.[writes],
r.[logical_reads],
r.[row_count],
q.[text],
q.[dbid],
p.[query_plan],
r.[plan_handle]
FROM [sys].[dm_os_waiting_tasks] w
INNER JOIN [sys].[dm_exec_sessions] s ON w.[session_id] = s.[session_id]
INNER JOIN [sys].[dm_exec_requests] r ON s.[session_id] = r.[session_id]
CROSS APPLY [sys].[dm_exec_sql_text](r.[plan_handle]) q
CROSS APPLY [sys].[dm_exec_query_plan](r.[plan_handle]) p
WHERE w.[session_id] > 50
      AND w.[wait_type] NOT IN ('DBMIRROR_DBM_EVENT',
'ASYNC_NETWORK_IO')
)
SELECT
b.[session_id] AS [WaitingSessionID],
b.[blocking_session_id] AS [BlockingSessionID],
b.[login_name] AS [WaitingUserSessionLogin],
s1.[login_name] AS [BlockingUserSessionLogin],
b.[original_login_name] AS [WaitingUserConnectionLogin],
s1.[original_login_name] AS [BlockingSessionConnectionLogin],
b.[wait_duration_ms] AS [WaitDuration],
b.[wait_type] AS [WaitType],
t.[request_mode] AS [WaitRequestMode],
UPPER(b.[status]) AS [WaitingProcessStatus],
UPPER(s1.[status]) AS [BlockingSessionStatus],
b.[wait_resource] AS [WaitResource],
t.[resource_type] AS [WaitResourceType],
t.[resource_database_id] AS [WaitResourceDatabaseID],
DB_NAME(t.[resource_database_id]) AS [WaitResourceDatabaseName],
b.[resource_description] AS [WaitResourceDescription],

```

```

b.[program_name] AS [WaitingSessionProgramName],
s1.[program_name] AS [BlockingSessionProgramName],
b.[host_name] AS [WaitingHost],
s1.[host_name] AS [BlockingHost],
b.[command] AS [WaitingCommandType],
b.[text] AS [WaitingCommandText],
b.[row_count] AS [WaitingCommandRowCount],
b.[percent_complete] AS [WaitingCommandPercentComplete],
b.[cpu_time] AS [WaitingCommandCPUTime],
b.[total_elapsed_time] AS [WaitingCommandTotalElapsedTime],
b.[reads] AS [WaitingCommandReads],
b.[writes] AS [WaitingCommandWrites],
b.[logical_reads] AS [WaitingCommandLogicalReads],
b.[query_plan] AS [WaitingCommandQueryPlan],
b.[plan_handle] AS [WaitingCommandPlanHandle]
FROM [Blocking] b
INNER JOIN [sys].[dm_exec_sessions] s1 ON b.[blocking_session_id] = s1.[session_id]
INNER JOIN [sys].[dm_tran_locks] t ON t.[request_session_id] = b.[session_id]
WHERE t.[request_status] = 'WAIT';

```

#### Что показывает запрос:

- `WaitingSessionID` — ID ожидающей сессии;
- `BlockingSessionID` — ID блокирующей сессии;
- `WaitDuration` — длительность ожидания (мс);
- `WaitType` — тип ожидания (например, `LCK_M_IX`);
- `WaitResource` — ресурс, на котором возникло ожидание;
- `WaitingCommandText` — текст SQL-запроса ожидающей сессии;
- `BlockingUserSessionLogin` — логин пользователя блокирующей сессии.

#### Шаг 2. Анализ активных блокировок

Запрос для просмотра текущих блокировок:

```

sql
SELECT
    resource_type,
    resource_database_id,
    (SELECT [name] FROM master.sys.databases WHERE database_id =
resource_database_id) AS [DBName],
    CASE resource_type
        WHEN 'OBJECT' THEN OBJECT_NAME(resource_associated_entity_id,
resource_database_id)
        ELSE NULL
    END AS ObjectName,
    resource_associated_entity_id,
    request_status,
    request_mode,

```

```
request_session_id,  
resource_description
```

```
FROM sys.dm_tran_locks;
```

#### Ключевые столбцы:

- `resource_type` — тип ресурса (DATABASE, OBJECT, PAGE и др.);
- `request_status` — статус запроса (`GRANT`, `WAIT`, `CONVERT`);
- `request_mode` — режим блокировки (например, `S` — Shared, `X` — Exclusive).

### Шаг 3. Диагностика длительных ожиданий

Запрос для выявления сессий с длительным ожиданием:

```
sql
```

#### SELECT

```
session_id,  
blocking_session_id,  
wait_time,  
wait_type,  
last_wait_type,  
wait_resource,  
transaction_isolation_level,  
lock_timeout
```

```
FROM sys.dm_exec_requests
```

```
WHERE blocking_session_id <> 0
```

```
AND wait_time > 5000; -- Ожидания дольше 5 секунд
```

### Шаг 4. Анализ плана запроса блокирующей сессии

Если известен `plan_handle` блокирующей сессии, получите её план выполнения:

```
sql
```

```
SELECT query_plan
```

```
FROM sys.dm_exec_query_plan(0x06000500...); -- Подставьте реальный plan_handle
```

#### Практические задания

##### 1. Выявите блокировки

Запустите запрос из Шага 1 и определите:

- сколько сессий находятся в состоянии ожидания;
- какие сессии являются блокирующими;
- типы ожиданий (`WaitType`).

##### 2. Проанализируйте ресурсы

Используя запрос из Шага 2, определите:

- какие объекты базы данных заблокированы;
- режимы блокировок (`request_mode`).

##### 3. Найдите длительные ожидания

Выполните запрос из Шага 3 и ответьте:

- какие сессии ждут дольше 10 секунд;
- каковы причины ожиданий (`wait_type`).

##### 4. Изучите планы запросов

Для одной из блокирующих сессий получите план выполнения (Шаг 4) и проанализируйте:

- есть ли операции, требующие больших ресурсов;
- возможные узкие места (например, сканирование таблиц).

### **Контрольные вопросы**

1. Что такое `wait_type` и как его интерпретировать?
2. В чём разница между `GRANT`, `WAIT` и `CONVERT` в `request_status`?
3. Как определить, какой объект базы данных заблокирован?
4. Какие действия можно предпринять для устранения блокировки?
5. Почему важно анализировать `plan_handle` блокирующей сессии?

### **Отчёт по работе**

Подготовьте отчёт, включающий:

1. Скриншоты результатов выполнения запросов.

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 59 Использование MongoDB Profiler для отслеживания производительности запросов.**

***Цель работы:*** освоить инструменты профилирования MongoDB для анализа производительности запросов, выявления «медленных» операций и оптимизации работы базы данных.

#### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

#### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

#### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.

- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### *Теоретические и учебно-методические материалы по теме практической работы*

#### Теоретические сведения

**MongoDB Profiler** — встроенный инструмент, собирающий детальную информацию о выполняемых операциях (запросы, вставки, обновления, удаления). Данные хранятся в специальной коллекции `system.profile` и включают:

- время выполнения операции (`millis`);
- тип операции (`op`);
- количество просмотренных документов (`nscannedObjects`, `nscanned`);
- использованные индексы (`keysExamined`);
- тип плана выполнения (`planSummary`: `IXSCAN` — поиск по индексу, `COLLSCAN` — полный просмотр коллекции);
- информацию о блокировках.

#### Ход работы

##### 1. Включение профилирования

Установите уровень профилирования и порог «медленных» запросов (например, 100 мс):

```
javascript
db.setProfilingLevel(1, { slowms: 100 })
```

##### Уровни профилирования:

- `0` — профилирование отключено (по умолчанию);
- `1` — регистрируются только операции дольше `slowms`;
- `2` — регистрируются все операции.

Проверьте текущий статус:

```
javascript
db.getProfilingStatus()
```

##### 2. Анализ собранных данных

Получите список записанных операций в удобочитаемом формате:

```
javascript
db.system.profile.find().pretty()
```

##### Ключевые поля для анализа:

- `op` — тип операции (`query`, `insert`, `update`, `remove`);

- `ns` — название коллекции;
- `millis` — время выполнения в миллисекундах;
- `nreturned` — количество возвращённых документов;
- `nscanned` — количество просмотренных документов;
- `planSummary` — стратегия выполнения (`IXSCAN`, `COLLSCAN`).

### 3. Выявление неэффективных запросов

Найдите запросы с полным просмотром коллекции (`COLLSCAN`), которые не используют индексы:

```
javascript
db.system.profile.find({
  "planSummary": "COLLSCAN",
  "op": "query"
}).sort({ "millis": -1 }).pretty()
```

Отсортируйте запросы по времени выполнения (самые медленные сверху):

```
javascript
db.system.profile.find({ "op": "query" })
  .sort({ "millis": -1 })
  .pretty()
```

### 4. Оптимизация запросов

#### а) Анализ использования индексов

Используйте метод `explain()` для понимания плана выполнения:

```
javascript
db.collection.find({ field: "value" }).explain("executionStats")
```

Ищите поля:

- `executionStats.executionStage.stage` —

`IXSCAN` (хорошо), `COLLSCAN` (плохо);

- `executionStats.totalKeysExamined` — количество проверенных ключей индекса;
- `executionStats.totalDocsExamined` — количество проверенных документов.

#### б) Создание индексов

Добавьте индекс для поля, используемого в запросе:

```
javascript
db.collection.createIndex({ field: 1 })
```

#### в) Оптимизация курсоров

Используйте методы `limit()`, `skip()`, `sort()` для снижения нагрузки:

```
javascript
db.collection.find()
  .sort({ field: 1 })
  .limit(10)
```

### 5. Отключение профилирования

После анализа отключите профилирование (чтобы избежать накладных расходов):

```
javascript
```

```
db.setProfilingLevel(0)
```

### **Контрольные вопросы**

1. Какие уровни профилирования существуют в MongoDB?
2. В какой коллекции хранятся данные профилировщика?
3. Как интерпретировать значение `planSummary: COLLSCAN`?
4. Какой метод используется для анализа плана выполнения запроса?
5. Почему важно ограничивать выборку данных (`limit()`) при работе с большими

коллекциями?

### **Задания для самостоятельной работы**

1. Включите профилирование с порогом 50 мс.
2. Выполните 5 различных запросов к вашей коллекции.
3. Проанализируйте собранные данные: найдите самый медленный запрос.
4. Оптимизируйте этот запрос (добавьте индекс, измените структуру).
5. Повторите запрос и сравните время выполнения до и после оптимизации.

### **Критерии оценки**

- Корректное включение/отключение профилирования: 2 балла.
- Умение анализировать данные из `system.profile`: 3 балла.
- Нахождение и оптимизация «медленного» запроса: 4 балла.
- Оформление отчёта с выводами: 1 балл.

**Итого:** 10 баллов.

### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### **Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## **Практическая работа 60 Настройка и тестирование автоматизированного сбора метрик базы данных с использованием Grafana.**

**Цель работы** изучить, как управлять учётными записями пользователей и контролировать доступ к базам данных

#### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### **Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### **Теоретические и учебно-методические материалы по теме практической работы**

Для выполнения практической работы по настройке и тестированию автоматизированного сбора метрик базы данных с использованием Grafana, необходимо интегрировать Grafana с системой сбора метрик (например, Prometheus) и настроить визуализацию данных. Ниже приведены ключевые шаги и рекомендации.

#### **1. Установка и настройка Prometheus**

Prometheus — система сбора метрик, которая будет получать данные от базы данных через экспортеры. Для мониторинга базы данных (например, PostgreSQL) потребуется соответствующий экспортер (например, `postgres_exporter`).

##### **Шаги:**

- Установите Prometheus на сервер мониторинга.
- Настройте экспортер для базы данных. Например, для PostgreSQL:
  - Создайте пользователя в PostgreSQL с правами на доступ к метрикам (`pg_monitor`).
  - Настройте `postgres_exporter`, указав параметры подключения к базе данных.
  - Добавьте конфигурацию в `prometheus.yml` для сбора метрик с экспортера (например, `scrape_configs` с указанием IP и порта экспортера).
- Перезапустите Prometheus для применения изменений.

#### **2. Подключение Prometheus к Grafana**

Grafana используется для визуализации метрик, собранных Prometheus.

##### **Шаги:**

1. Откройте Grafana по адресу `http://localhost:3000` (или другой порт, если он изменён).
2. Перейдите в раздел **Connections** → **Data sources** → **Add data source**.
3. Выберите **Prometheus** в качестве типа источника данных.
4. Укажите URL сервера Prometheus (например, `http://localhost:9090`).
5. Нажмите **Save & Test** для проверки подключения. При успешном подключении появится сообщение «Database connection OK».

#### **3. Создание дашборда в Grafana**

Дашборд — это набор панелей, отображающих метрики в виде графиков, таблиц и других визуализаций.

#### Шаги:

1. Перейдите в раздел **Dashboards** → **Manage** и нажмите **New Dashboard**.
2. Добавьте панель (Panel) и выберите тип графика (например, **Graph**).
3. В разделе **Query** выберите источник данных Prometheus.
4. Введите запрос для получения метрик. Например, для отображения загрузки

CPU:

```
promql
rate(node_cpu_seconds_total{mode="idle"}[5m])
```

5. Настройте визуальные параметры панели (цвет, оси, интервалы) и сохраните изменения.

Можно использовать готовые дашборды из каталога Grafana.com, импортировав их по ID.

#### 4. Тестирование и настройка метрик

- Проверьте, что метрики отображаются на дашборде. Для этого обновите страницу или используйте функцию **Refresh** в Grafana.
- Настройте дополнительные панели для других метрик (например, использование памяти, дискового пространства, количество запросов к базе данных).
- Используйте фильтры и агрегаты для анализа данных (например, `histogram_quantile` для расчёта перцентилей).

#### 5. Настройка алертов (опционально)

Алерты позволяют автоматически уведомлять о проблемах в системе.

##### Шаги:

1. Перейдите в раздел **Alerting** → **Notification channels** и добавьте канал уведомлений (например, Email или Slack).
2. В панели Grafana перейдите в раздел **Alert** и создайте новое правило.
3. Укажите условия срабатывания алерта (например, если загрузка CPU превышает 80% в течение 5 минут).

##### Рекомендации

- **Безопасность:** ограничьте доступ к интерфейсам Prometheus и Grafana через firewall или reverse proxy.
- **Резервное копирование:** регулярно создавайте резервные копии данных Prometheus (`/var/lib/prometheus`).
- **Масштабирование:** при росте инфраструктуры используйте сервис-дискавери (Consul, Kubernetes) вместо статических конфигураций.
- **Оптимизация:** анализируйте метрики для выявления узких мест и оптимизации работы базы данных.

Для более детальной настройки и решения возможных проблем рекомендуется обратиться к официальной документации Prometheus и Grafana, а также к примерам конфигураций в открытых источниках.

Если у вас есть конкретные вопросы по настройке или возникают ошибки, уточните их — помогу разобраться!

#### *Оформление результатов работы*

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

##### *Содержание отчёта*

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и

кодированию.

**Ответить на контрольные вопросы.(устно)**  
**Сформулировать выводы по результатам работы.**  
**Сдать и защитить работу.**

## **Тема 1.7. Протоколирование событий, возникающих в процессе работы баз данных**

### **Практическая работа 61 Настройка и анализ журнала ошибок (error log) в MySQL.**

**Цель работы:** освоить настройку, включение и анализ журнала ошибок MySQL (error log) для диагностики проблем сервера.

#### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

#### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

#### **Ход практического занятия:**

• Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.

- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

#### **Теоретические и учебно-методические материалы по теме практической работы**

#### **Теоретические сведения**

**Журнал ошибок MySQL (error log)** — основной лог, в который записываются:

- сообщения о запуске/остановке сервера `mysqld`;
- критические ошибки выполнения;
- предупреждения системы;
- информация о сбоях и исключительных ситуациях.

По умолчанию в большинстве дистрибутивов Linux журнал располагается в `/var/log/mysql/error.log` или `/var/log/mysqld.log`.

#### **Ход работы**

##### **1. Определение текущего местоположения и статуса error log**

Выполните в терминале:

```
bash
```

```
mysql -e "SHOW VARIABLES LIKE 'log_error'"
```

Или через командную строку:

```
bash
```

```
sudo grep -i log_error /etc/mysql/my.cnf /etc/mysql/mysql.conf.d/mysqld.cnf
```

## 2. Включение и настройка error log

**Шаг 1.** Откройте конфигурационный файл MySQL:

```
bash
```

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

**Шаг 2.** В секции `[mysqld]` добавьте/проверьте строки:

```
ini
```

```
[mysqld]
```

```
log_error = /var/log/mysql/mysql_error.log
```

**Шаг 3.** Для дополнительной детализации можно включить:

```
ini
```

```
log_warnings = 2
```

(уровень 2 — выводит все предупреждения)

**Шаг 4.** Сохраните файл и перезапустите MySQL:

```
bash
```

```
sudo systemctl restart mysql
```

## 3. Проверка работы журнала

**Способ 1.** Просмотр последних записей:

```
bash
```

```
sudo tail -n 50 /var/log/mysql/mysql_error.log
```

**Способ 2.** Мониторинг в реальном времени:

```
bash
```

```
sudo tail -f /var/log/mysql/mysql_error.log
```

**Способ 3.** Через MySQL-клиент:

```
sql
```

```
SELECT @@GLOBAL.log_error;
```

```
SHOW VARIABLES LIKE 'log_error%';
```

## 4. Анализ типичных записей журнала

Примеры сообщений и их интерпретация:

1. **Ошибка подключения:**

2. [Warning] IP address '192.168.1.100' could not be resolved: Name or service not known

→ Проверьте DNS-настройки или используйте IP вместо имени хоста.

3. **Нехватка памяти:**

4. [ERROR] Out of memory (Needed 123456 bytes)

→ Увеличьте `innodb_buffer_pool_size` в конфигурации.

5. **Сбой таблицы:**

6. [ERROR] Table './db/table' is marked as crashed and should be repaired

→ Выполните `REPAIR TABLE db.table;`

7. **Превышение времени ожидания:**

8. [Warning] Aborted connection 12345 to db: 'mydb' user: 'user' host: 'localhost' (Got timeout reading communication packets)

→ Проверьте `wait_timeout` и сеть.

9. **Ошибка репликации:**

10. [ERROR] Slave I/O: error connecting to master, Error\_code: 1045

→ Проверьте учётные данные репликации.

## 5. Ротация логов (опционально)

Для предотвращения разрастания файла настройте ротацию:

**Шаг 1.** Создайте конфигурационный файл:

```
bash
```

```
sudo nano /etc/logrotate.d/mysql-error
```

**Шаг 2.** Добавьте содержимое:

```
conf
/var/log/mysql/mysql_error.log {
    daily
    rotate 7
    compress
    missingok
    notifempty
    create 640 mysql mysql
}
```

**Шаг 3.** Примените:

```
bash
```

```
sudo logrotate -f /etc/logrotate.d/mysql-error
```

### Контрольные вопросы

1. Где по умолчанию хранится error log в MySQL под Linux?
2. Какая директива в `my.cnf` отвечает за путь к error log?
3. Как просмотреть последние 20 строк журнала ошибок?
4. Что означает ошибка «Out of memory» в логе?
5. Как включить детальное логирование предупреждений?

### Отчёт по работе

Подготовьте отчёт, включающий:

1. Скриншоты вывода команд из пунктов 2–4.
2. Примеры 3–5 реальных ошибок из вашего лога с анализом.
3. Описание выполненных действий по настройке ротации.
4. Ответы на контрольные вопросы.

### Дополнительные рекомендации

- Для продуктивных систем рассмотрите централизованный сбор логов (например, через `syslog-ng` или `journald`).
- Используйте инструменты анализа логов: `grep`, `awk`, `sed` или специализированные ПО (ELK Stack, Graylog).
- Регулярно проверяйте лог на наличие повторяющихся ошибок.

### Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### Содержание отчёта

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

**Практическая работа 62 Конфигурация и просмотр логов событий в PostgreSQL с использованием параметра `logging_collector`.**

**Цель работы:** освоить настройку механизма сбора логов в PostgreSQL через параметр `logging_collector`, научиться конфигурировать параметры ротации и хранения логов, а также просматривать и анализировать собранные данные.

**Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

**Ход практического занятия:**

• Для выполнения данного задания необходимо изучить теоретический материал по данной теме.

- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.

- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

**Теоретические и учебно-методические материалы по теме практической работы**

**Теоретическая часть**

`logging_collector` — это фоновый процесс PostgreSQL, который собирает сообщения, отправляемые в `stderr`, и перенаправляет их в журнальные файлы. Его использование предпочтительнее прямой записи в `syslog`, поскольку:

- предотвращает потерю сообщений (например, об ошибках динамического связывания);
- обеспечивает надёжную ротацию файлов;
- гарантирует, что сообщения не будут потеряны даже при высокой нагрузке.

**Практическая часть**

**Шаг 1. Включение `logging_collector`**

1. Откройте файл конфигурации PostgreSQL:

```
bash
```

```
sudo nano /etc/postgresql/<версия>/main/postgresql.conf
```

(замените `<версия>` на актуальную версию PostgreSQL).

2. Найдите параметр `logging_collector` и установите его в `on`:

```
3. logging_collector = on
```

4. Сохраните файл и перезапустите сервер:

```
bash
```

```
sudo systemctl restart postgresql
```

## Шаг 2. Настройка параметров ротации логов

В том же файле `postgresql.conf` настройте ключевые параметры:

1. **log\_directory** — директория для хранения логов (по умолчанию зависит от ОС):

```
2. log_directory = 'pg_log'
```

3. **log\_filename** — шаблон имени файла лога (использует формат `strftime`):

```
4. log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log'
```

*Пример:* `postgresql-2025-12-13_162146.log`.

5. **log\_rotation\_size** — максимальный размер файла перед ротацией (в КБ):

```
6. log_rotation_size = 10240 # 10 МБ
```

7. **log\_rotation\_age** — максимальное время жизни файла (в минутах):

```
8. log_rotation_age = 1440 # 24 часа
```

9. **log\_truncate\_on\_rotation** — перезаписывать файлы при ротации (вместо дописывания):

```
10. log_truncate_on_rotation = on
```

## Шаг 3. Дополнительные параметры логирования

Для детализации логов можно настроить:

1. **log\_min\_duration\_statement** — регистрировать запросы дольше указанного времени (в мс):

```
2. log_min_duration_statement = 1000 # 1 секунда
```

3. **log\_statement** — уровень детализации запросов:

```
4. log_statement = 'mod' # DDL + модификации (INSERT, UPDATE, DELETE)
```

*Варианты:* `none`, `ddl`, `mod`, `all`.

5. **log\_connections** и **log\_disconnections** — фиксация подключений/отключений:

```
6. log_connections = on
```

```
7. log_disconnections = on
```

## Шаг 4. Проверка работы logging\_collector

1. Убедитесь, что логи создаются в указанной директории:

```
bash
```

```
ls /var/log/postgresql/ # или путь из log_directory
```

2. Просмотрите последние записи:

```
bash
```

```
tail -f /var/log/postgresql/postgresql-<дата_время>.log
```

3. Проверьте статус сборщика через SQL:

```
sql
```

```
SELECT name, setting FROM pg_settings WHERE name = 'logging_collector';
```

## Шаг 5. Анализ логов

Типичные сценарии анализа:

1. **Поиск ошибок:**

```
bash
```

```
grep "ERROR" /var/log/postgresql/postgresql-*.log
```

2. **Анализ медленных запросов:**

```
bash
grep "duration:" /var/log/postgresql/postgresql-*.log | sort -n
```

### 3. Мониторинг подключений:

```
bash
grep "connection authorized" /var/log/postgresql/postgresql-*.log
```

### Контрольные вопросы

1. Для чего нужен параметр `logging_collector`?
2. Какие преимущества даёт использование `logging_collector` по сравнению с записью в `stderr`?
3. Как настроить ротацию логов по размеру и времени?
4. Какие параметры позволяют фильтровать записываемые запросы?
5. Где по умолчанию хранятся логи PostgreSQL в системах на базе

Debian/Red Hat?

### Возможные ошибки и их устранение

1. **Логи не создаются:**
  - проверьте, что `logging_collector = on`;
  - убедитесь в наличии прав на запись в `log_directory`.
2. **Файлы логов слишком большие:**
  - уменьшите `log_rotation_size` или `log_rotation_age`.
3. **Потеря сообщений при высокой нагрузке:**
  - увеличьте `log_rotation_size` для снижения частоты ротаций.

### Вывод

В ходе работы освоены:

- включение и базовая конфигурация `logging_collector`;
- настройка ротации логов;
- методы анализа собранных данных.

Правильное конфигурирование логирования позволяет:

- оперативно выявлять ошибки;
- оптимизировать производительность;
- вести аудит действий пользователей.

### Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### Содержание отчёта

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## Практическая работа 63 Настройка протоколирования аудита в Microsoft SQL Server с использованием Extended Events.

**Цель работы:** освоить настройку аудита событий в SQL Server посредством механизма **Extended Events** (XEvents) — лёгковесной системы мониторинга с минимальным потреблением ресурсов.

### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### **Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
  - Предъявить преподавателю результаты работы.
  - Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
  - Подготовить ответы на контрольные вопросы.

### **Теоретические и учебно-методические материалы по теме практической работы**

#### **Теоретические сведения**

**Extended Events** — это гибкий и производительный механизм сбора событий в SQL Server (начиная с версии 2008). В отличие от SQL Trace и SQL Server Profiler, XEvents:

- потребляет меньше ресурсов;
- поддерживает детальные фильтры и действия;
- позволяет сохранять данные в файлы `.xel` с ротацией;
- интегрируется с SSMS и T-SQL.

Основные компоненты сессии XEvents:

- **Event** — отслеживаемое событие (например, `sqlserver.login`, `sqlserver.error_reported`).
- **Action** — дополнительные данные, собираемые при событии (имя пользователя, приложение, хост и т.п.).
- **Target** — место сохранения данных (обычно `package0.event_file`).
- **Predicate (WHERE)** — условия фильтрации событий.

- **Session options** — настройки сессии (задержка, память, режим сохранения).

## Ход работы

### 1. Подготовка директории для файлов аудита

Перед созданием сессии убедитесь, что папка для файлов `.xel` существует.

Выполните в SSMS:

```
sql
EXEC xp_create_subdir 'C:\Audit';
GO
```

*Примечание:* замените путь на актуальный для вашего сервера.

### 2. Создание сессии аудита успешных и неудачных входов

Выполните скрипт, который создаёт сессию `LoginAudit_Success_Failed`:

```
sql
CREATE EVENT SESSION [LoginAudit_Success_Failed]
ON SERVER
ADD EVENT sqlserver.login
(
    ACTION (
        sqlserver.client_app_name,
        sqlserver.client_hostname,
        sqlserver.nt_username,
        sqlserver.server_principal_name,
        sqlserver.username,
        sqlserver.session_id
    )
),
ADD EVENT sqlserver.error_reported
(
    ACTION (
        sqlserver.client_app_name,
        sqlserver.client_hostname,
        sqlserver.nt_username,
        sqlserver.server_principal_name,
        sqlserver.username,
        sqlserver.session_id
    )
    WHERE ([error_number] = 18456) -- Ошибка входа
)
ADD TARGET package0.event_file
(
    SET filename = N'C:\Audit>LoginAudit_Success_Failed.xel',
    max_file_size = (10),      -- Макс. размер файла в МБ
    max_rollover_files = (10) -- Количество файлов ротации
)
WITH (
    MAX_MEMORY = 4096 KB,
```

```

EVENT_RETENTION_MODE = ALLOW_SINGLE_EVENT_LOSS,
MAX_DISPATCH_LATENCY = 30 SECONDS,
MAX_EVENT_SIZE = 0 KB,
MEMORY_PARTITION_MODE = NONE,
TRACK_CAUSALITY = OFF,
STARTUP_STATE = ON
);
GO

```

#### Пояснения к скрипту:

- `sqlserver.login` — фиксирует успешные входы.
- `sqlserver.error_reported` с фильтром `error_number = 18456` — фиксирует ошибки входа (неверный пароль, пользователь не существует и т. п.).
- `ACTION` — собирает дополнительную информацию о клиенте, пользователе и сессии.
- `event_file` — сохраняет данные в файлы с ротацией (до 10 файлов по 10 МБ).
- `STARTUP_STATE = ON` — сессия будет запускаться автоматически при старте сервера.

### 3. Запуск сессии

```

sql
ALTER EVENT SESSION [LoginAudit_Success_Failed] ON SERVER STATE = START;
GO

```

### 4. Проверка работы сессии

- В SSMS откройте *Object Explorer* → *Management* → *Extended Events* → *Sessions*.
- Найдите сессию `LoginAudit_Success_Failed` и кликните *Watch Live Data*.
- Попробуйте войти в SQL Server под разными учётными записями (в т. ч. с ошибочным паролем).
- В окне *Live Data* должны появиться события `login` и `error_reported`.

### 5. Анализ собранных данных

Чтобы прочитать данные из файлов `.xel`, используйте функцию `sys.fn_xe_file_target_read_file`:

```

sql
WITH event_data AS (
    SELECT CAST(event_data AS XML) AS data
    FROM sys.fn_xe_file_target_read_file(
        'C:\Audit\LoginAudit_Success_Failed*.xel',
        NULL, NULL, NULL
    )
)
SELECT
    data.value('(event/@name)[1]', 'varchar(50)') AS event_name,
    data.value('(event/data[@name="error_number"]/value)[1]', 'int') AS error_number,

```

```

        data.value('(event/action[@name="sqlserver.client_app_name"]/value)[1]', 'varchar(100)')
AS client_app,
        data.value('(event/action[@name="sqlserver.client_hostname"]/value)[1]', 'varchar(100)')
AS client_host,
        data.value('(event/action[@name="sqlserver.nt_username"]/value)[1]', 'varchar(100)') AS
nt_username,
        data.value('(event/action[@name="sqlserver.server_principal_name"]/value)[1]',
'varchar(100)') AS server_principal,
        data.value('(event/@timestamp)[1]', 'datetime2') AS timestamp
FROM event_data
ORDER BY timestamp DESC;

```

### Результат запроса:

- `event_name` — тип события (`login` или `error_reported`).
- `error_number` — код ошибки (для неудачных входов).
- `client_app`, `client_host` — приложение и хост клиента.
- `nt_username`, `server_principal` — учётные записи.
- `timestamp` — время события.

### 6. Остановка и удаление сессии (по необходимости)

```

sql
-- Остановка сессии
ALTER EVENT SESSION [LoginAudit_Success_Failed] ON SERVER STATE = STOP;
GO

-- Удаление сессии
DROP EVENT SESSION [LoginAudit_Success_Failed] ON SERVER;
GO

```

### Контрольные вопросы

1. В чём преимущество Extended Events перед SQL Trace?
2. Какие основные компоненты включает сессия Extended Events?
3. Как отфильтровать только ошибки входа в систему?
4. Где хранятся данные аудита и как их просмотреть?
5. Как автоматически запускать сессию при старте сервера?

### Отчёт по работе

В отчёте укажите:

1. Цель работы.
2. Пошаговые действия (с примерами скриптов).
3. Скриншоты из SSMS (список сессий, окно *Live Data*, результат запроса к `.xel`).
4. Ответы на контрольные вопросы.
5. Выводы (преимущества и ограничения XEvents).

### Дополнительные возможности

- **Фильтры по пользователям/приложениям:** добавьте `WHERE` с условиями на `sqlserver.username` или `sqlserver.client_app_name`.

- **Аудит других событий:** например, `sqlserver.sql_statement_completed` для отслеживания запросов.

- **Интеграция с SIEM:** файлы `.xel` можно парсить внешними системами мониторинга.

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 64 Включение и настройка логирования операций в MongoDB с использованием параметра `profilingLevel`.**

***Цель работы:*** освоить механизм профилирования (логирования) операций в MongoDB через настройку параметра `profilingLevel`, научиться включать, конфигурировать и анализировать данные профилировщика.

#### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

#### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

#### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.

- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).

- Подготовить ответы на контрольные вопросы.

## *Теоретические и учебно-методические материалы по теме практической работы*

### Теоретические основы

**Профилировщик MongoDB** — встроенный инструмент для сбора информации о выполняемых запросах к базе данных. Он фиксирует:

- CRUD-операции (создание, чтение, обновление, удаление);
- командные операции;
- время выполнения запросов;
- метаданные о контексте выполнения.

По умолчанию профилировщик **отключен** из-за потенциального влияния на производительность.

Данные записываются в коллекцию `system.profile` базы `admin`.

### Уровни профилирования (profilingLevel)

В MongoDB определены три уровня профилирования:

1. `0` — профилирование **отключено** (значение по умолчанию).

Никакие операции не записываются в лог.

2. `1` — логируются **только медленные операции**.

Записываются запросы, время выполнения которых превышает значение параметра `slowOpThresholdMs` (по умолчанию — 100 мс).

3. `2` — логируются **все операции** без исключения.

Фиксируются все запросы к БД, независимо от времени выполнения.

### Пошаговая инструкция

#### 1. Подключение к MongoDB

Откройте оболочку `mongo` или используйте клиент (например, MongoDB Compass,

Robo 3T):

```
bash
mongo
```

Перейдите в нужную базу данных (например, `test`):

```
javascript
use test
```

#### 2. Проверка текущего уровня профилирования

Выполните команду:

```
javascript
db.getProfilingLevel()
```

Результат:

- `0` — профилирование отключено;
- `1` — логируются медленные операции;
- `2` — логируются все операции.

Также можно получить подробную информацию:

```
javascript
db.runCommand({ getProfilingStatus: 1 })
```

Вывод включает:

- `was` — предыдущий уровень;
- `slowms` — пороговое время для «медленных» операций (в мс);
- `sampleRate` — доля запросов для логирования (по умолчанию `1.0` — все запросы).

### 3. Включение профилирования

Установите нужный уровень с помощью команды:

```
javascript
db.setProfilingLevel(<уровень>, { slowOpThresholdMs: <пороговое_время_мс> })
```

#### Примеры:

- Включить логирование всех операций:

```
javascript
db.setProfilingLevel(2)
```

- Включить логирование медленных операций (пороговое время — 200 мс):

```
javascript
db.setProfilingLevel(1, { slowOpThresholdMs: 200 })
```

- Отключить профилирование:

```
javascript
db.setProfilingLevel(0)
```

Ответ сервера при успешном выполнении:

```
json
{ "was": 0, "slowms": 100, "sampleRate": 1, "ok": 1 }
```

- `was` — предыдущий уровень;
- `slowms` — новое пороговое время;
- `ok: 1` — операция выполнена успешно.

### 4. Настройка порога «медленных» операций

Изменить пороговое время можно при установке уровня `1`:

```
javascript
db.setProfilingLevel(1, { slowOpThresholdMs: 150 })
```

Или отдельно (без смены уровня):

```
javascript
db.adminCommand({
  setParameter: 1,
  slowOpThresholdMs: 150
})
```

### 5. Просмотр логов профилировщика

Данные хранятся в коллекции `system.profile` базы `admin`. Переключитесь на неё:

```
javascript
use admin
```

Получите список последних записей (например, 5 самых свежих):

```
javascript
db.system.profile.find().sort({ ts: -1 }).limit(5)
```

**Ключевые поля в записях:**

- `op` — тип операции (`query`, `insert`, `update`, `remove`, `command`);
- `ns` — пространство имён (база.коллекция);
- `ms` — время выполнения (в миллисекундах);
- `ts` — временная метка операции;
- `client` — IP-адрес клиента;
- `query` — текст запроса (для операций чтения);
- `updateobj` — объект обновления (для `update`);
- `command` — команда (для командных операций).

Для удобного вывода можно отобразить конкретные поля:

```
javascript
db.system.profile.find(
  {},
  { op: 1, ns: 1, millis: 1, ts: 1, query: 1 }
).sort({ ts: -1 }).limit(5)
```

## 6. Фильтрация логов

Примеры запросов:

- Все операции дольше 500 мс:

```
javascript
db.system.profile.find({ millis: { $gt: 500 } }).sort({ ts: -1 })
```

- Все запросы к коллекции `users` в базе `test`:

```
javascript
db.system.profile.find({ ns: "test.users" }).sort({ ts: -1 })
```

- Все операции типа `update`:

```
javascript
db.system.profile.find({ op: "update" }).sort({ ts: -1 })
```

## 7. Отключение профилирования

Чтобы остановить логирование:

```
javascript
db.setProfilingLevel(0)
```

Это освободит ресурсы и прекратит запись в `system.profile`.

## 8. Очистка коллекции `system.profile`

После отключения профилирования можно удалить старые записи:

```
javascript
use admin
db.system.profile.drop()
```

**Внимание!** Команда `drop()` удаляет коллекцию полностью. Если нужно сохранить часть данных, используйте `deleteMany()` с фильтром.

#### **Рекомендации по использованию**

1. **Для диагностики производительности:**
  - Включите уровень `2` на короткое время, чтобы собрать полные данные.
  - Проанализируйте самые длинные операции.
  - Отключите профилирование после сбора данных.
2. **Для мониторинга в продакшене:**
  - Используйте уровень `1` с разумным `slowOpThresholdMs` (например, 200–500 мс).
  - Регулярно проверяйте логи на наличие аномалий.
3. **Оптимизация хранения:**
  - Коллекция `system.profile` может быстро расти при уровне `2`.
  - Рассмотрите ротацию логов или ограничение размера коллекции.
4. **Безопасность:**
  - Доступ к `system.profile` должен быть ограничен.
  - Не оставляйте профилирование включённым без необходимости.

#### **Контрольные вопросы**

1. Какой уровень профилирования включён по умолчанию?
2. Как проверить текущий уровень профилирования?
3. Какие данные записываются в `system.profile`?
4. Как отключить профилирование?
5. Как просмотреть самые медленные операции за последний час?

#### **Задания для самостоятельной работы**

1. Включите профилирование уровня `1` с порогом 150 мс.
2. Выполните несколько операций (вставка, запрос, обновление) и проверьте лог.
3. Переключите уровень на `2`, повторите операции, сравните лог.
4. Отключите профилирование и очистите коллекцию `system.profile`.
5. Напишите запрос, который выведет все операции типа `query` дольше 300 мс за последние 10 минут.

#### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### **Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

#### **Ответить на контрольные вопросы.(устно)**

*Сформулировать выводы по результатам работы.  
Сдать и защитить работу.*

## **Практическая работа 65 Настройка и просмотр журнала аудита (Audit Trail) в Oracle Database.**

**Цель работы** освоить включение и настройку профилирования (логирования) операций в MongoDB через параметр `profilingLevel`, научиться анализировать собранные данные.

### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### **Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### **Теоретические и учебно-методические материалы по теме практической работы**

#### **Теоретические сведения**

**Профилировщик MongoDB** (`MongoDB Profiler`) — инструмент для сбора детальной информации о командах, выполняемых на экземпляре `mongod`. Данные записываются в коллекцию `system.profile` каждой профилируемой базы данных.

#### **Важные особенности:**

- По умолчанию профилировщик **отключён** (уровень 0) из-за влияния на производительность и потребления памяти.
- Коллекция `system.profile` — ограниченная (`capped collection`), по умолчанию размером 1 МБ.

- Данные профилировщика могут содержать чувствительную информацию (тексты запросов).

### Уровни профилирования (profilingLevel)

#### 1. Уровень 0 (db.setProfilingLevel(0))

- Профилировщик выключен, данные не собираются.
- `mongod` записывает в лог только операции дольше

порога `slowOpThresholdMs` (по умолчанию 100 мс).

- **Значение по умолчанию.**

#### 2. Уровень 1 (db.setProfilingLevel(1))

- Собираются данные только по «медленным» операциям

(дольше `slowOpThresholdMs`).

- Пороговое значение можно изменить через параметр `slowOpThresholdMs` или команду `setParameter`.

#### 3. Уровень 2 (db.setProfilingLevel(2))

- Собираются данные по всем операциям БД.
- Максимально детальный лог, но наибольшая нагрузка на систему.

### Порядок выполнения работы

#### 1. Подключение к MongoDB

Откройте оболочку `mongosh` (или `mongo`) и подключитесь к нужному экземпляру MongoDB. Выберите базу данных, для которой хотите включить профилирование:

```
javascript
use myDatabase
```

#### 2. Проверка текущего уровня профилирования

Узнайте текущий уровень профилирования:

```
javascript
db.getProfilingLevel()
```

- Возвращает число: `0`, `1` или `2`.

Проверьте статус профилировщика:

```
javascript
db.getProfilingStatus()
```

- Показывает уровень, порог `slowms` и другие параметры.

#### 3. Включение профилирования

Установите нужный уровень профилирования. Примеры:

- **Только медленные операции** (порог 100 мс):

```
javascript
db.setProfilingLevel(1)
```

- **Все операции** (максимальный лог):

```
javascript
db.setProfilingLevel(2)
```

- **С изменением порога медленных операций** (например, 50 мс):

```
javascript
```

```
db.setProfilingLevel(1, 50)
```

○ Второй параметр — количество миллисекунд (`milliseconds`).

- **Отключение профилирования:**

```
javascript
```

```
db.setProfilingLevel(0)
```

#### 4. Просмотр собранных данных

Данные записываются в коллекцию `system.profile`. Чтобы увидеть лог:

- Вывести все записи:

```
javascript
```

```
db.system.profile.find().pretty()
```

- Найти операции дольше 1 секунды (1000 мс), отсортированные по времени

(новые первыми):

```
javascript
```

```
db.system.profile.find({ millis: { $gt: 1000 } }).sort({ ts: -1 }).pretty()
```

- Просмотреть последние 5 операций (помощник в оболочке):

```
javascript
```

```
show profile
```

#### 5. Анализ ключевых полей в `system.profile`

Основные поля записей:

- `op` — тип операции (`query`, `insert`, `update`, `remove`, `command`).

- `ns` — пространство имён (база.коллекция).

- `millis` — длительность операции в миллисекундах.

- `ts` — временная метка операции.

- `client` — адрес клиента.

- `query` — текст запроса (если есть).

- `nreturned` — количество возвращённых документов.

- `nscanned` — количество просканированных документов.

#### 6. Изменение размера коллекции `system.profile`

По умолчанию `system.profile` имеет размер 1 МБ. Чтобы увеличить (например, до 5 МБ):

1. Отключите профилирование:

```
javascript
```

```
db.setProfilingLevel(0)
```

2. Удалите текущую коллекцию:

```
javascript
```

```
db.system.profile.drop()
```

3. Создайте новую `capped`-коллекцию нужного размера:

```
javascript
```

```
db.createCollection("system.profile", { capped: true, size: 5000000 })
```

- `size: 5000000` — 5 МБ в байтах.

4. Включите профилирование снова:

```
javascript
db.setProfilingLevel(1) // или 2
```

### 7. Включение профилирования для всего экземпляра mongod

Чтобы применить профилирование ко **всем базам данных** экземпляра, запустите `mongod` с параметрами:

```
bash
mongod --profile 1 --slowms 15 --slowOpSampleRate 0.5
```

- `--profile 1` — уровень профилирования.
- `--slowms 15` — порог «медленных» операций (15 мс).
- `--slowOpSampleRate 0.5` — доля отслеживаемых медленных операций (50 %).

Либо укажите эти параметры в конфигурационном файле MongoDB.

### Контрольные вопросы

1. Каковы три уровня профилирования в MongoDB и чем они отличаются?
2. В какой коллекции хранятся данные профилировщика?
3. Как проверить текущий уровень профилирования?
4. Как включить профилирование для всех операций БД?
5. Почему по умолчанию профилировщик отключён?
6. Как изменить порог длительности для «медленных» операций?
7. Что такое `capped collection` и зачем она используется для `system.profile`?
8. Как увеличить размер коллекции `system.profile`?
9. Как включить профилирование для всего экземпляра `mongod`?
10. Какие ключевые поля записи в `system.profile` полезны для анализа

производительности?

### Требования к отчёту

1. Цель работы.
2. Краткие теоретические сведения (уровни профилирования, коллекция `system.profile`).
3. Последовательность выполненных действий (команды `mongosh` с пояснениями).
4. Примеры выведенных записей из `system.profile` (не менее 3-х, с комментариями).
5. Ответы на контрольные вопросы.
6. Выводы (что удалось настроить, какие данные получены, возможные сценарии использования профилировщика).

### Примечания

- В продуктивной среде используйте уровень 1 (только медленные операции) во избежание перегрузки.
- Регулярно проверяйте размер `system.profile` и при необходимости увеличивайте его.
- Учитывайте безопасность: лог может содержать конфиденциальные запросы.

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

### **Практическая работа 66 Конфигурация параметров логирования запросов в MySQL (slow\_query\_log) и анализ записей.**

***Цель работы*** освоить настройку формата и параметров логирования в PostgreSQL для эффективного мониторинга и диагностики работы СУБД.

#### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

#### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

#### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
- Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### ***Теоретические и учебно-методические материалы по теме практической работы***

### **Теоретические сведения**

**Журнал медленных запросов** (`slow_query_log`) — встроенный механизм MySQL для регистрации SQL-запросов, выполнение которых занимает больше заданного времени или не использует индексы.

Ключевые параметры конфигурации:

- `$slow_query_log$` — включение/выключение журнала (1/0 или ON/OFF);
- `$slow_query_log_file$` — путь к файлу журнала;
- `$long_query_time$` — пороговое время выполнения запроса (в секундах);
- `$log_queries_not_using_indexes$` — запись запросов без индексов;
- `$min_examined_row_limit$` — минимальное число проанализированных строк

для записи в журнал.

### 3. Порядок выполнения

#### Шаг 1. Проверка текущей конфигурации

Выполните запрос для просмотра текущих значений параметров:

```
sql
SHOW VARIABLES LIKE '%slow%';
```

#### Шаг 2. Включение журнала (для MySQL ≥ 5.1.6)

Выполните команды в консоли MySQL:

```
sql
SET GLOBAL slow_query_log = 'ON';
SET GLOBAL slow_query_log_file = '/var/log/mysql/slow-query.log';
SET GLOBAL long_query_time = 2;
SET GLOBAL log_queries_not_using_indexes = 'ON';
```

*Примечание:* путь к файлу должен быть доступен для записи пользователю `mysql`.

#### Шаг 3. Постоянная настройка (редактирование конфигурационного файла)

Добавьте в файл `my.cnf` (или `my.ini` для Windows) в секцию `[mysqld]`:

```
ini
[mysqld]
slow_query_log = 1
slow_query_log_file = /var/log/mysql/slow-query.log
long_query_time = 2
log_queries_not_using_indexes = 1
```

Перезапустите MySQL:

```
bash
sudo systemctl restart mysql
```

#### Шаг 4. Проверка работы журнала

Убедитесь, что журнал активен:

```
sql
SHOW VARIABLES LIKE 'slow_query_log';
```

Ожидаемый результат: `Value = ON`.

#### Шаг 5. Генерация тестовых медленных запросов

Для проверки работы журнала выполните запрос с искусственной задержкой:

```
sql
SELECT SLEEP(3);
```

Этот запрос должен попасть в журнал, так как его время выполнения (3 с) превышает порог `$long_query_time = 2$`.

## Шаг 6. Анализ журнала

### Способ 1. Просмотр файла напрямую

Используйте команду `tail` для мониторинга в реальном времени:

```
bash
tail -f /var/log/mysql/slow-query.log
```

### Способ 2. Анализ с помощью `mysqldumpslow`

Инструмент группирует запросы по шаблону и выводит статистику:

```
bash
mysqldumpslow /var/log/mysql/slow-query.log
```

Популярные опции:

- `-t 10` — показать топ-10 самых медленных запросов;
- `-s at` — сортировка по среднему времени выполнения.

### Способ 3. Анализ с помощью `pt-query-digest`

Более продвинутый инструмент из Percona Toolkit:

```
bash
pt-query-digest /var/log/mysql/slow-query.log
```

Выводит детализированную статистику: количество выполнений, суммарное время, среднее время и др.

## 4. Структура записи в журнале

Пример записи:

```
# Time: 2025-12-13T16:30:00.123456Z
# User@Host: user[user] @ localhost []
# Query_time: 3.123 Lock_time: 0.000 Rows_sent: 1 Rows_examined: 10000
SET timestamp=1734078600;
SELECT * FROM large_table WHERE unindexed_column = 'value';
```

Поля:

- `Time` — время записи запроса;
- `User@Host` — пользователь и хост;
- `Query_time` — время выполнения (сек);
- `Lock_time` — время ожидания блокировки;
- `Rows_sent` — количество возвращённых строк;
- `Rows_examined` — количество проанализированных строк.

## 5. Интерпретация результатов

Критические показатели для анализа:

1. **Высокое** `Query_time` — запрос выполняется слишком долго.
2. **Большое** `Rows_examined` при малом `Rows_sent` — неэффективный запрос (возможно, отсутствует индекс).
3. **Частые повторения одного шаблона** — потенциальная точка оптимизации.

## 6. Рекомендации по оптимизации

1. Добавьте индексы для столбцов в условиях `WHERE`, `JOIN`, `ORDER BY`.
2. Перепишите запросы с использованием `LIMIT` для ограничения выборки.
3. Избегайте `SELECT *` — указывайте только нужные столбцы.

4. Используйте `EXPLAIN` для анализа плана выполнения:

```
sql
EXPLAIN SELECT * FROM table WHERE column = 'value';
```

## 7. Отключение журнала

Для временного отключения:

```
sql
SET GLOBAL slow_query_log = 'OFF';
```

Для постоянного отключения — удалите или закомментируйте параметры в `my.cnf` и перезапустите MySQL.

## 8. Контрольные вопросы

1. Какие параметры необходимы для включения `slow_query_log`?
2. Как проверить, что журнал активен?
3. Какие инструменты используются для анализа журнала?
4. Что означает поле `Rows_examined` в записи журнала?
5. Как отключить журнал медленных запросов?

## 9. Отчет по работе

Подготовьте отчёт, включающий:

1. Скриншоты вывода команд `SHOW VARIABLES` до и после настройки.
2. Пример записи из журнала медленных запросов.
3. Результаты анализа с помощью `mysqldumpslow` или `pt-query-digest`.
4. Предложения по оптимизации на основе выявленных проблем.
5. Ответы на контрольные вопросы.

### *Оформление результатов работы*

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### *Содержание отчёта*

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## Практическая работа 67 Создание и настройка собственного формата логов в PostgreSQL.

**Цель работы** изучить, как управлять учётными записями пользователей и контролировать доступ к базам данных

#### **Задачи:**

- Систематизировать подходы к изучению предмета.

ИИ

- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

**Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

**Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

**Теоретические и учебно-методические материалы по теме практической работы**

**Теоретические основы**

PostgreSQL позволяет гибко настраивать логирование через файл конфигурации `postgresql.conf`. Основные параметры управляют:

- местом хранения логов;
- форматом вывода;
- уровнем детализации;
- ротацией файлов.

**Пошаговая инструкция**

**Шаг 1. Активация сборщика логов**

Включите `logging_collector` — он перенаправляет логи из терминала в файлы:

```
logging_collector = on
```

**Шаг 2. Задание директории для логов**

Укажите путь к каталогу для хранения логов:

```
log_directory = 'pg_log'
```

По умолчанию — подкаталог `pg_log` в директории данных PostgreSQL.

**Шаг 3. Настройка имени файлов логов**

Определите шаблон именования с временными метками:

```
log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log'
```

- `%Y` — год (4 цифры);
- `%m` — месяц (2 цифры);
- `%d` — день (2 цифры);
- `%H` — час (24-часовой формат);
- `%M` — минуты;
- `%S` — секунды.

**Шаг 4. Выбор формата вывода**

Задайте формат логов через `log_destination` (можно комбинировать):

```
log_destination = 'stderr,csvlog,jsonlog'
```

- `stderr` — стандартный формат текста;
- `csvlog` — CSV-формат (удобен для импорта в таблицы);
- `jsonlog` — JSON-формат (структурирован для парсинга).

### Шаг 5. Настройка префикса строк логов

Добавьте метаданные в начало каждой записи с помощью `log_line_prefix`. Пример:

```
log_line_prefix = '%m [%p] %q%u@%d '
```

Расшифровка кодов:

- `%m` — время в формате `yyyy-mm-dd hh:mm:ss.ms`;
- `%p` — PID процесса;
- `%q` — пустая строка для суперпользователя, иначе пробел;
- `%u` — имя пользователя;
- `%d` — имя базы данных.

### Шаг 6. Управление ротацией логов

Настройте ротацию по времени и размеру:

```
log_rotation_age = '1d' -- новый файл каждые 24 часа
```

```
log_rotation_size = '100MB' -- новый файл при достижении 100 МБ
```

### Шаг 7. Включение перезаписи старых логов

Чтобы старые файлы перезаписывались (а не дополнялись):

```
log_truncate_on_rotation = on
```

### Шаг 8. Настройка уровня детализации

Выберите, какие операции логировать:

```
log_statement = 'all' -- все SQL-запросы
```

-- или

```
log_statement = 'mod' -- только изменения данных (INSERT, UPDATE, DELETE)
```

-- или

```
log_statement = 'none' -- отключить логирование запросов
```

### Шаг 9. Логирование медленных запросов

Записывайте запросы, выполняющиеся дольше заданного времени (в миллисекундах):

```
log_min_duration_statement = 200 -- логировать запросы > 200 мс
```

### Шаг 10. Применение настроек

1. Сохраните изменения в `postgresql.conf`.
2. Перезапустите сервер PostgreSQL:
3. `sudo systemctl restart postgresql`

Или выполните команду в SQL:

```
sql
```

```
SELECT pg_reload_conf();
```

### Проверка результатов

1. Перейдите в каталог логов (`pg_log`).
2. Откройте свежий файл (например, `postgresql-2025-04-15_103000.log`).
3. Убедитесь, что записи содержат:
  - временную метку;
  - PID процесса;
  - имя пользователя и БД;
  - текст запроса (если `log_statement = 'all'`).

### Дополнительные настройки (по желанию)

- **Логирование подключений:**

- log\_connections = on
- log\_disconnections = on
- **Логирование контрольных точек:**
- log\_checkpoints = on
- **Формат CSV:**

Для `csvlog` добавьте:

- csvlog\_header = on -- включать заголовок в CSV

#### Типичные ошибки и их устранение

- Логи не появляются в каталоге**
  - Проверьте, включён ли `logging_collector`.
  - Убедитесь, что у PostgreSQL есть права на запись в `log_directory`.
- Файл лога не ротируется**
  - Проверьте значения `log_rotation_age` и `log_rotation_size`.
  - Убедитесь, что `logging_collector = on`.
- Не хватает деталей в логах**
  - Увеличьте `log_line_prefix` (добавьте `%l` для номера строки в файле).
  - Установите `log_statement = 'all'`.

#### Заключение

Вы настроили:

- хранение логов в файлах;
- формат и структуру записей;
- автоматическую ротацию;
- фильтрацию по времени выполнения запросов.

#### Рекомендации:

- Для продуктивных систем используйте `jsonlog` для удобного парсинга.
- Регулярно архивируйте старые логи.
- Ограничьте `log_min_duration_statement` (например, 500 мс), чтобы не перегружать диск.

#### **Оформление результатов работы**

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

##### **Содержание отчёта**

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

**Сформулировать выводы по результатам работы.**

**Сдать и защитить работу.**

## **Практическая работа 68 Протоколирование событий доступа к данным в Microsoft SQL Server и анализ логов.**

**Цель работы** изучить, протоколирование событий доступа к данным в Microsoft SQL Server и анализ логов включает настройку аудита, сбор данных о событиях и их анализ.

### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### **Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
  - Предъявить преподавателю результаты работы.
  - Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
  - Подготовить ответы на контрольные вопросы.

### **Теоретические и учебно-методические материалы по теме практической работы**

#### **Настройка аудита в SQL Server**

Аудит позволяет отслеживать действия пользователей и системные события.

Основные шаги:

##### **1. Создание сервера аудита:**

```
sql
```

```
CREATE SERVER AUDIT ServerAudit  
TO FILE (FILEPATH = 'D:\Audit\', MAXSIZE = 1GB)  
WITH (ON_FAILURE = CONTINUE);
```

##### **2. Включение сервера аудита:**

```
sql
```

```
ALTER SERVER AUDIT ServerAudit WITH (STATE = ON);
```

##### **3. Создание спецификации аудита:**

```
sql
```

```
CREATE DATABASE AUDIT SPECIFICATION SA_MyDb_Orders  
FOR SERVER AUDIT ServerAudit  
ADD (SELECT, UPDATE, INSERT, DELETE ON dbo.Orders BY PUBLIC),  
ADD (SELECT, UPDATE, INSERT, DELETE ON dbo.OrderDetails BY PUBLIC);
```

##### **4. Включение спецификации:**

```
sql
```

```
ALTER DATABASE AUDIT SPECIFICATION SA_MyDb_Orders WITH (STATE =  
ON);
```

## Анализ журналов транзакций

Журналы транзакций содержат информацию о всех операциях с базой данных (INSERT, UPDATE, DELETE, DDL-операции). Для их анализа можно использовать:

- **Функция** `fn_dblog()`: позволяет просматривать активные журналы

транзакций. Пример запроса:

```
sql
SELECT [current lsn], [operation], [transaction name], [transaction id], [transaction sid],
[spid], [begin time]
FROM fn_dblog(NULL, NULL);
```

- **Функция** `fn_dump_dblog()`: работает с активными и архивными журналами, но требует указания множества параметров.

- **Сторонние инструменты** (например, SysTools SQL Log Analyzer): упрощают анализ журналов, предоставляют графический интерфейс и возможности экспорта данных.

### Использование расширенных событий (Extended Events)

Extended Events — более современный и гибкий инструмент для мониторинга.

Пример создания сессии:

```
sql
CREATE EVENT SESSION MySession ON SERVER
ADD EVENT sqlserver.sql_batch_completed
ADD TARGET package0.ring_buffer;
```

```
ALTER EVENT SESSION MySession ON SERVER STATE = START;
```

Для анализа данных можно использовать функцию `sys.fn_xe_file_target_read_file`.

## Анализ журналов ошибок

Журналы ошибок SQL Server содержат информацию о системных и пользовательских ошибках, предупреждениях и операциях резервного копирования. Их можно просматривать через:

- **SQL Server Management Studio (SSMS)**: в Object Explorer перейдите в «Management» → «SQL Server Logs».

- **T-SQL**: используйте хранимую процедуру `xp_readerrorlog`. Например:

```
sql
EXEC xp_readerrorlog 0, 1;
```

### Рекомендации

- **Фильтрация данных**: используйте фильтры по базам данных, приложениям или пользователям для уменьшения объема захватываемых данных.

- **Регулярный анализ**: проверяйте журналы на предмет подозрительной активности, ошибок и узких мест производительности.

- **Резервное копирование**: регулярно создавайте резервные копии журналов, чтобы избежать потери данных.

- **Использование сторонних инструментов**: для сложных задач анализа рассмотрите специализированные решения, такие как dbForge Transaction Log или SQL Log Analyzer.

### Дополнительные инструменты

- **SQL Server Profiler:** устаревший инструмент для трассировки событий, но может быть полезен для быстрого анализа.
- **Vulnerability Assessment в SSMS:** позволяет сканировать базу данных на наличие уязвимостей и получать рекомендации по их устранению. [blog.sqlauthority.com](http://blog.sqlauthority.com) +1  
Для глубокого понимания процессов рекомендуется изучить документацию Microsoft и примеры из практики, например, на сайтах [и](#) .  
Если у вас есть конкретные задачи или вопросы по настройке аудита или анализу логов, уточните их — помогу разобраться!

### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## **Практическая работа 69 Настройка ротации логов и очистки устаревших записей в MongoDB.**

***Цель работы*** изучить, как управлять учётными записями пользователей и контролировать доступ к базам данных

### ***Задачи:***

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей ИИ

### ***Оборудование, технические средства и инструменты:***

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### ***Ход практического занятия:***

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.

- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

## *Теоретические и учебно-методические материалы по теме практической работы*

### 1. Введение

По умолчанию MongoDB **не имеет встроенной ротации логов**, что может привести к переполнению дискового пространства. В данной работе рассматриваются два ключевых аспекта:

- настройка ротации лог-файлов с помощью утилиты `logrotate`;
- механизмы очистки устаревших данных в MongoDB.

### 2. Настройка ротации логов через logrotate

#### 2.1. Подготовка конфигурации MongoDB

Отредактируйте файл конфигурации `/etc/mongod.conf`, добавив/изменив следующие параметры:

```
yaml
systemLog:
  destination: file
  path: "/var/log/mongodb/mongod.log"
  logAppend: true
  logRotate: reopen

processManagement:
  pidFilePath: "/var/run/mongodb.pid" # укажите актуальный путь
```

#### Пояснения:

- `logAppend: true` — дописывать новые записи в конец файла;
- `logRotate: reopen` — переоткрывать файл после ротации (требуется для внешней утилиты);
- `pidFilePath` — путь к PID-файлу процесса `mongod` (понадобится для сигнала ротации).

Перезапустите MongoDB:

```
bash
sudo systemctl restart mongod
```

#### 2.2. Создание конфигурации logrotate

Создайте файл `/etc/logrotate.d/mongodb` со следующим содержимым:

```
bash
/var/log/mongodb/mongod.log {
  rotate 7
  daily
  size 1M
  missingok
  create 0600 mongodb mongod
  delaycompress
```

```

compress
sharedscripts
postrotate
    /bin/kill -SIGUSR1 $(cat /var/run/mongodb.pid)
endscript
}

```

### Параметры:

- `rotate 7` — хранить 7 последних ротированных файлов;
- `daily` — ротировать ежедневно;
- `size 1M` — ротировать, если размер превысил 1 МБ (можно заменить на `weekly`, `monthly`);
- `missingok` — не выдавать ошибку, если файл отсутствует;
- `create 0600 mongodb mongodb` — создавать новый лог с правами `0600` и владельцем `mongodb`;
- `delaycompress` — не сжимать файл при первой ротации (оставляет `mongod.log.1` несжатым);
- `compress` — сжимать ротированные файлы (gzip);
- `sharedscripts` — выполнять скрипты один раз для всех совпадающих файлов;
- `postrotate` — команда, выполняемая после ротации: сигнал `SIGUSR1` процессу `mongod` для переоткрытия лог-файла.

## 2.3. Тестирование конфигурации

Протестируйте ротацию вручную:

```

bash
sudo logrotate --force /etc/logrotate.d/mongodb

```

Проверьте результат:

```

bash
ls -la /var/log/mongodb/

```

Ожидаемо:

- новый пустой `mongod.log`;
- архив `mongod.log.1.gz` (сжатый);
- возможно `mongod.log.1` (несжатый, если `delaycompress` активен).

## 3. Очистка устаревших данных в MongoDB

### 3.1. Использование TTL-индексов

TTL-индексы автоматически удаляют документы по истечении заданного времени.

**Пример:**

1. Создайте индекс с параметром `expireAfterSeconds`:

```

javascript
db.logs.createIndex({ "createdAt": 1 }, { expireAfterSeconds: 86400 })

```

- `createdAt` — поле даты/времени в документах;
  - `86400` — время жизни документа (1 день в секундах).
2. Вставляйте документы с полем `createdAt`:

```
javascript
db.logs.insert({
  message: "Пример лога",
  createdAt: new Date()
})
```

**Примечание:** MongoDB проверяет TTL-индексы **раз в минуту**, поэтому удаление может произойти с задержкой до 60 сек.

### 3.2. Ручное удаление старых данных

Для периодической очистки используйте скрипты или задания cron.

**Пример скрипта** (`cleanup.js`):

```
javascript
// Удалить документы старше 30 дней
db.logs.deleteMany({
  createdAt: { $lt: new Date(Date.now() - 30 * 24 * 60 * 60 * 1000) }
})

// Опционально: компакнуть коллекцию после удаления
db.runCommand({ compact: "logs" })
```

Запустите скрипт через `mongosh`:

```
bash
mongosh mongodb://localhost:27017/mydb --file cleanup.js
```

**Совет:** Добавьте скрипт в cron для регулярного выполнения:

```
bash
0 2 * * * /usr/bin/mongosh mongodb://localhost:27017/mydb --file /path/to/cleanup.js
```

### 3.3. Ограничение размера коллекций (Capped Collections)

Capped-коллекции имеют фиксированный размер и автоматически удаляют старые документы при переполнении.

**Создание:**

```
javascript
db.createCollection("logs", {
  capped: true,
  size: 10485760, // 10 МБ
  max: 1000      // максимум 1000 документов
})
```

**Особенности:**

- порядок вставки сохраняется;
- удаление старых документов происходит автоматически;
- нельзя удалять документы вручную (только через `drop()` коллекции).

## 4. Мониторинг и отладка

### 4.1. Проверка ротации логов

- Логи `logrotate`: `/var/log/syslog` или `/var/log/cron`.
- Состояние файлов: `ls -la /var/log/mongodb/`.
- PID процесса: `cat /var/run/mongodb.pid`.

#### 4.2. Проверка очистки данных

- Количество документов: `db.logs.countDocuments()`.
- Последние документы: `db.logs.find().sort({ createdAt: -1 }).limit(5)`.
- Статус TTL-индекса: `db.logs.getIndexes()`.

#### 5. Рекомендации

1. **Резервное копирование:** Перед массовыми операциями удаления создавайте бэкапы.
2. **Тестирование:** Проверяйте скрипты на тестовой среде.
3. **Логирование:** Ведите логи выполнения скриптов очистки.
4. **Мониторинг:** Отслеживайте размер логов и коллекций.
5. **Документация:** Фиксируйте настройки ротации и очистки в документации проекта.

#### 6. Заключение

Настроенная ротация логов через `logrotate` и механизмы очистки данных (TTL-индексы, скрипты, `sarred`-коллекции) обеспечивают:

- контроль размера лог-файлов;
- автоматическое удаление устаревших данных;
- стабильность работы MongoDB.

#### Итоговые команды для проверки:

```
bash
# Статус MongoDB
sudo systemctl status mongod

# Логи ротации
grep "mongodb" /var/log/syslog

# Размер логов
du -sh /var/log/mongodb/

# Количество документов в коллекции
mongosh mongodb://localhost:27017/mydb --eval "db.logs.countDocuments()"
```

#### Оформление результатов работы

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### Содержание отчёта

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и

кодированию.

**Ответить на контрольные вопросы.(устно)**  
**Сформулировать выводы по результатам работы.**  
**Сдать и защитить работу.**

## **Практическая работа 70 Разработка политики протоколирования событий и настройка соответствующих параметров в Oracle Database.**

**Цель работы** изучить, как управлять учётными записями пользователей и контролировать доступ к базам данных

### **Задачи:**

- Систематизировать подходы к изучению предмета.
- Научить правильному плану работы ИИ
- Показать основные приемы эффективного использования возможностей

ИИ

### **Оборудование, технические средства и инструменты:**

- Методические рекомендации для практических занятий
- Компьютер с подключением к сети Интернет

### **Ход практического занятия:**

- Для выполнения данного задания необходимо изучить теоретический материал по данной теме.
  - Руководствуясь методическими рекомендациями (инструкционная карта работы), студенты выполняют практическую работу. Предъявить преподавателю результаты работы.
- Предъявить преподавателю результаты работы.
- Оформить отчет по практической работе в соответствии с требованиями (содержание отчета см. ниже).
- Подготовить ответы на контрольные вопросы.

### **Теоретические и учебно-методические материалы по теме практической работы**

#### **1. Цели работы**

- освоить принципы построения политики аудита (протоколирования) в Oracle Database;
- научиться настраивать параметры аудита на уровне системы и объектов;
- отработать навыки анализа журналов аудита;
- сформировать понимание требований к хранению и защите логов.

#### **2. Теоретические основы**

**Аудит в Oracle** — это механизм регистрации действий пользователей и системных событий в специальных журналах для:

- контроля доступа к данным;

- выявления подозрительных операций;
- соответствия регуляторным требованиям (GDPR, ФЗ-152, PCI DSS и др.);
- расследования инцидентов.

#### Основные типы аудита:

- **Стандартный аудит** (через параметр `AUDIT_TRAIL`);
- **Fine-Grained Auditing (FGA)** — аудит по условиям для конкретных таблиц/столбцов;
  - **Унифицированный аудит** (Unified Auditing, начиная с Oracle 12c) — единый механизм с расширенными возможностями.

### 3. Этапы выполнения

#### Этап 1. Определение политики аудита

Сформулируйте требования к аудиту для вашей среды:

- Какие события логировать: вход/выход, DDL, DML, привилегии, ошибки.
- Уровень детализации: кто, когда, откуда, какие данные затронуты.
- Сроки хранения логов (например, 90 дней).
- Кто имеет доступ к логам (только DBA, Security Officer).

#### Пример политики:

Аудировать:

- все попытки входа/выхода (`LOGON`, `LOGOFF`);
- операции `CREATE`, `ALTER`, `DROP` для объектов схемы;
- изменения данных в таблицах `EMPLOYEES`, `SALARIES` (через FGA);
- привилегии `GRANT`, `REVOKE`.

Хранить логи 180 дней. Доступ — только группа `DBA_AUDIT`.

#### Этап 2. Включение и настройка стандартного аудита

1. Проверьте текущий режим аудита:

```
sql
SHOW PARAMETER AUDIT_TRAIL;
```

Возможные значения:

- `NONE` — аудит выключен;
- `DB` — логи в словаре данных (`SYS.AUD$`);
- `DB, EXTENDED` — с деталями SQL и bind-переменных;
- `OS` — в ОС-журнал;
- `XML` — в XML-файлы.

2. Включите аудит (требует перезапуска):

```
sql
ALTER SYSTEM SET AUDIT_TRAIL = DB, EXTENDED SCOPE=SPFILE;
```

3. Настройте аудит событий:

```
sql
-- Вход/выход
AUDIT CREATE SESSION;
AUDIT DROP SESSION;
```

```
-- DDL-операции
AUDIT CREATE TABLE, ALTER TABLE, DROP TABLE BY ACCESS;
AUDIT CREATE USER, ALTER USER, DROP USER BY ACCESS;
```

```
-- Привилегии
AUDIT GRANT ANY PRIVILEGE BY ACCESS;
AUDIT REVOKE ANY PRIVILEGE BY ACCESS;
```

### Этап 3. Настройка Fine-Grained Auditing (FGA)

Пример: аудит изменений в столбце `SALARY` таблицы `EMPLOYEES`:

```
sql
BEGIN
  DBMS_FGA.ADD_POLICY(
    object_schema => 'HR',
    object_name   => 'EMPLOYEES',
    policy_name   => 'AUDIT_SALARY_CHANGES',
    audit_condition => 'SALARY != OLD_SALARY', -- условие
    audit_column  => 'SALARY',
    handler_schema => NULL,
    handler_module => NULL,
    enable        => TRUE
  );
END;
/
```

### Этап 4. Настройка унифицированного аудита (для Oracle 12c+)

1. Включите унифицированный аудит:

```
sql
-- Остановка БД
SHUTDOWN IMMEDIATE;
STARTUP UPgrade;

-- Активация
@?/rdbms/admin/ua_setup.sql
EXEC DBMS_AUDIT_MGMT.INITIALIZE_UNIFIED_AUDITING();

-- Перезапуск
SHUTDOWN IMMEDIATE;
STARTUP;
```

2. Создайте политику:

```
sql
CREATE AUDIT POLICY audit_policy_dml
  ACTIONS INSERT, UPDATE, DELETE ON HR.EMPLOYEES
  WHEN 'SESSION_USER != "HR"' -- условие
  EVALUATE PER STATEMENT;
```

3. Включите политику:

```
sql
AUDIT POLICY audit_policy_dml;
```

### Этап 5. Управление хранением логов

1. Очистите старые записи:

```
sql
BEGIN
  DBMS_AUDIT_MGMT.CLEAN_AUDIT_TRAIL(
    audit_trail_type => DBMS_AUDIT_MGMT.AUDIT_TRAIL_ALL,
    use_last_arch_timestamp => TRUE
  );
END;
/
```

2. Настройте автоматическое архивирование:

```
sql
BEGIN
  DBMS_AUDIT_MGMT.SET_LAST_ARCHIVE_TIMESTAMP(
    audit_trail_type => DBMS_AUDIT_MGMT.AUDIT_TRAIL_UNIFIED,
    last_archive_time => SYSTIMESTAMP - 180 -- 180 дней
  );
END;
/
```

### Этап 6. Анализ логов

1. Запросы к стандартным логам (`AUD$`):

```
sql
SELECT TIMESTAMP, USERNAME, ACTION_NAME, OBJECT_NAME, SQLTEXT
FROM SYS.AUD$
WHERE TIMESTAMP > SYSDATE - 7;
```

2. Запросы к унифицированным логам:

```
sql
SELECT EVENT_TIMESTAMP, DB_USERNAME, ACTION_NAME,
OBJECT_SCHEMA, OBJECT_NAME
FROM UNIFIED_AUDIT_TRAIL
WHERE EVENT_TIMESTAMP > SYSTIMESTAMP - 7;
```

3. Отчёт по FGA:

```
sql
SELECT SQL_TEXT, DB_USER, TIMESTAMP
FROM DBA_FGA_AUDIT_TRAIL
WHERE POLICY_NAME = 'AUDIT_SALARY_CHANGES';
```

### 4. Контрольные задания

1. Включите стандартный аудит для событий `LOGON` и `LOGOFF`. Проверьте, что записи появляются в `AUD$`.
2. Настройте FGA для таблицы `HR.DEPARTMENTS` (столбец `MANAGER_ID`). Имитируйте изменение и убедитесь, что событие зафиксировано.
3. Сформируйте отчёт за последние 24 часа по всем операциям `DROP TABLE`.

4. Очистите логи старше 30 дней.

### 5. Требования к отчёту

Отчёт должен содержать:

- сформулированную политику аудита (текст);
- скрипты настройки аудита (стандартный, FGA, унифицированный);
- примеры запросов к логам;
- скриншоты результатов выполнения контрольных заданий.

### 6. Вопросы для самопроверки

1. В чём отличие `AUDIT_TRAIL=DB` от `AUDIT_TRAIL=OS`?
2. Как отключить аудит для конкретного события?
3. Какие привилегии нужны для просмотра `UNIFIED_AUDIT_TRAIL`?
4. Как экспортировать логи аудита для внешнего анализа?

### 7. Дополнительные рекомендации

• Используйте **Oracle Audit Vault and Database Firewall** для централизованного сбора и анализа логов.

• Настройте **уведомления** о критических событиях (например, через `DBMS_ALERT`).

• Регулярно **шифруйте** архивы логов.

• Ограничьте доступ к таблицам аудита (`AUD$`, `UNIFIED_AUDIT_TRAIL`)

через роли.

#### ***Оформление результатов работы***

Оформить отчёт о проделанной работе, который должен содержать исчерпывающие текстовые ответы на поставленные вопросы с решениями, пояснениями, результатами решения:

#### ***Содержание отчёта***

1. Цель.
2. Раскрытие темы работы с сопровождением необходимыми материалами по построению и кодированию.

**Ответить на контрольные вопросы.(устно)**

***Сформулировать выводы по результатам работы.***

***Сдать и защитить работу.***

## Критерии оценки

**Оценка «отлично»** ставится в том случае, если студент:

- свободно применяет полученные знания при выполнении практических заданий;
- выполнил работу в полном объеме с соблюдением необходимой последовательности действий;
- в письменном отчете по работе правильно и аккуратно выполнены все записи;
- при ответах на контрольные вопросы правильно понимает их сущность, дает точное определение и истолкование основных понятий, использует специальную терминологию дисциплины, не затрудняется при ответах на видоизмененные вопросы, сопровождает ответ примерами.

**Оценка «хорошо»** ставится, если:

- выполнены требования к оценке «отлично», но допущены 2 – 3 недочета при выполнении практических заданий и студент может их исправить самостоятельно или при небольшой помощи преподавателя;
- в письменном отчете по работе делает незначительные ошибки;
- при ответах на контрольные вопросы не допускает серьезных ошибок, легко устраняет отдельные неточности, но затрудняется в применении знаний в новой ситуации, приведении примеров.

**Оценка «удовлетворительно»** ставится, если:

- практическая работа выполнена не полностью, но объем выполненной части позволяет получить правильные результаты и выводы;
- в ходе выполнения работы студент продемонстрировал слабые практические навыки, были допущены ошибки;
- студент умеет применять полученные знания при решении простых задач по готовому алгоритму;
- в письменном отчете по работе допущены ошибки;
- при ответах на контрольные вопросы правильно понимает их сущность, но в ответе имеются отдельные пробелы и при самостоятельном воспроизведении материала требует дополнительных и уточняющих вопросов преподавателя.

**Оценка «неудовлетворительно»** ставится, если:

- практическая работа выполнена не полностью и объем выполненной работы не позволяет сделать правильных выводов, у студента имеются лишь отдельные представления об изученном материале, большая часть материала не усвоена;
- в письменном отчете по работе допущены грубые ошибки, либо он вообще отсутствует;
- на контрольные вопросы студент не может дать ответов, так как не овладел основными знаниями и умениями в соответствии с требованиями программы.