

Документ подписан Министерством науки и высшего образования Российской Федерации  
Информация о владельце:  
ФИО: Макаренко Елена Николаевна  
Должность: Ректор  
Дата подписания: 17.06.2026 13:24:14  
Уникальный программный ключ:  
c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ростовский государственный экономический университет (РИНХ)»  
Финансово-экономический колледж



УТВЕРЖДАЮ

Директор

Р. А. Сычев

2026г.

## Рабочая программа дисциплины Базы данных

Специальность

09.02.12 ТЕХНИЧЕСКАЯ ЭКСПЛУАТАЦИЯ И СОПРОВОЖДЕНИЕ  
ИНФОРМАЦИОННЫХ СИСТЕМ

Форма обучения                      очная

Часов по учебному плану 64

в том числе:

аудиторные занятия              48

самостоятельная работа        16

Ростов-на-Дону

2026 г.

**Распределение часов дисциплины по  
семестрам**

| Семестр<br>(<Курс>.<Семестр на курсе>) | 3 (2.1) |    | Итого |    |
|--|---------|----|-------|----|
|  | 16      |    |       |    |
| Неделя                                 | 16      |    |       |    |
| Вид занятий                            | УП      | РП | УП    | РП |
| Лекции                                 | 16      | 16 | 16    | 16 |
| Практические                           | 32      | 32 | 32    | 32 |
| Итого ауд.                             | 48      | 48 | 48    | 48 |
| Контактная работа                      | 48      | 48 | 48    | 48 |
| Сам. работа                            | 16      | 16 | 16    | 16 |
| Итого                                  | 64      | 64 | 64    | 64 |

**ОСНОВАНИЕ**

Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.12 ТЕХНИЧЕСКАЯ ЭКСПЛУАТАЦИЯ И СОПРОВОЖДЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ (приказ Министерство просвещения Российской Федерации от 10.03.2025 г. № 184)

Рабочая программа составлена по образовательной программе 09.02.12 ТЕХНИЧЕСКАЯ ЭКСПЛУАТАЦИЯ И СОПРОВОЖДЕНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ для набора 2026 года  
программа среднего профессионального образования

Учебный план утвержден учёным советом вуза от 03.03.2026 протокол № 9

Программу составил(и): Преп., Ленц С.С.

Председатель ЦМК: Ламин В. А.

Рассмотрено на заседании ЦМК от 06.03.2026 протокол № 7

| <b>1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ</b> |  |
|------------------------------------|--|
| 1.1                                | формирование навыков работы с базами данных, включая проектирование, администрирование и защиту. |

| <b>2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ</b> |  |
|--|--|
| Цикл (раздел) ООП:   | ОП   |
| <b>2.1</b>   | <b>Требования к предварительной подготовке обучающегося:</b>   |
| 2.1.1  | Операционные системы и среды   |
| 2.1.2  | Математический аппарат в отрасли информационных технологий   |
| 2.1.3  | Базы данных  |
| 2.1.4  | Архитектура аппаратных средств и основы сетевых технологий   |
| 2.1.5  | Основы алгоритмизации и программирования   |
| 2.1.6  | Тестирование и эксплуатация информационных систем  |
| 2.1.7  | Проектирование и разработка информационных систем  |
| 2.1.8  | Настройка и обеспечение работоспособности программных и аппаратных средств устройств инфокоммуникационных систем |
| <b>2.2</b>   | <b>Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:</b>     |
| 2.2.1  | Тестирование и эксплуатация информационных систем  |
| 2.2.2  | Демонстрационный экзамен   |

| <b>3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ</b>   |  |
|--|--|
| <b>3.1 Знать</b>   |  |
| <b>ОК 01.: Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам</b>   |  |
| актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте<br>алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах<br>структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности |  |
| <b>ОК 02.: Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности</b>   |  |
| номенклатуру информационных источников, применяемых в профессиональной деятельности<br>приемы структурирования информации; формат оформления результатов поиска информации<br>современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности  |  |
| <b>3.2 Уметь</b>   |  |
| <b>ОК 01.: Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам</b>   |  |
| распознавать задачу и/или проблему в профессиональном и/или социальном контексте;<br>анализировать задачу и/или проблему и выделять её составные части<br>определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы<br>составлять план действия; определять необходимые ресурсы; реализовывать составленный план; оценивать результат и последствия своих действий               |  |
| <b>ОК 02.: Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности</b>   |  |
| определять задачи для поиска информации; определять необходимые источники информации<br>планировать процесс поиска; структурировать получаемую информацию; выделять наиболее   |  |

значимое в перечне информации оформлять результаты поиска; применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение и цифровые средства

### 3.3 Владеть

#### **ОК 01.: Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам**

навыками распознавания и анализа профессиональных проблем

навыками поиска информации и планирования действий

навыками реализации плана и оценки результатов

#### **ОК 02.: Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности**

навыками определения цели поиска

навыками структурирования и анализа информации

навыками применения ИТ для решения профессиональных задач

## 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

| Код занятия | Наименование разделов и тем /вид занятия/   | Семестр / Курс | Часов | Компетенции      | Литература            | Примечание |
|-------------|---|----------------|-------|------------------|-----------------------|------------|
|             | <b>Раздел 1. Основы алгоритмизации, языки и системы программирования. Проектирование БД</b> |                |       |                  |                       |            |
| 1.1         | Основные положения теории баз данных, хранилищ данных, баз знаний /Лек/                     | 3              | 2     | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |            |
| 1.2         | Принципы построения концептуальной, логической и физической модели данных /Лек/             | 3              | 2     | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |            |
| 1.3         | Структуры данных СУБД. Нормализация (1НФ, 2НФ, 3НФ). Методы целостности /Лек/               | 3              | 2     | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |            |
| 1.4         | Сбор и анализ информации для проектирования БД /Пр/   | 3              | 2     | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |            |
| 1.5         | Проектирование реляционной схемы БД /Пр/  | 3              | 2     | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |            |
| 1.6         | Проектирование реляционной схемы БД /Пр/  | 3              | 2     | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |            |
| 1.7         | Приведение БД к нормальной форме 3НФ /Пр/   | 3              | 2     | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |            |
| 1.8         | Приведение БД к нормальной форме 3НФ /Пр/   | 3              | 2     | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |            |
| 1.9         | Дополнительные упражнения по моделированию данных /Пр/                                      | 3              | 2     | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |            |
| 1.10        | Изучение нормализации, выполнение домашнего задания по проектированию БД /Ср/               | 3              | 2     | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |            |
| 1.11        | Изучение нормализации, выполнение домашнего задания по                                      | 3              | 2     | ОК 01.<br>ОК 02. | Л1.1Л2.1.             |            |

|      |  |   |   |                  |                       |  |
|------|--|---|---|------------------|-----------------------|--|
|      | проектированию БД /Ср/   |   |   |                  | Э1.Э2.Э3              |  |
| 1.12 | Изучение нормализации, выполнение домашнего задания по проектированию БД /Ср/  | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
|      | <b>Раздел 2. Разработка и администрирование БД</b>   |   |   |                  |                       |  |
| 2.1  | Инструментальные средства проектирования схем БД. Технологии передачи и обмена данными /Лек/   | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 2.2  | Введение в SQL (DDL, DML, DCL). Установка и настройка PostgreSQL на Red OS Murom /Лек/   | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 2.3  | Автоматизация управления БД. Мониторинг, настройка обслуживания. Импорт/экспорт /Лек/  | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 2.4  | Создание базы данных в PostgreSQL (среда pgAdmin или psql) /Пр/  | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 2.5  | Организация локальной сети для доступа к PostgreSQL /Пр/   | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 2.6  | Установка и настройка PostgreSQL (конфигурационные файлы, pg_hba.conf) /Пр/  | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 2.7  | Экспорт данных из PostgreSQL в CSV/JSON /Пр/   | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 2.8  | Импорт данных в PostgreSQL из внешних источников /Пр/  | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 2.9  | Настройка автоматизации обслуживания (pg_cron, autovacuum), мониторинг (pg_stat activity) /Пр/                                       | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 2.10 | Написание SQL-запросов (SELECT, JOIN, GROUP BY), создание хранимых процедур (PL/pgSQL), изучение инструментов администрирования /Ср/ | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 2.11 | Написание SQL-запросов (SELECT, JOIN, GROUP BY), создание хранимых процедур (PL/pgSQL), изучение инструментов администрирования /Ср/ | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 2.12 | Написание SQL-запросов (SELECT, JOIN, GROUP BY), создание хранимых процедур (PL/pgSQL), изучение инструментов администрирования /Ср/ | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
|      | <b>Раздел 3. Организация защиты данных в хранилищах</b>  |   |   |                  |                       |  |
| 3.1  | Способы контроля доступа и управления привилегиями в PostgreSQL (GRANT, REVOKE, роли) /Лек/  | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |

|     |  |   |   |                  |                       |  |
|-----|--|---|---|------------------|-----------------------|--|
| 3.2 | Резервное копирование (pg_dump, pg_basebackup) и восстановление (pg_restore). Модели восстановления /Лек/                        | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 3.3 | Выполнение резервного копирования БД (pg_dump) /Пр/  | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 3.4 | Восстановление базы данных из резервной копии (pg_restore, psql) /Пр/  | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 3.5 | Реализация доступа пользователей (создание ролей, назначение привилегий). Мониторинг безопасности (логи, audit) /Пр/             | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 3.6 | Развертывание дополнительных средств защиты (SSL-сертификаты, настройка pg_hba.conf) /Пр/  | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 3.7 | Самостоятельная работа: изучение политик безопасности PostgreSQL, настройка прав доступа, анализ логов, подготовка к зачёту /Ср/ | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 3.8 | Самостоятельная работа: изучение политик безопасности PostgreSQL, настройка прав доступа, анализ логов, подготовка к зачёту /Ср/ | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |
| 3.9 | Экзамен  | 3 | 2 | ОК 01.<br>ОК 02. | Л1.1Л2.1.<br>Э1.Э2.Э3 |  |

## 5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

### 5.1. Фонд оценочных средств для проведения промежуточной аттестации

Промежуточная аттестация проходит в форме экзамена. Перечень вопросов к экзамену:

1. Дайте определение базы данных и СУБД. Назначение PostgreSQL.
2. Что такое реляционная модель данных Основные понятия (отношение, кортеж, атрибут, домен).
3. Что такое первичный ключ, внешний ключ Приведите примеры.
4. Перечислите этапы проектирования базы данных.
5. Что такое нормализация Объясните 1НФ, 2НФ, 3НФ.
6. Какие типы данных поддерживает PostgreSQL?
7. Что такое SQL Группы команд SQL (DDL, DML, DCL, TCL).
8. Напишите синтаксис основных команд: CREATE TABLE, INSERT, SELECT, UPDATE, DELETE.
9. Что такое JOIN? Виды JOIN в PostgreSQL (INNER, LEFT, RIGHT, FULL, CROSS).
10. Что такое подзапрос Приведите пример с EXISTS, IN.
11. Что такое агрегатные функции (COUNT, SUM, AVG, MAX, MIN)? Группировка GROUP BY, фильтрация HAVING.
12. Что такое индекс Зачем нужен Типы индексов в PostgreSQL (B-tree, Hash, GIN).
13. Что такое транзакция Свойства ACID. Команды BEGIN, COMMIT, ROLLBACK.
14. Что такое представление (VIEW) Для чего используется
15. Что такое хранимая процедура Отличие от функции в PostgreSQL.
16. Что такое триггер События BEFORE/AFTER, FOR EACH ROW/STATEMENT.
17. Какие способы обеспечения целостности данных вы знаете (ограничения, внешние ключи, триггеры).
18. Как выполняется резервное копирование в PostgreSQL Утилиты pg\_dump, pg\_basebackup.
19. Как восстановить базу данных из резервной копии (pg\_restore, psql).

20. Что такое пользователь, роль, привилегия в PostgreSQL Команды CREATE ROLE, GRANT, REVOKE.
21. Как настроить аутентификацию в PostgreSQL Файл pg\_hba.conf.
22. Что такое схема (schema) в PostgreSQL Зачем нужна
23. Как настроить SSL-соединение с PostgreSQL
24. Что такое автовакуум (autovacuum) Зачем он нужен
25. Как посмотреть активные сессии и выполняемые запросы (pg\_stat\_activity).
26. Что такое EXPLAIN Как анализировать план выполнения запроса
27. Назовите основные конфигурационные файлы PostgreSQL (postgresql.conf, pg\_hba.conf, pg\_ident.conf).
28. Как импортировать данные из CSV в таблицу PostgreSQL (команда COPY).
29. Как экспортировать результат запроса в файл
30. Какие существуют методы мониторинга производительности PostgreSQL
31. Что такое WAL (Write-Ahead Log) в PostgreSQL Для чего нужен
32. Что такое репликация в PostgreSQL Виды репликации.
33. Что такое блокировки (locks) в PostgreSQL Типы блокировок.
34. Что такое VACUUM и ANALYZE Для чего используются
35. Как настроить логирование в PostgreSQL Основные параметры.

#### Критерии оценивания

5 баллов – полный и правильный ответ на все вопросы билета с логическим обоснованием аргументов, в ответе нет ошибок.

4 балла – вопросы билета раскрыты полностью, но обоснования и доказательства недостаточны; допущены 2–3 несущественные ошибки, исправленные по требованию преподавателя.

3 балла – правильный ответ на вопросы билета, но допущено более одной ошибки по изложению фактов или более 2–3 недочётов в ответе.

2 балла – допущены существенные ошибки, показавшие, что обучающийся не обладает обязательными умениями по данной теме в полной мере

#### 5.2. Фонд оценочных средств для проведения текущего контроля

Представлен в Приложении 1 к рабочей программе дисциплины.

| <b>6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ</b>                       |   |  |                            |   |
|--|---|--|----------------------------|---|
| <b>6.1. Рекомендуемая литература</b>   |   |  |                            |   |
| <b>6.1.1. Основная литература</b>  |   |  |                            |   |
|  | Авторы,   | Заглавие   | Издательств                | Колич-во  |
| Л1.1   | Илюшечкин В. М.   | Основы использования и проектирования баз данных : учебник для СПО | М. : Юрайт, 2025. — 213 с. | <a href="https://web2.ura.it.ru/bcode/562514">https://web2.ura.it.ru/bcode/562514</a><br>неограниченный доступ зарегистрированным пользователям |
| <b>6.1.2. Дополнительная литература</b>  |   |  |                            |   |
|  | Авторы,   | Заглавие   | Издательств                | Колич-во  |
| Л2.1   | Волк В. К.  | Базы данных. PostgreSQL : учебник для СПО                          | М. : Юрайт, 2025. — 828 с. | <a href="https://web2.ura.it.ru/bcode/582051">https://web2.ura.it.ru/bcode/582051</a><br>неограниченный доступ зарегистрированным пользователям |
| <b>6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"</b> |   |  |                            |   |
| Э1   | Pgadmin4-документация по pgadmin <a href="https://www.pgadmin.org/docs/">https://www.pgadmin.org/docs/</a>                                      |  |                            |   |
| Э2   | Red Soft – документация по Red OS, установка ПО <a href="https://redsoft.ru">https://redsoft.ru</a>   |  |                            |   |
| Э3   | Официальная документация PostgreSQL <a href="https://postgrespro.ru/docs/postgresql/current">https://postgrespro.ru/docs/postgresql/current</a> |  |                            |   |
| <b>6.3. Перечень программного обеспечения</b>                                    |   |  |                            |   |
| 6.3.1  | Офисный пакет Red OS  |  |                            |   |
| 6.3.2  | Офисный пакет LibreOffice   |  |                            |   |
| <b>6.4 Перечень информационных справочных систем</b>                             |   |  |                            |   |
| 6.4.1  | ИСС «КонсультантПлюс»   |  |                            |   |

|       |              |
|-------|--------------|
| 6.4.2 | ИСС «Гарант» |
|-------|--------------|

|   |  |
|---|--|
| <b>7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)</b> |  |
|---|--|

|     |  |
|-----|--|
| 7.1 | Помещения для проведения всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения. |
|-----|--|

|   |  |
|---|--|
| <b>8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ</b> |  |
|---|--|

|  |  |
|--|--|
| Методические указания по освоению дисциплины представлены в Приложении 2 к рабочей программе дисциплины. |  |
|--|--|

## ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

## ОП. 04 Базы данных

## 1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

## 1.1 Показатели и критерии оценивания компетенций:

| УУД, составляющие компетенцию   | Показатели оценивания   | Критерии оценивания  | Средства оценивания                |
|---|---|--|------------------------------------|
| <b>ОК.01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам</b>  |   |  |                                    |
| <p><b>Знать:</b><br/>актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте<br/>алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах<br/>структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности</p> | <p><b>Получение систематических знаний</b> о структурах данных, нормализации, SQL-командах, индексах, транзакциях, целостности данных</p>                                   | <p><b>Уровень знаний</b> основных понятий реляционных БД, синтаксиса SQL, этапов проектирования, методов обеспечения целостности</p>               | <p><b>Т(1-35)</b></p>              |
| <p><b>Уметь:</b><br/>распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части<br/>определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы<br/>составлять план действия; определять необходимые ресурсы; реализовывать составленный план; оценивать результат и последствия своих действий</p>                  | <p><b>Сформировать систематическое умение</b> разрабатывать структуру БД, писать сложные SQL-запросы, применять JOIN и подзапросы, группировать данные</p>                  | <p><b>Уровень умения</b> создавать и модифицировать таблицы, выполнять выборки с условиями, объединять таблицы, вычислять агрегаты</p>             | <p><b>Т(1-35)<br/>ПЗ(1-16)</b></p> |
| <p><b>Владеть:</b><br/>навыками распознавания и анализа профессиональных проблем<br/>навыками поиска информации и планирования действий<br/>навыками реализации плана и оценки результатов</p>  | <p><b>Сформировать систематическое владение</b> методами нормализации, написания эффективных запросов, настройки прав доступа, выполнения бэкапов (pg_dump, pg_restore)</p> | <p><b>Уровень владения</b> инструментарием PostgreSQL для создания, наполнения и администрирования БД, обеспечения целостности и защиты данных</p> | <p><b>Т(1-35)<br/>ПЗ(1-16)</b></p> |
| <b>ОК.02: Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной</b>  |   |  |                                    |

| <b>деятельности</b>   |  |  |                                    |
|---|--|--|------------------------------------|
| <p><b>Знать:</b><br/>номенклатуру информационных источников, применяемых в профессиональной деятельности приемы структурирования информации; формат оформления результатов поиска информации<br/>современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности</p>  | <p><b>Получение систематических знаний</b> о возможностях PostgreSQL, конфигурационных файлах (postgresql.conf, pg_hba.conf), утилитах (pg_dump, pg_restore, psql)</p>                       | <p><b>Уровень знаний</b><br/>официальной документации PostgreSQL, типовых приёмов администрирования, методов мониторинга и бэкапа</p>            | <p><b>Т(1-35)</b></p>              |
| <p><b>Уметь:</b><br/>определять задачи для поиска информации; определять необходимые источники информации<br/>планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации оформлять результаты поиска; применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение и цифровые средства</p> | <p><b>Сформировать систематическое умение</b> настраивать подключения, создавать резервные копии, восстанавливать БД, импортировать данные из CSV/JSON, оптимизировать запросы</p>           | <p><b>Уровень умения</b><br/>применять инструменты PostgreSQL для администрирования, контроля производительности и защиты данных</p>             | <p><b>Т(1-35)<br/>ПЗ(1-16)</b></p> |
| <p><b>Владеть:</b><br/>навыками определения цели поиска<br/>навыками структурирования и анализа информации<br/>навыками применения ИТ для решения профессиональных задач</p>  | <p><b>Сформировать систематическое владение</b> приёмами администрирования PostgreSQL: создание ролей, управление привилегиями, настройка SSL, настройка автоматической очистки (vacuum)</p> | <p><b>Уровень владения</b><br/>современными средствами администрирования СУБД для обеспечения доступности, целостности и безопасности данных</p> | <p><b>Т(1-35)<br/>ПЗ(1-16)</b></p> |

*Т – тестовые задания, ПЗ – практические задания.*

**2 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы**

## Тестовые задания:

1. Что такое СУБД?
  - a. совокупность данных, организованных по определённым правилам
  - b. программное обеспечение, предназначенное для создания, ведения и совместного использования баз данных
  - c. язык структурированных запросов к данным
  - d. аппаратный комплекс для хранения резервных копий
2. Что является отношением в реляционной модели данных?
  - a. связь между двумя базами данных
  - b. таблица, состоящая из строк и столбцов
  - c. логическое выражение в условии WHERE
  - d. операция соединения двух таблиц
3. Первичный ключ (PRIMARY KEY) в таблице предназначен для:
  - a. ускорения выполнения запросов SELECT
  - b. организации связи с другой таблицей
  - c. однозначной идентификации каждой записи в таблице
  - d. хранения зашифрованных данных
4. Какой этап проектирования базы данных выполняется первым?
  - a. физическое проектирование
  - b. нормализация отношений
  - c. концептуальное (инфологическое) проектирование
  - d. создание индексов
5. Какое условие необходимо выполнить для приведения таблицы ко второй нормальной форме (2НФ)?
  - a. таблица должна находиться в 1НФ, и все неключевые атрибуты должны зависеть от полного первичного ключа (не должно быть частичной зависимости)
  - b. таблица не должна содержать повторяющихся групп данных
  - c. все атрибуты должны быть атомарными
  - d. не должно быть транзитивных зависимостей между неключевыми атрибутами
6. Какой из перечисленных типов данных не поддерживается в PostgreSQL?
  - a. JSONB
  - b. INET
  - c. IMAGE
  - d. UUID
7. К какой группе команд SQL относится оператор GRANT?
  - a. DDL (Data Definition Language)
  - b. DML (Data Manipulation Language)
  - c. DCL (Data Control Language)
  - d. TCL (Transaction Control Language)
8. Какой оператор SQL используется для добавления новых строк в таблицу?
  - a. UPDATE
  - b. INSERT
  - c. ALTER
  - d. CREATE
9. Какой тип JOIN возвращает только те строки, для которых найдено соответствие в обеих соединяемых таблицах?
  - a. LEFT JOIN
  - b. RIGHT JOIN
  - c. INNER JOIN
  - d. FULL OUTER JOIN
10. В каком случае используется оператор EXISTS в подзапросе?
  - a. для сравнения значения с результатом подзапроса, возвращающего ровно одно значение
  - b. для проверки, возвращает ли подзапрос хотя бы одну строку

- c. для проверки вхождения значения в множество значений, возвращаемых подзапросом
  - d. для вычисления агрегатной функции над результатом подзапроса
11. Какой оператор обязательно должен присутствовать в запросе, содержащем агрегатную функцию и GROUP BY для фильтрации сгруппированных данных?
- a. WHERE
  - b. HAVING
  - c. FILTER
  - d. ORDER BY
12. Для чего в PostgreSQL в первую очередь используются индексы типа B-Tree?
- a. для полнотекстового поиска
  - b. для ускорения операций с массивами
  - c. для ускорения поиска по операторам сравнения (=, >, <, BETWEEN)
  - d. для обеспечения уникальности значений внешних ключей
13. Какое свойство ACID гарантирует, что транзакция будет выполнена полностью или не выполнена вовсе?
- a. Consistency (Согласованность)
  - b. Isolation (Изолированность)
  - c. Durability (Долговечность)
  - d. Atomicity (Атомарность)
14. Что такое представление (VIEW) в базе данных?
- a. физическая таблица, хранящая копию данных другой таблицы
  - b. виртуальная таблица, содержимое которой определяется запросом SELECT
  - c. зашифрованное хранилище паролей пользователей
  - d. специальный тип индекса для ускорения запросов
15. Чем хранимая процедура в PostgreSQL отличается от функции?
- a. процедура не возвращает значение (кроме выходных параметров) и вызывается оператором CALL, функция возвращает значение и может использоваться в SELECT
  - b. процедура может содержать транзакции, а функция — нет
  - c. функция всегда пишется на языке SQL, процедура — только на PL/pgSQL
  - d. процедуры существуют только на уровне базы данных, функции — на уровне экземпляра сервера
16. На каком уровне срабатывает триггер с параметром FOR EACH ROW?
- a. на уровне всего сервера при запуске
  - b. для каждой изменяемой строки
  - c. один раз на весь оператор SQL
  - d. при каждом подключении клиента к базе данных
17. Какой механизм обеспечения целостности данных не относится к декларативным ограничениям?
- a. PRIMARY KEY
  - b. FOREIGN KEY
  - c. CHECK
  - d. ТРИГГЕР (TRIGGER)
18. Какая утилита PostgreSQL используется для создания логической резервной копии отдельной базы данных в виде SQL-скрипта или архивного файла?
- a. pg\_basebackup
  - b. pg\_dump
  - c. pg\_restore
  - d. pg\_ctl
19. Какой командой можно восстановить базу данных из файла дампа, созданного командой pg\_dump -Fc (custom format)?
- a. psql -f dump.sql
  - b. pg\_restore -d dbname dump.dump
  - c. pg\_ctl restore
  - d. pg\_recovery

20. Какая команда SQL используется для выдачи права на чтение таблицы другому пользователю?
- REVOKE SELECT ON table FROM user;
  - GRANT SELECT ON table TO user;
  - ALLOW SELECT ON table TO user;
  - CREATE PERMISSION SELECT ON table FOR user;
21. В каком файле настраиваются методы аутентификации клиентов PostgreSQL (например, trust, md5, scram-sha-256)?
- postgresql.conf
  - pg\_hba.conf
  - pg\_ident.conf
  - pg\_auth.conf
22. Что такое схема (schema) в PostgreSQL?
- графическое представление структуры базы данных
  - пространство имён (логический контейнер) для группировки таблиц, представлений и других объектов внутри базы данных
  - описание связей между таблицами
  - синоним термина «база данных»
23. Какой параметр необходимо установить в файле postgresql.conf для включения поддержки SSL-соединений?
- ssl = on
  - ssl = on (и указать пути к сертификатам)
  - enable\_ssl = true
  - encryption = 'ssl'
24. Какова основная задача процесса autovacuum в PostgreSQL?
- автоматическое создание резервных копий
  - автоматическая очистка «мёртвых» кортежей (dead tuples) и обновление статистики для планировщика
  - автоматическая перезагрузка сервера при сбое
  - автоматическое применение обновлений безопасности
25. В каком системном представлении PostgreSQL содержится информация о текущих активных сессиях и выполняемых запросах?
- pg\_locks
  - pg\_stat\_user\_tables
  - pg\_stat\_activity
  - pg\_current\_sessions
26. Что показывает команда EXPLAIN SELECT \* FROM users WHERE id = 1;?
- фактическое время выполнения запроса
  - план выполнения запроса (какие индексы используются, порядок сканирования таблиц)
  - результат выполнения запроса
  - список ошибок в синтаксисе запроса
27. Какой конфигурационный файл PostgreSQL содержит основные параметры сервера (общая память, количество подключений, WAL)?
- pg\_hba.conf
  - postgresql.conf
  - pg\_service.conf
  - pg\_ctl.conf
28. Какая команда SQL используется для импорта данных из CSV-файла в таблицу PostgreSQL?
- IMPORT FROM 'file.csv' INTO table;
  - COPY table FROM 'file.csv' DELIMITER ',' CSV HEADER;
  - LOAD DATA INFILE 'file.csv' INTO TABLE table;
  - BULK INSERT table FROM 'file.csv';
29. С помощью какой команды в psql можно сохранить результат запроса в текстовый файл?
- \save 'file.txt'

- b. \o file.txt
  - c. \export 'file.txt'
  - d. \write 'file.txt'
30. Какое расширение PostgreSQL предоставляет сбор подробной статистики по использованию функций и запросов (позволяет искать «тяжёлые» запросы)?
- a. pg\_stat\_statements
  - b. pg\_stat\_statements
  - c. pg\_buffercache
  - d. pg\_prewarm
31. Для чего служит механизм WAL (Write-Ahead Log) в PostgreSQL?
- a. для ведения журнала ошибок приложения
  - b. для обеспечения долговечности (Durability) данных и возможности восстановления после сбоя
  - c. для протоколирования всех команд пользователей
  - d. для синхронизации репликации между мастером и слейвом в реальном времени
32. Какой вид репликации в PostgreSQL подразумевает передачу и применение изменений на уровне блоков WAL?
- a. логическая репликация
  - b. физическая (потокковая) репликация
  - c. триггерная репликация
  - d. синхронная репликация на уровне приложения
33. Какой тип блокировки в PostgreSQL является наиболее строгим (конфликтует со всеми остальными блокировками, включая SELECT)?
- a. ROW EXCLUSIVE
  - b. SHARE UPDATE EXCLUSIVE
  - c. ACCESS EXCLUSIVE
  - d. SHARE ROW EXCLUSIVE
34. Для чего выполняется команда VACUUM в PostgreSQL?
- a. для сжатия табличного пространства на диске
  - b. для пометки «мёртвых» кортежей как свободного пространства для повторного использования
  - c. для удаления дублирующихся записей в таблице
  - d. для переиндексации всех индексов таблицы
35. Какой параметр в postgresql.conf отвечает за включение записи в лог всех выполняемых SQL-запросов (или запросов дольше определённого времени)?
- a. log\_destination = 'stderr'
  - b. log\_statement = 'all' (или log\_min\_duration\_statement)
  - c. logging\_collector = on
  - d. debug\_print\_parse = on

**Критерии оценивания:**

- 5 баллов выставляется, если правильные ответы даны на 85-100% тестовых заданий
- 4 балла выставляется студенту, если правильные ответы даны на 65-84% тестовых заданий
- 3 балла выставляется студенту, если правильные ответы даны на 50-64% тестовых заданий
- 2 балла выставляется студенту, если правильные ответы даны на менее 50% тестовых заданий

## Практические задания:

### №1

Тема: Сбор и анализ информации для проектирования базы данных

Цель: Сформировать навыки сбора, систематизации и документирования требований к базе данных на основе описания предметной области.

Теоретические вопросы:

1. Что понимается под предметной областью? Приведите примеры.
2. Какие сущности можно выделить в предметной области «Библиотека»?
3. Что такое атрибут сущности? Приведите примеры ключевых и неключевых атрибутов.
4. Какие типы связей между сущностями существуют (1:1, 1:M, M:N)?
5. Для чего составляется техническое задание на разработку базы данных?

Оборудование и ПО: Персональный компьютер, текстовый редактор, средство построения диаграмм ([draw.io](http://draw.io), MS Visio – опционально).

Задания:

1. Ознакомиться с описанием предметной области «Автосервис» (выдаётся преподавателем индивидуально или по вариантам).
2. Выделить сущности предметной области и определить их атрибуты. Результат оформить в виде таблицы:
  - Сущность
  - Атрибуты (с указанием предполагаемого типа данных)
  - Первичный ключ
3. Определить связи между сущностями, указать тип каждой связи (1:1, 1:M, M:N).
4. Сформулировать 4–5 функциональных требований к будущей базе данных (например, «хранить данные о клиентах и их автомобилях», «получать список выполненных работ за период»).
5. Подготовить отчёт, содержащий:
  - название предметной области;
  - таблицу сущностей и атрибутов;
  - описание связей;
  - перечень функциональных требований.

Контрольные вопросы:

1. Как определить, является ли некоторое понятие сущностью или атрибутом?
2. Как поступить, если между двумя сущностями обнаружена связь M:N?
3. Что такое домен атрибута? Приведите пример.
- 4.

### №2

Тема: Построение ER-диаграммы и разработка реляционной схемы базы данных

Цель: Приобрести практические навыки построения ER-диаграммы предметной области и её преобразования в реляционную схему (набор таблиц).

Теоретические вопросы:

1. Что такое ER-диаграмма? Какие основные элементы она содержит?
2. Какие нотации построения ER-диаграмм вам известны (Чена, IE, IDEF1X)?
3. Как на ER-диаграмме отображаются связи различной мощности?
4. Каковы правила преобразования ER-диаграммы в набор таблиц?

Оборудование и ПО: Персональный компьютер, средство построения диаграмм ([draw.io](http://draw.io), Dia, MS Visio) или миллиметровая бумага.

Задания:

1. На основе результатов практической работы №1 построить ER-диаграмму предметной области в любой нотации. На диаграмме должны быть отображены сущности, атрибуты, первичные ключи и связи с указанием их типа.
2. Преобразовать полученную ER-диаграмму в реляционную схему. Для каждой сущности определить:
  - имя будущей таблицы;
  - перечень столбцов с ориентировочными типами данных;
  - первичный ключ;

- внешние ключи (если есть).
- 3. Для связи M:N создать дополнительную связующую таблицу.
- 4. Подготовить SQL-скрипт создания всех таблиц (операторы CREATE TABLE с указанием первичных и внешних ключей).
- 5. Оформить отчёт: ER-диаграмма (скриншот), реляционная схема (табличное описание), SQL-скрипт.

Контрольные вопросы:

1. Каково назначение внешнего ключа в реляционной таблице?
2. Что означает опция ON DELETE CASCADE при определении внешнего ключа?
3. Может ли таблица иметь два разных первичных ключа? Почему?

### №3

Тема: Нормализация базы данных (приведение к 1НФ, 2НФ, 3НФ)

Цель: Освоить методику нормализации отношений; научиться выявлять и устранять избыточность данных, приводя таблицы к третьей нормальной форме.

Теоретические вопросы:

1. Что такое функциональная зависимость? Приведите пример полной и частичной функциональной зависимости.
2. Сформулируйте определения первой, второй и третьей нормальных форм.
3. Что такое транзитивная зависимость? Почему её наличие может приводить к аномалиям обновления?
4. В каких случаях допустима денормализация базы данных?

Оборудование и ПО: Персональный компьютер, текстовый редактор.

Задания:

1. Дано ненормализованное отношение (таблица) «Заказы»:

| № заказа | Дата       | Клиент | Телефон клиента | Товар      | Цена  | Количество |
|----------|------------|--------|-----------------|------------|-------|------------|
| 101      | 01.03.2026 | Иванов | 111-22-33       | Монитор    | 15000 | 1          |
| 101      | 01.03.2026 | Иванов | 111-22-33       | Клавиатура | 2000  | 2          |
| 102      | 02.03.2026 | Петров | 222-33-44       | Мышь       | 800   | 1          |

2. Выполнить последовательную нормализацию:
  - Привести отношение к 1НФ (устранить повторяющиеся группы, обеспечить атомарность атрибутов).
  - Привести отношение к 2НФ (устранить частичные зависимости неключевых атрибутов от составного ключа).
  - Привести отношение к 3НФ (устранить транзитивные зависимости).
3. Для полученного набора таблиц определить первичные и внешние ключи.
4. Написать SQL-скрипт создания таблиц в 3НФ.
5. Оформить отчёт, включив в него исходную таблицу, промежуточные результаты нормализации, финальную схему и вывод о том, какие аномалии были устранены.

Контрольные вопросы:

1. Почему отношение, находящееся в 1НФ, но не во 2НФ, может содержать избыточность?
2. Приведите пример транзитивной зависимости в отношении «Студент» (НомерЗачётки, ФИО, Группа, Факультет, Декан).
3. Всегда ли следует нормализовать базу данных до 3НФ? Обоснуйте ответ.
- 4.

### №4

Тема: Создание базы данных и таблиц в PostgreSQL (DDL)

Цель: Научиться создавать базу данных, таблицы, задавать ограничения целостности с помощью команд языка DDL в среде PostgreSQL.

Теоретические вопросы:

1. Какие команды языка SQL относятся к группе DDL?
2. Какие типы данных PostgreSQL используются для хранения целых чисел, строк переменной длины, даты и времени?
3. Какие ограничения целостности можно задать при создании таблицы (PRIMARY KEY, FOREIGN KEY, NOT NULL, UNIQUE, CHECK)?
4. Для чего используется ключевое слово SERIAL?

Оборудование и ПО: ПК с ОС Red OS Murom, PostgreSQL 15+, клиент psql или pgAdmin 4.

Задания:

1. Подключиться к серверу PostgreSQL под учётной записью суперпользователя.
2. Создать базу данных university и подключиться к ней.
3. На основе схемы, разработанной в работе №2, создать таблицы groups, students, disciplines, grades. Пример создания таблицы groups:

sql

```
CREATE TABLE groups (  
  id SERIAL PRIMARY KEY,  
  name VARCHAR(50) NOT NULL,  
  year INT CHECK (year >= 2000),  
  speciality VARCHAR(100)  
);
```

4. Для каждой таблицы определить не менее одного ограничения CHECK (например, допустимый диапазон оценки).
5. Убедиться, что таблицы созданы, с помощью команды \dt.
6. Просмотреть структуру таблицы students командой \d students.
7. Внести изменения в структуру таблицы: добавить столбец email в таблицу students (ALTER TABLE students ADD COLUMN email VARCHAR(100) UNIQUE).
8. Сохранить все выполненные SQL-команды в текстовый файл. Подготовить отчёт со скриншотами.

Контрольные вопросы:

1. Можно ли создать таблицу без первичного ключа? К каким проблемам это может привести?
2. Чем отличается ограничение UNIQUE от PRIMARY KEY?
3. Как удалить таблицу вместе со всеми данными?

## №5

Тема: Основы SQL: SELECT, INSERT, UPDATE, DELETE

Цель: Освоить базовые команды манипулирования данными (DML) в PostgreSQL: вставка, выборка, обновление и удаление записей.

Теоретические вопросы:

1. Какие команды входят в группу DML?
2. Какова структура оператора SELECT (обязательные и необязательные разделы)?
3. Какие операторы сравнения и логические операторы используются в условии WHERE?
4. Для чего служит ключевое слово DISTINCT?

Оборудование и ПО: ПК с ОС Red OS Murom, PostgreSQL 15+, клиент psql или pgAdmin 4.

Задания:

1. В базе данных university наполнить таблицы тестовыми данными (не менее 5 записей в каждой) с помощью команды INSERT. Пример:

sql

```
INSERT INTO groups (name, year, speciality) VALUES  
(‘ИС-21’, 2024, ‘Информационные системы’),  
(‘ПО-22’, 2024, ‘Программное обеспечение’);
```

2. Выполнить выборку всех данных из таблицы students.
3. Выполнить выборку с условиями:
  - студентов, чьё имя начинается на букву «А» (WHERE name LIKE 'A%');
  - студентов, родившихся после 01.01.2005.
4. Обновить данные: изменить название специальности у группы ИС-21 на «Информационные системы и программирование».

5. Удалить запись о студенте, у которого номер зачётной книжки равен заданному значению (предварительно добавив такого студента).
6. Выполнить удаление всех записей из таблицы grades, но сохранить структуру таблицы.
7. Оформить отчёт: привести все SQL-запросы и скриншоты результатов.

Контрольные вопросы:

1. Что произойдёт при выполнении команды DELETE FROM students; без условия WHERE?
2. Как вставить несколько строк одним оператором INSERT?
3. Каким образом можно отменить изменения, выполненные командой UPDATE?

## №6

Тема: Сложные запросы: JOIN, подзапросы, агрегация данных

Цель: Сформировать умение составлять многотабличные запросы с использованием различных типов соединений, подзапросов и агрегатных функций.

Теоретические вопросы:

1. Какие типы JOIN поддерживаются в PostgreSQL? В чём разница между INNER JOIN и LEFT OUTER JOIN?
2. Что такое подзапрос? Где он может располагаться в операторе SELECT?
3. Перечислите основные агрегатные функции. Для чего используется конструкция GROUP BY?
4. Чем условие HAVING отличается от WHERE?

Оборудование и ПО: ПК с ОС Red OS Murom, PostgreSQL 15+, клиент psql или pgAdmin 4.

Задания:

1. Написать запрос, выводящий список студентов с названиями их групп (использовать INNER JOIN).
2. Написать запрос, выводящий все группы, включая те, в которых нет студентов (использовать LEFT JOIN и RIGHT JOIN для демонстрации различий).
3. Написать запрос, вычисляющий средний балл каждого студента по всем дисциплинам. Отсортировать результат по убыванию среднего балла.
4. Написать запрос, выводящий дисциплины, по которым средний балл выше 4.0 (использовать HAVING).
5. Используя подзапрос с EXISTS, найти студентов, у которых есть хотя бы одна оценка «5».
6. Используя подзапрос с IN, найти студентов, обучающихся в группах, год поступления которых равен 2024.
7. Подготовить отчёт со всеми запросами и скриншотами результатов.

Контрольные вопросы:

1. В каком случае LEFT JOIN и RIGHT JOIN могут дать одинаковый результат?
2. Можно ли использовать агрегатные функции в условии WHERE? Почему?
3. В чём преимущество использования EXISTS перед IN в подзапросах?

## №7

Тема: Создание и использование индексов. Анализ плана выполнения запроса (EXPLAIN)

Цель: Научиться создавать индексы для ускорения выполнения запросов и анализировать план выполнения запроса с помощью команды EXPLAIN.

Теоретические вопросы:

1. Что такое индекс? Какие преимущества и недостатки даёт его использование?
2. Какие типы индексов поддерживает PostgreSQL (B-Tree, Hash, GIN, GiST, BRIN)?
3. В каких случаях использование индекса нецелесообразно?
4. Что показывает команда EXPLAIN? Как интерпретировать её вывод?

Оборудование и ПО: ПК с ОС Red OS Murom, PostgreSQL 15+, клиент psql.

Задания:

1. Создать таблицу employees с большим количеством строк (не менее 100 000) с помощью генератора (например, использовать generate\_series).
2. Выполнить запрос на выборку сотрудников по полю last\_name без индекса и зафиксировать время выполнения (\timing on).

3. Выполнить команду EXPLAIN ANALYZE для этого же запроса и проанализировать план выполнения.
4. Создать индекс типа B-Tree на столбец last\_name:

sql

```
CREATE INDEX idx_employees_lastname ON employees(last_name);
```

5. Повторно выполнить тот же запрос с измерением времени и командой EXPLAIN ANALYZE. Сравнить результаты.
6. Создать составной индекс на столбцы department и salary.
7. Написать запрос с условиями по этим двум полям и проверить использование индекса с помощью EXPLAIN.
8. Удалить созданные индексы командой DROP INDEX.
9. Подготовить отчёт с выводами о влиянии индексов на производительность.

Контрольные вопросы:

1. Какой тип индекса следует выбрать для столбца, по которому часто выполняется поиск по точному совпадению?
2. Что означает строка «Seq Scan» в выводе EXPLAIN?
3. Может ли индекс замедлить операции INSERT и UPDATE? Почему?

## №8

Тема: Транзакции и блокировки (ACID, BEGIN, COMMIT, ROLLBACK)

Цель: Изучить механизм транзакций, свойства ACID, научиться управлять транзакциями и анализировать блокировки в PostgreSQL.

Теоретические вопросы:

1. Дайте определение транзакции. Какие свойства ACID она должна обеспечивать?
2. Какие команды используются для управления транзакциями?
3. Что такое уровень изоляции транзакции? Какие уровни поддерживает PostgreSQL?
4. Что такое блокировки? Какие типы блокировок вы знаете?

Оборудование и ПО: ПК с ОС Red OS Murom, PostgreSQL 15+, два одновременных сеанса psql (в разных терминалах).

Задания:

1. В первом сеансе начать транзакцию (BEGIN), вставить новую запись в таблицу students, не завершая транзакцию.
2. Во втором сеансе попытаться прочитать эту же таблицу. Убедиться, что вставленная запись не видна (уровень изоляции Read Committed).
3. В первом сеансе выполнить COMMIT. Во втором сеансе снова выполнить SELECT – запись должна появиться.
4. Продемонстрировать откат транзакции: начать транзакцию, удалить несколько строк, выполнить ROLLBACK. Проверить, что данные остались неизменными.
5. Создать ситуацию взаимоблокировки (deadlock) между двумя сеансами (например, обновление двух разных строк в противоположном порядке). Зафиксировать сообщение об ошибке.
6. Изучить системное представление pg\_locks для анализа активных блокировок.
7. Подготовить отчёт с описанием экспериментов и скриншотами.

Контрольные вопросы:

1. Что произойдёт, если во время выполнения транзакции произойдёт сбой питания сервера?
2. Чем уровень изоляции REPEATABLE READ отличается от READ COMMITTED?
3. Каким образом PostgreSQL разрешает взаимоблокировки?

## №9

Тема: Представления (VIEW) и хранимые процедуры/функции

Цель: Научиться создавать и использовать представления, а также разрабатывать простые хранимые процедуры и функции на языке PL/pgSQL.

Теоретические вопросы:

1. Что такое представление (VIEW)? Для каких целей оно используется?
2. Можно ли выполнять операции INSERT, UPDATE, DELETE над представлением? При каких условиях?

3. Чем хранимая процедура отличается от функции в PostgreSQL?
4. Какие основные конструкции языка PL/pgSQL вы знаете?

Оборудование и ПО: ПК с ОС Red OS Murom, PostgreSQL 15+, клиент psql или pgAdmin 4.

Задания:

1. Создать представление student\_grades\_view, которое выводит ФИО студента, название дисциплины и оценку (результат соединения трёх таблиц).
2. Выполнить выборку из представления.
3. Попыаться обновить оценку через представление (если представление не обновляется, объяснить причину).
4. Создать простую функцию get\_student\_avg(student\_id INT), возвращающую средний балл студента. Использовать язык SQL или PL/pgSQL.

sql

```
CREATE OR REPLACE FUNCTION get_student_avg(s_id INT)
RETURNS NUMERIC AS $$
DECLARE
    avg_grade NUMERIC;
BEGIN
    SELECT AVG(grade) INTO avg_grade FROM grades WHERE student_id = s_id;
    RETURN avg_grade;
END;
$$ LANGUAGE plpgsql;
```

5. Вызвать функцию для нескольких студентов.
6. Создать хранимую процедуру update\_student\_email(student\_id INT, new\_email VARCHAR), которая обновляет email студента. Вызвать её с помощью CALL.
7. Удалить созданные объекты.
8. Подготовить отчёт с кодом объектов и результатами вызовов.

Контрольные вопросы:

1. В чём преимущество использования представлений?
2. Можно ли внутри функции выполнять транзакции (COMMIT, ROLLBACK)?
3. Как передать в функцию значение по умолчанию для параметра?

## №10

Тема: Триггеры и обеспечение целостности данных

Цель: Изучить механизм триггеров в PostgreSQL, научиться создавать триггеры для автоматизации проверки целостности и ведения аудита.

Теоретические вопросы:

1. Что такое триггер? Какие события могут вызвать его срабатывание?
2. В чём отличие триггеров BEFORE от AFTER?
3. Для чего нужна переменная NEW и OLD в триггерной функции?
4. Чем отличается триггер уровня строки (FOR EACH ROW) от триггера уровня оператора (FOR EACH STATEMENT)?

Оборудование и ПО: ПК с ОС Red OS Murom, PostgreSQL 15+, клиент psql или pgAdmin 4.

Задания:

1. Создать таблицу audit\_log для записи действий пользователей:

sql

```
CREATE TABLE audit_log (
    id SERIAL PRIMARY KEY,
    table_name TEXT,
    operation TEXT,
    changed_at TIMESTAMPTZ DEFAULT now(),
    user_name TEXT
);
```

2. Создать триггерную функцию log\_student\_changes(), которая при вставке, обновлении или удалении записи в таблице students добавляет соответствующую запись в audit\_log.
3. Создать триггер student\_audit\_trigger, срабатывающий AFTER INSERT OR UPDATE OR DELETE ON students FOR EACH ROW.

4. Протестировать триггер: выполнить вставку, обновление и удаление студента, проверить содержимое `audit_log`.
5. Создать триггер BEFORE INSERT OR UPDATE ON students, проверяющий, что значение поля email соответствует формату (содержит символ '@'). При несоответствии выбрасывать исключение.

sql

```
IF NEW.email NOT LIKE '%@%' THEN
    RAISE EXCEPTION 'Некорректный email: %', NEW.email;
END IF;
```

6. Протестировать ограничивающий триггер.
7. Оформить отчёт с кодом триггеров и скриншотами результатов.

Контрольные вопросы:

1. Может ли триггер вызвать другой триггер? Как избежать бесконечной рекурсии?
2. Можно ли в триггере BEFORE изменить значение столбца NEW?
3. Как отключить триггер на время выполнения массовой загрузки данных?

## №11

Тема: Экспорт и импорт данных (COPY, CSV, JSON)

Цель: Освоить приёмы экспорта данных из таблиц PostgreSQL в файлы форматов CSV и JSON, а также импорта данных из внешних источников.

Теоретические вопросы:

1. Какая команда SQL используется для импорта/экспорта данных в PostgreSQL?
2. В чём разница между командами COPY и \copy в psql?
3. Какие параметры формата CSV можно задать в команде COPY (разделитель, заголовок, кавычки)?
4. Как можно экспортировать результат произвольного запроса в файл?

Оборудование и ПО: ПК с ОС Red OS Murom, PostgreSQL 15+, клиент psql.

Задания:

1. Создать таблицу products и наполнить её тестовыми данными (не менее 15 записей).
2. Экспортировать содержимое таблицы products в CSV-файл с заголовком и разделителем «;»:

sql

```
\copy products TO '/tmp/products.csv' DELIMITER ';' CSV HEADER;
```

3. Экспортировать результат запроса (только товары дороже 1000 руб.) в файл expensive.csv.
4. Экспортировать данные в JSON-формат с помощью функции row\_to\_json и перенаправления вывода:

bash

```
psql -d dbname -c "SELECT row_to_json(products) FROM products;" -A -t > products.json
```

5. Очистить таблицу products и импортировать данные обратно из CSV-файла.
6. Создать новую таблицу imported\_products и импортировать в неё данные из CSV-файла, предварительно созданного в п.2.
7. Проверить корректность импорта.
8. Подготовить отчёт с командами и скриншотами первых строк файлов.

Контрольные вопросы:

1. Какие права необходимы для выполнения команды COPY TO на сервере?
2. Как обработать ситуацию, когда в CSV-файле встречаются символы-разделители внутри текстовых полей?
3. Можно ли импортировать данные из CSV-файла в таблицу с дополнительным автоинкрементным столбцом?

## №12

Тема: Резервное копирование и восстановление базы данных (pg\_dump, pg\_restore)

Цель: Сформировать практические навыки создания логических резервных копий баз данных PostgreSQL с помощью утилиты pg\_dump и восстановления из них с помощью pg\_restore и psql.

Теоретические вопросы:

1. Какие виды резервного копирования поддерживает PostgreSQL (логическое, физическое)?
2. Чем отличается резервная копия в формате plain SQL от custom-формата?
3. Какие параметры pg\_dump позволяют скопировать только схему или только данные?
4. Каким образом можно автоматизировать регулярное резервное копирование?

Оборудование и ПО: ПК с ОС Red OS Murom, PostgreSQL 15+, утилиты командной строки.

Задания:

1. Создать тестовую базу данных backup\_test с несколькими таблицами и данными.
2. Выполнить резервное копирование в виде SQL-скрипта:

bash

```
pg_dump backup_test > backup_test.sql
```

3. Выполнить резервное копирование только схемы базы данных:

bash

```
pg_dump -s backup_test > backup_test_schema.sql
```

4. Выполнить резервное копирование в сжатом custom-формате:

bash

```
pg_dump -Fc backup_test -f backup_test.dump
```

5. Создать новую пустую базу данных restore\_test.

6. Восстановить данные в неё из SQL-скрипта с помощью psql:

bash

```
psql restore_test < backup_test.sql
```

7. Удалить базу restore\_test, создать заново и восстановить из custom-формата с помощью pg\_restore:

bash

```
pg_restore -d restore_test backup_test.dump
```

8. Выполнить резервное копирование отдельной таблицы students:

bash

```
pg_dump -t students backup_test > students_backup.sql
```

9. Подготовить отчёт с описанием выполненных команд и проверкой восстановленных данных.

Контрольные вопросы:

1. Почему не рекомендуется выполнять физическое копирование файлов данных при работающем сервере PostgreSQL?
2. Что произойдёт, если восстанавливать базу из дампа поверх существующей базы с тем же именем?
3. Какая утилита используется для создания резервной копии на уровне файловой системы (физический бэкап)?

### №13

Тема: Управление пользователями, ролями и привилегиями (GRANT, REVOKE)

Цель: Изучить механизмы управления доступом в PostgreSQL: создание ролей, назначение привилегий, использование групповых ролей.

Теоретические вопросы:

1. Чем роль отличается от пользователя в терминологии PostgreSQL?
2. Какие привилегии могут быть предоставлены на таблицу, представление, схему?
3. Что означает опция WITH GRANT OPTION?
4. Как просмотреть текущие привилегии, назначенные роли?

Оборудование и ПО: ПК с ОС Red OS Murom, PostgreSQL 15+, клиент psql с правами суперпользователя.

Задания:

1. Создать три роли с возможностью входа:

sql

```
CREATE ROLE hr_user WITH LOGIN PASSWORD 'hr123';
```

```
CREATE ROLE manager WITH LOGIN PASSWORD 'mgr123';
```

```
CREATE ROLE auditor WITH LOGIN PASSWORD 'aud123';
```

2. Создать базу данных hr и таблицу employees.

3. Предоставить роли hr\_user права SELECT, INSERT, UPDATE на таблицу employees.

4. Предоставить роли manager права SELECT на таблицу employees.
5. Создать групповую роль hr\_staff и включить в неё hr\_user и manager:

sql

```
CREATE ROLE hr_staff NOLOGIN;
```

```
GRANT hr_staff TO hr_user, manager;
```

6. Предоставить групповой роли hr\_staff право SELECT на таблицу departments.
7. Проверить привилегии, подключившись к базе под каждой ролью, и выполнить операции SELECT, INSERT, UPDATE, DELETE.
8. Отозвать право DELETE у роли hr\_user (если оно было) и показать, что удаление становится невозможным.
9. Используя представление information\_schema.table\_privileges, вывести список привилегий на таблицу employees.
10. Оформить отчёт с командами, скриншотами успешных и неуспешных операций.

Контрольные вопросы:

1. Что произойдёт с привилегиями роли, если она будет удалена?
2. В чём разница между GRANT на таблицу и GRANT на все таблицы в схеме?
3. Можно ли запретить пользователю подключаться к определённой базе данных?

#### №14

Тема: Настройка аутентификации и SSL-соединения (pg\_hba.conf, SSL-сертификаты)

Цель: Научиться настраивать методы аутентификации клиентов в файле pg\_hba.conf и включать шифрование соединений с помощью SSL.

Теоретические вопросы:

1. Для чего предназначен файл pg\_hba.conf? Какова его структура?
2. Какие методы аутентификации поддерживает PostgreSQL (trust, md5, scram-sha-256, cert, peer)?
3. Какие шаги необходимо выполнить для включения поддержки SSL на сервере PostgreSQL?
4. Как проверить, что соединение с сервером защищено с помощью SSL?

Оборудование и ПО: ПК с ОС Red OS Murom, PostgreSQL 15+, доступ к конфигурационным файлам с правами sudo.

Задания:

1. Просмотреть текущее содержимое файла pg\_hba.conf (sudo cat /etc/postgresql/15/main/pg\_hba.conf).
2. Создать тестовую роль testssl с паролем.
3. Настроить правило в pg\_hba.conf, разрешающее подключение для роли testssl с локального хоста только по методу scram-sha-256.
4. Перезагрузить конфигурацию сервера (sudo systemctl reload postgresql).
5. Проверить подключение с правильным и неправильным паролем.
6. Включить поддержку SSL: убедиться, что в postgresql.conf параметр ssl = on, и указаны пути к сертификату и ключу. При необходимости создать самоподписанный сертификат с помощью openssl.
7. Настроить правило в pg\_hba.conf с опцией hostssl для принудительного использования SSL-соединений.
8. Подключиться к серверу с параметром sslmode=require и убедиться, что соединение зашифровано (команда \conninfo в psql).
9. Подготовить отчёт с фрагментами конфигурационных файлов и скриншотами проверок.

Контрольные вопросы:

1. Чем метод аутентификации peer отличается от md5?
2. Что означает запись host all all 0.0.0.0/0 md5 в pg\_hba.conf? Какие риски она несёт?
3. Обязательно ли перезапускать сервер после изменения pg\_hba.conf?

#### №15

Тема: Мониторинг производительности и настройка автовакуума (pg\_stat\_activity, autovacuum)

Цель: Изучить основные инструменты мониторинга активности и производительности в PostgreSQL, а также научиться управлять процессом автоматической очистки (autovacuum).

Теоретические вопросы:

1. Какие системные представления используются для мониторинга текущей активности (pg\_stat\_activity, pg\_stat\_user\_tables, pg\_stat\_bgwriter)?
2. Что такое «мёртвые кортежи» (dead tuples)? Как они влияют на производительность?
3. Для чего нужен процесс autovacuum? Какие параметры управляют его работой?
4. Какие метрики можно получить из представления pg\_stat\_user\_tables?

Оборудование и ПО: ПК с ОС Red OS Murom, PostgreSQL 15+, клиент psql.

Задания:

1. Выполнить запрос к pg\_stat\_activity для просмотра всех активных соединений и выполняемых запросов.
2. Выполнить несколько долгих запросов (например, SELECT pg\_sleep(10)) в отдельных сеансах и снова проанализировать pg\_stat\_activity.
3. Создать таблицу и выполнить множество операций UPDATE и DELETE, чтобы создать «мёртвые кортежи».
4. Проверить количество живых и мёртвых кортежей в таблице с помощью запроса к pg\_stat\_user\_tables (столбцы n\_live\_tup, n\_dead\_tup).
5. Выполнить ручную очистку таблицы командой VACUUM и снова проверить количество мёртвых кортежей.
6. Выполнить команду ANALYZE для обновления статистики планировщика.
7. Просмотреть параметры конфигурации, связанные с autovacuum (SHOW autovacuum;, SELECT name, setting FROM pg\_settings WHERE name LIKE '%vacuum%;').
8. Принудительно запустить autovacuum для таблицы (например, выполнив много обновлений) и убедиться, что он отработал, по логам или повторному запросу статистики.
9. Подготовить отчёт с запросами мониторинга и выводами о состоянии очистки.

Контрольные вопросы:

1. В чём отличие VACUUM от VACUUM FULL?
2. Почему не рекомендуется отключать autovacuum на production-сервере?
3. Какую информацию предоставляет столбец state в pg\_stat\_activity?

## №16

Тема: Конфигурирование PostgreSQL (postgresql.conf) и настройка логирования

Цель: Изучить основные параметры конфигурационного файла postgresql.conf, влияющие на производительность и логирование, и научиться изменять их.

Теоретические вопросы:

1. Где располагается основной конфигурационный файл PostgreSQL? Какие ещё конфигурационные файлы вы знаете?
2. Какие параметры отвечают за выделение общей памяти (shared\_buffers, effective\_cache\_size)?
3. Какие параметры управляют записью в журнал предзаписи (WAL)?
4. Как настроить логирование запросов, выполняющихся дольше заданного времени?

Оборудование и ПО: ПК с ОС Red OS Murom, PostgreSQL 15+, доступ к postgresql.conf с правами sudo.

Задания:

1. Найти файл postgresql.conf (обычно /etc/postgresql/15/main/postgresql.conf) и просмотреть его содержимое.
2. Ознакомиться с текущими значениями основных параметров:

```
sql
```

```
SHOW shared_buffers;
```

```
SHOW work_mem;
```

```
SHOW max_connections;
```

3. Изменить параметр log\_min\_duration\_statement на значение 1000 (логировать запросы, выполняющиеся дольше 1 секунды).

4. Включить запись в лог всех команд DDL (создание/удаление таблиц): `log_statement = 'ddl'`.
5. Перезагрузить конфигурацию без перезапуска сервера: `SELECT pg_reload_conf();`
6. Выполнить несколько запросов, в том числе медленный (например, с `pg_sleep(2)`), и проверить, что они появились в лог-файле PostgreSQL (обычно в `/var/log/postgresql/`).
7. Изменить параметр `shared_buffers` (с осторожностью, в тестовой среде) и перезапустить сервер (`sudo systemctl restart postgresql`). Убедиться, что новое значение применилось.
8. Подготовить отчёт с фрагментами конфигурационного файла, описанием внесённых изменений и скриншотами из логов.

Контрольные вопросы:

1. Почему изменение параметра `shared_buffers` требует перезапуска сервера, а `log_min_duration_statement` – нет?
2. Для чего используется параметр `effective_cache_size`?
3. Какой параметр отвечает за формат записи времени в логах (`log_timezone`, `log_line_prefix`)?

#### **Критерии оценивания:**

- 5 баллов выставляется, если правильные ответы даны на 85-100% практических заданий
- 4 балла выставляется студенту, если правильные ответы даны на 65-84% практических заданий
- 3 балла выставляется студенту, если правильные ответы даны на 50-64% практических заданий
- 2 балла выставляется студенту, если правильные ответы даны на менее 50% практических заданий

### **3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Процедура оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций состоит из текущего контроля.

**Текущий контроль** успеваемости проводится с использованием оценочных средств, представленных в п. 2 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации и учитываются при оценивании знаний, умений, навыков и (или) опыта деятельности.

## МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

### ОП.04 Базы данных

Методические указания для студентов по освоению дисциплины «Базы данных» являются частью рабочей программы дисциплины (РПД) (приложением к рабочей программе).

РПД – рабочая программа, утвержденная директором колледжа для изучения дисциплины «Базы данных». Она определяет цели и задачи дисциплины, формируемые в ходе ее изучения компетенции и их компоненты, содержание изучаемого материала, виды занятий и объем выделяемого учебного времени, а также порядок изучения и преподавания дисциплины «Базы данных».

Для самостоятельной учебной работы студента важное значение имеют разделы «Структура и содержание дисциплины (модуля)» и «Учебно-методическое и информационное обеспечение дисциплины (модуля)». В первом указываются разделы и темы изучаемой дисциплины «Базы данных», а также виды занятий и планируемый объем (в академических часах), во втором – рекомендуемая литература и перечень ресурсов информационно-телекоммуникационной сети «Интернет».

Для подготовки к текущему контролю студенты могут воспользоваться оценочными средствами, представленными в Приложении 1 к рабочей программе дисциплины.

#### 1. Описание последовательности действий студента

Приступая к изучению дисциплины «Базы данных» необходимо в первую очередь ознакомиться с содержанием РПД, где в разделе «Структура и содержание дисциплины (модуля)» приведено общее распределение часов аудиторных занятий и самостоятельной работы по темам дисциплины «Базы данных».

Залогом успешного освоения дисциплины «Базы данных» является регулярное посещение занятий и выполнение предусмотренных программой заданий. Пропуск одного, а тем более нескольких занятий может осложнить освоение разделов курса.

Лекции имеют целью дать систематизированные основы научных знаний по содержанию дисциплины «Базы данных». При изучении и проработке теоретического материала необходимо:

– повторить законспектированный на лекционном занятии материал и дополнить его с учетом рекомендованной по данной теме литературы;

– при самостоятельном изучении теоретической темы подготовить конспект, используя рекомендованные в РПД литературные источники и электронные образовательные ресурсы.

Практические занятия проводятся с целью углубления и закрепления знаний, полученных на лекциях и в процессе самостоятельной работы с учебной литературой.

В процессе практического занятия, как вида учебных занятий, обучающиеся выполняют одно или несколько практических заданий под руководством преподавателя в соответствии с изучаемым содержанием учебного материала.

Выполнение обучающимся практических работ проводится с целью:

- систематизации и закрепления полученных теоретических знаний и практических умений;

- углубления теоретических знаний в соответствии с заданной темой;

- формирования умений применять теоретические знания при решении поставленных задач;

- развития профессиональных компетенций у обучающихся;

- развития творческой инициативы, самостоятельности, ответственности и организованности.

Выполнение обучающимися практических заданий направлено на:

- обобщение, систематизацию, углубление, закрепление полученных теоретических знаний по конкретным темам дисциплины;

- формирование умений применять полученные знания на практике, реализацию единства интеллектуальной и практической деятельности;
- выработку при решении поставленных задач таких профессионально значимых качеств, как самостоятельность, ответственность, точность, творческая инициатива.

При подготовке к практическому занятию необходимо изучить или повторить лекционный материал по соответствующей теме.

## **2. Самостоятельная работа студента**

Самостоятельная работа студента – самостоятельная учебная деятельность студента, организуемая колледжем и осуществляемая без непосредственного руководства педагога, но по его заданиям и под его контролем.

Цели самостоятельной работы:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- воспитание самостоятельности, как личностного качества будущего специалиста.

Самостоятельная работа студента по дисциплине выполняется:

- самостоятельно вне расписания учебных занятий;
- с использованием современных образовательных технологий;
- работа со специальной литературой для подготовки к тестовым, практическим заданиям.

## **3. Рекомендации по работе с литературой и источниками**

Работу с литературой следует начинать с анализа РПД, содержащей список основной и дополнительной литературы, а также знакомства с учебно-методическими разработками.

В случае возникновения затруднений в понимании учебного материала следует обратиться к другим источникам, где изложение может оказаться более доступным.

Работа с литературой не только полезна как средство более глубокого изучения дисциплины «Базы данных», но и является неотъемлемой частью профессиональной деятельности будущего выпускника.