

**Автономная некоммерческая организация профессионального образования
«ПЕРМСКИЙ ГУМАНИТАРНО-ТЕХНОЛОГИЧЕСКИЙ КОЛЛЕДЖ»
(АНО ПО «ПГТК»)**

УТВЕРЖДЕНА
Педагогическим советом АНО ПО «ПГТК»
(протокол от 05.02.2026 № 01)
Председатель Педагогического совета, директор
И.Ф. Никитина



**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
УЧЕБНОЙ ДИСЦИПЛИНЫ**

**МДК.02.02 ТЕХНОЛОГИЯ РАЗРАБОТКИ И ЗАЩИТЫ БАЗ
ДАННЫХ**

для специальности

**09.02.13 «Интеграция решений с применением технологий
искусственного интеллекта»**

(код и наименование специальности)

Квалификация выпускника

Специалист по работе с искусственным интеллектом

Форма обучения

Очная

Пермь 2026

Фонд оценочных средств учебной дисциплины МДК.02.02 ТЕХНОЛОГИЯ РАЗРАБОТКИ И ЗАЩИТЫ БАЗ ДАННЫХ составлен в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.13 «Интеграция решений с применением технологий искусственного интеллекта» (утвержден приказом Министерства Просвещения Российской Федерации от 24 декабря 2024 г. N 1025).

Программа предназначена для студентов и преподавателей АНО ПО «ПГТК».

Автор – составитель: Могильникова Н.С., старший преподаватель.

Фонд оценочных средств учебной дисциплины рассмотрена и одобрена на заседании кафедры математических и естественно-научных дисциплин, протокол, № 01 от 04.02.2026.

Содержание ФОС УД

1. Паспорт фонда оценочных средств
 - 1.1. Область применения фонда оценочных средств
 - 1.2. Организация текущего контроля успеваемости и промежуточной аттестации по итогам освоения учебной дисциплины
2. Контроль и оценка достижения запланированных результатов обучения
 - 2.1. Перечень вопросов и заданий для текущего контроля знаний
 - 2.2. Перечень вопросов и заданий для промежуточной аттестации
 - 2.3. Критерии оценивания ПА
3. Рекомендуемая литература и иные источники

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

1.1. Область применения фонда оценочных средств

Фонд оценочных средств предназначен для оценивания достижений запланированных результатов по учебной дисциплины МДК.02.02 Технология разработки и защиты баз данных программы подготовки специалистов среднего звена (далее ППССЗ) по специальности 09.02.13 «Интеграция решений с применением технологий искусственного интеллекта».

Фонд оценочных средств (ФОС) представляет собой комплект материалов для проведения промежуточной аттестации и текущего контроля.

Результаты обучения - это усвоенные знания и освоенные умения по дисциплине в целях овладения предусмотренных стандартом общих и профессиональных компетенций.

Фонд оценочных средств позволяет оценивать формирование элементов профессиональных компетенций (ПК) и элементов общих компетенций (ОК) через освоение умений, знаний и навыков.

Код ОК, ПК	Уметь	Знать
ПК 2.1 Выявлять проблемы, возникающие в процессе эксплуатации баз данных. ПК 2.2 Осуществлять процедуры администрирования баз данных. ПК 2.3 Проводить аудит систем безопасности баз данных с использованием регламентов по защите информации. ПК 2.4 Формировать требования хранения данных для обучения. ПК 2.5 Подготавливать данные для базы знаний. ОК 01 Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	Производить идентификацию проблем, связанных с нормальным функционированием базы данных; Принимать решения по локализации проблем, связанных с нормальным функционированием базы данных; Документировать внештатные ситуации связанные с нормальным функционированием базы данных; Осуществлять основные функции по администрированию баз данных; Настраивать политики безопасности при работе с сервером баз данных Дать независимую оценку уровня безопасности Производить регламентное обновление программного обеспечения Разрабатывать перечень рекомендаций по дальнейшей эксплуатации БД с максимальной защитой хранящейся информации. Производить формирование требований к обработке данных и их извлечению; Добавлять, удалять и изменять данные в базе данных; Производить операции по импорту и экспорту данных в различных форматах распознавать задачу и/или проблему в профессиональном и/или социальном контексте, анализировать и выделять её составные части	Основные коды ошибок при работе с базой данных; Методы и средства устранения ошибок, возникающих при работе с базой данных; Тенденции развития баз данных; Технология установки и настройки сервера баз данных; Требования к безопасности сервера базы данных; Протоколы безопасности при работе с базой данных; Методы и средства защиты информации от несанкционированного доступа; Уровни угроз безопасности информации Формы документов, необходимых для формирования, ведения и использования баз данных Типы данных хранения информации в базе данных актуальный профессиональный и социальный контекст, в котором приходится работать и жить структура плана для решения задач, алгоритмы выполнения работ в профессиональной и смежных областях основные источники информации и ресурсы для решения задач и/или проблем в

<p>ОК 02 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности</p> <p>ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста</p>	<p>определять этапы решения задачи, составлять план действия, реализовывать составленный план, определять необходимые ресурсы выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы владеть актуальными методами работы в профессиональной и смежных сферах</p> <p>оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)</p> <p>определять задачи для поиска информации, планировать процесс поиска, выбирать необходимые источники информации</p> <p>выделять наиболее значимое в перечне информации, структурировать получаемую информацию, оформлять результаты поиска</p> <p>оценивать практическую значимость результатов поиска</p> <p>применять средства информационных технологий для решения профессиональных задач</p> <p>использовать современное программное обеспечение в профессиональной деятельности</p> <p>использовать различные цифровые средства для решения профессиональных задач</p> <p>грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке</p> <p>проявлять толерантность в рабочем коллективе</p>	<p>профессиональном и/или социальном контексте</p> <p>методы работы в профессиональной и смежных сферах</p> <p>порядок оценки результатов решения задач профессиональной деятельности</p> <p>номенклатура информационных источников, применяемых в профессиональной деятельности</p> <p>приемы структурирования информации</p> <p>формат оформления результатов поиска информации</p> <p>современные средства и устройства информатизации, порядок их применения</p> <p>программное обеспечение в профессиональной деятельности, в том числе цифровые средства</p> <p>психологические основы деятельности коллектива</p> <p>правила оформления документов</p> <p>правила построения устных сообщений</p> <p>особенности социального и культурного контекста</p>
---	---	---

1.2. Организация текущего контроля успеваемости и промежуточной аттестации по итогам освоения программы учебной дисциплины

В период обучения по образовательной программе СПО осуществляется текущий контроль успеваемости студентов, промежуточная аттестация по учебным дисциплинам и профессиональным модулям.

Текущий контроль осуществляется в пределах учебного времени, отведенного на учебную дисциплину, оценивается по пятибалльной шкале. Текущий контроль проводится с целью объективной оценки качества освоения программы дисциплины, а также стимулирования учебной деятельности студентов, подготовки к промежуточной аттестации и обеспечения максимальной эффективности учебного процесса. Для оценки качества подготовки используются различные формы и методы контроля. Текущий контроль учебной

дисциплины осуществляется в форме устного опроса; защиты практических заданий, реферата, творческих работ; выполнения контрольных и тестовых заданий; решения ситуационных задач и других форм контроля, предусмотренных программой учебной дисциплины.

Промежуточная аттестация проводится в форме, предусмотренной планом учебного процесса: дифференцированного зачета, экзамена.

В период сложной санитарно-эпидемиологической обстановки или других ситуациях невозможности очного обучения и проведения аттестации студентов колледж реализует образовательные программы или их части с применением электронного обучения, дистанционных образовательных технологий в предусмотренных законодательством формах обучения или при их сочетании, при проведении учебных занятий, практик, текущего контроля успеваемости, промежуточной аттестации обучающихся.

Форма промежуточной аттестации по учебной дисциплине МДК.02.02 Технология разработки и защиты баз данных - дифференцированный зачет.

2. КОНТРОЛЬ И ОЦЕНКА ОСВОЕНИЯ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Перечень вопросов и заданий для текущего контроля

В результате текущей аттестации по учебной дисциплине МДК.02.02 Технология разработки и защиты баз данных осуществляется проверка сформированности умений и знаний, направленных на формирование соответствующих ФГОС СПО общих и профессиональных компетенций.

Практическое занятие Нормализация отношений и построение концептуальных моделей данных

Цель занятия: научиться правильно проектировать базы данных, устраняя избыточность и аномалии данных путем приведения таблиц к третьей нормальной форме (3NF); освоить инструменты визуализации данных на примере диаграмм Entity Relationship (ER-diagram).

Этапы занятия:

Этап 1. Проектирование структуры БД и нормализация данных

Задача: Привести таблицы данных к третьей нормальной форме (3NF).

Задание:

Вы разработаете структуру баз данных небольшой библиотеки, состоящей из трех основных сущностей: книг, авторов и читателей. Таблицы приведены ниже:

1. Книги (Books)

book_id	title	author_name
1	Книга А	Автор X
2	Книга В	Автор Y
3	Книга С	Автор Z

2. Читатели (Readers)

reader_id	first_name	last_name	phone_number
1	Иван	Иванов	+7(999)-123-45-67
2	Петр	Петров	+7(999)-765-43-21

3. Авторы (Authors)

author_id	full_name
1	Автор X
2	Автор Y
3	Автор Z

При разработке нормализованной схемы необходимо учесть возможные повторяющиеся значения и устранить дублирование данных, приведя каждую таблицу к 3-й нормальной форме.

Этап 2. Создание концептуальной модели (ERR-диаграмма)

Задача: спроектировать ER-диаграмму, отражающую отношения между сущностями библиотек («Читатель», «Книга», «Автор»).

Инструкция:

Используя MySQL Workbench, создайте три сущности (таблицы): Book, Reader, Author. Затем установите связи между ними:

- Один читатель может брать одну или несколько книг.
- Одна книга написана одним автором (упрощённая схема).
- Связь между читателями и книгами — отношение типа многие-к-одному ("many-to-one").

Нарисуйте диаграмму, включив атрибуты каждой сущности, типы полей и ограничения целостности.

Этап 3. Работа с MySQL Workbench

Задача: импортировать созданные ранее таблицы и нормализованную схему в MySQL Workbench и сгенерировать SQL-код для дальнейшего развертывания базы данных.

Шаги:

1. Запустите MySQL Workbench.
2. Создайте новую схему (database schema) с именем library_db.
3. Импортируйте созданный файл ERD (*.mwb) в среду MySQL Workbench.
4. Выполните команду генерации SQL скрипта (Export to SQL Script).
5. Проверьте правильность синтаксиса полученного SQL-скрипта.

Пример оформления отчета:

Отчет должен содержать:

- Описание предметной области (библиотека);
- Диаграмму ERD с пояснением всех взаимосвязей и ограничений;
- Скриншоты интерфейса MySQL Workbench с импортируемыми таблицами и результатами нормализации;
- Генерируемый SQL код для создания базы данных.

Контрольные вопросы:

1. Почему важна нормализация данных в реляционных моделях?
2. Чем отличается первая нормальная форма (1NF) от второй (2NF)?
3. Какое основное правило используется при переходе к третьей нормальной форме (3NF)?
4. Для чего нужны внешние ключи в реляционной модели?
5. Объясните различия между однонаправленными и двунаправленными связями в ERD-диаграммах.

Критерии оценки:

1. Правильность нормализации отношений:

- **Соответствие первой нормальной форме (1NF):** проверка уникальности строк, отсутствие составных значений внутри одной ячейки.
- **Вторая нормальная форма (2NF):** наличие первичных ключей и устранение частичной зависимости атрибутов от ключа.
- **Третья нормальная форма (3NF):** исключение транзитивных зависимостей и лишней функциональной зависимости атрибутов друг от друга.

Баллы:

- Полностью правильная реализация (все формы соблюдены) — 5 баллов.
- Незначительные нарушения (не критичные отклонения) — 3 балла.
- Грубые ошибки, влияющие на целостность и эффективность схемы — 1 балл.

2. Качество ER-диаграммы:

- Корректность отображения сущностей и атрибутов.
- Четкость обозначенных типов связей (одно-на-один, один-ко-многим, многие-ко-многим).
- Использование правильного обозначения внешнего ключа и первичного ключа.
- Логичность распределения атрибутов по таблицам.

Баллы:

- Безошибочная ER-диаграмма — 5 баллов.
- Наличие незначительных ошибок (недостаточно четкое представление связей или атрибутов) — 3 балла.
- Нарушение ключевых принципов ER-моделирования (например, неправильно заданные связи) — 1 балл.

3. Полнота отчетного материала:

- Подробное описание предметной области.
- Качественное оформление схем и диаграмм.
- Представление генерируемых SQL-запросов.
- Ясность изложения результатов нормализации и обоснование принятых решений.

Баллы:

- Отчёт соответствует всем требованиям — 5 баллов.
- Отсутствуют некоторые элементы (например, пояснения или скриншоты) — 3 балла.
- Недостаточное понимание концепции или недостаточный объём представленного материала — 1 балл.

4. Самостоятельность и оригинальность подхода:

- Насколько студент проявил самостоятельность в выборе структуры и способов представления данных.
- Умение творчески подойти к задаче, предложить нестандартные подходы к решению проблемы.

Баллы:

- Проявлена высокая степень самостоятельности и творчества — 5 баллов.
- Решение типичное, шаблонное — 3 балла.
- Полное копирование готового примера без изменений — 1 балл.

Итоговая оценка:

Итоговую оценку студента можно рассчитать по формуле:

Общая оценка=(нормализация)+(ER-диаграмма)+(отчётность)+(самостоятельность)
 Общая оценка=(нормализация)+(ER-диаграмма)+(отчётность)+(самостоятельность)

Максимальное количество баллов — 20. По итоговому количеству баллов определяется уровень усвоения

материала и выставляется отметка:

- Отлично (от 17 до 20 баллов),
- Хорошо (от 13 до 16 баллов),
- Удовлетворительно (от 9 до 12 баллов),
- Неудовлетворительно (менее 9 баллов).

«Язык реляционных баз данных SQL» с фокусом на ключевые аспекты работы с MySQL:

Задание 1. Создание представлений в MySQL

Цель: научиться создавать и использовать представления для упрощения запросов и обеспечения безопасности данных.

Теория

Представление (View) — это виртуальная таблица, основанная на результирующем наборе SQL-запроса. Оно позволяет скрыть сложность многоуровневых запросов и обеспечивает уровень абстракции для пользователей.

Практическое задание

1. Создайте две таблицы: `employees` (сотрудники) и `departments` (отделы).
2. Создайте представление `department_employees`, которое объединяет данные из обеих таблиц и отображает ФИО сотрудника и название отдела.
3. Запустите запрос к этому представлению и убедитесь, что данные отображаются корректно.
4. Создайте второе представление `salary_report`, которое агрегирует данные о зарплате сотрудников по отделам.
5. Составьте отчёт по проделанной работе.

Задание 2. Создание пользовательских функций в MySQL

Цель: научиться разрабатывать и использовать пользовательские функции для повторного использования логики в базе данных.

Теория

Пользовательские функции (User Defined Functions) позволяют инкапсулировать сложную логику и многократно использовать её в различных частях базы данных.

Практическое задание

1. Создайте базу данных и таблицу `products` с полями `price` и `quantity`.
2. Разработайте функцию `calculate_total_price(price DECIMAL(10,2), quantity INT)` для расчёта общей стоимости товара.
3. Примените эту функцию в запросе SELECT для демонстрации её работы.
4. Протестируйте функцию с различными наборами данных.
5. Подготовьте отчёт о выполненных действиях.

Задание 3. Создание хранимых процедур в MySQL

Цель: овладеть созданием и использованием хранимых процедур для автоматизации и оптимизации работы с базой данных.

Теория

Хранимые процедуры (Stored Procedures) — это наборы SQL-команд, хранящихся на сервере базы данных и выполняемых единым блоком.

Практическое задание

1. Создайте простую хранимую процедуру ``insert_product(name VARCHAR(100), price DECIMAL(10,2))``, которая добавляет новый продукт в таблицу ``products``.
2. Вызовите процедуру несколько раз с разными параметрами.
3. Создайте вторую процедуру ``update_product(id INT, new_price DECIMAL(10,2))``, которая обновляет цену продукта по его идентификатору.
4. Проверьте работоспособность созданных процедур.
5. Оформите отчёт по результатам работы.

Задание 4. Создание триггеров в MySQL

Цель: познакомиться с механизмом триггеров и научиться использовать их для автоматического реагирования на события в базе данных.

Теория

Триггеры (Triggers) — это специальные процедуры, которые выполняются автоматически при наступлении определённых событий (вставки, обновления или удаления данных).

Практическое задание

1. Создайте триггер ``audit_insert_products``, который регистрирует каждую вставку нового продукта в отдельную таблицу журнала ``product_audit_log``.
2. Создайте второй триггер ``check_quantity``, который предотвращает ввод отрицательного количества товаров.
3. Проведите тесты обоих триггеров, добавив новые продукты и попытавшись внести недопустимые значения.
4. Сделайте отчёт о проведённой работе.

Задание 5. Управление доступом к базе данных. Обеспечение сохранности данных

Цель: изучить методы управления пользователями и правами доступа, обеспечивающими защиту и целостность данных.

Теория

Правильное управление доступом критично для предотвращения несанкционированного доступа и нарушения целостности данных.

Практическое задание

1. Создайте двух пользователей: администратора и обычного пользователя.
2. Назначьте администратору полные права на всю базу данных.

3. Обычному пользователю дайте право только на просмотр данных в некоторых таблицах.
4. Проверьте, что обычные пользователи не могут вносить изменения в защищённые таблицы.
5. Продемонстрируйте восстановление данных из резервной копии (используя утилиту mysqldump).
6. Подготовьте отчёт о проделанной работе.

Общая структура оценивания

Критерий Описание

Выполнение требований Соответствие поставленным условиям задания, выполнение всех пунктов

Корректность реализации Правильность написания SQL-кода, отсутствие синтаксических и логических ошибок

Оформление отчета Четкость, понятность и полнота описания выполненной работы

Соблюдение стандартов Использование лучших практик программирования и документации

Конкретные критерии по каждому типу задания

1. Создание представлений

Критерий Баллы

Реализовано правильное объединение таблиц 2

Выбраны необходимые поля и фильтры 2

Представление работает корректно и выдает ожидаемый результат 2

Правильно составлено описание назначения представления 1

Итоговая сумма баллов 7

2. Создание пользовательских функций

Критерий Баллы

Функция принимает правильные аргументы и возвращает верный результат 3

Код написан без ошибок и соответствует требованиям задания 2

Есть адекватная документация и пояснения к функции 1

Проверено функционирование функции на тестовых данных 2

Итоговая сумма баллов 8

3. Создание хранимых процедур

Критерий Баллы

Процедура реализует заявленную функциональность 3

Использованы операторы ветвления и циклов там, где это необходимо 2

Присутствует обработка возможных исключительных ситуаций 2

Документировано назначение процедуры и порядок её вызова 1

Тестирование процедуры на реальных данных проведено успешно 2

Итоговая сумма баллов 10

4. Создание триггеров

Критерий Баллы

Триггер срабатывает на нужное событие (INSERT, UPDATE, DELETE) 2

Логика триггера реализована верно и не влияет негативно на производительность 3

Нет нежелательных побочных эффектов 2

Документация включает описание условий срабатывания и целей триггера 1

Проведено тестирование работоспособности триггера 2

Итоговая сумма баллов 10

5. Управление доступом к базе данных

Критерий Баллы

Верно созданы разные уровни пользователей (администратор, обычный пользователь) 2

Адекватно распределены привилегии пользователям 3

Безопасность системы обеспечена (нет избыточных разрешений) 2

Отчетливо описаны принципы распределения прав доступа 1

Успешно продемонстрирована работа механизма восстановления данных 2

Итоговая сумма баллов 10

Итоговая система оценивания

Общая итоговая оценка складывается из суммы баллов по всем пяти категориям. Максимально возможное количество баллов составляет 45. Оценка выставляется следующим образом:

От 40 до 45 баллов: Отлично

От 30 до 39 баллов: Хорошо

От 20 до 29 баллов: Удовлетворительно

Менее 20 баллов: Неудовлетворительно

Настройка шифрования данных в MySQL с использованием встроенных функций (например, AES_ENCRYPT, AES_DECRYPT).

1. Создание таблицы Создаем таблицу, в которой будем хранить зашифрованные данные:

sql

Копировать

```
CREATE TABLE users (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    username VARCHAR(50),  
    password VARBINARY(255)  
);
```

Здесь мы используем тип данных `VARBINARY`, поскольку зашифрованные данные являются бинарными.

2. Шифрование данных при вставке Мы можем зашифровать пароль перед сохранением его в базу данных с помощью функции `AES_ENCRYPT`. Допустим, у нас есть простой скрипт для добавления пользователей:

sql

Копировать

```
INSERT INTO users (username, password)
VALUES ('ivan', AES_ENCRYPT('mypassword', 'my_secret_key'));
```

Где:

- "mypassword" — исходный пароль пользователя,
 - "my_secret_key" — ваш секретный ключ для шифрования.
3. **Дешифровка данных при чтении** Когда нам нужно прочитать данные обратно, мы используем функцию `AES_DECRYPT` для расшифровки пароля:

sql

Копировать

```
SELECT username, AES_DECRYPT(password, 'my_secret_key') AS decrypted_password FROM users
WHERE username = 'ivan';
```

4. **Безопасность ключа** Важно отметить, что хранение секретного ключа в открытом виде небезопасно. Лучше всего использовать отдельный механизм управления ключами, либо передавать ключ динамически через приложение, а не жестко закодированный в SQL-запросах.

Полезные советы:

- Всегда проверяйте длину вашего ключа (`AES_ENCRYPT` поддерживает ключи длиной 128, 192 или 256 бит).
- Используйте разные подходы к хранению ключей, чтобы минимизировать риск компрометации данных.
- Помните, что хотя шифрование помогает защитить конфиденциальность данных, оно также увеличивает нагрузку на сервер базы данных, особенно если таблица большая и часто запрашиваемая.

Реализация ролевой модели безопасности в PostgreSQL

1. **Создать роли:** создайте две роли с разными уровнями привилегий: администратора и обычного пользователя.
2. **Предоставить права:** назначьте каждому типу пользователя соответствующие права доступа.
3. **Проверить работу системы:** продемонстрируйте работу механизма через выполнение запросов пользователями разных ролей.

1. Создание ролей:

Создадим две роли: роль администратора (`admin_role`) и роль простого пользователя (`user_role`):

sql

Копировать

```
-- Создаем роль администратора
CREATE ROLE admin_role LOGIN PASSWORD 'admin_pass' SUPERUSER;
```

```
-- Создаем роль обычного пользователя
CREATE ROLE user_role LOGIN PASSWORD 'user_pass';
```

Примечание: Роль администратор получает расширенные права (`SUPERUSER`), а обычный пользователь имеет ограниченный доступ.

2. Предоставление прав:

Определим объекты базы данных, которым будут предоставлены права доступа. Например, создадим тестовую схему и таблицу:

sql

Копировать

```
-- Создаем новую схему
CREATE SCHEMA test_schema AUTHORIZATION admin_role;
```

```
-- Создаем таблицу внутри схемы
CREATE TABLE test_schema.users (
    id SERIAL PRIMARY KEY,
    username TEXT NOT NULL UNIQUE,
    email TEXT NOT NULL UNIQUE
);
```

Теперь назначим необходимые права пользователям:

sql

Копировать

```
-- Назначаем полные права на всю схему администратору
GRANT ALL PRIVILEGES ON SCHEMA test_schema TO admin_role;
```

```
-- Назначаем право SELECT обычному пользователю
GRANT USAGE ON SCHEMA test_schema TO user_role;
GRANT SELECT ON test_schema.users TO user_role;
```

Таким образом, администратор сможет создавать, изменять и удалять любые объекты, а обычный пользователь получит доступ только для чтения данных из таблицы.

3. Проверка работы системы:

Проверьте назначение прав, выполнив запросы от имени каждой роли:

Подключитесь от имени обычной роли:

bash

Копировать

```
psql -U user_role -d your_database_name
```

Попытайтесь добавить запись в таблицу:

sql

Копировать

```
INSERT INTO test_schema.users(username, email) VALUES ('test_user', 'test@example.com');
```

Результат: Ошибка, так как обычная роль не имеет достаточных прав.

Выполните запрос на чтение:

sql

Копировать

```
SELECT * FROM test_schema.users;
```

Результат: Запрос выполнится успешно, отображаются записи.

Теперь подключимся от имени админа:

bash

Копировать

```
psql -U admin_role -d your_database_name
```

Попробуйте добавить запись в таблицу:

sql

Копировать

```
INSERT INTO test_schema.users(username, email) VALUES ('admin_user',
'admin@example.com');
```

Результат: Данные успешно добавлены.

Дополнительные рекомендации:

- Используйте принцип наименьших привилегий, назначайте ровно столько прав, сколько реально необходимо.
- Регулярно проверяйте назначенные права, чтобы избежать избыточных разрешений.
- Периодически обновляйте настройки безопасности, используя команду `REVOKE`, если роль больше не должна иметь определенных полномочий.

Настройка аудита действий пользователей в Microsoft SQL Server

Задача: настроить аудит всех операций (SELECT, INSERT, UPDATE, DELETE) для определённых пользователей и объектов БД.

Этапы решения:

1. Подключение SQL Audit Log:

```
sql
Копировать
USE master;
GO
CREATE SERVER AUDIT audit_users_actions
    TO FILE (FILEPATH='C:\Audits\')
    WITH (QUEUE_DELAY = 1000, ON_FAILURE=CONTINUE);
ALTER SERVER AUDIT audit_users_actions STATE=ON;
```

2. Создание спецификации аудита для конкретной базы данных:

```
sql
Копировать
USE YourDatabaseName;
GO
CREATE DATABASE AUDIT SPECIFICATION audit_db_specification
    FOR SERVER AUDIT audit_users_actions
    ADD (SELECT, INSERT, UPDATE, DELETE ON dbo.YourTable BY public)
    WITH (STATE=ON);
```

Здесь замените `YourDatabaseName`, `dbo.YourTable` и `public` соответственно вашей базой данных, таблицей и пользователями, подлежащими аудиту.

3. Просмотр журнала аудита: Аудитированные события можно просмотреть следующим запросом:

```
sql
Копировать
SELECT * FROM sys.fn_get_audit_file('C:\Audits\*', default,
default);
```

Конфигурация шифрования трафика между клиентом и сервером базы данных (TLS/SSL)

Задача: обеспечить защищённый обмен данными между клиентскими приложениями и MS SQL Server с использованием TLS/SSL-шифрования.

Этапы решения:

- 1. Генерация сертификата SSL/TLS:** Можно воспользоваться инструментом OpenSSL для генерации самоподписанного сертификата или же купить сертификат у доверенного центра сертификации. Пример команды для генерации сертификата:

```
shell
Копировать
openssl req -newkey rsa:2048 -nodes -keyout server.key -x509 -days 365 -out server.cert
```
- 2. Установка сертификата на сервере:** Сертификат необходимо импортировать в хранилище сертификатов Windows. Затем на сервере SQL конфигурируем использование сертификата:
 - Открываем **SQL Server Configuration Manager**, переходим в пункт **SQL Server Network Configuration** → **Protocols for MSSQLSERVER** → **Properties** → **Certificate tab**.
 - Выбираем ранее установленный сертификат.
- 3. Включение принудительного шифрования соединений:** После установки сертификата включаем обязательное шифрование соединения в конфигурациях сети сервера:
 - Опять открываем **SQL Server Configuration Manager**, выбираем вкладку **Protocols for MSSQLSERVER** → **Properties** → **Flags tab**.
 - Включаем флажок **Force Encryption** и перезагружаем службу SQL Server

Организация резервного копирования с шифрованием в Oracle Database

Задача: создать полное резервное копирование всей базы данных с применением шифрования.

Этапы решения:

1. **Подготовка среды:** Убедитесь, что ваша база данных находится в режиме ARCHIVELOG, иначе нельзя создать полную копию.

```
sql
Копировать
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER DATABASE ARCHIVELOG;
ALTER DATABASE OPEN;
```

2. **Настройка мастер-ключа шифрования:** Прежде чем начать процесс резервного копирования с шифрованием, надо установить мастер-пароль шифрования (TDE Master Key):

```
sql
Копировать
ADMINISTER KEY MANAGEMENT SET ENCRYPTION WALLET IDENTIFIED BY
"<password>";
```

3. **Резервное копирование с шифрованием:** Используем утилиту RMAN для запуска полного резервного копирования:

```
bash
Копировать
rman target /
BACKUP DATABASE PLUS ARCHIVELOG FORMAT '/backup/%U.bkup' TAG
'FullBackup' ENCRYPTION ALGORITHM 'AES256';
Резервные копии сохраняются в указанном каталоге /backup/.
```

Разработка политики управления доступом к данным на уровне таблиц и столбцов

Задача: ограничить доступ к отдельным столбцам и таблицам, внедрив политику Row Level Security (RLS) и Column Level Security (CLS).

Этапы решения:

1. **Определение ролей и условий ограничения доступа:** Определите условия фильтрации строк и ограничение доступа к определенным столбцам для конкретных ролей.
2. **Реализация RLS (Row Level Security):** Пример правила ограничения доступа на уровне строки:

```
sql
Копировать
CREATE SECURITY POLICY row_level_policy
  ADD FILTER PREDICATE dbo.is_data_owner(row_owner_id) ON
  dbo.your_table
  WITH (STATE = ON);
```

Функция `is_data_owner` определяет, соответствует ли текущему пользователю условие владения записью.

3. **Реализация CLS (Column Level Security):** Ограничение доступа к отдельным столбцам реализуется с помощью масок доступа (Dynamic Data Masking):

```
sql
Копировать
ALTER TABLE employees ALTER COLUMN salary ADD MASKED WITH (FUNCTION
= 'default()');
```

Таким образом, столбец `salary` становится доступным только для разрешённых ролей.

Настройка защиты конфиденциальных данных с использованием маскирования данных (Data Masking) в Microsoft SQL Server

Задача: скрыть чувствительные данные в таблице сотрудников путём замещения реальных значений фиктивными значениями (маскировка).

Этапы решения:

1. **Выбор стратегии маскировки:** Выберите подходящую стратегию маскировки для конкретного типа данных (например, полная замена, случайное значение, регулярное выражение и др.).
2. **Применение маски на уровне столбца:** Применяем маску на столбце, содержащем персональные данные:

sql

Копировать

```
ALTER TABLE Employees ALTER COLUMN Email ADD MASKED WITH (FUNCTION = 'email()');  
ALTER TABLE Employees ALTER COLUMN Salary ADD MASKED WITH (FUNCTION = 'partial("****", "", "")');
```

В результате столбцы `Email` и `Salary` становятся замаскированными для большинства пользователей.

Организация двухфакторной аутентификации для доступа к базам данных

Задача: включить двухфакторную аутентификацию для повышения уровня безопасности входа в базу данных.

Этапы решения:

1. **Использование Azure Active Directory (AAD) для SQL Server:** Если ваша инфраструктура интегрирована с облаком Azure, примените Multi-Factor Authentication (MFA) для учетных записей, использующих Azure AD. Инструкция включает:
 - Добавление аккаунтов в Azure AD.
 - Использование подключения SSPI (Security Support Provider Interface) в SQL Server Management Studio (SSMS).
2. **Настройка двухэтапной проверки через VPN или прокси-сервер:** Пользователи сначала проходят стандартную проверку пароля, а затем дополнительно вводят одноразовый код, полученный через SMS или мобильное приложение.

Для допуска к промежуточной аттестации нужно выполнить все практические задания.

2.2. Перечень вопросов и заданий для промежуточной аттестации

Форма: дифференцированный зачет

Теоретические вопросы:

1. Основные понятия и концепции баз данных (СУБД, реляционная модель, нормальные формы).
2. Структура реляционной базы данных: таблицы, поля, индексы, связи.
3. Понятие транзакций и уровни изоляции транзакций.
4. Отличия OLTP и OLAP моделей обработки данных.
5. Нормализация данных и её цели.
6. Принцип ACID-транзакций (Atomicity, Consistency, Isolation, Durability).
7. Методы оптимизации производительности запросов.
8. Основы проектирования реляционных баз данных (Entity Relationship Diagram, ER-модель).
9. Типы индексов и их влияние на производительность запросов.
10. Механизмы репликации и шардинга в распределённых системах хранения данных.

11. Архитектуры NoSQL баз данных и их преимущества.
12. Безопасность баз данных: методы аутентификации и авторизации.
13. Модели доступа к данным (RBAC, ABAC, DAC).
14. Обзор современных технологий и инструментов для анализа больших данных (Big Data).
15. Современные тенденции развития облачных решений для хранения и обработки данных.
16. Применение виртуализации и контейнеризации в управлении базами данных.
17. Технологии шифрования данных на стороне клиента и сервера.
18. Особенности использования Materialized Views (материализованных представлений).
19. Репозиторий метаданных и управление жизненным циклом данных.
20. Кэширование результатов запросов и способы снижения нагрузки на БД.

Практические задания:

1. Проектирование структуры базы данных. Разработайте структуру реляционной базы данных для онлайн-магазина товаров с учётом нормализации данных до третьей нормальной формы.
2. Оптимизация запросов. Проанализируйте и оптимизируйте заданный SQL-запрос с целью уменьшения времени исполнения и потребления ресурсов.
3. Запросы с JOIN и агрегированными функциями. Напишите SQL-запрос, выполняющий выборку данных из нескольких таблиц с группировкой и агрегацией.
4. Работа с индексами. Предложите стратегию построения индексов для улучшения производительности запросов в большой таблице заказов интернет-магазина.
5. Архивация и восстановление данных. Продемонстрируйте процедуру архивации и восстановления базы данных средствами выбранной СУБД (например, PostgreSQL, MySQL, Oracle).
6. Импорт и экспорт данных. Подготовьте инструкцию по импорту и экспорту данных из одной базы данных в другую с использованием утилит (pg_dump, mysqldump и т.п.).
7. Политики доступа к данным. Разработайте политику разграничения доступа к ресурсам базы данных на уровне таблиц и полей с использованием механизмов RBAC.
8. Защита конфиденциальных данных. Предложите меры по защите персональных данных клиентов, включая шифрование, хеширование и механизмы маскирования данных.
9. Триггеры и хранимые процедуры. Разработайте триггеры и хранимую процедуру для автоматического контроля целостности данных и соблюдения бизнес-правил в базе данных.
10. Настройка мониторинга производительности. Настройте мониторинг ключевых показателей производительности базы данных (количество активных сессий, загрузка ЦПУ, потребление памяти и т.д.) с помощью встроенных средств СУБД.
11. Автоматизированное тестирование баз данных. Напишите автоматизированные тесты для проверки корректности функционирования прикладных процедур и триггеров в базе данных.
12. Управление транзакциями. Демонстрируйте правильную обработку ошибок и откаты транзакций в сложной среде многопользовательского доступа.
13. Создание резервных копий с шифрованием. Настройте создание резервных копий базы данных с обязательным шифрованием резервных файлов.
14. Аудит активности пользователей. Организуйте сбор и анализ информации о действиях пользователей в базе данных (просмотры, изменения, удаление данных).
15. Анализ и настройка конфигурации сервера базы данных. Проведите диагностику производительности базы данных и предложите меры по улучшению настроек сервера (конфигурационные файлы, кэширование и прочие средства ускорения работы).

Критерии оценивания устного ответа на экзамене

Оценка «5» («отлично») соответствует следующей качественной характеристике: «изложено правильное понимание вопроса и дан исчерпывающий на него ответ, содержание раскрыто полно, профессионально, грамотно».

Выставляется студенту,

- усвоившему взаимосвязь основных понятий дисциплины в их значении для приобретаемой профессии, проявившему творческие способности в понимании, изложении и использовании учебно-программного материала;
- обнаружившему всестороннее систематическое знание учебно-программного материала, четко и самостоятельно (без наводящих вопросов) отвечающему на вопрос билета.

Оценка «4» («хорошо») соответствует следующей качественной характеристике: «изложено правильное понимание вопроса, дано достаточно подробное описание предмета ответа, приведены и раскрыты в тезисной форме основные понятия, относящиеся к предмету ответа, ошибочных положений нет».

Выставляется студенту,

- обнаружившему полное знание учебно-программного материала, грамотно и по существу отвечающему на вопрос билета и не допускающему при этом существенных неточностей;
- показавшему систематический характер знаний по дисциплине и способному к их самостоятельному пополнению и обновлению в ходе дальнейшей учебы и профессиональной деятельности.

Оценка «3» («удовлетворительно»)

Выставляется студенту,

- обнаружившему знание основного учебно-программного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, справляющемуся с выполнением заданий, предусмотренных программой;
- допустившему неточности в ответе и при выполнении экзаменационных заданий, но обладающими необходимыми знаниями для их устранения под руководством преподавателя.

Оценка «2» («неудовлетворительно»)

Выставляется студенту,

- обнаружившему существенные пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий;
- давшему ответ, который не соответствует вопросу экзаменационного билета.

3. Рекомендуемая литература и иные источники

Основные источники:

1. Астапчук, В. А. Базы данных: проектирование и реализация : учебное пособие / В. А. Астапчук, Е. Н. Павенко, И. В. Эстрайх. — Новосибирск : Новосибирский государственный технический университет, 2023. — 111 с. — ISBN 978-5-7782-4917-2. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/155559.html>

Дополнительная литература:

1. Петрова, А. Н. Реализация баз данных : учебное пособие / А. Н. Петрова, В. Е. Степаненко. — Москва : Ай Пи Ар Медиа, 2022. — 143 с. — ISBN 978-5-4497-1026-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/105714.html> (дата обращения: 13.01.2026). — Режим доступа: для авторизир. пользователей. - DOI: <https://doi.org/10.23682/105714>

2. Введение в СУБД MySQL : учебное пособие / . — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2022. — 228 с. — ISBN 978-5-4497-0912-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102004.html> — Режим доступа: для авторизир. пользователей

3. Алексеев, В. А. Основы проектирования и реализации баз данных : методические указания к проведению лабораторных работ по курсу «Базы данных» / В. А. Алексеев. — Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2014. — 26 с. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/55122.html>

4. Оптимизация работы серверов баз данных Microsoft SQL Server 2005 : учебное пособие / . — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 372 с. — ISBN 978-5-4497-0901-1. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/102024.html> — Режим доступа: для авторизир. пользователей

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п.п.	Содержание изменения	Дата, номер протокола заседания кафедры, подпись зав.кафедрой
1	2	3
1		
2		
3		
4		