**Министерство образования и науки Забайкальского края**

**Государственное профессиональное образовательное учреждение**

 **«ЧИТИНСКИЙ ТЕХНИКУМ ОТРАСЛЕВЫХ ТЕХНОЛОГИЙ И БИЗНЕСА»**

**(ГПОУ «ЧТОТиБ»)**

**В.А. Чурсин**

**ОП.04 «Основы проектирования баз данных»**

Методические рекомендации для выполнения практических и самостоятельных

работ для студентов очного отделения

специальность **09.02.09 «Веб-разработка»**

Издательство ГПОУ «ЧТОТиБ»

2025 г.

**1. Пояснительная записка**

Учебная дисциплина ОП.04 «Основы проектирования баз данных» входит в общепрофессиональный цикл основной профессиональной образовательной программы по специальности 09.02.09 «Веб-разработка».

В рамках дисциплины предусмотрено выполнение практических и самостоятельных работ, способствующих закреплению теоретических знаний и приобретению практических навыков. Практические работы выполняются в аудитории под руководством преподавателя.

Методические рекомендации по выполнению практических работ составлены в соответствии с Государственными требованиями к минимуму содержания и уровню подготовки студентов по программам подготовки специалистов среднего звена.

Практические занятия стимулируют познавательную деятельность студентов, поскольку требуют их активного участия в проведении различных исследований. Они направлены на углубление и закрепление теоретических знаний в области расчета и анализа, а также на приобретение практических навыков для решения различных ситуационных задач, которые могут быть полезны в будущей профессиональной деятельности. Студенты учатся работать с разнообразной литературой, методическими и справочными материалами.

Перед выполнением практической работы обучающийся должен повторить или изучить материал, относящийся к теме работы, лекционным записям, учебной литературы и соответствующим методическим инструкциям.

Практические занятия служат связующим звеном между теорией и практикой. Они необходимы для закрепления теоретических знаний, полученных на уроках теоретического обучения, а также для получения практических знаний. Практические задания выполняются обучающимся самостоятельно, с применением знаний и умений, полученных на уроках, а также с использованием необходимых пояснений, полученных от преподавателя при выполнении практического задания. К практическому занятию от обучающегося требуется предварительная подготовка, которую он должен провести перед занятием.

Практические задания разработаны в соответствии с учебной программой. В зависимости от содержания они могут выполняться обучающимися индивидуально или фронтально.

Цель методических указаний – предоставить студентам рекомендации по выполнению практических и самостоятельных работ, а также обеспечить их необходимыми материалами для успешного освоения дисциплины.

В результате изучения дисциплины и выполнения работ студент должен:

**Уметь:**

 **У1.** Интерпретировать бизнес-требования заказчика для разработки концептуальной модели информационного ресурса.

 **У2.** Устанавливать систему управления базами данных (СУБД).

 **У3.** Использовать средства системы управления базами данных.

 **У4.** Выполнять регламентные процедуры по резервированию данных.

 **У5.** Применять регламентные процедуры управления правами доступа пользователей информационных ресурсов.

**Знать:**

 **З1**. Основы построения концептуальных моделей информационных ресурсов средствами графических нотаций.

 **З2.** Программные средства и платформы для разработки web-ресурсов.

 **З3.** Особенности систем управления базами данных.

 **З4.** Общие основы решения практических задач по созданию резервных копий.

 **З5.** Основы резервного развертывания и резервного копирования информационных ресурсов.

**2. Тематический план**

|  |  |
| --- | --- |
| **Практические работы** | **Количество часов** |
| **Раздел 1. Основные понятия теории проектирования баз данных** | **4** |
| **Тема 2.3. Реляционная алгебра** |
| **Практическая работа №1** «Использование операций: Объединение, Пресечение, Вычитание, Декартово произведение, Выборка, Проекция, Соединение, Деление» | 2 |
| **Самостоятельная работа №1** «Решение задач по реляционной алгебре» | 2 |
| **Раздел 3. Этапы проектирования баз данных** | **8** |
| **Тема 3.2. Концептуальное проектирование БД** | **2** |
| **Практическая работа №2** «Проектирование ER-диаграммы базы данных по нотации Чена» |
| **Тема 3.3. Логическое проектирование и нормализация БД** | **2** |
| **Практическая работа №3** «Нормализация таблиц реляционной БД» |
| **Тема 3.4. Физическое проектирование БД** | **4** |
| **Практическая работа №4** «Освоение принципов проектирования БД» | 2 |
| **Практическая работа №5** «Проектирование реляционной БД» | 2 |
| **Раздел 4. Подходы к реализации реляционных баз данных. Язык запросов SQL** | **20** |
| **Тема 4.1 Подходы к реализации реляционных баз данных. Язык запросов SQL** |
| **Практическая работа №6** «Написание простейших SELECT-запросов» | 2 |
| **Практическая работа №7** «Написание SELECT-запросов с логическими операторами, DISTINCT, условиями WHERE, операторами IS NULL, BETWEEN, IN и LIKE» | 2 |
| **Практическая работа №8** «Написание SELECT-запросов с сортировками, группировками и агрегатными функциями» | 2 |
| **Практическая работа №9** «Написание многотабличных запросов, запросов с ограничением выборки и подзапросов» | 2 |
| **Практическая работа №10** «Написание запросов с обобщенным табличным выражением, объединением запросов и условной логикой» | 2 |
| **Практическая работа №11** «Написание запросов на создание и удаление таблиц и БД, запросов на добавление, удаление и обновление данных» | 2 |
| **Практическая работа №12** «Представления» | 2 |
| **Практическая работа №13** «Хранимые процедуры и триггеры» | 2 |
| **Практическая работа №14** «Управление доступом к данным» | 2 |
| **Практическая работа №15** «Резервное копирование и восстановление данных» | 2 |
| **Раздел 5. Работа в СУБД** | **16** |
| **Тема 5.1**. **Работа в СУБД** |
| **Практическая работа №16 «**Работа с реляционной БД в СУБД MySQL Workbench» | 4 |
| **Практическая работа №17 «**Работа с реляционной БД в СУБД PostgreSQL» | 4 |
| **Практическая работа №18 «**Работа с реляционной БД в PhpMyAdmin» | 4 |
| **Практическая работа №19** «Проектирование реляционной БД на заданную тему в представленных СУБД» | 2 |
| **Самостоятельная работа №2** «Защита реляционных БД на заданную тему» | 2 |

**3. Методические рекомендации к практическим и самостоятельным работам**

**Практическая работа №1 «Использование операций: Объединение, Пресечение, Вычитание, Декартово произведение, Выборка, Проекция, Соединение, Деление»**

**Цель:** научиться применять основные операции реляционной алгебры к множествам данных для извлечения нужной информации.

**Теоретическая основа:**

Реляционная алгебра – это формальный язык запросов к реляционным БД, основанный на множественных операциях. В данной работе рассматриваются такие операции как:

* ∪ (Объединение).
* ∩ (Пересечение).
* − (Разность/Вычитание).
* × (Декартово произведение).
* σ (Селекция/Выборка).
* π (Проекция).
* ⨝ (Соединение).
* ÷ (Деление).

**Задание и исходные данные для выполнения работы:**

1. Сущность студентов (Students)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **StudentID** | **FullName** | **GroupID** | **Year** | **Year** |
| S1 | Алиса Павлова | G1 | 2023 | alisa@mail.ru |
| S2 | Борис Егоров | G2 | 2022 | boris@ya.ru |
| S3 | Виктория Громова | G1 | 2023 | vika@uni.edu |
| S4 | Григорий Захаров | G3 | 2024 | greg@school.ru |
| S5 | Диана Соколова | G2 | 2023 | diana@edu.ru |
| S6 | Алексей Миронов | G1 | 2023 | lex@site.ru |

1. Сущность групп (Groups)

|  |  |  |
| --- | --- | --- |
| **GroupID** | **GroupName** | **Curator** |
| G1 | Веб-разработка | Кузнецов А.П. |
| G2 | Системное ПО | Иванова И.А. |
| G3 | Информационная безопасность | Смирнов О.Н. |

1. Сущность курсов (Courses)

|  |  |  |  |
| --- | --- | --- | --- |
| **CourseID** | **CourseName** | **Hours** | **Type** |
| C1 | HTML/CSS | 36 | Практика |
| C2 | Алгоритмы | 72 | Теория |
| C3 | Веб-технологии | 54 | Практика |
| C4 | Базы данных | 60 | Теория |
| C5 | Операционные системы | 48 | Теория |

1. Сущность оценок (Grades)

|  |  |  |
| --- | --- | --- |
| **StudentID** | **CourseID** | **Grade** |
| S1 | C1 | 5 |
| S1 | C2 | 4 |
| S1 | C4 | 5 |
| S2 | C2 | 5 |
| S2 | C4 | 4 |
| S3 | C1 | 5 |
| S3 | C3 | 5 |
| S3 | C4 | 5 |
| S4 | C5 | 4 |
| S5 | C2 | 3 |
| S5 | C3 | 4 |
| S6 | C1 | 5 |
| S6 | C2 | 4 |
| S6 | C4 | 4 |

1. Сущность преподавателей (Teachers)

|  |  |  |  |
| --- | --- | --- | --- |
| **TeacherID** | **Name** | **Email** | **CourseID** |
| T1 | Кузнецов А.П. | kuz@col.edu | C1 |
| T2 | Иванова И.А. | iva@col.edu | C2 |
| T3 | Смирнов О.Н. | smir@col.edu | C3 |
| T4 | Захарова Л.В. | zah@col.edu | C4 |
| T5 | Митин В.А. | mitin@col.edu | C5 |

1. Сущность аудиторий (Classrooms)

|  |  |  |
| --- | --- | --- |
| **RoomID** | **RoomType** | **Capacity** |
| R101 | Лекция | 50 |
| R203 | Лаборатория | 25 |
| R305 | Комп. класс | 20 |

1. Сущность расписания (Schedule)

|  |  |  |  |
| --- | --- | --- | --- |
| **CourseID** | **RoomID** | **Day** | **TimeSlot** |
| C1 | R305 | Понедельник | 10:00 |
| C2 | R101 | Среда | 12:00 |
| C3 | R203 | Четверг | 14:00 |
| C4 | R101 | Вторник | 09:00 |
| C5 | R203 | Пятница | 13:00 |

1. Сущность проектов (Projects)

|  |  |  |  |
| --- | --- | --- | --- |
| **ProjectID** | **StudentID** | **Title** | **Grade** |
| P1 | S1 | Онлайн-магазин | 5 |
| P2 | S2 | Блог-платформа | 5 |
| P3 | S3 | Парсер новостей | 4 |
| P4 | S4 | SQL-тренажёр | 4 |
| P5 | S5 | Панель администратора | 5 |

**Задания:**

#### ****1. Объединение:**** Выведите всех студентов, которые сдали курс C2 или C4.

#### ****2. Пересечение:**** Выведите студентов, сдавших оба курса: C1 и C4.

#### ****3. Разность:**** Найдите студентов, которые сдали C2, но не C4.

#### ****4. Декартово произведение:**** Постройте все возможные комбинации студентов и преподавателей.

#### ****5. Проекция:**** Выведите уникальные названия курсов и тип занятия.

#### ****6. Селекция:**** Выведите всех студентов группы G1, обучающихся в 2023 году.

#### ****7. Соединение:**** Выведите ФИО студента, курс и оценку, используя соединения

#### ****8. Деление:**** Найдите студентов, которые сдали ****все теоретические курсы****.

#### ****9. Выборка + соединение:**** Выведите студентов, которые сдали хотя бы один предмет у преподавателя Ивановой И.А.

#### ****10. Сложная выборка:**** Выведите студентов, участвующих в проектах и сдавших не менее 3 курсов.

### **Контрольные вопросы:**

1. В чем разница между операцией объединения и пересечения?
2. Почему операция деления считается одной из самых сложных?
3. Как с помощью проекции убрать дубликаты в SQL?
4. Что произойдёт при соединении таблиц, если ключей нет?
5. Приведите пример задачи, где нужно использовать вычитание.

**Самостоятельная работа №1 «Решение задач по реляционной алгебре»**

## Цель работы: закрепить навыки применения основных операций реляционной алгебры для работы с отношениями, научиться формулировать решения с использованием операций выборки, проекции, объединения, пересечения, разности, декартова произведения и соединения.

## Задание:

1. Изучить основные операции реляционной алгебры.
2. Для заданных отношений (таблиц) выполнить задачи:
	* Операция **выборки** (σ) по заданному условию.
	* Операция **проекции** (π) для указанных атрибутов.
	* Операция **объединения** (∪) двух отношений.
	* Операция **пересечения** (∩) двух отношений.
	* Операция **разности** (−) двух отношений.
	* Операция **декартова произведения** (×) двух отношений.
	* Операция **естественного соединения** (⨝) двух отношений.
3. Оформить решения в виде последовательностей применённых операций, указывая исходные и полученные отношения.

Базовые отношения (таблицы):

**Студенты**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID\_Студента** | **ФИО** | **Группа** | **ГодПоступления** | **Возраст** |
| 1 | Иванов И.И. | ИС-21 | 2021 | 18 |
| 2 | Петров П.П. | ИС-22 | 2022 | 19 |
| 3 | Сидоров С.С. | ИС-21 | 2021 | 18 |
| 4 | Смирнова А.А. | ИС-23 | 2023 | 17 |
| 5 | Кузнецова Е.В. | ИС-22 | 2022 | 20 |

**Курсы**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID\_Курса** | **Название** | **Семестр** | **Часы** |
| 1 | Базы данных | 3 | 72 |
| 2 | Программирование | 2 | 90 |
| 3 | Компьютерные сети | 4 | 68 |
| 4 | Веб-технологии | 3 | 64 |

**Успеваемость**

|  |  |  |
| --- | --- | --- |
| **ID\_Студента** | **ID\_Курса** | **Оценка** |
| 1 | 1 | 5 |
| 1 | 2 | 4 |
| 2 | 1 | 3 |
| 2 | 3 | 4 |
| 3 | 1 | 5 |
| 4 | 2 | 5 |
| 5 | 3 | 4 |
| 5 | 4 | 5 |

## Задания (операции реляционной алгебры):

1. Выполнить **выборку** студентов группы ИС-21.
2. Выполнить **проекцию**: вывести только ФИО и Группу всех студентов.
3. Найти **объединение** студентов из группы ИС-21 и ИС-22.
4. Найти **пересечение** студентов, у которых ГодПоступления = 2021 и Группа = ИС-22.
5. Выполнить **разность** между всеми студентами и студентами, у которых возраст 18 лет.
6. Выполнить **декартово произведение** таблиц Студенты и Курсы (поясните результат).
7. Выполнить **естественное соединение** таблиц Студенты и Успеваемость по ID\_Студента.
8. Найти студентов, у которых **оценка** по курсу «Базы данных» выше 4.
9. Вывести студентов, изучающих **более одного курса** (по ID\_Студента).
10. Найти курсы, которые проходят в 3-м семестре и имеют более 60 часов.
11. Выбрать студентов, чей возраст находится **между 17 и 19 годами**.
12. Найти студентов, которые **не изучают** курс «Компьютерные сети».
13. Найти ФИО студентов, чья фамилия заканчивается на «ов» (**LIKE**-поиск).
14. Вывести курсы, которые проходят либо во 2-м, либо в 4-м семестре.
15. Определить студентов, которые получили оценку «5» по любому курсу.
16. Вывести имена студентов и названия курсов, которые они изучают.
17. Найти всех студентов, у которых нет оценок в таблице Успеваемость.
18. Вывести количество студентов в каждой группе (**GROUP BY**).
19. Вывести среднюю оценку по каждому курсу (**GROUP BY).**
20. Найти студентов, поступивших после 2021 года.
21. Определить студентов старше 18 лет, обучающихся на 3-м семестре.
22. Найти курсы, на которых оценка у студентов только 4 или 5.
23. Найти студентов, не обучающихся ни на одном курсе (использовать разность).
24. Определить студентов, которые учатся на одном и том же курсе.
25. Вывести студентов с минимальной оценкой по каждому курсу.

## Пояснение:

* Операция **Выборка** (σ) используется для фильтрации записей по условию.
* Операция **Проекция** (π) используется для выбора отдельных столбцов.
* Операция **Объединение** (∪) объединяет результаты двух таблиц.
* Операция **Пересечение** (∩) находит общие элементы двух таблиц.
* Операция **Разность** (−) исключает записи одной таблицы из другой.
* **Декартово произведение** (×) создает все возможные пары строк двух таблиц.
* **Естественное соединение** (⨝) объединяет строки с одинаковыми значениями по выбранному столбцу.

## Контрольные вопросы:

1. Что происходит при декартовом произведении двух отношений?
2. Когда имеет смысл использовать операцию пересечения?
3. Чем отличается выборка от проекции?
4. Как выполнить естественное соединение двух таблиц?
5. Для чего нужна операция разности в реляционной алгебре?

**Практическая работа №2 «Проектирование ER-диаграммы базы данных по нотации Чена»**

**Цель работы:** научиться проектировать и строить ER-диаграмму базы данных, используя нотацию Чена для описания сущностей, связей и атрибутов.

**Задание:**

1. **Ознакомьтесь с основными принципами нотации Чена**
* Сущности отображаются прямоугольниками.
* Атрибуты – эллипсами.
* Связи между сущностями – ромбами.



* Связи могут быть одного из типов: один-ко-многим, многие-ко-многим, один-к-одному.



1. **Спроектируйте базу данных для предметной области: Интернет-магазин по продаже книг.**

База данных должна содержать следующую информацию:

* + Сведения о книгах (название, автор, ISBN, год издания, жанр, цена).
	+ Сведения об авторах (ФИО, дата рождения, страна).
	+ Сведения о клиентах (ФИО, email, телефон, адрес).
	+ Сведения о заказах (номер заказа, дата заказа, статус заказа).
	+ Сведения о содержимом заказа (какие книги входят в каждый заказ, количество каждой книги).
1. **Требования к ER-диаграмме по нотации Чена:**
	* Не менее 5 сущностей.
	* У каждой сущности должно быть не менее 4 атрибутов.
	* Обязательно указать первичные ключи (PK).
	* Указать тип связей между сущностями (1:1, 1:\*, *:*).
	* Использовать слабые сущности, если требуется.
	* При необходимости указать составные атрибуты (например, ФИО может состоять из фамилии, имени, отчества).
2. **Порядок выполнения работы:**
	* Определите основные сущности предметной области.
	* Определите атрибуты каждой сущности.
	* Определите связи между сущностями и их типы.
	* Постройте ER-диаграмму в нотации Чена вручную или с использованием любого графического редактора (Draw.io, Creately).

**Контрольные вопросы:**

1. Что такое сущность в ER-диаграмме?
2. Что такое атрибут и какие типы атрибутов бывают?
3. Чем слабая сущность отличается от обычной сущности?
4. Какие бывают типы связей между сущностями?
5. Для чего используется первичный ключ?

**Практическая работа №3 «Нормализация таблиц реляционной БД»**

**Цель работы:** научиться выполнять нормализацию таблиц реляционных баз данных для устранения избыточности данных и обеспечения целостности данных.

**Задание:**

1. Ознакомьтесь с теоретическими понятиями нормальных форм:
	* Первая нормальная форма (1NF).
	* Вторая нормальная форма (2NF).
	* Третья нормальная форма (3NF).
2. Исходная ненормализованная таблица:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID Заказа** | **Дата заказа** | **ФИО клиента** | **Адрес клиента** | **Телефон клиента** | **Название книги** | **Автор книги** | **Кол-во экземпляров** | **Цена за штуку** |
| 101 | 10.04.2025 | Иванов И.И. | ул. Ленина, д.5 | 89999999999 | «Война и мир» | Толстой Л.Н. | 2 | 500 |
| 101 | 10.04.2025 | Иванов И.И. | ул. Ленина, д.5 | 89999999999 | «Анна Каренина» | Толстой Л.Н. | 1 | 450 |
| 102 | 10.04.2025 | Петров П.П. | ул. Мира, д.10 | 89998888888 | «Преступление и наказание» | Достоевский Ф.М. | 1 | 600 |

1. Требуется:
* Провести нормализацию данной таблицы до третьей нормальной формы (3NF).
* Описать каждый этап нормализации: каковы зависимости между атрибутами, какие атрибуты образуют ключи, какие атрибуты требуют вынесения в отдельные таблицы?
* Построить итоговые нормализованные таблицы (показать структуру и связи между ними).

Дополнительно:

* Указать типы связей между таблицами (один-ко-многим, многие-ко-многим).
* Построить схему базы данных после нормализации.

**Порядок выполнения:**

1. Определите функциональные зависимости в исходной таблице.
2. Проведите приведение к 1NF: устраните многозначные атрибуты.
3. Проведите приведение ко 2NF: удалите частичные зависимости.
4. Проведите приведение к 3NF: устраните транзитивные зависимости.
5. Постройте итоговую схему базы данных.

**Контрольные вопросы:**

1. Что называется нормализацией базы данных?
2. Что такое функциональная зависимость атрибутов?
3. Чем отличается 1NF от 2NF и 3NF?
4. Почему важно избегать избыточности данных?
5. Какие проблемы возникают в ненормализованных базах данных?

**Практическая работа №4 «Освоение принципов проектирования БД»**

## Цель работы: научиться применять основные принципы проектирования базы данных: анализ предметной области, выявление сущностей и атрибутов, определение связей между сущностями, построение концептуальной модели данных.

## Задание:

1. Изучите теоретические основы проектирования баз данных:
	* Понятие сущности и атрибута.
	* Типы связей между сущностями: один-ко-многим, многие-ко-многим, один-к-одному.
	* Принципы построения концептуальной схемы базы данных.
2. Предметная область: разработайте базу данных для **онлайн-сервиса бронирования конференц-залов.**

Сервис должен позволять:

* + Вести учет залов, их характеристик и доступности.
	+ Учитывать клиентов и их заказы на бронирование залов.
	+ Учитывать стоимость аренды и дополнительного оборудования.
	+ Вести учет оплат и статуса бронирования.
1. Выполните следующие действия:

**1) Выявите сущности и их атрибуты** (например: Клиент, Зал, Бронирование, Оплата, Оборудование).

**2) Определите связи между сущностями и их типы** (например: один клиент – много бронирований, один зал – много бронирований).

**3) Постройте концептуальную модель базы данных (ER-диаграмму) на основе выявленных сущностей и связей.**

**4) Определите ключевые атрибуты для каждой сущности (первичные ключи) и укажите внешние ключи.**

## Порядок выполнения:

1. Сформулировать задание по предметной области.
2. Выделить основные сущности и их атрибуты.
3. Определить связи между сущностями.
4. Построить концептуальную ER-диаграмму.
5. Сформировать описание структуры таблиц с указанием первичных и внешних ключей.

## Контрольные вопросы:

1. Что такое сущность и атрибут в проектировании БД?
2. Какие существуют типы связей между сущностями?
3. Чем отличается первичный ключ от внешнего ключа?
4. Для чего необходима концептуальная модель базы данных?
5. Какие принципы следует соблюдать при проектировании структуры базы данных?

**Практическая работа №5 «Проектирование реляционной БД»**

## Цель работы: освоить этапы проектирования реляционной базы данных на основе анализа предметной области: определение сущностей, атрибутов, связей и построение логической структуры базы данных.

## Задание:

1. Изучите основные принципы проектирования реляционных баз данных:
	* Понятие реляционной модели данных.
	* Принципы построения таблиц и установления связей между ними.
	* Роль первичных и внешних ключей в организации данных.
2. Предметная область: разработайте **реляционную базу данных для управления библиотекой.**

База данных должна обеспечивать:

* + Учёт книг и их характеристик.
	+ Учёт авторов и жанров.
	+ Учет читателей библиотеки.
	+ Учёт фактов выдачи и возврата книг.
1. Выполните следующие действия:

**1) Выделите сущности и их атрибуты в количестве, не менее 10 и включите обязательные из них:**

* + Книга (ID, название, год издания, издательство, жанр, автор)
	+ Автор (ID, ФИО, дата рождения)
	+ Жанр (ID, название жанра)
	+ Читатель (ID, ФИО, дата рождения, телефон, email)
	+ Выдача книги (ID, ID книги, ID читателя, дата выдачи, дата возврата)

**2) Определите связи между сущностями:**

**3) Постройте логическую модель базы данных в форме схемы связей.**

**4) Определите для каждой таблицы:**

* + Первичные ключи (PK)
	+ Внешние ключи (FK)
	+ Типы связей (один-ко-многим, многие-ко-многим).

## Порядок выполнения:

1. Провести анализ предметной области.
2. Составить список сущностей и атрибутов.
3. Построить логическую схему реляционной БД.
4. Описать ключи и связи между таблицами.
5. Составить описание структуры базы данных.

## Контрольные вопросы:

1. Что такое реляционная модель данных?
2. Чем отличается первичный ключ от внешнего?
3. Что такое нормализация базы данных?
4. Какие типы связей существуют между таблицами?
5. Для чего необходимо логическое проектирование БД?

**Практическая работа №6 «Написание простейших SELECT-запросов»**

## Цель работы: научиться составлять базовые SQL-запросы для извлечения данных из одной таблицы с использованием операторов SELECT и FROM.

## Задание:

1. Используйте следующую таблицу для выполнения запросов:

**Таблица STUDENTS (Студенты)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID\_Student** | **FullName** | **BirthDate** | **GroupName** | **AverageMark** |
| 1 | Иванов Иван | 2005-05-21 | Веб-23 | 4.5 |
| 2 | Петрова Анна | 2004-11-13 | ИС-22 | 4.0 |
| 3 | Сидоров Максим | 2005-03-07 | Веб-23 | 4.8 |
| 4 | Смирнова Екатерина | 2006-01-15 | ПО-24 | 4.2 |
| 5 | Кузнецов Артём | 2004-07-10 | ИС-22 | 3.9 |

1. Напишите SQL-запросы:

**1. Вывести все данные о студентах.**

1. **Вывести только имена студентов.**
2. **Вывести только даты рождения студентов.**
3. Вывести названия групп всех студентов.
4. Вывести средний балл всех студентов.
5. Вывести ID и ФИО студентов.
6. Вывести ФИО студентов и их средний балл.
7. Вывести дату рождения и название группы студентов.
8. Вывести ID студента и его группу.
9. Вывести все данные о студентах без использования знака \* (явное перечисление столбцов).

## Контрольные вопросы:

1. Для чего используется оператор SELECT в SQL?
2. Что делает оператор FROM в SQL?
3. Как выбрать все столбцы таблицы?
4. Как выбрать только некоторые столбцы таблицы?
5. Можно ли использовать SELECT без FROM?

**Практическая работа №7 «Написание SELECT-запросов с логическими операторами, DISTINCT, условиями WHERE, операторами IS NULL, BETWEEN, IN и LIKE»**

## Цель работы: освоить написание SQL-запросов с условиями выборки данных, использованием фильтрации, проверкой на пустые значения, выборкой по диапазону и шаблонам.

## Задание:

1. Используйте следующую таблицу для выполнения запросов:

**Таблица EMPLOYEES (Сотрудники)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID\_Employee** | **FullName** | **Position** | **Department** | **Salary** | **HireDate** | **PhoneNumber** |
| 1 | Иванов Иван | Менеджер | Продажи | 60000 | 2020-04-01 | 89991112233 |
| 2 | Петрова Анна | Бухгалтер | Финансы | 70000 | 2019-05-15 | NULL |
| 3 | Сидоров Максим | Программист | IT | 90000 | 2021-09-20 | 89994445566 |
| 4 | Кузнецова Дарья | Дизайнер | Маркетинг | 75000 | 2022-01-10 | 89990001122 |
| 5 | Орлов Максим | Программист | IT | 88000 | 2018-11-05 | NULL |

1. Напишите SQL-запросы:

1. Вывести всех сотрудников без повторений их должностей.

2. Вывести всех сотрудников, у которых зарплата больше 80000.

3. Вывести сотрудников отдела IT.

4. Найти сотрудников, у которых отсутствует номер телефона.

5. Найти сотрудников, принятых на работу с 1 января 2020 года по 31 декабря 2022 года.

6. Вывести сотрудников, занимающих должности «Программист» или «Дизайнер».

7. Найти сотрудников, чьи фамилии начинаются на букву «К».

8. Найти сотрудников, чьи имена содержат «а».

9. Вывести всех сотрудников, работающих в отделах «Продажи» или «Финансы», с зарплатой более 65000.

10. Вывести всех сотрудников, у которых зарплата не входит в диапазон от 60000 до 80000.

## Контрольные вопросы:

1. Что делает оператор DISTINCT в SQL?
2. Для чего используется условие WHERE?
3. Как проверить значение на NULL в SQL?
4. Что делает оператор BETWEEN?
5. Как использовать оператор LIKE для поиска по шаблону?

**Практическая работа №8 «Написание SELECT-запросов с сортировками, группировками и агрегатными функциями»**

## Цель работы: научиться писать SQL-запросы с использованием операторов ORDER BY, GROUP BY, HAVING, а также агрегатных функций (COUNT, SUM, AVG, MAX, MIN) для обработки данных.

## Задание:

1. Используйте следующую таблицу для выполнения запросов:

**Таблица ORDERS (Заказы)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID\_Order** | **CustomerName** | **Product** | **Quantity** | **PricePerUnit** | **OrderDate** |
| 1 | Иванов Иван | Ноутбук | 2 | 50000 | 2024-01-15 |
| 2 | Смирнова Анна | Планшет | 1 | 25000 | 2024-02-10 |
| 3 | Петров Петр | Смартфон | 3 | 30000 | 2024-02-18 |
| 4 | Иванов Иван | Планшет | 1 | 25000 | 2024-03-02 |
| 5 | Кузнецова Мария | Ноутбук | 1 | 50000 | 2024-03-05 |

1. Напишите SQL-запросы:

**1. Вывести все заказы, отсортированные по дате заказа по возрастанию.**

**2. Вывести все заказы, отсортированные по имени клиента в алфавитном порядке.**

**3. Посчитать общее количество заказов.**

**4. Найти максимальную цену за единицу товара.**

**5. Найти минимальное количество единиц в одном заказе.**

**6. Вычислить среднюю цену за единицу товара.**

**7. Посчитать общую сумму всех заказов (Quantity \* PricePerUnit).**

**8. Вывести количество заказов по каждому товару.**

**9. Вывести общую сумму заказов по каждому клиенту.**

**10. Вывести клиентов, у которых сумма заказов превышает 100000.**

## Контрольные вопросы:

1. Что делает оператор ORDER BY?
2. В чем разница между GROUP BY и ORDER BY?
3. Для чего нужны агрегатные функции в SQL?
4. Что делает оператор HAVING?
5. В чем отличие WHERE и HAVING?

**Практическая работа №9 «Написание многотабличных запросов, запросов с ограничением выборки и подзапросов»**

## Цель работы: научиться составлять SQL-запросы с объединением нескольких таблиц (JOIN), ограничением выборки (LIMIT) и использованием вложенных подзапросов (SELECT внутри SELECT).

## Задание:

1. Используйте следующие таблицы для выполнения запросов:

**Таблица CUSTOMERS (Клиенты)**

|  |  |  |
| --- | --- | --- |
| **CustomerID** | **CustomerName** | **City** |
| 1 | Иванов Иван | Москва |
| 2 | Смирнова Анна | Санкт-Петербург |
| 3 | Петров Петр | Казань |
| 4 | Кузнецова Мария | Новосибирск |

**Таблица ORDERS (Заказы)**

|  |  |  |  |
| --- | --- | --- | --- |
| **OrderID** | **CustomerID** | **OrderDate** | **Amount** |
| 101 | 1 | 2024-01-15 | 50000 |
| 102 | 2 | 2024-02-10 | 30000 |
| 103 | 1 | 2024-02-10 | 40000 |
| 104 | 3 | 2024-03-20 | 25000 |
| 105 | 2 | 2024-04-05 | 35000 |

1. Напишите SQL-запросы:

**1. Вывести список всех заказов вместе с именами клиентов.**

**2. Вывести заказы, оформленные клиентами из Москвы.**

**3. Найти клиентов, которые ещё не сделали ни одного заказа.**

**4. Вывести 3 самых дорогих заказа.**

**5. Найти клиентов, у которых сумма хотя бы одного заказа больше 40000.**

**6. Найти клиентов, у которых все заказы меньше 40000 (подзапрос).**

**7. Найти среднюю сумму заказа по каждому клиенту.**

**8. Вывести клиентов, у которых более одного заказа.**

## Контрольные вопросы:

1. Для чего используется оператор JOIN?
2. В чем разница между INNER JOIN, LEFT JOIN, RIGHT JOIN?
3. Как работает оператор LIMIT?
4. Что такое подзапрос? Когда его лучше использовать?
5. Можно ли использовать подзапрос в WHERE и FROM одновременно?

**Практическая работа №10 «Написание запросов с обобщенным табличным выражением, объединением запросов и условной логикой»**

## Цель работы: научиться использовать обобщённые табличные выражения (CTE), объединение запросов (UNION, UNION ALL) и условную логику (CASE) при составлении SQL-запросов.

## Задание:

1. Используйте следующие таблицы для выполнения запросов:

**Таблица EMPLOYEES (Сотрудники)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EmployeeID** | **FullName** | **Departments** | **Salary** | **HireDate** |
| 1 | Иванов Иван | Продажи | 60000 | 2020-01-15 |
| 2 | Смирнова Анна | IT | 90000 | 2019-03-10 |
| 3 | Петров Петр | Финансы | 75000 | 2021-07-25 |
| 4 | Кузнецова Мария | IT | 92000 | 2020-11-30 |
| 5 | Сидоров Максим | Продажи | 58000 | 2022-02-18 |

**Таблица CONTRACTORS (Подрядчики)**

|  |  |  |  |
| --- | --- | --- | --- |
| **ContractorID** | **ContractorName** | **ServiceType** | **Fee** |
| 1 | ООО «ТехноПро» | IT | 100000 |
| 2 | ИП «Бухгалтерия+» | Финансы | 60000 |
| 3 | ООО «Продавец» | Продажи | 55000 |

1. Напишите SQL-запросы:

**1. Создать обобщённое табличное выражение (CTE), которое выберет сотрудников с зарплатой выше 60000.**

**2. Объединить сотрудников и подрядчиков по отделам/видам услуг с использованием** UNION ALL**.**

**3. Объединить сотрудников и подрядчиков с использованием**UNION**(без дубликатов).**

**4. Добавить столбец «Уровень дохода» с условной логикой (**CASE**) для сотрудников:**

* Если зарплата меньше 60000 – «Низкий доход»
* От 60000 до 80000 – «Средний доход»
* Более 80000 – «Высокий доход»

**5. Добавить столбец «Тип» с условной логикой (CASE) для общего списка сотрудников и подрядчиков**

## Контрольные вопросы:

1. Что такое обобщённое табличное выражение (CTE)?
2. В чём разница между UNION и UNION ALL?
3. Как работает оператор CASE?
4. Можно ли использовать CASE в предложении SELECT?
5. Для чего используется WITH в SQL-запросах?

**Практическая работа №11 «Написание запросов на создание и удаление таблиц и БД, запросов на добавление, удаление и обновление данных»**

## Цель работы: научиться создавать и удалять базы данных и таблицы, а также выполнять базовые операции добавления (INSERT), удаления (DELETE) и обновления (UPDATE) данных.

## Задание:

1. **Создать новую базу данных** CompanyDB**.**
2. **Перейти к использованию этой базы данных.**
3. **Создать таблицу EMPLOYEES (Сотрудники):** EmployeeID – INT PRIMARY KEY AUTO\_INCREMENT, FullName – VARCHAR(100), Position – VARCHAR(50), Salary – DECIMAL(10,2), HireDate – DATE.
4. **Создать таблицу DEPARTMENTS (Отделы):** DepartmentID – INT PRIMARY KEY AUTO\_INCREMENT, DepartmentName – VARCHAR(100).
5. **Добавить несколько записей в таблицу EMPLOYEES.**
6. Добавить несколько записей в таблицу DEPARTMENTS.
7. **Изменить зарплату Иванова Ивана на 65000.**
8. **Переименовать отдел «Финансовый отдел» в «Финансы».**
9. **Удалить сотрудника по имени Петров Петр.**
10. **Удалить отдел «Продажи».**
11. **Удалить таблицу EMPLOYEES.**
12. **Удалить всю базу данных CompanyDB.**

## Контрольные вопросы:

1. Чем отличается DROP TABLE от DELETE FROM?
2. Что делает команда INSERT?
3. Как изменить существующие данные в таблице?
4. Что произойдет при выполнении команды DROP DATABASE?
5. В каких случаях используют UPDATE?

**Практическая работа №12 «Представления»**

## Цель работы: научиться создавать представления (VIEW) в базе данных для удобной работы с данными, а также обновлять и удалять представления.

## Задание:

Используйте следующие таблицы для работы:

**Таблица EMPLOYEES (Сотрудники)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EmployeeID** | **FullName** | **Position** | **Salary** | **DepartmentID** |
| 1 | Иванов Иван | Менеджер | 60000 | 1 |
| 2 | Смирнова Анна | Разработчик | 90000 | 2 |
| 3 | Петров Петр | Бухгалтер | 75000 | 3 |
| 4 | Кузнецова Мария | Разработчик | 92000 | 2 |

**Таблица DEPARTMENTS (Отделы)**

|  |  |
| --- | --- |
| **DepartmentID** | **DepartmentName** |
| 1 | Продажи |
| 2 | IT |
| 3 | Финансы |

**1. Создать представление, отображающее сотрудников только из отдела "IT".**

**2. Создать представление, показывающее ФИО сотрудников и название их отделов.**

**3. Создать представление, которое показывает сотрудников с зарплатой выше 80000.**

**4. Вывести данные из представления** IT\_Employees**.**

**5. Вывести данные из представления** Employee\_Department**.**

**6. Вывести данные из представления** HighSalaryEmployees**.**

**7. Обновить представление** HighSalaryEmployees**, чтобы отображать сотрудников с зарплатой выше 85000.**

**8. Удалить представление** IT\_Employees**.**

## Контрольные вопросы:

1. Что такое представление (VIEW) в базе данных?
2. Какие преимущества дает использование представлений?
3. Можно ли обновлять данные через представление?
4. В чем разница между CREATE VIEW и CREATE OR REPLACE VIEW?
5. Что происходит при удалении представления (DROP VIEW)?

**Практическая работа №13 «Хранимые процедуры и триггеры»**

## Цель работы: научиться создавать хранимые процедуры (PROCEDURE) и триггеры (TRIGGER) в базе данных для автоматизации обработки данных и обеспечения их целостности.

## Задание:

### **Этап 1. Подготовка**

Используйте следующую таблицу:

**Таблица EMPLOYEES (Сотрудники)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **EmployeeID** | **FullName** | **Position** | **Salary** | **HireDate** |
| 1 | Иванов Иван | Менеджер | 60000 | 2020-01-15 |
| 2 | Смирнова Анна | Разработчик | 90000 | 2019-03-10 |
| 3 | Петров Петр | Бухгалтер | 75000 | 2021-07-25 |
| 4 | Кузнецова Мария | Разработчик | 92000 | 2020-11-30 |

### **Этап 2. Хранимые процедуры**

**1. Создать процедуру для добавления нового сотрудника в таблицу EMPLOYEES.**

**2. Вызвать процедуру и добавить сотрудника:**

**3. Создать процедуру для повышения зарплаты сотрудника на 10% по его идентификатору.**

**4. Вызвать процедуру для повышения зарплаты Иванова Ивана.**

### Этап 3. Триггеры

**5. Создать таблицу SalaryChanges для фиксации изменений зарплаты, состоящей из полей:** ChangeID, EmployeeID, OldSalary, NewSalary, ChangeDate.

**6. Создать триггер, который будет срабатывать после обновления зарплаты в EMPLOYEES и записывать изменения в SalaryChanges.**

**7. Обновить зарплату сотруднику вручную.**

**8. Проверить, появилась ли запись в SalaryChanges.**

## Контрольные вопросы:

1. Что такое хранимая процедура? Какие у неё преимущества?
2. Что такое триггер? Когда его лучше использовать?
3. Чем отличается триггер BEFORE UPDATE от AFTER UPDATE?
4. Какие возможные риски существуют при использовании триггеров?
5. Можно ли вызывать процедуру внутри триггера?

**Практическая работа №14 «Управление доступом к данным»**

## Цель работы: научиться управлять правами доступа к базам данных и таблицам: предоставлять, изменять и отзывать права пользователей в СУБД.

## Задание:

Создайте тестовую базу данных AccessDB и таблицу Employees.

**1. Создать нового пользователя** user\_viewer **с паролем** viewer123**, который будет иметь права только на чтение таблицы** Employees**.**

**2. Создать нового пользователя** user\_editor **с паролем** editor123**, который будет иметь права на чтение и изменение данных в таблице** Employees**, но не на удаление.**

**3. Создать пользователя** user\_admin **с паролем** admin123**, который имеет полный доступ ко всей базе данных** AccessDB**.**

**4. Проверить, какие права имеет пользователь** user\_editor**.**

**5. Отозвать у** user\_editor **право на вставку новых записей (**INSERT**).**

**6. Изменить права пользователя** user\_viewer**, разрешив ему только просмотр имён сотрудников (через создание отдельного представления** EmployeesNames**).**

**7. Удалить пользователя** user\_editor**.**

## Контрольные вопросы:

1. Для чего необходимо управлять правами доступа в базе данных?
2. В чём разница между операциями GRANT и REVOKE?
3. Какие существуют уровни прав в СУБД (база данных, таблица, столбец)?
4. Что произойдет, если выдать пользователю право ALL PRIVILEGES?
5. Можно ли ограничить доступ к отдельным столбцам таблицы? Как это сделать?

**Практическая работа №15 «Резервное копирование и восстановление данных»**

## Цель работы: научиться выполнять резервное копирование баз данных и восстановление данных средствами Microsoft SQL Server Management Studio (SSMS) или через T-SQL.

## Задание:

Создайте базу данных BackupDB и таблицу Products:

**1. Выполните полное резервное копирование базы данных**BackupDB**в файл.**

**2. Удалите базу данных**BackupDB**.**

**3. Восстановите базу данных из файла**.bak**.**

**4. Подключитесь к восстановленной базе**BackupDB**и убедитесь, что таблица**Products**и данные восстановлены.**

## Контрольные вопросы:

1. Что делает команда BACKUP DATABASE в MS SQL Server?
2. Чем отличается параметр WITH FORMAT при бэкапе?
3. Что означает ключевое слово WITH REPLACE при восстановлении?
4. Какие существуют типы резервного копирования в MS SQL Server?
5. Как можно настроить автоматическое резервное копирование базы данных?

**Практическая работа №16 «Работа с реляционной БД в СУБД MySQL Workbench»**

## Цель работы: научиться работать с реляционной базой данных через графический интерфейс MySQL Workbench: создавать базу данных, проектировать таблицы, устанавливать связи между ними, заполнять данными, а также выполнять базовые SQL-запросы.

## Оборудование и ПО:

* Компьютер с установленной **MySQL Workbench** (версия 8.0 или выше).
* Сервер базы данных MySQL (локальный или удалённый).
* Доступ к MySQL-серверу с правами создания баз данных и таблиц.

## Ход работы:

**1. Создайте новую базу данных** с именем StudentCoursesDB **через SQL Editor** или через вкладку **«SCHEMAS»** – правая кнопка мыши – **«Create Schema»**.

**2. Создайте в базе данных три таблицы:**

**Таблица Students (Студенты):**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| StudentID | INT PRIMARY KEY AUTO\_INCREMENT | Идентификатор студента |
| FullName | VARCHAR(100) | Полное имя |
| BirthDate | DATE | Дата рождения |

**Таблица Courses (Курсы):**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| CourseID | INT PRIMARY KEY AUTO\_INCREMENT | Идентификатор курса |
| CourseName | VARCHAR(100) | Название курса |
| Duration | INT | Продолжительность (в часах) |

**Таблица Enrollments (Записи о зачислении):**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| EnrollmentID | INT PRIMARY KEY AUTO\_INCREMENT | Идентификатор записи |
| StudentID | INT | Внешний ключ на Students |
| CourseID | INT | Внешний ключ на Courses |
| EnrollmentDate | DATE | Дата зачисления |

**3. Настройте связи между таблицами через «EER Diagram» или вручную:**

* В таблице **Enrollments** поле StudentID ссылается на Students.StudentID (ON DELETE CASCADE).
* В таблице **Enrollments** поле CourseID ссылается на Courses.CourseID (ON DELETE CASCADE).

**4. Вставьте тестовые данные в таблицы.**

**5. Выполните следующие запросы:**

1. **Вывести всех студентов.**
2. **Вывести список всех курсов.**
3. **Вывести имена студентов и названия курсов, на которые они зачислены.**
4. **Посчитать количество студентов, зачисленных на каждый курс.**
5. **Найти студентов, которые записаны на более чем один курс.**

## Порядок выполнения:

1. Создать базу данных и таблицы.
2. Установить связи между таблицами.
3. Заполнить таблицы данными.
4. Построить EER-диаграмму в MySQL Workbench.
5. Выполнить предложенные SQL-запросы.
6. Сделать скриншоты всех этапов: создание таблиц, заполнение, EER-диаграмма, результаты запросов.
7. Оформить отчет по работе.

## Контрольные вопросы:

1. Что такое реляционная база данных?
2. Какие типы связей существуют между таблицами?
3. Для чего нужна внешняя связь (FOREIGN KEY)?
4. Что делает команда ON DELETE CASCADE?
5. Как в MySQL Workbench можно построить диаграмму связей (EER Diagram)?

**Практическая работа №17 «Работа с реляционной БД в СУБД PostgreSQL»**

## Цель работы: научиться работать с реляционной базой данных через командную строку PostgreSQL (psql) или через графический интерфейс (например, pgAdmin), а также освоить создание базы данных, таблиц, связей между ними, заполнение данными и выполнение запросов.

## Оборудование и ПО:

* Сервер базы данных PostgreSQL (версия 13 или выше).
* pgAdmin или любой доступ к консоли psql.
* Права администратора для создания баз данных и таблиц.

## Ход работы:

1. **Создать новую базу данных с именем** LibraryDB **через psql или pgAdmin**
2. **Подключиться к базе данных** LibraryDB**.**
3. **Создайте в базе данных три таблицы**

**Таблица Authors (Авторы):**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| AuthorID | SERIAL PRIMARY KEY | Уникальный идентификатор |
| AuthorName | VARCHAR(100) NOT NULL | Имя автора |
| BirthYear | INT | Год рождения |

**Таблица Books (Книги):**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| BookID | SERIAL PRIMARY KEY | Уникальный идентификатор |
| Title | VARCHAR(150) NOT NULL | Название книги |
| PublishYear | INT | Год публикации |
| AuthorID | INT | Внешний ключ на Authors |

**Таблица Borrowings (Выдачи):**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| BorrowingID | SERIAL PRIMARY KEY | Уникальный идентификатор |
| BookID | INT | Внешний ключ на Books |
| BorrowerName | VARCHAR(100) | Имя читателя |
| BorrowDate | DATE | Дата выдачи |

**4. Установите внешние ключи**

**5. Вставьте тестовые данные**

**6. Выполните следующие запросы**

1. **Вывести всех авторов.**
2. **Вывести все книги с именами их авторов.**
3. **Найти книги, выданные в феврале 2024 года.**
4. **Вывести количество книг у каждого автора.**
5. **Найти автора, книги которого ещё ни разу не брали.**

## Порядок выполнения:

1. Создать базу данных LibraryDB.
2. Создать и настроить таблицы с ограничениями целостности данных.
3. Заполнить таблицы тестовыми данными.
4. Выполнить все SQL-запросы.
5. Оформить отчет с текстом запросов и скриншотами результата выполнения.

## Контрольные вопросы:

1. В чем особенности типовых данных PostgreSQL (например, SERIAL)?
2. Чем отличается команда SERIAL от SEQUENCE в PostgreSQL?
3. Для чего нужны внешние ключи?
4. Что происходит при использовании ON DELETE CASCADE?
5. Как реализовать выборку данных из нескольких связанных таблиц?

**Практическая работа №18 «Работа с реляционной БД в PhpMyAdmin»**

## Цель работы: научиться использовать графический интерфейс PhpMyAdmin для создания базы данных, проектирования таблиц, установки связей, заполнения данными и выполнения базовых SQL-запросов.

## Оборудование и ПО:

* Веб-браузер.
* Доступ к локальному серверу (XAMPP, MAMP, WAMP) или удаленному серверу с установленным **PhpMyAdmin.**
* Учётная запись администратора или доступ к созданию БД.

## Ход работы:

**1. Через PhpMyAdmin создайте новую базу данных** ShopDB**:** Откройте PhpMyAdmin – вкладка «**Базы данных»** – введите имя ShopDB – выберите кодировку utf8mb4\_general\_ci – нажмите **Создать**.

**2. Создайте в базе данных три таблицы через интерфейс PhpMyAdmin:**

**Таблица Customers (Покупатели):**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| CustomerID | INT (AUTO\_INCREMENT) | Уникальный идентификатор |
| CustomerName | VARCHAR(100) | Имя покупателя |
| Phone | VARCHAR(20) | Номер телефона |

**Таблица Products (Товары):**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| ProductID | INT (AUTO\_INCREMENT) | Уникальный идентификатор |
| ProductName | VARCHAR(100) | Наименование товара |
| Price | DECIMAL(10,2) | Цена |

**Таблица Orders (Заказы):**

|  |  |  |
| --- | --- | --- |
| **Поле** | **Тип данных** | **Описание** |
| OrderID | INT (AUTO\_INCREMENT) | Уникальный идентификатор |
| CustomerID | INT | Внешний ключ на Customers |
| ProductID | INT | Внешний ключ на Products |
| OrderDate | DATE | Дата заказа |

**3. Настройте связи между таблицами через вкладку «Связи» (Relation view):**

* В таблице Orders, поле CustomerID ссылается на Customers.CustomerID.
* В таблице Orders, поле ProductID ссылается на Products.ProductID.
* Установите режим **ON DELETE CASCADE** для связей.

**4. Через вкладку «Вставить» заполните тестовыми данными:**

**Покупатели:**

* Иванов Иван, телефон: +79991234567
* Петрова Анна, телефон: +79998765432
* Сидоров Максим, телефон: +79995678901

**Товары:**

* Ноутбук – 50000
* Смартфон – 30000
* Планшет – 20000

**Заказы:**

* Иванов купил Ноутбук 10.03.2024
* Петрова купила Смартфон 12.03.2024
* Сидоров купил Планшет 15.03.2024

**5. Выполните следующие запросы:**

1. **Вывести всех покупателей.**
2. **Вывести все товары.**
3. **Показать все заказы с именами покупателей и названиями товаров.**
4. **Посчитать общее количество заказов для каждого покупателя.**
5. **Показать товары стоимостью выше 25000.**

## Порядок выполнения:

1. Создать базу данных ShopDB.
2. Создать таблицы и настроить их через интерфейс PhpMyAdmin.
3. Установить внешние ключи между таблицами.
4. Заполнить таблицы тестовыми данными.
5. Выполнить все SQL-запросы через вкладку «SQL».
6. Сделать скриншоты основных этапов: создание БД, таблиц, заполнение данных, выполнения запросов.
7. Оформить отчёт по результатам работы.

## Контрольные вопросы:

1. Для чего устанавливаются внешние ключи между таблицами?
2. Какой тип связи реализован между таблицами Заказы и Покупатели?
3. Что делает параметр ON DELETE CASCADE?
4. В чем особенности работы с PhpMyAdmin по сравнению с текстовым написанием SQL?
5. Как проверить целостность данных в связанном запросе?

**Практическая работа №19 «Проектирование реляционной БД на заданную тему в представленных СУБД»**

## Цель работы: научиться самостоятельно проектировать реляционные базы данных на основе выбранной предметной области, создавать структуры таблиц и реализовывать проект в выбранной СУБД (Microsoft SQL Server, MySQL Workbench или pgAdmin/PostgreSQL).

## Оборудование и ПО:

* Microsoft SQL Server Management Studio (MSSQL).
* MySQL Workbench.
* pgAdmin (PostgreSQL).

## Ход работы:

Выберите **одну** из предложенных тем для проектирования базы данных:

1. База данных книжного интернет-магазина.
2. База данных стоматологической клиники.
3. База данных доставки еды.
4. База данных автосалона.
5. База данных киностудии.
6. База данных фитнес-клуба.
7. База данных концертных мероприятий.
8. База данных туристического агентства.
9. База данных университета.
10. База данных ветеринарной клиники.
11. База данных аренды квартир.
12. База данных службы такси.
13. База данных онлайн-курсов.
14. База данных выставочного центра.
15. База данных службы технической поддержки.
16. База данных страховой компании.
17. База данных библиотечного фонда.
18. База данных языковой школы.
19. База данных сервиса бронирования отелей.
20. База данных корпоративного обучения сотрудников.
21. База данных агрокомпании (учет животных/посевов).
22. База данных онлайн-магазина одежды.
23. База данных теннисного клуба.
24. База данных медицинских обследований.
25. База данных аренды строительной техники.

**1. Определите сущности и атрибуты (какие таблицы нужны, какие у них будут поля).**

**2. Определите связи между сущностями**

* Один-к-одному (1:1)
* Один-ко-многим (1:N)
* Многие-ко-многим (N:N)

**3. Нарисуйте концептуальную ER-диаграмму.**

**4. Выберите СУБД для реализации проекта (MSSQL, MySQL или PostgreSQL).**

**5. Создайте базу данных и необходимые таблицы:**

* Укажите типы данных.
* Установите первичные ключи.
* Установите внешние ключи.
* Настройте ограничения (NOT NULL, UNIQUE, DEFAULT).

**6. Заполните таблицы тестовыми данными (не менее 10 записей на таблицу).**

**7. Напишите и выполните следующие SQL-запросы в количестве 40 штук:**

* Вывод всех данных из таблиц (SELECT).
* Вывод связанных данных через JOIN.
* Агрегация данных (COUNT, SUM, AVG, MAX, MIN).
* Сортировка данных (ORDER BY).
* Фильтрация данных (WHERE, LIKE, IN, BETWEEN).
* Группировка данных (GROUP BY, HAVING).

## Порядок выполнения:

1. Выбрать тему.
2. Спроектировать структуру базы данных.
3. Реализовать проект в выбранной СУБД.
4. Заполнить таблицы тестовыми данными.
5. Выполнить SQL-запросы для работы с данными.
6. Построить ER-диаграмму базы.
7. Подготовить отчёт:
	* Название темы.
	* Описание структуры базы данных.
	* Скриншоты кода создания таблиц.
	* Скриншоты тестовых данных.
	* Скриншоты выполнения SQL-запросов.
	* ER-диаграмму.

## Контрольные вопросы:

1. Что такое реляционная модель базы данных?
2. Чем отличается внешний ключ от первичного?
3. Как определить типы связей между таблицами?
4. Какие правила нормализации необходимо соблюдать при проектировании БД?
5. Какие преимущества даёт использование внешних ключей?

**Самостоятельная работа №2 «Защита реляционных БД на заданную тему»**

### **Цель работы:** научиться проектировать и реализовывать базовые меры защиты реляционных баз данных, развить навыки обеспечения целостности, конфиденциальности и доступности данных. Освоить создание пользователей с различными правами доступа, настройку резервного копирования, организацию контроля действий в базе данных с помощью триггеров и политик безопасности в выбранной СУБД.

### **План выступления**

**1. Вступление**

* Название проекта базы данных и выбранная предметная область.
* Краткое описание: для чего создавалась база данных.

**2. Цели разработки**

* Организация структурированного хранения информации.
* Обеспечение целостности, актуальности и безопасности данных.
* Оптимизация работы с данными: удобный доступ, обновление, удаление.
* Разграничение прав пользователей для защиты информации.

**3. Проектирование БД**

* Какие сущности и атрибуты были выделены.
* Как были установлены связи между таблицами.
* Какие ограничения применены (первичные, внешние ключи).

**4. Защита базы данных**

* Выделение ролей пользователей с разными уровнями доступа.
* Реализация резервного копирования для защиты данных.
* Создание триггеров или представлений для повышения безопасности.

**5. Проверка работы**

* Как тестировалась работа с ограничениями доступа.
* Как тестировалась возможность восстановления данных из резервных копий.

**6. Заключение**

* Как база данных решает поставленные задачи.
* Перспективы дальнейшего развития проекта (добавление новых функций безопасности, расширение функционала).

**Информационное обеспечение**

**Основные источники:**

Учебные пособия:

1. Фуфаев, Э.В. Базы данных: учеб. пособие дли студ. учреждений сред, проф. образования / Э. В.Фуфаев, Д. Э.Фуфаев. – 7-е изд., стер. – М: Издательский центр «Академия», 2022 — 320 с.: ISBN 978-5-7695-8959-1

2. Маркин, А. В. Программирование на SQL: учебное пособие для среднего профессионального образования / А. В. Маркин. – Москва: Издательство Юрайт, 2023 – 434 с. –(Профессиональное образование). – Текст: непосредственный ISBN 978-5-534-11093-7

**Дополнительные источники:**

Учебные пособия:

1. Голицына, О.Л. Базы данных / О.Л. Голицына, Н.В. Максимов, И.И. Попов. – М.: Форум, 2020. – 352 c.

2. Голицына, О.Л. Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. – М.: Форум, 2022. – 400 c.

3. Карпова, И.П. Базы данных: Учебное пособие / И.П. Карпова – СПб.: Питер, 2023. – 240 c.

4. Кириллов, В.В. Введение в реляционные базы данных. Введение в реляционные базы данных / В.В. Кириллов, Г.Ю. Громов. – СПб.: БХВ-Петербург, 2022. – 464 c.