**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ (ФИЛИАЛ) СПбГУТ**

**(АКТ (ф) СПбГУТ)**

**Отчеты по лабораторным и практическим работам**

**по ОП.08 ОПБД**

Студент: Чугин Е.В.

Группа: ИСПП-21

Преподаватель: Маломан Ю.С.

Архангельск 2024

# Практическая работа №1

# Изучение правил преобразования ER-модели в реляционную модель данных

1. **Цель работы**

1.1 Изучить правила преобразования ER-модели в реляционную модель данных с учетом вида связи;

1.2 Изучить процесс создания концептуальной модели данных и ее преобразования в логическую модель данных;

1.3 Научиться применять графические редакторы в процессе проектирования БД.

1. **Контрольные вопросы**

2.1 Что такое «сущность»?

Ответ: Сущность — это объект или понятие, которое может быть представлено в базе данных и о котором необходимо хранить информацию. Например, это может быть человек, книга или заказ.

2.2 Что такое «атрибут»?

Ответ: Атрибут — это характеристика или свойство сущности. Например, для сущности "Человек" атрибутами могут быть имя, дата рождения и адрес.

2.3 Что такое «ключевое поле»?

Ответ: Ключевое поле — это атрибут или набор атрибутов, которые уникально идентифицируют каждую запись в таблице. Ключевое поле позволяет отличать одну запись от другой.

2.4 Каково назначение первичных и внешних ключей?

Ответ: Первичный ключ — это уникальный идентификатор записи в таблице. Внешний ключ — это атрибут, который ссылается на первичный ключ другой таблицы, создавая связь между записями в разных таблицах.

2.5 Что такое «связь»?

Ответ: Связь — это ассоциация между двумя или более сущностями. Связи описывают, как сущности взаимодействуют друг с другом в базе данных.

2.6 Какие виды связей между сущностями существуют?

Ответ: Существуют три основных вида связей: 1. Один к одному (1:1), 2. Один ко многим (1), 3. Многие ко многим (M)

2.7 Какие элементы входят в ER-диаграммы?

Ответ: Основные элементы ER-диаграмм: 1. Сущности 2. Атрибуты

3. Связи 4. Множества связей (кардинальности)

1. **Вывод**

В ходе лабораторной работы были изучены правила преобразования ER-модели в реляционную модель данных с учетом вида связи, был изучен процесс создания концептуальной модели данных и ее преобразования в логическую модель данных, научились применять графические редакторы в процессе проектирования БД.

**Практической работе №2**

**Изучение процесса нормализации отношений**

1. **Цель работы**
   1. Изучить процесс приведения отношений от ненормализованного вида к четвертой нормальной форме;

1.2 Изучить процесс декомпозии отношений.

1. **Контрольные вопросы**

2.1 Что называется первичным ключом отношения?

Ответ: это один или несколько атрибутов (полей) в таблице базы данных, которые однозначно идентифицируют каждую запись в этой таблице. Первичный ключ должен быть уникальным для каждой строки и не должен содержать значение NULL. Он используется для обеспечения целостности данных и быстрого поиска записей.

2.2 Что называется внешним ключом отношения?

Ответ: это атрибут или набор атрибутов в одной таблице, который ссылается на первичный ключ другой таблицы. Внешние ключи используются для установления и поддержания ссылочной целостности между таблицами, позволяя связать данные из связанных таблиц.

2.3 В чем заключается процесс нормализации отношений?

Ответ: это процесс организации данных в базе данных для минимизации избыточности и устранения аномалий обновления, вставки и удаления. Процесс нормализации включает разбиение таблиц на более мелкие и более управляемые части и установление между ними отношений. Основные этапы нормализации включают приведение таблиц к первой нормальной форме (1НФ), второй нормальной форме (2НФ), третьей нормальной форме (3НФ) и, при необходимости, более высоким нормальным формам.

2.4 В каком случае атрибут А функционально зависит от атрибута В?

Ответ: Атрибут A функционально зависит от атрибута B, если для каждого значения B существует не более одного значения A. Иными словами, знание значения B позволяет однозначно определить значение A.

2.5 В каком случае атрибут А транзитивно зависит от атрибута В?

Ответ: Атрибут A транзитивно зависит от атрибута B, если существует такой атрибут C при этом A не является непосредственной функциональной зависимостью от B. Транзитивная зависимость представляет собой косвенную функциональную зависимость через промежуточный атрибут.

2.6 Каким требованиям должно отвечать отношение, находящееся в 1НФ? Ответ: Отношение находится в первой нормальной форме (1НФ), если оно удовлетворяет следующим требованиям:

1: Все атрибуты содержат только атомарные (неделимые) значения.

2: Каждое значение в колонке (атрибуте) является скалярным (то есть одно значение на ячейку).

3: Все строки (записи) уникальны, что подразумевает наличие потенциального первичного ключа.

2.7 Каким требованиям должно отвечать отношение, находящееся во 2НФ?

Ответ: Отношение находится во второй нормальной форме (2НФ), если оно удовлетворяет следующим требованиям:

1: Оно уже находится в первой нормальной форме (1НФ).

2: Все атрибуты, не входящие в состав первичного ключа, полностью функционально зависят от всего первичного ключа, а не от его части (для отношений с составным первичным ключом).

2.8 Каким требованиям должно отвечать отношение, находящееся в 3НФ?

Ответ: Отношение находится в третьей нормальной форме (3НФ), если оно удовлетворяет следующим требованиям:

1:Оно уже находится во второй нормальной форме (2НФ).

2: Все атрибуты, не входящие в состав первичного ключа, не зависят транзитивно от первичного ключа (то есть нет транзитивных зависимостей, когда один неключевой атрибут зависит от другого неключевого атрибута).

1. **Вывод**

В ходе лабораторной работы были изучены правила преобразования ER-модели в реляционную модель данных с учетом вида связи, был изучен процесс создания концептуальной модели данных и ее преобразования в логическую модель данных, научились применять графические редакторы в процессе проектирования БД.

**Практической работе №3  
Разработка физической модели данных**

**1 Цель работы**

* 1. Изучить процесс создания физической модели данных.

**2 Контрольные вопросы**

2.1 Что такое «физическая модель данных»?

Ответ: Физ. Модель Данных описывает, как данные будут храниться в системе управления базами данных (СУБД). Она определяет структуру таблиц, индексы, ограничения, типы данных и другие физические аспекты, необходимые для эффективного хранения, поиска и управления данными. Физическая модель данных является реализацией логической модели данных в конкретной СУБД.

2.2 Какие типы данных используются для хранения строковых значений?

Ответ:

1) CHAR(n): Хранит строки фиксированной длины nnn.

2) VARCHAR(n): Хранит строки переменной длины до nnn символов.

3) TEXT: Хранит большие объемы строковых данных, длина которых может варьироваться.

4) CLOB: (Character Large Object) Хранит очень большие текстовые данные.

2.3 Какие типы данных используются для хранения вещественных значений?

Ответ:

1) INT или INTEGER: Хранит целые числа.

2) SMALLINT: Хранит маленькие целые числа.

3) BIGINT: Хранит большие целые числа.

4) TINYINT: Хранит очень маленькие целые числа.

2.4 Какие типы данных используются для хранения вещественных значений?

Ответ:

FLOAT: Хранит числа с плавающей точкой.

REAL: Хранит числа с плавающей точкой меньшей точности.

DOUBLE или DOUBLE PRECISION: Хранит числа с плавающей точкой большей точности.

DECIMAL(p, s) или NUMERIC(p, s): Хранит точные числовые значения с фиксированной запятой, где ppp — это общее количество цифр, а sss — количество цифр после запятой.

2.5 Какие типы данных используются для хранения логических значений?

Ответ: Только BOOL: Хранит значения TRUE или FALSE.

2.6 Какие типы данных используются для хранения даты и/или времени?

Ответ:

DATE: Хранит дату (год, месяц, день).

TIME: Хранит время (часы, минуты, секунды).

DATETIME или TIMESTAMP: Хранит дату и время.

YEAR: Хранит только год.

INTERVAL: Хранит временные интервалы.

2.7 Что такое «ограничение целостности»?

Ответ: - это правило, которое накладывается на данные в базе данных для обеспечения их корректности и консистентности. Эти правила предотвращают некорректные изменения и обеспечивают, что данные в базе данных соответствуют определенным требованиям и бизнес-правилам.

2.8 Какие виды ограничений целостности существуют?

Ответ:

Первичный ключ (PRIMARY KEY): Гарантирует уникальность каждой записи в таблице и отсутствие значений NULL.

Уникальность (UNIQUE): Обеспечивает уникальность значений в одном или нескольких столбцах.

Не NULL (NOT NULL): Гарантирует, что значения в столбце не могут быть NULL.

Внешний ключ (FOREIGN KEY): Обеспечивает ссылочную целостность между таблицами, указывая, что значения в одном столбце должны соответствовать значениям первичного ключа в другой таблице.

Проверка (CHECK): Устанавливает условие, которому должны удовлетворять значения в столбце или группе столбцов.

Ограничение по домену (DOMAIN): Определяет набор допустимых значений для атрибута.

**3 Вывод**

В ходе лабораторной работы был изучен процесс создания физической модели данных.

**Практической работе №4  
Разработка ER-модели БД в Microsoft Visio**

**1 Цель работы**

1.1 Изучить процесс создания концептуальной, логической и физической модели данных в MS Visio.

**2. Контрольные вопросы:**

8.1 Что такое «MS Visio»?

Ответ: MS Visio — это программное обеспечение от компании Microsoft, предназначенное для создания диаграмм и схем. Оно широко используется для проектирования и документирования различных систем, включая базы данных, сетевые схемы, организационные структуры и другие визуальные представления информации.

8.2 В каких нотациях можно спроектировать схему БД в MS Visio?

Ответ:

Chen (Чена): Традиционная нотация ER-диаграмм, где сущности и отношения изображаются прямоугольниками и ромбами соответственно.

Crow's Foot (Воронья лапа): Нотация для ER-диаграмм, в которой связи изображаются с помощью символов, похожих на вороньи лапы, что позволяет легко визуализировать связи 1:1, 1:М и М:М.

IDEF1X: Нотация, используемая в моделировании данных для проектирования информационных систем и баз данных.

8.3 Какие элементы доступны для настройки схемы БД в MS Visio?

Ответ:

1) Сущности (Entities): Представляют таблицы в базе данных.

2) Атрибуты (Attributes): Представляют столбцы в таблицах.

3) Связи (Relationships): Определяют отношения между сущностями.

4) Метки связи (Relationship Labels): Обозначают тип связи и кардинальность.

5) Ключи (Keys): Указывают первичные и внешние ключи.

6) Ограничения и индексы (Constraints and Indexes): Используются для дополнительной настройки схемы базы данных.

8.4 Как задать в MS Visio первичный ключ?

Ответ:

Создайте или выберите сущность (таблицу).

Добавьте необходимые атрибуты (столбцы) в сущность.

Щелкните правой кнопкой мыши на атрибуте, который должен быть первичным ключом.

Выберите опцию Primary Key из контекстного меню. Атрибут будет отмечен как первичный ключ (обычно значком ключа).

8.5 Как задать в MS Visio внешний ключ?

Ответ:

1) Создайте или выберите сущности, между которыми должна быть установлена связь.

2) Выберите инструмент Relationship (Связь) на панели инструментов.

3) Соедините атрибут внешнего ключа одной сущности с атрибутом первичного ключа другой сущности, создавая таким образом связь.

4) Настройте свойства связи, чтобы определить её как внешний ключ.

8.6 Как в MS Visio изменить тип связи на 1:1, М:М, 1:М?

Ответ: Щелкните правой кнопкой мыши на связи, которую хотите изменить.

Выберите Database Properties (Свойства базы данных) или Database (База данных) из контекстного меню.

Перейдите на вкладку Relationships (Связи) и настройте кардинальность (1:1, 1:М, М:М) в соответствующих полях.

Сохраните изменения.

8.7 Как в MS Visio изменить связь на идентифицирующую и не

идентифицирующую?

Ответ: Щелкните правой кнопкой мыши на связи, которую хотите изменить.

Выберите Database Properties (Свойства базы данных) или Database (База данных) из контекстного меню.

Перейдите на вкладку Relationships (Связи).

Найдите опцию Identifying Relationship (Идентифицирующая связь) или Non-identifying Relationship (Неидентифицирующая связь).

Установите или снимите флажок, чтобы изменить тип связи.

Сохраните изменения.

**3 Вывод**

В ходе лабораторной работы был изучен процесс создания концептуальной, логической и физической модели данных в MS Visio.

**Практической работе №5  
Разработка ER-модели БД в MySQL Workbench**

**1 Цель работы**

1.1 Научиться применять MySQL Workbench в процессе создания схем моделей БД;

1.2 Научиться представлять логическую модель данных согласно нотациям ERD и IDEF1X.

**2 Контрольные вопросы**

2.1. Что такое «сущность»

Ответ: Сущность — это объект или понятие, которое может быть представлено в базе данных и о котором необходимо хранить информацию. Например, это может быть человек, книга или заказ.

2.2. Что такое «атрибут»

Ответ: Атрибут — это характеристика или свойство сущности. Например, для сущности "Человек" атрибутами могут быть имя, дата рождения и адрес.

2.3. Что такое «ключевое поле»?

Ответ: Ключевое поле — это атрибут или набор атрибутов, которые уникально идентифицируют каждую запись в таблице. Ключевое поле позволяет отличать одну запись от другой.

2.4. Каково назначение первичных и внешних ключей?

Ответ: Первичный ключ — это уникальный идентификатор записи в таблице. Внешний ключ — это атрибут, который ссылается на первичный ключ другой таблицы, создавая связь между записями в разных таблицах.

2.5. Что такое «связь»?

Ответ: Связь — это ассоциация между двумя или более сущностями. Связи описывают, как сущности взаимодействуют друг с другом в базе данных.

2.6. Какие виды связей между сущностями существуют?

Ответ: Существуют три основных вида связей: 1. Один к одному (1:1), 2. Один ко многим (1), 3. Многие ко многим (M)

2.7. Какие элементы входят в ER-диаграммы?

Ответ: Основные элементы ER-диаграмм:

Сущности 2.

Атрибуты 3.

Связи 4.

Множества связей (кардинальности)

2.8. Для чего применяются ER-диаграммы?

Ответ: ER-диаграммы используются для визуального представления структуры базы данных, её сущностей и связей между ними. Они помогают проектировать базы данных и понимать её структуру до её фактического создания.

**3 Вывод**

В ходе лабораторной работы научились применять MySQL Workbench в процессе создания схем моделей БД, научились представлять логическую модель данных согласно нотациям ERD и IDEF1X.

**Практической работе №6  
Создание БД MySQL**

**1 Цель работы**

1.1 Научиться создавать физическую модель данных БД MySQL, используя CASE-средство MySQL Workbench.

1.2 Научиться создавать БД MySQL, используя CASE-средство MySQL

**2 Контрольные вопросы**

2.1 Для чего применяется MySQL Workbench?

Ответ: Он используется для визуального создания, редактирования и управления базами данных, а также для выполнения SQL-запросов и миграции данных.

2.2 Как создать EER-модель в MySQL Workbench?

Ответ: Чтобы создать EER-модель в MySQL Workbench:

Откройте MySQL Workbench.

Перейдите в меню File и выберите New Model (Новая модель).

В панели Model Overview (Обзор модели) щелкните правой кнопкой мыши и выберите Add Diagram (Добавить диаграмму). Это создаст новую EER-диаграмму.

В появившемся окне вы можете добавлять таблицы, связи и другие элементы, используя инструменты на панели инструментов.

2.3 Из каких элементов состоит EER-модель?

Ответ:

Сущности (Entities): Таблицы в базе данных.

Атрибуты (Attributes): Столбцы в таблицах.

Связи (Relationships): Определяют связи между сущностями (таблицами).

Первичные ключи (Primary Keys): Атрибуты, однозначно идентифицирующие записи в таблицах.

Внешние ключи (Foreign Keys): Атрибуты, которые создают связи между таблицами.

Индексы (Indexes): Структуры данных, которые ускоряют выполнение запросов.

Ограничения (Constraints): Правила, обеспечивающие целостность данных (например, уникальность, NOT NULL).

Виды (Views): Виртуальные таблицы, созданные на основе запросов.

Хранимые процедуры (Stored Procedures) и функции (Functions): Программные компоненты для выполнения на сервере.

2.4 Что такое «прямой инжиниринг»?

Ответ: Прямой инжиниринг (Forward Engineering) — это процесс преобразования модели данных в реальную базу данных. Это включает создание SQL-скриптов для создания схемы базы данных, включая таблицы, индексы, связи и другие объекты, на основе разработанной модели.

2.5 Как провести прямой инжиниринг, используя MySQL Workbench?

Ответ: 1)Создайте или откройте существующую модель данных (EER-модель).

2) Перейдите в меню Database и выберите Forward Engineer.

3) В открывшемся мастере Forward Engineer следуйте шагам:

4) Укажите целевую схему и настройте параметры генерации.

5) Выберите объекты, которые нужно включить в процесс (таблицы, виды, процедуры и т.д.).

6) Сгенерируйте SQL-скрипт.

7) Вы можете просмотреть и отредактировать сгенерированный SQL-скрипт перед его выполнением.

8) Нажмите Execute (Выполнить), чтобы создать базу данных на сервере MySQL на основе вашей модели.

**3 Вывод**

В ходе лабораторной работы я 1) Научиться создавать физическую модель данных БД MySQL, используя CASE-средство MySQL Workbench.

2) Научиться создавать БД MySQL, используя CASE-средство MySQL

**Практической работе №7  
Изучение способов обеспечения целостности данных в СУБД MySQL**

**1 Цель работы**

1.1 Изучить способы обеспечения целостности данных в БД MySQL Server, используя CASE-средство MySQL Workbench.

**2.** **Контрольные вопросы**

2.1 Как в MySQL Workbench указать автоинкрементные значения?

Ответ:

1) Откройте MySQL Workbench и выберите вашу EER-диаграмму.

2) Дважды щелкните на таблицу, для которой вы хотите установить автоинкремент.

3) В открывшемся окне Table Editor выберите столбец, который должен быть автоинкрементным (обычно это первичный ключ).

4) Внизу найдите вкладку Column Details.

5) В разделе Column Details установите флажок AI (Auto Increment) для выбранного столбца.

2.2 Как в MySQL Workbench указать значения по умолчанию?

Ответ: 1) Откройте MySQL Workbench и выберите вашу EER-диаграмму.

2) Дважды щелкните на таблицу, для которой вы хотите установить значения по умолчанию.

3) В окне Table Editor выберите нужный столбец.

4)Внизу найдите вкладку Column Details.

5)В поле Default введите значение по умолчанию, которое вы хотите установить для этого столбца.

2.3 Как в MySQL Workbench указать обязательные для заполнения столбцы?

Ответ: 1) Откройте MySQL Workbench и выберите вашу EER-диаграмму.

2) Дважды щелкните на таблицу, для которой вы хотите установить обязательные столбцы.

3) В окне Table Editor выберите нужный столбец.

4) Внизу найдите вкладку Column Details.

5) Установите флажок NN (Not Null), чтобы сделать этот столбец обязательным для заполнения.

2.4 Как создаются индексы в MySQL Workbench?

Ответ: 1) Откройте MySQL Workbench и выберите вашу EER-диаграмму.

2) Дважды щелкните на таблицу, для которой вы хотите создать индексы.

3) В окне Table Editor перейдите на вкладку Indexes.

4) Нажмите на значок + (Add Index) для добавления нового индекса.

5) Укажите имя индекса, его тип (UNIQUE, FULLTEXT, SPATIAL и т.д.) и выберите столбцы, которые будут включены в индекс.

2.5 Какие виды индексов можно создать в таблицах MySQL?

Ответ:

1) PRIMARY KEY: Индекс первичного ключа, обеспечивающий уникальность и отсутствие NULL.

2) UNIQUE: Уникальный индекс, обеспечивающий уникальность значений в столбце или наборе столбцов.

3) INDEX: Обычный индекс, который ускоряет выполнение запросов.

4) FULLTEXT: Полнотекстовый индекс для быстрого поиска текстовых данных.

5) SPATIAL: Пространственный индекс для геометрических данных (например, для данных типа POINT, LINESTRING, POLYGON).

2.6 Как в MySQL Workbench указать настройки внешних ключей?

Ответ: 1) Откройте MySQL Workbench и выберите вашу EER-диаграмму.

2) Дважды щелкните на таблицу, в которой вы хотите установить внешний ключ.

3) В окне Table Editor перейдите на вкладку Foreign Keys.

4) Нажмите на значок + (Add Foreign Key) для добавления нового внешнего ключа.

5) Укажите имя внешнего ключа, выберите столбец, который будет внешним ключом, и укажите ссылочную таблицу и её столбец.

6) Установите параметры на вкладках On Delete и On Update для настройки поведения внешнего ключа при удалении или обновлении записей.

2.7 Как в MySQL Workbench указать уникальность значений в одном столбце?

Ответ: 1) Откройте MySQL Workbench и выберите вашу EER-диаграмму.

2) Дважды щелкните на таблицу, в которой вы хотите установить уникальность столбца.

3) В окне Table Editor выберите нужный столбец.

4) Внизу найдите вкладку Indexes.

5) Нажмите на значок + (Add Index) для добавления нового индекса.

6) В поле Index Type выберите UNIQUE и добавьте столбец, который должен быть уникальным.

2.8 Как в MySQL Workbench указать уникальность значений в наборе столбцов?

Ответ: 1) Откройте MySQL Workbench и выберите вашу EER-диаграмму.

2) Дважды щелкните на таблицу, в которой вы хотите установить уникальность набора столбцов.

3) В окне Table Editor перейдите на вкладку Indexes.

1) Нажмите на значок + (Add Index) для добавления нового индекса.

2) В поле Index Type выберите UNIQUE.

3) Добавьте столбцы, которые должны быть частью уникального индекса, перемещая их из доступных столбцов в секцию Index Columns.

**3.Вывод:**

В ходе лабораторной работы я изучил способы обеспечения целостности данных в БД MySQL Server, используя CASE-средство MySQL Workbench.

**Практической работе №8  
Создание БД Microsoft SQL Server**

**1 Цель работы**

1.1 Изучить процесс создания таблиц и связей между ними в СУБД Microsoft SQL Server (MSSQL).

1.2 Научиться работать в среде SQL Server Management Studio (SSMS).

**2 Контрольные вопросы**

2.1 Что такое SQL Server Management Studio?

Ответ: это интегрированная среда для управления и администрирования баз данных Microsoft SQL Server. SSMS предоставляет графический интерфейс для выполнения различных задач, таких как написание и выполнение SQL-запросов, управление объектами базы данных, настройка серверов, создание и изменение схем данных, а также мониторинг производительности.

2.2 Какие виды авторизации поддерживаются в MS SQL Server?

Ответ: MS SQL Server поддерживает два основных вида авторизации:

Windows Authentication: Авторизация с использованием учетных данных Windows. Это позволяет пользователям подключаться к серверу, используя свои учетные записи Windows, и обеспечивает интеграцию с Active Directory.

SQL Server Authentication: Авторизация с использованием логинов и паролей, созданных непосредственно в SQL Server. Это позволяет пользователям подключаться к серверу, используя отдельные учетные данные, независимо от учетных записей Windows.

2.3 Как создать диаграмму в SQL Server Management Studio?

Ответ: 1) Откройте SQL Server Management Studio и подключитесь к нужной базе данных.

2) В Object Explorer разверните узел базы данных, с которой вы работаете.

3) Разверните папку Tables (Таблицы).

4) Щелкните правой кнопкой мыши на папке Tables и выберите New Table (Новая таблица).

5) В открывшемся окне Table Designer укажите столбцы таблицы, их типы данных и ограничения.

6) Сохраните таблицу, нажав Save Table или выбрав File -> Save Table.

2.4 Как добавить таблицы в SQL Server Management Studio?

Ответ: 1) Откройте SQL Server Management Studio и подключитесь к нужной базе данных.

2) В Object Explorer разверните узел базы данных, с которой вы работаете.

3) Разверните папку Tables (Таблицы).

4) Щелкните правой кнопкой мыши на папке Tables и выберите New Table (Новая таблица).

5) В открывшемся окне Table Designer укажите столбцы таблицы, их типы данных и ограничения.

6) Сохраните таблицу, нажав Save Table или выбрав File -> Save Table.

2.5 Как создать связи между таблицами в SQL Server Management Studio?

Ответ: 1) Создайте или откройте диаграмму базы данных, как описано в пункте 2.3.

2) На диаграмме выберите таблицы, между которыми вы хотите установить связь.

3) Щелкните правой кнопкой мыши на таблице, содержащей внешний ключ, и выберите Relationships (Связи).

4) В окне Foreign Key Relationships нажмите Add (Добавить).

5) В секции Tables and Columns Specification (Спецификация таблиц и столбцов) выберите основную (Primary) и внешнюю (Foreign) таблицы и столбцы для создания связи.

6) Нажмите OK для сохранения связи.

**3 Вывод**

В ходе лабораторной работы был изучен процесс создания таблиц и связей между ними в СУБД Microsoft SQL Server (MSSQL), научились работать в среде SQL Server Management Studio (SSMS).

**Практической работе №9  
Изучение способов обеспечения целостности данных в СУБД Microsoft SQL Server**

**1 Цель работы**

1.1 Изучить способы обеспечения целостности данных в БД Microsoft SQL Server,используя SQL Server Management Studio (SSMS).

**2 Контрольные вопросы**

2.1 Как в SSMS указать автоинкрементные значения?

Ответ: 1) Откройте SSMS и подключитесь к нужной базе данных.

2) В Object Explorer разверните узел базы данных, в которой вы хотите создать или изменить таблицу.

3) Разверните папку Tables и выберите таблицу, которую хотите изменить, или создайте новую таблицу.

4) Щелкните правой кнопкой мыши на таблице и выберите Design (Конструктор).

5) В конструкторе таблиц выберите столбец, который должен быть автоинкрементным.

6) Внизу, в Column Properties (Свойства столбца), найдите свойство Identity Specification.

7) Раскройте это свойство и установите (Is Identity) в значение Yes.

8) Установите Identity Seed (начальное значение) и Identity Increment (шаг приращения) в нужные значения.

9) Сохраните изменения, нажав Save Table или выбрав File -> Save Table.

2.2 Как в SSMS указать значения по умолчанию?

Ответ: 1) Откройте SSMS и подключитесь к нужной базе данных.

2) В Object Explorer разверните узел базы данных и выберите таблицу, которую хотите изменить, или создайте новую таблицу.

3) Щелкните правой кнопкой мыши на таблице и выберите Design (Конструктор).

4) В конструкторе таблиц выберите столбец, для которого хотите задать значение по умолчанию.

5) Внизу, в Column Properties (Свойства столбца), найдите свойство Default Value or Binding (Значение по умолчанию или привязка).

6) Введите значение по умолчанию, которое хотите установить для этого столбца.

7) Сохраните изменения, нажав Save Table или выбрав File -> Save Table.

2.3 Как в SSMS указать необязательные для заполнения столбцы?

Ответ: 1) Откройте SSMS и подключитесь к нужной базе данных.

2) В Object Explorer разверните узел базы данных и выберите таблицу, которую хотите изменить, или создайте новую таблицу.

3) Щелкните правой кнопкой мыши на таблице и выберите Design (Конструктор).

4) В конструкторе таблиц выберите столбец, который должен быть необязательным.

5) В верхней части конструктора таблиц найдите свойство Allow Nulls и установите его флажок.

6) Сохраните изменения, нажав Save Table или выбрав File -> Save Table.

2.4 Как в SSMS указать настройки внешних ключей?

Ответ: 1) Откройте SSMS и подключитесь к нужной базе данных.

2) В Object Explorer разверните узел базы данных и выберите таблицу, в которой вы хотите создать внешний ключ.

3) Щелкните правой кнопкой мыши на таблице и выберите Design (Конструктор).

4) В конструкторе таблиц щелкните правой кнопкой мыши на таблице и выберите Relationships (Связи).

5) В окне Foreign Key Relationships нажмите Add (Добавить).

6) В секции Tables and Columns Specification (Спецификация таблиц и столбцов) нажмите на кнопку с многоточием (...).

7) Выберите основную (Primary) таблицу и столбец, а также внешний (Foreign) столбец, который должен ссылаться на первичный ключ.

8) Настройте поведение внешнего ключа в секциях Insert and Update Specification (Спецификация вставки и обновления), задав действия при удалении (ON DELETE) и обновлении (ON UPDATE).

9) Нажмите OK для сохранения настроек внешнего ключа.

10) Сохраните изменения, нажав Save Table или выбрав File -> Save Table.

2.5 Как в SSMS указать уникальность значений?

Ответ: 1) Откройте SSMS и подключитесь к нужной базе данных.

2) В Object Explorer разверните узел базы данных и выберите таблицу, которую хотите изменить, или создайте новую таблицу.

3) Щелкните правой кнопкой мыши на таблице и выберите Design (Конструктор).

4) В конструкторе таблиц выберите столбец или набор столбцов, для которых хотите задать уникальность.

5) Щелкните правой кнопкой мыши на выбранном столбце (или выделите несколько столбцов, удерживая клавишу Ctrl) и выберите Indexes/Keys (Индексы/Ключи).

6) В окне Indexes/Keys нажмите Add (Добавить).

7) В поле Index Type выберите Unique Key (Уникальный ключ).

8) В секции Columns нажмите на кнопку с многоточием (...) и выберите столбец или столбцы, которые должны быть уникальными.

9) Сохраните изменения, нажав OK и затем Save Table или выбрав File   
-> Save Table.

**3 Вывод**

В ходе лабораторной работы был изучен способы обеспечения целостности данных в БД Microsoft SQL Server, используя SQL Server Management Studio (SSMS).

**Практической работе №10  
Создание SQL-запросов на модификацию схемы БД**

**1 Цель работы**

1.1 Изучить процесс разработки DDL-команд в БД Microsoft SQL Server, используя SQL Server Management Studio (SSMS).

**2 Контрольные вопросы**

2.1 Какое предложение инструкции CREATE TABLE позволяет определить первичный ключ?

Ответ: В инструкции CREATE TABLE для определения первичного ключа используется предложение PRIMARY KEY.

CREATE TABLE TableName (

Column1 INT,

Column2 VARCHAR(255),

Column3 INT,

PRIMARY KEY (Column1)

);

2.2 Какое предложение инструкции CREATE TABLE позволяет определить внешний ключ?

Ответ: В инструкции CREATE TABLE для определения внешнего ключа используется предложение FOREIGN KEY.

CREATE TABLE ChildTable (

Column1 INT,

Column2 VARCHAR(255),

ParentID INT,

FOREIGN KEY (ParentID) REFERENCES ParentTable(ParentID)

);

2.3 Как в инструкции CREATE TABLE задать ограничения для значений столбца?

Ответ: В инструкции CREATE TABLE можно задать ограничения для значений столбца с помощью ключевых слов CONSTRAINT, NOT NULL, UNIQUE, CHECK и DEFAULT. Примеры:

NOT NULL: указывает, что столбец не может содержать NULL значения.

UNIQUE: указывает, что значения в столбце должны быть уникальными.

CHECK: позволяет задать логическое выражение, которому должны соответствовать значения столбца.

DEFAULT: задает значение по умолчанию для столбца.

2.4 Каково назначение предложения CONSTRAINT?

Ответ: CONSTRAINT используется для именования и определения ограничений в таблице. Оно позволяет явно задать ограничения для столбцов и таблиц, таких как первичные и внешние ключи, уникальность и проверки.

2.5 В каких случаях нельзя удалить столбец, таблицу?

Ответ: Столбец или таблицу нельзя удалить в следующих случаях:

Столбец:

Если столбец является частью первичного ключа, и другие таблицы ссылаются на него через внешний ключ.

Если столбец участвует в ограничениях (например, уникальности, проверки или внешнего ключа).

Таблица:

Если существуют другие таблицы, которые ссылаются на эту таблицу через внешние ключи.

Если таблица используется в представлениях, хранимых процедурах или триггерах.

2.6 Для чего предназначена команда ALTER TABLE?

Ответ: Команда ALTER TABLE предназначена для изменения структуры существующей таблицы. Она позволяет:

Добавлять, изменять или удалять столбцы.

Добавлять или удалять ограничения.

Переименовывать таблицу или столбцы.

Изменять типы данных столбцов.

2.7 Для чего предназначена команда DROP TABLE?

Ответ: Команда DROP TABLE предназначена для удаления существующей таблицы из базы данных. Это действие необратимо и приводит к удалению всех данных, связанных с этой таблицей, а также всех индексов, триггеров и ограничений, связанных с таблицей.

**3 Вывод**

В ходе лабораторной работы был изучен процесс разработки DDL-команд в БД Microsoft SQL Server, используя SQL Server Management Studio (SSMS).

**Практической работе №11  
Создание SQL-запросов на модификацию данных**

**1 Цель работы**

1.1 Изучить процесс создания таблиц и связей между ними в Microsoft SQL Server (MSSQL).

1.2 Научиться работать в среде SQL Server Management Studio (SSMS).

**2. Контрольные вопросы**

2.1 Каковы причины ошибок, связанных с выполнением DML-команд на

модификацию данных?

Ответ: Ошибки при выполнении DML-команд (Data Manipulation Language), таких как INSERT, UPDATE и DELETE, могут возникать по нескольким причинам:

Нарушение ограничений целостности:

Несоответствие типов данных:

Попытка вставить NULL в обязательное поле:

Проблемы с синтаксисом SQL:

Права доступа:

Блокировки:

Переполнение хранилища:

2.2 После какого ключевого слова в запросе на обновление записывается

обновляемое поле?

В запросе на обновление (UPDATE) обновляемое поле записывается после ключевого слова SET.

UPDATE TableName

SET ColumnName = NewValue

WHERE Condition;

2.3 Для чего используется команда INSERT?

Команда INSERT используется для добавления новых записей в таблицу базы данных.

INSERT INTO TableName (Column1, Column2, Column3) VALUES (Value1, Value2, Value3);

2.4 Для чего используется команда UPDATE?

Команда UPDATE используется для изменения существующих записей в таблице базы данных.

UPDATE TableName

SET Column1 = NewValue1, Column2 = NewValue2

WHERE Condition;

2.5 Для чего используется команда DELETE?

Команда DELETE используется для удаления записей из таблицы базы данных.

DELETE FROM TableName

WHERE Condition;

**3 Вывод**

В ходе лабораторной работы был изучен процесс создания таблиц и связей между ними в Microsoft SQL Server (MSSQL), научились работать в среде SQL Server Management Studio (SSMS)

**Практической работе №12  
Создание однотабличных SQL-запросов на выборку данных**

**1 Цель работы**

1.1 Изучить процесс создания однотабличных SQL-запросов на выборку.

**2 Контрольные вопросы**

2.1 Какова структура инструкции SELECT?

Ответ: Инструкция SELECT используется для выборки данных из базы данных и имеет следующую общую структуру:

SELECT columns

FROM table

[WHERE condition]

[GROUP BY columns]

[HAVING condition]

[ORDER BY columns [ASC|DESC]]

[LIMIT number\_of\_rows];

2.2 Что указывается после ключевого слова SELECT?

Ответ: После ключевого слова SELECT указываются столбцы, которые должны быть выбраны из таблицы. Это могут быть конкретные столбцы, выражения, функции агрегирования или символ \* для выборки всех столбцов.

2.3 Что указывается после ключевого слова FROM?

Ответ: После ключевого слова FROM указывается имя таблицы или нескольких таблиц, из которых будут выбраны данные. Также здесь могут указываться объединения (JOIN) для выборки данных из нескольких таблиц.

2.4 Что указывается после ключевого слова GROUP BY?

Ответ: После ключевого слова GROUP BY указываются столбцы, по которым будут сгруппированы результаты запроса. Это используется для агрегирования данных.

2.5 Что указывается после ключевого слова ORDER BY?

Ответ: После ключевого слова ORDER BY указываются столбцы, по которым будет выполнена сортировка результатов запроса. Можно указать направление сортировки: ASC (по возрастанию) или DESC (по убыванию).

**3 Вывод**

В ходе лабораторной работы был изучен процесс создания однотабличных SQL-запросов на выборку.

**Практической работе №13  
Создание однотабличных SQL-запросов на выборку с фильтрацией данных**

**1 Цель работы**

1.1 Изучить процесс фильтрации данных в SQL-запросах на выборку.

**2 Контрольные вопросы**

2.1 Что указывается после ключевого слова WHERE?

Ответ: После ключевого слова WHERE указывается условие, которое фильтрует строки возвращаемого результата на основе заданных критериев. Это условие может быть логическим выражением, использующим операторы сравнения (например, =, !=, <, >, <=, >=) и логические операторы (например, AND, OR, NOT).

2.2 Что указывается после ключевого слова HAVING?

Ответ: После ключевого слова HAVING указывается условие, которое фильтрует группы строк, созданных с помощью оператора GROUP BY в запросе SELECT. Это условие используется для применения фильтрации к агрегированным данным.

2.3 В чем отличие между разделами WHERE и HAVING?

Ответ: Основное отличие между WHERE и HAVING заключается в том, что:

WHERE применяется к отдельным строкам, фильтруя результаты запроса до их группировки, а HAVING применяется к группам строк, созданным с помощью оператора GROUP BY.

WHERE используется в запросах без оператора GROUP BY,

HAVING - в запросах с оператором GROUP BY.

WHERE фильтрует строки перед группировкой и агрегированием данных, а HAVING фильтрует группы после выполнения агрегации.

2.4 Что указывается после ключевого слова IN?

Ответ: После ключевого слова IN указывается список значений или подзапрос, возвращающий значения, с которыми сравниваются значения столбца в условии. Оператор IN используется для проверки, входит ли значение столбца в заданный список или результат подзапроса.

2.5 Что указывается после ключевого слова LIKE?

Ответ: После ключевого слова LIKE указывается шаблон символов, который используется для сопоставления строковых значений столбца. Этот шаблон может включать специальные символы:

% (процент) - для сопоставления любого количества символов.

\_ (подчеркивание) - для сопоставления одного символа.

2.6 Что указывается после ключевого слова BETWEEN?

Ответ: После ключевого слова BETWEEN указываются два значения, между которыми проверяется значение столбца. Оператор BETWEEN используется для проверки, входит ли значение столбца в заданный диапазон значений, включая начальное и конечное значения.

**3 Вывод**

В ходе лабораторной работы был изучен процесс фильтрации данных в SQL-запросах на выборку.

**Практической работе №14  
Создание многотабличных SQL-запросов на выборку данных**

**1 Цель работы**

1.1 Изучить процесс создавать многотабличных SQL-запросов на выборку.

**2. Контрольные вопросы**

2.1 Что указывается после ключевого слова FROM?

Ответ: После ключевого слова FROM указывается имя таблицы или таблиц, из которых будут выбраны данные для запроса. Это может быть одна таблица, или несколько таблиц, разделенных запятыми, если требуется выполнить запрос к нескольким таблицам.

2.2 Каков синтаксис инструкции объединения двух таблиц?

Ответ: Синтаксис инструкции объединения двух таблиц включает ключевые слова JOIN или INNER JOIN, а также условие объединения с помощью ключевого слова ON.

2.3 Какие виды объединения таблиц существуют?

Ответ: Существуют следующие виды объединения таблиц:

INNER JOIN: Возвращает строки, которые имеют соответствующие значения в обеих таблицах.

LEFT JOIN (или LEFT OUTER JOIN): Возвращает все строки из левой таблицы и соответствующие строки из правой таблицы. Если в правой таблице нет соответствия, используются NULL значения.

RIGHT JOIN (или RIGHT OUTER JOIN): Возвращает все строки из правой таблицы и соответствующие строки из левой таблицы. Если в левой таблице нет соответствия, используются NULL значения.

FULL JOIN (или FULL OUTER JOIN): Возвращает строки, которые имеют соответствие в любой из таблиц. Если нет соответствия, используются NULL значения.

2.4 Чем отличается объединение LEFT JOIN от RIGHT JOIN?

Ответ: Основное различие между LEFT JOIN и RIGHT JOIN заключается в том, что LEFT JOIN возвращает все строки из левой таблицы и соответствующие строки из правой таблицы, даже если в правой таблице нет соответствия. С другой стороны, RIGHT JOIN возвращает все строки из правой таблицы и соответствующие строки из левой таблицы, даже если в левой таблице нет соответствия.

2.5 Что такое подзапрос?

Ответ: Подзапрос (subquery) в SQL - это запрос, вложенный в другой запрос. Подзапрос может использоваться в различных частях SQL-запроса, таких как SELECT, INSERT, UPDATE или DELETE, и может возвращать одно значение, одну строку или несколько строк.

2.6 Для чего применяется ключевое слово UNION

Ответ: Ключевое слово UNION используется для объединения результатов двух или более запросов в один набор результатов. Он удаляет повторяющиеся строки из результирующего набора. Общее количество столбцов и их типы данных должны совпадать для всех запросов.

**3 Вывод**

В ходе лабораторной работы был изучен процесс создавать многотабличных SQL-запросов на выборку.

**Практической работе №15  
Применение встроенных функций SQL**

**1 Цель работы**

1.1 Изучить процесс создания SQL-запросов, содержащих встроенные функции в СУБД MS SQL Server.

**2. Контрольные вопросы**

2.1 В чем преимущество применения встроенных функций?

Ответ: **Универсальность**: Встроенные функции предоставляют готовые решения для широкого спектра задач, таких как математические вычисления, манипуляции строками, преобразования данных и т. д.

**Эффективность**: Они оптимизированы для работы с базами данных, что обычно обеспечивает более высокую производительность, чем пользовательские функции.

**Простота использования**: Встроенные функции обычно имеют простой синтаксис, что облегчает их использование и понимание.

**Надежность**: Эти функции широко тестируются и используются многими пользователями, что обычно обеспечивает их надежную работу.

2.2 Какие виды системных функций имеются в MSSQL?

Ответ: MSSQL предоставляет различные системные функции, включая:

Математические функции: ABS, CEILING, FLOOR, ROUND, SQRT и т. д.

Строковые функции: LEN, LEFT, RIGHT, SUBSTRING, CHARINDEX и т. д.

Дата и временные функции: GETDATE, DATEPART, DATEDIFF, DATEADD и т. д.

Преобразования данных: CONVERT, CAST и т. д.

Агрегатные функции: SUM, AVG, COUNT, MIN, MAX и т. д.

2.3 Каким образом указываются передаваемые в функцию параметры?

Ответ: Параметры передаются в функцию в скобках после имени функции. Для каждого параметра указывается его значение или выражение.

**3 Вывод**

В ходе лабораторной работы я изучил процесс создания SQL-запросов, содержащих встроенные функции в СУБД MS SQL Server.