

Министерство науки и высшего образования Российской Федерации  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Кузбасский государственный технический университет  
имени Т. Ф. Горбачева»

Институт профессионального образования

Кафедра информатики и информационных систем

Составитель О. С. Семенова

## **Основы проектирования баз данных**

### **Методические указания к самостоятельной работе**

Рекомендовано ЦМК специальности  
09.02.07 Информационные системы и программирования  
в качестве электронного издания для использования  
в образовательном процессе

Кемерово 2023

Рецензенты:

Е. А. Ощепкова – преподаватель кафедры информатики и информационных систем

**Семенова Ольга Сергеевна**

**Основы проектирования баз данных :** методические указания к самостоятельной работе для обучающихся специальности СПО 09.02.07 «Информационные системы и программирование» / сост. О. С. Семенова, Кузбасский государственный технический университет имени Т. Ф. Горбачева. – Кемерово, 2023. – Текст : электронный.

Приведенные методические указания к самостоятельной работе по курсу «Основы проектирования баз данных» позволяют углубить знания, полученные в ходе аудиторных занятий; способствуют закреплению теоретических положений; развивают навыки по их практическому применению.

© Кузбасский государственный  
технический университет  
имени Т. Ф. Горбачева, 2023  
© Семенова О. С.,  
составление, 2023

## **Часть №1. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО САМОСТОЯТЕЛЬНОЙ ПОДГОТОВКЕ К ИЗУЧЕНИЮ ТЕОРЕТИЧЕСКОГО МАТЕРИАЛА**

### **Раздел 1. Основные понятия баз данных**

#### ***Тема 1.1. Основные понятия теории БД***

##### **Содержание темы**

1. Основные понятия теории БД
2. Технологии работы с БД

**Литература [4].**

##### **Методические рекомендации**

В процессе изучения данной темы обучающийся должен освоить основные понятия баз данных.

База данных – именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

Банк данных – это система специальным образом организованных данных – баз данных, программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и многоцелевого использования данных.

Система управления базами данных (СУБД) – совокупность языковых и программных средств, предназначенных для создания и совместного использования БД многими пользователями.

Информационная система – это совокупность тем или иным способом структурированных данных (баз данных) и комплекса аппаратно-программных средств для хранения данных и манипулирования ими.

Предметная область – часть реального мира, для которой необходимо построить информационную модель. Примерами предметной области являются: предприятие, склад, универмаг, банк, университет и т.д.

Объект (сущность) – элемент информационной системы, это нечто существующее и различимое, т.е. объектом можно назвать, то «нечто», для которого существуют название и способ отличать один подобный объект от другого. Объект может быть реальным (человек, предмет, населенный пункт) и абстрактным (событие, счет покупателя, дисциплина).

Классом объектов называется совокупность объектов, обладающих одинаковым набором свойств. Объекты и их свойства являются понятиями реального мира.

Атрибут (свойство) – информационное отображение свойств объекта. Пример: клиент магазина, продающий автомобили имеет такие атрибуты как фамилию, имя, отчество, адрес, и возможно идентификационный номер. Атрибут при реализации информационной модели, на каком либо носителе информации часто называют элементом данных, полем данных или просто полем. Атрибут некоторого набора объектов сам может быть набором объектов, имеющим собственные атрибуты.

Таблица – некоторая регулярная структура, состоящая из конечного набора однотипных записей. Значения данных представляют собой действительные данные, содержащиеся в каждом элементе данных. От того, как элементы данных описывают объект, их значения могут быть количественными, качественными или описательными.

Экземпляр объекта – один набор значений его элементов данных. Запись данных - это совокупность значений связанных элементов данных.

Связь – это функциональная зависимость между сущностями.

Доменом называется набор записей одного типа, отвечающих поставленным условиям.

Ключевым элементом данных называется такой элемент, по которому можно определять значения других элементов данных. Первичный ключ – это атрибут (или группа атрибутов), которые единственным образом идентифицирует каждую строку в таблице. Альтернативный ключ – это атрибут (или группа атрибутов), несовпадающий с первичным ключом и уникально идентифицирующий экземпляр объекта.

Словарь данных – это централизованное хранилище сведений об объектах, элементах данных, взаимосвязях между объектами, их источниках, значениях, использовании и формах представления данных.

Нормализация отношений – это процесс построения оптимальной структуры таблиц и связей в реляционной модели данных.

SQL (Structured Query Language) – Структурированный Язык Запросов – стандартный язык запросов по работе с реляционными БД.

Триггеры – это программа действий, которая автоматически осуществляется при выполнении операций обновления, добавления или удаления данных. Ни пользователь, ни приложение не могут активизировать триггер. Триггер автоматически выполняется, когда пользователь или приложение выполняют с БД определенные действия. Триггер является мощным инструментом контроля над изменением данных в БД, а также помо-

гает программисту автоматизировать операции, которые должны выполняться в этом случае.

### ***Тема 1.2. Взаимосвязи в моделях и реляционный подход к построению моделей***

#### **Содержание темы**

1. Логическая и физическая независимость данных
2. Типы моделей данных. Реляционная модель данных
3. Реляционная алгебра

**Литература** [1,2,4].

#### **Методические рекомендации**

В процессе изучения данной темы обучающийся должен усвоить, что база данных (БД) – это совокупность специальным образом организованных и взаимосвязанных данных по конкретной предметной области, хранимых на внешних носителях информации и управляемых средствами СУБД.

В базе данных обеспечивается логическая взаимосвязь хранимых данных и их минимально необходимая избыточность.

По способу организации данных различают:

- Иерархические (рисунок 1)

Иерархическая модель данных – представление базы данных в виде древовидной (иерархической) структуры, состоящей из объектов (данных) различных уровней.

Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка (объект более близкий к корню) к потомку (объект более низкого уровня), при этом возможна ситуация, когда объект-предок не имеет потомков или имеет их несколько, тогда как у объекта-потомка обязательно только один предок. Объекты, имеющие общего предка, называются близнецами (в программировании применительно к структуре данных дерево устоялось название братья).



Рисунок 1 – Иерархическая БД

- Сетевые (рисунок 2)

Сетевая СУБД – СУБД, построенная на основе сетевой модели данных. К основным понятиям сетевой модели базы данных относятся: уровень, элемент (узел), связь.

Узел – это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа. В сетевой структуре каждый элемент может быть связан с любым другим элементом.

Сетевые базы данных подобны иерархическим, за исключением того, что в них имеются указатели в обоих направлениях, которые соединяют родственную информацию.

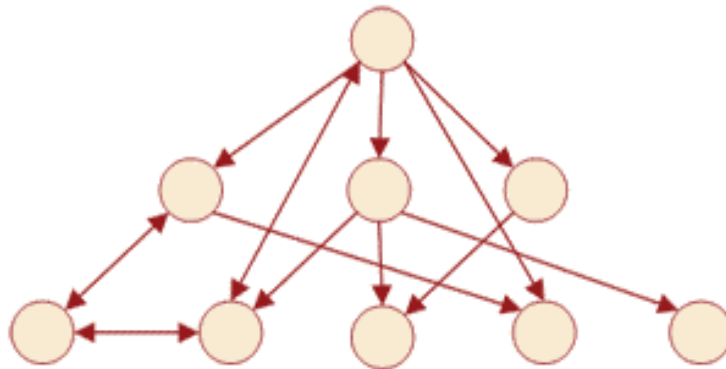


Рисунок 2 – Сетевая БД

- Реляционные базы данных (рисунок 3)

Данные структурированы в виде отдельных таблиц (рисунок 4). Таблицы могут быть связаны между собой.

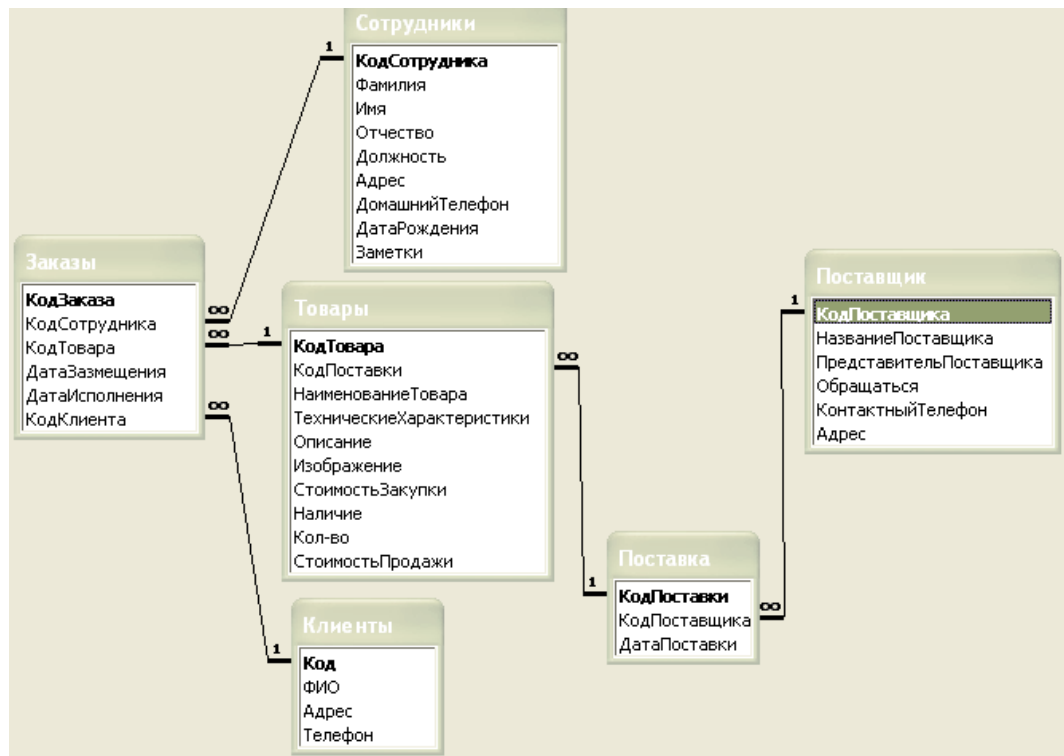


Рисунок – 3 Структура реляционной БД

Маршруты			
ID	Номер маршрута	Тип маршрута	
11	9	автобус	
12	10	автобус	
13	10т	автобус	
14	11т	автобус	
15	12	автобус	
16	13т	автобус	
17	14	автобус	
18	14т	автобус	
19	15т	автобус	
20	16	автобус	
21	16т	автобус	
22	17	автобус	
23	17т	автобус	
24	18	автобус	
25	18т	автобус	
26	19	автобус	
27	19т	автобус	

Рисунок 4 – Таблица реляционной БД

Процесс создания базы данных можно представить в виде трех этапов:

### 1. Инфологическое (концептуальное) описание баз данных

Исходными данными для осуществления инфологического проектирования является словесная и документальная характеристика предметной

области. На этом этапе решается вопрос о том, какие данные должны храниться в базе и какого типа информационные выборки и отчеты могут потребоваться пользователю БД.

## **2. Логическое проектирование баз данных**

На этом этапе осуществляется выбор подходящей системы управления базами данных (СУБД) и представление инфологической модели предметной области в форме структуры базы данных конкретной СУБД. Для реляционных баз данных на этом этапе производится описание структуры каждой таблицы и их взаимосвязей.

**3. Физическое проектирование.** Предполагает определение способов и мест размещения базы данных, оценку ее объема и других параметров.

### **Контрольные вопросы:**

1. Что такое базы данных?
2. Опишите первые модели данных.
3. Опишите иерархические СУБД.
4. Опишите сетевые базы данных.
5. Что такое реляционная модель данных?
6. Что такое инфологическое проектирование базы данных?
7. Что такое логическое проектирование базы данных?
8. Что такое физическое проектирование базы данных?

## **Раздел 2. Проектирование баз данных**

### **Тема 2.1. Этапы проектирования баз данных**

#### **Содержание темы**

1. Основные этапы проектирования БД
2. Концептуальное проектирование БД
3. Нормализация БД.

**Литература** [1,2,4].

#### **Методические рекомендации**

Большинство современных систем управления базами данных (СУБД) разработаны на основе реляционной алгебры.

Первая работа по реляционной модели данных «A Relational Model of Data for Large Shared Data Banks» была опубликована в 1970 г. Её автор - Эдгар Франк Кодд. В своей статье Э. Кодд вывел несколько правил, или форм, по упорядочиванию данных и их отношений.

**Нормализация БД** – это проектирование базы данных так, чтобы она была компактной и не несла логическую избыточность. Существует не-



сколько разновидностей нормализации, так называемые **нормальные формы**. Все они идут в порядке усложнения от простого к сложному.

Каждой нормальной форме соответствует некоторый определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений.

Всего существует 6 нормальных форм. На практике редко нормализуют выше 3-ей нормальной формы.

Существует специальная терминология, принятая в теории реляционных БД (рисунок 5):

- *Атрибут* соответствует столбцу таблицы, а именно – свойствам объектов, сведения о которых хранятся в ней. В СУБД ACCESS атрибуты называют **полями**.
- *Кортеж* соответствует заполненной строке таблицы. В СУБД ACCESS кортежи называют **записями**.
- *Кардинальное число* – количество кортежей в таблице в текущий момент времени.
- *Домен* – это общая совокупность значений, из которой берутся конкретные значения для конкретного атрибута.

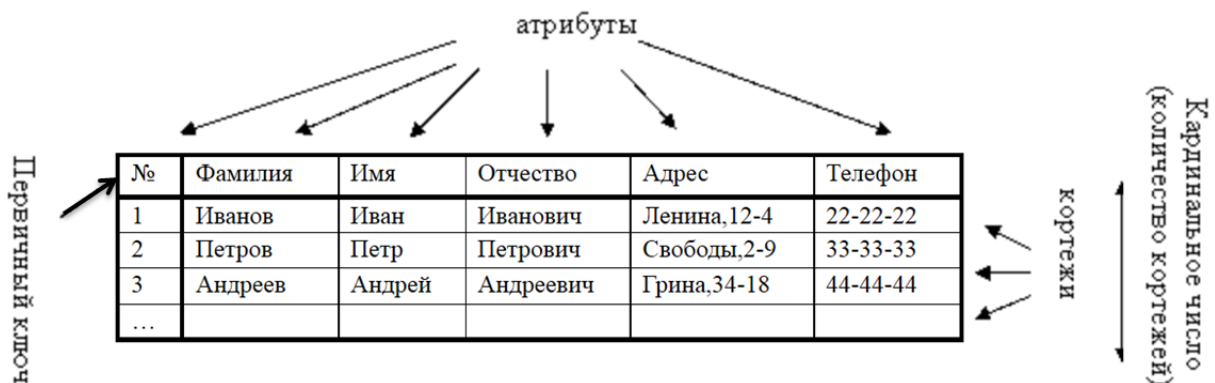


Рисунок 5 – Основные понятия БД

Ключи являются составляющей частью нормализованных таблиц. Бывают двух видов — внешние и первичные.

**Первичный ключ** — это атрибут, значения которого уникально идентифицируют каждую запись таблицы. Первичный ключ отвечает следующим условиям: он должен иметь значение, не NULL; быть неизменным; иметь уникальное значение для каждой строки.

**Внешние ключи** — это ссылки на первичные ключи других таблиц.

**Отношения** – это указатели, которые показывают, как соотносятся данные в одной таблице с данными в другой. Отношения бывают 3-х видов:

- 1) Связь «один-к-одному». Такая связь означает, что каждому значению реквизита *A* соответствует одно и только одно значение связанного с ним реквизита *B*, и наоборот.
- 2) Связь «один-ко-многим». Эта связь означает, что каждому значению реквизита *A* соответствует ноль, одно или несколько значений связанного с ним реквизита *B*, а каждому значению реквизита *B* соответствует одно и только одно значение реквизита *A*.
- 3) Связь «многие-ко-многим». Такая связь означает, что каждому значению реквизита *A* соответствует несколько значений связанного с ним реквизита *B*, и наоборот.

### ***Первая нормальная форма (1NF)***

Объект базы данных находится в первой нормальной форме тогда, когда каждый ее атрибут *атомарен*. Атрибут атомарен тогда, когда его значение теряет смысл при перестановке любой из его частей или при любом разбиении его на части. То есть, одно поле – одно значение.

### ***Вторая нормальная форма (2NF)***

Объект базы данных находится во второй нормальной форме тогда, когда он находится в первой нормальной форме и при этом любой его атрибут, не входящий в состав потенциального ключа, функционально полностью зависит от каждого потенциального ключа. Это правило говорит об отделении функционально полных зависимостей на отдельные структуры.

### ***Третья нормальная форма (3NF)***

Объект базы данных находится в третьей нормальной форме тогда, когда он находится во второй нормальной форме и отсутствуют транзитивные зависимости не ключевых объектов от ключевых.

*Транзитивная зависимость* – это очевидная зависимость между полями. Если поле *A* равно *x*, то поле *B* обязательно будет равно *y*. А если поле *B* равно *z*, то тогда поле *C* будет равно *m*. Такой зависимости между объектами быть не должно.

## ***Тема 2.2. Проектирование структур баз данных***

### **Содержание темы**

1. Средства проектирования структур БД
2. Организация интерфейса с пользователем

Литература [1,2,4].

### Методические рекомендации

Основными объектами базы данных являются таблицы.

**Таблица** – фундаментальная структура системы управления реляционными базами данных. Таблица – это объект, предназначенный для хранения данных в виде записей (строк) и полей (столбцов). При этом каждое поле содержит отдельную часть записи (например, фамилию, должность или инвентарный номер). Обычно каждая таблица используется для хранения сведений по одному конкретному вопросу (например, о сотрудниках или заказах).

Создать таблицу в Microsoft SQL Server можно тремя способами:

- с помощью визуального интерфейса SQL Server Management Studio (SSMS), выбрав в обозревателе объектов Tables и нажав в контекстном меню New (рисунок 6),
- с помощью инструкции на языке T-SQL,
- с помощью графического конструктора SSMS, в существующей диаграмме баз данных.

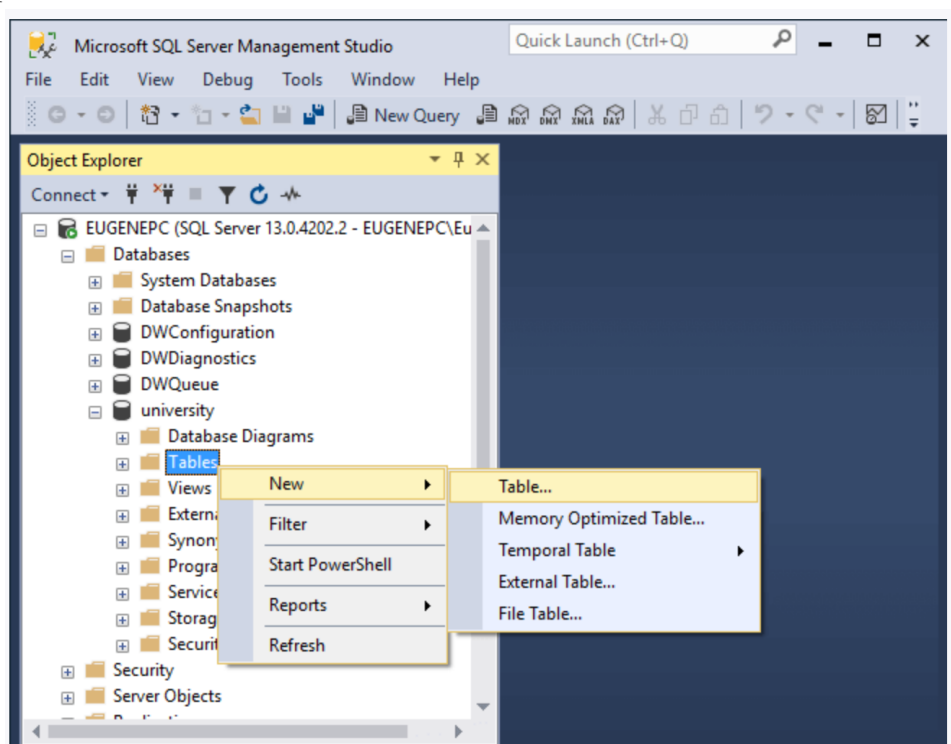


Рисунок 6 – Создание таблицы с помощью визуального интерфейса SQL Server Management Studio

### **Тема 2.3. Организация запросов SQL**

#### **Содержание темы**

1. Основные понятия языка SQL. Синтаксис операторов, типы данных.
2. Команды языка SQL для определения, обработки и администрирования данных.
3. Организация запросов на выборку данных, обновление, удаление. Сортировка и группировка данных в SQL.

**Литература** [1,2,4].

#### **Методические рекомендации**

**Запрос** – это запрограммированное на специальном языке (SQL) требование к системе на выполнение некоторых действий с записями одной или нескольких таблиц. Запросы создаются пользователем для выборки нужных сведений из одной или нескольких связанных таблиц. С помощью запроса можно также обновить, удалить или добавить данные в таблицы или создать новые таблицы на основе уже существующих.

Изначально SQL был основным способом работы пользователя с базой данных и позволял выполнять следующий набор операций:

- создание в базе данных новой таблицы;
- добавление в таблицу новых записей;
- изменение записей;
- удаление записей;
- выборка записей из одной или нескольких таблиц (в соответствии с заданным условием);
- изменение структур таблиц.

Со временем SQL усложнился – обогатился новыми конструкциями, обеспечил возможность описания и управления новыми хранимыми объектами (например, индексы, представления, триггеры и хранимые процедуры) – и стал приобретать черты, свойственные языкам программирования.

Для создания запросов с помощью языка структурированных запросов необходимо знать, что SQL предназначен для манипулирования данными в реляционных базах данных, определения структуры баз данных и для управления правами доступа к данным в многопользовательской среде. Поэтому, в язык SQL в качестве составных частей входят:

- язык манипулирования данными (Data Manipulation Language, DML)

- язык определения данных (Data Definition Language, DDL)
- язык управления данными (Data Control Language, DCL).

Язык манипулирования данными используется, как это следует из его названия, для манипулирования данными в таблицах баз данных. Он состоит из 4 основных команд:

SELECT       (выбрать)  
 INSERT       (вставить)  
 UPDATE       (обновить)  
 DELETE       (удалить)

Язык определения данных используется для создания и изменения структуры базы данных и ее составных частей – таблиц, индексов, представлений (виртуальных таблиц), а также триггеров и сохраненных процедур. Основными его командами являются:

CREATE DATABASE   (создать базу данных)  
 CREATE TABLE   (создать таблицу)  
 CREATE VIEW       (создать виртуальную таблицу)  
 CREATE INDEX      (создать индекс)  
 CREATE TRIGGER   (создать триггер)  
 CREATE PROCEDURE   (создать сохраненную процедуру)  
 ALTER DATABASE    (модифицировать базу данных)  
 ALTER TABLE      (модифицировать таблицу)  
 ALTER VIEW (модифицировать виртуальную таблицу)  
 ALTER INDEX       (модифицировать индекс)  
 ALTER TRIGGER   (модифицировать триггер)  
 ALTER PROCEDURE   (модифицировать сохраненную процедуру)  
 DROP DATABASE   (удалить базу данных)  
 DROP TABLE      (удалить таблицу)  
 DROP VIEW       (удалить виртуальную таблицу)  
 DROP INDEX       (удалить индекс)  
 DROP TRIGGER     (удалить триггер)  
 DROP PROCEDURE   (удалить сохраненную процедуру)

Язык управления данными используется для управления правами доступа к данным и выполнением процедур в многопользовательской среде. Более точно его можно назвать "язык управления доступом". Он состоит из двух основных команд:

GRANT (дать права)  
 REVOKE (забрать права)

SQL остаётся единственным механизмом связи между прикладным программным обеспечением и базой данных. В то же время современные СУБД, а также информационные системы, использующие СУБД, предоставляют пользователю развитые средства визуального построения запросов:

- 1) мастера запросов, который работает в диалоговом режиме;
- 2) конструктора запросов.

Принцип создания запросов с помощью конструктора запросов прост: после запуска конструктора автоматически появляется окно «Добавление таблицы», в котором выбирается одна или несколько таблиц, необходимых для решения поставленных целей; затем устанавливаются связи между таблицами, добавляются необходимые поля, устанавливается порядок сортировки, условия и т.д.

Оператор SELECT является наиболее часто используемым оператором в T-SQL и используется для извлечения данных из одной или нескольких таблиц. Чтобы выбрать данные из одной таблицы, необходимо использовать следующий синтаксис:

SELECT столбец1, столбец2, столбец3,...

FROM имя\_таблицы;

Для фильтрации данных необходимо использовать предложение WHERE, после которого необходимо записать условие, которое должно быть выполнено для извлечения данных.

Например, если необходимо получить клиентов только из России, мы можем изменить запрос следующим образом:

SELECT [Имя клиента], Страна, Город

FROM Клиент

WHERE Страна = 'Россия';

При извлечении данных из одной таблицы можно использовать условие BETWEEN, чтобы указать диапазон возвращаемых значений.

Например, предположим, что есть таблица с именем «Продажа», которая содержит столбцы для названия продукта, даты продажи и суммы. Чтобы получить все данные о продажах за январь можно использовать условие BETWEEN следующим образом:

SELECT [Название продукта], [Дата продажи], сумма

FROM Продажа

WHERE [Дата продажи] BETWEEN '2019-01-01' AND '2019-01-31';

Оператор LIKE используется для поиска строк, которые соответствуют заданному шаблону. Шаблон может содержать специальные символы, которые представляют любой символ или группу символов. Например, если мы хотим найти все продукты, содержащие слово "Яблоки", мы можем использовать следующий запрос:

```
SELECT * FROM Продажа
WHERE [Название продукта] LIKE '%Яблоки%'
```

Здесь знак процента (%) обозначает, что перед и после слова "Яблоки" могут быть любые символы. Таким образом, запрос найдет все строки, содержащие слово "Яблоки", вне зависимости от того, где это слово находится в строке. Кроме знака процента, есть еще два специальных символа, которые можно использовать в операторе LIKE:

- Знак подчеркивания (\_): означает любой один символ.
- Квадратные скобки ([]): означают любой один символ из заданного диапазона. Например, [a-z] означает любой один символ от a до z.

В условии отбора можно использовать встроенные функции. Например, чтобы получить все данные о продажах за январь можно использовать функцию Month():

```
SELECT [Название продукта], [Дата продажи], сумма
FROM Продажа
WHERE Month([Дата продажи])=1;
```

В SQL для создания запроса на обновление данных используется оператор UPDATE. Синтаксис запроса выглядит следующим образом:

```
UPDATE      имя_таблицы      SET      имя_столбца1=значение1,
имя_столбца2=значение2 WHERE условие;
```

Обратите внимание, что обновление данных возможно только в тех строках, которые удовлетворяют заданному условию.

Пример. Дана таблица Employees с полями id, name, salary. Необходимо увеличить зарплату сотрудника с id=1 на 10%:

```
UPDATE Employees SET salary=salary*1.1 WHERE id=1;
```

Для добавления данных в таблицу используется оператор INSERT INTO. Синтаксис запроса выглядит так:

```
INSERT INTO имя_таблицы (столбец1, столбец2, ...) VALUES (значение1, значение2, ...);
```

Обратите внимание, что в скобках после имени таблицы указываются названия столбцов, в которые будут вставляться данные. Если не указывать названия столбцов, то данные будут вставлены в столбцы в том по-

рядке, в котором они указаны в таблице.

Пример. Дана таблица Students с полями id, name, age. Необходимо добавить нового студента с именем "Иван" и возрастом 20 лет:

```
INSERT INTO students (name, age) VALUES ('Иван', 20);
```

Иногда возникает необходимость удалить данные из таблицы, например, если объект перестал существовать или изменил свое состояние. В SQL запросах для удаления данных используется команда DELETE:

```
DELETE FROM имя_таблицы
```

```
WHERE условие;
```

Здесь имя\_таблицы – это имя таблицы, из которой нужно удалить данные, а условие – это условие, которое определяет, какие именно строки нужно удалить.

Например, чтобы удалить все данные из таблицы Employees, можно использовать следующий запрос:

```
DELETE FROM Employees;
```

### **Контрольные вопросы:**

1. Что такое базы данных?
2. Опишите первые модели данных.
3. Опишите иерархические СУБД.
4. Опишите сетевые базы данных.
5. Что такое реляционная модель данных?
6. Дайте общую характеристику реляционной модели данных. Опишите типы данных, используемые в реляционной модели.
7. Дайте определение понятию «первичный ключ».
8. Дайте определение понятию «внешний ключ».
9. Укажите операции, которые могут нарушить ссылочную целостность.
10. Укажите стратегии поддержания ссылочной целостности.
11. Опишите этапы разработки базы данных.
12. Дайте определение первой нормальной форме.
13. Дайте определение второй нормальной форме.
14. Дайте определение третьей нормальной форме..
15. Как организовать интерфейс с пользователем?
16. Опишите средства проектирования структур БД.
17. Опишите процесс создания запросов.
18. Опишите операторы SQL, предназначенные для создания, удаления таблиц.



19. Опишите операторы SQL, предназначенные для выборки данных из таблиц.
20. Опишите операторы SQL, предназначенные для добавления/удаления полей из таблиц.
21. Для чего используется запрос на обновление?
22. Как проверить результат выполнения запроса?
23. Что осуществляет запрос на добавление записей?
24. Как можно узнать о числе обновляемых записей?
25. Какое есть условие для успешного выполнения запроса на добавление?

## **Часть №2 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ**

### **1. Индивидуальное задание по теме «Взаимосвязи в моделях и реляционный подход к построению моделей».**

Используя иерархическую, сетевую, реляционную, постреляционную, многомерную, объектно-ориентированную модели данных структурировать информацию из заданной предметной области.

1. Построить иерархическую, сетевую модель данных. Указать корневой тип, подтипы, узлы.
  - а) Генеалогическое древо;
  - б) Сайт kuzstu.ru;
  - в) Перевозка пассажиров между 5-ю населенными пунктами;
  - г) Транспортировка грузов со склада по 10 пунктам разгрузки;
  - д) Улично-дорожная сеть из 6-и транспортных узлов;
2. Построить реляционную модель данных. Указать атрибуты, кортежи, степень отношения, кардинальное число, домен.
  - а) Школьники;
  - б) Компьютерные комплектующие;
  - в) Запчасти к автомобилю;
3. Построить многомерную модель данных.
  - а) Расписание движения поездов;
  - б) Прибыль организаций за 10 лет;
  - в) Продажи топлива;
4. Построить объектно-ориентированную модель данных.
  - а) Объектная модель Word.
  - б) Объектная модель Excel

Литература: [4]

### **2. Индивидуальное задание по теме «Этапы проектирования баз данных».**

Осуществить инфологическое, логическое и физическое проектирование базы данных в выбранной предметной области (см. варианты ИЗ).

Разработка структуры БД начинается с изучения метода нормализации и метода построения ER-модели. Используя один из этих методов, строится логическая модель БД, производится разбивка на таблицы.

Затем определяется генеральный список полей. В соответствии с типом данных, размещаемых в каждом поле, определяют наиболее подходящий тип для каждого поля.

Далее распределяют поля генерального списка по базовым таблицам. На первом этапе распределение производят по функциональному признаку. Цель – обеспечить, чтобы ввод данных в одну таблицу производился, по возможности, в рамках одного подразделения, а еще лучше – на одном рабочем месте.

В каждой из таблиц намечают *ключевое поле*. В качестве такого выбирают поле, данные в котором повторяться не могут. Если в таблице нет полей, которые можно было бы использовать, как ключевые, всегда можно ввести дополнительное поле типа Счетчик – оно не может содержать повторяющихся данных по определению.

Определяют связи между таблицами. Связь между таблицами организуется на основе общего поля, причем в одной из таблиц оно обязательно должно быть ключевым.

Литература: [1,2,4]

### **3. Индивидуальное задание по теме «Проектирование структур баз данных».**

Создать в выбранной СУБД таблицы, заполнить их данными, установить между таблицами связи.

Используя разработанную физическую модель БД в среде SQL Server создать диаграмму БД. Произвести наполнение таблиц БД в соответствии с заданной предметной областью (см. варианты ИЗ).

Литература: [1,2,4]

### **4. Индивидуальное задание по теме «Организация запросов SQL».**

Создать запросы на выборку информации из одной или нескольких таблиц, запросы на обновление, удаление и добавление данных в таблицы (см. варианты ИЗ).

Литература: [1,2,4]

**Варианты домашних заданий И31-И34:*****Вариант 1. База данных Туризм.***

- Создать запрос для отображения всей информации о фирмах, предлагающих путевки на отдых в горнолыжных базах.
- Создать запрос для отображения информации о стоимости путевок в Египет с учетом предоставляемых скидок.
- Создать отчет по обеим таблицам.

***Вариант 2. База данных Библиотека.***

- Создать параметрический запрос для отображения фамилий и телефонов учеников, которые должны сдать книги до даты, определенной параметром, и названий этих книг.
- Создать запрос для отображения числа учеников каждого класса, взявших книги.
- Создать отчет по обеим таблицам.

***Вариант 3. База данных Банк.***

- Создать запрос на создание таблицы, отображающей информацию о курсах продажи и покупки долларов США и Канады.
- Создать запрос для отображения информации о сумме продажи по всем отделениям.
- Создать отчет по продаже и покупке валюты по всем отделениям с указанием общей суммы.

***Вариант 4. База данных Магазин.***

- Создать параметрический запрос для отображения всей информации о фирмах, поставляющих определенный значением параметра товар.
- Создать запрос для отображения общей суммы поставок каждого товара.
- Создать отчет по второй таблице, включив все поля, указать общую сумму поставок каждого товара.

### Список литературы

1. Федорова, Г. Н. Основы проектирования баз данных : учебное пособие для образовательных учреждений, реализующих программы среднего профессионального образования по специальности "Информационные системы (по отраслям)" : [для студентов СПО] / Г. Н. Федорова. – Москва : Академия, 2017. – 224 с. – Доступна электронная версия: <http://www.academia-moscow.ru/catalogue/4831/296505/>
2. Голицына, О. Л. Основы проектирования баз данных [Текст : электронный]. – Москва : НИЦ ИНФРА-М, 2016. – 416 с. – Режим доступа: <http://znanium.com/go.php?id=552969>. – Загл. с экрана. (14.12.2018)
3. Шустова, Л. И. Базы данных [Текст : электронный]. – Москва : НИЦ ИНФРА-М, 2018. – 304 с. – Режим доступа: <http://znanium.com/go.php?id=967755>. – Загл. с экрана. (14.12.2018)
4. Цветкова, М. С. Информатика [Текст : электронный] : учебник для использования в учебном процессе образовательных учреждений СПО на базе основного общего образования с получением среднего общего образования / М. С. Цветкова, И. Ю. Хлобыстова. – Москва : Академия, 2017. – 352 с. – Режим доступа: <http://academia-moscow.ru/reader/?id=227485#copy>. – Загл. с экрана. (14.12.2018)

## Содержание

Часть №1. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО САМОСТОЯТЕЛЬНОЙ ПОДГОТОВКЕ К ИЗУЧЕНИЮ ТЕОРЕТИЧЕСКОГО МАТЕРИАЛА .....	3
Часть №2 МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ВЫПОЛНЕНИЮ ИНДИВИДУАЛЬНЫХ ЗАДАНИЙ.....	18
Список литературы.....	21