# Практическая работа № 2

ВЫПОЛНЕНИЕ ЭЛЕМЕНТАРНЫХ ОПЕРАЦИЙ ПРИ РАБОТЕ С СУБД

1. **Цель работы**

Изучить технические возможности системы управления базами данных (СУБД) MySql при работе с простейшими базами данных (БД) и средства формирования SQL запросов к СУБД.

1. **Общие сведения**
   1. *Основные понятия БД*

В реляционной теории одним из главных понятий является отношение. Математически отношение определяется следующим образом. Пусть даны *n* множеств *D1*, *D2*, ..., *Dn*. Тогда *R* есть **отношение** над этими множествами, если *R* есть множество упорядоченных наборов вида <*d1*, *d2*, ..., *dn*>, где: *d1* – элемент из *D1*; *d2* – элемент из *D2*; ..., *dn* – элемент из *Dn*. При этом наборы вида <*d1*, d*2*, ..., dn> называются кортежами, а множества D*1*, *D2*, ..., *Dn –* **доменами**. Каждый кортеж состоит из элементов, выбираемых из своих доменов. Эти элементы называются атрибутами, а их значения – значениями атрибутов.

Поскольку в базах данных разнотипная информация может располагаться в отдельных файлах, то с точки зрения размещения данных отношение можно отождествлять с таблицей или, иногда, с файлом. Кортеж представляет собой строку в таблице, или, что-то же самое, запись. Атрибут же является столбцом таблицы, или полем в записи. Домен же представляется неким обобщенным типом, который может быть источником для типов полей в записи. Таким образом, следующие тройки терминов являются эквивалентными:

* + - отношение, таблица, файл (для локальных баз данных);
    - кортеж, строка, запись;
    - атрибут, столбец, поле.

Реляционная база данных представляет собой совокупность отношений, содержащих всю необходимую информацию и объединенных различными связями. Атрибут (или набор атрибутов), который может быть использован для однозначной идентификации конкретного кортежа (строки или записи), называется **первичным ключом**.

Во многих СУБД имеется механизм так называемых **внешних ключей**. Смысл этого механизма состоит в том, что некоему атрибуту (или группе атрибутов) одного отношения назначается ссылка на первичный ключ другого отношения; тем самым закрепляются связи подчиненностимежду этими отношениями.

* 1. *Синтаксис SQL команд описания таблиц, связи таблиц между собой, описания атрибутов и их типов*

**Structured Query Language** (SQL) – это непроцедурный язык, ис- пользуемый для формулировки запросов к БД в большинстве современных СУБД и в настоящий момент являющийся индустриальным стандартом.

Непроцедурность языка означает, что на нем можно указать, что нужно сделать с базой данных, но нельзя описать алгоритм этого процесса. Все алгоритмы обработки SQL-запросов генерируются самой СУБД и не зависят от пользователя. Язык SQL состоит из набора операторов, которые можно разделить на несколько категорий:

* + - **Data Definition Language** (DDL) – язык определения данных, позволяющий создавать, удалять и изменять объекты в базах данных;
    - **Data Manipulation Language** (DML) – язык управления данными, позволяющий модифицировать, добавлять и удалять данные в имеющихся объектах базы данных;
    - **Data Control Language** (DCL) – язык, используемый для управления пользовательскими привилегиями;
    - **Transaction Control Language** (TCL) – язык для управления изменениями, сделанными группами операторов;
    - **Cursor Control Language** (CCL) – язык для определения курсоров, подготовки операторов SQL к выполнению и некоторых других операций.

*2.2.1. Типы данных*

Язык SQL поддерживает большое к оли чест во т и п о в данных для представления и манипулирования информацией. Наиболее часто используемые типы данных даны в табл. 3. За дополнительной информацией по типам данных, поддерживаемых СУБД MySql, необходимо обратиться к документации.

* + 1. *Синтаксис DDL команд*

**Оператор создания таблиц БД** позволяет создавать новые таблицы и указывать связи между ними. Синтаксис (конструкции в квадратных скобках являются необязательными, конструкция { альтернатива1 | альтернатива2} предполагает использование одной из перечисленных альтернатив):

*CREATE [TEMPORARY] TABLE [IF NOT EXISTS] имя\_таблицы*

*(****определение\_столбца****,...) [свойства\_таблицы] [свойства\_раздела],*

где:

## Определение\_столбца::=

*{Имя\_столбца* ***описание\_столбца***

*| [CONSTRAINT [symbol]] PRIMARY KEY [index\_type] (имя\_столбца,...) [index\_option] ...*

*| {INDEX|KEY} [index\_name] [index\_type] (имя\_столбца\_индекса,...) [index\_option] ...*

*| [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY]*

*[index\_name] [index\_type] (имя\_столбца,...) [index\_option] ...*

*| {FULLTEXT|SPATIAL} [INDEX|KEY] [index\_name] (имя\_столбца,...) [index\_option] ...*

*| [CONSTRAINT [symbol]] FOREIGN KEY*

*[index\_name] (имя\_столбца,...)* ***reference\_definition***

*| CHECK (expr)}*

*<****Описание\_столбца****>::=*

***тип\_данных*** *[NOT NULL | NULL] [DEFAULT default\_value] [AUTO\_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]*

*[COMMENT 'string']*

*[COLUMN\_FORMAT {FIXED|DYNAMIC|DEFAULT}]*

***Тип\_данных****: BIT[(length)]*

*| TINYINT[(length)] [UNSIGNED] [ZEROFILL]*

*| SMALLINT[(length)] [UNSIGNED] [ZEROFILL]*

*| MEDIUMINT[(length)] [UNSIGNED] [ZEROFILL]*

*| INT[(length)] [UNSIGNED] [ZEROFILL]*

*| INTEGER[(length)] [UNSIGNED] [ZEROFILL]*

*| BIGINT[(length)] [UNSIGNED] [ZEROFILL]*

*| REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]*

*| DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]*

*| FLOAT[(length,decimals)] [UNSIGNED] [ZEROFILL]*

*| DECIMAL[(length[,decimals])] [UNSIGNED] [ZEROFILL]*

*| NUMERIC[(length[,decimals])] [UNSIGNED] [ZEROFILL]*

*| DATE*

*| TIME*

*| TIMESTAMP*

*| DATETIME*

*| YEAR*

*| CHAR[(length)]*

*[CHARACTER SET charset\_name] [COLLATE collation\_name]*

*| VARCHAR(length)*

*[CHARACTER SET charset\_name] [COLLATE collation\_name]*

*| BINARY[(length)]*

*| VARBINARY(length)*

*| TINYBLOB*

*| BLOB*

*| MEDIUMBLOB*

*| LONGBLOB*

*| TINYTEXT [BINARY]*

*[CHARACTER SET charset\_name] [COLLATE collation\_name]*

*| TEXT [BINARY]*

*[CHARACTER SET charset\_name] [COLLATE collation\_name]*

*| MEDIUMTEXT [BINARY]*

*[CHARACTER SET charset\_name] [COLLATE collation\_name]*

*| LONGTEXT [BINARY]*

*[CHARACTER SET charset\_name] [COLLATE collation\_name]*

*| ENUM(value1,value2,value3,...)*

*[CHARACTER SET charset\_name] [COLLATE collation\_name]*

*| SET(value1,value2,value3,...)*

*[CHARACTER SET charset\_name] [COLLATE collation\_name]*

## reference\_definition:

*REFERENCES имя\_таблицы (имя\_столбца,...) [MATCH FULL | MATCH PARTIAL | MATCH SIMPLE]*

*[ON DELETE {RESTRICT | CASCADE | SET NULL | NO ACTION}] [ON UPDATE {RESTRICT | CASCADE | SET NULL | NO ACTION}]*

Имена таблиц и определения столбцов должны быть уникальными. Назначение каждого атрибута в *<Ограничение столбца>*приведено далее:

* + - 1. UNIQUE – определяет уникальное (не повторяющееся значение атрибу- та);
      2. PRIMARY KEY– первичный ключ;
      3. FOREIGN KEY– внешний ключ;
      4. REFERENCES– определение ссылки на другую таблицу;
      5. NOT NULL– указывает, что значение атрибута не должно быть пустым;

**Оператор создания БД позволяет** создавать новые БД в СУБД для последующего использования при хранении и доступе к данным. Синтак- сис:*CREATE DATABASE имя\_базы\_данных*

Для осмысленного хранения информации как единых данных по предметной области, имеющих конкретный семантический смысл необхо- димо связать таблицы между собой. Связь между таблицами выполнена с использованием внешних ключей (FOREIGN KEY). Скрипты на языке SQL, описывающие таблицы и связи между ними, представлены далее.

*CREATE TABLE Disp(*

*ID\_Disp INTEGER not null primary key,*

*Disp\_Name CHAR(20) not null,*

*Cafedra CHAR(50),*

*Phone CHAR(8)*

*);REATE TABLE Person(*

*FIO CHAR(30) not null, Tab\_N INTEGER not null,*

*Ac\_Degree CHAR(20), PRIMARY KEY (Tab\_N)*

*);*

*CREATE TABLE Teach\_Load(*

*Teach\_ID INTEGER NOT NULL PRIMARY KEY,*

*Start\_Of\_Work DATE DEFAULT 'NOW' not null, Person\_ID INTEGER not null,*

*Disp\_ID INTEGER not null,*

*FOREIGN KEY (Person\_ID) REFERENCES Person (Tab\_N), FOREIGN KEY*

*(Disp\_ID) REFERENCES Disp (ID\_Disp));*

1. **Порядок выполнения работы**
   1. Создать базу данных.
   2. Создать и связать таблицы, схема данных для которых разрабатывалась в предыдущей работе.
2. **Содержание отчета**
   1. Цель работы
   2. Вариант индивидуального задания к работе.
   3. Тексты всех разработанных SQL скриптов для создания БД и таблиц внутри БД.
   4. Выводы по работе.