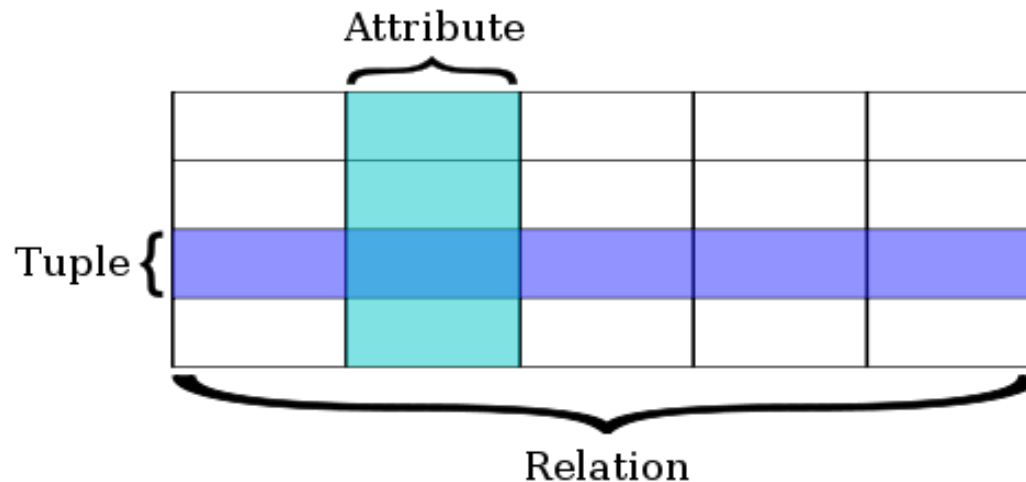


Реляционные базы данных

База данных MySQL

Что такое SQL?

- **Structured Query Language**
- SQL позволяет управлять базами данных
- SQL – это стандарт ANSI (American National Standards Institute)



Что может SQL?

- Выполнять **запросы** к базе данных
- **Извлекать** данные из базы данных
- **Вставлять** записи в базу данных
- **Обновлять** записи в базе данных
- **Удалять** записи из базы данных
- **Создавать** новые базы данных
- **Создавать** новые таблицы в базе данных
- **Создавать** хранимые процедуры в базе данных
- **Создавать** представления в базе данных
- Устанавливать **права доступа** к базам данных, таблицам, процедурам и т.п.

SQL – стандарт, но ...

- Существует множество реализаций стандарта
- Каждая база данных добавляет дополнительные возможности в язык
- Все базы данных, совместимые с стандартом ANSI, поддерживают одинаковое написание основных запросов

Использование SQL при разработке сайта

- Необходимо **СУБД** (**С**истема **у**правления **б**азами **д**анных)
- Нужен знать язык программирования, которые может исполняться на сервере, например, **PHP**
- Нужно знать **SQL**
- Уметь пользоваться **HTML** и **CSS**

Система управления реляционными базами данных

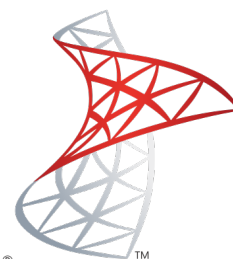
- **RDBMS** (**R**elational **D**atabase **M**anagement **S**ystem)
- **Таблица** – объект базы данных, в котором хранятся данные в **RDBMS**
- **Таблица** – это множество связанных записей данных – отношений
- Таблица состоит из **столбцов** или атрибутов (**column** or attribute) и **строчек** или кортежей (**row** or tuple)

Примеры RDBMS

ORACLE®



PostgreSQL



Microsoft®
SQL Server®

TERADATA



Типы операторов SQL

- **DDL, Data Definition Language** – операторы определения данных
- **DML, Data Manipulation Language** – операторы манипуляции данными ()
- **DCL, Data Control Language** – операторы определения доступа к данным
- **TCL, Transaction Control Language** – операторы управления транзакциями

Примеры операторов

- **SELECT** – выбрать данные
- **UPDATE** – изменить данные
- **DELETE** – удалить данные
- **INSERT INTO** – вставить данные
- **CREATE DATABASE** – создать базу данных
- **ALTER DATABASE** – изменить базу данных
- **CREATE TABLE** – создать таблицу
- **ALTER TABLE** – изменить таблицу
- **DROP TABLE** – удалить таблицу

CRATE DATABASE

- Синтаксис

```
CREATE DATABASE dbname;
```

- Пример

```
CREATE DATABASE cities;
```

CRATE TABLE

- Синтаксис

```
CREATE TABLE table_name
(
    column_name1 data_type(size),
    column_name2 data_type(size),
    column_name3 data_type(size),
    . . . .
);
```

CRATE TABLE

- Пример

```
CREATE TABLE city
(
    city_id INT NOT NULL AUTO_INCREMENT,
    city_name varchar(100) NOT NULL,
    country_name varchar(100) NOT NULL,
    population INT,
    PRIMARY KEY (city_id)
);
```

INSERT INTO

- Синтаксис

```
INSERT INTO table_name (column1, column2, column3, ...)
VALUES (value1, value2, value3, ...);
```

- Пример

```
INSERT INTO city(city_name, country_name, population)
VALUES ('Париж', 'Франция', 2300000);
```

Пример таблицы в базе данных

- Таблица CITY

city_id	city_name	country_name	population
1	Москва	Россия	12 000 000
2	Санкт-Петербург	Россия	5 000 000
3	Прага	Чехия	1 200 000
4	Минск	Белорусия	1 900 000
5	Киев	Украина	2 800 000
6	Лондон	Великобритания	8 500 000
7	Нью-Йорк	США	20 000 000

SELECT

- Синтаксис:
 - Выбрать набор столбцов из таблицы

```
SELECT column_name, column_name  
FROM table_name;
```

- Выбрать все столбцы из таблицы

```
SELECT * FROM table_name;
```

SELECT

- Пример:
 - Выбрать набор столбцов из таблицы

```
SELECT city_name, country_name  
FROM city;
```

- Выбрать все столбцы из таблицы

```
SELECT * FROM city;
```


SELECT и условие WHERE

- Синтаксис

```
SELECT column_name, column_name  
FROM table_name  
WHERE column_name operator value;
```

- Пример

```
SELECT *  
FROM city  
WHERE country_name = 'Россия';
```

Условные операторы

Оператор	Описание
=	Равно
<>	Неравно
>	Больше
<	Меньше
>=	Больше или равно
<=	Меньше или равно
IN	Множество допустимых значений
BETWEEN	Значение в промежутке
LIKE	Поиск шаблона

AND и OR

- Пример

```
SELECT *  
FROM city  
WHERE country_name = 'Россия'  
       OR (country_name like 'A%'  
           AND population > 1000000);
```

SELECT и сортировка ORDER

- Синтаксис

```
SELECT column_name, column_name  
FROM table_name  
ORDER BY column_name ASC|DESC, column_name ASC|DESC;
```

- Пример

```
SELECT *  
FROM city  
ORDER BY country_name, city_name DESC;
```

UPDATE

- Синтаксис

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE some_column = some_value;
```

- Пример

```
UPDATE city  
SET population = 5500000  
WHERE city_name = 'Санкт-Петербург';
```

DELETE

- Синтаксис

```
DELETE FROM table_name  
WHERE some_column = some_value;
```

- Пример

```
DELETE FROM city  
WHERE city_id = 115;
```

- Очистка таблицы

```
TRUNCATE TABLE table_name;
```

```
TRUNCATE TABLE city;
```

DROP

- Синтаксис

```
DROP TABLE table_name;
```

```
DROP DATABASE database_name;
```

- Пример

```
DROP TABLE city;
```

```
DROP DATABASE cities;
```

ALTER

- Синтаксис

```
ALTER TABLE table_name  
ADD column_name datatype;
```

```
ALTER TABLE table_name  
DROP COLUMN column_name;
```

- Пример

```
ALTER TABLE city  
ADD foundation date;
```

```
ALTER TABLE city  
DROP COLUMN foundation;
```


Типы данных в MySQL

- CHAR(n)
- VARCHAR(n)
- BINARY(n) или BYTE(n)
- VARBINARY(n)
- TEXT(n)
- INT или INTEGER
- BIGINT
- FLOAT
- DOUBLE или REAL
- CLOB
- BLOB
- TIME
- DATE
- DATETIME
- YEAR