

# **Основы базы данных SQL**

2020 г.



# Что такое SQL?



**SQL (Structured Query Language)** это язык программирования структурированных запросов, который используется в качестве эффективного способа сохранения данных, поиска их частей, обновления, извлечения из базы и удаления.

Произносится как **«Эскуэль/ЭсКьюЭль»**, реже **«СиКуЭль/СиКьюЭль»**, также можно услышать жаргонное **«Сиквэл/Сиквел»**.

Top 5 languages in terms of demand

	Language	Vacancies
1	SQL	95K
2	JavaScript	83K
3	Java	80K
4	C#	73K
5	Python	47K

Согласно данным **IT Jobs Watch**, SQL является самым запрашиваемым языком программирования среди работодателей.

*(IT Jobs Watch, собирает и анализирует данные о вакансиях в сфере ИТ в Великобритании.)*

# Что может SQL?

- SQL может выполнять запросы к базе данных
- SQL может извлекать данные из базы данных
- SQL может вставлять записи в базу данных
- SQL может обновлять записи в базе данных
- SQL может создавать новые базы данных
- SQL может удалять записи из базы данных



- SQL может создавать новые таблицы в базе данных
- SQL может создавать хранимые процедуры в базе данных
- SQL может создавать представления в базе данных
- SQL может устанавливать разрешения для таблиц, процедур и представлений

**Некоторые распространенные системы  
управления реляционными базами данных,  
использующие SQL**

**ORACLE®**

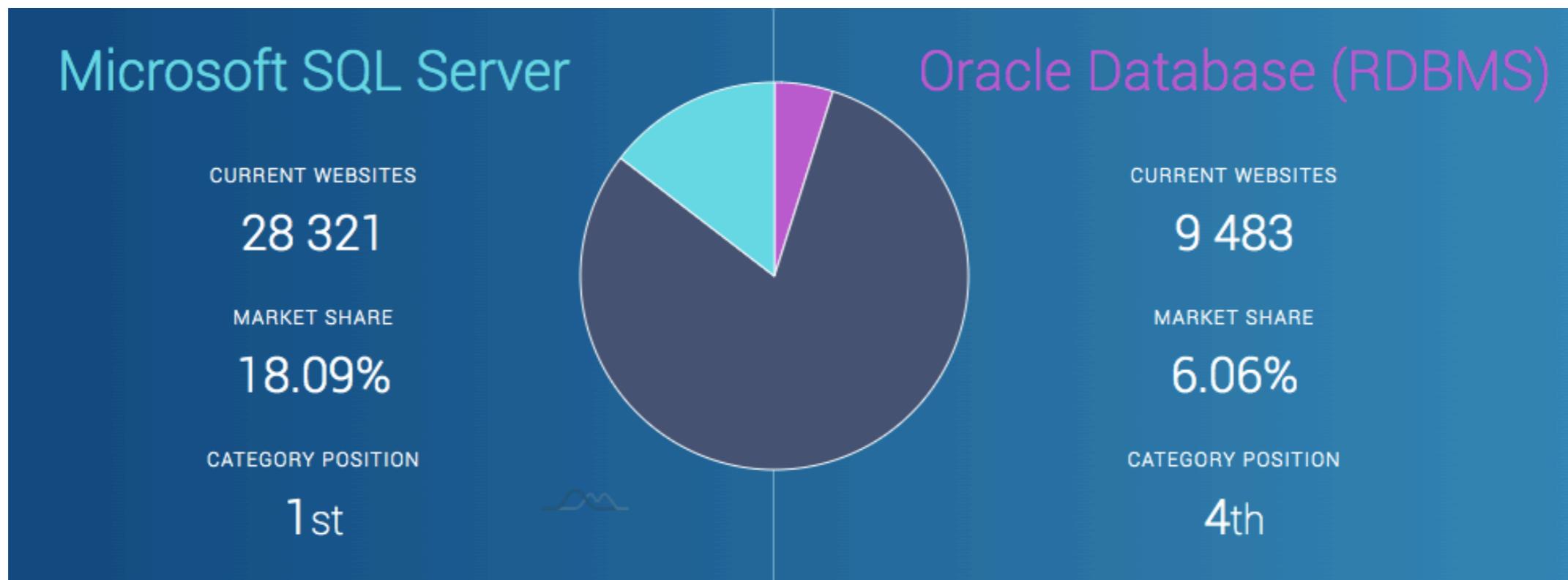
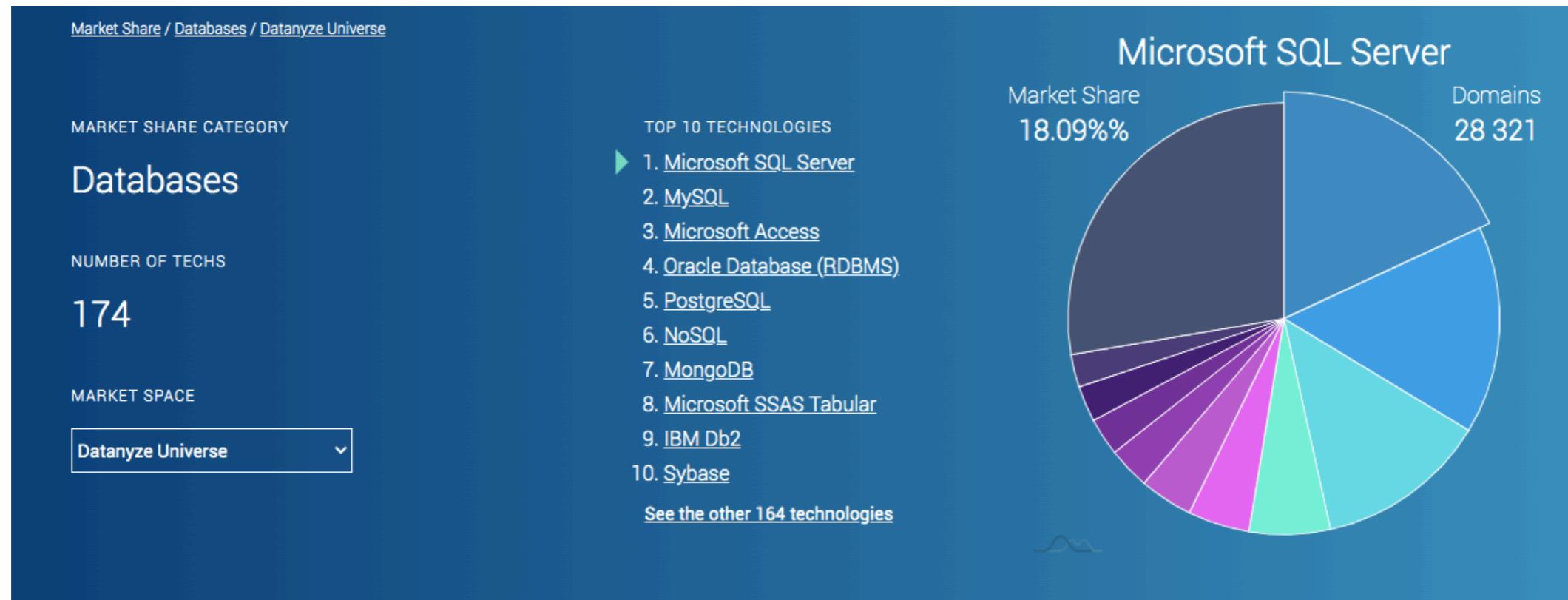
**SYBASE®**  
An  Company

Microsoft®  
**SQL Server®**



  
**MySQL®**

# Немного статистики



Данные с <https://www.datanyze.com/market-share/databases--272/Datanyze%20Universe>

# Database and Table



**Database**



**Table**

# Основные операторы



**CREATE** - создать новую базу/таблицу



**SELECT** - отобрать информацию из таблиц



**INSERT** - добавить информацию в таблицу



**UPDATE** - изменить информацию в таблице



**DELETE** - удалить информацию в таблице



**COMMIT** - зафиксировать внесенные изменения



**ROLLBACK** - откатить внесенные изменения

# CREATE

## 1. CREATE DATABASE

Оператор CREATE DATABASE используется для создания новой базы данных SQL.

```
CREATE DATABASE databasename;
```

## 2. CREATE TABLE

Оператор CREATE TABLE используется для создания новой таблицы в базе данных.

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
    ...
);
```

# PRIMARY KEY

**PRIMARY KEY** (Первичный ключ) — это параметр, который устанавливается для однозначной идентификации той или иной записи в таблице. Значения PRIMARY KEY должны быть всегда **уникальны**, а так же не содержать значений **NULL**.

```
CREATE TABLE table_name (
    column1 datatype NOT NULL PRIMARY KEY,
    column2 datatype,
    column3 datatype,
    .... );
```

**Примечание:** Лучше если каждая таблица будет иметь PRIMARY KEY, по которому можно однозначно идентифицировать записи в ней.

# FOREIGN KEY

**FOREIGN KEY** (Внешний ключ) - это поле (или набор полей) в одной таблице, которое ссылается на Первичный ключ (PRIMARY KEY) в другой таблице.

```
CREATE TABLE table_name2 (
    column3 datatype NOT NULL PRIMARY KEY,
    column4 datatype,
    column1 datatype FOREIGN KEY REFERENCES table_name1(column2);
```

```
CREATE TABLE Orders (
    OrderID int NOT NULL PRIMARY KEY,
    OrderNumber int NOT NULL,
    PersonID int FOREIGN KEY REFERENCES Persons(PersonID)
);
```

"Persons" table:

PersonID	LastName	FirstName	Age
1	Hansen	Ola	30
2	Svendson	Tove	23
3	Pettersen	Kari	20

"Orders" table:

OrderID	OrderNumber	PersonID
1	77895	3
2	44678	3
3	22456	2
4	24562	1

# AUTO INCREMENT

MS SQL использует ключевое слово **IDENTITY** для выполнения функции автоматического увеличения.

```
CREATE TABLE table_name (
    column1 datatype IDENTITY(1,1) PRIMARY KEY,
    column2 datatype,
    column3 datatype,
    .... );
```

**Примечание:** чтобы указать, что столбец «*column1*» должен начинаться со значения 10 и увеличиваться на 5, измените его на **IDENTITY (10,5)**.

**Примечание:** при добавлении записи с помощью **INSERT** можно не писать данные в ручную, они автоматически добавятся

# SELECT

Оператор SELECT используется для выбора данных из базы данных.

Возвращенные данные хранятся в таблице результатов, которая называется набором результатов.

```
SELECT column1, column2, ...  
FROM table_name;
```

Здесь *column1, column2, ...* являются именами полей таблицы, из которой вы хотите выбрать данные.

Если вы хотите выбрать все поля, доступные в таблице, используйте следующий синтаксис:

```
SELECT * FROM table_name;
```

# ALTER TABLE

Оператор **ALTER TABLE** используется тогда, когда необходимо внести изменения в структуру уже существующей таблицы.

1. Добавить колонку в таблицу

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

2. Удалить колонку с таблицы

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

3. Создать PRIMARY KEY для столбца *column\_name*, когда таблица уже создана

```
ALTER TABLE table_name
ADD PRIMARY KEY (column_name);
```

# INSERT

Оператор **INSERT INTO** используется для вставки/добавления новых записей в таблицу.

*Написать оператор **INSERT INTO** можно двумя способами.*

*1. Первый способ определяет имена столбцов и значения для вставки:*

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

*2. Если вы добавляете значения для всех столбцов таблицы, вам не нужно указывать имена столбцов в запросе SQL. Однако убедитесь, что порядок значений соответствует порядку столбцов в таблице.*

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

■ **INSERT** - написанием отличается от других типов запросов.

# UPDATE

Оператор **UPDATE** используется для изменения существующих записей в таблице.

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

!!! Будьте осторожны при обновлении записей в таблице! Обратите внимание на *WHERE* в *UPDATE*.

*WHERE* указывает, какие записи должны быть обновлены. Если вы пропустите *WHERE*, все записи в таблице будут обновлены!

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ..
```

# DELETE & DROP

Оператор **DELETE** используется для удаления существующих записей в таблице.

```
DELETE FROM table_name WHERE condition;
```

Можно удалить все строки в таблице, не удаляя таблицу. Это означает, что структура таблицы, атрибуты и индексы не будут повреждены:

```
DELETE FROM table_name
```

Оператор **DROP TABLE** используется для удаления существующей таблицы в базе данных.

```
DROP TABLE table_name;
```

!!!*Будьте осторожны при удалении, для защиты используйте BEGIN TRANSACTION*

# BEGIN TRANSACTION & ROLLBACK & COMMIT

**BEGIN TRANSACTION** - Начинает транзакцию

`BEGIN TRANSACTION tran_name`

or

`BEGIN TRAN tran_name`

**ROLLBACK** - В случае выполнения операции ROLLBACK отменяет операции.

`ROLLBACK TRAN tran_name`

**COMMIT** - Завершает транзакцию. После завершения транзакции не сможете вернуть данные в первоначальное положение.

`COMMIT`

# COMMENT

Комментарии используются для объяснения разделов операторов SQL и для удобства при написании скрипта

1. Однострочные комментарии начинаются с --

```
-- this script will ADD column to table
ALTER TABLE table_name
ADD column_name datatype;
```

2. Многострочные комментарии начинаются с /\* и заканчиваются \*/.  
Любой текст между /\* и \*/ будет проигнорирован в скрипте

```
/*
this script will
ADD column to table
*/
ALTER TABLE table_name
ADD column_name datatype;
```

# Questions?

