

## Как составлять SQL-запросы - подробные примеры

Каждый из нас регулярно сталкивается и пользуется различными базами данных. Когда мы выбираем адрес электронной почты, мы работаем с базой данных. Базы данных используют поисковые сервисы, банки для хранения данных о клиентах и т.д.

Но, несмотря на постоянное использование баз данных, даже для многих разработчиков программных систем остается много «белых пятен» из-за разного толкования одних и тех же терминов. Мы дадим краткое определение основных терминов баз данных перед рассмотрением языка SQL. Итак:

**База данных** – файл или набор файлов для хранения упорядоченных данных и их взаимосвязей. Очень часто базой данных называют систему управления базой данных (СУБД). База данных это только хранилище данных в определенном формате и может работать с различными СУБД.

**Таблица** – представим себе папку, в которой хранятся документы, сгруппированные по определенному признаку, например список заказов за последний месяц. Это и есть таблица в компьютерной базе данных. Каждая таблица имеет уникальное имя.

**Тип данных** – вид информации, разрешенной для хранения в отдельном столбце или строке. Это могут быть числа или текст определенного формата.

**Столбец и строка** – все мы работали с электронными таблицами, в которых также присутствуют строки и столбцы. Любая реляционная база данных работает с таблицами аналогичным образом. строки иногда называют *записями*.

**Первичный ключ** – каждая строка таблицы может иметь один или несколько столбцов для ее уникальной идентификации. Без первичного ключа очень трудно производить обновление, изменение и удаление нужных строк.

### Что такое SQL?

Язык программирования SQL (англ. Structured Query Language - язык структурированных запросов) был разработан только для работы с базами данных и в настоящий момент является стандартом для всех популярных СУБД. Синтаксис языка состоит из небольшого количества операторов и прост в изучении. Но, несмотря на внешнюю простоту он позволяет выполнять сложные запросы и операции с БД любого размера.

С 1992 г. существует общепринятый стандарт называемый ANSI SQL. Он определяет базовый синтаксис и функции операторов и поддерживается всеми лидерами рынка СУБД, такими как ORACLE Microsoft SQL Server. Рассмотреть все возможности языка в одной небольшой статье невозможно, поэтому мы кратко рассмотрим только основные операторы SQL:

- типы данных
- создание баз и таблиц
- выборка данных
- добавление записей

- модификация и удаление информации

## Типы данных SQL

Каждый столбец в таблице базы данных имеет свой тип данных. Типы данных в SQL такие же как и в других языках программирования.

Тип данных	Описание
INT	Целые числа
REAL	Числа с плавающей запятой
TEXT	Символьная строка с переменной длиной
DATE	Дата в различных форматах
TIME	Время
CHAR	Текстовые строки фиксированной длины.

## Создание баз данных и таблиц

Создавать новые базы и таблицы можно двумя способами:

- Операторами SQL через консоль СУБД
- Используя интерактивные средства администрирования, входящие в состав сервера баз данных.

Новая база данных создается оператором *CREATE DATABASE <имя базы данных>;*. Как видим синтаксис прост и лаконичен.

Таблицы внутри базы данных создаем оператором *CREATE TABLE* со следующими параметрами:

- наименование таблицы
- имена и типы данных столбцов

В качестве примера создадим таблицу Product со следующими столбцами:

Столбец	Описание
product_id	ID продукта
vendor_id	ID поставщика (ключ внешней таблицы Vendors)
product_name	Наименование продукта
product_price	Стоимость
product_desc	Описание

Создаем таблицу:

```
CREATE TABLE Product
(
product_id CHAR(10) NOT NULL,
vendor_id CHAR(10) NOT NULL,
product_name CHAR(254) NOT NULL,
```

```
product_price DECIMAL(8,2) NOT NULL,  
product_desc VARCHAR(1000) NULL );
```

Таблица состоит из пяти столбцов. После наименования идет тип данных, столбцы разделяются запятыми. Значение столбца может принимать пустое значение (NULL) или нет (NOT NULL) и это определяется при создании таблицы.

### **Выборка данных из таблицы**

Оператор выборки данных SELECT самый часто используемый в SQL. Для получения информации необходимо указать, что мы хотим выбрать и с такой таблицы. Вначале простой пример:

```
SELECT product_name  
FROM Product;
```

После оператора SELECT указываем имя столбца для получения информации, а FROM определяет таблицу.

Результатом выполнения запроса будут все строки таблицы со значениями *product\_name* в том порядке, в котором они были внесены в базу данных т.е. без всякой сортировки. Для упорядочивания результата используется дополнительный оператор ORDER BY.

Для запроса по нескольким полям перечисляем их через запятую как в следующем примере:

```
SELECT product_id, product_name, product_price  
FROM Product;
```

Есть возможность получить как результат запроса значение всех столбцов строки. Для этого используется знак «\*»:

```
SELECT *  
FROM Product;
```

Дополнительно SELECT поддерживает:

- Сортировку данных (оператор ORDER BY)
- Выбор согласно условиям (WHERE)
- Группировку строк (GROUP BY)

### **Добавляем строку**

Для добавления строки в таблицу используется оператор INSERT. Добавление может производиться тремя способами:

- добавляем новую целую строку
- часть строки

- результаты запроса

Для добавления полной строки необходимо указать имя таблицы и значения столбцов(полей) новой строки. Приведем пример

```
INSERT INTO Product  
VALUES('106 '  
50'  
Coca-Cola ,  
1.68,  
No Alcogol ,  
);
```

Пример добавляет в таблицу новый товар. Значения указываются после VALUES для каждого столбца. Если нет соответствующего значения для столбца, то необходимо указывать NULL. Столбцы заполняются значениями в порядке, указанном при создании таблицы.

В случае добавления только части строки необходимо явно указать наименования столбцов, как в примере:

```
INSERT INTO Product (product_id,  
vendor_id,  
product_name)  
VALUES('106 '  
'50'  
Coca-Cola,  
);
```

Мы ввели только идентификаторы товара, поставщика и его наименование, а остальные поля оставили пустыми.

### **Добавление результатов запроса**

В основном INSERT используется для добавления строк, но может использоваться и для добавления результатов оператора SELECT.

### **Изменение данных**

Для изменения данных полей таблицы необходимо использовать оператор UPDATE. Оператор может применяться двумя способами:

- Обновляются все строки в таблице
- Только определенной строки

UPDATE состоит из трех основных элементов:

- таблица, в которой необходимо производить изменения
- имена полей и их новые значения

- условия выбора строк для изменения.

Рассмотрим пример. Допустим, у товара с ID=106 изменилась стоимость, поэтому эту строку необходимо обновить. Пишем следующий оператор:

```
UPDATE Product
SET product_price = '3.2'
WHERE product_id = '106';
```

Мы указали имя таблицы, в нашем случае *Product*, где будет производиться обновление, затем после SET новое значение столбца и нашли нужную запись, указав в WHERE нужное значение ID.

Для изменения нескольких столбцов после оператора SET указываются несколько пар столбец-значение разделенных запятыми. Смотрим пример в котором обновляется наименование и цена товара:

```
UPDATE Product
SET product_name='Fanta', product_price = '3.2'
WHERE product_id = '106';
```

Для удаления информации в столбце можно присвоить ему значение NULL, если это позволяет структура таблицы. Необходимо помнить, что NULL это именно «никакое» значение а не ноль в виде текста или числа. Удалим описание товара:

```
UPDATE Product
SET product_desc = NULL
WHERE product_id = '106';
```

### Удаление строк

Для удаления строк в таблице предназначен оператор DELETE. Есть два варианта использования:

- в таблице удаляются определенные строки;
- удаляются все строки в таблице.

Пример удаления одной строки из таблицы:

```
DELETE FROM Product
WHERE product_id = '106';
```

После *DELETE FROM* указываем имя таблицы, в которой будут удаляться строки. Оператор WHERE содержит условие, по которому будут выбираться строки для удаления. В примере мы удаляем строку товара с ID=106. Указывать WHERE очень важно т.к. пропуск этого оператора приведет к удалению всех строк в таблице. Это относится и к изменению значения полей.

В операторе DELETE не указываются наименования столбцов и метасимволы. Он полностью удаляет строки, а удалить отдельный столбец он не может.

### **Использование SQL в Microsoft Access**

Microsoft Access обычно используется в интерактивном режиме для создания таблиц, баз данных, для управления, изменения и анализа данным в базе данных, но в Access еще присутствует интерактивный конструктор запросов (Query Designer), используя который можно построить и немедленно выполнить операторов SQL любой сложности.

Также поддерживается режим доступа к серверу, при котором СУБД Access может использоваться как генератор SQL-запросов к любому ODBC источнику данных. Это возможность позволяет приложениям Access взаимодействовать с базами данных любого формата.