

# **ОПЕРАТОР ВЫБОРКИ SELECT**

Ключевое слово **SELECT** сообщает базе данных, что данное предложение является запросом на *извлечение* информации.

После слова **SELECT** через запятую перечисляются *наименования полей* (список атрибутов), содержимое которых запрашивается

Обязательным ключевым словом в предложении-запросе **SELECT** является слово **FROM** (из). За ключевым словом **FROM** указывается список разделенных запятыми имен таблиц, из которых извлекается информация

## Пример .

Представить клиентов с указанием номера телефона и количества заказанного товара.

## Вариант1.

```
SELECT Client.surname, Client.phone, Order.qty  
FROM Client INNER JOIN Order  
ON Client.id=Order.id_client;
```

## Вариант2.

```
FROM Client, Order  
WHERE Client.id=Order.id_client;
```

## Синтаксис

---

**SELECT** [**DISTINCT**] <список атрибутов> - запрос на извлечение информации

**FROM** <список таблиц>- список имен таблиц, из которых извлекается информация.

[**WHERE** <условие выборки>]- определяет, какие записи указанных таблиц должны быть выбраны

Предложение **WHERE** используется для указания условий.

При указании условий можно использовать операторы **AND** или **OR**.

Оператор **WHERE** также может использоваться с запросом **DELETE** и **UPDATE**

## Пример .

Выбрать клиентов с количеством товара в заказе, превышающем 100.

```
SELECT Client.surname, Order.qty  
FROM Client, Order  
WHERE Client.id=Order.id_client AND  
Order.qty>100;
```

С оператором **WHERE**  
можно использовать  
знаки сравнения (<, >, =,  
<>), а также операторы  
**LIKE, IN, NOT IN** и  
**BETWEEN**



Пример.

**Вывести фамилии и телефоны  
клиентов из Москвы.**

**SELECT** surname, phone

**FROM** client

**WHERE** address **LIKE** '%Москва%';

**[ORDER BY <список атрибутов>]** – определяет порядок записей в выходной таблице.

Обеспечивает сортировку данных по определенному столбцу в порядке возрастания или убывания.

По умолчанию сортировка выполняется в порядке возрастания, но мы также можем явно указать способ сортировки с помощью ключевых слов **ASC** (по возрастанию) и **DESC** (по убыванию)

# Оператор **LIMIT**

Оператор **LIMIT** используется в **SELECT** для указания количества отображаемых строк при выборке информации из таблиц базы данных

```
LIMIT row_count;
```

Если оператору **LIMIT** передается один параметр, то он указывает на количество извлекаемых строк.

Пример:

```
SELECT *  
FROM Product  
LIMIT 3;
```

**Если передается два параметра, то первый параметр устанавливает смещение относительно начала, то есть сколько строк нужно пропустить, а второй параметр также указывает количество извлекаемых строк.**

**Пример:**

**SELECT \***

**FROM Product**

**LIMIT 2, 3;**

Как правило, оператор **LIMIT**  
используется вместе с  
оператором **ORDER BY**

**SELECT \***

**FROM Product**

**ORDER BY name\_prod**

**LIMIT 2, 3;**

Оператор **CONCAT** позволяет вывести информацию из разных полей в одну строку.

Пример.

Представить данные о клиентах в следующем виде: Иванов Олег, тел. 89062356521.

```
SELECT CONCAT(surname, name,  
тел.',phone) AS Dannyae  
FROM Client;
```

# Итоговые(агрегирующие) функции

---

**AVG()** Функция возвращает среднее значение столбца.

**COUNT()** Функция возвращает число строк в столбце.

**MAX()** Функция возвращает самое большое значение в столбце.

**MIN()** Функция возвращает самое маленькое значение в столбце.

**SUM()** Функция возвращает сумму значений столбца.



## Пример.

**Вычислить общее количество продуктов во всех заказах**

```
SELECT Client.name, sum(Order.qty)  
FROM Client, Order  
WHERE Client.id = Order.id_client
```

**В случае использования  
итоговых функций или  
вычисляемых столбцов,  
необходимо столбцам дать  
новые имена.**

**С помощью  
оператора **AS** можно изменить  
название выходного столбца  
или определить его псевдоним.**

## Пример.

**Вычислить общее количество продуктов во всех заказах**

```
SELECT Client.name, sum(Order.qty) AS  
summa  
FROM Client INNER JOIN Order ON  
Client.id = Order.id_client
```

**[GROUP BY <список атрибутов>]** – определяет атрибуты, по которым производится группировка информации.

Пример.

Вычислить общее количество продуктов по заказам каждого клиента

```
SELECT Client.name, sum(Order.qty)
```

```
FROM Client, Order
```

```
WHERE Client.id = Order.id_client
```

```
GROUP BY Client.name
```

**[HAVING <условие>]**- определяет группы, которые войдут в выходную таблицу

Пример.

Вычислить общее количество продуктов по заказам каждого клиента и вывести на экран тех клиентов, общее количество продуктов в заказах больше 10.

```
SELECT Client.name, sum(Order.qty)
```

```
FROM Client, Order
```

```
GROUP BY Client.name
```

```
WHERE Client.id = Order.id_client
```

```
HAVING sum(Order.qty) > 10;
```

## Пример.

Вывести информацию о продуктах с ценой, увеличенной в 2 раза для тех продуктов, которые имеются на складе.

```
SELECT name_prod, qty, price*2 AS CEN  
FROM product  
WHERE qty > 0 OR qty IS NOT NULL;
```

**[UNION <выражение с оператором SELECT>];** - используется для объединения выходных данных двух или более SQL-запросов в единое множество строк и столбцов.

Оператор UNION используется для объединения наборов результатов из 2 или более SELECT предложений. Он удаляет повторяющиеся строки между различными предложениями SELECT

**Каждое предложение SELECT в операторе UNION должно иметь одинаковое количество полей в наборах результатов с одинаковыми типами данных**