

# ЯЗЫК **SQL**

Команда выборки

# Синтаксис команды выборки

```
SELECT [ALL | DISTINCT] <список_вывода> [INTO <Ид_нов_таб>]  
FROM <источник_выборки>  
[ WHERE <условие> ]  
[ GROUP BY <список_группировки>]  
[ HAVING <условие> ]  
[ORDER BY <имя_столбца> [ASC | DESC],... ]
```

## Формат списка вывода

Выражение [ as <имя поля результирующей таблицы>]

# Пример использования ключа **DISTINCT**

ФИО	Номер_отдела	Зарплата	Рейтинг
Иващенко	1	1200000	2
Потапов	2	1000000	1
Бабкина	2	1200000	2
Филипова	1	1200000	1

**SELECT Рейтинг, Зарплата  
FROM Сотрудники;**

Зарплата	Рейтинг
1200000	2
1000000	1
1200000	2
1200000	1

**SELECT DISTINCT Рейтинг,  
Зарплата FROM Сотрудники;**

Зарплата	Рейтинг
1000000	1
1200000	2
1200000	1

# Пример использования фильтрации

ФИО	Номер_отдела	Зарплата	Рейтинг
Иващенко	1	1300000	2
Потапов	2	1000000	1
Бабкина	2	1400000	2
Филипова	1	1200000	1

**SELECT ФИО, Зарплата FROM Сотрудники  
WHERE ((Зарплата >=1200000) AND (Зарплата <=1300000)) OR (Рейтинг = 1);**

**SELECT ФИО, Зарплата FROM Сотрудники  
WHERE Зарплата BETWEEN 1200000 AND 1300000 OR (Рейтинг = 1) ;**

ФИО	Зарплата
Иващенко	1300000
Потапов	1000000
Филипова	1200000

# Пример использования операции **LIKE**

<u>Заказчики</u>	
Имя_заказчика	Адрес
Афанасьев И.В.	пр. Победы 74-145
Павлов П.Ю.	ул. Чкалова 54-34
Кирилов С.Е.	пл. Победы 33-5

```
SELECT Имя_заказчика  
FROM Заказчики WHERE Адрес LIKE '*Победы*';
```

Имя_заказчика
Афанасьев И.В.
Кирилов С.Е.

# Пример использования вычисляемого поля

Сотрудники		
ФИО	Номер_отдела	Зарплата
Белова	1	2000000
Иванов	1	1600000
Бабкина	2	2000000

```
SELECT ФИО, round(Зарплата/2800,2) as [Зарплата в $]  
FROM Сотрудники  
WHERE Номер_отдела =1;
```

ФИО	Зарплата в \$
Белова	714,29
Иванов	571,43

# Источники выборки

Тип источника выборки	Синтаксис раздела FROM
Таблица	<Ид_таблицы>
Объединение таблиц декартовым произведением	<Ид_таблицы1>, <Ид_таблицы2>
Объединение таблиц внутренним объединением	<Ид_таблицы1> INNER JOIN <Ид_таблицы2> ON выражение
Объединение таблиц внешним объединением	<Ид_таблицы1> LEFT RYGHT JOIN <Ид_таблицы2> ON выражение
Подзапрос	(Текст_подзапроса <Ид_подзапроса>)

<Ид\_таблицы1>.<Ид\_поля> – полное имя поля

[Идентификатор с разделителем] – синтаксис идентификатора, содержащего разделительный символ (пробел)

\* – обозначает полный список полей таблицы

# Объединение декартовым произведением

Отделы	
Номер_отдела	Наименование
1	№1
2	№2

Сотрудники		
Табельный_номер	ФИО	Номер_отдела
1	Белова	1
7	Иванов	1
8	Сидоров	2

```
SELECT Отделы.Наименование, Сотрудники.ФИО  
FROM Отделы, Сотрудники;
```

Наименование	ФИО
№1	Белова
№2	Белова
№1	Иванов
№2	Иванов
№1	Сидоров
№2	Сидоров

# Внутреннее объединение

Отделы	
Номер_отдела	Наименование
1	№1
2	№2
3	№3

Сотрудники		
Табельный_номер	ФИО	Номер_отдела
2	Новиков	
4	Иващенко	
1	Белова	1
7	Иванов	1
8	Сидоров	2

```
SELECT Отделы.Наименование, Сотрудники.ФИО  
FROM Отделы INNER JOIN Сотрудники ON Отделы.Номер_отдела = Сотрудники.  
Номер_отдела;
```

Наименование	ФИО
№1	Белова
№1	Иванов
№2	Сидоров

# Левое внешнее объединение

Отделы	
Номер_отдела	Наименование
1	№1
2	№2
3	№3

Сотрудники		
Табельный_номер	ФИО	Номер_отдела
2	Новиков	
4	Иващенко	
1	Белова	1
7	Иванов	1
8	Сидоров	2

```
SELECT Отделы.Наименование, Сотрудники.ФИО  
FROM Отделы LEFT JOIN Сотрудники ON Отделы.Номер_отдела = Сотрудники.  
Номер_отдела;
```

Наименование	ФИО
№1	Белова
№1	Иванов
№2	Сидоров
№3	

# Правое внешнее объединение

Отделы	
Номер_отдела	Наименование
1	№1
2	№2
3	№3

Сотрудники		
Табельный_номер	ФИО	Номер_отдела
2	Новиков	
4	Иващенко	
1	Белова	1
7	Иванов	1
8	Сидоров	2

```
SELECT Отделы.Наименование, Сотрудники.ФИО  
FROM Отделы RIGHT JOIN Сотрудники ON Отделы.Номер_отдела = Сотрудники.  
Номер_отдела;
```

Наименование	ФИО
№1	Белова
№1	Иванов
№2	Сидоров
	Новиков
	Иващенко

## Объединение трёх таблиц

Сотрудники		
Табельный_номер	ФИО	Номер_отдела
1	Белова	1
7	Иванов	1
8	Сидоров	2

Исполнители	
Табельный_номер	Номер_контракта
7	1
7	2
1	3
8	4
1	5

Заказы		
Номер_контракта	Сумма	Дата
1	540,75	13.02.2009
2	588	10.06.2008
3	52,5	02.06.2008
4	22,05	19.07.2008
5	24,15	25.07.2008

```
SELECT Сотрудники.Табельный_номер, Сотрудники.ФИО, Исполнители.  
Табельный_номер, Исполнители.Номер_контракта, Заказы.Номер_контракта,  
Заказы.Дата  
FROM  
Заказы INNER JOIN  
(Сотрудники INNER JOIN Исполнители ON Сотрудники.Табельный_номер =  
Исполнители.Табельный_номер)  
ON Заказы.Номер_контракта = Исполнители.Номер_контракта;
```

## Объединение трёх таблиц

Сотрудники		
Табельный_номер	ФИО	Номер_отдела
1	Белова	1
7	Иванов	1
8	Сидоров	2

Исполнители	
Табельный_номер	Номер_контракта
7	1
7	2
1	3
8	4
1	5

Заказы		
Номер_контракта	Сумма	Дата
1	540,75	13.02.2009
2	588	10.06.2008
3	52,5	02.06.2008
4	22,05	19.07.2008
5	24,15	25.07.2008

ФИО	Сотрудники. Табельный_номер	Исполнители. Табельный_номер	Исполнител и. Номер_контракта	Заказы. Номер_контракта	Дата
Иванов	7	7	1	1	13.02.2009
Иванов	7	7	2	2	10.06.2008
Белова	1	1	3	3	02.06.2008
Сидоров	8	8	4	4	19.07.2008
Белова	1	1	5	5	25.07.2008

# Группировка данных

**<список\_группировки>** - состоит из выражений по значениям которых будет проводиться группировка.

**Агрегатные функции** используют в качестве аргумента последовательность значений.

**AVG(<список значений>)** - среднее

**COUNT(<список значений>)** или **COUNT (\*)** – число значений

**MAX(<список значений>)** - максимальное из всех значений

**MIN(<список значений>)** - минимальное из всех значений

**SUM(<список значений>)** - сумма всех значений

При наличие группировки агрегатные функции получают в качестве аргумента значения вычисляются для каждой группы.

Раздел **Having** содержит фильтрующие логическое выражение, которое накладывается на сгруппированную таблицу.

В данном разделе можно использовать агрегатные функции.

## Пример: одиночная группировка

Наименование	Табельный_номер
Коммерческий	12
Коммерческий	11
Маркетинговый	9
Коммерческий	8
Транспортный	7
Транспортный	1

```
SELECT Наименование, Count(Табельный_номер) AS Количество  
FROM Отделы INNER JOIN Сотрудники ON  
Отделы.Номер_отдела = Сотрудники.Номер_отдела  
GROUP BY Наименование ;
```

Наименование	Количество
Коммерческий	3
Маркетинговый	1
Транспортный	2

## Пример: использование Having

Сумма	ФИО	Номер_контракта
650000	Васильев В.А.	41
650000	Петров Е.А.	41
800000	Белова Е.А.	40
800000	Белова Е.А.	42
800000	Павлов П.А.	42
950000	Белова Е.А.	5
950000	Иванов С.В.	5
1500000	Сидоров П.П.	2

ФИО	Сумма
Белова Е.А.	4150000
Васильев В.А.	4750000
Иванов С.В.	7050000
Павлов П.А.	2900000
Петров Е.А.	650000
Сидоров П.П.	7600000

```

SELECT ФИО, sum(Сумма) AS Сумма
FROM Заказы INNER JOIN (Сотрудники INNER JOIN Исполнители ON
Сотрудники.Табельный_номер=Исполнители.Табельный_номер) ON
Заказы.Номер_контракта=Исполнители.Номер_контракта
GROUP BY ФИО
HAVING sum(Сумма)> 5000000;
    
```

2100000	Павлов П.А.	38	
2100000	Сидоров П.П.	38	
2500000	Иванов С.В.	1	7050000
2500000	Сидоров П.П.	1	7600000

## Пример: использование Having

Наименование	Рейтинг
Транспортный	100
Транспортный	100
Маркетинговый	100
Коммерческий	24
Коммерческий	120
Коммерческий	50

Наименование	Рейтинг	Count-Рейтинг
Коммерческий	24	1
Коммерческий	50	1
Коммерческий	120	1
Маркетинговый	100	1
Транспортный	100	2

```
SELECT DISTINCT Наименование
FROM Отделы INNER JOIN Сотрудники ON Отделы.Номер_отдела =
Сотрудники.Номер_отдела
GROUP BY Наименование, Рейтинг
HAVING (((Count(Рейтинг))>1));
```

Наименование
--------------

Транспортный
--------------

## Подзапросы

Подзапрос может использоваться внутри основного запроса в качестве:

- Списка
- Константы
- Таблицы

Вывести сведения о всех сотрудниках, чья зарплата превышает среднюю зарплату по первому отделу.

```
SELECT ФИО, Зарплата
FROM Сотрудники
WHERE Зарплата > (
SELECT avg(Зарплата) FROM Сотрудники where Номер_отдела = 1;
);
```

ФИО	Зарплата
Белова Е.А.	2000000
Васильев В.А.	2000000

## Подзапросы

Сколько заработали сотрудники каждого отдела, учитывая что за каждый контракт сотрудник получает 5 % от его стоимости

```
SELECT ФИО, Sum(Сумма*0.05)+Зарплата AS [Sum-Сумма], Сотрудники.  
Номер_отдела AS N  
FROM Сотрудники INNER JOIN (Заказы INNER JOIN Исполнители ON Заказы.  
Номер_контракта=Исполнители.Номер_контракта) ON Сотрудники.  
Табельный_номер=Исполнители.Табельный_номер  
GROUP BY ФИО, Зарплата, Сотрудники.Номер_отдела
```

ФИО	Sum-Сумма	N
Белова Е.А.	2207500	1
Васильев В.А.	2237500	4
Иванов С.В.	1952500	1
Павлов П.А.	1645000	4
Петров Е.А.	1632500	2

```
SELECT [SubQ].N AS [Номер отдела], Sum(SubQ.[Sum-Сумма]) AS [Общий  
заработок]  
FROM ( Текст подзапроса) AS SubQ  
GROUP BY [SubQ].N;
```

# Псевдоним

Задаёт новое имя таблицы или подзапроса, которое можно использовать в пределах основного запроса

Синтаксис:

**Ид\_таблицы|Текст подзапроса Псевдоним**

Пример:

Вывести имена всех сотрудников с такой же зарплатой, как у Иванова, так чтобы этот запрос.

```
SELECT а.ФИО,а.зарплата, б.ФИО,б.зарплата
```

```
FROM сотрудники as а, сотрудники as б
```

```
WHERE а.зарплата = б.зарплата
```

```
AND б.Табельный_номер = 9
```

```
AND а.ФИО <> б.ФИО
```

а.ФИО	а.зарплата	б.ФИО	б.зарплата
Белова Е.	2000000	Белова Е.А.	2000000
а.ФИО	а. зарплата	б.ФИО	б.зарплата
Петров Е.А.	1600000	Петров Е.	1600000
а.ФИО	а. зарплата	б.ФИО	б. зарплата
Иванов С. В.	1600000	Петров Е. А.	1600000
Петров Е.	1600000	Иванов С.В.	1600000

## Подзапрос в разделе **select**

Условие: подзапрос должен возвращать одно значение для каждой строки.

Количество сотрудников в отделах

**SELECT** Номер\_отдела,

**(select COUNT(Наименование) FROM**

**Сотрудники a WHERE a.Номер\_отдела = b.**

**Номер\_отдела) AS Отдел**

**FROM Отделы AS b;**