

БАЗЫ ДАННЫХ

ВЕЛИКИЙ И УЖАСНЫЙ SELECT

ПРЕДЛОЖЕНИЯ SQL

ВЫБОРКА - SELECT

SELECT [предикат] { * | таблица.* | [таблица.]поле_1
[AS псевдоним_1] [, [таблица.]поле_2 [AS псевдоним_2] [, ...]] }
FROM выражение [, ...] [IN внешняяБазаДанных]
[WHERE...]
[GROUP BY...]
[HAVING...]
[ORDER BY...]
[WITH OWNERACCESS OPTION]

ПРЕДЛОЖЕНИЯ SQL ПРОСТЕЙШИЙ ВАРИАНТ SELECT'a

```
SELECT column_name FROM table_name;
```

ПРЕДЛОЖЕНИЯ SQL

ПРОСТЕЙШИЙ ВАРИАНТ SELECT'a

Persons

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Pettersen	Kari	Storgt 20	Stavanger

```
SELECT LastName, FirstName FROM Persons;
```

LastName	FirstName
Hansen	Ola
Svendson	Tove
Pettersen	Kari

ПРЕДЛОЖЕНИЯ SQL

ПРОСТЕЙШИЙ ВАРИАНТ SELECT'a

Persons

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Pettersen	Kari	Storgt 20	Stavanger

```
SELECT * FROM Persons;
```

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Pettersen	Kari	Storgt 20	Stavanger

ПРЕДЛОЖЕНИЯ SQL

АЛИАСЫ

```
SELECT column AS column_alias FROM table;
```

```
SELECT column FROM table AS table_alias;
```

ПРЕДЛОЖЕНИЯ SQL

АЛИАСЫ

Persons

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Pettersen	Kari	Storgt 20	Stavanger

```
SELECT LastName AS Family, FirstName AS Name FROM Persons;
```

Family	Name
Hansen	Ola
Svendson	Tove
Pettersen	Kari

ПРЕДЛОЖЕНИЯ SQL

АЛИАСЫ

Persons

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Pettersen	Kari	Storgt 20	Stavanger

SELECT LastName, FirstName FROM Persons **AS** Employees

Employees

LastName	FirstName
Hansen	Ola
Svendson	Tove
Pettersen	Kari

ПРЕДЛОЖЕНИЯ SQL

SELECT без повторов

```
SELECT DISTINCT column_name(s) FROM table_name;
```

ПРЕДЛОЖЕНИЯ SQL

SELECT без повторов

"Orders"

Company	OrderNumber
Sega	3412
W3Schools	2312
Trio	4678
W3Schools	6798

```
SELECT Company  
FROM Orders;
```

Company
Sega
W3Schools
Trio
W3Schools

```
SELECT DISTINCT Company  
FROM Orders;
```

Company
Sega
W3Schools
Trio

ПРЕДЛОЖЕНИЯ SQL

SELECT с условием

SELECT column FROM table
WHERE column operator value;

=	Равно
<>	Не равно
>	Больше чем
<	Меньше чем
>=	Больше или равно
<=	Меньше или равно
BETWEEN	Между
LIKE	Похоже на

ПРЕДЛОЖЕНИЯ SQL

SELECT с условием

"Persons"

LastName	FirstName	Address	City	Year
Hansen	Ola	Timoteivn 10	Sandnes	1951
Svendson	Tove	Borgvn 23	Sandnes	1978
Svendson	Stale	Kaivn 18	Sandnes	1980
Pettersen	Kari	Storgt 20	Stavanger	1960

SELECT * FROM Persons **WHERE** City='Sandnes';

LastName	FirstName	Address	City	Year
Hansen	Ola	Timoteivn 10	Sandnes	1951
Svendson	Tove	Borgvn 23	Sandnes	1978
Svendson	Stale	Kaivn 18	Sandnes	1980

ПРЕДЛОЖЕНИЯ SQL

SELECT с условием

 SELECT * FROM Persons WHERE FirstName='Tove';

 SELECT * FROM Persons WHERE FirstName=Tove;

 SELECT * FROM Persons WHERE Year>1965;

 SELECT * FROM Persons WHERE Year>'1965';

ПРЕДЛОЖЕНИЯ SQL

SELECT с условием LIKE

```
SELECT column FROM table  
WHERE column LIKE pattern;
```

ПРЕДЛОЖЕНИЯ SQL

SELECT с условием LIKE

Store_Information

store_name	Sales	Date
LOS ANGELES	\$1500	Jan-05-1999
SAN DIEGO	\$250	Jan-07-1999
SAN FRANCISCO	\$300	Jan-08-1999
BOSTON	\$700	Jan-08-1999

```
SELECT * FROM Store_Information  
WHERE store_name LIKE '%AN%';
```

store_name	Sales	Date
LOS AN GELES	\$1500	Jan-05-1999
SAN FRANCISCO	\$300	Jan-08-1999
SAN DIEGO	\$250	Jan-07-1999

ПРЕДЛОЖЕНИЯ SQL

SELECT с условием LIKE

```
SELECT * FROM Persons WHERE FirstName LIKE 'O%';
```

```
SELECT * FROM Persons WHERE FirstName LIKE '%a';
```

```
SELECT * FROM Persons WHERE FirstName LIKE '%la%';
```

```
SELECT * FROM Persons WHERE FirstName LIKE '*la*';
```

```
SELECT * FROM Persons WHERE FirstName LIKE '%la__ _a';
```

ПРЕДЛОЖЕНИЯ SQL BETWEEN

```
SELECT column_name FROM table_name  
WHERE column_name BETWEEN value1 AND value2;
```

ЗАВИСИТ ОТ КОНКРЕТНОЙ СУБД!

ПРЕДЛОЖЕНИЯ SQL

BETWEEN

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Nordmann	Anna	Neset 18	Sandnes
Pettersen	Kari	Storgt 20	Stavanger
Svendson	Tove	Borgvn 23	Sandnes

```
SELECT * FROM Persons  
WHERE LastName BETWEEN 'Hansen' AND 'Pettersen';
```

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Nordmann	Anna	Neset 18	Sandnes

ПРЕДЛОЖЕНИЯ SQL

BETWEEN

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Nordmann	Anna	Neset 18	Sandnes
Pettersen	Kari	Storgt 20	Stavanger
Svendson	Tove	Borgvn 23	Sandnes

```
SELECT * FROM Persons  
WHERE LastName NOT BETWEEN 'Hansen' AND 'Pettersen';
```

LastName	FirstName	Address	City
Pettersen	Kari	Storgt 20	Stavanger
Svendson	Tove	Borgvn 23	Sandnes

ПРЕДЛОЖЕНИЯ SQL

ORDER BY

Orders

Company	OrderNumber
Sega	3412
ABC Shop	5678
W3Schools	2312
W3Schools	6798

```
SELECT Company, OrderNumber  
FROM Orders  
ORDER BY Company;
```

Company	OrderNumber
ABC Shop	5678
Sega	3412
W3Schools	6798
W3Schools	2312

ПРЕДЛОЖЕНИЯ SQL

ORDER BY

Orders

Company	OrderNumber
Sega	3412
ABC Shop	5678
W3Schools	2312
W3Schools	6798

```
SELECT Company, OrderNumber  
FROM Orders  
ORDER BY Company,  
OrderNumber;
```

Company	OrderNumber
ABC Shop	5678
Sega	3412
W3Schools	6798
W3Schools	2312

Company	OrderNumber
ABC Shop	5678
Sega	3412
W3Schools	2312
W3Schools	6798

ПРЕДЛОЖЕНИЯ SQL

ORDER BY

Orders

Company	OrderNumber
Sega	3412
ABC Shop	5678
W3Schools	2312
W3Schools	6798

```
SELECT Company, OrderNumber  
FROM Orders  
ORDER BY Company DESC;
```

Company	OrderNumber
ABC Shop	5678
Sega	3412
W3Schools	6798
W3Schools	2312

Company	OrderNumber
W3Schools	6798
W3Schools	2312
Sega	3412
ABC Shop	5678

ПРЕДЛОЖЕНИЯ SQL

ORDER BY

Orders

Company	OrderNumber
Sega	3412
ABC Shop	5678
W3Schools	2312
W3Schools	6798

```
SELECT Company, OrderNumber
FROM Orders
ORDER BY Company DESC,
OrderNumber ASC;
```

Company	OrderNumber
ABC Shop	5678
Sega	3412
W3Schools	2312
W3Schools	6798

Company	OrderNumber
W3Schools	2312
W3Schools	6798
Sega	3412
ABC Shop	5678

ПРЕДЛОЖЕНИЯ SQL

ORDER BY

```
SELECT store_name, Sales, Date  
FROM Store_Information  
ORDER BY 2 DESC
```



ПО ВТОРОМУ
СТОЛБЦУ

ПРЕДЛОЖЕНИЯ SQL

AND & OR

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Svendson	Stephen	Kaivn 18	Sandnes

```
SELECT * FROM Persons  
WHERE FirstName='Tove' AND LastName='Svendson';
```

LastName	FirstName	Address	City
Svendson	Tove	Borgvn 23	Sandnes

ПРЕДЛОЖЕНИЯ SQL

AND & OR

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Svendson	Stephen	Kaivn 18	Sandnes

```
SELECT * FROM Persons  
WHERE firstname='Tove' OR lastname='Svendson' ;
```

LastName	FirstName	Address	City
Svendson	Tove	Borgvn 23	Sandnes
Svendson	Stephen	Kaivn 18	Sandnes

ПРЕДЛОЖЕНИЯ SQL

AND & OR

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Svendson	Tove	Borgvn 23	Sandnes
Svendson	Stephen	Kaivn 18	Sandnes

SELECT * FROM Persons

WHERE (FirstName='Tove' **OR** FirstName='Stephen')
AND LastName='Svendson' ;

LastName	FirstName	Address	City
Svendson	Tove	Borgvn 23	Sandnes
Svendson	Stephen	Kaivn 18	Sandnes

ПРЕДЛОЖЕНИЯ SQL

AND & OR

Store_Information

store_name	Sales	Date
Los Angeles	\$1500	Jan-05-1999
San Diego	\$250	Jan-07-1999
San Francisco	\$300	Jan-08-1999
Boston	\$700	Jan-08-1999

store_name
Los Angeles
San Francisco

```
SELECT store_name FROM Store_Information  
WHERE Sales > 1000 OR (Sales < 500 AND Sales > 275);
```


ПРЕДЛОЖЕНИЯ SQL ФУНКЦИИ

- **AVG** – среднее значение в столбце
- **COUNT** – число значений в столбце
- **MAX** – самое большое значение в столбце
- **MIN** – самое малое значение в столбце
- **SUM** – сумма значений по столбцу

ПРЕДЛОЖЕНИЯ SQL

AVG

Persons

Name	Age
Hansen, Ola	34
Svendson, Tove	45
Pettersen, Kari	19

SELECT **AVG**(Age) FROM Persons

32.67

SELECT **AVG**(Age) FROM Persons WHERE Age>20

39.5

ПРЕДЛОЖЕНИЯ SQL

COUNT

Store_Information

store_name	Sales	Date
Los Angeles	\$1500	Jan-05-1999
San Diego	\$250	Jan-07-1999
Los Angeles	\$300	Jan-08-1999
Boston	\$700	Jan-08-1999

```
SELECT COUNT(store_name) FROM Store_Information;
```

Count(store_name)

4

ПРЕДЛОЖЕНИЯ SQL

COUNT

Store_Information

store_name	Sales	Date
Los Angeles	\$1500	Jan-05-1999
San Diego	\$250	Jan-07-1999
Los Angeles	\$300	Jan-08-1999
Boston	\$700	Jan-08-1999

```
SELECT COUNT(DISTINCT store_name)  
FROM Store_Information;
```

Count(store_name)

3

ПРЕДЛОЖЕНИЯ SQL

MAX

Persons

Name	Age
Hansen, Ola	34
Svendson, Tove	45
Pettersen, Kari	19

```
SELECT MAX(Age) FROM Persons
```

45

ПРЕДЛОЖЕНИЯ SQL

MIN

Persons

Name	Age
Hansen, Ola	34
Svendson, Tove	45
Pettersen, Kari	19

```
SELECT MIN(Age) FROM Persons
```

19

ПРЕДЛОЖЕНИЯ SQL

SUM

Store_Information

store_name	Sales	Date
Los Angeles	\$1500	Jan-05-1999
San Diego	\$250	Jan-07-1999
Los Angeles	\$300	Jan-08-1999
Boston	\$700	Jan-08-1999

SELECT **SUM**(Sales) FROM Store_Information;

\$1500 + \$250 + \$300 + \$700 = \$2750

SUM(Sales)
\$2750

Некоторые функции MS Access

Функция	Описание
AVG(column)	Returns the average value of a column
COUNT(column)	Returns the number of rows (without a NULL value) of a column
COUNT(*)	Returns the number of selected rows
FIRST(column)	Returns the value of the first record in the specified field
LAST(column)	Returns the value of the last record in the specified field
MAX(column)	Returns the highest value of a column
MIN(column)	Returns the lowest value of a column
STDEV(column)	
STDEVP(column)	
SUM(column)	Returns the total sum of a column
VAR(column)	
VARP(column)	

Некоторые функции MS Access

Функция	Описание
UCASE(c)	Converts a field to upper case
LCASE(c)	Converts a field to lower case
MID(c,start[,end])	Extract characters from a text field
LEN(c)	Returns the length of a text field
INSTR(c)	Returns the numeric position of a named character within a text field
LEFT(c,number_of_char)	Return the left part of a text field requested
RIGHT(c,number_of_char)	Return the right part of a text field requested
ROUND(c,decimals)	Rounds a numeric field to the number of decimals specified
MOD(x,y)	Returns the remainder of a division operation
NOW()	Returns the current system date
FORMAT(c,format)	Changes the way a field is displayed
DATEDIFF(d,date1,date2)	Used to perform date calculations

Некоторые функции MS SQL Server

Функция	Описание
AVG(column)	Returns the average value of a column
BINARY_CHECKSUM	
CHECKSUM	
CHECKSUM_AGG	
COUNT(column)	Returns the number of rows (without a NULL value) of a column
COUNT(*)	Returns the number of selected rows
COUNT(DISTINCT column)	Returns the number of distinct results
FIRST(column)	Returns the value of the first record in the specified field (not supported in SQLServer2K)
LAST(column)	Returns the value of the last record in the specified field (not supported in SQLServer2K)
MAX(column)	Returns the highest value of a column
MIN(column)	Returns the lowest value of a column
STDEV(column)	
STDEVP(column)	
SUM(column)	Returns the total sum of a column
VAR(column)	
VARP(column)	

ПРЕДЛОЖЕНИЯ SQL

SELECT IN

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Nordmann	Anna	Neset 18	Sandnes
Pettersen	Kari	Storgt 20	Stavanger
Svendson	Tove	Borgvn 23	Sandnes

SELECT * FROM Persons WHERE LastName **IN** ('Hansen','Pettersen');

LastName	FirstName	Address	City
Hansen	Ola	Timoteivn 10	Sandnes
Pettersen	Kari	Storgt 20	Stavanger

где, *expression* — любое символьное выражение
pattern — шаблон, по которому будет происходить проверка
выражения *expression*. Шаблон может включать в себя
следующие спец. символы:

Символ	Описание	Примеры
%	Строка любой длины	Пример 1
—	Любой одиночный символ	Пример 2
[]	Диапазон или последовательность символов	Пример 3
[^]	Исключающий диапазон или последовательность символов	Пример 4

Примеры оператора SQL LIKE

ID	UniversityName	Students	Faculties	Professores	Location	Site
1	Perm State National Research University	12400	12	1229	Perm	psu.ru
2	Saint Petersburg State University	21300	24	13126	Saint-Petersburg	spbu.ru
3	Novosibirsk State University	7200	13	1527	Novosibirsk	nsu.ru
4	Moscow State University	35100	39	14358	Moscow	msu.ru
5	Higher School of Economics	20335	12	1615	Moscow	hse.ru
6	Ural Federal University	57000	19	5640	Yekaterinburg	urfu.ru
7	National Research Nuclear University	8600	10	936	Moscow	mephi.ru

```
SELECT * FROM Universities WHERE Site LIKE '____.ru'
```

ID	UniversityName	Students	Faculties	Professores	Location	Site
2	Saint Petersburg State University	21300	24	13126	Saint-Petersburg	spbu.ru
6	Ural Federal University	57000	19	5640	Yekaterinburg	urfu.ru

**SELECT * FROM Universities
WHERE Site LIKE '[k-o]%'**

ID	UniversityName	Students	Faculties	Professores	Location	Site
3	Novosibirsk State University	7200	13	1527	Novosibirsk	nsu.ru
4	Moscow State University	35100	39	14358	Moscow	msu.ru
7	National Research Nuclear University	8600	10	936	Moscow	mephi.ru


```
SELECT * FROM Universities
WHERE Location LIKE '_[^e-s]%'
```

ID	UniversityName	Students	Faculties	Professores	Location	Site
2	Saint Petersburg State University	21300	24	13126	Saint-Petersburg	spbu.ru
7	National Research Nuclear University	8600	10	936	Moscow	mephi.ru

```
SELECT supplier_city, supplier_state  
FROM suppliers  
WHERE supplier_name = 'Intel'  
ORDER BY supplier_city DESC, supplier_state ASC;
```

возвращает все отсортированные записи по полю supplier_city в порядке убывания, а по полю supplier_state в порядке возрастания

```
SELECT * FROM mytable
```

```
ORDER BY column1 ASC, column2 DESC,  
column3 ASC
```

Первый столбец по возрастанию, второй по убыванию, третий опять по возрастанию.

Запрос упорядочит строки по первому столбцу, затем, **не разрушая первого правила**, по второму столбцу. Затем, так же, **не нарушая имеющихся правил**, по