

Язык SQL

Инструкция SELECT

Инструкция SELECT

- *Назначение инструкции* – запрос данных из таблицы, некоторая логическая обработка и возврат результата
- Логическая обработка запроса – абстрактная процедура, которая в языке SQL определяется как необходимость обработки запроса и получения конечного результата
- СУБД MS SQL Server может не следовать строго логической обработке запроса, применяя оптимизацию в ходе физической обработки запроса

Таблица Orders

orderid	empid	custid	orderts	qty
2	1	2	2013-09-29 13:23:02.213	100
3	1	71	2013-10-01 00:00:00.000	10000
4	1	3	2013-10-06 13:18:01.000	234567
5	1	4	2013-10-07 00:00:00.000	567890
6	1	71	2012-10-04 00:00:00.000	1234567
7	2	71	2011-09-01 00:00:00.000	34567890
8	2	71	2011-10-10 00:00:00.000	3456
9	1	71	2013-09-07 00:00:00.000	1233

orderid	empid	custid	orderts	qty
2	1	2	2013-09-29 13:23:02.213	100
3	1	71	2013-10-01 00:00:00.000	10000
4	1	3	2013-10-06 13:18:01.000	234567
5	1	4	2013-10-07 00:00:00.000	567890
6	1	71	2012-10-04 00:00:00.000	1234567
7	2	71	2011-09-01 00:00:00.000	34567890
8	2	71	2011-10-10 00:00:00.000	3456
9	1	71	2013-09-07 00:00:00.000	1233

USE testdb;

```

SELECT empid, YEAR(orderts) AS orderyear,
       COUNT(*) AS numorders
FROM dbo.Orders
WHERE custid = 71
GROUP BY empid, YEAR(orderts)
HAVING COUNT(*) > 1
ORDER BY empid, orderyear;

```

Логическая обработка запроса

1. FROM	1. FROM dbo.Orders
2. WHERE	2. WHERE custid = 71
3. GROUP BY	3. GROUP BY empid, YEAR(orderts)
4. HAVING	4. HAVING COUNT(*) > 1
5. SELECT	5. SELECT empid, YEAR(orderts) AS
6. ORDER BY	orderyear, COUNT(*) AS numorders
	6. ORDER BY empid, orderyear

Элемент FROM

FROM `dbo.Orders`

В элементе **FROM** задаются имена таблиц, из которых нужно достать данные и табличные операции, которые нужно применить к этим таблицам

Пример

```
SELECT orderid, empid, custid,  
       YEAR(orderts) AS orderyear, qty  
FROM dbo.Orders
```

	orderid	empid	custid	orderyear	qty
1	2	1	2	2013	100
2	3	1	71	2013	10000
3	4	1	3	2013	234567
4	5	1	4	2013	567890
5	6	1	71	2012	1234567
6	7	2	71	2011	34567890
7	8	2	71	2011	3456
8	9	1	71	2013	1233

Элемент WHERE

FROM `dbo.Orders`

WHERE `custid = 71`

- В условии **WHERE** задается логическое выражение для отбора строк, возвращаемых **FROM**
- Стадия **WHERE** передаст в последующую стадию логической обработки запроса только те строки, для которых вычисляемое логическое выражение равно **TRUE**

Пример

```
SELECT empid, YEAR(orderts) AS orderyear  
FROM dbo.Orders  
WHERE custid = 71
```

	empid	orderyear
1	1	2013
2	1	2012
3	2	2011
4	2	2011
5	1	2013

Элемент GROUP BY

FROM `dbo.Orders`

WHERE `custid = 71`

GROUP BY `empid, YEAR(orderts)`

- Стадия **GROUP BY** позволяет организовать группы строк, возвращенных предыдущей стадией логической обработки запроса
- Группы определяются элементами, которые задаются в условии **GROUP BY**

	<code>empid</code>	<code>orderyear</code>
1	1	2013
2	1	2012
3	2	2011
4	2	2011
5	1	2013

Пример

```
SELECT empid, YEAR(orderts) AS orderyear ,  
       COUNT(*) AS numorders  
FROM dbo.Orders  
WHERE custid = 71  
GROUP BY empid, YEAR(orderts)
```

	empid	orderyear	numorders
1	2	2011	2
2	1	2012	1
3	1	2013	2

Элемент GROUP BY

- Группирующая функция возвращает одно значение на группу
- Элементы, не включенные в список условия **GROUP BY**, разрешены только как входные параметры для статистических или агрегатных функций

COUNT, SUM, AVG, MIN, MAX

Пример

```
SELECT empid, YEAR(orderts) AS orderyear,  
       SUM(qty) AS totalqty,  
       COUNT(*) AS numorders  
FROM dbo.Orders  
WHERE custid = 71  
GROUP BY empid, YEAR(orderts)
```

	empid	orderyear	totalqty	numorders
1	2	2011	34571346	2
2	1	2012	1234567	1
3	1	2013	11233	2

Пример

```
SELECT empid, YEAR(orderts) AS orderyear,  
       qty,  
       COUNT(*) AS numorders  
FROM dbo.Orders  
WHERE custid = 71  
GROUP BY empid, YEAR(orderts)
```

Столбец "dbo.Orders.qty" недопустим в списке выбора, поскольку он не содержится ни в статистической функции, ни в предложении GROUP BY.

Элемент GROUP BY

- Все агрегатные функции, кроме **COUNT (*)**, игнорируют значения **NULL**

qty	COUNT (*)	COUNT (qty)	COUNT (DISTINCT qty)	SUM(qty)	SUM (DISTINCT qty)
30	5	4	2	60	40
10					
NULL					
10					
10					

AVG (qty) = ? AVG (DISTINCT qty) = ?

Элемент HAVING

- С помощью условия **HAVING** можно задать логическое выражение для выбора групп, а не отдельных строк, которые были отобраны на стадии **GROUP BY**
- В следующую стадию логической обработки запроса стадия **HAVING** передаст только те группы, для которых заданное в элементе **HAVING** логическое выражение равно **TRUE**

```
FROM dbo.Orders  
WHERE custid = 71  
GROUP BY empid, YEAR(orderts)  
HAVING COUNT(*) > 1
```

	empid	orderyear	numorders
1	2	2011	2
2	1	2012	1
3	1	2013	2

Пример

```
SELECT empid, YEAR(orderts) AS orderyear,  
       COUNT(*) AS numorders  
FROM dbo.Orders  
WHERE custid = 71  
GROUP BY empid, YEAR(orderts)  
HAVING COUNT(*) > 1
```

	empid	orderyear	numorders
1	2	2011	2
2	1	2013	2

Элемент SELECT

- В элементе **SELECT** задаются атрибуты (столбцы), которые необходимо получить в результирующей таблице запроса
- В списке столбцов можно использовать выражения (встроенные функции SQL)
- С помощью элемента **AS** можно вводить псевдонимы для результирующих атрибутов

```
SELECT empid, YEAR(orderts) AS orderyear,  
       COUNT(*) AS numorders
```

Пример

```
SELECT empid orders  
FROM dbo.Orders
```

	orders
1	1
2	1
3	1
4	1
5	1
6	2
7	2
8	1

Пример

```
SELECT empid, YEAR(orderts) AS orderyear,  
       COUNT(*) AS numorders  
FROM dbo.Orders  
WHERE custid = 71  
GROUP BY empid, orderyear  
HAVING numorders > 1
```

Недопустимое имя столбца "orderyear".

Недопустимое имя столбца "numorders".

Пример

```
SELECT empid,  
       YEAR(orderts) AS orderyear  
FROM dbo.Orders  
WHERE custid = 71
```

	empid	orderyear
1	1	2013
2	1	2012
3	2	2011
4	2	2011
5	1	2013

```
SELECT DISTINCT empid,  
               YEAR(orderts) AS orderyear  
FROM dbo.Orders  
WHERE custid = 71
```

	empid	orderyear
1	1	2012
2	1	2013
3	2	2011

Пример

SELECT *

FROM dbo.Orders

	orderid	empid	custid	orderts	qty
1	2	1	2	2013-09-29 13:23:02.213	100
2	3	1	71	2013-10-01 00:00:00.000	10000
3	4	1	3	2013-10-06 13:18:01.000	234567
4	5	1	4	2013-10-07 00:00:00.000	567890
5	6	1	71	2012-10-04 00:00:00.000	1234567
6	7	2	71	2011-09-01 00:00:00.000	34567890
7	8	2	71	2011-10-10 00:00:00.000	3456
8	9	1	71	2013-09-07 00:00:00.000	1233

Пример

```
SELECT orderid, YEAR(orderts) AS orderyear,  
       orderyear + 1 AS nextyear  
FROM dbo.Orders
```

Недопустимое имя столбца "orderyear".

```
SELECT orderid, YEAR(orderts) AS orderyear,  
       YEAR(orderts) + 1 AS nextyear  
FROM dbo.Orders
```

	orderid	orderyear	nextyear
1	2	2013	2014
2	3	2013	2014
3	4	2013	2014
4	5	2013	2014
5	6	2012	2013
6	7	2011	2012
7	8	2011	2012
8	9	2013	2014

Элемент ORDER BY

- Позволяет сортировать строки в выходном наборе
- Последний обрабатываемый элемент с точки зрения логической обработки данных

```
SELECT empid,  
       YEAR(orderts) AS orderyear,  
       COUNT(*) AS numorders  
FROM dbo.Orders  
WHERE custid = 71  
GROUP BY empid, YEAR(orderts)  
HAVING COUNT(*) > 1  
ORDER BY empid, orderyear;
```

empid	orderyear	numorders
1	2013	2
2	2011	2

Элемент ORDER BY

- Результат запроса с элементом **ORDER BY** не может интерпретироваться как таблица, т.к. у него гарантированный порядок следования строк
- Запрос с элементом **ORDER BY** возвращает *курсор*, т.е. нереляционный результат с гарантированным порядком следования строк

Пример

```
SELECT empid, YEAR(orderts) AS orderyear,  
       COUNT(*) AS numorders  
FROM   dbo.Orders  
WHERE  custid = 71  
GROUP BY empid, YEAR(orderts)  
HAVING COUNT(*) > 1  
ORDER BY empid DESC, orderyear;
```

empid	orderyear	numorders
1	2013	2
2	2011	2

Пример

```
SELECT empid, YEAR(orderts) AS orderyear,  
       COUNT(*) AS numorders  
FROM   dbo.Orders  
WHERE  custid = 71  
GROUP BY empid, YEAR(orderts)  
HAVING COUNT(*) > 1  
ORDER BY 1, 2;
```

Пример

```
SELECT empid, firstname, lastname, salary
FROM dbo.Employees
ORDER BY hiredate;
```

empid	firstname	lastname	salary
2	qwerty	user	9876544.00
1	abc	abc	1234567890.00

Пример

```
SELECT DISTINCT empid, firstname,  
                lastname, salary  
FROM dbo.Employees  
ORDER BY hiredate;
```

Элементы ORDER BY должны входить в список выбора, если указывается SELECT DISTINCT.

Элемент TOP

- Необязательный элемент, позволяющий ограничить количество или процентную долю строк, возвращаемых запросом

```
SELECT TOP(5) orderid, empid, orderts, custid  
FROM dbo.Orders  
ORDER BY orderts DESC;
```

orderid	empid	orderts	custid
5	1	2013-10-07 00:00:00.000	4
4	1	2013-10-06 13:18:01.000	3
3	1	2013-10-01 00:00:00.000	71
2	1	2013-09-29 13:23:02.213	2

Элемент TOP

```
SELECT TOP (1) PERCENT orderid,  
                        empid, orderts, custid  
FROM dbo.Orders  
ORDER BY orderts DESC;
```

orderid	empid	orderts	custid
5	1	2013-10-07 00:00:00.000	4

Справочные материалы

Справочник по Transact-SQL

[http://technet.microsoft.com/ru-ru/library/bb510741\(v=sql.105\).aspx](http://technet.microsoft.com/ru-ru/library/bb510741(v=sql.105).aspx)