

SQL Задачи и решения

Учебник. Сергей Моисеенко.



◀ назад листать вперед ▶

Язык манипуляции данными в SQL

Оператор SELECT

Простой оператор SELECT стр. 2

Чтобы упорядочить строки результирующего набора, можно выполнить сортировку по любому количеству полей, указанных в предложении **SELECT**. Для этого используется предложение **ORDER BY** список полей, являющееся всегда последним предложением в операторе **SELECT**. При этом в списке полей могут указываться как имена полей, так и их порядковые позиции в списке предложения **SELECT**. Так, если требуется упорядочить результирующий набор по объему оперативной памяти в порядке убывания, можно записать:



Выполнить



Консоль

```
1. SELECT DISTINCT speed, ram
2. FROM PC
3. ORDER BY ram DESC;
```

или



Выполнить



Консоль

```
1. SELECT DISTINCT speed, ram
2. FROM PC
3. ORDER BY 2 DESC;
```

Результат, приведенный ниже, будет, естественно, одним и тем же.

speed	ram
600	128
750	128
900	128
450	64
500	64
450	32
500	32

Сортировку можно проводить по возрастанию (параметр **ASC** принимается по умолчанию) или по убыванию (параметр **DESC**).

Примечание:

Не рекомендуется в приложениях использовать запросы с сортировкой по номерам столбцов. Это связано с тем, что со временем структура таблицы может измениться, например, в результате добавления/удаления столбцов. Как следствие, запрос типа



Выполнить



Консоль

```
1. SELECT *
2. FROM PC
3. ORDER BY 3;
```

может давать совсем другую последовательность или вообще вызывать ошибку, ссылаясь на отсутствующий столбец.

Сортировка по двум полям



Выполнить



Консоль

```
1. SELECT DISTINCT speed, ram
2. FROM PC
3. ORDER BY ram DESC, speed DESC;
```

даст следующий результат:

speed	ram
900	128
750	128
600	128
500	64
450	64
500	32
450	32

Горизонтальную выборку реализует предложение **WHERE** предикат, которое записывается после предложения **FROM**. При этом в результирующий набор попадут только те строки из источника записей, для каждой из которых значение предиката равно **TRUE**. То есть предикат проверяется для каждой записи. Например, запрос «получить информацию о частоте процессора и объеме оперативной памяти для компьютеров с ценой ниже \$500» можно сформулировать следующим образом:

```
1. SELECT DISTINCT speed, ram
2. FROM PC
3. WHERE price < 500
4. ORDER BY 2 DESC;
```

speed	Ram
450	64
450	32
500	32

В последнем запросе был применен предикат сравнения с использованием операции сравнения «<» (меньше чем). Кроме этой операции сравнения могут использоваться: «=» (равно), «>» (больше), «>=» (больше или равно), «<=» (меньше или равно) и «<>» (не равно). Выражения в предикатах сравнения могут содержать константы и любые поля из таблиц, указанных в предложении **FROM**. Символьные строки и константы типа дата/время записываются в апострофах.

Примеры простых предикатов сравнения:

предикат	описание
price < 1000	Цена меньше 1000
type = 'laptop'	Типом продукции является портативный компьютер
cd = '24x'	24-скоростной CD-ROM
color <> 'y'	Не цветной принтер
ram - 128 > 0	Объем оперативной памяти свыше 128 Мбайт
Price <= speed*2	Цена не превышает удвоенной частоты процессора

Рекомендуемые упражнения: 1, 2, 3, 4, 5, 6, 9, 14, 31, 33, 42.

Последние изменения:	
Упражнение 151 (подсказки и решения)	
Предикат LIKE	
Приложение 2. Список задач стр. 2	
Приложение 2. Список задач Футбол	
Приложение 1. Описание учебных баз данных	
Функция STRING_AGG стр. 2	
Оператор UPDATE стр. 2	
Упражнение 151 стр. 4	



Тэги:

поиск по тэгам

ALL AND AUTO_INCREMENT AVG
battles CASE CAST CHAR
CHARINDEX CHECK classes
COALESCE CONSTRAINT Convert
COUNT CROSS APPLY CTE
DATEADD DATEDIFF DATENAME
DATEPART DATETIME DDL
DEFAULT DELETE DISTINCT DML
EXCEPT EXISTS EXTRACT
FOREIGN KEY FROM FULL JOIN
GROUP BY Guadalcanal HAVING
IDENTITY IN
INFORMATION_SCHEMA INNER
JOIN insert INTERSECT IS NOT
NULL IS NULL ISNULL laptop LEFT
LEFT OUTER JOIN LEN maker

[Больше тэгов](#)

Учебник обновлялся
несколько дней назад

©SQL-EX,2008 [Развитие] [Связь] [О проекте] [Ссылки] [Team]
Перепечатка материалов сайта возможна только с разрешения автора.

