

# SQL Задачи и решения

Учебник. Сергей Моисеенко.



◀ назад    листать    вперед ▶

Оператор SELECT

Получение итоговых значений

## Сортировка и NULL-значения стр. 1

Если столбец, по которому выполняется сортировка, допускает **NULL**-значения, то при использовании **SQL Server** следует иметь в виду, что при сортировке по возрастанию NULL-значения будут идти в начале списка, а при сортировке по убыванию - в конце.

Поскольку в доступных в учебнике базах NULL-значения отсутствуют в представленных данных (коль скоро они согласованы с данными в открытых базах, используемых на сайте sql-ex.ru), я создал копию таблицы PC с именем PC\_, в которую добавил строку, содержащую NULL в столбце *price*:

```
1. INSERT INTO PC_
2. VALUES (13,2112,600,64,8,'24x',NULL);
```

Следует отметить, что это не противоречит [схеме данных](#).

Теперь вы сами можете убедиться в сказанном, выполнив пару приведенных ниже запросов.



Выполнить



Консоль

```
1. SELECT * FROM PC_ ORDER BY price;
```



Выполнить



Консоль

```
1. SELECT * FROM PC_ ORDER BY price DESC;
```

Почему это важно? Дело в том, что при поиске экстремальных значений часто используют метод, основанный на сортировке. Рассмотрим, например, такую задачу.

**Найти модели ПК, имеющих минимальную цену.**

Иногда эту задачу решают следующим образом:



Выполнить



Консоль

```
1. SELECT TOP 1 WITH ties model
2. FROM PC_
3. ORDER BY price;
```

Конструкция **WITH TIES** используется для того, чтобы вывести все модели с наименьшей ценой, если их окажется несколько. Однако в результате мы получим модель 2112, цена которой неизвестна, в то время как должны получить модели 1232 и 1260, имеющих действительно минимальные цены. Мы их и получим, если исключим из рассмотрения модели с неизвестными ценами:



Выполнить



Консоль

```
1. SELECT TOP 1 WITH ties model
2. FROM PC_
3. WHERE price IS NOT NULL
4. ORDER BY price;
```

Но тут появляется еще одна проблема, связанная с дубликатами. Поскольку есть два ПК модели 1232 с минимальной ценой, то обе они будут выводиться в результирующем наборе. DISTINCT без указания в списке столбцов предложения SELECT тех, по которым выполняется сортировка, использовать мы не можем, о чем и сообщает ошибка, если мы попытаемся это сделать



Выполнить







Консоль

```
1. SELECT DISTINCT TOP 1 WITH ties model
2. FROM PC_
3. WHERE price IS NOT NULL
4. ORDER BY price;
```

(ORDER BY items must appear in the select list if SELECT DISTINCT is specified.)

Чтобы получить решение в требуемом виде, мы можем добавить *price* в список выводимых столбцов, а потом использовать полученный запрос в качестве подзапроса. Итак,



  Выполнить



  Консоль

```
1. SELECT model FROM (
2. SELECT DISTINCT TOP 1 WITH ties model, price
3. FROM PC_
4. WHERE price IS NOT NULL
5. ORDER BY price
6. ) X;
```

**Примечание:**

При использовании агрегатных функций проблемы с NULL-значениями не возникает, т.к. они автоматически исключаются из рассмотрения. Хотя при этом тоже придется использовать подзапрос:

  Выполнить

  Консоль

```
1. SELECT DISTINCT model FROM PC_
2. WHERE price = (SELECT MIN(price) FROM PC_);
```

Заметим также, что это стандартное решение будет работать под любыми СУБД, т.к. не использует специфических особенностей диалекта.

А как, кстати, обстоят дела с использованием метода на основе сортировки в других СУБД?

Страницы: 1 2 3

SQL Server sql-ex.ru сортировка NULL WITH TIES

 [Предыдущая](#) [Комбинация детализированных и агрегированных данных] [Агрегатная функция от агрегатной функции] [Следующая](#) 

**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

Упражнение 151



**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  
*месяц назад*  
<https://exchangesumo.com/obmen/XL>  
PUMBUAH/ . Как осуществить  
расчет перекрытия?

---

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

