

SQL Задачи и решения

Учебник. Сергей Моисеенко.



◀ назад листать вперед ▶

Новое в стандарте и реализациях языка SQL

CROSS APPLY / OUTER APPLY стр. 1

Оператор **CROSS APPLY** появился в SQL Server 2005. Он позволяет выполнить соединение двух табличных выражений. При этом каждая строка из левой таблицы сочетается с каждой строкой из правой.

Давайте попробуем разобраться в том, какие преимущества дает нам использование этого нестандартного оператора.

Первый пример.

➡  Выполнить

  Консоль

```
1. SELECT * FROM
2. Product
3. CROSS APPLY
4. Laptop;
```

Мы получили просто декартово произведение таблиц Product и Laptop. Аналогичный результат мы можем получить с помощью следующих стандартных запросов:

➡  Выполнить

  Консоль

```
1. SELECT * FROM
2. Product
3. CROSS JOIN
4. Laptop;
```

или

➡  Выполнить

  Консоль

```
1. SELECT * FROM
2. Product, Laptop;
```

Поставим теперь более осмысленную задачу.

Для каждого ноутбука дополнительно вывести имя производителя.

Эту задачу мы можем решить с помощью обычного соединения:

➡  Выполнить

  Консоль

```
1. SELECT P.make, L.* FROM
2. Product P JOIN Laptop L ON P.model= L.model;
```

С помощью CROSS APPLY решение этой же задачи можно написать так:

➡  Выполнить

  Консоль

```
1. SELECT P.make, L.* FROM
2. Product P
3. CROSS APPLY
4. (SELECT * FROM Laptop L WHERE P.model= L.model) L;
```

"И что тут нового?" - спросите вы. Запрос стал даже более громоздким. Пока да, можно согласиться. Но уже здесь можно заметить весьма важную вещь, которая отличает CROSS APPLY от других видов соединений. А именно, мы используем коррелирующий подзапрос в предложении FROM, передавая в него значения из левого табличного выражения. В данном примере это значения из столбца P.model. Т.е. для каждой строки из левой таблицы правая таблица будет своя.

Поняв это, мы можем воспользоваться данными преимуществами. Рассмотрим следующую задачу.

Для каждого ноутбука дополнительно вывести максимальную цену среди ноутбуков того же производителя.



Эту задачу мы можем решить с помощью коррелирующего подзапроса в предложении SELECT:



Выполнить



Консоль

```
1. SELECT *, (SELECT MAX(price) FROM Laptop L2
2. JOIN Product P1 ON L2.model=P1.model
3. WHERE maker = (SELECT maker FROM Product P2 WHERE P2.model= L1.model)) max_price
4. FROM laptop L1;
```

Пока решение, использующее CROSS APPLY, будет мало чем отличаться от вышеприведенного:



Выполнить



Консоль

```
1. SELECT *
2. FROM laptop L1
3. CROSS APPLY
4. (SELECT MAX(price) max_price FROM Laptop L2
5. JOIN Product P1 ON L2.model=P1.model
6. WHERE maker = (SELECT maker FROM Product P2 WHERE P2.model= L1.model)) X;
```

Страницы: 1 2 3 4

SQL Server коррелирующие подзапросы CROSS APPLY



Предыдущая [COUNT DISTINCT и оконные функции]

[Функция CONCAT] Следующая



Последние изменения:

Метод наименьших квадратов

Вставка строк в таблицу, содержащую автоинкрементируемое поле стр. 5

Сортировка по дням рождения стр. 2

Функция STRING_AGG стр. 2

Функция TRANSLATE

Упражнение 151 (подсказки и решения)

Предикат LIKE

Приложение 2. Список задач стр. 2

Приложение 2. Список задач

Футбол



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

