

SQL Задачи и решения

Учебник. Сергей Моисеенко.



◀ назад листать вперед ▶

Оператор SELECT

Получение итоговых значений

Получение итоговых данных с помощью оператора ROLLUP

Посчитаем сумму прихода на каждый из пунктов по таблице Income. Это несложно сделать при помощи запроса



Выполнить



Консоль

```
1. SELECT point, SUM(inc) Qty
2. FROM Income GROUP BY point;
```

Пусть наряду с этим нам требуется вывести сумму по всем пунктам, т.е. результат должен выглядеть так:

point	Qty
1	66500.00
2	13000.00
3	3100.00
ALL	82600.00

Для решения подобной задачи в операторе SELECT имеется спецификация **ROLLUP**. С её помощью достичь требуемого результата не составляет труда:



Выполнить



Консоль

```
1. SELECT CASE WHEN point IS NULL THEN 'ALL' ELSE CAST(point AS varchar) END point,
2. SUM(inc) Qty
3. FROM Income GROUP BY point WITH ROLLUP;
```

Поскольку значения столбца должны быть одного типа, номер пункта приёма приводится к символьному типу.

Последний запрос можно переписать в иной (стандартной) синтаксической форме:



Выполнить



Консоль

```
1. SELECT CASE WHEN point IS NULL THEN 'ALL' ELSE CAST(point AS varchar) END point,
2. SUM(inc) Qty
3. FROM Income GROUP BY ROLLUP(point);
```

Вместо ROLLUP в нашем запросе можно также использовать оператор CUBE:



Выполнить



Консоль

```
1. SELECT CASE WHEN point IS NULL THEN 'ALL' ELSE CAST(point AS varchar) END point,
2. SUM(inc) Qty
3. FROM Income
4. GROUP BY point WITH CUBE;
```

Подробнее о различиях между этими двумя операторами вы можете почитать в [статье Бена Ричардсона](#).

Если СУБД не поддерживает конструкцию ROLLUP, можно использовать либо **UNION**, либо внешнее соединение (**FULL JOIN**), что позволяет объединить два запроса в один.

Ниже приводятся эти решения.

UNION



Выполнить



Консоль

```
1. SELECT CAST(point AS varchar) point, SUM(inc) Qty
2. FROM Income GROUP BY point
3. UNION ALL
```

```
4. SELECT 'ALL', SUM(inc)
5. FROM Income;
```



FULL JOIN



Выполнить



Консоль

```
1. SELECT coalesce(X.point,Y.point) point, coalesce(X.Qty,Y.Qty) Qty FROM
2. (SELECT CAST(point AS varchar) point, SUM(inc) Qty
3. FROM Income GROUP BY point) X
4. FULL JOIN
5. (SELECT 'ALL' point, SUM(inc) Qty
6. FROM Income) Y ON 1 = 2;
```

В последнем решении следует обратить внимание на то, что соединение выполняется по заведомо ложному предикату, т.к. нам нужны строки из обеих таблиц, которые бы не конкатенировались друг с другом.

Рекомендуемые упражнения: **120**

группировка FULL JOIN UNION ALL ROLLUP CUBE



Предыдущая [Предложение HAVING]

[Комбинация детализированных и агрегированных данных]

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

продажа евразийский банк .

Особенности укладки водяного

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

