SQL Задачи и решения

Учебник. Сергей Моисеенко.

Предложение GROUP BY

Предложение **GROUP BY** используется для определения групп выходных строк, к которым могут применяться агрегатные функции (**COUNT**, **MIN**, **MAX**, **AVG** и **SUM**). Если это предложение отсутствует, и используются агрегатные функции, то все столбцы с именами, упомянутыми в **SELECT**, должны быть включены в агрегатные функции, и эти функции будут применяться ко всему набору строк, которые удовлетворяют предикату запроса. В противном случае все столбцы списка **SELECT**, не вошедшие в агрегатные функции, должны быть указаны в предложении **GROUP BY**. В результате чего все выходные строки запроса разбиваются на группы, характеризуемые одинаковыми комбинациями значений в этих столбцах. После чего к каждой группе будут применены агрегатные функции. Следует иметь в виду, что для **GROUP BY** все значения **NULL** трактуются как равные, то есть при группировке по полю, содержащему **NULL**-значения, все такие строки попадут в одну группу.

Если при наличии предложения **GROUP BY**, в предложении **SELECT** отсутствуют агрегатные функции, то запрос просто вернет по одной строке из каждой группы. Эту возможность, наряду с ключевым словом **DISTINCT**, можно использовать для исключения дубликатов строк в результирующем наборе.

Рассмотрим простой пример:

Выполнить Консоль

```
1.  SELECT model, COUNT(model) AS Qty_model,
2.       AVG(price) AS Avg_price
3.  FROM PC
4.  GROUP BY model;
```

В этом запросе для каждой модели ПК определяется их количество и средняя стоимость. Все строки с одинаковыми значениями model (номер модели) образуют группу, и на выходе **SELECT** вычисляются количество значений и средняя цена для каждой группы. Результатом выполнения запроса будет следующая таблица

model	Qty_model	Avg_price
1121	3	850
1232	4	425
1233	3	843,333333333333
1260	1	350

Если бы в **SELECT** присутствовал столбец с датой, то можно было бы вычислять эти показатели для каждой конкретной даты. Для этого нужно добавить дату в качестве группирующего столбца, и тогда агрегатные функции вычислялись бы для каждой комбинации значений {модель, дата}.

Существует несколько определенных правил выполнения агрегатных функций.

- Если в результате выполнения запроса не получено ни одной строки (или ни одной строки для данной группы), то исходные данные для вычисления любой из агрегатных функций отсутствуют. В этом случае результатом выполнения функций **COUNT** будет нуль, а результатом всех других функций — **NULL**.
  Данное свойство может дать не всегда очевидный результат. Рассмотрим, например, такой запрос:

  Выполнить Консоль

  ```
  1.  SELECT 1 a WHERE
  2.  EXISTS(SELECT  MAX(price)
  3.            FROM PC
  4.            WHERE price<0);
  ```

  Подзапрос в предикате EXISTS возвращает одну строку с NULL в качестве значения столбца. Поэтому, несмотря на то, что ПК с отрицательными ценами нет в базе данных, запрос в примере вернет 1.

- Аргумент агрегатной функции не может сам содержать агрегатные функции (функция от функции). То есть в простом запросе (без подзапросов) нельзя, скажем, получить максимум средних значений.
- Результат выполнения функции **COUNT** есть целое число (INTEGER). Другие агрегатные функции наследуют типы данных обрабатываемых значений.
- Если при выполнении функции **SUM** будет получен результат, превышающий максимально возможное значение для используемого типа данных, возникает ошибка.

Итак, агрегатные функции, включенные в предложение **SELECT** запроса, не содержащего предложения **GROUP BY**, исполняются над всеми результирующими строками этого запроса. Если же запрос содержит предложение **GROUP BY**, каждый набор строк, который имеет одинаковые значения столбца или группы столбцов, заданных в предложении **GROUP BY**, составляют группу, и агрегатные функции выполняются для каждой группы отдельно.

Рекомендуемые упражнения: **15**, **19**, **20**, **21**, **22**, **28**, **30**, **32**, **40**, **41**, **55**, **56**, **57**, **59**, **63**, **66**, **67**, **68**, **70**, **72**, **74**, **76**, **84**, **86**, **87**, **89**, **102**, **111**, **112**, **113**, **114**, **117**, **119**, **130**

GROUP BY AVG SUM COUNT агрегатные функции группировка

Предыдущая [Получение итоговых значений]

[Предложение HAVING] 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 *несколько дней назад*
https://exchangesumo.com/obmen/PL HCBRUB/ . Книга все об электрике.