IT CAREER HUB

Урок 5.2. Фильтрация агрегированных данных Having

IT CAREER HUB

Фильтрация данных

Фильтрация данных — это процесс отбора строк в наборе данных, соответствующих определенными условиям.

В SQL для фильтрации используются операторы WHERE и HAVING, которые применяются на разных этапах выполнения запроса.

Оператор WHERE используется для фильтрации строк в таблице на этапе выборки данных (до группировки). Он отбирает только те строки, которые удовлетворяют заданным условиям.

Особенности:

- WHERE применяется на этапе выборки данных, то есть до того, как строки будут агрегированы или сгруппированы.
- WHERE не может фильтровать результаты на основе агрегатных функций (например, нельзя использовать WHERE COUNT(*) > 1).

Чтобы отобрать необходимые нам данные после применения GROUP BY, необходимо использовать специальный оператор HAVING.

Оператор HAVING — это оператор, который применяется после оператора GROUP BY и используется для фильтрации результатов агрегатных функций на основе заданных условий.

- Позволяет отобрать только те группы, которые соответствуют определенным условиям.
- GROUP BY может использоваться без HAVING, но HAVING никогда не используется без GROUP BY.

```
Unset

SELECT столбец1, агрегатная_функция(столбец2)

FROM таблица

WHERE условия

GROUP BY столбец1
```

```
HAVING условие_для_группы;
```

столбец1 — столбец, по которому происходит группировка.

агрегатная_функция(столбец2) — агрегатная функция, применяемая к группам.

условие_для_группы — условие, определяющее, какие группы попадут в итоговый результат.

IT CAREER HUB

Задание для закрепления

1. Выбрать supplier_ids для тех поставщиков, у которых количество продуктов больше 2. Используем таблицу products.

```
Unset

SELECT supplier_ids

FROM products

GROUP BY supplier_ids

HAVING COUNT(id) >2
```

Вы можете использовать несколько условий в HAVING.

2. Сгруппировать продукты по standard_cost и list_price.

3. Посчитать количество продуктов и вывести только те данные, где количество продуктов не менее 2.

```
Unset
SELECT standard_cost, list_price, count(product_name)

FROM products

GROUP BY 1,2

HAVING count(product_name) > 1
```

Часто HAVING и WHERE используются вместе, чтобы максимально сузить набор данных перед тем, как применять агрегатные функции и группировку.

4. Выбрать только те продукты в quantity_per_unit встречается слово 'oz' как в нижнем, так и в верхнем регистрах.

5. Сгруппировать по standard_cost.

6. Оставить только данные, где количество продуктов не менее 3.

```
Unset
SELECT standard_cost, COUNT(product_name)

FROM products

WHERE lower(quantity_per_unit) LIKE '%oz%'

GROUP BY 1

HAVING COUNT(product_name) > 2
```

IT CAREER HUB

Особенности HAVING

- Производительность:

  Запросы с HAVING могут быть медленнее, чем запросы с WHERE, поскольку HAVING применяется уже после группировки и вычисления агрегатных функций. Это может привести к обработке большего объёма данных.

- Четкое понимание порядка выполнения:

  HAVING фильтрует данные после всех операций, связанных с выборкой и группировкой, тогда как WHERE работает до них.

Синтаксис SQL-запроса:

```
Unset
SELECT столбцы

FROM таблица

WHERE условия

GROUP BY столбцы

HAVING условия_группировки

ORDER BY столбцы

LIMIT количество;
```

Этот порядок написания операторов логичен с точки зрения структуры запроса, но он не отражает фактический порядок выполнения команд на сервере базы данных.

IT CAREER HUB

Фактический порядок выполнения SQL-запроса

СУБД (Система управления базами данных) обрабатывает запросы в определенном порядке, чтобы получить результат.

Порядок выполнения запросов:

1. **FROM** — Определение исходных данных.
2. **WHERE** — Фильтрация строк данных.
3. **GROUP BY** — Группировка строк данных.
4. **HAVING** — Фильтрация сгруппированных данных.
5. **SELECT** — Выборка столбцов для вывода.
6. **ORDER BY** — Сортировка результата.
7. **LIMIT** — Ограничение количества строк в результате.

- На этапе **FROM** происходит определение таблицы (или таблиц), из которых будут выбраны данные. Выполняется первым: Потому что сначала нужно определить, откуда будут извлекаться данные.

- Оператор **WHERE** фильтрует строки на основе указанных условий. На этом этапе удаляются строки, которые не соответствуют условиям.Выполняется вторым: Потому что перед группировкой (или любой другой обработкой) необходимо отобрать только те строки, которые соответствуют заданным критериям.

- Оператор **GROUP BY** группирует строки, выбранные на предыдущем этапе, по одному или нескольким столбцам. Группировка позволяет применить агрегатные функции (например, SUM, COUNT, AVG и т.д.) к этим группам.Выполняется третьим: После того как данные были отобраны и отфильтрованы, они группируются по указанным столбцам.

- Оператор **HAVING** фильтрует группы, созданные на этапе GROUP BY, на основе условий, которые могут включать агрегатные функции.Выполняется четвертым: Потому что он фильтрует результаты после их группировки.

- Оператор **ORDER BY** сортирует строки результата на основе одного или нескольких столбцов.Выполняется шестым: Потому что сортировка данных возможна только после того, как определён окончательный набор строк, которые нужно отсортировать.

- Оператор **LIMIT** ограничивает количество строк в итоговом результате.Выполняется последним: Потому что ограничивать количество строк можно только после сортировки и выборки данных.