

Database

Фильтрация агрегированных данных Having



Преподаватель

Портрет

Имя Фамилия

Текущая должность

Количество лет опыта

Какой у Вас опыт - ключевые кейсы


Самые яркие проекты


Дополнительная информация по вашему усмотрению


Корпоративный e-mail


Социальные сети (по желанию)


Важно

- 

Камера должна быть включена на протяжении всего занятия
- 

В течение занятия вопросы задавать в чате или когда преподаватель спрашивает, есть ли у Вас вопросы
- 

Вести себя уважительно и этично по отношению к остальным участникам занятия
- 

Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях
- 

Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя

Повторение



Агрегирующие функции Оператор Group BY



Основные агрегирующие функции



Оператор GROUP BY

План занятия

- Фильтрация данных
- Особенности HAVING
- Фактический порядок выполнения SQL-запроса



ОСНОВНОЙ БЛОК





Фильтрация данных



Фильтрация данных

Это процесс отбора строк в наборе данных, соответствующих определенными условиям.

Фильтрация данных



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

Фильтрация данных

Особенности WHERE

Применяется на этапе выборки данных.

Не может фильтровать результаты на основе агрегатных функций.



Оператор HAVING

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

Оператор HAVING



Позволяет отобрать только те группы, которые соответствуют определенным условиям.



GROUP BY может использоваться без HAVING, но HAVING никогда не используется без GROUP BY.

Оператор Having



Определение

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.

Код

```
SELECT столбец1, агрегатная_функция
(столбец2)
FROM таблица
WHERE условия
GROUP BY столбец1
HAVING условие_для_группы;
```



ВОПРОСЫ





ЗАДАНИЕ



Задание

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



Решение

```
SELECT supplier_ids
FROM products
GROUP BY supplier_ids
HAVING COUNT(id) >2
```



Задание

Вы можете использовать несколько условий в HAVING

Сгруппировать продукты по standard_cost и list_price
Посчитать количество продуктов и вывести только те данные, где количество продуктов не менее 2



Решение

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

```
FROM products
```

```
GROUP BY 1,2
```

```
HAVING count(product_name) > 1
```



Задание

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

Выбрать только те продукты в quantity_per_unit встречается слово 'oz' как в нижнем так и в верхнем регистрах Сгруппировать по standard_cost Оставить только данные где количество продуктов не менее 3



Решение

```
SELECT standard_cost, COUNT(product_name)
FROM products
WHERE lower(quantity_per_unit) LIKE '%oz%'
GROUP BY 1
HAVING COUNT(product_name) > 2
```





ВОПРОСЫ



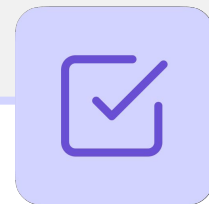
Фильтрация данных

Особенности HAVING

Производительность

Четкое понимание порядка выполнения

Синтаксис SQL-запроса



```
SELECT столбцы
FROM таблица
WHERE условия
GROUP BY столбцы
HAVING условия_группировки
ORDER BY столбцы
LIMIT количество;
```



Фактический порядок выполнения SQL-запроса

Создание и настройка проекта



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.



ВОПРОСЫ



Домашнее задание

1. Посчитайте основные статистики - среднее, сумму, минимум, максимум столбца `unit_cost`.
2. Посчитайте количество уникальных заказов `purchase_order_id`.
3. Посчитайте количество продуктов `product_id` в каждом заказе `purchase_order_id`. Отсортируйте полученные данные по убыванию количества.
4. Посчитайте заказы по дате доставки `date_received`. Считаем только те продукты, количество `quantity` которых больше 30.

Домашнее задание

5. Посчитайте суммарную стоимость заказов в каждую из дат. Стоимость заказа - произведение quantity на unit_cost.
6. Сгруппируйте товары по unit_cost и вычислите среднее и максимальное значение quantity только для товаров, где purchase_order_id не больше 100.
7. Выберите только строки где есть значения в столбце inventory_id.
 - Создайте столбец category - если unit_cost > 20 то 'Expensive' в остальных случаях 'others'.
 - Посчитайте количество продуктов в каждой категории.

Заключение

