1. Простые агрегаты (без GROUP BY)

a) Найдите общую сумму всех продаж. Назовите колонку total\_sales\_sum

**select** **sum** (amount) **as** *total\_sale\_sum* **from** sales\_61

b) Найдите среднюю сумму продажи. Назовите колонку sales\_avg\_amt

**select** **avg** (amount) **as** *sales\_avg\_amt* **from** sales\_61

c) Подсчитайте общее количество продаж. Назовите колонку total\_sales\_cnt

**select** **count** (amount) **as** *total\_sales\_cnt* **from** sales\_61

d) Найдите максимальную сумму продаж. Назовите колонку max\_sales\_amt

**select** **max**(amount) **as** *max\_total\_sales* **from** sales\_61

e) Найдите, когда был самый первый заказ. Нужно вывести дату этого заказа.

**select** **min**(sale\_date) **as** *min\_sale\_date* **from** sales\_61

Группировка данных

1. Сгруппируйте данные по товарам (product\_name) и найдите общую сумму продаж по каждому товару. Нужно вывести product\_name и сумму продаж. Отсортировать по сумме продаж по возрастанию.

**select** product\_name, **sum**(amount) **as** *total\_sale\_sum* **from** sales\_61

**group** **by** product\_name

**order** **by** *total\_sale\_sum* **asc**

1. Сгруппируйте данные по покупателям (customer\_id) и посчитайте количество покупок каждого клиента. Нужно вывести customer\_id и количество покупок. Отсортировать по убыванию customer\_id

**select** customer\_id, **count**(\*) **as** *total\_sales\_cnt* **from** sales\_61

**group** **by** customer\_id

**order** **by** customer\_id **desc**

3. Группировка с фильтрацией

a) Найдите товары, общая сумма продаж которых превышает 10,000. Используйте HAVING. Выведите id товара и сумму продаж.

**select** customer\_id , **sum**(amount) **as** *total\_sale\_sum* **from** sales\_61

**group** **by** customer\_id

**having** **sum**(amount) > 10000

1. Выведите клиентов, которые совершили более 3 покупок. Выведите id клиента и количество покупок. Отсортируйте по количество покупок.

**select** customer\_id, **count**(\*) **as** *total\_sales\_cnt* **from** sales\_61

**group** **by** customer\_id

**having** **count**(\*) > 3

**order** **by** *total\_sales\_cnt*

1. Напишите запрос, который выведет среднюю сумму продаж по каждому покупателю и отсортирует их по убыванию. Нужно вывести id покупателя и среднюю сумму.

**select** customer\_id, **avg**(amount) **as** *sales\_avg\_amount* **from** sales\_61

**group** **by** customer\_id

**order** **by** customer\_id **desc**

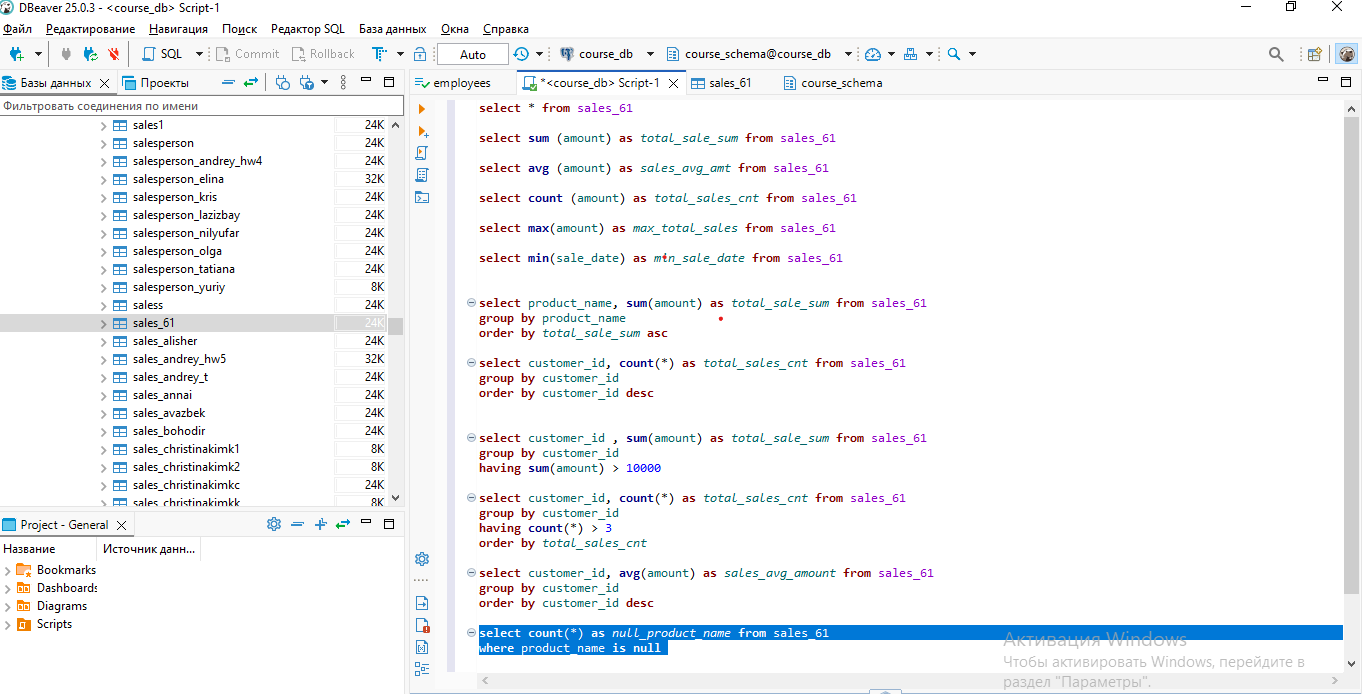
4. Работа с NULL\*

a) Определите, есть ли товары с NULL в названии (product\_name), и как это влияет на подсчет количества продаж.

**select** **count**(\*) **as** *null\_product\_name* **from** sales\_61

**where** product\_name **is** **null**

Null не мешают подсчету; но могут быть исключены из аналитики



**1. Какие агрегатные функции существуют в SQL и в каких случаях их**

**применяют?**

Агрегатные функции: MAX - максимальное значение,

COUNT - подсчет кол-во строк или значений,

AVG - среднее значение,

SUM- сумирует значение,

MIN- минимальное значение

1. **Как работает оператор GROUP BY в SQL?**

GROUP BY используется для группировки строк, имеющих одинаковые значения в указанных столбцах

1. **Чем отличается HAVING от WHERE?**

WHERE, когда фильтруешь отдельные строки до группировки.

HAVING, когда фильтруешь группы строк после применения агрегатных функций (SUM, AVG, и т.д.).

1. **В чем разница между COUNT(\*) и COUNT(column\_name)?**

COUNT(\*) - считает все строки   
COUNT(column\_name)- считает все строки кроме пустых (NULL)

**5. Напишите порядок выполнения запросов с учетом GROUP BY и**

**HAVING.**

**select** customer\_id, **count**(\*) **as** *total\_sales\_cnt* **from** sales\_61

**group** **by** customer\_id

**having** **count**(\*) > 3