**3.11 Практика по темам «Агрегатные функции» и «Фильтрация после группировки, оператор HAVING»**

Описание

В этом курсе мы предусмотрели опциональные мини-задания для самостоятельной практики. Их не нужно сдавать на проверку. Нажав на кнопку внизу страницы, вы найдёте алгоритм и советы по выполнению этого задания с перечнем использованных команд. Постарайтесь решить задание самостоятельно, но если возникнет вопрос, то вы всегда сможете посмотреть в эту подсказку.

В предыдущих видео вы узнали о том, что такое агрегатные функции и как пользоваться фильтрацией вывода при группировке. На практике часто приходится группировать данные по некоторому полю с учётом дополнительных условий, а агрегатные функции облегчают анализ такой выборки.

В этом задании вам необходимо выполнить несколько SQL-запросов к базе данных shop.sql:

1. Выведите статистику по каждому повторяющемуся товару: его название, количество повторений, минимальное количество, среднее количество и максимальное количество.
2. Выведите имя, email, общие траты и среднее количество заказанных товаров для пользователей, которые:
   * суммарно потратили больше 1000 рублей,
   * имеют почтовый адрес с доменом Великобритании,
   * сделали более одного заказа.

Эталонное решение

1. Вывод статистики по каждому повторяющемуся товару: его название, количество повторений, минимальное количество, среднее количество и максимальное количество.

Результат запроса (первые десять записей из 28):

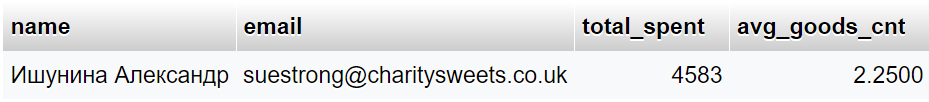


Запрос:

SELECT  
    `name`,  
    COUNT(\*) `repeat\_count`,  
    MIN(`count`) `min\_count`,  
    AVG(`count`) `avg\_count`,  
    MAX(`count`) `max\_count`  
FROM `good`  
GROUP BY `name`  
HAVING `repeat\_count` > 1;

1. Вывод имени, email, общих трат и среднего количества заказанных товаров для тех пользователей, которые суммарно потратили больше 1000 рублей, имеют почтовый адрес с доменом Великобритании и сделали более одного заказа.

Результат выполнения запроса:



Запрос:

SELECT  
    u.`name`,  
    u.`email`,  
    SUM(o2g.`count` \* g.`price`) `total\_spent`,  
    AVG(o2g.`count`) `avg\_goods\_cnt`  
FROM `user` u  
JOIN `order` o ON  
    o.user\_id = u.id  
JOIN `order2good` o2g ON  
    o2g.order\_id = o.id  
JOIN `good` g ON  
    g.id = o2g.good\_id  
GROUP BY o.user\_id  
HAVING  
    `total\_spent` > 1000 AND  
    u.`email` LIKE '%.uk' AND  
    COUNT(\*) > 1;