**4.5 Практика по теме «Вложенные запросы»**

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 предыдущем видео вы узнали о подзапросах. Пользоваться ими удобно при составлении запросов, ссылающихся на таблицы, которых как таковых и нет.

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

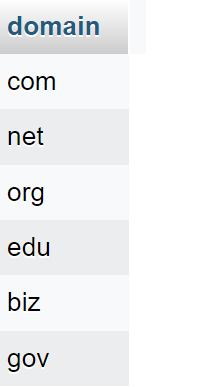
1. Используя вложенные запросы, выведите все уникальные почтовые домены верхнего уровня длиной в три символа.
2. Используя вложенные запросы, выведите для каждого товара, который находится в категории со средним количеством товаров меньше 185:
   * ID категории,
   * наименование товара,
   * количество товара,
   * стоимость такого товара.

Эталонное решение

1. Используя вложенные запросы, вывести все уникальные почтовые домены верхнего уровня длиной в три символа.

**Комментарий**: вся идея в таблице, которую мы будем использовать с оператором FROM. Это позволит не повторять строчные функции.

Результат выполнения запроса:



Здесь и далее — скриншоты из сервиса phpMyAdmin

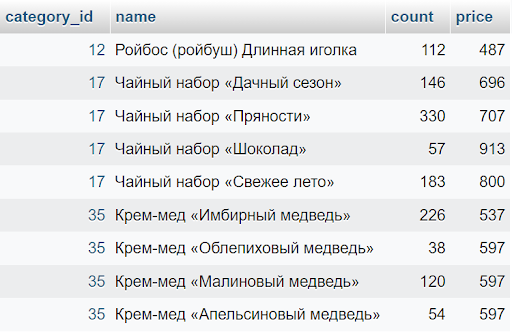
Запрос:

SELECT DISTINCT  
    `domain`  
FROM (  
    SELECT  
        REPLACE(  
            SUBSTR(`email`, -3),  
            '.',  
            '') `domain`  
    FROM `user`  
) t  
WHERE CHAR\_LENGTH(`domain`) = 3;

1. Используя вложенные запросы, вывести для каждого товара, который находится в категории со средним количеством товаров меньше 185:
   * ID категории,
   * наименование товара,
   * количество товара,
   * стоимость такого товара.

**Комментарий**: чтобы выводить информацию по элементам сгруппированных данных, нужно представить группу как подзапрос, который должен пристыковаться к исходной таблице до момента группировки. Задание можно решить как с помощью оператора WHERE, так и с помощью JOIN.

Результат выполнения запроса:



Запрос с использованием JOIN:

SELECT g.category\_id, g.name, g.count, g.price  
FROM `good` g   
JOIN (  
    SELECT category\_id  
    FROM `good`  
    GROUP BY category\_id  
    HAVING AVG(count) < 185  
) gg ON  
    gg.category\_id = g.category\_id;

Запрос с использованием WHERE:

SELECT g.category\_id, g.name, g.count, g.price  
FROM `good` g  
WHERE g.category\_id IN (  
    SELECT category\_id  
    FROM `good`  
    GROUP BY category\_id  
    HAVING AVG(count) < 185  
)