**4.8 Практика по теме «Представления»**

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, что такое представления и для чего они нужны, а также изучили запросы для создания, изменения и удаления представлений.

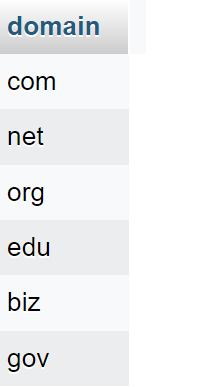
В этом задании вам необходимо выполнить несколько SQL-запросов к базе данных shop.sql:

1. Используя представления, выведите все уникальные почтовые домены верхнего уровня длиной в три символа.
2. Выведите общую стоимость товаров в каждой категории, в которой более 50 наименований. Воспользуйтесь представлениями.
3. Создайте представление с ID пользователя, ID заказа и общей стоимостью внутри такого пользовательского заказа, который переходил из статуса «Доставлен» в статус «Оплачен». Выведите содержимое представления.
4. Используя полученное в предыдущем задании представление, выведите:
   * ID пользователя;
   * строку со всеми товарами, которые были у пользователя в максимальном по общей стоимости заказе. Выводите товары, разделяя их запятой. Пробелы должны стоять только после запятой;
   * ID такого заказа;
   * общую стоимость. Если есть несколько заказов с одинаковой стоимостью, такого пользователя не рассматривайте.

Эталонное решение

1. Используя представления, выведите все уникальные почтовые домены верхнего уровня длиной в три символа.

Результат выполнения запроса:



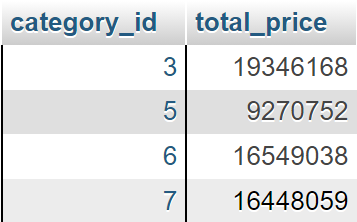
Запрос:

CREATE OR REPLACE VIEW `domains` AS  
SELECT  
    REPLACE(  
        SUBSTR(`email`, -3),  
        '.',  
        ''  
    ) `domain`  
FROM `user`;  
  
SELECT DISTINCT `domain`  
FROM `domains`  
WHERE CHAR\_LENGTH(`domain`) = 3;

1. Вывод имени категории и её общей стоимости товаров при условии, что в категории больше 50 наименований. Используем представления.

**Комментарий**: это задача из видео «Вложенные запросы». После создания представления полезно его изучить отдельно, чтобы понимать, с чем приходится иметь дело.

Результат выполнения запроса:



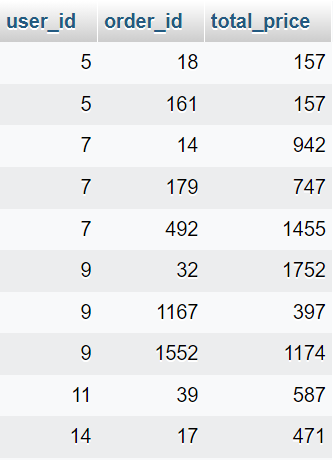
Запрос:

CREATE OR REPLACE VIEW `vast\_goods\_categories` AS  
SELECT `category\_id` `id`  
FROM `good`  
GROUP BY `category\_id`  
HAVING COUNT(\*) > 50;  
SELECT  
    g.`category\_id`,  
    SUM(`count` \* `price`) total\_price  
FROM `good` g  
JOIN `vast\_goods\_categories` vgc ON  
    vgc.id = g.`category\_id`  
WHERE g.`category\_id` IN(vgc.id)  
GROUP BY g.`category\_id`;

1. Создание представления с ID пользователя, ID заказа и общей стоимостью внутри такого пользовательского заказа, который переходил из статуса «Доставлен» в статус «Оплачен». Вывод содержимого представления.

**Комментарий**: группируйте по двум полям, на каждого пользователя может быть несколько заказов, в то же время в каждом заказе может быть несколько товаров.

Результат выполнения запроса (первые десять из 970):



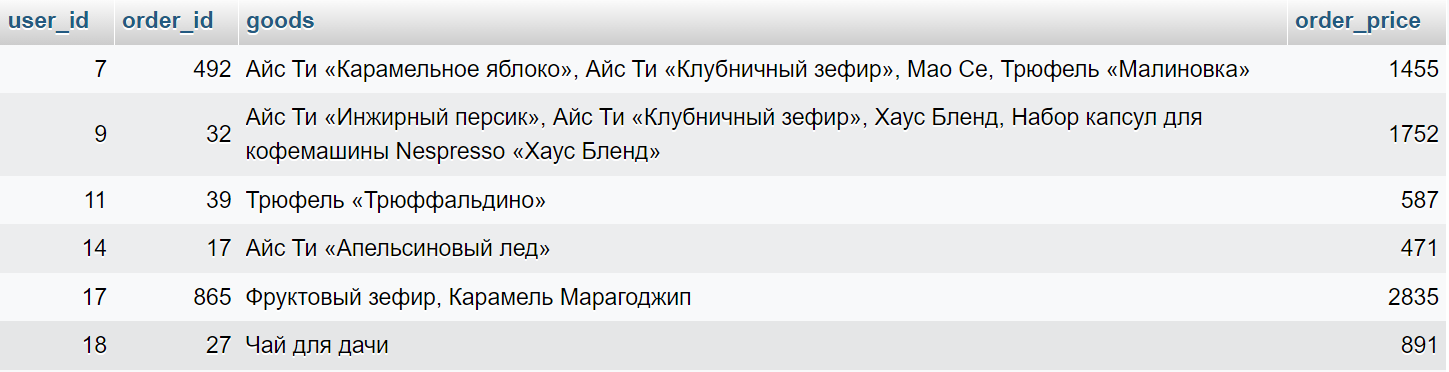
Запрос:

CREATE OR REPLACE VIEW `users\_orders` AS  
SELECT  
    u.id user\_id,  
    o.id order\_id,  
    SUM(o2g.count \* g.price) total\_price  
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  
JOIN `order\_status\_change` osc ON  
    osc.order\_id = o.id  
WHERE  
    osc.src\_status\_id = 6 AND  
    osc.dst\_status\_id = 7  
GROUP BY u.id, o.id;

1. Используя полученное в предыдущем задании представление, вывести:
   * ID пользователя;
   * строку со всеми товарами, которые были у пользователя в максимальном по общей стоимости заказе. Вывести товары, разделяя их запятой. Пробел должен стоять только после запятой;
   * ID такого заказа;
   * общую стоимость. Если есть несколько заказов с одинаковой стоимостью, такого пользователя не рассматривать.

**Комментарий**: чтобы воспользоваться максимальным по стоимости заказом каждого пользователя, придётся выполнить подзапрос на стыковку представления с самим собой. Далее присоединяются таблицы для взятия наименования товара, а чтобы избавиться от повторяющихся заказов с одинаковой стоимостью, выполняем фильтрацию на группе.

Результат выполнения запроса (первые шесть записей из 619):



Запрос:

SELECT  
    uo.user\_id,  
    uo.order\_id,  
    GROUP\_CONCAT(TRIM(g.name) SEPARATOR ', ') `goods`,  
    max\_order\_price order\_price  
FROM `users\_orders` uo  
JOIN (  
    SELECT uoi.user\_id, MAX(uoi.total\_price) `max\_order\_price`  
    FROM `users\_orders` uoi  
    GROUP BY uoi.user\_id  
) muo ON  
    muo.user\_id = uo.user\_id AND `max\_order\_price` = uo.total\_price  
JOIN `order` o ON  
    o.id = uo.order\_id  
JOIN `order2good` o2g ON  
    o2g.order\_id = o.id  
JOIN `good` g ON  
    g.id = o2g.good\_id  
GROUP BY uo.user\_id  
HAVING COUNT(DISTINCT uo.order\_id) = 1;