**3.6 Практика по теме «Функции работы со строками»**

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 про работу со строками в запросах SQL. Иметь дело со строками приходится часто, поэтому полезно иметь багаж опыта и функций для работы с ними.

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

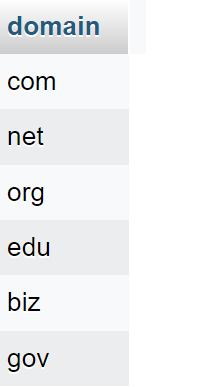
1. Выведите все уникальные почтовые домены верхнего уровня длиной в три символа.
2. Выведите родительские категории и все товары, входящие в них, а также их количество.  
   Если вам захочется больше практики, то мы подготовили дополнительную задачу повышенной сложности:
3. Выведите строки формата ‘Статус заказа номер <order id> пользователя “<user name>” изменился <дата без времени> с <src status code> на <dst status code>’ для заказов, которые переходили со статусов «Доставлен» на «Оплачен» и созданных 25 мая 2015 года.

Эталонное решение

1. Вывод всех уникальных почтовых доменов верхнего уровня длиной в три символа.

**Комментарий**: это задание можно выполнить разными способами. Вы можете использовать функции из предыдущего видео или найти другую функцию, которая позволит выполнить задание чуть более элегантно. Умение искать подходящие функции под задачу важно в работе с базами данных разработчика любого профиля.

Результат запроса:



Запросы:

Вариант 1: на основе материала из видео.  
SELECT DISTINCT  
    REPLACE(SUBSTR(`email`, -3), '.', '') `domain`  
FROM `user`  
WHERE CHAR\_LENGTH(REPLACE(SUBSTR(`email`, -3), '.', '')) = 3;

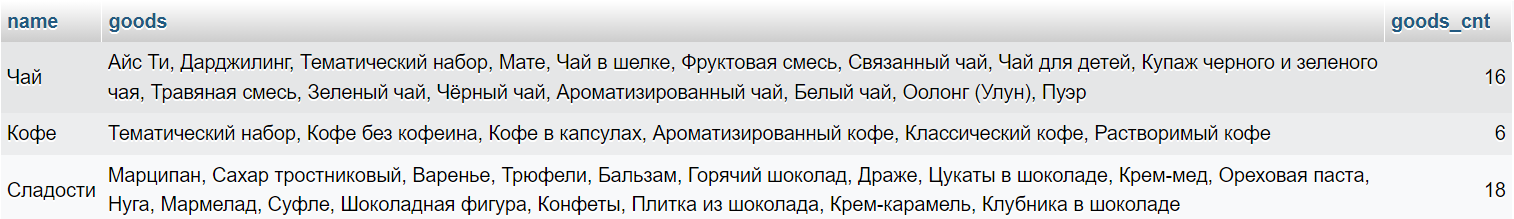
Вариант 2: элегантное применение новой функции.

SELECT DISTINCT  
    @var `domain`  
FROM `user`  
WHERE CHAR\_LENGTH((@var := SUBSTRING\_INDEX(`email`, '.', -1))) = 3;

1. Вывод родительских категорий и всех товаров, входящих в них, а также их количества.

**Комментарий**: чтобы трюк получился, нужно сделать JOIN таблицы с самой собой.

Результат запроса:



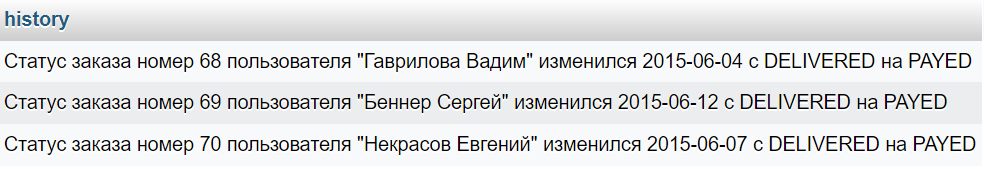
Запрос:

SELECT  
    gc2.`name`,  
    GROUP\_CONCAT(TRIM(gc.`name`) SEPARATOR ', ') `goods`,  
    COUNT(\*) `goods\_cnt`  
FROM `good\_category` gc  
JOIN `good\_category` gc2 ON  
    gc2.id = gc.parent\_id  
GROUP BY gc.parent\_id;

1. Вывод строки формата ‘Статус заказа номер <order id> пользователя “<user name>” изменился <дата без времени> с <src status code> на <dst status code>’ для заказов, которые переходили со статусов «Доставлен» на «Оплачен» и созданных 25 мая 2015 года.

**Комментарий**: нужно будет пристыковать два раза одну и ту же таблицу, чтобы получить коды статусов. Дату, поскольку она имеет чёткий формат, удобно менять с помощью взятия подстроки или же можно воспользоваться функцией DATE.

Результат выполнения запроса:



Запрос:

SELECT  
    CONCAT(  
        'Статус заказа номер ',  
        o.id,  
        ' пользователя "',  
        u.`name`,  
        '" изменился ',  
        SUBSTR(osc.`time`, 1, 10), # или DATE(osc.`time`)  
        ' с ',  
        os\_src.`code`,  
        ' на ',  
        os\_dst.`code`  
    ) `history`  
FROM `user` u  
JOIN `order` o ON  
    o.user\_id = u.id  
JOIN `order\_status\_change` osc ON  
    osc.order\_id = o.id  
JOIN `order\_status` os\_src ON  
    os\_src.id = osc.src\_status\_id  
JOIN `order\_status` os\_dst ON  
    os\_dst.id = osc.dst\_status\_id  
WHERE  
    osc.src\_status\_id = 6 AND  
    osc.dst\_status\_id = 7 AND  
    o.creation\_date >= '2015-05-25' AND  
    o.creation\_date < '2015-05-26';