

Задание OZON Camp блок Аналитика, SQL

Выполнил:

Титарчук Константин Валериевич

2021-11-26 23:00:00

Есть 3 таблицы:

1. users – пользователи
2. purchases – покупки
3. skus – товары

(процесс создания и заполнения таблиц представлен ниже)

Вывести по этим таблицам ответы на вопросы:

- **Сколько людей купило товар с sku.id = 5 10 октября?**

Результат запроса должен возвращать одну цифру.

Запрос:

```
SELECT COUNT(DISTINCT user_id) AS cnt_users_who_buy  
FROM purchases  
WHERE sku_id = 5 AND created_at = '2021-10-10';
```

результат выполнения запроса:

	cnt_users_who_buy
1	4

- **Сколько людей из Москвы купило телефоны в сентябре и октябре?**

Результат запроса должен возвращать таблицу формата.

month	people
september	n
october	m

n, m – кол-во людей, купивших телефоны в формате integer.

Запрос:

```
SELECT mnth AS `month`, COUNT(id) AS `people`  
FROM (SELECT users.id, MONTHNAME(purchases.created_at) AS mnth  
FROM users  
JOIN purchases ON users.id = purchases.user_id  
JOIN skus ON skus.id = purchases.sku_id  
WHERE city = 'Moscow'  
AND MONTHNAME(purchases.created_at) IN ('September', 'October')  
AND skus.category = 'phone'  
) AS request  
GROUP BY mnth;
```

результат выполнения запроса:

	ABC month	123 people
1	September	3
2	October	1

- Какая средняя стоимость 5-ой покупки у пользователей в разных городах?

Результат запроса должен возвращать таблицу формата.

city	avg_price_5th_purchase
Moscow	n
Novosibirsk	m

n, m – средняя цена, в формате float, округленная до 2-х знаков после запятой.

Запрос:

```
SELECT city, ROUND(AVG(price), 2) AS `avg_price_5th_purchase`  
FROM (  
    SELECT users.id, users.city, skus.price, ROW_NUMBER() OVER(PARTITION BY  
users.id ORDER BY purchases.created_at) AS `row_number`  
    FROM users  
    JOIN purchases ON users.id = purchases.user_id  
    JOIN skus ON skus.id = purchases.sku_id  
    ) AS request2  
WHERE `row_number` = 5  
GROUP BY city;
```

результат выполнения запроса:

	ABC city	123 avg_price_5th_purchase
1	Moscow	21,953.33
2	Novosibirsk	20,935

Создание и наполнение таблиц для осуществления запросов к ним:

IDE среда DBeaver, файл формата *.sql для быстрого запуска скрипта доступен по ссылке:

https://drive.google.com/file/d/1pJlsYsjOM3BxIBN_CkZJu4SKlvgC4xZ6/view?usp=sharing

users – данные о людях

Поля:

- id – уникальный идентификатор
- created_at – дата и время по UTC

- name – имя пользователя в формате строки
- city – город человека в формате enum. Включает в себя: Moscow, Novosibirsk.

purchases – данные о покупках

Поля:

- id – уникальный идентификатор
- created_at – дата и время по UTC
- user_id – уникальный идентификатор человека из таблицы users
- sku_id – уникальный идентификатор товара из таблицы skus

skus – данные о товарах

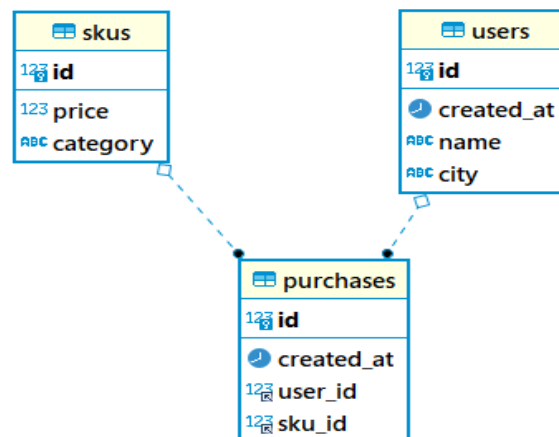
Поля:

- id – уникальный идентификатор товара.
- price – цена товара.
- category – категория товара. Включает: tv, phone, audio'

Примечание:

Не все люди из таблицы users совершали покупки.

В таблицах users и purchases данные только за 2021 год.



Создание таблиц:

```
DROP DATABASE IF EXISTS train_ozon;
```

```
CREATE DATABASE train_ozon;
```

```
USE train_ozon;
```

```
DROP TABLE IF EXISTS users;
```

```
CREATE TABLE users (
    id BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    created_at DATETIME,
    name VARCHAR(100),
    city ENUM('Moscow', 'Novosibirsk')
);
```

```
DROP TABLE IF EXISTS skus;
```

```
CREATE TABLE skus (
    id BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,
```

```

        price BIGINT,
        category ENUM('tv', 'phone', 'audio')
    );

DROP TABLE IF EXISTS purchases;
CREATE TABLE purchases (
    id BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    created_at DATETIME,
    user_id BIGINT,
    sku_id BIGINT,

    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (sku_id) REFERENCES skus(id)
);

```

Заполнение таблиц:

```

INSERT INTO users VALUES
(NULL, '2021-11-26' - INTERVAL 100 DAY, 'Ivan', 'Moscow'),
(NULL, '2021-11-26' - INTERVAL 95 DAY, 'Aleksa', 'Moscow'),
(NULL, '2021-11-26' - INTERVAL 90 DAY, 'Karen', 'Novosibirsk'),
(NULL, '2021-11-26' - INTERVAL 85 DAY, 'Ivana', 'Novosibirsk'),
(NULL, '2021-11-26' - INTERVAL 80 DAY, 'Alexey', 'Moscow'),
(NULL, '2021-11-26' - INTERVAL 75 DAY, 'Anton', 'Novosibirsk'),
(NULL, '2021-11-26' - INTERVAL 70 DAY, 'Akat', 'Moscow'),
(NULL, '2021-11-26' - INTERVAL 65 DAY, 'Katy', 'Moscow'),
(NULL, '2021-11-26' - INTERVAL 60 DAY, 'Kristina', 'Novosibirsk'),
(NULL, '2021-11-26' - INTERVAL 55 DAY, 'Kirill', 'Novosibirsk'),
(NULL, '2021-11-26' - INTERVAL 50 DAY, 'Ivan', 'Moscow'),
(NULL, '2021-11-26' - INTERVAL 45 DAY, 'Mikhail', 'Moscow'),
(NULL, '2021-11-26' - INTERVAL 40 DAY, 'Gosha', 'Novosibirsk'),
(NULL, '2021-11-26' - INTERVAL 35 DAY, 'Avan', 'Novosibirsk'),
(NULL, '2021-11-26' - INTERVAL 30 DAY, 'Petr', 'Novosibirsk'),
(NULL, '2021-11-26' - INTERVAL 25 DAY, 'Petr', 'Moscow'),
(NULL, '2021-11-26' - INTERVAL 20 DAY, 'Iraqlii', 'Novosibirsk'),
(NULL, '2021-11-26' - INTERVAL 15 DAY, 'Evgenii', 'Moscow'),
(NULL, '2021-11-26' - INTERVAL 10 DAY, 'Inna', 'Moscow'),
(NULL, '2021-11-26' - INTERVAL 5 DAY, 'Anna', 'Novosibirsk');

```

```

INSERT INTO skus VALUES
(NULL, 8099, 'tv'),
(NULL, 9055, 'tv'),
(NULL, 10300, 'tv'),
(NULL, 11450, 'tv'),
(NULL, 12669, 'tv'),
(NULL, 13700, 'tv'),
(NULL, 14899, 'tv'),
(NULL, 15500, 'tv'),
(NULL, 16800, 'tv'),
(NULL, 17880, 'tv'),
(NULL, 18900, 'tv'),
(NULL, 19099, 'tv'),

```

(NULL, 20100, 'tv'),
(NULL, 21500, 'tv'),
(NULL, 22099, 'tv'),
(NULL, 23099, 'tv'),
(NULL, 24430, 'tv'),
(NULL, 25100, 'tv'),
(NULL, 26111, 'tv'),
(NULL, 27555, 'tv'),
(NULL, 8555, 'phone'),
(NULL, 9599, 'phone'),
(NULL, 10550, 'phone'),
(NULL, 11588, 'phone'),
(NULL, 12577, 'phone'),
(NULL, 13566, 'phone'),
(NULL, 14560, 'phone'),
(NULL, 15533, 'phone'),
(NULL, 16520, 'phone'),
(NULL, 17560, 'phone'),
(NULL, 18540, 'phone'),
(NULL, 19544, 'phone'),
(NULL, 20555, 'phone'),
(NULL, 21566, 'phone'),
(NULL, 22599, 'phone'),
(NULL, 23599, 'phone'),
(NULL, 24599, 'phone'),
(NULL, 25599, 'phone'),
(NULL, 26590, 'phone'),
(NULL, 27580, 'phone'),
(NULL, 8999, 'audio'),
(NULL, 9910, 'audio'),
(NULL, 10999, 'audio'),
(NULL, 11999, 'audio'),
(NULL, 12989, 'audio'),
(NULL, 13979, 'audio'),
(NULL, 14956, 'audio'),
(NULL, 15952, 'audio'),
(NULL, 16960, 'audio'),
(NULL, 17912, 'audio'),
(NULL, 18996, 'audio'),
(NULL, 19966, 'audio'),
(NULL, 20944, 'audio'),
(NULL, 21950, 'audio'),
(NULL, 22960, 'audio'),
(NULL, 23990, 'audio'),
(NULL, 24999, 'audio'),
(NULL, 25940, 'audio'),
(NULL, 26988, 'audio'),
(NULL, 27999, 'audio');

INSERT INTO purchases VALUES

(NULL, '2021-11-26' - INTERVAL 95 DAY, 1, 1),
 (NULL, '2021-11-26' - INTERVAL 94 DAY, 1, 15),
 (NULL, '2021-11-26' - INTERVAL 91 DAY, 2, 7),
 (NULL, '2021-11-26' - INTERVAL 91 DAY, 2, 50),
 (NULL, '2021-11-26' - INTERVAL 85 DAY, 4, 10),
 (NULL, '2021-11-26' - INTERVAL 78 DAY, 5, 25),
 (NULL, '2021-11-26' - INTERVAL 78 DAY, 5, 27),
 (NULL, '2021-11-26' - INTERVAL 65 DAY, 7, 10),
 (NULL, '2021-11-26' - INTERVAL 65 DAY, 7, 35),
 (NULL, '2021-11-26' - INTERVAL 65 DAY, 7, 59),
 (NULL, '2021-11-26' - INTERVAL 64 DAY, 8, 2),
 (NULL, '2021-11-26' - INTERVAL 64 DAY, 8, 5),
 (NULL, '2021-11-26' - INTERVAL 53 DAY, 10, 16),
 (NULL, '2021-11-26' - INTERVAL 50 DAY, 11, 22),
 (NULL, '2021-11-26' - INTERVAL 49 DAY, 11, 20),
 (NULL, '2021-11-26' - INTERVAL 47 DAY, 11, 5),
 (NULL, '2021-11-26' - INTERVAL 47 DAY, 1, 5),
 (NULL, '2021-11-26' - INTERVAL 47 DAY, 1, 5),
 (NULL, '2021-11-26' - INTERVAL 47 DAY, 3, 5),
 (NULL, '2021-11-26' - INTERVAL 47 DAY, 7, 5),
 (NULL, '2021-11-26' - INTERVAL 39 DAY, 13, 1),
 (NULL, '2021-11-26' - INTERVAL 34 DAY, 14, 5),
 (NULL, '2021-11-26' - INTERVAL 30 DAY, 16, 9),
 (NULL, '2021-11-26' - INTERVAL 30 DAY, 16, 10),
 (NULL, '2021-11-26' - INTERVAL 30 DAY, 16, 51),
 (NULL, '2021-11-26' - INTERVAL 30 DAY, 16, 52),
 (NULL, '2021-11-26' - INTERVAL 29 DAY, 3, 8),
 (NULL, '2021-11-26' - INTERVAL 29 DAY, 3, 26),
 (NULL, '2021-11-26' - INTERVAL 28 DAY, 4, 50),
 (NULL, '2021-11-26' - INTERVAL 27 DAY, 6, 12),
 (NULL, '2021-11-26' - INTERVAL 27 DAY, 6, 13),
 (NULL, '2021-11-26' - INTERVAL 24 DAY, 18, 4),
 (NULL, '2021-11-26' - INTERVAL 23 DAY, 18, 20),
 (NULL, '2021-11-26' - INTERVAL 20 DAY, 19, 5),
 (NULL, '2021-11-26' - INTERVAL 20 DAY, 19, 10),
 (NULL, '2021-11-26' - INTERVAL 19 DAY, 2, 5),
 (NULL, '2021-11-26' - INTERVAL 18 DAY, 1, 58),
 (NULL, '2021-11-26' - INTERVAL 18 DAY, 2, 60),
 (NULL, '2021-11-26' - INTERVAL 18 DAY, 2, 55),
 (NULL, '2021-11-26' - INTERVAL 16 DAY, 16, 49),
 (NULL, '2021-11-26' - INTERVAL 16 DAY, 16, 2),
 (NULL, '2021-11-26' - INTERVAL 15 DAY, 3, 10),
 (NULL, '2021-11-26' - INTERVAL 15 DAY, 3, 10),
 (NULL, '2021-11-26' - INTERVAL 15 DAY, 6, 11),
 (NULL, '2021-11-26' - INTERVAL 14 DAY, 14, 19),
 (NULL, '2021-11-26' - INTERVAL 14 DAY, 14, 20),
 (NULL, '2021-11-26' - INTERVAL 13 DAY, 14, 57),
 (NULL, '2021-11-26' - INTERVAL 12 DAY, 14, 56),
 (NULL, '2021-11-26' - INTERVAL 10 DAY, 14, 55);

конец задания