**УПРАЖНЕНИЯ**

Задание выполняется на основе Датаcета: <https://www.kaggle.com/datasets/smayanj/e-commerce-transactions-dataset>

## --- Базовая Задача: «Создать таблицу»

**create** **table** Ecommerce\_Transactions

(

Transaction\_ID **INTEGER**,

User\_Name **VARCHAR**(50),

Age **INTEGER**,

Country **VARCHAR**(50),

Product\_Category **VARCHAR**(100),

Purchase\_Amount **INTEGER**,

Payment\_Method **VARCHAR**(50),

Transaction\_Date **TIMESTAMP**

**);**

## --- Базовая Задача: «Загрузить данные»

--- Импортировать csv данные в созданную таблицу. :

### --- **→ Folder Publiс ↓ →Папка Таблицы**

--- →Папка Таблицы: «Импорт данных»

--- → На этапе «Входные файлы» - выбрать корректный формат «Значение» для отображения CSV таблицы. «; или , ».

--- → --- → На этапе «Соответствие столбцов» - проверить формат «Значение» CSV таблицы в «Просмотр данных» - это предварительная загрузка информации. Тестовый просмотр.

## --- 1. Показать имена пользователей и страны, из которых они совершали покупки (без дубликатов).

**select** **distinct**(user\_name), country

**from** Ecommerce\_Transactions

**group** **by** **distinct**(user\_name), country ; --- Результат: 1000 значений (Не разобрался как вкл. сортировку в примере запроса. 1 столбец выдачи "П/П" не сортируется) <https://disk.yandex.ru/i/wbX6j64DP3VCjQ>

--- Проверка:

**select** (user\_name), country, Transaction\_ID

**from** Ecommerce\_Transactions

**ORDER** **BY** Transaction\_ID **desc**

**limit** 5 ; --- Результат: 7596 значений

https://disk.yandex.ru/i/YUcHkRk4\_lY9zg

## --- 2. Найти общую сумму покупок по каждой категории товаров.

**select** product\_Category, **sum**(purchase\_Amount)

**from** Ecommerce\_Transactions

**group** **by** product\_category ;

https://disk.yandex.ru/i/7ArCvG9iCxEM8g

## --- 3. Определить средний возраст покупателей для каждой страны.

**select** country, **round**(**avg**(age), 0) **as** *avg\_age*

--- country - выбирает название страны

--- avg(age) - вычисляет среднее арифметическое возраста всех user\_name

--- round(..., 0) - округляет результат до целого числа

--- as avg\_age - дает псевдоним столбцу с результатом

**from** Ecommerce\_Transactions

--- Откуда забирать данные

**group** **by** country ;

--- Группирует все записи по странам. Для каждой страны вычисляется отдельное среднее арифметическое значение.

https://disk.yandex.ru/i/6TUxA\_eAcEdBag

## --- 4. Найти топ-3 самых крупных покупки. Указать имя, категорию и сумму.

**select** User\_name, Product\_Category, Purchase\_Amount

**from** Ecommerce\_Transactions

**order** **by** Purchase\_Amount **desc**

**limit** 3

;

https://disk.yandex.ru/i/vQbAEqfUI-9CKA

## --- 5. Посчитать количество транзакций, сделанных каждым пользователем.

**select** user\_name, **count**(purchase\_amount) **as** *count\_transaction*

**from** Ecommerce\_Transactions

**group** **by** user\_name

**order** **by** *count\_transaction* **desc**

;

https://disk.yandex.ru/i/IZk8rp1tu1U\_2A

## --- 6. Показать все транзакции за 2024 год.

---1 вариант:

**select** \*

--- Функция " SELECT " + " \* ": возвращает все столбцы и строки из указанной таблицы -> SELECT \* "FROM" "table\_name"

**from** Ecommerce\_Transactions

--- Источник данных

**where** **extract** ---Извлекает год из даты без фильтрации

(**year** **from** Transaction\_Date) = 2024

--- Возвращает полную информацию о каждой транзакции

--- "= 2024" - Фильтрует только те записи, где год равен 2024 в транзакциях(дате)

;

---2 вариант, через диапозон дат:

**select** \*

**from** Ecommerce\_Transactions

**WHERE** Transaction\_Date >= '2024-01-01'

**AND** Transaction\_Date < '2024-12-31'

;

---3816 транзакций за 2024 год: https://disk.yandex.ru/i/6PejUq7Iwf2iDQ

## --- 7. Найти пользователей, у которых сумма покупки превышает 500.

**select** user\_name, purchase\_amount

**from** Ecommerce\_Transactions

**where** purchase\_amount >500

--- Фильтрует общую сумму, потраченную на транзакцию и оставляет значение больше 500

--- Результат: 3798 записей

## --- 8. Определить самый популярный способ оплаты (по количеству транзакций).

--- 8А.

**select** payment\_method, **count**(\*) **as** *count\_usage*

/\*

\* count(\*) - Подсчитывает все строки - независимо от значений в них

\* Включает все записи - даже те, где есть NULL значения

\* Работает с дубликатами - считает все строки, включая повторяющиеся

\* as - Это ключевое слово для создания псевдонимов (aliases) для столбцов или таблиц.

\* - Переименовывает столбцы в результатах запроса

\* - Создает понятные названия для вычисляемых полей

\* - Укорачивает имена таблиц для удобства написания запросов

\* count\_usage - псевдоним

\*/

**from** ecommerce\_transactions

**group** **by** payment\_method

**order** **by** *count\_usage* **desc** --- Сортирует от Я до А (Обратный порядок. Убывание)

**limit** 1 --- Ограничения вывода информации по количеству строк

;

--- 8Б. вариант 2

**select** payment\_method, **count**(transaction\_ID) **as** *count\_transaction*

**from** Ecommerce\_Transactions

**group** **by** payment\_method

**order** **by** *count\_transaction* **desc**

**limit** 1

;

## --- 9. Вывести пользователей из Франции, отсортированных по убыванию суммы покупки.

**select** user\_name, country, purchase\_amount

**from** Ecommerce\_Transactions

**where** country = 'France'

**order** **by** purchase\_amount **desc**

--- Результат: 736

## --- 10. Найти общую выручку за каждый год.

---

**select** **extract** (**year** **from** transaction\_date) **as** **year**, **sum**(purchase\_amount)

/\*

\* 1. extract - это функция для извлечения конкретных частей из даты/времени.

\* 2. (year - часть даты: год

\* 3. from transaction\_date) 2 извлеки из 3 только год

\* as year - назвали операцию "Year"

\*/

**from** Ecommerce\_Transactions

**group** **by** **extract**(**year** **from** transaction\_date)

**order** **by** **year** **desc**

;

--- Оптимизировано )

**select** **extract** (**year** **from** transaction\_date) **as** **year**, **sum**(purchase\_amount)

**from** Ecommerce\_Transactions

**group** **by** **year**

**order** **by** **year** **desc**

## --- 11. Показать самого молодого и самого старого покупателя в одной строке

**select** **min**(age) **as** *min\_age*, **max**(age) **as** *max\_age*

**from** Ecommerce\_Transactions

## --- 12. Вывести категории товаров, средняя цена покупки в которых выше 400.

-- В1 - не рабочий

**select**

Product\_Category,

**Round**(**AVG**(Purchase\_Amount), 0) **as** *avg\_Purchase*

**from** Ecommerce\_Transactions

**group** **by** Product\_Category

**having** **AVG**(Purchase\_Amount) > 400

/\*

\* НЕПРАВИЛЬНО: GROUP BY Product\_Category AND AVG(avg\_purchase) = > 400

\* НЕПРАВИЛЬНО: = > 400 -- пробел между = и >

\* ПРАВИЛЬНО: >= 400

\* НЕПРАВИЛЬНО: AVG(avg\_purchase) -- нельзя использовать алиас в агрегатной функции

\*/

**order** **by** Purchase\_Amount **desc**

-- НЕПРАВИЛЬНО: ORDER BY Purchase\_Amount DESC -- это отдельная покупка, а не среднее

;

-- В2

**SELECT**

Product\_Category,

**ROUND**(**AVG**(Purchase\_Amount), 2) **AS** *Avg\_Purchase*

**FROM** ecommerce\_transactions

**GROUP** **BY** Product\_Category

**HAVING** **AVG**(Purchase\_Amount) > 400 -- фильтрует группы, оставляя только те, где средняя сумма покупки ≥ 400

**ORDER** **BY** *Avg\_Purchase* **desc**

;

## --- 13. Найти всех пользователей с именем 'Sophia' и показать их последнюю транзакцию (по дате).

-- Варинат 1

**select**

user\_name,

country,

**max**(transaction\_date)

**from** Ecommerce\_Transactions

**where** user\_name **like** 'Sophia%'

-- НЕПРАВИЛЬНО: ❌ GROUP BY (User\_name) = 'Sophia %' - неправильный синтаксис фильтрации

**group** **by** user\_name, country

/\*

\* **LIKE** - оператор для поиска по шаблону в текстовых полях. Он позволяет искать строки, которые соответствуют определенному шаблону, а не точному значению. (Находит все имена, которые начинаются с "Sophia")

\* **%** заменяет любые символы после "Sophia"

\*/

;

-- Варинат 2 (оптимизирован)

**select**

user\_name,

**max**(transaction\_date)

**from** Ecommerce\_Transactions

**where** user\_name **ilike** 'Sophi\_%'

**group** **by** user\_name

/\*

\*  **iLIKE** - **LIKE** мощный инструмент для гибкого поиска текстовых данных, когда вам нужно найти не точное совпадение, а строки, соответствующие определенному шаблону.

\* Важные особенности: В PostgreSQL: чувствителен (лучше использовать ILIKE)

\* **Wildcard-символы**: "\_" - Нижнее подчеркивание. Ровно один любой символ после условия:

\* Имена из 5 букв, начинающиеся на "Soph"

\* WHERE name LIKE 'Soph\_'

\* Найдет: 'Sophi', 'Sophe'

\* Не найдет: 'Sophia' (6 букв), 'Soph' (4 буквы)

\*/

;

## --- 14. Посчитать количество уникальных стран, в которых были совершены покупки.

-- Вариант 1

**select** **count**(**distinct** country) **as** *Unique\_Countries\_By\_Transactions*

-- НЕПРАВИЛЬНО: ❌ select count(distinct country), as Unique\_Countries\_By\_Transactions

-- РЕШЕНИЕ: Удалить запятую

**FROM** ecommerce\_transactions

;

-- Вариант 2

**select**

**count**(**distinct** country) **as** Unique\_Countries\_By\_Transactions,

-- НЕПРАВИЛЬНО: ❌ Отсутсвие запятятой в запроса "count(distinct country) as Unique\_Countries\_By\_Transactions"

**STRING\_AGG** (**DISTINCT** country, ', ' **ORDER** **BY** country) **as** Unique\_Countries\_By\_Transactions\_List

/\* [Превращает множество строк в одну удобочитаемую строку с заданным разделителем и порядком](**STRING\_AGG**)

\* [Уникальные страны]((**DISTINCT** country,)

\* **STRING\_AGG**[Объединение с разделителем]((**', '**))

\* [Сортировка по странам](**ORDER BY** country))

\* [**Aliases**](**AS** Unique\_Countries\_By\_Transactions\_List)

\*/

**FROM** ecommerce\_transactions

;

## --- 15. Найти страны, где средняя сумма покупки превышает 400, и показать количество транзакций в этих страна

**SELECT**

Country,

**COUNT**(\*) **AS** *Transaction\_Count*,

**ROUND**(**AVG**(Purchase\_Amount), 2) **AS** *Avg\_Purchase\_Amount*

**FROM** Ecommerce\_Transactions

**GROUP** **BY** Country

**HAVING** **AVG**(Purchase\_Amount) > 400

**ORDER** **BY** *Avg\_Purchase\_Amount* **desc**

;

## --- 16. Найти способы оплаты, которые использовались более 2 раз и при этом средняя сумма покупки превышает 300.

**SELECT**

Payment\_Method,

**COUNT**(\*) **AS** *Usage\_Count*,

**ROUND**(**AVG**(Purchase\_Amount), 2) **AS** *Avg\_Purchase\_Amount*,

**SUM**(Purchase\_Amount) **AS** *Total\_Amount*

**FROM** ecommerce\_transactions

**GROUP** **BY** Payment\_Method

**HAVING** **COUNT**(\*) > 2 **AND** **AVG**(Purchase\_Amount) > 300

**ORDER** **BY** *Avg\_Purchase\_Amount* **desc**

;

## --- 17. Найти способы оплаты, которые использовались более 2 раз и при этом средняя сумма покупки превышает 300.

-- Варианат 1: Самые прибылье категории в разрезе стран. Сортировка по Доход.

**SELECT**

Country,

Product\_Category,

**COUNT**(\*) **as** *Transaction\_Count*,

**SUM**(Purchase\_Amount) **as** *Profit*

**FROM** ecommerce\_transactions

**GROUP** **BY** Country, Product\_Category

**ORDER** **BY** *Profit* **desc**

;

-- Варианат 2: Самые прибылье категории в разрезе стран. Сортировка по стране, а в ней Доход.

**SELECT**

Country,

Product\_Category,

**COUNT**(\*) **as** *Transaction\_Count*,

**SUM**(Purchase\_Amount) **as** *Profit*

**FROM** ecommerce\_transactions

**GROUP** **BY** Country, Product\_Category

**ORDER** **BY** Country, *Profit* **desc**

;