César Mata Rodríguez- Data Analisys - TP4

Part 1: Initial Dataset Exploration

1. **Describe the customers table: Carefully examine the customers table and describe each column it contains, including the data type and potential meaning or usage of each field.**

   The customers table stores information about the clients of the platform. It contains a unique identifier, customer_id, which is an integer value used to distinguish each customer uniquely. The name and email columns are text fields that store the customer's full name and email address, which are used for identification and communication purposes. The city column is also a text field and indicates the customer's place of residence.

   The birthdate column is a date-type field that stores the customer's date of birth and can be used to estimate the customer's age. The join_date column is another date-type field that represents the date when the customer registered on the platform and can be useful for analyzing customer tenure and behavior over time.

2. **Draw a Schema/Diagram: Illustrate the relationships between the customers, products, and purchases tables. Indicate the primary keys and foreign keys and how these tables connect with each other.**

   The database is composed of three tables: customers, products, and purchases. The customers table uses customer_id as its primary key, while the products table uses product_id as its primary key. The purchases table has purchase_id as its primary key and includes customer_id and product_id as foreign keys that reference the customers and products tables respectively.

   Each customer can make multiple purchases, and each product can be purchased multiple times. Therefore, there is a many-to-many relationship between customers and products, which is resolved by the purchases table. The purchases table acts as a transactional (fact) table, where each purchase record is linked to exactly one customer and one product.

3. **Identify Qualitative and Quantitative Variables: Review the three tables and classify each column as a qualitative (categorical) or quantitative (numerical) variable.**

   In the customers table, customer_id, name, email, and city are qualitative variables. The customer_id column is an identifier used for referencing customers rather than for analysis. The birthdate and join_date columns are quantitative temporal variables, as they represent dates and allow time-based analysis.

   In the products table, product_id, product_name, and category are qualitative variables, with product_id serving as an identifier. The price column is a quantitative numerical variable and represents a continuous measure.

In the purchases table, purchase_id, customer_id, and product_id are qualitative identifier variables. The quantity column is a quantitative discrete numerical variable, as it represents a count of items purchased. The purchase_date column is a quantitative temporal variable that records when each transaction occurred.

Part 2: Data Analysis Questions

1. **Which city has the highest number of customers?**
   ```
   SELECT city, COUNT(*) AS number_of_customers
   FROM customers
   GROUP BY city
   ORDER BY number_of_customers DESC
   LIMIT 1
   ```

2. **Which product category is the most popular?**
   ```
   SELECT category, SUM(quantity) AS total_sales
   FROM purchases
   JOIN products
   ON purchases.product_id = products.product_id
   GROUP BY category
   ORDER BY total_sales DESC
   LIMIT 1
   ```

3. **Identify the top three customers by the number of purchases.**
   ```
   SELECT customer_id, COUNT(*) AS number_of_purchases
   FROM purchases
   GROUP BY customer_id
   ORDER BY number_of_purchases DESC
   LIMIT 3
   ```

4. **Calculate the total revenue generated by each product category.**
   ```
   SELECT category,
   SUM(products.price * purchases.quantity) AS total_revenue
   FROM purchases
   JOIN products
   ON purchases.product_id = products.product_id
   GROUP BY category
   ```

5. **Which products were bought in the largest quantities in April 2023?**
   ```
   SELECT product_name, SUM(quantity) AS total_quantity
   FROM purchases
   JOIN products ON purchases.product_id = products.product_id
   WHERE purchase_date BETWEEN '2023-04-01' AND '2023-04-30'
   GROUP BY product_name
   ORDER BY total_quantity DESC
   ```

6. **List the customers who have spent more than €500 since they joined**

```
SELECT customer_id, SUM(products.price * purchases.quantity) AS total_spent
FROM purchases
JOIN products
ON purchases.product_id = products.product_id
GROUP BY customer_id
HAVING total_spent > 500
```

7. **Compute the minimum, maximum and the average age of each customer. Use CURRENT_TIMESTAMP - birthdate to get an approximation of the customer age.**

```
SELECT avg(CURRENT_TIMESTAMP - birthdate), min(CURRENT_TIMESTAMP - birthdate),
max(CURRENT_TIMESTAMP - birthdate)
FROM customers
```

8. **Which three cities have the highest revenue?**

```
SELECT city, SUM(products.price * purchases.quantity) AS total_revenue
FROM purchases
JOIN products
ON purchases.product_id = products.product_id
JOIN customers
ON purchases.customer_id = customers.customer_id
GROUP BY city
ORDER BY total_revenue DESC
LIMIT 3
```

9. **Which product generates the most revenue?**

```
SELECT product_name, SUM(products.price * purchases.quantity) AS revenue
FROM purchases
JOIN products
ON purchases.product_id = products.product_id
GROUP BY product_name
ORDER BY revenue DESC
LIMIT 1
```

10. **Identify customers who have bought from at least two different product categories.**

```
SELECT customer_id
FROM purchases
JOIN products
ON purchases.product_id = products.product_id
GROUP BY customer_id
HAVING COUNT(DISTINCT category) >= 2
```

11. **Calculate the repeat purchase rate: how many customers have made more than one purchase?**

```
SELECT customer_id, COUNT(*) AS number_of_purchases
FROM purchases
GROUP BY customer_id
HAVING COUNT(*) > 1
```

12. **Create a visualization showing the sales trend for each product during April 2023.**

    ```
    SELECT purchase_date, product_name, SUM(quantity) AS total_quantity
    FROM purchases
    JOIN products
    ON purchases.product_id = products.product_id
    WHERE purchase_date BETWEEN '2023-04-01' AND '2023-04-30'
    GROUP BY purchase_date, product_name
    ORDER BY purchase_date
    ```

13. **What is the average number of products bought per order?**

    ```
    SELECT AVG(quantity) AS average_products_per_order
    FROM purchases
    ```

14. **Investigate whether there is a correlation between customer age and their total spending.**

    ```
    SELECT CURRENT_DATE - birthdate as age, SUM(products.price * purchases.quantity) AS total_spent
    FROM purchases
    JOIN products
    ON purchases.product_id = products.product_id
    JOIN customers
    ON purchases.customer_id = customers.customer_id
    GROUP BY purchases.customer_id
    ```