Real-Case Business Scenario for SQL Database

Business Overview:

You are responsible for creating a database system for an e-commerce platform that sells a wide variety of products. The platform manages customer information, product listings, categories, and order records. The following scenario provides the basis for constructing and querying a database to manage this e-commerce operation.

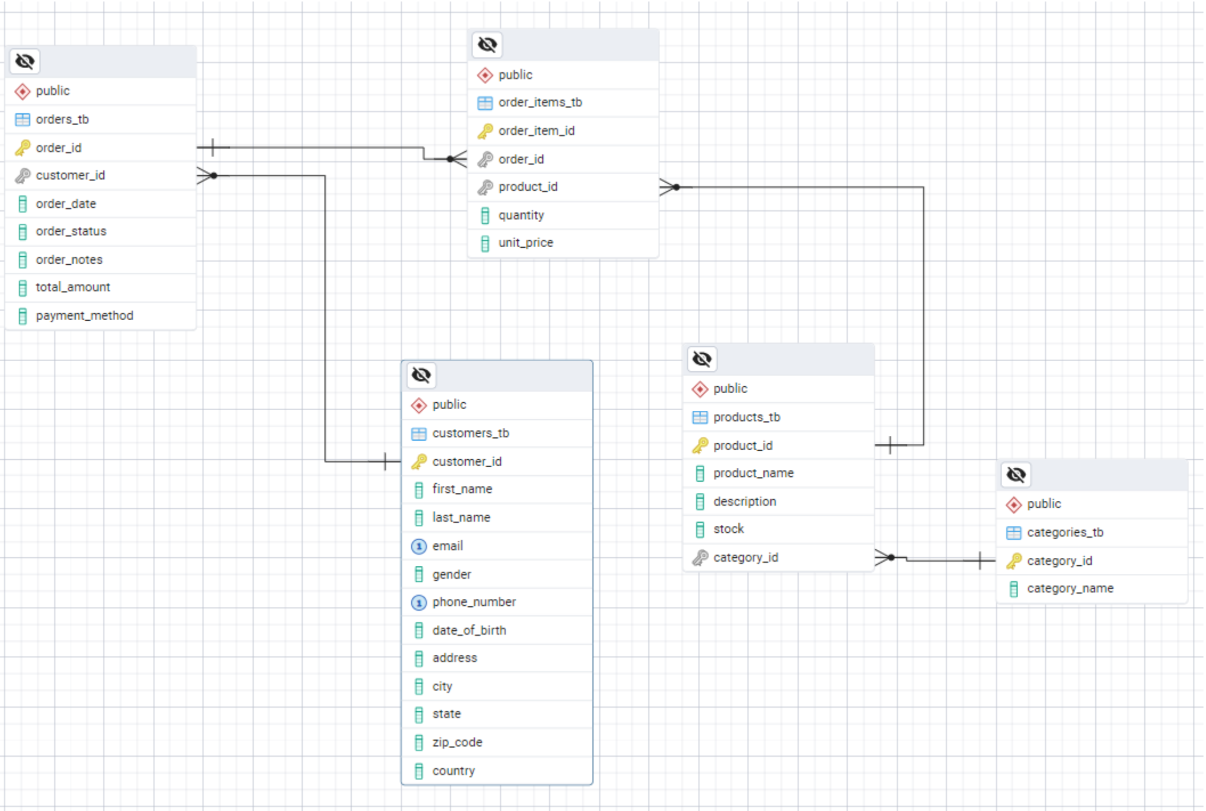
Specifications:

1. **Customer Management:**
   - The company has customers who provide their personal details like name, email, gender, and city of residence. This information is stored in the customers_tb table.
   - The company needs to keep track of customer sign-up dates to understand customer acquisition trends (which is something you can add later on).
   - Customers might place multiple orders, and all of their orders should be stored and linked to their profile.
2. **Product Management:**
   - The store offers various products categorised into sections like Electronics, Clothing, Home Appliances, Books, Sports equipment.
   - Each product belongs to a specific category and has information like the product name, description and available stock.
   - Products may run out of stock, and stock management is crucial for the business. The platform should track how much stock is available for each product and ensure stock quantities are non-negative.
   - The company needs the flexibility to handle product deletions; for instance, if a product is deleted, it will be set to NULL in orders and order items to ensure historical data integrity.
3. **Orders and Sales:**
   - The platform allows customers to place orders. Each order is associated with a specific customer and has a date and total amount.
   - Orders consist of one or more products. Each product in an order has its own quantity and price, which must be non-negative.

- If a customer is deleted from the system (for instance, due to GDPR requests), all of their associated orders and order items must be automatically deleted to comply with regulations.

4. **Business Expansion & Future-Proofing:**
   - The platform is expected to scale in the future, and new product categories might be added. Ensuring the system supports seamless product category management is essential.
   - The system should handle high sales volumes while keeping track of product availability and order histories without performance issues.

After the execution of the SQL queries provided by your tutor, here is the schema of the database you are supposed to have:



Your 1st task as the database manager is to check if this database design is optimal and if no, how it could be improved (normalisation). Write the SQL Script that will allow you to reach that objective, and present your optimal database design.

Secondly, you are asked to help the company to answer certain business questions:

Simple queries:

1. Retrieve a list of all customers.
2. List all orders with their total amounts and payment methods.

3. Retrieve all products with their stock and descriptions.
4. Find customers living in a specific city.
5. List all orders placed by a specific customer.
6. Retrieve all products and their categories.

Intermediate queries:

7. Find total orders placed by each customer.
8. Calculate the total revenue generated by each product.
9. Find the top 5 most frequently ordered products.
10. Retrieve orders along with their total number of items.

Advanced queries:

11. Find the customer who has spent the most on orders.
12. Identify products with stock levels below a certain threshold.
13. Use a CTE to find customers with more than 3 orders.
14. Create a view showing customer orders and their total number of items.
15. Find the most popular product category based on sales.
16. Use a derived table to list orders with at least 5 items.
17. Use a transaction to place a new order and update stock.

Expert queries:

18. Find the total revenue, number of orders, and average order value for each customer.
19. Find the top 3 customers by total revenue, along with the names of the products they ordered.
20. Calculate the monthly revenue, and show the month with the highest revenue.
21. Find products that were never ordered.
22. Identify the most frequently ordered product in each category.
23. List customers who have ordered products from more than 3 different categories.
24. Find the average number of items per order for each customer.
25. Calculate the stock-to-sales ratio for each product.
26. Generate a sales forecast by extrapolating the average monthly growth.
27. Detect inactive customers (no orders in the last 6 months).
28. Rank customers based on the percentage of their spending relative to total revenue.