E- Commerce Database System

Outline

- Introduction?
- Topic/Problem Statement?
- SQL Query?
- Explanation?
- Relational Algebra?
- Schema Diagram?

Introduction?

An **E-commerce Database System** streamlines online shopping by managing customers, products, orders, and payments, ensuring efficient inventory control, order tracking, and seamless transaction processing.

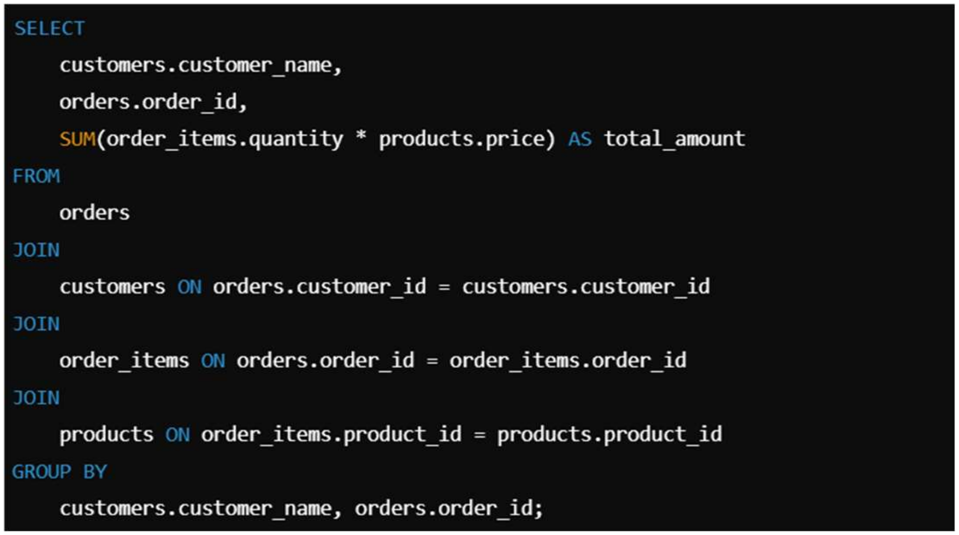
Topic/Problem Statement?

- E-commerce
- Database System
- Online Shopping
- Customers
- Products
- Orders
- Payments
- Order Processing
- Inventory Management
- Financial Transactions

❖ These terms emphasize the core functionalities and purpose of the system

SQL Query?

This query retrieves the total amount spent by each customer on their orders:



```sql
SELECT
    customers.customer_name,
    orders.order_id,
    SUM(order_items.quantity * products.price) AS total_amount
FROM
    orders
JOIN
    customers ON orders.customer_id = customers.customer_id
JOIN
    order_items ON orders.order_id = order_items.order_id
JOIN
    products ON order_items.product_id = products.product_id
GROUP BY
    customers.customer_name, orders.order_id;
```

Explanation

❖ Tables Involved:

❑ Customers: Stores customer details (e.g., customer_id, customer_name).

❑ Products: Stores product details (e.g., product_id, product_name, price).

❑ Orders: Captures order details (e.g., order_id, customer_id, order_date).

❑ Order_Items: Links products and orders with quantity.

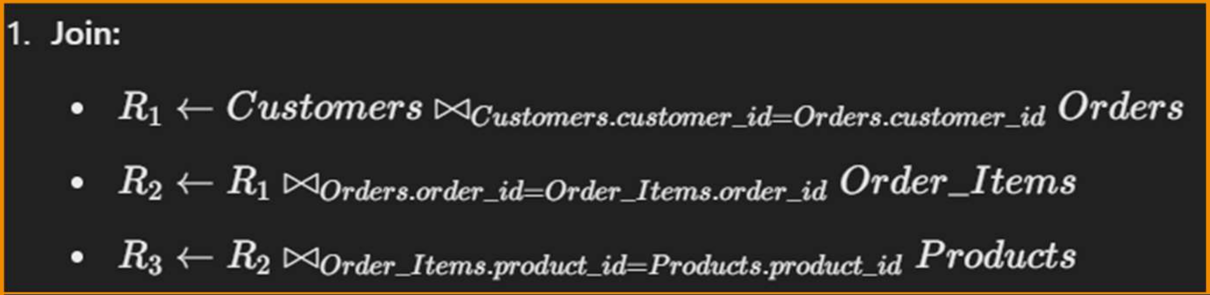
❖ Query Breakdown:

❑ Joins tables to combine relevant data.

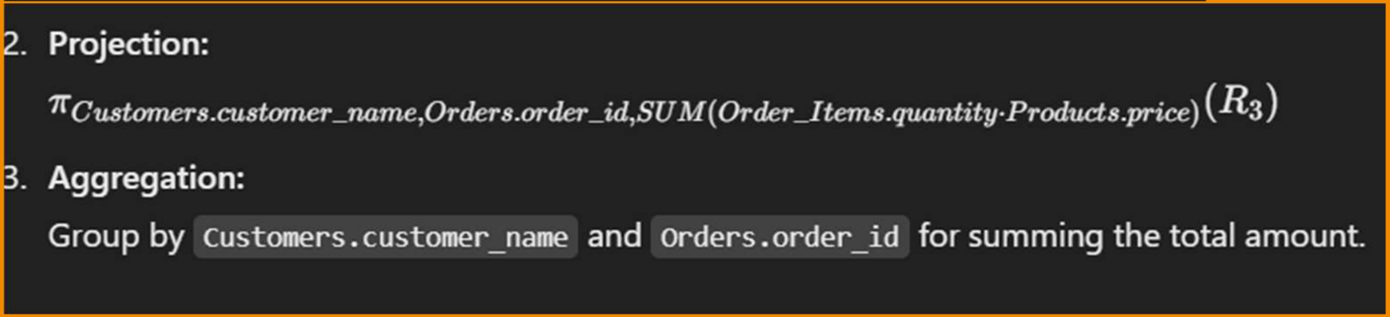
❑ Groups by customer_name and order_id to calculate the total amount spent per order

❑ Uses the SUM function to calculate the total amount for each order based on product prices and quantities.

Relational Algebra



1. **Join:**
   - $R_1 \leftarrow Customers \bowtie_{Customers.customer\_id=Orders.customer\_id} Orders$
   - $R_2 \leftarrow R_1 \bowtie_{Orders.order\_id=Order\_Items.order\_id} Order\_Items$
   - $R_3 \leftarrow R_2 \bowtie_{Order\_Items.product\_id=Products.product\_id} Products$



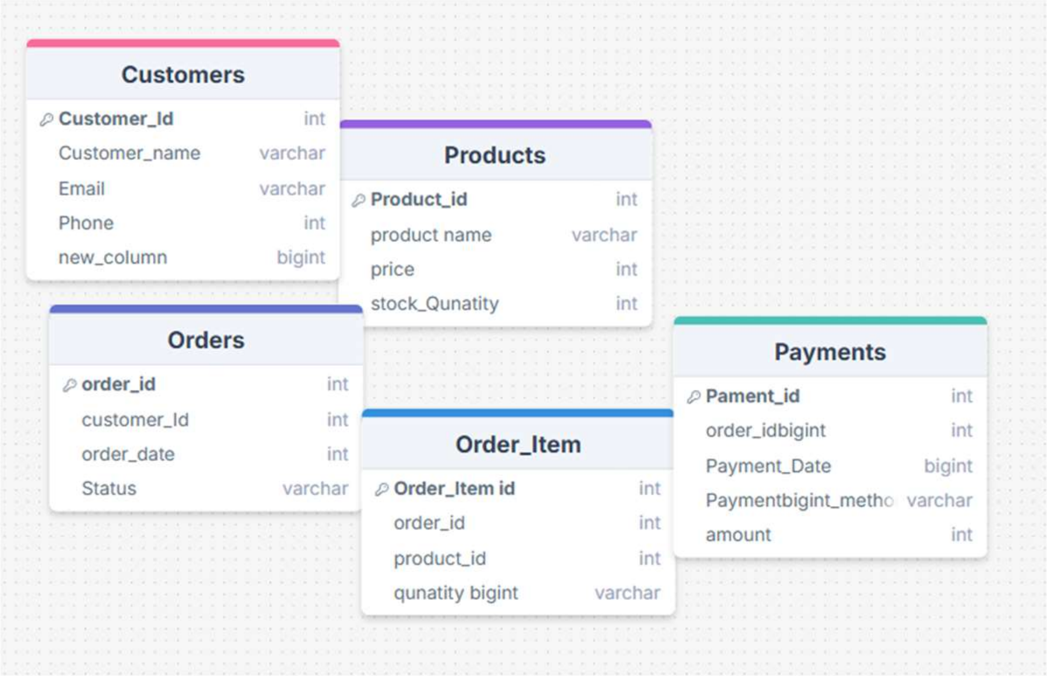
2. **Projection:**

   $\pi_{Customers.customer\_name,Orders.order\_id,SUM(Order\_Items.quantity \cdot Products.price)}(R_3)$

3. **Aggregation:**

   Group by `Customers.customer_name` and `Orders.order_id` for summing the total amount.

Schema Diagram



Customers	
🔑 **Customer_Id**	int
Customer_name	varchar
Email	varchar
Phone	int
new_column	bigint

Products	
🔑 **Product_id**	int
product name	varchar
price	int
stock_Qunatity	int

Orders	
🔑 **order_id**	int
customer_Id	int
order_date	int
Status	varchar

Order_Item	
🔑 **Order_Item id**	int
order_id	int
product_id	int
qunatity bigint	varchar

Payments	
🔑 **Pament_id**	int
order_idbigint	int
Payment_Date	bigint
Paymentbigint_metho	varchar
amount	int

Conclusion

An e-commerce database system effectively stores and organizes customer, product, order, and payment information, ensuring seamless transactions, efficient inventory management, and improved user experience for both customers and businesses.

Questions & answers

Thank You
Promod Chandra Das
ID: 231002005