**🔶 SECTION A – Real-Time Business Model**

Customers place orders for food from various restaurants. Each order can contain multiple food items (from potentially different restaurants).  
Each item is delivered by a delivery agent. Customers can make payments, leave feedback, and track delivery status.

**🔷 SECTION B – SQL Practical Questions**

**✅ Joins & Query Logic (1–5)**

**1.** Write a query to display:  
CustomerName, RestaurantName, FoodName, Quantity, SubPrice for all orders.

**2.** Display names of customers who have ordered **"Paneer Butter Masala"**.

**3.** Show delivery agents who handled more than **3 deliveries**.

**4.** List orders where the total amount is **greater than ₹500**.

**5.** Display orders placed from more than **one restaurant**.

**✅ Aggregation & Grouping (6–8)**

**6.** Show **total number of items ordered** per customer.

**7.** Show **total revenue generated per restaurant**.

**8.** Show **average delivery time** per delivery agent.

**✅ Subqueries & Filtering (9–11)**

**9.** Find customers who have **never placed an order**.

**10.** Display the **most ordered food item** by quantity.

**11.** List customers who made the **highest total payment**.

**✅ Views, Functions, Procedures (12–15)**

**12.** Create a **view** vw\_OrderSummary to show:  
OrderID, CustomerName, TotalAmount, OrderDate, PaymentStatus

**13.** Write a **scalar function** that takes @RestaurantID and returns **total orders placed** for that restaurant.

**14.** Create a stored procedure sp\_AssignDeliveryAgent that takes @OrderItemsID, @AgentID and inserts a new delivery.

**15.** Write a stored procedure sp\_NewOrder to insert a new order into Orders table.

**✅ Triggers & Constraints (16–17)**

**16.** Write a **trigger** to prevent placing an order if TotalAmount is less than ₹100.

**17.** Write a trigger to **update DeliveryStatus to 'Dispatched'** automatically when a delivery is inserted.

**✅ UDDTs & Indexes (18–19)**

**18.** Create a **user-defined data type** PhoneType as VARCHAR(15) and apply it to CustomerPhone.

**19.** Create a **non-clustered index** on RestaurantName in Restaurants, include Phone.

**✅ Business Rule Logic (20)**

**20.** Add a unique constraint so that a customer **can only give one feedback** per restaurant.