

# **Scenario Analysis and its Solution**

## **Scenario-8: Restaurant Management System.**

### **Requirements:**

A restaurant chain needs a database system to manage menu items, customer orders, and staff schedules. The system should store details about menu items, including name, description, price, ingredients, and category (e.g., appetizer, main course, dessert).

Customer orders should be tracked, recording details such as order date, items ordered, table number, and total amount. The system should also manage customer profiles, including personal details, contact information, and order history.

Staff schedules need to be maintained, recording shift timings, assigned tasks, and attendance. The restaurant wants to monitor staff performance, manage payroll, and handle leave requests.

Additionally, the system should support inventory management functions like tracking ingredient stock levels, generating purchase orders, and monitoring waste. The ability to generate reports on sales performance, popular menu items, and customer feedback is also essential.

### **Questions:**

- Find the menu item with the highest sales revenue.
- Calculate the average order value for each customer.
- Retrieve the minimum quantity of any ingredient in stock.
- Find the staff member who worked the most hours in the last month.
- List the top 5 most ordered menu categories.

**Tables required for this scenario:** As per our analysis 11 tables are required which are given below:

1. **Menu\_Items** (item\_id (Primary Key), name, description, price, ingredients, category)
2. **Orders**(order\_id (Primary Key),order\_date,customer\_id (Foreign Key referencing Customers),table\_number,total\_amount)
3. **Order\_Items**(order\_item\_id (Primary Key), order\_id (Foreign Key referencing Orders), item\_id (Foreign Key referencing Menu\_Items), quantity)
4. **Customers**(customer\_id (Primary Key), name, contact\_info, order\_history )
5. **Staff**(staff\_id (Primary Key), name,contact\_info, position)
6. **Staff\_Schedules**(schedule\_id (Primary Key), staff\_id (Foreign Key referencing Staff), shift\_date, shift\_start\_time, shift\_end\_time, assigned\_tasks, attendance)
7. **Inventory**(ingredient\_id (Primary Key), name, quantity\_in\_stock, unit, reorder\_level)
8. **Purchase Orders**(purchase\_order\_id (Primary Key), ingredient\_id (Foreign Key referencing Inventory), quantity\_ordered, order\_date, supplier\_id (Foreign Key referencing Suppliers))
9. **Suppliers**(supplier\_id (Primary Key), name, contact\_info)
10. **Payroll**(payroll\_id (Primary Key), staff\_id (Foreign Key referencing Staff), pay\_period, hours\_worked, pay\_amount, leave\_requests)
11. **Customer\_Feedback**(feedback\_id (Primary Key), customer\_id (Foreign Key referencing Customers), order\_id (Foreign Key referencing Orders), feedback,rating)

## Create all tables:

- i. CREATE TABLE Menu\_Items ( item\_id INT PRIMARY KEY  
AUTO\_INCREMENT,  
name VARCHAR(255) NOT NULL,  
description TEXT,  
price DECIMAL(10, 2) NOT NULL,  
ingredients TEXT,  
category VARCHAR(100) NOT NULL);
- ii. CREATE TABLE Orders ( order\_id INT PRIMARY KEY AUTO\_INCREMENT,  
order\_date DATE NOT NULL,  
customer\_id INT,  
table\_number INT,  
total\_amount DECIMAL(10, 2) NOT NULL,  
FOREIGN KEY (customer\_id) REFERENCES  
Customers(customer\_id));
- iii. CREATE TABLE Order\_Items (order\_item\_id INT PRIMARY KEY  
AUTO\_INCREMENT,  
order\_id INT,  
item\_id INT,  
quantity INT NOT NULL,  
FOREIGN KEY (order\_id) REFERENCES  
Orders(order\_id),  
FOREIGN KEY (item\_id) REFERENCES  
Menu\_Items(item\_id));
- iv. CREATE TABLE Customers (customer\_id INT PRIMARY KEY  
AUTO\_INCREMENT,  
name VARCHAR(255) NOT NULL,  
contact\_info VARCHAR(255));
- v. CREATE TABLE Staff (staff\_id INT PRIMARY KEY AUTO\_INCREMENT,  
name VARCHAR(255) NOT NULL,  
contact\_info VARCHAR(255),  
position VARCHAR(100));



[illegible]

### Five questions and their answers:

1. Find the menu item with the highest sales revenue.

**Ans:**

```
SELECT m.name, SUM(oi.quantity * m.price) AS total_revenue
FROM Order_Items oi
JOIN Menu_Items m ON oi.item_id = m.item_id
GROUP BY m.name
ORDER BY total_revenue DESC
LIMIT 1;
```

2. Calculate the average order value for each customer.

**Ans:**

```
SELECT c.name, AVG(o.total_amount) AS average_order_value
FROM Orders o
JOIN Customers c ON o.customer_id = c.customer_id
GROUP BY c.name;
```

3. Retrieve the minimum quantity of any ingredient in stock.

**Ans:**

```
SELECT name, MIN(quantity_in_stock) AS min_stock
FROM Inventory;
```

4. Find the staff member who worked the most hours in the last month.

**Ans:**

```
SELECT s.name, SUM(TIMESTAMPDIFF(HOUR, ss.shift_start_time,
                                   ss.shift_end_time)) AS total_hours
FROM Staff_Schedules ss
JOIN Staff s ON ss.staff_id = s.staff_id
WHERE ss.shift_date BETWEEN DATE_SUB(CURDATE(), INTERVAL 1
                                       MONTH) AND CURDATE()

GROUP BY s.name
ORDER BY total_hours DESC
LIMIT 1;
```

5. List the top 5 most ordered menu categories.

**Ans:**

```
SELECT m.category, COUNT(oi.item_id) AS order_count
FROM Order_Items oi
JOIN Menu_Items m ON oi.item_id = m.item_id
GROUP BY m.category
ORDER BY order_count DESC
LIMIT 5;
```