

- q1) Display restaurants with more than 5 food items in their menu
- q2) List users who have placed more than 3 orders
- q3) Show order details with user name, restaurant name, food item, and price
- q4) Find total revenue generated by each restaurant
- q5) Display orders where a coupon code was applied
- q6) Find top 3 most ordered food items based on order count
- q7) Find users who ordered food from more than one restaurant
- q8) Top-rated restaurant per category.

-- q1

```
SELECT r.res_id, r.name, COUNT(m.item_id) AS total_items
FROM Restaurant r
JOIN Menu m ON r.res_id = m.rest_id
GROUP BY r.res_id, r.name
HAVING COUNT(m.item_id) > 5;
```

res_id	name	total_items
1	Spice Hub	7

-- q2

```
SELECT Name from User
where user_Id in (SELECT user_id from Orders GROUP BY user_id having
count(user_id)>3);
```

Name
Amit Patil

-- q3

```
SELECT U.name as user_name,R.name as rest_name,
m.name as item_name,OI.price from Orders O
JOIN OrderItems OI ON OI.order_id=O.order_id
JOIN Menu m ON OI.item_id=m.item_id
JOIN User U ON U.user_id = O.user_id
JOIN Restaurant R ON R.res_id=O.restaurant_id;
```

user_name	rest_name	item_name	price
Amit Patil	Spice Hub	Veg Pizza	398.00
Amit Patil	Food Plaza	Chicken Biryani	498.00
Sneha Kulkarni	Green Leaf	Veg Biryani	199.00
Amit Patil	Tandoori Town	Butter Chicken	299.00
Rohit Kale	Urban Bites	Paneer Burger	149.00
Amit Patil	Desi Dhaba	Masala Dosa	129.00
Vikas More	Punjabi Treat	Fried Rice	358.00
Neha Patil	South Spice	Hakka Noodles	189.00
Sanket Shah	Biryani House	Samosa	87.00
Kunal Joshi	Quick Eats	Samosa	58.00

-- q4

```
SELECT R.name , SUM(O.total_amount) as Revenue
FROM Restaurant R
LEFT JOIN Orders O ON R.res_id=O.restaurant_id
GROUP BY R.name;
```

name	Revenue
Spice Hub	399.00
Food Plaza	499.00
Green Leaf	299.00
Tandoori Town	349.00
Urban Bites	199.00
Desi Dhaba	229.00
Punjabi Treat	449.00
South Spice	159.00
Biryani House	99.00
Quick Eats	59.00
Rama international	NULL

-- q5

```
SELECT * from Orders
where offer_id is NOT NULL;
```

```
+-----+-----+-----+-----+-----+-----+-----+
| order_id | user_id | restaurant_id | offer_id | order_status | total_amount | order_time | deliver_time |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | 1 | 1 | 1 | success | 399.00 | 2026-01-22 10:07:33 | 2026-01-20 13:00:00 |
| 2 | 1 | 2 | 2 | success | 499.00 | 2026-01-22 10:07:33 | 2026-01-21 14:00:00 |
| 3 | 3 | 3 | 3 | pending | 299.00 | 2026-01-22 10:07:33 | NULL |
| 4 | 1 | 4 | 4 | success | 349.00 | 2026-01-22 10:07:33 | 2026-01-22 15:30:00 |
| 5 | 5 | 5 | 5 | failed | 199.00 | 2026-01-22 10:07:33 | NULL |
| 6 | 1 | 6 | 6 | success | 229.00 | 2026-01-22 10:07:33 | 2026-01-23 16:00:00 |
| 7 | 7 | 7 | 7 | success | 449.00 | 2026-01-22 10:07:33 | 2026-01-24 18:00:00 |
| 8 | 8 | 8 | 8 | pending | 159.00 | 2026-01-22 10:07:33 | NULL |
| 9 | 9 | 9 | 9 | success | 99.00 | 2026-01-22 10:07:33 | 2026-01-25 19:00:00 |
| 10 | 10 | 10 | 10 | success | 59.00 | 2026-01-22 10:07:33 | 2026-01-26 20:00:00 |
+-----+-----+-----+-----+-----+-----+
```

-- q6

```
SELECT name
FROM Menu
WHERE item_id IN (SELECT item_id FROM OrderItems GROUP BY item_id ORDER BY
COUNT(item_id) DESC
```

```
)  
LIMIT 3;
```

name
Veg Pizza
Paneer Burger
Chicken Biryani

-- q7

```
select name from User
where user_id in(select user_id from orders
group by user_id having count(distinct restaurant_id)>1);
```

name
Amit Patil

-- q8

```
SELECT rest_name, res_type, Total_Rating
FROM ( SELECT
        R.name AS rest_name,
        R.res_type,
        SUM(rr.rating) AS Total_Rating,
        DENSE_RANK() OVER (
```

```

        PARTITION BY R.res_type
        ORDER BY SUM(rr.rating) DESC
    ) AS rnk
FROM Restaurant R
JOIN REST_RATING rr
    ON rr.restaurant_id = R.res_id
GROUP BY R.name, R.res_type
) ranked
WHERE rnk = 1;

```

rest_name	res_type	Total_Rating
Green Leaf	veg	11
Food Plaza	non-veg	2
Tandoori Town	non-veg	2

SQL:

```

-- User
CREATE TABLE User (
    User_Id BIGINT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Email VARCHAR(150) NOT NULL UNIQUE,
    Phone VARCHAR(15) NOT NULL UNIQUE,
    Password VARCHAR(255),
    Role ENUM('Rest. manager', 'Customer', 'Delivery Agent') DEFAULT 'Customer',
    IsActive BOOLEAN DEFAULT TRUE,
    Created_At TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    Updated_At TIMESTAMP DEFAULT CURRENT_TIMESTAMP
        ON UPDATE CURRENT_TIMESTAMP
);
-- Restaurant
CREATE TABLE Restaurant (
    res_id BIGINT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(150) NOT NULL,
    phone VARCHAR(15) UNIQUE,
    email VARCHAR(150) NOT NULL,
    opening_time TIME DEFAULT '09:00:00',

```

```
        closing TIME DEFAULT '09:00:00',
        res_manager_id BIGINT,
        res_type ENUM('veg', 'non-veg') DEFAULT 'veg',
        FOREIGN KEY (res_manager_id) REFERENCES User(User_Id)
    );
```

-- Address

```
CREATE TABLE Address (
    add_id BIGINT AUTO_INCREMENT PRIMARY KEY,
    user_id BIGINT,
    Street VARCHAR(50) NOT NULL,
    Area VARCHAR(50) NOT NULL,
    City VARCHAR(50) NOT NULL,
    State VARCHAR(50) NOT NULL,
    Pincode CHAR(6) NOT NULL,
    Type ENUM('home', 'office', 'current', 'rest'),
    FOREIGN KEY (user_id) REFERENCES User(User_Id)
);
```

-- Menu

```
CREATE TABLE Menu (
    item_id BIGINT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    Description VARCHAR(255),
    actual_prize DECIMAL(7,2) NOT NULL,
    striked_price DECIMAL(7,2) NOT NULL,
    availability BOOLEAN DEFAULT TRUE,
    rest_id BIGINT,
    type ENUM('veg', 'non-veg') NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP
        ON UPDATE CURRENT_TIMESTAMP,
    created_by BIGINT,
    FOREIGN KEY (rest_id) REFERENCES Restaurant(res_id),
    FOREIGN KEY (created_by) REFERENCES User(User_Id),
    CHECK (striked_price > actual_prize)
);
```

-- Offers

```
CREATE TABLE Offers (
    Offer_Id BIGINT AUTO_INCREMENT PRIMARY KEY,
    Code VARCHAR(50),
    Description VARCHAR(150),
    discount_value INT,
    Discount ENUM('Flat', 'percentage'),
    Valid_From DATETIME,
    Valid_till DATETIME,
    CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    UpdatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP
```

```

    ON UPDATE CURRENT_TIMESTAMP
);

-- Orders
CREATE TABLE Orders (
    order_id BIGINT AUTO_INCREMENT PRIMARY KEY,
    user_id BIGINT,
    restaurant_id BIGINT,
    offer_id BIGINT,
    order_status ENUM('success', 'failed', 'pending') DEFAULT 'pending',
    total_amount DECIMAL(10,2) NOT NULL,
    order_time DATETIME DEFAULT CURRENT_TIMESTAMP,
    deliver_time DATETIME,
    FOREIGN KEY (user_id) REFERENCES User(User_Id),
    FOREIGN KEY (restaurant_id) REFERENCES Restaurant(res_id),
    FOREIGN KEY (offer_id) REFERENCES Offers(Offer_Id)
);

-- OrderItems
CREATE TABLE OrderItems (
    order_items_id BIGINT AUTO_INCREMENT PRIMARY KEY,
    order_id BIGINT,
    item_id BIGINT,
    quantity INT NOT NULL,
    price DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (order_id) REFERENCES Orders(order_id),
    FOREIGN KEY (item_id) REFERENCES Menu(item_id)
);

-- Payment
CREATE TABLE Payment (
    p_id BIGINT AUTO_INCREMENT PRIMARY KEY,
    order_id BIGINT,
    trans_id BIGINT NOT NULL,
    payment_mode ENUM('cash', 'upi', 'card') NOT NULL,
    payment_status ENUM('success', 'failed', 'pending') NOT NULL,
    payment_date_and_time DATETIME NOT NULL,
    FOREIGN KEY (order_id) REFERENCES Orders(order_id)
);

-- Menu_Review
CREATE TABLE Menu_Review (
    mr_id BIGINT PRIMARY KEY,
    menu_item_id BIGINT,
    ratings ENUM('satisfied', 'not satisfied', 'neutral') DEFAULT 'neutral',
    reviews VARCHAR(255),
    user_id BIGINT,
    FOREIGN KEY (menu_item_id) REFERENCES Menu(item_id),

```

```

    FOREIGN KEY (user_id) REFERENCES User(User_Id)
);

-- rest_rating
CREATE TABLE REST_RATING(
rest_rating_id BIGINT AUTO_INCREMENT PRIMARY KEY,
restaurant_id BIGINT,
rating ENUM('1','2','3','4','5') DEFAULT 1,
user_id BIGINT

);

-- User
INSERT INTO User (Name, Email, Phone, Password, Role) VALUES
('Amit Patil', 'amit@gmail.com', '9000000001', 'pass123', 'Customer'),
('Rahul Deshmukh', 'rahul@gmail.com', '9000000002', 'pass123', 'Customer'),
('Sneha Kulkarni', 'sneha@gmail.com', '9000000003', 'pass123', 'Customer'),
('Pooja Joshi', 'pooja@gmail.com', '9000000004', 'pass123', 'Customer'),
('Rohit Kale', 'rohit@gmail.com', '9000000005', 'pass123', 'Delivery Agent'),
('Akash Pawar', 'akash@gmail.com', '9000000006', 'pass123', 'Delivery Agent'),
('Vikas More', 'vikas@gmail.com', '9000000007', 'pass123', 'Rest. manager'),
('Neha Patil', 'neha@gmail.com', '9000000008', 'pass123', 'Rest. manager'),
('Sanket Shah', 'sanket@gmail.com', '9000000009', 'pass123', 'Customer'),
('Kunal Joshi', 'kunal@gmail.com', '9000000010', 'pass123', 'Customer');

-- Restaurant
INSERT INTO Restaurant (name, phone, email, opening_time, closing, res_manager_id, res_type) VALUES
('Spice Hub', '8800000001', 'spice@gmail.com', '09:00', '22:00', 7, 'veg'),
('Food Plaza', '8800000002', 'plaza@gmail.com', '10:00', '23:00', 8, 'non-veg'),
('Green Leaf', '8800000003', 'leaf@gmail.com', '08:00', '21:00', 7, 'veg'),
('Tandoori Town', '8800000004', 'tandoor@gmail.com', '11:00', '23:30', 8, 'non-veg'),
('Urban Bites', '8800000005', 'urban@gmail.com', '09:30', '22:30', 7, 'veg'),
('Desi Dhaba', '8800000006', 'dhaba@gmail.com', '10:00', '00:00', 8, 'non-veg'),
('Punjabi Treat', '8800000007', 'punjabi@gmail.com', '09:00', '23:00', 7, 'veg'),
('South Spice', '8800000008', 'south@gmail.com', '08:00', '21:30', 8, 'veg'),
('Biryani House', '8800000009', 'biryani@gmail.com', '11:00', '23:59', 7, 'non-veg'),
('Quick Eats', '8800000010', 'quick@gmail.com', '07:00', '21:00', 8, 'veg'),
('Rama international', '8800000012', 'quick@gmail.com', '07:00', '21:00', 8, 'veg');

-- Address
INSERT INTO Address (user_id, Street, Area, City, State, Pincode, Type) VALUES
(1, 'MG Road', 'Camp', 'Pune', 'Maharashtra', '411001', 'home'),

```

```
(2,'FC Road','Shivaji Nagar','Pune','Maharashtra','411005','home'),
(3,'College Rd','Gangapur','Nashik','Maharashtra','422005','home'),
(4,'CIDCO','Ambad','Nashik','Maharashtra','422010','office'),
(5,'MIDC','Satpur','Nashik','Maharashtra','422007','current'),
(6,'Tidke Colony','Canada','Nashik','Maharashtra','422002','home'),
(7,'Paithan Rd','Cidco','Aurangabad','Maharashtra','431001','rest'),
(8,'Jalna Rd','Seven Hills','Aurangabad','Maharashtra','431003','rest'),
(9,'JM Road','Deccan','Pune','Maharashtra','411004','office'),
(10,'Ambegaon','Katraj','Pune','Maharashtra','411046','home');
```

-- Menu

```
INSERT INTO Menu
(name, Description, actual_prize, strucked_price, availability, rest_id, type, created_by)
VALUES
('Veg Pizza','Cheese loaded pizza',199,249,1,1,'veg',7),
('Paneer Burger','Paneer patty burger',149,199,1,1,'veg',7),
('Chicken Biryani','Hyderabadi style',249,299,1,1,'non-veg',8),
('Veg Biryani','Aromatic basmati rice',199,249,1,1,'veg',7),
('Butter Chicken','Creamy gravy',299,349,1,1,'non-veg',8),
('Masala Dosa','South Indian dosa',129,159,1,1,'veg',8),
('Fried Rice','Veg fried rice',179,219,1,1,'veg',7),
('Hakka Noodles','Chinese noodles',189,229,1,5,'veg',7),
('Pav Bhaji','Mumbai special',149,199,1,6,'veg',8),
('Samosa','Crispy snack',29,39,1,10,'veg',8);
```

-- Offers

```
INSERT INTO Offers
(Code, Description, discount_value, Discount, Valid_From, Valid_till)
VALUES
('NEW10','New user offer',10,'percentage','2026-01-01','2026-03-01'),
('SAVE50','Flat discount',50,'Flat','2026-01-10','2026-02-10'),
('WEEKEND20','Weekend offer',20,'percentage','2026-01-05','2026-02-05'),
('FEST15','Festival offer',15,'percentage','2026-01-15','2026-02-15'),
('FLAT100','Flat 100 off',100,'Flat','2026-01-20','2026-02-20'),
('HOT30','Hot deal',30,'percentage','2026-02-01','2026-03-01'),
('MEGA25','Mega discount',25,'percentage','2026-01-01','2026-01-31'),
('FIRST50','First order',50,'Flat','2026-01-01','2026-04-01'),
('COMBO20','Combo offer',20,'percentage','2026-02-05','2026-02-28'),
('DAILY5','Daily deal',5,'percentage','2026-01-01','2026-12-31');
```

-- Orders

```
INSERT INTO Orders
(user_id, restaurant_id, offer_id, order_status, total_amount, deliver_time)
VALUES
(1,1,1,'success',399,'2026-01-20 13:00'),
(1,2,2,'success',499,'2026-01-21 14:00'),
(3,3,3,'pending',299,NULL),
(1,4,4,'success',349,'2026-01-22 15:30'),
(5,5,5,'failed',199,NULL),
(1,6,6,'success',229,'2026-01-23 16:00'),
(7,7,7,'success',449,'2026-01-24 18:00'),
(8,8,8,'pending',159,NULL),
(9,9,9,'success',99,'2026-01-25 19:00'),
(10,10,10,'success',59,'2026-01-26 20:00');
```

-- OrderItems

```
INSERT INTO OrderItems (order_id, item_id, quantity, price) VALUES
(1,1,2,398),
(2,3,2,498),
(3,4,1,199),
(4,5,1,299),
(5,2,1,149),
(6,6,1,129),
(7,7,2,358),
(8,8,1,189),
(9,10,3,87),
(10,10,2,58);
```

-- Payment

```
INSERT INTO Payment
(order_id, trans_id, payment_mode, payment_status, payment_date_and_time)
VALUES
(1,10001,'upi','success','2026-01-20 12:45'),
(2,10002,'card','success','2026-01-21 13:50'),
(3,10003,'upi','pending','2026-01-22 14:10'),
(4,10004,'cash','success','2026-01-22 15:40'),
(5,10005,'upi','failed','2026-01-23 16:10'),
(6,10006,'card','success','2026-01-23 16:20'),
(7,10007,'upi','success','2026-01-24 18:10'),
(8,10008,'cash','pending','2026-01-25 19:20'),
(9,10009,'upi','success','2026-01-25 19:30'),
(10,10010,'cash','success','2026-01-26 20:10');
```

-- Menu\_Review

```
INSERT INTO Menu_Review (mr_id, menu_item_id, ratings, reviews, user_id) VALUES
```

```
(1,1,'satisfied','Very tasty',1),
(2,2,'neutral','Good',2),
(3,3,'satisfied','Excellent',3),
(4,4,'not satisfied','Too spicy',4),
(5,5,'satisfied','Loved it',5),
(6,6,'neutral','Okayish',6),
(7,7,'satisfied','Worth price',7),
(8,8,'not satisfied','Cold food',8),
(9,9,'satisfied','Best pav bhaji',9),
(10,10,'neutral','Average',10);
```

```
-- rating data
```

```
INSERT INTO REST_RATING (restaurant_id,rating,user_id) VALUES
('3','2','1'),
('3','4','1'),('3','5','2'),('2','2','3'),('1','2','3'),('4','2','2');
```

```
-- Q1
```

```
SELECT r.res_id, r.name, COUNT(m.item_id) AS total_items
FROM Restaurant r
JOIN Menu m ON r.res_id = m.rest_id
GROUP BY r.res_id, r.name
HAVING COUNT(m.item_id) > 5;
```

```
-- q2
```

```
SELECT Name from User
where user_id in (SELECT user_id from Orders GROUP BY user_id having
count(user_id)>3);
```

```
-- q3
```

```
SELECT U.name as user_name,R.name as rest_name,
m.name as item_name,OI.price from Orders O
JOIN OrderItems OI ON OI.order_id=O.order_id
JOIN Menu m ON OI.item_id=m.item_id
JOIN User U ON U.user_id = O.user_id
JOIN Restaurant R ON R.res_id=O.restaurant_id;
```

```
-- q4
```

```
SELECT R.name , SUM(O.total_amount) as Revenue
FROM Restaurant R
LEFT JOIN Orders O ON R.res_id=O.restaurant_id
GROUP BY R.name;
```

```
-- q5
```

```
SELECT * from Orders
where offer_id is NOT NULL;
```

```
-- q6
SELECT name
FROM Menu
WHERE item_id IN (SELECT item_id FROM OrderItems GROUP BY item_id ORDER BY
COUNT(item_id) DESC

)
LIMIT 3;

-- q7
select name from User
where user_id in(select user_id from orders
group by user_id having count(distinct restaurant_id)>1);

-- q8
SELECT rest_name, res_type, Total_Rating
FROM (
    SELECT
        R.name AS rest_name,
        R.res_type,
        SUM(rr.rating) AS Total_Rating,
        DENSE_RANK() OVER (
            PARTITION BY R.res_type
            ORDER BY SUM(rr.rating) DESC
        ) AS rnk
    FROM Restaurant R
    JOIN REST_RATING rr
        ON rr.restaurant_id = R.res_id
    GROUP BY R.name, R.res_type
) ranked
WHERE rnk = 1;
```