QUERIES

-- 1. sort restaurants by their average rating

```
SELECT Restaurantname, R.avg_rating FROM Restaurant NATURAL JOIN
(SELECT restaurantid,avg(ratings) as avg_rating FROM restaurant_ratings
GROUP BY restaurantid) AS R
ORDER BY avg_rating DESC;
```

--2 give the city wise best restaurant's list (best in terms of avg_rating)?

```
CREATE VIEW V1 AS (
SELECT restaurantname,restaurantid,restaurant_city ,R.avg_rating FROM restaurant
NATURAL JOIN
(SELECT restaurantid,avg(ratings) as avg_rating FROM restaurant_ratings
GROUP BY restaurantid) AS R );

CREATE VIEW V2 AS (
(SELECT R.restaurant_city,max(avg_rating) AS MAX_RATING from (
SELECT restaurant_city, restaurantid,avg(ratings) as avg_rating FROM restaurant natural
join restaurant_ratings
GROUP BY restaurant_city,restaurantid
)as R group by R.restaurant_city));

SELECT v1.restaurantname,v1.restaurantid,v2.restaurant_city from
v1,v2 where v1.restaurant_city=v2.restaurant_city and v1.avg_rating=v2.max_rating;
```

--3 Find the top 5 most ordered items along with their total quantity sold

```
SELECT itemname,R.itemid,R.total_qty FROM items join
(SELECT itemid,sum(quantity) as total_qty
FROM ordered_item
GROUP BY itemid
ORDER BY total_qty DESC
LIMIT 5) AS R on items.itemid=r.itemid ORDER BY total_qty DESC;
```

--4. Find the top 3 most ordered items by a specific user by quantity ordered:

```
SELECT itemname,R.itemid,R.total_orders FROM items JOIN
(SELECT itemid,sum(quantity) as total_orders FROM ordered_item
JOIN orders ON ordered_item.orderid =orders.orderid
WHERE userid = 'user001'
```

```
GROUP BY itemid
ORDER BY total_orders DESC
LIMIT 3) AS R ON items.itemid=R.itemid;


--5. Give the list of all the past orders for particular user

SELECT username,"DATE",orderid,paymentid,amount
FROM users NATURAL JOIN orders NATURAL JOIN payment
WHERE userid='user016';

---6. sort cuisines by price (asending/decending)

SELECT itemname,price,restaurantname
FROM items NATURAL JOIN restaurant ORDER BY price ASC;



--7.Find the total revenue generated by each restaurant and list them in descending order by
total revenue

SELECT restaurantid,restaurantname,SUM(price*quantity) AS total_revenue FROM
ordered_item
NATURAL JOIN items
NATURAL JOIN restaurant
GROUP BY restaurantid,restaurantname
ORDER BY total_revenue DESC;


--8.Find the total number of orders delivered by each delivery guy

SELECT delievery_partner_id,dpname,COUNT(orderid) AS TotalOrdersDelivered
FROM delievery_guy
NATURAL JOIN orders
GROUP BY delievery_partner_id,dpname;


--9.List the users who have placed orders for items from at least 2 different categories:
SELECT  users.userid,users.username FROM users
JOIN orders ON users.userid = orders.userid
JOIN ordered_item ON orders.orderid=ordered_item.orderid
JOIN items ON ordered_item.itemid =items.itemid
JOIN category ON items.categoryid = category.categoryid
GROUP BY users.userid,users.username
HAVING COUNT(DISTINCT category.categoryid)>=2;


--10. list top  5 most popular category
```

```
SELECT categoryid,categoryname,COUNT(orderid) AS totalorders FROM  ordered_item
NATURAL JOIN items NATURAL JOIN category
GROUP BY categoryid,categoryname
ORDER BY totalorders DESC
LIMIT 5;

--11.  list top  5 most popular items
SELECT itemid,itemname,COUNT(orderid) AS totalorders FROM  ordered_item NATURAL
JOIN items
GROUP BY itemid,itemname
ORDER BY totalorders DESC
LIMIT 5;

--12.give the list of items that was ordered with particular orderid

SELECT * FROM ordered_item where orderid = 'order020';


--13.Find users who have not placed any orders yet:
SELECT UserID, Username
FROM Users
WHERE UserID NOT IN (SELECT DISTINCT UserID FROM Orders);

--14. what payment method was use for particular order
SELECT orderid,payment_method FROM orders natural join payment
WHERE orderid='order001';

--15.number of offers running on the particular restaurant (ex. restaurant id= "res002")
SELECT restaurantname,offername FROM restaurant
NATURAL JOIN have_offer
NATURAL JOIN offer
WHERE restaurantid='res002';

--16. Give the best-selling items for a particular restaurant  (ex. restaurant id= "res005")

SELECT itemid,itemname,sum(quantity) AS tot_order FROM ordered_item
NATURAL JOIN items
WHERE restaurantid ='res005'
GROUP BY itemid,itemname
ORDER BY tot_order DESC
LIMIT 5;

-- 17.which items from restaurant is not sold till  (ex. restaurant id= "res001").

SELECT itemid FROM items
WHERE restaurantid = 'res009' AND itemid NOT IN
(SELECT DISTINCT itemid FROM ordered_item
```

```
NATURAL JOIN items
WHERE restaurantid = 'res009');

--18.For given date number of total order (ex. DATE =15-04-2024 ).

--full details of orders
SELECT * FROM orders where "DATE" = '2024-04-15';
--number of oreders
SELECT count(orderid)FROM orders where "DATE" = '2024-04-15';

-- 19.what is amount spent by user (overall)

SELECT username ,sum(amount) as Total_Spend FROM(users NATURAL JOIN orders)
NATURAL JOIN Payment
WHERE userid='user005' group by username;

--20.What is number of Premium member ?
SELECT count(userid) AS No_Premium_User
FROM users
WHERE is_premium_user='Y';

--21.Give the best selling items for particular restaurant in specific date interval  (ex.
restaurant id= "res005")

SELECT itemid,itemname,sum(quantity) AS tot_order FROM
(ordered_item NATURAL JOIN orders)
NATURAL JOIN items
WHERE restaurantid ='res005' AND "DATE" > '2024-04-01' AND  "DATE" < '2024-04-30'
GROUP BY itemid,itemname
ORDER BY tot_order DESC
LIMIT 5;
```