BASIC SQL PROJECT

ZOMATO FOOD DELIVERY

In this project I, implemented all the knowledge gained from my whole SQL practice. In this project all the queries will be solved that will be according to the how online food delivery system retrieve all the data either of food or as customer details about the order. This project consist of all the details regarding online delivery of food like top restaurants, top food , total sales , total profit/month , etc.

DATA –

USERS -

user_id	name	email	password
1	Sagar	Sagar@gmail.com	oxqwe
2	Anjali	Anjali@gmail.com	23rfw
3	Vrushika	vrushika@gmail.com	wqnm6
4	Ankit	ankit@gmail.com	34klb
5	Nehal	nehal@gmail.com	02rxt
6	Vayu	Vayu@gmail.com	mkvgf
7	Dev	Dev@gmail.com	67pof
NULL	NULL	NULL	NULL

FOOD -

f_id	f_name	type
1	Non-veg Pizza	Non-veg
2	Veg Pizza	Veg
3	Choco Lava cake	Veg
4	Chicken Wings	Non-veg
5	Chicken Popcorn	Non-veg
6	Rice Meal	Veg
7	Roti meal	Veg
8	Masala Dosa	Veg
9	Rava Idli	Veg
10	Schezwan Noodles	Veg
11	Veg Manchurian	Veg

ORDERS -

O

order_id	user_id	r_id	amount	date	partner_id	delivery_time	delivery_rating	restaurant_rating
1001	1	1	550	2022-05-10	1	25	5	3
1002	1	2	415	2022-05-26	1	19	5	2
1003	1	3	240	2022-06-15	5	29	4	
1004	1	3	240	2022-06-29	4	42	3	5
1005	1	3	220	2022-07-10	1	58	1	4
1006	2	1	950	2022-06-10	2	16	5	
1007	2	2	530	2022-06-23	3	60	1	5
1008	2	3	240	2022-07-07	5	33	4	5
1009	2	4	300	2022-07-17	4	41	1	
1010	2	5	650	2022-07-31	1	67	1	4
1011	3	1	450	2022-05-10	2	25	3	1
1012	3	4	180	2022-05-20	5	33	4	1
1013	3	2	230	2022-05-30	4	45	3	
1014	3	2	230	2022-06-11	2	55	1	2
1015	3	2	230	2022-06-22	3	21	5	
1016	4	4	300	2022-05-15	3	31	5	5
1017	4	4	300	2022-05-30	1	50	1	
1018	4	4	400	2022-06-15	2	40	3	5
1019	4	5	400	2022-06-30	1	70	2	4
1020	4	5	400	2022-07-15	3	26	5	3
1021	5	1	550	2022-07-01	5	22	2	
1022	5	1	550	2022-07-08	1	34	5	1

(

id	order_id	f_id
1	1001	1
2	1001	3
3	1002	4
4	1002	3
5	1003	6
6	1003	3
7	1004	6
8	1004	3
9	1005	7
10	1005	3
11	1006	1
12	1006	2
13	1006	3
14	1007	4
15	1007	3
16	1008	6
17	1008	3
18	1009	8
19	1009	9

r_id	r_name	cuisine
1	dominos	Italian
2	kfc	American
3	box8	North Indian
4	Dosa Plaza	South Indian
5	China Town	Chinese

DELIVERY PARTNER -

partner_id	partner_name
1	Suresh
2	Amit
3	Lokesh
4	Kartik
5	Gyandeep

MENU -

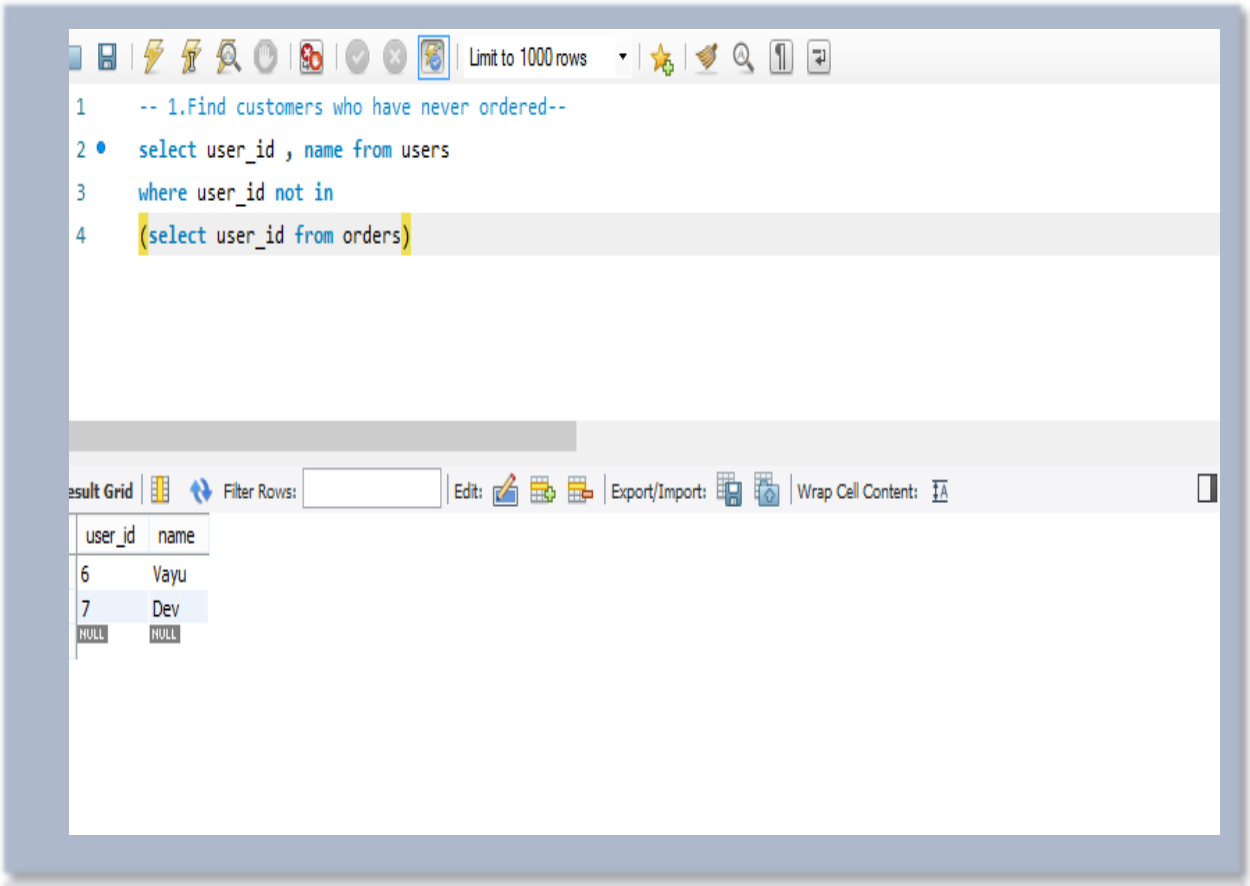
menu_id	r_id	f_id	price
1	1	1	450
2	1	2	400
3	1	3	100
4	2	3	115
5	2	4	230
6	2	5	300
7	3	3	80
8	3	6	160
9	3	7	140
10	4	6	230
11	4	8	180
12	4	9	120
13	5	6	250
14	5	10	220
15	5	11	180

QUERIES -

1.Find customers who have never ordered.

```
SELECT user_id, name FROM users
 WHERE user_id not in
(Select user_id from orders);
```

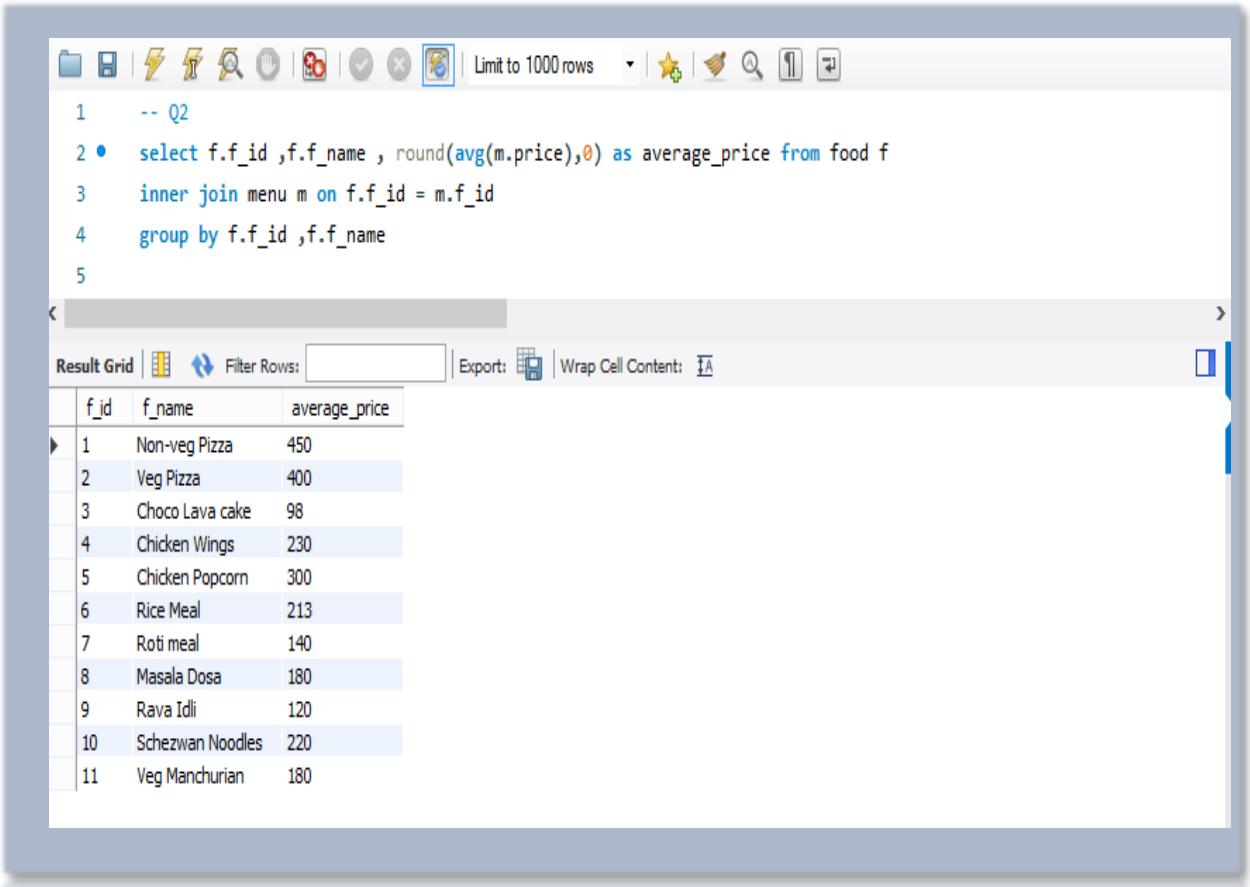
OUTPUT -



2. Average Price/dish.

```
SELECT f.f_id, f.f_name, ROUND(avg(m.price),0) as
average_price FROM food f
JOIN menu m ON f.f_id = m.f_id
GROUP BY f.f_id, f.f_name;
```

OUTPUT -



f_id	f_name	average_price
1	Non-veg Pizza	450
2	Veg Pizza	400
3	Choco Lava cake	98
4	Chicken Wings	230
5	Chicken Popcorn	300
6	Rice Meal	213
7	Roti meal	140
8	Masala Dosa	180
9	Rava Idli	120
10	Schezwan Noodles	220
11	Veg Manchurian	180

3. Find the top restaurant in terms of the number of orders for a given month.

```
SELECT monthname(o.date) as order_month,
count(o.order_id) AS total_orders, r.r_name
FROM order_details od
JOIN orders o on od.order_id = o.order_id
JOIN restaurants r ON o.r_id = r.r_id
GROUP BY o.date, r.r_id, r.r_name
ORDER BY extract(month from o.date)
```

OUTPUT -

Limit to 1000 rows

```
-- Q3
 select  monthname(o.date) as order_month,
count(o.order_id) as total_orders , r.r_name
from order_details od
inner join orders o on od.order_id = o.order_id
inner join restaurants r on o.r_id = r.r_id
group by o.date , r.r_id , r.r_name
order by  extract(month from o.date)
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

order_month	total_orders	r_name
May	3	dominos
May	2	kfc
May	1	Dosa Plaza
May	1	kfc
May	2	Dosa Plaza
May	2	Dosa Plaza
June	2	box8
June	2	box8
June	3	dominos
June	2	kfc
June	1	kfc
June	1	kfc
June	2	Dosa Plaza
June	2	China Town
July	2	box8
July	2	box8
July	2	Dosa Plaza
July	3	China Town

4. Restaurants with monthly sales greater than 1000.

```
SELECT o.r_id, r.r_name , sum(o.amount) AS total_amounts,
monthname(o.date) AS order_month
FROM orders o
JOIN restaurants r on o.r_id = r.r_id
GROUP BY o.r_id , r.r_name,  monthname(o.date)
HAVING sum(o.amount) > 500
ORDER BY o.r_id ASC
```

OUTPUT -

Limit to 1000 rows

```
-- Q4
SELECT o.r_id , r.r_name , sum(o.amount) as total_amounts , monthname(o.date) as order_month
from orders o
inner join restaurants r on o.r_id = r.r_id
group by o.r_id , r.r_name ,  monthname(o.date)
having sum(o.amount) > 500
order by o.r_id asc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

r_id	r_name	total_amounts	order_month
1	dominos	1100	July
1	dominos	950	June
1	dominos	1000	May
2	kfc	1935	July
2	kfc	990	June
2	kfc	645	May
4	Dosa Plaza	780	May
5	China Town	1050	July

5. Show all orders with order details for a particular customer in a particular date range.

```
SELECT o.user_id, u.name, r.r_name, o.date, f.f_name FROM orders o
JOIN users u ON o.user_id = u.user_id
JOIN restaurants r ON o.r_id = r.r_id
JOIN menu m ON r.r_id = m.r_id
JOIN food f ON m.f_id = f.f_id
GROUP BY o.user_id, u.name, r.r_name, o.date, f.f_name;
```

OUTPUT -

Limit to 1000 rows

```
-- Q5
SELECT o.user_id ,u.name , r.r_name , o.date , f.f_name from orders o
inner join users u on o.user_id = u.user_id
inner join restaurants r on o.r_id = r.r_id
inner join menu m on r.r_id = m.r_id
inner join food f on m.f_id = f.f_id
group by o.user_id ,u.name , r.r_name , o.date , f.f_name
```

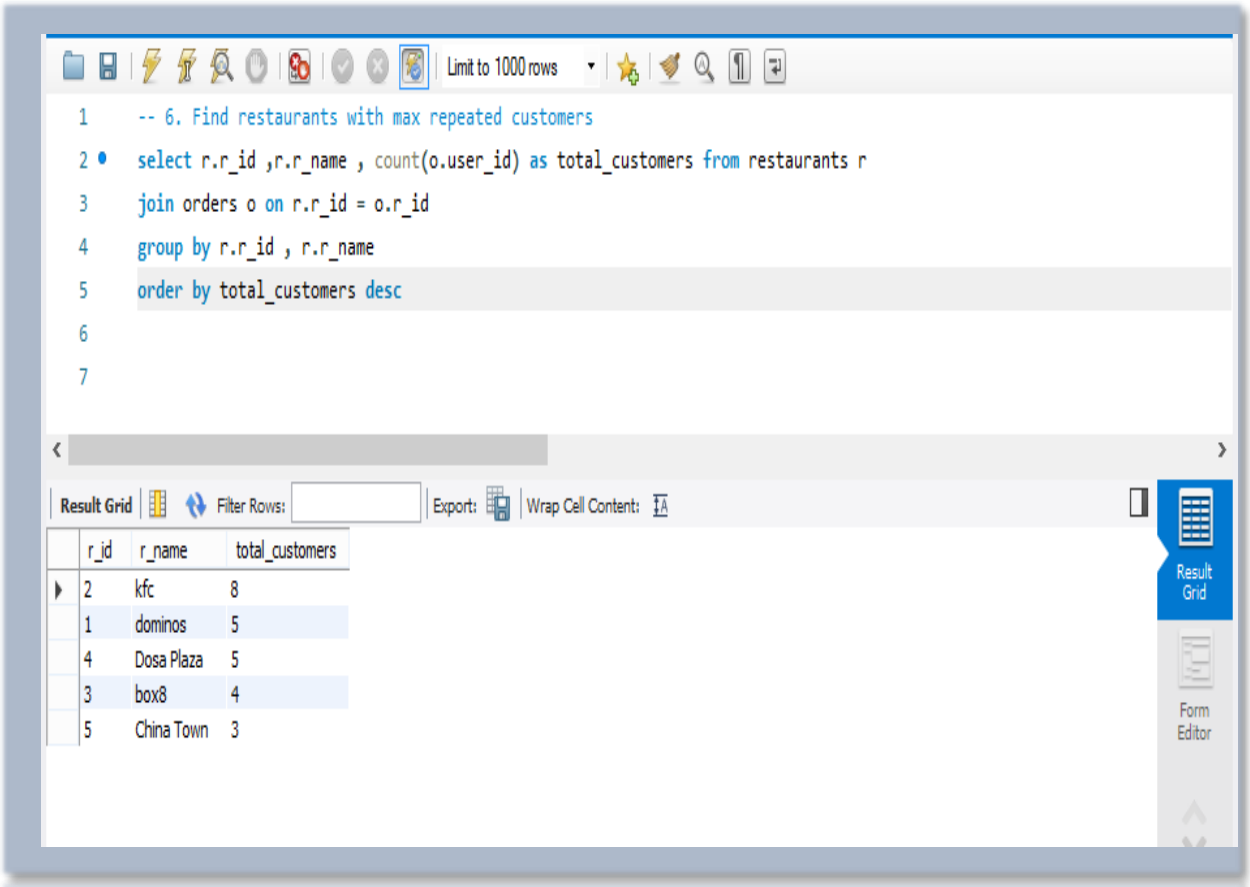
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

user_id	name	r_name	date	f_name
1	Sagar	dominos	2022-05-10	Choco Lava cake
1	Sagar	dominos	2022-05-10	Veg Pizza
1	Sagar	dominos	2022-05-10	Non-veg Pizza
1	Sagar	kfc	2022-05-26	Chicken Popcorn
1	Sagar	kfc	2022-05-26	Chicken Wings
1	Sagar	kfc	2022-05-26	Choco Lava cake
1	Sagar	box8	2022-06-15	Roti meal
1	Sagar	box8	2022-06-15	Rice Meal
1	Sagar	box8	2022-06-15	Choco Lava cake
1	Sagar	box8	2022-06-29	Roti meal
1	Sagar	box8	2022-06-29	Rice Meal
1	Sagar	box8	2022-06-29	Choco Lava cake
1	Sagar	box8	2022-07-10	Roti meal
1	Sagar	box8	2022-07-10	Rice Meal
1	Sagar	box8	2022-07-10	Choco Lava cake
2	Anjali	dominos	2022-06-10	Choco Lava cake
2	Anjali	dominos	2022-06-10	Veg Pizza
2	Anjali	dominos	2022-06-10	Non-veg Pizza
2	Anjali	kfc	2022-06-23	Chicken Popcorn

6. Find restaurants with max repeated customers.

```
SELECT r.r_id, r.r_name, count(o.user_id) AS total_customers
from restaurants r
JOIN orders o ON r.r_id = o.r_id
GROUP BY r.r_id, r.r_name
ORDER BY total_customers DESC;
```

OUTPUT -

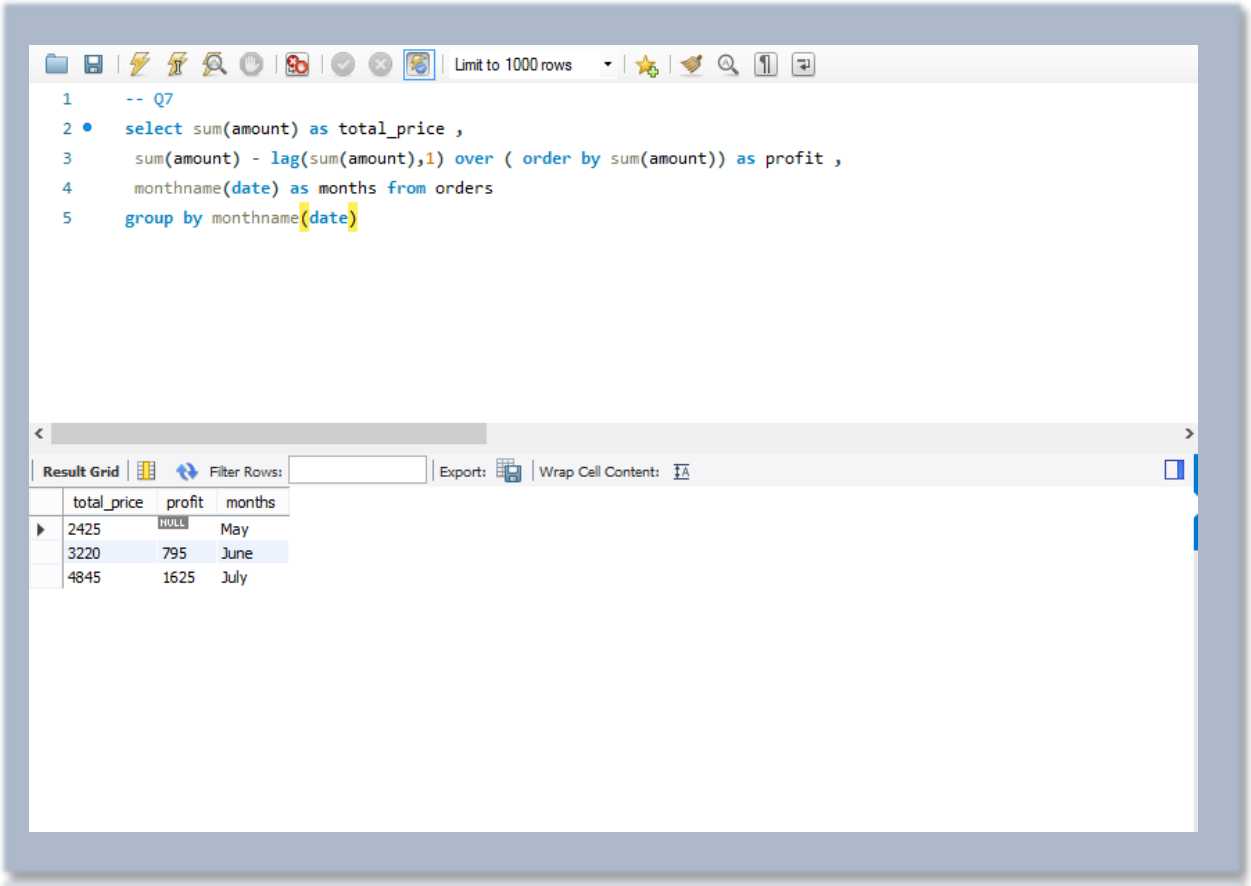


r_id	r_name	total_customers
2	kfc	8
1	dominos	5
4	Dosa Plaza	5
3	box8	4
5	China Town	3

7. Month over month revenue growth of ZOMATO.

```
SELECT SUM(amount) AS total_price,
SUM(amount) - LAG(sum(amount),1) OVER (order by sum(amount))
AS profit,
MONTHNAME(date) AS months FROM orders
GROUP BY MONTHNAME(date);
```

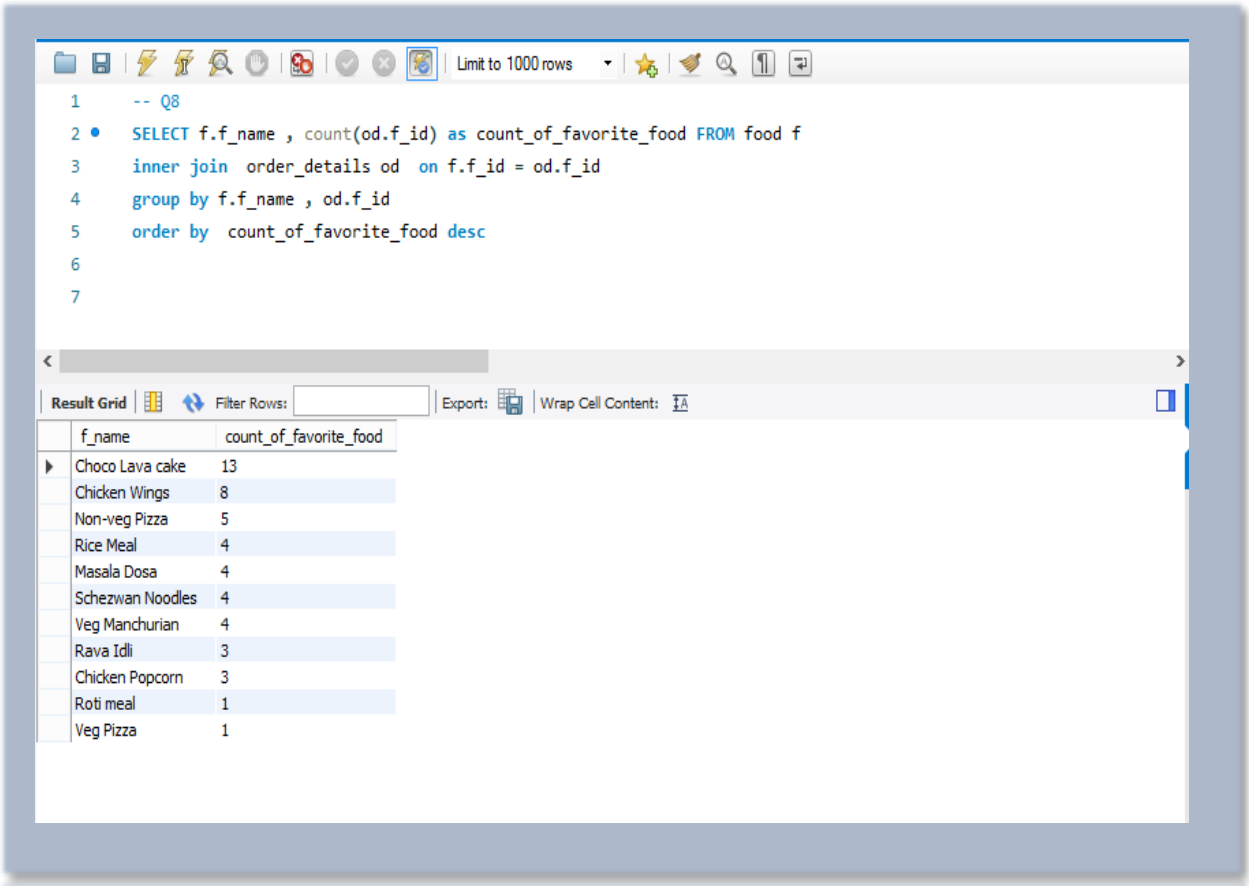
OUTPUT -



8. Customer's most favourite food.

```
SELECT f.f_name, count(od.f_id) AS count_of_favorite_food
FROM food f
JOIN order_details od ON f.f_id = od.f_id
GROUP BY f.f_name, od.f_id
ORDER BY count_of_favorite_food DESC;
```

OUTPUT -



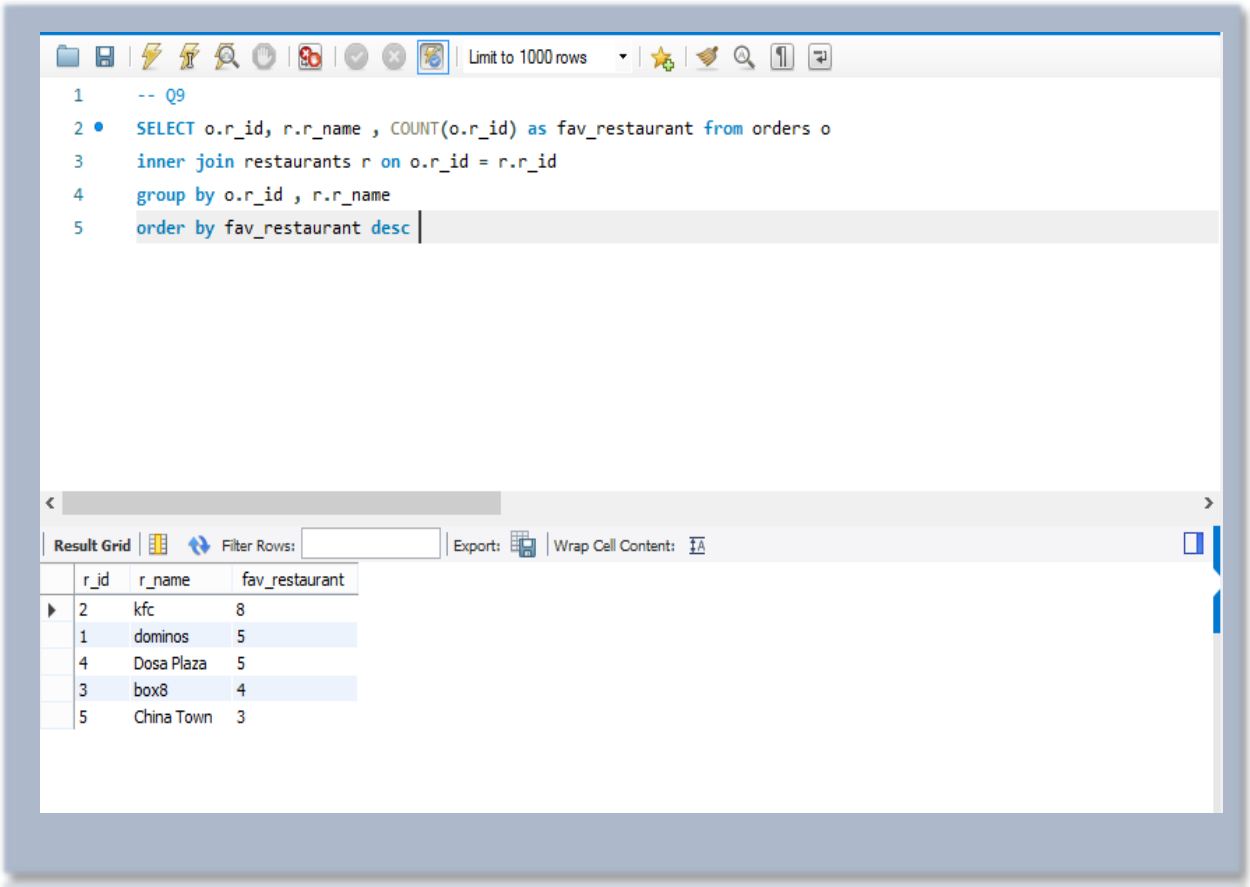
```
-- Q8
SELECT f.f_name , count(od.f_id) as count_of_favorite_food FROM food f
inner join  order_details od  on f.f_id = od.f_id
group by f.f_name , od.f_id
order by  count_of_favorite_food desc
```

f_name	count_of_favorite_food
Choco Lava cake	13
Chicken Wings	8
Non-veg Pizza	5
Rice Meal	4
Masala Dosa	4
Schezwan Noodles	4
Veg Manchurian	4
Rava Idli	3
Chicken Popcorn	3
Roti meal	1
Veg Pizza	1

9. Customer’s most favourite restaurant by orders.

```
SELECT o.r_id, r.r_name, COUNT(o.r_id) AS fav_restaurant FROM
orders o
JOIN restaurants r on o.r_id = r.r_id
GROUP BY o.r_id, r.r_name
ORDER BY fav_restaurant DESC;
```

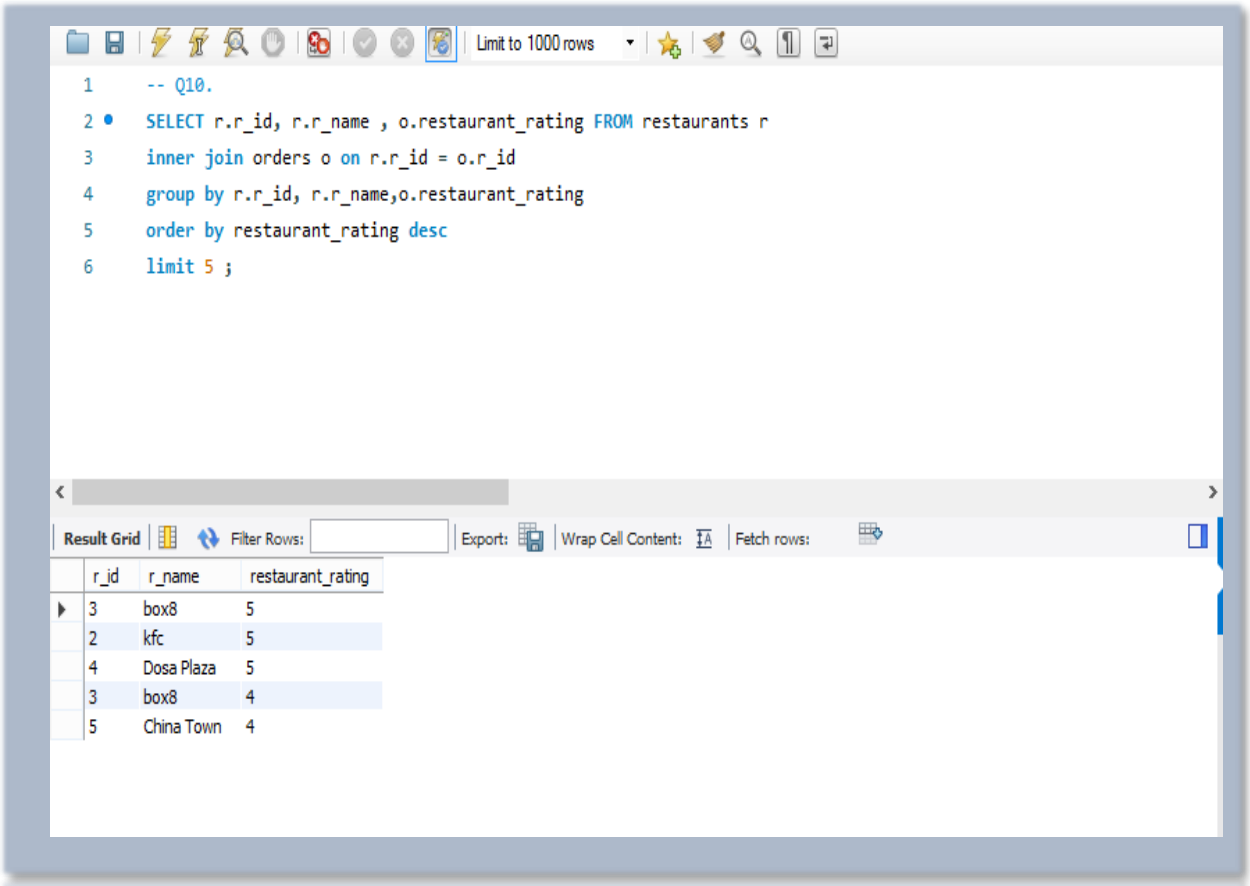
OUTPUT -



10. Top 5 restaurants by rating.

```
SELECT o.r_id, r.r_name, COUNT(o.r_id) AS fav_restaurant FROM
orders o
JOIN restaurants r on o.r_id = r.r_id
GROUP BY o.r_id, r.r_name
ORDER BY fav_restaurant DESC;
```

OUTPUT -



r_id	r_name	restaurant_rating
3	box8	5
2	kfc	5
4	Dosa Plaza	5
3	box8	4
5	China Town	4