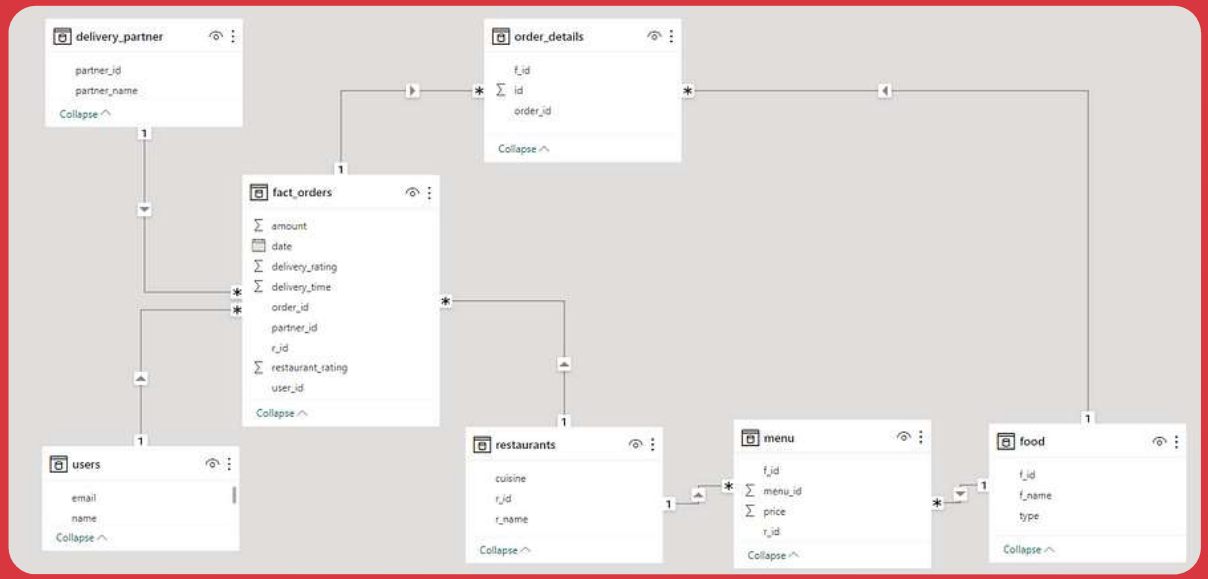
DATA MODEL



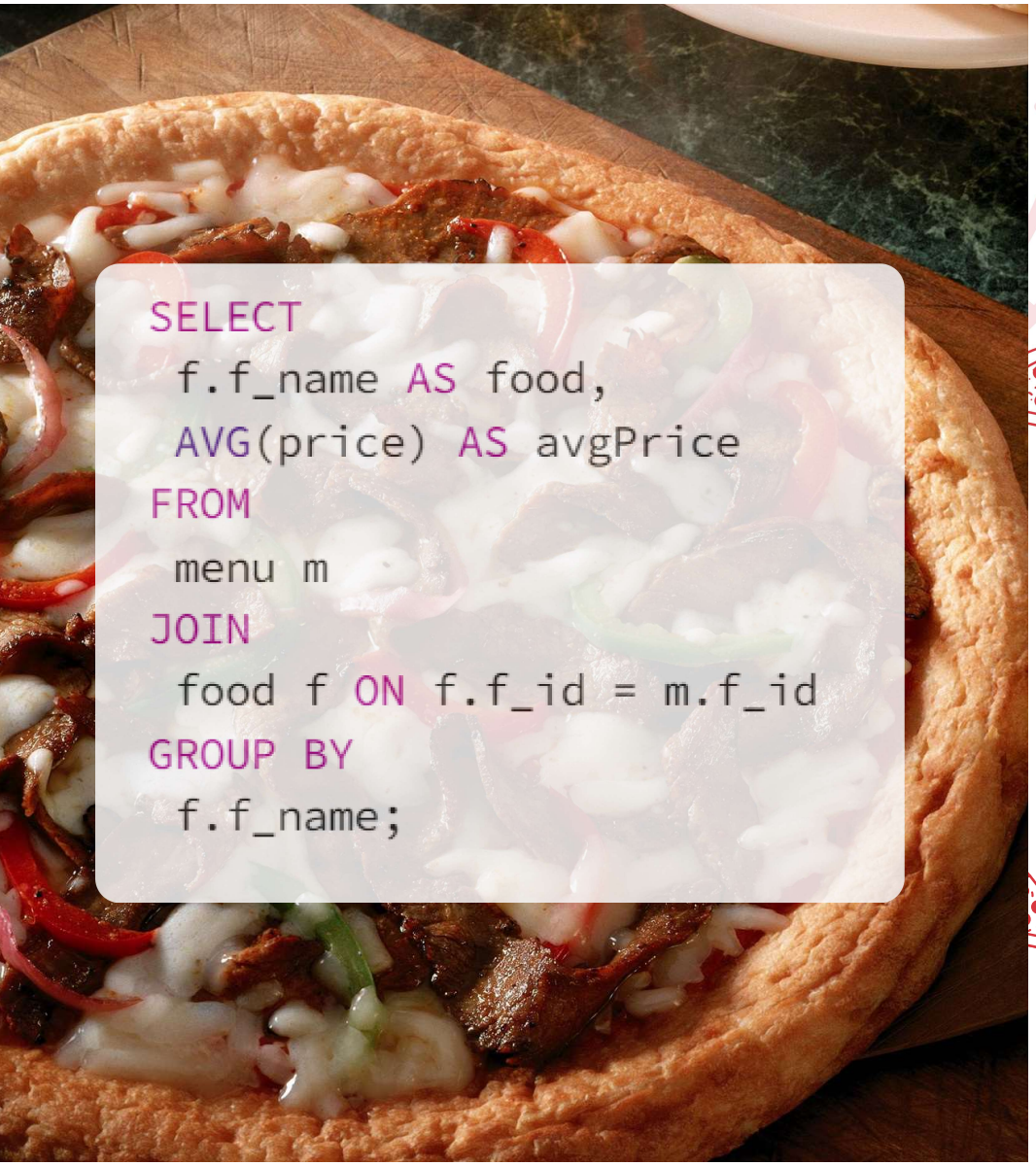
- **Snowflake schema** designed using Power BI.
- 'restaurants' table normalized to 'menu' and further to 'food'.
- **Normalization** helps to deal with redundant and repetitive data.

1.FIND CUSTOMERS WHO HAVE NEVER ORDERED.

Result Grid

name
Anupama
Rishabh

```
SELECT
 name
FROM
 users
WHERE
 user_id NOT IN
(
SELECT user_id
FROM orders
);
```



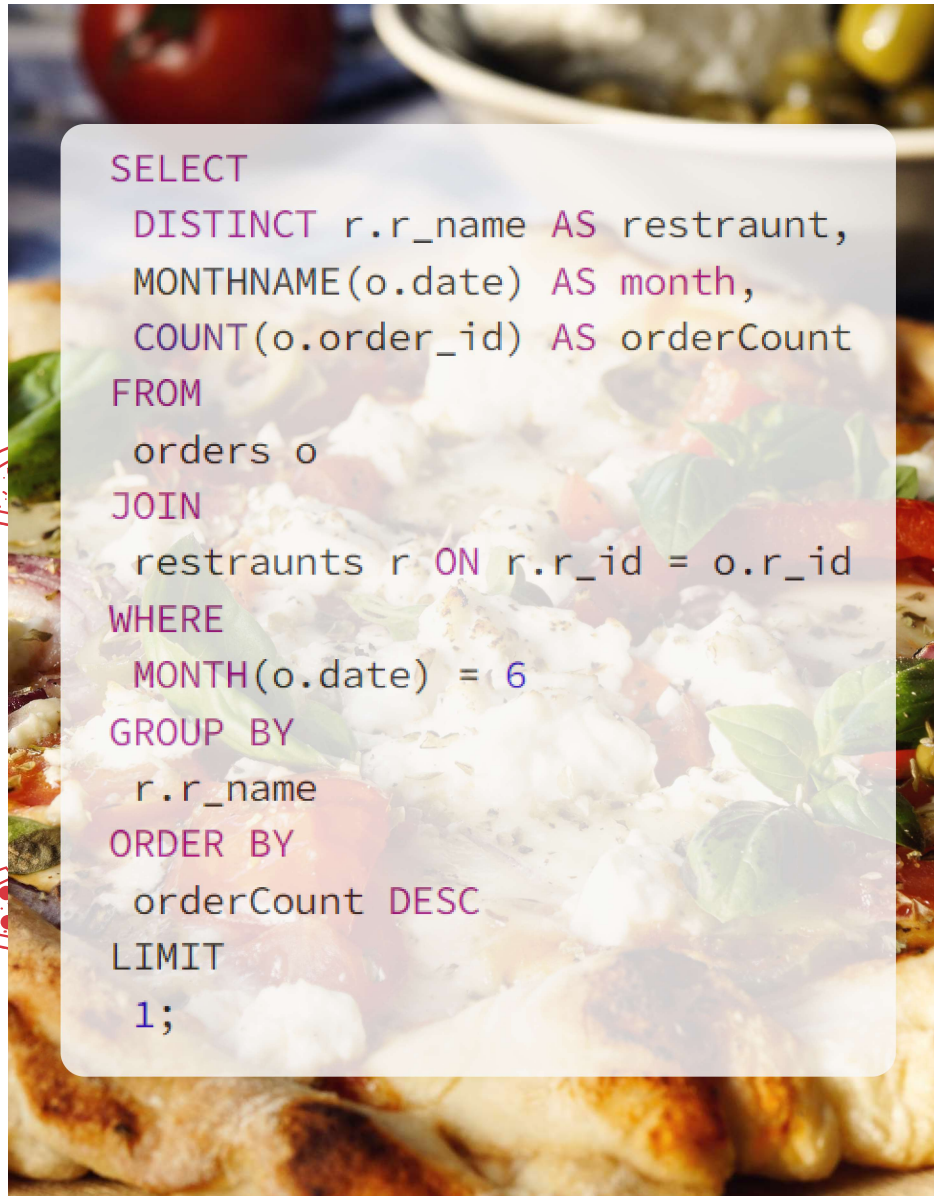
```sql
SELECT
 f.f_name AS food,
 AVG(price) AS avgPrice
FROM
 menu m
JOIN
 food f ON f.f_id = m.f_id
GROUP BY
 f.f_name;
```

2. AVERAGE PRICE/DISH

food	avgPrice
Non-veg Pizza	450.0000
Veg Pizza	400.0000
Choco Lava cake	98.3333
Chicken Wings	230.0000
Chicken Popcorn	300.0000
Rice Meal	213.3333
Roti meal	140.0000
Masala Dosa	180.0000
Rava Idli	120.0000
Schezwan Noodles	220.0000
Veg Manchurian	180.0000

3. FIND THE TOP RESTAURANT IN TERMS OF THE NUMBER OF ORDERS IN JUNE.

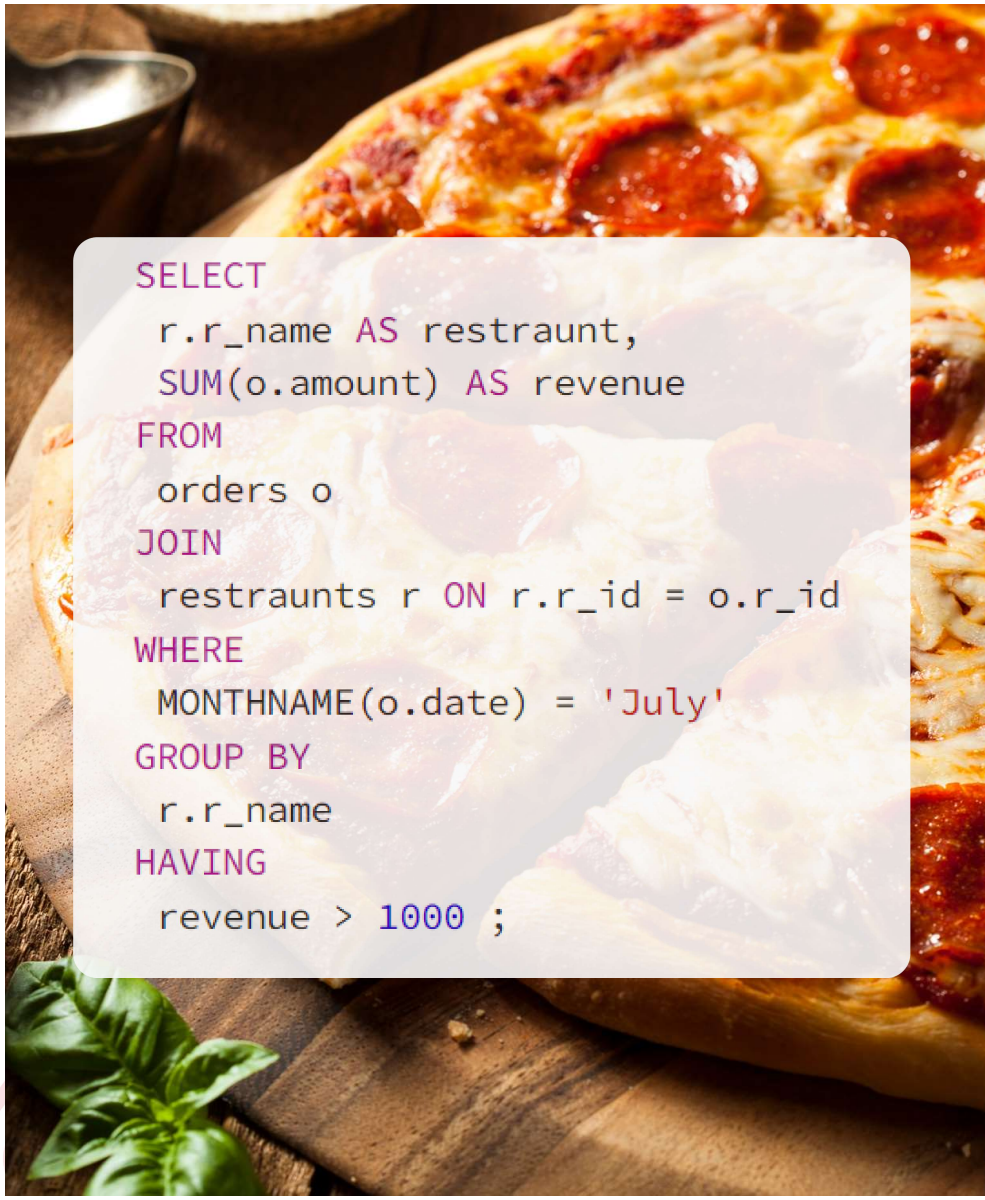
	restraunt	month	orderCount
▶	kfc	June	3



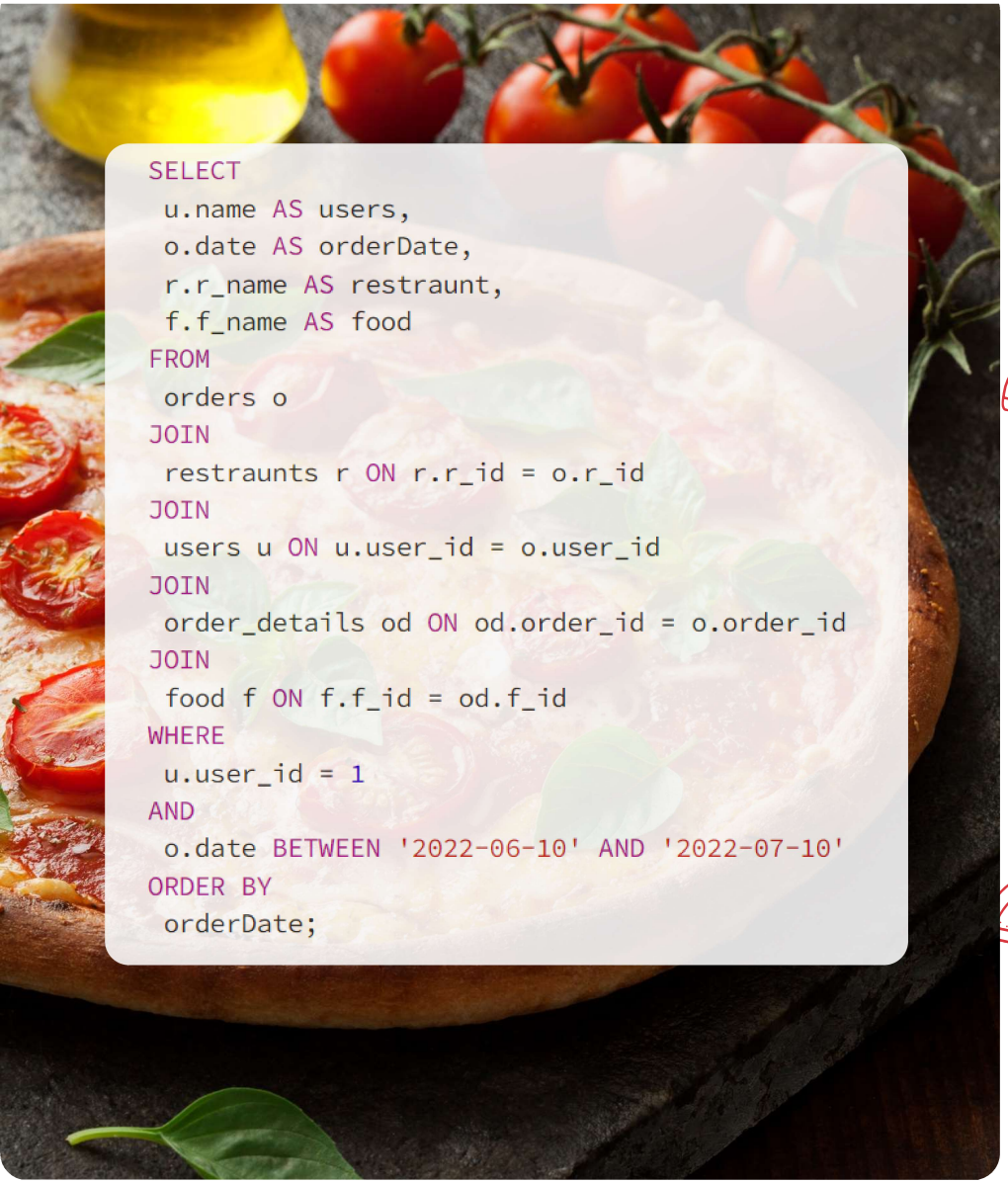
```sql
SELECT
 DISTINCT r.r_name AS restraunt,
 MONTHNAME(o.date) AS month,
 COUNT(o.order_id) AS orderCount
FROM
 orders o
JOIN
 restraunts r ON r.r_id = o.r_id
WHERE
 MONTH(o.date) = 6
GROUP BY
 r.r_name
ORDER BY
 orderCount DESC
LIMIT
 1;
```

4. RESTAURANTS WITH MONTHLY SALES GREATER THAN 1000 FOR JULY.

	restraunt	revenue
▶	China Town	1050
	dominos	1100
	kfc	1935



```sql
SELECT
 r.r_name AS restraunt,
 SUM(o.amount) AS revenue
FROM
 orders o
JOIN
 restraunts r ON r.r_id = o.r_id
WHERE
 MONTHNAME(o.date) = 'July'
GROUP BY
 r.r_name
HAVING
 revenue > 1000 ;
```



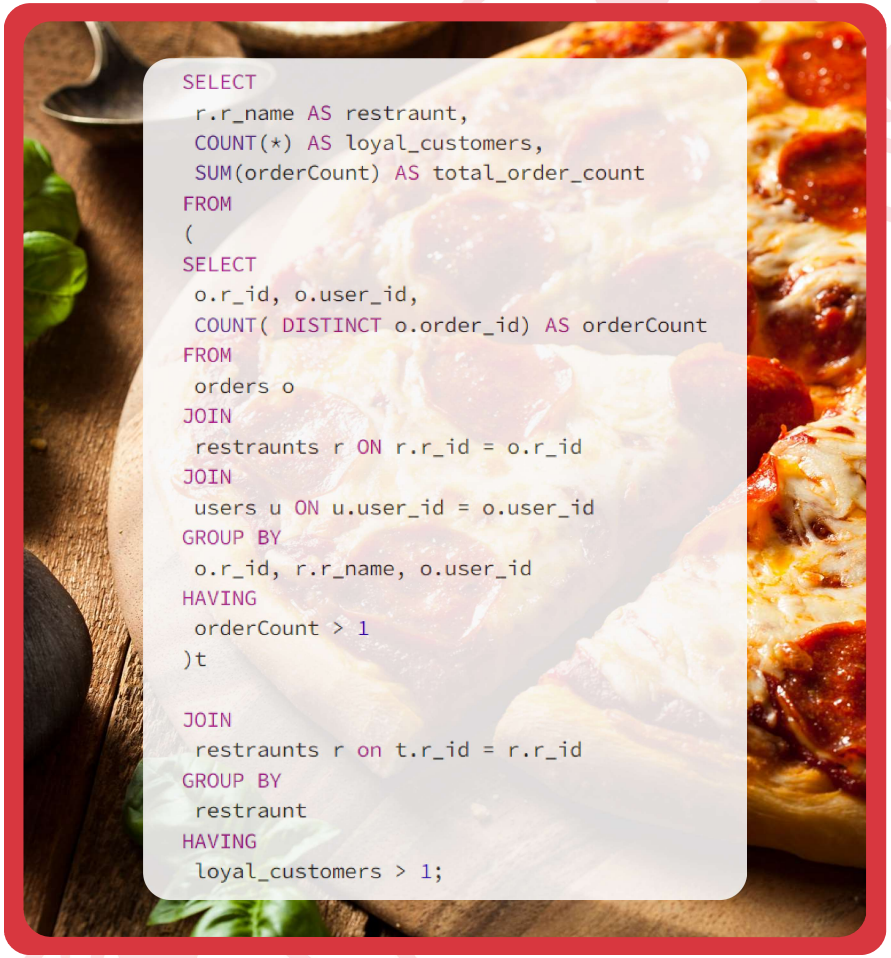
```sql
SELECT
 u.name AS users,
 o.date AS orderDate,
 r.r_name AS restraunt,
 f.f_name AS food
FROM
 orders o
JOIN
 restraunts r ON r.r_id = o.r_id
JOIN
 users u ON u.user_id = o.user_id
JOIN
 order_details od ON od.order_id = o.order_id
JOIN
 food f ON f.f_id = od.f_id
WHERE
 u.user_id = 1
AND
 o.date BETWEEN '2022-06-10' AND '2022-07-10'
ORDER BY
 orderDate;
```

5. SHOW ALL ORDERS WITH ORDER DETAILS OF NITISH (USER_ID = 1) FROM 10TH JUNE'22 TO 10TH JULY'22.

orderDate	users	restraunt	food
2022-06-15	Nitish	box8	Choco Lava cake
2022-06-15	Nitish	box8	Rice Meal
2022-06-29	Nitish	box8	Choco Lava cake
2022-06-29	Nitish	box8	Rice Meal
2022-07-10	Nitish	box8	Choco Lava cake
2022-07-10	Nitish	box8	Roti meal

6. FIND RESTAURANTS WITH MAXIMUM REPEAT CUSTOMERS.

	restraunt	loyal_customers	total_order_count
▶	kfc	2	6



```sql
SELECT
 r.r_name AS restraunt,
 COUNT(*) AS loyal_customers,
 SUM(orderCount) AS total_order_count
FROM
(
SELECT
 o.r_id, o.user_id,
 COUNT( DISTINCT o.order_id) AS orderCount
FROM
 orders o
JOIN
 restraunts r ON r.r_id = o.r_id
JOIN
 users u ON u.user_id = o.user_id
GROUP BY
 o.r_id, r.r_name, o.user_id
HAVING
 orderCount > 1
)t

JOIN
 restraunts r on t.r_id = r.r_id
GROUP BY
 restraunt
HAVING
 loyal_customers > 1;
```

```sql
WITH T AS
(
SELECT
 MONTHNAME (date) AS month,
 SUM(amount) AS revenue,
 LAG (SUM(amount)) OVER (ORDER BY date) AS prevRevenue
FROM
 orders
GROUP BY
 month
ORDER BY
 date
)
SELECT
 month,
 revenue,
 ((revenue-prevRevenue)/ prevRevenue) * 100 AS 'MOM revenue growth (%)'
FROM
T;
```

7. MONTH-OVER-MONTH REVENUE GROWTH OF ZOMATO.

	month	revenue	MoM revenue growth(%)
▶	May	2425	NULL
	June	3220	32.7835
	July	4845	50.4658

8. CUSTOMER AND THIER FAVORITE FOOD.

	Name	FavoriteFood	OrderCount
▶	Ankit	Schezwan Noodles	3
	Ankit	Veg Manchurian	3
	Khushboo	Choco Lava cake	3
	Neha	Choco Lava cake	5
	Nitish	Choco Lava cake	5
	Vartika	Chicken Wings	3

```sql
WITH T AS
(
SELECT
 o.user_id, u.name, od.order_id,
 od.f_id, f.f_name as favouriteFood,
 COUNT(od.f_id) AS orderCount,
 RANK() OVER (PARTITION BY u.name ORDER BY COUNT(od.f_id) DESC) AS orderRank
FROM
 orders o
JOIN
 order_details od ON od.order_id = o.order_id
JOIN
 food f ON f.f_id = od.f_id
JOIN
 users u ON u.user_id = o.user_id
GROUP BY
 u.user_id,
    f.f_name
ORDER BY
 u.name,
    orderCount DESC
)
SELECT
 name,
 favouriteFood,
    orderCount
FROM
 T
WHERE
 orderRank = 1;
```

```sql
WITH X AS
(
SELECT
 u.name, r.r_name,
 COUNT(o.r_id) AS OrderCount,
 DENSE_RANK() OVER (PARTITION BY r.r_name ORDER BY COUNT(o.r_id) DESC) AS row_rank
FROM
 orders o
JOIN
 restraunts r ON o.r_id = r.r_id
JOIN
 users u ON o.user_id = u.user_id
GROUP BY
  o.user_id,
  o.r_id
)
SELECT
 *
FROM
 X
WHERE
 row_rank = 1;
```

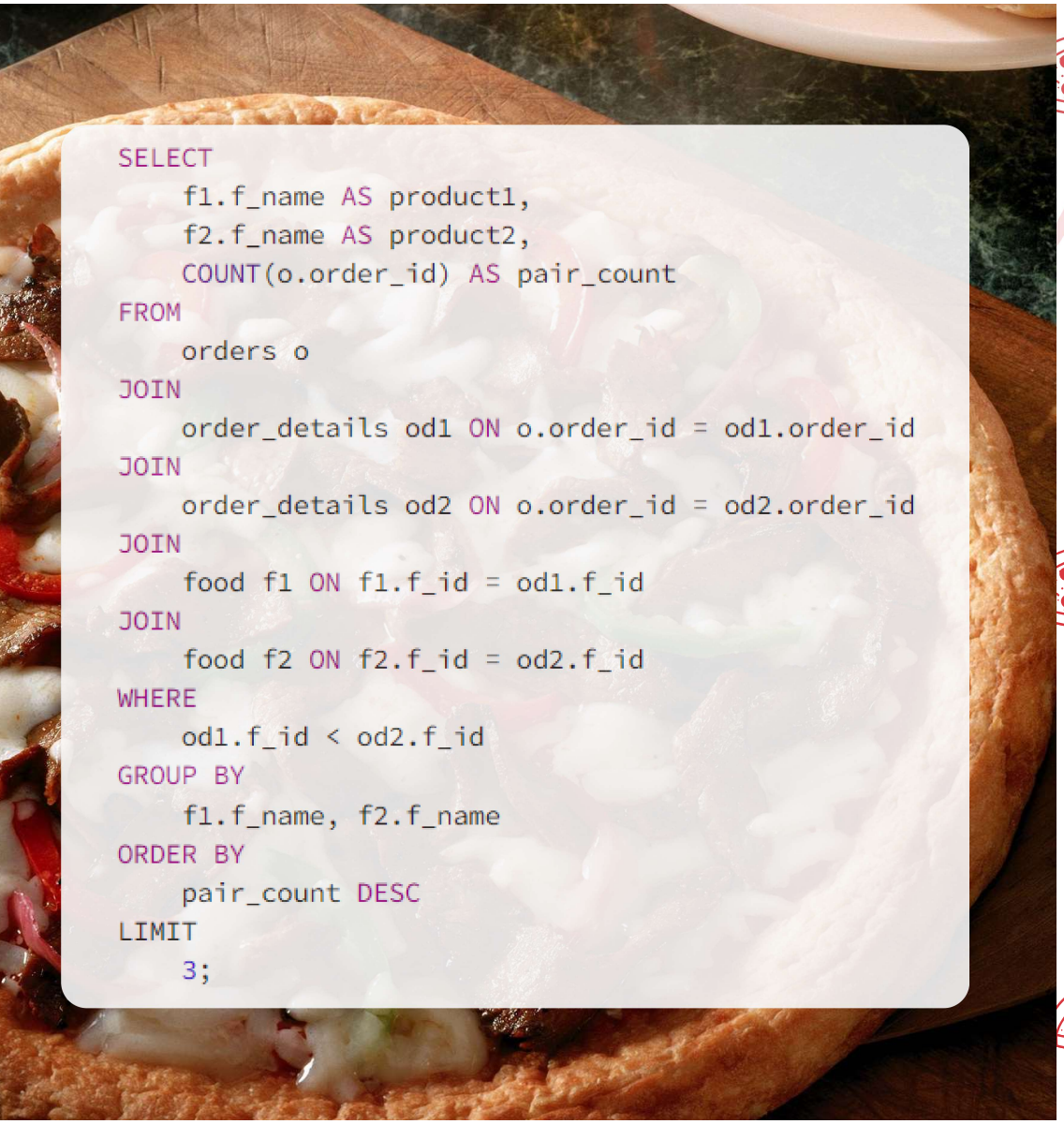
9. FIND THE MOST LOYAL CUSTOMERS FOR ALL RESTAURANTS.

	name	restraunt	order_count	row_rank
▶	Nitish	box8	3	1
	Ankit	China Town	2	1
	Neha	dominos	2	1
	Ankit	Dosa Plaza	3	1
	Vartika	kfc	3	1
	Neha	kfc	3	1

10. MONTH-OVER-MONTH REVENUE GROWTH OF EACH RESTAURANT.

	restraunt	monthName	MoM revenue growth(%)
▶	dominos	May	NULL
	dominos	June	-5.0000
	dominos	July	15.7895
	kfc	May	NULL
	kfc	June	53.4884
	kfc	July	95.4545
	box8	June	NULL
	box8	July	-4.1667
	Dosa Plaza	May	NULL
	Dosa Plaza	June	-48.7179
	Dosa Plaza	July	-25.0000
	China Town	June	NULL
	China Town	July	162.5000

```sql
WITH X AS
(
SELECT
 o.r_id, r.r_name As restraunt,
 MONTHNAME(o.date) AS monthName,
 SUM(o.amount) AS revenue,
 LAG(SUM(o.amount)) OVER (PARTITION BY r.r_name ORDER BY MONTH(o.date)) AS prevRev
FROM
 orders o
JOIN
 restraunts r ON o.r_id = r.r_id
GROUP BY
 r.r_name,
 MONTHNAME(o.date)
)
SELECT
 restraunt,
 monthName,
 ((revenue - prevRevenue)/ NULLIF (prevRevenue, 0))*100 AS 'MoM revenue growth (%)'
FROM
 X
ORDER BY
 r_id;
```



```sql
SELECT
    f1.f_name AS product1,
    f2.f_name AS product2,
    COUNT(o.order_id) AS pair_count
FROM
    orders o
JOIN
    order_details od1 ON o.order_id = od1.order_id
JOIN
    order_details od2 ON o.order_id = od2.order_id
JOIN
    food f1 ON f1.f_id = od1.f_id
JOIN
    food f2 ON f2.f_id = od2.f_id
WHERE
    od1.f_id < od2.f_id
GROUP BY
    f1.f_name, f2.f_name
ORDER BY
    pair_count DESC
LIMIT
    3;
```

11. TOP 3 PAIR PRODUCTS ORDERED TOGETHER.

	product1	product2	pair_count
▶	Choco Lava cake	Chicken Wings	5
	Non-veg Pizza	Choco Lava cake	4
	Schezwan Noodles	Veg Manchurian	4

IMPORTANT INSIGHT !!

CHICKEN WINGS AND CHOCO LAVA CAKE ARE THE MOST ORDERED FOOD ITEMS TOGETHER.
AND I VOUCH FOR THAT ♥♥

X