AMAN MAHESHWARI
Data Analyst
amansandesh33@gmail.com

PIZZA HUT

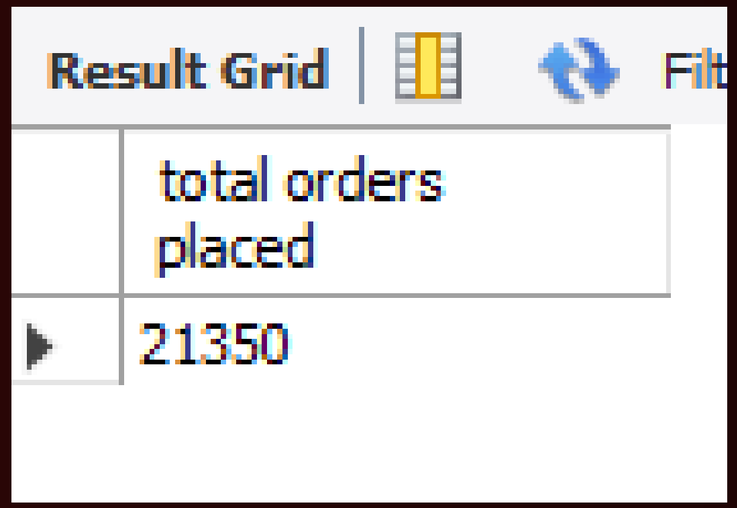
- SQL PROJECT

AMAN MAHESHWARI
Data Analyst

ABOUT PROJECT

Utilized SQL to perform comprehensive data analysis on Pizza Hut's operational database, extracting actionable insights through complex queries involving joins, aggregations, and subqueries. Developed analytical solutions for order trends, customer behavior patterns, and inventory optimization, transforming raw transactional data into strategic business intelligence to support data driven decision making and operational efficiency improvements.

```sql
-- The total number of orders placed.

select count(order_id) as 'total orders placed' from orders;
```



```sql
-- the total revenue generated from pizza sales.

SELECT
    ROUND(SUM(order_detail.quantity * pizzas.price),
            2) AS 'Total Revenue'
FROM
    order_detail
        JOIN
    pizzas ON pizzas.pizza_id = order_detail.pizza_id;
```

	Total Revenue
▶	817860.05

```sql
-- the highest-priced pizza.

SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result Grid | Filter Row

name	price
The Greek Pizza	35.95

```sql
-- the most common pizza size ordered.

SELECT
    pizzas.size,
    COUNT(order_detail.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    order_detail ON pizzas.pizza_id = order_detail.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

Result Grid | Filter R

	size	order_count
▶	L	18526

```sql
-- top 5 most ordered pizza types along with their quantities.

SELECT
    pizza_types.name, SUM(order_detail.quantity)
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_detail ON order_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY SUM(order_detail.quantity) DESC
LIMIT 5;
```

Result Grid | Filter Rows: | Export

name	sum(order_detail.quantity)
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

```sql
-- the total quantity of each pizza category ordered.

SELECT
    pizza_types.category, SUM(order_detail.quantity)
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_detail ON order_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY SUM(order_detail.quantity) DESC;
```

Result Grid | Filter Rows:

category	sum(order_detail.quantity)
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

```sql
-- the distribution of orders by hour of the day.

SELECT
    HOUR(oder_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(oder_time);
```

Result Grid | Filter Rows:

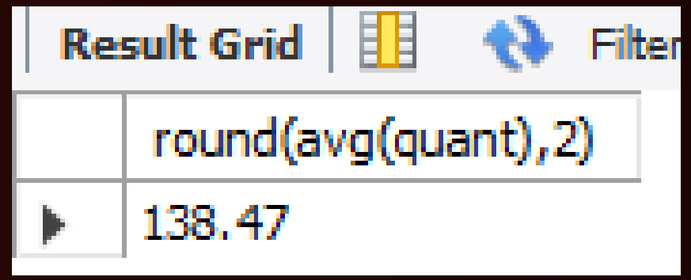
HOUR(oder_time)	COUNT(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

```sql
-- category-wise distribution of pizzas.

SELECT
    COUNT(name), category
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Rows:

count(name)	category
6	Chicken
8	Classic
9	Supreme
9	Veggie

```sql
-- Group the orders by date and calculate the average number of pizzas ordered per day.


SELECT
    ROUND(AVG(quant), 2)
FROM
    (SELECT
        DATE(orders.order_date), SUM(order_detail.quantity) AS quant
    FROM
        orders
    JOIN order_detail ON orders.order_id = order_detail.order_id
    GROUP BY DATE(orders.order_date)) AS order_quant;
```



```sql
-- top 3 most ordered pizza types based on revenue.

SELECT
    pizza_types.name,
    ROUND(SUM(order_detail.quantity * pizzas.price),
            2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_detail ON order_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5

```sql
-- the percentage contribution of each pizza type to total revenue.

SELECT
    pizza_types.category,
    round((SUM(order_detail.quantity * pizzas.price) / (SELECT
            ROUND(SUM(order_detail.quantity * pizzas.price),
                        2) AS 'Total Revenue'
        FROM
            order_detail
                JOIN
            pizzas ON pizzas.pizza_id = order_detail.pizza_id)) * 100,2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_detail ON order_detail.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
```

Result Grid | Filte

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

```sql
-- the cumulative revenue generated over time.

select order_date, sum(revenue) over(order by order_date) as cum_revenue from
(select orders.order_date,
sum(order_detail.quantity * pizzas.price) as revenue
from order_detail join pizzas
on order_detail.pizza_id=pizzas.pizza_id
join orders
on orders.order_id=order_detail.order_id
group by orders.order_date) as sales;
```

Result Grid | Filter Rows:

order_date	cum_revenue
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003
2015-01-14	32358.700000000004

```sql
-- the top 3 most ordered pizza types based on revenue for each pizza category.


select category, name , revenue from
(select category, name, revenue, rank() over(partition by category order by revenue desc) as rn
from
(select pizza_types.category ,pizza_types.name,
sum((order_detail.quantity)*pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_detail
on order_detail.pizza_id=pizzas.pizza_id
group by pizza_types.category ,pizza_types.name) as work) as final
where rn<=3;
```

Result Grid | Filter Rows: | Export:

category	name	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5