PIZZA HUT

HOME ABOUT MORE

Pizza Sales Project using SQL

Hello!

My Name is Vipul Agarwal
And in this Project I have utilized SQL queries to solve business-related questions based on pizza sales data. The goal was to turn raw data into meaningful insights to support better decision-making.

OBJECTIVE

- The main objective was to explore sales trends, identify best-selling pizzas, and evaluate revenue performance.
- We calculated the total number of orders and total revenue generated. The most popular pizza size was identified, along with the top 5 pizzas based on quantity ordered
- Further analysis showed which pizza categories sold the most, how orders were distributed by time of day, and which pizzas brought in the most revenue.

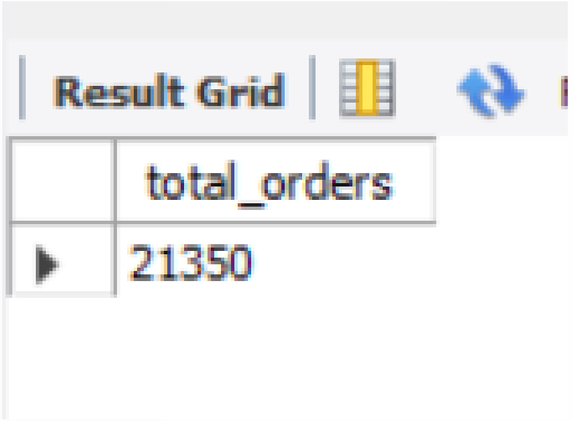
Key insights include:

- Medium size pizzas are the most ordered.
- A few pizza types dominate in sales and revenue.
- Peak sales occur around lunch and dinner hours.
- Revenue steadily increased over time.

Retrieve the total number of orders placed.

```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

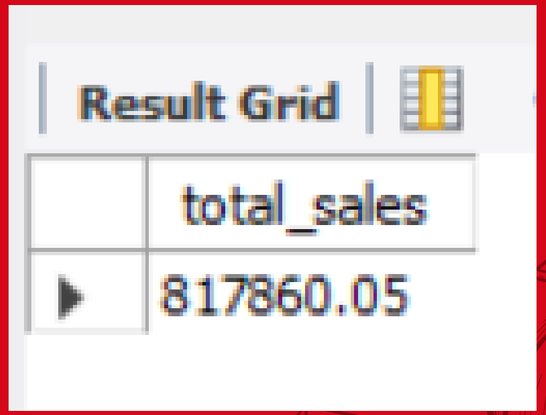
Output :-



Calculate the total revenue generated from pizza sales.

```sql
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

Output :-



Identify the most common pizza size ordered.

```sql
SELECT
    pizzas.size,
    COUNT(order_details.order_id) AS Most_ordered_id
FROM
    pizzas
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizzas.size
ORDER BY Most_ordered_id DESC
LIMIT 1;
```

Output :-

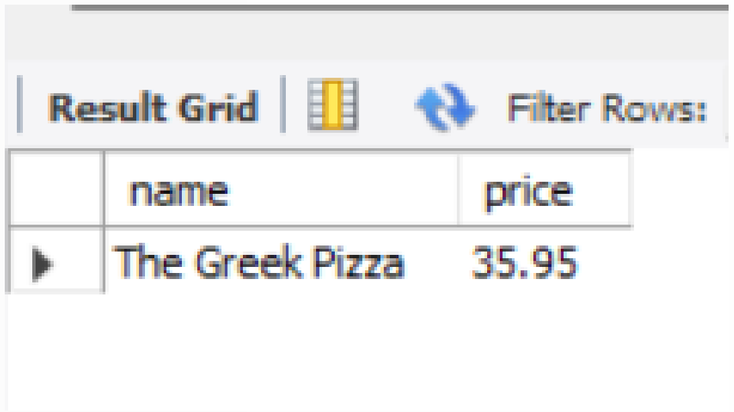
Result Grid | Filter Rows:

	size	Most_ordered_id
▶	L	18526

Identify the highest-priced pizza.

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

Output :-



List the top 5 most ordered pizza types along with their quantities.

```sql
select
pizza_types.name as Name, sum(order_details.quantity) as Quantity_Ord
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on  order_details.pizza_id = pizzas.pizza_id
group by Name
order by Quantity_Ord desc limit 5;
```

Output :-

Result Grid | Filter Rows:

Name	Quantity_Ord
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
select
sum(order_details.quantity) as Total_Quantity, pizza_types.category
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join pizza_types
on pizza_types.pizza_type_id = pizzas.pizza_type_id
group by pizza_types.category
order by Total_Quantity desc ;
```

Output :-

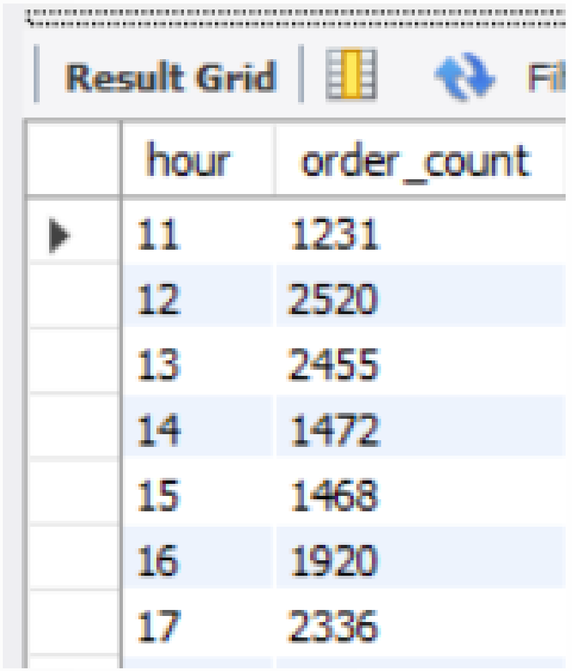
Result Grid | Filter Rows:

Total_Quantity	category
14888	Classic
11987	Supreme
11649	Veggie
11050	Chicken

Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

Output :-



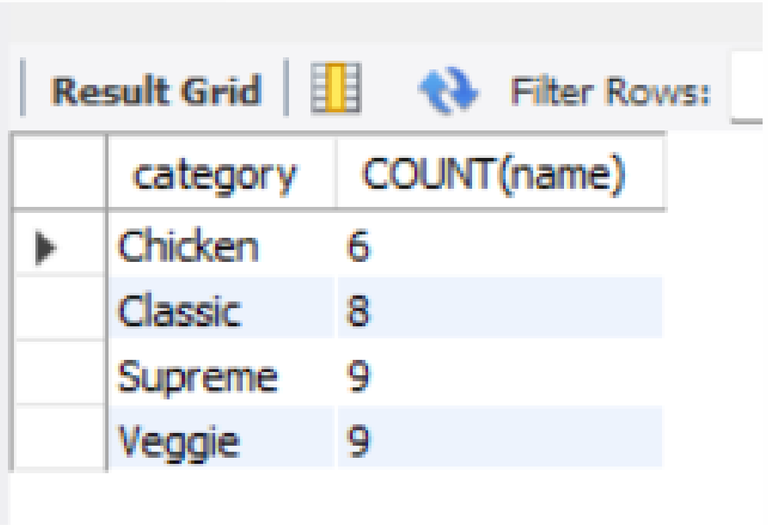
Result Grid

hour	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336

Join relevant tables to find the category-wise distribution of pizzas.

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

Output :-



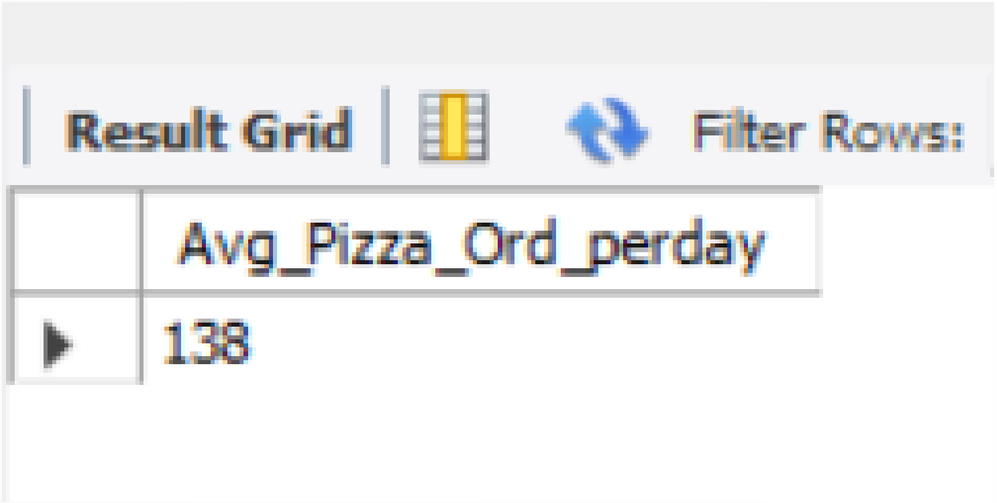
Result Grid | Filter Rows:

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

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

```sql
select round(avg(quantity),0) as Avg_Pizza_Ord_perday
from
(select orders.order_date as Order_date, sum(order_details.quantity) as quantity
from orders join order_details
on orders.order_id = order_details.order_id
group by Order_date) as Order_quantity;
```

Output :-



Determine the top 3 most ordered pizza types based on revenue.

```sql
select pizza_types.name as Name,
Round(sum((order_details.quantity * pizzas.price)),0) as Revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by Name
order by Revenue desc limit 3;
```

Output :-

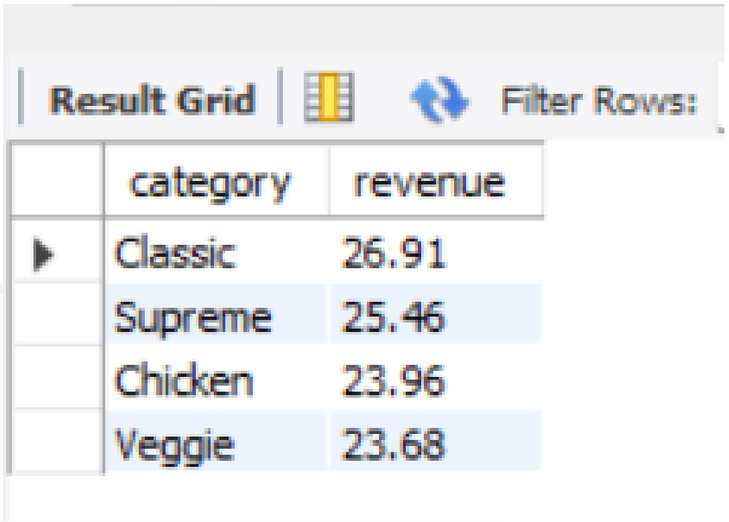
Result Grid | Filter Rows:

Name	Revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

Calculate the percentage contribution of each pizza type to total revenue.

```sql
select pizza_types.category,
round(sum(order_details.quantity * pizzas.price) / (SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,2) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category
order by revenue desc;
```

Output :-



category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

PIZZA STORE

HOME ABOUT MORE