Pizza Sales Analysis Using MySQL Project

By Mangesh Pakhale

About Us

Hello, my name is Mangesh Pakhale and I am passionate about working with data. In this project, I utilized SQL queries to analyze pizza sales.

01

Retrieve the total number of orders placed.

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

Result Grid

	total_orders
▶	21350

02 Calculate the total revenue generated from pizza sales.

```sql
SELECT
    ROUND(SUM(quantity * price), 2) revenue
FROM
    order_details o
        JOIN
    pizzas p USING (pizza_id);
```

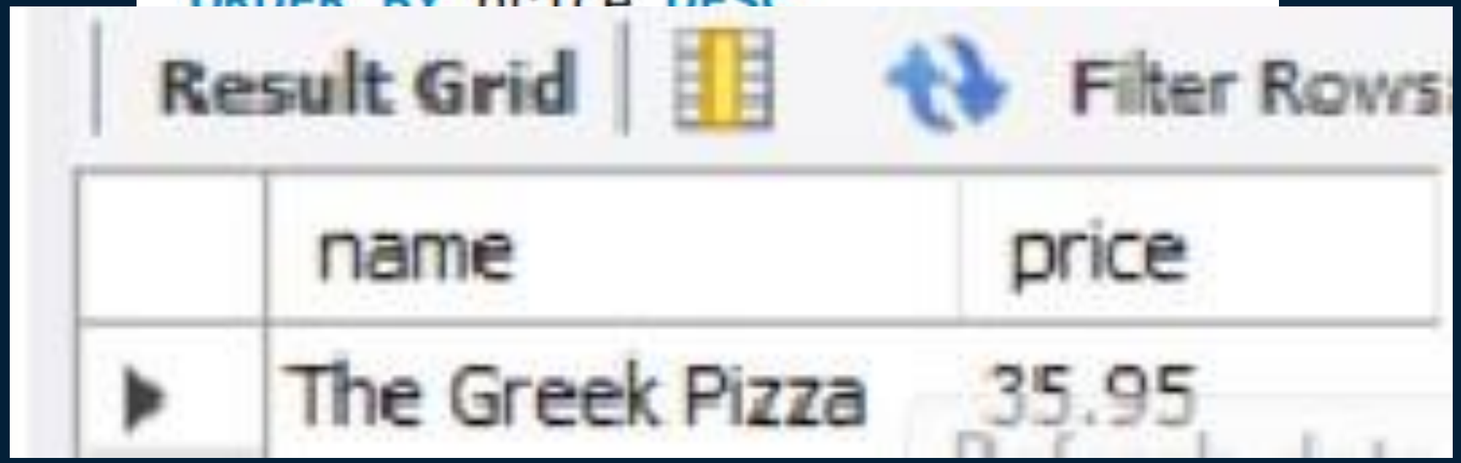
Result Grid

	revenue
▶	817860.05

03

Identify the highest-priced pizza.

```
SELECT
    name, price
FROM
    pizza_types pt
        JOIN
    pizzas p USING (pizza_type_id)
ORDER BY price DESC
```



04 Identify the most common pizza size ordered.

```
SELECT
    size, COUNT(quantity)
FROM
    pizzas
        JOIN
    order_details USING (pizza_id)
GROUP BY size
ORDER BY COUNT(quantity) DESC
```

Result Grid

	size	quantity
▶	L	18526

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

```sql
SELECT
    name, SUM(quantity) quantities
FROM
    pizza_types
        JOIN
    pizzas USING (pizza_type_id)
        JOIN
    order_details USING (pizza_id)
GROUP BY name
ORDER BY quantities DESC
LIMIT 5;
```

name	quantities
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

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

```sql
SELECT
    category, SUM(quantity) quantity
FROM
    order_details
        JOIN
    pizzas USING (pizza_id)
        JOIN
    pizza_types USING (pizza_type_id)
GROUP BY category;
```

Result Grid | Filter Rows:

category	quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

07 Determine the distribution of orders by hour of the day.

```sql
SELECT
    HOUR(order_time), COUNT(order_id)
FROM
    orders
GROUP BY HOUR(order_time)
ORDER BY COUNT(order_id) DESC;
```

Result Grid | Filter Rows:

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

08

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

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

Result Grid | Filter Ro

category	count
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

```sql
with cte as(
select order_date,sum(quantity) q
from order_details
join orders using(order_id)
group by order_date)
select *, round(avg(q) over(),0) avg_quantity_order_per_day from cte;
```

Result Grid | Filter Rows: | Export:

order_date	quantity	avg_quantity_order_per_day
2015-01-01	162	138
2015-01-02	165	138
2015-01-03	158	138
2015-01-04	106	138
2015-01-05	125	138
2015-01-06	147	138
2015-01-07	138	138
2015-01-08	173	138

10 determine the top 3 most ordered pizza types based on revenue.

```sql
SELECT
    name, SUM(quantity * price) revenue
FROM
    order_details
        JOIN
    pizzas USING (pizza_id)
        JOIN
    pizza_types USING (pizza_type_id)
GROUP BY 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

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

```sql
with ct as(
select name, sum(quantity*price) revenue
from order_details
join pizzas using(pizza_id)
join pizza_types using(pizza_type_id)
group by name)
select *, concat(round((revenue/sum(revenue) over())*100,2),'%') 'revenue_%' from ct
order by concat(round((revenue/sum(revenue) over())*100,2),'%') desc;
```

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

name	revenue	revenue_%
The Four Cheese Pizza	32265.70000000065	3.95%
The Sicilian Pizza	30940.5	3.78%
The Pepperoni Pizza	30161.75	3.69%
The Greek Pizza	28454.100000000013	3.48%
The Mexicana Pizza	26780.75	3.27%
The Five Cheese Pizza	26066.5	3.19%
The Pepper Salami Pizza	25529	3.12%
The Italian Capocollo Pizza	25094	3.07%
The Vegetables + Vegetabl...	24374.75	2.98%
The Prosciutto and Arugula...	24193.25	2.96%
The Napolitana Pizza	24087	2.95%

30940.5

12 Calculate the percentage contribution of each pizza category to total revenue.

```sql
with ct as(
select category, sum(quantity*price) revenue
from order_details
join pizzas using(pizza_id)
join pizza_types using(pizza_type_id)
group by category)
select *, concat(round((revenue/sum(revenue) over())*100,2),'%') 'revenue_%' from ct
order by concat(round((revenue/sum(revenue) over())*100,2),'%') desc;
```

Result Grid | Filter Rows: | Export:

category	revenue	revenue_%
Classic	220053.1000000001	26.91%
Supreme	208196.99999999822	25.46%
Chicken	195919.5	23.96%
Veggie	193690.45000000298	23.68%

13 Analyze the cumulative revenue generated over time.

```
with cte as(
select order_date,sum(quantity*price) as revenue
from order_details
join pizzas using(pizza_id)
join orders using(order_id)
group by order_date)
select *, sum(revenue) over(order by order_date) cum_revenue from cte;
```

Result Grid | Filter Rows: | Export:

order_date	revenue	cum_revenue
2015-01-01	2713.8500000000004	2713.8500000000004
2015-01-02	2731.8999999999996	5445.75
2015-01-03	2662.3999999999996	8108.15
2015-01-04	1755.4500000000003	9863.6
2015-01-05	2065.95	11929.55
2015-01-06	2428.95	14358.5
2015-01-07	2202.2000000000003	16560.7
2015-01-08	2838.3499999999995	19399.05

14 Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
with ctt as(
with ct as(
select name,category, sum(quantity*price) revenue
from order_details
join pizzas using(pizza_id)
join pizza_types using(pizza_type_id)
group by name,category
order by category)
select *, rank() over(partition by category order by revenue desc) rk from ct)
select * from ctt where rk<=3;
```

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

name	category	revenue	rk
The Thai Chicken Pizza	Chicken	43434.25	1
The Barbecue Chicken Pizza	Chicken	42768	2
The California Chicken Pizza	Chicken	41409.5	3
The Classic Deluxe Pizza	Classic	38180.5	1
The Hawaiian Pizza	Classic	32273.25	2
The Pepperoni Pizza	Classic	30161.75	3
The Spicy Italian Pizza	Supreme	34831.25	1
The Italian Supreme Pizza	Supreme	33476.75	2
The Sicilian Pizza	Supreme	30940.5	3
The Four Cheese Pizza	Veggie	32265.70000000065	1
The Mexicana Pizza	Veggie	26780.75	2
The Five Cheese Pizza	Veggie	26066.5	3

conclusion :

The Pizza Sales Analysis project showcases MySQL's effectiveness in managing sales data for business decisions.
It highlights my skills in designing efficient databases, generating actionable insights, and presenting them visually.
I aim to apply these skills in future projects and roles.

Thank You