Project presentation

Presented by Pramod S H

Summary

In this Project, I have utilized SQL Queries to solve the questions related to Pizza Sales using MySQL DBMS

DATASET SCHEMA

order_details_id	order_id	pizza_id	quantity
1	1	hawaiian_m	1
2	2	classic_dlx_m	1

01. Order Details

order_id	date	time
1	01-01-2015	11:38:36
2	01-01-2015	11:57:40

02. Orders

pizza_type_id	name	category	ingredients
bbq_ckn	The BB Chicken Pizza	Chicken	Green Peppers, Tomatoes
cali_ckn	The California Chicken Pizza	Chicken	Artichoke, Spinach

03. pizza_types

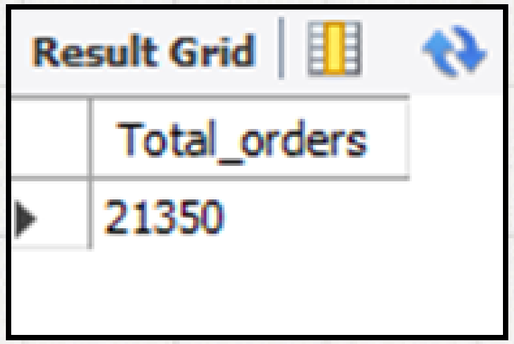
pizza_id	pizza_type_id	size	price
bbq_ckn_s	bbq_ckn	S	12.75
bbq_ckn_m	bbq_ckn	M	16.75

04. pizzas

Retrieve the total number of orders placed

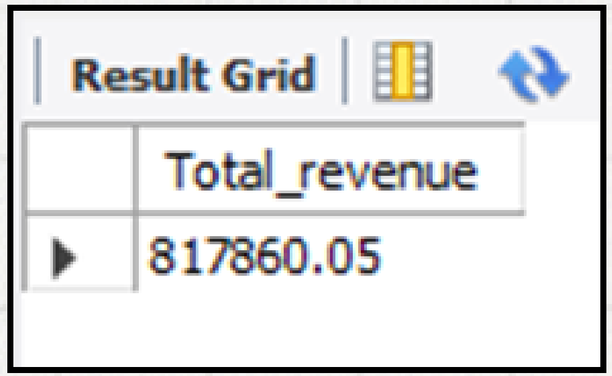
```sql
-- Retrieve the total number of orders placed.

select count(*) as Total_orders
from orders
```



Calculate the total revenue generated from pizza sales

```
select round(sum(od.quantity*pi.price),2) as Total_revenue
from pizzas pi join order_details od
on pi.pizza_id = od.pizza_id
```

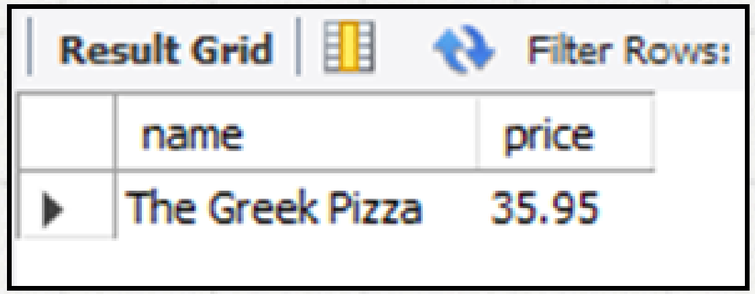


Result Grid

	Total_revenue
▶	817860.05

Identify the highest-priced pizza

```sql
select pt.name, pi.price
from pizza_types pt join pizzas pi
on pt.pizza_type_id = pi.pizza_type_id
order by pi.price desc limit 1;
```



Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered

```
select pi.size, count(od.order_details_id) as count
from pizzas pi join order_details od
on pi.pizza_id = od.pizza_id
group by pi.size;
```

Result Grid

size	count
M	15385
L	18526
S	14137
XL	544
XXL	28

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

```sql
SELECT pt.name AS pizza_name, SUM(od.quantity) AS quantity
FROM pizza_types AS pt
JOIN pizzas AS pi ON pt.pizza_type_id = pi.pizza_type_id
JOIN order_details AS od ON od.pizza_id = pi.pizza_id
GROUP BY pt.name
ORDER BY quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

pizza_name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Find the total quantity of each pizza category ordered

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

Result Grid | Filter Rows

category	Total_Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hour of the day

```sql
select hour(order_time) as hour , count(order_id) as count
from orders
group by hour;
```

Result Grid

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

Find the category-wise distribution of pizzas

```
select category, count(name)
from pizza_types
group by category;
```

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_Quantity
from (select o.order_date as order_date , sum(od.quantity) as quantity
    from orders o join order_details od
    on o.order_id = od. order_id
    group by order_date) as order_quantity
```

Result Grid

	Avg_Quantity
▶	138

Determine the top 3 most ordered pizza types based on revenue

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

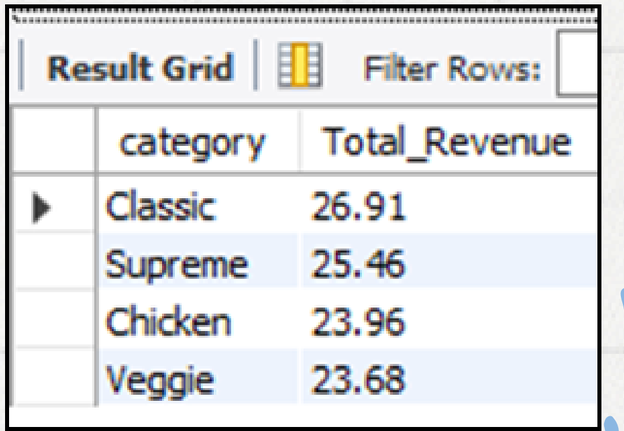
Result Grid | Filter Rows:

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

Calculate the percentage contribution of each pizza type to total revenue

```sql
with revenue as (SELECT
                    ROUND(SUM(od.quantity * pi.price), 2) rev
                FROM pizzas pi JOIN order_details od
                ON pi.pizza_id = od.pizza_id)

SELECT pt.category AS category,
       ROUND((SUM(od.quantity * pi.price)) * 100 / (select rev from revenue) ,2) AS Total_Revenue
FROM pizza_types pt JOIN pizzas pi
ON pt.pizza_type_id = pi.pizza_type_id
JOIN order_details od
ON od.pizza_id = pi.pizza_id
GROUP BY category
ORDER BY Total_Revenue DESC;
```



Result Grid | Filter Rows:

category	Total_Revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze the cumulative revenue generated over time

```sql
select Order_Date, round(sum(revenue) over(order by Order_Date),2) as CUM_Revenue
from (select o.order_date as Order_Date , sum(od.quantity*pi.price) as revenue
      from order_details od join pizzas pi
      on od.pizza_id = pi.pizza_id
      join orders o
      on o.order_id = od.order_id
      group by Order_Date) as sales
```

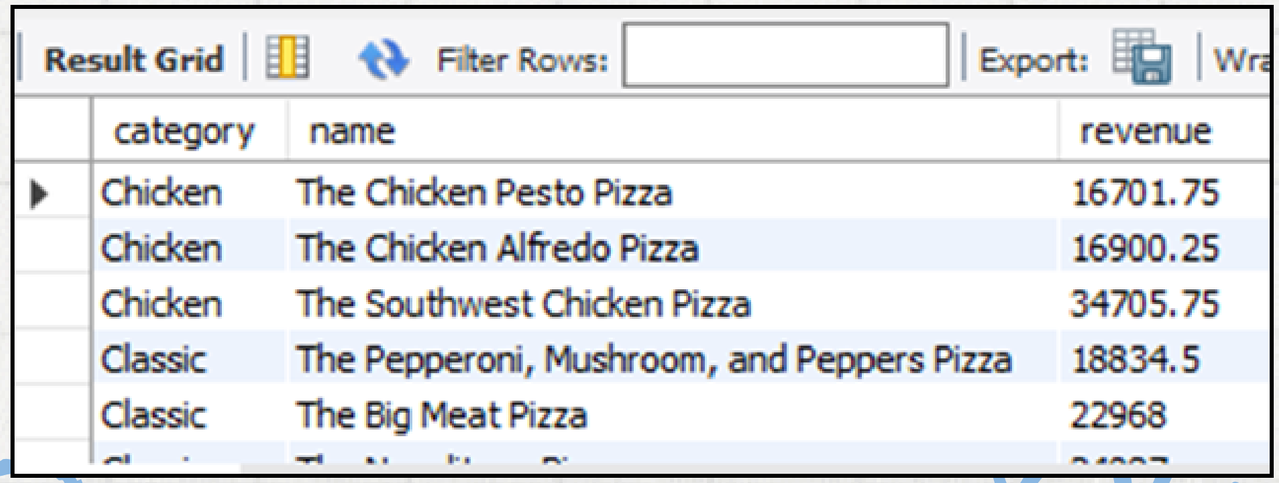
Result Grid | Filter Rows:

Order_Date	CUM_Revenue
2015-01-01	2713.85
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6
2015-01-05	11929.55

.....

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

```sql
select category,name,revenue
from  (select *, rank() over (partition by category order by revenue ) as rn
      from (select pt.category as category,pt.name as name, sum(od.quantity*pi.price) as revenue
         from pizza_types pt join pizzas pi
         on pt.pizza_type_id = pi.pizza_type_id
         join order_details od
         on od.pizza_id = pi.pizza_id
         group by category, name) as a) as b
where rn<=3;
```



category	name	revenue
Chicken	The Chicken Pesto Pizza	16701.75
Chicken	The Chicken Alfredo Pizza	16900.25
Chicken	The Southwest Chicken Pizza	34705.75
Classic	The Pepperoni, Mushroom, and Peppers Pizza	18834.5
Classic	The Big Meat Pizza	22968

.....

Thank you very much!