SQL Project on Pizza Sales

Varun H M

Introduction

Welcome to my SQL Project on Pizza Sales. I'm Varun H M, and I used four comprehensive datasets to explore various aspects of pizza sales. This project aims to provide valuable insights into sales patterns, revenue generation, and customer preferences through detailed SQL queries.

Key Deliverables:

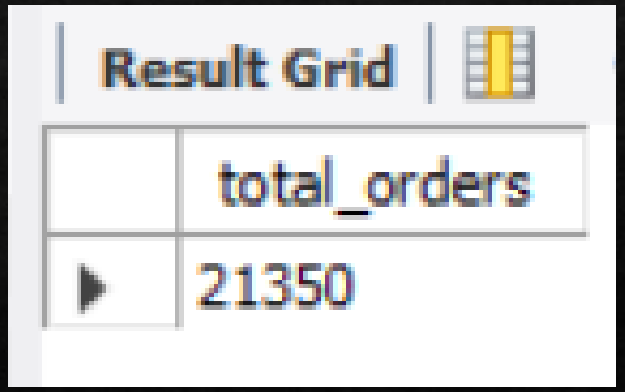
1. Basic Analysis:
   - Total orders, revenue generation, and popular pizza types.
   - Identification of the highest-priced pizza and common sizes.

2. Intermediate Insights:
   - Aggregated data on pizza categories and order distribution by time.
   - Analysis of daily average orders and revenue-based rankings.

3. Advanced Exploration:
   - Revenue contribution breakdown by pizza type.
   - Cumulative revenue analysis and top types per category.

Retrieve the total number of orders placed.

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



Calculate the total revenue generated from pizza sales.

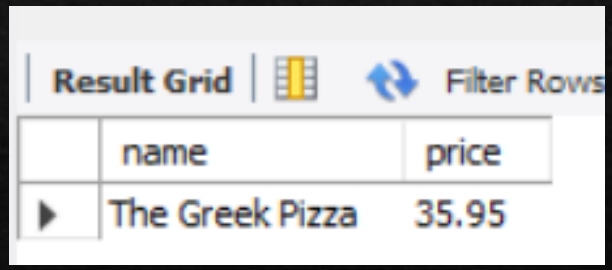
```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```

Result Grid

total_sales
817860.05

Identify the highest-priced pizza.

```sql
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;
```



Identify the most common pizza size ordered.

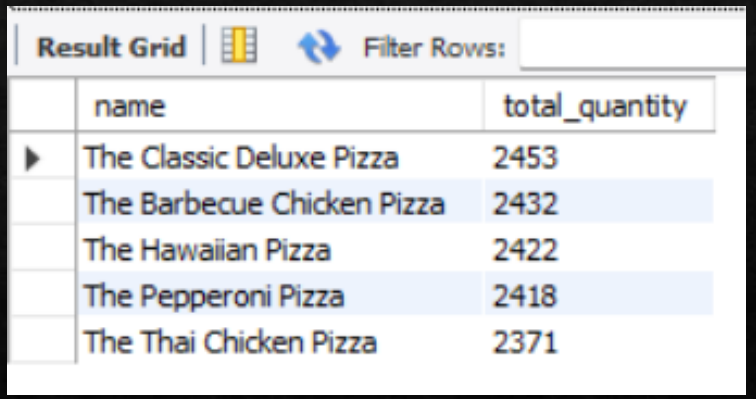
```sql
SELECT
    pizzas.size,
    count(order_details.order_details_id) AS NO_of_time_ordered
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY NO_of_time_ordered DESC
LIMIT 1;
```

Result Grid | Filter Rows:

size	NO_of_time_ordered
L	18526

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

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS total_quantity
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.name
ORDER BY total_quantity DESC
LIMIT 5;
```

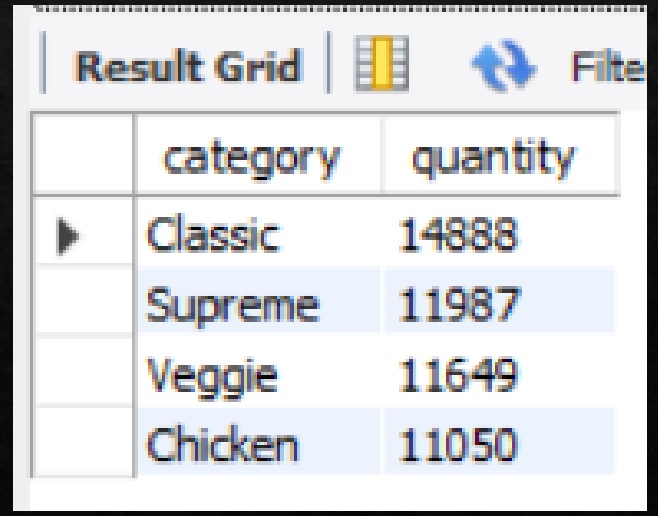


Result Grid | Filter Rows:

name	total_quantity
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
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
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 quantity DESC;
```



	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the day.

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

Result Grid

hour	order_count
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

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 Row

	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(orders), 0) as average_pizzas_ordered_per_day
FROM
    (SELECT
        DATE(orders.orders_date) AS dates,
            SUM(order_details.quantity) AS orders
    FROM
        orders
    JOIN order_details ON orders.orders_id = order_details.order_id
    GROUP BY dates) AS order_quantity;
```

Result Grid | Filter Rows:

	average_pizzas_ordered_per_day
▶	138

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

```sql
SELECT
    pizza_types.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;
```

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 order_details.pizza_id = pizzas.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 pizzas.pizza_id = order_details.pizza_id
GROUP BY category
ORDER BY revenue;
```

Result Grid | Filt

	category	revenue
▶	Veggie	23.68
	Chicken	23.96
	Supreme	25.46
	Classic	26.91

Analyze the cumulative revenue generated over time.

```sql
select  orders_date,
sum(revenue) over(order by orders_date) as cum_revenue
from
(select orders.orders_date,
round(sum(order_details.quantity * pizzas.price),0) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.orders_id = order_details.order_id
group by orders.orders_date) as sales;
```

Result Grid | Filter Rows:

orders_date	cum_revenue
2015-01-01	2714
2015-01-02	5446
2015-01-03	8108
2015-01-04	9863
2015-01-05	11929
2015-01-06	14358
2015-01-07	16560
2015-01-08	19398
2015-01-09	21525
2015-01-10	23989
2015-01-11	25861
2015-01-12	27780
2015-01-13	29830
2015-01-14	32357
2015-01-15	34342
2015-01-16	36936

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

```sql
select *
from
(select category,name,revenue,
rank() over(partition by category order by revenue desc) as ranks
from
(select pizza_types.category,pizza_types.name,
sum((order_details.quantity * pizzas.price)) 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,pizza_types.name)as a)as b
where ranks <=3;
```

Result Grid | Filter Rows: | Export: | Wrap

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