SQL Project on

Pizza Sales

Hello, my name is Sweta Maheshwari and I did some analysis on Pizza sales of a pizza company using SQL queries

Details:

We have a database named pizzahut, the database contains 4 tables having information of what kind of pizza they have, what is the price, size , and the orders they have received and date and time for the order

I have calculated the total revenue generated, what pizza type customers are ordering the most , what is the cumulative revenue, the average numbers of orders we have received every day and many more.

Retrieve the total number of orders placed.

```
SELECT
    COUNT(order_id) AS Total_orders
FROM
    orders;
```

Result Grid

	Total_orders
▶	21350

Calculate the total revenue generated from pizzas.

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

Result Grid

	Total_Revenue
▶	817860.05

Identify the highest priced pizza (pizza name)

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

Result Grid | Filter Rows:

price	Highest_Priced_Pizza
35.95	The Greek Pizza

Identify the most common pizza size ordered.

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

Result Grid | Filter Rows

	size	Total_orders
▶	L	18526

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

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS Total_quantity
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.name
ORDER BY SUM(order_details.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.

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

Result Grid | Filter Ro

category	total_quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Determine the distribution of orders by hours of the day

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

Result Grid

hour	total_orders
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

Join relevant tables to find the categorywise distribution of pizzas

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

Result Grid

category	names
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_per_day), 0)
FROM
    (SELECT
        orders.order_date,
            SUM(order_details.quantity) AS quantity_per_day
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS data_table;
```

Result Grid | Filter Rows:

	ROUND(AVG(quantity_per_day), 0)
▶	138

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

```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS pizza_revenue
FROM
    pizzas
        JOIN
    pizza_types 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 SUM(order_details.quantity * pizzas.price) DESC
LIMIT 3;
```

Result Grid Filter Rows:

name	pizza_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
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    SUM(order_details.quantity * pizzas.price)
                FROM
                    order_details
                        JOIN
                    pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
            2) AS percentage_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 SUM(order_details.quantity * pizzas.price) DESC;
```

Result Grid | Filter Rows:

category	percentage_revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

Analyze cumulative revenue generated over time.

```sql
select order_date, sum(revenue) over(order by order_date) as cumulative_revenue
from
(select orders.order_date, round(sum(pizzas.price*order_details.quantity),2) as revenue
from
orders join order_details
on orders.order_id = order_details.order_id
join
pizzas on pizzas.pizza_id = order_details.pizza_id
group by orders.order_date) as data_table;
```

Result Grid | Filter Rows:

order_date	cumulative_reven
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
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399.05

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

```
select category, name, revenue
from
    (select *, rank() over(partition by category order by revenue desc)as rank_
from
    (select pizza_types.category, pizza_types.name,
    sum(pizzas.price * order_details.quantity) 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 data_table
                    ) as second_table
                        where rank_ <=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