SQL Project: Pizza Sales Data Analysis

This project focuses on analyzing a pizza sales dataset using SQL to uncover key business insights. The objective is to help stakeholders understand sales performance, customer preferences, and product trends through data-driven analysis.

By writing optimized SQL queries, I performed in-depth analysis on various aspects of the pizza business, such as revenue, order patterns, and product demand. This project strengthened my skills in joins, group by, aggregate functions, subqueries, and window functions.

Key Analysis Performed:

Total number of orders placed

- Total revenue generated from pizza sales
- Identification of the highest-priced pizza
- Most common pizza size ordered
- Top 5 most ordered pizza types by quantity
- Category-wise total quantity of pizzas sold
- Hourly distribution of orders.
- Category-wise pizza distribution.
- Average number of pizzas ordered per day
- Top 3 most ordered pizza types based on revenue
- Percentage contribution of each pizza type to total revenue
- Cumulative revenue generated over time
- Top 3 revenue-generating pizzas for each category

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

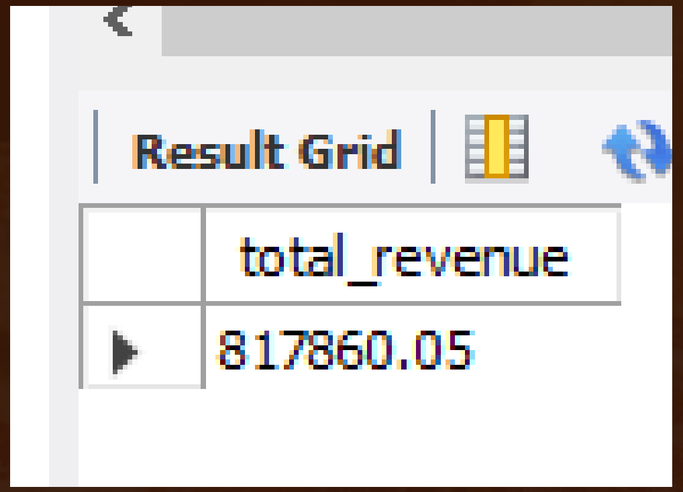
```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

Result Grid

	total_orders
▶	21350

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

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



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;
```

Result Grid | Filter

	name	price
▶	The Greek Pizza	35.95

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

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

Result Grid | Filter Ro

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

LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITY

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

Result Grid | Filter Rows:

name	quantity
The Barbecue Chicken Pizza	4864
13043	4288
13047	4261
The Vegetables + Vegetables Pizza	3052
13045	2453

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
    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 pizza_types.category
ORDER BY total_quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

category	total_quantity
Supreme	11497
Veggie	10319
Chicken	4864
napolitana_l	2453
mexicana_m	2422

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

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

Result Grid | Filter Rows:

HOUR(time)	order_count
11	1166
12	2370
13	2319
14	1369
NULL	1326
15	1361

JOIN RELEVANT TABLES TO FIND THE -- CATEGORY-WISE DISTRIBUTION OF PIZZAS.

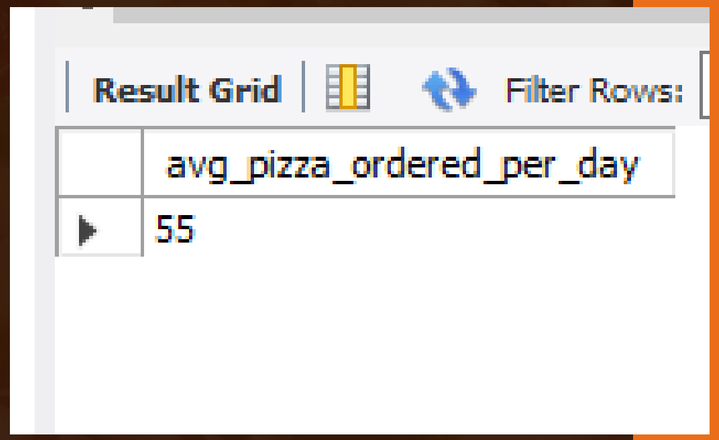
```sql
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

Result Grid | Filter Rows:

category	COUNT(name)
Chicken	2
cali_ckn_m	2
four_cheese_m	2
southw_ckn_m	2
spinach_supr_s	2
southw_ckn_l	2

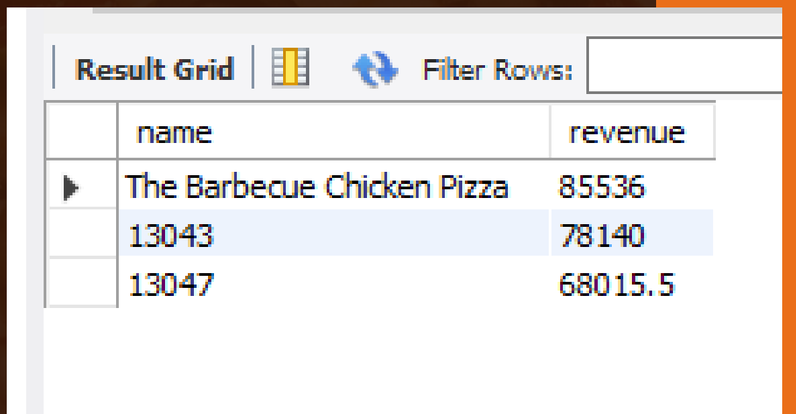
GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE -- NUMBER OF PIZZAS ORDERED PER DAY.

```sql
SELECT
    ROUND(AVG(quantity), 0) as avg_pizza_ordered_per_day
FROM
    (SELECT
        orders.date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.date) AS order_quantity;
```



DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
SELECT
    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.name
ORDER BY revenue DESC
LIMIT 3;
```



name	revenue
The Barbecue Chicken Pizza	85536
13043	78140
13047	68015.5

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    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;
```

Result Grid | Filter Rows:

category	revenue
Supreme	24.04
Veggie	20.42
Chicken	10.46
spinach_supr_s	5.31
napolitana_l	4.67
southw_ckn_m	4.24

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME

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

Result Grid | Filter Rows

date	cum_revenue
14647	16.5
14651	29
14657	65
14664	81.5
14744	92
14759	104

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```sql
select name, revenue from
(select category, name, revenue, rank()
over(partition by category order by revenue desc) as rn
from
(select pizza_types.category, pizza_types.name,
sum((order_details.quantity) * pizzas.price) as revenue
from pizza_types join pizzas
on pizzas.pizza_type_id = pizza_types.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 rn <=3;
```

Result Grid | Filter Rows:

name	revenue
The Pepperoni Pizza	30161.75
13042	16900.25
The Barbecue Chicken Pizza	85536
13042	16701.75
13047	25094
13047	24087

Result 1

Pizza Resto Presentation

THANK YOU FOR YOUR TIME AND ATTENTION!

IF YOU HAVE ANY QUESTIONS, I'D BE HAPPY TO ANSWER THEM.