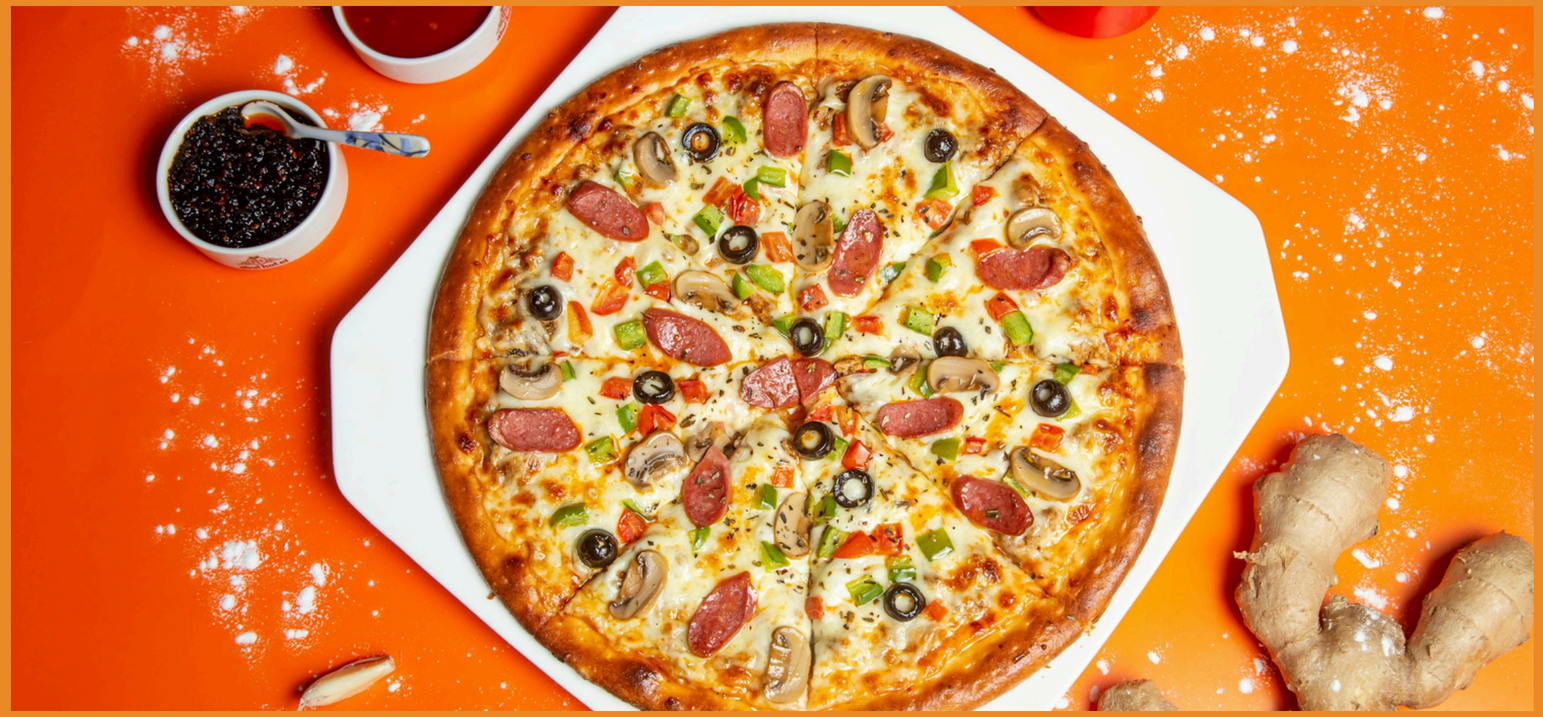
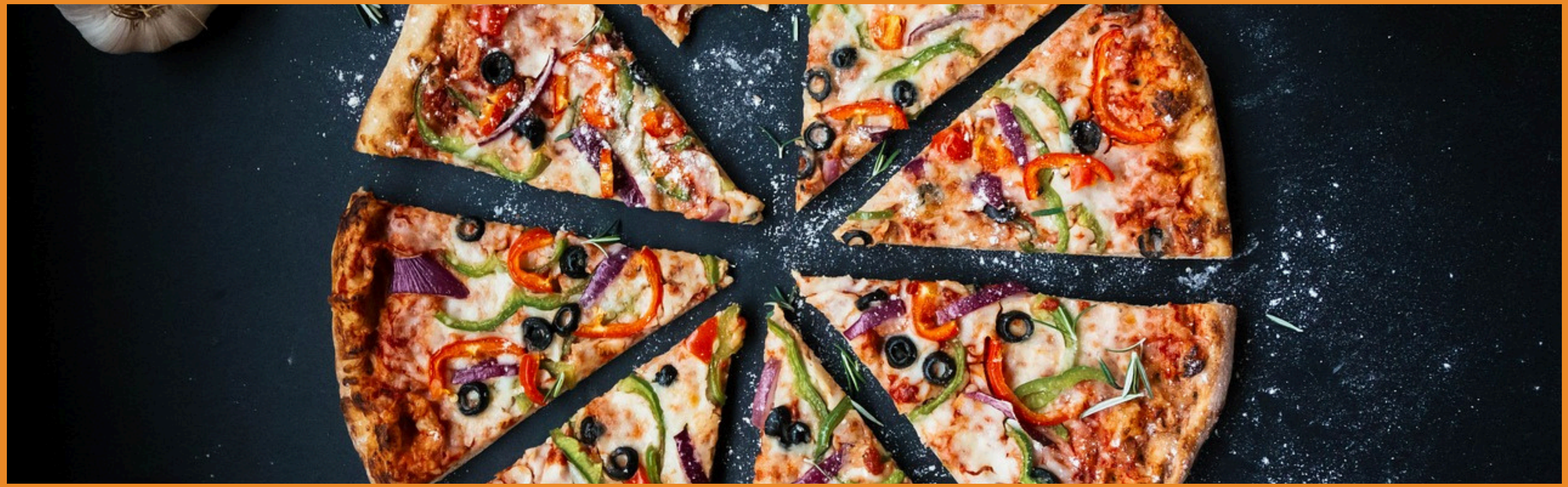
SQL Pizza Sales Data Analysis



GOAL : This project focuses on analyzing pizza sales data using SQL queries. The project is organized into three main sections: Basic, Intermediate, and Advanced queries.



Retrive the total number of orders placed;

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

Result Grid | Filter Rows:

	total_orders
▶	21350

Calculate the total revenue generated from pizza sales

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

Result Grid | Filter Rows:

total_revenue
817860.049999993

Identify the highest-priced pizza

```
select pizza_types.name ,pizzas.price from pizza_types
join pizzas on pizzas.pizza_type_id=pizza_types.pizza_type_id
where pizzas.price =(select max(price) from pizzas);
```

Result Grid | Filter Rows:

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
```

Result Grid | Filter Rows:

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

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

```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 order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5;
```

Result Grid | Filter Rows:

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

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

```sql
SELECT
    pizza_types.category, SUM(order_details.quantity) AS total
FROM
    pizzas
        JOIN
    order_details 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;
```

Result Grid | Filter Rows:

category	total
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

Determine the distribution of orders by hour of the day

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

Result Grid Filter Rows:

hour	order_count
12	2520
13	2455
18	2399
17	2336
19	2009
16	1920
20	1642

Result 48 Result 49 Result 50

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	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_perday
FROM
    (SELECT
        orders.order_date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS quantity;
```

Result Grid | Filter R

	Avg_perday
▶	138

Identify the highest-priced pizza

```sql
SELECT
    pizza_types.name,
    SUM(pizzas.price * order_details.quantity) AS total_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.name
ORDER BY total_revenue DESC
LIMIT 3;
```

Result Grid | Filter Rows:

name	total_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
                    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 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
```

Result Grid | Filter Rows:

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

Analyze the cumulative revenue generated over time.

```
select order_date,
sum(revenue) over (order by order_date) as cum_revenue
from
(select orders.order_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.order_date) as sales;
```

Result Grid | Filter Rows:

order_date	cum_revenue
2015-01-01	2713.850000000004
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.

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

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25

Result 104 Result 105 Result 106 Result 107

Thanks