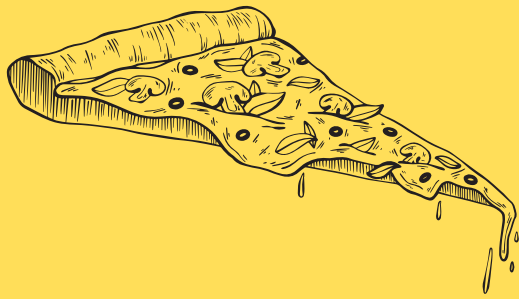
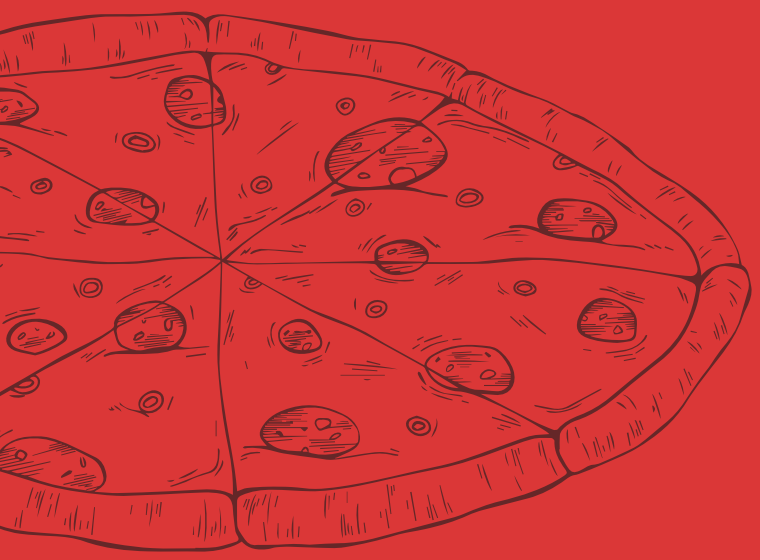
Pizza





Executed Pizza Sales Analysis using SQL.

Himanshu John Marandi

Retrieve the total number of orders placed.

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

Result Grid

	total_orders
▶	18918

Calculate the total revenue generated from pizza sales.

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

Result Grid

	total_revenue
▶	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 price DESC
LIMIT 1;
```

Result Grid | Filter Rc

	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

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

Result Grid

	size	quantity
▶	L	18526

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

```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS top
FROM
    pizzas
        JOIN
    pizza_types 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 top DESC
LIMIT 5;
```

Result Grid | Filter Rows:

name	top
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
    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 quantity DESC;
```

Result Grid | Filter R

category	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 quantity
FROM
    orders
GROUP BY HOUR(order_time);
```

Result Grid

hour	quantity
11	1086
12	2217
13	2173
14	1330
15	1313

Join relevant tables to find the category-wise distribution of pizzas.

```sql
SELECT
    pizza_types.category, COUNT(pizza_types.pizza_type_id) as Distribution
FROM
    pizza_types
GROUP BY pizza_types.category;
```

Result Grid | Filter R

	category	Distribution
▶	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(Total_order), 0) as Average
FROM
    (SELECT
        orders.order_date,
            SUM(order_details.quantity) AS Total_order
    FROM
        orders
    JOIN order_details ON order_details.order_id = orders.order_id
    GROUP BY orders.order_date) AS avg_perday;
```

Result Grid

	Average
▶	138

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 pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.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
SELECT
    pizza_types.category,
    ROUND((SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                2)
                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) * 100),
            2) AS Total_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 pizza_types.category;
```

Result Grid | Filter Rows:

category	Total_revenue
Classic	26.91
Veggie	23.68
Supreme	25.46
Chicken	23.96

Analyze the cumulative revenue generated over time.

```sql
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

pizzas join order_details on

pizzas.pizza_id=order_details.pizza_id

join orders on order_details.order_id=orders.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
2015-01-06	14358.5
2015-01-07	16560.7

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

```sql
select category,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

 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 rn<=3;
```

Result Grid | Filter Rows: | Expo

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