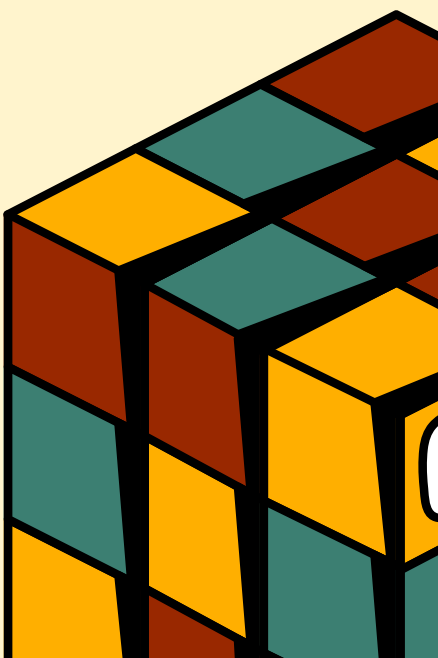
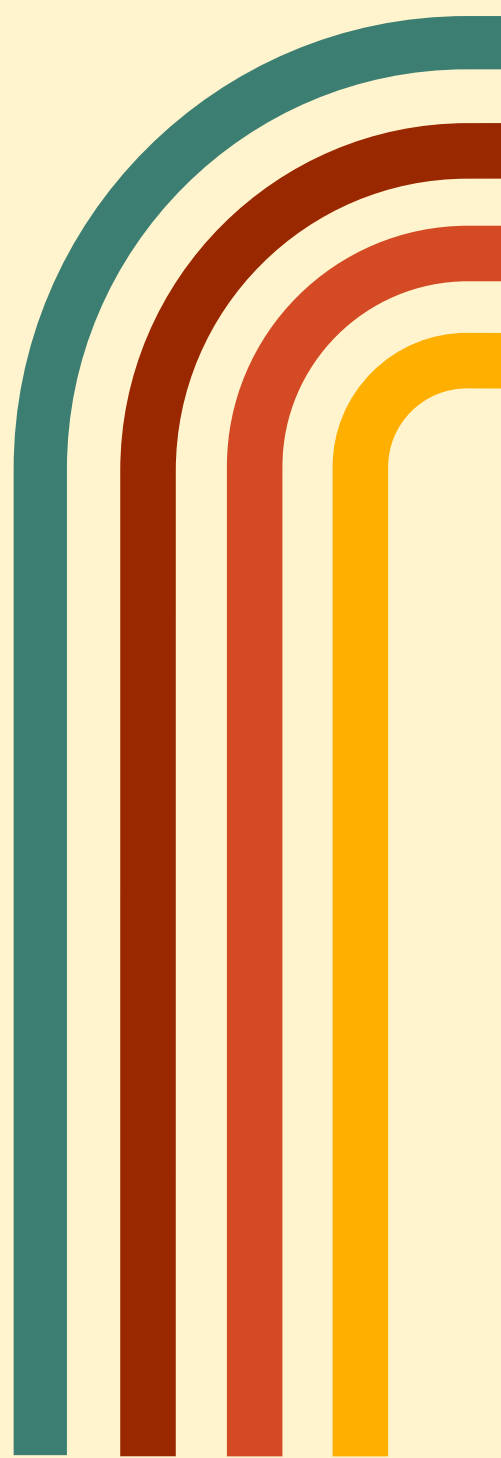


# Project on SQL

**SQL QUERIES ON PIZZA SALES DATA SET**

15 June, 2024



# 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 pizza sales.

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

Result Grid	
	Revenue
▶	817860.05

# Identify the highest-priced pizza.

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

Result Grid			Filter Rows	
	name	price		
▶	The Greek Pizza	35.95		

# Identify the most common pizza size ordered.

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

Result Grid			Filter
	Order_count	size	
▶	18526	L	

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

```
SELECT
    name, SUM(quantity)
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 name
ORDER BY SUM(quantity) DESC
LIMIT 5;
```

Result Grid			Filter Rows:
	name	SUM(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.

```
SELECT
    category, SUM(quantity)
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 category
ORDER BY SUM(quantity) DESC
LIMIT 5;
```

Result Grid			Filter Rows:
	category	SUM(quantity)	
▶	Classic	14888	
	Supreme	11987	
	Veggie	11649	
	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_time);
```

	Hour	Order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336



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

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

	category	Number
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

**Group the orders by date and calculate the average number of pizzas ordered per day.**

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

Result Grid			Filter Rows
	avg_pizza_ordered_per_day		
	138		

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



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

```
SELECT
    category,
    round((SUM(quantity * price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS Revenue
    FROM
        order_details
        JOIN
        pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100 ,2)AS revenue
FROM
    pizzas
    JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
    JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY category
ORDER BY revenue DESC;
```

Result Grid					Filter
	category	revenue			
▶	Classic	26.91			
	Supreme	25.46			
	Chicken	23.96			
	Veggie	23.68			

# 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	2713.8500000000000004		
	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 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
where rn<=3;
```

Result Grid				Filter Rows:
	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	