

# **SQL PROJECT ON PIZZA SALES FOR DATA ANALYSIS**

**Name : Sarvesh Vinayak Wankhade**

# SQL Questions on Pizza Sales

## Basic:

- 1] Retrieve the total number of orders placed.
- 2] Calculate the total revenue generated from pizza sales.
- 3] Identify the highest-priced pizza.
- 4] Identify the most common pizza size ordered.
- 5] List the top 5 most ordered pizza types along with their quantities.

## Intermediate / Advanced:

- 6] Join the necessary tables to find the total quantity of each pizza category ordered.
- 7] Determine the distribution of orders by hour of the day.
- 8] Join relevant tables to find the category-wise distribution of pizzas.
- 9] Group the orders by date and calculate the average number of pizzas ordered per day.
- 10] Determine the top 3 most ordered pizza types based on revenue.
- 11] Calculate the percentage contribution of each pizza type to total revenue.

⋮ ⋮ ⋮

⋮ ⋮ ⋮

# 1] Retrieve the total number of orders placed

```
SELECT  
    COUNT(*) AS TOTALS_ORDERS  
FROM  
    ORDERS;
```

## Output :

Result Grid	
	TOTALS_ORDERS
	21350

## 2] Calculate the total revenue generated from pizza sales

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

### Output :

Result Grid	
	total_revenue
	817860.05

### 3] Identify the highest-priced pizza

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

### Output :

Result Grid    Filter Rows:  Search

	name	highest_priced_pizza
	The Greek Pizza	35.95

## 4] Identify the most common pizza size ordered

```
SELECT
    pizzas.size, COUNT(orders_details.orders_details_id)
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY COUNT(orders_details.orders_details_id) DESC
LIMIT 1;
```

### Output :

Result Grid    Filter Rows:

size	Common_Size
L	18526

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

```
SELECT
    pizza_types.name, sum(orders_details.order_quantity) as quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
        group by pizza_types.name order by sum(orders_details.order_quantity) desc limit 5;
```

### Output :

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

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

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

### Output :

Result Grid		Filter
	category	quantity
	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

## 7] 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);
```

### Output :

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

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

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

Output :

Result Grid Filter

category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

## 9] Group the orders by date and calculate the average number of pizzas ordered per day.

```
SELECT |  
    ROUND(AVG(quantity), 0) as avg_pizza_per_day  
FROM  
    (SELECT  
        orders.order_date,  
        SUM(orders_details.order_quantity) AS quantity  
    FROM  
        orders  
    JOIN orders_details ON orders.order_id = orders_details.order_id  
    GROUP BY orders.order_date) AS order_quantity_day;
```

### Output :

Result Grid			F
avg_pizza_per_d...			
138			

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

```
SELECT
    pizza_types.name,
    ROUND(SUM(orders_details.order_quantity * pizzas.price),
          2) AS total_revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY total_revenue DESC
LIMIT 3;
```

### Output :

Result Grid    Filter Rows:    Search

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

# 11] Calculate the percentage contribution of each pizza type to total revenue

```
select pizza_types.category, round((sum(orders_details.order_quantity * pizzas.price) /(SELECT  
ROUND(SUM(orders_details.order_quantity * pizzas.price),2)  
AS total_revenue  
FROM  
orders_details  
JOIN  
pizzas ON pizzas.pizza_id = orders_details.pizza_id ))*100,2) as revenue  
from pizza_types join pizzas  
on pizza_types.pizza_type_id = pizzas.pizza_type_id  
join orders_details  
on orders_details.pizza_id = pizzas.pizza_id  
group by pizza_types.category order by revenue desc;
```

## Output:

Result Grid    Filter Rows:

	category	revenue
	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

# THANK YOU



<https://github.com/SarveshWankhade30>

<https://www.linkedin.com/in/sarvesh-wankhade/>