# Pizza Sales

## A Data Analysis Project using MySQL

- Moyya Deepak

# INTRODUCTION

We’re embarking on an exciting data analysis project related to a pizza sales store. We have four interesting tables to investigate, and our mission is to solve 13 practical business challenges using MySQL queries. 😊

# PROBLEMS TO SOLVE

Here are the problems which we have to solve from pizza sales, the questions are listed below:

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.
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.
12. Analyze the cumulative revenue generated over time.
13. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

# TABLE OF SCHEMAS

## 01 ORDERS

| Column Name | Datatype |
|---|---|
| order_id | INT |
| order_date | DATE |
| order_time | TIME |

## 02 ORDERS DETAILS

| Column Name | Datatype |
|---|---|
| order_details_id | INT |
| order_id | INT |
| pizza_id | TEXT |
| quantity | INT |

## 03 PIZZAS

| Column Name | Datatype |
|---|---|
| pizza_type_id | TEXT |
| size | TEXT |
| price | DOUBLE |
| pizzascol | VARCHAR(45) |

## 04 PIZZA_TYPES

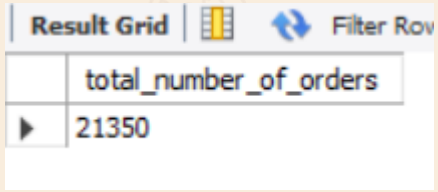
| Column Name | Datatype |
|---|---|
| pizza_type_id | TEXT |
| name | TEXT |
| category | TEXT |
| ingredients | TEXT |

1. “Retrieve the total number of orders placed.”

**Query:**

```
select count(order_id) as total_number_of_orders
from orders;
```

**Output:** Total number of orders placed are 21350



## 2. “Calculate the total revenue generated from pizza sales.”

**Query:**

```
select round(sum(orders_details.quantity * pizzas.price), 2)
as total_revenue
from orders_details join pizzas
on orders_details.pizza_id = pizzas.pizza_id;
```

**Output:** The Total Revenue is 8,17,860.05

| | total_revenue |
|---|---|
| ▶ | 817860.05 |

# 3. “Identify the highest-priced pizza.”

**Query:**

```
select pizza_types.name, pizzas.price as Highest_Price
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
order by price desc limit 1;
```

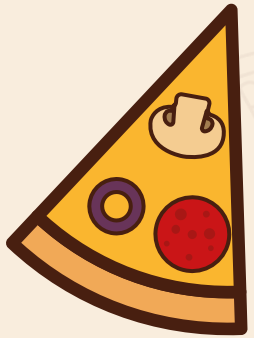
**Output:** Therefore, the highest priced pizza is “The Greek Pizza” for 35.95

| name | Highest_Price |
| --- | --- |
| The Greek Pizza | 35.95 |

# 4. “Identify the most common pizza size ordered.”

**Query:**

```
select pizzas.size as Common_size,
 count(orders_details.order_details_id) as Orders
from pizzas join orders_details
on pizzas.pizza_id = orders_details.pizza_id
group by pizzas.size order by Orders desc limit 1;
```



**Output:** The most common pizza size is “Large (L)”

| Common_size | Orders |
|---|---|
| L | 18526 |

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

**Query:**

```
select pizza_types.name as Top5_pizzas,
 sum(orders_details.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 Quantity desc limit 5;
```

**Output:** These are the top-5 most ordered pizzas

| Top5_pizzas | 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.”

**Query:**

```
select pizza_types.category, sum(orders_details.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:** Each Pizza Category and their quantities

| category | Quantity |
|---|---|
| Classic | 14888 |
| Supreme | 11987 |
| Veggie | 11649 |
| Chicken | 11050 |

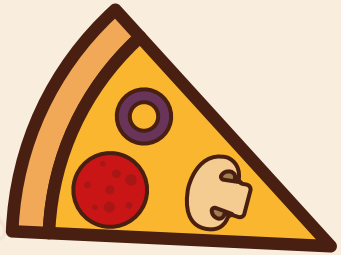
# 7. “Determine the distribution of orders by hour of the day.”

**Query:**

```
select hour(order_time), count(order_id)
from orders group by hour(order_time);
```

**Output:**

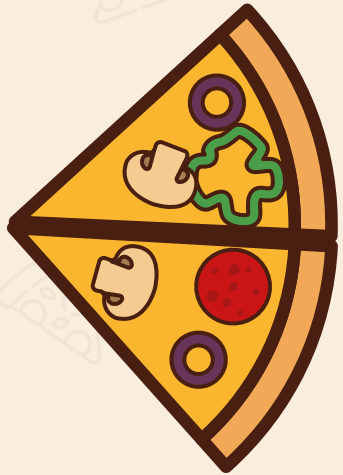
| hour(order_time) | count(order_id) |
|---|---|
| 11 | 1231 |
| 12 | 2520 |
| 13 | 2455 |
| 14 | 1472 |
| 15 | 1468 |
| 16 | 1920 |
| 17 | 2336 |
| 18 | 2399 |
| 19 | 2009 |
| 20 | 1642 |
| 21 | 1198 |
| 22 | 663 |



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

**Query:**

```
select category, count(name)
 from pizza_types group by category;
```



**Output:** Category-wise distribution of Pizzas.

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

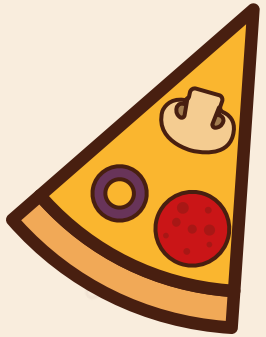
# 9. “Group the orders by date and calculate the average number of pizzas ordered per day.”

**Query:**

```
select round(avg(Quantity), 0) as average_pizzas_order_per_day from
(select orders.order_date, sum(orders_details.quantity) as Quantity
from orders join orders_details on orders.order_id = orders_details.order_id
group by orders.order_date) as Order_quantity;
```

**Output:** Therefore, the average orders per day are “138”

| average_pizzas_order_per_day |
| --- |
| 138 |



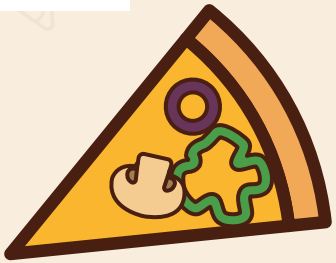
# 10. “Determine the top 3 most ordered pizza types based on revenue.”

**Query:**

```
select pizza_types.name, sum(orders_details.quantity * pizzas.price) 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.name order by revenue desc limit 3;
```

**Output:** Top-3 most ordered pizzas based on Revenue

| name | 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.”

**Query:**

```
select pizza_types.category, round(sum(orders_details.quantity * pizzas.price)
/ (select round(sum(orders_details.quantity * pizzas.price), 2)
as total_revenue
from orders_details join pizzas on orders_details.pizza_id = pizzas.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:**

| category | revenue |
|---|---|
| Classic | 26.91 |
| Supreme | 25.46 |
| Chicken | 23.96 |
| Veggie | 23.68 |

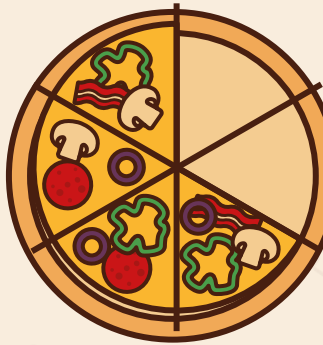
# 12. “Analyze the cumulative revenue generated over time.”

**Query:**

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

**Output:**

| order_date | cum_revenue |
|---|---|
| 2015-01-01 | 2713.8500000000004 |
| 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 |



# 13. “Determine the top 3 most ordered pizza types based on revenue for each pizza category.”

**Query:**

```
select category, name, revenue,
rank() over(partition by category order by revenue desc) as Rnk from
(select category, name, revenue,
rank() over(partition by category order by revenue desc) as Rnk from
(select pizza_types.category, pizza_types.name, sum((orders_details.quantity)*pizzas.price) 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, pizza_types.name order by revenue desc) as A) as B
where Rnk <=3;
```

**Output:**

| category | name | revenue | Rnk |
|---|---|---|---|
| Chicken | The Thai Chicken Pizza | 43434.25 | 1 |
| Chicken | The Barbecue Chicken Pizza | 42768 | 2 |
| Chicken | The California Chicken Pizza | 41409.5 | 3 |
| Classic | The Classic Deluxe Pizza | 38180.5 | 1 |
| Classic | The Hawaiian Pizza | 32273.25 | 2 |
| Classic | The Pepperoni Pizza | 30161.75 | 3 |
| Supreme | The Spicy Italian Pizza | 34831.25 | 1 |
| Supreme | The Italian Supreme Pizza | 33476.75 | 2 |
| Supreme | The Sicilian Pizza | 30940.5 | 3 |
| Veggie | The Four Cheese Pizza | 32265.70000000065 | 1 |
| Veggie | The Mexicana Pizza | 26780.75 | 2 |
| Veggie | The Five Cheese Pizza | 26066.5 | 3 |

# THANK YOU

moyyadeepak2003@gmail.com

www.linkedin.com/in/deepak-moyya-554493239

github.com/MoyyaDeepak