# SQL PROJECT

# Introduction

- Hello, my name is Shubham Singh and in this project I have used SQL queries to solve multiple questions ranging from difficulty levels of easy to hard.
- I have used MySQL to solve the given questions.
- We have 4 tables in the database which provide the details about Pizza Sales of an outlet.

# Q1) Calculate the total revenue generated from pizza sales.

```
 9 •   select round(sum(pizzas.price*order_details.quantity),2) as revenue
10     from pizzas join order_details
11     on pizzas.pizza_id = order_details.pizza_id
12
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| revenue |
| --- |
| 817860.05 |

# Q2) Identify the highest-priced pizza.

```
 7 •  select name, price, size from pizza_types join pizzas
 8    on pizza_types.pizza_type_id = pizzas.pizza_type_id
 9    order by price desc
10    limit 1
11
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conten

| | name | price | size |
|---|---|---|---|
| ▶ | The Greek Pizza | 35.95 | XXL |

# Q3) Join the necessary tables to find the total quantity of each pizza category ordered.

```
 5 •  select sum(order_details.quantity) as total_quantity, pizza_types.category
 6    from pizza_types join pizzas
 7    on pizza_types.pizza_type_id = pizzas.pizza_type_id
 8    join order_details
 9    on order_details.pizza_id = pizzas.pizza_id
10    group by pizza_types.category
11    order by total_quantity desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

# Q4) Determine the distribution of orders by hour of the day.

```
10 •    select hour(order_time), count(order_id) from orders
11      group by hour(order_time)
12
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

| hour(order_time) | count(order_id) |
|---|---|
| 11 | 1200 |
| 12 | 2467 |
| 13 | 2403 |
| 14 | 1443 |
| 15 | 1431 |
| 16 | 1871 |

# Q5) 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
    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
ORDER BY revenue DESC
LIMIT 3
```

| name | revenue |
|---|---|
| The Thai Chicken Pizza | 43434.25 |
| The Barbecue Chicken Pizza | 42768 |
| The California Chicken Pizza | 41409.5 |

# Q6) Calculate the percentage contribution of each pizza type to total revenue.

```
select pizza_types.category, round((sum(order_details.quantity*pizzas.price) /
(select round(sum(order_details.quantity*pizzas.price),0)
from order_details join pizzas
on order_details.pizza_id= pizzas.pizza_id)*100),0)  as percentage_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
```

| | category | percentage_revenue |
|---|---|---|
| ▶ | Classic | 27 |
| | Veggie | 24 |
| | Supreme | 25 |
| | Chicken | 24 |

# Q7) 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 b
where rn<=3
```

| 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 |
| Supreme | The Sicilian Pizza | 30940.5 |
| Veggie | The Four Cheese Pizza | 32265.70000000065 |
| Veggie | The Mexicana Pizza | 26780.75 |
| Veggie | The Five Cheese Pizza | 26066.5 |

# DATABASE SCHEMA

**pizzas**

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| pizza_id | text | YES | | NULL | |
| pizza_type_id | text | YES | | NULL | |
| size | text | YES | | NULL | |
| price | double | YES | | NULL | |

**orders**

| Field | Type | Null | Key | Default |
|---|---|---|---|---|
| order_id | int | NO | PRI | NULL |
| order_date | date | NO | | NULL |
| order_time | time | NO | | NULL |

**pizza_types**

| Field | Type | Null | Key | Default |
|---|---|---|---|---|
| pizza_type_id | text | YES | | NULL |
| name | text | YES | | NULL |
| category | text | YES | | NULL |
| ingredients | text | YES | | NULL |

**order_details**

| Field | Type | Null | Key | Default |
|---|---|---|---|---|
| order_details_id | int | NO | PRI | NULL |
| order_id | int | NO | | NULL |
| pizza_id | text | NO | | NULL |
| quantity | int | NO | | NULL |

# THANK YOU