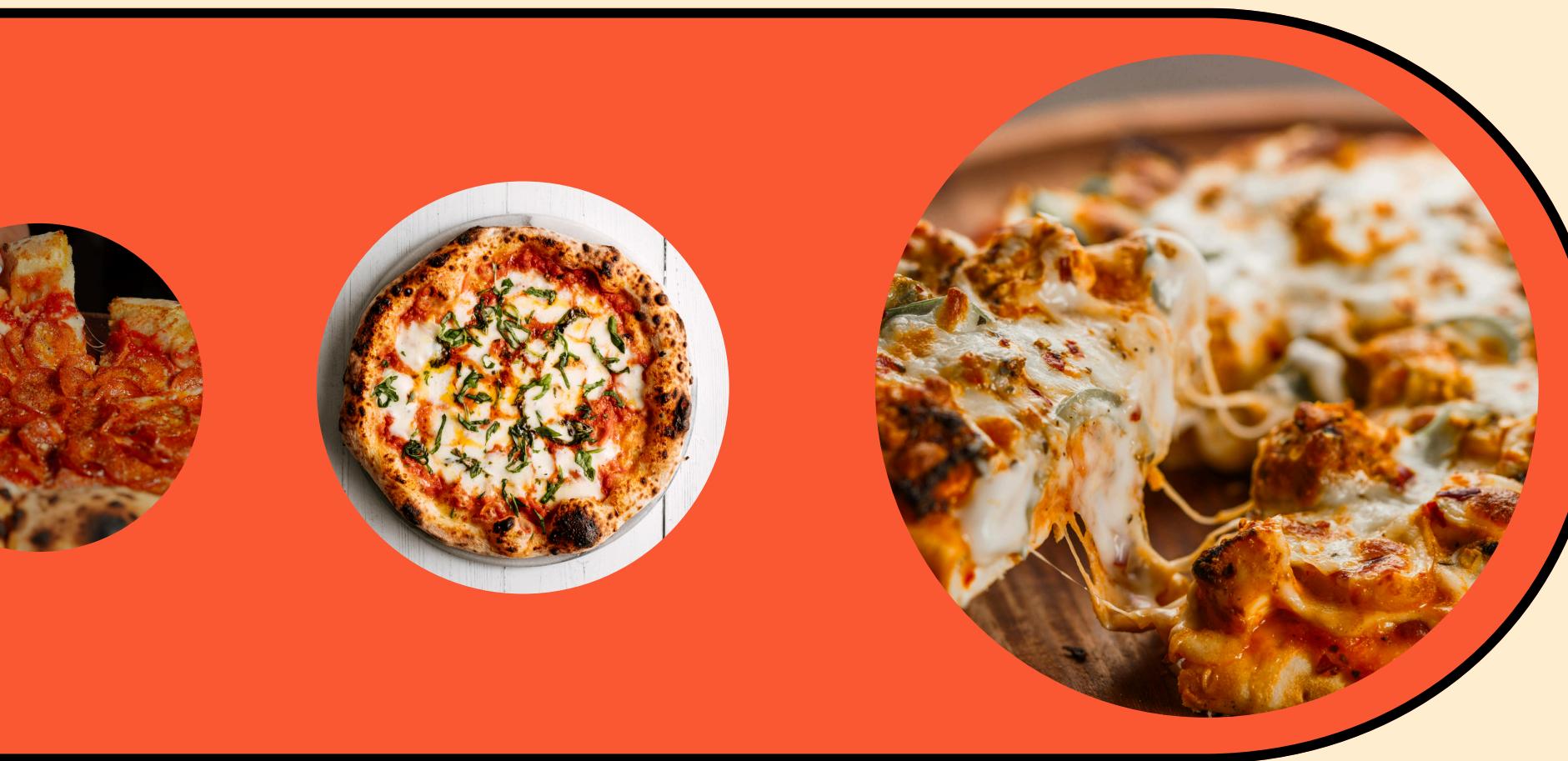


PIZZA

ORDER NOW



WELCOME OUR PIZZA



This project analyzes a pizza shop's sales data using MySQL to extract meaningful business insights. The database consists of orders, order details, pizzas, and pizza categories. SQL queries were used to calculate total and cumulative revenue, identify top-selling pizza types, measure percentage contribution to total revenue, and determine category-wise top performers.

The project demonstrates strong knowledge of SQL concepts such as JOINs, GROUP BY, aggregate functions, subqueries, and window functions to transform raw transactional data into actionable business insights.

1. Retrieve the total number of orders placed.

```
3 • select count(order_id) from orders;
```

Result Grid | Filter Rows:

| | count(order_id) |
|---|-----------------|
| ▶ | 21350 |



2. Calculate the total revenue generated from pizza sales.

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

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



3. Identify the highest-priced pizza.

```
select pt.pizza_type_id, p.price  
from pizza_types pt join pizzas p  
on pt.pizza_type_id = p.pizza_type_id  
order by p.price desc limit 1;
```

Result Grid | Filter Rows:

| | pizza_type_id | price |
|---|---------------|-------|
| ▶ | the_greek | 35.95 |



4. Identify the most common pizza size ordered.

```
select p.size, count(order_details_id) as order_count
from pizzas p join orders_details o
on p.pizza_id = o.pizza_id
group by p.size
order by order_count desc;
```

Result Grid | Filter Rows:

| | size | order_count |
|---|------|-------------|
| ▶ | L | 18526 |
| | M | 15385 |
| | S | 14137 |
| | XL | 544 |
| | XXL | 28 |



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

```
select pt.name, sum(od.quantity) as total_quantity
from pizza_types pt join pizzas p
on pt.pizza_type_id = p.pizza_type_id
join orders_details od
on od.pizza_id = p.pizza_id
group by pt.name
order by total_quantity desc limit 5;
```

| | name | total_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 pt.category, sum(od.quantity) as total_quantity
from pizza_types pt join pizzas p
on pt.pizza_type_id = p.pizza_type_id
join orders_details od
on od.pizza_id = p.pizza_id
group by pt.category
order by total_quantity desc;
```

Result Grid | Filter Rows:

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



7. Join the necessary tables to find the total quantity of each pizza category ordered.

```
select hour(order_time) as hours, count(order_id) as total_orders  
from orders  
group by hours  
order by hours;
```

| | hours | total_orders |
|----|-------|--------------|
| 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 | |



8. find the category-wise distribution of pizzas.

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

| | category | pizzas |
|---|----------|--------|
| ▶ | 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)) as avg_orders_per_day  
from (select o.order_date, sum(od.quantity) as quantity  
from orders o join orders_details od  
on o.order_id = od.order_id  
group by o.order_date) as order_quantity;
```

| | avg_orders_per_day |
|---|--------------------|
| ▶ | 138 |



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

```
select pt.name, sum(od.quantity * p.price) as revenue
from pizza_types pt join pizzas p
on pt.pizza_type_id = p.pizza_type_id
join orders_details od
on od.pizza_id = p.pizza_id
group by pt.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 |



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

- ```
select pt.category, concat(round(sum(od.quantity * p.price) /
 (select round(sum(o.quantity * p.price), 2) as total_revenue
 from orders_details o join pizzas p
 on o.pizza_id = p.pizza_id) * 100, 2), '%') as revenue_percentage
 from pizza_types pt join pizzas p
 on p.pizza_type_id = pt.pizza_type_id
 join orders_details od
 on od.pizza_id = p.pizza_id
 group by pt.category
 order by revenue_percentage desc;
```

|   | category | revenue_percentage |
|---|----------|--------------------|
| ▶ | Classic  | 26.91%             |
|   | Supreme  | 25.46%             |
|   | Chicken  | 23.96%             |
|   | Veggie   | 23.68%             |



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

```
SELECT
 order_date,
 SUM(revenue) OVER (ORDER BY order_date) AS cum_revenue
FROM (
 SELECT
 o.order_date,
 SUM(p.price * od.quantity) AS revenue
 FROM orders_details od
 JOIN pizzas p
 ON od.pizza_id = p.pizza_id
 JOIN orders o
 ON o.order_id = od.order_id
 GROUP BY o.order_date
) AS day_revenue;
```

|   | 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            |
|   | 2015-01-07 | 16560.7            |
|   | 2015-01-08 | 19399.05           |
|   | 2015-01-09 | 21526.4            |
|   | 2015-01-10 | 23990.350000000002 |
|   | 2015-01-11 | 25862.65           |
|   | 2015-01-12 | 27781.7            |
|   | 2015-01-13 | 29831.300000000003 |
|   | 2015-01-14 | 32752.700000000004 |



# 13. 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 total_rank
 from
 (select
 pt.category, pt.name, sum(od.quantity * p.price) as revenue
 from
 pizza_types pt join pizzas p
 on pt.pizza_type_id = p.pizza_type_id
 join orders_details od
 on od.pizza_id = p.pizza_id
 group by pt.category, pt.name) as a) as b
where total_rank <=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           |





**THANK YOU**