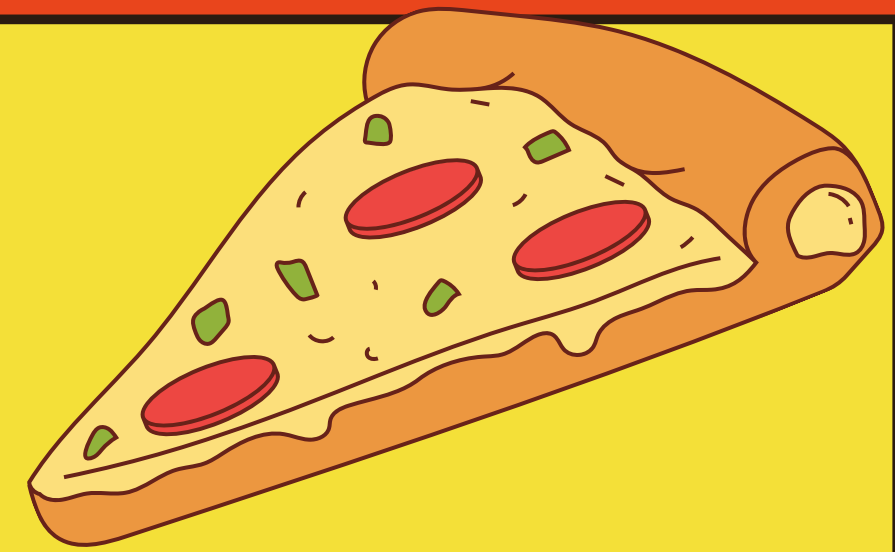
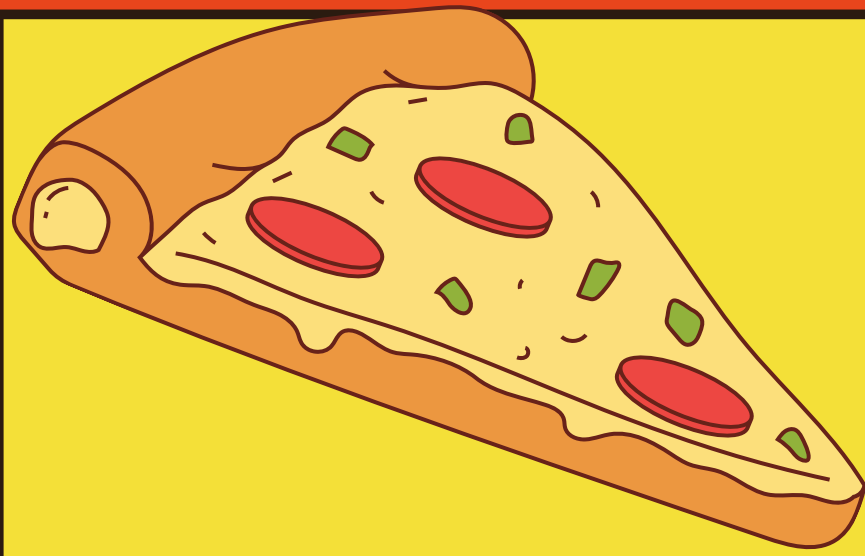


PIZZAS SALES

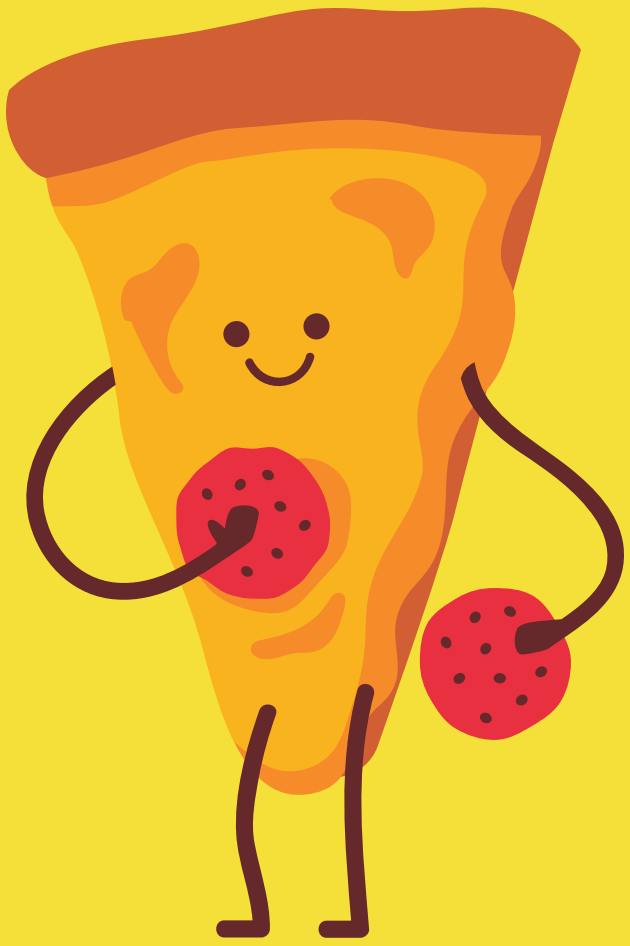




Hello

**My name is sharada pujari and in this project I
have utilise SQL queries to solve a question that
were related to pizzas sales**

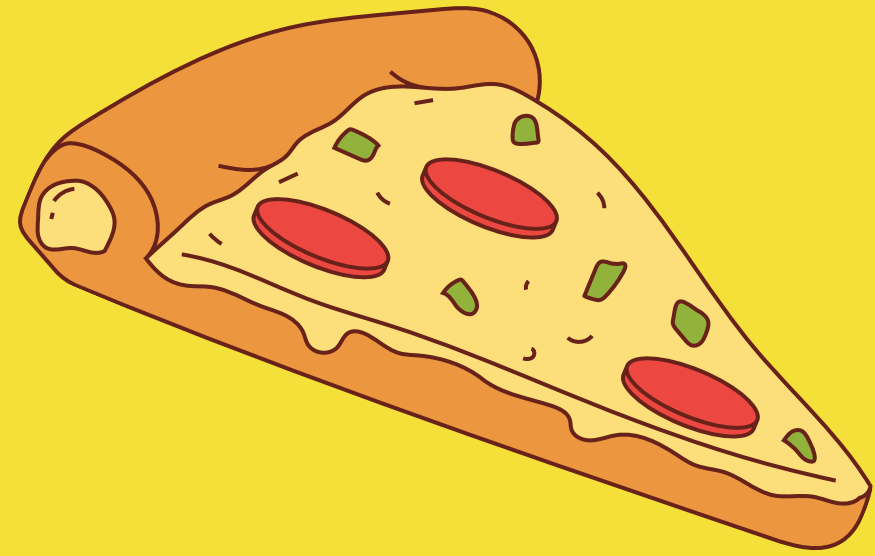




- total number of orders placed.
- total revenue generated from pizza sales
- total quantity of each pizza category ordered.
- distribution of orders by hour of the day
- top 3 most ordered pizza types based on revenue.
- percentage contribution of each pizza type to total revenue.
- cumulative revenue generated over time



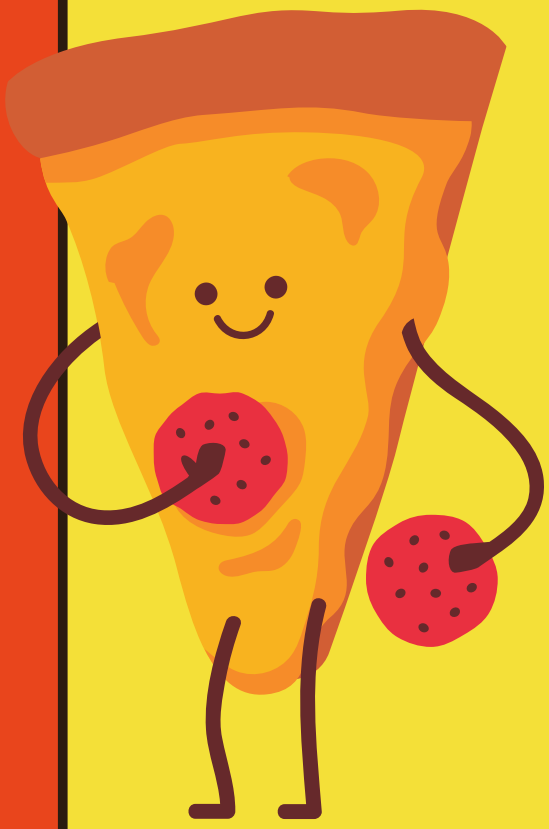
total number of orders placed.



```
1  -- Q1 Retrieve the total number of orders placed.
2  • SELECT
3      COUNT(order_id) AS tatal_order
4  FROM
5      orders;
6
```



| Result Grid | | Filter |
|-------------|-------------|--------|
| | tatal_order | |
| ▶ | 21350 | |



Calculate the total revenue generated from pizza sales.

```
1  -- Q2 Calculate the total revenue generated from pizza sales.
2  • SELECT
3      ROUND(SUM(orders_details.quantity * pizzas.price),
4             2) AS total_sales
5  FROM
6      orders_details
7      JOIN
8      pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

| Result Grid | |
|-------------|-------------|
| | total_sales |
| ▶ | 817860.05 |



Identify the highest-priced pizza.

```
1  -- Identify the highest-priced pizza.
2  • SELECT
3      pizza_types.name, pizzas.price
4  FROM
5      pizza_types
6      JOIN
7      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8  ORDER BY pizzas.price DESC
9  LIMIT 1;
```

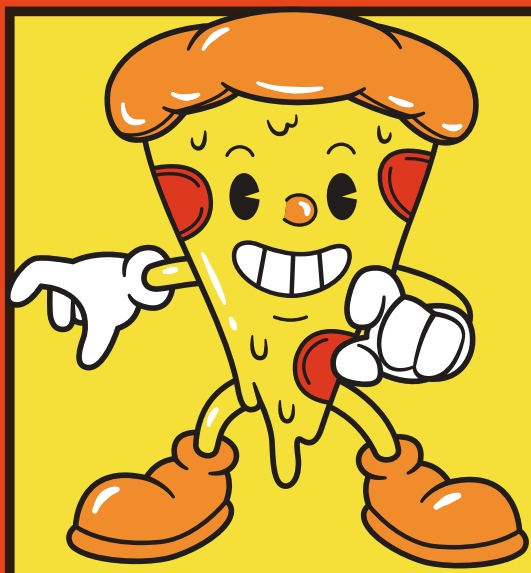
| Result Grid | | | Filter Rows: |
|-------------|-----------------|-------|--------------|
| | name | price | |
| ▶ | The Greek Pizza | 35.95 | |

Identify the most common pizza size ordered.

```
2 • SELECT
3     pizzas.size,
4     COUNT(orders_details.order_details_id) AS order_count
5 FROM
6     pizzas
7     JOIN
8     orders_details ON pizzas.pizza_id = orders_details.pizza_id
9 GROUP BY pizzas.size
10 ORDER BY order_count DESC
11 LIMIT 1;
```



| < | | |
|--------------|------|-------------|
| Result Grid | | |
| Filter Rows: | | |
| | size | order_count |
| ▶ | L | 18526 |



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

```
2 • SELECT
3     pizza_types.name, SUM(orders_details.quantity) AS quantity
4 FROM
5     pizza_types
6     JOIN
7     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8     JOIN
9     orders_details ON orders_details.pizza_id = pizzas.pizza_id
10 GROUP BY pizza_types.name
11 ORDER BY quantity DESC
12 LIMIT 5;
```

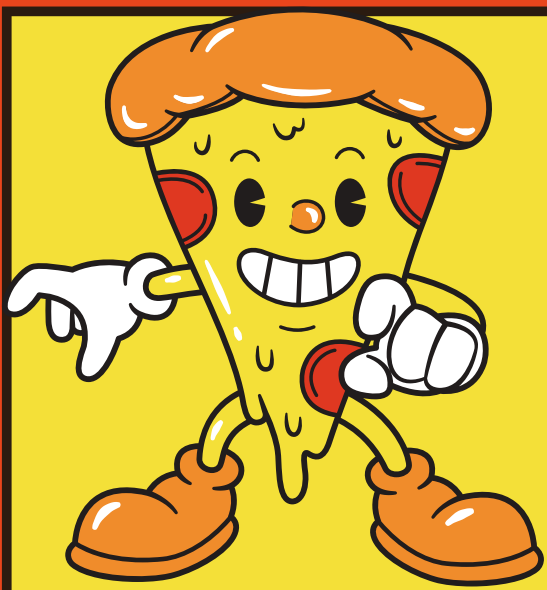
| Result Grid | | | Filter Rows: |
|-------------|----------------------------|----------|--------------|
| | name | quantity | |
| ▶ | The Classic Deluxe Pizza | 2453 | |
| | The Barbecue Chicken Pizza | 2432 | |
| | The Hawaiian Pizza | 2422 | |
| | The Pepperoni Pizza | 2418 | |
| | The Thai Chicken Pizza | 2371 | |



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

```
2 • SELECT
3     pizza_types.category,
4     SUM(orders_details.quantity) AS quantity
5 FROM
6     pizza_types
7     JOIN
8     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9     JOIN
10    orders_details ON orders_details.pizza_id = pizzas.pizza_id
11 GROUP BY pizza_types.category
12 ORDER BY quantity DESC;
```

| Result Grid | | | Filter Rows: |
|-------------|----------|----------|--------------|
| | category | quantity | |
| ▶ | Classic | 14888 | |
| | Supreme | 11987 | |
| | Veggie | 11649 | |
| | Chicken | 11050 | |



Determine the distribution of orders by hour of the day.

```
1 Determine the distribution of orders by hour of the day.
2 • SELECT
3     HOUR(order_time), COUNT(order_id)
4 FROM
5     orders
6 GROUP BY HOUR(order_time)
7
```

| Result Grid | | Filter Rows: |
|-------------|------------------|-----------------|
| | HOUR(order_time) | COUNT(order_id) |
| | 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 |

Result 1 ×

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

```
1  -- Join relevant tables to find the category-wise distribution of pizzas.  
2 • select category, count(name) from pizza_types  
3  group by category
```

| Result Grid | | | Filter Rows: |
|-------------|----------|-------------|--------------|
| | category | count(name) | |
| ▶ | Chicken | 6 | |
| | Classic | 8 | |
| | Supreme | 9 | |
| | Veggie | 9 | |

Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```
2 • select name, revenu from
3 (select category, name, revenu,
4 rank()over(partition by category order by revenu desc)as rn
5 from
6 (select pizza_types.category, pizza_types.name,
7 sum((orders_details.quantity)* pizzas.price)as revenu
8 from pizza_types join pizzas
9 on pizza_types.pizza_type_id=pizzas.pizza_type_id
10 join orders_details
11 on orders_details.pizza_id=pizzas.pizza_id
12 group by pizza_types.category, pizza_types.name)as a) as b
13 where rn<3 limit 3;
```

| Result Grid | | | Filter Rows: |
|-------------|----------------------------|----------|--------------|
| | name | revenu | |
| ▶ | The Thai Chicken Pizza | 43434.25 | |
| | The Barbecue Chicken Pizza | 42768 | |
| | The Classic Deluxe Pizza | 38180.5 | |

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

```
1  -- Calculate the percentage contribution of each pizza type to total revenue.
2  • select pizza_types.category,
3     round(sum(orders_details.quantity * pizzas.price) / (SELECT
4     ROUND(SUM(orders_details.quantity * pizzas.price),
5     2) AS total_sales
6  FROM
7     orders_details
8     JOIN
9     pizzas ON pizzas.pizza_id = orders_details.pizza_id), 4)*100 as revenu
10 from pizza_types join pizzas
11 on pizza_types.pizza_type_id=pizzas.pizza_type_id
12 join orders_details
13 on orders_details.pizza_id=pizzas.pizza_id
14 group by pizza_types.category order by revenu desc;
```

| Result Grid | | | Filter Rows: |
|-------------|----------|--------|--------------|
| | category | revenu | |
| ▶ | Classic | 26.91 | |
| | Supreme | 25.46 | |
| | Chicken | 23.96 | |
| | Veggie | 23.68 | |

Result 1 ✕

CONCLUSION

The pizza sales project using SQL has provided valuable insights into our sales performance, customer preferences, and operational efficiency. By leveraging SQL for data analysis, we were able to uncover significant trends and patterns that can drive strategic decision-making.

THANK YOU !

