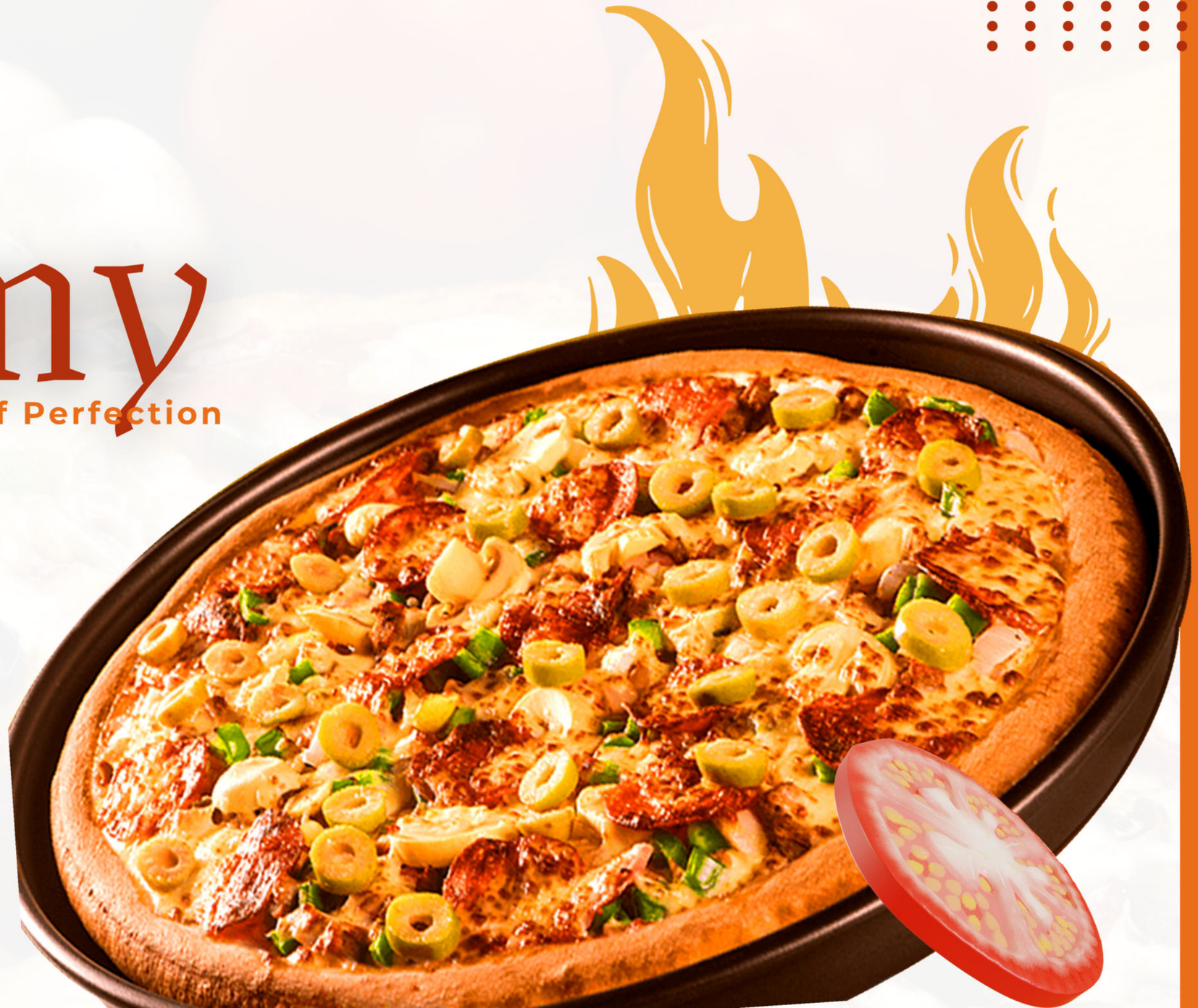
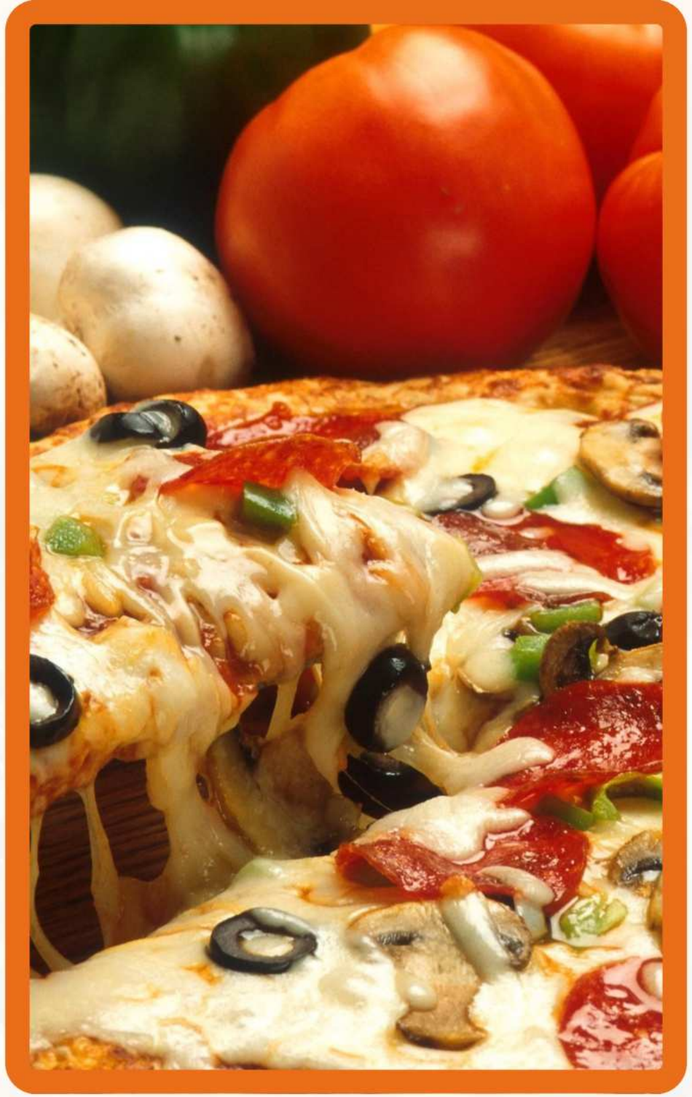
# Yummy Pizza

Where Every Slice is a Taste of Perfection



presented by
smruti

Unvesiling Insights from Pizza sales



# About Pizza

OBJECTIVE:

This Project aims to leverage the power of SQL Quries to unlock valuable insights from the Pizza Sales dataset. By crafting intricate joins and utilizing advanced analytical techniques, we will provide the necessary information to make informed strategic decisions across various aspects of the business.

# Key Questions :

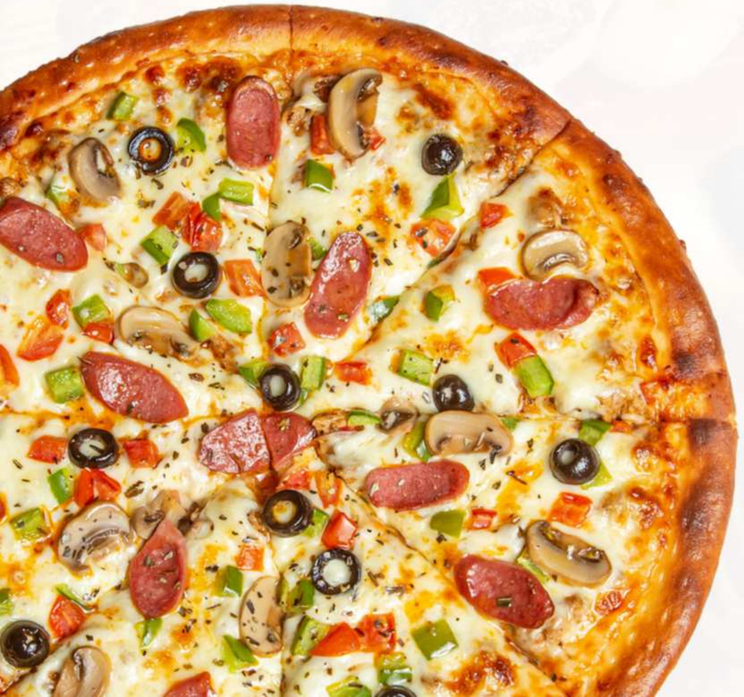
- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza with name.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

- Join the necessary tables to find the total quantity of each pizza category ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the category wise distribution of pizzas.

- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.
- Analyze the cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.

# Retrieve the total number of orders placed.

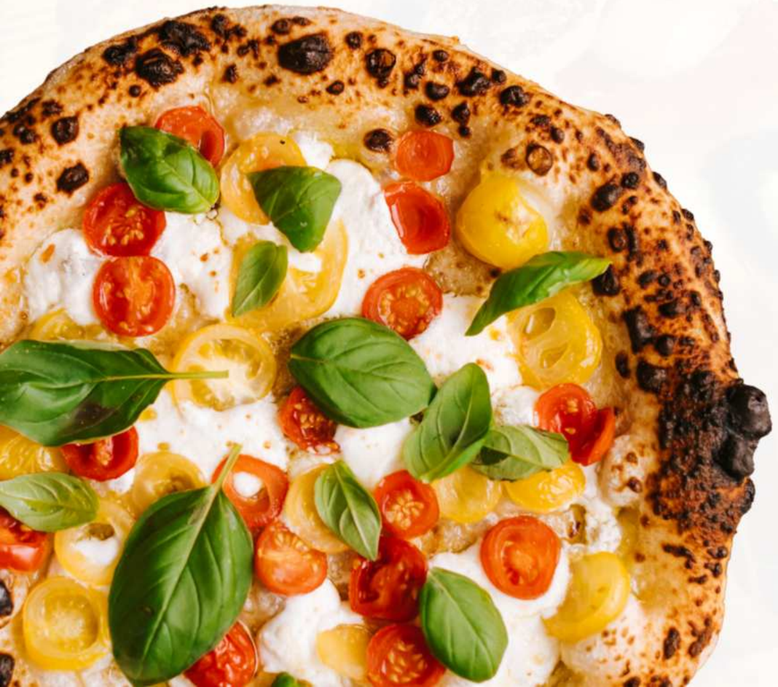
```
SELECT
    COUNT(order_id) AS total_ord
FROM
    orders;
```



| | total_ord |
|---|---|
| ▶ | 250 |

# Calculate the total revenue generated from pizza sales.

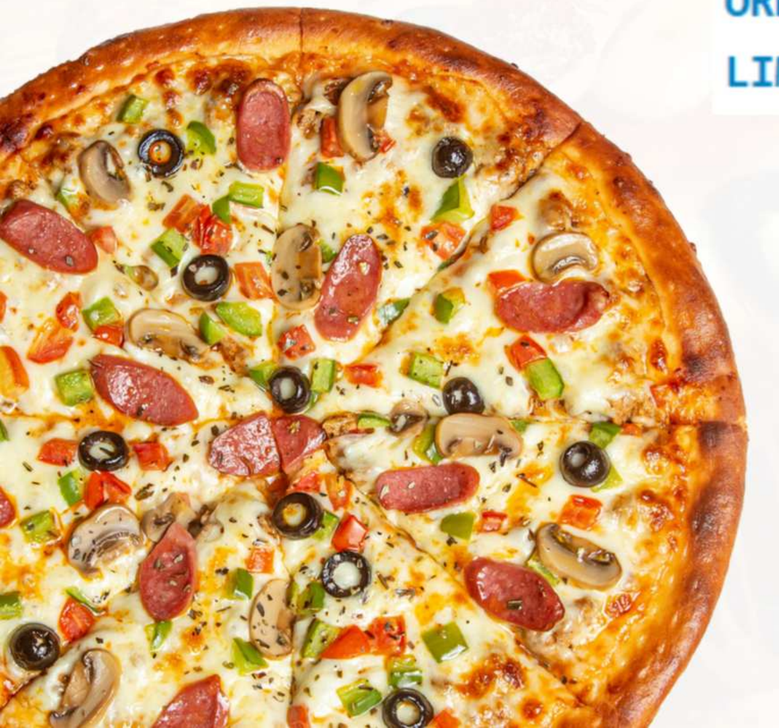
```sql
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id;
```



| | total_sales |
|---|---|
| ▶ | 9744.55 |

# Identify the highest-priced pizza with name.

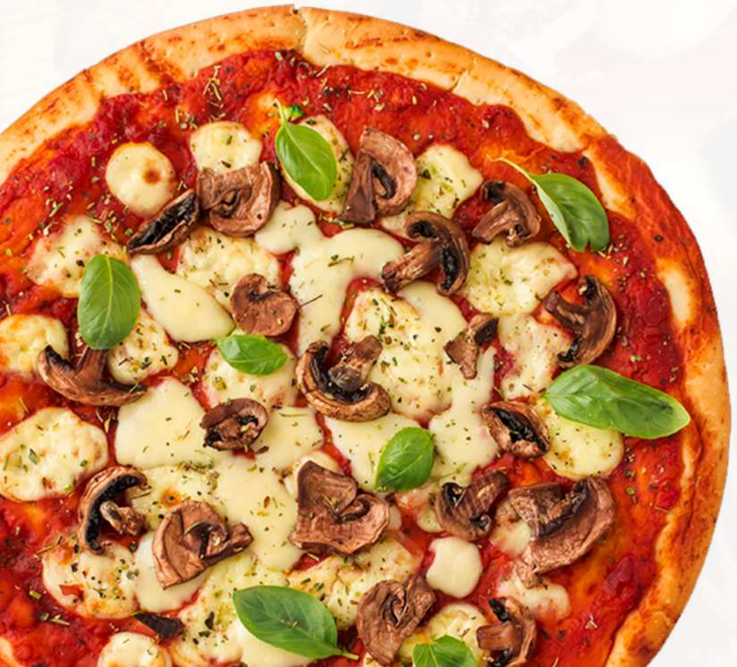
```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```



| name | price |
| --- | --- |
| The Greek Pizza | 35.95 |

# Identify the most common pizza size ordered.

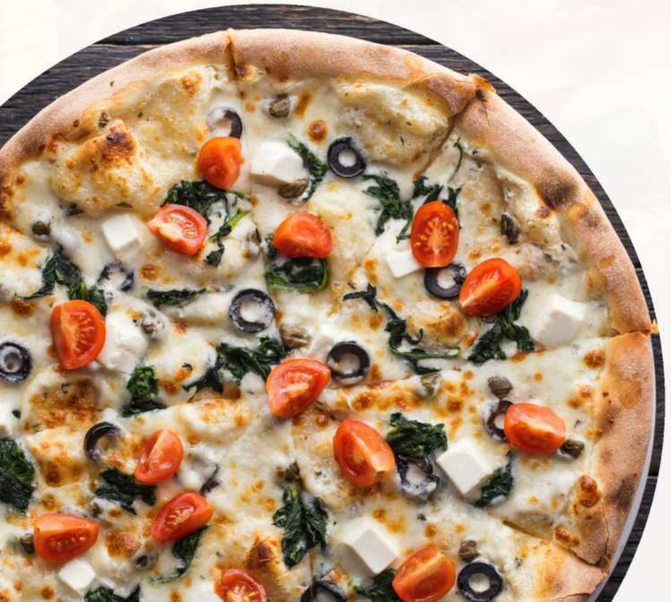
```sql
select pizzas.size,
count(pizza_types.pizza_type_id) as order_count
from pizzas join pizza_types
on pizzas.pizza_type_id = pizza_types.pizza_type_id
group by pizzas.size order by order_count desc;
```



| size | order_count |
|---|---|
| S | 32 |
| M | 31 |
| L | 31 |
| XL | 1 |
| XXL | 1 |

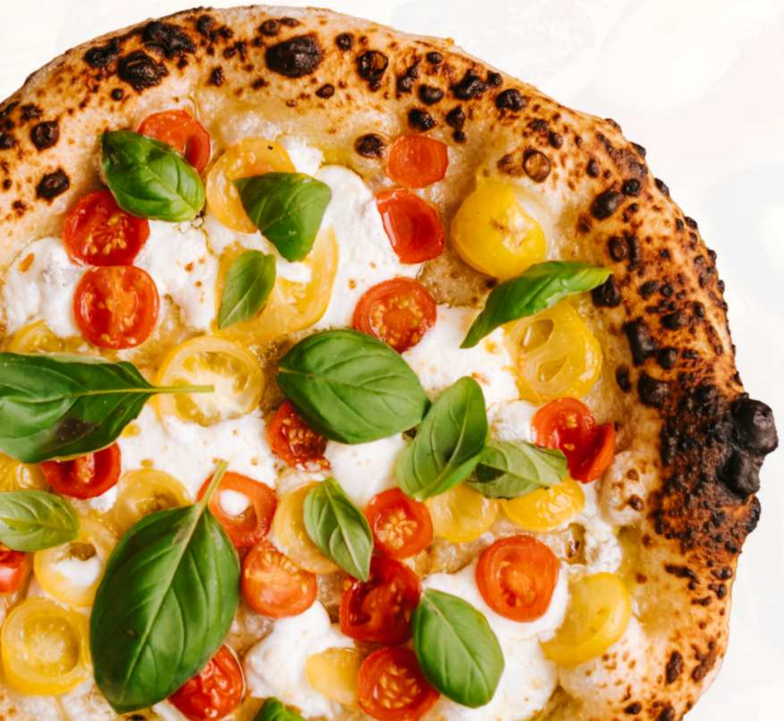
# List the top 5 most ordered pizza types along with their quantities.

```sql
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
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.name
ORDER BY quantity DESC
LIMIT 5;
```



| name | quantity |
|---|---|
| The Barbecue Chicken Pizza | 37 |
| The Italian Supreme Pizza | 32 |
| The Thai Chicken Pizza | 32 |
| The Pepperoni Pizza | 30 |
| The Hawaiian Pizza | 26 |

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

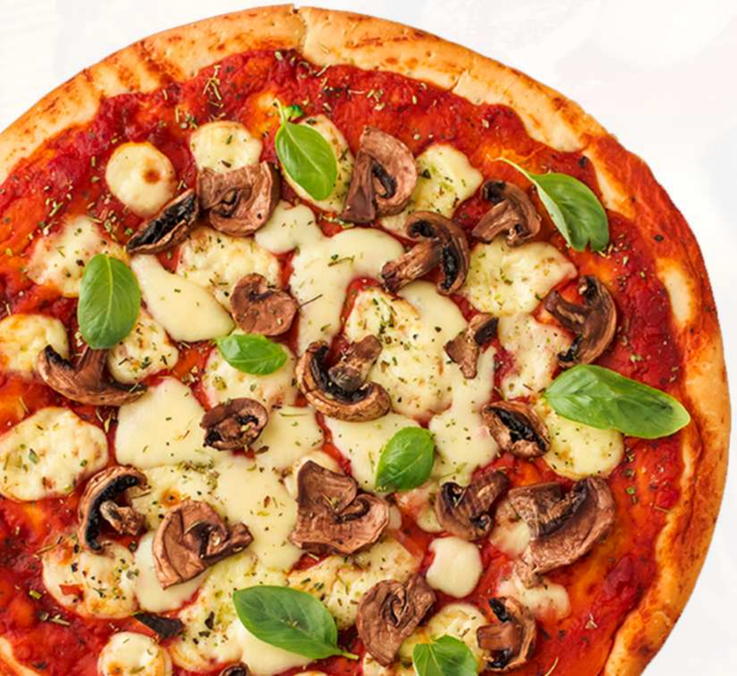


```sql
SELECT
    SUM(order_details.quantity) AS quantity,
    pizza_types.category
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.category
ORDER BY quantity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

| quantity | category |
|---|---|
| 176 | Classic |
| 141 | Supreme |
| 137 | Chicken |
| 131 | Veggie |

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

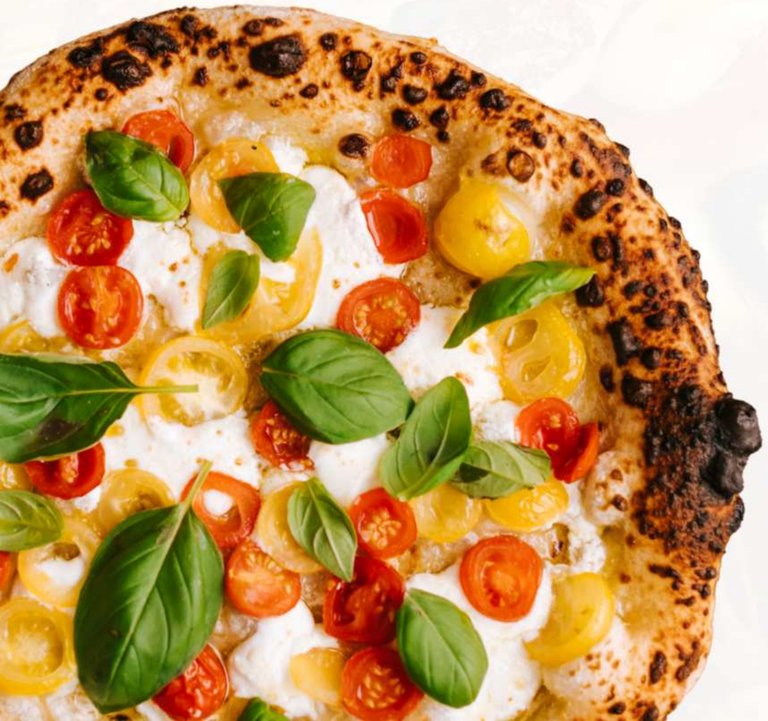
```sql
select hour(time) as hour, count(order_id)as order_count
from orders group by hour(time);
```



| hour | order_count |
|---|---|
| 11 | 13 |
| 12 | 23 |
| 13 | 22 |
| 14 | 19 |
| 15 | 23 |
| 16 | 19 |
| 17 | 28 |
| 18 | 32 |
| 19 | 21 |
| 20 | 25 |
| 21 | 14 |
| 22 | 11 |

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

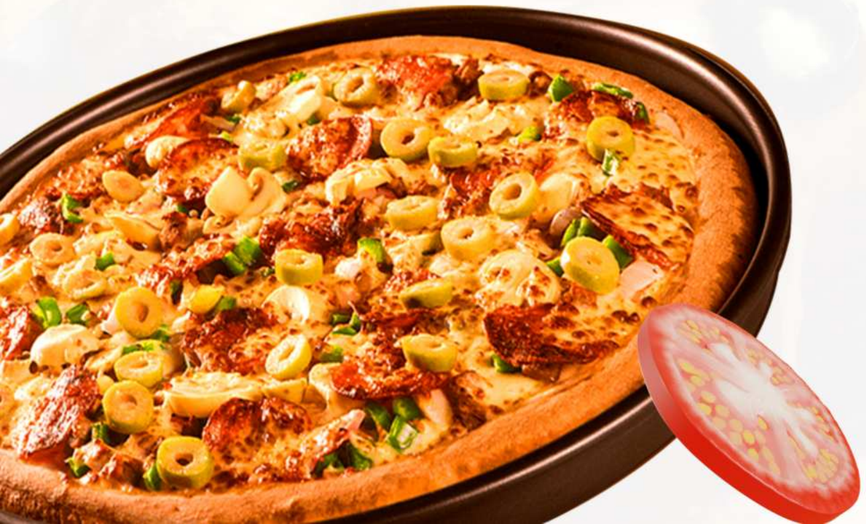
```
select category, count(name)as count
 from pizza_types group by category;
```



| category | count |
|---|---|
| Chicken | 6 |
| Classic | 8 |
| Supreme | 9 |
| Veggie | 9 |

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

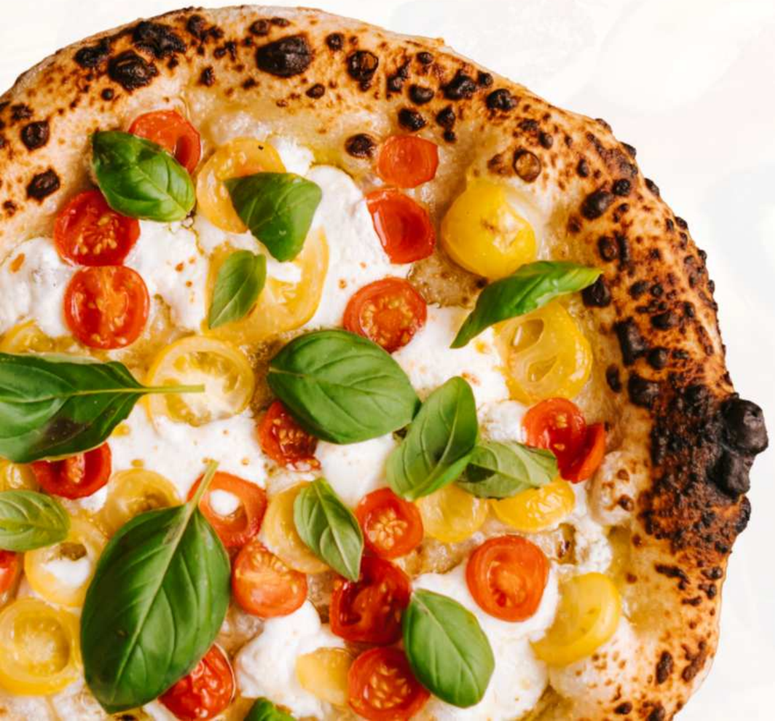
```sql
SELECT
    ROUND(AVG(quantity), 0) AS avg_no_pizzas
FROM
    (SELECT
        orders.date, SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.date) AS order_quantity;
```



| avg_no_pizzas |
| --- |
| 146 |

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

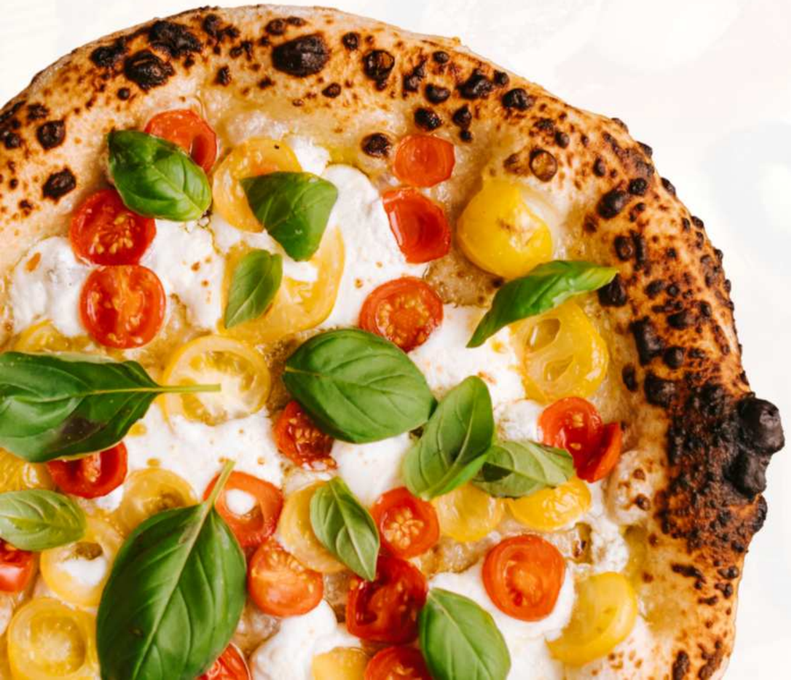
```sql
SELECT
    pizza_types.name,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```



| name | revenue |
|---|---|
| The Barbecue Chicken Pizza | 667.75 |
| The Thai Chicken Pizza | 588 |
| The Italian Supreme Pizza | 571.5 |

# Analyze the cumulative revenue generated over time.

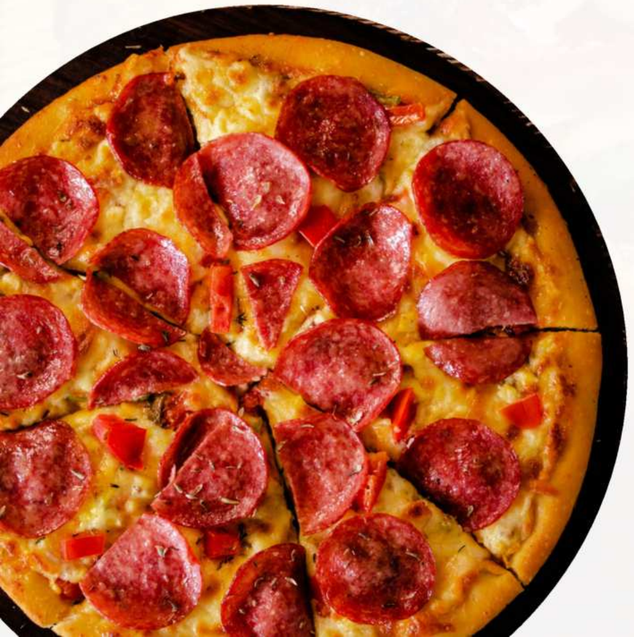
```
select date,sum(revenue) over(order by date)as cum_rev from (select orders.date,
sum(order_details.quantity * pizzas.price)as revenue
from order_details join pizzas
on order_details.pizza_id  = pizzas.pizza_id
join orders on orders.order_id = order_details.order_id
group by orders.date)as sales;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



| date | cum_rev |
|---|---|
| 01-01-2015 | 2713.8500 |
| 02-01-2015 | 5445.75 |
| 03-01-2015 | 8108.15 |
| 04-01-2015 | 9744.55 |

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

```sql
select name,revenue 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((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 rnk<=3;
```



| name | revenue |
|---|---|
| The Barbecue Chicken Pizza | 667.75 |
| The Thai Chicken Pizza | 588 |
| The California Chicken Pizza | 475.5 |
| The Classic Deluxe Pizza | 407.5 |
| The Greek Pizza | 389 |
| The Pepperoni Pizza | 377.75 |
| The Italian Supreme Pizza | 571.5 |
| The Spicy Italian Pizza | 493.5 |
| The Sicilian Pizza | 329 |
| The Five Cheese Pizza | 370 |
| The Mexicana Pizza | 342.75 |
| The Four Cheese Pizza | 333.39... |

Pizza Sales Presentation

# THANK YOU FOR ATTENTION