



# Pizza Sales Analysis

## Using SQL



# Special chesse

-- Retrieve the total number of orders placed?  
use **pizzahut;**  
**SELECT**  
**COUNT(order\_id) AS total\_orders**  
**FROM**  
**pizzahut.orders;**

DISCOUNT UP TO 15% OFF  
DISCOUNT UP TO 15%



*Delicious*

# PIZZA

-- Calculate the total revenue generated from pizza sales.?

SELECT

ROUND(SUM(e.quantity \* p.price),

2) AS total\_revenue

FROM

order\_details e

LEFT JOIN

pizzas p ON e.pizza\_id = p.pizza\_id;

-- Identify the highest price pizza.?

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

-- Identify the most common pizza size ordered?

```
SELECT
    e.size, COUNT(f.order_id) AS order_count
    FROM
        pizzas e
    JOIN
        order_details f ON e.pizza_id = f.pizza_id
    GROUP BY e.size
    ORDER BY order_count DESC
    LIMIT 1;
```

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

```
SELECT
e.name, SUM(g.quantity) AS pizza_quantity
FROM
pizza_types e
JOIN
pizzas f ON e.pizza_type_id = f.pizza_type_id
JOIN
order_details g ON g.pizza_id = f.pizza_id
GROUP BY e.name
ORDER BY pizza_quantity DESC
LIMIT 5;
```

-- Find the total quantity of each pizza category ordered?

```
SELECT
    e.category, SUM(g.quantity) AS sum_quantity
    FROM
        pizza_types e
        JOIN
            pizzas f ON e.pizza_type_id = f.pizza_type_id
        JOIN
            order_details g ON g.pizza_id = f.pizza_id
    GROUP BY e.category;
```

-- Determine the distribution of the orders by the hour  
of the day?

```
SELECT  
    HOUR(order_time) AS hour, COUNT(order_id) AS  
        order_count  
    FROM  
        orders  
    GROUP BY HOUR(order_time);
```

-- Find category wise distribution of pizzas?

```
SELECT  
    category, COUNT(category)  
    FROM  
        pizza_types  
    GROUP BY category;
```

-- Group the orders by date and calculate the average number of pizzas ordered per day?

```
SELECT
    round(avg(pizzas_quantity),0) AS
        avg_pizzas_ordered
    FROM
        (SELECT
            e.order_date, SUM(f.quantity) AS pizzas_quantity
            FROM
                orders e
            JOIN order_details f ON e.order_id = f.order_id
            GROUP BY e.order_date) AS order_quantity;
```

-- Determine the top 3 most ordered pizza types based on revenue?

```
SELECT
e.name, SUM(f.price * g.quantity) AS pizza_revenue
FROM
pizza_types e
JOIN
pizzas f ON e.pizza_type_id = f.pizza_type_id
UP TO
JOIN
order_details g ON g.pizza_id = f.pizza_id
OFF
GROUP BY e.name
ORDER BY pizza_revenue DESC
LIMIT 3;
```

-- Calculate the percentage contribution of each pizza type to total revenue?

```
SELECT
    e.category,
    round(SUM(f.price * g.quantity) / (SELECT
        SUM(f.price * g.quantity)
    FROM
        order_details g
    JOIN
        pizzas f ON f.pizza_id = g.pizza_id) * 100,2) AS
    total_sales
FROM
    pizza_types e
JOIN
    pizzas f ON e.pizza_type_id = f.pizza_type_id
JOIN
    order_details g ON g.pizza_id = f.pizza_id
GROUP BY e.category
ORDER BY total_sales;
```

-- Analyse the cumulated revenue generated over time?

```
SELECT  
order_date,SUM(revenue) OVER(ORDER BY  
order_date) AS cum_revenue  
FROM  
(SELECT  
e.order_date,SUM(f.quantity*g.price) AS  
revenue  
FROM  
orders e  
JOIN  
order_details f ON e.order_id=f.order_id  
JOIN  
pizzas g on f.pizza_id=g.pizza_id  
GROUP BY e.order_date) AS sales;
```

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

```
SELECT name,revenue
      FROM
        (SELECT category,name,revenue,
RANK() OVER(PARTITION BY category ORDER BY
revenue DESC) AS rn
      FROM
(SELECT e.category,e.name,SUM(g.quantity*f.price)
      AS revenue
      FROM
pizza_types e
      JOIN
pizzas f ON e.pizza_type_id=f.pizza_type_id
      JOIN
order_details g ON g.pizza_id=f.pizza_id
GROUP BY e.category,e.name) as a) as b
      WHERE rn<=3;
```