

```
SELECT * FROM pizzahunt.pizzas;
```

```
SELECT * FROM pizzahunt.order_details;
```

```
SELECT * FROM pizzahunt.orders;
```

```
SELECT * FROM pizzahunt.pizza_types;
```

```
SELECT * FROM orders;
```

```
SELECT count(order_id) from orders;
```

```
-- --retrieve the total number of order place—
```

```
SELECT count(order_id) as total_orders from orders;
```

```
-- count the total revenue generated from pizza sales-----
```

```
SELECT
```

```
    ROUND(SUM(order_details.quantity * pizzas.price), 2) AS total_sales
```

```
FROM
```

```
    order_details
```

```
JOIN
```

```
    pizzas
```

```
ON
```

```
    pizzas.pizza_id = order_details.pizza_id;
```

```
-- Find the highest pizza price -----
```

```
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;
```

-- Find the most common pizza is ordered

```
SELECT
    quantity, COUNT(order_details_id)
FROM
    order_details
GROUP BY quantity;
```

```
SELECT
    pizzas.size,
    COUNT(order_details.order_details_id) AS order_count
FROM
    pizzas
    JOIN
        order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC;
```

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

```
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;
```

-- join the curial tables to find the total quantity of each pizza category ordered

SELECT

 pizza_types.category,

 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.category

ORDER BY quantity DESC;

-- Find the distribution of orders by hours of the day

SELECT

 HOUR(order_time) AS hour, COUNT(order_id) AS order_count

FROM

 orders

GROUP BY HOUR(order_time);

-- Join relevant table to find the category wise distribution of pizzas

select category , count(name) from pizza_types

group by category;

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

SELECT

 orders.order_date, SUM(order_details.quantity)

FROM

 orders

JOIN

order_details ON orders.order_id = order_details.order_id

GROUP BY orders.order_date;

-- find the top 3 most order pizza type based on the revenue

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;

-- calculate the perentage contribution of each pizza type to total revenu.....

SELECT pizza_types.name,

SUM(order_details.quantity * pizzas.price) AS revenue,

(SUM(order_details.quantity * pizzas.price) / total_revenue.total * 100) AS
percentage_contribution

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

JOIN (SELECT SUM(order_details.quantity * pizzas.price) AS total

FROM pizzas

JOIN order_details ON order_details.pizza_id = pizzas.pizza_id) AS total_revenue

GROUP BY pizza_types.name, total_revenue.total

ORDER BY revenue DESC;

-- analyze the cumulative revenue generated over time

```
SELECT order_date,
       revenue,
       SUM(revenue) OVER (ORDER BY order_date) AS cum_revenue
FROM (
  SELECT orders.order_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.order_date
) AS sales
ORDER BY order_date;
```

-- find the top 3 most order pizza type based on revenue for each pizza category

```
select 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;
```