

Basic:

Q1 - Retrieve the total number of orders placed.

```
SELECT count(order_id) as Total_num_order From orders
```

Q2 - Calculate the total revenue generated from pizza sales.

```
SELECT  
ROUND(SUM(pizzas.price * order_details.quantity)) AS Total_Revenue  
FROM  
order_details  
JOIN  
pizzas ON pizzas.pizza_id = order_details.pizza_id
```

Q3 - Identify the highest-priced pizza.

```
SELECT pizza_type.name, pizzas.price as highest_price_pizza  
from pizza_type  
JOIN pizzas  
ON pizzas.pizza_type_id = pizza_type.pizza_type_id  
order by pizzas.price DESC  
limit 2
```

Q4 - Identify the most common pizza size ordered.

```
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
```

Q5 - List the top 5 most ordered pizza types along with their quantities.

```
select pizza_type.name , sum(order_details.quantity) as quantity
from pizza_type
Join pizzas on pizzas.pizza_type_id = pizza_type.pizza_type_id
Join order_details ON order_details.pizza_id = pizzas.pizza_id
group by pizza_type.name
order by quantity DESC
limit 5
```

Intermediate:

Q6 - Join the necessary tables to find the total quantity of each pizza category ordered.

```
select pizza_type.category , sum(order_details.quantity) as quantity
from pizza_type
join pizzas ON pizza_type.pizza_type_id = pizzas.pizza_type_id
join order_details ON order_details.pizza_id = pizzas.pizza_id
group by pizza_type.category
order by quantity DESC
limit 5
```

Q7 - Determine the distribution of orders by hour of the day.

```
select HOUR(time) as order_hour, count(order_id)as order_count
from orders
group by HOUR(time)
```

Q8 - Join relevant tables to find the category-wise distribution of pizzas.

```
select category , count(name) from pizza_type
group by category
```

Q9 - Group the orders by date and calculate the average number of pizzas ordered per day.

```
select round(Avg(quantity)) as average_quantity 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
```

Q10 - Determine the top 3 most ordered pizza types based on revenue.

```
select pizza_type.name , round(sum(order_details.quantity * pizzas.price))
as revenue from pizza_type
join pizzas ON pizza_type.pizza_type_id = pizzas.pizza_type_id
JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
group by pizza_type.name
order by revenue desc
limit 5
```

Advanced:

Q11- Calculate the percentage contribution of each pizza type to total revenue.

USE pizzahut;

```
select pizza_type.category,  
round(sum(pizzas.price * order_details.quantity)/  
(select round(sum(order_details.quantity * pizzas.price),2) as total_dales  
from order_details  
JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id) *100 ,2)as  
revenue  
from pizza_type  
JOIN pizzas ON pizza_type.pizza_type_id = pizzas.pizza_type_id  
JOIN order_details ON order_details.pizza_id = pizzas.pizza_id  
group by pizza_type.category
```

Q12 - Analyze the cumulative revenue generated over time.

```

select order_date, sum(total_revenue) over(order by order_date) as
com_revenue
from
(select orders.date as order_date,
round(sum(order_details.quantity * pizzas.price))as total_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

```

Q13 - Determine the top 3 most ordered pizza types 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_type.category, pizza_type.name,sum(order_details.quantity * pizzas.price) as
revenue
from pizza_type
JOIN pizzas ON pizza_type.pizza_type_id = pizzas.pizza_type_id
JOIN order_details ON order_details.pizza_id = pizzas.pizza_id
group by pizza_type.category, pizza_type.name) as a) as b
where rn <= 3

```