

```
1  /*Basic:
2  Retrieve the total number of orders placed.
3  Calculate the total revenue generated from pizza sales.
4  Identify the highest-priced pizza.
5  Identify the most common pizza size ordered.
6  List the top 5 most ordered pizza types along with their quantities.*/
7  create database pizzahut;
8  select* from order_details
9  select count(order_id) as "total-orders-placed" from orders
10 select * from orders
11 select* from pizza_types
12 select* from pizzas
13 select round(sum(o.quantity*p.price),2) as revenue_generated from      ↗
    order_details as o left join pizzas as p on o.pizza_id=p.pizza_id
14 select top 1 round( max(p.price),2) as highest_priced , pt.name from    ↗
    pizzas p left join pizza_types pt on p.pizza_type_id=pt.pizza_type_id  ↗
    group by pt.name
15 order by max(p.price) desc
16 select quantity , count(order_details_id) from order_details group by  ↗
    quantity
17 select p.size , count(od.order_details_id)as totalpizzas from pizzas p  ↗
    join order_details od on p.pizza_id=od.pizza_id group by p.size order by ↗
    totalpizzas desc
18 select top 5 sum(o.quantity) as mostorderedpizzas, p.pizza_type_id from  ↗
    pizzas p join order_details o on p.pizza_id=o.pizza_id group by        ↗
    p.pizza_type_id
19 order by sum(o.quantity) desc
20
----- ↗
----- ↗
----- ↗
21 /*Intermediate:
22 Join the necessary tables to find the total quantity of each pizza category ↗
    ordered.
23 Determine the distribution of orders by hour of the day.
24 Join relevant tables to find the category-wise distribution of pizzas.
25 Group the orders by date and calculate the average number of pizzas ordered ↗
    per day.
26 Determine the top 3 most ordered pizza types based on revenue.*/
27 select p.category,sum(o.quantity) as totalquantity from pizza_types p join ↗
    pizzas on p.pizza_type_id=pizzas.pizza_type_id join order_details o
28 on o.pizza_id=pizzas.pizza_id group by p.category order by sum(o.quantity) ↗
    desc
29 select DATEPART(hour, time) as hour, sum(order_id)as totalorders from    ↗
    orders group by time order by time asc
30 select category, count(name) orders from pizza_types group by category
31 select avg(quantity)as average_per_day_order from (select o.date,sum      ↗
    (od.quantity)quantity from orders o join order_details od on            ↗
    o.order_id=od.order_id group by o.date) as order_quantity
32 select top 3 sum(order_details.quantity*pizzas.price) as revenue ,      ↗
    pizza_types.name from pizzas join order_details on                    ↗
    pizzas.pizza_id=order_details.pizza_id
33 join pizza_types on pizzas.pizza_type_id=pizza_types.pizza_type_id
```

```

34 group by pizza_types.name order by revenue desc
35 -----
36 -----
37 /* Advanced
38 Calculate the percentage contribution of each pizza type to total revenue.
39 Analyze the cumulative revenue generated over time.
40 Determine the top 3 most ordered pizza types based on revenue for each
   pizza category.*/
41 select pt.category, round(sum(od.quantity*p.price)/ (select round(sum
   (od.quantity*p.price),2) as revenue_generated
42 from order_details as od left join pizzas as p on od.pizza_id=p.pizza_id)
   *100,2)
43 as revenue_percent from pizzas p join order_details od on
   p.pizza_id=od.pizza_id
44 join pizza_types pt on p.pizza_type_id=pt.pizza_type_id
45 group by pt.category order by revenue_percent
46 select date, sum(revenue_generated) over(order by date) as
   cumulative_revenue from
47 (select orders.date, round(sum(od.quantity*p.price),2) as revenue_generated
   from
48 order_details as od left join pizzas as p on od.pizza_id=p.pizza_id
49 left join orders on orders.order_id=od.order_id group by orders.date) as
   sales ;
50 select name , category, revenue
51 from
52 (select category, name, revenue , rank() over ( partition by category order
   by revenue desc) as rn
53 from
54 (select pt.category , pt.name, sum(p.price*od.quantity) as revenue from
   pizzas p join order_details od
55 on p.pizza_id=od.pizza_id
56 join pizza_types pt on p.pizza_type_id=pt.pizza_type_id
57 group by pt.category, pt.name ) as a ) b
58 where rn<=3 ;
59 ----- PROJECT
60 -----
61 -----

```