

## PIZZA SALES PROJECT USING MYSQL SOLUTIONS

Select \* from orders

select \* from order\_details

select \* from pizzas

select \* from pizza\_types

### **BASIC LEVEL: -**

1) Retrieve the total number of orders placed

- Select count(order\_id) as Total\_Orders  
from orders;

2) Calculate the total revenue generated from pizza sales.

- Select  
round(sum(order\_details.quantity\*pizzas.price),2) as total\_revenue  
from order\_details  
join pizzas  
on pizzas.pizza\_id=order\_details.pizza\_id;

3)Identify the highest-priced pizza.

- 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;

4)Identify the most common pizza size ordered.

- select quantity,count(order\_details\_id) as total\_orders  
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;

5)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;

### **INTERMEDIATE LEVEL: -**

1)Join the necessary 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;

2) Determine the distribution of orders by hour of the day.

- ```
select hour(order_time) as hours, count(order_id) as order_count
from orders
group by hours
order by hours;
```

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

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

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

- ```
select round(avg(quantity),0) as avg_pizza_order_per_day
from ( select orders.order_date, sum(order_details.quantity) as quantity
      from orders
      join order_details
      on orders.order_id=order_details.order_id
      group by orders.order_date ) as order_quantity;
```

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

- ```
select pizza_types.name, round(sum(order_details.quantity * pizzas.price),0) 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;
```

## **ADVANCE LEVEL: -**

1) Calculate the percentage contribution of each pizza type to total revenue.

- ```
select pizza_types.category, round(sum(order_details.quantity * pizzas.price)/
(
    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) * 100,2) 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.category
order by revenue desc
```

2) Analyze the cumulative revenue generated over time.

- ```
select order_date,
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;
```

3) 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\_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

limit 3;