



sales Pizza

Hey, my name is Harshit Tiwari. In this project. I have utilise sql queries to solve question related to pizza sales.

A photograph of a pizza on a round wooden board. The pizza is cut into eight slices and topped with melted cheese, ham, and pepperoni. A hand wearing a blue and white striped shirt is visible at the top left, holding a folded napkin.

*Retrieve the total number of
orders placed*

```
SELECT  
COUNT(order_id) AS  
total_orders  
FROM  
orders;
```

*RetrieveCalculate the total revenue
generated from pizza sales. the total
number of orders placed*



```
SELECT  
    ROUND(SUM((orders_details.  
        quantity * pizzas.price)),  
        2) AS total_sales  
FROM  
    orders_details  
    JOIN  
        pizzas ON  
    pizzas.pizza_id =  
    orders_details.pizza_id;
```

A photograph of a pizza on a round wooden board. The pizza is cut into eight slices and topped with melted cheese, ham, and pepperoni. A hand wearing a blue and white striped cuff is visible at the top left, holding a folded napkin.

*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;
```

A photograph of a pizza on a wooden board. A hand is visible at the top left, holding a blue and white striped napkin. The pizza is cut into slices and topped with cheese, ham, and pepperoni.

*Identify the most common
pizza size ordered.*

```
SELECT
  pizzas.size,
  COUNT(orders_details.orders_details_id) AS
  total_order
FROM
  pizzas
JOIN
  orders_details ON
  pizzas.pizza_id =
  orders_details.pizza_id
GROUP BY pizzas.size
ORDER BY total_order DESC
LIMIT 1;
```

A photograph of a pizza on a wooden board. A hand-drawn white outline of a heart surrounds one slice of the pizza. The pizza has various toppings like cheese, ham, and pepperoni. A blue napkin is visible in the top left corner.

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

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

A photograph of a pizza on a round wooden board. The pizza is cut into eight slices and topped with melted cheese, ham, and pepperoni. A hand wearing a blue and white striped shirt is visible at the top left, holding a folded napkin.

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

```
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity
    FROM
    pizza_types
    JOIN
    pizzas ON
    pizza_types.pizza_type_id =
    pizzas.pizza_type_id
    JOIN
    orders_details ON
    orders_details.pizza_id =
    pizzas.pizza_id
    GROUP BY
    pizza_types.category
    ORDER BY quantity desc;
```

A photograph of a pizza on a wooden board. A hand wearing a blue and white striped shirt is visible at the top left, holding a folded napkin. The pizza is cut into slices and topped with cheese, ham, and pepperoni.

Determine the distribution of
orders by hour of the day.

```
SELECT  
    HOUR(order_time) AS  
hours, COUNT(order_id) AS  
count_order  
FROM  
    orders  
GROUP BY  
    HOUR(order_time);
```

A photograph of a pizza on a round wooden board. The pizza is cut into several slices and topped with melted cheese, ham, and pepperoni. A hand wearing a blue and white striped shirt is visible at the top left, holding a folded napkin.

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

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

A photograph of a pizza on a wooden board. A hand is visible at the top left, holding a blue and white striped napkin. The pizza is cut into slices and topped with cheese, ham, and pepperoni.

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

```
SELECT
    ROUND(AVG(quantity), 0)
    AS average_pizza_order
    FROM
        (SELECT
            orders.order_date,
            SUM(orders_details.quantity) AS quantity
        FROM
            orders
        JOIN orders_details ON
            orders.order_id =
            orders_details.order_id
        GROUP BY
            orders.order_date) AS
            order_quantity ;
```

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



```
SELECT
  pizza_types.name,
  SUM(orders_details.quantity * pizzas.price) AS revenue
  FROM
  pizza_types
  JOIN
  pizzas ON
  pizza_types.pizza_type_id =
  pizzas.pizza_type_id
  JOIN
  orders_details ON
  orders_details.pizza_id =
  pizzas.pizza_id
  GROUP BY
  pizza_types.name
  ORDER BY revenue DESC
  LIMIT 3;
```

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



```
SELECT
    pizza_types.category,
    ROUND(SUM(orders_details.quantity *
    pizzas.price) / (SELECT
        ROUND(SUM((orders_details.quantity *
        pizzas.price)),
        2) AS total_sales
    FROM
        orders_details
    JOIN
        pizzas ON pizzas.pizza_id =
        orders_details.pizza_id) * 100,
    2) AS revenue
    FROM
        pizza_types
    JOIN
        pizzas ON pizza_types.pizza_type_id =
        pizzas.pizza_type_id
    JOIN
        orders_details ON orders_details.pizza_id =
        pizzas.pizza_id
    GROUP BY pizza_types.category
    ORDER BY revenue DESC;
```

Analyze the cumulative revenue generated
over time.

```
select order_date,  
sum(revenue)over(order by order_date) as  
cummu_revenue  
from  
(select orders.order_date,  
sum(orders_details.quantity * pizzas.price) as  
revenue  
from orders_details join pizzas  
on orders_details.pizza_id = pizzas.pizza_id  
join orders  
on orders.order_id = orders_details.order_id  
group by orders.order_date) as sales;
```



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(orders_details.quantity *  
      pizzas.price) as revenue  
      from pizza_types join pizzas  
      on pizza_types.pizza_type_id =  
      pizzas.pizza_type_id  
      join orders_details  
      on orders_details.pizza_id =  
      pizzas.pizza_id  
      group by  
pizza_types.category,pizza_types.name )  
      as a ) as b  
      where rn<=3 ;
```



