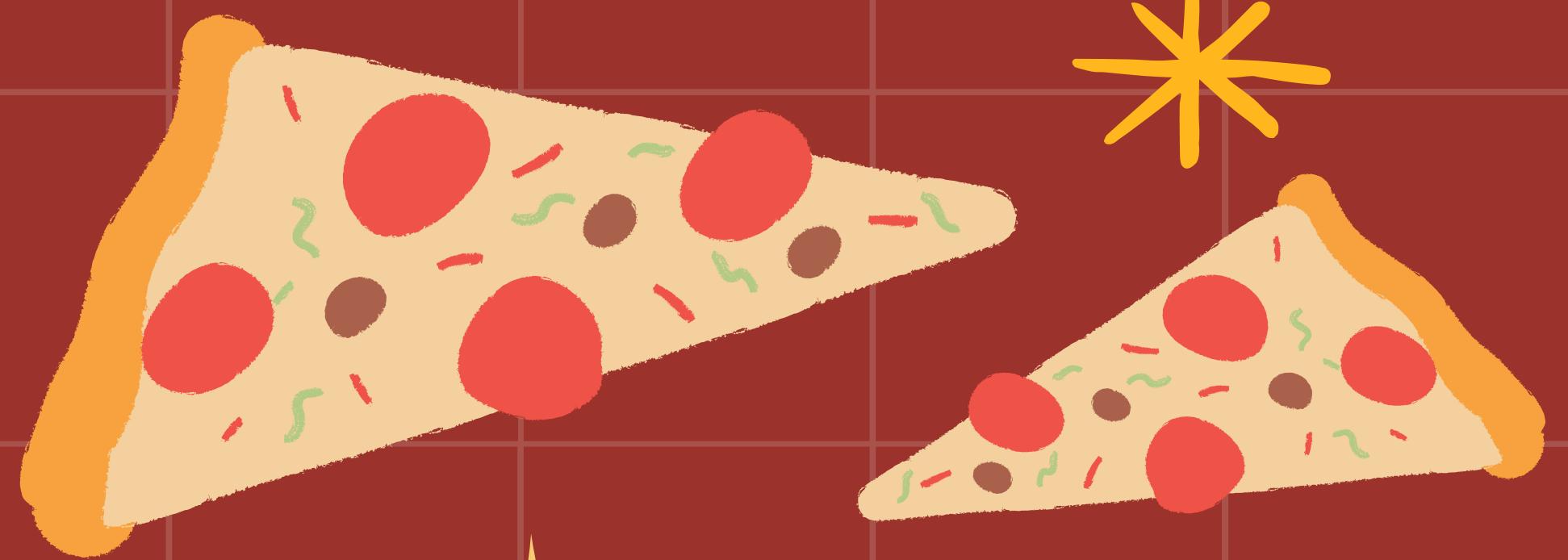


REPORT OF PIZZA SALE



A cartoon illustration of two children against a red grid background. On the left, a girl with dark hair and round glasses wears a yellow starburst-shaped cape and holds a slice of pizza. On the right, a boy with a green baseball cap and a yellow striped shirt also holds a slice of pizza. Both children are smiling.

LIST OF CONTENTS

Introduction

questions

summary

INTRODUCTION

This report analyzes pizza sales using SQL in MySQL Workbench, focusing on revenue, popular pizzas, and customer trends to optimize our offerings.

Let's start our adventure in the world of pizza!

QUESTIONS



- Retrieve the total number of orders placed.
- Calculate the total revenue generated from pizza sales.
- Identify the highest-priced pizza.
- Identify the most common pizza size ordered.
- List the top 5 most ordered pizza types along with their quantities.

- Join the necessary tables to find the total quantity of each pizza ordered.
- Determine the distribution of orders by hour of the day.
- Join relevant tables to find the Category wise distribution of pizzas.
- Group the orders by date and calculate the average number of pizzas ordered per day.
- Determine the top 3 most ordered pizza types based on revenue.

- Calculate the percentage contribution of each pizza type to total revenue.
- Analyze the Cumulative revenue generated over time.
- Determine the top 3 most ordered pizza types based on revenue for each pizza category.



RETRIEVE THE TOTAL NUMBER
OF ORDERS PLACED.

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



CALCULATE 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
```



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;
```

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  
LIMIT 1;
```



LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONE 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 NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA 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
```

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT
```

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

```
FROM
```

```
orders
```

```
GROUP BY HOUR(order_time);
```



JOIN RELEVANT TABLES TO FIND THE CATEGORY WISE DISTRIBUTION OF PIZZAS.

```
SELECT  
  
ROUND(AVG(quantity), 0)  
as avg_pizza_ordered_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 ;
```



DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON 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
```



GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
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 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 revenue desc;
```

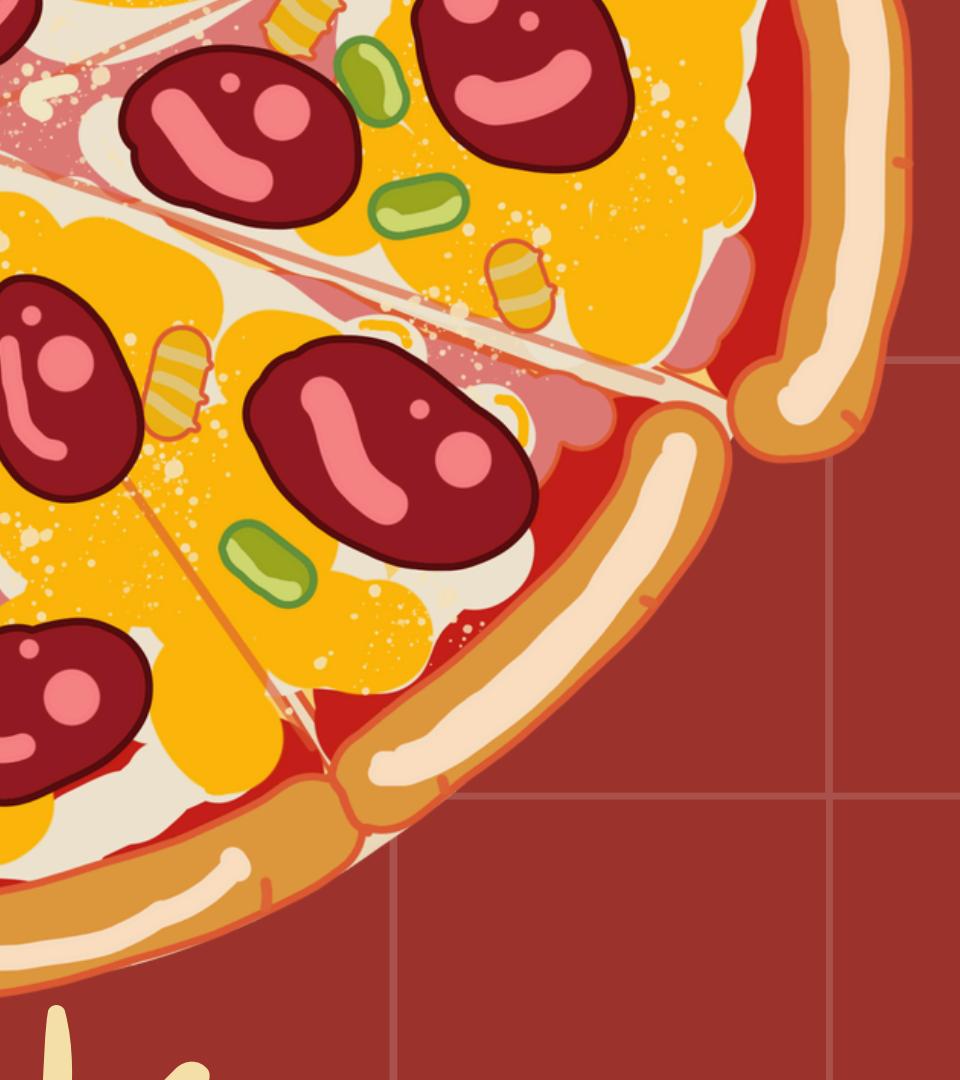


CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
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
```

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;
```



DETERMINE THE TOP 3 POST 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;
```

SUMMARY

In summary, the pizza sales report indicates a strong overall performance, with notable success in our top-selling categories such as Pepperoni and Margherita. Regional analysis shows significant variations, suggesting opportunities for targeted marketing strategies in underperforming areas.

THANK YOU