PIZZA SALES ANLYSIS of PIZZA KINGS USING MySQL

ROHAN KAPOOR

INTRODUCTION

This project focuses on leveraging SQL queries to analyze sales data from **Pizza Kings**. The goal of this project is to uncover key insights into customer behavior, pizza preferences, and overall sales performance.
By working with a structured dataset containing information on orders, pizza types, and customer details, MySQL is utilized to query, organize, and extract valuable business insights.

These queries help in understanding the driving factors behind sales trends and in making data-driven decisions for optimizing inventory, enhancing marketing strategies, and improving customer satisfaction at Pizza King.

The project involves executing a range of SQL queries to answer critical business questions such as:

- 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 category 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.

01

Retrieve the total number of orders placed

Retrieve the total number of orders placed – SQL

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

Result Grid

	total_orders
▶	21350

02

Calculate the total revenue generated from pizza sales.– SQL

Calculate the total revenue generated from pizza sales– SQL

```sql
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;
```

Result Grid

	total_sales
▶	817860.05

03

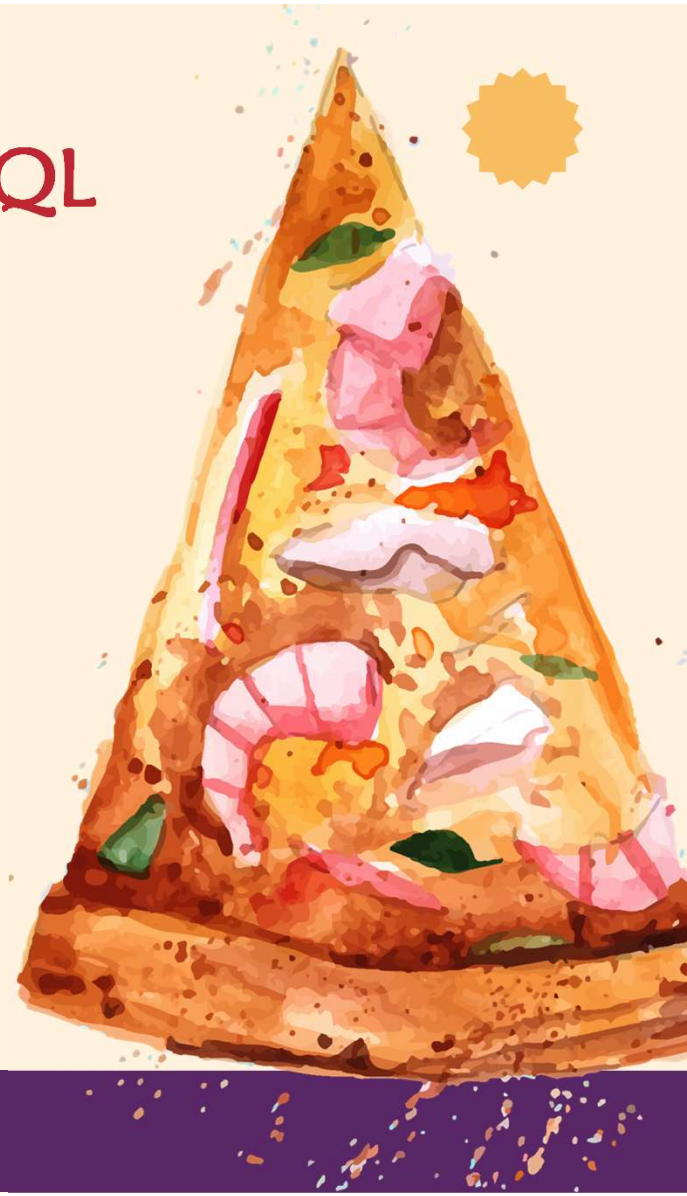
Identify the highest-priced pizza

Identify the highest-priced pizza--SQL

```sql
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;
```

Result Grid | Filter Rows:

name	price
The Greek Pizza	35.95



04

Identify the most common pizza size ordered.-SQL

Identify the most common pizza size ordered.-SQL

```sql
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 3;
```

Result Grid | Filter Rows

size	order_count
L	18526
M	15385
S	14137

05

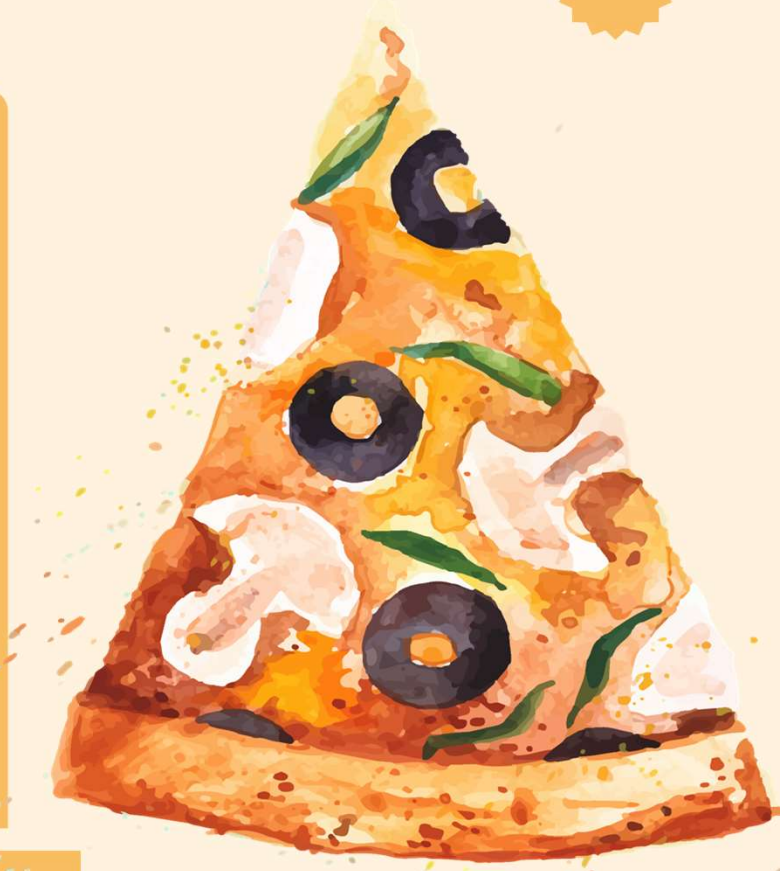
List the top 5 most ordered pizza types along with their quantities -- SQL

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

```sql
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;
```

Result Grid | Filter Rows:

name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371



06

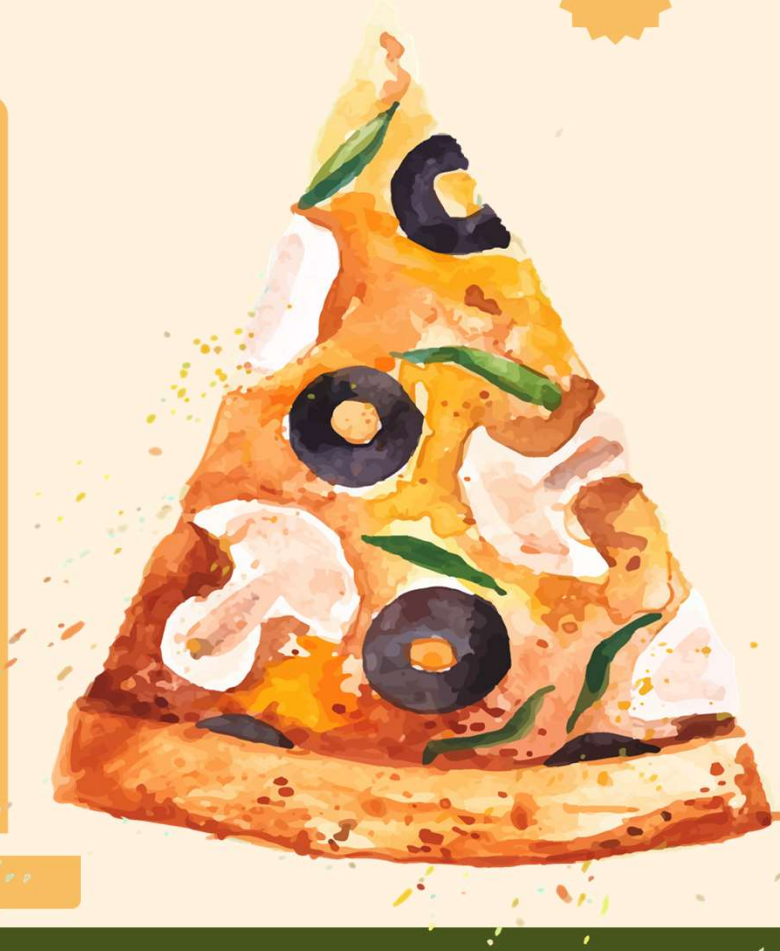
Join the necessary tables to find the total quantity of each pizza category ordered

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

```sql
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;
```

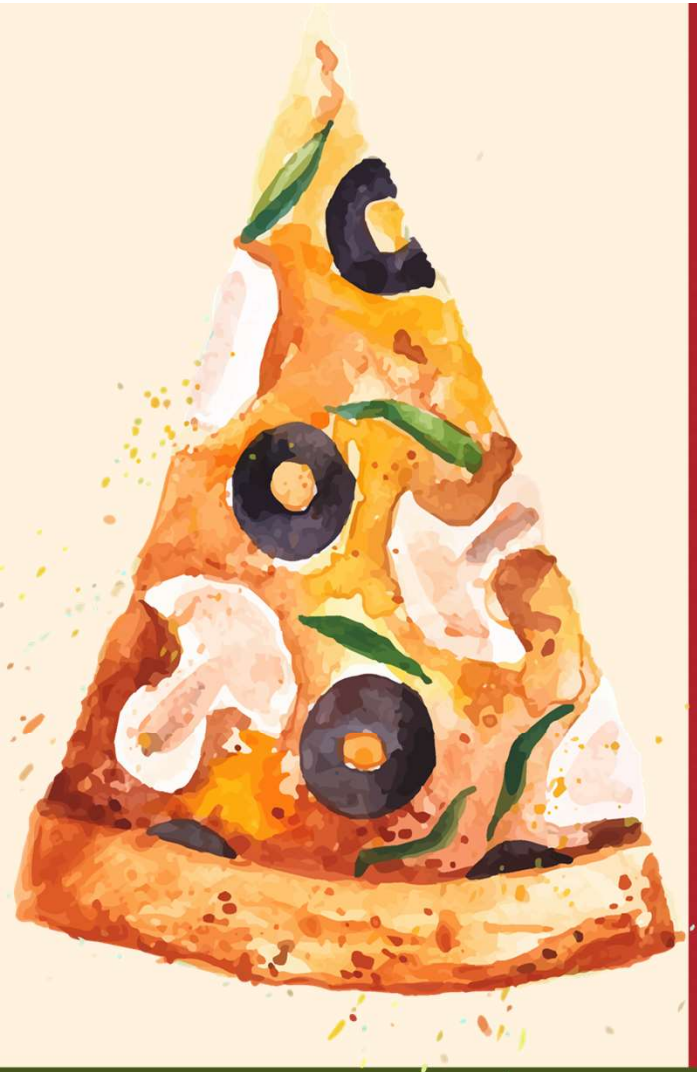
Result Grid | Filte

category	Quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050



07

Determine the distribution of orders by hour of the day.



Determine the distribution of orders by hour of the day

```sql
SELECT
    HOUR (ORDER_TIME), COUNT(order_id) AS orders_count
FROM
    orders
GROUP BY HOUR(order_time);
```

Result Grid | Filter Rows:

HOUR (ORDER_TIME)	orders_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336

08

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

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

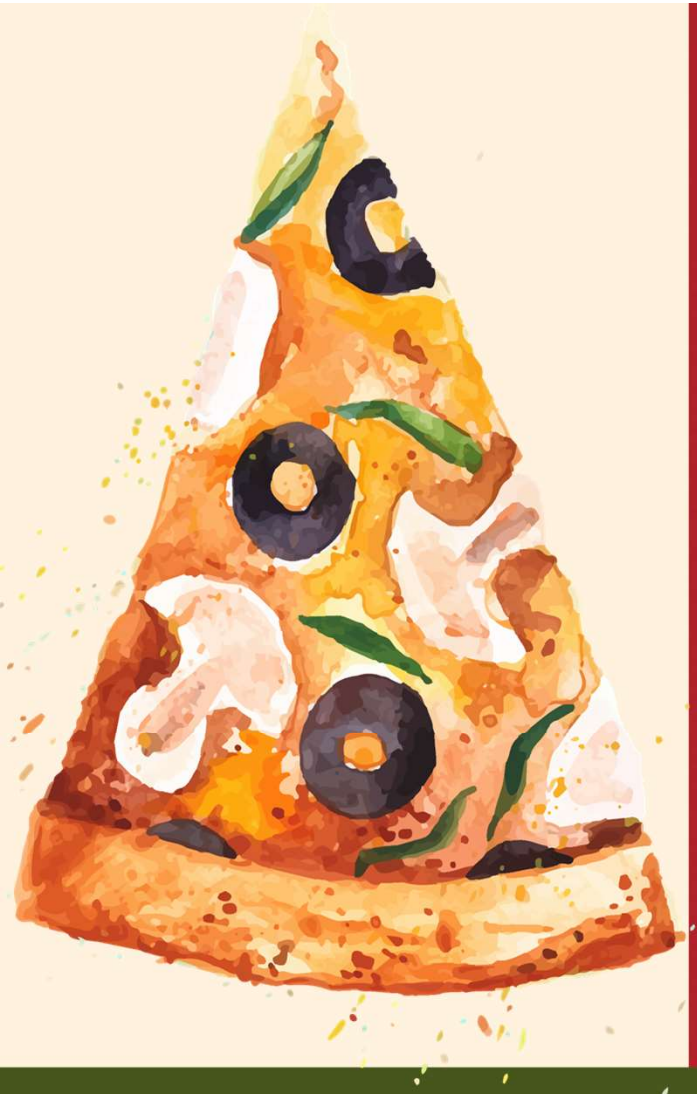
```
SELECT round(
  AVG(quantity),0)
FROM  (SELECT
      orders.Order_date, SUM(order_details.quantity) as quantity
  FROM
      orders
  JOIN order_details ON orders.order_id = order_details.order_id
  GROUP BY orders.order_date) AS order_quantity;
```

Result Grid

	round(AVG(quantity),0)
▶	138

09

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



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

```
Select category,count(name) from pizza_types
Group by category;
```

Result Grid | Filter Ro

category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

10

Determine the top 3 most ordered pizza types based on revenue

Determine the top 3 most ordered pizza types based on revenue

```sql
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 2;
```

Result Grid | Filter Rows:

NAME	REVENUE
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768

THANKS

rohankapoorrk714@gmail.com

linkedin.com/in/rohan-kapoor-1b449b273

https://github.com/rohankapoor93