HELLO

This project focuses on analyzing pizza sales data using MySQL. Through this case study, I have written and executed SQL queries to extract meaningful insights such as total revenue, best-selling pizzas, peak order times, and customer preferences. The goal of this project is to demonstrate practical SQL skills in data retrieval, aggregation, and reporting, using real-world business scenarios.

DATASET DESCRIPTION

This project is based on a structured pizza sales dataset, designed to simulate real-world transactions in a pizza restaurant. The dataset is divided into four relational tables, each serving a specific purpose:

1. **Orders.csv** : Contains unique order id along with date and time of each order.

2. **Orders_details.csv** : Contains fields like order id , pizza id and quantity , helping to identify what was ordered and how much.

3. **Pizza_types.csv :** Lists detailed descriptions of each pizza type using pizza_type id , along with name and category and a list of ingredients.

4. **Pizzas.csv** : Maps each pizza_id to its pizza_type_id , size and price

Retrieve the total number of orders placed.

```
SELECT
    COUNT(order_details_id) AS total_orders_placed
FROM
    orders_details;
```

	total_orders_placed
▶	48620

Calculate the total revenue generated from pizza sales.

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

	total_sales
▶	817860.05

Identify the highest-priced pizza.

```sql
SELECT
    pizza_types.name, pizzas.price
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

	name	price
▶	The Greek Pizza	35.95

Identify the most common pizza size ordered.

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

	size	quantity_ordered
▶	L	18526

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

```sql
SELECT
    pizza_types.name,
    SUM(orders_details.quantity) AS quantity_ordered
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_ordered DESC
LIMIT 5;
```

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

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

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

	category	total_quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Determine the distribution of orders by hour of the day.

```
SELECT
    HOUR(order_time), COUNT(order_id) AS quantity
FROM
    orders
GROUP BY HOUR(orders.order_time);
```

	HOUR(order_time)	quantity
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920

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

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

	category	COUNT(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

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

```sql
SELECT
    AVG(quantity)
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;
```

	AVG(quantity)
▶	138.4749

Determine the top 3 most ordered pizza types based on revenue

```sql
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 ;
```

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5

THANKYOU