🍽 Restaurant Order Analysis Using SQL

📌 Overview

This project involves analyzing a restaurant's ordering system using structured SQL queries. The objective was to explore menu offerings, understand customer ordering behavior, and identify patterns in high-spend transactions. The analysis was performed using two primary datasets: 'menu_items' and 'order_details'.

🎯 Project Objectives

1. Menu Analysis

- View and explore all menu items.
- Identify the most and least expensive dishes.
- Analyze category-wise pricing and item distribution.
- Deep-dive into Italian cuisine: price range and item count.

2. Order Analysis

- Determine the date range of customer orders.
- Count total orders and total items ordered.
- Identify the order with the most items.
- Detect high-volume orders (more than 12 items).

3. Sales Insights (Join Analysis)

- Combine menu_items with order_details for detailed analysis.
- Find the most and least ordered items and their categories.
- Identify top 5 highest-revenue orders.
- Analyze category-level breakdown of top orders.

🧠 Key Insights

- High-revenue orders often include multiple high-priced items from key categories.
- Certain Italian dishes are both frequently ordered and high-priced.
- Menu categories show varied pricing and item popularity, offering opportunities for optimization.

Technologies Used

- SQL (MySQL)
- Database: restaurant_db
- Tables: menu_items, order_details

More Sample SQL Queries

Top 5 Highest Revenue Orders:

```
SELECT order_id, SUM(price) AS total_spend
FROM order_details od
LEFT JOIN menu_items mi ON od.item_id = mi.menu_item_id
GROUP BY order_id
ORDER BY total_spend DESC
LIMIT 5;
```

Most & Least Ordered Items:

```
SELECT item_name, COUNT(order_details_id) AS num_purchases, category
FROM order_details od
LEFT JOIN menu_items mi ON od.item_id = mi.menu_item_id
GROUP BY item_name, category
ORDER BY num_purchases DESC
LIMIT 1;

SELECT item_name, COUNT(order_details_id) AS num_purchases, category
FROM order_details od
LEFT JOIN menu_items mi ON od.item_id = mi.menu_item_id
GROUP BY item_name, category
ORDER BY num_purchases ASC
LIMIT 1;
```

Order with Most Items:

```
SELECT order_id, COUNT(item_id) AS num_items
FROM order_details
GROUP BY order_id
```

```
ORDER BY num_items DESC
LIMIT 1;
```

Dish Count by Category:

```
SELECT category, COUNT(*) AS num_dishes
FROM menu_items
GROUP BY category;
```

Average Price per Category:

```
SELECT category, AVG(price) AS avg_price
FROM menu_items
GROUP BY category;
```