

```

In [1]: # Import necessary libraries
import sqlite3
import pandas as pd
import matplotlib.pyplot as plt

# Create an in-memory SQLite database
conn = sqlite3.connect(':memory:')
cursor = conn.cursor()

# Load and adjust the SQL script for SQLite compatibility
with open('create_restaurant_db.sql', 'r') as sql_file:
    sql_script = sql_file.read()

# Remove SCHEMA-related commands and adjust data types for SQLite
sql_script_adjusted = sql_script.replace("DROP SCHEMA IF EXISTS restaurant_db;", "")
sql_script_adjusted = sql_script_adjusted.replace("CREATE SCHEMA restaurant_db;", "")
sql_script_adjusted = sql_script_adjusted.replace("USE restaurant_db;", "")
sql_script_adjusted = sql_script_adjusted.replace("SMALLINT", "INTEGER")
sql_script_adjusted = sql_script_adjusted.replace("VARCHAR(45)", "TEXT")
sql_script_adjusted = sql_script_adjusted.replace("DECIMAL(5,2)", "REAL")

# Execute the adjusted SQL script
cursor.executescript(sql_script_adjusted)
conn.commit()

# Objective 1: Explore the menu_items table
# 1. Number of items on the menu
query_1 = "SELECT COUNT(*) FROM menu_items;"
num_items = cursor.execute(query_1).fetchone()[0]
print(f"Number of items on the menu: {num_items}")

# 2. Least and most expensive items
query_2 = "SELECT item_name, price FROM menu_items ORDER BY price ASC LIMIT 1;"
least_expensive = cursor.execute(query_2).fetchone()
query_3 = "SELECT item_name, price FROM menu_items ORDER BY price DESC LIMIT 1;"
most_expensive = cursor.execute(query_3).fetchone()
print(f"Least expensive item: {least_expensive[0]} - ${least_expensive[1]:.2f}")
print(f"Most expensive item: {most_expensive[0]} - ${most_expensive[1]:.2f}")

# 3. Number of Italian dishes and their price range
query_4 = "SELECT COUNT(*) FROM menu_items WHERE category='Italian';"
italian_count = cursor.execute(query_4).fetchone()[0]
query_5 = "SELECT item_name, price FROM menu_items WHERE category='Italian' ORDER BY price ASC LIMIT 1;"
least_expensive_italian = cursor.execute(query_5).fetchone()
query_6 = "SELECT item_name, price FROM menu_items WHERE category='Italian' ORDER BY price DESC LIMIT 1;"
most_expensive_italian = cursor.execute(query_6).fetchone()
print(f"Number of Italian dishes: {italian_count}")
print(f"Least expensive Italian dish: {least_expensive_italian[0]} - ${least_expensive_italian[1]:.2f}")
print(f"Most expensive Italian dish: {most_expensive_italian[0]} - ${most_expensive_italian[1]:.2f}")

# 4. Number of dishes and average price per category
query_7 = "SELECT category, COUNT(*) AS num_dishes, AVG(price) AS avg_price FROM menu_items GROUP BY category;"
df_category = pd.read_sql_query(query_7, conn)
print(df_category)

# Visualization: Average price per category
plt.figure(figsize=(10, 6))
plt.bar(df_category['category'], df_category['avg_price'])
plt.xlabel('Category')
plt.ylabel('Average Price ($)')
plt.title('Average Price per Menu Category')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# Objective 2: Explore the order_details table

```

```

# 1. Date range of orders
query_8 = "SELECT MIN(order_date), MAX(order_date) FROM order_details;"
date_range = cursor.execute(query_8).fetchone()
print(f"Order date range: {date_range[0]} to {date_range[1]}")

# 2. Total number of orders and items ordered
query_9 = "SELECT COUNT(DISTINCT order_id), COUNT(*) FROM order_details;"
order_stats = cursor.execute(query_9).fetchone()
print(f"Total orders: {order_stats[0]}")
print(f"Total items ordered: {order_stats[1]}")

# 3. Order with the most items
query_10 = """
SELECT order_id, COUNT(*) AS item_count
FROM order_details
GROUP BY order_id
ORDER BY item_count DESC
LIMIT 1;
"""
max_items_order = cursor.execute(query_10).fetchone()
print(f"Order with the most items: Order ID {max_items_order[0]} with {max_items_order[1]} items")

# 4. Number of orders with more than 12 items
query_11 = """
SELECT COUNT(*)
FROM (SELECT order_id FROM order_details GROUP BY order_id HAVING COUNT(*) > 12);
"""
orders_over_12 = cursor.execute(query_11).fetchone()[0]
print(f"Number of orders with more than 12 items: {orders_over_12}")

# Objective 3: Analyze customer behavior
# 1. Least and most ordered items
query_12 = """
SELECT mi.item_name, mi.category, COUNT(od.item_id) AS order_count
FROM order_details od
JOIN menu_items mi ON od.item_id = mi.menu_item_id
GROUP BY mi.item_name, mi.category
ORDER BY order_count ASC
LIMIT 1;
"""
least_ordered_item = cursor.execute(query_12).fetchone()
print(f"Least ordered item: {least_ordered_item[0]} ({least_ordered_item[1]}) with {least_ordered_item[2]} orders")

query_13 = query_12.replace('ASC', 'DESC')
most_ordered_item = cursor.execute(query_13).fetchone()
print(f"Most ordered item: {most_ordered_item[0]} ({most_ordered_item[1]}) with {most_ordered_item[2]} orders")

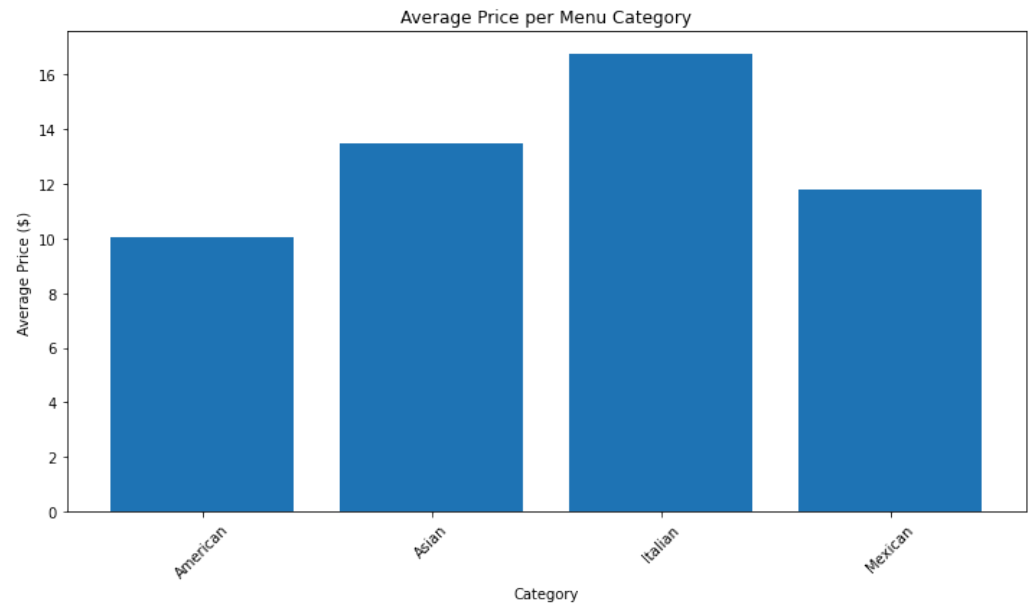
# 2. Top 5 highest spend orders
query_14 = """
SELECT od.order_id, SUM(mi.price) AS total_spent
FROM order_details od
JOIN menu_items mi ON od.item_id = mi.menu_item_id
GROUP BY od.order_id
ORDER BY total_spent DESC
LIMIT 5;
"""
df_top_orders = pd.read_sql_query(query_14, conn)
print(df_top_orders)

# Most expensive order
most_expensive_order = df_top_orders.iloc[0]['total_spent']
print(f"Most expensive order amount: {most_expensive_order:.2f}")

```

Number of items on the menu: 32
 Least expensive item: Edamame – \$5.00
 Most expensive item: Shrimp Scampi – \$19.95
 Number of Italian dishes: 9
 Least expensive Italian dish: Spaghetti – \$14.50
 Most expensive Italian dish: Shrimp Scampi – \$19.95

	category	num_dishes	avg_price
0	American	6	10.066667
1	Asian	8	13.475000
2	Italian	9	16.750000
3	Mexican	9	11.800000



Order date range: 2023-01-01 to 2023-03-31
 Total orders: 5370
 Total items ordered: 12234
 Order with the most items: Order ID 4482 with 14 items
 Number of orders with more than 12 items: 23
 Least ordered item: Chicken Tacos (Mexican) with 123 orders
 Most ordered item: Hamburger (American) with 622 orders

	order_id	total_spent
0	440	192.15
1	2075	191.05
2	1957	190.10
3	330	189.70
4	2675	185.10

Most expensive order amount: 192.15