



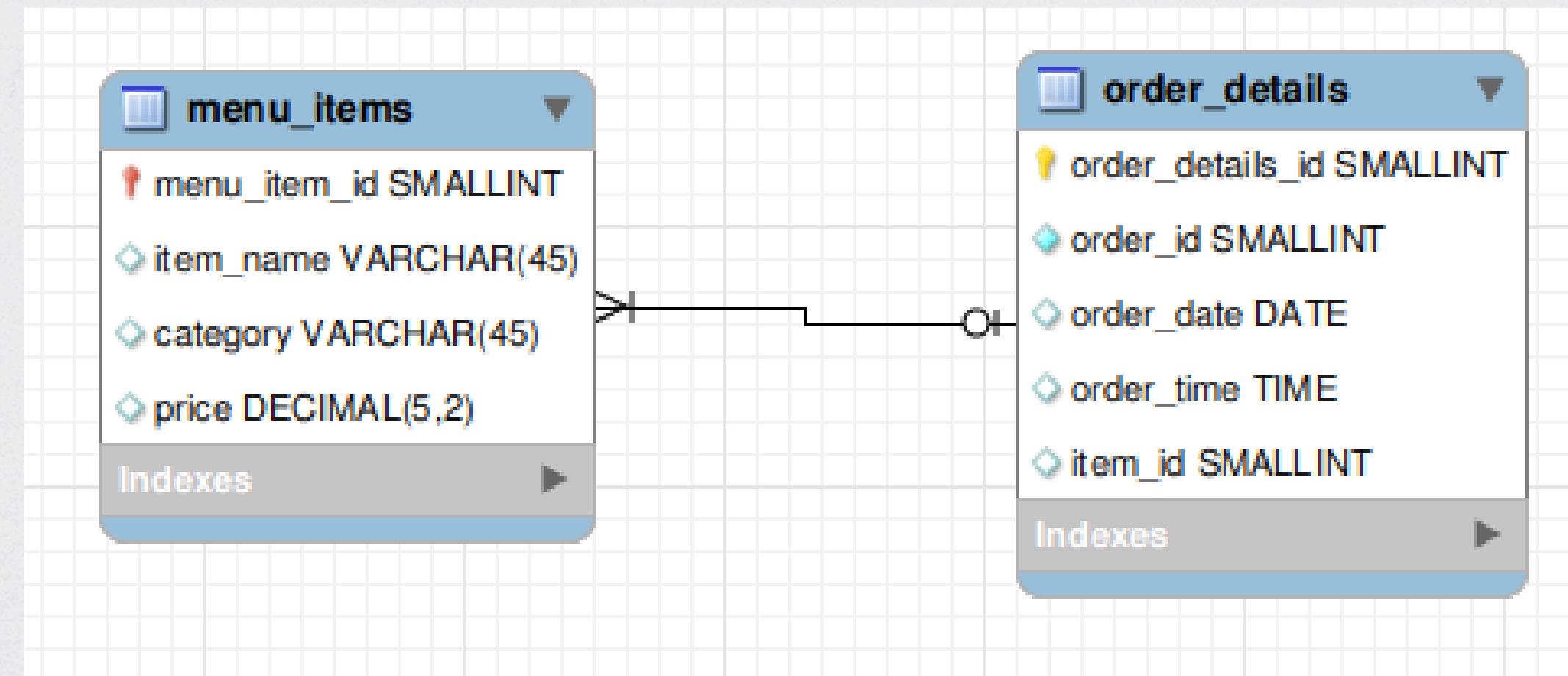
# Restaurant Data Analysis using SQL

# Introduction

*Hello, my name is Siddhant Jain. In this project, I analyzed restaurant data using SQL to extract valuable insights. This analysis focuses on understanding how different menu items, their categories, and prices contribute to overall sales. By identifying top-selling items and evaluating category performance, this project provides actionable insights that can help shape future business strategies, optimize menu offerings, and improve pricing decisions to enhance profitability.*



# Data Model



# Project Analysis

- Count the total number of menu items in the menu\_items table.
- Calculate the total sales amount by summing up the prices of all ordered items.
- Find the number of orders placed each day.
- Determine the average price of menu items.
- Calculate the total sales amount for each category of menu items.
- Retrieve a list of orders along with the names of the menu items in each order.
- List the menu items that were ordered in January 2023.
- Identify the top 5 most ordered menu items.
- Find the 5 most expensive orders.
- Generate a daily sales summary showing the total sales for each day.
- Calculate the cumulative sales over time, showing the running total sales amount for each day.
- Compute the average order value over the last 7 days for each day in the dataset.
- Get the top 3 menu item from each category based on their total sales



**Count the total number of menu items in the menu\_items table.**

```
SELECT  
    COUNT(item_name) AS total_items  
FROM  
    restaurant_db.menu_items;
```

#	total_items
1	32

# Calculate the total sales amount by summing up the prices of all ordered items.

```
SELECT
    SUM(menu_items.price) AS total_price
FROM
    restaurant_db.menu_items
    JOIN
    restaurant_db.order_details ON menu_items.menu_item_id = order_details.item_id;
```

#	total_price
1	159217.90

# Find the number of orders placed each day.

**SELECT**

**order\_date, COUNT(\*) AS total\_orders**

**FROM**

**restaurant\_db.order\_details**

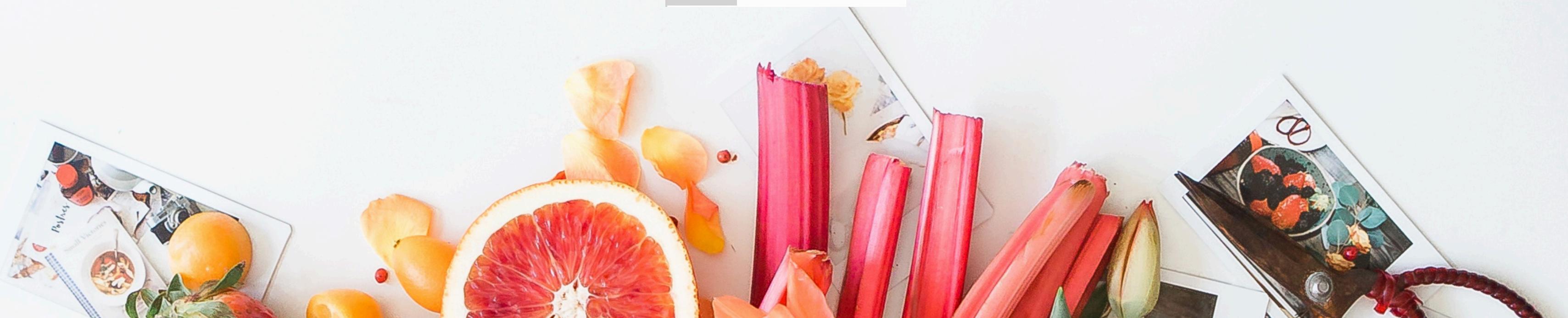
**GROUP BY order\_date;**

#	order_date	total_orders
1	2023-01-01	161
2	2023-01-02	160
3	2023-01-03	154
4	2023-01-04	106
5	2023-01-05	121
6	2023-01-06	144

# Determine the average price of menu items.

```
SELECT  
    ROUND(AVG(price), 2) AS avg_price  
FROM  
    restaurant_db.menu_items;
```

#	avg_price
1	13.29



**Calculate the total sales amount for each category of menu items.**

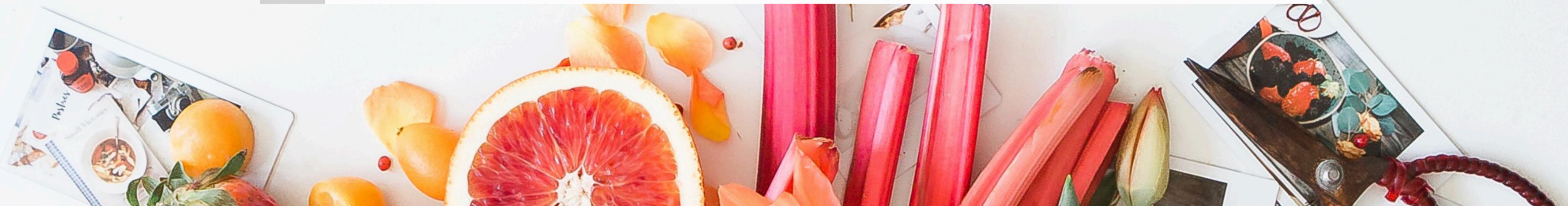
```
SELECT  
    category, SUM(price) AS total_price  
FROM  
    restaurant_db.menu_items  
GROUP BY category;
```

#	category	total_price
1	American	60.40
2	Asian	107.80
3	Mexican	106.20
4	Italian	150.75

# Retrieve a list of orders along with the names of the menu items in each order.

```
SELECT  
    order_details.* , menu_items.item_name  
FROM  
    restaurant_db.menu_items  
    JOIN  
    restaurant_db.order_details ON menu_items.menu_item_id = order_details.item_id;
```

#	order_details_id	order_id	order_date	order_time	item_id	item_name
1	1	1	2023-01-01	11:38:36	109	Korean Beef Bowl
2	2	2	2023-01-01	11:57:40	108	Tofu Pad Thai
3	3	2	2023-01-01	11:57:40	124	Spaghetti
4	4	2	2023-01-01	11:57:40	117	Chicken Burrito
5	5	2	2023-01-01	11:57:40	129	Mushroom Ravioli
6	6	2	2023-01-01	11:57:40	106	French Fries
7	7	3	2023-01-01	12:12:28	117	Chicken Burrito
8	8	3	2023-01-01	12:12:28	119	Chicken Torta



# List the menu items that were ordered in January 2023.

```
SELECT
    menu_items.item_name, order_details.order_date
FROM
    restaurant_db.menu_items
    JOIN
    restaurant_db.order_details ON menu_items.menu_item_id = order_details.item_id
WHERE
    order_details.order_date BETWEEN '2023-01-01' AND '2023-01-31';
```

#	item_name	order_date
1	Korean Beef Bowl	2023-01-01
2	Tofu Pad Thai	2023-01-01
3	Spaghetti	2023-01-01
4	Chicken Burrito	2023-01-01
5	Mushroom Ravioli	2023-01-01
6	French Fries	2023-01-01
7	Chicken Burrito	2023-01-01
8	Chicken Torta	2023-01-01
9	Chicken Burrito	2023-01-01
10	Chicken Burrito	2023-01-01

# Identify the top 5 most ordered menu items.

```
SELECT
    menu_items.item_name,
    COUNT(order_details.item_id) AS total_orders
FROM
    restaurant_db.menu_items
    JOIN
    restaurant_db.order_details ON menu_items.menu_item_id = order_details.item_id
GROUP BY menu_items.item_name
ORDER BY total_orders DESC
LIMIT 5;
```

#	item_name	total_orders
1	Hamburger	622
2	Edamame	620
3	Korean Beef Bowl	588
4	Cheeseburger	583
5	French Fries	571



# Find the 5 most expensive orders.

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

#	order_id	total_price
1	440	192.15
2	2075	191.05
3	1957	190.10
4	330	189.70
5	2675	185.10

# Generate a daily sales summary showing the total sales for each day.

**SELECT**

```
order_details.order_date,  
SUM(menu_items.price) AS daily_sales
```

**FROM**

```
restaurant_db.order_details
```

**JOIN**

```
restaurant_db.menu_items ON menu_items.menu_item_id = order_details.item_id
```

**GROUP BY** order\_details.order\_date;

#	order_date	daily_sales
1	2023-01-01	2091.60
2	2023-01-02	1994.70
3	2023-01-03	1983.70
4	2023-01-04	1356.85
5	2023-01-05	1589.85
6	2023-01-06	1888.00
7	2023-01-07	1691.10
8	2023-01-08	2258.10
9	2023-01-09	1540.40
10	2023-01-10	1866.40

# Calculate the cumulative sales over time, showing the running total sales amount for each day.

```
select sales.order_date,  
       sum(sales.daily_sales) over(order by sales.order_date) as cum_sales from  
(SELECT  
       order_details.order_date,  
       SUM(menu_items.price) AS daily_sales  
FROM  
       restaurant_db.order_details  
       JOIN  
       restaurant_db.menu_items ON menu_items.menu_item_id = order_details.item_id  
GROUP BY order_details.order_date) as sales;
```

#	order_date	cum_sales
1	2023-01-01	2091.60
2	2023-01-02	4086.30
3	2023-01-03	6070.00
4	2023-01-04	7426.85
5	2023-01-05	9016.70
6	2023-01-06	10904.70
7	2023-01-07	12595.80
8	2023-01-08	14853.90
9	2023-01-09	16394.30
10	2023-01-10	18260.70

# Compute the average order value over the last 7 days for each day in the dataset.

```
select sales.order_date,  
       avg(sales.total_price) OVER (ORDER BY order_date  
                                     rows between 6 preceding and current row) AS avg_order_value_last_7_days  
  from  
(SELECT order_date,  
        SUM(menu_items.price) as total_price  
     FROM order_details  
    JOIN menu_items ON order_details.item_id = menu_items.menu_item_id  
   GROUP BY order_date  
  ORDER BY order_date) as sales;
```

#	order_date	avg_order_value_last_7_day
1	2023-01-01	2091.600000
2	2023-01-02	2043.150000
3	2023-01-03	2023.333333
4	2023-01-04	1856.712500
5	2023-01-05	1803.340000
6	2023-01-06	1817.450000
7	2023-01-07	1799.400000
8	2023-01-08	1823.185714
9	2023-01-09	1758.285714
10	2023-01-10	1741.528571

# Get the top 3 menu item from each category based on their total sales

```
select sales.category, sales.item_name, sales.total_sales
from
(select menu_items.category, menu_items.item_name,
sum(menu_items.price) as total_sales,
rank() over(partition by menu_items.category order by sum(menu_items.price) desc) as ranking
from restaurant_db.menu_items
join restaurant_db.order_details
on menu_items.menu_item_id = order_details.item_id
group by menu_items.category, menu_items.item_name) as sales
where sales.ranking <=3
```

#	category	item_name	total_sales
1	American	Cheeseburger	8132.85
2	American	Hamburger	8054.90
3	American	French Fries	3997.00
4	Asian	Korean Beef Bowl	10554.60
5	Asian	Tofu Pad Thai	8149.00
6	Asian	Orange Chicken	7524.00
7	Italian	Spaghetti & Meatballs	8436.50
8	Italian	Eggplant Parmesan	7119.00
9	Italian	Chicken Parmesan	6533.80
10	Mexican	Steak Torta	6821.55
11	Mexican	Chicken Burrito	5892.25
12	Mexican	Steak Burrito	5292.30

# THANK YOU

For seeing data and sql queries used, please visit

<https://github.com/siddhantjain603/Restaurant-Data-Analysis-using-SQL>

