# Restaurant Orders
# SQL Analysis

## What were the least and most ordered items? What categories were they in?

```
Most ordered items:
SELECT COUNT(O.ORDER_ID) AS total_orders,
       m.item_name,
       m.category
FROM order_details AS o
FULL JOIN menu_items AS m
ON o.item_id = m.menu_item_id
GROUP BY m.item_name,
         m.category
ORDER BY order_num DESC
LIMIT 3;
```

---

Results:

| | total_orders | item_name | category |
|---|---|---|---|
| 1 | 622 | Hamburger | American |
| 2 | 620 | Edamame | Asian |
| 3 | 588 | Korean Beef Bowl | Asian |

## What were the least and most ordered items? What categories were they in?

```
Least ordered items:
SELECT COUNT(O.ORDER_ID) AS total_orders,
       m.item_name,
       m.category
FROM order_details AS o
FULL JOIN menu_items AS m
ON o.item_id = m.menu_item_id
GROUP BY m.item_name,
         m.category
ORDER BY order_num
LIMIT 3;
```

---

Results:

| | total_orders | item_name | category |
|---|---|---|---|
| 1 | 123 | Chicken Tacos | Mexican |
| 2 | 205 | Potstickers | Asian |
| 3 | 207 | Cheese Lasagna | Italian |

**What do the highest spend orders look like? Which items did they buy and how much did they spend?**

```
Customer with order number 17
SELECT SUM(m.price) AS total_price,
       item_name
FROM menu_items AS m
FULL JOIN order_details AS o
ON m.menu_item_id = o.item_id
WHERE o.order_id = 17
GROUP BY  o.order_time,
          o.order_id,
          item_name
ORDER BY order_time,
         order_id,
         total_price DESC;
```

Results:

| | total_price | item_name |
|---|---|---|
| 1 | 17.95 | Meat Lasagna |
| 2 | 17.95 | Spaghetti & Meatballs |
| 3 | 17.95 | Chicken Parmesan |
| 4 | 15.50 | Mushroom Ravioli |
| 5 | 14.95 | Steak Burrito |
| 6 | 14.50 | Spaghetti |
| 7 | 13.95 | Steak Tacos |
| 8 | 12.95 | Chicken Burrito |
| 9 | 12.95 | Hamburger |

**What do the highest spend orders look like? Which items did they buy and how much did they spend?**

```
Customer with order number 9
SELECT SUM(m.price) AS total_price,
       item_name
FROM menu_items AS m
FULL JOIN order_details AS o
ON m.menu_item_id = o.item_id
WHERE o.order_id = 9
GROUP BY  o.order_time,
          o.order_id,
          item_name
ORDER BY order_time,
         order_id,
         total_price DESC;
```

Results:

| | total_price | item_name |
|---|---|---|
| 1 | 25.90 | Chicken Burrito |
| 2 | 19.95 | Shrimp Scampi |
| 3 | 17.95 | Pork Ramen |
| 4 | 16.95 | Eggplant Parmesan |
| 5 | 15.50 | Mushroom Ravioli |
| 6 | 14.50 | Fettuccine Alfredo |
| 7 | 14.50 | Tofu Pad Thai |
| 8 | 7.00 | Chips & Salsa |

**What do the highest spend orders look like? Which items did they buy and how much did they spend?**

```
Customer with order number 2
SELECT SUM(m.price) AS total_price,
       item_name
FROM menu_items AS m
FULL JOIN order_details AS o
ON m.menu_item_id = o.item_id
WHERE o.order_id = 2
GROUP BY  o.order_time,
          o.order_id,
          item_name
ORDER BY order_time,
         order_id,
         total_price DESC;
```

---

Results:

| | total_price | item_name |
|---|---|---|
| 1 | 15.50 | Mushroom Ravioli |
| 2 | 14.50 | Spaghetti |
| 3 | 14.50 | Tofu Pad Thai |
| 4 | 12.95 | Chicken Burrito |
| 5 | 7.00 | French Fries |

## Were there certain times that had more or less orders?

```sql
With this query we can check all of the order numbers on every given day
SELECT COUNT(o.order_id) AS date_count,
       o.order_date
FROM menu_items AS m
FULL JOIN order_details AS o
ON m.menu_item_id = o.item_id
GROUP BY o.order_date
ORDER BY o.order_date;
```

---

Results:

| | number_of_orders | order_date |
|---|---|---|
| 1 | 160 | 2023-01-01 |
| 2 | 159 | 2023-01-02 |
| 3 | 150 | 2023-01-03 |
| 4 | 106 | 2023-01-04 |
| 5 | 121 | 2023-01-05 |

## Were there certain times that had more or less orders?

```sql
Or analyze the AVG/MIN/MAX number of orders by weeks to see if we have some outlying data
WITH dates AS
(SELECT COUNT(o.order_id) AS date_count,
        EXTRACT('week' FROM o.order_date) AS week
FROM menu_items AS m
FULL JOIN order_details AS o
ON m.menu_item_id = o.item_id
GROUP BY o.order_date)

SELECT  ROUND(AVG(date_count),2) AS avg_orders,
        MIN(date_count) AS min_orders,
        MAX(date_count) AS max_orders,
        week
FROM dates
GROUP BY week
ORDER BY week;
```

Results:

| | avg_orders | min_orders | max_orders | week |
|---|---|---|---|---|
| 1 | 139.86 | 106 | 170 | 1 |
| 2 | 124.14 | 114 | 142 | 2 |
| 3 | 134.86 | 118 | 153 | 3 |
| 4 | 125.00 | 101 | 148 | 4 |
| 5 | 146.00 | 132 | 186 | 5 |
| 6 | 134.00 | 121 | 151 | 6 |
| 7 | 134.00 | 116 | 158 | 7 |
| 8 | 130.00 | 91 | 168 | 8 |

## Which cuisines should we focus on developing more menu items for based on the data?

```
We could look at the top 5 most paying costumers (only included the first)
WITH orders AS (
SELECT SUM(m.price) as total_price,
       m.category
FROM menu_items AS m
FULL JOIN order_details AS o
ON m.menu_item_id = o.item_id
WHERE o.order_id = 17
GROUP BY m.item_name,
         m.category)

SELECT COUNT(category) AS number_of_foods,
       category
FROM orders
GROUP BY category
--------------------------------------------------------------------------------
Results:
       number_of_foods          category
1      6                        Italian
2      3                        Mexican
3      1                        American
```

## Which cuisines should we focus on developing more menu items for based on the data?

```
Or we can look at it from revenue by category
WITH orders AS (
SELECT SUM(m.price) AS total_price,
       m.category
FROM menu_items AS m
FULL JOIN order_details AS o
ON m.menu_item_id = o.item_id
GROUP BY m.item_name,
         m.category)

SELECT SUM(total_price) AS income,
       category
FROM orders
GROUP BY category
ORDER BY income DESC;
```

---

Results:

| | income | category |
|---|---|---|
| 1 | 49462.70 | Italian |
| 2 | 46720.65 | Asian |
| 3 | 34796.80 | Mexican |
| 4 | 28237.75 | American |