Restaurant Orders Analysis

Profile

Restaurant Orders is A quarter’s worth of orders from a fictitious restaurant serving international cuisine.

Problem & Objective

Problem :

How was the sales performance of Restaurant Orders?

Objective :

- To evaluate Restaurant Orders performance.
- To identify the highest-performing menu at Restaurant Orders.
- To identify the highest-performing menu categories at Restaurant Orders.

Data Preparation

Data Preparation

Data use from: Maven Analytics,

In Order_details table:

- order_details_id = Unique ID of an item in an order
- order_id = ID of an order
- order_date = Date an order was put in (MM/DD/YY)
- order_time = Time an order was put in (HH:MM:SS AM/PM)
- item_id = Matches the menu_item_id in the menu_items table

In menu_items table:

- menu_item_id = Unique ID of a menu item
- item_name = Name of a menu item
- category = Category or type of cuisine of the menu item
- price = Price of the menu item (US Dollars $)

Data Cleaning

The total data count is 12,234, but there are 137 missing values in the order_details. Since we don't know which menu items are missing, we decided to drop the null entries. After remove the null entries, the count is 12.097.

```sql
-- Count total
SELECT COUNT(*) total FROM order_details;

-- Count total missing values
SELECT COUNT(*) missing_values FROM order_details
WHERE item_id IS NULL;

-- DROP NULL
DELETE FROM order_details
WHERE item_id IS NULL;
```

total
12234

total
12097

missing_values
137

EDA

Data Analysis

```sql
-- Category
SELECT DISTINCT(category) total_category FROM menu_items;
-- Total four category
```

	total_category
▶	American
	Asian
	Mexican
	Italian

```sql
-- Total Menu
SELECT COUNT(item_name) total_menu FROM menu_items;
-- Total 32 menu
```

	total_menu
▶	32

```sql
-- Total Menu per Category
SELECT category, COUNT(item_name) total_menu FROM menu_items
GROUP BY 1;
-- There are 6 menu in American Category,
-- 8 menu in Asian Category,
-- 9 menu in Mexican Category, and
-- 9 menu in Italian Category
```

	category	total_menu
▶	American	6
	Asian	8
	Mexican	9
	Italian	9

Data Analysis

```sql
-- Average Daily Sales
WITH total_sales_per_day AS (
SELECT order_date, SUM(PRICE) total_sales
FROM order_details od JOIN menu_items mi ON od.item_id = mi.menu_item_id
GROUP BY 1
)
SELECT ROUND(AVG(total_sales), 2) average_sales_daily FROM total_sales_per_day;
-- Average Daily Sales is 1769.09
```

average_sales_daily
1769.09

```sql
-- Average Customer
WITH avgcus AS (
SELECT
    order_date,
    COUNT(DISTINCT(order_id)) total_customer
FROM order_details
GROUP BY 1
)
SELECT ROUND(AVG(total_customer)) average_total_customer FROM avgcusg;
-- The Average Customer is 59 customer
```

average_total_customer
59

```sql
-- Total Sales, Total Order, Total Customer
SELECT
    COUNT(DISTINCT(order_id)) total_customer,
    COUNT(order_details_id) total_order,
    SUM(price) total_sales
FROM order_details od JOIN menu_items mi
    ON od.item_id = mi.menu_item_id;
-- Total Customer is 5343, Total Order 12097, and Total Sales 159217.90
```

total_customer	total_order	total_sales
5343	12097	159217.90

Data Analysis

```sql
-- What were the least and most ordered items? What categories were they in?
SELECT
    mi.item_name,
    category,
    COUNT(*) total
FROM
    order_details od JOIN menu_items mi
        ON od.item_id = mi.menu_item_id
GROUP BY 1, 2
ORDER BY 3 DESC;
```

item_name	category	total
Hamburger	American	622
Edamame	Asian	620
Korean Beef Bowl	Asian	588
Cheeseburger	American	583
French Fries	American	571
Tofu Pad Thai	Asian	562
Steak Torta	Mexican	489
Spaghetti & Meatballs	Italian	470
Mac & Cheese	American	463
Chips & Salsa	Mexican	461
Orange Chicken	Asian	456
Chicken Burrito	Mexican	455
Eggplant Parmesan	Italian	420
Chicken Torta	Mexican	379
Spaghetti	Italian	367
Chicken Parmesan	Italian	364

Data Analysis

```
WITH most_order AS(
    SELECT
        mi.item_name,
        category,
        COUNT(*) total
    FROM
        order_details od JOIN menu_items mi
            ON od.item_id = mi.menu_item_id
GROUP BY 1, 2
ORDER BY 3 DESC
LIMIT 1
),
```

```
least_order AS(
    SELECT
        mi.item_name,
        category,
        COUNT(*) total
    FROM
        order_details od JOIN menu_items mi
            ON od.item_id = mi.menu_item_id
GROUP BY 1, 2
ORDER BY 3
LIMIt 1
)
```

```
SELECT *, 'most_order' orders FROM most_order
UNION
SELECT *, 'least_order' orders FROM least_order;
```

item_name	category	total	orders
Hamburger	American	622	most_order
Chicken Tacos	Mexican	123	least_order

Hamburger is the most orders with total 662 orders, and Chicken Tacos is the least orders with total 123 orders.

Data Analysis

```sql
-- What do the highest spend orders look like? Which items did they buy and how much did they spend?
SELECT order_id, SUM(price) total_spent FROM order_details od JOIN menu_items mi
ON od.item_id = mi.menu_item_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1;
```

	order_id	total_spent
▶	440	192.15

The highest spender is customer with order_id 440
with total spent $192.15.

Data Analysis

```
WITH highest_spend AS(
SELECT order_id, SUM(price) total_spent FROM order_details od JOIN menu_items mi
ON od.item_id = mi.menu_item_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1)
SELECT hs.order_id, item_name, price,
sum(price) OVER() total_spent FROM order_details od JOIN menu_items mi
ON od.item_id = mi.menu_item_id
JOIN highest_spend hs ON od.order_id = hs.order_id;
```

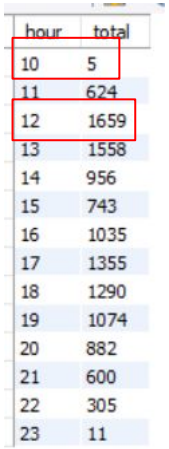
order_id	item_name	price	total_spent
440	Steak Tacos	13.95	192.15
440	Hot Dog	9.00	192.15
440	Spaghetti	14.50	192.15
440	Spaghetti & Meatballs	17.95	192.15
440	Spaghetti & Meatballs	17.95	192.15
440	Fettuccine Alfredo	14.50	192.15
440	Fettuccine Alfredo	14.50	192.15
440	Korean Beef Bowl	17.95	192.15
440	Meat Lasagna	17.95	192.15
440	Edamame	5.00	192.15
440	Chips & Salsa	7.00	192.15
440	Chicken Parmesan	17.95	192.15
440	French Fries	7.00	192.15
440	Eggplant Parmesan	16.95	192.15

total spending for all menus

There are 13 menu that customer with order id 440.

Data Analysis

```sql
-- Were there certain times that had more or less orders?
SELECT HOUR(order_time) hour, COUNT(*) total FROM order_details
GROUP BY 1
ORDER BY hour;
```



hour	total
10	5
11	624
12	1659
13	1558
14	956
15	743
16	1035
17	1355
18	1290
19	1074
20	882
21	600
22	305
23	11

We can develop strategies by offering targeted promotions.

12 pm is the most orders and 10 am is the less orders.

Data Analysis

```
-- Which cuisines should we focus on developing more menu items for based on the data?
SELECT category, COUNT(*) total_sold, SUM(price) total_sales
FROM order_details od JOIN menu_items mi ON od.item_id = mi.menu_item_id
GROUP BY 1
ORDER BY 2 DESC;
```

category	total_sold	total_sales
Asian	3470	46720.65
Italian	2948	49462.70
Mexican	2945	34796.80
American	2734	28237.75

Asian cuisine has the highest total items sold and total sales.

We can focused on Asian and Italian Cuisines. These categories have shown the highest sales performance, suggesting by expanding their menu option.

Data Analysis

```sql
-- MoM Growth
WITH cteone AS (
SELECT DATE_FORMAT(order_date, '%Y-%m') month, SUM(price) total_sales
FROM order_details od JOIN menu_items mi

    ON od.item_id = mi.menu_item_id
GROUP BY 1
),
ctetwo AS(
SELECT *,
    COALESCE(LAG(total_sales) OVER(ORDER BY month), total_sales) previous_sales
FROM cteone
)
```

```sql
SELECT
    *,
    CONCAT(ROUND((total_sales - previous_sales)
    /
    previous_sales * 100, 2), '%') growth_percentage
FROM ctetwo
ORDER BY month;
```

total_sales	previous_sales	growth_percentage
53816.95	53816.95	0.00%
50790.35	53816.95	-5.62%
54610.60	50790.35	7.52%

Sales decreased in February, but they increased in March.

Data Analysis

```sql
-- Comparison Daily Total Sales and Average Total Sales
WITH rts AS (
SELECT
order_date,
SUM(price) total_sales
FROM order_details od JOIN menu_items mi ON od.item_id = mi.menu_item_id
GROUP BY 1
)
SELECT *,
ROUND(AVG(total_sales) OVER(), 2) average_total_sales
FROM rts
ORDER BY 1, 2;
```

order_date	total_sales	average_total_sales
2023-01-01	2091.60	1769.09
2023-01-02	1994.70	1769.09
2023-01-03	1983.70	1769.09
2023-01-04	1356.85	1769.09
2023-01-05	1589.85	1769.09
2023-01-06	1888.00	1769.09
2023-01-07	1691.10	1769.09
2023-01-08	2258.10	1769.09
2023-01-09	1540.40	1769.09
2023-01-10	1866.40	1769.09
2023-01-11	1473.70	1769.09
2023-01-12	1510.35	1769.09

Dashboard

Dashboard

```sql
-- Query use on Power BI

SELECT
    order_id,
    order_details_id,
    order_date, dayname(order_date) dayname,hour(order_time) hour,
    CASE
        WHEN hour(order_time) BETWEEN 10 AND 11 THEN 'Morning'
        WHEN hour(order_time) BETWEEN 12 AND 18 THEN 'Afternoon'
        ELSE 'Night'
    END time_category,
    item_name,
    category,
    COUNT(DISTINCT(order_id)) total_customer,
    COUNT(order_details_id) total_order,
    SUM(price) total_sales
FROM order_details od JOIN menu_items mi
    ON od.item_id = mi.menu_item_id
GROUP BY 1, 2, 3, 4, 5, 6;
```

Do a query to make interactive in Power BI.

Dashboard

```sql
-- DoD Growth
WITH cte1 AS (
SELECT order_date, SUM(price) total_sales
FROM order_details od JOIN menu_items mi
    ON od.item_id = mi.menu_item_id
GROUP BY 1
),
cte2 AS(
SELECT *,
    COALESCE(LAG(total_sales) OVER(ORDER BY order_date), total_sales) previous_sales
FROM cte1
)
SELECT
    *,
    CONCAT(ROUND((total_sales - previous_sales) / previous_sales * 100, 2), '%') growth_percentage
FROM cte2;
```

Do a query to make interactive in Power BI.

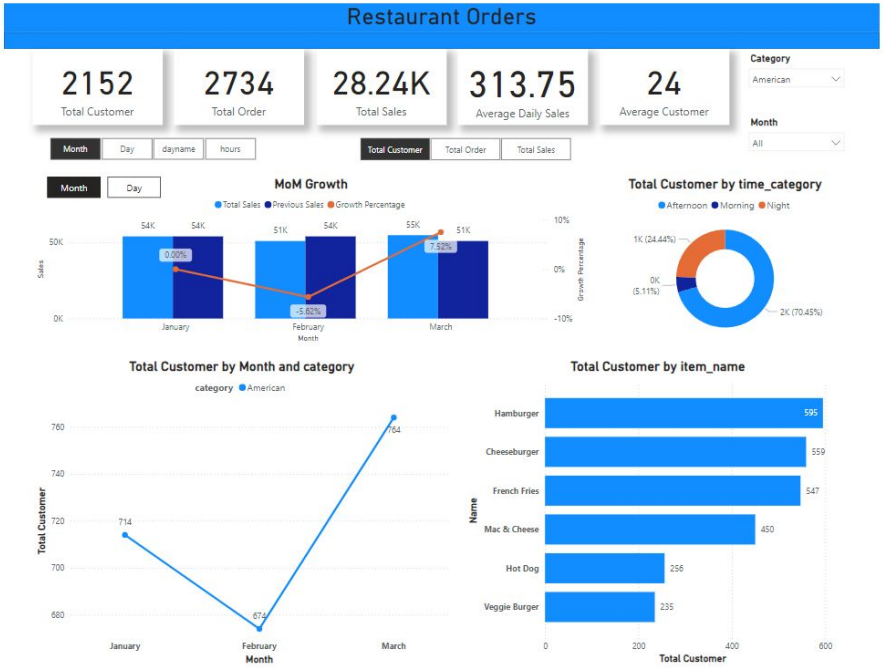
Dashboard

```sql
-- MoM Growth
WITH cteone AS (
SELECT DATE_FORMAT(order_date, '%Y-%m') month, SUM(price) total_sales
FROM order_details od JOIN menu_items mi

    ON od.item_id = mi.menu_item_id
GROUP BY 1
),
ctetwo AS(
SELECT *,
    COALESCE(LAG(total_sales) OVER(ORDER BY month), total_sales) previous_sales
FROM cteone
)
SELECT
    *,
    CONCAT(ROUND((total_sales - previous_sales) / previous_sales * 100, 2), '%') growth_percentage
FROM ctetwo
ORDER BY month;
```

Do a query to make interactive in Power BI.

Dashboard



Click Here : [Link Power BI]

Insights

Insights

- **March 2023** has the highest total sales compared to January and February.
- **12 pm** is the **most orders** and 10 am is the less orders.
- **Hamburger** is the **most order** with **622** orders and Chicken Tacos is the. least order with 123 orders.
- **Asian** and **Italian Category** have demonstrated the **best performance**.
- The **Average Daily Sales** is **$1769.09**.
- The **Average Customer** is **59** customer.

Recommendation

Recommendation

- We can **developed more menu items** within the **top 5** total orders such as **Hamburger, Edamame, Korean Beef Bowl, Cheeseburger, and French Fries**.
- We can focus on **developing more menu options** within the **Asian and Italian categories** to cater to customers who favor these cuisines.
- We can also implemented **special promotions**, particularly for **Asian and Italian menus**, such as **discounts** or **special menu deals**, to attract more customers.

Thank You