Created By: Divya Pardeshi

Date: 23-03-2024

Restaurant Order Analysis

Objective:

To Analyze order data to identify the most and least popular menu items and types of cuisine.

Task 1: Explore the items table

Your first objective is to better understand the items table by finding the number of rows in the table, the least and most expensive items, and the item prices within each category.

Q1) View the menu_items table and write a query to find the number of items on the menu

```
USE restaurant_db;

-- View the menu_items table.

SELECT * FROM menu_items;

-- Find the number of items on the menu.

SELECT COUNT(*) FROM menu_items;
```

Insights: The database contains 32 menu items.

Q2) What are the least and most expensive items on the menu?

```
-- Least Expensive Items

SELECT * FROM menu_items

ORDER BY price;

-- Most Expensive Items

SELECT * FROM menu_items

ORDER BY price DESC;
```

Insights:

a. The least expensive item is 'Edamame' priced at 5.00.
b. The most expensive item is 'Shrimp Scampi' priced at 19.95.

Q3) How many Italian dishes are on the menu? What are the least and most expensive Italian dishes on the menu?

```
SELECT COUNT(*) FROM menu_items
WHERE category = 'Italian';
```

Insights:

There are 9 Italian dishes available on the menu

Q4) What are the least and most expensive Italian dishes on the menu?

```
-- Least Expensive Italian Dishes
SELECT * FROM menu_items
WHERE category = 'Italian'
ORDER BY price;
-- Most Expensive Italian Dishes
SELECT * FROM menu_items
WHERE category = 'Italian'
ORDER BY price DESC;
```

Insights:

a. Least expensive Italian dish is ‘Spaghetti’ priced at '14.50'
b. Most Expensive Italian dish is 'Shrimp Scampi' priced at '19.95'

Q5) How many dishes are in each category?

```
SELECT category, COUNT(menu_item_id) as num_dishes
FROM menu_items
GROUP BY category;
```

Insights:

category	num_dishes
American	6
Asian	8
Mexican	9
Italian	9

Q6) What is the average dish price within each category?

```
SELECT category, AVG(Price) as avg_price
FROM menu_items
```

GROUP BY category;

Output:

category	avg_price
American	10.066667
Asian	13.475000
Mexican	11.800000
Italian	16.750000

Task 2: Explore the orders table

Your second objective is to better understand the orders table by finding the date range, the number of items within each order, and the orders with the highest number of items.

Q1) View the order_details table.

SELECT * FROM order_details;

Q2) What is the date range of the table?

SELECT * FROM order_details

ORDER BY order_date;

SELECT * FROM order_details

ORDER BY order_date DESC;

-- Alternative Way

SELECT MIN(order_date), MAX(order_date) FROM order_details;

Insights:

Orders were placed between ‘2023-01-01’ and '2023-03-31'

Q3) How many orders were made within this date range?

SELECT COUNT(DISTINCT order_id) FROM order_details;

Insights:

There were 5370 orders placed within the specified date range.

Q4) How many items were ordered within this date range?

```
SELECT COUNT(*) FROM order_details;
```

Insights:

'12234' items were ordered within the given date range

Q5) Which orders had the most number of items?

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

Insights:

Order ID 330 had the highest number of items, totaling 14 items.

Q6) How many orders had more than 12 items?

```
SELECT COUNT(*) FROM
(SELECT order_id, COUNT(item_id) AS num_items
FROM order_details
GROUP BY order_id
HAVING num_items > 12) AS num_orders;
```

Insights:

There were 20 orders with more than 12 items.

Task 3: Analyze customer behavior

Your final objective is to combine the items and orders tables, find the least and most ordered categories, and dive into the details of the highest spend orders.

Q1) Combine the menu_items and order_details tables into a single table

```
-- to view menu_items table
SELECT * FROM menu_items;
-- to view order_details table
```

```
SELECT * FROM order_details;


SELECT * FROM order_details od
LEFT JOIN menu_items mi
ON od.item_id = mi.menu_item_id
```

Q2) What were the least and most ordered items?

```
-- Least Ordered
SELECT item_name,COUNT(order_details_id) AS num_purchases
FROM order_details od
LEFT JOIN menu_items mi
ON od.item_id = mi.menu_item_id
GROUP BY item_name
ORDER BY num_purchases;


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

Insights:

a. Least Ordered: 'Chicken Tacos' with 123 purchases.
b. Most Ordered: 'Hamburger' with 622 purchases.

Q3) What categories were they in?

```
-- Least Ordered Categories
SELECT item_name,category, COUNT(order_details_id) AS num_purchases
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;

-- Most Ordered Categories
SELECT item_name,category, COUNT(order_details_id) AS num_purchases
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;
```

Insights:

a. Least ordered category is 'Mexican'
b. Most ordered category is 'American'

Q4) What were the top 5 orders that spent the most money?

```
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;
```

Output:

Order_id	Total_spend
440	192.15
2075	191.05
1957	190.10
330	189.70
2675	185.10

Q5) View the details of the highest spend order. Which specific items were purchased?

```
SELECT category, COUNT(item_id) AS num_items
FROM order_details od
LEFT JOIN menu_items mi
ON od.item_id = mi.menu_item_id
WHERE order_id = 440
GROUP BY category;
```

Output:

category	num_items
Mexican	2
American	2
Italian	8
Asian	2

Q6) View the details of the top 5 highest spend orders

```
SELECT order_id, category, COUNT(item_id) AS num_items
FROM order_details od
LEFT JOIN menu_items mi
ON od.item_id = mi.menu_item_id
WHERE order_id IN (440,2075,1957,330,2675)
GROUP BY order_id, category;
```