zomato

Data Analysis

Using MySQL

Presented by:

Md Saif Ali Khan

Introduction

In an era dominated by technology, the food industry has undergone a significant transformation with the advent of online food delivery platforms. Zomato, a prominent player in this space, has not only revolutionized the way people discover and explore restaurants but has also redefined the dynamics of the food industry.

The Zomato Data Analysis project leverages a simulated dataset inspired by the Zomato platform, presenting an opportunity to delve into the intricate details of user interactions, restaurant performances, and operational dynamics using advanced MySQL queries.

Project Objectives

Understand the structure of the dataset, including information about users, restaurants, menu items, orders, reviews, delivery status, complaints, payment transactions, delivery personnel, and order deliveries.

- Investigate user behavior, including the frequency of orders, spending patterns, and relationships with multiple restaurants.

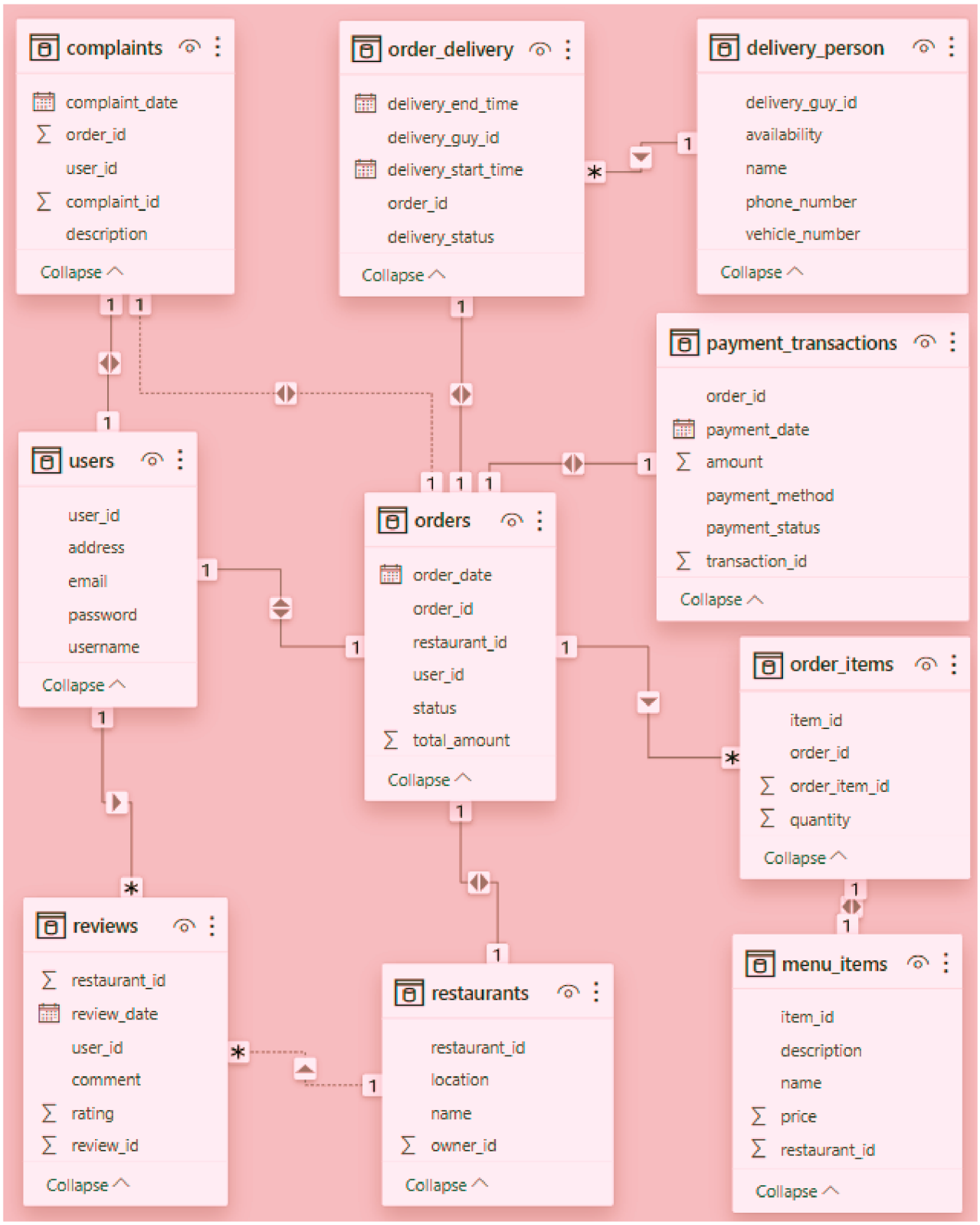
- Explore restaurant performance metrics, such as average ratings, total revenue, and common customer complaints.

- Examine the efficiency and reliability of the delivery system, identifying top-performing delivery personnel and analyzing delivery times.

List Of Tables in the Database

- Users table
- Restaurants table
- Menu items table
- Orders table
- Order items table (to store items in each order)
- Reviews table
- Complaints table
- Payment transactions table
- Order delivery table
- Delivery person table

Data Model


complaints
complaint_date
order_id
user_id
complaint_id
description
Collapse
order_delivery
delivery_end_time
delivery_guy_id
delivery_start_time
order_id
delivery_status
Collapse
delivery_person
delivery_guy_id
availability
name
phone_number
vehicle_number
Collapse
payment_transactions
order_id
payment_date
amount
payment_method
payment_status
transaction_id
Collapse
users
user_id
address
email
password
username
Collapse
orders
order_date
order_id
restaurant_id
user_id
status
total_amount
Collapse
order_items
item_id
order_id
order_item_id
quantity
Collapse
reviews
restaurant_id
review_date
user_id
comment
rating
review_id
Collapse
restaurants
restaurant_id
location
name
owner_id
Collapse
menu_items
item_id
description
name
price
restaurant_id
Collapse

QUERY - 1

Find the top 5 restaurants with the highest average customer ratings

```
(SELECT
    res.name as restaurant_name,
    ROUND(AVG(rev.rating), 1) AS avg_rating
FROM
    restaurants res
        JOIN
    reviews rev ON res.restaurant_id = rev.restaurant_id
GROUP BY res.name
ORDER BY avg_rating DESC
LIMIT 5);
```

Result Table

restaurant_name	avg_rating
Delhi Diner	4.5
Hyderabad House	4.0
Kolkata Bites	4.4
Kolkata Kitchens	4.0
Mumbai Masala	4.0

QUERY - 2

List the top 5 users who have placed the most orders

```
(SELECT
    u.username as user_name, COUNT(o.order_id) AS total_orders
FROM
    users u
        JOIN
    orders o ON u.user_id = o.user_id
GROUP BY u.username
ORDER BY total_orders DESC
LIMIT 5);
```

Result Table

user_name	total_orders
VikasGupta	5
AnitaChopra	4
AmitPatel	4
NehaSingh	4
KarthikIyer	4

QUERY - 3

Find the total revenue generated by each restaurant

```
(SELECT
    r.name AS restaurant_name, SUM(p.amount) AS total_revenue
FROM
    restaurants r
        LEFT JOIN
    orders o ON r.restaurant_id = o.restaurant_id
        LEFT JOIN
    payment_transactions p ON o.order_id = p.order_id
WHERE
    p.payment_status = 'Paid'
GROUP BY r.name
ORDER BY total_revenue DESC);
```

Result Table

restaurant_name	total_revenue
Jaipur Rasoi	3450.00
Kolkata Bites	3363.00
Delhi Delight	2744.00
Gujarati Swad	2527.00
South Indian Treat	2203.00
Spice Haven	1874.00
Hyderabad House	1140.00
Punjabi Dhaba	1061.00
Delhi Darbar	600.00
Agra Sweets	459.00

QUERY - 4

Identify the most ordered menu item across all restaurants

```
(SELECT
    m.name AS item_name, SUM(o.quantity) AS total_orders
FROM
    menu_items m
        JOIN
    order_items o ON m.item_id = o.item_id
GROUP BY m.name
ORDER BY total_orders DESC
LIMIT 1);
```

Result Table

item_name	total_orders
Paneer Tikka	82

QUERY - 5

Find the percentage of orders that were paid using online methods

```
(WITH online AS (
    SELECT COUNT(*) AS online_payments
    FROM payment_transactions
    WHERE payment_method IN ('Online Banking', 'Online Wallet')
)
SELECT
    ROUND((online.online_payments / (SELECT COUNT(*) FROM payment_transactions)) * 100, 2)
    AS online_payment_percentage
FROM online);
```

Result Table

online_payment_percentage
34.29

QUERY - 6

Determine the average time taken for order delivery for each restaurant

```
(SELECT
    r.name as restaurant_name,
    SEC_TO_TIME(AVG(TIME_TO_SEC(TIMEDIFF(od.delivery_end_time,
                            od.delivery_start_time)))) AS avg_delivery_time
FROM
    order_delivery od
        JOIN
    orders o ON od.order_id = o.order_id
        JOIN
    restaurants r ON o.restaurant_id = r.restaurant_id
WHERE
    od.delivery_status = 'Delivered'
GROUP BY r.name
HAVING avg_delivery_time IS NOT NULL
ORDER BY avg_delivery_time);
```

Result Table

restaurant_name	avg_delivery_time
Punjabi Dhaba	00:14:00.0000
South Indian Treat	00:25:30.0000
Jaipur Rasoi	00:31:20.0000
Delhi Delight	00:36:20.0000
Gujarati Swad	00:41:00.0000
Kolkata Bites	00:49:00.0000

QUERY - 7

List the top 3 delivery persons with the highest number of successful deliveries

```
(SELECT
    dp.name AS delivery_person,
    COUNT(od.delivery_status) AS successful_deliveries
FROM
    delivery_person dp
        JOIN
    order_delivery od ON dp.delivery_guy_id = od.delivery_guy_id
WHERE
    od.delivery_status = 'Delivered'
GROUP BY delivery_person
ORDER BY successful_deliveries DESC
LIMIT 3);
```

Result Table

delivery_person	successful_deliveries
Suresh Kumar	7
Meenakshi Yadav	7
Geeta Sharma	4

QUERY - 8

Based on keywords, find the 3 most common complaint type

```
(SELECT
    SUBSTRING_INDEX(description, ' ', 2) AS common_complaint_type
FROM
    complaints
GROUP BY common_complaint_type
ORDER BY COUNT(*) DESC
LIMIT 3);
```

Result Table

common_complaint_type
Unprofessional behavior
Late delivery
Received wrong

QUERY - 9

Find the users who have ordered from the same restaurant multiple times

```
(SELECT
    u.username AS user,
    r.name AS restaurant_name,
    COUNT(o.order_id) AS total_orders
FROM
    users u
        JOIN
    orders o ON u.user_id = o.user_id
        JOIN
    restaurants r ON o.restaurant_id = r.restaurant_id
GROUP BY u.username , r.name
HAVING total_orders > 1);
```

Result Table

user	restaurant_name	total_orders
AmitPatel	Punjabi Dhaba	2
KarthikIyer	Jaipur Rasoi	3

QUERY - 10

Determine the percentage of orders that experienced complaints

```sql
SELECT
    ROUND((COUNT(c.order_id) / COUNT(o.order_id)) * 100,
            2) AS complaint_percentage
FROM
    orders o
        LEFT JOIN
    complaints c ON o.order_id = c.order_id;
```

Result Table

complaint_percentage
38.46

QUERY - 11

List the restaurants that have received complaints related to late deliveries

```
(SELECT DISTINCT
    r.name AS restaurant_name
FROM
    restaurants r
        JOIN
    orders o ON r.restaurant_id = o.restaurant_id
        JOIN
    order_delivery od ON o.order_id = od.order_id
        JOIN
    complaints c ON o.order_id = c.order_id
WHERE
    c.description LIKE '%delayed%'
        OR c.description LIKE '%late%'
        OR c.description LIKE '%delay%');
```

Result Table

restaurant_name
Spice Haven
Jaipur Rasoi
South Indian Treat

QUERY - 12

Determine the percentage contribution of each order item to the total order amount

```
(SELECT
  oi.order_id,
  m.item_id,
  oi.quantity,
  m.price,
  ROUND((m.price * oi.quantity / SUM(m.price * oi.quantity)
  OVER (PARTITION BY oi.order_id)) * 100, 2) AS percentage_contribution
FROM
    order_items oi
        JOIN
            menu_items m on oi.item_id = m.item_id);
```

Result Table

order_id	item_id	quantity	price	percentage_contribution
20	8	4	280.00	30.27
20	10	4	200.00	21.62
20	16	5	180.00	24.32
20	6	4	220.00	23.78
21	17	6	320.00	70.59
21	19	4	150.00	22.06
21	5	2	100.00	7.35

QUERY - 13

Find users who have placed more orders than the average number of orders

```
(SELECT user_id, username, orders_placed_by_user, avg_orders
FROM (
    SELECT
        u.user_id, u.username,
        COUNT(o.order_id) AS orders_placed_by_user,
        AVG(COUNT(o.order_id)) OVER () AS avg_orders
    FROM
        users u
    LEFT JOIN
        orders o ON u.user_id = o.user_id
    GROUP BY
        u.user_id, u.username
) AS user_orders
WHERE
    orders_placed_by_user > avg_orders
ORDER BY orders_placed_by_user DESC);
```

Result Table

user_id	username	orders_placed_by_user	avg_orders
7	VikasGupta	5	2.5000
3	AmitPatel	4	2.5000
4	NehaSingh	4	2.5000
8	AnitaChopra	4	2.5000
15	KarthikIyer	4	2.5000
11	RajeshKumar	3	2.5000
17	RohanMehra	3	2.5000
18	AnanyaChoudhury	3	2.5000

QUERY - 14

Identify users who have placed orders from restaurants with average ratings above 4.0

```
(SELECT
    user_id, username
FROM
    users
WHERE
    user_id IN (SELECT DISTINCT
            u.user_id
        FROM
            users u
                JOIN
            orders o ON u.user_id = o.user_id
                JOIN
            restaurants r ON o.restaurant_id = r.restaurant_id
        WHERE
            r.restaurant_id IN (SELECT
                    restaurant_id
                FROM
                    reviews
                GROUP BY restaurant_id
                HAVING AVG(rating) > 4.0)));
```

Result Table

user_id	username
1	RahulKumar
4	NehaSingh
8	AnitaChopra
18	AnanyaChoudhury

QUERY - 15

Calculate the total number of orders placed on each day of the week

```
(WITH OrderDayOfWeek AS (
  SELECT
    order_id,
    DAYOFWEEK(order_date) AS day_of_week
  FROM orders
)
SELECT
  day_of_week,
  COUNT(order_id) AS order_count
FROM OrderDayOfWeek
GROUP BY day_of_week order by day_of_week);
```

Result Table

day_of_week	order_count
1	6
2	7
3	5
4	6
5	13
6	7
7	6

THE END