Zomato Food Delivery

**Zomato Food Delivery Data Insights**

## **1. Project Overview**

As a data scientist at Zomato, the goal of this project is to enhance operational efficiency and improve customer satisfaction by analyzing food delivery data. The interactive Streamlit tool enables seamless data entry and management of orders, customers, restaurants, and deliveries. The system supports robust database operations like adding columns or creating new tables dynamically while maintaining compatibility with existing code.

## **2. Source Code**

This project consists of multiple Python scripts for dataset generation, database management, and Streamlit app development. The key components are:

**a. Dataset Generation (datasetGenerator.py)**

* Generates synthetic food delivery data.
* Exports data in CSV format for initial database seeding.

**b. Database Management (DBConnection.py)**

* Establishes a connection to an MySQL Database.
* Provides functions to create, update, and query tables dynamically.
* Ensures compatibility with existing schema.

**c. Streamlit App (ZomatoStreamlit.py)**

* Interactive interface for data entry and real-time analysis.
* Allows users to insert, update, and view records.
* Supports on-the-fly schema modifications.

## **3. Streamlit App**

The **Streamlit** app is the front-end of this tool, designed to provide a user-friendly interface for managing and analyzing food delivery data.

**Key Features:**

* **Data Entry:** Users can add new records for orders, customers, restaurants, and deliveries.
* **Dynamic Table Management:** Ability to create new tables or add columns dynamically.
* **Visual Analytics:** Data visualization through interactive charts.
* **SQL Query Execution:** Run custom SQL queries for deeper insights.

## **4. Database Schema**

The SQL database includes the following tables:

a. tbl\_customers

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| **customer\_id** | **INT** | **Auto-increment primary key for each customer.** |
| **name** | **VARCHAR(255)** | **Name of the customer.** |
| **email** | **VARCHAR(255)** | **Email address of the customer, must be unique.** |
| **phone** | **VARCHAR(20)** | **Phone number of the customer, must be unique.** |
| **location** | **TEXT** | **Customer's location details.** |
| **signup\_date** | **DATE** | **The date when the customer signed up.** |
| **is\_premium** | **BOOLEAN** | **Whether the customer is a premium member (true/false).** |
| **preferred\_cuisine** | **VARCHAR(100)** | **Preferred cuisine of the customer (e.g., Italian, Chinese).** |
| **total\_orders** | **INT** | **Total number of orders placed by the customer.** |
| **average\_rating** | **DECIMAL(3,2)** | **Average rating given by the customer, ranging from 0.00 to 5.00.** |

b. tbl\_delivery\_persons

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| **delivery\_person\_id** | **INT** | **Auto-increment primary key for each delivery person.** |
| **name** | **VARCHAR(255)** | **Name of the delivery person.** |
| **contact\_number** | **VARCHAR(20)** | **Contact number of the delivery person, must be unique.** |
| **vehicle\_type** | **ENUM('Bike','Scooter','Motorcycle','Electric Bike','Electric Scooter','Moped')** | **Type of vehicle used by the delivery person.** |
| **total\_deliveries** | **INT** | **Total number of deliveries made by the delivery person, must be non-negative.** |
| **average\_rating** | **DECIMAL(2,1)** | **Average rating of the delivery person, ranging from 1.0 to 5.0.** |
| **location** | **TEXT** | **Delivery person's location details.** |

c. tbl\_order\_details

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| **order\_id** | **INT** | **Auto-increment primary key for each order.** |
| **customer\_id** | **INT** | **Foreign key referencing the customer\_id from the tbl\_customers table.** |
| **restaurant\_id** | **INT** | **Foreign key referencing the restaurant\_id from the tbl\_restaurant table.** |
| **order\_date** | **DATETIME** | **The date and time when the order was placed.** |
| **delivery\_time** | **DATETIME** | **The date and time when the delivery is completed.** |
| **status** | **ENUM('Pending', 'Delivered', 'Cancelled')** | **Status of the order (e.g., Pending, Delivered, Cancelled).** |
| **total\_amount** | **DECIMAL(10,2)** | **Total amount for the order.** |
| **payment\_mode** | **ENUM('Credit Card', 'Cash', 'UPI')** | **Mode of payment (e.g., Credit Card, Cash, UPI).** |
| **discount\_applied** | **DECIMAL(10,2)** | **The discount applied to the order, if any.** |
| **feedback\_rating** | **DECIMAL(2,1)** | **Feedback rating for the order, ranging from 1.0 to 5.0.** |

d. tbl\_deliveries

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| **delivery\_id** | **INT** | **Auto-increment primary key for each delivery record.** |
| **order\_id** | **INT** | **Foreign key referencing the order\_id from the tbl\_order\_details table.** |
| **delivery\_person\_id** | **INT** | **Foreign key referencing the delivery\_person\_id from the tbl\_delivery\_persons table.** |
| **delivery\_status** | **ENUM('Pending', 'Delivered', 'Cancelled')** | **Status of the delivery (e.g., On the way, Delivered, or Failed).** |
| **distance** | **DECIMAL(5,2)** | **Distance traveled for the delivery, must be a positive value.** |
| **delivery\_time** | **INT** | **Delivery time in minutes, must be a positive value.** |
| **estimated\_time** | **INT** | **Estimated delivery time in minutes, must be a positive value.** |
| **delivery\_fee** | **DECIMAL(10,2)** | **Delivery fee charged for the delivery, must be a positive value.** |
| **vehicle\_type** | **ENUM('Bike','Scooter','Motorcycle','Electric Bike','Electric Scooter','Moped')** | **Type of vehicle used for delivery** |

**e.** **tbl\_restaurant**

| **Column Name** | **Data Type** | **Description** |
| --- | --- | --- |
| **restaurant\_id** | **INT** | **Auto-increment primary key for each restaurant.** |
| **name** | **VARCHAR(255)** | **Name of the restaurant.** |
| **cuisine\_type** | **VARCHAR(100)** | **Type of cuisine served at the restaurant (e.g., Italian, Chinese).** |
| **location** | **TEXT** | **Location details of the restaurant.** |
| **owner\_name** | **VARCHAR(255)** | **Name of the restaurant owner.** |
| **average\_delivery\_time** | **INT** | **Average time taken for delivery from the restaurant, must be a non-negative value.** |
| **contact\_number** | **VARCHAR(20)** | **Contact number for the restaurant, must be unique.** |
| **rating** | **DECIMAL(2,1)** | **Rating of the restaurant, ranging from 1.0 to 5.0.** |
| **total\_orders** | **INT** | **Total number of orders placed at the restaurant.** |
| **is\_active** | **BOOLEAN** | **Whether the restaurant is currently active (true/false).** |

## 

## **5. SQL Queries for Data Analysis**

Below are 20 SQL queries to analyze food delivery trends:

1. Total number of orders placed:

SELECT COUNT(\*) FROM tbl\_order\_details;

1. Most popular restaurant:

SELECT restaurant\_id, COUNT(\*) as order\_count FROM tbl\_order\_details GROUP BY restaurant\_id ORDER BY order\_count DESC LIMIT 1;

1. Average order value:

SELECT AVG(total\_amount) From tbl\_order\_details;

1. Count of customers who placed more than 5 orders:

SELECT customer\_id, COUNT(\*) From tbl\_order\_details GROUP BY customer\_id HAVING COUNT(\*) > 5;

1. Number of pending orders:

SELECT COUNT(\*) From tbl\_order\_details WHERE status = 'Pending';

1. Orders by cuisine type:

SELECT r.cuisine\_type, COUNT(o.order\_id) From tbl\_order\_details o JOIN tbl\_restaurant r ON o.restaurant\_id = r.restaurant\_id GROUP BY r.cuisine\_type;

1. Top 5 customers based on total spending:

SELECT customer\_id, SUM(total\_amount) AS total\_spent From tbl\_order\_details GROUP BY customer\_id ORDER BY total\_spent DESC LIMIT 5;

1. Orders completed within 30 minutes:

SELECT COUNT(\*) FROM tbl\_deliveries WHERE delivery\_time - order\_date <= INTERVAL '30 MINUTES';

1. Percentage of completed deliveries:

SELECT (COUNT(\*) FILTER(WHERE delivery\_status = 'Completed') \* 100.0) / COUNT(\*) AS completion\_rate FROM tbl\_deliveries;

1. Average delivery time:

SELECT AVG(EXTRACT(EPOCH FROM delivery\_time - order\_date) / 60) AS avg\_delivery\_time FROM tbl\_deliveries;

1. Restaurant with the highest revenue:

SELECT restaurant\_id, SUM(total\_amount) From tbl\_order\_details GROUP BY restaurant\_id ORDER BY SUM(total\_amount) DESC LIMIT 1;

1. Orders by month:

SELECT DATE\_TRUNC('month', order\_date) AS month, COUNT(\*) From tbl\_order\_details GROUP BY month;

1. Customer retention rate:

SELECT COUNT(DISTINCT customer\_id) \* 100.0 / (SELECT COUNT(\*) FROM tbl\_customers) From tbl\_order\_details;

1. Percentage of orders that include delivery:

SELECT (COUNT(DISTINCT order\_id) \* 100.0) / (SELECT COUNT(\*) From tbl\_order\_details) FROM tbl\_deliveries;

1. Restaurant with the most late deliveries:

SELECT restaurant\_id, COUNT(\*) FROM tbl\_deliveries WHERE delivery\_status = 'Pending' GROUP BY restaurant\_id ORDER BY COUNT(\*) DESC LIMIT 1;

1. Average number of items per order:

SELECT AVG(item\_count) From tbl\_order\_details;

1. Peak order hours:

SELECT EXTRACT(HOUR FROM order\_date) AS hour, COUNT(\*) From tbl\_order\_details GROUP BY hour ORDER BY COUNT(\*) DESC;

1. Revenue per customer:

SELECT customer\_id, SUM(total\_amount) From tbl\_order\_details GROUP BY customer\_id;

1. Most frequently ordered item:

SELECT cuisine\_type, COUNT(\*) FROM tbl\_restaurant GROUP BY item\_name ORDER BY COUNT(\*) DESC LIMIT 1;

1. Order frequency per restaurant:

SELECT restaurant\_id, COUNT(\*) From tbl\_order\_details GROUP BY restaurant\_id ORDER BY COUNT(\*) DESC;

## **6. Instructions to Run the Project**

**Prerequisites**

* Install **Python 3.8+**
* Install dependencies using:

pip install streamlit pandas numpy sqlalchemy

* Set up the database using the schema provided.

**Run the Streamlit App**

streamlit run app.py

## **Conclusion**

This tool empowers Zomato’s team with efficient food delivery data management, real-time analytics, and enhanced customer insights.