Swiggy Data Analysis Project

SQL-Based Business Insights Report

Name: Pankaj Raju Badipahadi
Role Targeted: Business Analyst
Tools Used: SQL (MySQL)
Project Type: Data Analysis & Business Reporting

📌 Project Overview

This project analyzes Swiggy food delivery data using SQL to extract meaningful business insights related to revenue trends, customer behavior, and restaurant performance. The objective was to support data-driven decision-making and operational optimization.

--A] Data Cleaning & Validation

--1) Null Check

```
SELECT
        SUM(CASE WHEN state IS NULL THEN 1 ELSE 0 END) AS null_state,
        SUM(CASE WHEN city IS NULL THEN 1 ELSE 0 END) AS null_city,
        SUM(CASE WHEN order_date IS NULL THEN 1 ELSE 0 END) AS null_order_date,
        SUM(CASE WHEN restaurant_name IS NULL THEN 1 ELSE 0 END) AS null_restaurant,
        SUM(CASE WHEN location IS NULL THEN 1 ELSE 0 END) AS null_location,
        SUM(CASE WHEN category IS NULL THEN 1 ELSE 0 END) AS null_category,
        SUM(CASE WHEN dish_name IS NULL THEN 1 ELSE 0 END) AS null_dish,
        SUM(CASE WHEN price_inr IS NULL THEN 1 ELSE 0 END) AS null_price,
        SUM(CASE WHEN rating IS NULL THEN 1 ELSE 0 END) AS null_rating,
        SUM(CASE WHEN rating_Count IS NULL THEN 1 ELSE 0 END) AS null_rating_count
FROM swiggy_data;
```

--2) Blank & Empty String

```
SELECT *
from swiggy_data
where
state='' OR state IS NULL
OR city='' OR city IS NULL
OR restaurant_name='' OR restaurant_name IS NULL
OR location='' OR location IS NULL
OR category='' OR category IS NULL
OR dish_name='' OR dish_name IS NULL
OR price_inr IS NULL
OR rating IS NULL
OR rating_count IS NULL;
```

--3) Duplicate Detection

SELECT state, city, order_date, restaurant_name, location, category,

dish_name, price_inr, rating, rating_count, count(*) as CNT

from swiggy_data

Group By state, city, order_date, restaurant_name, location, category,

dish_name, price_inr, rating, rating_count

Having count(*)>1;

--4) Delete Duplication

WITH cte AS(

SELECT ctid, ROW_NUMBER() Over(

       PARTITION BY state, city, order_date, restaurant_name, location, category,

dish_name, price_inr, rating, rating_count

ORDER BY ctid

) AS rn

FROM swiggy_data

)

DELETE FROM swiggy_data

WHERE ctid IN(

select ctid

from cte

where rn>1

);

---B] Creating Schema

--Creating Dimension Tables

--1) Date Table

SELECT * from dim_date;

Create Table dim_date(

       date_id SERIAL Primary Key,

       full_date DATE,

       year INT,

       month INT,

       month_name varchar(20),

```
        quarter INT,

        day INT,

        week INT

)
```

--2) Location Table

```
SELECT * from dim_location;


Create Table dim_location(

        location_id SERIAL Primary Key,

        state VARCHAR(100),

        city VARCHAR(100),

        location VARCHAR(100)

)
```

--3) Restaurant Table

```
SELECT * from dim_restaurant;


Create Table dim_restaurant(

        restaurant_id SERIAL Primary KEY,

        restaurant_name VARCHAR(200)

)
```

--4) Category Table

```
SELECT * from dim_category;


Create Table dim_category(

        category_id SERIAL Primary KEY,

        category VARCHAR(200)

)
```

--5) Dish Table

```
SELECT * from dim_dish;


Create Table dim_dish(
```

```
        dish_id SERIAL Primary Key,

        dish_name VARCHAR(200)
)
```

---C] Creating Fact

--Creating Fact Tables

--1) Swiggy Orders

```
SELECT * from fact_swiggy_orders;


Create Table fact_swiggy_orders(

        order_id Serial Primary Key,


        date_id Serial,

        price_inr Decimal(10,2),

        rating Decimal(4,2),

        rating_count INT,


        location_id Serial,

        restaurant_id Serial,

        category_id Serial,

        dish_id Serial,


        FOREIGN KEY (date_id) REFERENCES dim_date(date_id),

        FOREIGN KEY (location_id) REFERENCES dim_location(location_id),

        FOREIGN KEY (restaurant_id) REFERENCES dim_restaurant(restaurant_id),

        FOREIGN KEY (category_id) REFERENCES dim_category(category_id),

        FOREIGN KEY (dish_id) REFERENCES dim_dish(dish_id)


)
```

--D] Insert Data in Tables

--1) dim_date

```
SELECT  * from dim_date;


INSERT INTO dim_date(full_date, year, month, month_name, quarter, day, week)
SELECT DISTINCT

        order_date,
```

```sql
        EXTRACT(YEAR FROM order_date) AS year,
    EXTRACT(MONTH FROM order_date) AS month,
    TO_CHAR(order_date, 'Month') AS month_name,
    EXTRACT(QUARTER FROM order_date) AS quarter,
    EXTRACT(DAY FROM order_date) AS day,
    EXTRACT(WEEK FROM order_date) AS week
from swiggy_data
WHERE order_date IS NOT NULL;
```

--2) dim_location

```sql
SELECT * from dim_location;


INSERT INTO dim_location (state, city, location)
SELECT DISTINCT
        state,
        city,
        location
from swiggy_data;
```

--3) dim_restaurant

```sql
SELECT * from dim_restaurant;


INSERT INTO dim_restaurant(restaurant_name)
SELECT DISTINCT
        restaurant_name
from swiggy_data;
```

--4) dim_category

```sql
SELECT * from dim_category;


INSERT INTO dim_category(category)
SELECT DISTINCT
        category
from swiggy_data;
```

--5) dim_dish

```sql
SELECT * from dim_dish;


INSERT INTO dim_dish(dish_name)
SELECT DISTINCT
        dish_name
from swiggy_data;
```

--6) fact_swiggy_order

```sql
SELECT * from fact_swiggy_orders;


INSERT INTO fact_swiggy_orders(
        date_id,
        price_inr,
        rating,
        rating_count,
        location_id,
        restaurant_id,
        category_id,
        dish_id
)
SELECT
        dd.date_id,
        s.price_inr,
        s.rating,
        s.rating_count,

        dl.location_id,
        dr.restaurant_id,
        dc.category_id,
        dsh.dish_id
from swiggy_data s

JOIN dim_date dd
        ON dd.full_date = s.order_date

JOIN dim_location dl
        ON dl.state = s.state
```

```
            AND dl.city = s.city

            AND dl.location = s.location


JOIN dim_restaurant dr

            ON dr.restaurant_name = s.restaurant_name


JOIN dim_category dc

            ON dc.category = s.category


JOIN dim_dish dsh

            ON dsh.dish_name = s.dish_name;
```

-- COMPLETE TABLE

```
SELECT * FROM fact_swiggy_orders f
JOIN dim_date d ON f.date_id = d.date_id
JOIN dim_location l ON f.location_id = l.location_id
JOIN dim_restaurant r ON f.restaurant_id = r.restaurant_id
JOIN dim_category c ON f.category_id = c.category_id
JOIN dim_dish di ON f.dish_id = di.dish_id;
```

--E] KPI's

--1) Total Orders

```
SELECT COUNT(*) AS total_order
from fact_swiggy_orders;
```

--2) Total Revenue (INR Million)

```
SELECT
    TO_CHAR(SUM(price_inr) / 1000000.0, 'FM999999990.00') || ' INR Million'
    AS total_revenue
FROM fact_swiggy_orders;
```

--3) Average Dish Price

```
SELECT
    TO_CHAR(AVG(price_inr), 'FM999999990.00') || 'INR'
```

```
    AS average_price
FROM fact_swiggy_orders;
```

--4) Average Rating

```
SELECT AVG(rating) AS avg_rating
from fact_swiggy_orders;
```

--F] Deep Dive Business Analysis

--1) Monthly Order Trends

```
SELECT
d.year,
d.month,
d.month_name,
COUNT(*) AS total_orders
from fact_swiggy_orders f
JOIN dim_date d ON f.date_id = d.date_id
Group By d.year,
d.month,
d.month_name;
```

--2) Total Revenue

```
SELECT
d.year,
d.month,
d.month_name,
SUM(price_inr) AS total_revenue
from fact_swiggy_orders f
JOIN dim_date d ON f.date_id = d.date_id
Group By d.year,
d.month,
d.month_name
ORDER By SUM(price_inr) DESC;
```

--3) Quaterly Trend

```
SELECT
```

```sql
d.year,

d.quarter,

COUNT(*) AS total_orders

from fact_swiggy_orders f

JOIN dim_date d ON f.date_id = d.date_id

Group By d.year,

d.quarter;
```

--4) Yearly Trend
```sql
SELECT

d.year,

COUNT(*) AS total_orders

from fact_swiggy_orders f

JOIN dim_date d ON f.date_id = d.date_id

Group By d.year;
```

--5) Order by day of week (MON-SUN)
```sql
SELECT

    TO_CHAR(d.full_date, 'Day') AS day_name,

    COUNT(*) AS total_orders

FROM fact_swiggy_orders f

JOIN dim_date d

    ON f.date_id = d.date_id

GROUP BY

    TO_CHAR(d.full_date, 'Day'),

    EXTRACT(DOW FROM d.full_date)

ORDER BY

    EXTRACT(DOW FROM d.full_date);
```

--6) Top 10 Cities by order volume
```sql
SELECT

l.city,

COUNT(*) AS total_orders

FROM fact_swiggy_orders f

JOIN dim_location l

ON l.location_id = f.location_id
```

```
Group By l.city
Order By Count(*) DESC
LIMIT 10;
```

--7) Top 10 cities by Revenue

```
SELECT
l.city,
SUM(f.price_inr) AS total_revenue
FROM fact_swiggy_orders f
JOIN dim_location l
ON l.location_id = f.location_id
Group By l.city
Order By SUM(f.price_inr) DESC
LIMIT 10;
```

--8) Revenue contribution by states

```
SELECT
l.state,
SUM(f.price_inr) AS total_revenue
FROM fact_swiggy_orders f
JOIN dim_location l
ON l.location_id = f.location_id
Group By l.state
Order By SUM(f.price_inr) DESC;
```

--8) Orders contribution by states

```
SELECT
l.state,
COUNT(*) AS total_orders
FROM fact_swiggy_orders f
JOIN dim_location l
ON l.location_id = f.location_id
Group By l.state
Order By COUNT(*) DESC;
```

--9) Top 10 Restaurant by orders

SELECT

r.restaurant_name,

COUNT(*) AS total_orders

FROM fact_swiggy_orders f

JOIN dim_restaurant r

ON r.restaurant_id = f.restaurant_id

Group By r.restaurant_name

Order By Count(*) DESC

LIMIT 10;

--10) Top Categories by orders values

SELECT

  c.category,

  COUNT(*) AS total_orders

FROM fact_swiggy_orders f

JOIN dim_category c ON f.category_id = c.category_id

GROUP By c.category

Order By total_orders DESC;

--11) Most Ordered Dish

SELECT

  d.dish_name,

  COUNT(*) AS order_count

FROM fact_swiggy_orders f

JOIN dim_dish d ON f.dish_id = d.dish_id

Group By d.dish_name

Order By order_count DESC;

--12) Cuisine performance (Orders + AVG Rating)

SELECT

 c.category,

 COUNT(*) AS total_orders,

 AVG(f.rating) AS avg_rating

FROM fact_swiggy_orders f

JOIN dim_category c

```
    ON f.category_id = c.category_id
GROUP BY
    c.category
ORDER BY
    total_orders DESC;
```

--13) Total Orders By Price Range

```
SELECT
    CASE
        WHEN price_inr < 100 THEN 'Under 100'
        WHEN price_inr BETWEEN 100 AND 199 THEN '100 - 199'
        WHEN price_inr BETWEEN 200 AND 299 THEN '200 - 299'
        WHEN price_inr BETWEEN 300 AND 399 THEN '300 - 499'
        ELSE '500+'
    END AS price_range,
    COUNT(*) AS total_orders
FROM fact_swiggy_orders
GROUP BY
    CASE
        WHEN price_inr < 100 THEN 'Under 100'
        WHEN price_inr BETWEEN 100 AND 199 THEN '100 - 199'
        WHEN price_inr BETWEEN 200 AND 299 THEN '200 - 299'
        WHEN price_inr BETWEEN 300 AND 399 THEN '300 - 499'
        ELSE '500+'
    END
ORDER BY total_orders DESC;
```

--14) Rating Count Distribution (1-5)

```
SELECT
        rating,
        Count(*) AS rating_count
FROM fact_swiggy_orders
Group By rating
Order By COUNT(*) DESC;
```