Swiggy Sales Performance Analysis Project Report

(SQL + BI project)

By: Harshal Sharma

Tools Used: SQL (SSMS) and Microsoft Power BI

Resume Project - Data Analyst Portfolio

Project Description:

This project focuses on analyzing Swiggy's order and sales data using SQL-based data modeling and Power BI dashboards to derive actionable business insights and key performance indicators (KPIs).

Tools & Technologies Used:

- **Database:** SQL Server SSMS
- **Query Language:** SQL
- **Data Modeling:** Star Schema (Fact & Dimension Tables)
- **ETL:** SQL (Data Cleaning, Transformation, Insertion)
- **Visualization Tool:** Power BI
- **Measures Language:** DAX

1. Project Overview

“This project analyzes Swiggy sales data using **SQL Server (SSMS)** for data cleaning, KPI development, and business analysis, with insights visualized in Power BI.”

The objective was to clean raw transactional data, design a **star schema**, and calculate **key performance indicators (KPIs)** that help understand customer behaviour , revenue trends, food preferences, and regional performance.

The final output of this project is a **clean analytics-ready dataset** and **business KPIs** that can directly power BI dashboards for decision-making.

2. Business Problem Statement

Food delivery platforms like Swiggy need answers to questions such as:

- Which cities and states generate the highest revenue?
- What price ranges customers prefer?
- Which restaurants, dishes, and cuisines perform best?
- How do orders and revenue change over time?
- How do ratings impact food performance?

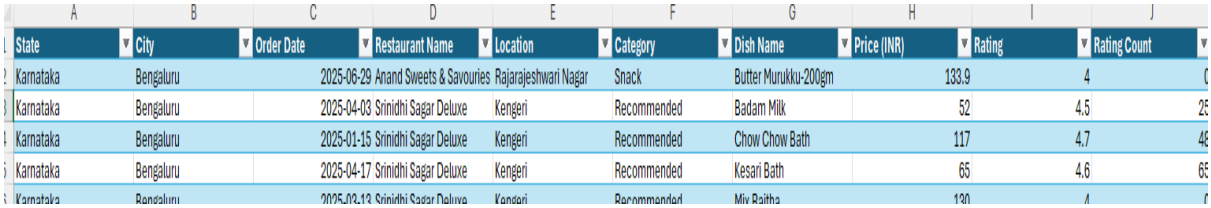
This project solves these problems using structured SQL analysis.

Dataset Overview

The dataset used in this project consists of **197K rows and 10 columns**, representing transactional order-level data. Each row corresponds to a single order, while the columns capture details related to customers, restaurants, locations, categories, dishes, pricing, ratings, and order timestamps.

Source Table: swiggy_data
Data Type: Transaction-level food delivery records

Screenshot of Raw table data



State	City	Order Date	Restaurant Name	Location	Category	Dish Name	Price (INR)	Rating	Rating Count
Karnataka	Bengaluru	2025-06-29	Anand Sweets & Savouries	Rajarajeshwari Nagar	Snack	Butter Murukku-200gm	133.9	4	0
Karnataka	Bengaluru	2025-04-03	Srinidhi Sagar Deluxe	Kengeri	Recommended	Badam Milk	52	4.5	25
Karnataka	Bengaluru	2025-01-15	Srinidhi Sagar Deluxe	Kengeri	Recommended	Chow Chow Bath	117	4.7	48
Karnataka	Bengaluru	2025-04-17	Srinidhi Sagar Deluxe	Kengeri	Recommended	Kesari Bath	65	4.6	65
Karnataka	Bengaluru	2025-02-13	Srinidhi Sagar Deluxe	Kengeri	Recommended	Mix Raitha	120	4	0

Key Columns:

- Order_Date
- State, City, Location
- Restaurant_Name
- Category (Cuisine)
- Dish_Name
- Price_INR
- Rating, Rating_Count

By Using SQL SSMS

To support analytical reporting and KPI calculation, the raw transactional data was transformed and loaded into a star schema consisting of dimension tables and a fact table.

Dimension tables store descriptive attributes such as customer, restaurant, city, and date details, while the fact table captures measurable business events like orders, revenue, quantity, and ratings.

SQL was used to clean, standardize, and insert data into these tables to ensure data consistency, referential integrity, and optimized query performance.

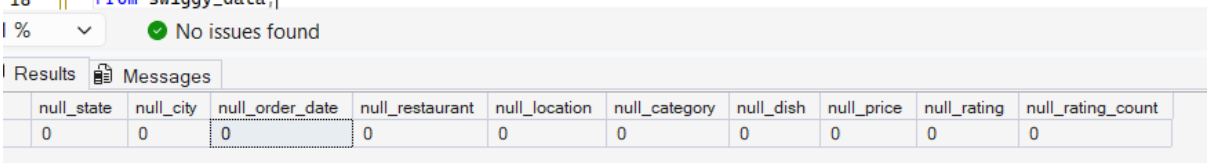
4. Data Cleaning & Validation (SQL)

Before analysis, data quality checks were performed to ensure accurate insights.

a) Null & Blank Value Checks

- Identified missing or empty values in:
    - State, City, Category
    - Dish_Name, Restaurant_Name
    - Price_INR, Rating

```
4   -- Data Validation and Cleaning
5   --Null Check
6
7   Select
8   SUM (CASE WHEN State IS NULL THEN 1 ELSE 0 END) AS null_state,
9   SUM (CASE WHEN City IS NULL THEN 1 ELSE 0 END) AS null_city,
10  SUM (CASE WHEN Order_Date IS NULL THEN 1 ELSE 0 END) AS null_order_date,
11  SUM (CASE WHEN Restaurant_Name IS NULL THEN 1 ELSE 0 END) AS null_restaurant,
12  SUM (CASE WHEN Location IS NULL THEN 1 ELSE 0 END) AS null_location,
13  SUM (CASE WHEN Category IS NULL THEN 1 ELSE 0 END) AS null_category,
14  SUM (CASE WHEN Dish_Name IS NULL THEN 1 ELSE 0 END) AS null_dish,
15  SUM (CASE WHEN Price_INR IS NULL THEN 1 ELSE 0 END) AS null_price,
16  SUM (CASE WHEN Rating IS NULL THEN 1 ELSE 0 END) AS null_rating,
17  SUM (CASE WHEN Rating_Count IS NULL THEN 1 ELSE 0 END) AS null_rating_count
18  From swiggy_data;
```



null_state	null_city	null_order_date	null_restaurant	null_location	null_category	null_dish	null_price	null_rating	null_rating_count
0	0	0	0	0	0	0	0	0	0

```sql
-- Check Empty data (Strings)
select * from swiggy_data
where state= '' OR City= '' OR Location= ''
OR Category= '' OR Dish_Name= '';
```

81 % No issues found

Results Messages

State	City	Order_Date	Restaurant_Name	Location	Category	Dish_Name	Price_INR	Rating	Rating_Count

Why important:
Null or blank values can distort revenue, average price, and rating calculations.

b) Duplicate Detection & Removal

- Used ROW_NUMBER() with partitioning on business-critical columns.
- Retained only one valid record per unique order.

```sql
-- Check Duplicate Values

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;
```

81 % No issues found

Results Messages

	State	City	Order_date	Restaurant_Name	Location
1	Gujarat	Ahmedabad	2025-01-02	McDonald's Gourmet Burger Collection	Ghatlodia
2	Gujarat	Ahmedabad	2025-01-16	McDonald's Gourmet Burger Collection	Ghatlodia
3	Gujarat	Ahmedabad	2025-01-17	McDonald's Gourmet Burger Collection	Ghatlodia

```sql
-- Delete Duplication

WITH CTE AS(
SELECT *, ROW_NUMBER () Over(
  PARTITION BY State, City, Order_date, Restaurant_Name, Location,
Category, Dish_Name, Price_INR, Rating, Rating_Count
ORDER BY (SELECT NULL)
) AS rn
from swiggy_data
)
DELETE FROM CTE WHERE rn>1
```

No issues found

Messages

```
(29 rows affected)

Completion time: 2026-01-15T23:10:49.1092467+05:30
```

Why important:
Duplicates inflate order count and revenue, leading to incorrect KPIs.

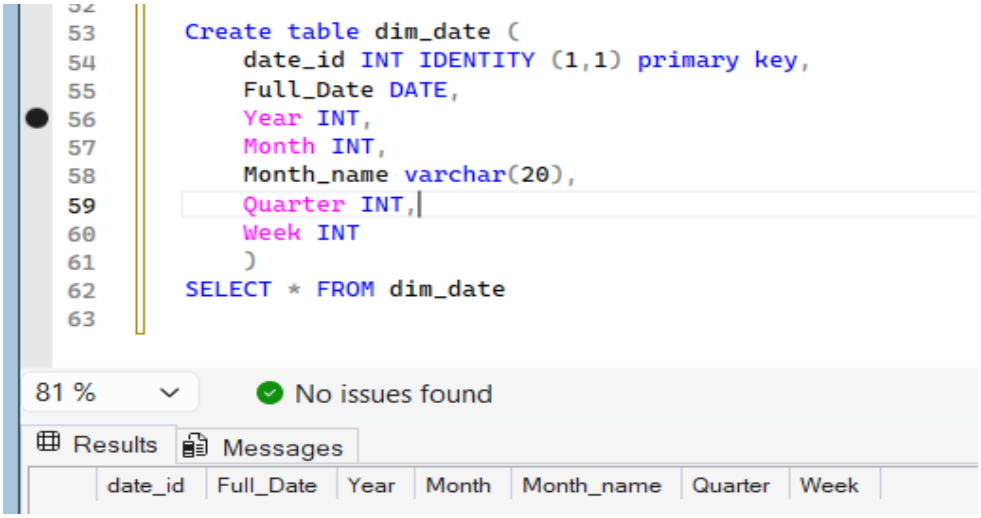
5. Dimensional Modelling (Star Schema)

To make the dataset scalable and BI-friendly, a **Star Schema** was designed.

Dimension Tables

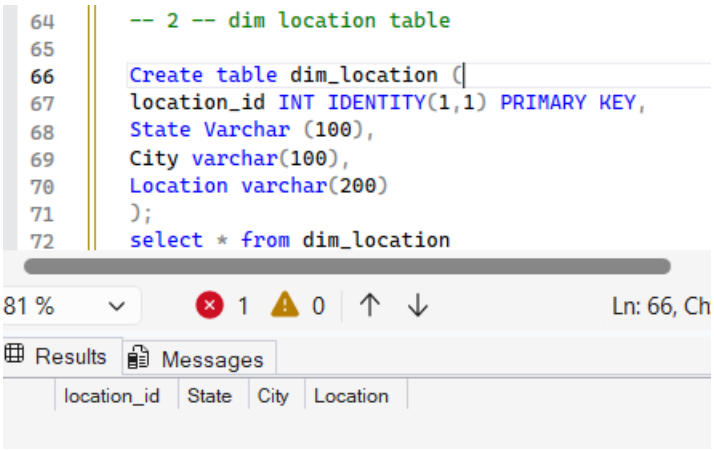
- **dim_date** → Year, Month, Quarter, Week

  The **dim_date** table was created to enable **time-based analysis** such as daily, monthly, and yearly trends. Date attributes like year, month, and day were derived using SQL date functions to support efficient filtering and aggregation in BI dashboards.



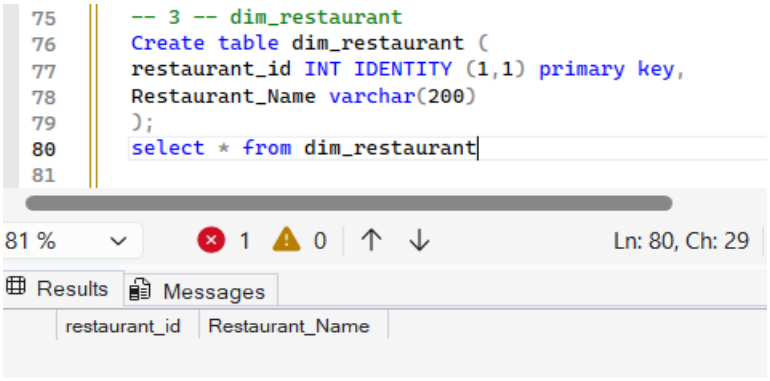
- **dim_location** → State, City, Location

  The dim_location table was created to store **geographical information** such as country, state, city, and area associated with each order.



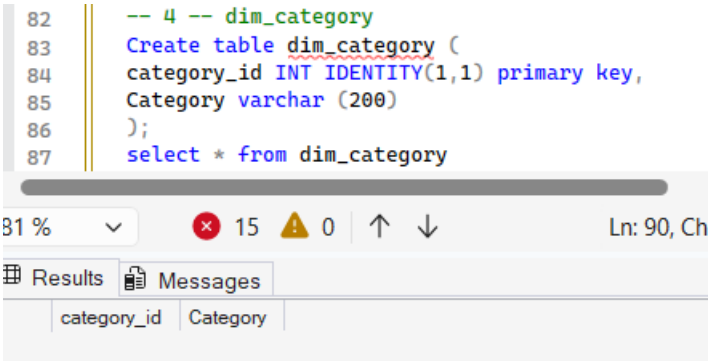
- **dim_restaurant** → Restaurant_Name

  The **dim_restaurant** table was created to store **restaurant-level descriptive information** such as restaurant name, category (cuisine type), price range, and ratings.This dimension enables detailed **restaurant performance analysis**, including order volume, revenue contribution, average ratings, and category-wise comparison.



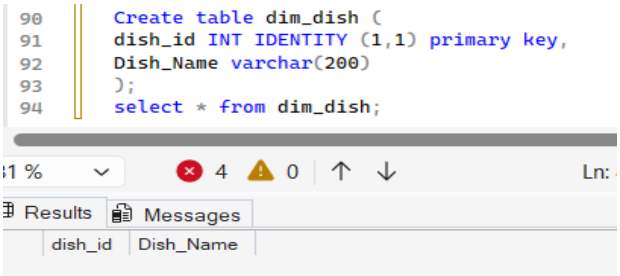
- **dim_category** → Cuisine / Category

  The **dim_category** table was created to store **food category / cuisine information** such as category name and cuisine type associated with restaurant orders.This dimension enables **category-wise analysis** including order distribution, revenue contribution, and customer preference trends across different food categories.



- **dim_dish** → Dish_Name

  The **dim_dish** table was created to store **dish-level information** such as dish name, dish category, and price.This dimension enables **item-level analysis**, helping identify top-selling dishes, low-performing items, and pricing impact on customer orders.

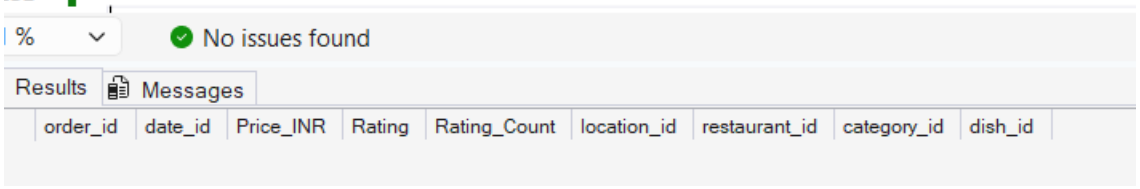


Fact Table

- **fact_swiggy_orders**
    - Price_INR
    - Rating
    - Rating_Count
    - Foreign keys to all dimensions

    The **fact_swiggy_orders** table was designed to store **transactional data at the order level**, representing measurable business events in the system.This table captures key metrics such as order amount, quantity, delivery time, and customer ratings, which are essential for calculating KPIs and performance indicators.

```
109  -- Fact Table
110  Create table fact_swiggy_orders (
111  order_id INT IDENTITY (1,1) Primary key,
112
113  date_id INT,
114  Price_INR Decimal (10,2),
115  Rating decimal(4,2),
116  Rating_Count INT,
117
118  location_id INT,
119  restaurant_id INT,
120  category_id INT,
121  dish_id INT,
122
123  FOREIGN KEY (date_id) REFERENCES dim_date(date_id),
124  FOREIGN KEY (location_id) REFERENCES dim_location(location_id),
125  FOREIGN KEY (restaurant_id) REFERENCES dim_restaurant(restaurant_id),
126  FOREIGN KEY (category_id) REFERENCES dim_category(category_id),
127  FOREIGN KEY (dish_id) REFERENCES dim_dish(dish_id)
128  );
129
130  select * from fact_swiggy_orders
131
```



Why Star Schema?

- Faster queries
- Cleaner data structure
- Industry-standard for analytics & BI tools

Data Insertion into Dimension and Fact Tables

After designing the data model, data was inserted into the **dimension and fact tables** using SQL as part of the ETL (Extract, Transform, Load) process.

Raw transactional data was first **cleaned, transformed, and standardized** to ensure accuracy and consistency. Dimension tables were populated with **unique descriptive attributes** using deduplication logic, while the fact table was populated with **transaction-level metrics** linked to all relevant dimensions through surrogate keys.

This structured data insertion approach ensures **referential integrity**, minimizes data redundancy, and prepares the dataset for efficient KPI calculation and business intelligence reporting.

```
144  -- INSERT DATA IN TABLES
145  -- dim_date
146  INSERT INTO dim_date (Full_Date, Year, Month, Month_Name, Quarter, Week)
147  Select distinct
148     Order_Date,
149     YEAR(Order_Date),
150     MONTH(Order_Date),
151     DATENAME(MONTH, Order_Date),
152     DATEPART(QUARTER, Order_Date),
153     DATEPART(WEEK, Order_Date)
154
155     from swiggy_data
156     Where Order_Date IS NOT NULL;
157
158     select * from dim_date;
```

81 % No issues found

Results Messages

	date_id	Full_Date	Year	Month	Month_name	Quarter	Week
1	1	2025-03-29	2025	3	March	1	13
2	2	2025-05-30	2025	5	May	2	22
3	3	2025-03-13	2025	3	March	1	11

```
159
160     -- dim_location
161
162  INSERT INTO dim_location (State, City, Location)
163  select distinct
164   State,
165   City,
166   Location
167   from swiggy_data;
168
169  select * from dim_Location
```

81 % No issues found

Results Messages

	location_id	State	City	Location
1	1	Assam	Guwahati	Amingaon
2	2	Assam	Guwahati	Azara

```sql
-- dim_category
INSERT INTO dim_category (Category)
select distinct
Category from swiggy_data

select * from dim_category;
```

81 % 1 0

Results Messages

	category_id	Category
1	1	Tea Cakes
2	2	Bakery
3	3	No Deal Like McDeal

```sql
-- dim_restaurant
INSERT INTO dim_restaurant (Restaurant_Name)
select distinct
Restaurant_Name
from swiggy_data

select * from dim_restaurant;
```

81 % 1 0 Ln: 173, Ch: 1

Results Messages

	restaurant_id	Restaurant_Name
1	1	Mr. Pizza
2	2	Khana Khazana (Shankar Nagar)
3	3	The Grill Republic

```sql
-- dim_dish
INSERT INTO dim_dish (Dish_Name)
select distinct
Dish_Name from swiggy_data;

select * from dim_dish
```

81 % No issues found Ln:

Results Messages

	dish_id	Dish_Name
1	1	Nandhana Special Andhra Veg Carrier Meals
2	2	Shavige Bhath + Uddina Vada(1pcs)
3	3	Paneer Stuff Masala

```sql
-- fact_table
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
```

Joins Between Tables

The project follows a **star schema data model**, where the fact table acts as the central table and is connected to multiple dimension tables through **foreign key relationships**.

Since dimension tables store descriptive data and the fact table stores measurable metrics, joins are essential to bring contextual information into analytical queries without duplicating data. This approach improves data consistency, reduces storage redundancy, and enhances query performance.

```sql
-- JOIN date dimension
JOIN dim_date dd
    ON dd.Full_Date = s.Order_Date

-- JOIN location dimension
JOIN dim_location dl
    ON dl.State = s.State
   AND dl.City = s.City
   AND dl.Location = s.Location

-- JOIN restaurant dimension
JOIN dim_restaurant dr
    ON dr.Restaurant_Name = s.Restaurant_Name

-- JOIN category dimension
JOIN dim_category dc
    ON dc.Category = s.Category

-- JOIN dish dimension
JOIN dim_dish dsh
    ON dsh.Dish_Name = s.Dish_Name;

select * from fact_swiggy_orders;
```

81 % No issues found

Results Messages

	order_id	date_id	location_id	restaurant_id	category_id	dish_id	Price_INR	Rating	Rating_Count
1	1	149	157	598	4557	46751	439.00	4.40	0
2	2	205	28	598	4557	45095	549.00	3.60	10

KPI Development & Explanation

Basic Business KPIs

1. **Total Orders -** Counts total food orders placed.

```sql
-- KPIs
-- 1 -- Total Orders
SELECT COUNT(*) AS Total_Orders
  FROM fact_swiggy_orders;
```

74 % No issues found

Results Messages

	Total_Orders
1	197401

Business Value:
Measures platform demand and customer engagement.

2. **Total Revenue (INR) -** Sum of Price_INR from all orders.

```sql
-- 2 -- Total Revenue
SELECT
    CAST(SUM(price_INR) / 1000000.0 AS DECIMAL(18,2)) AS Total_Revenue_Million
FROM fact_swiggy_orders;

-- 3 -- Average Dish Price
```

74 % No issues found

Results Messages

	Total_Revenue_Million
1	53.00

Business Value:
Tracks financial performance and growth.

3. **Average Dish Price -** Average price customers pay per dish.

```sql
-- 3 -- Average Dish Price
SELECT
    CAST(AVG(price_INR) AS DECIMAL(10,2)) AS Avg_Dish_Price_INR
FROM fact_swiggy_orders;
```

74 % No issues found

Results Messages

	Avg_Dish_Price_INR
1	268.50

Business Value:
Helps pricing teams identify affordable vs premium positioning.

4. **Average Rating -** Average customer rating across dishes.

```sql
-- 4 -- Average Rating
SELECT
    CAST(AVG(Rating) AS DECIMAL(5,2)) AS Avg_Rating
FROM fact_swiggy_orders;
```

74 % No issues found

Results Messages

	Avg_Rating
1	4.34

Business Value:
Indicates overall food quality and customer satisfaction.

Time-Based Analysis

Monthly & Quarterly Trends

- Identified peak and low-demand periods.
- Useful for marketing campaigns and discounts.

```sql
-- 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
Order by count(*) desc;
```

74 % No issues found Ln: 2

Results Messages

	year	month	month_name	Total_Orders
1	2025	1	January	25393
2	2025	8	August	25227

```sql
-- 2 -- Quarterly trend
select
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
Order by count(*) desc;
```

74 % No issues found Ln: 29

Results Messages

	year	quarter	Total_Orders
1	2025	2	74154

Year-wise Growth

- Shows business expansion over time.

```sql
-- 3 -- Yearly Trend (2025)
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;
```

74 % No issues found Ln: 30

Results Messages

	year	Total_Orders
1	2025	197401

Day-of-Week Analysis

- Highlights weekends vs weekdays demand patterns.

```sql
-- 4 -- Weekly Orders
select datename (Weekday, d.full_date) 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 datename (Weekday, d.full_date), datepart (weekday, d.full_date)
order by datepart (Weekday, d.full_date);
```

74 % 1 0 Ln: 309, Ch: 74 SPC CRLF Windows 1

Results Messages

	Day_name	Total_orders
1	Sunday	28469
2	Monday	27568
3	Tuesday	27413

Location-Based Insights

Top Cities by Orders

- Identifies high-demand urban markets.

```sql
-- 5 -- Top 10 cities by order volume
select top 10 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;
```

74 % No issues found Ln: 320, Ch: 2

Results Messages

	city	Total_Orders
1	Bengaluru	20072
2	Mumbai	10507
3	Hyderabad	10308

Revenue by State

- Helps regional teams focus on high-performing states.

```sql
-- 6 -- Revenue by states
select l.state,
sum(f.price_INR) AS Total_Revenue_INR
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;
```

89 % No issues found Ln: 329, Ch: 3

Results Messages

	state	Total_Revenue_INR
1	Karnataka	5455887.73
2	Uttar Pradesh	3117359.65
3	Telangana	3021656.62
4	Maharashtra	3015573.35

Business Impact:
Optimizes logistics, restaurant onboarding, and promotions.

Food & Restaurant Performance

- **Top 10 Restaurants -** Based on order volume.

```sql
-- 7 --  TOP 10 Restaurants by Orders
select top 10 r.restaurant_name,
sum(f.price_INR) AS Total_Revenue_INR
from fact_swiggy_orders f
JOIN dim_restaurant r
ON r.restaurant_id = f.restaurant_id
GROUP by r.restaurant_name
ORDER BY sum(f.price_INR) desc;
```

89 % No issues found Ln: 337, Ch: 27

Results Messages

	restaurant_name	Total_Revenue_INR
1	KFC	4245461.78
2	McDonald's	3342455.58
3	Pizza Hut	2133265.69
4	Burger King	1900518.09
5	Domino's Pizza	1832985.32
6	Olio - The Wood Fired Pizzeria	1232731.00
7	LunchBox - Meals and Thalis	1101141.00
8	Baskin Robbins - Ice Cream Desserts	860591.94
9	Faasos - Wraps, Rolls & Shawarma	780215.00
10	The Good Bowl	673343.00

Top Categories (Cuisines)

- Indian, Chinese, Fast Food, etc.

```
340   -- 8 -- Top categories by Order Volume
341   select
342   c.category, count(*) AS Total_orders
343   from fact_swiggy_orders f
344   JOIN dim_category c ON f.category_id = c.category_id
345   GROUP BY c.Category
346   ORDER BY total_orders desc;
```

89 % No issues found Ln: 346, Ch: 28 SPC CRLF W

Results Messages

	category	Total_orders
1	Recommended	24097
2	Desserts	3582
3	Main Course	2983
4	Beverages	2682

Most Ordered Dishes

- Reveals customer taste preferences.

```
348   -- 9 -- Most Ordered dish
349   select top 10 d.dish_name, count(*) AS order_count
350   from fact_swiggy_orders f
351   JOIN dim_dish d ON f.dish_id = d.dish_id
352   GROUP BY d.dish_name
353   ORDER BY order_count desc;
```

89 % No issues found Ln: 350, Ch: 27 SPC CRLF

Results Messages

	dish_name	order_count
1	Veg Fried Rice	321
2	Choco Lava Cake	303
3	Jeera Rice	265
4	Paneer Butter Masala	262
5	French Fries	248
6	Chicken Sausage	230
7	Chicken Fried Rice	228
8	Butter Naan	218
9	Margherita Pizza	203
10	Green Salad	197

Cuisine Performance

- Orders + Average Rating combined.

```
355    -- 10 --  Cuisine Performance
356    select c.category, count(*) AS Total_orders,
357   cast(avg(f.Rating) AS decimal(4,1)) AS Avg_Rating
358    from fact_swiggy_orders f
359    JOIN dim_category c ON f.category_id = c.category_id
360    group by c.Category
361    order by total_orders desc;
```

89 % | No issues found | Ln: 358, Ch: 28 | SPC | CRLF | Wind

Results | Messages

	category	Total_orders	Avg_Rating
1	Recommended	24097	4.3
2	Desserts	3582	4.4
3	Main Course	2983	4.3
4	Beverages	2682	4.4
5	BURGERS	2538	4.3
6	Sweets	1954	4.5
7	McSaver Combos (2 Pc Meals)	1884	4.4

Business Value:
Improves menu optimization and partner restaurant strategy.

Customer Spending Analysis

Order Value Distribution Buckets

- Under 100
- 100–199
- 200–299
- 300–499
- 500+

```sql
-- Customer spending insights

-- Under 100, 100-199, 200-299,
--       300-499, 500+

SELECT
   price_range,
   COUNT(*) AS Total_Orders
FROM (
   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 499 THEN '300 - 499'
     ELSE '500+'
        END AS price_range
   FROM fact_swiggy_orders
) t
GROUP BY price_range
ORDER BY Total_Orders DESC;
```

74 % | No issues found | Ln: 380, Ch: 6 | SPC

Results | Messages

	price_range	Total_Orders
1	100 - 199	56189
2	200 - 299	54567
3	300 - 499	43758
4	Under 100	26795
5	500+	16092

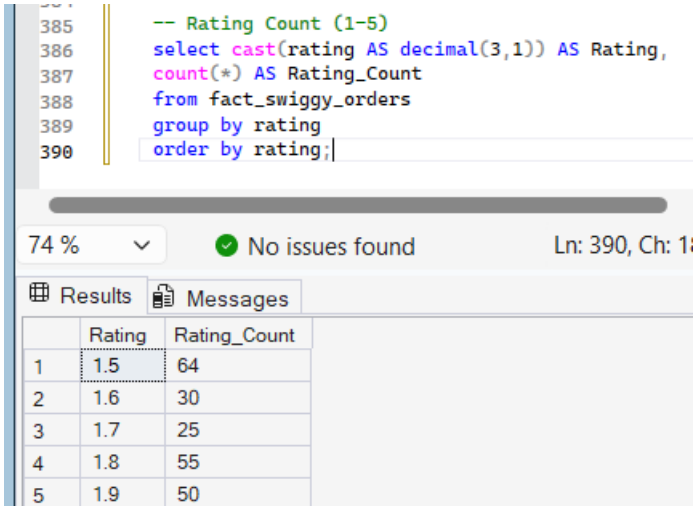
Why this matters:

- Shows dominant customer price range.
- Helps design coupons, combo offers, and premium plans.

Ratings Analysis

Rating Distribution (1–5 Stars)

- Identifies quality gaps.
- Helps focus on low-rated dishes/restaurants.



Dashboard Creation

After completing data modeling, ETL, and KPI calculations, an **interactive business intelligence dashboard** was created to visualize key insights and support data-driven decision-making.

The dashboard was built using **Power BI**, with data sourced from the SQL database containing the finalized fact and dimension tables. KPIs and measures were created using **DAX** to ensure accurate aggregation and dynamic filtering.

The dashboard presents critical business metrics such as total orders, total revenue, average order value, ratings, and delivery performance, enabling stakeholders to quickly understand overall performance and identify trends, patterns, and improvement areas.

SQL to Power BI Data Connection

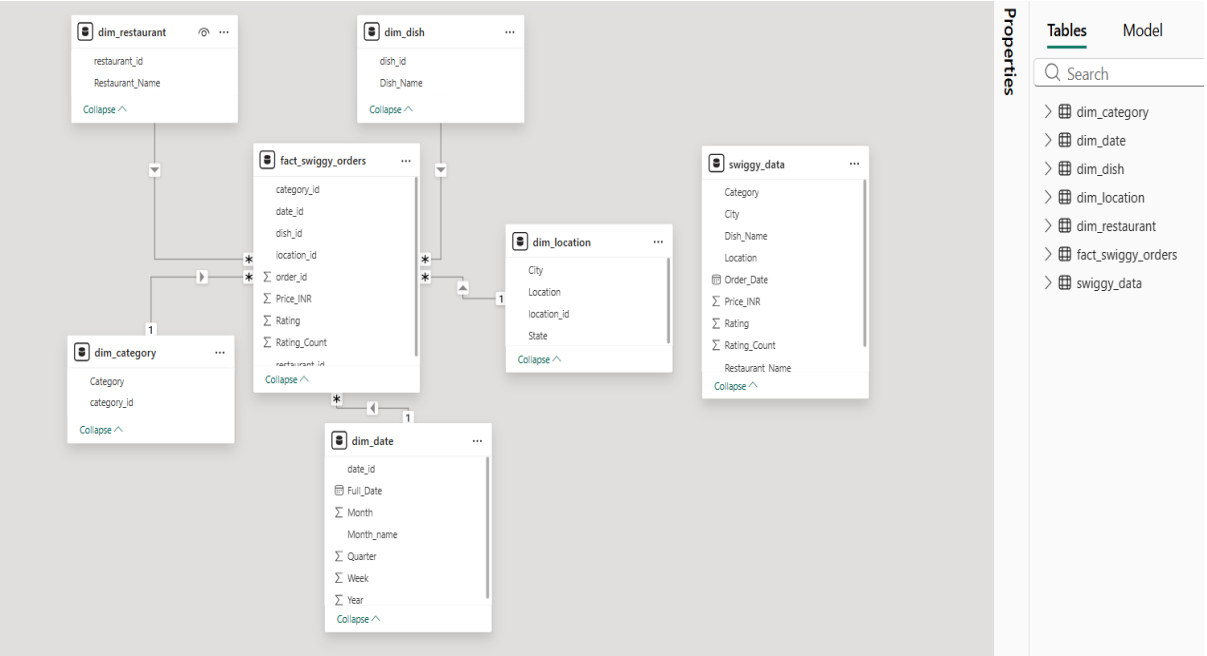
After completing data cleaning, dimensional modeling, and KPI execution in SQL, the finalized tables were connected to **Power BI** for visualization and analysis.

Data Model Used

The dataset follows a **Star Schema**, optimized for analytics and BI reporting:

- **Fact Table**
    - fact_swiggy_orders
    - Contains measurable metrics such as Price_INR, Rating, Rating_Count and foreign keys to dimensions
- **Dimension Tables**
    - dim_date – Date, Month, Quarter, Year
    - dim_location – State, City, Location
    - dim_restaurant – Restaurant Name
    - dim_category – Food Category / Cuisine
    - dim_dish – Dish Name

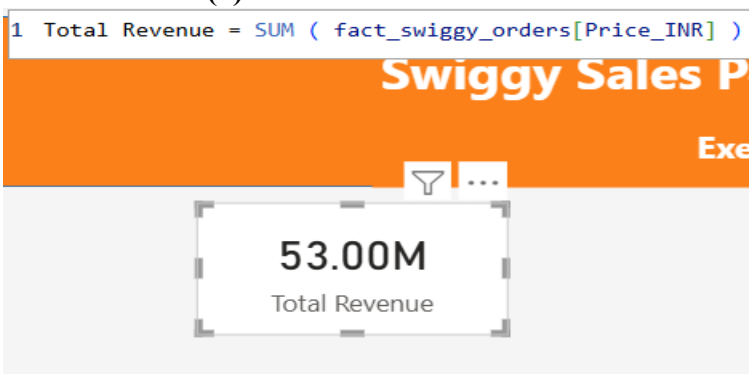
Table overview



Dax Measures

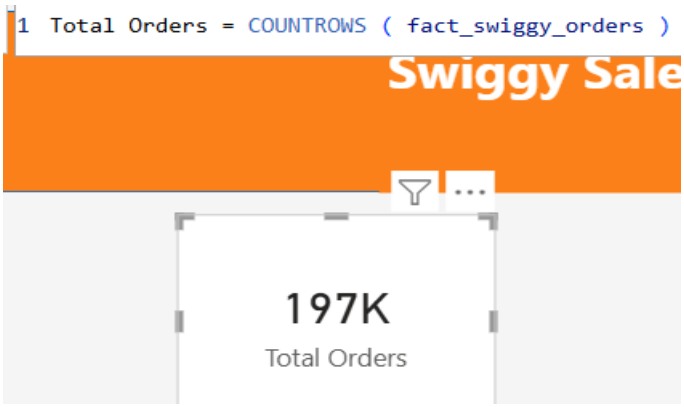
- **Total Revenue (₹)**

```
1  Total Revenue = SUM ( fact_swiggy_orders[Price_INR] )
```



- **Total Orders**

```
1  Total Orders = COUNTROWS ( fact_swiggy_orders )
```

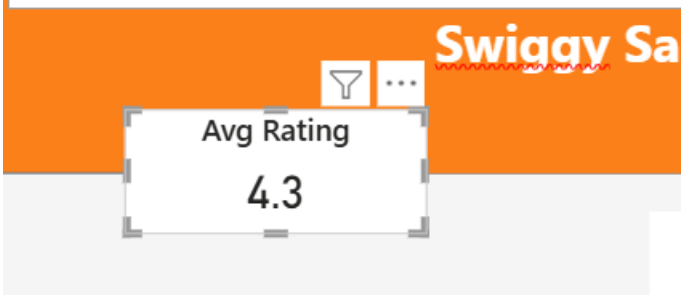


- **Average Dish Price**

```
1  Avg Dish Price = AVERAGE(fact_swiggy_orders[Price_INR])
```

- **Average Rating**

```
1  Avg Rating = AVERAGE(fact_swiggy_orders[Rating])
```



- **Weekday KPI**

```
1  Day Name =
2  FORMAT(dim_date[full_date], "dddd")
```

Shows which days of the week generate the highest order volume.

```
1  Day Number =
2  WEEKDAY ( dim_date[full_date], 2 )
```

Orders categories in logical business order for clear interpretation.

- **Order Value Bucket**

```
Order Value Bucket =
VAR Price = 'fact_swiggy_orders'[Price_INR]
RETURN
SWITCH(
    TRUE(),
    Price < 100, "Below 100",
    Price >= 100 && Price < 200, "100 - 199",
    Price >= 200 && Price < 300, "200 - 299",
    Price >= 300 && Price < 500, "300 - 499",
    Price >= 500, "500+"
)
```

Shows distribution of orders by spend range.

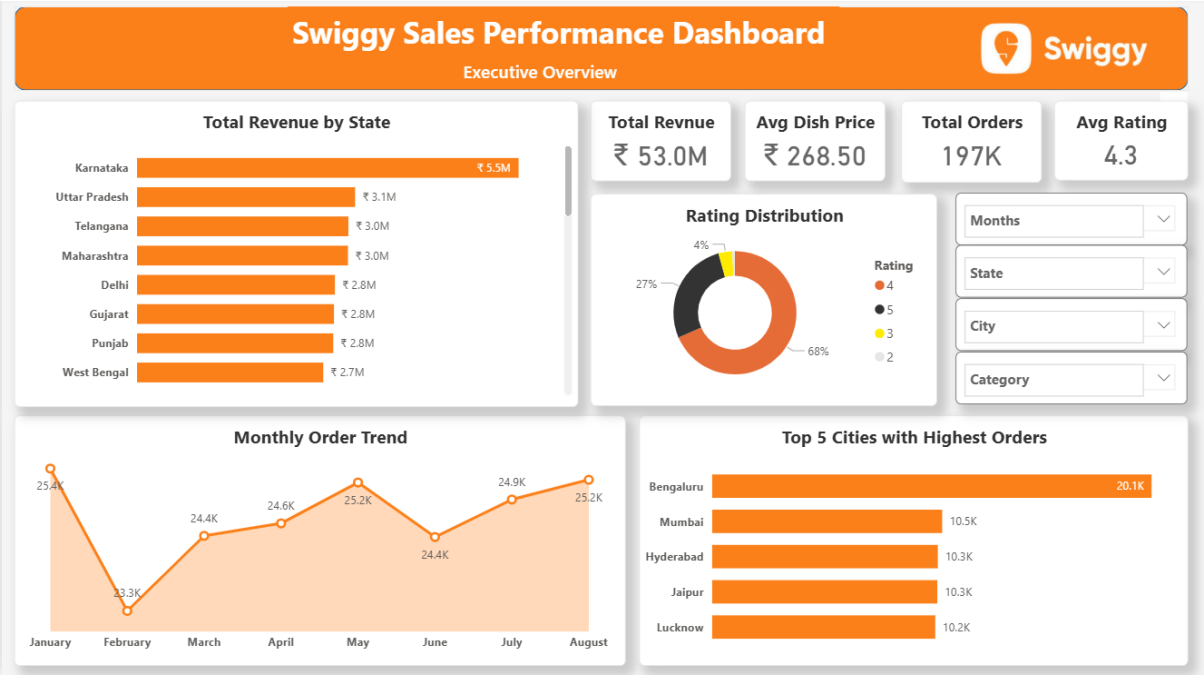
```
Order Value Sort =
VAR Price = 'fact_swiggy_orders'[Price_INR]
RETURN
SWITCH(
    TRUE(),
    Price < 100, 1,
    Price < 200, 2,
    Price < 300, 3,
    Price < 500, 4,
    5
)
```

Displays buckets from lowest to highest order value.

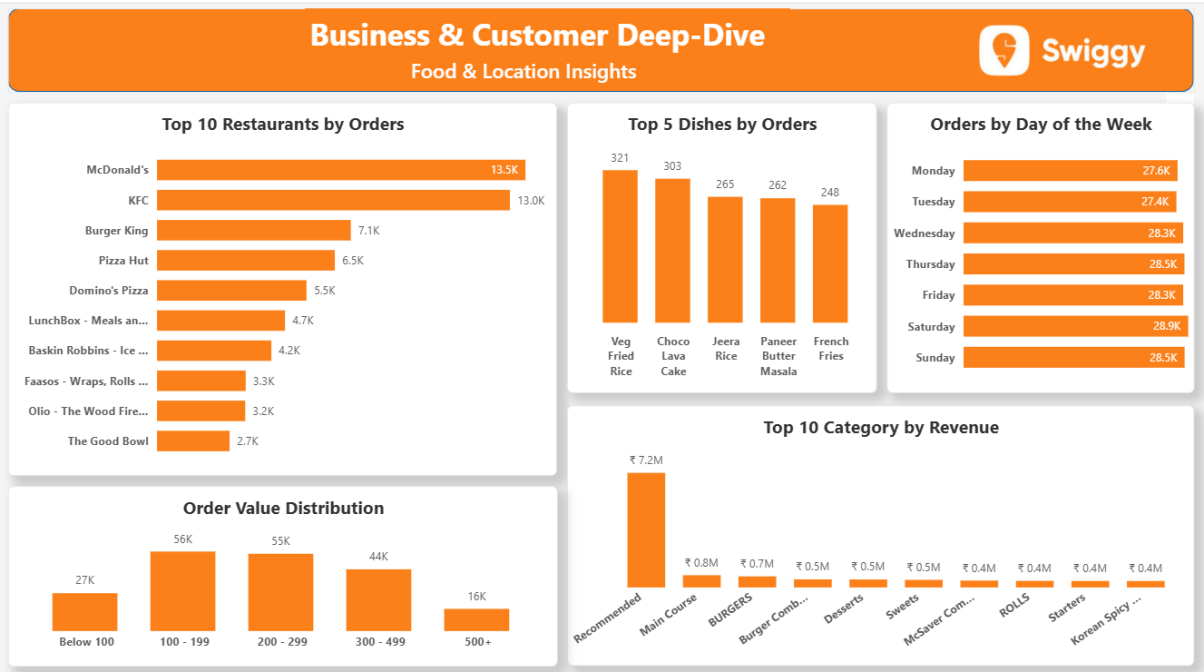
Final Dashboard

This dashboard presents executive-level insights into Swiggy sales performance, customer behavior, and revenue drivers using SQL-validated KPIs visualized in Power BI.

Page 1 – Executive Overview



Page 2 – Food & Location Insights



Key Observations & Insights

1. Demand & Revenue Concentration

- A small set of cities and states contribute a disproportionately high share of total orders and revenue.
- Urban and metro regions show higher order frequency and higher average spend compared to smaller cities.

Insight:
Swiggy's growth is currently city-driven rather than evenly distributed, indicating untapped potential in mid-tier cities.

2. Customer Price Sensitivity

- Majority of orders fall into Under ₹300 price buckets, especially ₹100–199 and ₹200–299.
- High-value orders (₹500+) exist but form a smaller, premium segment.

Insight:
Customers are highly price-sensitive, preferring affordable and mid-range meals rather than premium pricing.

3. Cuisine & Dish Performance

- A limited number of cuisines (e.g., Indian, Fast Food, Chinese) dominate both order volume and revenue.
- Top dishes contribute a significant portion of total orders, following a Pareto (80/20) pattern.

Insight:
Customer preferences are concentrated, meaning menu optimization can directly impact revenue.

4. Restaurant Performance Inequality

- A small group of restaurants generates very high order volumes, while many remain underutilized.
- Some restaurants receive high ratings but low order volumes, suggesting visibility or pricing issues.

Insight:
Not all quality restaurants are commercially successful — discoverability and promotion matter as much as ratings.

5. Rating & Quality Gaps

- Orders with higher ratings correlate with repeat demand.
- Low-rated dishes/restaurants still receive orders, indicating lack of feedback loops or quality enforcement.

Insight:
Ratings are underutilized as a control mechanism for food quality and customer satisfaction**.**

6. Time-Based Demand Patterns

- Clear weekly and monthly demand cycles exist.
- Certain days consistently outperform others in order volume.

Insight:
Swiggy has strong opportunities for time-based promotions and demand forecasting.

Identified Business Problems

1. Revenue dependency on limited cities and restaurants
2. High customer price sensitivity limiting AOV growth
3. Underperforming restaurants despite good ratings
4. Weak action on low-rated dishes/restaurants
5. Inefficient targeting of promotions across time and location

Business Recommendations

1. City-Level Growth Strategy

- Expand restaurant onboarding and localized offers in mid-performing cities.
- Use top-city success models (pricing, cuisine mix) to replicate growth in smaller cities.

Business Impact: Revenue diversification & reduced dependency on metro cities.

2. Smart Pricing & Offer Design

- Focus discounts and combos in ₹100–299 price range, where demand is highest.
- Introduce premium bundles for ₹500+ customers instead of flat discounts.

Business Impact*:* Higher order volume without sacrificing margins.

3. Menu & Cuisine Optimization

- Promote top-performing cuisines and dishes through homepage placement.
- Encourage restaurants to:
    - Remove low-performing dishes
    - Introduce variations of high-demand items

Business Impact: Increased conversion rate and customer satisfaction.

4. Restaurant Performance Improvement

- Identify high-rated but low-order restaurants and boost visibility using:
    - Sponsored listings
    - Targeted promotions
- Provide insights dashboards to restaurant partners.

Business Impact: Better restaurant utilization and platform trust**.**

5. Rating-Driven Quality Control

- Flag consistently low-rated dishes/restaurants for:
    - Quality audits
    - Temporary de-ranking
- Reward high-rated restaurants with visibility boosts.

Business Impact: Improved customer experience & repeat orders.

6. Time-Based Marketing & Operations

- Run campaigns on high-performing days to maximize ROI.
- Use low-demand days for:
    - Cashback offers
    - Free delivery incentives

Business Impact: Balanced demand & optimized delivery operations.

Conclusion

This project demonstrates how SQL-driven data modeling and Power BI analytics can transform raw transactional data into actionable business insights. By combining KPI analysis, customer behavior patterns, pricing distribution, location performance, and rating trends, the analysis identifies clear opportunities to improve revenue growth, customer satisfaction, and operational efficiency.