

# OUTPUT FOR BUSINESS REQUIREMENTS

## Data Validation & Cleaning

- 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;
```

- Blank or Empty Cells

```
Select * from Swiggy_Data where City= '' or State= '' or Restaurant_Name= '' or
Location= '' or Category= '' or Dish_Name= ''
```

- Duplicate Cells

```
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
```

- Remove Duplicates

```
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;
```

## Create Snow Flake Schema

- Dimension Tables

### 1. Dim\_Date Table

```
create Table Dim_Date (
    date_id int IDENTITY(1,1) Primary Key,
    Full_Date Date,
    Year Int,
    Month Int,
    Month_Name Varchar(20),
    Quarter Int,
    Week int,
    Day int
)
```

### 2. Dim\_Location Table

```
create table Dim_Location(
    Location_Id int Identity(1,1) Primary Key,
    State Varchar(100),
    City varchar(100),
    Location Varchar(200)
)
```

### 3. Dim\_Restaurant Table

```
create table Dim_Restaurant(
    Restaurant_id int identity(1,1) primary key,
    Restaurant_Name Varchar(200)
)
```

### 4. Dim\_Category Table

```
create table Dim_Category(
    Category_id int identity(1,1) primary key,
    Category_Name Varchar(200)
)
```

### 5. Dim\_Dish Table

```
create table Dim_Dish(
    Dish_id int identity(1,1) primary key,
    Dish_Name Varchar(200)
)
```

- Fact Swiggy Table

```
Create Table Fact_Swiggy(
    order_id int identity(1,1) primary key,
    date_id int,
    dish_id int,
    Category_id int,
    Location_id int,
    Restaurant_id int,
    Price_INR Decimal(10,2),
    Rating Decimal(10,2),
    Rating_Count int,
    foreign key (date_id) references Dim_Date(date_id),
    foreign key (dish_id) references Dim_Dish(dish_id),
    foreign key (Location_id) references Dim_Location(Location_id),
    foreign key (Category_id) references Dim_Category(Category_id),
    foreign key (Restaurant_id) references Dim_Restaurant(Restaurant_id))
```

## Inserting Data to Dim & Fact Table

- **Dim\_Date**

```
insert into Dim_Date(Full_Date, Year, Month, Month_Name, Quarter, Week, Day)
select Distinct
    Order_Date,
    Year(Order_Date),
    Month(Order_Date),
    DATENAME(Month,Order_Date),
    DATEPART(quarter,Order_Date),
    Datepart(Week,Order_Date),
    Day (Order_Date)
from Swiggy_Data
where Order_Date is not null;
```

- **Dim\_Location**

```
insert into Dim_Location(State, City, Location)
select distinct
    state,
    city,
    location
from Swiggy_Data
```

- **Dim\_Category**

```
insert into Dim_Category(Category_Name)
select distinct Category
from Swiggy_Data
```

- **Dim\_Restaurant**

```
insert into Dim_Restaurant(Restaurant_Name)
select distinct
    Restaurant_Name
from Swiggy_Data
```

- **Dim\_Dish**

```
insert into Dim_Dish(Dish_Name)
select distinct
    Dish_Name
from Swiggy_Data
```

- **Fact\_Swiggy**

```
INSERT INTO Fact_Swiggy(
    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_Name = s.Category
JOIN dim_dish dsh ON dsh.Dish_Name = s.Dish_Name;
```

- **Fact\_Swiggy Data**

```
select * from Fact_Swiggy
```

- **To Get Full\_Data\_in One table**

```
SELECT *
FROM Fact_Swiggy 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;
```

### Key Performance Indicators

- **Total Orders**

```
select count(order_id) as Total_Orders from Fact_Swiggy
```

Total_Orders	
1	197401

- **Total Revenue (INR Million)**

```
select
format(SUM(convert(float,Price_INR)) / 1000000, 'N2') + ' INR Million'
as Total_Revenue
from Fact_Swiggy
```

Total_Revenue	
1	53.00 INR Million

- **Average Dish Price**

```
select
format(Avg(convert(float,Price_INR)), 'N2') + ' INR'
as Average_Dish_Price
from Fact_Swiggy
```

Average_Dish_Price	
1	268.50 INR

- **Average Rating**

```
select Avg(rating) as Average_Rating from Fact_Swiggy
```

Average_Rating	
1	4.341577

## Deep Dive Analysis

- **Month Order Trends**

```
select d.Year, d.Month, d.Month_Name, COUNT(*) as Total_orders from Fact_Swiggy f
join Dim_Date d on f.date_id= d.date_id
group by d.Year, d.Month, d.Month_Name
```

	Year	Month	Month_Name	Total_orders
1	2025	1	January	25393
2	2025	2	February	23291
3	2025	3	March	24400
4	2025	4	April	24584
5	2025	5	May	25188
6	2025	6	June	24382
7	2025	7	July	24936
8	2025	8	August	25227

- **Extra \*Month Revenue Trends\***

```
select d.Year, d.Month, d.Month_Name, Sum(Price_INR) as Total_Revenue from Fact_Swiggy 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
```

	Year	Month	Month_Name	Total_Revenue
1	2025	1	January	6823981.33
2	2025	5	May	6792621.35
3	2025	8	August	6790899.22
4	2025	7	July	6650268.51
5	2025	4	April	6590449.00
6	2025	3	March	6572738.20
7	2025	6	June	6513676.19
8	2025	2	February	6268106.67

- **Quarterly Order Trends**

```
select d.Year, d.Quarter, COUNT(*) as Total_orders from Fact_Swiggy f
join Dim_Date d on f.date_id= d.date_id
group by d.Year, d.Quarter
```

	Year	Quarter	Total_orders
1	2025	1	73084
2	2025	2	74154
3	2025	3	50163

- **Yearly orders**

```
select d.year, COUNT(*) as Total_Orders from Fact_Swiggy f
join Dim_Date d on d.date_id = f.date_id
group by d.Year
order by d.year asc
```

	year	Total_Orders
1	2025	197401

- **Orders by day of the week**

```
select
DATENAME(weekday, d.Full_Date) as Day_Week, COUNT(*) as Total_Orders from Fact_Swiggy 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)
```

	Day_Week	Total_Orders
1	Sunday	28469
2	Monday	27568
3	Tuesday	27413
4	Wednesday	28284
5	Thursday	28450
6	Friday	28284
7	Saturday	28933

- **Top 10 cities by Order**

```
select top 10 l.City, COUNT(*) as Total_Orders from Fact_Swiggy f
join Dim_Location l on f.Location_id=l.Location_Id
group by l.City
order by COUNT(*) desc
```

	City	Total_Orders
1	Bengaluru	20072
2	Mumbai	10507
3	Hyderabad	10308
4	Jaipur	10285
5	Lucknow	10192
6	New Delhi	10191
7	Ahmedabad	10175
8	Chandigarh	10060
9	Kolkata	10044
10	Chennai	10042

- **Revenue Contribution By states**

```
select l.State, sum(Price_INR) as Total_Revenue from Fact_Swiggy f
join Dim_Location l on f.Location_id =l.Location_Id
group by l.State
order by sum(Price_INR) desc
```

	State	Total_Revenue
1	Karnataka	5455887.73
2	Uttar Pradesh	3117359.65
3	Telangana	3021656.62
4	Maharashtra	3015573.35
5	Delhi	2829180.60
6	Gujarat	2815536.27
7	Punjab	2804991.82
8	West Bengal	2662213.76
9	Tamil Nadu	2642594.63
10	Rajasthan	2502833.61

- **Top 10 Restaurants by Orders**

```
select Top 10 r.Restaurant_Name, COUNT(*) as Total_Orders from Fact_Swiggy f
join Dim_Restaurant r on f.Restaurant_id=r.Restaurant_id
group by r.Restaurant_Name
order by Count(*) Desc
```

	Restaurant_Name	Total_Orders
1	McDonald's	13528
2	KFC	12957
3	Burger King	7115
4	Pizza Hut	6529
5	Domino's Pizza	5489
6	LunchBox - Meals and Thalis	4700
7	Baskin Robbins - Ice Cream Desserts	4197
8	Faasos - Wraps, Rolls & Shawarma	3256
9	Olio - The Wood Fired Pizzeria	3239
10	The Good Bowl	2665

- **Top Categories by Revenue**

```
select top 20 c.Category_Name, Sum(Price_INR) as Total_Revenue from Fact_Swiggy f
join Dim_Category c on f.Category_id=c.Category_id
group by c.Category_Name
order by Sum(Price_INR) desc
```

	Category_Name	Total_Revenue
1	Recommended	7188217.53
2	Main Course	767175.00
3	BURGERS	694960.48
4	Burger Combos ( 3 Pc Meals )	507773.55
5	Desserts	500588.72
6	Sweets	475068.40
7	McSaver Combos (2 Pc Meals)	431336.65
8	ROLLS	410715.89
9	Starters	407731.00
10	Korean Spicy Fest(Limited Time Only)	404892.00

- **Top Categories by Orders**

```
Select d.Dish_Name, count(*) as Total_Orders from Fact_Swiggy f
join Dim_Dish d on f.dish_id=d.Dish_id
group by d.Dish_Name
order by count(*) desc
```

	Category_Name	Total_Orders
1	Recommended	24097
2	Desserts	3582
3	Main Course	2983
4	Beverages	2682
5	BURGERS	2538
6	Sweets	1954
7	McSaver Combos (2 Pc Meals)	1884
8	Exclusive Deals (Save upto 40...)	1717
9	Starters	1692
10	ROLLS	1652

- **Most ordered dishes**

```
Select d.Dish_Name, count(*) as Total_Orders from Fact_Swiggy f
join Dim_Dish d on f.dish_id=d.Dish_id
group by d.Dish_Name
order by count(*) desc
```

	Dish_Name	Total_Orders
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
11	Dal Makhani	184
12	Margherita	184
13	Cold Coffee	182
14	Double Chicken Roll	180
15	Veggie Supreme	179

- **Cuisine performance → Orders + Avg Rating**

```
SELECT c.Category_Name, COUNT(*) AS Total_Orders, AVG(CONVERT(FLOAT, f.rating)) AS
Avg_Rating FROM Fact_Swiggy f
JOIN dim_category c ON f.category_id = c.category_id
GROUP BY c.Category_Name
ORDER BY total_orders DESC;
```

	Category_Name	Total_Orders	Avg_Rating
1	Recommended	24097	4.32178279453843
2	Desserts	3582	4.3716359575655
3	Main Course	2983	4.31019108280243
4	Beverages	2682	4.36871737509312
5	BURGERS	2538	4.3249408983451
6	Sweets	1954	4.45552712384845
7	McSaver Combos (2 Pc Meals)	1884	4.41380042462842
8	Exclusive Deals (Save upto 40%)	1717	4.34880605707629
9	Starters	1692	4.30484633569735
10	ROLLS	1652	4.2463680387409
11	Snacks	1438	4.31244784422806
12	Breads	1422	4.34781997187058
13	DESSERTS & BEVERAGES	1333	4.33780945236308
14	Burger Combos ( 3 Pc Meals )	1331	4.37821187077384
15	DOTD	1307	4.3143075745983

- **Total Orders by Price Range**

```

select
CASE
    when Convert(Float,Price_INR) < 100 Then 'Under 100'
    when Convert(Float,Price_INR) Between 100 and 200 Then 'B/W 100 - 200'
    when Convert(Float,Price_INR) Between 200 and 300 Then 'B/W 200 - 300'
    when Convert(Float,Price_INR) Between 400 and 500 Then 'B/W 400 - 500'
    Else '500+'
End as Price_range,
Count(*) as Total_Orders from Fact_Swiggy
Group by
CASE
    when Convert(Float,Price_INR) < 100 Then 'Under 100'
    when Convert(Float,Price_INR) Between 100 and 200 Then 'B/W 100 - 200'
    when Convert(Float,Price_INR) Between 200 and 300 Then 'B/W 200 - 300'
    when Convert(Float,Price_INR) Between 400 and 500 Then 'B/W 400 - 500'
    Else '500+'
End
order by Total_Orders Desc

```

	Price_range	Total_Orders
1	B/W 100 - 200	58279
2	B/W 200 - 300	54026
3	500+	45612
4	Under 100	26795
5	B/W 400 - 500	12689

- **Rating Count Distribution**

```

Select Rating, COUNT(Rating_Count) as Rating_Count from Fact_Swiggy
group by Rating
order by Rating

```

	Rating	Rating_Count
1	1.50	64
2	1.60	30
3	1.70	25
4	1.80	55
5	1.90	50
6	2.00	247
7	2.10	161
8	2.20	209
9	2.30	227
10	2.40	234
11	2.50	283
12	2.60	316
13	2.70	436
14	2.80	552
15	2.90	549
16	3.00	740
17	3.10	701
18	3.20	996
19	3.30	1278
20	3.40	1340