



# SWIGGY EDA

By Yuvaraj M

# About the Company



Swiggy is one of India's leading food delivery platforms, founded in 2014 and headquartered in Bengaluru. It allows users to order food from local restaurants via its app and also offers grocery delivery through Swiggy Instamart and parcel services via Swiggy Genie. The company operates in over 500 cities and serves millions of customers across the country. Backed by major investors, Swiggy has expanded into quick commerce and acquired Dineout to enter the dining-out space. Its main competitor is Zomato, and both dominate India's online food delivery market.



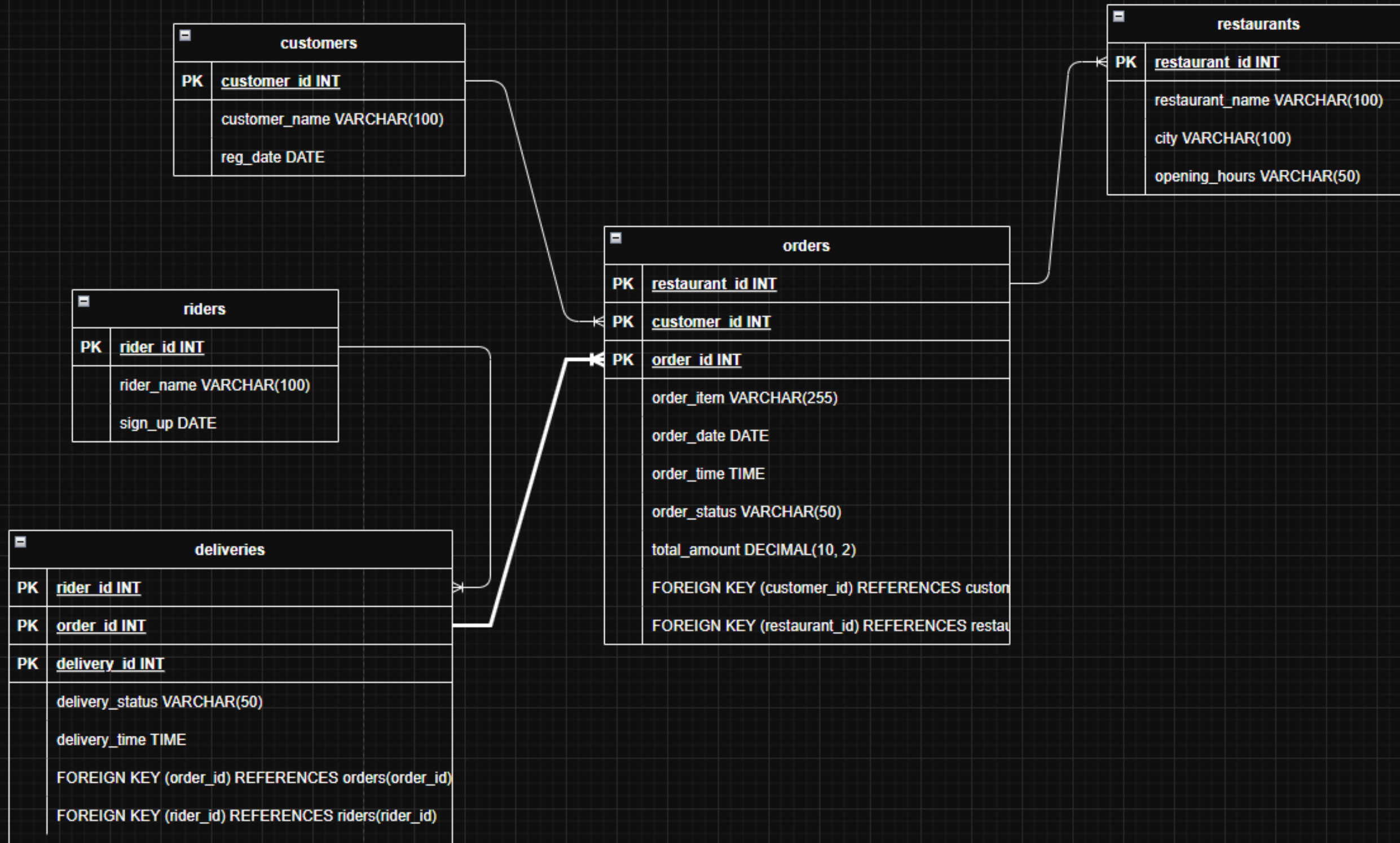
# EDA PART 1 ANALYSIS

- ✔ ER Diagram
- ✔ Time Period Exploration
- ✔ Dimension Exploration
- ✔ Measure Exploration
- ✔ Generate Keymetrics report and view
- ✔ Query Insights





# ER Diagram



## Table Information

### Query

```
SELECT * FROM INFORMATION_SCHEMA.TABLES
```

### Output

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
1	swiggy	dbo	customers	BASE TABLE
2	swiggy	dbo	orders	BASE TABLE
3	swiggy	dbo	deliveries	BASE TABLE
4	swiggy	dbo	restaurants	BASE TABLE
5	swiggy	dbo	riders	BASE TABLE
6	swiggy	dbo	sysdiagrams	BASE TABLE
7	swiggy	dbo	Keymetrics	VIEW

## Time Period Exploration

### Query

```
SELECT
    Min(order_date) as FirstOrderDate,
    MAX(order_date) as LastOrderDate
FROM
    orders
```

### Output

	FirstOrderDate	LastOrderDate
1	2023-01-01	2024-01-25

### Query

```
SELECT
    DATEDIFF(YEAR,MIN(order_date),Max(order_date)) as OrderRangeInYears
FROM orders
```

### Output

	OrderRangeInYears
1	1

Why Time Period Exploration?

"Time Period Exploration" is crucial in data analysis and reporting because it helps us understand how things change over time — revealing trends, seasonality, patterns, and anomalies that inform better decision-making.



## Time Period Exploration

### Query

```
SELECT
    YEAR(order_date) AS OrderYear,
    DATENAME(MONTH, order_date) AS MonthName,
    MIN(order_date) AS FirstOrderDate,
    MAX(order_date) AS LastOrderDate
FROM
    orders
GROUP BY
    YEAR(order_date),
    DATENAME(MONTH, order_date),
    MONTH(order_date)
ORDER BY
    OrderYear,
    MONTH(order_date);
```

### Output

	OrderYear	MonthName	FirstOrderDate	LastOrderDate
1	2023	January	2023-01-01	2023-01-31
2	2023	February	2023-02-01	2023-02-28
3	2023	March	2023-03-01	2023-03-31
4	2023	April	2023-04-01	2023-04-30
5	2023	May	2023-05-01	2023-05-31
6	2023	June	2023-06-01	2023-06-30
7	2023	July	2023-07-01	2023-07-31
8	2023	August	2023-08-01	2023-08-31
9	2023	September	2023-09-01	2023-09-30
10	2023	October	2023-10-01	2023-10-31
11	2023	November	2023-11-01	2023-11-30
12	2023	December	2023-12-01	2023-12-31
13	2024	January	2024-01-01	2024-01-25

## Dimension Exploration

### Query

```
SELECT distinct TOP 10
    c.customer_name as CustomerName,
    r.restaurant_name as RestaurantName,
    o.order_item as OrderItem,
    r.city as City
FROM
    orders o
left join customers c on o.customer_id = c.customer_id
left join restaurants r on r.restaurant_id = o.restaurant_id
left join deliveries d on d.order_id = o.order_id
```

### Output

	CustomerName	RestaurantName	OrderItem	City
1	Karan Kapoor	Almond House	Pasta Alfredo	Hyderabad
2	Rohan Iyer	SodaBottleOpenerWala	Chicken Biryani	Delhi
3	Aman Gupta	Yauatcha	Masala Dosa	Mumbai
4	Manish Kulkarni	Peshawri	Chicken Shawarma	Chennai
5	Rahul Verna	Diggin	Samosa	Delhi
6	Anjali Saxena	Cafe Bahar	Burger	Hyderabad
7	Nikhil Jain	Punjabi By Nature	Mutton Rogan Josh	Delhi
8	Shreya Ghosh	Punjabi By Nature	Paneer Butter Masala	Delhi
9	Arjun Mehta	Toit	Pasta Alfredo	Bengaluru
10	Rahul Verna	Gajalee	Paneer Lababdar	Mumbai

Why Dimension Exploration?  
"Dimension Exploration" is uncovers structure, improves visualization and sets up better modeling outcomes. Bridges the gap between raw data and actionable insight.



# Measure Exploration

## Queries

```
SELECT 'TotalCustomers' AS KeyMetrics, COUNT(*) AS Value FROM customers
```

```
SELECT 'TotalOrders' AS KeyMetrics, COUNT(DISTINCT order_id) AS Value FROM orders
```

```
SELECT 'TotalAmount' AS KeyMetrics, SUM(total_amount) AS Value FROM orders
```

```
SELECT 'TotalRestaurant' AS KeyMetrics, COUNT(*) AS Value FROM restaurants
```

```
SELECT 'TotalDishes' AS KeyMetrics, COUNT(DISTINCT order_item) AS Value FROM orders;
```

## Output

	KeyMetrics	Value
1	TotalCustomers	33
2	TotalOrders	10000
3	TotalAmount	3228216
4	TotalRestaurant	71
5	TotalDishes	23

Why Measure Exploration?  
"Measure Exploration" is  
refers to analyzing  
quantitative (numeric) data  
in your dataset.

## Generated Keymetrics report and view

### Query

```
CREATE VIEW Keymetrics AS
SELECT 'TotalCustomers' AS KeyMetrics, COUNT(*) AS Value FROM customers
UNION ALL
SELECT 'TotalOrders' AS KeyMetrics, COUNT(DISTINCT order_id) AS Value FROM orders
UNION ALL
SELECT 'TotalAmount' AS KeyMetrics, SUM(total_amount) AS Value FROM orders
UNION ALL
SELECT 'TotalRestaurant' AS KeyMetrics, COUNT(*) AS Value FROM restaurants
UNION ALL
SELECT 'TotalDishes' AS KeyMetrics, COUNT(DISTINCT order_item) AS Value FROM orders;

SELECT * FROM Keymetrics
```

### Output

	KeyMetrics	Value
1	TotalCustomers	33
2	TotalOrders	10000
3	TotalAmount	3228216
4	TotalRestaurant	71
5	TotalDishes	23



## Query Insights

->City wise customer Count and orders Count

### Query

```
SELECT
    DISTINCT r.city as City,
    COUNT(DISTINCT c.customer_id) as CustomerCount,
    COUNT(DISTINCT o.order_id) as OrdersCount
FROM
    Orders o
LEFT JOIN customers c on o.customer_id = c.customer_id
LEFT JOIN restaurants r on o.restaurant_id = r.restaurant_id
GROUP BY r.city
ORDER BY r.city DESC, CustomerCount DESC, OrdersCount DESC
```

### Output

	City	CustomerCount	OrdersCount
1	Mumbai	24	4723
2	Hyderabad	22	955
3	Delhi	22	1199
4	Chennai	22	894
5	Bengaluru	23	2229



## Query Insights

->Month wise order count

### Query

```
SELECT
    DATENAME(MONTH,order_date) AS MonthName,
    MONTH(order_date) AS MonthNumber,
    COUNT (DISTINCT Order_id) as OrderCount
FROM orders
GROUP BY DATENAME(MONTH,order_date) ,MONTH(order_date)
ORDER BY MONTH(order_date)
```

### Output

	MonthName	MonthNumber	OrderCount
1	January	1	896
2	February	2	727
3	March	3	891
4	April	4	827
5	May	5	854
6	June	6	782
7	July	7	874
8	August	8	794
9	September	9	797
10	October	10	892
11	November	11	812
12	December	12	854



## Query Insights

->Highest Sales in which Month

### Query

```
SELECT
    DATENAME(MONTH,order_date) AS MonthName,
    MONTH(order_date) AS MonthNumber,
    SUM(total_amount) AS TotalSales
FROM orders
GROUP BY DATENAME(MONTH,order_date) ,MONTH(order_date)
ORDER BY TotalSales DESC
```

### Output

	MonthName	MonthNumber	TotalSales
1	March	3	288309
2	October	10	286519
3	July	7	284818
4	January	1	283600
5	May	5	276827
6	December	12	276097
7	April	4	271615
8	November	11	264235
9	September	9	259743
10	August	8	255234
11	June	6	247780
12	February	2	233439



## Query Insights

->Restaurant wise order count

### Query

```
SELECT
    r.restaurant_id,
    COUNT (DISTINCT O.order_id) AS OrderCount
FROM orders o
left join restaurants r
on o.restaurant_id = r.restaurant_id
GROUP BY r.restaurant_id
ORDER BY OrderCount DESC
```

### Output

	restaurant_id	OrderCount
1	6	487
2	3	485
3	5	480
4	9	479
5	2	477
6	1	474
7	8	468
8	10	464
9	7	459
10	4	450





## Query Insights

->Dish Wise Order Count

### Query

```
select
    order_item,
    Count(DISTINCT order_id) as OrderCount
FROM Orders o
GROUP BY order_item
ORDER BY order_item
```

### Output

	order_item	OrderCount
1	Burger	428
2	Butter Chicken	333
3	Chicken Biryani	754
4	Chicken Shawarma	426
5	Chicken Tikka	111
6	Chole Bhature	410
7	Dal Makhani	446
8	Egg Curry	410
9	Fish Curry	315
10	Grilled Chicken Sandwich	300
11	Hyderabadi Biryani	450



## Query Insights

->Dish wise Highest Sales

### Query

```
SELECT
    order_item,
    sum(Total_amount) as TotalSales
FROM orders
GROUP BY order_item
ORDER BY TotalSales DESC
```

### Output

	order_item	TotalSales
1	Chicken Biryani	238767
2	Paneer Butter Masala	238280
3	Masala Dosa	236148
4	Pasta Alfredo	231529
5	Mutton Rogan Josh	230244
6	Vegetable Fried Rice	148675
7	Dal Makhani	145565
8	Hyderabadi Biryani	144343
9	Chicken Shawarma	141923
10	Mutton Biryani	140667

## Query Insights

->Delivered Order and Not Delivery Ordered

### Query

```
SELECT
  SUM(CASE
    WHEN delivery_status = 'Delivered'
    THEN 1 ELSE 0 END) AS Delivered,
  SUM(CASE
    WHEN delivery_status IS NOT NULL
    AND delivery_status = 'Not Delivered'
    THEN 1 ELSE 0 END) AS Not_Delivered
FROM
  deliveries;
```

### Output

	Delivered	Not_Delivered
1	8953	543



## Query Insights

->Rider Wise Delivered and Not Delivery Ordered status

### Query

```
SELECT
  rider_id,
  SUM(CASE WHEN delivery_status = 'Delivered' THEN 1 ELSE 0 END) AS Delivered,
  SUM(CASE WHEN delivery_status IS NOT NULL AND delivery_status = 'Not Delivered' THEN 1
ELSE 0 END) AS Not_Delivered
FROM
  deliveries
GROUP BY rider_id
Order by rider_id, Delivered DESC,Not_Delivered DESC
```

### Output

	rider_id	Delivered	Not_Delivered
1	1	717	41
2	2	767	38
3	3	721	42
4	4	683	46
5	5	328	25
6	6	286	25
7	7	288	20
8	8	321	28
9	9	699	43
10	10	735	43

The image features decorative geometric patterns in the corners, consisting of overlapping diagonal stripes in shades of blue and dark blue. The central text is flanked by three blue dots on each side, and a single blue diamond is positioned below the text.

...

Thank you

...

