```sql
1  SELECT * FROM swiggy_data
2
3  -- Data Validation & Cleaning
4  -- Null Check
5  SELECT
6      SUM(CASE WHEN State IS NULL THEN 1 ELSE 0 END) AS null_state,
7      SUM(CASE WHEN City IS NULL THEN 1 ELSE 0 END) AS null_city,
8      SUM(CASE WHEN Order_Date IS NULL THEN 1 ELSE 0 END) AS
          null_order_date,
9      SUM(CASE WHEN Restaurant_Name IS NULL THEN 1 ELSE 0 END) AS
          null_restaurant,
10     SUM(CASE WHEN Location IS NULL THEN 1 ELSE 0 END) AS null_location,
11     SUM(CASE WHEN Category IS NULL THEN 1 ELSE 0 END) AS null_category,
12     SUM(CASE WHEN Dish_Name IS NULL THEN 1 ELSE 0 END) AS null_dish,
13     SUM(CASE WHEN Price_INR IS NULL THEN 1 ELSE 0 END) AS null_price,
14     SUM(CASE WHEN Rating IS NULL THEN 1 ELSE 0 END) AS null_rating,
15     SUM(CASE WHEN Rating_Count IS NULL THEN 1 ELSE 0 END) AS
          null_rating_count
16 FROM swiggy_data;
17
18 --Blank or Empty Strings
19 SELECT *
20 FROM swiggy_data
21 WHERE
22 State ='' OR City='' OR Restaurant_Name='' OR Location='' OR Category=''
     OR Dish_Name=''
23
24
25 -- Duplicate Detection
26 -- Duplicate Detection
27 SELECT
28     State,City,order_date,restaurant_name,location,category,
29     dish_name,price_INR,rating,rating_count,
30     COUNT(*) AS CNT
31 FROM swiggy_data
32 GROUP BY
33     State,City,order_date,restaurant_name,location,category,
34     dish_name,price_INR,rating,rating_count
35 HAVING COUNT(*) > 1;
36
37
38 -- Delete Duplication
39 WITH CTE AS (
40 SELECT *, ROW_NUMBER() Over(
41      PARTITION BY State,City,order_date,restaurant_name,location,category,
42     dish_name,price_INR,rating,rating_count
43     ORDER BY (SELECT NULL)
44     ) AS rn
45     FROM swiggy_data
```

```sql
46      )
47      DELETE FROM CTE WHERE rn>1
48
49
50      -- CREATING SCHEMA
51      -- DIMENSION TABLES
52      -- DATE TABLE
53      CREATE TABLE dim_date (
54          date_id INT IDENTITY(1,1) PRIMARY KEY,
55          Full_Date DATE,
56          Year INT,
57          Month INT,
58          Month_Name varchar(20),
59          Quarter INT,
60          Day INT,
61          Week INT,
62          )
63
64
65      -- dim_location
66  CREATE TABLE dim_location (
67      location_id INT IDENTITY(1,1) PRIMARY KEY,
68      State VARCHAR(100),
69      City VARCHAR(100),
70      Location VARCHAR(200)
71  );
72
73  -- dim_restaurant
74  CREATE TABLE dim_restaurant (
75      restaurant_id INT IDENTITY(1,1) PRIMARY KEY,
76      Restaurant_Name VARCHAR(200)
77  );
78
79  -- dim_category
80  CREATE TABLE dim_category (
81      category_id INT IDENTITY(1,1) PRIMARY KEY,
82      Category VARCHAR(200)
83  );
84
85  -- dim_dish
86  CREATE TABLE dim_dish (
87      dish_id INT IDENTITY(1,1) PRIMARY KEY,
88      Dish_Name VARCHAR(200)
89  );
90
91
92  -- FACT TABLE
93  CREATE TABLE fact_swiggy_orders (
94      order_id INT IDENTITY(1,1) PRIMARY KEY,
```

```sql
 95
 96        date_id INT,
 97        Price_INR DECIMAL(10,2),
 98        Rating DECIMAL(4,2),
 99        Rating_Count INT,
100
101        location_id INT,
102        restaurant_id INT,
103        category_id INT,
104        dish_id INT,
105
106        FOREIGN KEY (date_id) REFERENCES dim_date(date_id),
107        FOREIGN KEY (location_id) REFERENCES dim_location(location_id),
108        FOREIGN KEY (restaurant_id) REFERENCES dim_restaurant(restaurant_id),
109        FOREIGN KEY (category_id) REFERENCES dim_category(category_id),
110        FOREIGN KEY (dish_id) REFERENCES dim_dish(dish_id)
111 );
112
113
114     SELECT * FROM fact_swiggy_orders
115
116     -- INSERT DATA IN TABLES
117     --dim_date
118     INSERT INTO dim_date (Full_Date, Year , Month , Month_Name ,
          Quarter , Day , Week)
119     SELECT DISTINCT
120          Order_Date,
121          YEAR(Order_Date),
122          MONTH(Order_Date),
123          DATENAME(MONTH, Order_Date),
124          DATEPART(QUARTER , Order_Date),
125          DAY(Order_Date),
126          DATEPART(WEEK , Order_Date)
127     FROM swiggy_data
128     WHERE Order_Date IS NOT NULL ;
129
130 SELECT * FROM dim_location
131
132 --dim_location
133 INSERT INTO dim_location ( State , City , location)
134 SELECT DISTINCT
135      State,
136      City,
137      Location
138 FROM swiggy_data;
139
140 --dim restaurant
141 INSERT INTO dim_restaurant (Restaurant_Name)
142 SELECT DISTINCT
```

```
143         Restaurant_Name
144  FROM swiggy_data;
145
146  --dim_category
147  INSERT INTO dim_category (Category)
148  SELECT DISTINCT
149      Category
150  FROM swiggy_data;
151
152  --dim_dish
153  INSERT INTO dim_dish (Dish_Name)
154  SELECT DISTINCT
155      Dish_Name
156  FROM swiggy_data;
157
158
159  --fact_table
160  INSERT INTO fact_swiggy_orders
161  (
162      date_id,
163      Price_INR,
164      Rating,
165      Rating_Count,
166      location_id,
167      restaurant_id,
168      category_id,
169      dish_id
170  )
171  SELECT
172      dd.date_id,
173      s.Price_INR,
174      s.Rating,
175      s.Rating_Count,
176
177      dl.location_id,
178      dr.restaurant_id,
179      dc.category_id,
180      dsh.dish_id
181  FROM swiggy_data s
182
183  JOIN dim_date dd
184      ON dd.Full_Date = s.Order_Date
185
186  JOIN dim_location dl
187      ON dl.State = s.State
188     AND dl.City = s.City
189     AND dl.Location = s.Location
190
191  JOIN dim_restaurant dr
```

```sql
192        ON dr.Restaurant_Name = s.Restaurant_Name
193
194 JOIN dim_category dc
195        ON dc.Category = s.Category
196
197 JOIN dim_dish dsh
198        ON dsh.Dish_Name = s.Dish_Name;
199
200
201 SELECT * FROM fact_swiggy_orders f
202 JOIN dim_date d ON f.date_id = d.date_id
203 JOIN dim_location l ON f.location_id = l.location_id
204 JOIN dim_restaurant r ON f.restaurant_id = r.restaurant_id
205 JOIN dim_category c ON f.category_id = c.category_id
206 JOIN dim_dish di ON f.dish_id = di.dish_id;
207
208 -- KPI'S
209 --Total Orders
210 SELECT COUNT(*) AS Total_Orders
211 FROM fact_swiggy_orders
212
213 -- Total Revenue ( INR Million )
214 SELECT
215 FORMAT(SUM(CONVERT(FLOAT,price_INR))/1000000, 'N2') + ' INR Million '
216 AS Total_Revenue
217 FROM fact_swiggy_orders
218
219 -- Average Dish Price
220 SELECT
221 FORMAT(AVG(CONVERT(FLOAT,price_INR)), 'N2') + ' INR '
222 AS Total_Revenue
223 FROM fact_swiggy_orders
224
225 --Average Rating
226 SELECT
227 AVG(Rating) AS Avg_Rating
228 FROM fact_swiggy_orders
229
230
231 --Deep-Dive Business Analysis
232
233 -- Monthly Order Trends
234 SELECT
235 d.year,
236 d.month,
237 d.month_name,
238 count(*) AS Total_Orders
239 FROM fact_swiggy_orders f
240 JOIN dim_date d ON f.date_id = d.date_id
```

```sql
241 GROUP BY d.year,
242 d.month,
243 d.month_name
244 ORDER BY count(*) DESC
245
246 -- QUARTERLY TRENDS
247 SELECT
248 d.year,
249 d.quarter,
250 count(*) AS Total_Orders
251 FROM fact_swiggy_orders f
252 JOIN dim_date d ON f.date_id = d.date_id
253 GROUP BY d.year,
254 d.quarter
255 ORDER BY count(*) DESC
256
257
258 --Yearly Trends
259 SELECT
260 d.year,
261 count(*) AS Total_Orders
262 FROM fact_swiggy_orders f
263 JOIN dim_date d ON f.date_id = d.date_id
264 GROUP BY d.year
265 ORDER BY count(*) DESC
266
267
268 -- Orders by Day of Week (Mon-Sun)
269
270 SELECT
271     DATENAME(WEEKDAY, d.full_date) AS day_name,
272     COUNT(*) AS total_orders
273 FROM fact_swiggy_orders f
274 JOIN dim_date d
275     ON f.date_id = d.date_id
276 GROUP BY
277     DATENAME(WEEKDAY, d.full_date),
278     DATEPART(WEEKDAY, d.full_date)
279 ORDER BY
280     DATEPART(WEEKDAY, d.full_date);
281
282 --Top 10 cities by order volume
283 SELECT TOP 10
284 l.city,
285 COUNT(*) AS Total_Orders FROM fact_swiggy_orders f
286 JOIN dim_location l
287 ON l.location_id = f.location_id
288 GROUP BY l.city
289 ORDER BY COUNT (*) DESC
```

```sql
290
291  --BOTTOM 10 cities by order volume
292  SELECT TOP 10
293  l.city,
294  COUNT(*) AS Total_Orders FROM fact_swiggy_orders f
295  JOIN dim_location l
296  ON l.location_id = f.location_id
297  GROUP BY l.city
298  ORDER BY COUNT (*) ASC
299
300  --Top 10 cities by Revenue
301  SELECT TOP 10
302  l.city,
303  SUM(f.price_INR) AS Revenue FROM fact_swiggy_orders f
304  JOIN dim_location l
305  ON l.location_id = f.location_id
306  GROUP BY l.city
307  ORDER BY SUM(f.price_INR) DESC
308
309  -- Revenue Contribution by States
310  SELECT
311  l.State,
312  SUM(f.price_INR) AS Revenue FROM fact_swiggy_orders f
313  JOIN dim_location l
314  ON l.location_id = f.location_id
315  GROUP BY l.State
316  ORDER BY SUM(f.price_INR) DESC
317
318
319  -- Top 10 restaurants by orders
320  SELECT TOP 10
321  r.restaurant_name,
322  SUM(f.price_INR) AS Revenue FROM fact_swiggy_orders f
323  JOIN dim_restaurant r
324  ON r.restaurant_id = f.restaurant_id
325  GROUP BY r.restaurant_name
326  ORDER BY SUM(f.price_INR) DESC
327
328
329  -- TOP Categories by Order Volume
330  SELECT
331      c.category,
332      COUNT(*) AS total_orders
333  FROM fact_swiggy_orders f
334  JOIN dim_category c
335      ON f.category_id = c.category_id
336  GROUP BY c.category
337  ORDER BY total_orders DESC;
338
```

```sql
339
340  -- Most Ordered Dishes
341  SELECT TOP 10
342      d.dish_name,
343      COUNT(*) AS order_count
344  FROM fact_swiggy_orders f
345  JOIN dim_dish d
346      ON f.dish_id = d.dish_id
347  GROUP BY d.dish_name
348  ORDER BY order_count DESC;
349
350  --Cuisine Performance (orders + avg rating )
351  SELECT
352      c.category,
353      COUNT(*) total_orders,
354      AVG(CONVERT(FLOAT,f.rating)) AS avg_rating
355  FROM fact_swiggy_orders f
356  JOIN dim_category c ON f.category_id = c.category_id
357  GROUP BY c.category
358  ORDER BY total_orders DESC;
359
360
361  -- Total Orders by Price Range
362  SELECT
363      CASE
364          WHEN CONVERT(FLOAT, price_inr) < 100 THEN 'Under 100'
365          WHEN CONVERT(FLOAT, price_inr) BETWEEN 100 AND 199 THEN '100 -
                199'
366          WHEN CONVERT(FLOAT, price_inr) BETWEEN 200 AND 299 THEN '200 -
                299'
367          WHEN CONVERT(FLOAT, price_inr) BETWEEN 300 AND 499 THEN '300 -
                499'
368          ELSE '500+'
369      END AS price_range,
370      COUNT(*) AS total_orders
371  FROM fact_swiggy_orders
372  GROUP BY
373      CASE
374          WHEN CONVERT(FLOAT, price_inr) < 100 THEN 'Under 100'
375          WHEN CONVERT(FLOAT, price_inr) BETWEEN 100 AND 199 THEN '100 -
                199'
376          WHEN CONVERT(FLOAT, price_inr) BETWEEN 200 AND 299 THEN '200 -
                299'
377          WHEN CONVERT(FLOAT, price_inr) BETWEEN 300 AND 499 THEN '300 -
                499'
378          ELSE '500+'
379      END
380  ORDER BY total_orders DESC;
381
```

```sql
--Rating Count Distribution (1-5)
SELECT
    rating,
    COUNT(*) AS rating_count
    FROM fact_swiggy_orders
    GROUP BY rating
    ORDER BY COUNT(*) DESC;
```