

DELIVERY MANAGEMENT SYSTEM

Team Go_pro

Group member:

Raunak Singh Matharu - 002785039

Vatsal Doshi - 002776613

Bhavanish Dhamnaskar - 002792028

Samiksha Mundas - 002776646

Description:

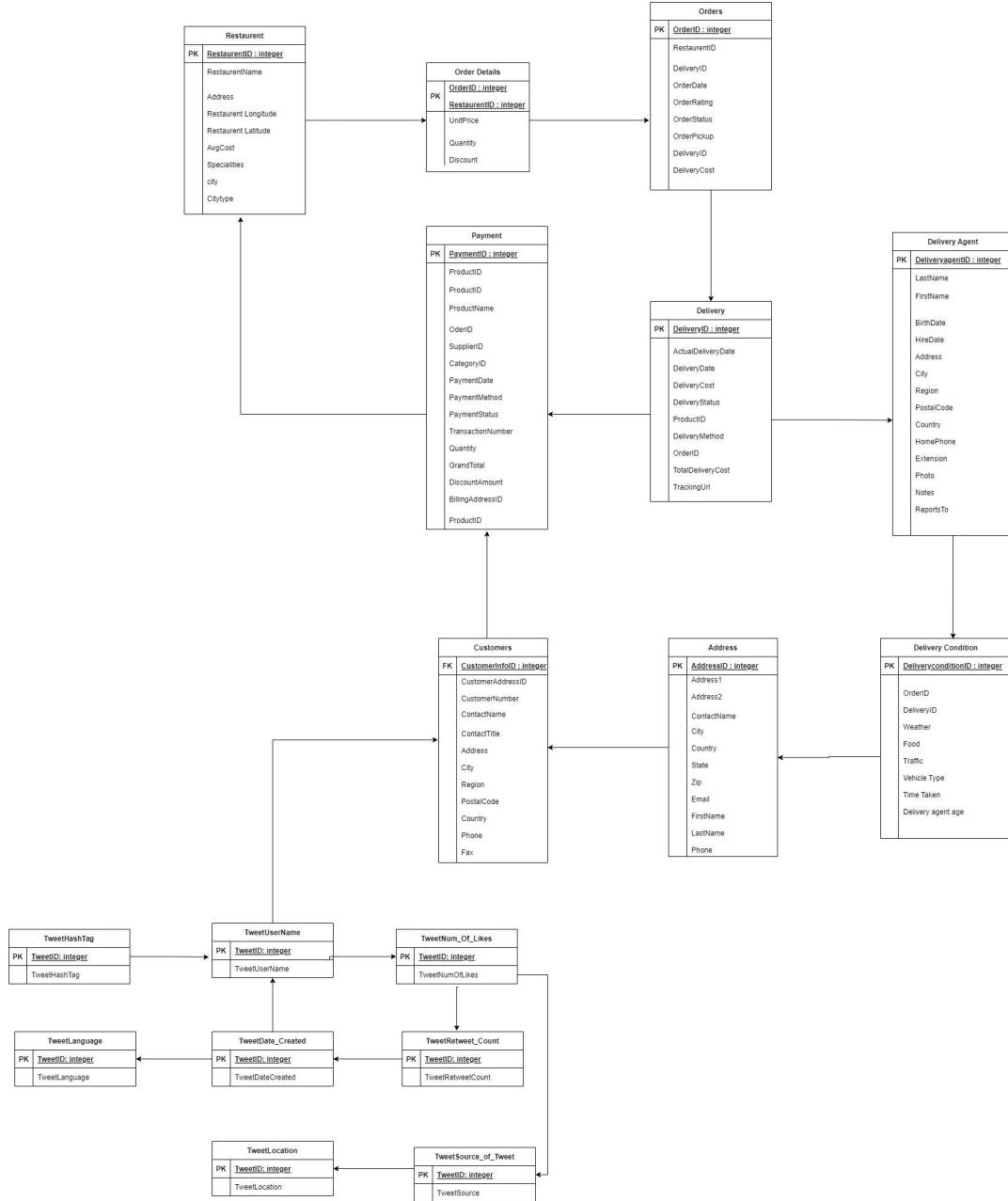
Our project is about developing a food delivery system that will allow your businesses to accept and handle online orders for delivery or takeout. Customers use an app or website to explore a digital menu and place and pay for their order online. The order data will then be received by the venue using their preferred online food ordering system, and the order will be prepared for delivery or client pickup. While internet meal ordering systems have been available for some time, demand for these technology skyrocketed during the epidemic. With lengthier lockdowns and limitations, hospitality establishments immediately shifted their operations to include online ordering alternatives so that their clients could continue to enjoy restaurant-quality cuisine at home.

To place an order through our delivery system, the customer must first fill out his information, which is then saved in the database. The consumer may then completely browse eateries and order anything he desires. The order information will be shown appropriately, and the order will be verified and delivered in accordance with the details contained in the database. Thus, the order will be received by the client via the delivery man whose details will be saved in the database. Our system also gathers information about delivery conditions such as weather, traffic, and the time it takes the deliveryman in such settings.

The flow of our project was as follows: initially, we examined the data; next, we scraped the data from Twitter, gathered the apis, and connected it via tweepy; and last, we scraped data related to zomato and grubhub. Its characteristics were attached. Then we got the datasets, utilized a beautiful soup library, and used regex to validate the data. The data was then displayed using matplotlib in the form of a bar graph and a pie chart. Our final step was to normalize the data in order to reduce redundant data, decrease errors, and simplify the query process. As a result, data will be standardized and workflow will be enhanced.

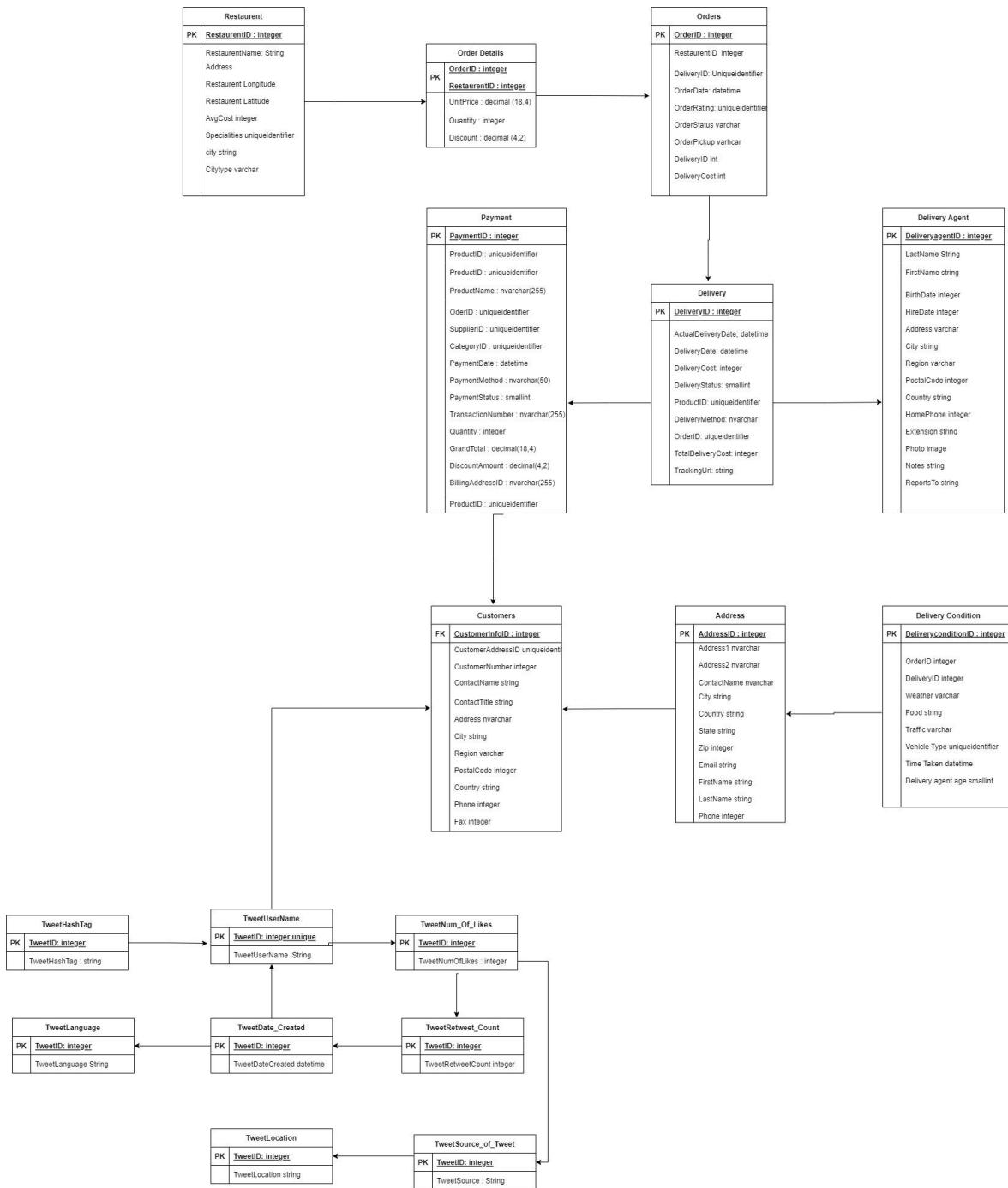
ER Diagrams

UML Diagram



<https://drive.google.com/file/d/1jH2Wb5vjCpIT2MdztIWgBBWTcj5KUJUO/view?usp=sharing>

Conceptual Diagram



<https://drive.google.com/file/d/1YSONQmRD7rLJsKz7THjPSJ0-AcV-SLKD/view?usp=sharing>

The flow of the project

We started the project in steps which started from:

1. Twitter scraping,
2. Data gathering and data munging
3. Normalisation:

Chapter 1: Twitter Scraping:

1. We scraped the data from Twitter using the tweepy library and we choose Zomato and Grubhub account for the scraping and also scraped using the hashtags.
2. We used tweepy attributes like
 - A. tweet.id, to get data of tweet account id from which user has tweeted
 - B. tweet.created_at,to get information when was the tweet time
 - C. tweet.created_at,to get information when was the tweet time
 - D. tweet.created_at,to get information when was the tweet time
 - E. tweet.favorite_count, to get information of tweet like
 - F. tweet.source, to get information from which source user has tweeted (ios phone, android phone,browser)
 - G. tweet.text, to get information of what is users text
 - H. tweet.text, to get information of what is users text
 - I. tweet.lang,to get information of what was tweet language
 - J. tweet.user.screen_name, , to get information of what is the username of user id
 - K. tweet.in_reply_to_status_id_str, , to get information of of what are reply to tweets
 - L. tweet.retweet_count, , to get information of of tweet count
 - M. tweet.entities, , to get information of hashtag used in tweets
 - N. tweet.user.location, , to get information on user's location at the tweet time
 - O. tweet.user.created_at to get information when did user join the Twitter
- 3.We drew a UML diagram and ER diagram accordingly

Steps followed :

Steps followed by us were

- Collected all the API key required to connect to twitter
- Connected to twitter using "tweepy"
- Scrapped the data related to twitter profile which was zomato and Grubhub
- Data was appended to list on all the attributes
- Created a Dictionary and then converted it to data frame using pandas library of the extracted data
- Separating the hashtag list
- Connection to sqlite and pushing data to twitter data in sql

Chapter 2.

Data Scraping , Data Munging,Data cleaning, Visualisation:

- 1) We have downloaded dataset from Kaggle, and then split this data into three tables:
 - a) Order
 - b) Delivery
 - c) Restaurant
- 2) We Web scrapped the data from the Website using **beautifulsoup library** and then converted thisa to csv file. We imported this CSV file and then concatenated the web-scraped data with the Kaggle data.

RESULT:

- 1) We extracted the data from the Web using beautifulsoup library .
- 2) We cleaned the data using REGEX and replace function.
- 3) We visualized the data using matplot library(bar graph and pie chart). We used Seaborn library for correlation graph and pair-plot graph.

CODE:

- 1) We used 'beautiful soup' library to extract web data using the function 'find all' in which we included tags and class of a website.
- 2) We validated the data using REGEX, unique, drop and replace function.
- 3) Plotted the bar graph and pie chart using the matplot lib which consisted the count of particular data and numeric value of the column of the data frame.
- 4) Concatenated web scraped data and Kaggle data

Kaggle Source:

- 1) <https://www.kaggle.com/datasets/gauravmalik26/food-delivery-dataset>
- 2) <https://www.kaggle.com/datasets/rishikeshkonapure/zomato>

OUTPUT:

Concatenating Webscraped data and kaggle Data

Before :

In [15]:	1	df_Restaurent					
Out[15]:							
	Order_ID	Restaurent	Speciality	Rating	Avgcost	Timetaken	City
0	0x4607	Le Petit Souffle	Mutton Galouti Kabab, Kali Mirch Chicken, Kaba...	4.9	1200	24	Mumbai
1	0xb379	Izakaya Kikufuji	Onion Naan, Kurkure Momo, Kadai Paneer, Shahi ...	4.5	1600	33	Bangalore
2	0x5d6d	Heat - Edsa Shangri-La	Chocolate Truffle Cake, Cakes, Tea, Breads, Ch...	4.4	700	26	Bhopal
3	0x7a6a	Ooma	Tawa Chicken, Chicken Curry, Paneer Lababdar, ...	4.7	645	21	Ahmedabad
4	0x70a2	Sambo Kojin	Coconut Milk Dessert, Honey Caviar, Kaffir Lim...	4.6	745	30	Goa
5	0x9bb4	Din Tai Fung	Irish Cream Tiramisu, Penne Vodka, Mississippi...	4.8	800	26	Amritsar
6	0x95b4	Buffet 101	Malleshwari Dosa, Benne Khali Dosa, Obbattu, R...	4.7	695	40	Allahabad

```
In [20]: 1 df_Web_Scrape
```

Out[20]:

	Order_ID	Restaurent	Speciality	Rating	Avgcost	Timetaken	City
0	f5ec435b-85f3-4489-8e55-8ebfc110d80c	Fast & Fresh	Pizzas	5.0	400	59	Mumbai
1	6c1844db-ab28-4beb-83a2-debdb7825d71	Sizzling Spices	Chinese, Biryani, Thai, Indian	5.0	300	57	Mumbai
2	b6856455-3105-4655-9634-6ef059070c09	Sandwich Shack	Snacks	5.0	300	50	Mumbai
3	bfc47c73-3304-4879-b215-393df8f3e659	House Of Beriyan	Biryani, Mughlai	5.0	500	47	Mumbai
4	27e5950d-7636-4ce0-935b-72e40e4538a4	FB Cafe By Frozen Bottle	Desserts, Pizzas, Bakery	5.0	500	33	Mumbai
5	5128774a-da4c-40ab-b67e-fc0fd7e46f96	Delhi Zayaka	Indian, Biryani, Chinese	4.9	400	53	Mumbai
6	2aad34d6-8004-4b3b-b31d-580db2a5521e	Ice Cream Works	Ice Cream, Desserts	4.9	300	41	Mumbai

After:

```
In [21]: 1 df4=pd.concat ([df_Restaurent, df_Web_Scrape], axis=0, ignore_index=True)
```

In [22]: 1 df4

994	0xa644	Anand Dhaba	Tandoori Jalandhari, Chicken Jalandhari, Eggs,...	4.7	1450	25	Gurgaon
995	0x8dc0	Bikaneri Sweets & Restaurant	Masala Pao Bhaji, Tawa Pulao, Dahi Puri, Bhel ...	4.6	750	39	Ahmedabad
996	0x8753	Delicieux Ice Cream Rolls	Chocolate Almond, Chocolate Coffee, Eggless Ca...	4.9	150	10	Bangalore
997	0xd71b	Flying Tandoor	Live Kitchen, Lovely Ambience, Live Music, Ama...	4.4	450	40	Gurgaon
998	0x9247	Food Station	Kiwi Blast, Brewed Wheat Beer, Chocolate Pan, ...	4.9	750	41	Agra
999	f5ec435b-85f3-4489-8e55-8ebfc110d80c	Fast & Fresh	Pizzas	5.0	400	59	Mumbai
1000	6c1844db-ab28-4beb-83a2-debdb7825d71	Sizzling Spices	Chinese, Biryani, Thai, Indian	5.0	300	57	Mumbai
1001	b6856455-3105-4655-9634-6ef059070c09	Sandwich Shack	Snacks	5.0	300	50	Mumbai

Cleaning Data:

Before (Webscraped data):

	Order_ID	Restaurent	Speciality	Rating	Av_cost	Timetaken	City
0	541b09d7-6dd7-4524-9007-5b3aab1c461f	Fast & Fresh	Pizzas	5.0	₹400 FOR TWO	64	Mumbai
1	eaaf1ae4-d2c5-4381-bc0a-3135fadd6346	Sizzling Spices	Chinese, Biryani, Thai, Indian	5.0	₹300 FOR TWO	61	Mumbai
2	2b274f26-f8e1-4aff-931a-b8d1e65f5151	Sandwich Shack	Snacks	5.0	₹300 FOR TWO	52	Mumbai
3	8fa3b8e4-86dc-48a0-8a99-fe1eee6754a5	House Of Beriyan	Biryani, Mughlai	5.0	₹500 FOR TWO	52	Mumbai
4	c35402aa-2fba-4f25-a74b-e8539fdc9881	FB Cafe By Frozen Bottle	Desserts, Pizzas, Bakery	5.0	₹500 FOR TWO	34	Mumbai
...
59	96f52944-cca5-4cda-ad40-4e10ff60fb	Elementaria Bakery & Cafe	Desserts, Bakery	4.5	₹800 FOR TWO	26	Mumbai
60	594e31fb-942f-44aa-b8b4-8f91c7b0eb27	Dabba Garam (Homely Meals & More)	Home Food, North Indian, Thalis	4.5	₹150 FOR TWO	28	Mumbai
61	cbffb671-128c-4ce7-96fb-d38acf7b9247	Mad Over Donuts	Desserts, Sweets, Bakery	4.5	₹200 FOR TWO	26	Mumbai
62	7f077095-db35-47a3-ab5f-0402011f1f1f	Baskin Robbins	Desserts, Ice Cream	4.5	₹250 FOR TWO	24	Mumbai

In [52]: 1 List_Timetaken

Out[52]: ['5.0•64 MINS•₹400 FOR TWO',
 '5.0•61 MINS•₹300 FOR TWO',
 '5.0•52 MINS•₹300 FOR TWO',
 '5.0•52 MINS•₹500 FOR TWO',
 '5.0•34 MINS•₹500 FOR TWO',
 '4.9•43 MINS•₹300 FOR TWO',
 '4.9•42 MINS•₹600 FOR TWO',
 '4.8•57 MINS•₹250 FOR TWO',
 '4.8•65 MINS•₹250 FOR TWO',
 '4.8•46 MINS•₹200 FOR TWO',
 '4.8•55 MINS•₹300 FOR TWO',
 '4.8•30 MINS•₹500 FOR TWO',
 '4.8•40 MINS•₹350 FOR TWO',
 '4.8•25 MINS•₹100 FOR TWO',
 '4.7•49 MINS•₹300 FOR TWO',
 '4.7•48 MINS•₹500 FOR TWO',
 '4.7•60 MINS•₹300 FOR TWO',
 '4.7•65 MINS•₹250 FOR TWO',
 '4.7•52 MINS•₹150 FOR TWO',
 '4.7•61 MINS•₹300 FOR TWO',
 '4.7•37 MINS•₹400 FOR TWO',
 '4.7•32 MINS•₹300 FOR TWO',
 '4.7•38 MINS•₹300 FOR TWO',
 '4.7•31 MINS•₹150 FOR TWO',
 '4.7•32 MINS•₹300 FOR TWO',
 '4.7•25 MINS•₹250 FOR TWO',
 '4.7•25 MINS•₹300 FOR TWO']

After (Webscraped data):

```
In [66]: 1 #used to clean special character like ₹ in dataframe column Avgcost
2 df['Av_cost'] = df['Av_cost'].str.replace('₹', '', regex=True)

In [67]: 1 df
2 #df["Timetaken"].replace("57Mins", "Mins")

Out[67]:
```

	Order_ID	Restaurent	Speciality	Rating	Av_cost	Timetaken	City
0	541b09d7-6dd7-4524-9007-5b3aab1c461f	Fast & Fresh	Pizzas	5.0	400FORTWO	64	Mumbai
1	eaaf1ae4-d2c5-4381-bc0a-3135fadd6346	Sizzling Spices	Chinese, Biryani, Thai, Indian	5.0	300FORTWO	61	Mumbai
2	2b274f26-f8e1-4aff-931a-b8d1e65f5151	Sandwich Shack	Snacks	5.0	300FORTWO	52	Mumbai
3	8fa3b8e4-86dc-48a0-8a99-fe1eee6754a5	House Of Beriyan	Biryani, Mughlai	5.0	500FORTWO	52	Mumbai
4	c35402aa-2fba-4f25-a74b-e8539fdc9881	FB Cafe By Frozen Bottle	Desserts, Pizzas, Bakery	5.0	500FORTWO	34	Mumbai
...
59	96f52944-cca5-4cda-ad40-4e10ff60fb	Elementaria Bakery & Cafe	Desserts, Bakery	4.5	800FORTWO	26	Mumbai
60	594e31fb-942f-44aa-b8b4-8f91c7b0eb27	Dabba Garam (Homely Meals & More)	Home Food, North Indian, Thalis	4.5	150FORTWO	28	Mumbai

```
64 rows × 7 columns

In [68]: 1 #used to remove FORTWO Letters in Avgcost Column in DataFrame
2 df.replace("FORTWO", " ", regex = True)

In [69]: 1 df

Out[69]:
```

	Order_ID	Restaurent	Speciality	Rating	Av_cost	Timetaken	City
0	541b09d7-6dd7-4524-9007-5b3aab1c461f	Fast & Fresh	Pizzas	5.0	400	64	Mumbai
1	eaaf1ae4-d2c5-4381-bc0a-3135fadd6346	Sizzling Spices	Chinese, Biryani, Thai, Indian	5.0	300	61	Mumbai
2	2b274f26-f8e1-4aff-931a-b8d1e65f5151	Sandwich Shack	Snacks	5.0	300	52	Mumbai
3	8fa3b8e4-86dc-48a0-8a99-fe1eee6754a5	House Of Beriyan	Biryani, Mughlai	5.0	500	52	Mumbai
4	c35402aa-2fba-4f25-a74b-e8539fdc9881	FB Cafe By Frozen Bottle	Desserts, Pizzas, Bakery	5.0	500	34	Mumbai
...
59	96f52944-cca5-4cda-ad40-4e10ff60fb	Elementaria Bakery & Cafe	Desserts, Bakery	4.5	800	26	Mumbai

In [71]:

```
1 #used to replace the blank space with not available
2 df = df.replace("", "Not Available ", regex = True)
```

In [72]:

23	5e20985f-0553-43c0-0418-26bd396b9a2d	Moussestruck	Ice Cream, Desserts, Sweets	4.7	150	31	Mumbai		
24	d7aab2d1-da46-437a-9533-06ed1fec822f	99 Pancakes	Not Available	4.7	300	32	Mumbai		
25	7df097b0-4d4d-4e30-a972-80db63b24887	Giani's	Not Available	4.7	250	25	Mumbai		
26	99a95d29-eeab-4063-9ad3-ddae04dec1f	Haagen Dazs	Desserts	4.7	400	25	Mumbai		
27	99d6a514-20ad-435c-a868-cde28cbc441	Apsara Ice Creams	Ice Cream, Desserts	4.7	150	23	Mumbai		
28	63c6e0b6-0e9a-4f3d-b1fb-9a1338319c56	The Canary by Origin	Fast Food, British, Italian	4.6	600	63	Mumbai		
29	9468c01d-a464-4f13-a7f2-95bfaaf39f5b	Yummy Business	Snacks, Fast Food, Beverages	4.6	250	60	Mumbai		

In [53]:

```
1 # data cleaning only taking the 5 and 6 index of List_Timetaken to extract time taken
2 List_Timetaken1=[]
3 for i in range(len(List_Timetaken)):
4     List_Timetaken1.append(List_Timetaken[i][4:6])
5 #List_Timetaken1.append
6 List_Timetaken1
7
```

Out[53]:

```
['64',
 '61',
 '52',
 '52',
 '34',
 '43',
 '42',
 '57',
 '65',
 '46',
 '55',
 '30',
 '40',
 '25',
 '49',
 '48',
 '60',
 '65',
 '52',
```

Before (Kaggle data):

	Address	OrderDate	TimeOrderd	TimeOrderpicked	Weatherconditions	Roadtrafficdensity	Typeofvehicle	Timetaken	Avcost
1	6 & 7, Upvan Building, Near Indian Oil Colony,...	19-03- 2022	11:30:00	11:45:00	conditions Sunny	High	motorcycle	(min) 24	1200
2	BluPetal Hotel, 60 Jyoti Nivas College Road, K...	25-03- 2022	19:45:00	19:50:00	conditions Stormy	Jam	scooter	(min) 33	1600
3	12-13, Food Court, DB City Mall, Maharana Prat...	19-03- 2022	08:30:00	08:45:00	conditions Stormy	Low	motorcycle	(min) 26	700
4	Opposite Sindhu Bhawan, Bodakdev,	05-04- 2022	18:00:00	18:10:00	conditions Sunny	Medium	motorcycle	(min) 21	645
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									
31									
32									
33									
34									
35									
36									
37									
38	0x4bda	01-04-2022	09:45:00	09:55:00	11	Sunny			
39	0x9d26	28-03-2022	08:40:00	08:55:00	11	Stormy			
40	0xb18	18-03-2022	23:00:00	23:10:00	28	Fog			
41	0xd99	14-03-2022	17:25:00	17:30:00	33	Cloudy			
42	0xf0	06-04-2022	Nan	18:35:00	33	Cloudy			
43	0xb796	04-04-2022	19:45:00	19:50:00	52	Sandstorms			
44	0x85b4	08-03-2022	19:10:00	19:25:00	22	Stormy			
45	0xc644	13-02-2022	10:55:00	11:00:00	16	Fog			
46	0x6999	24-03-2022	21:40:00	21:45:00	11	Sunny			
47	0x63b6	19-03-2022	19:00:00	19:15:00	25	Windy			

ID	Address	OrderDate	TimeOrderd	TimeOrderpicked	Weatherconditions	Roadtrafficdensity
38	0x4bda	01-04-2022	09:45:00	09:55:00	11	Sunny
39	0x9d26	28-03-2022	08:40:00	08:55:00	11	Stormy
40	0xb18	18-03-2022	23:00:00	23:10:00	28	Fog
41	0xd99	14-03-2022	17:25:00	17:30:00	33	Cloudy
42	0xf0	06-04-2022	Nan	18:35:00	33	Cloudy
43	0xb796	04-04-2022	19:45:00	19:50:00	52	Sandstorms
44	0x85b4	08-03-2022	19:10:00	19:25:00	22	Stormy
45	0xc644	13-02-2022	10:55:00	11:00:00	16	Fog
46	0x6999	24-03-2022	21:40:00	21:45:00	11	Sunny
47	0x63b6	19-03-2022	19:00:00	19:15:00	25	Windy

After(Kaggle data) :

```
In [4]: 1 df = df.replace("(min)", " ",regex = True)
2 df = df.replace("conditions", " ",regex = True)

In [5]: 1 df['Timetaken'] = df['Timetaken'].str.replace('\W', ' ', regex=True)

In [6]: 1 pd.set_option('display.max_columns', None)
2 pd.set_option('display.max_rows', None)

In [7]: 1 df
```

Type	Address	OrderDate	TimeOrderd	TimeOrderpicked	Weatherconditions	Roadtrafficdensity	Typeofvehicle	Timetaken	Avcost
Urban	6 & 7, Upvan Building, Near Indian Oil Colony,....	19-03-2022	11:30:00	11:45:00	Sunny	High	motorcycle	24	1200
Urban	BluPetal Hotel, 60 Jyoti Nivas College Road, K...	25-03-2022	19:45:00	19:50:00	Stormy	Jam	scooter	33	1600
Urban	12-13, Food Court, DB City Mall, Maharana Prat...	19-03-2022	08:30:00	08:45:00	Stormy	Low	motorcycle	26	700

```
In [4]: 1 df = df.replace("(min)", " ",regex = True)
2 df = df.replace("conditions", " ",regex = True)

In [5]: 1 df['Timetaken'] = df['Timetaken'].str.replace('\W', ' ', regex=True)

In [6]: 1 pd.set_option('display.max_columns', None)
2 pd.set_option('display.max_rows', None)

In [7]: 1 df
```

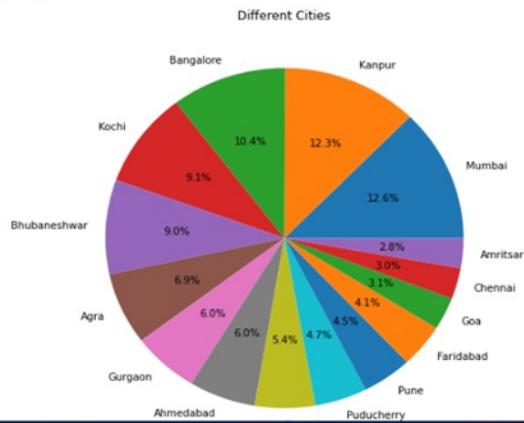
Type	Address	OrderDate	TimeOrderd	TimeOrderpicked	Weatherconditions	Roadtrafficdensity	Typeofvehicle	Timetaken	Avcost
Urban	6 & 7, Upvan Building, Near Indian Oil Colony,....	19-03-2022	11:30:00	11:45:00	Sunny	High	motorcycle	24	1200
Urban	BluPetal Hotel, 60 Jyoti Nivas College Road, K...	25-03-2022	19:45:00	19:50:00	Stormy	Jam	scooter	33	1600
Urban	12-13, Food Court, DB City Mall, Maharana Prat...	19-03-2022	08:30:00	08:45:00	Stormy	Low	motorcycle	26	700

Visualisation:

Pie chart for Different cities

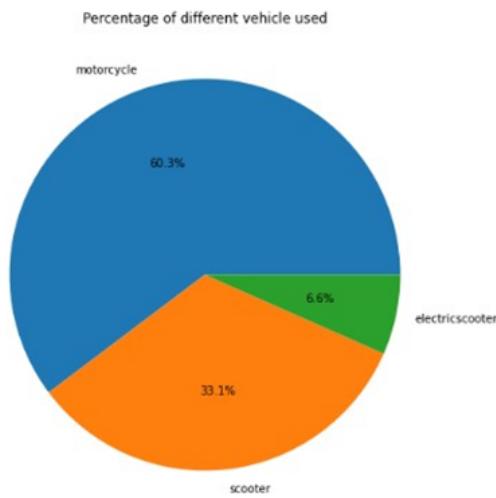
```
In [27]: 1 df_Order.to_csv('./Order_Table.csv', index = False)

In [28]: 1 plt.figure(figsize = (50,8))
2 df4["City"].value_counts().sort_values(ascending=False).head(15).plot.pie(y="Headquarters", autopct="%0.1f%%")
3 plt.title("Different Cities")
4 plt.axis("off")
5 plt.show()
<
```



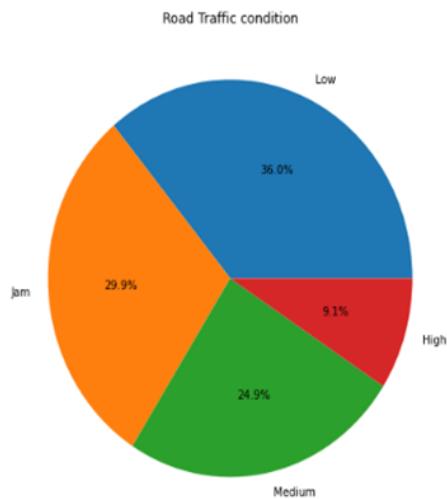
Pie chart for Different Vehicle used

```
In [29]: 1 plt.figure(figsize = (50,8))
2 df_Delivery["Vehicle"].value_counts().sort_values(ascending=False).head(15).plot.pie(y="Headquarters", autopct="%0.1f%%")
3 plt.title("Percentage of different vehicle used")
4 plt.axis("off")
5 plt.show()
```



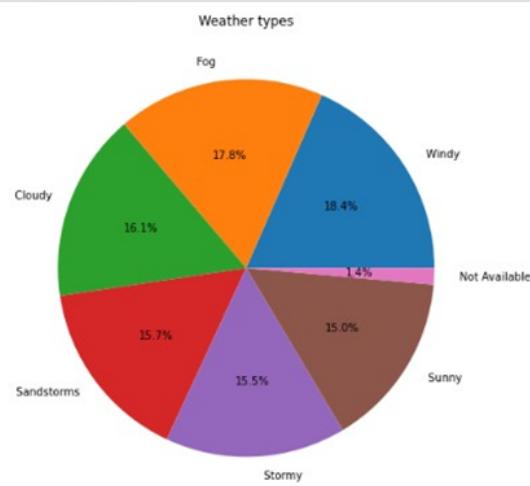
Pie chart for Traffic condition

```
In [30]: 1 plt.figure(figsize = (50,8))
2 df_Delivery[\"Road_traffic_density\"].value_counts().sort_values(ascending=False).head(15).plot.pie(y=\"Headquarters\", autopct="
3 plt.title(\"Road Traffic condition\")
4 plt.axis(\"off\")
5 plt.show()
```



Pie chart for Weather types

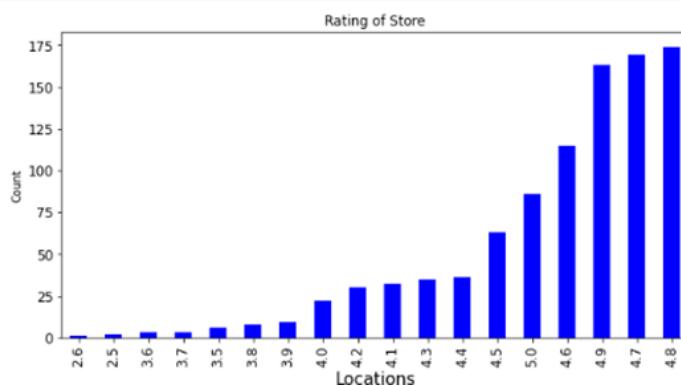
```
In [68]: 1 plt.figure(figsize = (50,8))
2 df_order["Weather"].value_counts().sort_values(ascending=False).head(15).plot.pie(y="Headquarters", autopct="%0.1f%%")
3 plt.title("Weather types")
4 plt.axis("off")
5 plt.show()
```



```
In [69]: 1 plt.figure(figsize = (10,5))
```

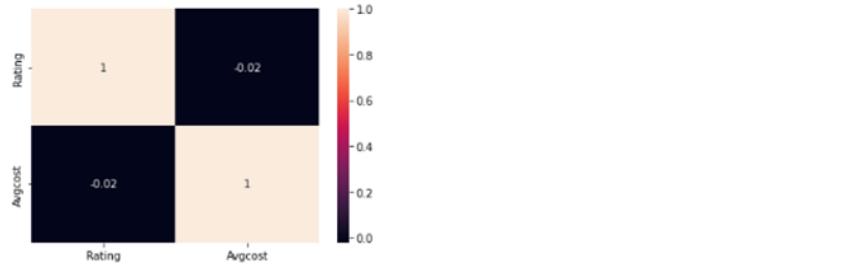
Bar plot for rating of store with count

```
In [32]: 1 plt.figure(figsize = (10,5))
2 df["DeliverypersonRatings"].value_counts().sort_values(ascending=True).head(20).plot.bar(color= "Blue", fontsize=12)
3 plt.title("Rating of Store")
4 plt.xlabel("Locations",fontsize=15)
5 plt.ylabel("Count")
6 plt.show()
```



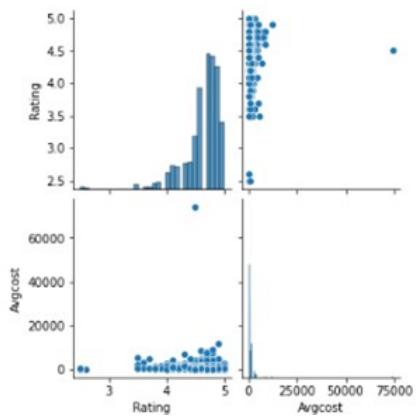
Seaborn corelation

```
In [65]: 1 corelation = df4.corr()
2 sns.heatmap(corelation, xticklabels=corelation.columns, yticklabels=corelation.columns, annot=True)
Out[65]: <AxesSubplot:>
```



Seaborn Pairplot

```
In [66]: 1 sns.pairplot(df4)
Out[66]: <seaborn.axisgrid.PairGrid at 0x21e28f6ed90>
```



Chapter 3: NORMALISATION:

1NF:

Before

Before delivery name and specialities column was not in normalized form

CityType	Address	OrderDate	TimeOrderd	TimeOrderpicked	Climate	Roadtrafficdensity	Typeofvehicle	Timetaken	Avcost	Delivery_Name	DeliveryID
Metropolitan	1/48, Delhi Gate, Station Road, Raja Mandi, Ci...	27-03-2022	19:30:00	19:45:00	Windy	Jam	scooter	39	680	Oliver, Brooke	10078
Metropolitan	ITC Mughal, Fatehabad Road, Tajganj, Agra	02-04-2022	22:35:00	22:45:00	Cloudy	Low	scooter	30	800	Onque, Jasmine	10121
Metropolitan	ITC Mughal, Fatehabad Road, Tajganj, Agra	23-03-2022	21:25:00	21:40:00	Stormy	Jam	motorcycle	39	2200	Owad, Clinton	10281
Metropolitan	Courtyard by Marriott Agra, Phase 2.	01-03-2022	10:25:00	10:40:00	Stormy	Low	motorcycle	15	1200	Ozark, Travis	10041

Out[4]:	OrderID	DeliverypersonAge	DeliverypersonRatings	Specialties	RestaurantName	Restaurentid	Ci
	0 0xb816	33	4.3	Cheese Souffle, Fettucine, Scallops, Gnocchi, ...	Chez Michou	42	Ag
	1 0x30c8	32	4.6	Panneer Butter Masala, Noodle, Chilli Garlic N...	Taco Pep	43	Ag
	2 0x95fb	25	4.2	Chilli Garlic Bread, Choco Lava Cake, Pastas, ...	Harvest Moon	45	Ag
	3 0x4863	25	4.6	Pineapple Curry, Vindaloo, ..	Last Resort Grill	46	Ag

DATASET AFTER:

Speciality column is separated in rows

In [19]:	1	Df1									
Out[19]:											
	Order_ID	Restaurant_Id	Restaurant_Name	Speciality	City	Address	Avg_Cost	DeliveryID	Agent_FirstName	Agent_LastName	Agent_Age
0	0xb816	42	Chez Michou	Cheese Souffle	Agra	1/48, Delhi Gate, Station Road, Raja Mandi, Ci...	680	10078	Oliver	Brooke	33
1	0xb816	42	Chez Michou	Fettucine	Agra	1/48, Delhi Gate, Station Road, Raja Mandi, Ci...	680	10078	Oliver	Brooke	33
2	0x30c8	43	Taco Pep	Panner Butter Masala	Agra	ITC Mughal, Fatehabad Road, Tajganj, Agra	800	10121	Onque	Jasmine	32
3	0x30c8	43	Taco Pep	Noodle	Agra	ITC Mughal, Fatehabad Road, Tajganj, Agra	800	10121	Onque	Jasmine	32
4	0x95fb	45	Harvest Moon	Chilli Garlic Bread	Agra	ITC Mughal, Fatehabad Road, Tajganj, Agra	2200	10281	Owad	Clinton	25

Employee name has been split in columns : First and Last Name

City	Address	Avg_Cost	DeliveryID	Agent_FirstName	Agent_LastName	Agent_Age	Rating	Climate	Traffic	Time_1
Agra	1/48, Delhi Gate, Station Road, Raja Mandi, Ci...	680	10078	Oliver	Brooke	33	4.3	Windy	Jam	ITC Mughal
Agra	1/48, Delhi Gate, Station Road, Raja Mandi, Ci...	680	10078	Oliver	Brooke	33	4.3	Windy	Jam	ITC Mughal
Agra	ITC Mughal, Fatehabad Road, Tajganj, Agra	800	10121	Onque	Jasmine	32	4.6	Cloudy	Low	ITC Mughal
Agra	ITC Mughal, Fatehabad Road, Tajganj, Agra	800	10121	Onque	Jasmine	32	4.6	Cloudy	Low	ITC Mughal
Agra	ITC Mughal, Fatehabad Road, Tajganj, Agra	2200	10281	Owad	Clinton	25	4.2	Stormy	Jam	ITC Mughal

2NF

Continuing with data that was in the 1NF form we have done the changes. We removed the partial dependencies from the table

In [39]:

Out[39]:

	Order_ID	Restaurent_Id	Restaurent_Name
0	0xb816	42	Chez Michou
1	0xb816	42	Chez Michou
2	0x30c8	43	Taco Pep
3	0x30c8	43	Taco Pep
4	0x95fb	45	Harvest Moon
5	0x95fb	45	Harvest Moon
6	0x4863	46	Last Resort Grill
7	0x4863	46	Last Resort Grill
8	0xb46c	47	Sconyers Bar B Que
9	0xb46c	47	Sconyers Bar B Que
10	0xd7ec	48	Los Beto's
11	0xb4ba	49	The Hamburg Inn No. 2 Inc.

In [45]:

Df3

Out[45]:

	Order_ID	Restaurent_Id	Restaurent_Name	City	Address	DeliveryID	Agent_FirstName	Agent_LastName	Rating	Avg_Cost
0	0xb816	42	Chez Michou	Agra	1/48, Delhi Gate, Station Road, Raja Mandi, Ci...	10078	Oliver	Brooke	4.3	680
1	0x30c8	43	Taco Pep	Agra	ITC Mughal, Fatehabad Road, Tajganj, Agra	10121	Onque	Jasmine	4.6	800
2	0x95fb	45	Harvest Moon	Agra	ITC Mughal, Fatehabad Road, Tajganj, Agra	10281	Owad	Clinton	4.2	2200
3	0x4863	46	Last Resort Grill	Agra	Courtyard by Marriott Agra, Phase 2, Fatehabad...	10041	Ozark	Travis	4.6	1200
4	0xb46c	47	Sconyers Bar B Que	Agra	1/48, Delhi Gate, Station Road, Raja Mandi, Ci...	10148	Panjwani	Nina	4.7	5500
5	0xd7ec	48	Los Beto's	Agra	ITC Mughal, Fatehabad Road, Tajganj, Agra	10005	Patronick	Lucas	4.3	1200
6	0xb4ba	49	The Hamburg Inn No. 2 Inc.	Agra	Courtyard by Marriott Agra, Phase 2, Fatehabad...	10259	Pearson	Randall	4.9	1600
7	0xb7085	50	Fuji Japanese Steak	Agra	1/48, Delhi Gate, Station	10286	Smith	Martin	4.7	800

3NF

We removed the transitive dependencies from the table

In[49]:

	Order_ID	Restaurent_Id	DeliveryID	Time_Taken
0	0xb816	42	10078	39
1	0x30c8	43	10121	30
2	0x95fb	45	10281	39
3	0x4863	46	10041	15
4	0xb46c	47	10148	23
5	0xd7ec	48	10005	28
6	0xb4ba	49	10259	17
7	0x7085	50	10286	16
8	0x52e2	51	10297	36
9	0xdd96	52	10171	25
10	0x3329	53	10032	30
11	0xaef2	54	10130	31

In[52]:

	Restaurent_Id	Restaurent_Name	City	Address	Avg_Cost
0	42	Chez Michou	Agra	1/48, Delhi Gate, Station Road, Raja Mandi, Ci...	680
1	43	Taco Pep	Agra	ITC Mughal, Fatehabad Road, Tajganj, Agra	800
2	45	Harvest Moon	Agra	ITC Mughal, Fatehabad Road, Tajganj, Agra	2200
3	46	Last Resort Grill	Agra	Courtyard by Marriott Agra, Phase 2, Fatehabad...	1200
4	47	Sconyers Bar B Que	Agra	1/48, Delhi Gate, Station Road, Raja Mandi, Ci...	5500
5	48	Los Beto's	Agra	ITC Mughal, Fatehabad Road, Tajganj, Agra	1200
6	49	The Hamburg Inn No. 2 Inc.	Agra	Courtyard by Marriott Agra, Phase 2, Fatehabad...	1600
7	50	Fuji Japanese Steak House	Agra	1/48, Delhi Gate, Station Road, Raja Mandi, Ci...	800
8	51	Bailey's Bar-B-Que	Agra	ITC Mughal, Fatehabad Road, Tajganj, Agra	1600
9	52	Frick's Tap	Agra	23/453, Opposite Sanjay Cinema, Wazipura Road,...	750
10	53	Catfish Charlie's	Agra	Courtyard by Marriott Agra, Phase 2, Fatehabad...	1800
11	54	Atlanta Highway Seafood Market	Agra	23/453, Opposite Sanjay Cinema, Wazipura Road	1600

In [62]:

	DeliveryID	Agent_FirstName	Agent_LastName	Agent_Age
0	10078	Oliver	Brooke	33
1	10121	Onque	Jasmine	32
2	10281	Owad	Clinton	25
3	10041	Ozark	Travis	25
4	10148	Panjwani	Nina	34
5	10005	Patronick	Lucas	29
6	10259	Pearson	Randall	25
7	10286	Smith	Martin	21
8	10297	Pelletier	Emine	36
9	10171	Perry	Shakira	32
10	10032	Peters	Lauren	36
11	10130	Peterson	Ephones	32

In [67]:

1 Df12

Out[67]:

	Agent_Age	Climate	Traffic	Time_Taken
0	33	Windy	Jam	39
1	32	Cloudy	Low	30
2	25	Stormy	Jam	39
3	25	Stormy	Low	15
4	34	Sunny	High	23
5	29	Sandstorms	Low	28
6	25	Sandstorms	Low	17
7	21	Sunny	Low	16
8	36	Sandstorms	Medium	36
9	32	Sunny	Jam	25
10	36	Fog	Medium	30
11	32	Stormy	Medium	31

Create and insert Statement:

```
CREATE TABLE "Agents_detail" ( "OrderID" TEXT PRIMARY KEY , "Rest_id" INTEGER, "EmpID" INTEGER, "Ratings" REAL, "OrderDate" TEXT, "TimeOrderd" TEXT, "TimeOrderpicked" TEXT )
```

```
CREATE TABLE "Delivery_Details" ( "OrderID" TEXT PRIMARY KEY, "Restaurant_latitude" REAL, "Restaurant_longitude" REAL, "Delivery_location_latitude" REAL, "Delivery_location_longitude" REAL, "Address" TEXT )
```

```
CREATE TABLE "Delivery_condition" ( "OrderID" TEXT PRIMARY KEY, "EmpID" INTEGER, "Weather" TEXT, "Roadtrafficdensity" TEXT, "Typeofvehicle" TEXT, "Timetaken" INTEGER )
```

```
CREATE TABLE "Order_Detail" ( "OrderID" TEXT PRIMARY KEY, "Rest_id" INTEGER, "EmpID" INTEGER, "Ratings" REAL, "OrderDate" TEXT, "TimeOrderd" TEXT, "TimeOrderpicked" TEXT )
```

```
CREATE TABLE "Restaurent_Detail" ( "Rest_id" INTEGER PRIMARY KEY, "RestaurantName" TEXT, "Address" TEXT, "Restaurant_latitude" REAL, "Restaurant_longitude" REAL, "Specialities" TEXT, "City" TEXT, "CityType" TEXT, "Avgcost" INTEGER )
```

```
CREATE TABLE "tweet_gopro" ( "tweetID" REAL PRIMARY KEY, "Date_Created" TEXT, "Number_of_Likes" INTEGER, "Source_of_Tweet" TEXT, "Tweet" TEXT, "language" TEXT, "username" TEXT, "inreplyto_status_id" TEXT, "retweet_count" INTEGER, "entities" TEXT, "location" TEXT, "user_created" TEXT, "ht_Createdatlast24hrs" TEXT, "ht_text" TEXT, "ht_userid" TEXT )
```

```
INSERT INTO Agents_detail(OrderID,Rest_id,EmpID,Ratings,OrderDate,TimeOrderd,TimeOrderpicked)
VALUES ('0x21jna','213456','19769','4.9', '19-04-2022', '11:30:00','11:40:00')
INSERT INTO Restaurent_Table(Order_ID,Restaurent,Employee_Name,EmplID,Avgcost,Rating,City,Speciality, Timetaken) VALUES ('3f1963e8-38c1-41dd-be5f-40e0ca7646fd','Cormint','Adinolfi, Wilson K',10026,280,4.6,'Hyderabad','North Indian, Indian',25);
INSERT INTO Restaurent_Table(Order_ID,Restaurent,Employee_Name,EmplID,Avgcost,Rating,City,Speciality, Timetaken) VALUES ('23d0ecc4-a466-41db-899b-6991ac8037cd','Kwality Walls Frozen Dessert and Ice Cream Shop','Ait Sidi, Karthikeyan',10084,300,4.5,'Hyderabad','Desserts, Ice Cream, Ice Cream Cakes',21);
INSERT INTO Restaurent_Table(Order_ID,Restaurent,Employee_Name,EmplID,Avgcost,Rating,City,Speciality,
```

Timetaken) VALUES ('ab3ceff1-6265-4265-a1f1-859a28672ec0','Kumbakonam degree coffee','Akinkuolie, Sarah',10196,200,4.5,'Hyderabad','Indian, Beverages',42);
INSERT INTO
Restaurent_Table(Order_ID,Restaurent,Employee_Name,EmplID,Avgcost,Rating,City,Speciality, Timetaken) VALUES ('fcc68108-732a-4157-9586-7ac9889aeddcc','Fruit Bite Natural Juices & Salads','Alagbe,Trina',10088,200,4.5,'Hyderabad','Indian',39);

INSERT INTO
Restaurent_Table(Order_ID,Restaurent,Employee_Name,EmplID,Avgcost,Rating,City,Speciality, Timetaken) VALUES ('784213ea-6a27-4a0a-901c-0f546963241c','Varalakshmi Tiffins','Anderson, Carol',10069,300,4.5,'Hyderabad','South Indian',27);

INSERT INTO
Restaurent_Table(Order_ID,Restaurent,Employee_Name,EmplID,Avgcost,Rating,City,Speciality, Timetaken) VALUES ('04f5f732-b1b6-4b2b-92a2-91778bcf7ac9','Go foodies','Anderson, Linda',10002,100,4.4,'Hyderabad','Snacks, Ice Cream',34);

INSERT INTO
Restaurent_Table(Order_ID,Restaurent,Employee_Name,EmplID,Avgcost,Rating,City,Speciality, Timetaken) VALUES ('1a9153d3-ea0e-4f95-a90f-8fc2a96a7730','MADHURAM TIFFINS','Andreola, Colby',10194,150,4.4,'Hyderabad','South Indian',31);

INSERT INTO
Restaurent_Table(Order_ID,Restaurent,Employee_Name,EmplID,Avgcost,Rating,City,Speciality, Timetaken) VALUES ('04aeec47-833a-4b7d-94ff-a2591f078283','New Srinivasa Bhavan','Athwal, Sam',10062,100,4.4,'Hyderabad','South Indian',40);

INSERT INTO
Restaurent_Table(Order_ID,Restaurent,Employee_Name,EmplID,Avgcost,Rating,City,Speciality, Timetaken) VALUES ('34fefc4d-db34-4949-a9c5-2f1ad75af612','FitChai Raghavendra Colony','Bachiochi, Linda',10114,125,4.3,'Hyderabad','Beverages, Snacks',24);

INSERT INTO
Restaurent_Table(Order_ID,Restaurent,Employee_Name,EmplID,Avgcost,Rating,City,Speciality, Timetaken) VALUES ('8d1dc024-4e09-4563-a714-1b4783ef0b5a','Chai Samosa','Barone, Francesco A',10265,120,4.3,'Hyderabad','Beverages, Snacks, Fast Food',42);

INSERT INTO
Restaurent_Table(Order_ID,Restaurent,Employee_Name,EmplID,Avgcost,Rating,City,Speciality, Timetaken) VALUES ('32cbd11b-7c18-4a04-a5fc-87b3c0795dcc','Crumbcoat','Barton, Nader',10066,150,4.3,'Hyderabad','Desserts, Waffle, Beverages, Snacks, Ice Cream',35);

USE CASES :

1. SELECT DISTINCT Ratings FROM Order_Detail;
2. SELECT EmpID, Ratings FROM Order_Detail where Ratings>4.9;
3. create view 'Av_Rating' AS
select avg(Ratings) from Order_Detail
select * from Av_Rating
4. select * from Restaurent_Detail where City="Mumbai"
5. select Rest_id, RestaurantName from Restaurent_Detail where Avcost>1000
6. select distinct(City) from Restaurent_Detail
7. select * from Delivery_Details where Restaurant_latitude between 18.0 and 19.0
8. select * from Delivery_Details where Delivery_location_longitude>80.0
9. select distinct(Roadtrafficdensity) from Delivery_condition
10. select distinct(Typeofvehicle) from Delivery_condition
11. select Roadtrafficdensity, Typeofvehicle from Delivery_condition where
Roadtrafficdensity="Jam" and Typeofvehicle="scooter"
12. select distinct(Weather) from Delivery_condition
13. select t.Timetaken , d.Delivery_location_latitude from Delivery_Details d
join Delivery_condition t
on t.OrderID=d.OrderID
14. select count(EmpID) from Agents_detail
15. select min(Ratings), avg(Ratings) from Agents_detail

16. select d.OrderID ,min(t.Timetaken) as 'time', RestaurantName from Restaurent_Detail t
join Delivery_Details d
on t.OrderID=d.OrderID
17. create view 'ID' AS
select OrderID, Rest_id,EmpID
from Agents_detail
select * from ID
18. select distinct(language) from tweet_gopro
19. select max(retweet_count) from tweet_gopro
20. select * from tweet_gopro where username="Grubhub"
21. select * from tweet_gopro where inreplyto_status_id IS NULL
22. select d.Address ,t.City, t.CityType from Restaurent_Detail t
join Delivery_Details d
on t.Restaurant_longitude=d.Restaurant_longitude
23. select a.OrderID, t.RestaurantName, avg(t.Avcost) as 'avg' from Restaurent_Detail t
join Agents_detail a
on t.Rest_id=a.Rest_id
24. select * from Delivery_condition
WHERE
Typeofvehicle like '%scooter%'
25. select d.Timetaken,a.OrderDate,a.TimeOrderd,a.TimeOrderpicked
from Agents_detail a
inner join Delivery_condition d
on a.OrderID = d.OrderID
26. select * from Restaurent_Detail
WHERE Specialities like '%Pav bhaji%'
27. SELECT DISTINCT (location) from tweet_gopro

28. SELECT * FROM Restaurent_Detail WHERE Avgcost > (SELECT avg(Avgcost) FROM Restaurent_Detail)
29. SELECT * FROM Restaurent_Detail
WHERE CityType IN ('Urban')
30. SELECT DISTINCT(City), CityType
from Restaurent_Detail
WHERE CityType = "Urban"
31. select tweetID, max(Number_of_Likes) as maxLikes, Tweet, username, location from tweet_gopro where username like "zomato" group by username
32. select count(Date_Created) ,username from tweet_gopro
where Date_Created BETWEEN "2022-11-09" AND "2022-11-11"
group by username
33. CREATE view 'Special' as SELECT RestaurantName, Specialities from
Restaurent_Detail
SELECT * from Special
34. CREATE VIEW 'Twitter' AS
SELECT tweetID, Number_of_Likes, Tweet, retweet_count from tweet_gopro
SELECT * FROM Twitter
35. SELECT * from tweet_gopro
WHERE language = "hi" AND location = "India"
36. SELECT * FROM tweet_gopro
WHERE user_created BETWEEN '2008-08-18' AND '2009-03-08'
37. SELECT ht_text from tweet_gopro
WHERE ht_text like '%NOT%'
38. SELECT * FROM Delivery_condition
WHERE Typeofvehicle = "motorcycle" group by Weather
39. select min(Ratings), avg(Ratings) from Agents_detail

40. select * from Delivery_Details where Restaurant_latitude between 20.0 and 22.0

VIEW STATEMENTS:

```
1 create view 'Av_Rating' AS  
2 select avg(Ratings) from Order_Detail  
3  
4 select * from Av_Rating  
5  
5
```

	avg(Ratings)
1	4.44524524524525

```

1  create view 'ID' AS
2    select OrderID, Rest_id, EmpID
3    from Agents_detail
4
5    select * from ID
6
7
8

```

	OrderID	Rest_id	EmpID
1	0x4607	134629	10311
2	0xb379	134630	10310
3	0x5d6d	134631	10309
4	0x7a6a	134632	10308
5	0x70a2	134633	10307
6	0x9bb4	134634	10306
7	0x95b4	134635	10305
8	0x9eb2	134636	10304
9	0x1102	134637	10303

Execution finished without errors.
 Result: 1000 rows returned in 45ms
 At line 5:
 select * from ID

```

1  CREATE VIEW 'Twitter' AS
2    SELECT tweetID, Number_of_Likes, Tweet, retweet_count from tweet_gopro
3    SELECT * FROM Twitter

```

	tweetID	Number_of_Likes	Tweet	retweet_count
1	1.59496546859473e+18	860	@elonmusk if you tweet that counts ...	22
2	1.59491997048783e+18	2353	namaste 🙏 https://t.co/dQZozyXaJJ	113
3	1.59456315211686e+18	960	'gol gappe' is such a basic name for ...	41
4	1.59347949216479e+18	25810	guys don't worry, if twitter shuts dow...	1365
5	1.59311591411512e+18	1606	all I need in life is more snacks than ...	124
6	1.59247755008503e+18	2252	the world now has 8 billion people an...	112
7	1.59201987626861e+18	4831	happy children's day to people jinki ...	263
8	1.59175708866946e+18	21083	no combo would have come close to ...	1289
9	1.59066111988639e+18	23278	oreo pakoda would taste a lot better ...	1451

Execution finished without errors.
 Result: 100 rows returned in 11ms
 At line 3:
 SELECT * FROM Twitter

```
1 CREATE view 'Special' as
2 SELECT RestaurantName, Specialities
3 from Restaurent_Detail
4
5 SELECT * from Special
6
7
8
```

	RestaurantName	Specialities
1	Le Petit Souffle	Mutton Galouti Kabab, Kali Mirch ...
2	Izakaya Kikufuji	Onion Naan, Kurkure Momo, Kadai ...
3	Heat - Edsa Shangri-La	Chocolate Truffle Cake, Cakes, Tea, ...
4	Ooma	Tawa Chicken, Chicken Curry, Paneer...
5	Sambo Kojin	Coconut Milk Dessert, Honey Caviar, ...
6	Din Tai Fung	Irish Cream Tiramisu, Penne Vodka, ...
7	Buffet 101	Malleshwari Dosa, Benne Khali Dosa,...
8	Vikings	Classic Martini, Cocktails, Bloody Mary...
9	Sniral - Sofitel Philippine Plaza Manila	Watermelon Ice Cream.. Choco Bite....

Execution finished without errors.
Result: 999 rows returned in 46ms
At line 2:
SELECT * from Special