In [1]:
```
#librarys
import pandas as pd
```

In [2]:
```
#File extraction
df =pd.read_csv("C:/Users/kunal/Downloads/uber_data.csv")
```

In [3]:
```
#Information
df.head()
```

Out[3]:

	VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	pickup_longitude	pickup_latitude	RatecodeID	store_and_fwd_flag	dropoff_longitude	dropoff_latitude	payment_type	fare_amount	extra
0	1	2016-03-01 00:00:00	2016-03-01 00:07:55	1	2.50	-73.976746	40.765152	1	N	-74.004265	40.746128	1	9.0	0.5
1	1	2016-03-01 00:00:00	2016-03-01 00:11:06	1	2.90	-73.983482	40.767925	1	N	-74.005943	40.733166	1	11.0	0.5
2	2	2016-03-01 00:00:00	2016-03-01 00:31:06	2	19.98	-73.782021	40.644810	1	N	-73.974541	40.675770	1	54.5	0.5
3	2	2016-03-01 00:00:00	2016-03-01 00:00:00	3	10.78	-73.863419	40.769814	1	N	-73.969650	40.757767	1	31.5	0.0
4	2	2016-03-01 00:00:00	2016-03-01 00:00:00	5	30.43	-73.971741	40.792183	3	N	-74.177170	40.695053	1	98.0	0.0

In [4]:
```
#check whether column has right Datatype
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 19 columns):
 #   Column                 Non-Null Count   Dtype  
---  ------                 --------------   -----  
 0   VendorID               100000 non-null  int64  
 1   tpep_pickup_datetime   100000 non-null  object 
 2   tpep_dropoff_datetime  100000 non-null  object 
 3   passenger_count        100000 non-null  int64  
 4   trip_distance          100000 non-null  float64
 5   pickup_longitude       100000 non-null  float64
 6   pickup_latitude        100000 non-null  float64
 7   RatecodeID             100000 non-null  int64  
 8   store_and_fwd_flag     100000 non-null  object 
 9   dropoff_longitude      100000 non-null  float64
 10  dropoff_latitude       100000 non-null  float64
 11  payment_type           100000 non-null  int64  
 12  fare_amount            100000 non-null  float64
 13  extra                  100000 non-null  float64
 14  mta_tax                100000 non-null  float64
 15  tip_amount             100000 non-null  float64
 16  tolls_amount           100000 non-null  float64
 17  improvement_surcharge  100000 non-null  float64
 18  total_amount           100000 non-null  float64
dtypes: float64(12), int64(4), object(3)
memory usage: 14.5+ MB
```

In [5]:
```
df['tpep_pickup_datetime']= pd.to_datetime(df['tpep_pickup_datetime'])
df['tpep_dropoff_datetime']= pd.to_datetime(df['tpep_dropoff_datetime'])
```

In [6]:
```
df = df.drop_duplicates().reset_index(drop=True)
df['trip_id'] = df.index
```

In [7]:
```
#converting pickup and dropoff into Datetime formate
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 20 columns):
 #   Column                 Non-Null Count   Dtype         
---  ------                 --------------   -----         
 0   VendorID               100000 non-null  int64         
 1   tpep_pickup_datetime   100000 non-null  datetime64[ns]
 2   tpep_dropoff_datetime  100000 non-null  datetime64[ns]
 3   passenger_count        100000 non-null  int64         
 4   trip_distance          100000 non-null  float64       
 5   pickup_longitude       100000 non-null  float64       
 6   pickup_latitude        100000 non-null  float64       
 7   RatecodeID             100000 non-null  int64         
 8   store_and_fwd_flag     100000 non-null  object        
 9   dropoff_longitude      100000 non-null  float64       
 10  dropoff_latitude       100000 non-null  float64       
 11  payment_type           100000 non-null  int64         
 12  fare_amount            100000 non-null  float64       
 13  extra                  100000 non-null  float64       
 14  mta_tax                100000 non-null  float64       
 15  tip_amount             100000 non-null  float64       
 16  tolls_amount           100000 non-null  float64       
 17  improvement_surcharge  100000 non-null  float64       
 18  total_amount           100000 non-null  float64       
 19  trip_id                100000 non-null  int64         
dtypes: datetime64[ns](2), float64(12), int64(5), object(1)
memory usage: 15.3+ MB
```

In [8]:
```
# modifing the data into column of hour,day,month,year and weekday
datetime_dim = df[['tpep_pickup_datetime','tpep_dropoff_datetime']].drop_duplicates().reset_index(drop=True)

datetime_dim['pick_hour'] = datetime_dim['tpep_pickup_datetime'].dt.hour

datetime_dim['pick_day'] = datetime_dim['tpep_pickup_datetime'].dt.day

datetime_dim['pick_month'] = datetime_dim['tpep_pickup_datetime'].dt.month

datetime_dim['pick_year'] = datetime_dim['tpep_pickup_datetime'].dt.year

datetime_dim['pick_weekday'] = datetime_dim['tpep_pickup_datetime'].dt.weekday

##
datetime_dim['dropoff_hour'] = datetime_dim['tpep_dropoff_datetime'].dt.hour

datetime_dim['dropoff_day'] = datetime_dim['tpep_dropoff_datetime'].dt.day

datetime_dim['dropoff_month'] = datetime_dim['tpep_dropoff_datetime'].dt.month

datetime_dim['dropoff_year'] = datetime_dim['tpep_dropoff_datetime'].dt.year

datetime_dim['dropoff_weekday'] = datetime_dim['tpep_dropoff_datetime'].dt.weekday
```

In [9]:
```
datetime_dim
```

Out[9]:

	tpep_pickup_datetime	tpep_dropoff_datetime	pick_hour	pick_day	pick_month	pick_year	pick_weekday	dropoff_hour	dropoff_day	dropoff_month	dropoff_year	dropoff_weekday
0	2016-03-01 00:00:00	2016-03-01 00:07:55	0	1	3	2016	1	0	1	3	2016	1
1	2016-03-01 00:00:00	2016-03-01 00:11:06	0	1	3	2016	1	0	1	3	2016	1
2	2016-03-01 00:00:00	2016-03-01 00:31:06	0	1	3	2016	1	0	1	3	2016	1
3	2016-03-01 00:00:00	2016-03-01 00:00:00	0	1	3	2016	1	0	1	3	2016	1
4	2016-03-01 00:00:01	2016-03-01 00:16:04	0	1	3	2016	1	0	1	3	2016	1
...	...	...	...	...	...	...	...	...	...	...	...	...
99848	2016-03-01 06:17:10	2016-03-01 06:22:15	6	1	3	2016	1	6	1	3	2016	1
99849	2016-03-01 06:17:10	2016-03-01 06:32:41	6	1	3	2016	1	6	1	3	2016	1
99850	2016-03-01 06:17:10	2016-03-01 06:37:23	6	1	3	2016	1	6	1	3	2016	1
99851	2016-03-01 06:17:10	2016-03-01 06:22:09	6	1	3	2016	1	6	1	3	2016	1
99852	2016-03-01 06:17:11	2016-03-01 06:22:00	6	1	3	2016	1	6	1	3	2016	1

99853 rows × 12 columns

In [10]:
```
#giving Index to table
datetime_dim['datetime_id'] = datetime_dim.index
```

In [11]:
```
datetime_dim
```

Out[11]:

	tpep_pickup_datetime	tpep_dropoff_datetime	pick_hour	pick_day	pick_month	pick_year	pick_weekday	dropoff_hour	dropoff_day	dropoff_month	dropoff_year	dropoff_weekday	datetime_id
0	2016-03-01 00:00:00	2016-03-01 00:07:55	0	1	3	2016	1	0	1	3	2016	1	0
1	2016-03-01 00:00:00	2016-03-01 00:11:06	0	1	3	2016	1	0	1	3	2016	1	1
2	2016-03-01 00:00:00	2016-03-01 00:31:06	0	1	3	2016	1	0	1	3	2016	1	2
3	2016-03-01 00:00:00	2016-03-01 00:00:00	0	1	3	2016	1	0	1	3	2016	1	3
4	2016-03-01 00:00:01	2016-03-01 00:16:04	0	1	3	2016	1	0	1	3	2016	1	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...
99848	2016-03-01 06:17:10	2016-03-01 06:22:15	6	1	3	2016	1	6	1	3	2016	1	99848
99849	2016-03-01 06:17:10	2016-03-01 06:32:41	6	1	3	2016	1	6	1	3	2016	1	99849
99850	2016-03-01 06:17:10	2016-03-01 06:37:23	6	1	3	2016	1	6	1	3	2016	1	99850
99851	2016-03-01 06:17:10	2016-03-01 06:22:09	6	1	3	2016	1	6	1	3	2016	1	99851
99852	2016-03-01 06:17:11	2016-03-01 06:22:00	6	1	3	2016	1	6	1	3	2016	1	99852

99853 rows × 13 columns

In [12]:
```
#arrange the table
datetime_dim = datetime_dim[['datetime_id','tpep_pickup_datetime','pick_hour','pick_day','pick_month',
             'pick_year','pick_weekday','tpep_dropoff_datetime','dropoff_hour',
             'dropoff_day','dropoff_month','dropoff_year','dropoff_weekday']]
```

In [13]:
```
datetime_dim
```

Out[13]:

	datetime_id	tpep_pickup_datetime	pick_hour	pick_day	pick_month	pick_year	pick_weekday	tpep_dropoff_datetime	dropoff_hour	dropoff_day	dropoff_month	dropoff_year	dropoff_weekday
0	0	2016-03-01 00:00:00	0	1	3	2016	1	2016-03-01 00:07:55	0	1	3	2016	1
1	1	2016-03-01 00:00:00	0	1	3	2016	1	2016-03-01 00:11:06	0	1	3	2016	1
2	2	2016-03-01 00:00:00	0	1	3	2016	1	2016-03-01 00:31:06	0	1	3	2016	1
3	3	2016-03-01 00:00:00	0	1	3	2016	1	2016-03-01 00:00:00	0	1	3	2016	1
4	4	2016-03-01 00:00:01	0	1	3	2016	1	2016-03-01 00:16:04	0	1	3	2016	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
99848	99848	2016-03-01 06:17:10	6	1	3	2016	1	2016-03-01 06:22:15	6	1	3	2016	1
99849	99849	2016-03-01 06:17:10	6	1	3	2016	1	2016-03-01 06:32:41	6	1	3	2016	1
99850	99850	2016-03-01 06:17:10	6	1	3	2016	1	2016-03-01 06:37:23	6	1	3	2016	1
99851	99851	2016-03-01 06:17:10	6	1	3	2016	1	2016-03-01 06:22:09	6	1	3	2016	1
99852	99852	2016-03-01 06:17:11	6	1	3	2016	1	2016-03-01 06:22:00	6	1	3	2016	1

99853 rows × 13 columns

In [14]:
```
passenger_count_dim = df[['passenger_count']].reset_index(drop=True)
passenger_count_dim['passenger_count_id'] = passenger_count_dim.index
passenger_count_dim = passenger_count_dim[['passenger_count_id','passenger_count']]

trip_distance_dim = df[['trip_distance']].reset_index(drop=True)
trip_distance_dim['trip_distance_id'] = trip_distance_dim.index
trip_distance_dim = trip_distance_dim[['trip_distance_id','trip_distance']]
```

In [15]:
```
rate_code_type = {
    1:"Standard rate",
    2:"JFK",
    3:"Newark",
    4:"Nassau or Westchester",
    5:"Negotiated fare",
    6:"Group ride"
}

rate_code_dim = df[['RatecodeID']].reset_index(drop=True)
rate_code_dim['rate_code_id'] = rate_code_dim.index
rate_code_dim['rate_code_name'] = rate_code_dim['RatecodeID'].map(rate_code_type)
rate_code_dim = rate_code_dim[['rate_code_id','RatecodeID','rate_code_name']]
```

In [16]:
```
rate_code_dim.head()
```

Out[16]:

	rate_code_id	RatecodeID	rate_code_name
0	0	1	Standard rate
1	1	1	Standard rate
2	2	1	Standard rate
3	3	1	Standard rate
4	4	3	Newark

In [17]:
```
pickup_location_dim = df[['pickup_longitude', 'pickup_latitude']].drop_duplicates().reset_index(drop=True)
pickup_location_dim['pickup_location_id'] = pickup_location_dim.index
pickup_location_dim = pickup_location_dim[['pickup_location_id','pickup_latitude','pickup_longitude']]


dropoff_location_dim = df[['dropoff_longitude', 'dropoff_latitude']].drop_duplicates().reset_index(drop=True)
dropoff_location_dim['dropoff_location_id'] = dropoff_location_dim.index
dropoff_location_dim = dropoff_location_dim[['dropoff_location_id','dropoff_latitude','dropoff_longitude']]
```

In [18]:
```
payment_type_name = {
    1:"Credit card",
    2:"Cash",
    3:"No charge",
    4:"Dispute",
    5:"Unknown",
    6:"Voided trip"
}
payment_type_dim = df[['payment_type']].drop_duplicates().reset_index(drop=True)
payment_type_dim['payment_type_id'] = payment_type_dim.index
payment_type_dim['payment_type_name'] = payment_type_dim['payment_type'].map(payment_type_name)
payment_type_dim = payment_type_dim[['payment_type_id','payment_type','payment_type_name']]
```

In [19]:
```
payment_type_dim.head()
```

Out[19]:

	payment_type_id	payment_type	payment_type_name
0	0	1	Credit card
1	1	2	Cash
2	2	3	No charge
3	3	4	Dispute

In [20]:
```
fact_table = df.merge(passenger_count_dim, left_on='trip_id', right_on='passenger_count_id') \
              .merge(trip_distance_dim, left_on='trip_id', right_on='trip_distance_id') \
              .merge(rate_code_dim, left_on='trip_id', right_on='rate_code_id') \
              .merge(pickup_location_dim, left_on='trip_id', right_on='pickup_location_id') \
              .merge(dropoff_location_dim, left_on='trip_id', right_on='dropoff_location_id')\
              .merge(datetime_dim, left_on='trip_id', right_on='datetime_id') \
              .merge(payment_type_dim, left_on='trip_id', right_on='payment_type_id') \
              [['trip_id','VendorID', 'datetime_id', 'passenger_count_id',
                'trip_distance_id', 'rate_code_id', 'store_and_fwd_flag', 'pickup_location_id', 'dropoff_location_id',
                'payment_type_id', 'fare_amount', 'extra', 'mta_tax', 'tip_amount', 'tolls_amount',
                'improvement_surcharge', 'total_amount']]
```

In [21]:
```
payment_type_dim.columns
```

Out[21]:
```
Index(['payment_type_id', 'payment_type', 'payment_type_name'], dtype='object')
```

In [22]:
```
fact_table.columns
```

Out[22]:
```
Index(['trip_id', 'VendorID', 'datetime_id', 'passenger_count_id',
       'trip_distance_id', 'rate_code_id', 'store_and_fwd_flag',
       'pickup_location_id', 'dropoff_location_id', 'payment_type_id',
       'fare_amount', 'extra', 'mta_tax', 'tip_amount', 'tolls_amount',
       'improvement_surcharge', 'total_amount'],
      dtype='object')
```

In [23]:
```
fact_table
```

Out[23]:

	trip_id	VendorID	datetime_id	passenger_count_id	trip_distance_id	rate_code_id	store_and_fwd_flag	pickup_location_id	dropoff_location_id	payment_type_id	fare_amount	extra	mta_tax	tip_amount	tolls_amount	improveme
0	0	1	0	0	0	0	N	0	0	0	9.0	0.5	0.5	2.05	0.00	
1	1	1	1	1	1	1	N	1	1	1	11.0	0.5	0.5	3.05	0.00	
2	2	2	2	2	2	2	N	2	2	2	54.5	0.5	0.5	8.00	0.00	
3	3	2	3	3	3	3	N	3	3	3	31.5	0.0	0.5	3.78	5.54	

In [ ]: