- ETL Project -
Restaurant Data with Consumer Ratings

By Hari, Chadwick and Sam

Extraction

The public platform Kaggle lead us to the "Restaurant Data with Consumer Ratings" website which had 5 different datasets having following information:

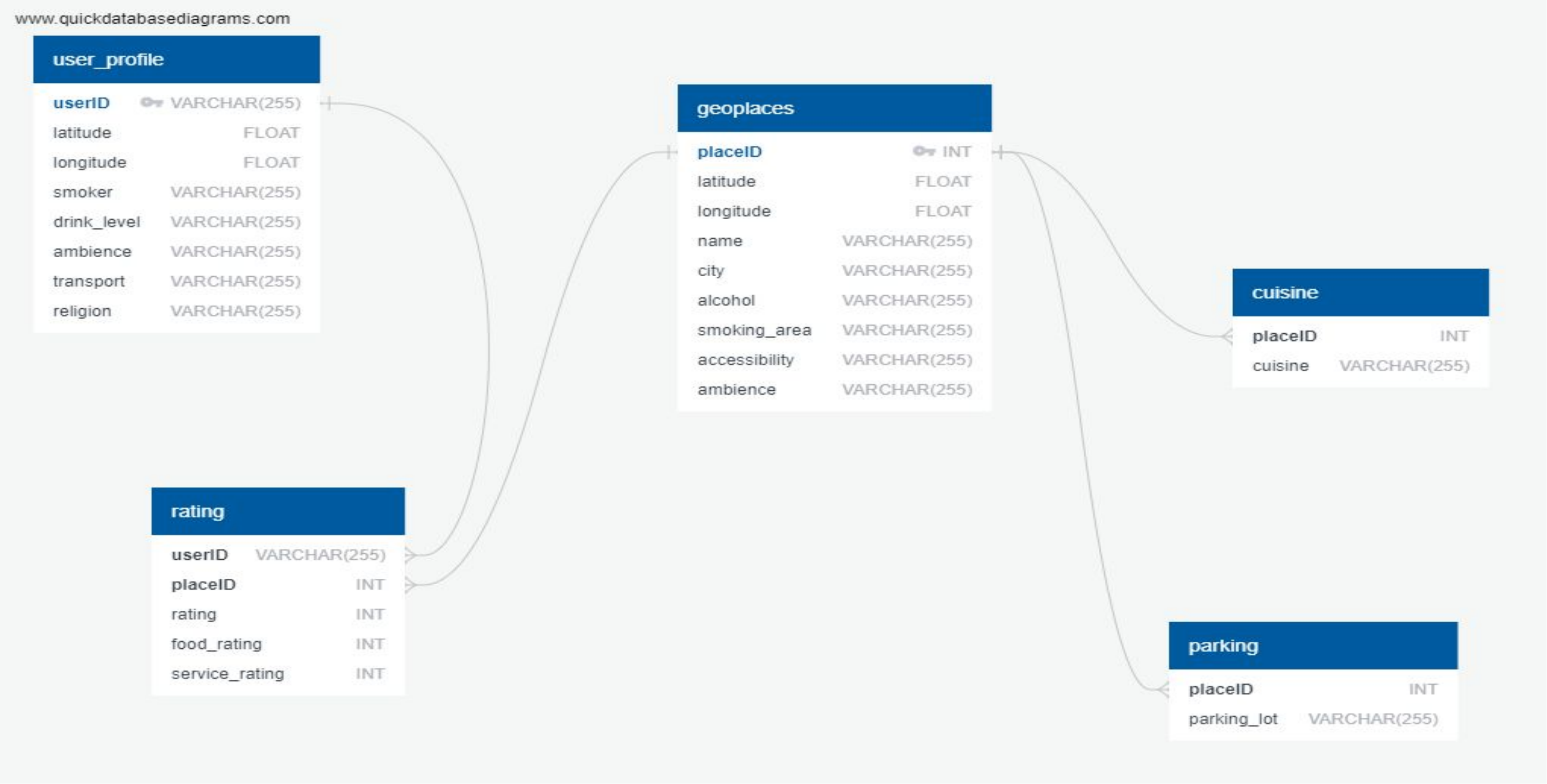
User profile and their preferences	userprofile.csv
Restaurant Related Information	geoplaces2.csv
Restaurant Ratings	rating_final.csv
Restaurant Cuisine Speciality	chefmozcuisine.csv
Restaurant Parking Availability	chefmozparking.csv

The following is the source our datasets used:

https://www.kaggle.com/uciml/restaurant-data-with-consumer-ratings/metadata

ER Diagram

The real datasets had more number of fields however we choose to pick fields of interest:



Observations

The following table illustrates the observations we would like to address:

Top 5 best restaurants for families
Top 5 worst restaurants based on service rating
Does Parking lot affect the restaurant rating
Which City has most elevated no of restaurants with higher rating
Does alcohol/no alcohol affect the restaurant rating
No of restaurants based on Cuisine

Transformation

In order to transform the restaurant data and use it in our study we performed the following:

- Reviewed the Datasets and transformed into data frames

```
# Use Pandas to read data
userprofile_df = pd.read_csv(userprofile, encoding="ISO-8859-1")
userprofile_df.head()
```

- Selected the specific columns from Datasets with respect to observations

```
# get list of all columns
userprofile_df.columns
```

```
3]: Index(['userID', 'latitude', 'longitude', 'smoker', 'drink_level',
           'dress_preference', 'ambience', 'transport', 'marital_status', 'hijos',
           'birth_year', 'interest', 'personality', 'religion', 'activity',
           'color', 'weight', 'budget', 'height'],
          dtype='object')
```

```
# Select specific columns
userprofile_df = userprofile_df[["userID", "latitude", "longitude", "smoker", "drink_level", "ambience",
                                 "transport", "religion"]]
userprofile_df.head()
```

Transformation

- Used Pandas functions to rename the columns to avoid data load failure with Postgres

```
# Rename UserID column to userid
userprofile_df = userprofile_df.rename(columns={"userID": "userid"})
```

- Used Pandas functions to catch Missing/Null values

```
# check all columns with any missing/null values
userprofile_df.isna().sum()
```

```
]: userid          0
   latitude        0
   longitude       0
   smoker          0
   drink_level     0
   ambience        0
   transport       0
   religion        0
   dtype: int64
```

- Some of the Datasets had "?" as opposed to Missing/Null values, we have replaced them with 'Not Recorded' to avoid Referential Integrity errors for Postgres.

```
# Replace the values having ? with Nan
userprofile_df = userprofile_df.replace('?', "Not Recorded")
userprofile_df.head()
```

Transformation

- Used Pandas functions to catch the duplicate values and drop them to get the clean Dataset

```
# check all duplicate rows
duplicate_rows_df = userprofile_df[userprofile_df.duplicated()]
print (f"Number of duplicate rows: {duplicate_rows_df.shape}")
```

```
Number of duplicate rows: (0, 8)
```

- City column from Geoplaces table had multiple variations for City names, we corrected those to get the clean dataset

```
# City coulm had lot of variations for "San Luis Potosi"
places_df["city"].unique()
```

```
.2]: array(['Cuernavaca', 's.l.p.', 'San Luis Potosi', 'victoria ', 'victoria',
       'Cd Victoria', 'Not Recorded', 'san luis potosi', 'Jiutepec',
       'cuernavaca', 'slp', 'Soledad', 'san luis potos',
       'san luis potosi ', 'Ciudad Victoria', 'Cd. Victoria', 's.l.p'],
      dtype=object)
```

```
# Replace the values 's.l.p.', 'san luis potosi', 'slp', 'san luis potos', 'san luis potosi ' and 's.l.p'
# with "San Luis Potosi" to get the correct data loaded into Postgres for further analysis
places_df = places_df.replace(['s.l.p.','san luis potosi','slp','san luis potos','san luis potosi ','s.l.p'], "San Luis Potos
places_df["city"].unique()
```

```
.3]: array(['Cuernavaca', 'San Luis Potosi', 'victoria ', 'victoria',
       'Cd Victoria', 'Not Recorded', 'Jiutepec', 'cuernavaca', 'Soledad',
       'Ciudad Victoria', 'Cd. Victoria'], dtype=object)
```

Transformation

- Used Sqlalchemy to connect to Postgres Database

```
# connect to Postgres
engine = create_engine(f"postgresql://postgres:{password}@localhost/restaurant_rating_db")
conn = engine.connect()
```

- Used Pandas functions to load all 5 CSV Datasets into Postgres Database.

```
# Insert data into User_Profile table
userprofile_df.to_sql(name='user_profile', con=engine, if_exists='append', index=False)
```

Load

After we pulled in the CSV files and loaded them into the data frames, we did an initial connection to the Postgres database using PGAdmin to store our transformed data sets.

We used the quick database website to create the initial table schema that got loaded into the Postgres database that generated the first set of tables by maintaining the Referential integrity.

USER_PROFILE

Query Editor Query History

```
SELECT * FROM USER_PROFILE;
```

Data Output Explain Messages Notifications

	userid [PK] character varying (255)	latitude double precision	longitude double precision	smoker character varying (255)	drink_level character varying (255)	ambience character varying (255)	transport character varying (255)	religion character varying (255)
1	U1001	22.139997	-100.978803	false	abstemious	family	on foot	none
2	U1002	22.150087	-100.983325	false	abstemious	family	public	Catholic
3	U1003	22.119847	-100.946527	false	social drinker	family	public	Catholic
4	U1004	18.867	-99.18299999999999	false	abstemious	family	public	none
5	U1005	22.183477	-100.959891	false	abstemious	family	public	Catholic

Load

GEOPLACES

Query Editor | Query History

```
SELECT * FROM GEOPLACES;
```

Data Output | Explain | Messages | Notifications

	placeid [PK] integer	latitude double precision	longitude double precision	name character varying (255)	city character varying (255)	alcohol character varying (255)	smoking_area character varying (255)	accessibility character varying (255)	ambience character varying (255)
1	134999	18.915421	-99.184871	Kiku Cuernavaca	Cuernavaca	No_Alcohol_Served	none	no_accessibility	familiar
2	132825	22.1473922	-100.983092	puesto de tacos	San Luis Potosi	No_Alcohol_Served	none	completely	familiar
3	135106	22.149708800000003	-100.97609279999999	El Rinc�n de San Francisco	San Luis Potosi	Wine-Beer	only at bar	partially	familiar
4	132667	23.7526973	-99.1633594	little pizza Emilio Portes Gil	Ciudad Victoria	No_Alcohol_Served	none	completely	familiar
5	132613	23.7529035	-99.165076	carnitas_mata	Ciudad Victoria	No_Alcohol_Served	permitted	completely	familiar

RATING

Query Editor | Query History

```
SELECT * FROM RATING;
```

Data Output | Explain | Messages | Notifications

	userid character varying (255)	placeid integer	rating integer	food_rating integer	service_rating integer
1	U1077	135085	2	2	2
2	U1077	135038	2	2	1
3	U1077	132825	2	2	2
4	U1077	135060	1	2	2
5	U1068	135104	1	1	2

Load

CUISINE

Query Editor | Query History

```sql
SELECT * FROM CUISINE;
```

Data Output | Explain | Messages | Notifica

	placeid integer	cuisine character varying (255)
1	134999	Dutch-Belgian
2	132825	Seafood
3	135106	International
4	132667	Seafood
5	132613	French

PARKING

Query Editor | Query History

```sql
SELECT * FROM PARKING;
```

Data Output | Explain | Messages | Notifica

	placeid integer	parking_lot character varying (255)
1	134999	public
2	132825	none
3	135106	none
4	132667	street
5	132613	street

Summary

The time constraint and limited set of information were a portion of the primary imperatives which influenced our investigation of this ETL project. But, we were still managed to come up with below observations which should provide sufficient thought about the dataset what we are dealing herewith -

Which City has most elevated no of restaurants with higher rating?

The city "**San Luis Potosi**" has one of the best rated restaurants in Mexico. The highest number of better rated restaurants makes it ideal destination for tourism.

```sql
-- Which City has highest no of restaurants with higher rating
SELECT COUNT(R.RATING),G.CITY
FROM USER_PROFILE UP,
RATING R,
GEOPLACES G
WHERE UP.USERID = R.USERID
AND R.PLACEID = G.PLACEID
AND G.CITY != 'Not Recorded'
GROUP BY G.CITY
HAVING MAX(R.RATING) = 2 --- 2 being the highest rating
ORDER BY COUNT(R.RATING) DESC;
```

Data Output | Explain | Messages | Notifications

	count bigint	city character varying (255)
1	834	San Luis Potosi
2	89	Cuernavaca
3	88	Ciudad Victoria
4	19	Jiutepec
5	17	Soledad

Summary

```sql
SELECT P.PARKING_LOT,R.RATING, COUNT(R.RATING) AS "TOTAL RATING"
FROM USER_PROFILE UP,
RATING R,
GEOPLACES G,
PARKING P
WHERE UP.USERID = R.USERID
AND R.PLACEID = G.PLACEID
AND G.PLACEID = P.PLACEID
GROUP BY P.PARKING_LOT,R.RATING
ORDER BY COUNT(R.RATING) DESC
LIMIT 5;
```

Data Output Explain Messages Notifications

	parking_lot character varying (255)	rating integer	TOTAL RATING bigint
1	yes	2	182
2	none	2	177
3	yes	1	163
4	none	1	144
5	yes	0	100

Does Parking lot affect the restaurant rating?

When it comes to overall rating of restaurants, users doesn’t seems to care much about availability of parking. The second record shows even though parking is unavailable still users has given higher rating.