- 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:

www.quickdatabasediagrams.com

user_profile

userID	VARCHAR(255)
latitude	FLOAT
longitude	FLOAT
smoker	VARCHAR(255)
drink_level	VARCHAR(255)
ambience	VARCHAR(255)
transport	VARCHAR(255)
religion	VARCHAR(255)

geoplaces

placeID	INT
latitude	FLOAT
longitude	FLOAT
name	VARCHAR(255)
city	VARCHAR(255)
alcohol	VARCHAR(255)
smoking_area	VARCHAR(255)
accessibility	VARCHAR(255)
ambience	VARCHAR(255)

cuisine

| placeID | INT |
| cuisine | VARCHAR(255) |

rating

userID	VARCHAR(255)
placeID	INT
rating	INT
food_rating	INT
service_rating	INT

parking

| placeID | INT |
| parking_lot | VARCHAR(255) |

Observations

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

Which restaurants have the best family ratings?
What restaurant is preferred by particular religion?
Which City has the most elevated no of restaurants with higher rating
Does having a Parking lot affect the restaurant rating
Does serving alcohol affect user rating
Number of restaurants based on cuisine type

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

```python
# 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

```python
# 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')
```

```python
# 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

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

- Used Pandas functions to catch Missing/Null values

```python
# 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.

```python
# 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

```python
# 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

```python
# 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)
```

```python
# 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

- Removed few special characters from "City" names to have clean dataset

```python
places_df = places_df.replace(["El Rinc◆n de San Francisco"], "El Rincon de San Francisco")
places_df = places_df.replace(["vips"], "VIPS")

places_df["name"].unique()
```

```
]: array(['Kiku Cuernavaca', 'puesto de tacos', 'El Rincon de San Francisco',
          'little pizza Emilio Portes Gil', 'carnitas_mata',
          'Restaurant los Compadres', 'Taqueria EL amigo ', 'shi ro ie',
```

- Used Sqlalchemy to connect to Postgres Database

```python
# 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.

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

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

```
1
2  SELECT * FROM USER_PROFILE;
3
```

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

Query Editor   Query History

```
1
2   SELECT * FROM GEOPLACES;
3
```

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	-100.976092799999999	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

```
1
2   SELECT * FROM RATING;
3
```

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

PARKING

CUISINE

Query Editor   Query History

```
1
2   SELECT * FROM CUISINE;
3
```

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

Query Editor   Query History

```
1
2   SELECT * FROM PARKING;
3
```

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 restaurants have the best family ratings?

Overall, there was not much of a difference between the different rating criteria. The places that did well in food and service rating, correlated with the overall rating, and vice versa.

```sql
-- Getting the 5 best family restaurants based on overall rating
SELECT geoplaces.placeid, geoplaces.name, geoplaces.ambience, rating.placeid, SUM(rating.rating) AS "Overall Rating Sum"
FROM rating
INNER JOIN geoplaces ON
geoplaces.placeid = rating.placeid
WHERE geoplaces.ambience = 'familiar'
GROUP BY geoplaces.placeid, rating.placeid
ORDER BY SUM(rating.rating) DESC
LIMIT 5;
```

	placeid integer	name character varying (255)	ambience character varying (255)	placeid integer	Overall Rating Sum bigint
1	135085	Tortas Locas Hipocampo	familiar	135085	48
2	132825	puesto de tacos	familiar	132825	41
3	135032	Cafeteria y Restaurant El Paci…	familiar	135032	33
4	135052	La Cantina Restaurante	familiar	135052	32
5	135038	Restaurant la Chalita	familiar	135038	29

Summary

What restaurant is preferred by particular religion?

Based on results, **"Tortas Locas Hipocampo"** restaurant is top choice for Catholic people while other religion's don't have any preference to specific restaurant.

	religion character varying (255)	name character varying (255)	cnt bigint	max_cnt bigint
1	Catholic	Tortas Locas Hipocampo	28	28

	religion character varying (255)	name character varying (255)	cnt bigint	max_cnt bigint
2	Christian	Mariscos Tia Licha	2	2
3	Christian	Taqueria EL amigo	2	2
4	Christian	palomo tec	2	2

	religion character varying (255)	name character varying (255)	cnt bigint	max_cnt bigint
20	Mormon	Cafeteria y Restaurant El Pacifico	1	1
21	Mormon	Gorditas Doa Gloria	1	1
22	Mormon	Unicols Pizza	1	1
23	Mormon	Tortas Locas Hipocampo	1	1

```
73  SELECT SRC.RELIGION,SRC.NAME,SRC.CNT,SRC.MAX_CNT
74  FROM
75  (
76      SELECT UP.RELIGION,G.NAME,COUNT(G.NAME) AS CNT,
77      MAX(COUNT(G.NAME)) OVER (PARTITION BY UP.RELIGION) AS MAX_CNT
78      FROM USER_PROFILE UP,
79      RATING R,
80      GEOPLACES G
81      WHERE UP.USERID = R.USERID
82      AND R.PLACEID = G.PLACEID
83      AND UP.RELIGION != 'none'
84      GROUP BY UP.RELIGION,G.NAME
85      ORDER BY COUNT(G.NAME) DESC
86  ) SRC
87  WHERE SRC.CNT = SRC.MAX_CNT
```

	religion character varying (255)	name character varying (255)	cnt bigint	max_cnt bigint
12	Jewish	carnitas_mata	1	1
13	Jewish	Gorditas Dona Tota	1	1
14	Jewish	Little Cesarz	1	1
15	Jewish	little pizza Emilio Portes Gil	1	1

Summary

Which City has the 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.

```
31  -- Which City has highest no of restaurants with higher rating
32  SELECT COUNT(R.RATING),G.CITY
33  FROM USER_PROFILE UP,
34  RATING R,
35  GEOPLACES G
36  WHERE UP.USERID = R.USERID
37  AND R.PLACEID = G.PLACEID
38  AND G.CITY != 'Not Recorded'
39  GROUP BY G.CITY
40  HAVING MAX(R.RATING) = 2 --- 2 being the highest rating
41  ORDER BY COUNT(R.RATING) DESC;
42
```

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

```
23  SELECT P.PARKING_LOT,R.RATING, COUNT(R.RATING) AS "TOTAL RATING"
24  FROM USER_PROFILE UP,
25  RATING R,
26  GEOPLACES G,
27  PARKING P
28  WHERE UP.USERID = R.USERID
29  AND R.PLACEID = G.PLACEID
30  AND G.PLACEID = P.PLACEID
31  GROUP BY P.PARKING_LOT,R.RATING
32  ORDER BY COUNT(R.RATING) DESC
33  LIMIT 5;
34
```

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 having a 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.

Does serving alcohol affect user rating?

Alcohol was broken down into 3 categories and based on the highest rated restaurants, serving alcohol does not affect the total rating.

```
12   SELECT g.alcohol, r.rating, COUNT(r.rating) AS "Total Rating"
13   FROM rating r, geoplaces g
14   WHERE r.placeid = g.placeid
15   GROUP BY g.alcohol, r.rating
16   HAVING MAX(r.rating) = 2 --- 2 being the highest rating
17   ORDER BY COUNT(r.rating) DESC;
18
```

Data Output   Explain   Notifications   Messages

	alcohol character varying (255)	rating integer	Total Rating bigint
1	No_Alcohol_Served	2	303
2	Wine-Beer	2	146
3	Full_Bar	2	37

Summary

Number of restaurants based on cuisine type.

Among the cities in the dataset, restaurants that serve Mexican cuisine are more popular followed by International, Dutch, American and Italian cuisine.

```sql
18  SELECT c.cuisine, COUNT(g.name)
19  FROM geoplaces g, cuisine c
20  WHERE g.placeid = c.placeid
21  AND g.city != 'Not Recorded'
22  GROUP BY c.cuisine
23  ORDER BY COUNT(g.name) DESC
24  LIMIT 5;
25
```

Data Output   Explain   Notifications   Messages

	cuisine character varying (255)	count bigint
1	Mexican	101
2	International	48
3	Dutch-Belgian	44
4	American	35
5	Italian	31