Restaurant-Customer Ratings Database

Group 7
Heer Parekh
Joshna Devi Vadapalli
Vamshikrushna Lakavath

Department of Applied Data Science
San Jose State University

Major Professor: Dr. Eduardo Chan

CONTENTS

Abstract	**3**
Description	**4**
Database Initial Study	**5**
Database Design - Conceptual Design	**8**
Database Design - Logical Design	**10**
Database Design - Physical Design	**17**
DBMS Selection	**19**
Implementation & Loading	**20**
Testing & Evaluation	**33**
Query Descriptions	**41**
Summary & Conclusion	**51**
Future Work	**52**

Abstract

With the increase in the number of restaurants people often get confused about the best-suited restaurant according to their preferences. In addition to that, people face a hard time finding out the best place and food to eat, especially when they are new to that place. The Restaurant-consumer rating database helps build a restaurant recommendation system that suggests a top-n-list of restaurants based on the user cuisine preferences. This will save time for users while choosing their preferred restaurants. It is so convenient to get a list of restaurants that match our preferences without much clicking, comparing, and browsing through a long list of reviews for every single business. In addition, restaurants can also improve their business by providing better services based on user ratings.

Description

Today, online reviews are the biggest source of social proof, and they have a clear impact on sales. Since more than 80 percent of customers incorporate ratings into their purchasing decisions especially for restaurants and cafes compared to other types of business.

Restaurants can build significant trust and credibility from a steady stream of positive reviews. Many restaurants today ask their customers for reviews and ratings to help them serve better. This rating strategy will not only help the restaurants to improve customer satisfaction but also in the successful growth of the restaurant.

Not every customer spends time reading through the reviews, but prefer a glance at the overall ratings of the restaurant. So, we have taken ratings from the customers based on the below attributes and provide an overall rating for the restaurant.

Keeping this in mind, we chose to study different restaurant's datasets according to the customer's ratings and user fondness for a particular cuisine.

We design a database that shows the different salient features of the restaurant based on the customer preference for the various cuisine types and will sort out the restaurants for the customer.

The database takes the food preference and ratings into consideration to recommend food to the users. The database uses item-based collaborative filtering and user-based collaborative filtering method to recommend the food to the users. The database takes user ratings for different food items and stores them into the database. The database then recommends food items to the users based on their ratings.

The above consideration would be a win-win situation for both the customers and the restaurant as it will help the restaurants to provide better service, increase their revenue and reputation in the market and the customers will be benefited by knowing the top restaurants in the city based on their cuisine preference.

Database Initial Study

● Analysis & Requirements

We have three main datasets (Restaurant User, Restaurant profile and Restaurant ratings).The datasets describe the restaurants, their customers, and ratings given by the customers. The datasets include both numerical and categorical fields.

Restaurant Customer data has 10 attributes for 138 customers with each customer having a unique userID. Restaurant data has 8 attributes for 707 restaurants with each restaurant having a unique placeID.Restaurant Ratings data has 8 attributes and total 1161 customer ratings that customers (userID) provided for various restaurants (placeID).

The Restaurant dataset details include:

- Restaurant name and address
- Cuisine Type
- Working Hours
- Payment Method
- Restaurant Facilities

The Customer dataset details include:

- User Profile
- Cuisine Preference
- Payment Preference

The Ratings dataset is based on below ratings given by the user for each restaurant:

- Cost
- Food
- Service
- Parking
- Waiting Time
- Overall Rating

- ## Problems

The problem in finding the best restaurants according to the user choice has been a challenging task nowadays since the restaurant businesses are ever-increasing.
Also for the restaurants to understand and apply different marketing strategies to attract customers is difficult.In order to address this problem we have designed a recommendation system which shows the restaurants and its ratings.

- ## Constraints

Only the registered restaurants will be listed in the database. Also, the registered users will be allowed to rate the restaurants and their services.One user will be allowed to rate a restaurant only once.

- ## Objective

The main objective is to provide restaurant details and facilities namely restaurant name,address,services and cuisines available. This will help restaurants to know what types of customers are there in the region. Next by suggesting best restaurants to the customers according to their budget, taste, accessibility, ambiance preference and food choice. Also, providing Economic status of restaurants based on the customers budget. Making the data available to customers regarding the restaurants according to customer preferences. Lastly,determining customer satisfaction and requirements based on food quality,cost,waiting time,parking availability and services provided by the restaurant. To conclude, this database will help customers find restaurants based on their cuisine preferences, affordability, accessibility and restaurant ratings and also the restaurants to find the user preferences and budget which improves the business in the market.

- ### Scope

All the restaurants in Mexico city can be listed in the database. It will help the customers to choose the restaurants based on their preferences. Also, restaurants can mould their business according to the users preferences by analysing their ratings. In the future we can use the same Database design for those restaurants which are in any region or country.

This recommendation database system can be used by all the people who want to visit and rate the restaurants in Mexico.Also, can be accessed by restaurant owners or

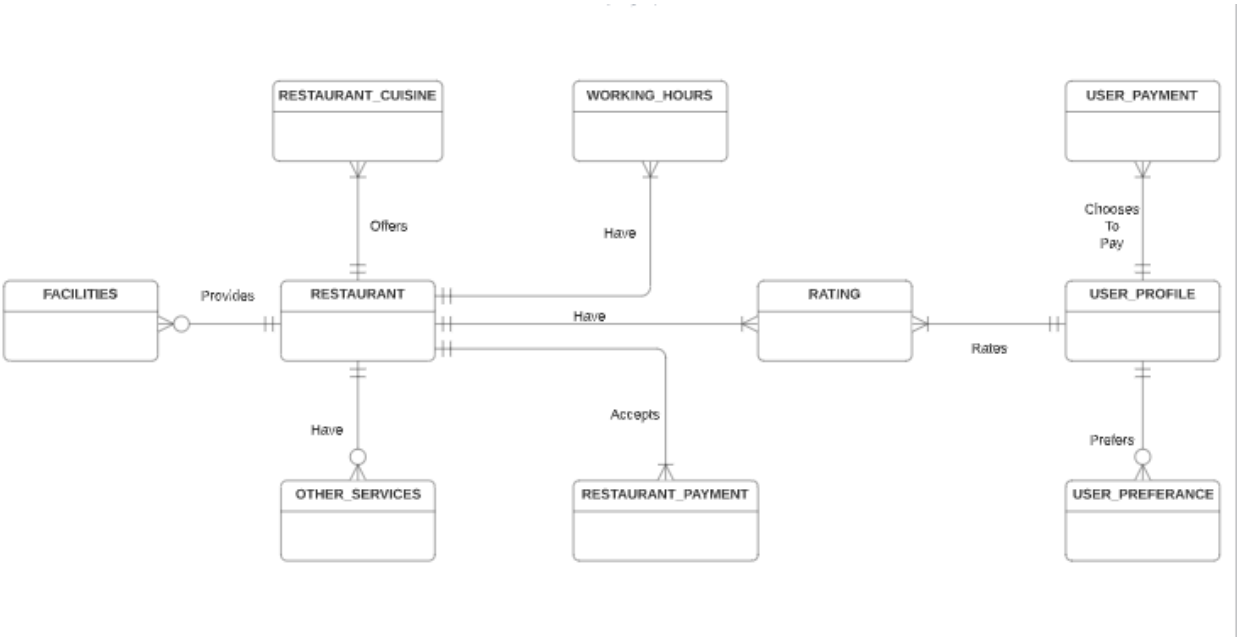
managers to understand its reputation in the restaurant business which would help them increase their revenues.

- **Boundaries**

  We can only provide the details of those restaurants which are listed in our dataset, and while calculating the restaurant's rating we can only consider the ratings of those customers whose userID are listed in the dataset.
  Currently, In our dataset, we have only those restaurant details which are in Mexico. Also, only the registered users will be allowed to rate the restaurants and their services. And one user will be allowed to rate a restaurant only once. But our database is unique; it will work for any country's restaurants.

Database Design - Conceptual Design

The conceptual model is the overall view of the entire database by the entire organization. It consists of the entities and their relationships as shown below in the diagram.



ENTITY	RELATIONSHIP	CONNECTIVITY	ENTITY
RESTAURANT	provides	1:M	FACILITIES
RESTAURANT	have	0:M	OTHER_SERVICES
RESTAURANT	offers	1:M	RESTAURANT_CUISINES
RESTAURANT	has	1:M	RATING
RESTAURANT	accepts	1:M	RESTAURANT_PAYMENT
RESTAURANT	have	1:M	WORKING_HOURS
USER_PROFILE	rates	1:M	RATINGS
USER_PROFILE	choses to pay	1:M	USER_PAYMENT
USER_PROFILE	prefers	0:M	USER_PREFERENCES

The restaurant offers at least one type of cuisine and one cuisine is a part of one restaurant.Therefore the relationship between restaurant cuisine and the restaurant is (1:M).

- **Business Rules**

  A restaurant can provide many cuisines. Each cuisine can be provided by one restaurant.

  A restaurant may offer zero or more facilities. Each facility can be offered by one restaurant.

  A restaurant can have many working hours. Each working hour can be assigned to one restaurant.

  A restaurant can provide many payment methods. Each payment is associated with one restaurant.

  A restaurant can have many services. Each service is being offered by one restaurant.

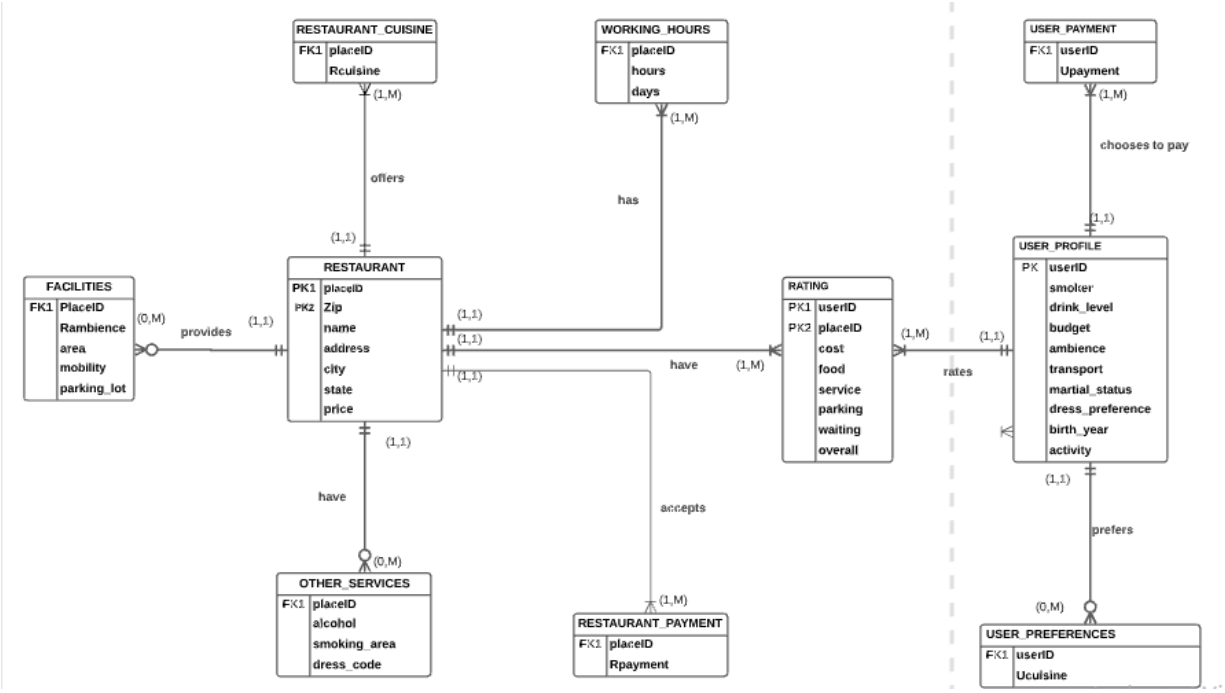
  A rating can be provided by one user to each restaurant. Each restaurant can be rated by many users.

  A user can have zero or many cuisine preferences. Each cuisine preference is associated with one user.

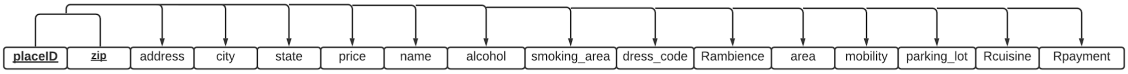
  A user can have many payment modes. Each payment is associated with a single user.

Database Design - Logical Design

Logical Database Design helps us to decide how to sort and group the attributes of the entities into tables of a relational database. We have created proficient structured tables and properly reflect our Restaurant-Customer relationship.The tables store the data as entities in a non-redundant manner with their respective foreign keys that are placed in the tables in order to support all the relationships between the entities.We have graphically represented this information in the form of the below ER Diagram with Crow's Foot Notation..



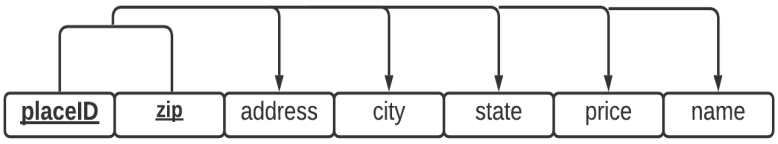
The Restaurant profile dataset details includes:



Since all other attributes are partially dependent on placeID, to remove the partial dependency , we have divided the restaurant profile table to different new tables as below and reassigned corresponding dependent attributes to the new tables.

- **RESTAURANT**

1. **Restaurant Name** : The name of the restaurant.
2. **PlaceID** : is the restaurant ID.This is a unique ID that is provided to the restaurant and cannot be kept NULL.
3. **Address**: is the address of the restaurant.
4. **City**: where the restaurant is located.
5. **State**: state where the restaurant is located.
6. **Zip**: is the zip code for the restaurant.
7. **Price**: Price range like low,medium or high of the restaurant.Based on the customer's budget, the restaurant has been given this price range.

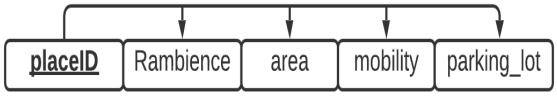


- **RESTAURANT CUISINE**

1. **PlaceID**: The unique restaurant ID.
2. **RCuisine**: : Consists of all the cuisines that are provided by the restaurant.

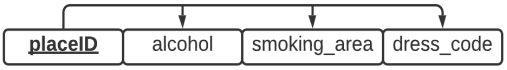
- **FACILITIES**

1. **PlaceID**: The unique restaurant ID
2. **Rambience**: The restaurant ambience if it is quite or familiar.
3. **Area**: The restaurant area whether it is open or closed.
4. **Mobility**:This feature or attribute is specially for the customers with disabilities for restaurant accessibility.It specifies if the restaurant is completely or partially accessible by the customers.
5. **Parking_lot**: Specifies if the restaurant has any parking space available or no.



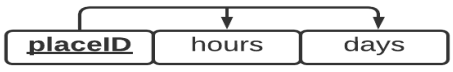
- **OTHER SERVICES**

1. **PlaceID**:The unique restaurant ID
2. **Alcohol**: If the restaurant serves any alcoholic beverages.
3. **Smoking Area**: The restaurant does or does not have a dedicated area for smoking and also if smoking is permitted in the restaurant.
4. **Dress_Code**: The preferred dress code in the restaurants.



- **WORKING_HOURS:**

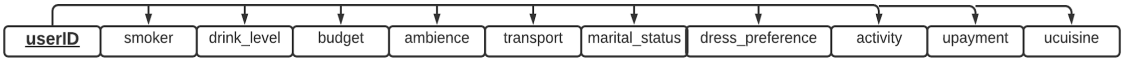
1. **PlaceID**:The unique restaurant ID
2. **Hours:** The timings for the restaurants are specified here.
3. **Days**: The days on which the restaurant is operational.



- **RESTAURANT_PAYMENT**

1. **PlaceID**: The unique restaurant ID
2. **RPayment**: The preferred payment method accepted by the restaurant.

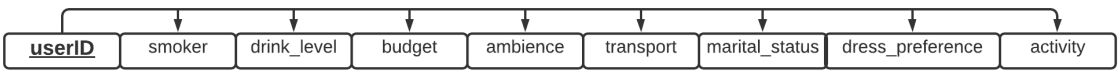
The Users dataset details include:



Since all other attributes are partially dependent on userID, to remove the partial dependency dividing the restaurant profile table to different new tables as below and reassigning corresponding dependent attributes to the new tables.

- **USER_PROFILE:**

1. **User ID**: is the customer's unique ID.Every Customer who registers would be given a unique ID that cannot be NULL.
2. **Smoker**: represents whether a user is a smoker or not
3. **Drink level** : represents the level of drinking nature of the customer
4. **Budget** : represents the budget range of the customer like low,medium or high.
5. **Ambiance**: is the ambiance that the user prefers.
6. **Transport**: the mode of transportation that the user has and mostly prefers.
7. **Marital status**: represents the marital status of the user.
8. **Dress preference**: The dress code the user prefers.
9. **Activity** : the profession of the user.



- **USER_PREFERENCE**

1. **User_ID:** customer's unique ID
2. **UCuisine:** represents the users cuisine preferences.

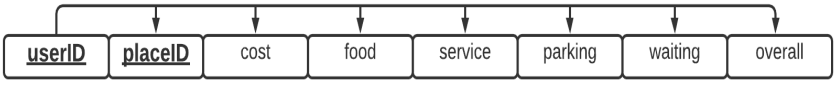
- **USER_PAYMENT**

1. **User_ID:** customer's unique ID
2. **UPayment:** The preferred payment method used by the user

The Ratings dataset details include:

- **RATING**

1. **Cost**: rating for the affordability of restaurant based on quality
2. **Food**: rating for the food quality and quantity
3. **Service**: other services provided by the restaurant
4. **Parking**: rating for the parking facility provided by the restaurant
5. **Waiting Time**: If there is a waiting time the customers rating for the restaurant
6. **Overall Rating**: the average of the all other ratings

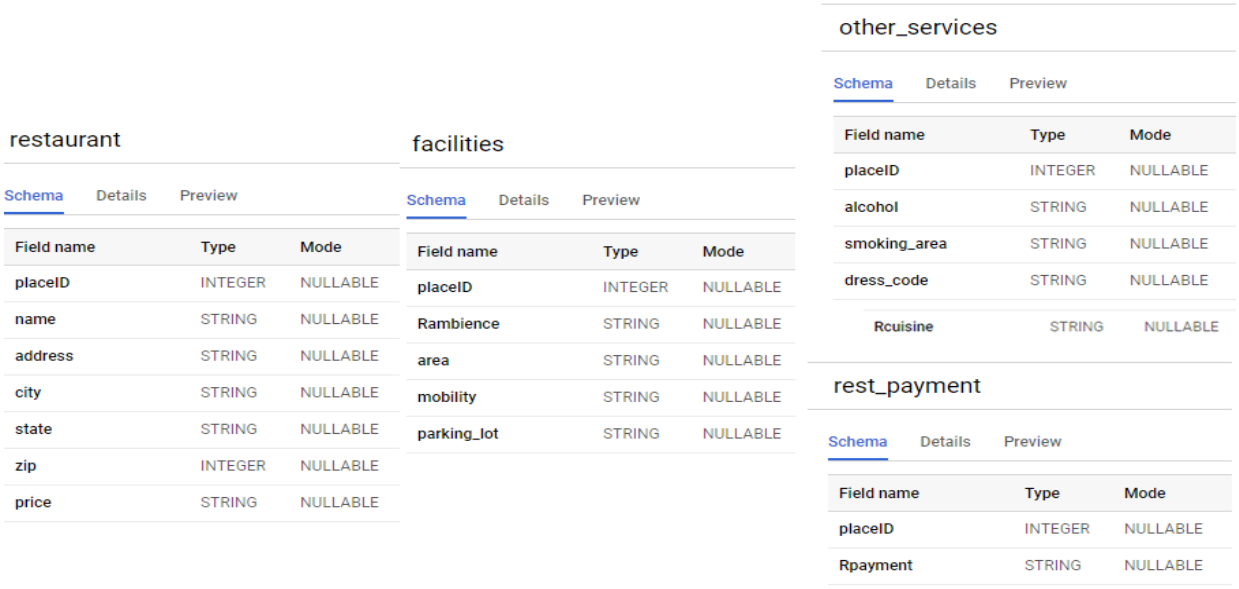


- # Schema & Constraints

In bigquery, the database schema is specified at the time of loading the data into the tables automatically. The database schema consists of three main columns:

1. Name: Signifies the column name and should be defined only in letters,numbers or underscores and must start with a letter or underscore.
2. Data Type: Data Types like INT64 for integer, NUMERIC for numeric data, STRING for string data are some of the data types that are used in the schema.
3. Mode: Mode specifies, if the respective column values can be NULL or not.If the mode is not specified, the column will default to NULLABLE
    - NULLABLE: Column values can be NULL
    - REQUIRED: Column values cannot be NULL
    - REPEATED: Column contains array of values of specific type.

In our database, we have the below mentioned tables and there specific schemas



restaurant

Schema Details Preview

Field name	Type	Mode
placeID	INTEGER	NULLABLE
name	STRING	NULLABLE
address	STRING	NULLABLE
city	STRING	NULLABLE
state	STRING	NULLABLE
zip	INTEGER	NULLABLE
price	STRING	NULLABLE

facilities

Schema Details Preview

Field name	Type	Mode
placeID	INTEGER	NULLABLE
Rambience	STRING	NULLABLE
area	STRING	NULLABLE
mobility	STRING	NULLABLE
parking_lot	STRING	NULLABLE

other_services

Schema Details Preview

Field name	Type	Mode
placeID	INTEGER	NULLABLE
alcohol	STRING	NULLABLE
smoking_area	STRING	NULLABLE
dress_code	STRING	NULLABLE
Rcuisine	STRING	NULLABLE

rest_payment

Schema Details Preview

Field name	Type	Mode
placeID	INTEGER	NULLABLE
Rpayment	STRING	NULLABLE

user_profile

Schema Details Preview

Field name	Type	Mode
userID	STRING	NULLABLE
smoker	STRING	NULLABLE
drink_level	STRING	NULLABLE
budget	STRING	NULLABLE
ambience	STRING	NULLABLE
transport	STRING	NULLABLE
marital_status	STRING	NULLABLE
dress_preference	STRING	NULLABLE
activity	STRING	NULLABLE

user_payment

Schema Details Preview

Field name	Type	Mode
userID	STRING	NULLABLE
Upayment	STRING	NULLABLE

user_preference

Schema Details Preview

Field name	Type	Mode
userID	STRING	NULLABLE
Ucuisine	STRING	NULLABLE

rating

Schema Details Preview

Field name	Type	Mode
userID	STRING	NULLABLE
placeID	INTEGER	NULLABLE
cost	INTEGER	NULLABLE
food	INTEGER	NULLABLE
service	INTEGER	NULLABLE
parking	INTEGER	NULLABLE
waiting	INTEGER	NULLABLE
overall	FLOAT	NULLABLE

**As we have uploaded the csvs for the above tables in bigquery, the mode for these tables is NULLABLE as there is no option to change/edit the schema once upload is done.

Database Design - Physical Design

- **Access Control:**

  As we are using BigQuery as our database, it provides several layers of access control. The top layer is primitive roles, which acts at the project level. They're administered through IAM, the Identity and Access Management system.

  - An owner has editor permissions, but can also delete all datasets and see all jobs for all users in the project.DB Admin has the owner role for all the tables and dataset.
  - The restaurant manager will have the editor role for the restaurant table. The user will have an editor role for his profile .
  - Restaurant manager and users have viewer role for all the other data tables.

- **Physical storage details:**

```
SELECT
  table_id,
  row_count AS rowscount,
  ROUND(size_bytes/1024) AS KB
FROM data225-spring-2021.termproject225.__TABLES__;
```

Row	table_id	rowscount	KB
1	facilities	114	5.0
2	other_services	114	5.0
3	rating	1161	71.0
4	ratingView	0	0.0
5	rest_payment	253	5.0
6	restaurant	707	67.0
7	restaurant_data	850	256.0
8	restautant_cuisine	553	10.0
9	user	416	75.0
10	user_payment	182	3.0

Security:

BigQuery automatically encrypts all data before it is written to disk. The data is automatically decrypted when read by an authorized user. By default, Google manages the key encryption keys used to protect your data. We can also use customer-managed encryption keys, and encrypt individual values within a table.

To encrypt individual values within a BigQuery table, use the Authenticated Encryption with Associated Data (AEAD).

DBMS Selection

- **DBMS selection**

  We selected BigQuery as our Data Base Management System.

- **DBMS analysis**

  Google BigQuery is a Cloud Datawarehouse run by Google. It is capable of analysing terabytes of data in seconds.one can access BigQuery by using the GCP console or the classic web UI, by using a command-line tool, or by making calls to BigQuery Rest API using a variety of Client Libraries such as Java, .Net, or Python. There are also a variety of third-party tools that one can use to interact with BigQuery, such as visualising the data or loading the data.

- **Key Features of BigQuery**
  1. Ease of Implementation: Building your own is expensive, time-consuming, and difficult to scale. With BigQuery, you need to load data first and pay only for what you use.
  2. Speed: Process billions of rows in seconds and handles real-time analysis of Streaming data
  3. Columnar storage.
  4. Flexibility and functionality for Nested/Repeated fields.
  5. Support for the DML and DDL languages
  6. No Index: Single full table scan.

- **Hardware & software:**

We are working on the Google Cloud Platform(GCP). On GCP we created a project from that project we connected to the BigQuery. From BigQuery we have connected to the SQL workspace and created a database and uploaded our datasets.

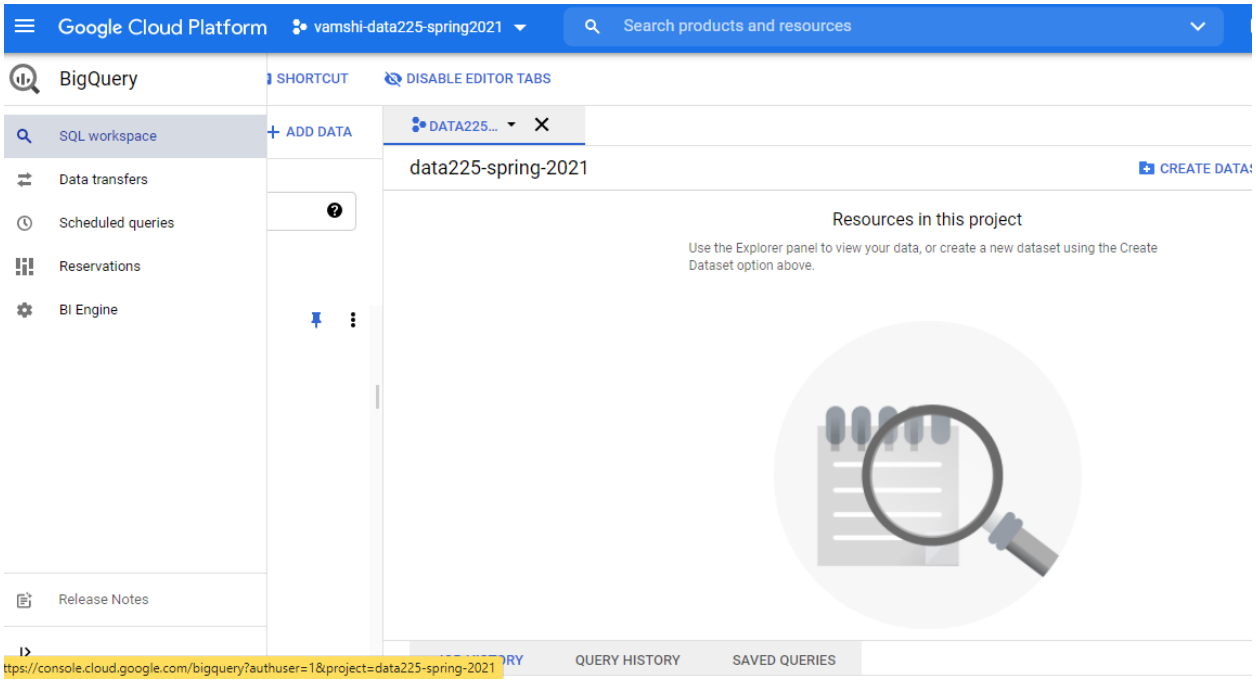
Implementation & Loading

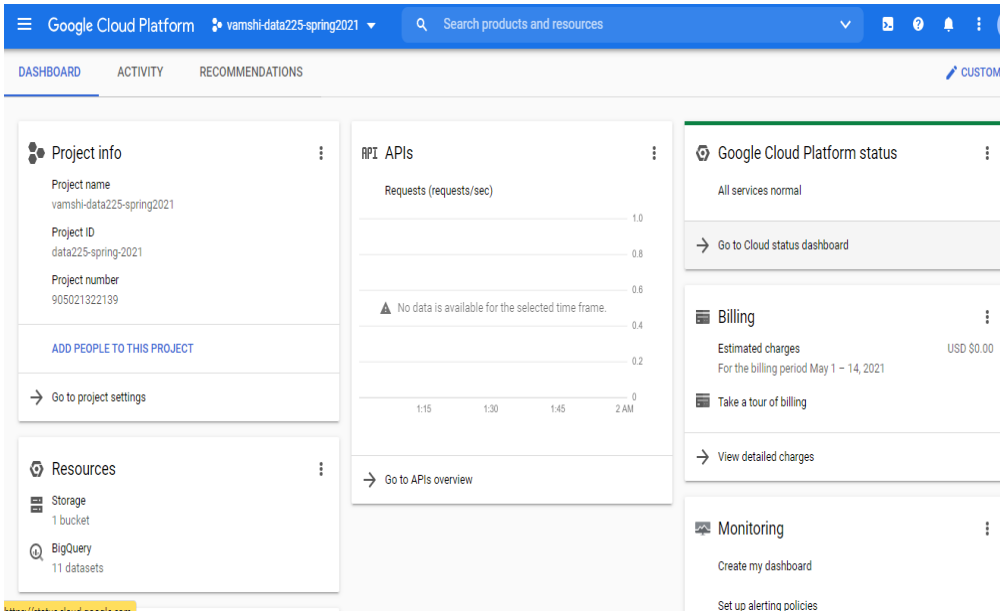
- **DBMS Installation:**

We chose to use Bigquery as our database which is a cloud database prominently used for data analytics platform. To make use of this database we should create a billing account in Google Cloud Platform (GCP).

For this project we have created a billing account which has been shared among the team numbers to access the projects and its resources.

Since, We are using BigQuery we need not install it manually in our local system; it is pre-installed and available virtually on the Google Cloud Platform (GCP).





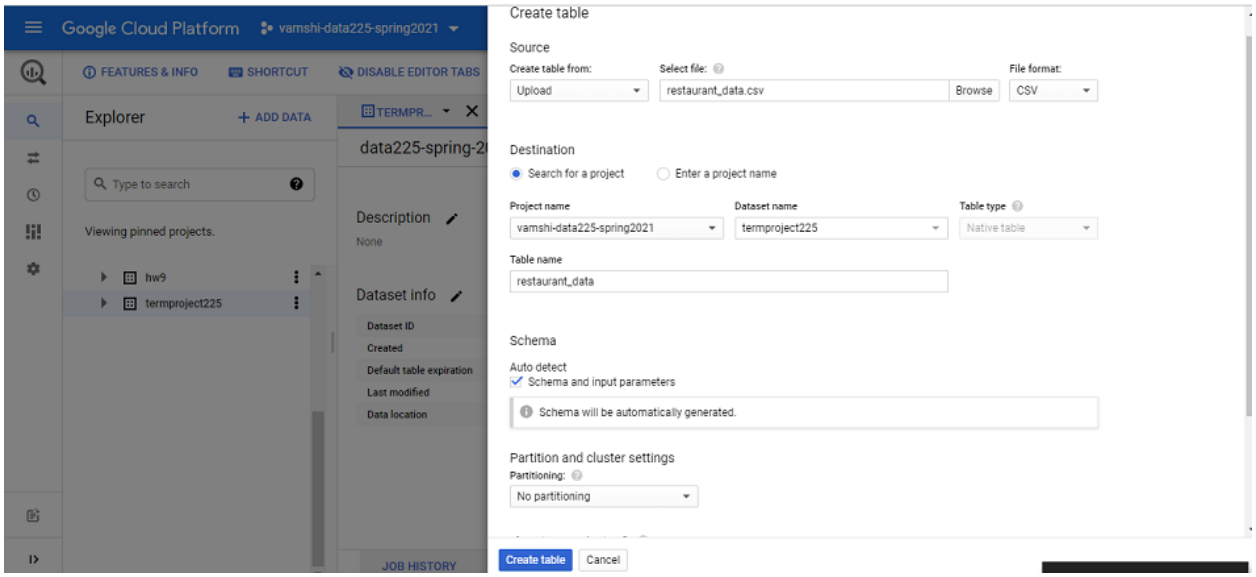
- **Schema creation:**

Before creating a schema for the tables, we have imported tour datasets that are in CSV format. Initially, we had three main datasets which are Restaurant profile, Users details, Ratings.

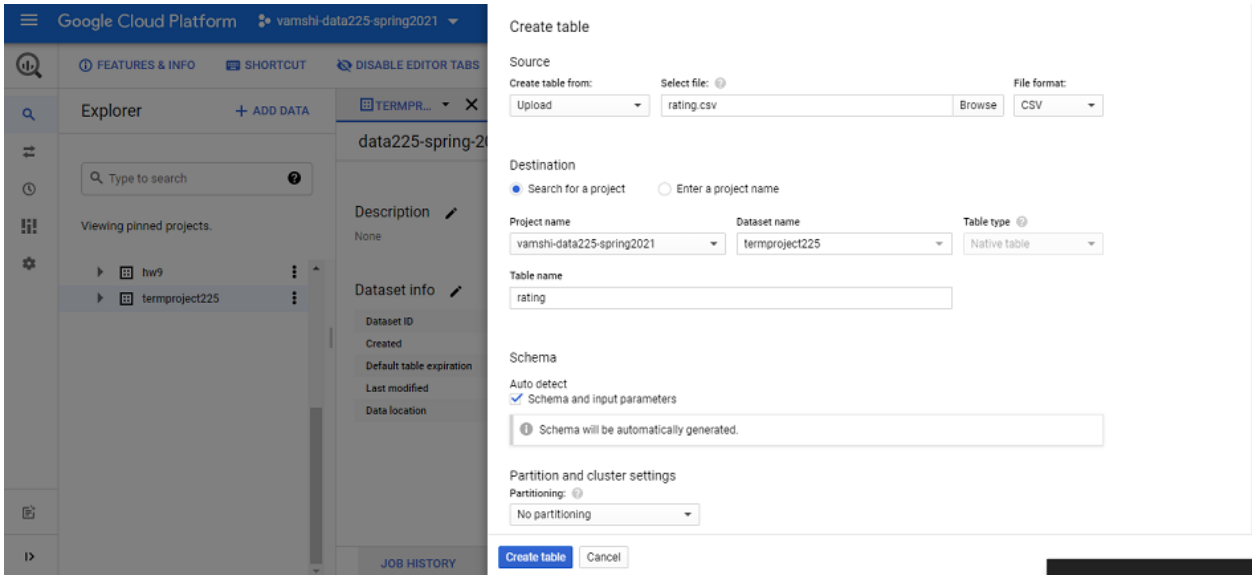
- **Data loading & conversion:**

1. Screenshots for import csvs.

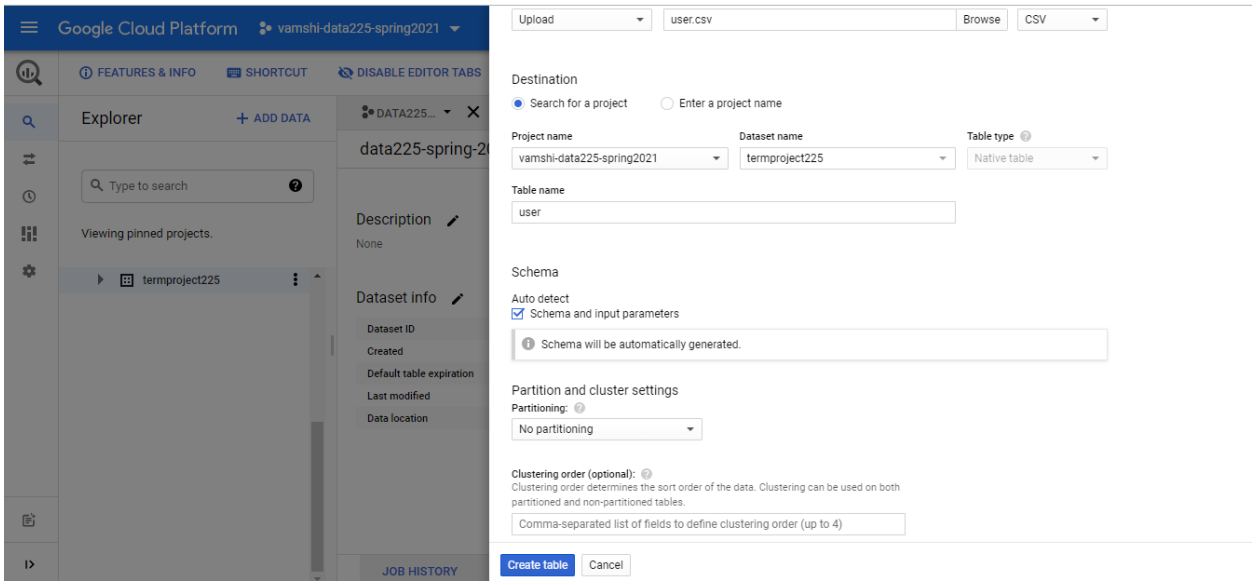
   Screenshot of importing restaurant_data dataset in to the BigQuery



Screenshot of importing ratings dataset in to the BigQuery



Screenshot of importing customer details dataset:

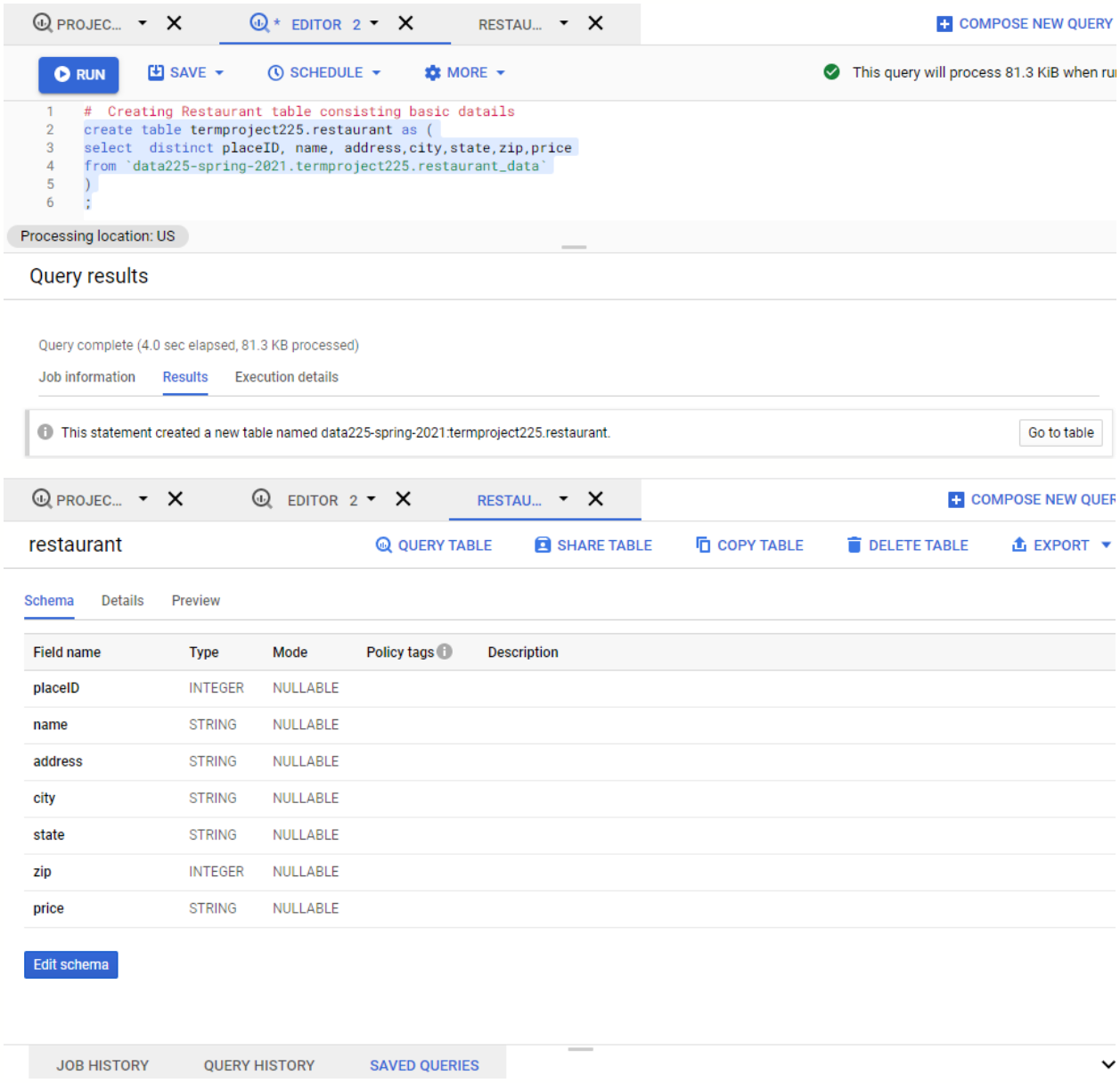


From these tables we have created other tables which we got from the normalization of the data. Below are the schema creations for the tables which are created from the restaurant profile table.

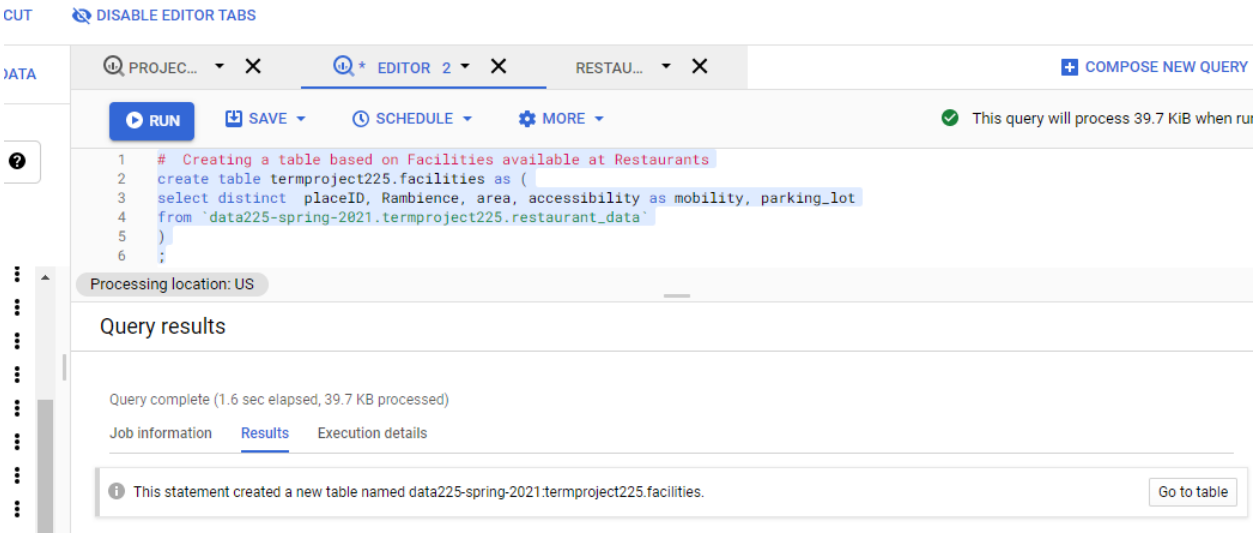
- **Data loading & conversion:**

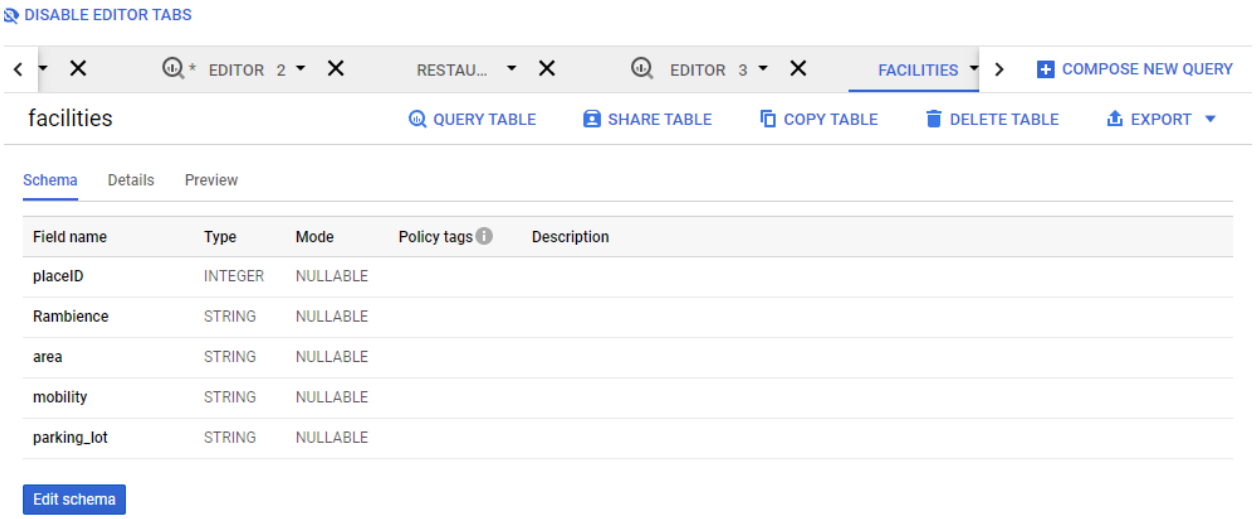
Creating Restaurant table consisting basic details of the restaurants like PlaceID, name, address, city, state, zip, price

```
create table termproject225.restaurant as (
select  distinct placeID, name, address,city,state,zip,price
from `data225-spring-2021.termproject225.restaurant_data`
)
;
```



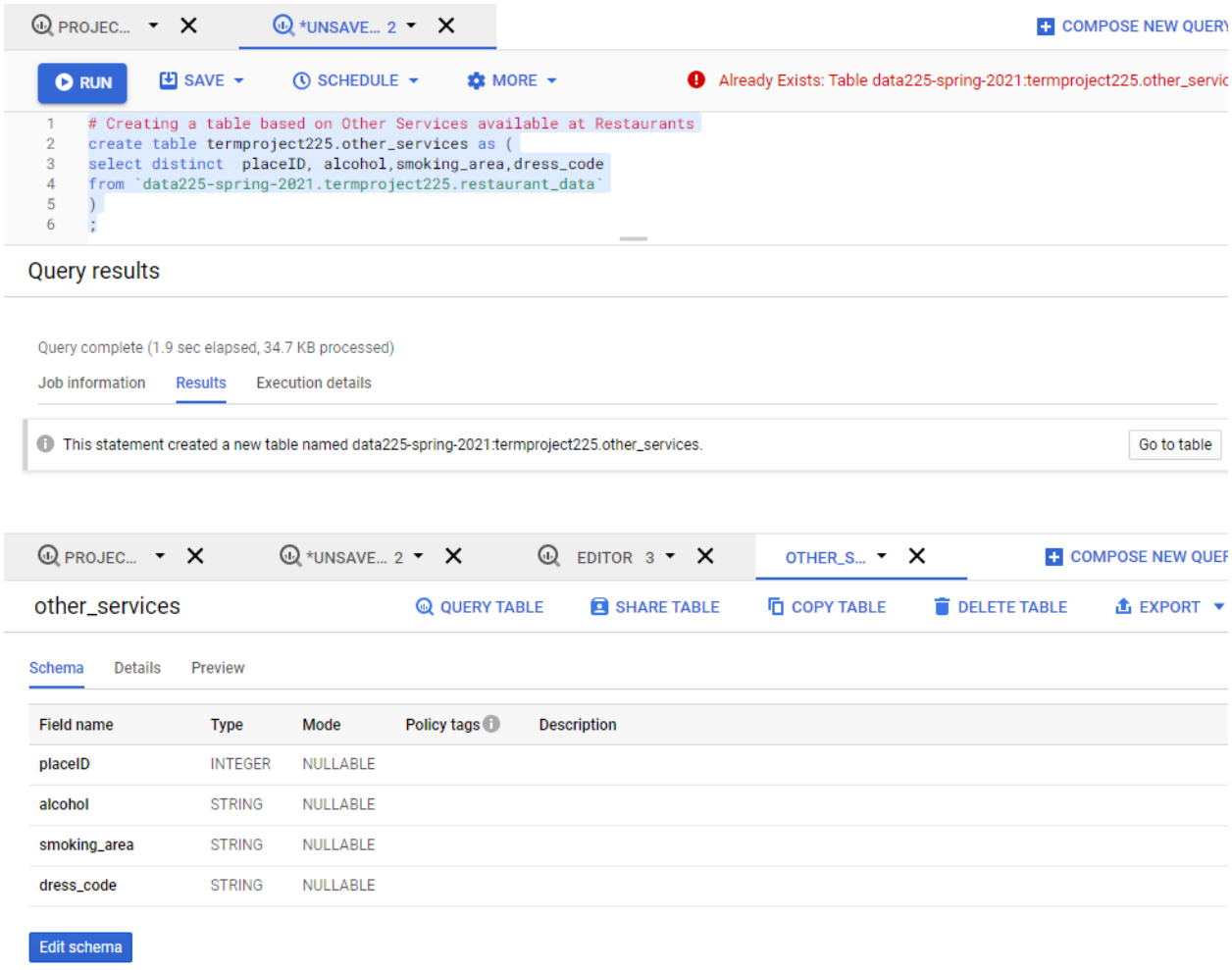

```
#  Creating Facilities table based on Facilities available at Restaurants
create table termproject225.facilities as (
select distinct  placeID, Rambience, area, accessibility as mobility, parking_lot
from `data225-spring-2021.termproject225.restaurant_data`
)
;
```





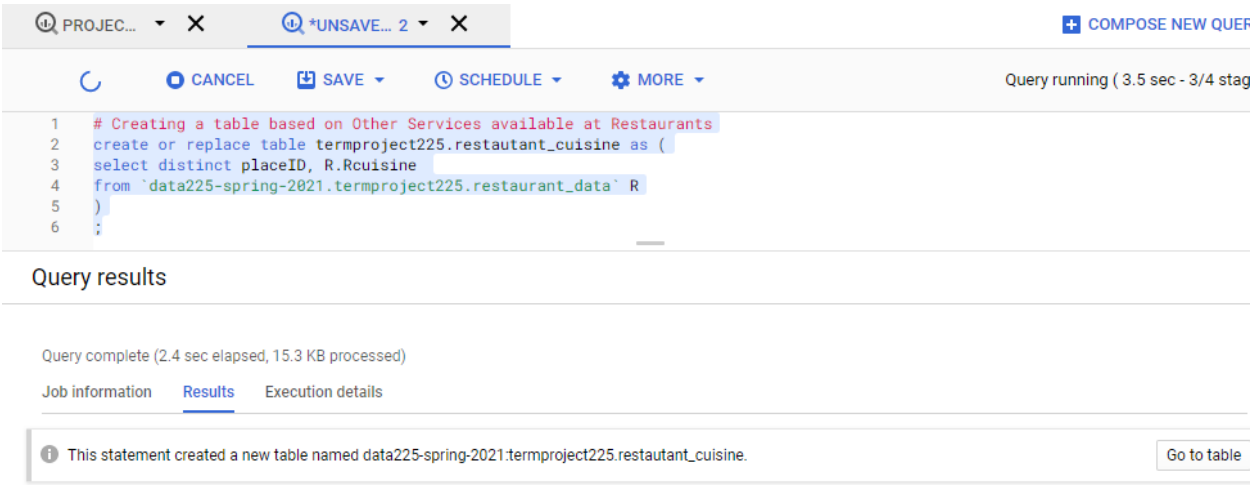
Creating other_services table based on Other Services available at Restaurants that consists of data like availability of alcohol, smoking area, and if there is any dress code or not.

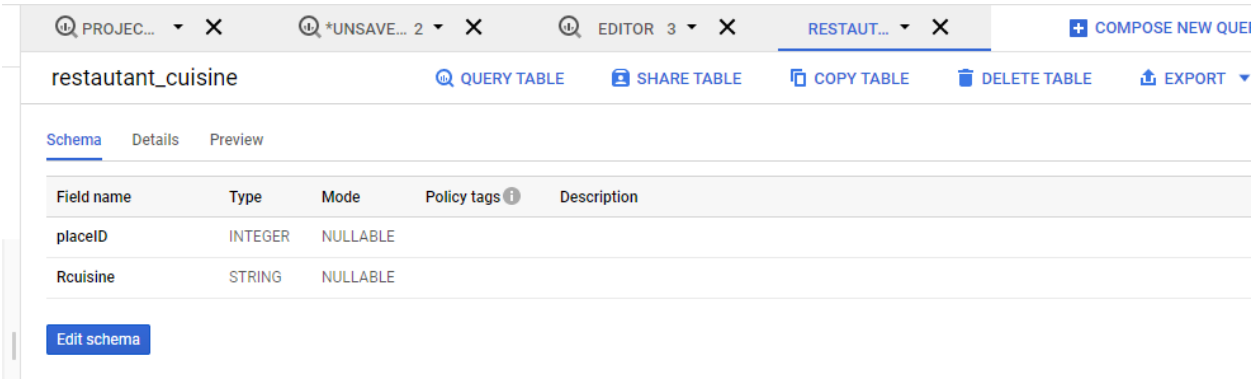
```
create table termproject225.other_services as (
select distinct  placeID, alcohol,smoking_area,dress_code
from `data225-spring-2021.termproject225.restaurant_data`
)
;
```



Creating restautant_cuisine table based on the different Cuisines available at Restaurants

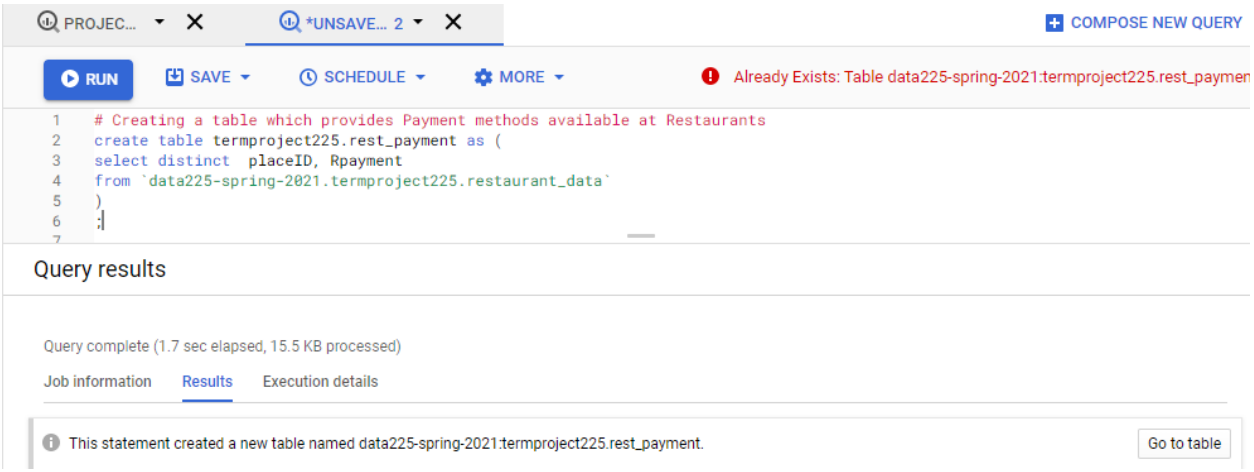
```
create table termproject225.restautant_cuisine as (
select distinct placeID, R.Rcuisine
from `data225-spring-2021.termproject225.restaurant_data` R
)
;
```

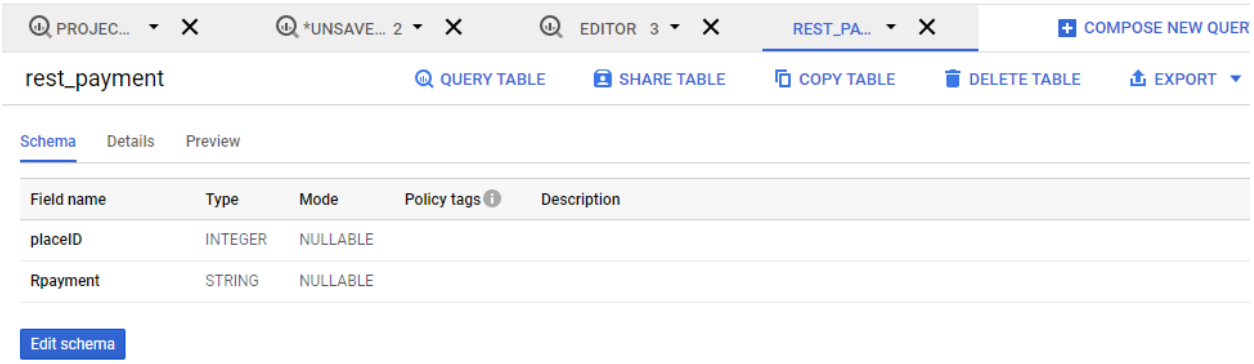




Creating restaurant payment table which provides Payment methods available at Restaurants

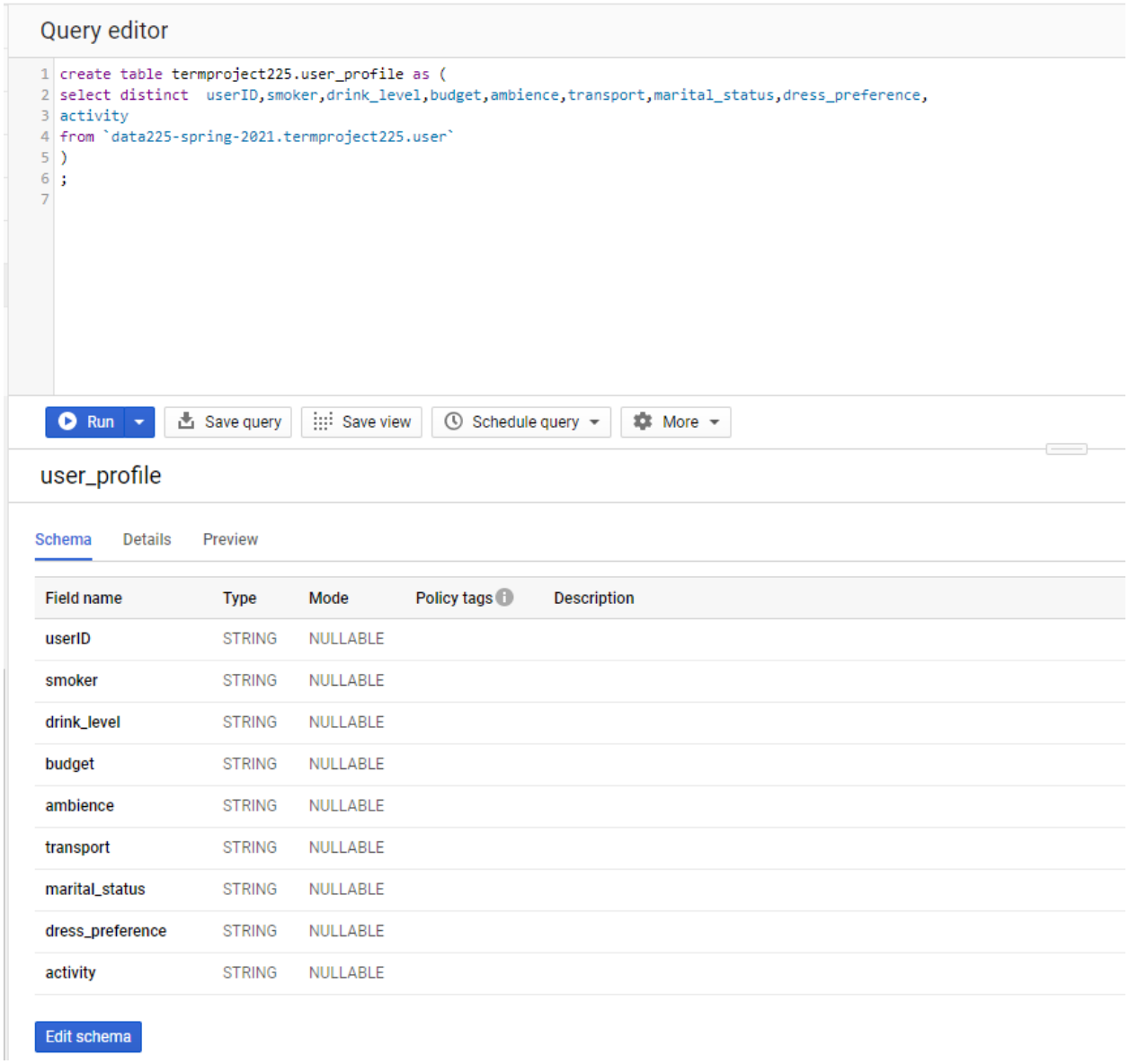
```
create table termproject225.rest_payment as (
select distinct  placeID, Rpayment
from `data225-spring-2021.termproject225.restaurant_data`
);
```





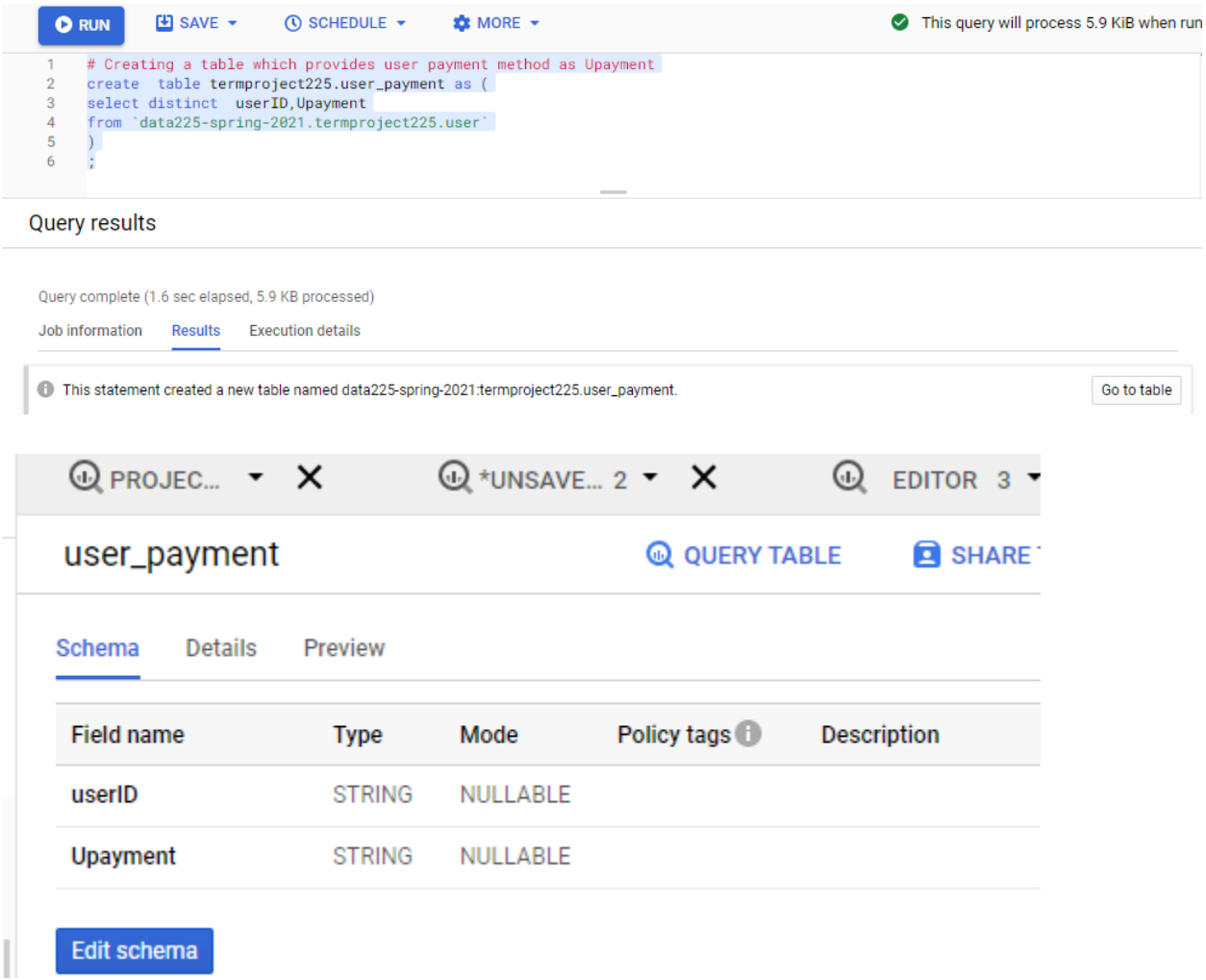
Creating a user profile table which provides user details like userID, if the user is a smoker or not, user's drinking level, budget range, ambiance preference, mode of transportation, marital status, and dress preference.

```
create table termproject225.user_profile as (
select distinct
userID,smoker,drink_level,budget,ambience,transport,marital_status,dre
ss_preference,
activity
from `data225-spring-2021.termproject225.user`
)
;
```



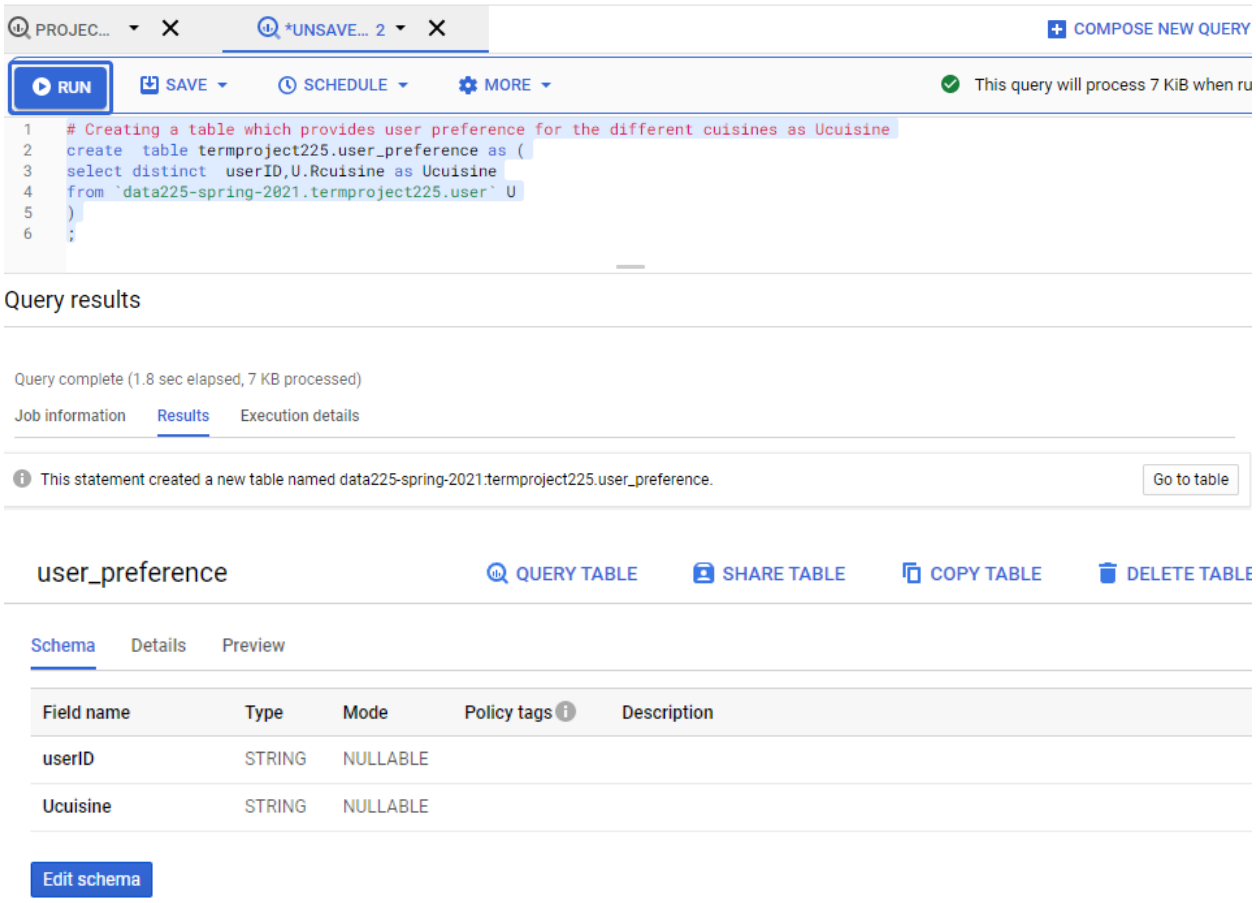
Creating a table which provides user payment method as Upayment

```
create  table termproject225.user_payment as (
select distinct  userID,Upayment
from `data225-spring-2021.termproject225.user`
)
;
```



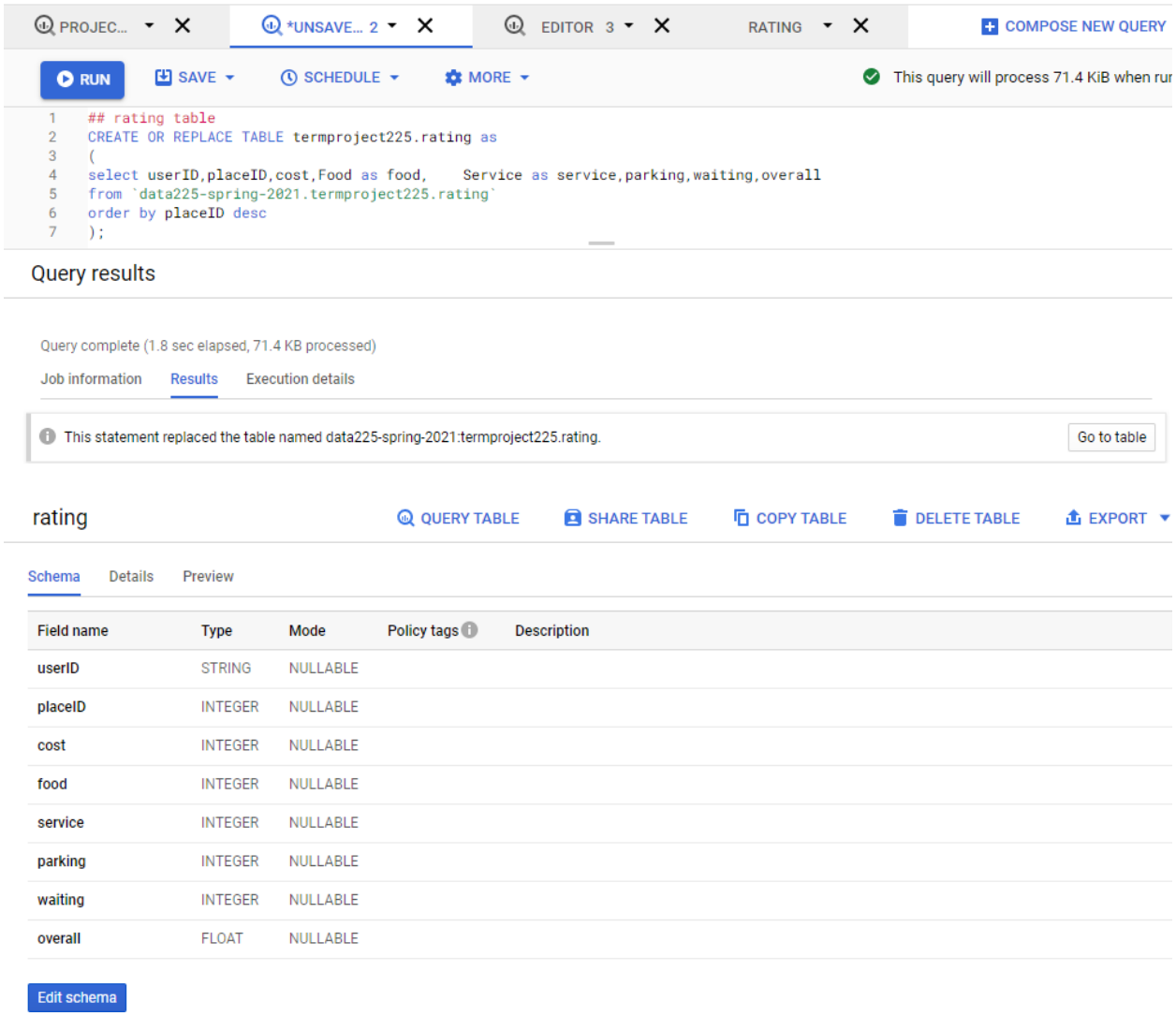
Creating a User preference table which provides user preference for the different cuisines as Ucuisine

```
create  table termproject225.user_preference as (
select distinct  userID,U.Rcuisine as Ucuisine
from `data225-spring-2021.termproject225.user` U
)
;
```



Ratings table schema:

```
create or replace table termproject225.rating as
(
select userID,placeID,cost,Food as food,    Service as service,parking,waiting,overall
from `data225-spring-2021.termproject225.rating`
order by placeID desc
);
```


PROJEC...
*UNSAVE... 2
EDITOR 3
RATING
COMPOSE NEW QUERY
RUN
SAVE
SCHEDULE
MORE
This query will process 71.4 KiB when rur
1 ## rating table
2 CREATE OR REPLACE TABLE termproject225.rating as
3 (
4 select userID,placeID,cost,Food as food, Service as service,parking,waiting,overall
5 from `data225-spring-2021.termproject225.rating`
6 order by placeID desc
7);
Query results
Query complete (1.8 sec elapsed, 71.4 KB processed)
Job information
Results
Execution details
This statement replaced the table named data225-spring-2021:termproject225.rating.
Go to table
rating
QUERY TABLE
SHARE TABLE
COPY TABLE
DELETE TABLE
EXPORT
Schema
Details
Preview
Field name
Type
Mode
Policy tags
Description
userID STRING NULLABLE
placeID INTEGER NULLABLE
cost INTEGER NULLABLE
food INTEGER NULLABLE
service INTEGER NULLABLE
parking INTEGER NULLABLE
waiting INTEGER NULLABLE
overall FLOAT NULLABLE
Edit schema

Testing & Evaluation

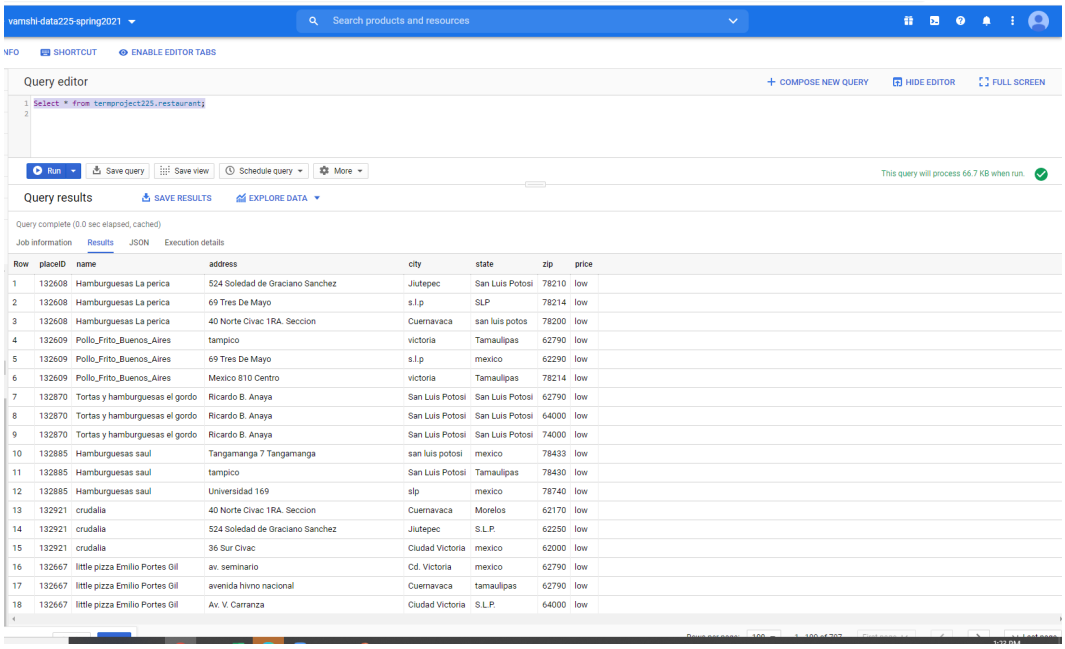
Database testing and evaluation consists of data validity, data integrity,performance check related to the database and testing for different queries like procedures, functions,views, etc in the database.

In our relational database, following checks are performed:

1. We have stored all the customer and restaurant information. While querying the database should provide accurate figures or data to the customer as and when required.
2. During data loading, the data should not be lost.
3. Testing each object in the schema.

For all the below tables we ensure referential integrity and entity integrity is followed.

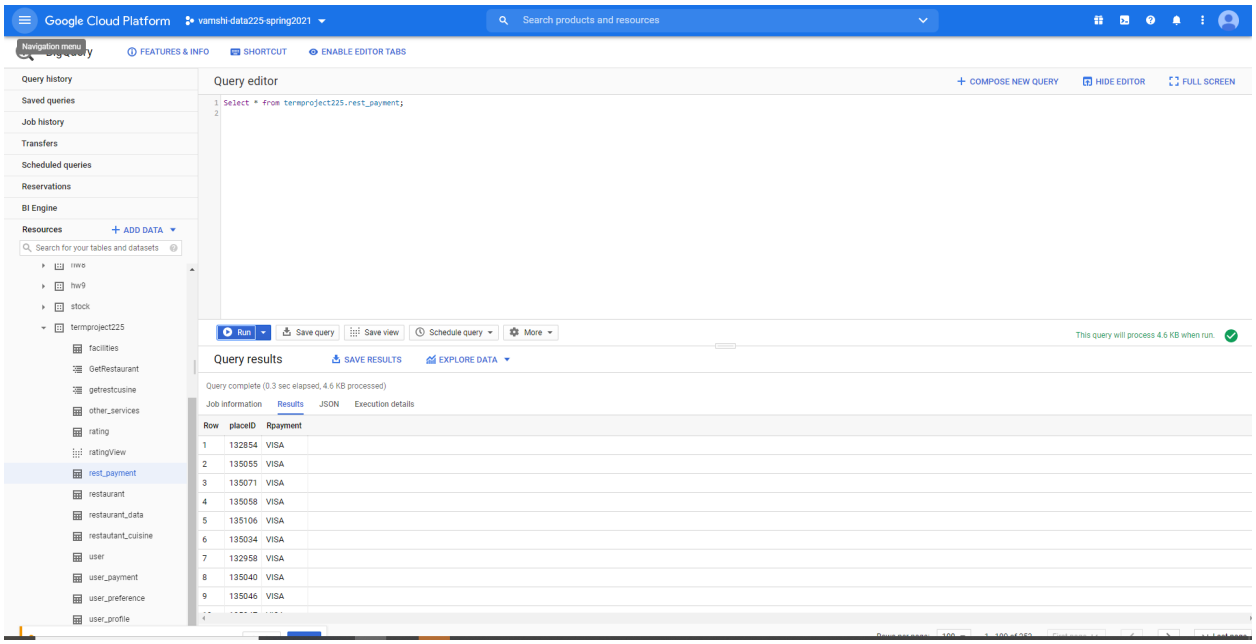
- Select * from termproject225.restaurant;



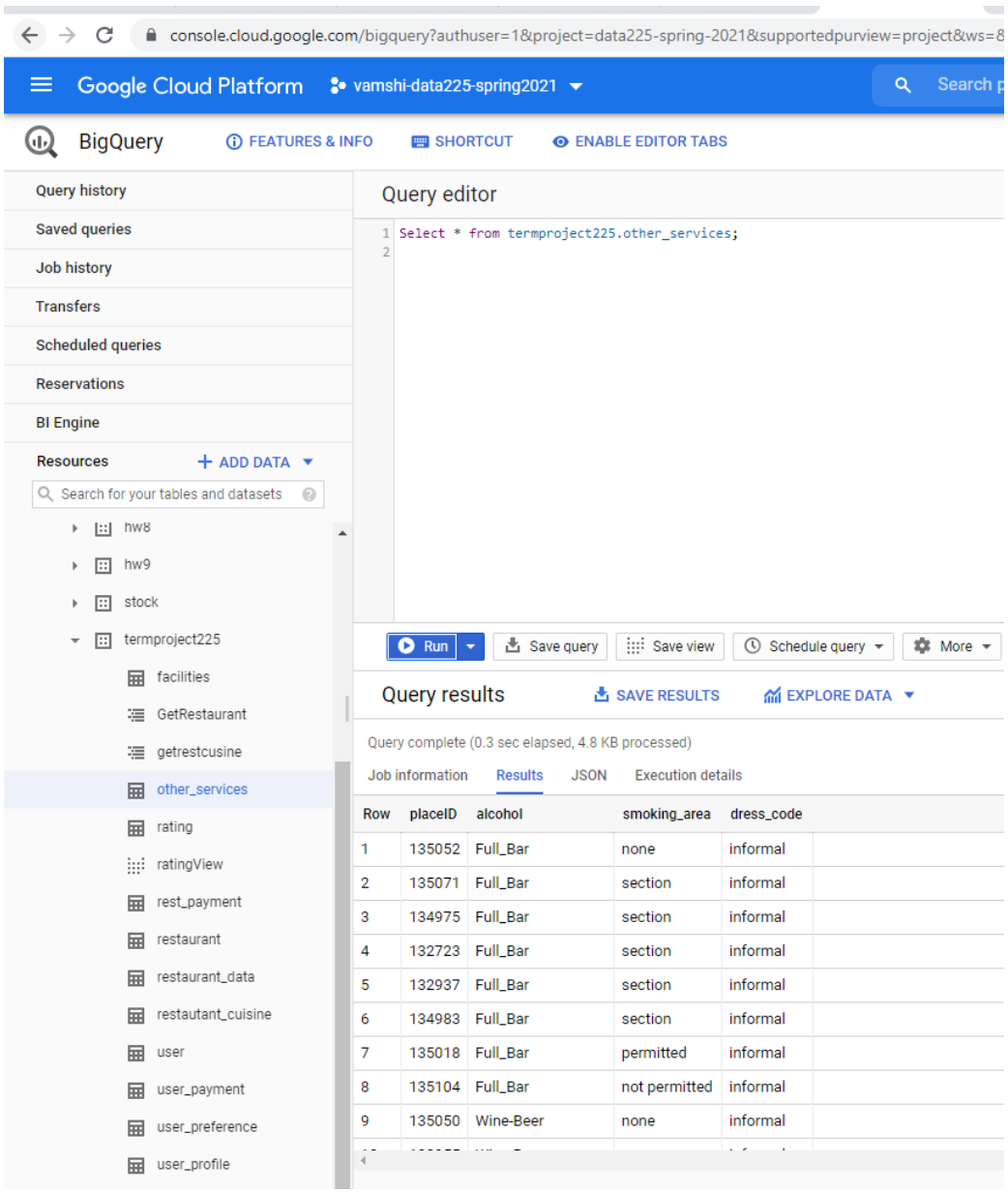
Row	placeID	name	address	city	state	zip	price
1	132608	Hamburguesas La perica	524 Soledad de Graciano Sanchez	Jiutepec	San Luis Potosi	78210	low
2	132608	Hamburguesas La perica	69 Tres De Mayo	s.l.p	SLP	78214	low
3	132608	Hamburguesas La perica	40 Norte Civac 1RA. Seccion	Cuernavaca	san luis potos	78200	low
4	132609	Pollo_Frito_Buenos_Aires	tampico	victoria	Tamaulipas	62790	low
5	132609	Pollo_Frito_Buenos_Aires	69 Tres De Mayo	s.l.p	mexico	62290	low
6	132609	Pollo_Frito_Buenos_Aires	Mexico 810 Centro	victoria	Tamaulipas	78214	low
7	132870	Tortas y hamburguesas el gordo	Ricardo B. Anaya	San Luis Potosi	San Luis Potosi	62790	low
8	132870	Tortas y hamburguesas el gordo	Ricardo B. Anaya	San Luis Potosi	San Luis Potosi	64000	low
9	132870	Tortas y hamburguesas el gordo	Ricardo B. Anaya	San Luis Potosi	San Luis Potosi	74000	low
10	132885	Hamburguesas saul	Tangamanga 7 Tangamanga	san luis potosi	mexico	78433	low
11	132885	Hamburguesas saul	tampico	San Luis Potosi	Tamaulipas	78430	low
12	132885	Hamburguesas saul	Universidad 169	slp	mexico	78740	low
13	132921	crudalia	40 Norte Civac 1RA. Seccion	Cuernavaca	Morelos	62170	low
14	132921	crudalia	524 Soledad de Graciano Sanchez	Jiutepec	S.L.P.	62250	low
15	132921	crudalia	36 Sur Civac	Ciudad Victoria	mexico	62000	low
16	132667	little pizza Emilio Portes Gil	av. seminario	Cd. Victoria	mexico	62790	low
17	132667	little pizza Emilio Portes Gil	avenida hivno nacional	Cuernavaca	tamaulipas	62790	low
18	132667	little pizza Emilio Portes Gil	Av. V. Carranza	Ciudad Victoria	S.L.P.	64000	low

In the above restaurant table the composite primary key (placeID,zip) follows entity integrity rule by showing unique results.

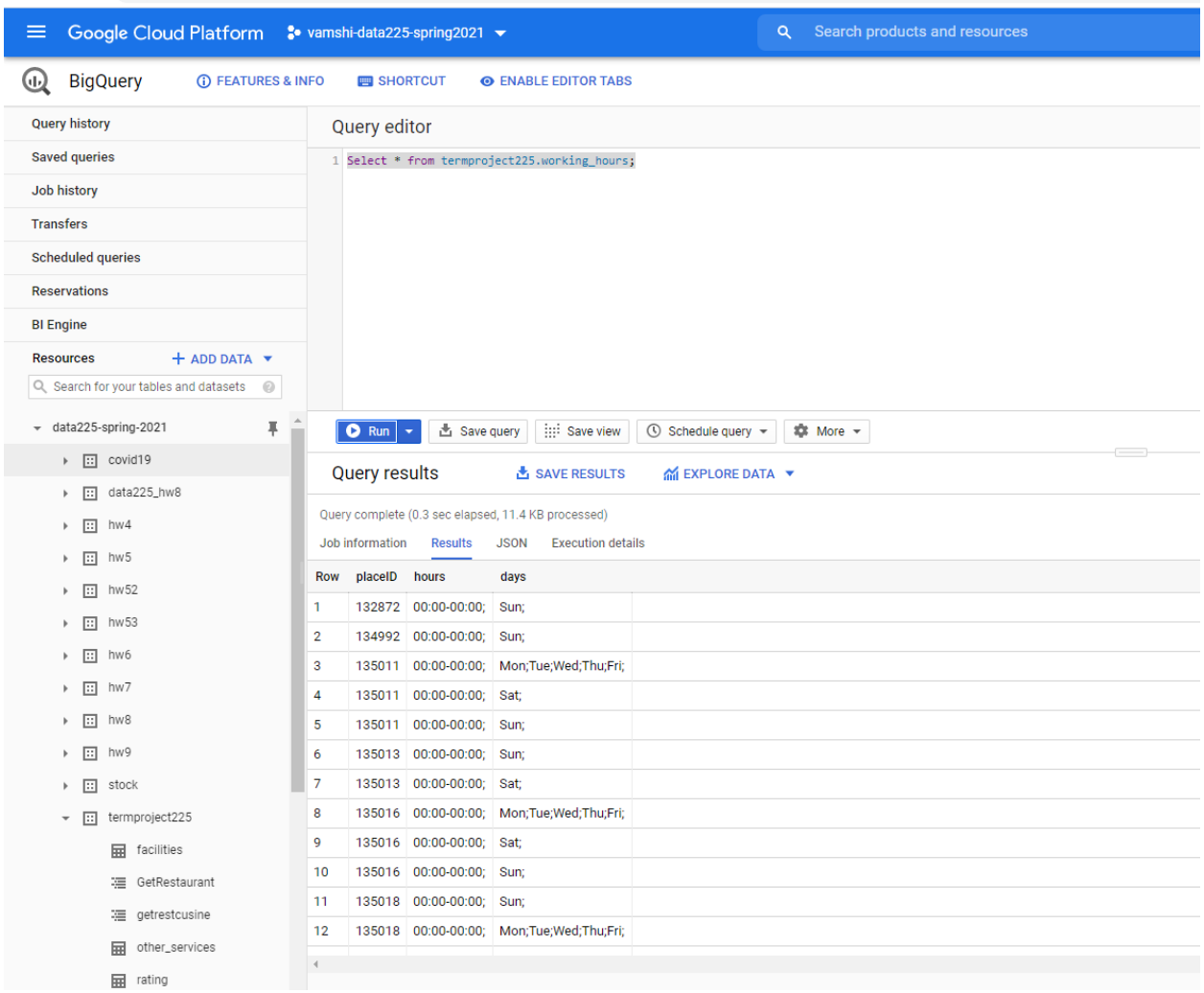
Select * from termproject225.rest_payment;



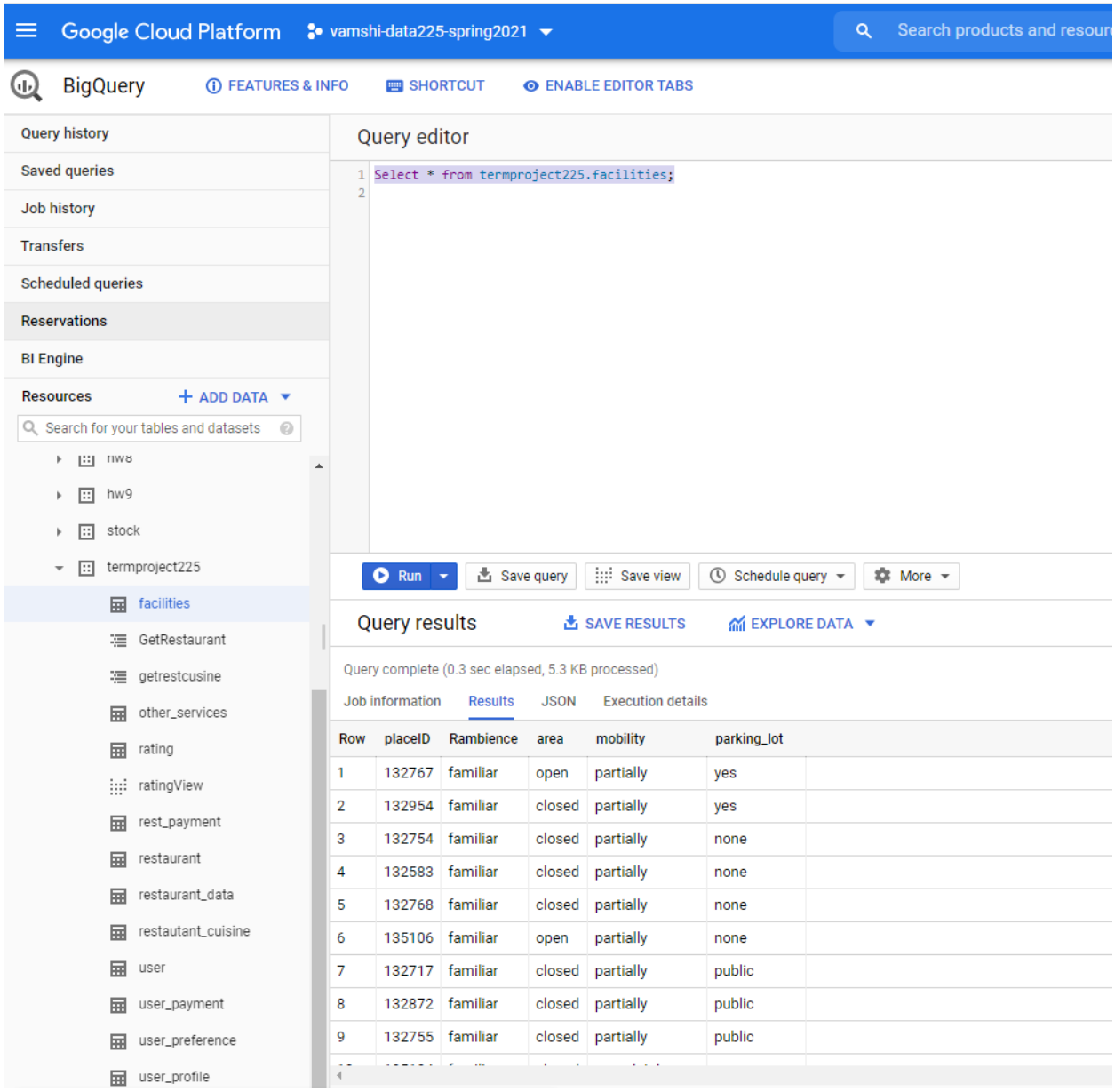
Select * from termproject225.other_services;



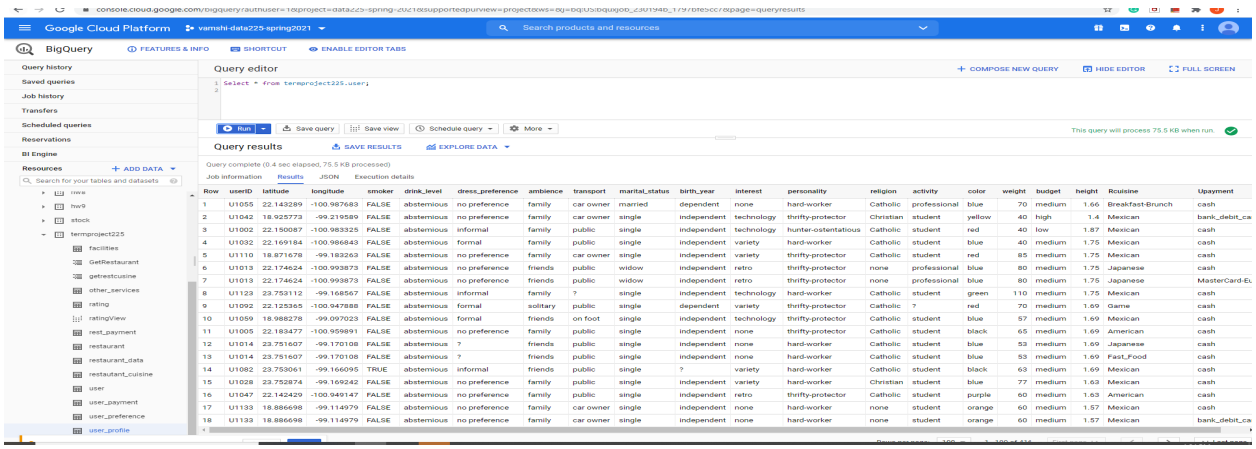
Select * from termproject225.working_hours;



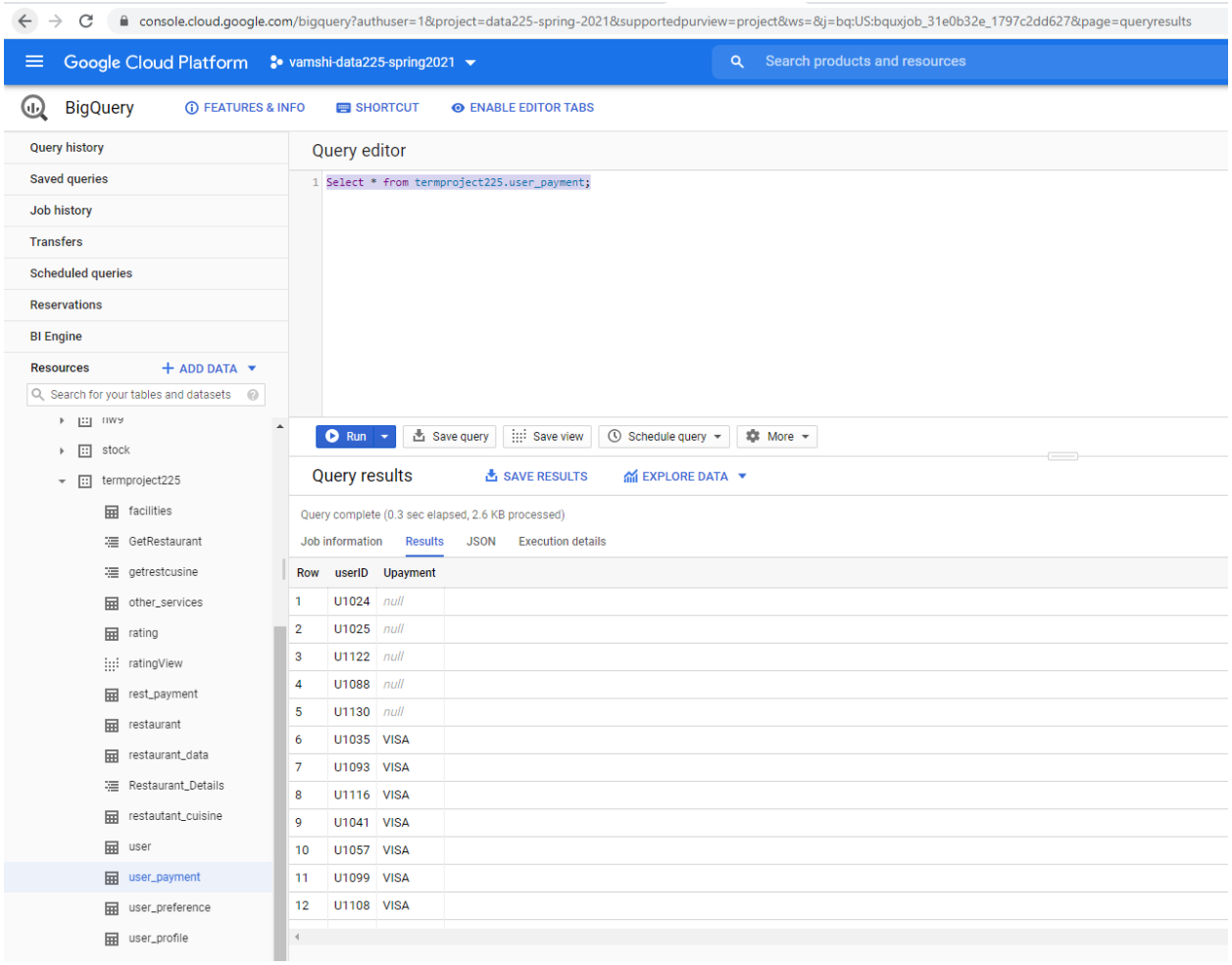
Select * from termproject225.facilities;



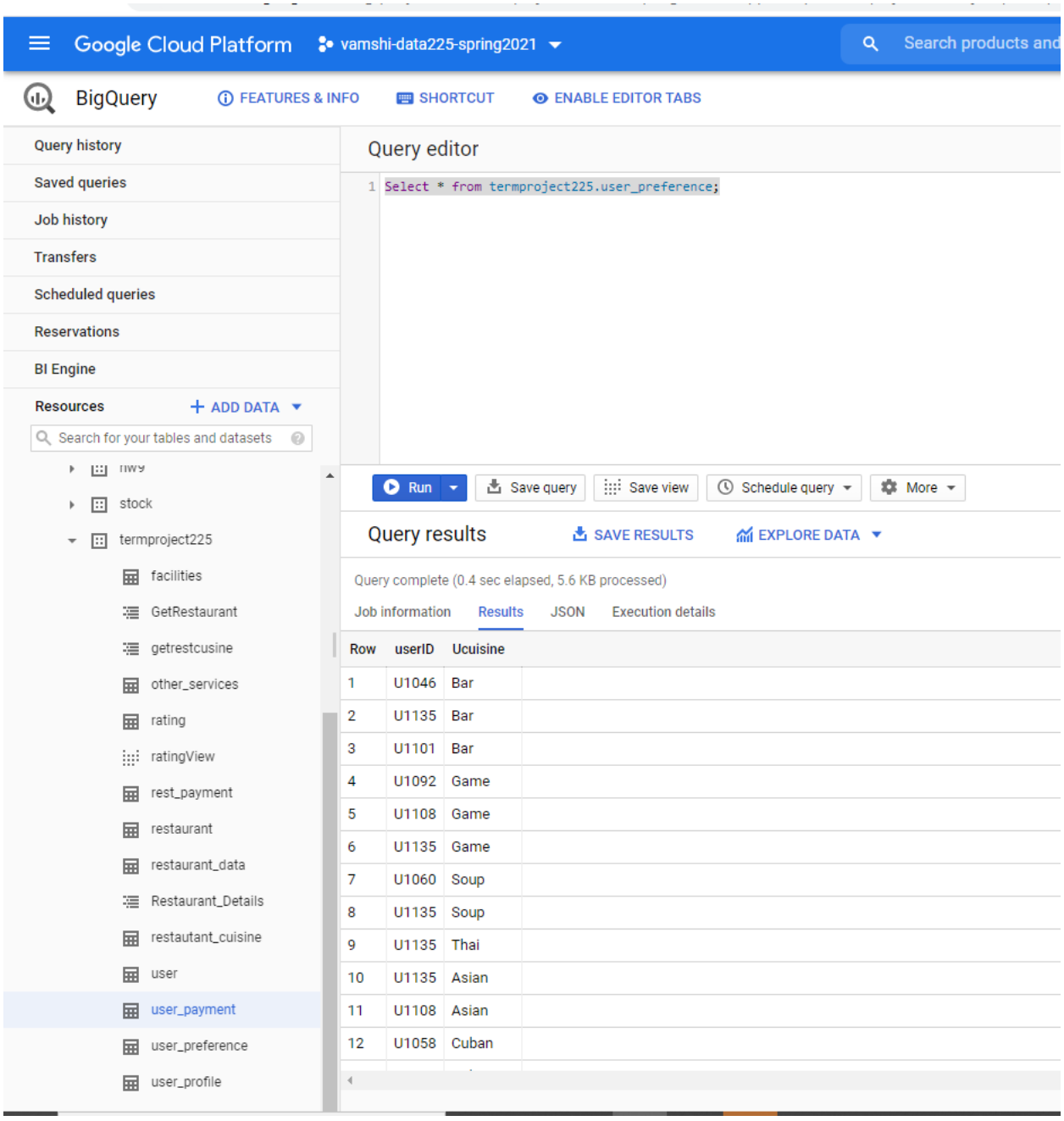
- Select * from termproject225.user;



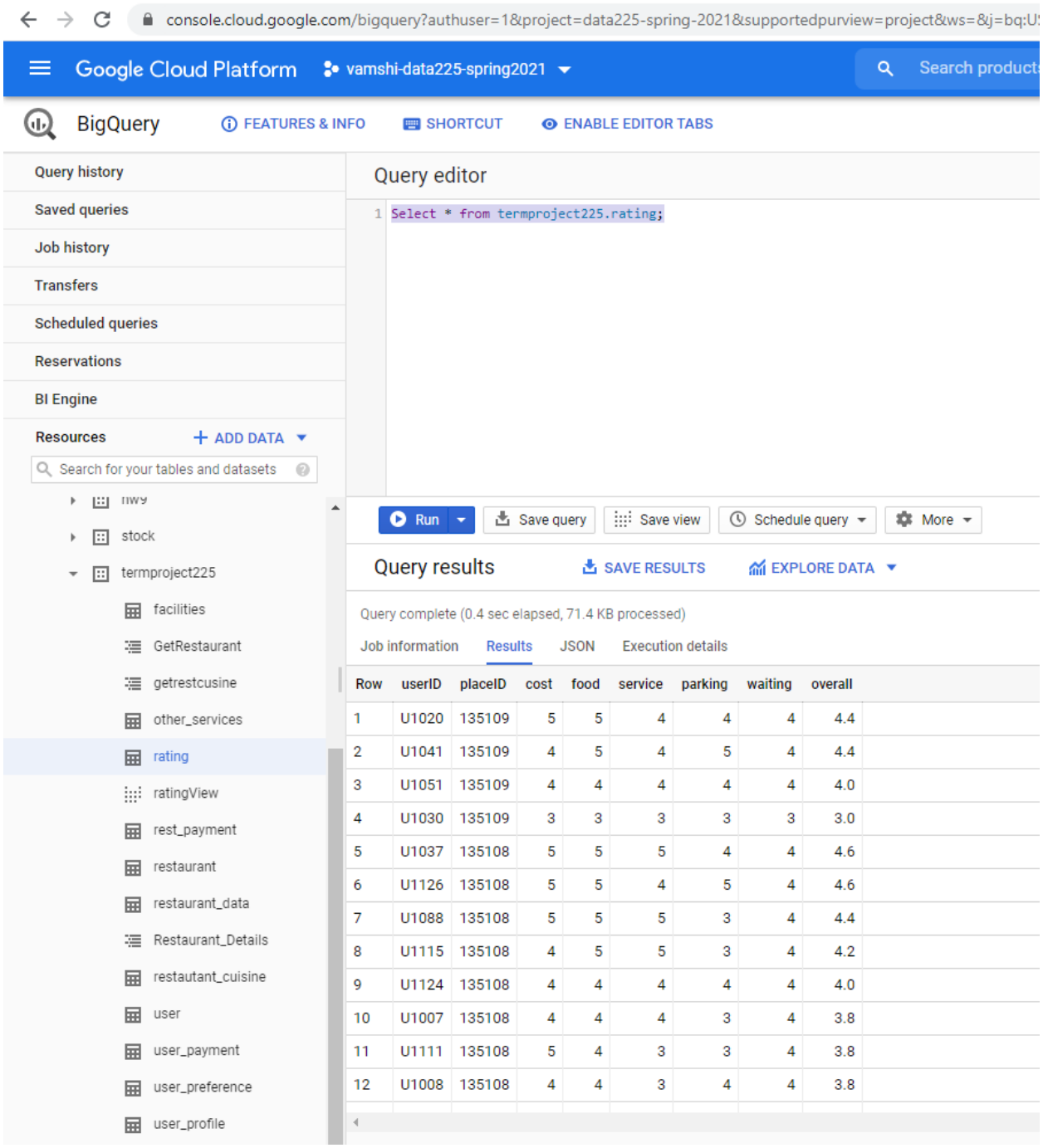
Select * from termproject225.user_payment;



Select * from termproject225.user_preference;



Select * from termproject225.rating;

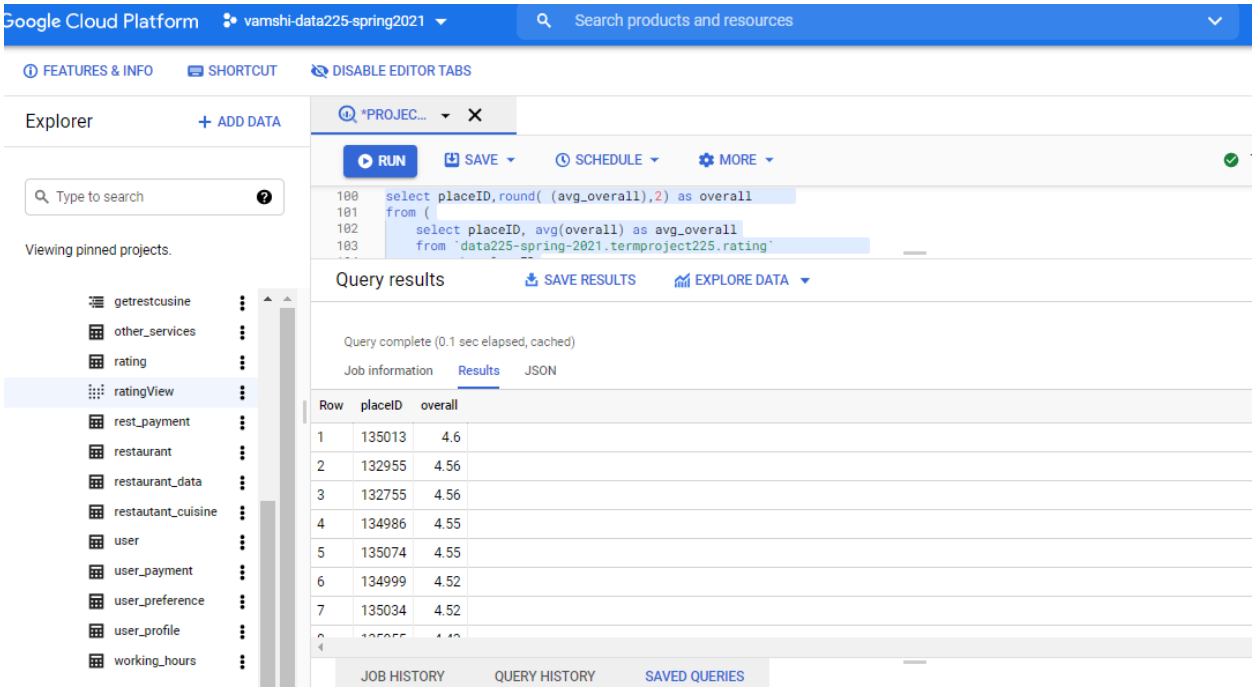


Row	userID	placeID	cost	food	service	parking	waiting	overall
1	U1020	135109	5	5	4	4	4	4.4
2	U1041	135109	4	5	4	5	4	4.4
3	U1051	135109	4	4	4	4	4	4.0
4	U1030	135109	3	3	3	3	3	3.0
5	U1037	135108	5	5	5	4	4	4.6
6	U1126	135108	5	5	4	5	4	4.6
7	U1088	135108	5	5	5	3	4	4.4
8	U1115	135108	4	5	5	3	4	4.2
9	U1124	135108	4	4	4	4	4	4.0
10	U1007	135108	4	4	4	3	4	3.8
11	U1111	135108	5	4	3	3	4	3.8
12	U1008	135108	4	4	3	4	4	3.8

Query Descriptions

 # Top 10 rated restaurants based on customer's overall ratings

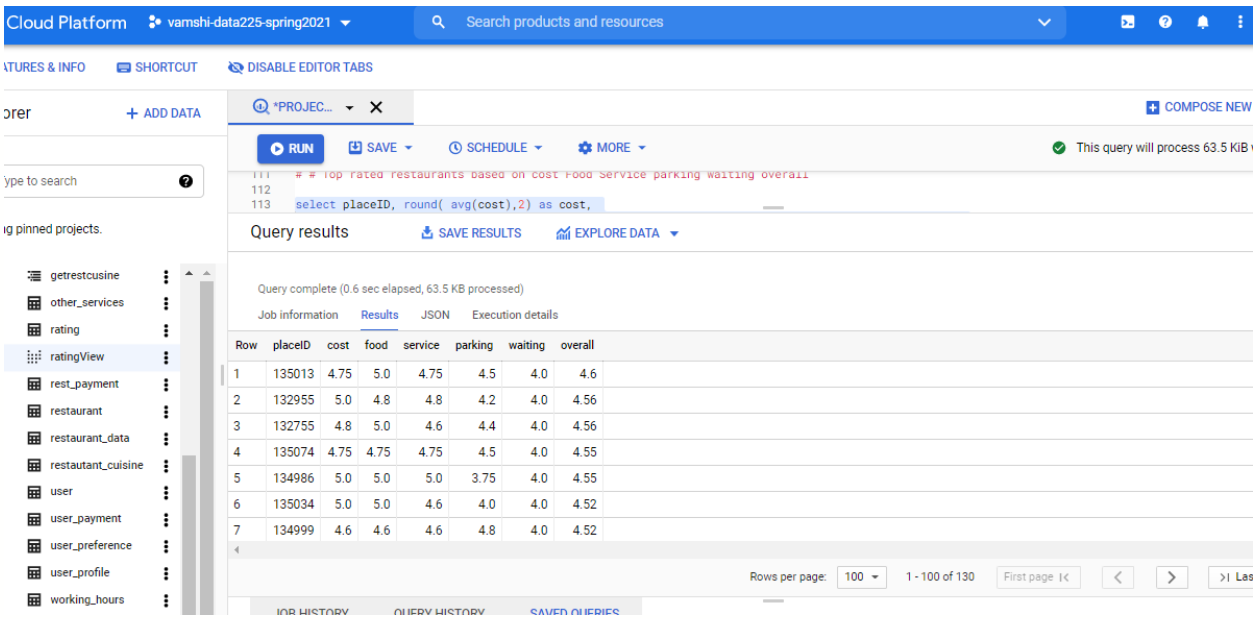
```
select placeID,round( (avg_overall),2) rest_overall
from (
    select placeID, avg(overall) as avg_overall
    from `data225-spring-2021.termproject225.rating`
    group by placeID
    order by avg_overall
)
order by avg_overall desc
limit 10 ;
```



The above results show the top 10 restaurants considering their overall rating. This will help users to find highest rated restaurants.

 # Overall ratings of restaurants based on cost, Food, Service, parking, waiting, overall

```
select placeID, round( avg(cost),2) as cost,
    round(avg(Food),2) as food, round(avg(Service),2) as service,
   round(avg(parking),2) as parking,
    round( avg(waiting),2) as waiting, round( avg(overall),2) as overall
from termproject225.rating
group by placeID
order by overall desc
;
```



Row	placeID	cost	food	service	parking	waiting	overall
1	135013	4.75	5.0	4.75	4.5	4.0	4.6
2	132955	5.0	4.8	4.8	4.2	4.0	4.56
3	132755	4.8	5.0	4.6	4.4	4.0	4.56
4	135074	4.75	4.75	4.75	4.5	4.0	4.55
5	134986	5.0	5.0	5.0	3.75	4.0	4.55
6	135034	5.0	5.0	4.6	4.0	4.0	4.52
7	134999	4.6	4.6	4.6	4.8	4.0	4.52

The above result shows the top rated restaurants and their ratings individually in terms of cost (affordability), food (food quality), services (services provided at restaurant), parking availability, waiting time, and overall is average of all other fields.

 # Creating a View for restaurants based on cost Food Service parking waiting overall ratings

```
CREATE or REPLACE VIEW termproject225.ratingView AS
SELECT *
from (
        select placeID, round( avg(cost),2) as cost,
            round(avg(Food),2) as food, round(avg(Service),2) as service,
   round(avg(parking),2) as parking,
            round( avg(waiting),2) as waiting, round( avg(overall),2) as overall
        from termproject225.rating
        group by placeID
        order by overall desc
);
```

For the above problem we have created a VIEW.

VIEWS

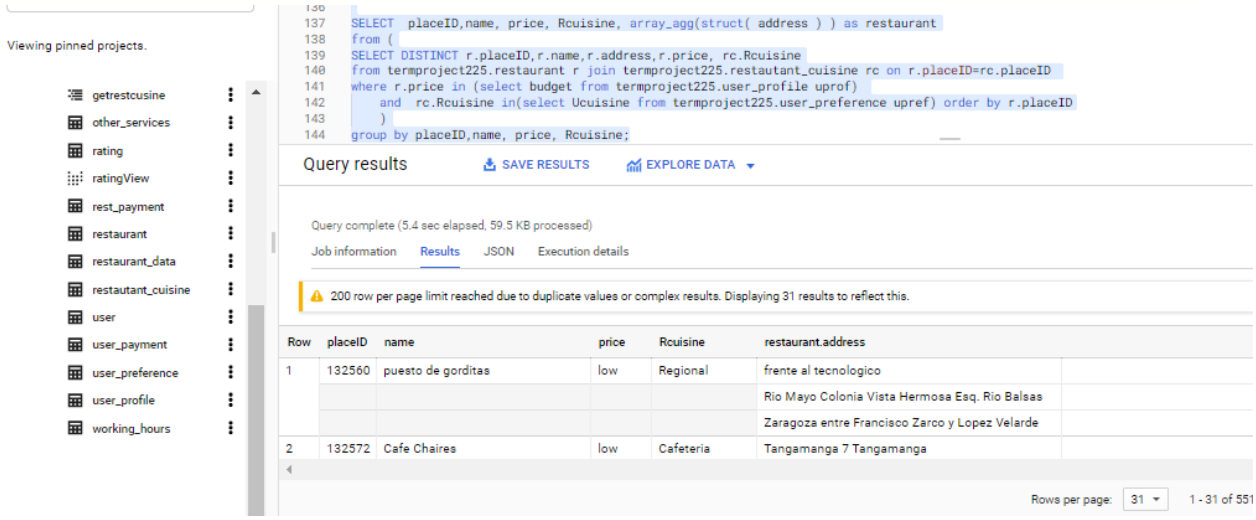
Views**:**

-
- are virtual tables that can be a great way to optimize your database experience. Not only are views good for defining a table without using extra storage, but they also accelerate data analysis and can provide your data extra security.

- Using our view customers can choose top rated restaurants. And Restaurants can monitor their ratings from time to time.

 # Restaurant based on customers cuisine choice (Ucuisine) and budget

```
SELECT  placeID,name, price, Rcuisine, array_agg(struct( address ) ) as restaurant
from (
SELECT DISTINCT r.placeID,r.name,r.address,r.price, rc.Rcuisine
from termproject225.restaurant r join termproject225.restautant_cuisine rc on
   r.placeID=rc.placeID
where r.price in (select budget from termproject225.user_profile uprof)
    and  rc.Rcuisine in(select Ucuisine from termproject225.user_preference upref)
   order by r.placeID
    )
group by placeID,name, price, Rcuisine;
```



From the above results customers can find Restaurant and its address based on their cuisine preference and budget range(low, medium, or high).

 # Restaurant based on address/zip

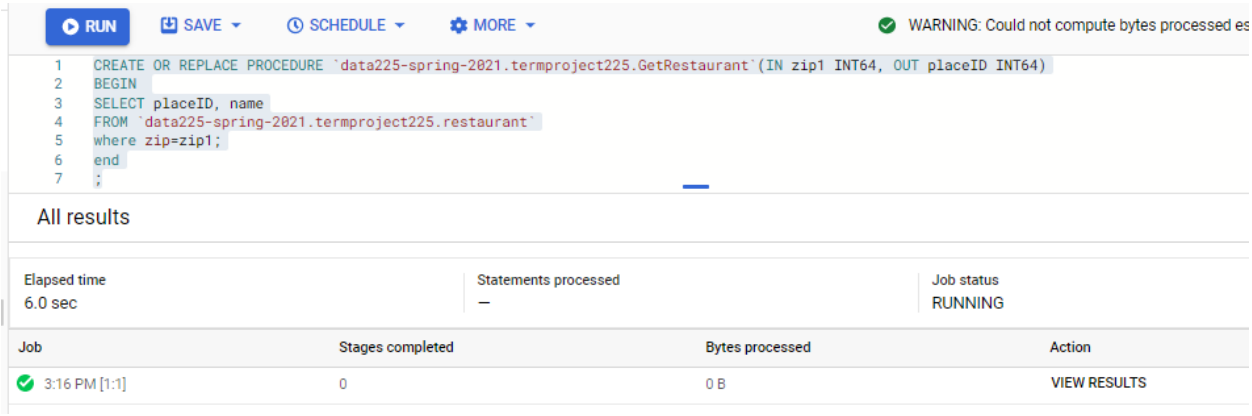
For this we have created a stored procedure. Using this the customer can find the restaurants which are located in that ZIP code by providing the zip code. This is a simple and easy way to call any number of times.

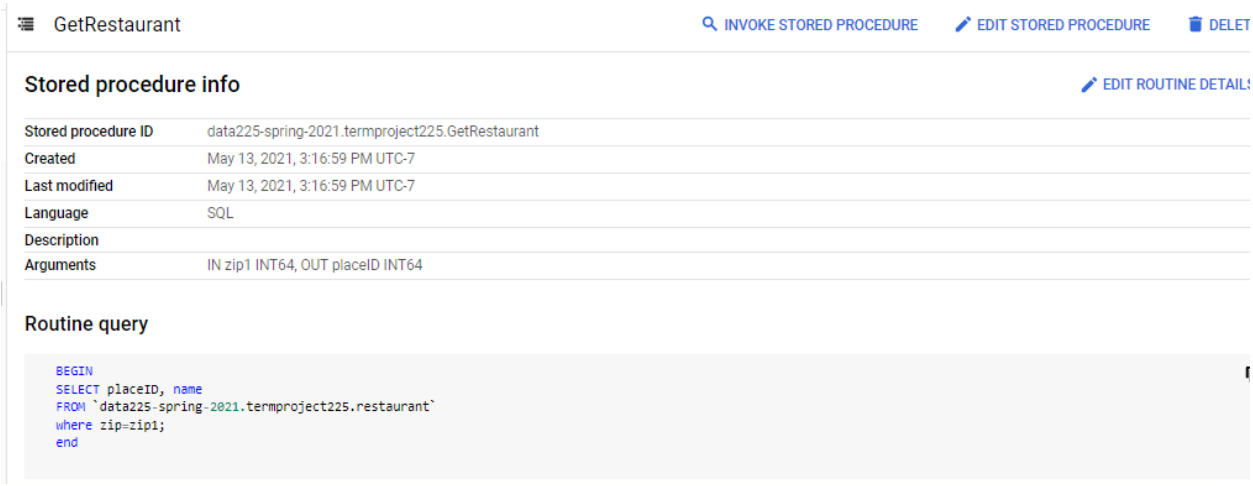
Benefits of stored procedures:

- Reduce the Network Traffic
- Easy to maintain
- Secure
- Create once call any number of times

- When you execute it, instead of sending multiple queries, we are sending only the name and the parameters of the stored procedure.

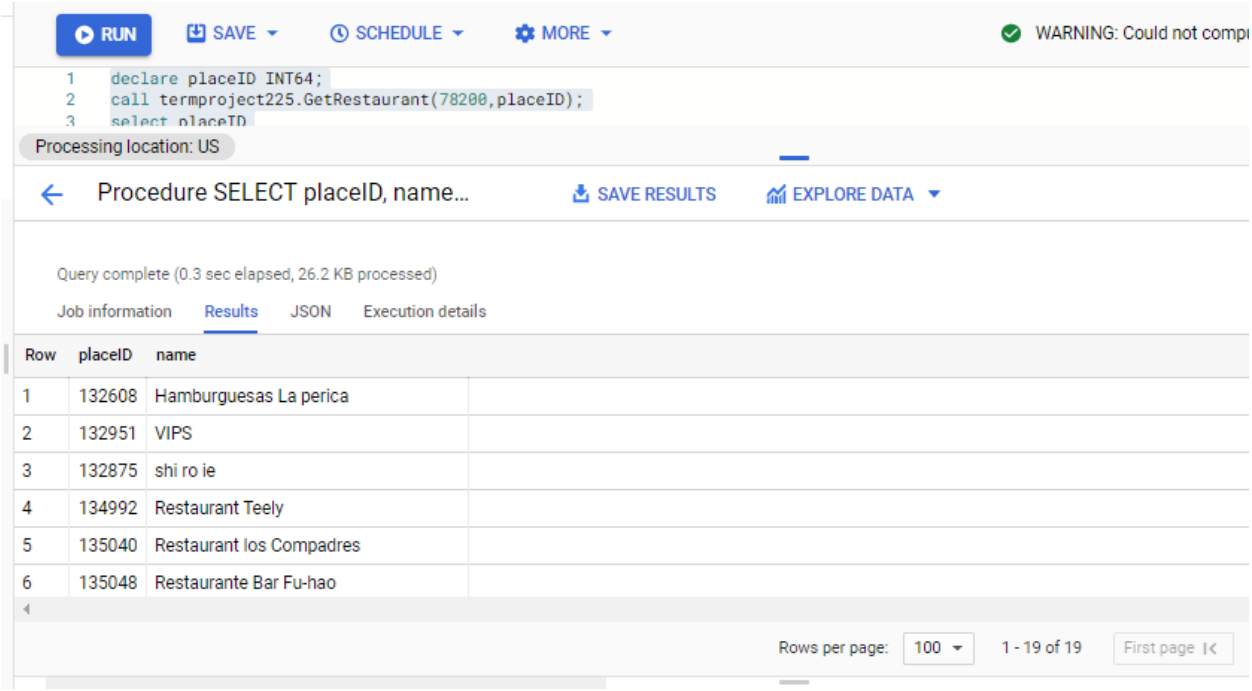
```
CREATE OR REPLACE PROCEDURE `data225-spring-2021.termproject225.GetRestaurant`(IN
   zip1 INT64, OUT placeID INT64)
BEGIN
SELECT placeID, name
FROM `data225-spring-2021.termproject225.restaurant`
where zip=zip1;
end
;
```





Calling the procedure to find the restaurants placed in a zip code area by providing zip code.

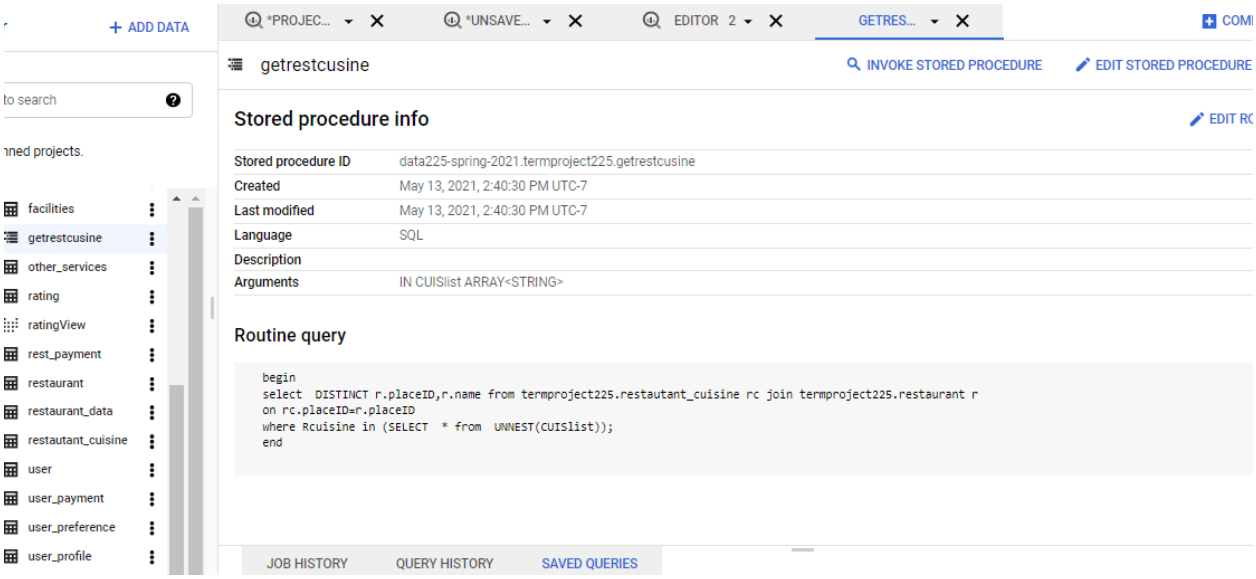
```
declare placeID INT64;
call termproject225.GetRestaurant(78200,placeID);
select placeID
;
```



The above stored procedure result shows the list of restaurants placed in zip code 78200

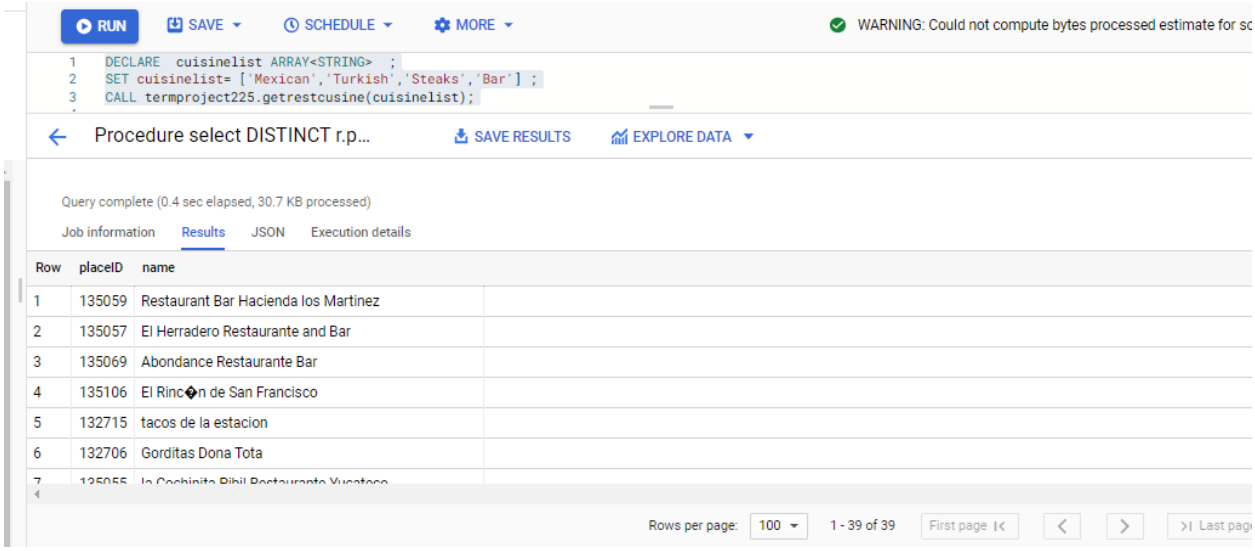
 # Finding restaurants based on combination of cuisines and statement

```
create or replace procedure termproject225.getrestcusine(IN CUISlist ARRAY<STRING>)
begin
select  DISTINCT r.placeID,r.name from termproject225.restautant_cuisine rc join
   termproject225.restaurant r
      on rc.placeID=r.placeID
where Rcuisine in (SELECT  * from  UNNEST(CUISlist));
End;
```



Calling the procedure to give the details of the restaurants which are providing cuisines like Mexican or Turkish or Steaks or Bar.

```
DECLARE  cuisinelist ARRAY<STRING>  ;
SET cuisinelist= ['Mexican','Turkish','Steaks','Bar'] ;
CALL termproject225.getrestcusine(cuisinelist);
```



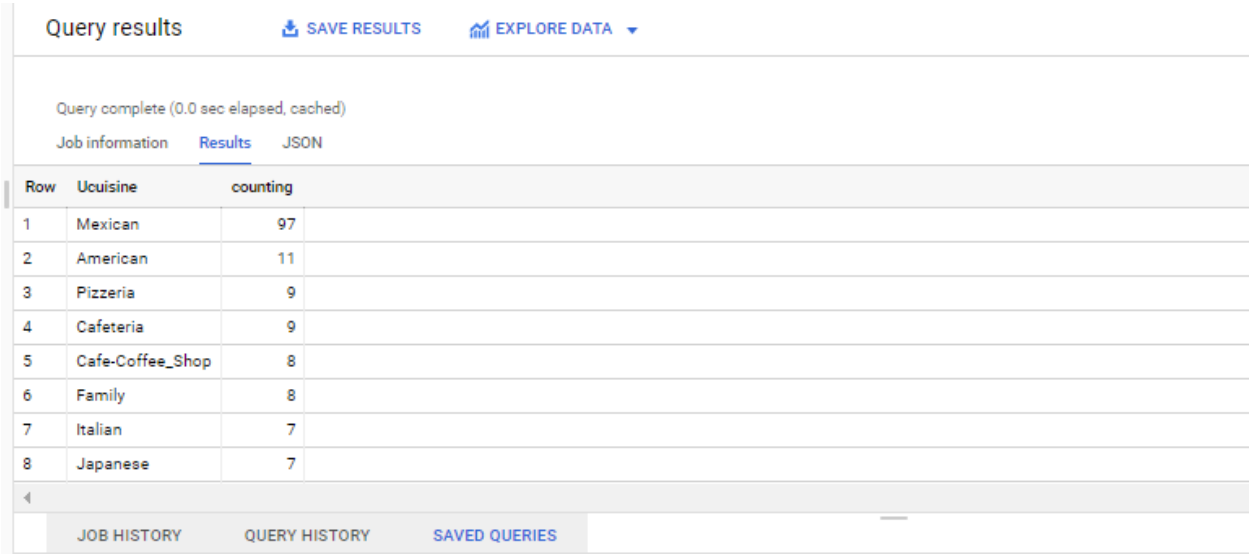
 # Customer's rating style based on avg of overall rating User gave to restaurants.

```
select userID,round( (avg_overall),2) as user_overall
from (
    select userID, avg(overall) as avg_overall
    from `data225-spring-2021.termproject225.rating`
    group by userID
    order by avg_overall
)
order by avg_overall asc
limit 10 ;
```

From the above results we can find each customer's rating style, as few customers are giving low ratings to all restaurants irrespective of restaurants food quality, services, and facilities. If some customers give low ratings to all restaurants and that is completely unmatchable with the rest of others' ratings in such situations their ratings can be ignored for better understanding the restaurant.

 # Top 10 trending cuisine among customers with count of users

```
select Ucuisine, count(Ucuisine) as counting
from `data225-spring-2021.termproject225.user_preference`
group by Ucuisine
order by counting desc
limit 10 ;
```



Row	Ucuisine	counting
1	Mexican	97
2	American	11
3	Pizzeria	9
4	Cafeteria	9
5	Cafe-Coffee_Shop	8
6	Family	8
7	Italian	7
8	Japanese	7

The above result gives the 10 most preferred cuisines in Mexico among the customers.

 # Most preferred mode of payment by user vs Rpayment

```
select RR.Rpayment max_Rpayment, RR.count_Rpayment , max(Upayment) as max_Upayment,
   count(Upayment) as count_Upayment
from`data225-spring-2021.termproject225.user_payment` U,
    (select Rpayment,count(Rpayment) as count_Rpayment
    from `data225-spring-2021.termproject225.rest_payment`
    group by Rpayment
    order by count_Rpayment desc
    limit 1
    ) as RR
group by RR.Rpayment,RR.count_Rpayment, Upayment
order by RR.count_Rpayment,count_Upayment desc
limit 1
;
```

```
select RR.Rpayment max_Rpayment, RR.count_Rpayment , max(Upayment) as max_Upayment, count(Upaym
from`data225-spring-2021.termproject225.user_payment` U,
    (select Rpayment,count(Rpayment) as count_Rpayment
    from `data225-spring-2021.termproject225.rest_payment`
    group by Rpayment
    order by count_Rpayment desc
    limit 1
    ) as RR
group by RR.Rpayment,RR.count_Rpayment, Upayment
order by RR.count_Rpayment,count_Upayment desc
limit 1
```

Query results SAVE RESULTS EXPLORE DATA

Query complete (0.7 sec elapsed, 4 KB processed)

Job information | Results | JSON | Execution details

Row	max_Rpayment	count_Rpayment	max_Upayment	count_Upayment
1	cash	113	cash	131

The above results shows the most preferred mode of payment for users is cash which is available at 131 customers and for restaurants also cash is the most preferable mode of option. 113 Restaurants are providing cash payment options to their customers.

Summary & Conclusion

Customer preferences towards restaurant cuisines are increasing everyday, and they are now more demanding in choosing better restaurant choices based on what they can get from their decision. This decision making can help restaurateurs understand restaurant customer perception of key factors when selecting a restaurant, but also form appropriate marketing strategies to attract existing and potential customers and outperform competitors. Faced with the complex phenomenon of eating-out, our study extends the body of knowledge on the relative importance of restaurant selection criteria. Our analysis into customers' importance of restaurant selection and how they vary across situational factors, namely dining ambience,payment methods and restaurant segments, presents empirical evidence regarding customers choice of restaurant. Our study has three important findings. First, the customer's preference of cuisine becomes the most important criteria when customers choose a restaurant to eat-out. Secondly, the facilities that the restaurant would provide i.e ambience of the restaurant, payment method accepted by the restaurant, if the restaurant serves alcohol or has a smoking area, etc.Third,ratings given to the restaurant that will help them and the customers. This demonstrates that there are many restaurant alternatives with similar cuisines or facilities in the city.Therefore, here the decision is then made on the ratings that is provided to the restaurants by different customers. It is concluded that customers' perceived importance of restaurant cuisine and ratings are important considerations for choice of restaurant.

Future Work

There are several curbs to this study that we need to address for future work. First, we have data collected for only one country Mexico, hence limiting the generalizability of the inferences. Other country restaurants can be studied to obtain comparative results. Second, investigating more on the budget ranges to classify the restaurants into low, medium and high categories. Thirdly, add the reviews section for the customers which would also impact the ratings and ranking of the restaurants.Fourth, provide guest access for some of the customers.

Finally,collecting all of the data and putting it together in the form of a mobile application or a website that can be more accessible to the people. Since we are using Bigquery for our application,with the increase in number of records or data we can accomodate by adding credits to the billing account.