3309 Assignment 3 Report

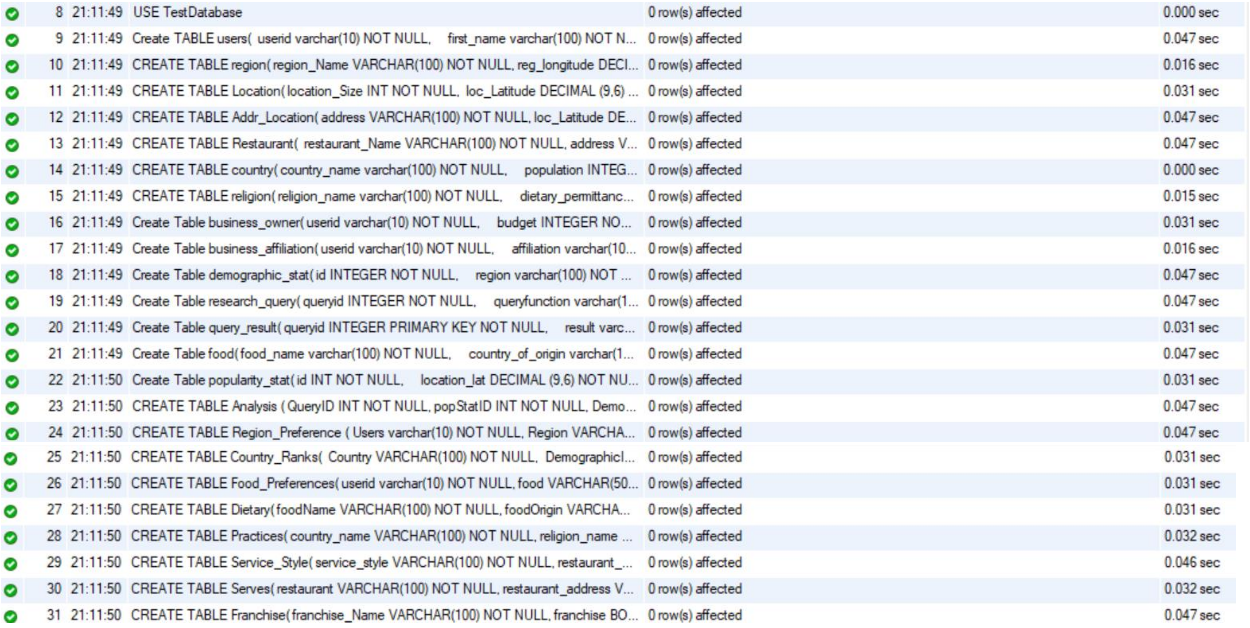
Changes from Assignment 2

Here is a summary of all the changes made from the assignment 2:

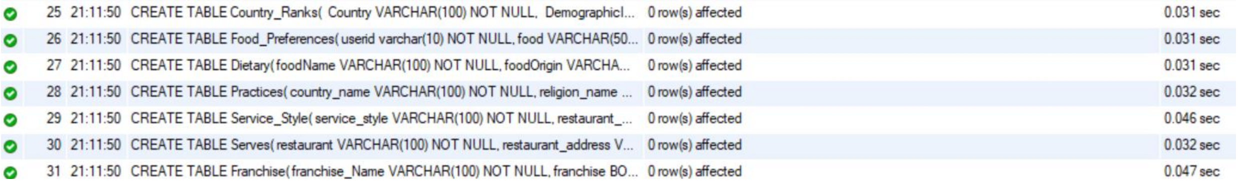
- User relation renamed to users since user is a reserved keyword in SQL.
- Query relation renamed to research_query since query is a reserved keyword.
- We changed the naming convention from wordWord to word_word.

Table Creation and Descriptions

Editor response to the creation of each table:



```
 8  21:11:49  USE TestDatabase                                                                  0 row(s) affected   0.000 sec
 9  21:11:49  Create TABLE users(  userid varchar(10) NOT NULL,    first_name varchar(100) NOT N...  0 row(s) affected   0.047 sec
10  21:11:49  CREATE TABLE region( region_Name VARCHAR(100) NOT NULL, reg_longitude DECI...         0 row(s) affected   0.016 sec
11  21:11:49  CREATE TABLE Location( location_Size INT NOT NULL,  loc_Latitude DECIMAL (9,6) ...      0 row(s) affected   0.031 sec
12  21:11:49  CREATE TABLE Addr_Location( address VARCHAR(100) NOT NULL, loc_Latitude DE...         0 row(s) affected   0.047 sec
13  21:11:49  CREATE TABLE Restaurant(  restaurant_Name VARCHAR(100) NOT NULL, address V...         0 row(s) affected   0.047 sec
14  21:11:49  CREATE TABLE country( country_name varchar(100) NOT NULL,    population INTEG...      0 row(s) affected   0.000 sec
15  21:11:49  CREATE TABLE religion( religion_name varchar(100) NOT NULL,    dietary_permittanc...  0 row(s) affected   0.015 sec
16  21:11:49  Create Table business_owner( userid varchar(10) NOT NULL,    budget INTEGER NO...     0 row(s) affected   0.031 sec
17  21:11:49  Create Table business_affiliation( userid varchar(10) NOT NULL,    affiliation varchar(10...  0 row(s) affected   0.016 sec
18  21:11:49  Create Table demographic_stat( id INTEGER NOT NULL,    region varchar(100) NOT ...    0 row(s) affected   0.047 sec
19  21:11:49  Create Table research_query( queryid INTEGER NOT NULL,    queryfunction varchar(1...  0 row(s) affected   0.047 sec
20  21:11:49  Create Table query_result( queryid INTEGER PRIMARY KEY NOT NULL,    result varc...   0 row(s) affected   0.031 sec
21  21:11:49  Create Table food( food_name varchar(100) NOT NULL,    country_of_origin varchar(1...  0 row(s) affected   0.047 sec
22  21:11:50  Create Table popularity_stat( id INT NOT NULL,    location_lat DECIMAL (9,6) NOT NU...  0 row(s) affected   0.031 sec
23  21:11:50  CREATE TABLE Analysis ( QueryID INT NOT NULL, popStatID INT NOT NULL, Demo...        0 row(s) affected   0.047 sec
24  21:11:50  CREATE TABLE Region_Preference ( Users varchar(10) NOT NULL, Region VARCHA...       0 row(s) affected   0.047 sec
25  21:11:50  CREATE TABLE Country_Ranks(  Country VARCHAR(100) NOT NULL,  DemographicI...        0 row(s) affected   0.031 sec
26  21:11:50  CREATE TABLE Food_Preferences( userid varchar(10) NOT NULL, food VARCHAR(50...      0 row(s) affected   0.031 sec
27  21:11:50  CREATE TABLE Dietary( foodName VARCHAR(100) NOT NULL, foodOrigin VARCHA...         0 row(s) affected   0.031 sec
28  21:11:50  CREATE TABLE Practices( country_name VARCHAR(100) NOT NULL, religion_name ...       0 row(s) affected   0.032 sec
29  21:11:50  CREATE TABLE Service_Style( service_style VARCHAR(100) NOT NULL, restaurant_...    0 row(s) affected   0.046 sec
30  21:11:50  CREATE TABLE Serves( restaurant VARCHAR(100) NOT NULL, restaurant_address V...     0 row(s) affected   0.032 sec
31  21:11:50  CREATE TABLE Franchise( franchise_Name VARCHAR(100) NOT NULL, franchise BO...      0 row(s) affected   0.047 sec
```

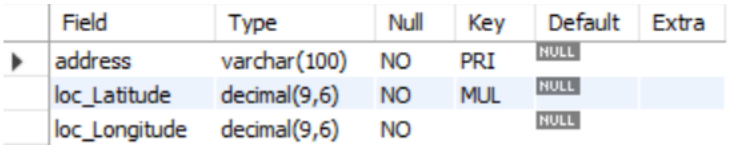


```
25  21:11:50  CREATE TABLE Country_Ranks(  Country VARCHAR(100) NOT NULL,  Demographicl...  0 row(s) affected  0.031 sec
26  21:11:50  CREATE TABLE Food_Preferences( userid varchar(10) NOT NULL, food VARCHAR(50...  0 row(s) affected  0.031 sec
27  21:11:50  CREATE TABLE Dietary( foodName VARCHAR(100) NOT NULL, foodOrigin VARCHA...  0 row(s) affected  0.031 sec
28  21:11:50  CREATE TABLE Practices( country_name VARCHAR(100) NOT NULL, religion_name ...  0 row(s) affected  0.032 sec
29  21:11:50  CREATE TABLE Service_Style( service_style VARCHAR(100) NOT NULL, restaurant_...  0 row(s) affected  0.046 sec
30  21:11:50  CREATE TABLE Serves( restaurant VARCHAR(100) NOT NULL, restaurant_address V...  0 row(s) affected  0.032 sec
31  21:11:50  CREATE TABLE Franchise( franchise_Name VARCHAR(100) NOT NULL, franchise BO...  0 row(s) affected  0.047 sec
```

Describe Queries for Every Relation:

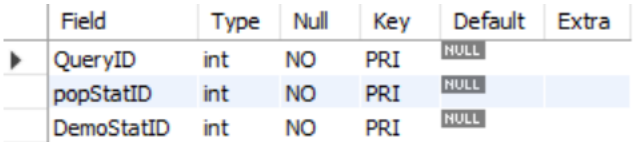
In alphabetical order:

addr_locaiton:



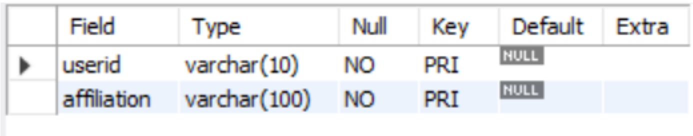
Field	Type	Null	Key	Default	Extra
address	varchar(100)	NO	PRI	NULL	
loc_Latitude	decimal(9,6)	NO	MUL	NULL	
loc_Longitude	decimal(9,6)	NO		NULL	

analysis:



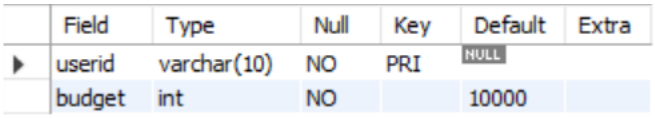
Field	Type	Null	Key	Default	Extra
QueryID	int	NO	PRI	NULL	
popStatID	int	NO	PRI	NULL	
DemoStatID	int	NO	PRI	NULL	

business_affiliation:



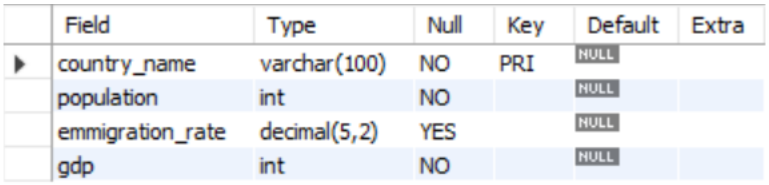
Field	Type	Null	Key	Default	Extra
userid	varchar(10)	NO	PRI	NULL	
affiliation	varchar(100)	NO	PRI	NULL	

business_owner:



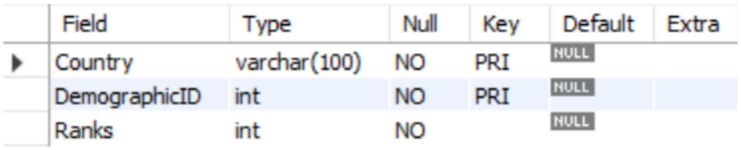
Field	Type	Null	Key	Default	Extra
userid	varchar(10)	NO	PRI	NULL	
budget	int	NO		10000	

country:



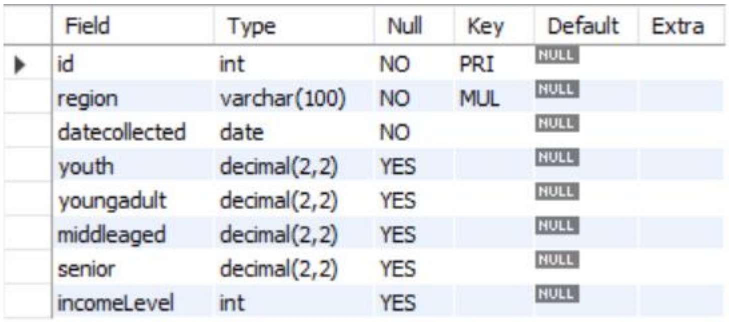
Field	Type	Null	Key	Default	Extra
country_name	varchar(100)	NO	PRI	NULL	
population	int	NO		NULL	
emmigration_rate	decimal(5,2)	YES		NULL	
gdp	int	NO		NULL	

country_rank:



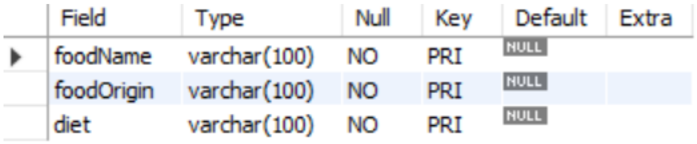
Field	Type	Null	Key	Default	Extra
Country	varchar(100)	NO	PRI	NULL	
DemographicID	int	NO	PRI	NULL	
Ranks	int	NO		NULL	

demographic_stat:



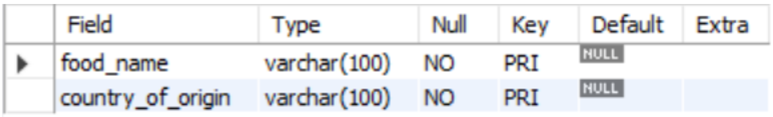
	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	
	region	varchar(100)	NO	MUL	NULL	
	datecollected	date	NO		NULL	
	youth	decimal(2,2)	YES		NULL	
	youngadult	decimal(2,2)	YES		NULL	
	middleaged	decimal(2,2)	YES		NULL	
	senior	decimal(2,2)	YES		NULL	
	incomeLevel	int	YES		NULL	

dietary:



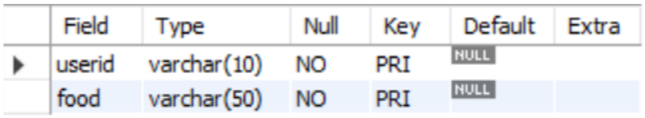
	Field	Type	Null	Key	Default	Extra
▶	foodName	varchar(100)	NO	PRI	NULL	
	foodOrigin	varchar(100)	NO	PRI	NULL	
	diet	varchar(100)	NO	PRI	NULL	

food:



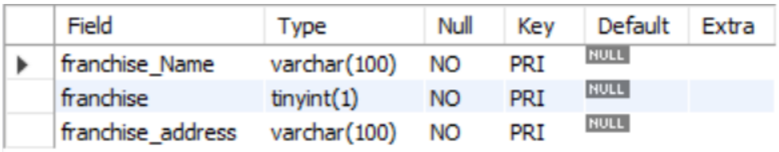
	Field	Type	Null	Key	Default	Extra
▶	food_name	varchar(100)	NO	PRI	NULL	
	country_of_origin	varchar(100)	NO	PRI	NULL	

food_preference:



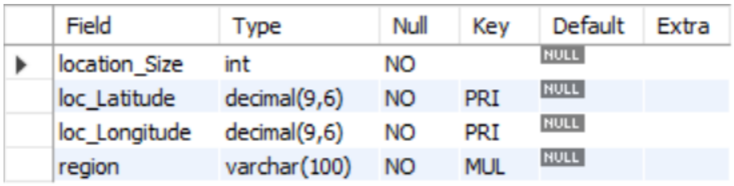
	Field	Type	Null	Key	Default	Extra
▶	userid	varchar(10)	NO	PRI	NULL	
	food	varchar(50)	NO	PRI	NULL	

franchise:



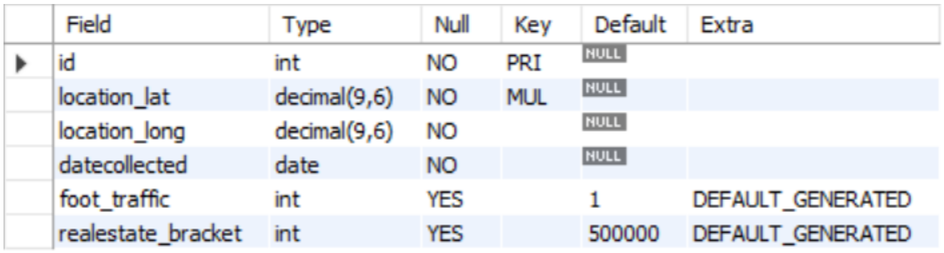
	Field	Type	Null	Key	Default	Extra
▶	franchise_Name	varchar(100)	NO	PRI	NULL	
	franchise	tinyint(1)	NO	PRI	NULL	
	franchise_address	varchar(100)	NO	PRI	NULL	

location:



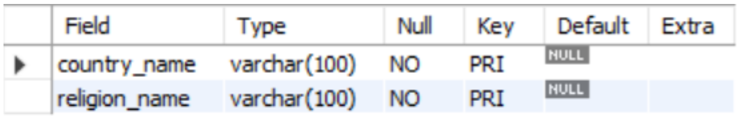
	Field	Type	Null	Key	Default	Extra
▶	location_Size	int	NO		NULL	
	loc_Latitude	decimal(9,6)	NO	PRI	NULL	
	loc_Longitude	decimal(9,6)	NO	PRI	NULL	
	region	varchar(100)	NO	MUL	NULL	

popularity_stat:



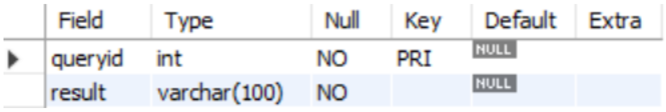
	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	
	location_lat	decimal(9,6)	NO	MUL	NULL	
	location_long	decimal(9,6)	NO		NULL	
	datecollected	date	NO		NULL	
	foot_traffic	int	YES		1	DEFAULT_GENERATED
	realestate_bracket	int	YES		500000	DEFAULT_GENERATED

practices:



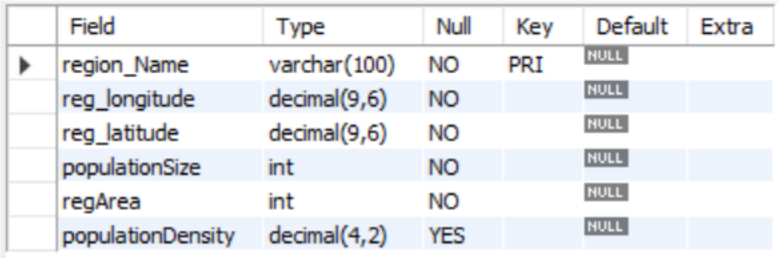
	Field	Type	Null	Key	Default	Extra
▶	country_name	varchar(100)	NO	PRI	NULL	
	religion_name	varchar(100)	NO	PRI	NULL	

query_results:



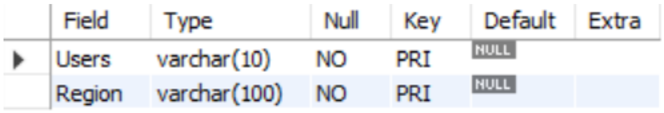
	Field	Type	Null	Key	Default	Extra
▶	queryid	int	NO	PRI	NULL	
	result	varchar(100)	NO		NULL	

region:



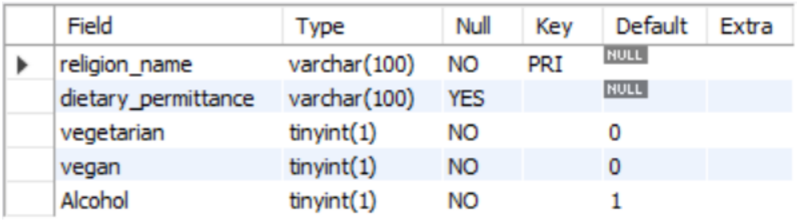
	Field	Type	Null	Key	Default	Extra
▶	region_Name	varchar(100)	NO	PRI	NULL	
	reg_longitude	decimal(9,6)	NO		NULL	
	reg_latitude	decimal(9,6)	NO		NULL	
	populationSize	int	NO		NULL	
	regArea	int	NO		NULL	
	populationDensity	decimal(4,2)	YES		NULL	

region_preferences:



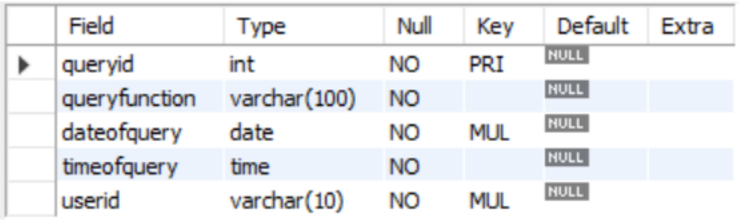
	Field	Type	Null	Key	Default	Extra
▶	Users	varchar(10)	NO	PRI	NULL	
	Region	varchar(100)	NO	PRI	NULL	

religion:



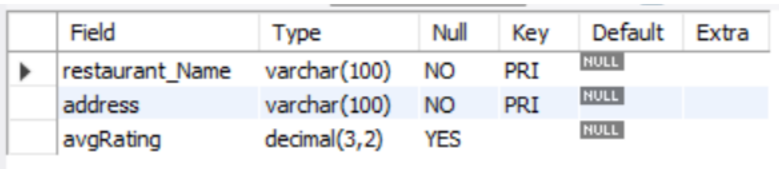
	Field	Type	Null	Key	Default	Extra
▶	religion_name	varchar(100)	NO	PRI	NULL	
	dietary_permittance	varchar(100)	YES		NULL	
	vegetarian	tinyint(1)	NO		0	
	vegan	tinyint(1)	NO		0	
	Alcohol	tinyint(1)	NO		1	

research_query:



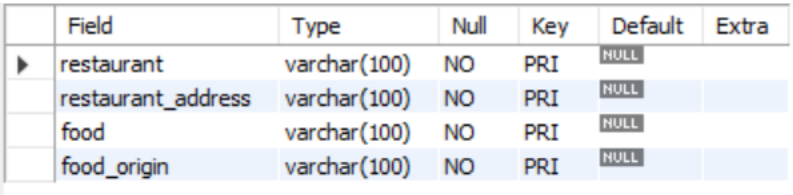
	Field	Type	Null	Key	Default	Extra
▶	queryid	int	NO	PRI	NULL	
	queryfunction	varchar(100)	NO		NULL	
	dateofquery	date	NO	MUL	NULL	
	timeofquery	time	NO		NULL	
	userid	varchar(10)	NO	MUL	NULL	

restaurant:



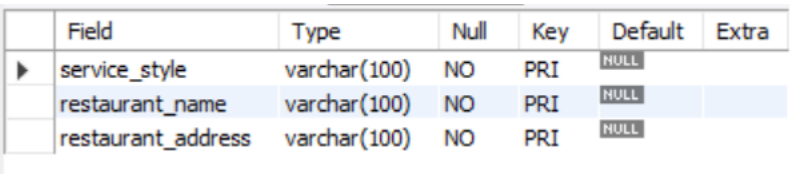
	Field	Type	Null	Key	Default	Extra
▶	restaurant_Name	varchar(100)	NO	PRI	NULL	
	address	varchar(100)	NO	PRI	NULL	
	avgRating	decimal(3,2)	YES		NULL	

serves:



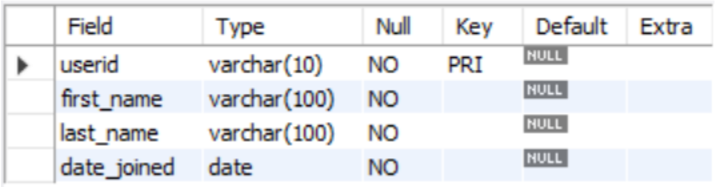
	Field	Type	Null	Key	Default	Extra
▶	restaurant	varchar(100)	NO	PRI	NULL	
	restaurant_address	varchar(100)	NO	PRI	NULL	
	food	varchar(100)	NO	PRI	NULL	
	food_origin	varchar(100)	NO	PRI	NULL	

service_style:



	Field	Type	Null	Key	Default	Extra
▶	service_style	varchar(100)	NO	PRI	NULL	
	restaurant_name	varchar(100)	NO	PRI	NULL	
	restaurant_address	varchar(100)	NO	PRI	NULL	

users:



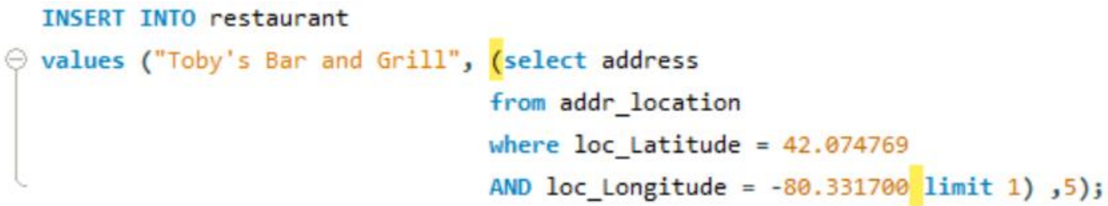
	Field	Type	Null	Key	Default	Extra
▶	userid	varchar(10)	NO	PRI	NULL	
	first_name	varchar(100)	NO		NULL	
	last_name	varchar(100)	NO		NULL	
	date_joined	date	NO		NULL	

Insert Statements

Insert 1:

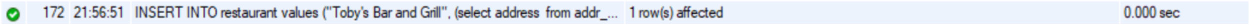
Create a 5-star restaurant where the address is the first address alphabetically in a chosen location:

Code:



```
INSERT INTO restaurant
values ("Toby's Bar and Grill", (select address
                                 from addr_location
                                 where loc_Latitude = 42.074769
                                 AND loc_Longitude = -80.331700 limit 1) ,5);
```

Output:

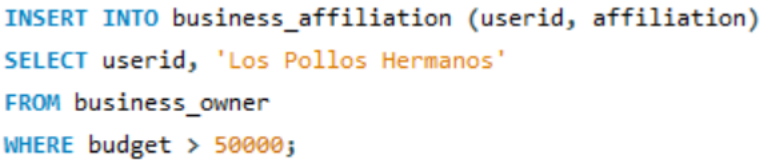


```
172  21:56:51  INSERT INTO restaurant values ("Toby's Bar and Grill", (select address  from addr_...  1 row(s) affected  0.000 sec
```

Insert 2:

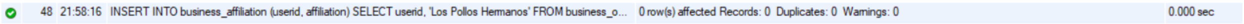
The insert of "Los Pollos Hermanos" is allocated via a SELECT statement such that all business owners with a budget of over $50,000. This was added as a gimmick to represent the possibility of co ownership via stock.

Code:



```
INSERT INTO business_affiliation (userid, affiliation)
SELECT userid, 'Los Pollos Hermanos'
FROM business_owner
WHERE budget > 50000;
```

Output:

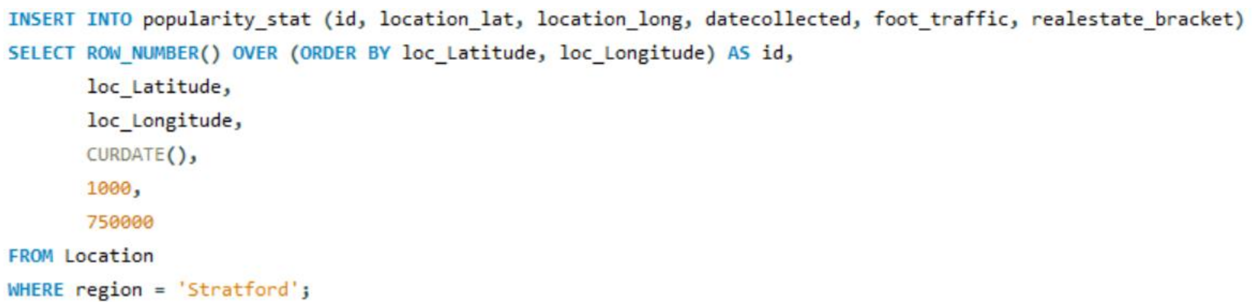


```
48  21:58:16  INSERT INTO business_affiliation (userid, affiliation) SELECT userid, 'Los Pollos Hermanos' FROM business_o...  0 row(s) affected Records: 0  Duplicates: 0  Warnings: 0  0.000 sec
```

Insert 3:

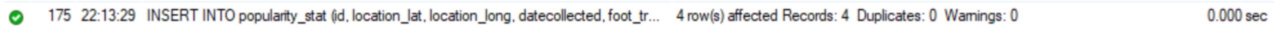
Latitudes and longitudes from "Locations" and a region, "Stratford", are taken and given values for foot traffic and real-estate bracket. They are then inserted into popularity stats.

Code:



```
INSERT INTO popularity_stat (id, location_lat, location_long, datecollected, foot_traffic, realestate_bracket)
SELECT ROW_NUMBER() OVER (ORDER BY loc_Latitude, loc_Longitude) AS id,
       loc_Latitude,
       loc_Longitude,
       CURDATE(),
       1000,
       750000
FROM Location
WHERE region = 'Stratford';
```

Output:

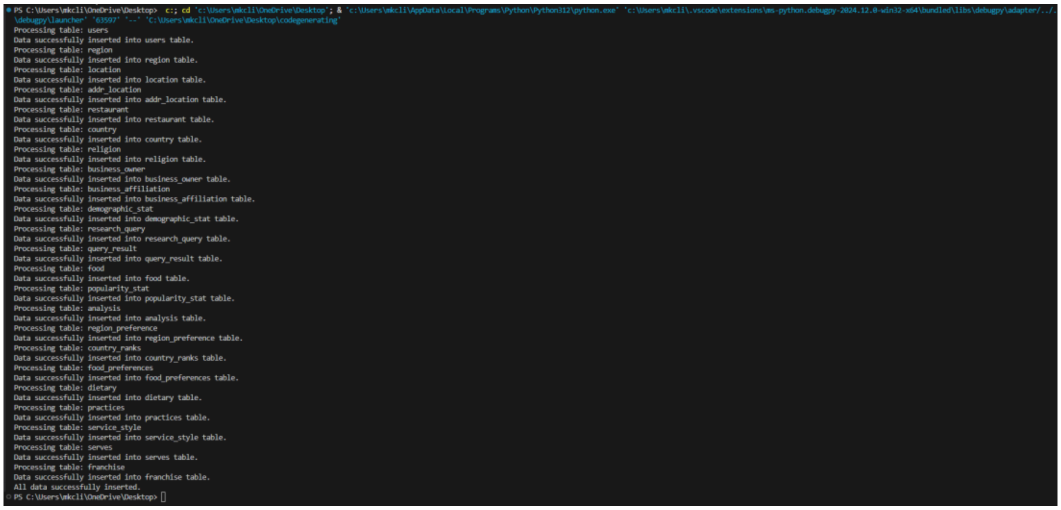


```
175  22:13:29  INSERT INTO popularity_stat (id, location_lat, location_long, datecollected, foot_tr...  4 row(s) affected Records: 4  Duplicates: 0  Warnings: 0    0.000 sec
```

Data Generation

Dataset information was created in google colab using predefined entries for each attribute, or a random generation and faker and outputted as a csv file. These csv files were then imported into a python script that uploaded the data into mysql by specifying the local dataset, username, password, and root.

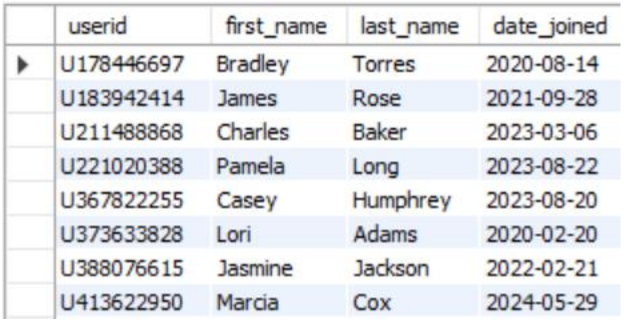
Data Upload from Python



```
● PS C:\Users\mkcli\OneDrive\Desktop>  c:; cd 'c:\Users\mkcli\OneDrive\Desktop'; & 'c:\Users\mkcli\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\mkcli\.vscode\extensions\ms-python.debugpy-2024.12.0-win32-x64\bundled\libs\debugpy\adapter/../.
  \debugpy\launcher' '63597' '--' 'C:\Users\mkcli\OneDrive\Desktop\codegenerating'
  Processing table: users
  Data successfully inserted into users table.
  Processing table: region
  Data successfully inserted into region table.
  Processing table: location
  Data successfully inserted into location table.
  Processing table: addr_location
  Data successfully inserted into addr_location table.
  Processing table: restaurant
  Data successfully inserted into restaurant table.
  Processing table: country
  Data successfully inserted into country table.
  Processing table: religion
  Data successfully inserted into religion table.
  Processing table: business_owner
  Data successfully inserted into business_owner table.
  Processing table: business_affiliation
  Data successfully inserted into business_affiliation table.
  Processing table: demographic_stat
  Data successfully inserted into demographic_stat table.
  Processing table: research_query
  Data successfully inserted into research_query table.
  Processing table: query_result
  Data successfully inserted into query_result table.
  Processing table: food
  Data successfully inserted into food table.
  Processing table: popularity_stat
  Data successfully inserted into popularity_stat table.
  Processing table: analysis
  Data successfully inserted into analysis table.
  Processing table: region_preference
  Data successfully inserted into region_preference table.
  Processing table: country_ranks
  Data successfully inserted into country_ranks table.
  Processing table: food_preferences
  Data successfully inserted into food_preferences table.
  Processing table: dietary
  Data successfully inserted into dietary table.
  Processing table: practices
  Data successfully inserted into practices table.
  Processing table: service_style
  Data successfully inserted into service_style table.
  Processing table: serves
  Data successfully inserted into serves table.
  Processing table: franchise
  Data successfully inserted into franchise table.
  All data successfully inserted.
○ PS C:\Users\mkcli\OneDrive\Desktop> ▯
```

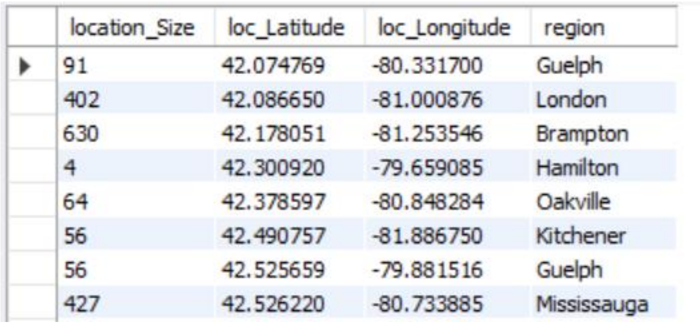
Data samples for each table:

users:



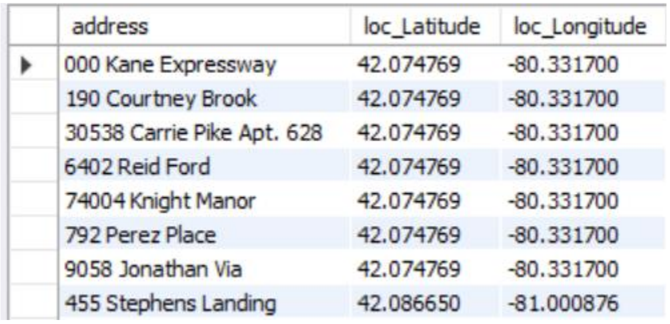
	userid	first_name	last_name	date_joined
▶	U178446697	Bradley	Torres	2020-08-14
	U183942414	James	Rose	2021-09-28
	U211488868	Charles	Baker	2023-03-06
	U221020388	Pamela	Long	2023-08-22
	U367822255	Casey	Humphrey	2023-08-20
	U373633828	Lori	Adams	2020-02-20
	U388076615	Jasmine	Jackson	2022-02-21
	U413622950	Marcia	Cox	2024-05-29

location:



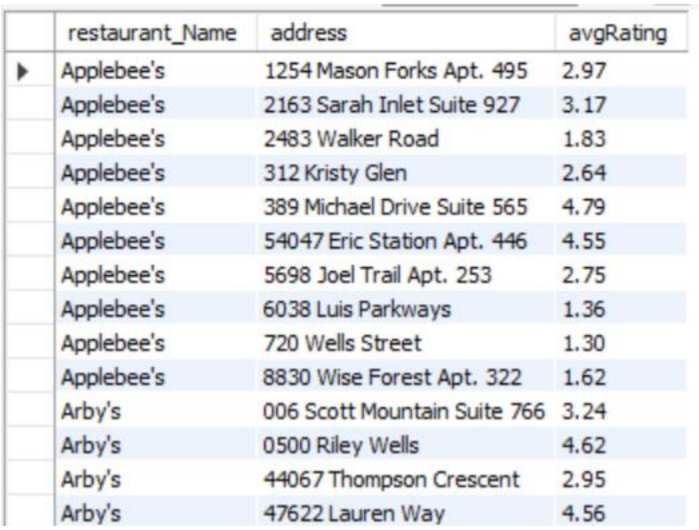
location_Size	loc_Latitude	loc_Longitude	region
91	42.074769	-80.331700	Guelph
402	42.086650	-81.000876	London
630	42.178051	-81.253546	Brampton
4	42.300920	-79.659085	Hamilton
64	42.378597	-80.848284	Oakville
56	42.490757	-81.886750	Kitchener
56	42.525659	-79.881516	Guelph
427	42.526220	-80.733885	Mississauga

addr_location:



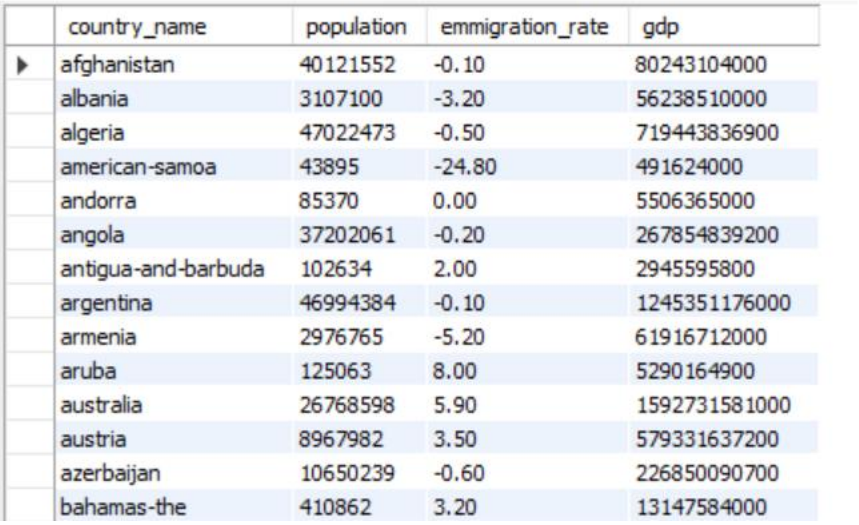
address	loc_Latitude	loc_Longitude
000 Kane Expressway	42.074769	-80.331700
190 Courtney Brook	42.074769	-80.331700
30538 Carrie Pike Apt. 628	42.074769	-80.331700
6402 Reid Ford	42.074769	-80.331700
74004 Knight Manor	42.074769	-80.331700
792 Perez Place	42.074769	-80.331700
9058 Jonathan Via	42.074769	-80.331700
455 Stephens Landing	42.086650	-81.000876

restaurant:



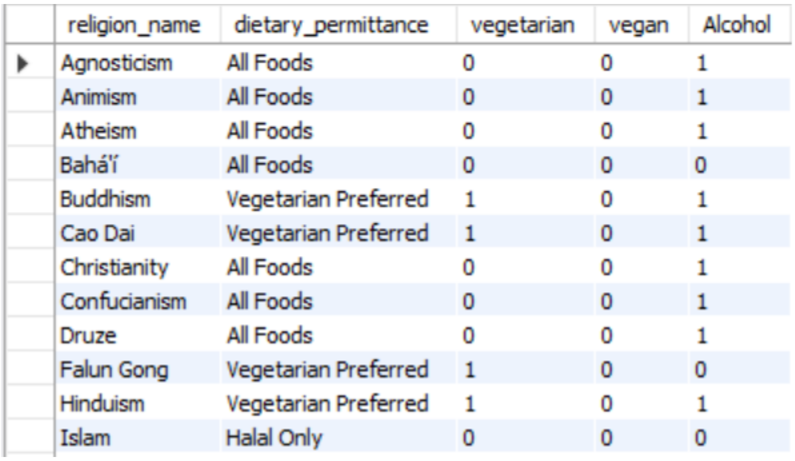
restaurant_Name	address	avgRating
Applebee's	1254 Mason Forks Apt. 495	2.97
Applebee's	2163 Sarah Inlet Suite 927	3.17
Applebee's	2483 Walker Road	1.83
Applebee's	312 Kristy Glen	2.64
Applebee's	389 Michael Drive Suite 565	4.79
Applebee's	54047 Eric Station Apt. 446	4.55
Applebee's	5698 Joel Trail Apt. 253	2.75
Applebee's	6038 Luis Parkways	1.36
Applebee's	720 Wells Street	1.30
Applebee's	8830 Wise Forest Apt. 322	1.62
Arby's	006 Scott Mountain Suite 766	3.24
Arby's	0500 Riley Wells	4.62
Arby's	44067 Thompson Crescent	2.95
Arby's	47622 Lauren Way	4.56

country:



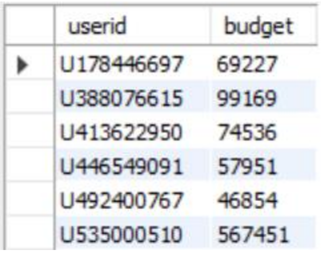
	country_name	population	emmigration_rate	gdp
▶	afghanistan	40121552	-0.10	80243104000
	albania	3107100	-3.20	56238510000
	algeria	47022473	-0.50	719443836900
	american-samoa	43895	-24.80	491624000
	andorra	85370	0.00	5506365000
	angola	37202061	-0.20	267854839200
	antigua-and-barbuda	102634	2.00	2945595800
	argentina	46994384	-0.10	1245351176000
	armenia	2976765	-5.20	61916712000
	aruba	125063	8.00	5290164900
	australia	26768598	5.90	1592731581000
	austria	8967982	3.50	579331637200
	azerbaijan	10650239	-0.60	226850090700
	bahamas-the	410862	3.20	13147584000

religion:



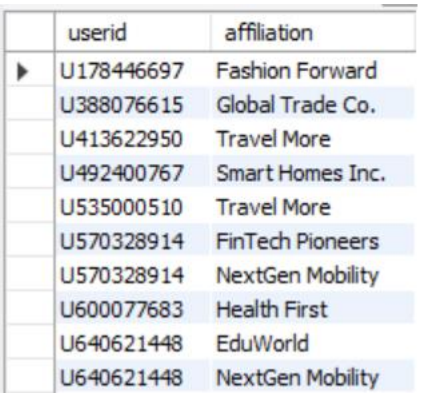
	religion_name	dietary_permittance	vegetarian	vegan	Alcohol
▶	Agnosticism	All Foods	0	0	1
	Animism	All Foods	0	0	1
	Atheism	All Foods	0	0	1
	Bahá'í	All Foods	0	0	0
	Buddhism	Vegetarian Preferred	1	0	1
	Cao Dai	Vegetarian Preferred	1	0	1
	Christianity	All Foods	0	0	1
	Confucianism	All Foods	0	0	1
	Druze	All Foods	0	0	1
	Falun Gong	Vegetarian Preferred	1	0	0
	Hinduism	Vegetarian Preferred	1	0	1
	Islam	Halal Only	0	0	0

business_owner:



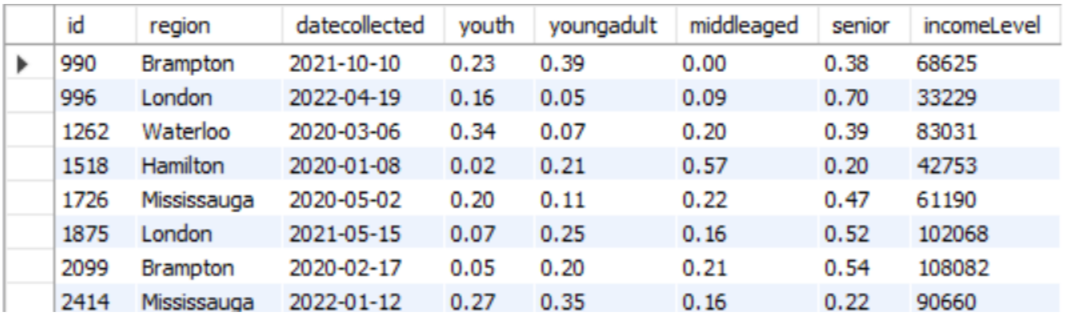
	userid	budget
▶	U178446697	69227
	U388076615	99169
	U413622950	74536
	U446549091	57951
	U492400767	46854
	U535000510	567451

business_affiliation:



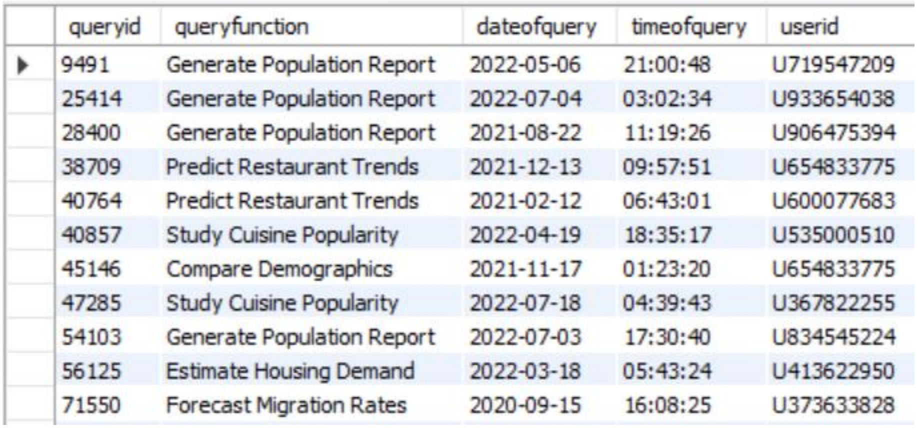
userid	affiliation
U178446697	Fashion Forward
U388076615	Global Trade Co.
U413622950	Travel More
U492400767	Smart Homes Inc.
U535000510	Travel More
U570328914	FinTech Pioneers
U570328914	NextGen Mobility
U600077683	Health First
U640621448	EduWorld
U640621448	NextGen Mobility

demographic_stat:



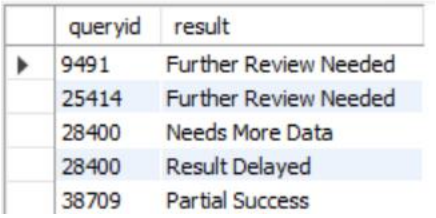
id	region	datecollected	youth	youngadult	middleaged	senior	incomeLevel
990	Brampton	2021-10-10	0.23	0.39	0.00	0.38	68625
996	London	2022-04-19	0.16	0.05	0.09	0.70	33229
1262	Waterloo	2020-03-06	0.34	0.07	0.20	0.39	83031
1518	Hamilton	2020-01-08	0.02	0.21	0.57	0.20	42753
1726	Mississauga	2020-05-02	0.20	0.11	0.22	0.47	61190
1875	London	2021-05-15	0.07	0.25	0.16	0.52	102068
2099	Brampton	2020-02-17	0.05	0.20	0.21	0.54	108082
2414	Mississauga	2022-01-12	0.27	0.35	0.16	0.22	90660

research_query:



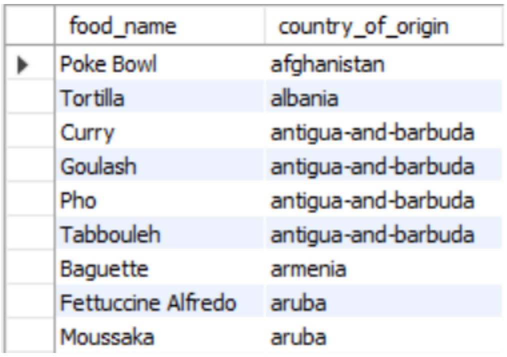
queryid	queryfunction	dateofquery	timeofquery	userid
9491	Generate Population Report	2022-05-06	21:00:48	U719547209
25414	Generate Population Report	2022-07-04	03:02:34	U933654038
28400	Generate Population Report	2021-08-22	11:19:26	U906475394
38709	Predict Restaurant Trends	2021-12-13	09:57:51	U654833775
40764	Predict Restaurant Trends	2021-02-12	06:43:01	U600077683
40857	Study Cuisine Popularity	2022-04-19	18:35:17	U535000510
45146	Compare Demographics	2021-11-17	01:23:20	U654833775
47285	Study Cuisine Popularity	2022-07-18	04:39:43	U367822255
54103	Generate Population Report	2022-07-03	17:30:40	U834545224
56125	Estimate Housing Demand	2022-03-18	05:43:24	U413622950
71550	Forecast Migration Rates	2020-09-15	16:08:25	U373633828

query_result:



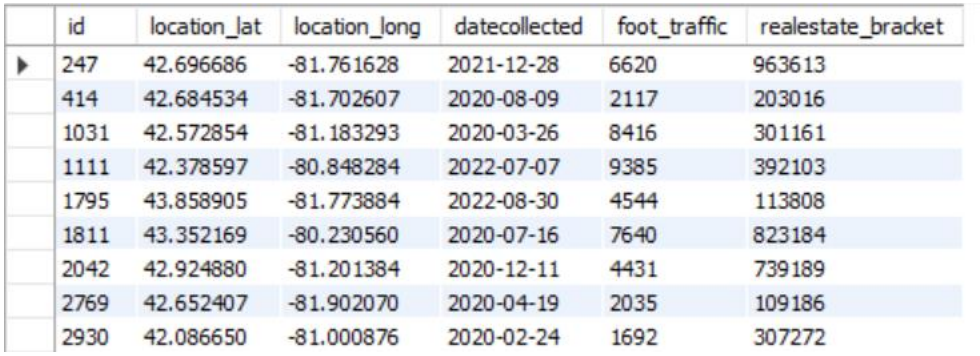
queryid	result
9491	Further Review Needed
25414	Further Review Needed
28400	Needs More Data
28400	Result Delayed
38709	Partial Success

food:



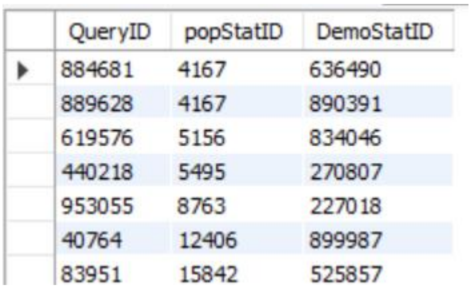
food_name	country_of_origin
Poke Bowl	afghanistan
Tortilla	albania
Curry	antigua-and-barbuda
Goulash	antigua-and-barbuda
Pho	antigua-and-barbuda
Tabbouleh	antigua-and-barbuda
Baguette	armenia
Fettuccine Alfredo	aruba
Moussaka	aruba

popularity_stat:



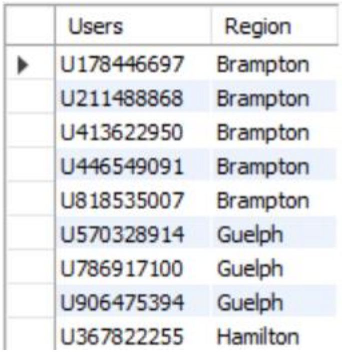
id	location_lat	location_long	datecollected	foot_traffic	realestate_bracket
247	42.696686	-81.761628	2021-12-28	6620	963613
414	42.684534	-81.702607	2020-08-09	2117	203016
1031	42.572854	-81.183293	2020-03-26	8416	301161
1111	42.378597	-80.848284	2022-07-07	9385	392103
1795	43.858905	-81.773884	2022-08-30	4544	113808
1811	43.352169	-80.230560	2020-07-16	7640	823184
2042	42.924880	-81.201384	2020-12-11	4431	739189
2769	42.652407	-81.902070	2020-04-19	2035	109186
2930	42.086650	-81.000876	2020-02-24	1692	307272

analysis:



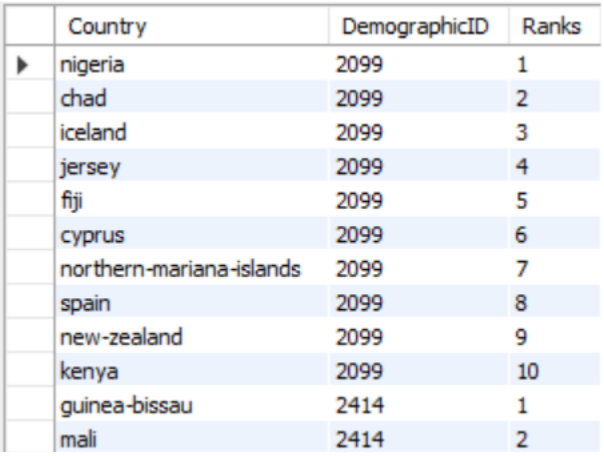
QueryID	popStatID	DemoStatID
884681	4167	636490
889628	4167	890391
619576	5156	834046
440218	5495	270807
953055	8763	227018
40764	12406	899987
83951	15842	525857

region_preference:



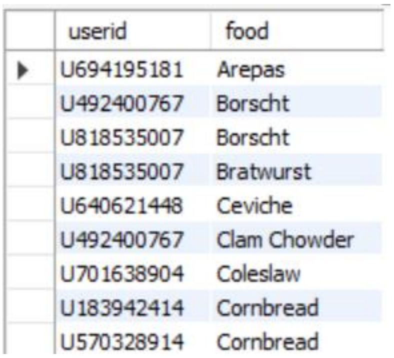
Users	Region
U178446697	Brampton
U211488868	Brampton
U413622950	Brampton
U446549091	Brampton
U818535007	Brampton
U570328914	Guelph
U786917100	Guelph
U906475394	Guelph
U367822255	Hamilton

country_rank:



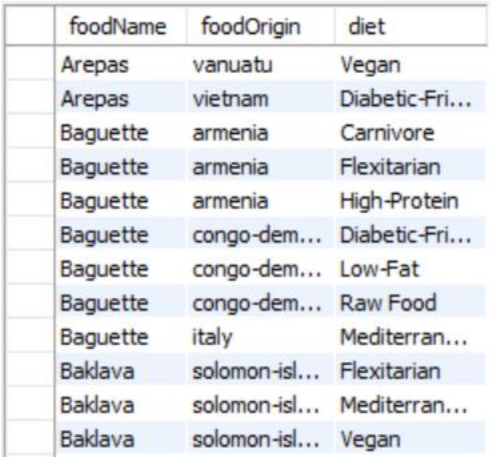
Country	DemographicID	Ranks
nigeria	2099	1
chad	2099	2
iceland	2099	3
jersey	2099	4
fiji	2099	5
cyprus	2099	6
northern-mariana-islands	2099	7
spain	2099	8
new-zealand	2099	9
kenya	2099	10
guinea-bissau	2414	1
mali	2414	2

food_preference:



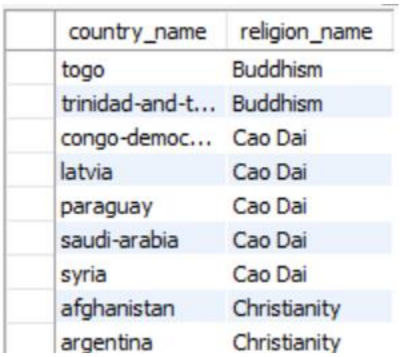
userid	food
U694195181	Arepas
U492400767	Borscht
U818535007	Borscht
U818535007	Bratwurst
U640621448	Ceviche
U492400767	Clam Chowder
U701638904	Coleslaw
U183942414	Cornbread
U570328914	Cornbread

dietary:



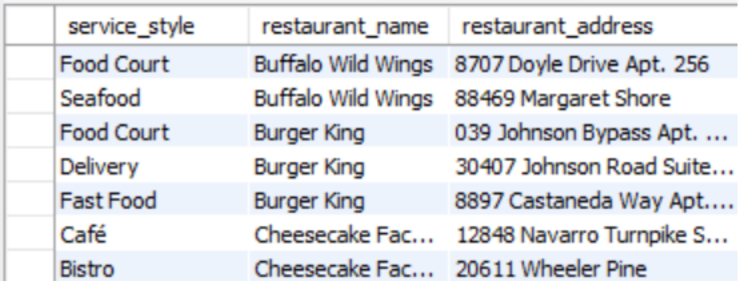
foodName	foodOrigin	diet
Arepas	vanuatu	Vegan
Arepas	vietnam	Diabetic-Fri...
Baguette	armenia	Carnivore
Baguette	armenia	Flexitarian
Baguette	armenia	High-Protein
Baguette	congo-dem...	Diabetic-Fri...
Baguette	congo-dem...	Low-Fat
Baguette	congo-dem...	Raw Food
Baguette	italy	Mediterran...
Baklava	solomon-isl...	Flexitarian
Baklava	solomon-isl...	Mediterran...
Baklava	solomon-isl...	Vegan

practices:



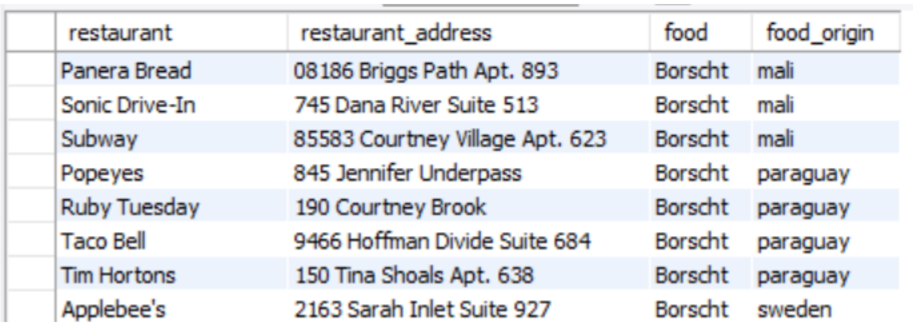
country_name	religion_name
togo	Buddhism
trinidad-and-t...	Buddhism
congo-democ...	Cao Dai
latvia	Cao Dai
paraguay	Cao Dai
saudi-arabia	Cao Dai
syria	Cao Dai
afghanistan	Christianity
argentina	Christianity

service_style:



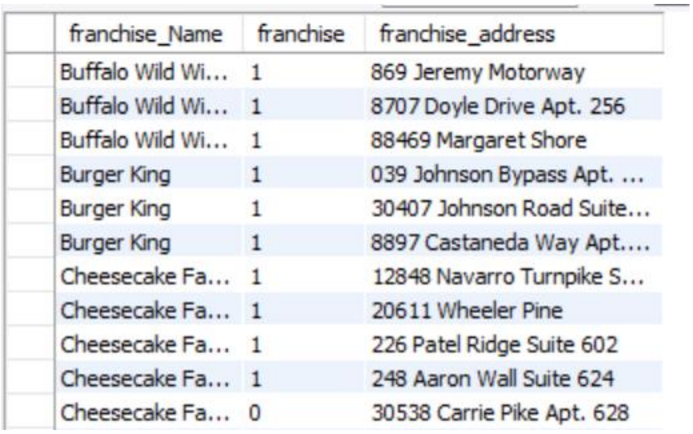
service_style	restaurant_name	restaurant_address
Food Court	Buffalo Wild Wings	8707 Doyle Drive Apt. 256
Seafood	Buffalo Wild Wings	88469 Margaret Shore
Food Court	Burger King	039 Johnson Bypass Apt. ...
Delivery	Burger King	30407 Johnson Road Suite...
Fast Food	Burger King	8897 Castaneda Way Apt....
Café	Cheesecake Fac...	12848 Navarro Turnpike S...
Bistro	Cheesecake Fac...	20611 Wheeler Pine

serves:



	restaurant	restaurant_address	food	food_origin
	Panera Bread	08186 Briggs Path Apt. 893	Borscht	mali
	Sonic Drive-In	745 Dana River Suite 513	Borscht	mali
	Subway	85583 Courtney Village Apt. 623	Borscht	mali
	Popeyes	845 Jennifer Underpass	Borscht	paraguay
	Ruby Tuesday	190 Courtney Brook	Borscht	paraguay
	Taco Bell	9466 Hoffman Divide Suite 684	Borscht	paraguay
	Tim Hortons	150 Tina Shoals Apt. 638	Borscht	paraguay
	Applebee's	2163 Sarah Inlet Suite 927	Borscht	sweden

franchise:



	franchise_Name	franchise	franchise_address
	Buffalo Wild Wi...	1	869 Jeremy Motorway
	Buffalo Wild Wi...	1	8707 Doyle Drive Apt. 256
	Buffalo Wild Wi...	1	88469 Margaret Shore
	Burger King	1	039 Johnson Bypass Apt. ...
	Burger King	1	30407 Johnson Road Suite...
	Burger King	1	8897 Castaneda Way Apt....
	Cheesecake Fa...	1	12848 Navarro Turnpike S...
	Cheesecake Fa...	1	20611 Wheeler Pine
	Cheesecake Fa...	1	226 Patel Ridge Suite 602
	Cheesecake Fa...	1	248 Aaron Wall Suite 624
	Cheesecake Fa...	0	30538 Carrie Pike Apt. 628

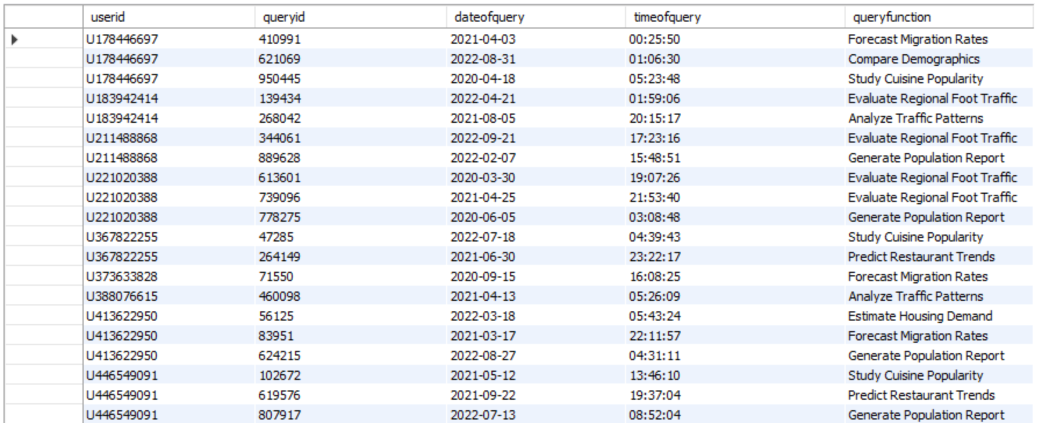
Seven Queries:

Turn in a copy of all your SQL queries and the system's response to convince us that your commands run successfully. Please do not turn in query results that are thousands (or hundreds of thousands) of lines long!

Query 1: List the Queries for a Given User

This query will create a list of the history of queries that users have created.

```
#Query 1: Queries Users have Created
SELECT u.userid, r.queryid, r.dateofquery, r.timeofquery, r.queryfunction
FROM users u JOIN research_Query r USING (userid)
LIMIT 20;
```

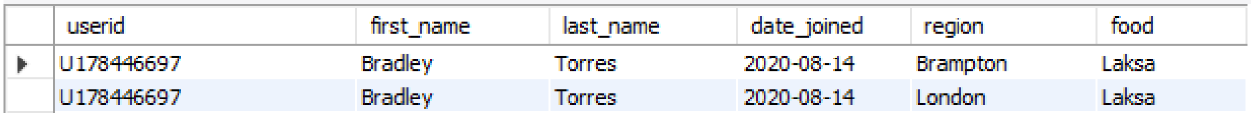


userid	queryid	dateofquery	timeofquery	queryfunction
U178446697	410991	2021-04-03	00:25:50	Forecast Migration Rates
U178446697	621069	2022-08-31	01:06:30	Compare Demographics
U178446697	950445	2020-04-18	05:23:48	Study Cuisine Popularity
U183942414	139434	2022-04-21	01:59:06	Evaluate Regional Foot Traffic
U183942414	268042	2021-08-05	20:15:17	Analyze Traffic Patterns
U211488868	344061	2022-09-21	17:23:16	Evaluate Regional Foot Traffic
U211488868	889628	2022-02-07	15:48:51	Generate Population Report
U221020388	613601	2020-03-30	19:07:26	Evaluate Regional Foot Traffic
U221020388	739096	2021-04-25	21:53:40	Evaluate Regional Foot Traffic
U221020388	778275	2020-06-05	03:08:48	Generate Population Report
U367822255	47285	2022-07-18	04:39:43	Study Cuisine Popularity
U367822255	264149	2021-06-30	23:22:17	Predict Restaurant Trends
U373633828	71550	2020-09-15	16:08:25	Forecast Migration Rates
U388076615	460098	2021-04-13	05:26:09	Analyze Traffic Patterns
U413622950	56125	2022-03-18	05:43:24	Estimate Housing Demand
U413622950	83951	2021-03-17	22:11:57	Forecast Migration Rates
U413622950	624215	2022-08-27	04:31:11	Generate Population Report
U446549091	102672	2021-05-12	13:46:10	Study Cuisine Popularity
U446549091	619576	2021-09-22	19:37:04	Predict Restaurant Trends
U446549091	807917	2022-07-13	08:52:04	Generate Population Report

Query 2: Shows the Preferences of a Given User

We can look up the preferences that have been chosen by a given user.

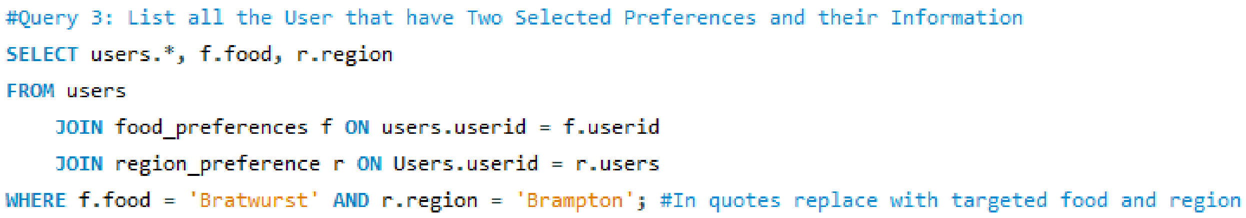
```
#Query 2: Shows the Preferences of a Given User
SELECT s.*, p.region, f.food
FROM users s
    JOIN Region_Preference p ON s.userid = p.users
    JOIN Food_Preferences f ON s.userid = f.userid
WHERE s.userid = ("U178446697") #In quotes replace with targeted userid
LIMIT 20;
```



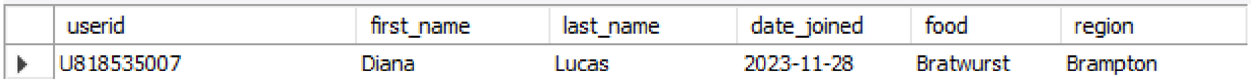
userid	first_name	last_name	date_joined	region	food
U178446697	Bradley	Torres	2020-08-14	Brampton	Laksa
U178446697	Bradley	Torres	2020-08-14	London	Laksa

Query 3: List all the User that have Two Selected Preferences and their Information

For the given set of region and food, we see a list of users and their information, where those two choices are true

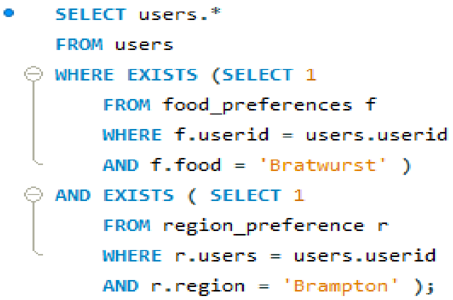


```
#Query 3: List all the User that have Two Selected Preferences and their Information
SELECT users.*, f.food, r.region
FROM users
    JOIN food_preferences f ON users.userid = f.userid
    JOIN region_preference r ON Users.userid = r.users
WHERE f.food = 'Bratwurst' AND r.region = 'Brampton'; #In quotes replace with targeted food and region
```

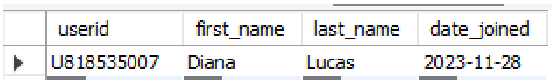


	userid	first_name	last_name	date_joined	food	region
▶	U818535007	Diana	Lucas	2023-11-28	Bratwurst	Brampton

using exists, only shows the list of users and their information



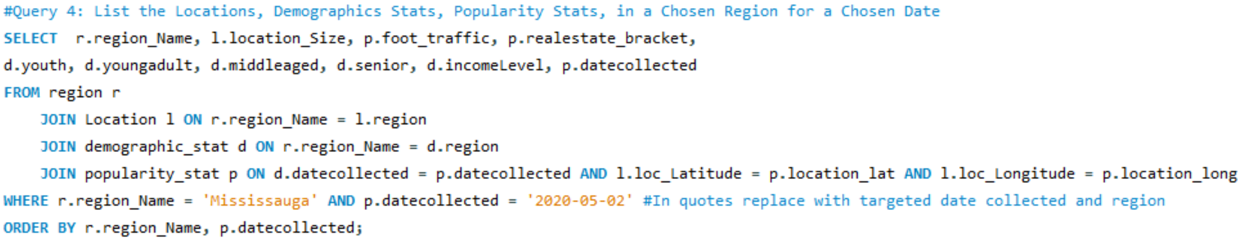
```
SELECT users.*
FROM users
WHERE EXISTS (SELECT 1
    FROM food_preferences f
    WHERE f.userid = users.userid
    AND f.food = 'Bratwurst' )
AND EXISTS ( SELECT 1
    FROM region_preference r
    WHERE r.users = users.userid
    AND r.region = 'Brampton' );
```



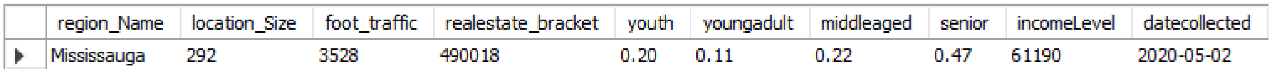
	userid	first_name	last_name	date_joined
▶	U818535007	Diana	Lucas	2023-11-28

Query 4: List the Locations, Demographics Stats, Popularity Stats, in a Chosen Region for a Chosen Date

For a given date and region, we can see information about popularity and demographics



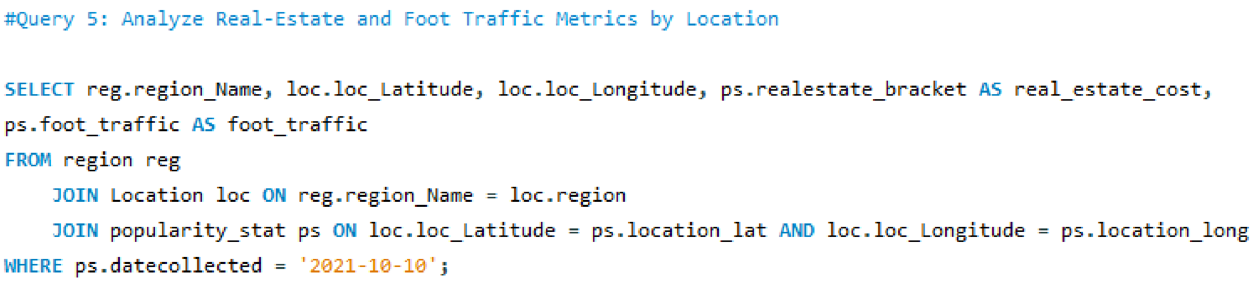
```
#Query 4: List the Locations, Demographics Stats, Popularity Stats, in a Chosen Region for a Chosen Date
SELECT  r.region_Name, l.location_Size, p.foot_traffic, p.realestate_bracket,
d.youth, d.youngadult, d.middleaged, d.senior, d.incomeLevel, p.datecollected
FROM region r
    JOIN Location l ON r.region_Name = l.region
    JOIN demographic_stat d ON r.region_Name = d.region
    JOIN popularity_stat p ON d.datecollected = p.datecollected AND l.loc_Latitude = p.location_lat AND l.loc_Longitude = p.location_long
WHERE r.region_Name = 'Mississauga' AND p.datecollected = '2020-05-02' #In quotes replace with targeted date collected and region
ORDER BY r.region_Name, p.datecollected;
```



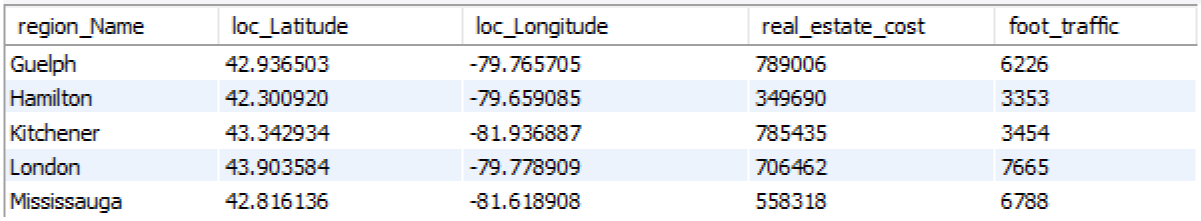
	region_Name	location_Size	foot_traffic	realestate_bracket	youth	youngadult	middleaged	senior	incomeLevel	datecollected
▶	Mississauga	292	3528	490018	0.20	0.11	0.22	0.47	61190	2020-05-02

Query 5: Analyze Real-Estate and Foot Traffic Metrics by Location

Retrieves real-estate costs and foot traffic metrics for each location within a region in the present. This provides an overview of location-specific metrics for the evaluation of affordability and popularity. Startups can use this data to select locations that balance high visibility with affordability.



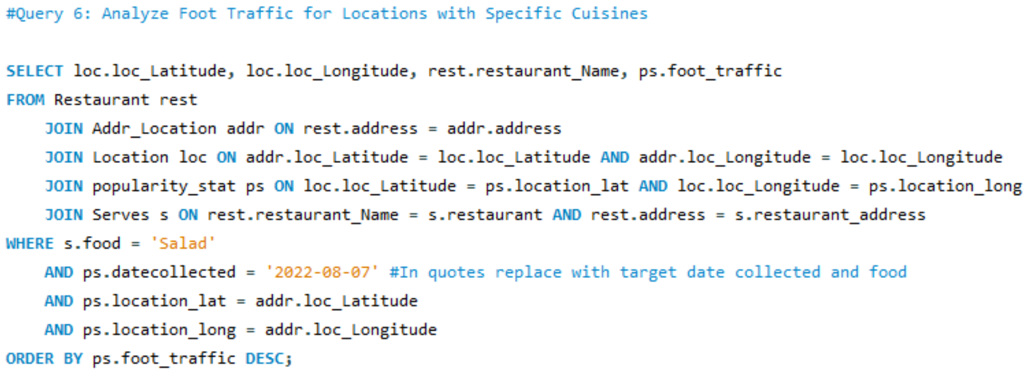
```
#Query 5: Analyze Real-Estate and Foot Traffic Metrics by Location

SELECT reg.region_Name, loc.loc_Latitude, loc.loc_Longitude, ps.realestate_bracket AS real_estate_cost,
ps.foot_traffic AS foot_traffic
FROM region reg
    JOIN Location loc ON reg.region_Name = loc.region
    JOIN popularity_stat ps ON loc.loc_Latitude = ps.location_lat AND loc.loc_Longitude = ps.location_long
WHERE ps.datecollected = '2021-10-10';
```



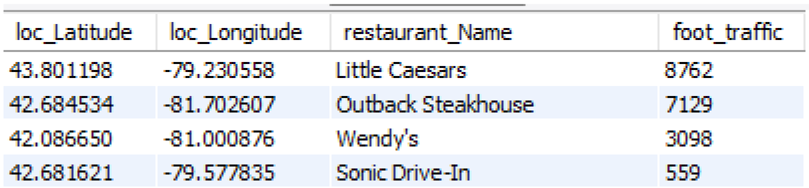
region_Name	loc_Latitude	loc_Longitude	real_estate_cost	foot_traffic
Guelph	42.936503	-79.765705	789006	6226
Hamilton	42.300920	-79.659085	349690	3353
Kitchener	43.342934	-81.936887	785435	3454
London	43.903584	-79.778909	706462	7665
Mississauga	42.816136	-81.618908	558318	6788

Query 6: Analyze Foot Traffic for Locations with Specific Cuisines

Identifies locations where restaurants serving a specific cuisine have a high visibility. Provides actionable insights for startups looking to serve particular cuisines in high-traffic areas by prioritizing locations where target cuisines are already popular ensuring a better chance of success.



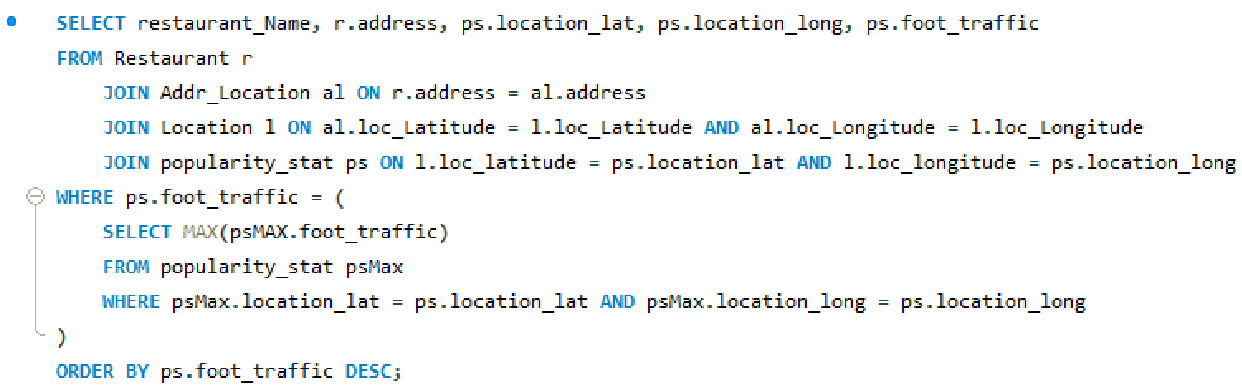
```
#Query 6: Analyze Foot Traffic for Locations with Specific Cuisines

SELECT loc.loc_Latitude, loc.loc_Longitude, rest.restaurant_Name, ps.foot_traffic
FROM Restaurant rest
    JOIN Addr_Location addr ON rest.address = addr.address
    JOIN Location loc ON addr.loc_Latitude = loc.loc_Latitude AND addr.loc_Longitude = loc.loc_Longitude
    JOIN popularity_stat ps ON loc.loc_Latitude = ps.location_lat AND loc.loc_Longitude = ps.location_long
    JOIN Serves s ON rest.restaurant_Name = s.restaurant AND rest.address = s.restaurant_address
WHERE s.food = 'Salad'
    AND ps.datecollected = '2022-08-07' #In quotes replace with target date collected and food
    AND ps.location_lat = addr.loc_Latitude
    AND ps.location_long = addr.loc_Longitude
ORDER BY ps.foot_traffic DESC;
```



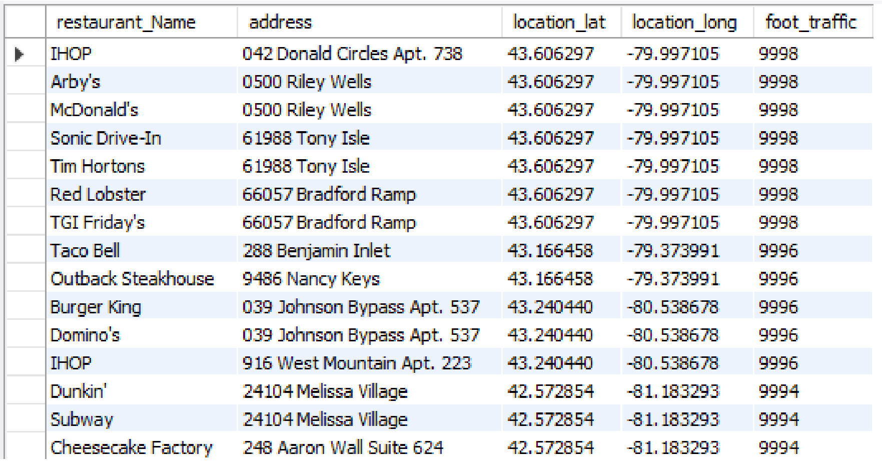
loc_Latitude	loc_Longitude	restaurant_Name	foot_traffic
43.801198	-79.230558	Little Caesars	8762
42.684534	-81.702607	Outback Steakhouse	7129
42.086650	-81.000876	Wendy's	3098
42.681621	-79.577835	Sonic Drive-In	559

Query 7: List the Most Popular Restaurants by Foot Traffic

This query identifies the most popular restaurants based on foot traffic at their respective locations. It uses a subquery to find the maximum foot traffic for each location and filters for restaurants with matching traffic values. The results are ordered in descending order of foot traffic to display the most popular restaurants with respect to foot traffic at the top. This query could provide insights into successful restaurant locations which can guide startups or business expansions.



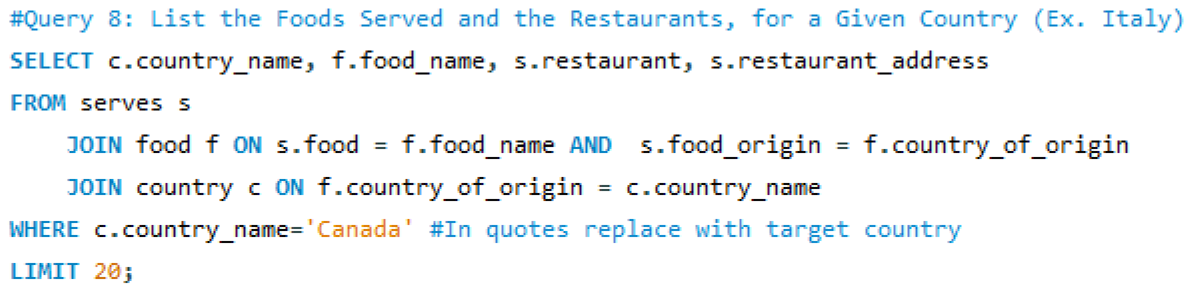
```
SELECT restaurant_Name, r.address, ps.location_lat, ps.location_long, ps.foot_traffic
FROM Restaurant r
    JOIN Addr_Location al ON r.address = al.address
    JOIN Location l ON al.loc_Latitude = l.loc_Latitude AND al.loc_Longitude = l.loc_Longitude
    JOIN popularity_stat ps ON l.loc_latitude = ps.location_lat AND l.loc_longitude = ps.location_long
WHERE ps.foot_traffic = (
    SELECT MAX(psMAX.foot_traffic)
    FROM popularity_stat psMax
    WHERE psMax.location_lat = ps.location_lat AND psMax.location_long = ps.location_long
)
ORDER BY ps.foot_traffic DESC;
```



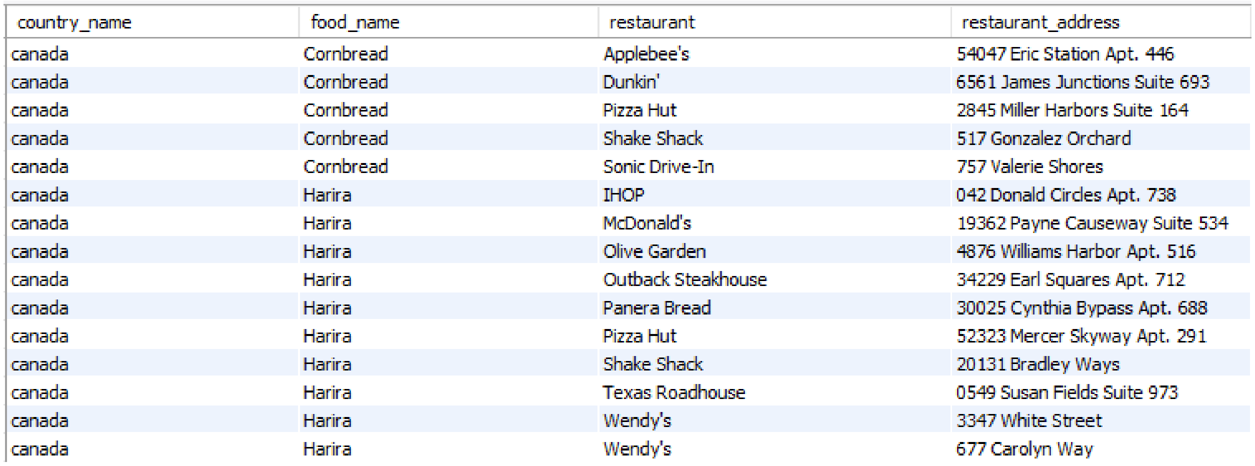
restaurant_Name	address	location_lat	location_long	foot_traffic
IHOP	042 Donald Circles Apt. 738	43.606297	-79.997105	9998
Arby's	0500 Riley Wells	43.606297	-79.997105	9998
McDonald's	0500 Riley Wells	43.606297	-79.997105	9998
Sonic Drive-In	61988 Tony Isle	43.606297	-79.997105	9998
Tim Hortons	61988 Tony Isle	43.606297	-79.997105	9998
Red Lobster	66057 Bradford Ramp	43.606297	-79.997105	9998
TGI Friday's	66057 Bradford Ramp	43.606297	-79.997105	9998
Taco Bell	288 Benjamin Inlet	43.166458	-79.373991	9996
Outback Steakhouse	9486 Nancy Keys	43.166458	-79.373991	9996
Burger King	039 Johnson Bypass Apt. 537	43.240440	-80.538678	9996
Domino's	039 Johnson Bypass Apt. 537	43.240440	-80.538678	9996
IHOP	916 West Mountain Apt. 223	43.240440	-80.538678	9996
Dunkin'	24104 Melissa Village	42.572854	-81.183293	9994
Subway	24104 Melissa Village	42.572854	-81.183293	9994
Cheesecake Factory	248 Aaron Wall Suite 624	42.572854	-81.183293	9994

Query 8: List the foods served and the restaurants that serve it.

Shows the foods and the restaurant information for a given country.



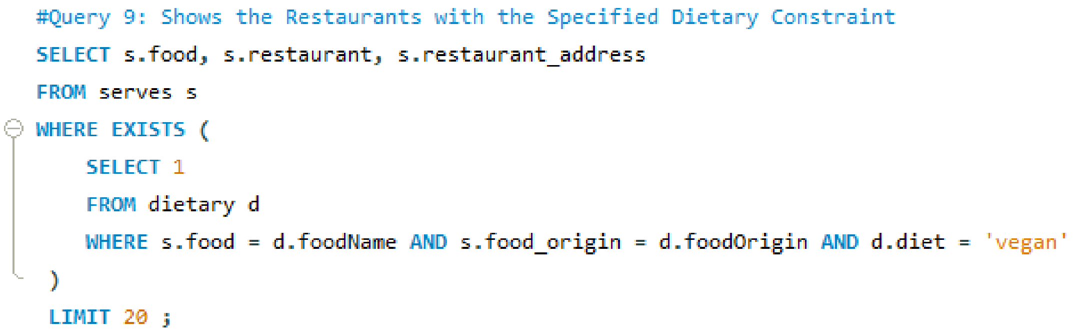
```
#Query 8: List the Foods Served and the Restaurants, for a Given Country (Ex. Italy)
SELECT c.country_name, f.food_name, s.restaurant, s.restaurant_address
FROM serves s
    JOIN food f ON s.food = f.food_name AND  s.food_origin = f.country_of_origin
    JOIN country c ON f.country_of_origin = c.country_name
WHERE c.country_name='Canada' #In quotes replace with target country
LIMIT 20;
```



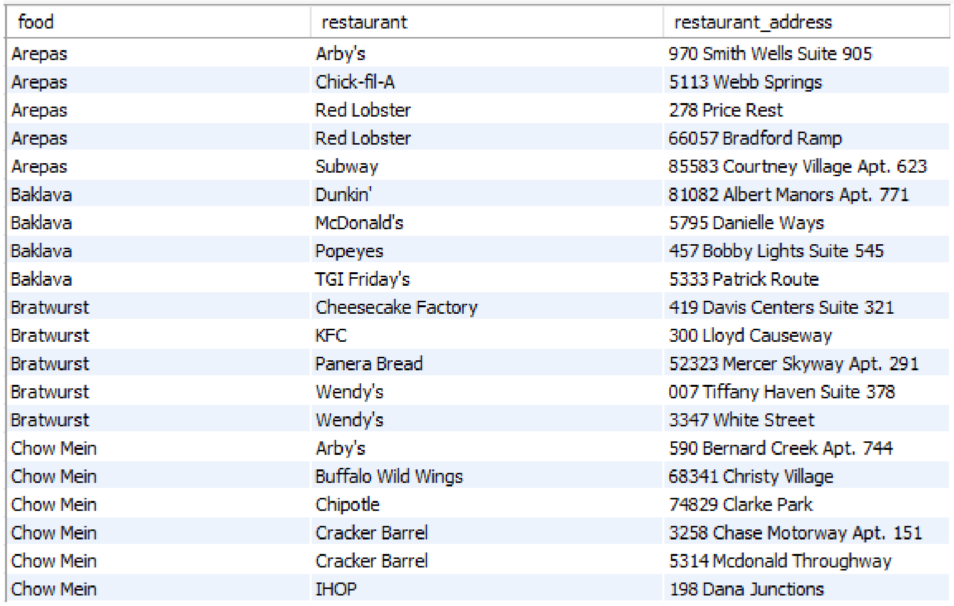
country_name	food_name	restaurant	restaurant_address
canada	Cornbread	Applebee's	54047 Eric Station Apt. 446
canada	Cornbread	Dunkin'	6561 James Junctions Suite 693
canada	Cornbread	Pizza Hut	2845 Miller Harbors Suite 164
canada	Cornbread	Shake Shack	517 Gonzalez Orchard
canada	Cornbread	Sonic Drive-In	757 Valerie Shores
canada	Harira	IHOP	042 Donald Circles Apt. 738
canada	Harira	McDonald's	19362 Payne Causeway Suite 534
canada	Harira	Olive Garden	4876 Williams Harbor Apt. 516
canada	Harira	Outback Steakhouse	34229 Earl Squares Apt. 712
canada	Harira	Panera Bread	30025 Cynthia Bypass Apt. 688
canada	Harira	Pizza Hut	52323 Mercer Skyway Apt. 291
canada	Harira	Shake Shack	20131 Bradley Ways
canada	Harira	Texas Roadhouse	0549 Susan Fields Suite 973
canada	Harira	Wendy's	3347 White Street
canada	Harira	Wendy's	677 Carolyn Way

Query 9: Shows the Restaurants with the Specified Dietary Constraint

For a chosen dietary restriction (EX. Vegan), we list the restaurants that serve it



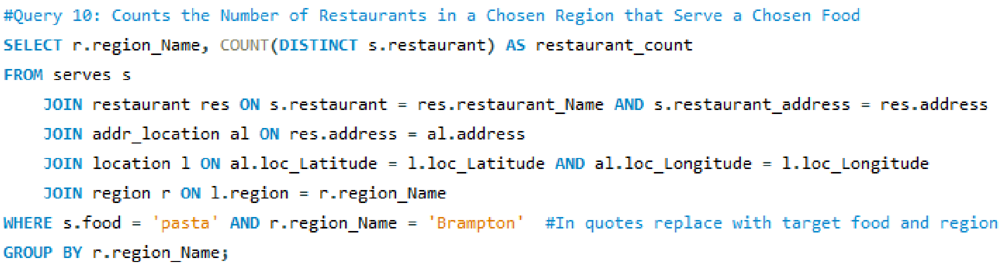
```
#Query 9: Shows the Restaurants with the Specified Dietary Constraint
SELECT s.food, s.restaurant, s.restaurant_address
FROM serves s
WHERE EXISTS (
    SELECT 1
    FROM dietary d
    WHERE s.food = d.foodName AND s.food_origin = d.foodOrigin AND d.diet = 'vegan'
 )
 LIMIT 20 ;
```



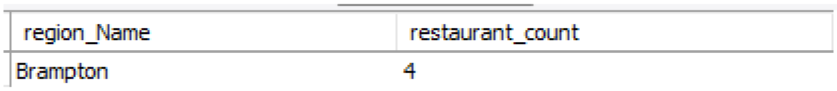
food	restaurant	restaurant_address
Arepas	Arby's	970 Smith Wells Suite 905
Arepas	Chick-fil-A	5113 Webb Springs
Arepas	Red Lobster	278 Price Rest
Arepas	Red Lobster	66057 Bradford Ramp
Arepas	Subway	85583 Courtney Village Apt. 623
Baklava	Dunkin'	81082 Albert Manors Apt. 771
Baklava	McDonald's	5795 Danielle Ways
Baklava	Popeyes	457 Bobby Lights Suite 545
Baklava	TGI Friday's	5333 Patrick Route
Bratwurst	Cheesecake Factory	419 Davis Centers Suite 321
Bratwurst	KFC	300 Lloyd Causeway
Bratwurst	Panera Bread	52323 Mercer Skyway Apt. 291
Bratwurst	Wendy's	007 Tiffany Haven Suite 378
Bratwurst	Wendy's	3347 White Street
Chow Mein	Arby's	590 Bernard Creek Apt. 744
Chow Mein	Buffalo Wild Wings	68341 Christy Village
Chow Mein	Chipotle	74829 Clarke Park
Chow Mein	Cracker Barrel	3258 Chase Motorway Apt. 151
Chow Mein	Cracker Barrel	5314 Mcdonald Throughway
Chow Mein	IHOP	198 Dana Junctions

Query 10: Counts the Number of Restaurants in a Chosen Region that Serve a Chosen Food

By specifying a region and a food, we can see how many restaurants serve it.



```
#Query 10: Counts the Number of Restaurants in a Chosen Region that Serve a Chosen Food
SELECT r.region_Name, COUNT(DISTINCT s.restaurant) AS restaurant_count
FROM serves s
    JOIN restaurant res ON s.restaurant = res.restaurant_Name AND s.restaurant_address = res.address
    JOIN addr_location al ON res.address = al.address
    JOIN location l ON al.loc_Latitude = l.loc_Latitude AND al.loc_Longitude = l.loc_Longitude
    JOIN region r ON l.region = r.region_Name
WHERE s.food = 'pasta' AND r.region_Name = 'Brampton'  #In quotes replace with target food and region
GROUP BY r.region_Name;
```



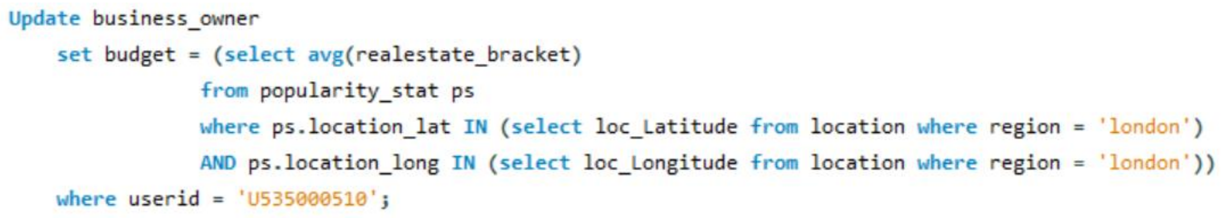
region_Name	restaurant_count
Brampton	4

Data Modification

Modification 1:

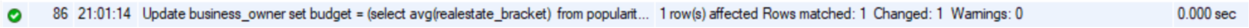
This update sets a business owners budget to the average price of real estate in a target region. The interesting feature of this update is that is inserts the results of a search, using a subquery with an aggregate.

Code:



```
Update business_owner
    set budget = (select avg(realestate_bracket)
                from popularity_stat ps
                where ps.location_lat IN (select loc_Latitude from location where region = 'london')
                AND ps.location_long IN (select loc_Longitude from location where region = 'london'))
    where userid = 'U535000510';
```

Output:

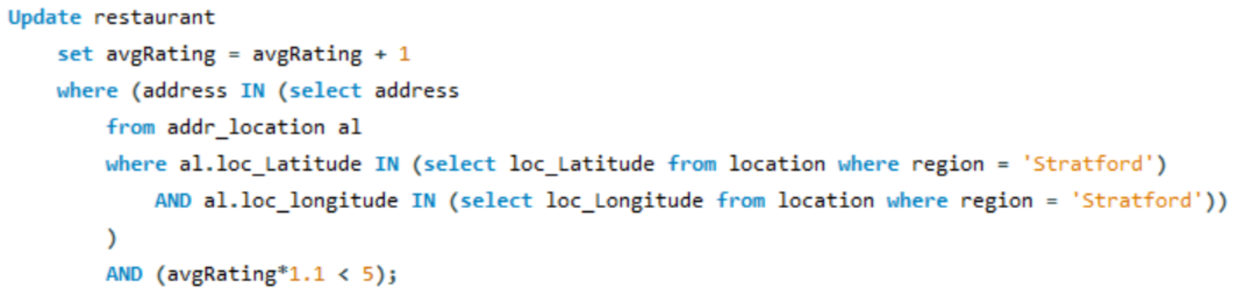


```
86  21:01:14  Update business_owner set budget = (select avg(realestate_bracket)  from popularit...  1 row(s) affected Rows matched: 1  Changed: 1  Warnings: 0    0.000 sec
```

Modification 2:

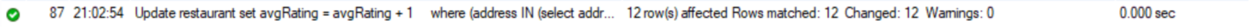
This update increases the average rating of every restaurant in a region, unless that increase would put the rating over 5 out of 5. The interesting feature of this update is that it modifies the values of many tuples based on their current values, and the where condition uses a double nested subquery.

Code:



```
Update restaurant
    set avgRating = avgRating + 1
    where (address IN (select address
        from addr_location al
        where al.loc_Latitude IN (select loc_Latitude from location where region = 'Stratford')
            AND al.loc_longitude IN (select loc_Longitude from location where region = 'Stratford'))
        )
        AND (avgRating*1.1 < 5);
```

Output:

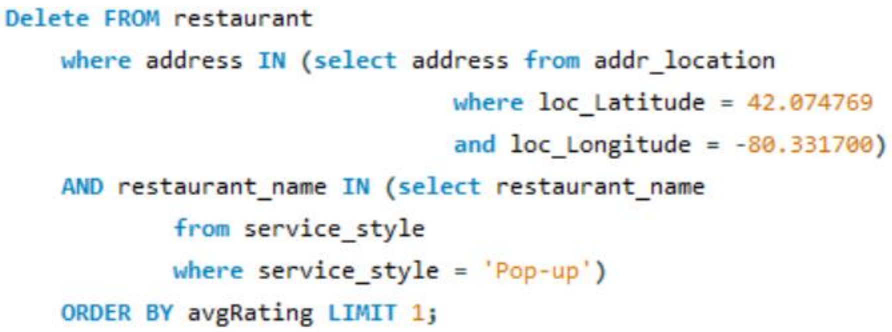


```
87  21:02:54  Update restaurant set avgRating = avgRating + 1    where (address IN (select addr...  12 row(s) affected Rows matched: 12  Changed: 12  Warnings: 0    0.000 sec
```

Modification 3:

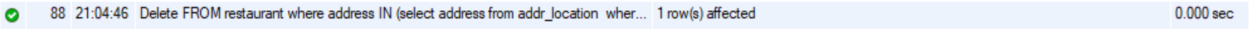
This modification deletes the lowest rated restaurant for a certain service style in a location. The interesting feature of this deletion is that it uses two subqueries for the where conditions and only deletes the tuple with the lowest average rating.

Code:



```
Delete FROM restaurant
    where address IN (select address from addr_location
                                where loc_Latitude = 42.074769
                                and loc_Longitude = -80.331700)
    AND restaurant_name IN (select restaurant_name
            from service_style
            where service_style = 'Pop-up')
    ORDER BY avgRating LIMIT 1;
```

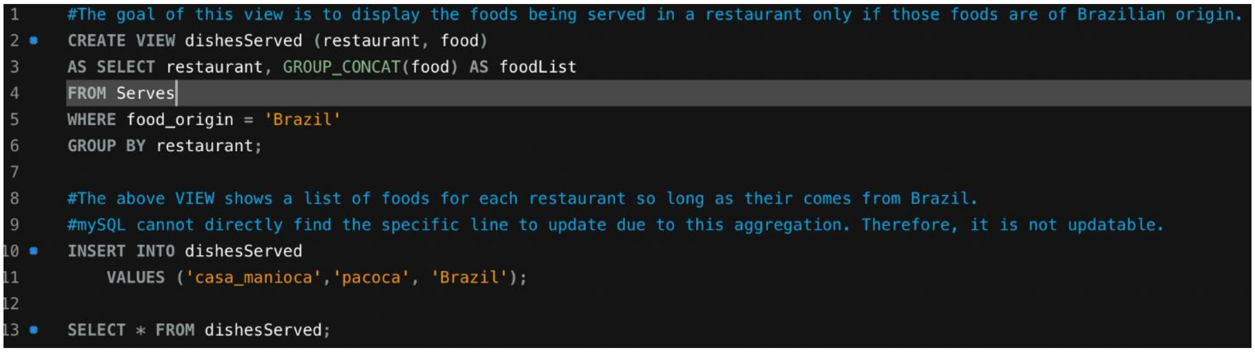
Output:



```
✓   88  21:04:46  Delete FROM restaurant where address IN (select address from addr_location  wher...  1 row(s) affected      0.000 sec
```

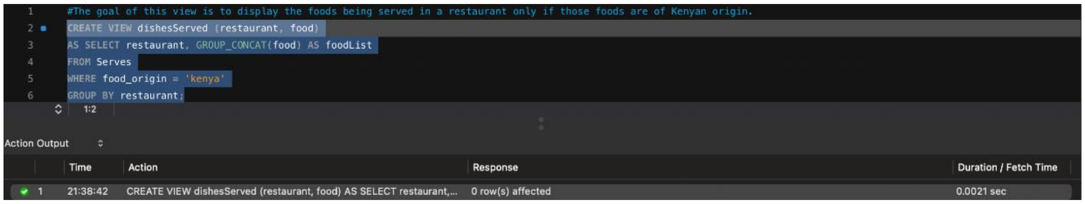
Views

First view:



```
#The goal of this view is to display the foods being served in a restaurant only if those foods are of Brazilian origin.
CREATE VIEW dishesServed (restaurant, food)
AS SELECT restaurant, GROUP_CONCAT(food) AS foodList
FROM Serves
WHERE food_origin = 'Brazil'
GROUP BY restaurant;

#The above VIEW shows a list of foods for each restaurant so long as their comes from Brazil.
#mySQL cannot directly find the specific line to update due to this aggregation. Therefore, it is not updatable.
INSERT INTO dishesServed
    VALUES ('casa_manioca','pacoca', 'Brazil');

SELECT * FROM dishesServed;
```

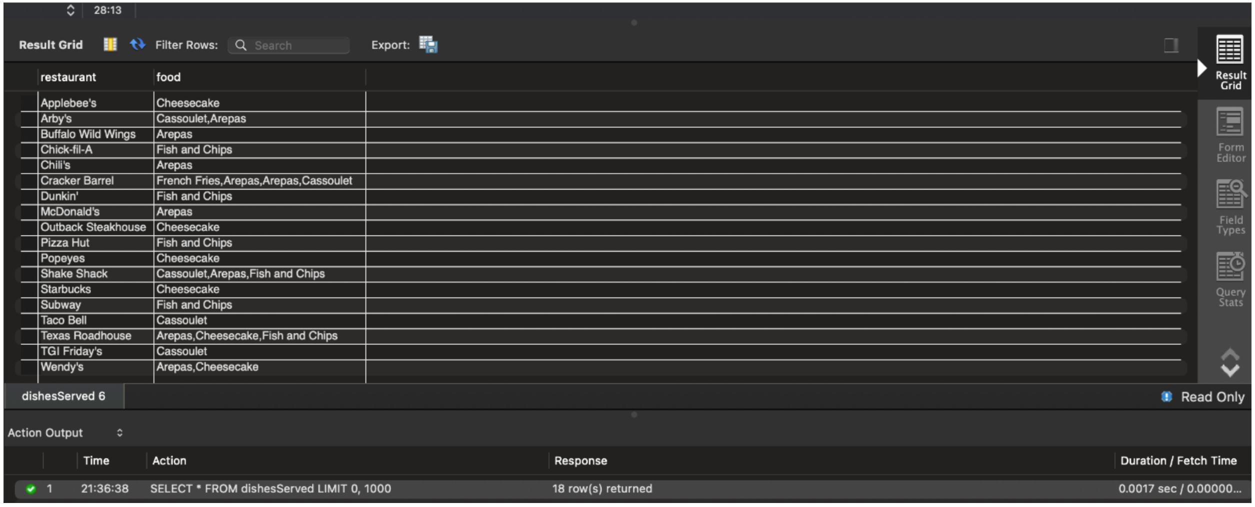
System response:



```
#The goal of this view is to display the foods being served in a restaurant only if those foods are of Kenyan origin.
CREATE VIEW dishesServed (restaurant, food)
AS SELECT restaurant, GROUP_CONCAT(food) AS foodList
FROM Serves
WHERE food_origin = 'kenya'
GROUP BY restaurant;
```

```
     Time       Action                                                       Response            Duration / Fetch Time
1    21:38:42   CREATE VIEW dishesServed (restaurant, food) AS SELECT restaurant,...   0 row(s) affected   0.0021 sec
```

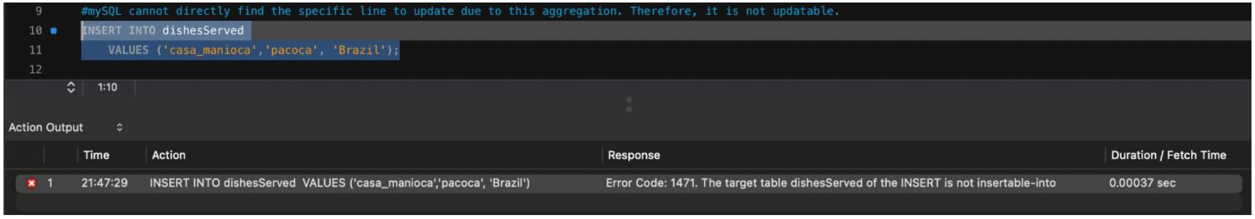
Query of the view:



restaurant	food
Applebee's	Cheesecake
Arby's	Cassoulet,Arepas
Buffalo Wild Wings	Arepas
Chick-fil-A	Fish and Chips
Chili's	Arepas
Cracker Barrel	French Fries,Arepas,Arepas,Cassoulet
Dunkin'	Fish and Chips
McDonald's	Arepas
Outback Steakhouse	Cheesecake
Pizza Hut	Fish and Chips
Popeyes	Cheesecake
Shake Shack	Cassoulet,Arepas,Fish and Chips
Starbucks	Cheesecake
Subway	Fish and Chips
Taco Bell	Cassoulet
Texas Roadhouse	Arepas,Cheesecake,Fish and Chips
TGI Friday's	Cassoulet
Wendy's	Arepas,Cheesecake

```
     Time       Action                                  Response            Duration / Fetch Time
1    21:36:38   SELECT * FROM dishesServed LIMIT 0, 1000   18 row(s) returned   0.0017 sec / 0.00000...
```

System response to inserting data:



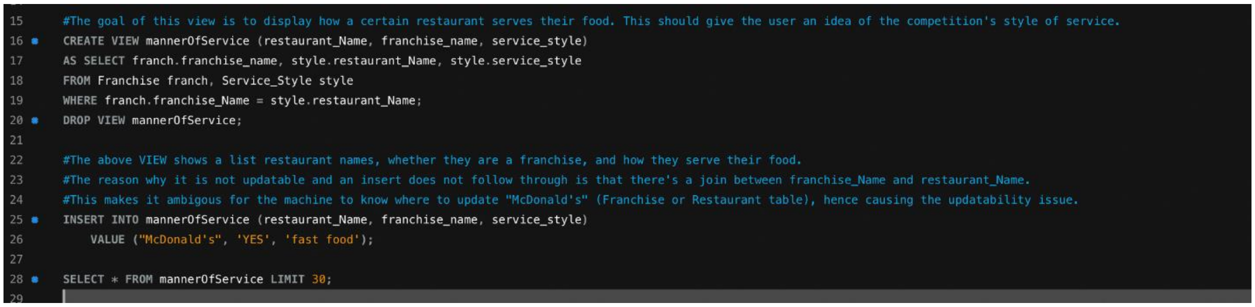
```
#mySQL cannot directly find the specific line to update due to this aggregation. Therefore, it is not updatable.
INSERT INTO dishesServed
    VALUES ('casa_manioca','pacoca', 'Brazil');
```

```
1:10
Action Output
     Time      Action                                                        Response                                                                  Duration / Fetch Time
✖ 1  21:47:29  INSERT INTO dishesServed VALUES ('casa_manioca','pacoca', 'Brazil')  Error Code: 1471. The target table dishesServed of the INSERT is not insertable-into  0.00037 sec
```

The above VIEW shows a list of foods for each restaurant so long as their comes from Kenya.

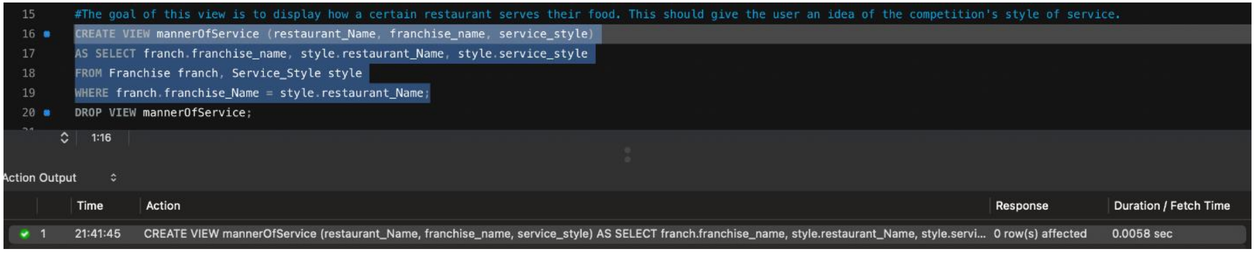
MySQL cannot directly find the specific line to update due to this aggregation. Therefore, it is not updatable.

Second view:



```
#The goal of this view is to display how a certain restaurant serves their food. This should give the user an idea of the competition's style of service.
CREATE VIEW mannerOfService (restaurant_Name, franchise_name, service_style)
AS SELECT franch.franchise_name, style.restaurant_Name, style.service_style
FROM Franchise franch, Service_Style style
WHERE franch.franchise_Name = style.restaurant_Name;
DROP VIEW mannerOfService;

#The above VIEW shows a list restaurant names, whether they are a franchise, and how they serve their food.
#The reason why it is not updatable and an insert does not follow through is that there's a join between franchise_Name and restaurant_Name.
#This makes it ambigous for the machine to know where to update "McDonald's" (Franchise or Restaurant table), hence causing the updatability issue.
INSERT INTO mannerOfService (restaurant_Name, franchise_name, service_style)
    VALUE ("McDonald's", 'YES', 'fast food');

SELECT * FROM mannerOfService LIMIT 30;
```

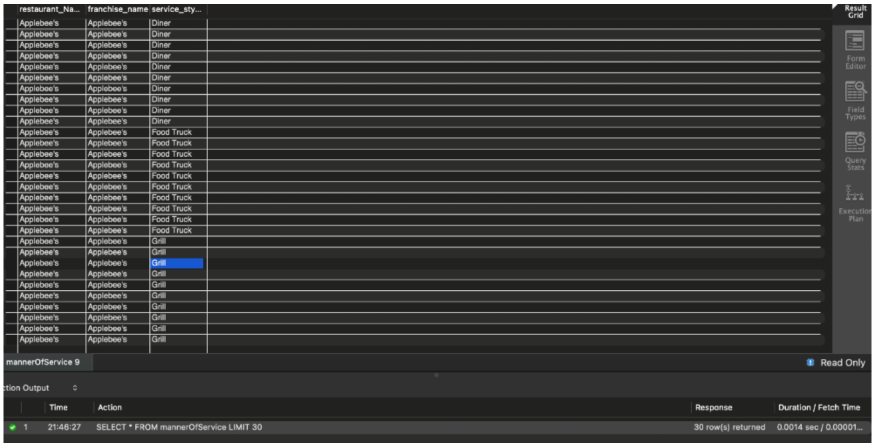
System response:



```
#The goal of this view is to display how a certain restaurant serves their food. This should give the user an idea of the competition's style of service.
CREATE VIEW mannerOfService (restaurant_Name, franchise_name, service_style)
AS SELECT franch.franchise_name, style.restaurant_Name, style.service_style
FROM Franchise franch, Service_Style style
WHERE franch.franchise_Name = style.restaurant_Name;
DROP VIEW mannerOfService;
```

```
1:16
Action Output
     Time      Action                                                                                                                               Response             Duration / Fetch Time
✓ 1  21:41:45  CREATE VIEW mannerOfService (restaurant_Name, franchise_name, service_style) AS SELECT franch.franchise_name, style.restaurant_Name, style.servi...  0 row(s) affected    0.0058 sec
```

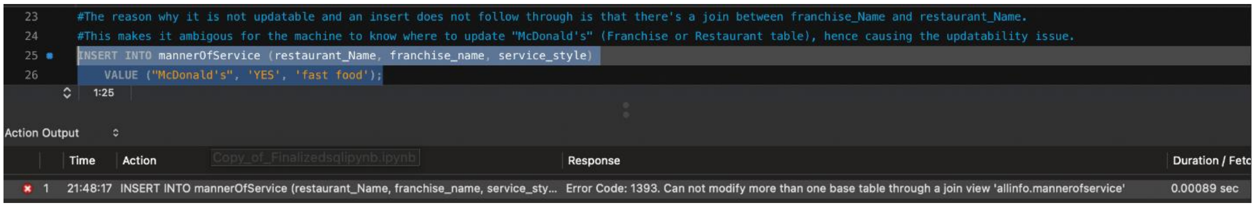
Query of the view:



restaurant_Na...	franchise_name	service_sty...
Applebee's	Applebee's	Diner
Applebee's	Applebee's	Diner
Applebee's	Applebee's	Diner
Applebee's	Applebee's	Diner
Applebee's	Applebee's	Diner
Applebee's	Applebee's	Diner
Applebee's	Applebee's	Diner
Applebee's	Applebee's	Diner
Applebee's	Applebee's	Diner
Applebee's	Applebee's	Diner
Applebee's	Applebee's	Food Truck
Applebee's	Applebee's	Food Truck
Applebee's	Applebee's	Food Truck
Applebee's	Applebee's	Food Truck
Applebee's	Applebee's	Food Truck
Applebee's	Applebee's	Food Truck
Applebee's	Applebee's	Food Truck
Applebee's	Applebee's	Food Truck
Applebee's	Applebee's	Food Truck
Applebee's	Applebee's	Food Truck
Applebee's	Applebee's	Grill
Applebee's	Applebee's	Grill
Applebee's	Applebee's	Grill
Applebee's	Applebee's	Grill
Applebee's	Applebee's	Grill
Applebee's	Applebee's	Grill
Applebee's	Applebee's	Grill
Applebee's	Applebee's	Grill
Applebee's	Applebee's	Grill
Applebee's	Applebee's	Grill

```
ction Output
     Time      Action                                  Response           Duration / Fetch Time
✓ 1  21:46:27  SELECT * FROM mannerOfService LIMIT 30  30 row(s) returned  0.0014 sec / 0.00001...
```

System response to inserting data:



```
#The reason why it is not updatable and an insert does not follow through is that there's a join between franchise_Name and restaurant_Name.
#This makes it ambigous for the machine to know where to update "McDonald's" (Franchise or Restaurant table), hence causing the updatability issue.
INSERT INTO mannerOfService (restaurant_Name, franchise_name, service_style)
    VALUE ("McDonald's", 'YES', 'fast food');
```

```
    Time      Action                                                         Response                                                                                   Duration / Fetc
✖ 1 21:48:17  INSERT INTO mannerOfService (restaurant_Name, franchise_name, service_sty...  Error Code: 1393. Can not modify more than one base table through a join view 'allinfo.mannerofservice'  0.00089 sec
```

The reason why it is not updatable and an insert does not follow through is that there's a join between franchise_Name and restaurant_Name.

This makes it ambigous for the machine to know where to update "McDonald's" (Franchise or Restaurant table), hence causing the updatability issue.