

Database Population and SQL Queries (Milestone 3)

Queries and Descriptions

Webpage 1. Business searcher & recommender

1.1 Business searcher

A. Provide a list of businesses that meet users' selection criteria

Description	Users select state and city (required parameters) and input business name, category, postal code, rating star range, price range (optional parameters), return a list of businesses with their information that meet the selection criteria, sorted by descending order of rating stars and review count.																																																		
Input parameters	state, city (required parameters) business name category, postal code, rating star range, price range (optional parameters)																																																		
SQL	<pre>SELECT business_id, name, address, city, state, postal_code, stars, review_count, categories, RestaurantsPriceRange2 as price_range FROM Business WHERE Name like '%starbucks%' AND postal_code='97202' AND city = 'Portland' AND state = 'OR' AND stars>= 3 AND stars<=5 AND RestaurantsPriceRange2>=1 AND RestaurantsPriceRange2<=3 ORDER BY stars DESC, review_count DESC;</pre>																																																		
Result sample	<table><tr><th>business_id</th><th>name</th><th>address</th><th>city</th><th>state</th><th>postal_code</th><th>stars</th><th>review_count</th><th>categories</th><th>price_range</th></tr><tr><td>6YJLzNqyyXGwwfsp...</td><td>Starbucks</td><td>1305 SE Tacoma St</td><td>Portland</td><td>OR</td><td>97202</td><td>3.5</td><td>22</td><td>Coffee & Tea, Food</td><td>1</td></tr><tr><td>mAvk5jvzTWTPsb0x...</td><td>Starbucks</td><td>4437 SE 39th Ave, A</td><td>Portland</td><td>OR</td><td>97202</td><td>3.5</td><td>19</td><td>Coffee & Tea, Food</td><td>2</td></tr><tr><td>0FKrWzTdBLFEQwFW...</td><td>Starbucks</td><td>7001 SE Milwaukee Ave</td><td>Portland</td><td>OR</td><td>97202</td><td>3.5</td><td>17</td><td>Food, Coffee & Tea</td><td>1</td></tr><tr><td>2yDk3cEj2C0Ut2dC...</td><td>Starbucks</td><td>3623 SE Powell</td><td>Portland</td><td>OR</td><td>97202</td><td>3</td><td>37</td><td>Food, Coffee & Tea</td><td>2</td></tr></table>	business_id	name	address	city	state	postal_code	stars	review_count	categories	price_range	6YJLzNqyyXGwwfsp...	Starbucks	1305 SE Tacoma St	Portland	OR	97202	3.5	22	Coffee & Tea, Food	1	mAvk5jvzTWTPsb0x...	Starbucks	4437 SE 39th Ave, A	Portland	OR	97202	3.5	19	Coffee & Tea, Food	2	0FKrWzTdBLFEQwFW...	Starbucks	7001 SE Milwaukee Ave	Portland	OR	97202	3.5	17	Food, Coffee & Tea	1	2yDk3cEj2C0Ut2dC...	Starbucks	3623 SE Powell	Portland	OR	97202	3	37	Food, Coffee & Tea	2
business_id	name	address	city	state	postal_code	stars	review_count	categories	price_range																																										
6YJLzNqyyXGwwfsp...	Starbucks	1305 SE Tacoma St	Portland	OR	97202	3.5	22	Coffee & Tea, Food	1																																										
mAvk5jvzTWTPsb0x...	Starbucks	4437 SE 39th Ave, A	Portland	OR	97202	3.5	19	Coffee & Tea, Food	2																																										
0FKrWzTdBLFEQwFW...	Starbucks	7001 SE Milwaukee Ave	Portland	OR	97202	3.5	17	Food, Coffee & Tea	1																																										
2yDk3cEj2C0Ut2dC...	Starbucks	3623 SE Powell	Portland	OR	97202	3	37	Food, Coffee & Tea	2																																										

B. Provide detailed information of the user selected business

Description	Users selected a specific business and this query will provide detailed information of this business and its photos.
Input parameters	Business_id
SQL	<pre>WITH num_photos (business_id, num_photo) AS (SELECT business_id, count(*) as num_photo from photo WHERE business_id = ' _--ScmaNumIoT2gQanACvg') SELECT B.business_id, name, categories, RestaurantsPriceRange2 as price_range, stars, review_count, is_open, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, address, city, state, postal_code, RestaurantsTakeOut, garage, lot, street, valet, validated, photo_id, caption, label, num_photo FROM Business B join photo P on B.business_id = P.business_id join num_photos N on P.business_id = N.business_id WHERE B.business_id = ' _--ScmaNumIoT2gQanACvg';</pre>

SQL	<pre> WITH num_photos (business_id, num_photo) AS (SELECT business_id, count(*) as num_photo from photo WHERE business_id = 'IVl8FC2S9ICn3aTSsaBmVQ') SELECT B.business_id, name, categories, RestaurantsPriceRange2 as price_range, stars, review_count, is_open, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, address, city, state, postal_code, RestaurantsTakeOut, garage, lot, street, valet, validated, photo_id, caption, label, num_photo FROM Business B join photo P on B.business_id = P.business_id join num_photos N on P.business_id = N.business_id WHERE B.business_id = 'IVl8FC2S9ICn3aTSsaBmVQ'; </pre>
Result sample	

Webpage 2. Find Users' N-Connection










2.1 User Login.

Description	User input Name and last 6 digit of UID to login, return the full user_ID for other query.
Input parameters	Name, last 6 digit of UID
SQL	<pre> SELECT user_id FROM user WHERE name="Don" AND user_id LIKE "%0ZxoUw"; </pre>
Result sample	

2.2 Users' favorite Business

Description	Use returned user_ID from User Login to search the user's 5 star reviewed business
Input parameters	user_ID (from User Login)
SQL	<pre> SELECT DISTINCT RP.business_id, Business.name, Business.address, Business.city, Business.State, Business.new_categories FROM review_Portland RP JOIN Business ON RP.business_id=Business.business_id WHERE RP.user_id="Pf7FI00ukC_CEcCz0ZxoUw" AND RP.stars=5 ; </pre>
Result sample	

2.3 Business related 1-connection and 2-connection

Description	User select one business, and search for the business related 1-connection and 2-connection friends												
Input parameters	Business_ID(from User favorite Business), user_ID(from User Login)												
SQL	<pre>WITH ONE AS (SELECT DISTINCT RP.user_id, user.name, 1 AS N FROM review_Portland RP JOIN user ON RP.user_id = user.user_id WHERE RP.stars = 5 AND RP.business_id = "LfLNQC_C7xlnC_ojHMI7fw" AND RP.user_id!="Pf7FI0OukC_CEcCz0ZxoUw"), TWO_B AS(SELECT DISTINCT review_Portland.business_id FROM ONE LEFT JOIN review_Portland ON review_Portland.user_id=ONE.user_id WHERE review_Portland.stars=5), TWO AS (SELECT DISTINCT review_Portland.user_id, user.name, 2 AS N FROM review_Portland JOIN TWO_B ON review_Portland.business_id=TWO_B.business_id JOIN user ON review_Portland.user_id= user.user_id WHERE review_Portland.stars=5 AND review_Portland.user_id!="Pf7FI0OukC_CEcCz0ZxoUw"), TOT AS (SELECT * FROM ONE UNION ALL SELECT * FROM TWO) SELECT user_id, name, MIN(N) AS N FROM TOT GROUP BY user_id ORDER BY N;</pre>												
Result sample	<table><thead><tr><th></th><th> user_id</th><th> name</th><th> N</th></tr></thead><tbody><tr><td>1</td><td>xKtauW_XxIDByQdrIDkPdg</td><td>Kimberly</td><td>1</td></tr><tr><td>2</td><td>hmmzEyrKx8eo1jK1XwDQ7w</td><td>Sam</td><td>1</td></tr></tbody></table>		 user_id	 name	 N	1	xKtauW_XxIDByQdrIDkPdg	Kimberly	1	2	hmmzEyrKx8eo1jK1XwDQ7w	Sam	1
	 user_id	 name	 N										
1	xKtauW_XxIDByQdrIDkPdg	Kimberly	1										
2	hmmzEyrKx8eo1jK1XwDQ7w	Sam	1										

Webpage 3. Business Scientist

3.1 Review star frequency distribution (count & percentage) at different geographic level

A. State's review star frequency distribution by city

Description	User select one State, export the star distribution of each city in the state
Input parameters	state name

SQL	<pre>SELECT city, SUM(IF(stars<2,1,0)) AS 1star_count,SUM(IF(stars<3 AND stars>=2,1,0)) AS 2star_count,SUM(IF(stars<4 AND stars>=3,1,0)) AS 3star_count, SUM(IF(stars<5 AND stars>=4,1,0)) AS 4star_count,SUM(IF(stars=5,1,0)) AS 5star_count,SUM(IF(stars<2,1,0))/SUM(IF(stars>0,1,0)) AS 1star_percent, SUM(IF(stars<3 AND stars>=2,1,0))/SUM(IF(stars>0,1,0)) AS 2star_percent, SUM(IF(stars<4 AND stars>=3,1,0))/SUM(IF(stars>0,1,0)) AS 3star_percent,SUM(IF(stars<5 AND stars>=4,1,0))/SUM(IF(stars>0,1,0)) AS 4star_percent, SUM(IF(stars=5,1,0))/SUM(IF(stars>0,1,0)) AS 5star_percent FROM Business WHERE state="MA" GROUP BY city;</pre>																																										
Result sample	<table><tr><th>city</th><th>1star_count</th><th>2star_count</th><th>3star_count</th><th>4star_count</th><th>5star_count</th><th>1star_percent</th></tr><tr><td>Somerville</td><td>10</td><td>87</td><td>293</td><td>423</td><td>89</td><td>0.0111</td></tr><tr><td>Concord</td><td>3</td><td>14</td><td>51</td><td>62</td><td>16</td><td>0.0205</td></tr><tr><td>Lynn</td><td>18</td><td>38</td><td>99</td><td>96</td><td>31</td><td>0.0638</td></tr><tr><td>Cambridge</td><td>15</td><td>198</td><td>635</td><td>750</td><td>161</td><td>0.0085</td></tr><tr><td>Beverly</td><td>9</td><td>54</td><td>96</td><td>156</td><td>44</td><td>0.0251</td></tr></table>	city	1star_count	2star_count	3star_count	4star_count	5star_count	1star_percent	Somerville	10	87	293	423	89	0.0111	Concord	3	14	51	62	16	0.0205	Lynn	18	38	99	96	31	0.0638	Cambridge	15	198	635	750	161	0.0085	Beverly	9	54	96	156	44	0.0251
city	1star_count	2star_count	3star_count	4star_count	5star_count	1star_percent																																					
Somerville	10	87	293	423	89	0.0111																																					
Concord	3	14	51	62	16	0.0205																																					
Lynn	18	38	99	96	31	0.0638																																					
Cambridge	15	198	635	750	161	0.0085																																					
Beverly	9	54	96	156	44	0.0251																																					

B. City's review star frequency distribution by zip

Description	User select one city, export the star distribution of each zip area in the city																																										
Input parameters	city name																																										
SQL	<pre>SELECT postal_code, SUM(IF(stars<2,1,0)) AS 1star_count,SUM(IF(stars<3 AND stars>=2,1,0)) AS 2star_count,SUM(IF(stars<4 AND stars>=3,1,0)) AS 3star_count, SUM(IF(stars<5 AND stars>=4,1,0)) AS 4star_count,SUM(IF(stars=5,1,0)) AS 5star_count,SUM(IF(stars<2,1,0))/SUM(IF(stars>0,1,0)) AS 1star_percent, SUM(IF(stars<3 AND stars>=2,1,0))/SUM(IF(stars>0,1,0)) AS 2star_percent, SUM(IF(stars<4 AND stars>=3,1,0))/SUM(IF(stars>0,1,0)) AS 3star_percent,SUM(IF(stars<5 AND stars>=4,1,0))/SUM(IF(stars>0,1,0)) AS 4star_percent, SUM(IF(stars=5,1,0))/SUM(IF(stars>0,1,0)) AS 5star_percent FROM Business WHERE city="Somerville" GROUP BY postal_code;</pre>																																										
Result sample	<table><tr><th>postal_code</th><th>'1star_count'</th><th>'2star_count'</th><th>'3star_count'</th><th>'4star_count'</th><th>'5star_count'</th><th>'1star_percent'</th></tr><tr><td>02143</td><td>2</td><td>35</td><td>109</td><td>178</td><td>50</td><td>0.0053</td></tr><tr><td>02144</td><td>3</td><td>27</td><td>102</td><td>144</td><td>21</td><td>0.0101</td></tr><tr><td>02145</td><td>5</td><td>25</td><td>80</td><td>96</td><td>15</td><td>0.0226</td></tr><tr><td>02141</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0.0000</td></tr><tr><td>02130</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0.0000</td></tr></table>	postal_code	'1star_count'	'2star_count'	'3star_count'	'4star_count'	'5star_count'	'1star_percent'	02143	2	35	109	178	50	0.0053	02144	3	27	102	144	21	0.0101	02145	5	25	80	96	15	0.0226	02141	0	0	0	0	1	0.0000	02130	0	0	1	0	0	0.0000
postal_code	'1star_count'	'2star_count'	'3star_count'	'4star_count'	'5star_count'	'1star_percent'																																					
02143	2	35	109	178	50	0.0053																																					
02144	3	27	102	144	21	0.0101																																					
02145	5	25	80	96	15	0.0226																																					
02141	0	0	0	0	1	0.0000																																					
02130	0	0	1	0	0	0.0000																																					

C. Zip level review star distribution (individual zip)

Description	User select one postal code, export the star distribution
Input parameters	zip code

SQL	<pre>SELECT stars, COUNT(*) AS count, COUNT(*) / SUM(COUNT(*) OVER ()) AS percent FROM Business WHERE postal_code="2143" GROUP BY stars ORDER BY stars;</pre>																								
Result sample	<table><tr><th>stars</th><th>count</th><th>percent</th></tr><tr><td>1.5</td><td>2</td><td>0.0053</td></tr><tr><td>2</td><td>13</td><td>0.0345</td></tr><tr><td>2.5</td><td>24</td><td>0.0637</td></tr><tr><td>3</td><td>37</td><td>0.0981</td></tr><tr><td>3.5</td><td>73</td><td>0.1936</td></tr><tr><td>4</td><td>90</td><td>0.2387</td></tr><tr><td>4.5</td><td>88</td><td>0.2334</td></tr></table>	stars	count	percent	1.5	2	0.0053	2	13	0.0345	2.5	24	0.0637	3	37	0.0981	3.5	73	0.1936	4	90	0.2387	4.5	88	0.2334
stars	count	percent																							
1.5	2	0.0053																							
2	13	0.0345																							
2.5	24	0.0637																							
3	37	0.0981																							
3.5	73	0.1936																							
4	90	0.2387																							
4.5	88	0.2334																							

Note: A-C in 3.1 together will be a function module for the web

3.2 Price range frequency distribution (count & percentage) at different geographic level

A. State's price range frequency distribution by city

Description	User select one state, export the business price distribution of each city in the state																																				
Input parameters	state name																																				
SQL	<pre>SELECT city, SUM(IF(RestaurantsPriceRange2=1,1,0)) AS 1price_count, SUM(IF(RestaurantsPriceRange2=2,1,0)) AS 2price_count, SUM(IF(RestaurantsPriceRange2=3,1,0)) AS 3price_count, SUM(IF(RestaurantsPriceRange2=4,1,0)) AS 4price_count, SUM(IF(RestaurantsPriceRange2=1,1,0)) / SUM(IF(RestaurantsPriceRange2>0,1,0)) AS 1price_percent, SUM(IF(RestaurantsPriceRange2=2,1,0)) / SUM(IF(RestaurantsPriceRange2>0,1,0)) AS 2price_percent, SUM(IF(RestaurantsPriceRange2=3,1,0)) / SUM(IF(RestaurantsPriceRange2>0,1,0)) AS 3price_percent, SUM(IF(RestaurantsPriceRange2=4,1,0)) / SUM(IF(RestaurantsPriceRange2>0,1,0)) AS 4price_percent FROM Business WHERE State="MA" GROUP BY city HAVING SUM(IF(RestaurantsPriceRange2>0,1,0))>0;</pre>																																				
Result sample	<table><tr><th>city</th><th>'1price_count'</th><th>'2price_count'</th><th>'3price_count'</th><th>'4price_count'</th><th>'1price_percent'</th></tr><tr><td>Somerville</td><td>220</td><td>362</td><td>32</td><td>5</td><td>0.3554</td></tr><tr><td>Concord</td><td>12</td><td>65</td><td>20</td><td>3</td><td>0.1200</td></tr><tr><td>Lynn</td><td>101</td><td>64</td><td>11</td><td>1</td><td>0.5706</td></tr><tr><td>Cambridge</td><td>353</td><td>741</td><td>175</td><td>19</td><td>0.2741</td></tr><tr><td>Beverly</td><td>81</td><td>119</td><td>20</td><td>3</td><td>0.3632</td></tr></table>	city	'1price_count'	'2price_count'	'3price_count'	'4price_count'	'1price_percent'	Somerville	220	362	32	5	0.3554	Concord	12	65	20	3	0.1200	Lynn	101	64	11	1	0.5706	Cambridge	353	741	175	19	0.2741	Beverly	81	119	20	3	0.3632
city	'1price_count'	'2price_count'	'3price_count'	'4price_count'	'1price_percent'																																
Somerville	220	362	32	5	0.3554																																
Concord	12	65	20	3	0.1200																																
Lynn	101	64	11	1	0.5706																																
Cambridge	353	741	175	19	0.2741																																
Beverly	81	119	20	3	0.3632																																

B. City's price range frequency distribution by zip

Description	User select one city, export the business price distribution of each zip area in the city
Input parameters	city name

SQL	<pre>SELECT postal_code, SUM(IF(RestaurantsPriceRange2=1,1,0)) AS 1price_count, SUM(IF(RestaurantsPriceRange2=2,1,0)) AS 2price_count, SUM(IF(RestaurantsPriceRange2=3,1,0)) AS 3price_count, SUM(IF(RestaurantsPriceRange2=4,1,0)) AS 4price_count, SUM(IF(RestaurantsPriceRange2=1,1,0))/SUM(IF(RestaurantsPriceRange2>0,1,0)) AS 1price_percent, SUM(IF(RestaurantsPriceRange2=2,1,0))/SUM(IF(RestaurantsPriceRange2>0,1,0)) AS 2price_percent, SUM(IF(RestaurantsPriceRange2=3,1,0))/SUM(IF(RestaurantsPriceRange2>0,1,0)) AS 3price_percent, SUM(IF(RestaurantsPriceRange2=4,1,0))/SUM(IF(RestaurantsPriceRange2>0,1,0)) AS 4price_percent FROM Business WHERE city="Somerville" GROUP BY postal_code HAVING SUM(IF(RestaurantsPriceRange2>0,1,0))>0;</pre>																																				
Result sample	<table><tr><th>postal_code</th><th>1price_count</th><th>2price_count</th><th>3price_count</th><th>4price_count</th><th>1price_percent</th></tr><tr><td>02143</td><td>79</td><td>141</td><td>19</td><td>3</td><td>0.3264</td></tr><tr><td>02144</td><td>82</td><td>128</td><td>9</td><td>1</td><td>0.3727</td></tr><tr><td>02145</td><td>58</td><td>90</td><td>4</td><td>1</td><td>0.3791</td></tr><tr><td>02141</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0.0000</td></tr><tr><td>02114</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0.0000</td></tr></table>	postal_code	1price_count	2price_count	3price_count	4price_count	1price_percent	02143	79	141	19	3	0.3264	02144	82	128	9	1	0.3727	02145	58	90	4	1	0.3791	02141	0	1	0	0	0.0000	02114	0	1	0	0	0.0000
postal_code	1price_count	2price_count	3price_count	4price_count	1price_percent																																
02143	79	141	19	3	0.3264																																
02144	82	128	9	1	0.3727																																
02145	58	90	4	1	0.3791																																
02141	0	1	0	0	0.0000																																
02114	0	1	0	0	0.0000																																

C. Zip level price range frequency distribution (individual zip)

Description	User select one zip, export the business price distribution										
Input parameters	zip code										
SQL	<pre> SELECT RestaurantsPriceRange2, COUNT(RestaurantsPriceRange2) / SUM(COUNT(RestaurantsPriceRange2)) OVER () AS percent FROM Business WHERE postal_code="02143" AND RestaurantsPriceRange2 IS NOT NULL GROUP BY RestaurantsPriceRange2 ORDER BY RestaurantsPriceRange2; </pre>										
Result sample	<table> <tr> <th>RestaurantsPriceRange2</th><th>percent</th></tr> <tr> <td>1</td><td>0.3320</td></tr> <tr> <td>2</td><td>0.5779</td></tr> <tr> <td>3</td><td>0.0779</td></tr> <tr> <td>4</td><td>0.0123</td></tr> </table>	RestaurantsPriceRange2	percent	1	0.3320	2	0.5779	3	0.0779	4	0.0123
RestaurantsPriceRange2	percent										
1	0.3320										
2	0.5779										
3	0.0779										
4	0.0123										

Note: A-C in 3.2 together will be a function module for the web










3.3 Average price and review star summary

Description	Summary by states, cities, or zip codes, to get the average price and review star of business in each area
Input parameters	None

SQL	<pre>SELECT postal_code, COUNT(stars) AS num,AVG(RestaurantsPriceRange2) AS avg_price, AVG(stars) AS avg_review FROM Business GROUP BY postal_code ORDER BY num DESC; ##### SELECT city, COUNT(stars) AS num,AVG(RestaurantsPriceRange2) AS avg_price, AVG(stars) AS avg_review FROM Business GROUP BY city ORDER BY num DESC; ##### SELECT state, COUNT(stars) AS num,AVG(RestaurantsPriceRange2) AS avg_price, AVG(stars) AS avg_review FROM Business GROUP BY state ORDER BY num DESC;</pre>																																																												
Result sample	<table><tr><th>postal_code</th><th>num</th><th>avg_price</th><th>avg_review</th></tr><tr><td>97214</td><td>1375</td><td>1.7828</td><td>4.19418</td></tr><tr><td>02116</td><td>1078</td><td>2.3460</td><td>3.75139</td></tr><tr><td>97209</td><td>1062</td><td>1.9987</td><td>4.09087</td></tr><tr><td>97202</td><td>928</td><td>1.6651</td><td>4.16433</td></tr></table> <table><tr><th>city</th><th>num</th><th>avg_price</th><th>avg_review</th></tr><tr><td>Portland</td><td>13614</td><td>1.7217</td><td>3.98046</td></tr><tr><td>Boston</td><td>5638</td><td>1.9531</td><td>3.70105</td></tr><tr><td>Cambridge</td><td>1759</td><td>1.8913</td><td>3.71717</td></tr><tr><td>Beaverton</td><td>1721</td><td>1.7510</td><td>3.72719</td></tr><tr><td>Somerville</td><td>902</td><td>1.7124</td><td>3.801</td></tr><tr><td>Quincy</td><td>759</td><td>1.6913</td><td>3.63439</td></tr></table> <table><tr><th>state</th><th>num</th><th>avg_price</th><th>avg_review</th></tr><tr><td>1 MA</td><td>25782</td><td>1.8558</td><td>3.6783</td></tr><tr><td>2 OR</td><td>18991</td><td>1.7356</td><td>3.91327</td></tr></table>	postal_code	num	avg_price	avg_review	97214	1375	1.7828	4.19418	02116	1078	2.3460	3.75139	97209	1062	1.9987	4.09087	97202	928	1.6651	4.16433	city	num	avg_price	avg_review	Portland	13614	1.7217	3.98046	Boston	5638	1.9531	3.70105	Cambridge	1759	1.8913	3.71717	Beaverton	1721	1.7510	3.72719	Somerville	902	1.7124	3.801	Quincy	759	1.6913	3.63439	state	num	avg_price	avg_review	1 MA	25782	1.8558	3.6783	2 OR	18991	1.7356	3.91327
postal_code	num	avg_price	avg_review																																																										
97214	1375	1.7828	4.19418																																																										
02116	1078	2.3460	3.75139																																																										
97209	1062	1.9987	4.09087																																																										
97202	928	1.6651	4.16433																																																										
city	num	avg_price	avg_review																																																										
Portland	13614	1.7217	3.98046																																																										
Boston	5638	1.9531	3.70105																																																										
Cambridge	1759	1.8913	3.71717																																																										
Beaverton	1721	1.7510	3.72719																																																										
Somerville	902	1.7124	3.801																																																										
Quincy	759	1.6913	3.63439																																																										
state	num	avg_price	avg_review																																																										
1 MA	25782	1.8558	3.6783																																																										
2 OR	18991	1.7356	3.91327																																																										

3.4 Business categories frequency distribution in a given geographic area

Description	Summary by states, cities, or zip codes, to get the percentage of each different categories of business
Input parameters	State, or city, or zip
SQL	<pre> SELECT new_categories, COUNT(*) AS count, COUNT(*) / SUM(COUNT(*)) OVER () AS percent FROM Business WHERE postal_code="02143" GROUP BY new_categories ORDER BY count DESC ; ##### </pre>

	<pre>SELECT new_categories, COUNT(*) AS count, COUNT(*) / SUM(COUNT(*)) OVER () AS percent FROM Business WHERE city="Portland" GROUP BY new_categories ORDER BY count DESC ; ##### SELECT new_categories, COUNT(*) AS count, COUNT(*) / SUM(COUNT(*)) OVER () AS percent FROM Business WHERE State="MA" GROUP BY new_categories ORDER BY count DESC ; </pre>																					
Result sample	<table><tr><th> new_categories ÷</th><th> count ÷</th><th> percent ÷</th></tr><tr><td>Restaurants</td><td>129</td><td>0.3422</td></tr><tr><td>Shopping</td><td>56</td><td>0.1485</td></tr><tr><td>Food</td><td>43</td><td>0.1141</td></tr><tr><td>Local Services</td><td>31</td><td>0.0822</td></tr><tr><td>Automotive</td><td>30</td><td>0.0796</td></tr><tr><td>Beauty & Spas</td><td>24</td><td>0.0637</td></tr></table>	 new_categories ÷	 count ÷	 percent ÷	Restaurants	129	0.3422	Shopping	56	0.1485	Food	43	0.1141	Local Services	31	0.0822	Automotive	30	0.0796	Beauty & Spas	24	0.0637
 new_categories ÷	 count ÷	 percent ÷																				
Restaurants	129	0.3422																				
Shopping	56	0.1485																				
Food	43	0.1141																				
Local Services	31	0.0822																				
Automotive	30	0.0796																				
Beauty & Spas	24	0.0637																				

3.5 County health rankings and review star (or price range) frequency distribution within a given state

Description	User selected a specific state and the query will provide the health ranking, frequency distribution (i.e. count and percentage) of businesses' stars and price levels by county within the state.
Input parameters	state
SQL	<pre> WITH H (business_id, price_range, stars, type, zip, city, state, fips, county, num_ranked_counties, health_factors_rank) AS (SELECT B.business_id, RestaurantsPriceRange2, stars, new_categories, B.postal_code, B.city, B.state, R.fips, R.county, R.number_of_ranked_counties, R.health_factors_rank FROM Business B join Zip_county_crosswalk Z on B.postal_code = Z.zip AND B.city = Z.city AND B.state = Z.state join County_health_ranking R on Z.fips = R.fips WHERE B.state = 'MA') SELECT H.state, H.num_ranked_counties, H.fips, H.county, H.health_factors_rank, SUM(IF(H.price_range = 1, 1, 0)) as price1_count, SUM(IF(H.price_range = 2, 1, 0)) as price2_count, SUM(IF(H.price_range = 3, 1, 0)) as price3_count, SUM(IF(H.price_range = 4, 1, 0)) as price4_count, SUM(IF(H.price_range >0, 1, 0)) as price_count, SUM(IF(H.price_range = 1, 1, 0))/SUM(IF(H.price_range >0, 1, 0)) as price1_pct, </pre>

	<pre>SUM(IF(H.price_range = 2, 1, 0))/SUM(IF(H.price_range >0, 1, 0)) as price2_pct, SUM(IF(H.price_range = 3, 1, 0))/SUM(IF(H.price_range >0, 1, 0)) as price3_pct, SUM(IF(H.price_range = 4, 1, 0))/SUM(IF(H.price_range >0, 1, 0)) as price4_pct, SUM(IF(H.stars <= 1, 1, 0)) as star0_1_count, SUM(IF(H.stars>1 AND stars<=2, 1, 0)) as star1_2_count, SUM(IF(H.stars>2 AND stars<=3, 1, 0)) as star2_3_count, SUM(IF(H.stars>3 AND stars<=4, 1, 0)) as star3_4_count, SUM(IF(H.stars>4 AND stars<=5, 1, 0)) as star4_5_count, SUM(IF(H.stars >0, 1, 0)) as star_count, SUM(IF(H.stars <= 1, 1, 0))/SUM(IF(H.stars >0, 1, 0)) as star0_1_pct, SUM(IF(H.stars>1 AND stars<=2, 1, 0))/SUM(IF(H.stars >0, 1, 0)) as star1_2_pct, SUM(IF(H.stars>2 AND stars<=3, 1, 0))/SUM(IF(H.stars >0, 1, 0)) as star2_3_pct, SUM(IF(H.stars>3 AND stars<=4, 1, 0))/SUM(IF(H.stars >0, 1, 0)) as star3_4_pct, SUM(IF(H.stars>4 AND stars<=5, 1, 0))/SUM(IF(H.stars >0, 1, 0)) as star4_5_pct FROM H GROUP BY H.state, H.fips, H.county, H.health_factors_rank ORDER BY H.state, H.health_factors_rank ;</pre>																																																												
Result sample	<table><tr><th></th><th>state</th><th>num_ranked_counties</th><th>fips</th><th>county</th><th>health_factors_rank</th><th>price1_pct</th><th>price2_pct</th><th>price3_pct</th><th>price4_pct</th></tr><tr><td>1</td><td>MA</td><td>14</td><td>25021</td><td>Norfolk</td><td>1</td><td>0.2716</td><td>0.6079</td><td>0.1059</td><td>0.0146</td></tr><tr><td>2</td><td>MA</td><td>14</td><td>25017</td><td>Middlesex</td><td>2</td><td>0.2875</td><td>0.5984</td><td>0.0989</td><td>0.0151</td></tr><tr><td>3</td><td>MA</td><td>14</td><td>25023</td><td>Plymouth</td><td>7</td><td>0.2838</td><td>0.5904</td><td>0.1919</td><td>0.0148</td></tr><tr><td>4</td><td>MA</td><td>14</td><td>25009</td><td>Essex</td><td>9</td><td>0.3078</td><td>0.5864</td><td>0.0935</td><td>0.0123</td></tr><tr><td>5</td><td>MA</td><td>14</td><td>25025</td><td>Suffolk</td><td>12</td><td>0.3035</td><td>0.5258</td><td>0.1467</td><td>0.0247</td></tr></table>		state	num_ranked_counties	fips	county	health_factors_rank	price1_pct	price2_pct	price3_pct	price4_pct	1	MA	14	25021	Norfolk	1	0.2716	0.6079	0.1059	0.0146	2	MA	14	25017	Middlesex	2	0.2875	0.5984	0.0989	0.0151	3	MA	14	25023	Plymouth	7	0.2838	0.5904	0.1919	0.0148	4	MA	14	25009	Essex	9	0.3078	0.5864	0.0935	0.0123	5	MA	14	25025	Suffolk	12	0.3035	0.5258	0.1467	0.0247
	state	num_ranked_counties	fips	county	health_factors_rank	price1_pct	price2_pct	price3_pct	price4_pct																																																				
1	MA	14	25021	Norfolk	1	0.2716	0.6079	0.1059	0.0146																																																				
2	MA	14	25017	Middlesex	2	0.2875	0.5984	0.0989	0.0151																																																				
3	MA	14	25023	Plymouth	7	0.2838	0.5904	0.1919	0.0148																																																				
4	MA	14	25009	Essex	9	0.3078	0.5864	0.0935	0.0123																																																				
5	MA	14	25025	Suffolk	12	0.3035	0.5258	0.1467	0.0247																																																				

Credentials

- We use Amazon AWS database (MySQL) to host our data and below is the information to access to the database.

```

{
  "rds_host": "database-newbee.cmya8s5xkkje.us-east-1.rds.amazonaws.com",
  "rds_port": "3306",
  "rds_user": "newbeedata",
  "rds_password": "newbeel234",
  "rds_db": "Yelp_project",
  "server_host": "127.0.0.1",
  "server_port": "8080"
}

```

- We use Amazon S3 to host our photos and below is a link of sample photo:
<https://yelpphoto.s3.amazonaws.com/CCbMJ0qYIYAB3GJ8DA-pFg.jpg>