MLDS-413 Introduction to Databases and Information Retrieval

Homework 7: Regular expressions; Common Table Expressions; Recursive networks

Name 1: Ayush Agarwal

NetID 1: scg1143

Name 2: Lowan(Sydney) Li

NetID 2: chr0390

Instructions

You should submit this homework assignment via Canvas. Acceptable formats are word files, text files, and pdf files. Paper submissions are not allowed and they will receive an automatic zero.

As explained during lecture and in the syllabus, assignments are done in groups. The groups have been created and assigned. Each group needs to submit only one assignment (i.e., there is no need for both partners to submit individually the same homework assignment).

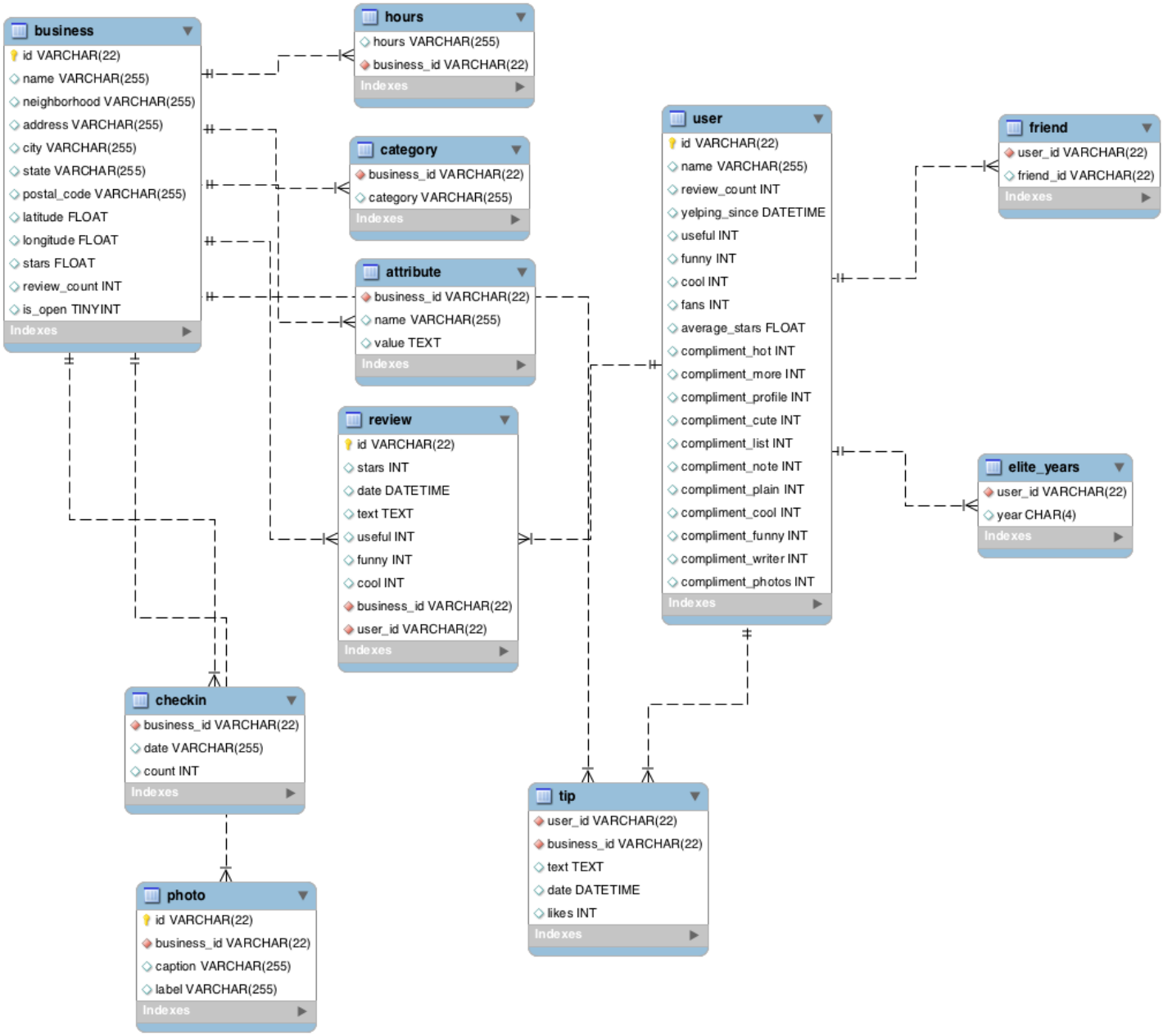
Each group can submit solutions multiple times (for example, you may discover an error in your earlier submission and choose to submit a new solution set). We will grade only the last submission and ignore earlier ones.

Make sure you submit your solutions before the deadline. The policies governing academic integrity, tardiness and penalties are detailed in the syllabus.

Yelp Database (yelp)

The database "yelp" has data from the Yelp business review app (http://yelp.com/). Please follow the instrutions from Homework 6 to connect to the yelp database on MSiA's Postgres server.

The database schema is provided below:



Note that the position of the linking lines does not directly indicate which columns are linked; there is no such requirement or standard for ER diagrams. You will need to infer which columns are the ones linking the tables.

You will use this database to answer the following questions. Unless otherwise noted, for each question please provide:

- The query you constructed
- The output of that query
- Any other information requested by the question (e.g., timing results)

1) **(10 points)** Find the name of the businesses for which there is a review that contains the case-**insensitive** text string "wing" at least 25 times in the same review. *Hint 1:* You do not have to search for complete words but only for **text strings** that are case-insensitive, i.e., "sunwing", "wing", "winging", "Wings", "WiNg" are all hits. *Hint 2*: The regular expressions format in PostgreSQL is different than the MySQL variant we discussed in class. PostgreSQL does pattern matching with regular expressions using the `SIMILAR TO` operator, instead of the `REGEXP` operator. In the `SIMILAR TO` operator "_" matches any character and "%" matches any sequence of zero or more characters. The remaining rules are similar to the ones we learned in class, e.g., parentheses "`( )`" are used to group items together into a single logical item, square brackets "`[ ]`" are used to denote a class of characters, angled brackets "`{ }`" are used to denote repetition, etc. The regular expressions syntax rules for PostgreSQL 10 can be found at Section 9.7.2 at https://www.postgresql.org/docs/10/functions-matching.html.

```
SELECT b.name
FROM public.business b
JOIN public.review r ON b.id = r.business_id
WHERE LOWER(r.text) SIMILAR TO '%(wing%){25,}%';
```

```
Ans:
- Wing Time
- Wingstop
- The Firehall Cool Bar Hot Grill
- Buffalo Wild Wings
- Puck'n Wings
```

	name
1	Wing Time
2	Wingstop
3	The Firehall Cool Bar Hot Grill
4	Buffalo Wild Wings
5	Puck'n Wings

2) **(10 points)** What is the name, address (including city, state, postal code), and **average** rating of the highest-rated **restaurant** with "McDonald" in its name? *Hint 1*: You must use the category named "`Restaurants`", otherwise you'll get results for other types of businesses with "`McDonald`" in the name. *Hint 2*: We are asking for the restaurant with the highest ***average*** rating. Many such restaurants have at least one 5-star rating, but only one location has a star rating **average** close to 5. *Hint 3*: You do not need to concatenate the address into a single string. It is OK for the address, city, state and postal code to occupy a separate column each in your result table.

```
select
   b.name,
   b.address ,
   b.city ,
   b.state ,
   b.postal_code ,
   AVG(r.stars) as average_rating
FROM
   business b
inner join
   category c on b.id = c.business_id
left join
```

```
   review r  on b.id = r.business_id
where
   c.category = 'Restaurants' and b.name like '%McDonald%'
group by
   b."id"
order by
   average_rating desc
limit 1;
```

Ans:

	name	address	city	state	postal_code	average_rating
1	McDonald's McCafe	100 King Street W, Exchange Tower	Toronto	ON	M5X 2A2	4.8333333333

3) **(10 points)** What are the names of the businesses for which there are at least 5 reviews where each one of these reviews contains the text "`barf`"? *Hint:* Similarly to question 2, you do not have to match individual words, but only sub-strings. For example, "barf", "barfing" and "barfday" should all be considered hits.

```
select b.name
from
   business b
inner join
   review r on b.id = r.business_id
where
   lower(r."text") like '%barf%'
group by
   b.id
having count(*) >= 5;
```

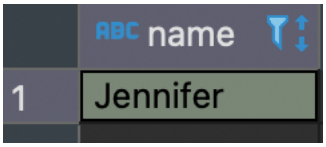
Ans:

	name
1	Barfly
2	Barfly
3	Spirit Airlines
4	Wicked Spoon
5	Barfitness

4) **(10 points)** With execution timing on, find the name of the user with id `'CxDOIDnH8gp9KXzpBHJYXw'`. Include the time it took to execute the query in your answer. *Note 1:* you may want to run this ~10 times and get the average timing across all runs to get a more reliable measurement.

```
select
   u.name
from
   "user" u
where
   u.id = 'CxDOIDnH8gp9KXzpBHJYXw';
```

Ans:



10 Execution times (Executed on Dbeaver)

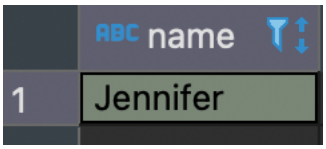
19ms, 17ms, 20ms, 21ms, 21ms, 19ms, 22ms, 18ms, 17ms, 17ms

Average Execution Times: 19.1ms

5) **(10 points)** With execution timing on, find the name of the user with 3336 compliment_plain compliments. Include the time it took to execute the query in your answer. *Note:* you may want to run this ~10 times and get the average timing across all runs to get a more reliable measurement.

```
select *
from
public."user" u
where u.compliment_plain = 3336;
```

Ans:



10 Execution times (Executed on Dbeaver)

349ms,335ms, 263ms, 343ms, 268ms, 324ms, 266ms, 323ms, 375ms, 263ms

Average Execution Time: 312.9 ms

6) **(10 points)** Which query is faster, query 5 or query 6, and by how much, and why is it faster? *Note*: this question does not ask you to write a query or provide a query's output. Simply provide your answers below.

Query 4 is faster, as the average execution time is around 0.067 ms. Query 4 saves 94.355 ms to get the outputs we want. The "id" column in the user table is indexed (which is common for primary key fields), and the database can quickly locate the row with the specified id value. On the other hand, the compliment_plain is not indexed, which means that the database has to perform a full table scan to count the compliments, making it significantly slower.

7) **(10 points)** Find the absolute number and percentage of businesses that have photos in the database, and businesses without any photo. *Hint*: to obtain a floating-point result in SQL arithmetic operations, at least one of the arithmetic operands must be a floating point number.

```
select
    SUM(case when x.b_id is not null then 1 else 0 end) as bus_with_photo,
    ROUND(100.0 * SUM(case when x.b_id is not null then 1 else 0 end) / COUNT(b.id),2) as bus_with_photo_perc,
    SUM(case when x.b_id is null then 1 else 0 end) as bus_with_no_photo,
```

```
    ROUND(100.0 * SUM(case when x.b_id is null then 1 else 0 end) / COUNT(b.id),2) as
bus_with_no_photo_perc
from
    business b
left join
    (select distinct(p.business_id) as b_id from photo p) as x
on b.id = x.b_id
```

Ans:

	123 bus_with_photo	123 bus_with_photo_perc	123 bus_with_no_photo	123 bus_with_no_photo_perc
1	27,850	17.78	128,789	82.22

8) **(10 points)** Some businesses are open fewer days of the week than others. Use a common table expression to find airports that are open only once a week and report their business id, name, and hours of operation.

```
with bwh as (
   select
        h.business_id as b_id,
        COUNT(*) over(partition by h.business_id) as num_days_open,
        h.hours as w_hours
   from
        hours as h
)
select b.id as business_id , b.name as name, bwh.w_hours as
working_hours
from
business b
inner join
bwh on b.id = bwh.b_id
where
bwh.num_days_open = 1 and b.name like '%Airport%';
```

	business_id	name	working_hours
1	6VaeaNoma3zRLslDrF1Cjg	Howard Johnson Phoenix Airport/ Downtown Area	Monday|6:00-6:00

9) **(20 points)** You are tasked with doing some city planning, which requires that you find clusters of businesses that are physically located very close to each other. Your first task is to find the IDs, names and GPS coordinates (latitude, longitude) of businesses that are clustered around McDonald's at address Av. Maip 2779. A business is considered part of the cluster if it is within 0.005 degrees away from any other business in the cluster. *Hint 1:* When you need to include an apostrophe as part of a text string in PostgreSQL, you need to escape it with another apostrophe, e.g., to find all "McDonald's" you need a query like `SELECT * FROM business WHERE name='McDonald''s';` Note the use of two apostrophes between letters d and s. *Hint 2:* You can use the Pythagorean theorem to find businesses within the requested range like in question 3. *Hint 3:* You need recursion!

```
WITH RECURSIVE mc_cluster(id,name,latitude,longitude) AS (
   select
        id ,
        name ,
        latitude,
        longitude
```

```
   from
        business
   where
        name = 'McDonald''s'
        and
        address  = 'Av. Maip 2779'
   union
   select
        b.id ,
        b.name,
        b.latitude,
        b.longitude
   from
        mc_cluster as c
   join
        business b on sqrt(pow(b.latitude-c.latitude,2) +
pow(b.longitude-c.longitude,2)) <= 0.005
)
select *
from
mc_cluster;
```

	id	name	latitude	longitude
1	softZjpREG65wpAns2FaWA	McDonald's	-34.51	-58.4911
2	bGxzQDGOTpab_6hdqsqv9g	Burger King	-34.5089	-58.4919
3	i1e8Ksly1ELvl7G6mvvZkw	Havanna	-34.5133	-58.4894
4	m-SUr48X9gMHtwvraM-KmA	Compaa del Sol	-34.5134	-58.4896
5	WNsimvxr-0NimM57I5gj4A	Arnaldo	-34.5137	-58.4888
6	yadScsa2pShYsQAVXbNivw	La Farola de Olivos	-34.5108	-58.4908
7	YBaWP2r64BPJazkmyf1fig	Almacn de Pizzas	-34.5089	-58.4916
8	zMAiU0s8ScUYHwAESCB8Qg	Prosciutto	-34.5122	-58.4898
9	4-xLjGavuWFqEfNuznxL3A	D' Lucky	-34.516	-58.4884
10	AwpX8mheEmMhaluIqEhMkA	Estacin Mitre - Lnea Mitre	-34.515	-58.4897
11	Ss6J7HFhMCxoq7M8wXqc8A	Salve Bruna	-34.5159	-58.488