This assignment is a continuation of assignment 4, you might need the code you wrote in assignment 4 for this assignment (You can also use the solution of assignment 4 after we published it).
You might need to refer the most updated version of this assignment at Assignment 5

Question 1: Data Preprocessing

For this assignment, we are providing you with 5 csv files which you can download. The link the directory with the files in ALLDATA.
These files contain data that you have already seen for your previous assignment. In assignment 4 we provided you with a sample of the data in the files. These files contain larger datasets.
You will need to create the tables in MySQL to store the data in the files. The first row of each file provides you with the schema of that file.

(1) Take a look at the csv file's first row, create a table in MySQL for it with proper data type for each attribute;

(2) Insert all other rows except the first row into the table you created;
You can do these in python or MySQL. For MySQL, as a reference, you can try to use:

```
LOAD DATA INFILE '/path/to/data.csv'
INTO TABLE my_table
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;
```

(3) After loading the file, we need to make sure that the number of rows in the corresponding tables inMySQL matches the number of rows in the csv file.

For each table, capture the screenshot of your MySQL execution result for "SELECT count(*) FROM my_table;" - run one such query for each table.

Note: since there are 5 tables, you will be submitting 5 screenshots.

For the second question, you will need to first transfer the tables you created in Assignment 4 to MySQL. In particular, go ahead and create three tables Astar_Based_Rating, Balanced_Rating, and General_Rating, that you had created in the previous assignment, and write the SQL queries to populate these tables over the dataset we have provided in the csv files with this assignment. You can likely cut and paste your SQL part of your python code from Assignment 4 for this purpose into Workbench.

Question 2: Ranking Universities

Write an SQL query to rank universities based on their scores for each of the three tables above: i.e., Astar_Based_Rating, Balanced_Rating, and General_Rating. You will be writing three queries, one for each table. The university with the highest score will have a rank 1, the second highest score rank 2, and so on.
Output the following attributes for each table:

- university_name
- score
- Ranking

Each output should be ordered based on rank and should contain only top 50 universities.
Save these outputs into csv files which you will use in your next question. Let us refer to these csv files as star, balanced, and general

To submit the answer to this question, submit the SQL query. Also, run the query from workbench and submit a screenshot showing the output for each query (note: each output will contain 50 university names and their ranks)

Question 3: Analyze Rankings Using Python

1. Read the three csv files from the last question into python. Also, we need to load up rankings based on usnews data as well in your python code . You can do this directly from the cvs file and generate rankings based on usnews inside python or write a corresponding query to extract university name and ranking in SQL based on the table storing usnews ranking. Limit the usnews ranking to only the top 52 universities not 50 since some universities rank the same at the boundary if you checked the usnewsranking csv file.

We will next generate a correlation between a list in files star, balanced, and general to the usnews rankings using kendall tau correlation described below.

2. Kendall Tau Correlation: Consider two lists L1, L2 of universities (E.g., L1 is from file f1 and L2 is based on usnews rankings).

- Compute the union of all universities in the two lists L = L1 union L2 (note: the union may have more than 50 tuples since not all universities in L1 may be in L2 and vice versa).
- For each pair of universities u_i, and u_j in L (i.e., the union of the two lists), determine if the two lists agree or disagree on the ranking of u_i and u_j . We say that the two L1 and L2 agree on the rank of u_i and u_j if it is the case that rank(u_i, L1) < rank(u_j, L1) then rank(u_i, L2) <rank(u_j, L2), else they disagree. Please see example below which shows how to determine if lists agree or disagree u clearly including corner cases when one or both university might not be in a list.

- Once we have determined the set of agreements and disagreements based on pairs of universities, we can compute correlation as follows:

$$\tau = \frac{n_c - n_d}{n(n-1)/2}$$

n_c is the the number of agreed pairs / n_d is the number of disagreed pairs / n is the size of the union of the two lists (L1 union L2)

Compute and output the correlation between the us news world rankings and each of the ratings ((i.e., star, balanced, and general) based on the above formula above. Submit (a) execution result and (b) your code of calculating KTC.

Example: illustrating how to compute *the set of agreement and disagreements between two ranking lists.*

Suppose we have two ordered lists: T1(a1, a2, a4), T2(a1, a3, a2). n=|T1 U T2|=4.

[a1, a2], T1 agrees T2 since they both say a1 ranks better than a2

[a1, a3], T1 agrees T2 since rank(a1, T2) < rank(a3, T2) and

rank(a1, T1) ≤ 3 < rank(a3, T1)

[a1, a4], T1 agrees T2 since rank(a1, T1) < rank(a4, T1) and

rank(a1, T2) ≤ 3 < rank(a4, T2)

[a2, a3], T1 disagrees T2, since rank(a2, T1) ≤ 3 < rank(a3, T1) and

rank(a2, T2) > rank(a3, T2)

[a2, a4], T1 agrees T2 since rank(a2, T1) < rank(a4, T1) and

$rank(a2, T2) \leq 3 < rank(a4, T2)$

[a3, a4], T1 disagrees T2, since $rank(a3, T1) \leq 3 < rank(a4, T1)$ and

$rank(a4, T2) > 3 \geq rank(a3, T2)$

n_c=4, n_d=2, n=4 in this example (if you look at n(n-1)/2, you will find it equals 6 pairs).

So the correlation is 2/6 = 1/3

Some corner cases:

a) *What if a pair of universities are missing from one of the lists?*

Say we have two ordered lists, the first is T1(a1, a2, a3),

and the second is T2(a1, a4, a5).

For pair [a2, a3], we know that $rank(a2, T1) < rank(a3, T1)$,

But for T2, we don't know the ranking relationship between a2 and a3.

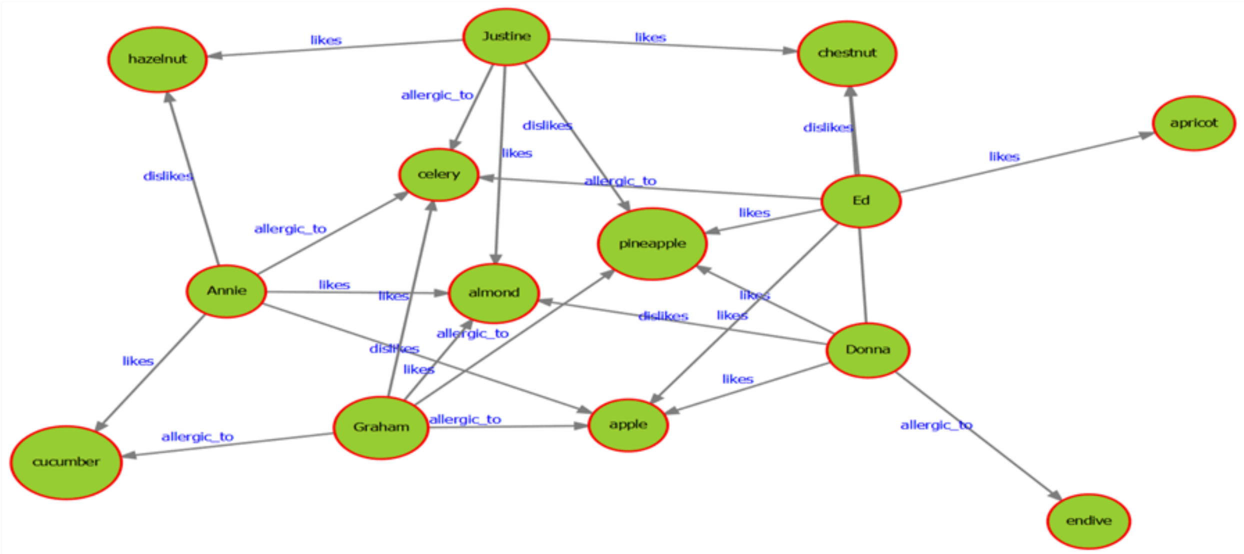
As specified in our strategy, we will assume $rank(a2, T2) = rank(a3, T2)$ in our assignment.

Hence T1 disagrees with T2 for this case.

b) *What if two universities in both lists are equal in ranking (i.e., have the same rank)*

In cases where both lists contain the equal ranking for a pair, we will consider T1 agrees with T2.

Question 4. Spark SQL



Consider a Sparql query below over the RDF graph above. What will be the output?
(the edge (Donna, chestnut) and edge (Ed, chestnut) both are dislike, edge (Graham, pineapple is allergic_to), There is no edge between donna and ed. That's just the edge from donna to chestnut)

```
SELECT ?a ?b ?c ?d
WHERE {
  ?a likes ?c
  ?b dislikes  ?c
  ?a dislikes ?d
  ?b likes ?d
}
```

Question 5: Json query

Consider the following sequence of SQL commands which are executed:

```
CREATE TABLE users (
    user_id INT PRIMARY KEY,
    user_info JSON
);
```

```
INSERT INTO users (user_id, user_info)
VALUES
    (1, '{"name": "Alice", "age": 30, "email": "alice@example.com"}'),
    (2, '{"name": "Bob", "age": 25, "email": "bob@example.com", "preferences": {"newsletter": true, "sms": false}}'),
    (3, '{"name": "Charlie", "age": 35, "email": "charlie@example.com"}'),
    (4, '{"name": "Diana", "age": 28, "email": "diana@example.com", "preferences": {"newsletter": false, "sms": true}}');
```

A.

```
UPDATE users
SET user_info = JSON_SET(user_info, '$.email', 'bob.new@example.com')
WHERE user_id = 2;
```

```
UPDATE users
SET user_info = JSON_SET(user_info, '$.phone', '123-456-7890')
WHERE user_id = 1;

UPDATE users
SET user_info = JSON_REMOVE(user_info, '$.preferences.sms')
WHERE user_id = 4;

SELECT *
FROM users
WHERE JSON_EXTRACT(user_info, '$.age') > 27;
```

*What will be the output of the above SELECT * ... query*