Assignment 2

Due date: March 15, 2019, 11:55pm IST

General Instructions

1. Please complete this assignment *individually*, on your own.
2. You will submit 2 files: EntryNumber.sql and EntryNumber.pdf, corresponding to the dataset you created locally.
3. Use PostgreSQL 11 for your homework. See `https://www.postgresql.org/download/` for instructions on how to download and install it on your OS. The .sql files are run automatically using the psql command using the \i option, so please ensure that there are no syntax errors in the file. *If we are unable to run your file, you get an automatic reduction to 0 marks.*

   To understand how to run many queries at once from text file, a dummy query file example.sql is available. To run example.sql in postgres, type the following command in the terminal:

   sudo -u postgres psql dbname

   and then type

   `\i /address/to/example.sql`

   This command will run all the queries listed in example.sql at once.
4. The format of the .sql file should be as follows. One line should identify the query number (note the two hyphens before and after the query number), followed by the actual, syntactically correct SQL query. Leave a blank line after each query.

   - -1- -

   SQL QUERY

   - -2- -

   SQL QUERY

   - -3- -

   SQL QUERY

   - -CLEANUP- -

   CLEANUP EVERYTHING YOU CREATED HERE
5. Assume set semantics, unless stated otherwise.
6. Each correct query carries 2 marks.
7. There is no data provided for these queries. For testing, make your own datasets. The .sql file that you submit however, should contain only queries

8. You can create the relevant table using following queries

```
CREATE TABLE TrainSchedule(
    Train_ID  varchar,
    Source varchar,
    Destination varchar,
    Distance integer,
    Departure_Time time,
    Arrival_Time time
    );
```

9. You can also assume that there is no circular path (there is no path like Delhi → Kolkata → Jaipur → Bhopal → Kolkata) in your database, so that you can fetch the results properly.
10. You can also assume that all the journeys are within a day i.e every train completes its journey in the same day.
11. No changes are allowed in the i) attribute names, ii) table name
12. Expected output of each question is given based on the following table. (Note that this is sample table only for clarification)

```
Test Data:
 train_id | source   | destination | distance | departure_time | arrival_time
----------+----------+-------------+----------+----------------+--------------
    22435 | Kolkata | Bhopal      |     1200 | 08:00:00       | 17:00:00
    21484 | Mumbai  | Jaipur      |      500 | 13:23:44       | 13:23:45
    21424 | Delhi   | Mumbai      |      800 | 13:23:44       | 15:05:40
    12456 | Bhopal  | Mumbai      |      800 | 11:00:00       | 23:00:00
    12453 | Banaras | Mumbai      |      500 | 11:00:00       | 21:00:00
    21514 | Jaipur  | Madras      |     1500 | 10:05:00       | 13:23:45
    21414 | Delhi   | Kolkata     |      800 | 14:05:00       | 15:05:40
    23432 | Bhopal  | Hyderabad   |      670 | 12:00:00       | 20:20:00
(8 rows)
```

Queries

1. Find all cities reachable from Delhi through a series of one or more connecting Trains (Show your output in ascending order).

```
cities_reachable
------------------
 Bhopal
 Hyderabad
 Jaipur
 Kolkata
 Madras
 Mumbai
(6 rows)
```

2. Find all cities reachable from Delhi through a chain of one or more connecting trains, with no more than one hour spent on any connection. (That is, every connecting train must depart within an hour of the arrival of the previous train in the chain.) note: Show your output in ascending order

```
destination
-------------
 Kolkata
 Mumbai
(2 rows)
```

3. Find the shortest time to travel from Delhi to Mumbai, using a chain of one or more connecting trains.

```
shortest_time
---------------
 01:41:56
```

4. Find the Train ID of all trains that do not depart from Delhi or a city that is reachable from Delhi through a chain of Trains.

```
train_id
----------
    12453
```

5. Find all the pairs of cities (c1, c2) such that there is a path from c1 to c2, such that successive travel times between the consecutive cities is in increasing order. note: sort your output based on the first column.

```
source  | destination
---------+-------------
 Banaras | Mumbai
 Bhopal  | Mumbai
 Bhopal  | Hyderabad
 Delhi   | Mumbai
 Delhi   | Kolkata
 Delhi   | Bhopal
 Jaipur  | Madras
 Kolkata | Mumbai
 Kolkata | Bhopal
 Mumbai  | Jaipur
 Mumbai  | Madras
(11 rows)
```

6. Find all the pairs of cities (c1, c2) such that there is a path from c1 to c2, such that alternate travel times between the consecutive cities is in decreasing order. (alternate travel times implies the ordering like 1-3-5-...i.e. Starting from the first interval check alternate intervals for decreasing order of time.) note: sort your output based on first column.

```
source  | destination
---------+-------------
 Banaras | Mumbai
 Banaras | Jaipur
 Banaras | Madras
 Bhopal  | Jaipur
 Bhopal  | Mumbai
 Bhopal  | Hyderabad
 Bhopal  | Madras
 Delhi   | Bhopal
 Delhi   | Kolkata
 Delhi   | Mumbai
 Delhi   | Jaipur
 Jaipur  | Madras
 Kolkata | Madras
 Kolkata | Hyderabad
 Kolkata | Mumbai
 Kolkata | Jaipur
 Kolkata | Bhopal
 Mumbai  | Madras
 Mumbai  | Jaipur
(19 rows)
```

7. Find all the pairs of cities (c1, c2) such that there is no path from c1 to c2.
   note: sort your output based on first column.

```
 source    | destination
-----------+-------------
 Mumbai    | Kolkata
 Hyderabad | Kolkata
 Hyderabad | Mumbai
 Jaipur    | Kolkata
 Jaipur    | Mumbai
 Bhopal    | Kolkata
 Banaras   | Bhopal
 Banaras   | Delhi
 Madras    | Mumbai
 Madras    | Kolkata
 Banaras   | Hyderabad
 Hyderabad | Delhi
 Hyderabad | Bhopal
 Mumbai    | Hyderabad
 Jaipur    | Bhopal
 Jaipur    | Delhi
 Madras    | Banaras
 Hyderabad | Madras
 Jaipur    | Hyderabad
 Hyderabad | Jaipur
 Kolkata   | Delhi
 Delhi     | Banaras
 Mumbai    | Bhopal
 Bhopal    | Banaras
 Mumbai    | Delhi
 Jaipur    | Banaras
 Madras    | Delhi
 Madras    | Bhopal
 Hyderabad | Banaras
 Bhopal    | Delhi
 Mumbai    | Banaras
 Kolkata   | Banaras
 Madras    | Jaipur
 Madras    | Hyderabad
 Banaras   | Kolkata
(35 rows)
```

8. Given source city Delhi, destination city Mumbai return the number of paths between these two cities.

```
no_of_paths
-------------
           2
```

9. Find all the cities that are reachable form Delhi by exactly one path (means list all cities 'c' such that there is exactly one path from Delhi to 'c').
   note: sort your output.

```
 cities_havingexactly_onepath
------------------------------
 Bhopal
 Hyderabad
 Kolkata
(3 rows
```

10. Given source city Delhi, destination city Hyderabad return the number of paths between these two cities, that also pass through another city Bhopal.
    note: sort your output.

```
count
-------
     1
```