LAB ASSIGNMENT 6 (DBMS)

U20CS110
Krishna Pandey
B110

In reference to the **Bus database** given below, attempt the following questions:

The queries to create the tables and the data to be inserted are listed below. Run these queries against the database to have your tables and data ready.

Some notes on terms used:

+ table "**bushalts"** contains one row for every halt of a bus.

-- id : id of the bus which is the primary key.
-- seqno : The halt number. Assume that the starting station has seqno as 0.
-- stcode : station code of this halt.
-- timein : time at which the bus arrives at this station. (will be null for the starting station of a bus)
-- timeout: time at which the bus departs this station. (will be null for the terminating station of a bus)
-- If a bus passes through a station without stopping, then there will be an entry with timein = timeout.

+ table "**track"** stores the distances between directly connected stations stcode1 and stcode2.

-- Assume that this represents a directed track. i.e., for two stations A and B, there will be an entry corresponding to (A, B, distance) and another for (B, A, distance).

Script to be run to set up the tables and insert the respective data with the following constraints:

- The stcode of table bushalts is the code which is available in the station table only.
- The track table should have distance value greater than 0.
- The seqno of the table bushalts should be automatically generated using the "SEQUENCE".

create table bus (id varchar(5) , name varchar(20), primary key (id));	create table station (stcode varchar(5), name varchar(20), primary key (stcode));
create table track (stcode1 varchar(5) , stcode2 varchar(5), distance integer ,	create table bushalts (id varchar(5) , seqno integer , stcode varchar(10),

primary key (stcode1, stcode2));	timein varchar(5) , timeout varchar(5) , primary key (id, seqno));

```
insert into bus values ('KP11' ,'ST-KYN');

insert into bus values ('KP11L' ,'ST-KYN_LOCAL');

insert into bus values ('T129' ,'ST-TNA_LOCAL');

insert into bus values ('A63' ,'ST-DL_LOCAL');

insert into bus values ('K101' ,'ST-KYN_LOCAL');

insert into bus values ('N27' ,'ST-TNA_LOCAL');

insert into bus values ('S33' ,'ST-KGR_LOCAL');

insert into bus values ('A65' ,'ST-AMR_LOCAL');
insert into station values ('ST' ,'MUMBAI');

insert into station values ('BYC' ,'BYCULLA');

insert into station values ('DR' ,'DADAR');

insert into station values ('KRL' ,'KURLA');

insert into station values ('GPR' ,'GHATKOPAR');

insert into station values ('TNA' ,'THANE');

insert into station values ('DL' ,'DOMBIVALI');

insert into station values ('AMR' , 'AMBARNATH');

insert into station values ('KYN' ,'KALYAN');

insert into station values ('KSR' ,'KASARA');

insert into track values ('ST' ,'BYC', 5);

insert into track values ('ST' ,'DR', 9);
insert into track values ('ST' ,'KRL', 16);

insert into track values ('ST' ,'GPR', 20);

insert into track values ('ST' ,'TNA', 34);

insert into track values ('ST' ,'DL', 49);

insert into track values ('ST' ,'KYN', 54);

insert into track values ('ST' ,'KSR', 77);

insert into track values ('ST' ,'AMR', 65);

insert into track values ('BYC' ,'DR', 4);

insert into track values ('BYC' ,'KRL', 11);

insert into track values ('GRP' ,'TNA', 14);

insert into track values ('DR' ,'TNA', 25);

insert into track values ('KRL' ,'KYN', 38);

insert into track values ('TNA' ,'KYN', 20);

insert into track values ('TNA' ,'KSR', 43);

insert into bushalts values ('KP11' , 0 , 'ST' , NULL, '20.23');

insert into bushalts values ('KP11' , 1 , 'BYC' , '20.31', '20.32');

insert into bushalts values ('KP11' , 2 , 'DR' , '20.41', '20.42');

insert into bushalts values ('KP11' , 3 , 'GPR' , '20.52', '20.53');

insert into bushalts values ('KP11' , 4 , 'GPR' , '20.52', '20.53');

insert into bushalts values ('KP11' , 5 , 'DR' , '20.41', '20.42');

insert into bushalts values ('KP11' , 6 , 'GPR' , '20.58', '20.59');

insert into bushalts values ('KP11' , 7 , 'TNA' , '21.21', '21.22');
insert into bushalts values ('KP11' , 8 , 'DL' , '21.45', '21.46');

insert into bushalts values ('KP11' , 9 , 'KYN' , '21.54', NULL);

insert into bushalts values ('A65' , 0 , 'ST' , NULL , '20.52');

insert into bushalts values ('A65' , 1 , 'BYC' , '21.00' , '21.01');

insert into bushalts values ('A65' , 2 , 'DR' , '21.10' , '21.11');

insert into bushalts values ('A65' , 3 , 'KRL' , '21.22' , '21.23');

insert into bushalts values ('A65' , 4 , 'GPR' , '21.28' , '21.29');

insert into bushalts values ('A65' , 5 , 'TNA' , '21.49' , '21.50');

insert into bushalts values ('A65' , 6 , 'DL' , '22.13' , '22.14');

insert into bushalts values ('A65' , 7 , 'KYN' , '22.22' , '22.23');

insert into bushalts values ('A65' , 8 , 'AMR' , '22.36' , NULL);
```

After the database tables and data are set; write the following queries:

1. Display all the pairs of stations with total distance for given source and destinations.

```
-> SELECT * FROM TRACK;
```

2. Find the pairs of stations (station codes) which have a track with distance more than 15km between them.

```
-> SELECT STCODE1,STCODE2 FROM TRACK WHERE DISTANCE>15;
```

3. Find the IDs of all the buses which have a stop at DADAR.

```
-> SELECT DISTINCT ID FROM BUSHALTS,STATION WHERE
STATION.STCODE=BUSHALTS.STCODE AND NAME='DADAR'AND
TIMEIN<>TIMEOUT;
```

4. Find the ordered list of names of all buses that start at THANE.

```
->SELECT BUS.NAME FROM BUS,BUSHALTS,STATION WHERE
BUS.ID=BUSHALTS.ID AND STATION.STCODE=BUSHALTS.STCODE AND
STATION.NAME='THANE' AND TIMEIN IS NULL;
```

5. List all the stations in order of visit by the bus ‘ST- DL_LOCAL'.

```
-> SELECT STCODE FROM BUSHALTS,BUS WHERE BUSHALTS.ID=BUS.ID AND
BUS.NAME='ST-DL_LOCAL' ORDER BY SEQNO;
```

6. Find the name of the buses which stop at KALYAN, before the 3rd stop in the route of the bus.

```
-> SELECT BUS.NAME FROM BUS,BUSHALTS,STATION WHERE
BUS.ID=BUSHALTS.ID AND BUSHALTS.STCODE=STATION.STCODE AND
STATION.NAME='KALYAN' AND SEQNO<3;
```

7. Display the pair of stations (i.e. station names) having maximum distance between them.

```
-> SELECT STATION.NAME FROM STATION,TRACK WHERE
(STATION.STCODE=STCODE1 OR STATION.STCODE=STCODE2) AND
DISTANCE=(SELECT MAX(DISTANCE) FROM TRACK);
```

8. Display id of the bushalt having highest time out.

```
-> SELECT ID FROM BUSHALTS WHERE TIMEOUT=(SELECT MAX(TIMEOUT) FROM
BUSHALTS);
```

9. Remove Track “ST” from the track table. Note: If any track is removed from the track table, then that track related information also should be removed from the other tables.
10. Remove Track “KP11” from the bus table. If any bus is removed from the bus table that track related information also should be removed from the other tables.
11. Delete the train "KP11" and all halt stations of that train.

```
DELETE FROM STATION WHERE STCODE = 'ST';
DELETE FROM TRACK WHERE STCODE1 = 'ST' OR STCODE2 = 'ST'; DELETE FROM
BUSHALTS WHERE STCODE = 'ST';
DELETE FROM BUS WHERE ID = 'KP11'; DELETE FROM BUSHALTS WHERE ID =
'KP11';
```

*