DBMS ASSIGNMENT – 5

***Roll Number: U19CS012***

***Name: BHAGYA VINOD RANA***

In reference to the **Railway 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.**

**SQL-Code [**SQLite 3.29.0**]:**

BEGIN TRANSACTION;

create table train(

    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 ,

    primary key (stcode1, stcode2)

);

create table trainhalts(

    id varchar(5) ,

    seqno integer ,

    stcode varchar(10),

    timein varchar(5) ,

    timeout varchar(5) ,

    primary key (id, seqno)

);

insert into train values ('KP11' ,'CST-KYN');

insert into train values ('KP11L' ,'CST-KYN\_LOCAL');

insert into train values ('T129' ,'CST-TNA\_LOCAL');

insert into train values ('A63' ,'CST-DL\_LOCAL');

insert into train values ('K101' ,'CST-KYN\_LOCAL');

insert into train values ('N27' ,'CST-TNA\_LOCAL');

insert into train values ('S33' ,'CST-KGR\_LOCAL');

insert into train values ('A65' ,'CST-AMR\_LOCAL');

insert into station values ('CST' ,'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 ('CST' ,'BYC', 5);

insert into track values ('CST' ,'DR', 9);

insert into track values ('CST' ,'KRL', 16);

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

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

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

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

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

insert into track values ('CST' ,'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 trainhalts values ('KP11' , 0 , 'CST' , NULL, '20.23');

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

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

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

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

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

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

insert into trainhalts values ('KP11' , 7 , 'TNA' , '21.21', '21.22');

insert into trainhalts values ('KP11' , 8 , 'DL' , '21.45', '21.46');

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

insert into trainhalts values ('A65' , 0 , 'CST' , NULL , '20.52');

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

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

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

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

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

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

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

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

*-- Saving the Work*

COMMIT;

*-------------------------------------------------------------------*

*-- For Output Formatting [Human Understandable Form] in SQLite*

.mode column

.headers on

.separator ROW "\n"

.nullvalue NULL

*-------------------------------------------------------------------*

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

SELECT s1.name AS "START-POINT",s2.name AS "END-POINT", t.distance AS "DISTANCE"

FROM station s1,station s2, track t

WHERE s1.stcode==t.stcode1 AND s2.stcode==t.stcode2;

*-- 2. Find the pairs of stations (station codes) which have a track with distance less than 20Kms*

*-- between them.*

SELECT s1.name AS "START-POINT",s2.name AS "END-POINT", t.distance AS "DISTANCE"

FROM station s1,station s2, track t

WHERE s1.stcode==t.stcode1 AND s2.stcode==t.stcode2 AND t.distance<20;

*-- 3. Find the IDs of all the trains which have a stop at GHATKOPAR*

SELECT DISTINCT t1.id AS "TRAIN-NO", t1.name AS "TRAIN-NAME"

FROM trainhalts th1,train t1

WHERE (th1.stcode=='GPR' AND th1.id==t1.id);

*-- 4. Find the ordered list of names of all trains that start at MUMBAI.*

SELECT s.name AS "TRAINS\_STARTING\_AT\_MUMBAI"

FROM track t,station s

WHERE t.stcode1=='CST' AND t.stcode2==s.stcode ORDER BY s.name;

*-- 5. List all the stations in order of visit by the train 'CST-AMR\_LOCAL'.*

SELECT s.name AS "STATIONS VISITED BY TRAIN 'CST-AMR\_LOCAL'"

FROM trainhalts th, station s, train t

WHERE (t.name == 'CST-AMR\_LOCAL' AND t.id == th.id AND th.stcode == s.stcode)

*-- 6. Find the name of the trains which stop at Thane, before the 6th stop in the route of the train.*

SELECT t.name AS "TRAINS STOPS AT THANE [BEFORE 6TH STOP IN ROUTE]"

FROM trainhalts th, train t

WHERE (th.seqno<6 AND th.id==t.id AND th.stcode == 'TNA');

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

SELECT s1.name AS "MAX DISTANCE STATION\_1 [START]",s2.name AS "MAX DISTANCE STATION\_2 [END]"

FROM track t,station s1,station s2

WHERE (t.distance = (SELECT MAX(distance) FROM track)) AND (t.stcode1==s1.stcode) AND (t.stcode2==s2.stcode);

*-- 8. Display id of the trainhalt having second highest time out.*

SELECT id AS "TRAIN-HALT ID WITH 2ND HIGHEST TIMEOUT",timeout FROM trainhalts

WHERE timeout = (

SELECT MAX(timeout)

FROM trainhalts

WHERE timeout NOT IN (SELECT MAX(timeout) FROM trainhalts));

*-- 9. Remove Track “CST” 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.*

DELETE FROM station WHERE stcode = 'CST';

DELETE FROM track WHERE (stcode1 = 'CST' OR stcode2 = 'CST');

DELETE FROM trainhalts WHERE (stcode = 'CST');

*-- -- For Checking*

SELECT \* FROM station;

SELECT \* FROM track;

SELECT \* FROM trainhalts;

*-- 10. Remove Track “KP11” from the train table.If any train is removed from the train table that track*

*-- related information also should be removed from the other tables.*

DELETE FROM station WHERE stcode = 'KP11';

DELETE FROM track WHERE (stcode1 = 'KP11' OR stcode2 = 'KP11');

DELETE FROM trainhalts WHERE (stcode = 'KP11');

*-- For Checking*

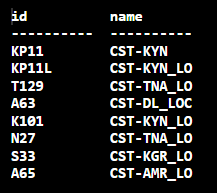
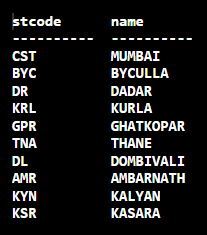
SELECT \* FROM station;

SELECT \* FROM track;

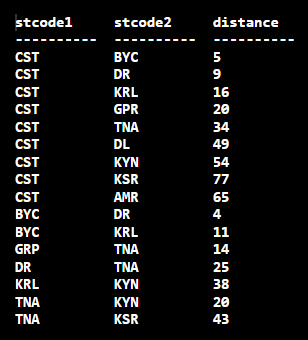
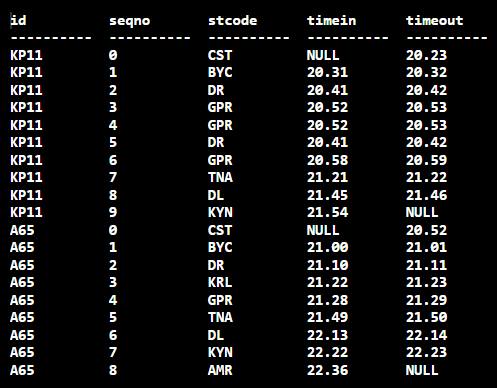
SELECT \* FROM trainhalts;

**Initial Table:**

A.) TRAIN TABLE B.) STATION TABLE

C.) TRACK TABLE D.) TRAIN HALTS TABLE

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

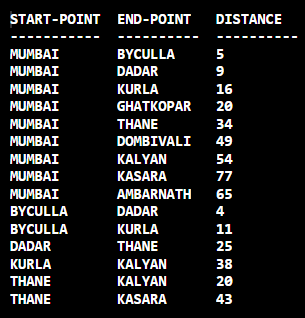
**Query:**

SELECT s1.name AS "START-POINT",s2.name AS "END-POINT", t.distance AS "DISTANCE"

FROM station s1,station s2, track t

WHERE s1.stcode==t.stcode1 AND s2.stcode==t.stcode2;

**Output:**



2. Find the pairs of stations (station codes) which have a track with distance less than 20Kms between them.

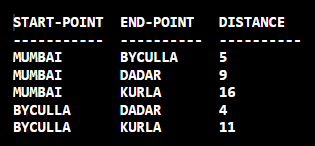
**Query:**

SELECT s1.name AS "START-POINT",s2.name AS "END-POINT", t.distance AS "DISTANCE"

FROM station s1,station s2, track t

WHERE s1.stcode==t.stcode1 AND s2.stcode==t.stcode2 AND t.distance<20;

**Output:**



3. Find the IDs of all the trains which have a stop at GHATKOPAR

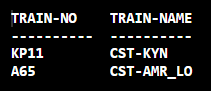
**Query:**

SELECT DISTINCT t1.id AS "TRAIN-NO", t1.name AS "TRAIN-NAME"

FROM trainhalts th1,train t1

WHERE (th1.stcode=='GPR' AND th1.id==t1.id);

**Output:**



4. Find the ordered list of names of all trains that start at MUMBAI.

**Query:**

SELECT s.name AS "TRAINS\_STARTING\_AT\_MUMBAI"

FROM track t,station s

WHERE t.stcode1=='CST' AND t.stcode2==s.stcode ORDER BY s.name;

**Output:**



By Default Also, The Output is Order by Name. [~~ORDER BY s.name~~]

5. List all the stations in order of visit by the train 'CST-AMR\_LOCAL'.

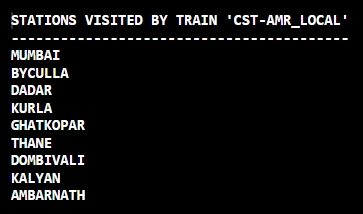
**Query:**

SELECT s.name AS "STATIONS VISITED BY TRAIN 'CST-AMR\_LOCAL'"

FROM trainhalts th, station s, train t

WHERE (t.name == 'CST-AMR\_LOCAL' AND t.id == th.id AND th.stcode == s.stcode)

**Output:**



6. Find the name of the trains which stop at Thane, before the 6th stop in the route of the train.

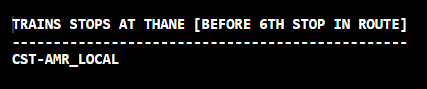
**Query:**

SELECT t.name AS "TRAINS STOPS AT THANE [BEFORE 6TH STOP IN ROUTE]"

FROM trainhalts th, train t

WHERE (th.seqno<6 AND th.id==t.id AND th.stcode == 'TNA');

**Output:**



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

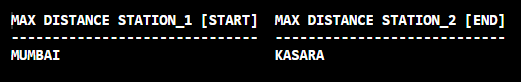
**Query:**

SELECT s1.name AS "MAX DISTANCE STATION\_1 [START]",s2.name AS "MAX DISTANCE STATION\_2 [END]"

FROM track t,station s1,station s2

WHERE (t.distance = (SELECT MAX(distance) FROM track)) AND (t.stcode1==s1.stcode) AND (t.stcode2==s2.stcode);

**Output:**



8. Display id of the trainhalt having second highest time out.

**Query:**

SELECT id AS "TRAIN-HALT ID WITH 2ND HIGHEST TIMEOUT",timeout FROM trainhalts

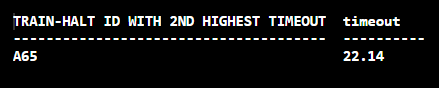
WHERE timeout = (

SELECT MAX(timeout)

FROM trainhalts

WHERE timeout NOT IN (SELECT MAX(timeout) FROM trainhalts));

**Output:**



Highest Timeout = **22.23**|| Second Highest Timeout = **22.14**

9. Remove Track “CST” 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.

**Query:**

DELETE FROM station WHERE stcode = 'CST';

DELETE FROM track WHERE (stcode1 = 'CST' OR stcode2 = 'CST');

DELETE FROM trainhalts WHERE (stcode = 'CST');

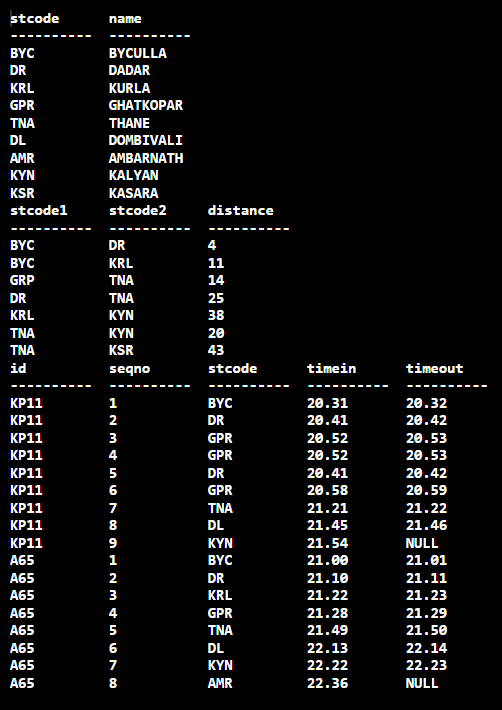
*-- For Checking*

SELECT \* FROM station;

SELECT \* FROM track;

SELECT \* FROM trainhalts;

**Output:**



10. Remove Track “KP11” from the train table. If any train is removed from the train table that track related information also should be removed from the other tables.

**Query:**

DELETE FROM station WHERE stcode = 'KP11';

DELETE FROM track WHERE (stcode1 = 'KP11' OR stcode2 = 'KP11');

DELETE FROM trainhalts WHERE (stcode = 'KP11');

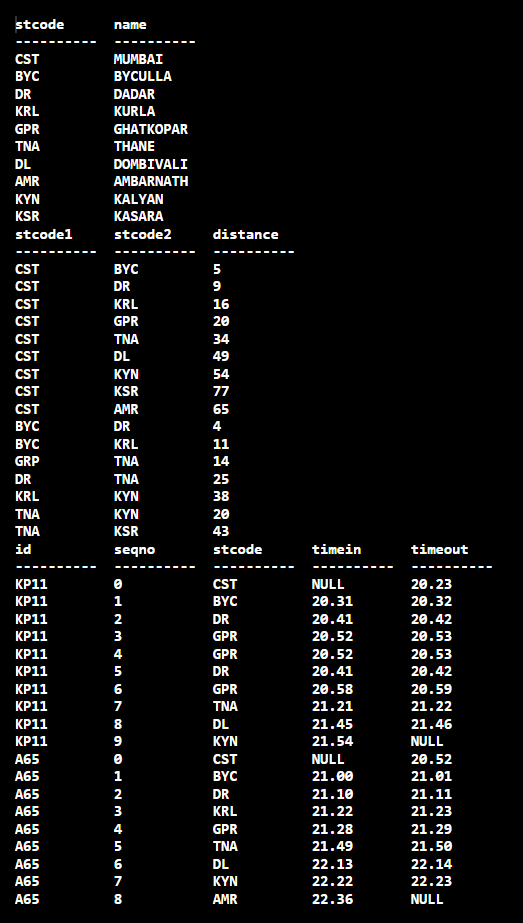
*-- For Checking*

SELECT \* FROM station;

SELECT \* FROM track;

SELECT \* FROM trainhalts;

**Output:**



**Submitted By:**

**BHAGYA VINOD RANA**

**U19CS012**