Solution of SQL Assignment-Major

1. . Create a table “Station” to store information about weather

observation stations:

CREATE TABLE Station

(

ID int primary key,

CITY CHAR(20),

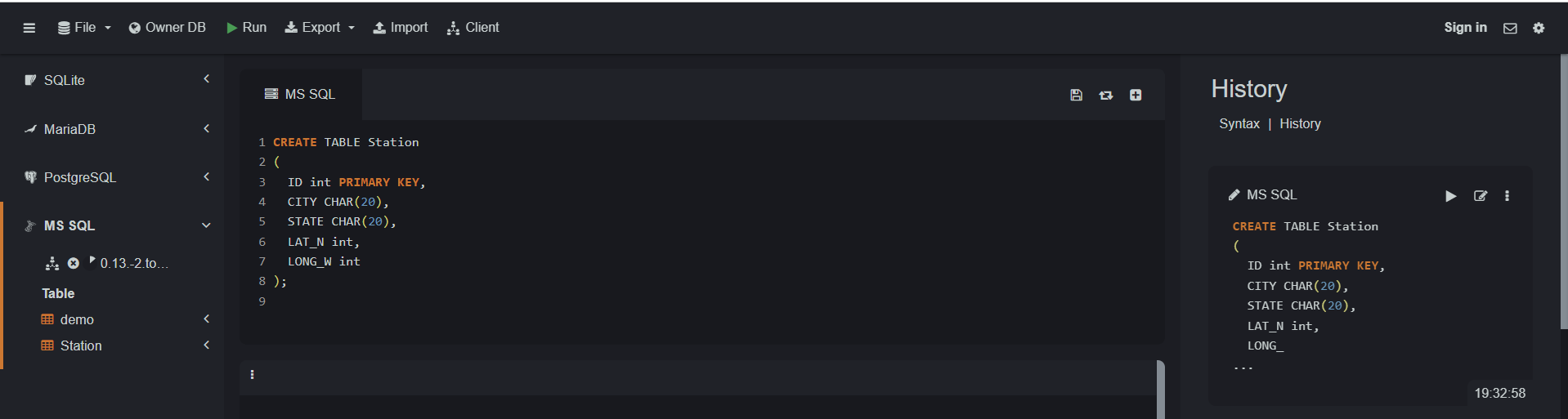
STATE CHAR(20),

LAT\_N int,

LONG\_W int

);

**OUTPUT**



2. . Create a table “Station” to store information about weather

observation stations:

INSERT into Station

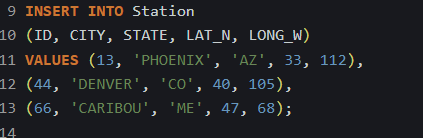
(ID, CITY, STATE, LAT\_N, LONG\_W)

VALUES (13, 'PHOENIX', 'AZ', 33, 112),

(44, 'DENVER', 'CO', 40, 105),

(66, 'CARIBOU', 'ME', 47, 68);

**OUTPUT**



3. Execute a query to look at table STATION in undefined order.

SELECT \* FROM Station;

**OUTPUT**

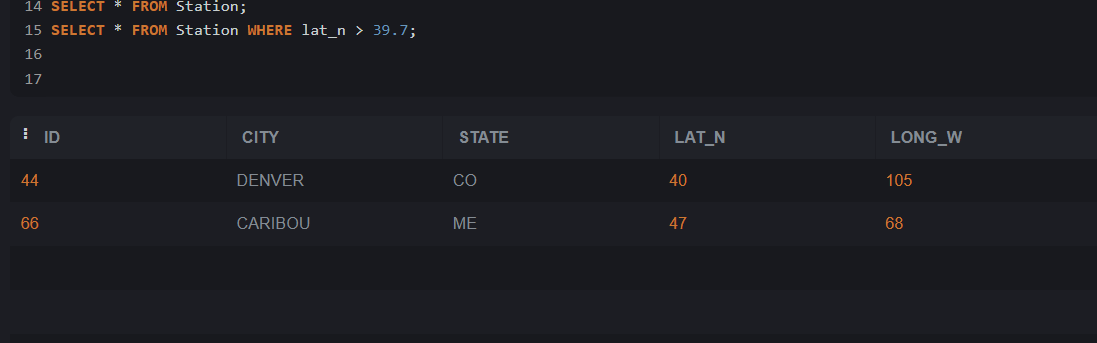


4. Execute a query to select Northern stations (Northern latitude >

39.7).

SELECT \* FROM Station WHERE lat\_n > 39.7;

**OUTPUT**



5. Create another table, ‘STATS’, to store normalized temperature and

precipitation data:

CREATE TABLE STATS

(

ID NUMBER,

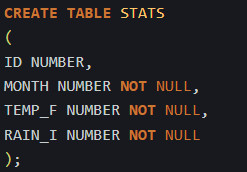
MONTH NUMBER NOT NULL,

TEMP\_F NUMBER NOT NULL,

RAIN\_I NUMBER NOT NULL

);

**OUTPUT**



6. Populate the table STATS with some statistics for January and July:

INSERT INTO STATS VALUES (13, 1, 57.4, 0.31);

INSERT INTO STATS VALUES (13, 7, 91.7, 5.15);

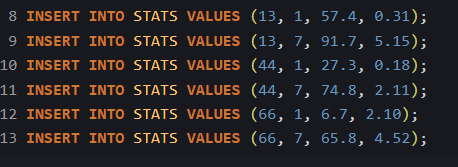
INSERT INTO STATS VALUES (44, 1, 27.3, 0.18);

INSERT INTO STATS VALUES (44, 7, 74.8, 2.11);

INSERT INTO STATS VALUES (66, 1, 6.7, 2.10);

INSERT INTO STATS VALUES (66, 7, 65.8, 4.52);

**OUTPUT**



7. Execute a query to display temperature stats (from STATS table) for

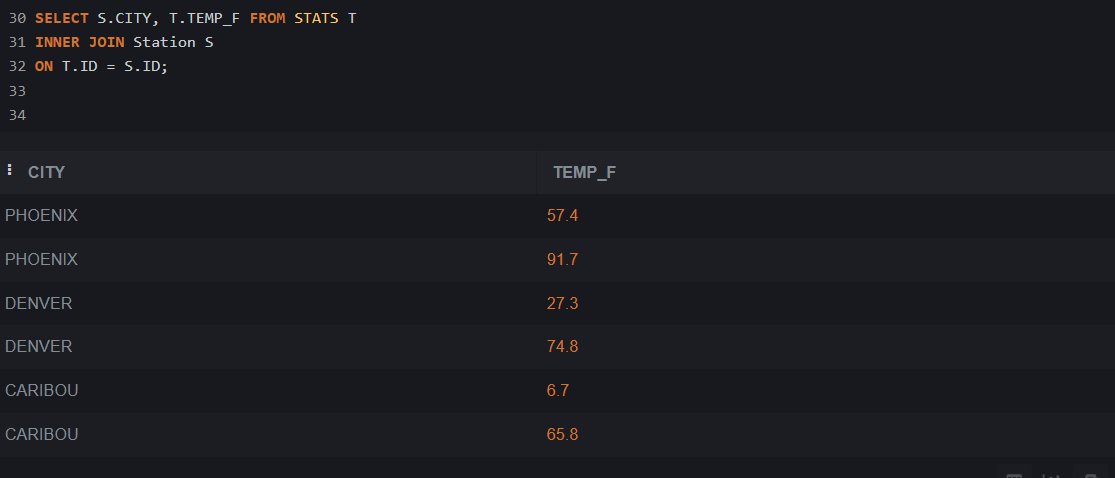
each city (from Station table).

SELECT S.CITY, T.TEMP\_F FROM STATS T

INNER JOIN Station S

ON T.ID = S.ID;

**OUTPUT**



8 . Execute a query to look at the table STATS, ordered by month and

greatest rainfall, with columns rearranged. It should also show the

corresponding cities.

SELECT T.MONTH , S.city, T.ID, T.RAIN\_I, T.TEMP\_F

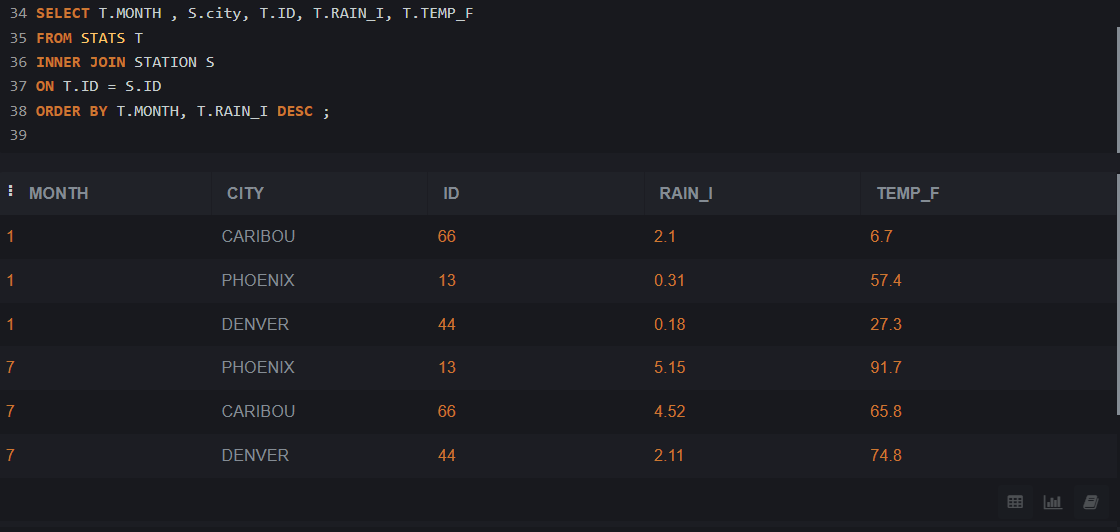
FROM STATS T

INNER JOIN STATION S

ON T.ID = S.ID

ORDER BY T.MONTH, T.RAIN\_I DESC ;

**OUTPUT**



9. Execute a query to look at temperatures for July from table STATS,

lowest temperatures first, picking up city name and latitude.

SELECT S.LAT\_N, S.CITY, T.TEMP\_F

FROM STATS T , STATION s

WHERE MONTH = 7

AND T.ID = S.ID

ORDER BY TEMP\_F;

**OUTPUT**



10. Execute a query to show MAX and MIN temperatures as well as

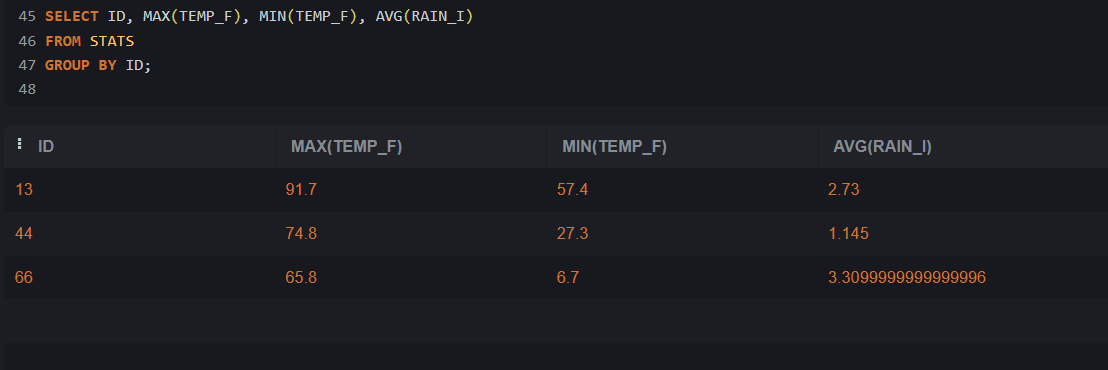
average rainfall for each city.

SELECT ID, MAX(TEMP\_F), MIN(TEMP\_F), AVG(RAIN\_I)

FROM STATS

GROUP BY ID;

**OUTPUT**



11. Execute a query to display each city’s monthly temperature in

Celcius and rainfall in Centimeter.

SELECT S.ID, S.City,

T.MONTH,

ROUND((TEMP\_F - 32) \* 5 /9) As TEMP\_CELCIUS,

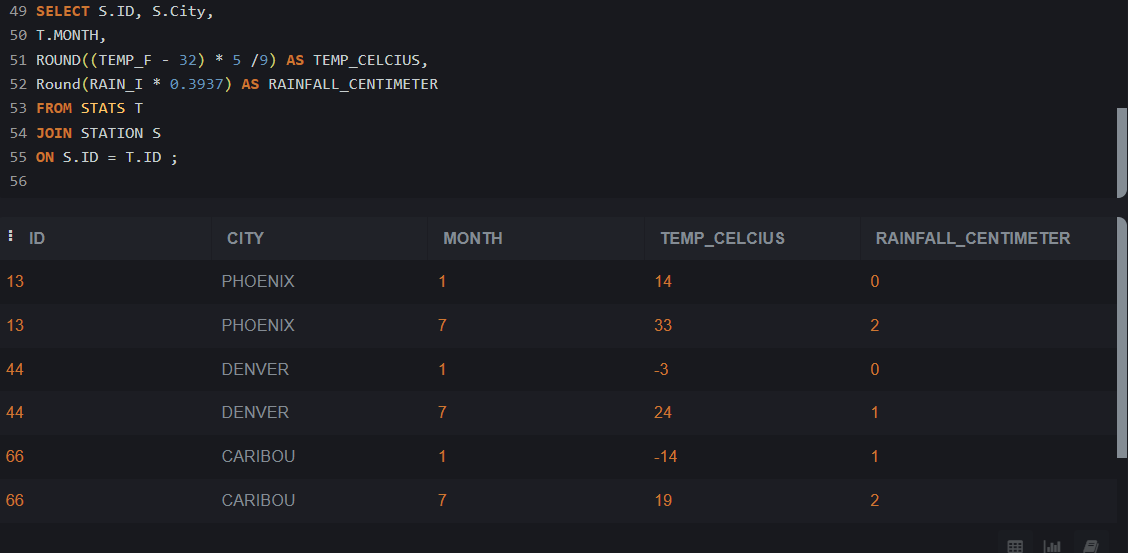
Round(RAIN\_I \* 0.3937) As RAINFALL\_CENTIMETER

From STATS T

JOIN STATION S

ON S.ID = T.ID ;

**OUTPUT**



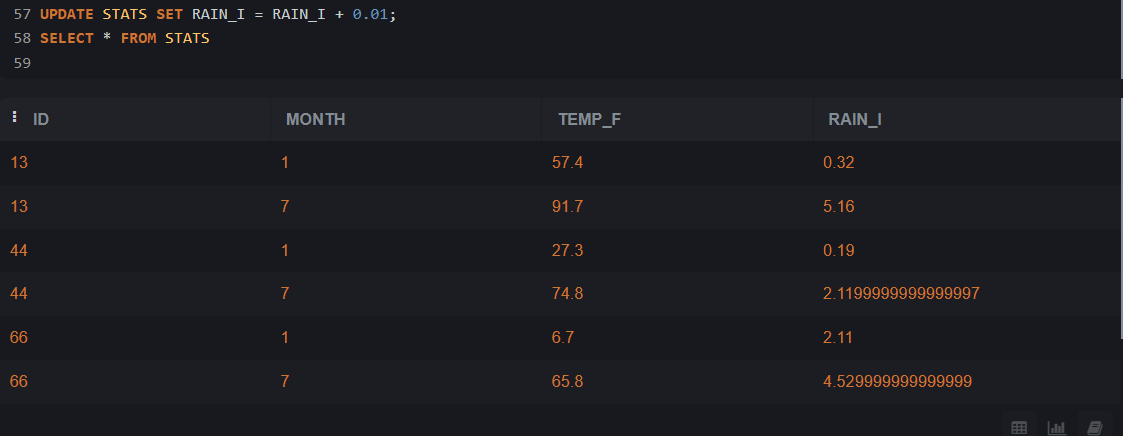
12. Update all rows of table STATS to compensate for faulty rain gauges

known to read 0.01 inches low.

UPDATE STATS SET RAIN\_I = RAIN\_I + 0.01;

SELECT \* FROM STATS

**OUTPUT**



13. Update Denver's July temperature reading as 74.9

UPDATE STATS SET TEMP\_F = 74.9

WHERE ID = 44 AND MONTH = 7;

SELECT \* FROM STATS

**OUTPUT**

