Q-1: Create table ‘STATION’ to store information about weather observation stations:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

ANS:

CREATE TABLE STATION

(

ID NUMBER PRIMARY KEY,

CITY CHAR(20),

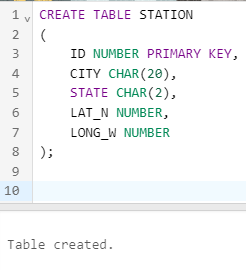
STATE CHAR(2),

LAT\_N NUMBER,

LONG\_W NUMBER

);

OUTPUT:



Q-2: Insert following records into the table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

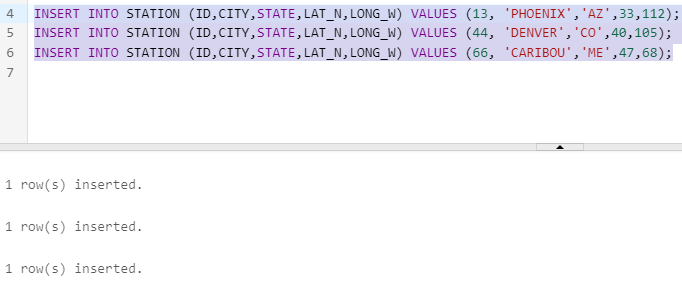
ANS:

INSERT INTO STATION (ID,CITY,STATE,LAT\_N,LONG\_W) VALUES (13, 'PHOENIX','AZ',33,112);

INSERT INTO STATION (ID,CITY,STATE,LAT\_N,LONG\_W) VALUES (44, 'DENVER','CO',40,105);

INSERT INTO STATION (ID,CITY,STATE,LAT\_N,LONG\_W) VALUES (66, 'CARIBOU','ME',47,68);

OUTPUT:

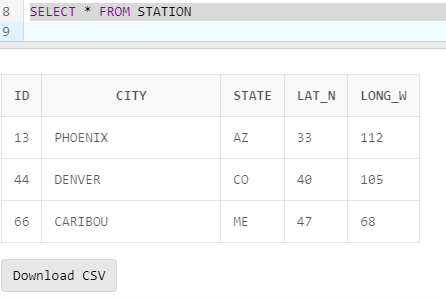


Q-3:Execute a query to look at table STATION in undefined order.

ANS:

SELECT \* FROM STATION;

OUTPUT:

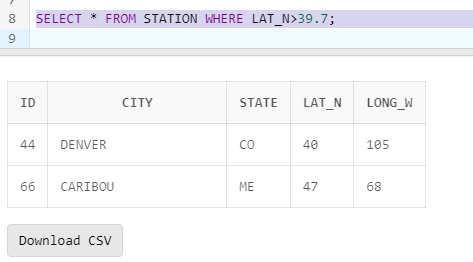


Q-4:Execute the query to select Northern stations (Northern latitude>39.7)

ANS:

SELECT \* FROM STATION WHERE LAT\_N>39.7;

OUTPUT:



Q-5:Create another table ‘STATS’, to store normalized temperature and percapitation data:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

There will be no Duplicate ID and MONTH combination.

ANS:

CREATE TABLE STATS

(

ID NUMBER,

MONTH NUMBER CHECK(MONTH BETWEEN 1 AND 12),

TEMP\_F NUMBER,

RAIN\_I NUMBER,

CONSTRAINT ID\_MON\_UNIQUE UNIQUE(ID,MONTH),

CONSTRAINT FK\_ID FOREIGN KEY(ID)

REFERENCES STATION(ID)

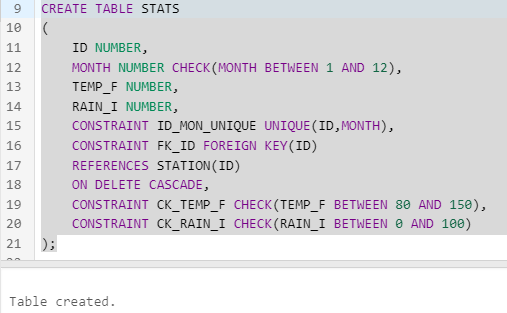
ON DELETE CASCADE,

CONSTRAINT CK\_TEMP\_F CHECK(TEMP\_F BETWEEN 80 AND 150),

CONSTRAINT CK\_RAIN\_I CHECK(RAIN\_I BETWEEN 0 AND 100)

);

OUTPUT:



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

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

ANS:

ALTER TABLE STATS

DROP CONSTRAINT CK\_TEMP\_F; -- DROP CONSTRAINT DUE LOW VALUE INSERTION IN TEMP\_F COLUMN

ALTER TABLE STATS

ADD CONSTRAINT CK\_TEMP\_F

CHECK(TEMP\_F BETWEEN 0 AND 150);-- CHANGE IT TO 0 TO 150 DUE TO ERROR IN INSERTION

INSERT INTO STATS(ID,MONTH,TEMP\_F,RAIN\_I) VALUES (13,1,57.4,.31);

INSERT INTO STATS(ID,MONTH,TEMP\_F,RAIN\_I) VALUES (13,7,91.7,5.15);

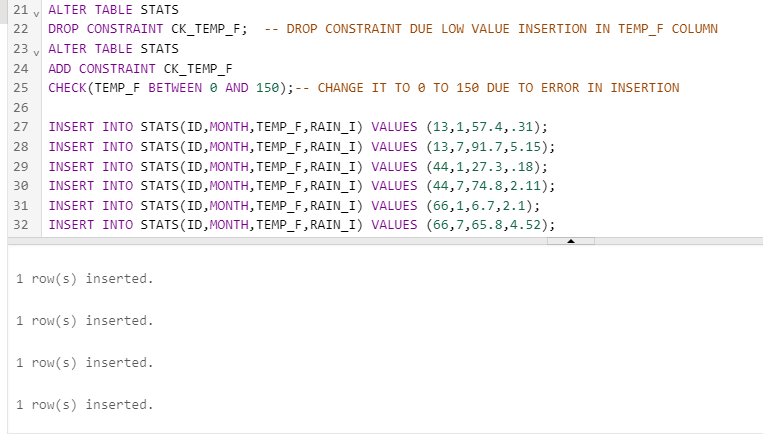
INSERT INTO STATS(ID,MONTH,TEMP\_F,RAIN\_I) VALUES (44,1,27.3,.18);

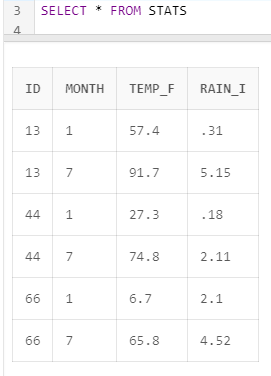
INSERT INTO STATS(ID,MONTH,TEMP\_F,RAIN\_I) VALUES (44,7,74.8,2.11);

INSERT INTO STATS(ID,MONTH,TEMP\_F,RAIN\_I) VALUES (66,1,6.7,2.1);

INSERT INTO STATS(ID,MONTH,TEMP\_F,RAIN\_I) VALUES (66,7,65.8,4.52);

OUTPUT:



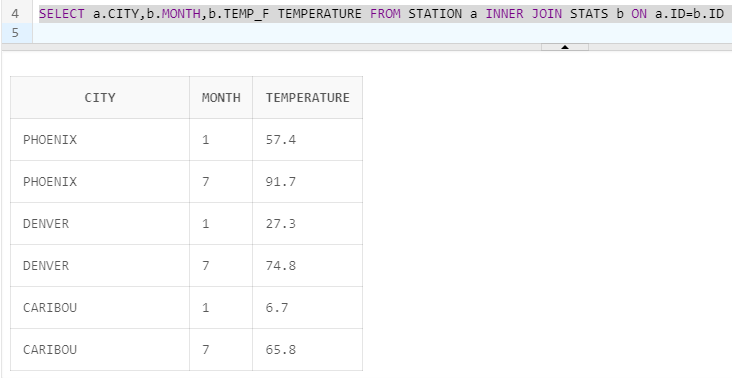


Q-7:Execute query to display temperature stats(from the STATS table) for each city(for the STATION table)

ANS:

SELECT a.CITY,b.MONTH,b.TEMP\_F TEMPERATURE FROM STATION a INNER JOIN STATS b ON a.ID=b.ID;

OUTPUT:

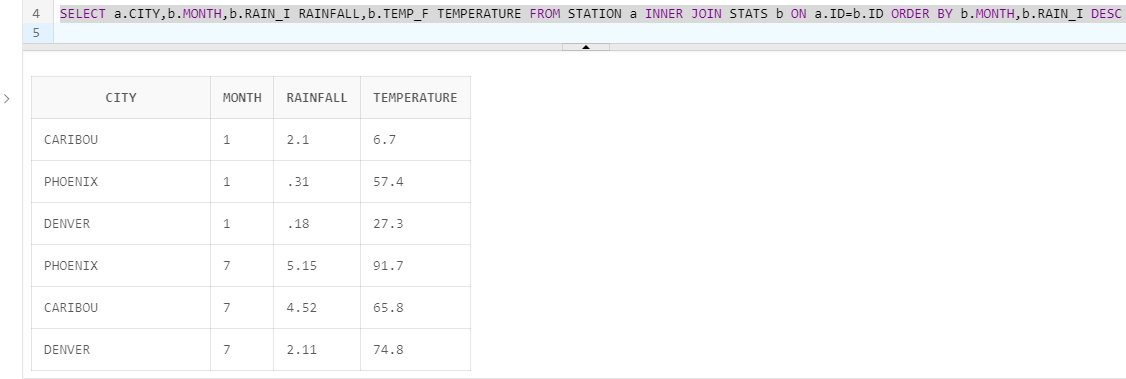


Q-8: Execute a query to look at the table STATS, ordered by month and greatest rainfall, with columns rearranged. It should also show the corresponding cities.

ANS:

SELECT a.CITY,b.MONTH,b.RAIN\_I RAINFALL,b.TEMP\_F TEMPERATURE FROM STATION a INNER JOIN STATS b ON a.ID=b.ID ORDER BY b.MONTH,b.RAIN\_I DESC;

OUTPUT:



Q-9:Execute a query to look at temperature for July from tables STATS, lowest temperature first, picking up city name and latitude.

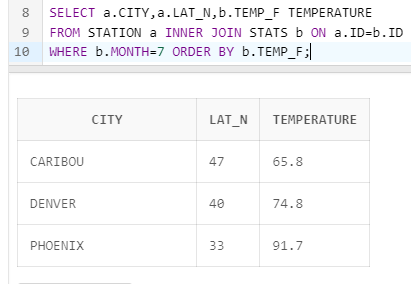
ANS:

SELECT a.CITY,a.LAT\_N,b.TEMP\_F TEMPERATURE

FROM STATION a INNER JOIN STATS b ON a.ID=b.ID

WHERE b.MONTH=7 ORDER BY b.TEMP\_F;

OUTPUT:



Q-10: Execute a query to show MAX and MIN temperatures as well as average rainfall for each city.

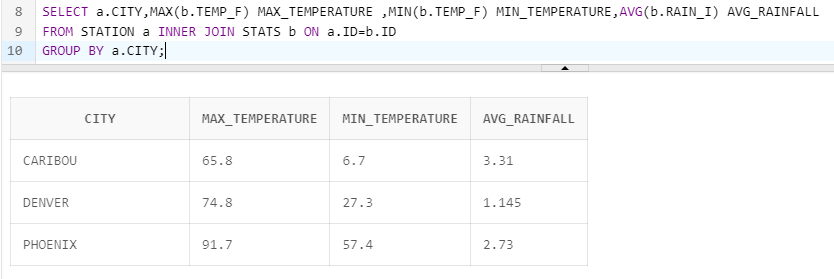
ANS:

SELECT a.CITY,MAX(b.TEMP\_F) MAX\_TEMPERATURE ,MIN(b.TEMP\_F) MIN\_TEMPERATURE,AVG(b.RAIN\_I) AVG\_RAINFALL

FROM STATION a INNER JOIN STATS b ON a.ID=b.ID

GROUP BY a.CITY;

OUTPUT:



Q-11: Execute a query to display each city’s monthly temperature in Celsius and rainfall in Centimetre.

ANS:

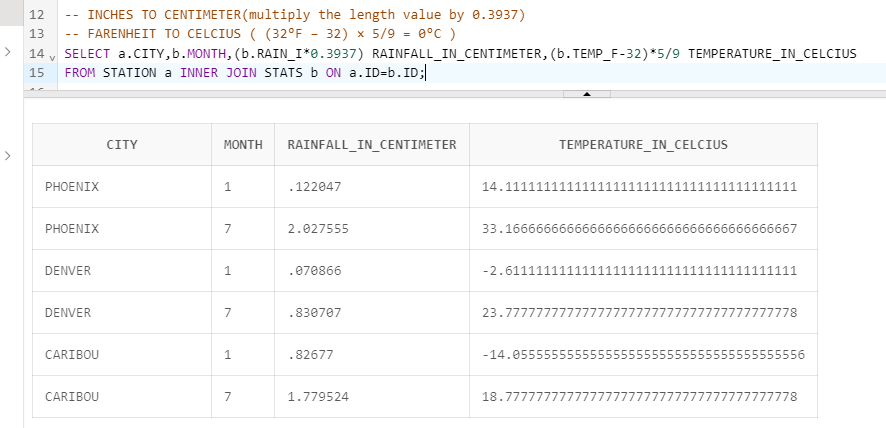
-- INCHES TO CENTIMETER(multiply the length value by 0.3937)

-- FARENHEIT TO CELCIUS ( (32°F − 32) × 5/9 = 0°C )

SELECT a.CITY,b.MONTH,(b.RAIN\_I\*0.3937) RAINFALL\_IN\_CENTIMETER,(b.TEMP\_F-32)\*5/9 TEMPERATURE\_IN\_CELCIUS

FROM STATION a INNER JOIN STATS b ON a.ID=b.ID;

OUTPUT:

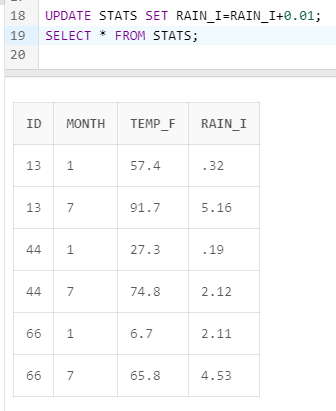


Q-12: Update all rows of table STATS to compensate for faulty rain gauges known to read 0.01 inches low.

ANS:

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

OUTPUT:



Q-13: Update DENVER’s July temperature reading as 74.9.

ANS:

UPDATE STATS SET TEMP\_F=74.9 WHERE MONTH=7 AND ID IN (SELECT ID FROM STATION WHERE CITY='DENVER'); -- UPDATE

SELECT a.CITY,b.TEMP\_F TEMPERATURE

FROM STATION a INNER JOIN STATS b ON a.ID=b.ID

WHERE b.MONTH=7 AND a.CITY='DENVER'; --SHOWING RESU

OUTPUT:

