SQL ASSIGNMENT -MAJOR

1. **Create a table “Station” to store information about weather observation stations. Ans:**

Query

Output

CREATE TABLE **Station**

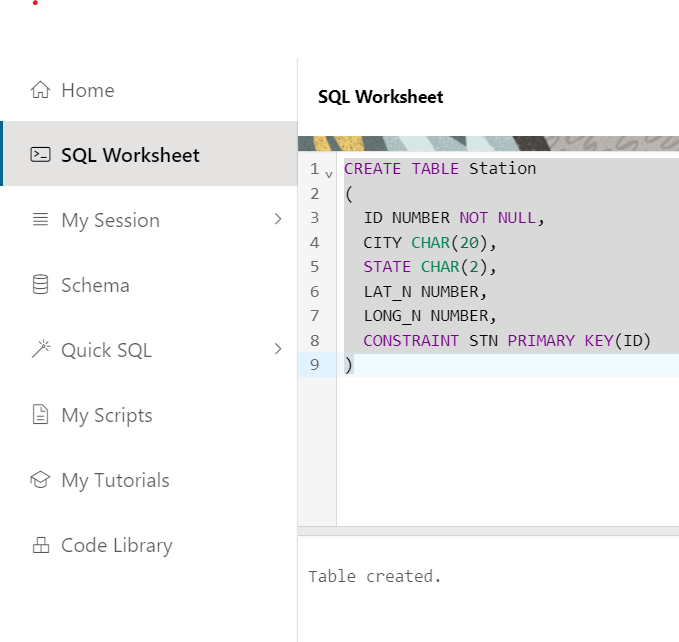
(

ID NUMBER NOT NULL, CITY CHAR (20),

STATE CHAR (2), LAT\_N NUMBER, LONG\_N NUMBER,

CONSTRAINT STN PRIMARY KEY(ID)

);



1. **Insert the following records into the table**

Ans:

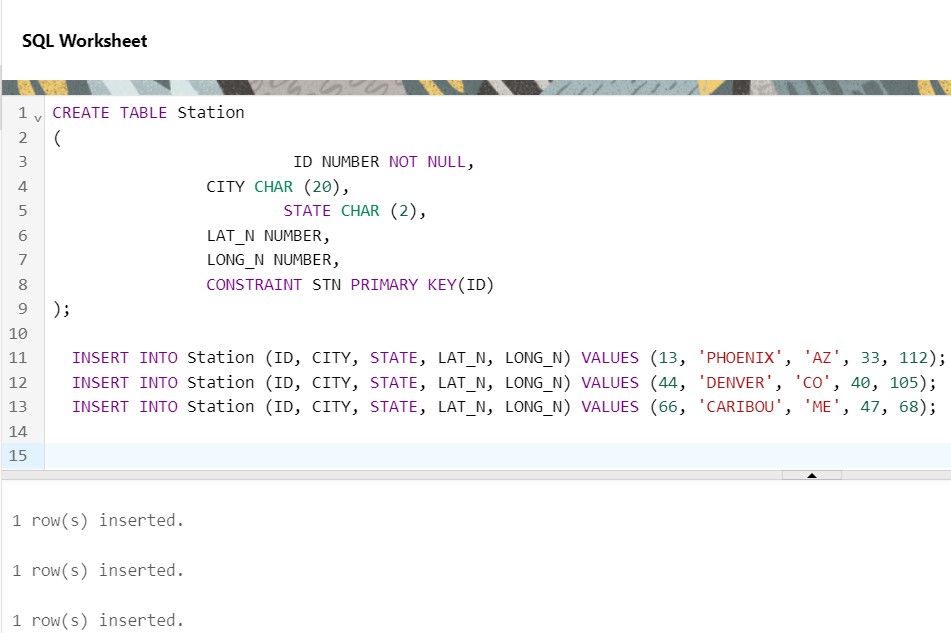
Query:

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

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

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

Output:

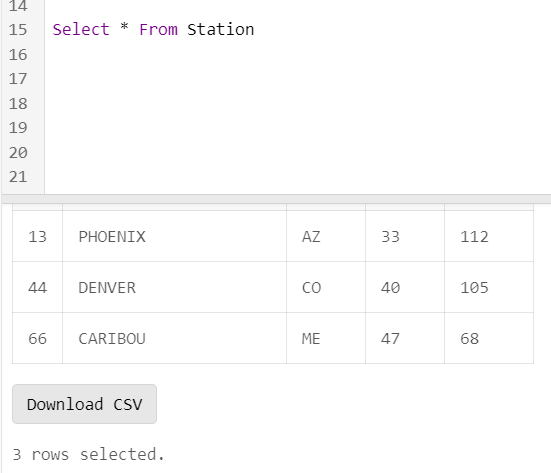


1. **Execute a query to look at table Station in undefined order. Ans:**

Query:

SELECT \* FROM Station

Output:

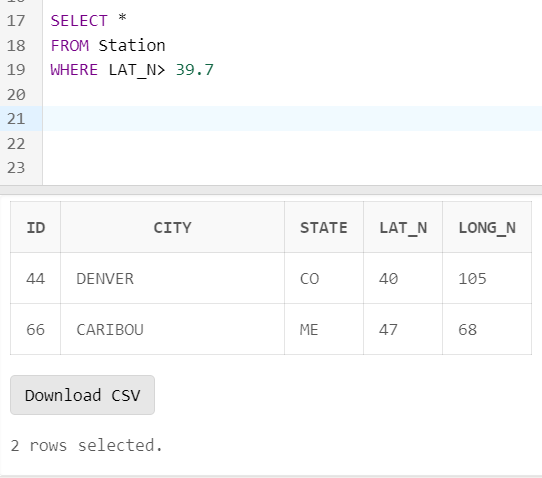


1. **Execute query to select northern stations (Northern Latitude > 39.7) Ans:**

Query:

Output:

SELECT \* FROM Station WHERE LAT\_N > 39.7



1. **Create another table, ‘STATS’, to store normalized temperature and precipitation**

data: There will be no Duplicate ID and MONTH combination.

Ans:

Query:

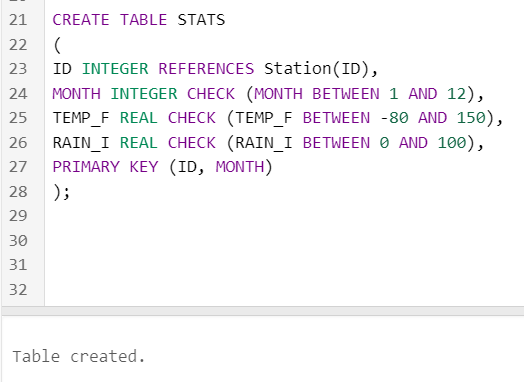
Output:

CREATE TABLE STATS (

ID INTEGER REFERENCES Station(ID),

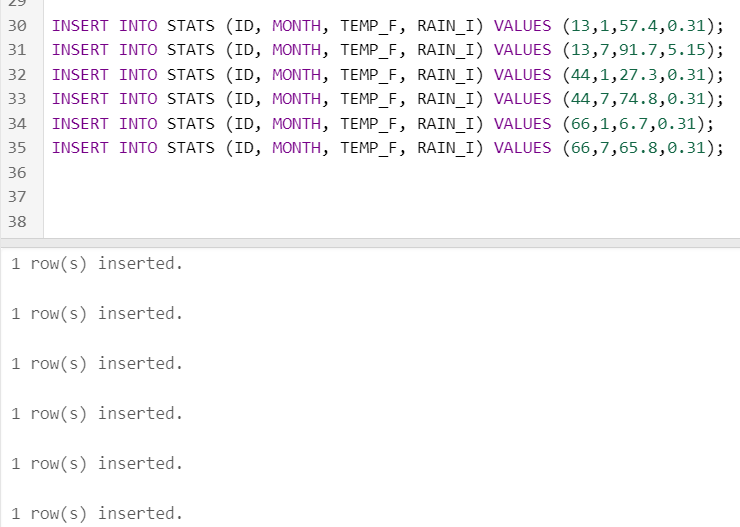
MONTH INTEGER CHECK (MONTH BETWEEN 1 AND 12), TEMP\_F REAL CHECK (TEMP\_F BETWEEN -80 AND 150), RAIN\_I REAL CHECK (RAIN\_I BETWEEN 0 AND 100), PRIMARY KEY (ID, MONTH)

);



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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ans:**  **Query:** |  | | | | | |
|  | INSERT INTO STATS | (ID, | MONTH, | TEMP\_F, | RAIN\_I) | VALUES |
|  | (13,1,57.4,0.31); |  |  |  |  |  |
|  | INSERT INTO STATS (13,7,91.7,5.15); | (ID, | MONTH, | TEMP\_F, | RAIN\_I) | VALUES |
|  | INSERT INTO STATS (44,1,27.3,0.31); | (ID, | MONTH, | TEMP\_F, | RAIN\_I) | VALUES |
|  | INSERT INTO STATS (44,7,74.8,0.31); | (ID, | MONTH, | TEMP\_F, | RAIN\_I) | VALUES |
|  | INSERT INTO STATS (66,1,6.7,0.31); | (ID, | MONTH, | TEMP\_F, | RAIN\_I) | VALUES |
|  | INSERT INTO STATS (66,7,65.8,0.31); | (ID, | MONTH, | TEMP\_F, | RAIN\_I) | VALUES |
| **Output:** |  |  |  |  |  |  |



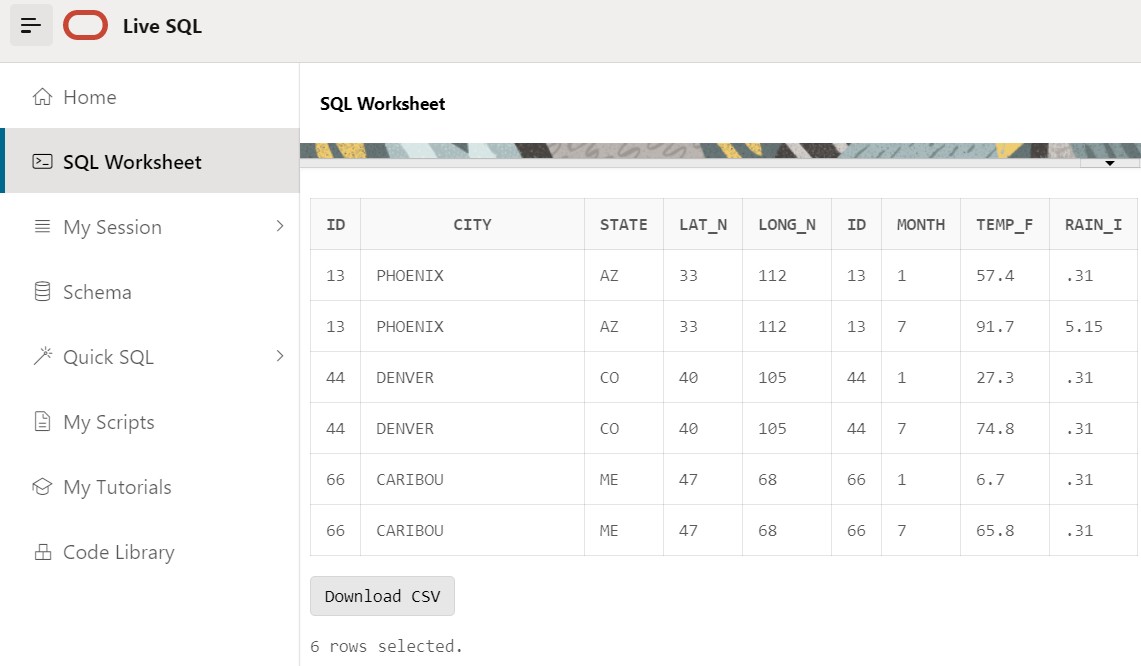
1. **Execute a query to display temperature stats (from STATS table) for each city (from Station table).**

Ans:

Query:

Output:

SELECT \* FROM Station, STATS WHERE Station.ID = STATS.ID



1. **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.**

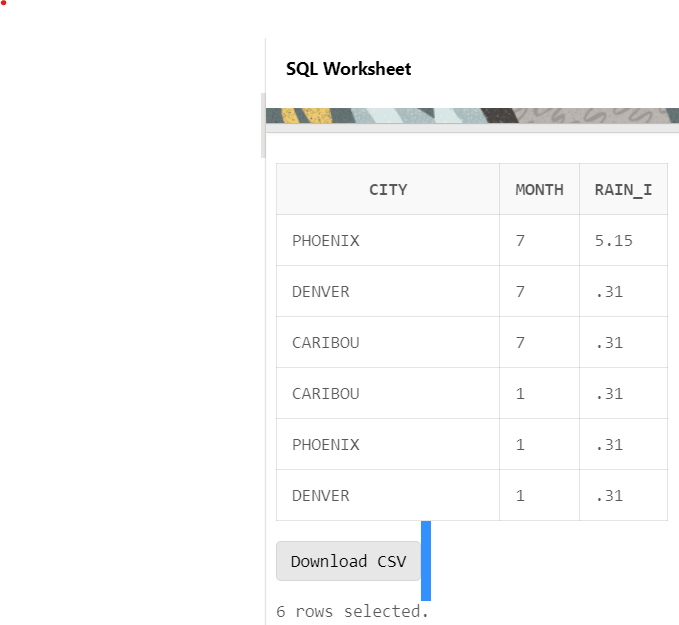
Ans:

Query:

SELECT CITY, MONTH, RAIN\_I

FROM Station, STATS WHERE Station.ID = STATS.ID

ORDER BY MONTH DESC, RAIN\_I DESC;

Output:

1. **Execute a query to look at temperatures for July from table STATS, lowest temperatures first, picking up city name and latitude.**

Ans:

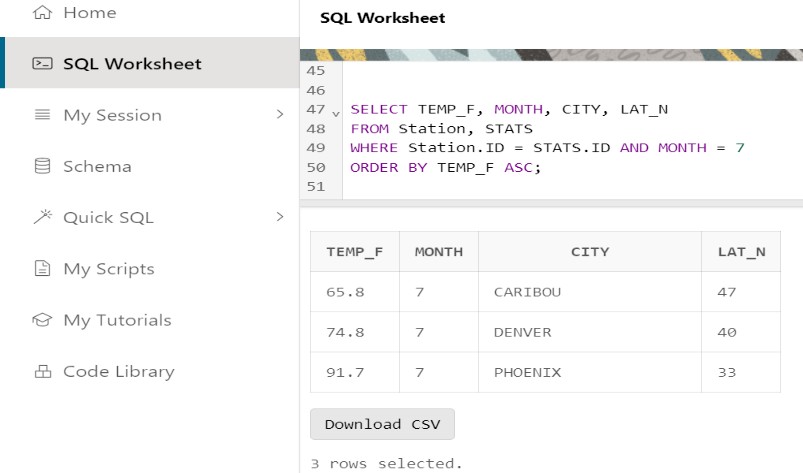
Query:

Output:

SELECT TEMP\_F, MONTH, CITY, LAT\_N

FROM Station, STATS

WHERE Station.ID = STATS.ID AND MONTH = 7 ORDER BY TEMP\_F ASC;



1. **Execute a query to show MAX and MIN temperatures as well as average rainfall for each city.**

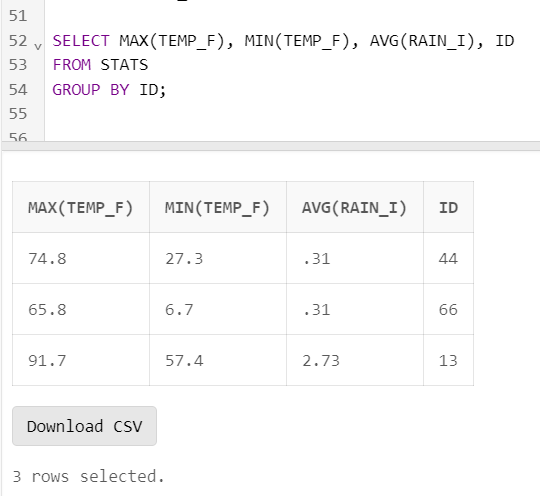
Ans:

Query:

Output:

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

GROUP BY ID;



1. **Execute a query to display each city’s monthly temperature in Celsius and rainfall in Centimeter.**

Ans:

Query:

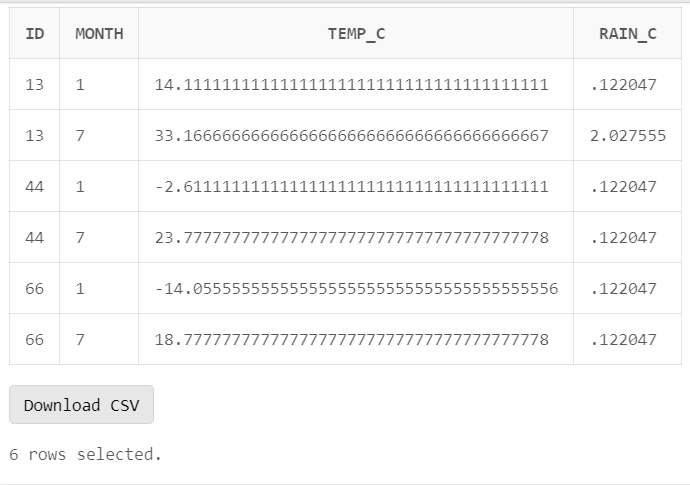
CREATE VIEW CRITERIA\_STATS (ID, MONTH, TEMP\_C, RAIN\_C) AS SELECT ID, MONTH,

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

RAIN\_I \* 0.3937 FROM STATS;

SELECT \*FROM CRITERIA\_STATS

Output:

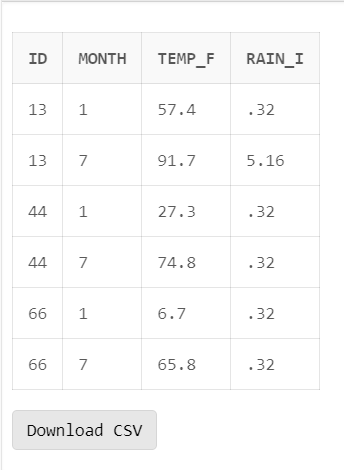


1. **Update all rows of table STATS to compensate for faulty rain gauges known to read 0.01 inches low.**

Ans:

**Query:** UPDATE STATS SET RAIN\_I = RAIN\_I + 0.01; SELECT \* FROM STATS

Output:



1. **Update Denver's July temperature reading as 74.9 Ans:**

Query:

Output:

UPDATE STATS SET TEMP\_F = 74.9

WHERE MONTH = 7 AND ID = 44; SELECT \* FROM STATS;

