

SQL ASSIGNMENT -MAJOR

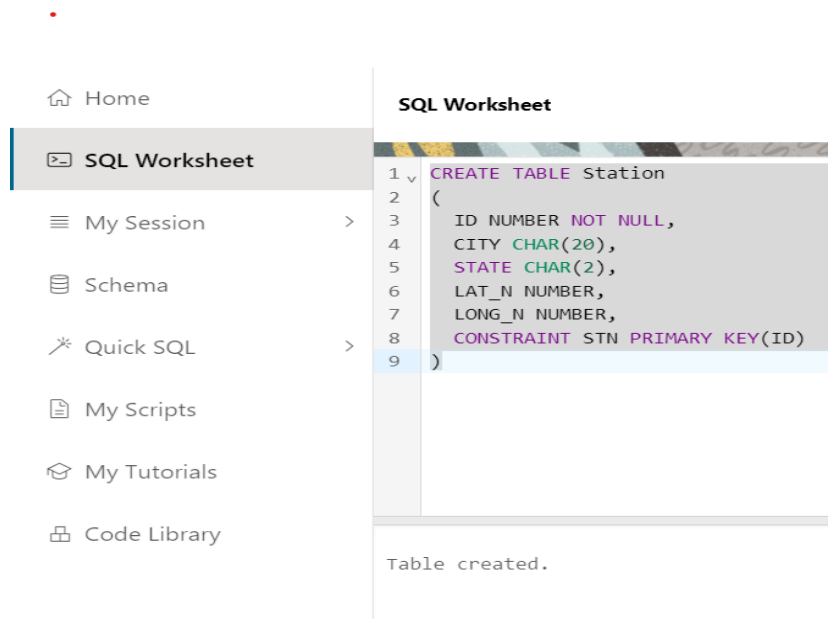
1. Create a table "Station" to store information about weather observation stations.

Ans:

Query

```
CREATE TABLE Station
(
    ID NUMBER NOT NULL,
    CITY CHAR (20),
    STATE CHAR (2),
    LAT_N NUMBER,
    LONG_N NUMBER,
    CONSTRAINT STN PRIMARY KEY(ID)
);
```

Output



2. 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:

```
SQL Worksheet

1 CREATE TABLE Station
2 (
3     ID NUMBER NOT NULL,
4     CITY CHAR (20),
5     STATE CHAR (2),
6     LAT_N NUMBER,
7     LONG_N NUMBER,
8     CONSTRAINT STN PRIMARY KEY(ID)
9 );
10
11 INSERT INTO Station (ID, CITY, STATE, LAT_N, LONG_N) VALUES (13, 'PHOENIX', 'AZ', 33, 112);
12 INSERT INTO Station (ID, CITY, STATE, LAT_N, LONG_N) VALUES (44, 'DENVER', 'CO', 40, 105);
13 INSERT INTO Station (ID, CITY, STATE, LAT_N, LONG_N) VALUES (66, 'CARIBOU', 'ME', 47, 68);
14
15

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

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

Ans:

Query:

```
SELECT * FROM Station
```

Output:

```
14
15  select * from Station
16
17
18
19
20
21
```

13	PHOENIX	AZ	33	112
44	DENVER	CO	40	105
66	CARIBOU	ME	47	68

[Download CSV](#)

3 rows selected.

4. Execute query to select northern stations (Northern Latitude > 39.7)

Ans:

Query:

```
SELECT * FROM Station
WHERE LAT_N > 39.7
```

Output:

```
17  SELECT *
18  FROM Station
19  WHERE LAT_N > 39.7
20
21
22
23
```

ID	CITY	STATE	LAT_N	LONG_N
44	DENVER	CO	40	105
66	CARIBOU	ME	47	68

[Download CSV](#)

2 rows selected.

5. Create another table, 'STATS', to store normalized temperature and precipitation data: There will be no Duplicate ID and MONTH combination.

Ans:

Query:

```
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)
);
```

Output:

```
21 CREATE TABLE STATS
22 (
23 ID INTEGER REFERENCES Station(ID),
24 MONTH INTEGER CHECK (MONTH BETWEEN 1 AND 12),
25 TEMP_F REAL CHECK (TEMP_F BETWEEN -80 AND 150),
26 RAIN_I REAL CHECK (RAIN_I BETWEEN 0 AND 100),
27 PRIMARY KEY (ID, MONTH)
28 );
29
30
31
32
```

Table created.

6. 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 (ID, MONTH, TEMP_F, RAIN_I) VALUES  
(13,7,91.7,5.15);
```

```
INSERT INTO STATS (ID, MONTH, TEMP_F, RAIN_I) VALUES  
(44,1,27.3,0.31);
```

```
INSERT INTO STATS (ID, MONTH, TEMP_F, RAIN_I) VALUES  
(44,7,74.8,0.31);
```

```
INSERT INTO STATS (ID, MONTH, TEMP_F, RAIN_I) VALUES  
(66,1,6.7,0.31);
```

```
INSERT INTO STATS (ID, MONTH, TEMP_F, RAIN_I) VALUES  
(66,7,65.8,0.31);
```

Output:

```
30 INSERT INTO STATS (ID, MONTH, TEMP_F, RAIN_I) VALUES (13,1,57.4,0.31);  
31 INSERT INTO STATS (ID, MONTH, TEMP_F, RAIN_I) VALUES (13,7,91.7,5.15);  
32 INSERT INTO STATS (ID, MONTH, TEMP_F, RAIN_I) VALUES (44,1,27.3,0.31);  
33 INSERT INTO STATS (ID, MONTH, TEMP_F, RAIN_I) VALUES (44,7,74.8,0.31);  
34 INSERT INTO STATS (ID, MONTH, TEMP_F, RAIN_I) VALUES (66,1,6.7,0.31);  
35 INSERT INTO STATS (ID, MONTH, TEMP_F, RAIN_I) VALUES (66,7,65.8,0.31);  
36  
37  
38
```

```
1 row(s) inserted.
```

```
1 row(s) inserted.
```

```
1 row(s) inserted.
```

```
1 row(s) inserted.
```

```
1 row(s) inserted.
```

```
1 row(s) inserted.
```

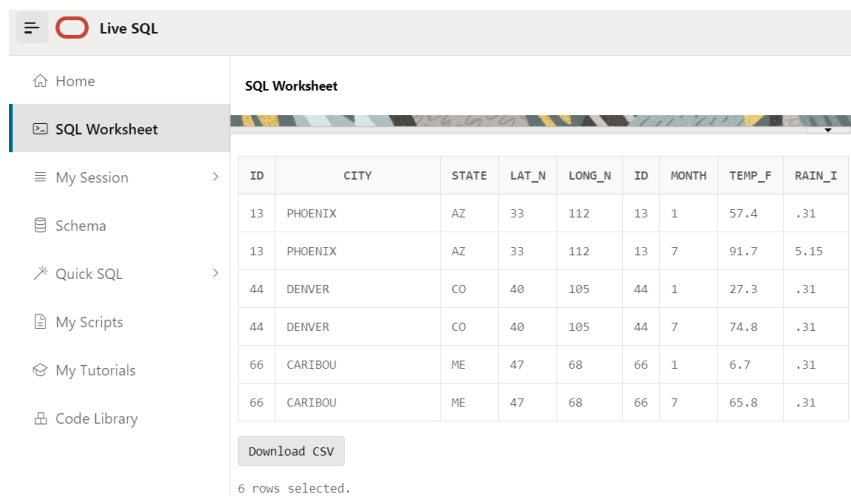
7. Execute a query to display temperature stats (from STATS table) for each city (from Station table).

Ans:

Query:

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

Output:



ID	CITY	STATE	LAT_N	LONG_N	ID	MONTH	TEMP_F	RAIN_I
13	PHOENIX	AZ	33	112	13	1	57.4	.31
13	PHOENIX	AZ	33	112	13	7	91.7	5.15
44	DENVER	CO	40	105	44	1	27.3	.31
44	DENVER	CO	40	105	44	7	74.8	.31
66	CARIBOU	ME	47	68	66	1	6.7	.31
66	CARIBOU	ME	47	68	66	7	65.8	.31

Download CSV

6 rows selected.

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:

Query:

```
SELECT CITY, MONTH, RAIN_I  
FROM Station, STATS  
WHERE Station.ID = STATS.ID  
ORDER BY MONTH DESC, RAIN_I DESC;
```

Output:

SQL Worksheet

CITY	MONTH	RAIN_I
PHOENIX	7	5.15
DENVER	7	.31
CARIBOU	7	.31
CARIBOU	1	.31
PHOENIX	1	.31
DENVER	1	.31

Download CSV

6 rows selected.

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

Ans:

Query:

```
SELECT TEMP_F, MONTH, CITY, LAT_N
FROM Station, STATS
WHERE Station.ID = STATS.ID AND MONTH = 7
ORDER BY TEMP_F ASC;
```

Output:

SQL Worksheet

```
45
46
47 SELECT TEMP_F, MONTH, CITY, LAT_N
48 FROM Station, STATS
49 WHERE Station.ID = STATS.ID AND MONTH = 7
50 ORDER BY TEMP_F ASC;
51
```

TEMP_F	MONTH	CITY	LAT_N
65.8	7	CARIBOU	47
74.8	7	DENVER	40
91.7	7	PHOENIX	33

Download CSV

3 rows selected.

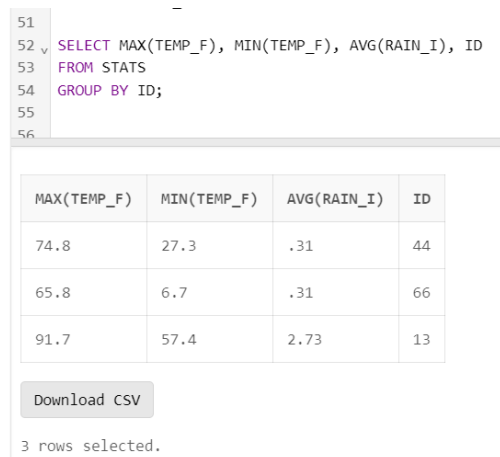
10. Execute a query to show MAX and MIN temperatures as well as average rainfall for each city.

Ans:

Query:

```
SELECT MAX(TEMP_F), MIN(TEMP_F), AVG(RAIN_I), ID
FROM STATS
GROUP BY ID;
```

Output:



The screenshot shows a SQL query editor with the following text:

```
51 --
52 v SELECT MAX(TEMP_F), MIN(TEMP_F), AVG(RAIN_I), ID
53 FROM STATS
54 GROUP BY ID;
55
56
```

Below the editor is a table with the following data:

MAX(TEMP_F)	MIN(TEMP_F)	AVG(RAIN_I)	ID
74.8	27.3	.31	44
65.8	6.7	.31	66
91.7	57.4	2.73	13

Below the table is a button labeled "Download CSV" and a status message: "3 rows selected."

11. 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:

[illegible][Download CSV](#)

```
6 rows selected.
```

12. 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:

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	.32
13	7	91.7	5.16
44	1	27.3	.32
44	7	74.8	.32
66	1	6.7	.32
66	7	65.8	.32

[Download CSV](#)

13. Update Denver's July temperature reading as 74.9

Ans:

Query:

```
UPDATE STATS  
SET TEMP_F = 74.9  
WHERE MONTH = 7 AND ID = 44;  
SELECT * FROM STATS;
```

Output:

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	.32
13	7	91.7	5.16
44	1	27.3	.32
44	7	74.9	.32
66	1	6.7	.32
66	7	65.8	.32

Download CSV