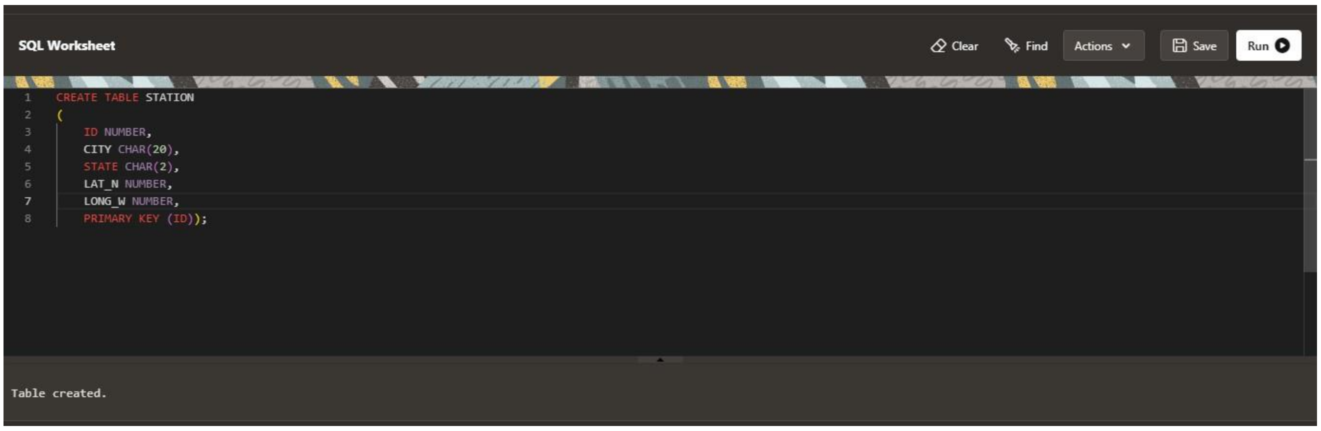
Assignment – Advance SQL [Major] by Hemant Raikwar

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

ID	Number	Primary key
CITY	CHAR(20)	
STATE	CHAR(2)	
LAT_N	Number	

QUERY -

```
CREATE TABLE STATION
(
    ID NUMBER,
    CITY CHAR(20),
    STATE CHAR(2),
    LAT_N NUMBER,
    LONG_W NUMBER,
    PRIMARY KEY (ID));
```



```
CREATE TABLE STATION
(
    ID NUMBER,
    CITY CHAR(20),
    STATE CHAR(2),
    LAT_N NUMBER,
    LONG_W NUMBER,
    PRIMARY KEY (ID));
```

```
Table created.
```

2. Insert the following records into the table:

ID	CITY	STATE	LAT_N	LONG_W
13	PHONEIX	AZ	33	112
44	DENVER	CO	40	105
66	CARIBOU	ME	47	68

QUERY--

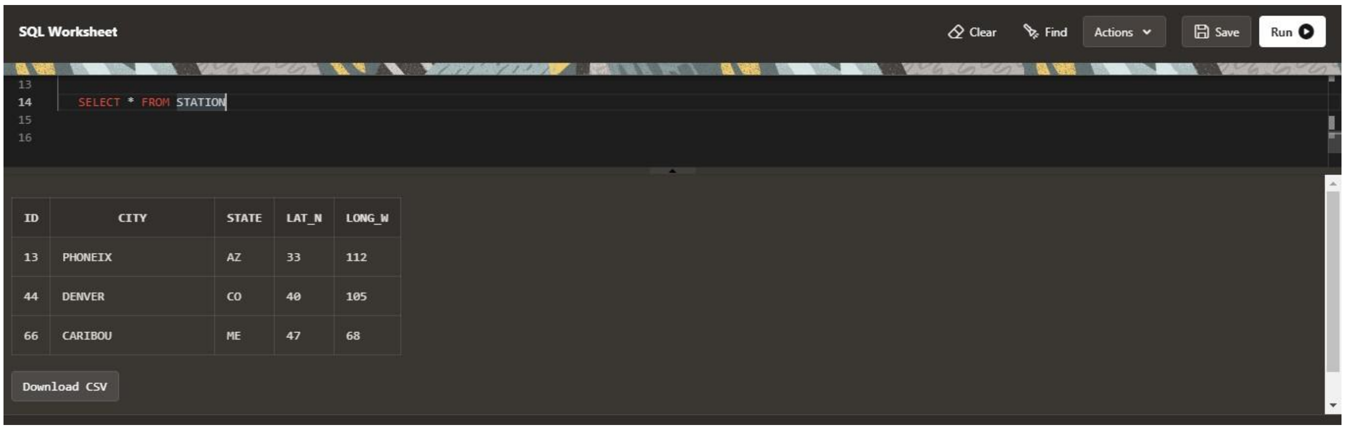
```
INSERT INTO STATION (ID, CITY, STATE, LAT_N, LONG_W) VALUES (13, 'PHONEIX', 'AZ', 33, 112);

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);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

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

QUERY -

SELECT * FROM STATION



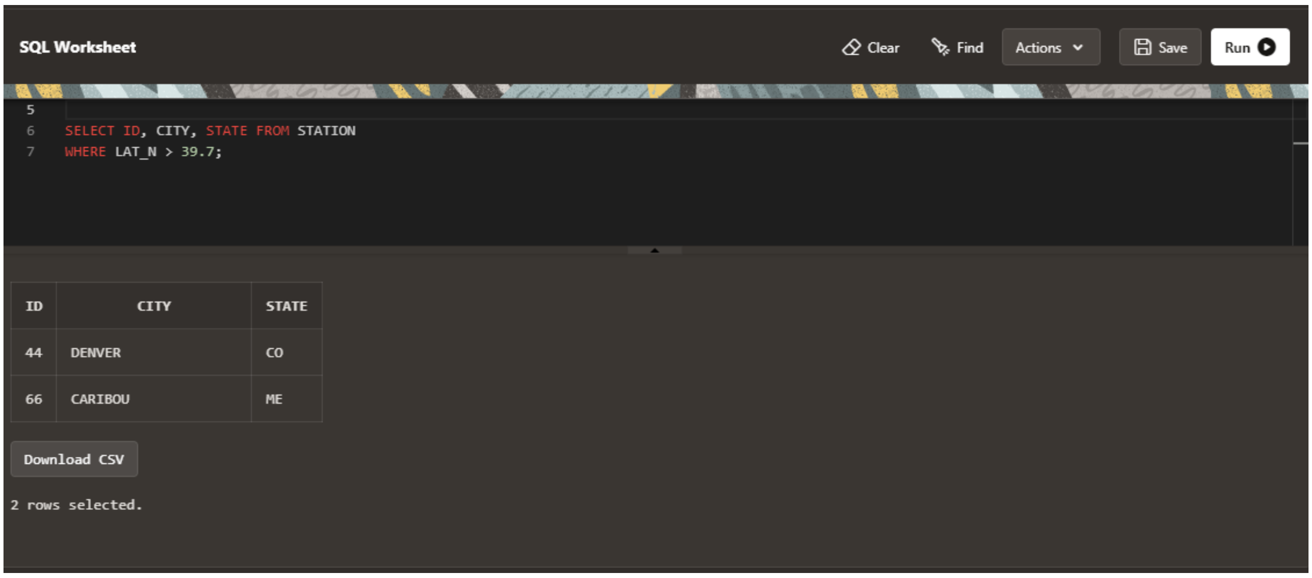
```
SELECT * FROM STATION
```

ID	CITY	STATE	LAT_N	LONG_W
13	PHONEIX	AZ	33	112
44	DENVER	CO	40	105
66	CARIBOU	ME	47	68

4. Execute a query to select Northern stations (Northern latitude >39.7).

QUERY-

SELECT ID, CITY, STATE FROM STATION
WHERE LAT_N > 39.7;



```
SELECT ID, CITY, STATE FROM STATION
WHERE LAT_N > 39.7;
```

ID	CITY	STATE
44	DENVER	CO
66	CARIBOU	ME

```
2 rows selected.
```

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

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

Column	Data type	Remark
ID	Number	must match some STATION table ID(so name & location will be known).
MONTH	Number	Range between 1 and 12

TEMP_F	Number	in Fahrenheit degrees, Range between -80 and 150
RAIN_I	Number	in inches, Range between 0 and 100

There will be no Duplicate ID and MONTH combination.

QUERY –

CREATE TABLE STATS

(

ID NUMBER,

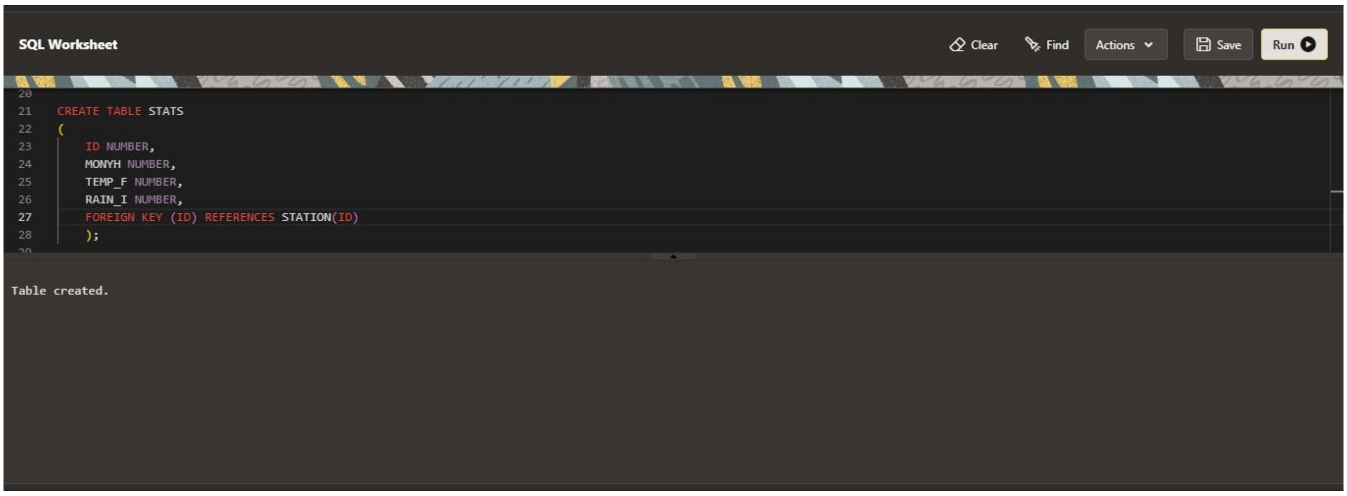
MONTH NUMBER,

TEMP_F NUMBER,

RAIN_I NUMBER,

FOREIGN KEY (ID) REFERENCES STATION(ID)

);



```
CREATE TABLE STATS
(
    ID NUMBER,
    MONYH NUMBER,
    TEMP_F NUMBER,
    RAIN_I NUMBER,
    FOREIGN KEY (ID) REFERENCES STATION(ID)
    );
```

```
Table created.
```

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

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	.31
13	7	91.7	5.15
44	1	27.3	.18
44	7	74.8	2.11
66	1	6.7	2.1
66	7	65.8	4.52

QUERY –

INSERT INTO STATS (ID,MONTH,TEMP_F,RAIN_I) VALUES (13,1,57.4,.31);

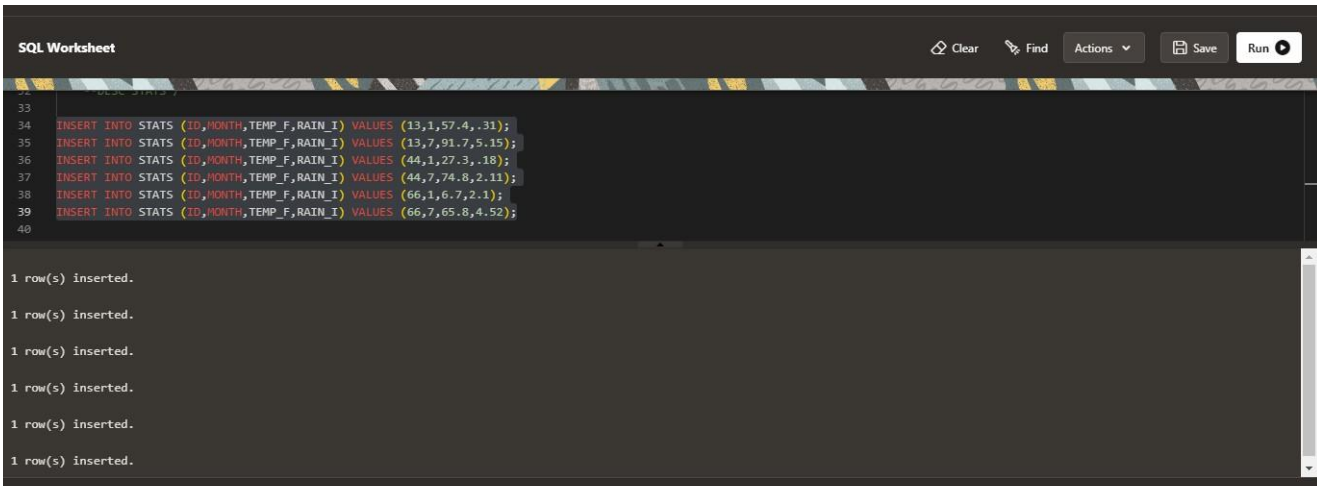
INSERT INTO STATS (ID,MONTH,TEMP_F,RAIN_I) VALUES (13,7,91.7,5.15);

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);



```
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);
```

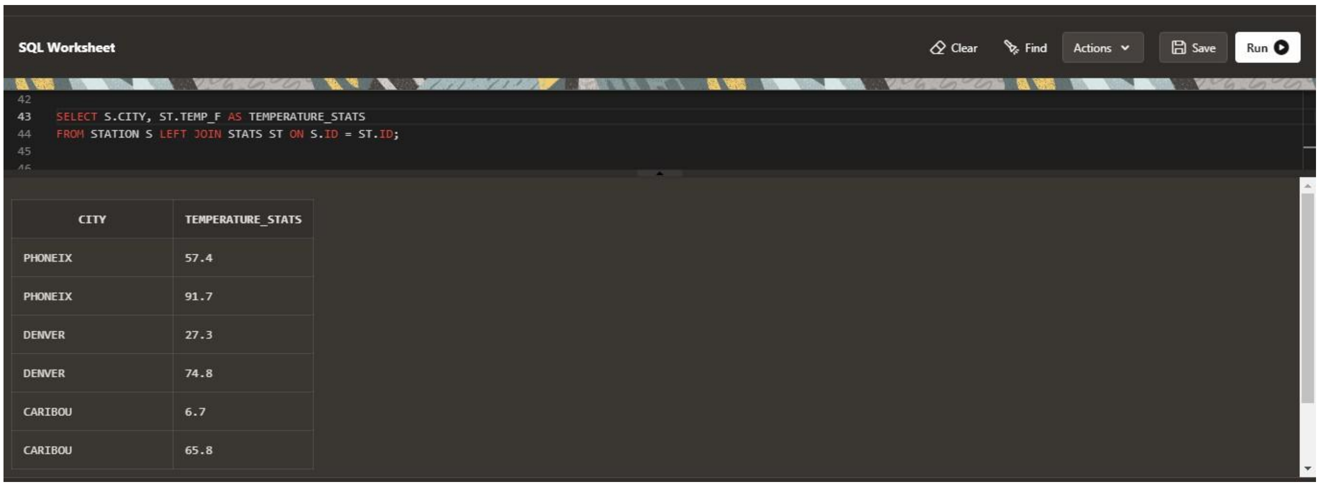
```
1 row(s) inserted.

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).QUERY-

SELECT S.CITY, ST.TEMP_F AS TEMPERATURE_STATS

FROM STATION S LEFT JOIN STATS ST ON S.ID = ST.ID;



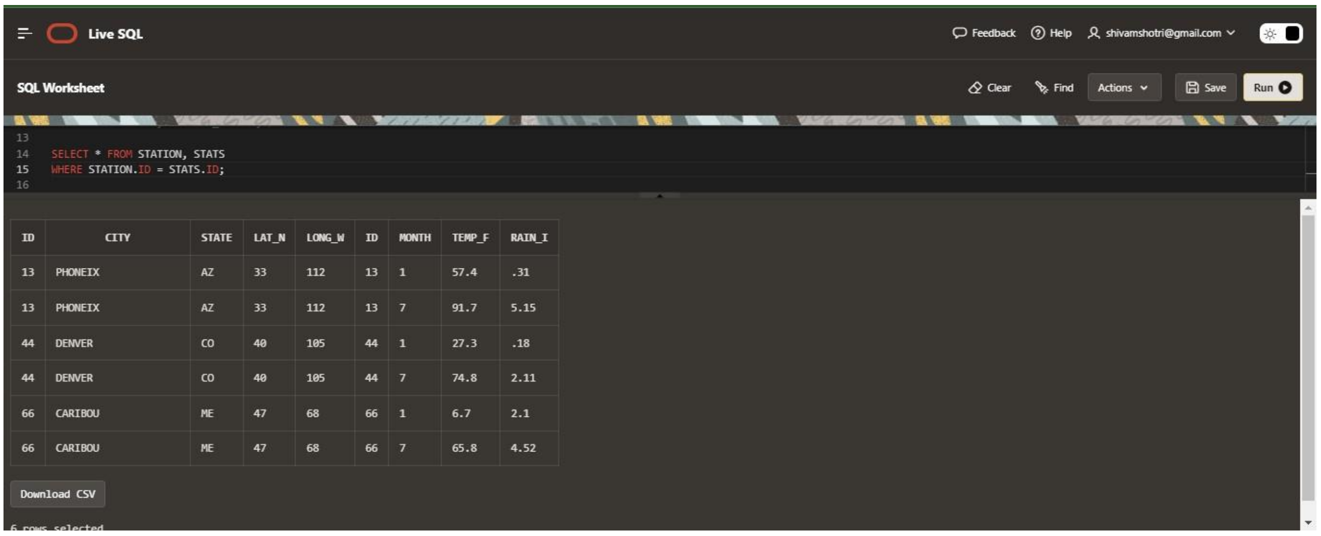
```
SELECT S.CITY, ST.TEMP_F AS TEMPERATURE_STATS
FROM STATION S LEFT JOIN STATS ST ON S.ID = ST.ID;
```

CITY	TEMPERATURE_STATS
PHONEIX	57.4
PHONEIX	91.7
DENVER	27.3
DENVER	74.8
CARIBOU	6.7
CARIBOU	65.8

QUERY –

SELECT * FROM STATION, STATS

WHERE STATION.ID=STATS.ID



```
SELECT * FROM STATION, STATS
WHERE STATION.ID = STATS.ID;
```

ID	CITY	STATE	LAT_N	LONG_W	ID	MONTH	TEMP_F	RAIN_I
13	PHONEIX	AZ	33	112	13	1	57.4	.31
13	PHONEIX	AZ	33	112	13	7	91.7	5.15
44	DENVER	CO	40	105	44	1	27.3	.18
44	DENVER	CO	40	105	44	7	74.8	2.11
66	CARIBOU	ME	47	68	66	1	6.7	2.1
66	CARIBOU	ME	47	68	66	7	65.8	4.52

```
6 rows selected.
```

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

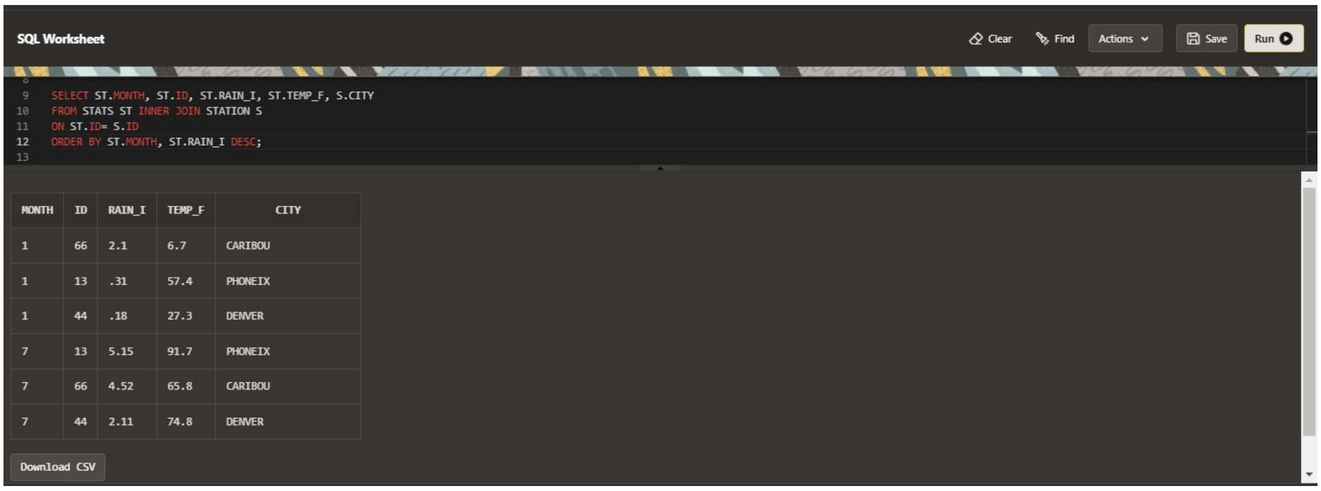
QUERY –

SELECT ST.MONTH, ST.ID, ST.RAIN_I, ST.TEMP_F, S.CITY

FROM STATS ST INNER JOIN STATION S

ON ST.ID= S.ID

ORDER BY ST.MONTH, ST.RAIN_I DESC;



```
SELECT ST.MONTH, ST.ID, ST.RAIN_I, ST.TEMP_F, S.CITY
FROM STATS ST INNER JOIN STATION S
ON ST.ID= S.ID
ORDER BY ST.MONTH, ST.RAIN_I DESC;
```

MONTH	ID	RAIN_I	TEMP_F	CITY
1	66	2.1	6.7	CARIBOU
1	13	.31	57.4	PHONEIX
1	44	.18	27.3	DENVER
7	13	5.15	91.7	PHONEIX
7	66	4.52	65.8	CARIBOU
7	44	2.11	74.8	DENVER

Download CSV

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

QUERY -

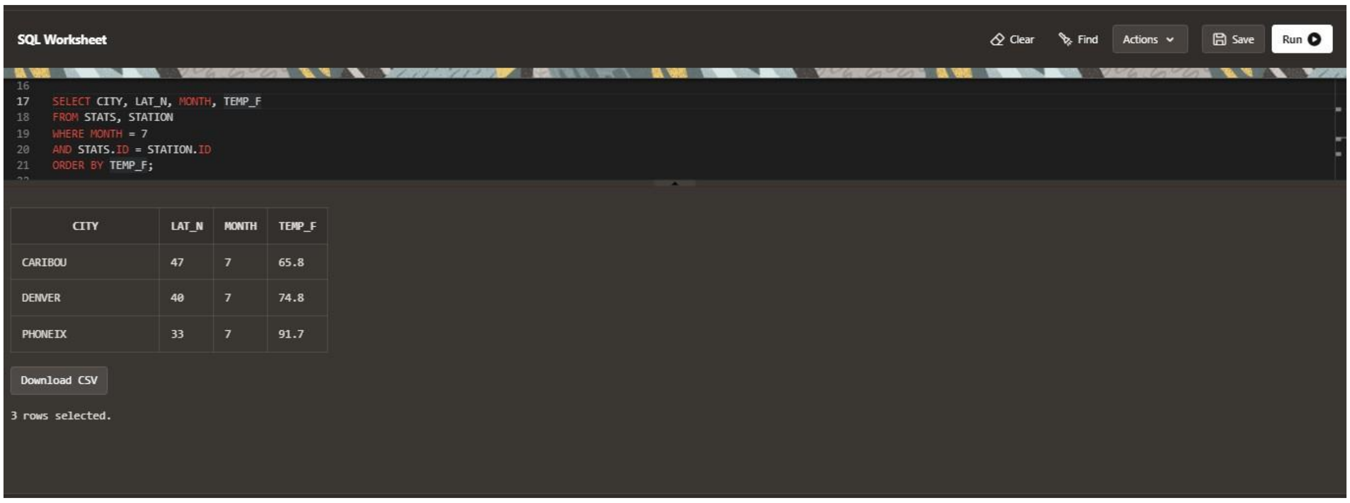
SELECT CITY, LAT_N, MONTH, TEMP_F

FROM STATS, STATION

WHERE MONTH = 7

AND STATS.ID = STATION.ID

ORDER BY TEMP_F;



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

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

Download CSV

```
3 rows selected.
```

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

SELECT ID, MAX(TEMP_F) AS MAX_TEMPERATURE,

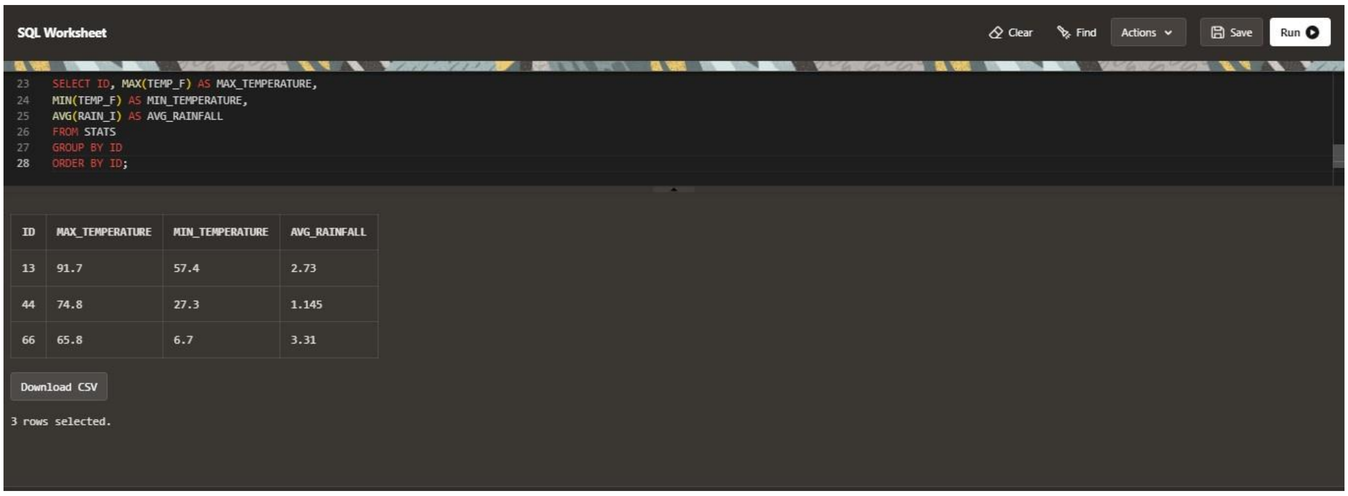
MIN(TEMP_F) AS MIN_TEMPERATURE,

AVG(RAIN_I) AS AVG_RAINFALL

FROM   STATS

GROUP BY ID

ORDER BY ID;



```
SELECT ID, MAX(TEMP_F) AS MAX_TEMPERATURE,
MIN(TEMP_F) AS MIN_TEMPERATURE,
AVG(RAIN_I) AS AVG_RAINFALL
FROM STATS
GROUP BY ID
ORDER BY ID;
```

ID	MAX_TEMPERATURE	MIN_TEMPERATURE	AVG_RAINFALL
13	91.7	57.4	2.73
44	74.8	27.3	1.145
66	65.8	6.7	3.31

```
3 rows selected.
```

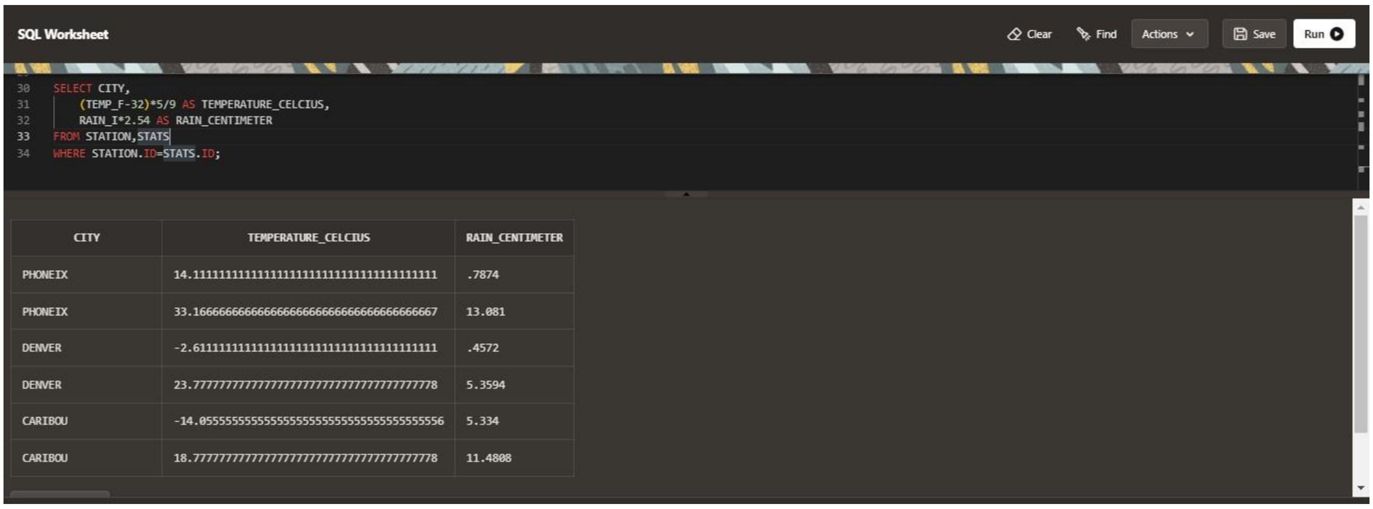
11. Execute a query to display each city's monthly temperature in Celcius and rainfall in Centimeter.QUERY –

SELECT CITY, (TEMP_F-32)*5/9 AS TEMPERATURE_CELCIUS,

RAIN_I*2.54 AS RAIN_CENTIMETER

FROM STATION,STATS

WHERE STATION.ID=STATS.ID;



```
SELECT CITY,
    (TEMP_F-32)*5/9 AS TEMPERATURE_CELCIUS,
    RAIN_I*2.54 AS RAIN_CENTIMETER
FROM STATION,STATS
WHERE STATION.ID=STATS.ID;
```

CITY	TEMPERATURE_CELCIUS	RAIN_CENTIMETER
PHONEIX	14.11111111111111111111111111111111111111	.7874
PHONEIX	33.16666666666666666666666666666666666667	13.081
DENVER	-2.61111111111111111111111111111111111111	.4572
DENVER	23.77777777777777777777777777777777777778	5.3594
CARIBOU	-14.05555555555555555555555555555555555556	5.334
CARIBOU	18.77777777777777777777777777777777777778	11.4808

NOTE – CONVERSION DETAILS OF FAHERNHEIT INTO CELCIUS

(TEMP. IN FAHERNHEIT -32) X 5/9

CONVERSION DETAILS OF INCHES TO CENTIMETRE

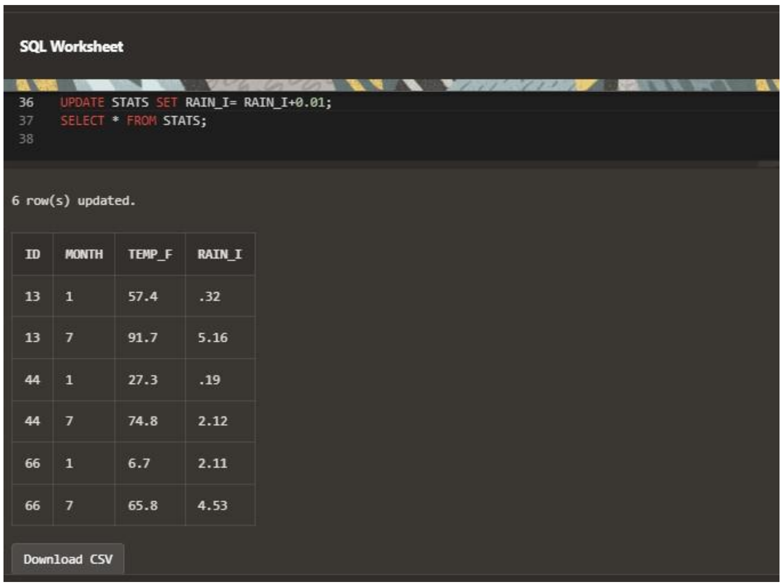
CENTIMETRE = INCHES X 2.54 (1 INCH = 2.54 CENTIMETRE)

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

QUERY –

UPDATE STATS SET RAIN_I = RAIN_I+0.01 ;

JUST TO SEE THE UPDATE - SELECT * FROM STATS;



```
UPDATE STATS SET RAIN_I= RAIN_I+0.01;
SELECT * FROM STATS;
```

```
6 row(s) updated.
```

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	.32
13	7	91.7	5.16
44	1	27.3	.19
44	7	74.8	2.12
66	1	6.7	2.11
66	7	65.8	4.53

13. 13. Update Denver's July temperature reading as 74.9.

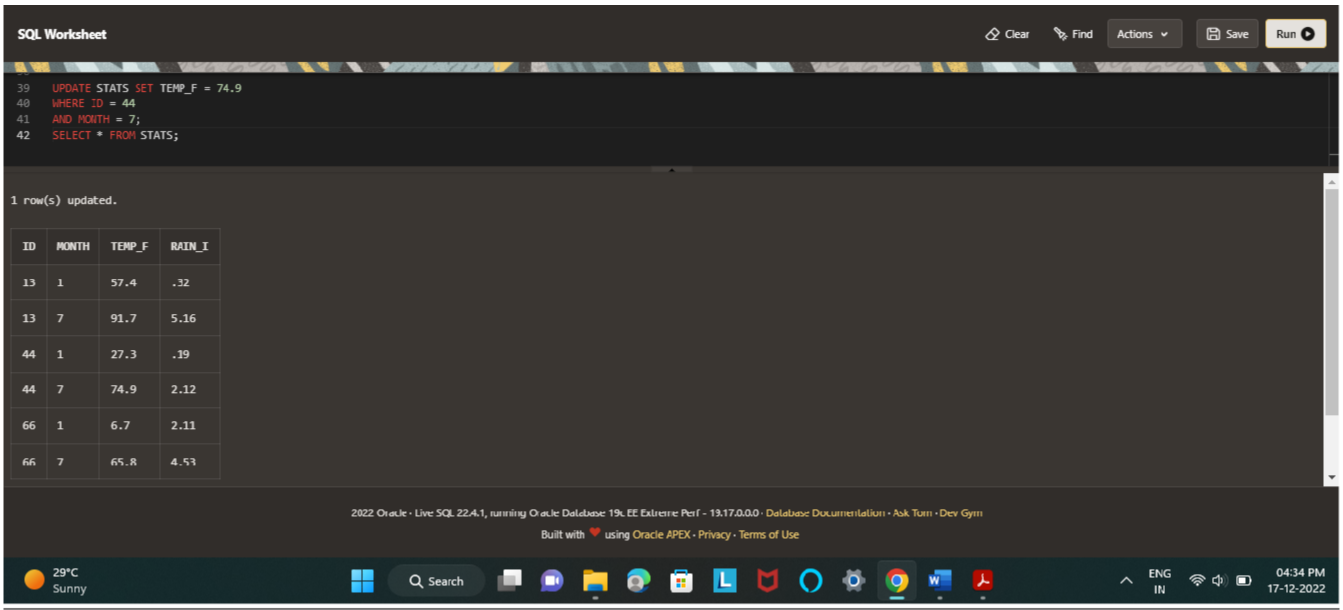
QUERY –

UPDATE STATS SET TEMP_F = 74.9

WHERE ID = 44

AND MONTH = 7;

JUST TO SEE THE UPDATE - SELECT * FROM STATS;



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

```
1 row(s) updated.
```

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	.32
13	7	91.7	5.16
44	1	27.3	.19
44	7	74.9	2.12
66	1	6.7	2.11
66	7	65.8	4.53

2022 Oracle · Live SQL 22.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym

Built with ♥ using Oracle APEX · Privacy · Terms of Use