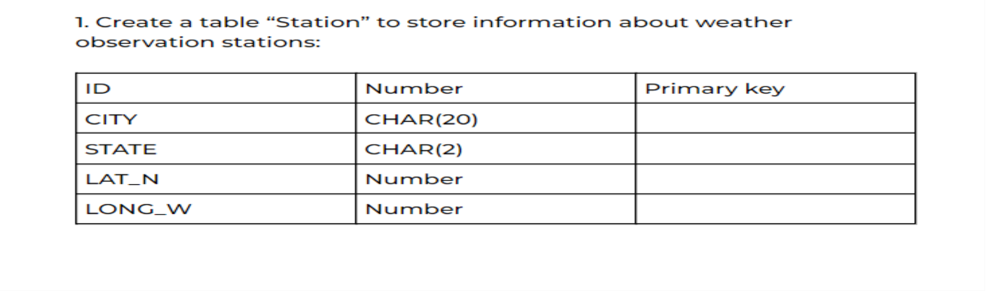
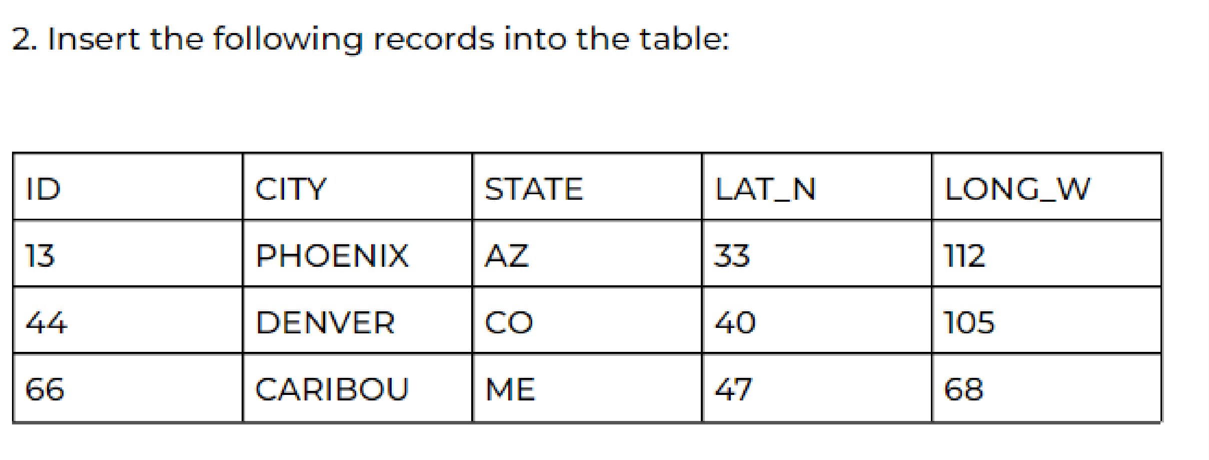


1. Create a table “Station” to store information about weather observation stations:

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

```
create table STATION
(    ID number,
    CITY char(20),
    STATE char(2),
    lAT_N Number,
    LONG_W   Number
)
```



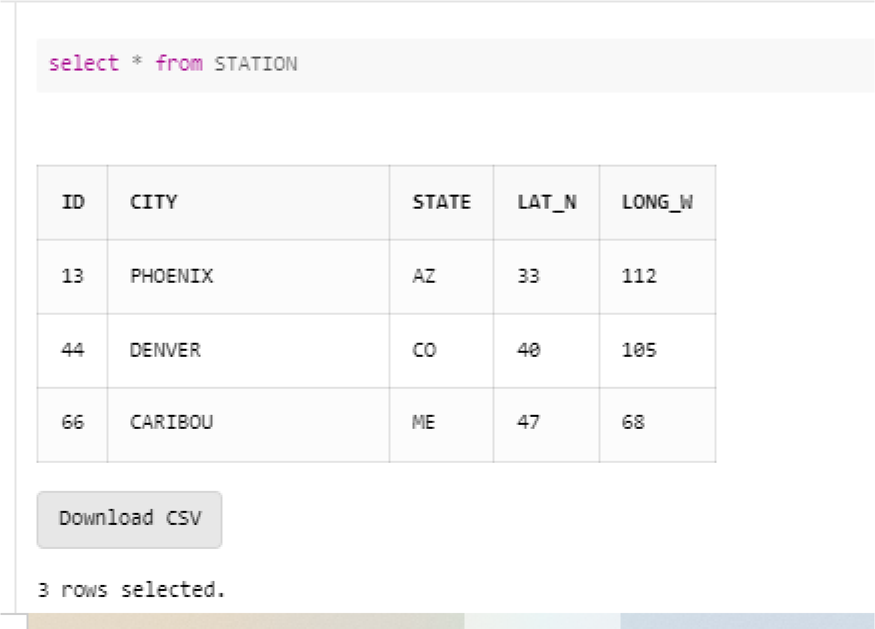
2. Insert the following records into the table:

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

```
insert into STATION VALUES (13,'PHOENIX','AZ',33,112)

insert into STATION VALUES (44,'DENVER','CO',40,105)

insert into STATION VALUES (66,'CARIBOU','ME',47,68)

select * from STATION
```

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

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

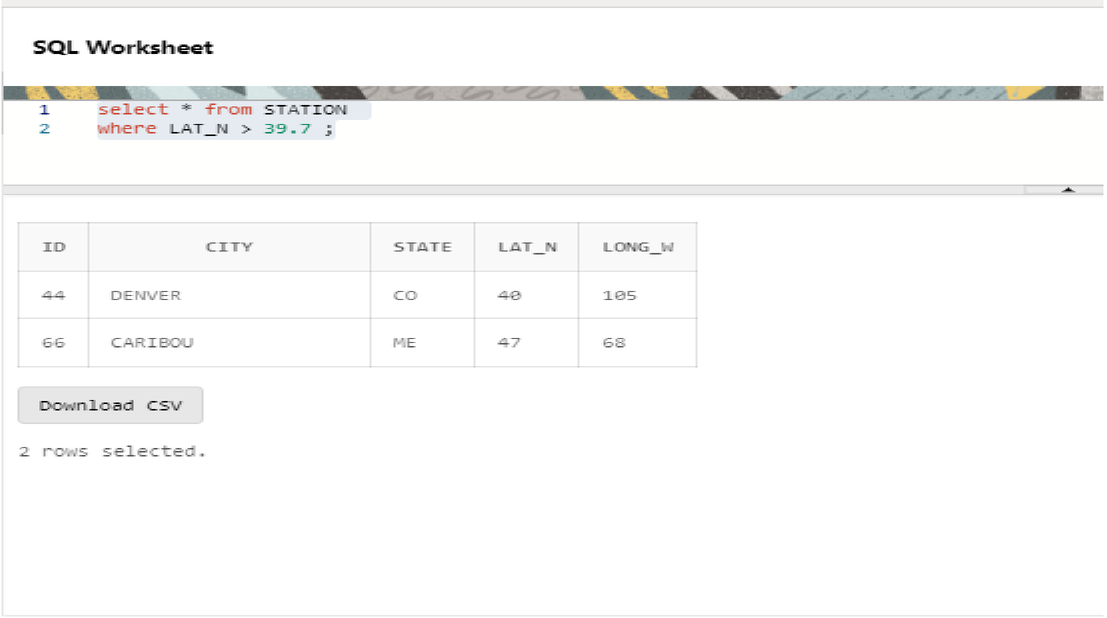


```
select * from STATION
```

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

```
3 rows selected.
```

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



```
select * from STATION
where LAT_N > 39.7 ;
```

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

```
2 rows selected.
```

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

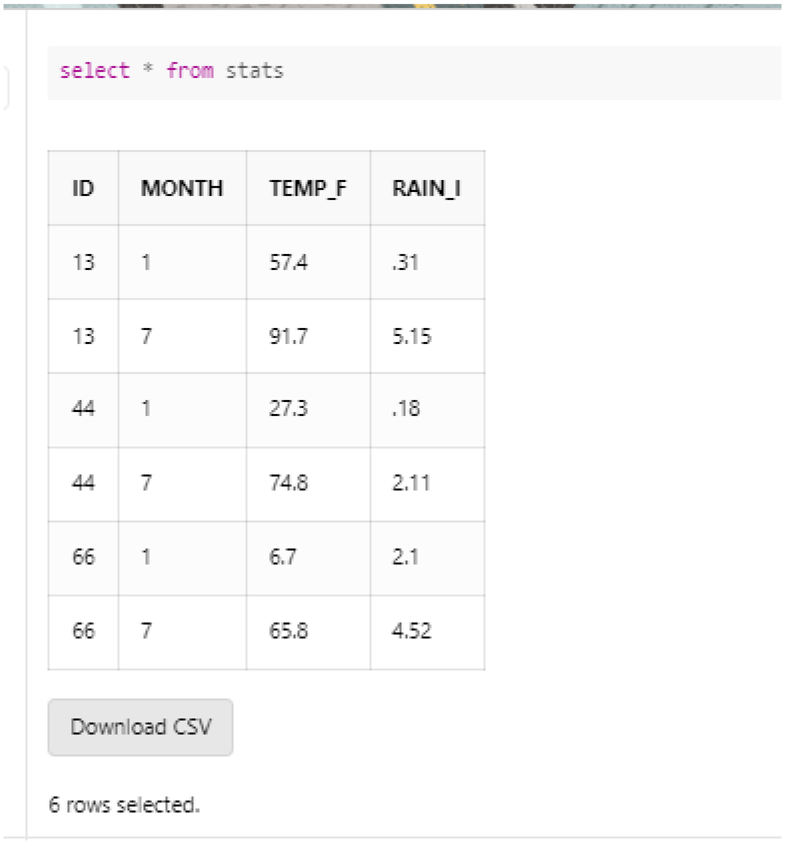
Column	Data type	Remark
ID	Number	must match some STATION table ID(so name & location will be known).
MONTH	Number	Range between 1 and 12
TEMP_F	Number	in Fahrenheit degrees,Range between -80 and 150
RAIN_I	Number	in inches, Range between 0 and 100

There will be no Duplicate ID and MONTH combination.

```
create table STATS
(    ID number,
    MONTH Number,
    TEMP_F Number,
    RAIN_I Number
 )
```

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

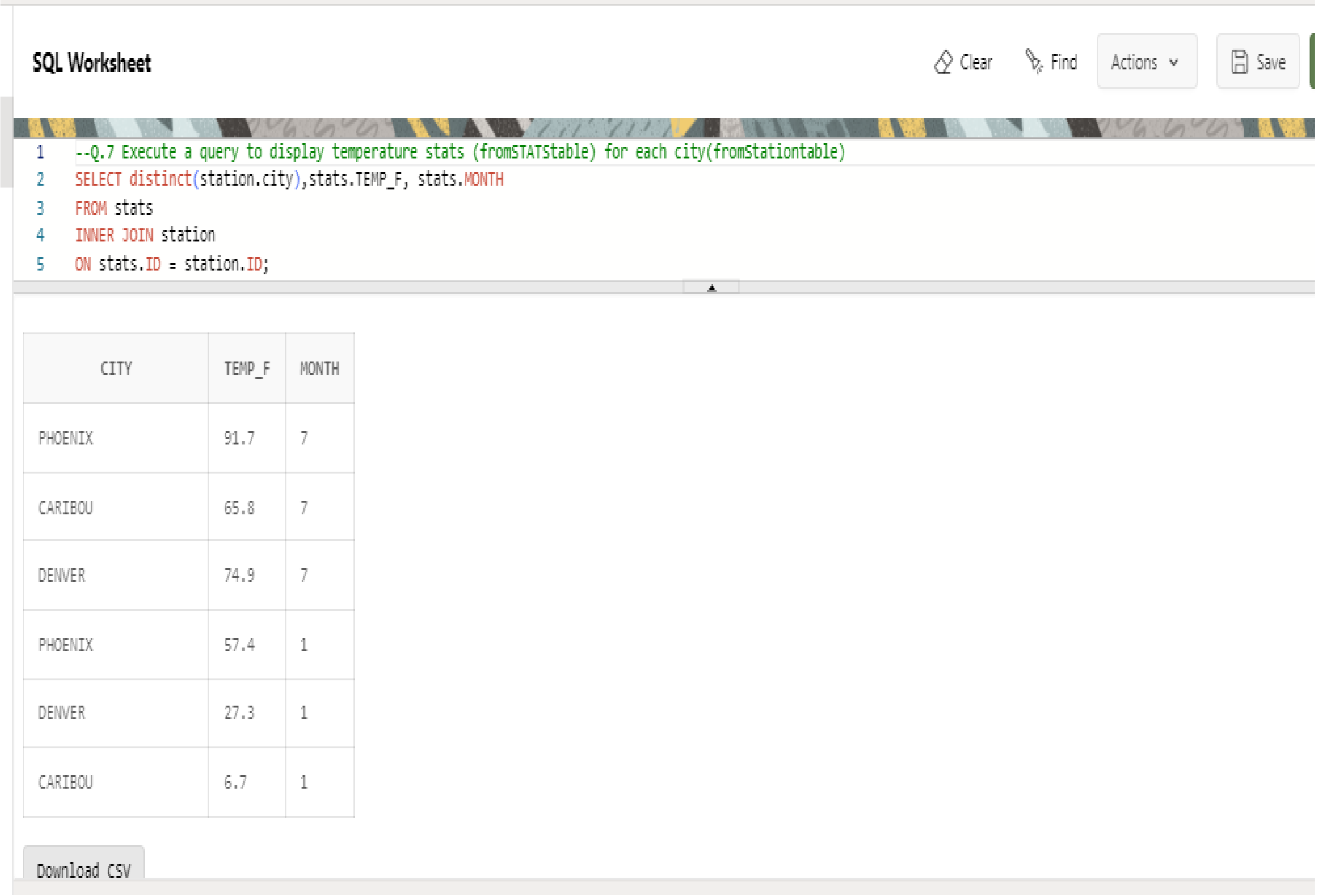


```
select * from stats
```

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

6 rows selected.

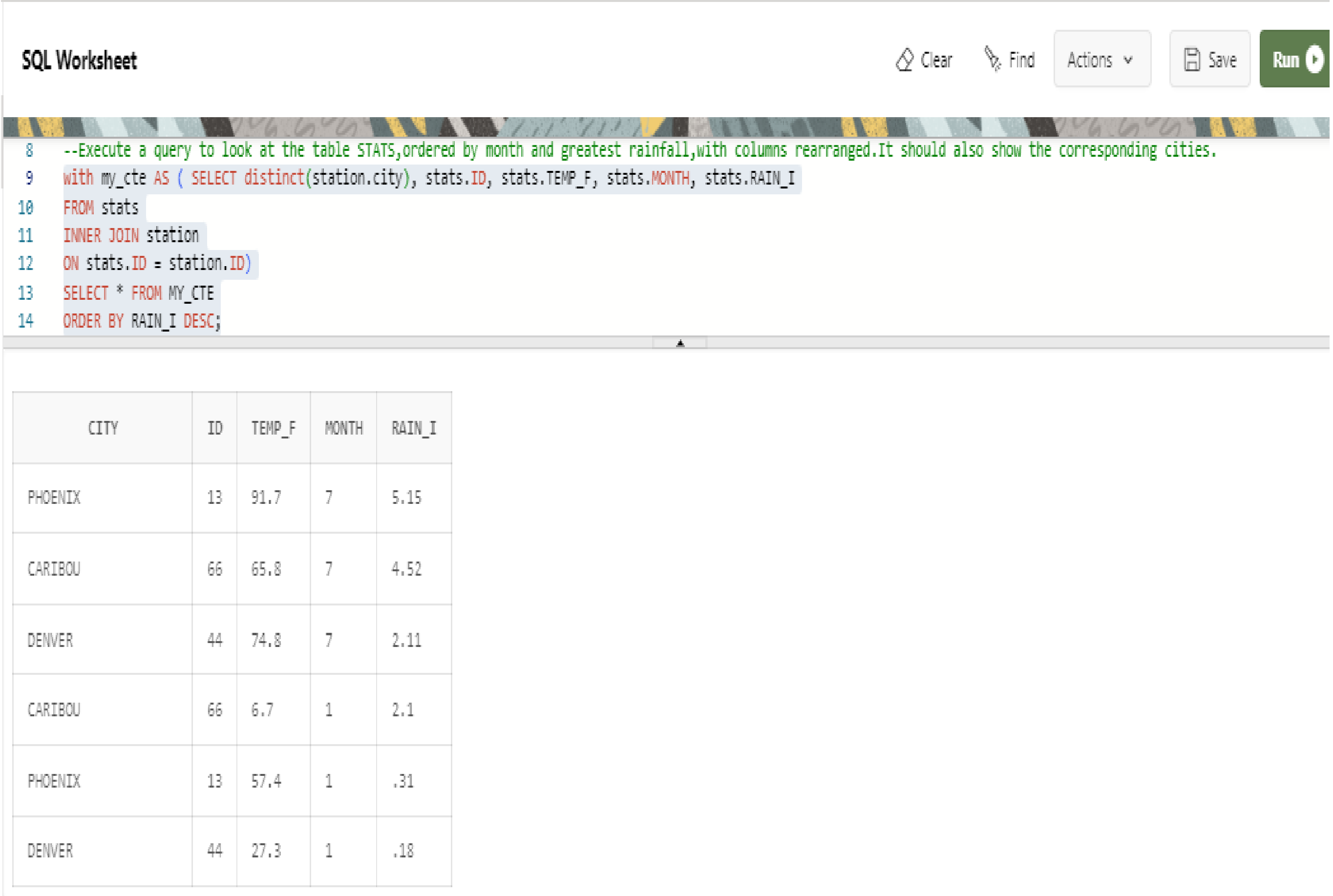
q.7



```
--Q.7 Execute a query to display temperature stats (fromSTATStable) for each city(fromStationtable)
SELECT distinct(station.city),stats.TEMP_F, stats.MONTH
FROM stats
INNER JOIN station
ON stats.ID = station.ID;
```

CITY	TEMP_F	MONTH
PHOENIX	91.7	7
CARIBOU	65.8	7
DENVER	74.9	7
PHOENIX	57.4	1
DENVER	27.3	1
CARIBOU	6.7	1

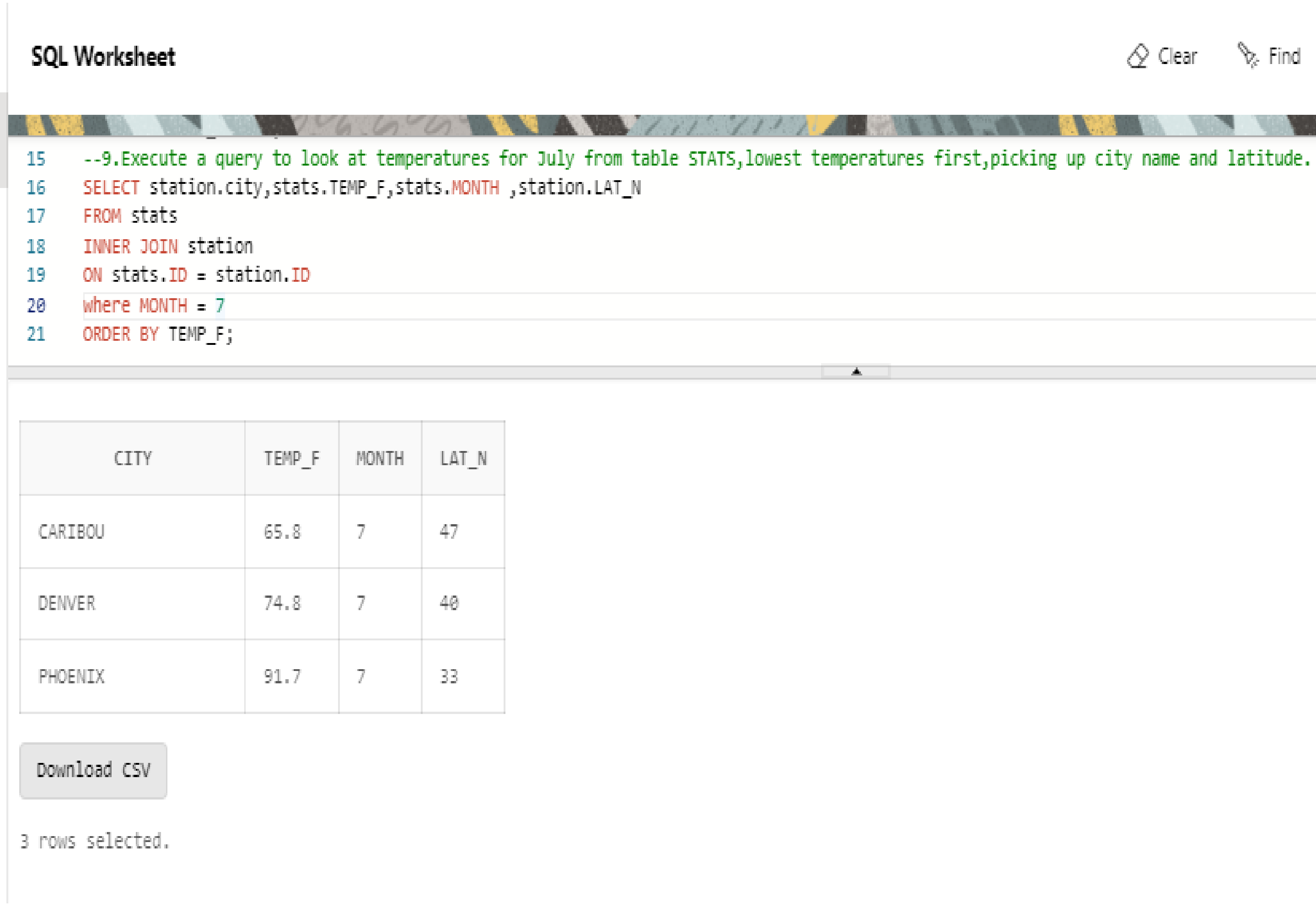
Q.8



```sql
--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.
with my_cte AS ( SELECT distinct(station.city), stats.ID, stats.TEMP_F, stats.MONTH, stats.RAIN_I
FROM stats
INNER JOIN station
ON stats.ID = station.ID)
SELECT * FROM MY_CTE
ORDER BY RAIN_I DESC;
```

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

Q.9

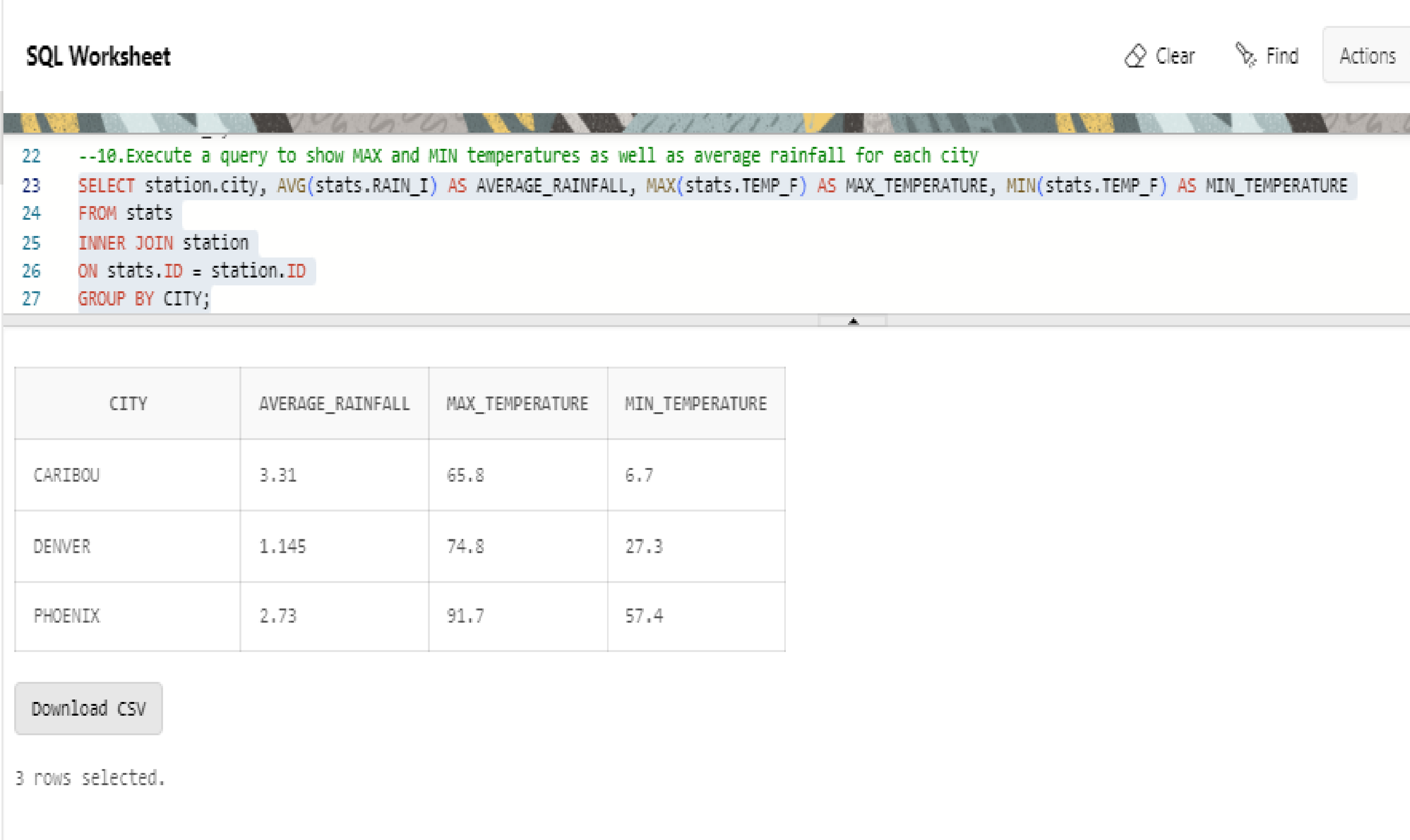


```sql
--9.Execute a query to look at temperatures for July from table STATS,lowest temperatures first,picking up city name and latitude.
SELECT station.city,stats.TEMP_F,stats.MONTH ,station.LAT_N
FROM stats
INNER JOIN station
ON stats.ID = station.ID
where MONTH = 7
ORDER BY TEMP_F;
```

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

```
3 rows selected.
```

Q.10



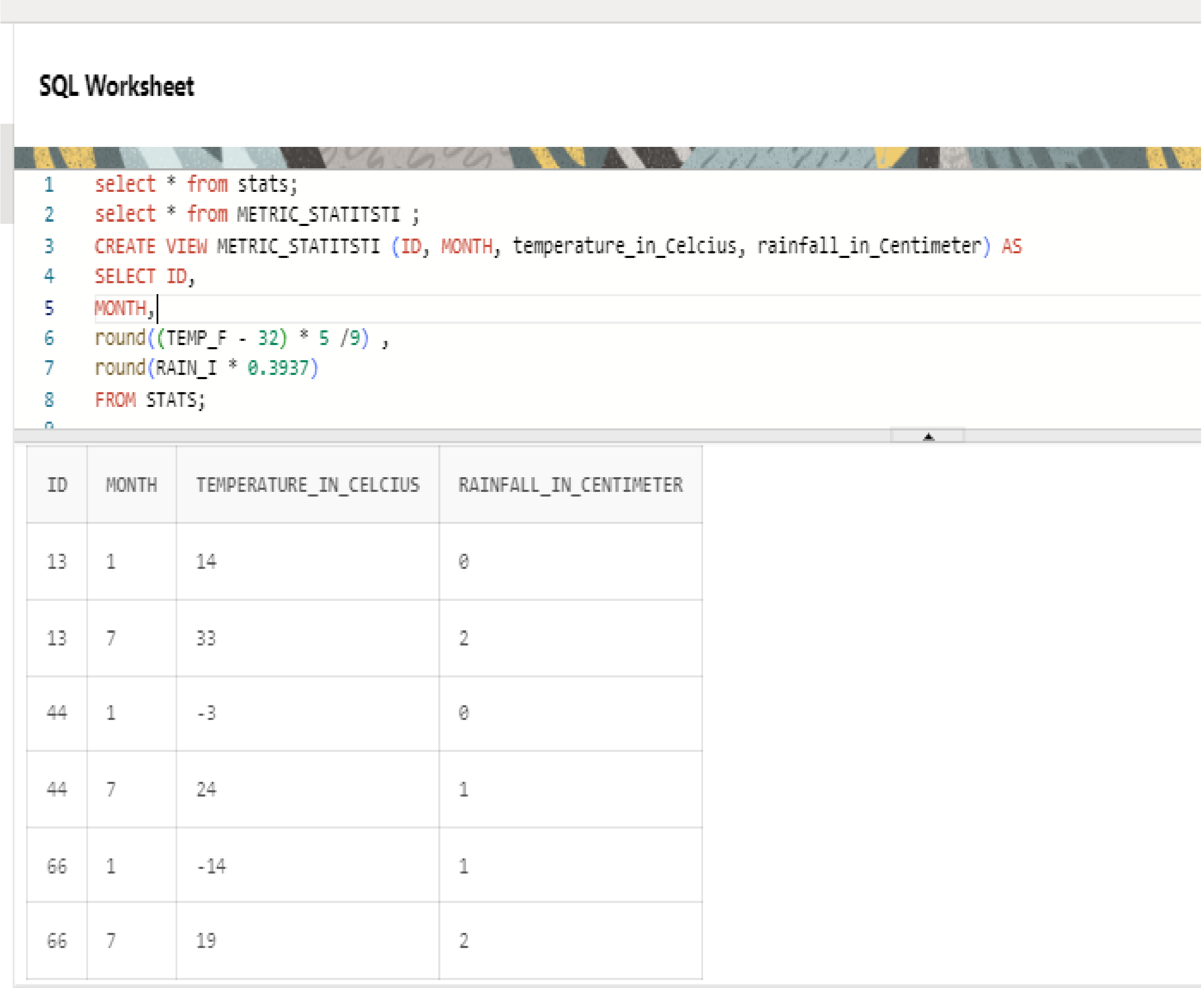
```
--10.Execute a query to show MAX and MIN temperatures as well as average rainfall for each city
SELECT station.city, AVG(stats.RAIN_I) AS AVERAGE_RAINFALL, MAX(stats.TEMP_F) AS MAX_TEMPERATURE, MIN(stats.TEMP_F) AS MIN_TEMPERATURE
FROM stats
INNER JOIN station
ON stats.ID = station.ID
GROUP BY CITY;
```

CITY	AVERAGE_RAINFALL	MAX_TEMPERATURE	MIN_TEMPERATURE
CARIBOU	3.31	65.8	6.7
DENVER	1.145	74.8	27.3
PHOENIX	2.73	91.7	57.4

3 rows selected.

Q.

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

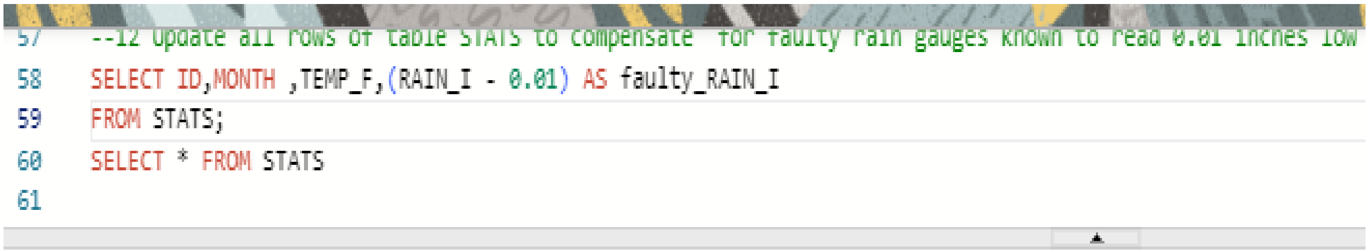


SQL Worksheet

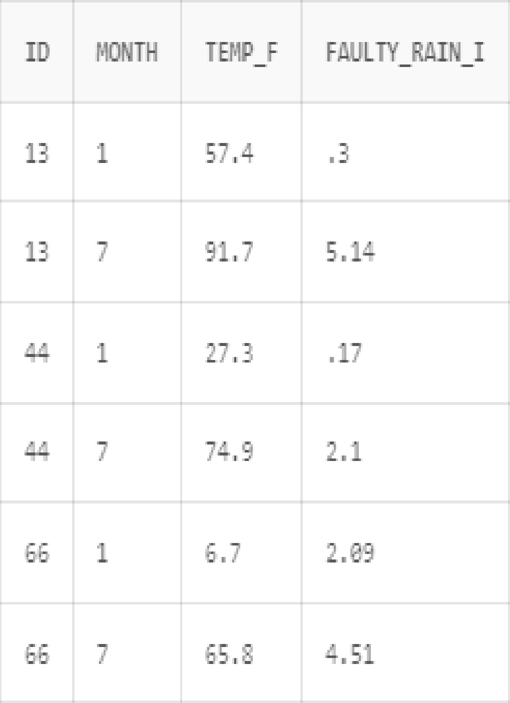
```
select * from stats;
select * from METRIC_STATITSTI ;
CREATE VIEW METRIC_STATITSTI (ID, MONTH, temperature_in_Celcius, rainfall_in_Centimeter) AS
SELECT ID,
MONTH,
round((TEMP_F - 32) * 5 /9) ,
round(RAIN_I * 0.3937)
FROM STATS;
```

ID	MONTH	TEMPERATURE_IN_CELCIUS	RAINFALL_IN_CENTIMETER
13	1	14	0
13	7	33	2
44	1	-3	0
44	7	24	1
66	1	-14	1
66	7	19	2

Q.12

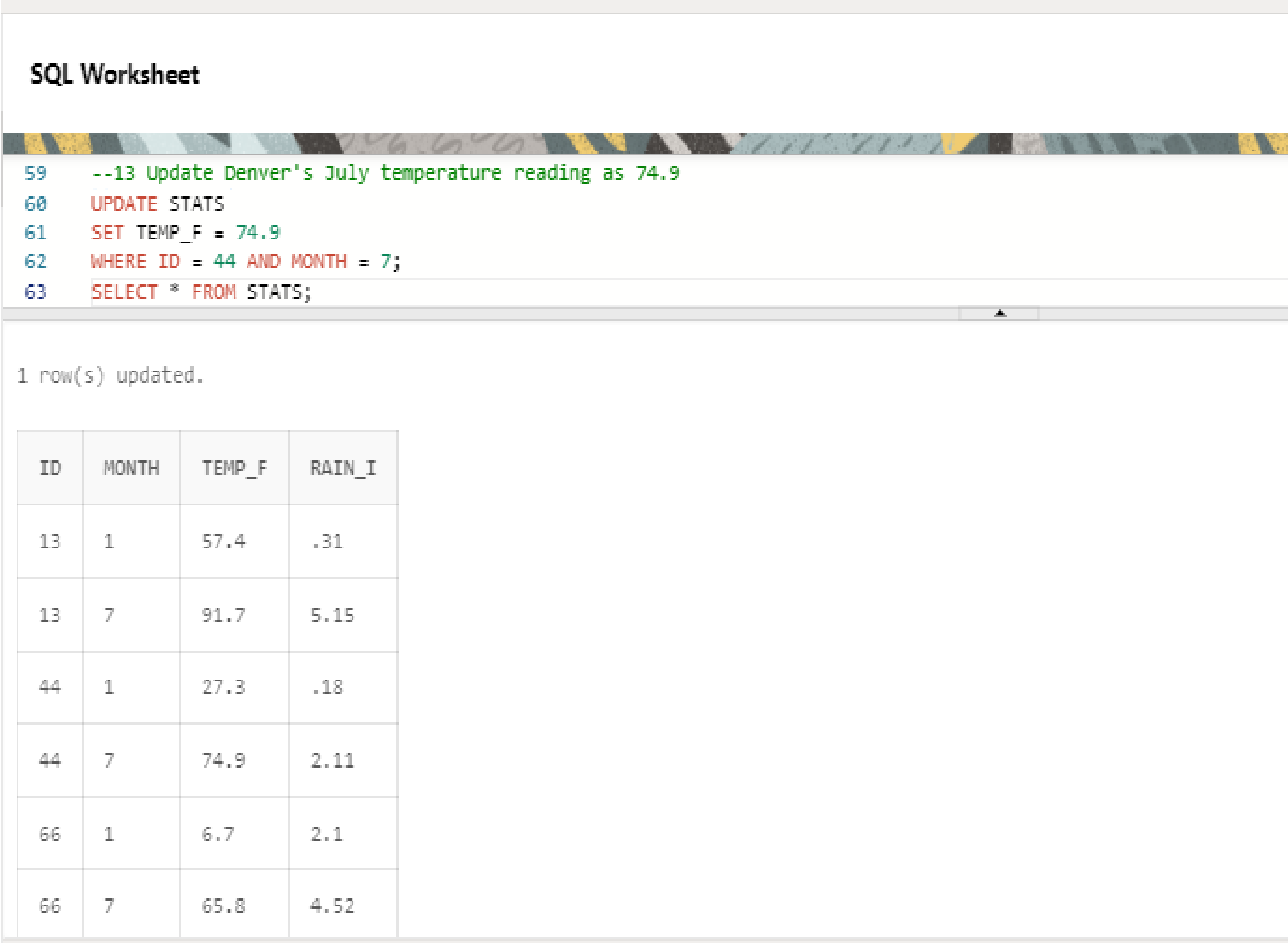


```
--12 update all rows of table STATS to compensate  for faulty rain gauges known to read 0.01 inches low
SELECT ID,MONTH ,TEMP_F,(RAIN_I - 0.01) AS faulty_RAIN_I
FROM STATS;
SELECT * FROM STATS

```



ID	MONTH	TEMP_F	FAULTY_RAIN_I
13	1	57.4	.3
13	7	91.7	5.14
44	1	27.3	.17
44	7	74.9	2.1
66	1	6.7	2.09
66	7	65.8	4.51

Q.13



```
--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;
```

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

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	.31
13	7	91.7	5.15
44	1	27.3	.18
44	7	74.9	2.11
66	1	6.7	2.1
66	7	65.8	4.52