Name = Sakshi Suhas Shinde

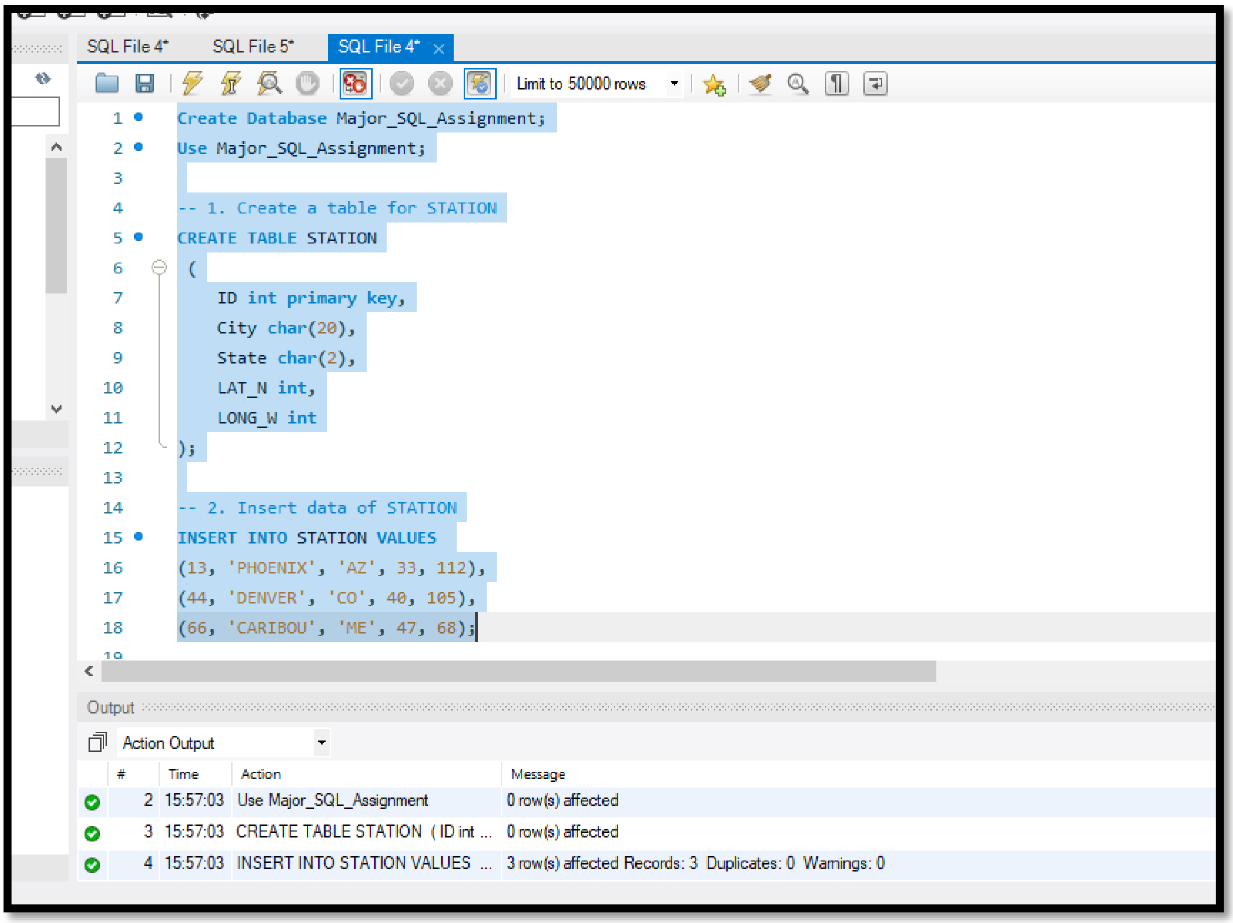
Email = sakshishinde209@gmail.com

Course = Business Analyst

Major SQL Assignment

Q1) Create a table “STATION ”

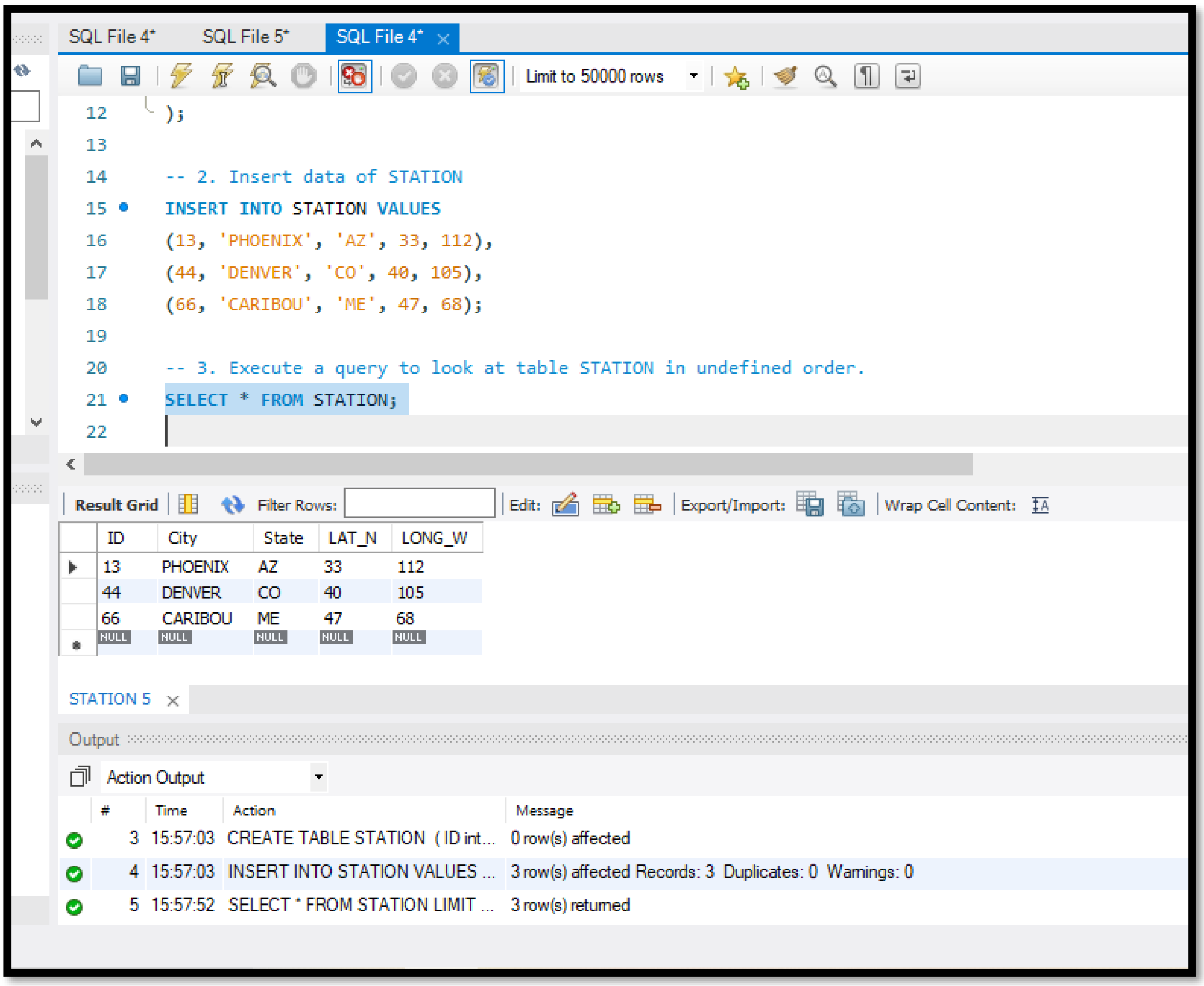
Q2) Insert the following records into the table:



```sql
Create Database Major_SQL_Assignment;
Use Major_SQL_Assignment;

-- 1. Create a table for STATION
CREATE TABLE STATION
 (
    ID int primary key,
    City char(20),
    State char(2),
    LAT_N int,
    LONG_W int
);

-- 2. Insert data of STATION
INSERT INTO STATION VALUES
(13, 'PHOENIX', 'AZ', 33, 112),
(44, 'DENVER', 'CO', 40, 105),
(66, 'CARIBOU', 'ME', 47, 68);
```

```
#  Time      Action                             Message
2  15:57:03  Use Major_SQL_Assignment           0 row(s) affected
3  15:57:03  CREATE TABLE STATION  ( ID int ... 0 row(s) affected
4  15:57:03  INSERT INTO STATION VALUES  ...    3 row(s) affected Records: 3  Duplicates: 0  Warnings: 0
```

Q3) Execute a query to look at table STATION in undefined order.

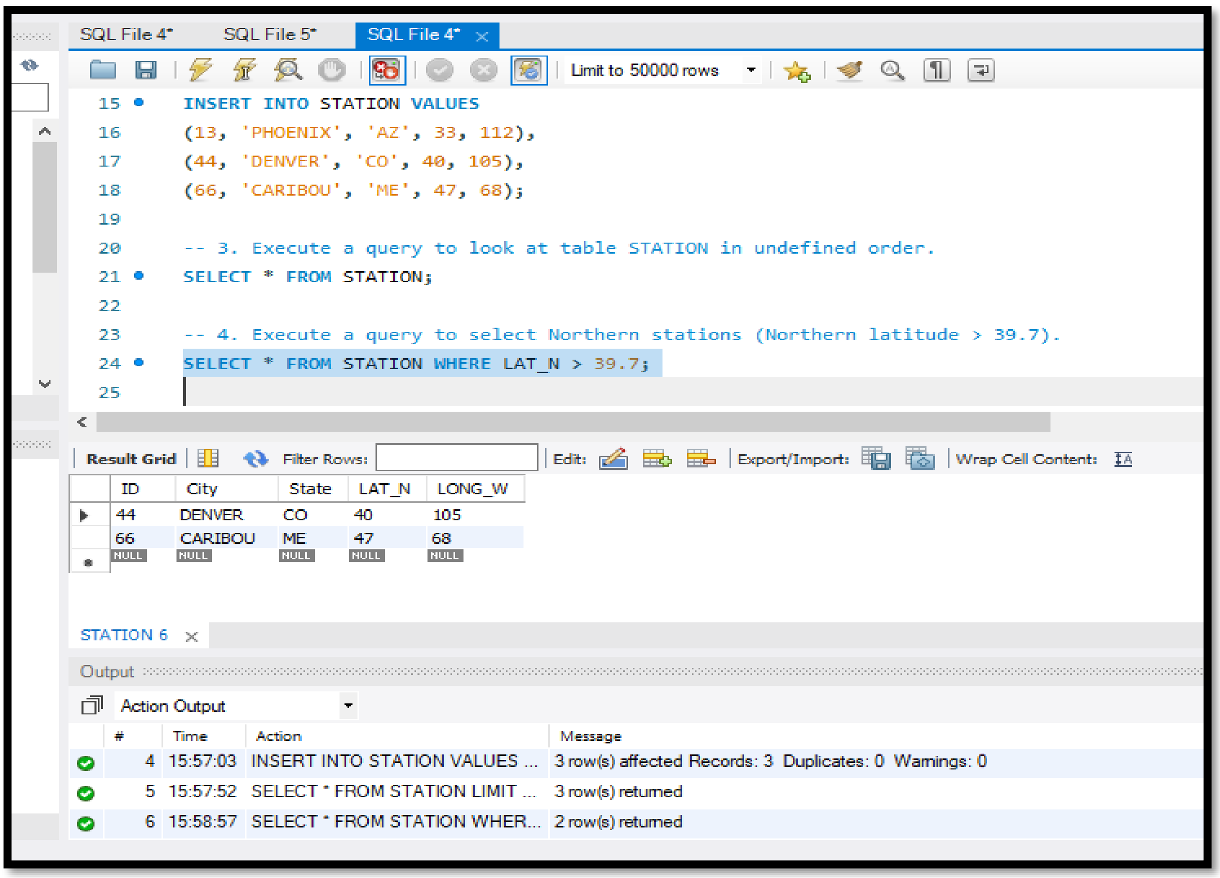


```
);

-- 2. Insert data of STATION
INSERT INTO STATION VALUES
(13, 'PHOENIX', 'AZ', 33, 112),
(44, 'DENVER', 'CO', 40, 105),
(66, 'CARIBOU', 'ME', 47, 68);

-- 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
NULL	NULL	NULL	NULL	NULL

#	Time	Action	Message
3	15:57:03	CREATE TABLE STATION (ID int...	0 row(s) affected
4	15:57:03	INSERT INTO STATION VALUES ...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0
5	15:57:52	SELECT * FROM STATION LIMIT ...	3 row(s) returned

Q4) Execute a query to select Northern stations (Northern latitude > 39.7).

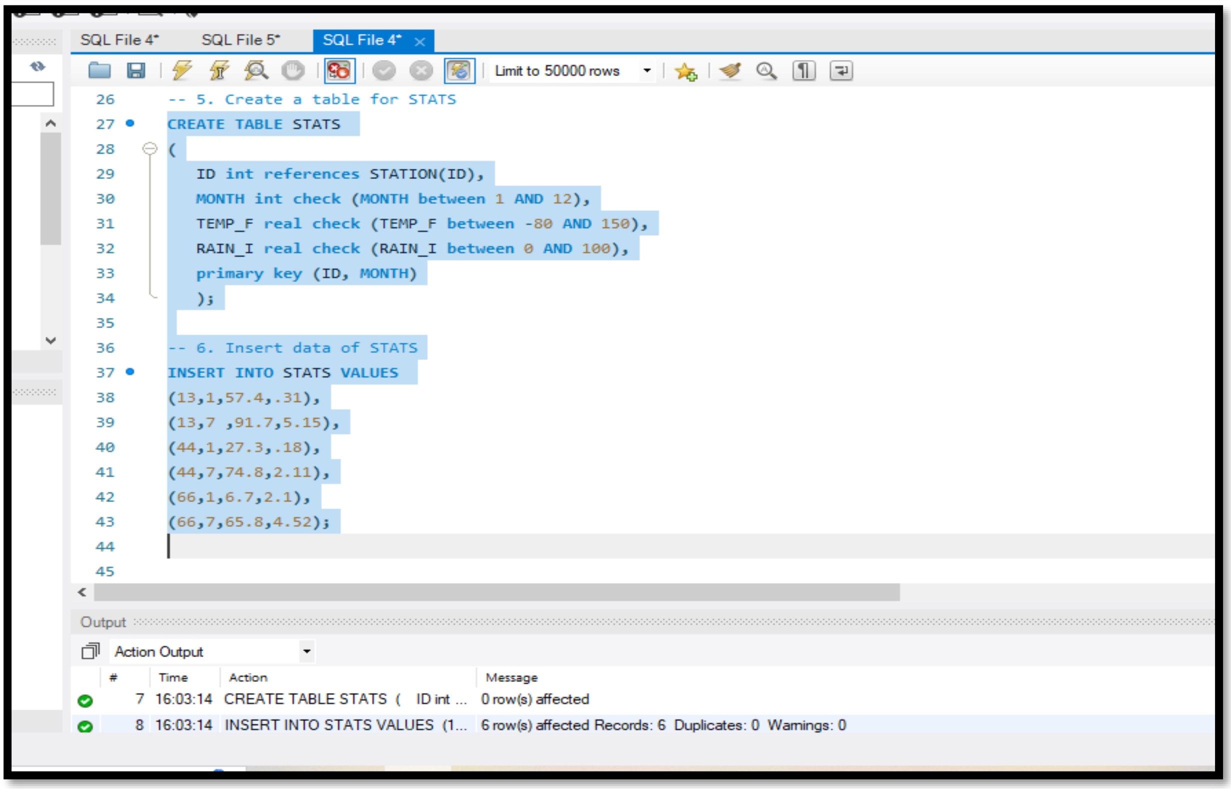


```
INSERT INTO STATION VALUES
(13, 'PHOENIX', 'AZ', 33, 112),
(44, 'DENVER', 'CO', 40, 105),
(66, 'CARIBOU', 'ME', 47, 68);

-- 3. Execute a query to look at table STATION in undefined order.
SELECT * FROM STATION;

-- 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
NULL	NULL	NULL	NULL	NULL

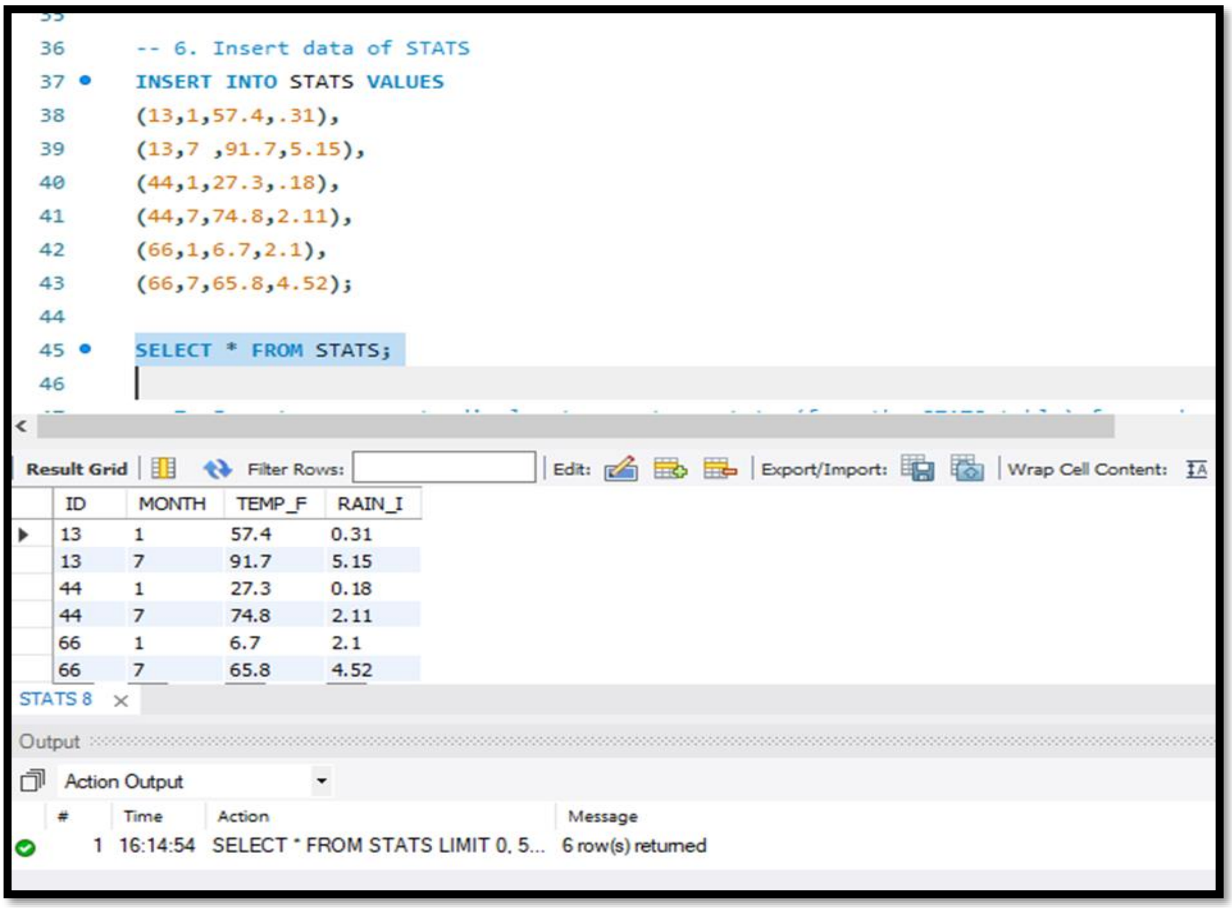
#	Time	Action	Message
4	15:57:03	INSERT INTO STATION VALUES ...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0
5	15:57:52	SELECT * FROM STATION LIMIT ...	3 row(s) returned
6	15:58:57	SELECT * FROM STATION WHER...	2 row(s) returned

Q5) Create another table, 'STATS'   Q6) Populate the table



```
-- 5. Create a table for STATS
CREATE TABLE STATS
(
   ID int references STATION(ID),
   MONTH int 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)
   );

-- 6. Insert data of STATS
INSERT INTO STATS VALUES
(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);
```

```
Output
Action Output
#  Time      Action                        Message
7  16:03:14  CREATE TABLE STATS (   ID int ...  0 row(s) affected
8  16:03:14  INSERT INTO STATS VALUES  (1...  6 row(s) affected Records: 6  Duplicates: 0  Warnings: 0
```

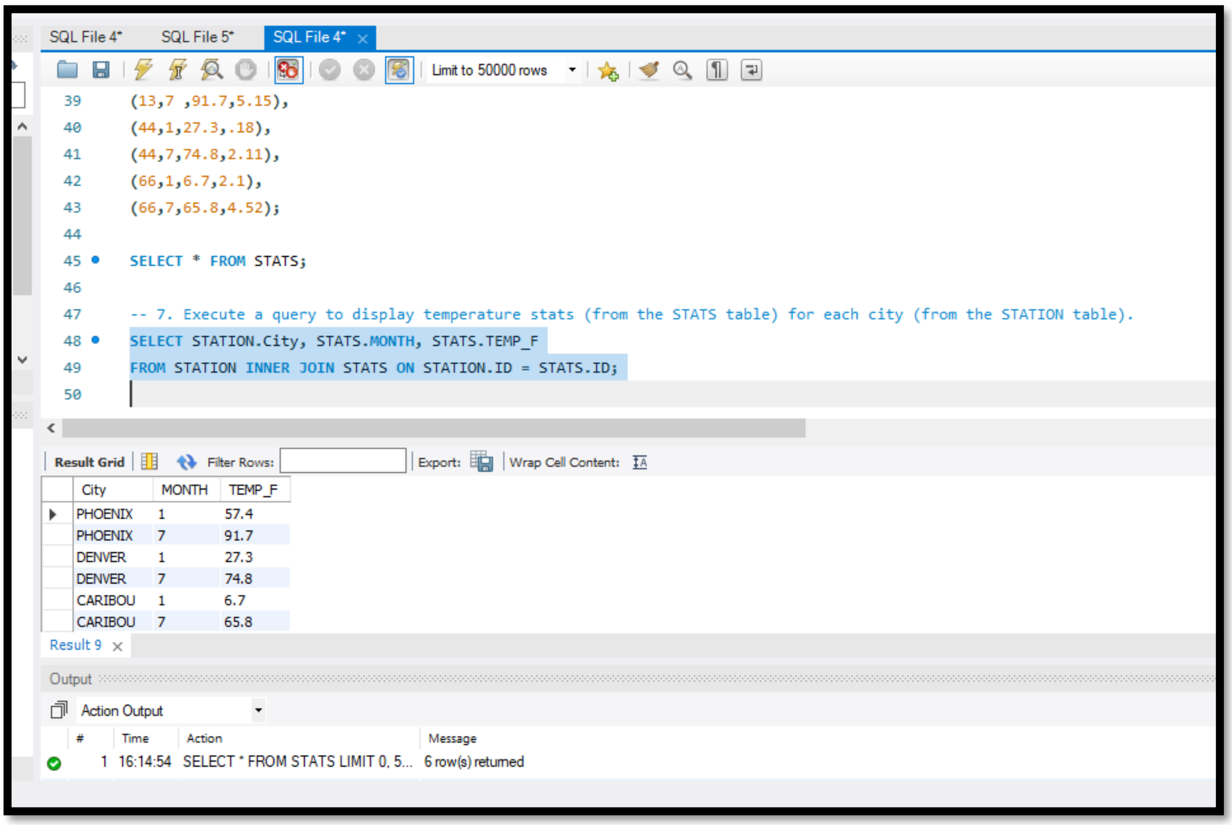


```
-- 6. Insert data of STATS
INSERT INTO STATS VALUES
(13,1,57.4,.31),
(13,7 ,91.7,5.15),
(44,1,27.3,.18),
(44,7,74.8,2.11),
(66,1,6.7,2.1),
(66,7,65.8,4.52);

SELECT * FROM STATS;
```

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	0.31
13	7	91.7	5.15
44	1	27.3	0.18
44	7	74.8	2.11
66	1	6.7	2.1
66	7	65.8	4.52

```
Output
Action Output
#  Time      Action                           Message
1  16:14:54  SELECT * FROM STATS LIMIT 0, 5...  6 row(s) returned
```

Q7) Execute a query to display temperature stats (from the STATS table) for each city (from the STATION table).



SQL File 4* SQL File 5* SQL File 4*

Limit to 50000 rows

```
(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;

-- 7. Execute a query to display temperature stats (from the STATS table) for each city (from the STATION table).
SELECT STATION.City, STATS.MONTH, STATS.TEMP_F
FROM STATION INNER JOIN STATS ON STATION.ID = STATS.ID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

City	MONTH	TEMP_F
PHOENIX	1	57.4
PHOENIX	7	91.7
DENVER	1	27.3
DENVER	7	74.8
CARIBOU	1	6.7
CARIBOU	7	65.8

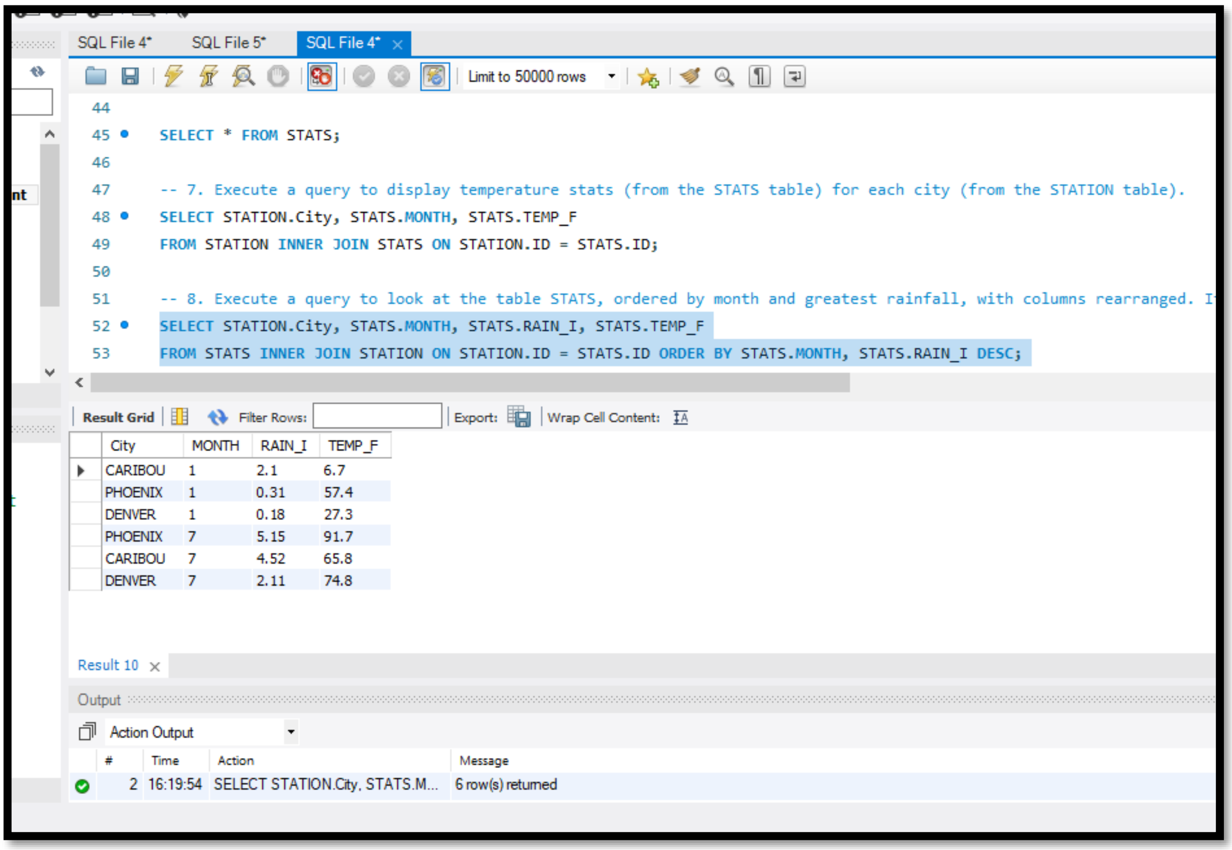
Result 9

Output

Action Output

#	Time	Action	Message
1	16:14:54	SELECT * FROM STATS LIMIT 0, 5...	6 row(s) returned

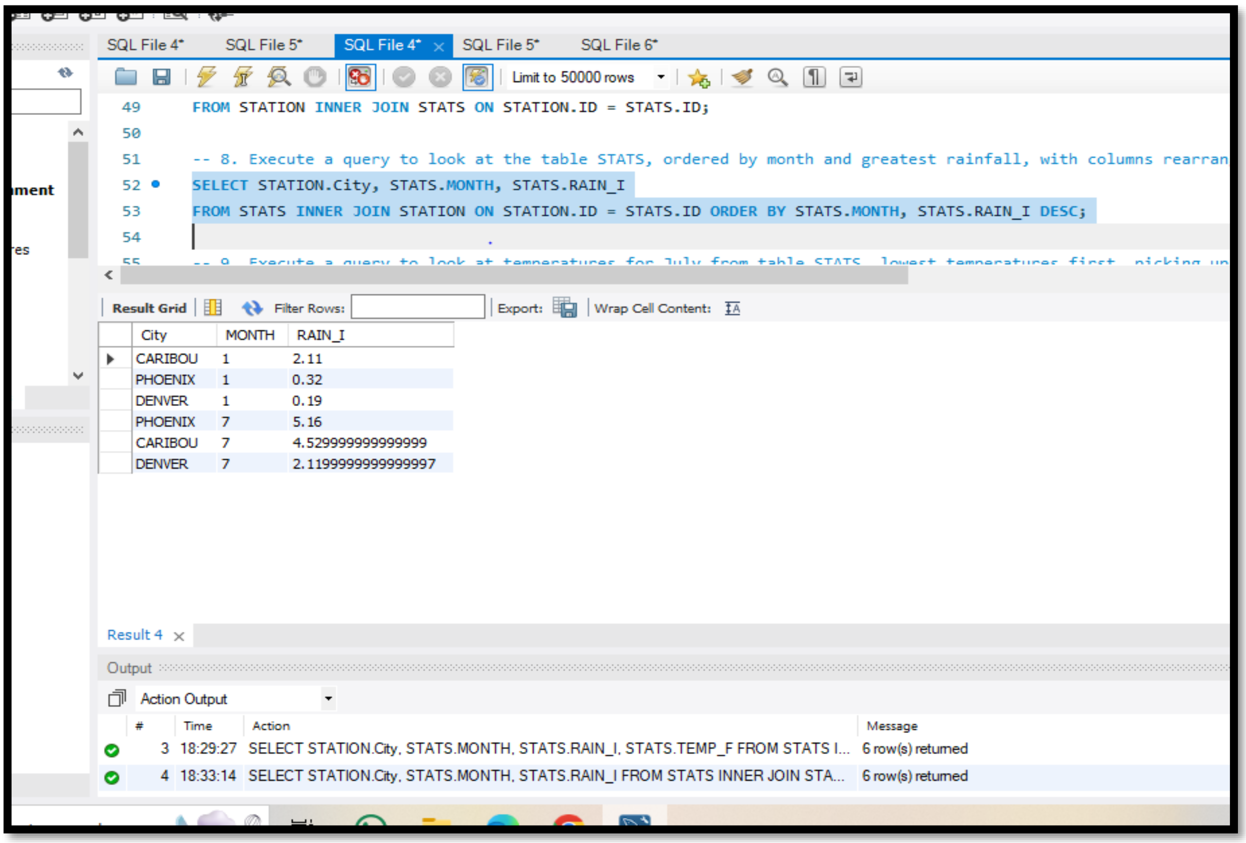
Q8) Execute a query to look at the table STATS, ordered by month and greatest rainfall, with columns rearranged. It should also show the corresponding cities.



```
SELECT * FROM STATS;

-- 7. Execute a query to display temperature stats (from the STATS table) for each city (from the STATION table).
SELECT STATION.City, STATS.MONTH, STATS.TEMP_F
FROM STATION INNER JOIN STATS ON STATION.ID = STATS.ID;

-- 8. Execute a query to look at the table STATS, ordered by month and greatest rainfall, with columns rearranged. I
SELECT STATION.City, STATS.MONTH, STATS.RAIN_I, STATS.TEMP_F
FROM STATS INNER JOIN STATION ON STATION.ID = STATS.ID ORDER BY STATS.MONTH, STATS.RAIN_I DESC;
```

City	MONTH	RAIN_I	TEMP_F
CARIBOU	1	2.1	6.7
PHOENIX	1	0.31	57.4
DENVER	1	0.18	27.3
PHOENIX	7	5.15	91.7
CARIBOU	7	4.52	65.8
DENVER	7	2.11	74.8

#	Time	Action	Message
2	16:19:54	SELECT STATION.City, STATS.M...	6 row(s) returned

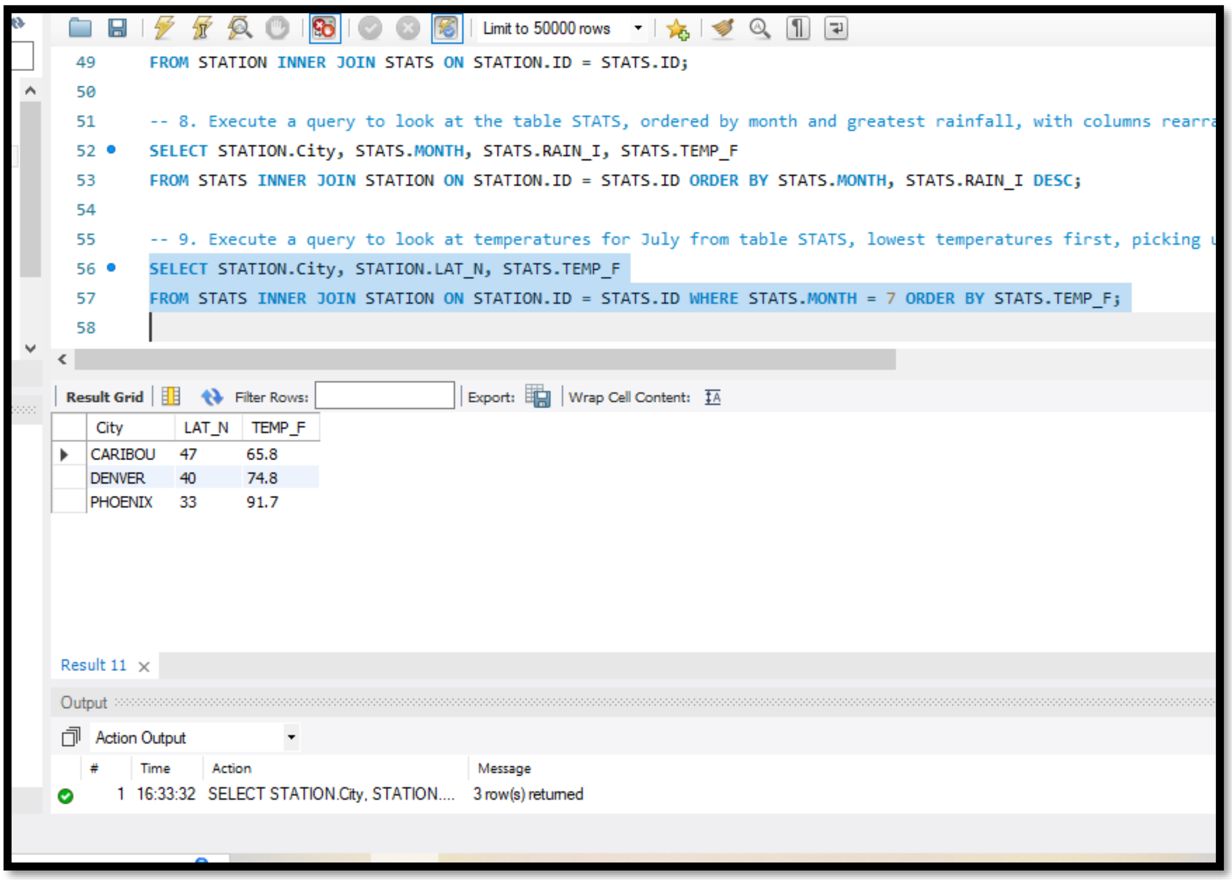


```
FROM STATION INNER JOIN STATS ON STATION.ID = STATS.ID;

-- 8. Execute a query to look at the table STATS, ordered by month and greatest rainfall, with columns rearran
SELECT STATION.City, STATS.MONTH, STATS.RAIN_I
FROM STATS INNER JOIN STATION ON STATION.ID = STATS.ID ORDER BY STATS.MONTH, STATS.RAIN_I DESC;
```

City	MONTH	RAIN_I
CARIBOU	1	2.11
PHOENIX	1	0.32
DENVER	1	0.19
PHOENIX	7	5.16
CARIBOU	7	4.529999999999999
DENVER	7	2.1199999999999997

#	Time	Action	Message
3	18:29:27	SELECT STATION.City, STATS.MONTH, STATS.RAIN_I, STATS.TEMP_F FROM STATS I...	6 row(s) returned
4	18:33:14	SELECT STATION.City, STATS.MONTH, STATS.RAIN_I FROM STATS INNER JOIN STA...	6 row(s) returned

Q9) Execute a query to look at temperatures for July from table STATS, lowest temperatures first, picking up city name and latitude.

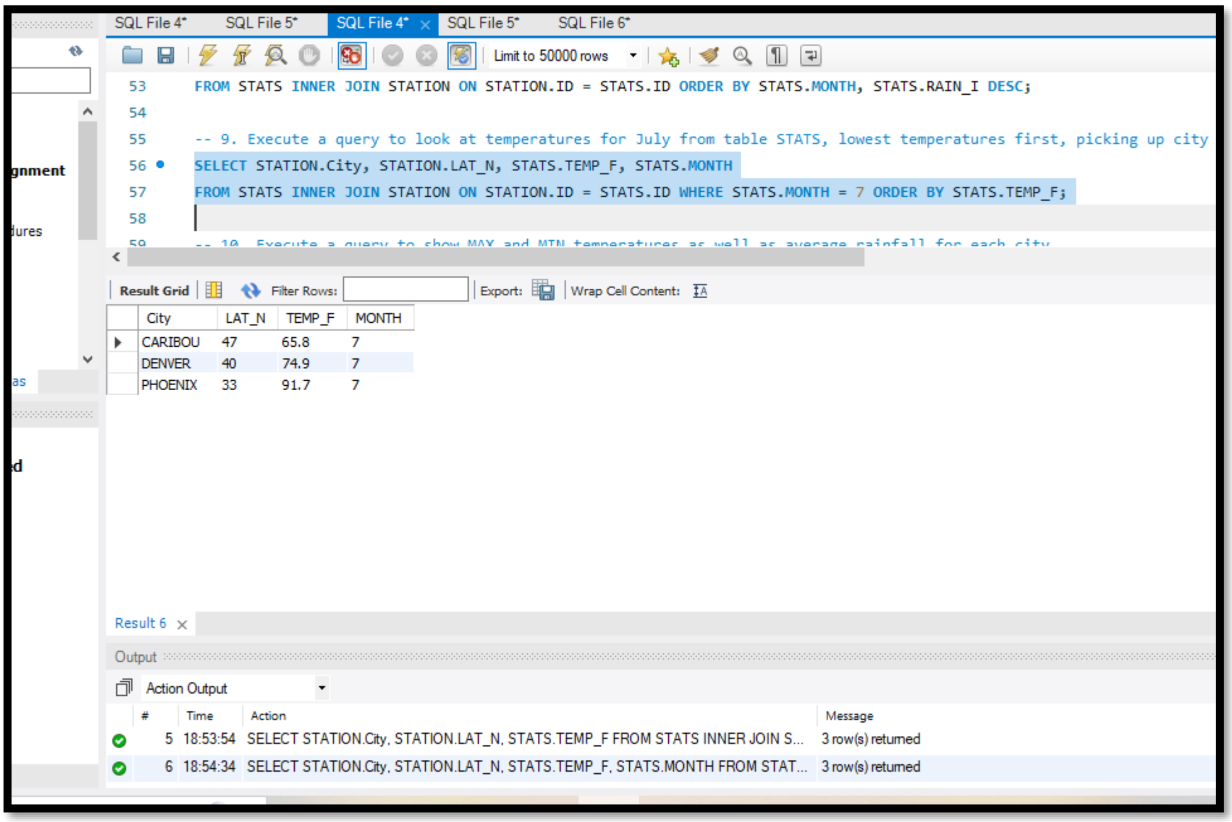


```sql
FROM STATION INNER JOIN STATS ON STATION.ID = STATS.ID;

-- 8. Execute a query to look at the table STATS, ordered by month and greatest rainfall, with columns rearra
SELECT STATION.City, STATS.MONTH, STATS.RAIN_I, STATS.TEMP_F
FROM STATS INNER JOIN STATION ON STATION.ID = STATS.ID ORDER BY STATS.MONTH, STATS.RAIN_I DESC;

-- 9. Execute a query to look at temperatures for July from table STATS, lowest temperatures first, picking u
SELECT STATION.City, STATION.LAT_N, STATS.TEMP_F
FROM STATS INNER JOIN STATION ON STATION.ID = STATS.ID WHERE STATS.MONTH = 7 ORDER BY STATS.TEMP_F;

```

City	LAT_N	TEMP_F
CARIBOU	47	65.8
DENVER	40	74.8
PHOENIX	33	91.7

Result 11

```
#   Time      Action                          Message
1   16:33:32  SELECT STATION.City, STATION....  3 row(s) returned
```



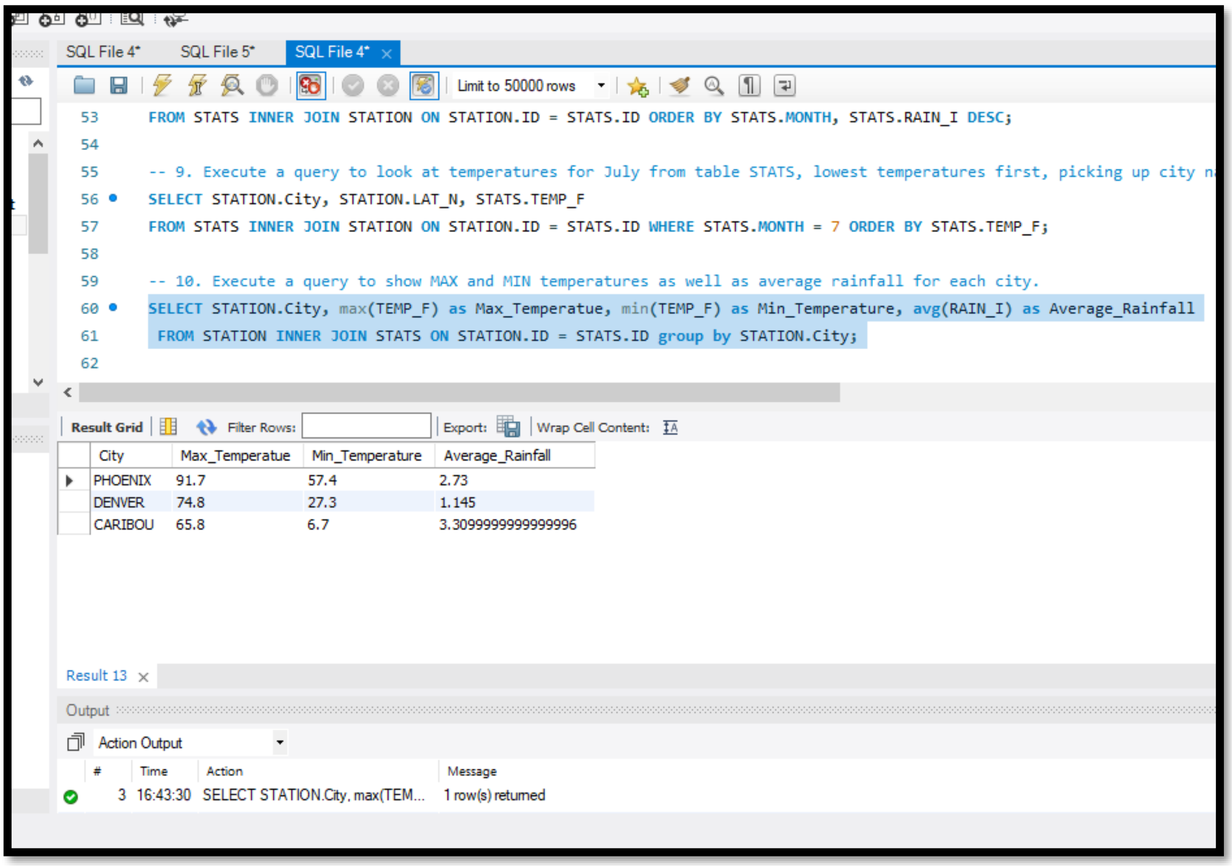
```sql
FROM STATS INNER JOIN STATION ON STATION.ID = STATS.ID ORDER BY STATS.MONTH, STATS.RAIN_I DESC;

-- 9. Execute a query to look at temperatures for July from table STATS, lowest temperatures first, picking up city
SELECT STATION.City, STATION.LAT_N, STATS.TEMP_F, STATS.MONTH
FROM STATS INNER JOIN STATION ON STATION.ID = STATS.ID WHERE STATS.MONTH = 7 ORDER BY STATS.TEMP_F;

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

City	LAT_N	TEMP_F	MONTH
CARIBOU	47	65.8	7
DENVER	40	74.9	7
PHOENIX	33	91.7	7

Result 6

```
#   Time      Action                                                                   Message
5   18:53:54  SELECT STATION.City, STATION.LAT_N, STATS.TEMP_F FROM STATS INNER JOIN S...  3 row(s) returned
6   18:54:34  SELECT STATION.City, STATION.LAT_N, STATS.TEMP_F, STATS.MONTH FROM STAT...   3 row(s) returned
```

Q10) Execute a query to show MAX and MIN temperatures as well as average rainfall for each city.



```sql
FROM STATS INNER JOIN STATION ON STATION.ID = STATS.ID ORDER BY STATS.MONTH, STATS.RAIN_I DESC;

-- 9. Execute a query to look at temperatures for July from table STATS, lowest temperatures first, picking up city n
SELECT STATION.City, STATION.LAT_N, STATS.TEMP_F
FROM STATS INNER JOIN STATION ON STATION.ID = STATS.ID WHERE STATS.MONTH = 7 ORDER BY STATS.TEMP_F;

-- 10. Execute a query to show MAX and MIN temperatures as well as average rainfall for each city.
SELECT STATION.City, max(TEMP_F) as Max_Temperatue, min(TEMP_F) as Min_Temperature, avg(RAIN_I) as Average_Rainfall
 FROM STATION INNER JOIN STATS ON STATION.ID = STATS.ID group by STATION.City;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

City	Max_Temperatue	Min_Temperature	Average_Rainfall
PHOENIX	91.7	57.4	2.73
DENVER	74.8	27.3	1.145
CARIBOU	65.8	6.7	3.3099999999999996

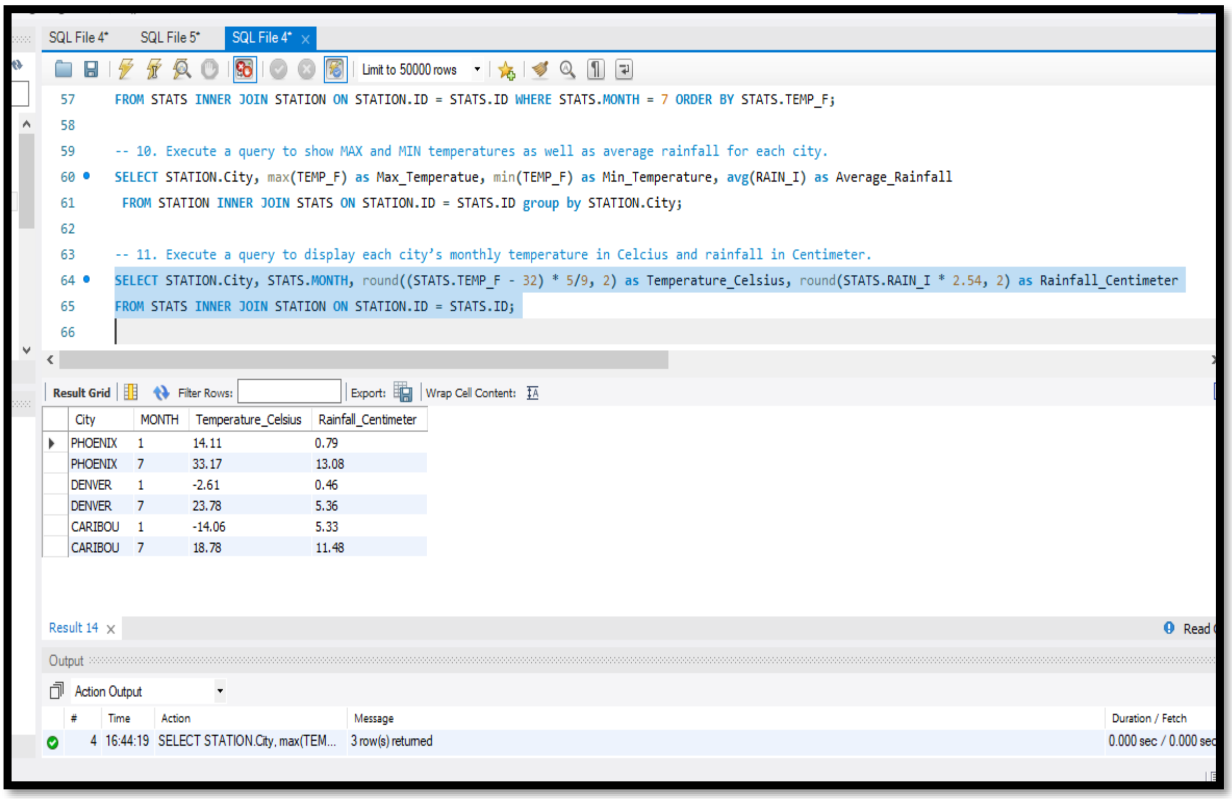
Result 13

Output

Action Output

#	Time	Action	Message
3	16:43:30	SELECT STATION.City, max(TEM...	1 row(s) returned

Q11) Execute a query to display each city's monthly temperature in Celsius and rainfall in Centimetre.



SQL File 4* SQL File 5* SQL File 4*

Limit to 50000 rows

```
FROM STATS INNER JOIN STATION ON STATION.ID = STATS.ID WHERE STATS.MONTH = 7 ORDER BY STATS.TEMP_F;

-- 10. Execute a query to show MAX and MIN temperatures as well as average rainfall for each city.
SELECT STATION.City, max(TEMP_F) as Max_Temperatue, min(TEMP_F) as Min_Temperature, avg(RAIN_I) as Average_Rainfall
 FROM STATION INNER JOIN STATS ON STATION.ID = STATS.ID group by STATION.City;

-- 11. Execute a query to display each city's monthly temperature in Celcius and rainfall in Centimeter.
SELECT STATION.City, STATS.MONTH, round((STATS.TEMP_F - 32) * 5/9, 2) as Temperature_Celsius, round(STATS.RAIN_I * 2.54, 2) as Rainfall_Centimeter
FROM STATS INNER JOIN STATION ON STATION.ID = STATS.ID;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

City	MONTH	Temperature_Celsius	Rainfall_Centimeter
PHOENIX	1	14.11	0.79
PHOENIX	7	33.17	13.08
DENVER	1	-2.61	0.46
DENVER	7	23.78	5.36
CARIBOU	1	-14.06	5.33
CARIBOU	7	18.78	11.48

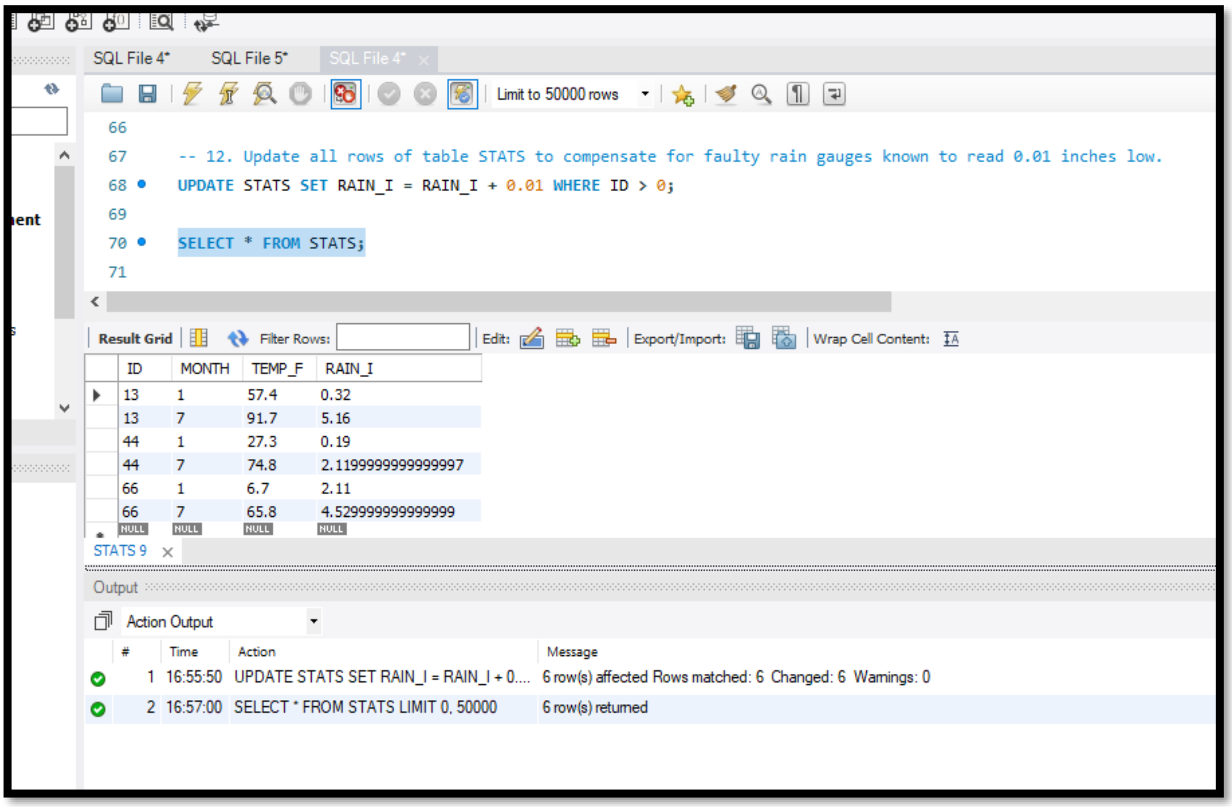
Result 14 Read

Output

Action Output

```
#  Time      Action                           Message             Duration / Fetch
4  16:44:19  SELECT STATION.City, max(TEM...  3 row(s) returned   0.000 sec / 0.000 sec
```

Q12) Update all rows of table STATS to compensate for faulty rain gauges known to read 0.01 inches low.

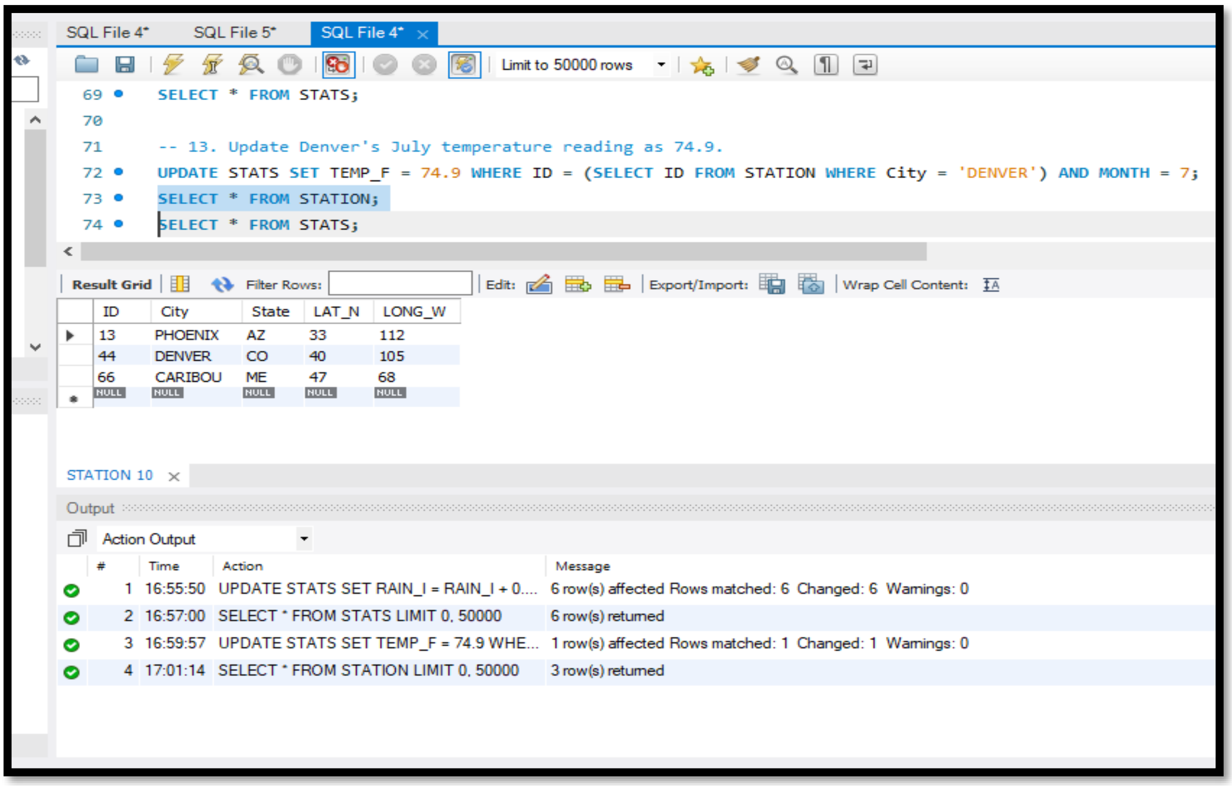


```
-- 12. Update all rows of table STATS to compensate for faulty rain gauges known to read 0.01 inches low.
UPDATE STATS SET RAIN_I = RAIN_I + 0.01 WHERE ID > 0;

SELECT * FROM STATS;
```

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	0.32
13	7	91.7	5.16
44	1	27.3	0.19
44	7	74.8	2.1199999999999997
66	1	6.7	2.11
66	7	65.8	4.529999999999999
NULL	NULL	NULL	NULL

```
#  Time      Action                                  Message
1  16:55:50  UPDATE STATS SET RAIN_I = RAIN_I + 0....  6 row(s) affected Rows matched: 6  Changed: 6  Warnings: 0
2  16:57:00  SELECT * FROM STATS LIMIT 0, 50000      6 row(s) returned
```

Q13) Update Denver's July temperature reading as 74.9



SQL File 4* | SQL File 5* | SQL File 4*

Limit to 50000 rows

```sql
SELECT * FROM STATS;

-- 13. Update Denver's July temperature reading as 74.9.
UPDATE STATS SET TEMP_F = 74.9 WHERE ID = (SELECT ID FROM STATION WHERE City = 'DENVER') AND MONTH = 7;
SELECT * FROM STATION;
SELECT * FROM STATS;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

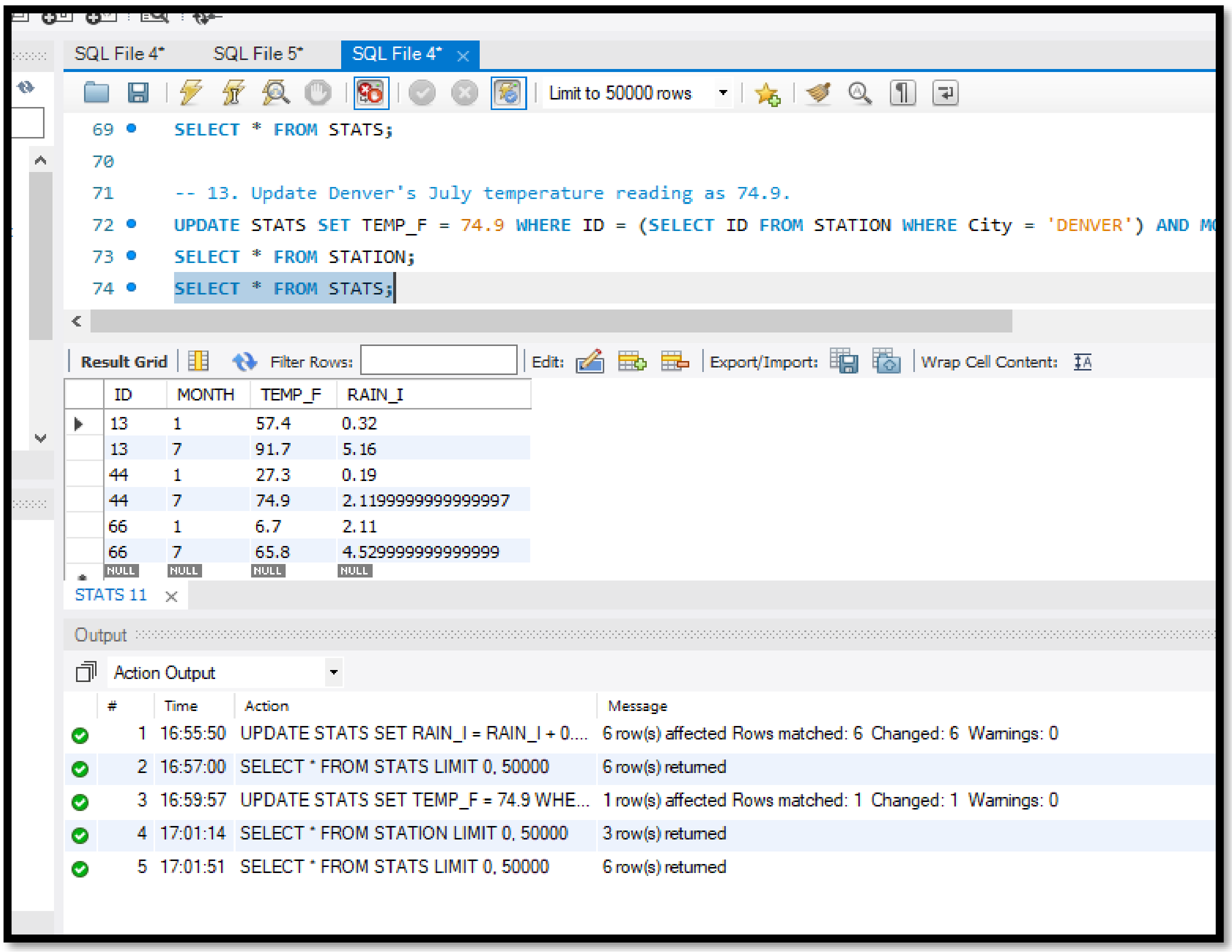
ID	City	State	LAT_N	LONG_W
13	PHOENIX	AZ	33	112
44	DENVER	CO	40	105
66	CARIBOU	ME	47	68
NULL	NULL	NULL	NULL	NULL

STATION 10

Output

Action Output

#	Time	Action	Message
1	16:55:50	UPDATE STATS SET RAIN_I = RAIN_I + 0....	6 row(s) affected Rows matched: 6 Changed: 6 Warnings: 0
2	16:57:00	SELECT * FROM STATS LIMIT 0, 50000	6 row(s) returned
3	16:59:57	UPDATE STATS SET TEMP_F = 74.9 WHE...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
4	17:01:14	SELECT * FROM STATION LIMIT 0, 50000	3 row(s) returned



SQL File 4* | SQL File 5* | SQL File 4*

Limit to 50000 rows

```sql
SELECT * FROM STATS;

-- 13. Update Denver's July temperature reading as 74.9.
UPDATE STATS SET TEMP_F = 74.9 WHERE ID = (SELECT ID FROM STATION WHERE City = 'DENVER') AND MO
SELECT * FROM STATION;
SELECT * FROM STATS;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	0.32
13	7	91.7	5.16
44	1	27.3	0.19
44	7	74.9	2.1199999999999997
66	1	6.7	2.11
66	7	65.8	4.529999999999999
NULL	NULL	NULL	NULL

STATS 11

Output

Action Output

#	Time	Action	Message
1	16:55:50	UPDATE STATS SET RAIN_I = RAIN_I + 0....	6 row(s) affected Rows matched: 6 Changed: 6 Warnings: 0
2	16:57:00	SELECT * FROM STATS LIMIT 0, 50000	6 row(s) returned
3	16:59:57	UPDATE STATS SET TEMP_F = 74.9 WHE...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
4	17:01:14	SELECT * FROM STATION LIMIT 0, 50000	3 row(s) returned
5	17:01:51	SELECT * FROM STATS LIMIT 0, 50000	6 row(s) returned