

Q1: 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	

Page | 1

Answer:

```
1 • create table STATION (  
2     ID integer Primary key ,  
3     CITY CHAR(20),  
4     STATE CHAR(2),  
5     LAT_N integer,  
6     LONG_W integer  
7 );  
8
```

Output :

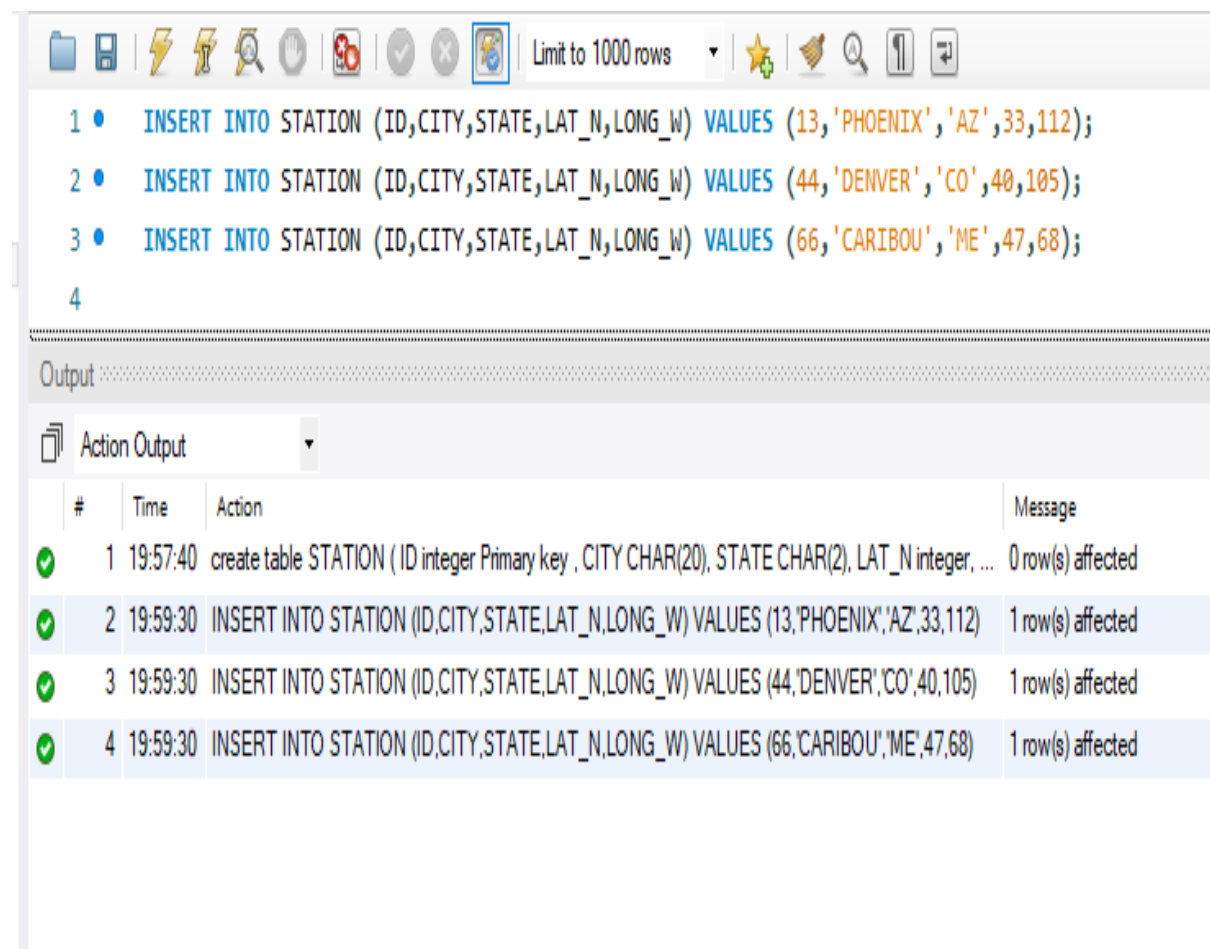
Action Output

#	Time	Action	Message
✓ 1	19:57:40	create table STATION (ID integer Primary key , CITY CHAR(20), STATE CHAR(2), LAT_N integer, ...	0 row(s) affected

Q 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

ANSWER:



The screenshot shows a SQL IDE interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 1000 rows". Below the toolbar, three SQL statements are listed:

- 1 • `INSERT INTO STATION (ID,CITY,STATE,LAT_N, LONG_W) VALUES (13,'PHOENIX','AZ',33,112);`
- 2 • `INSERT INTO STATION (ID,CITY,STATE,LAT_N, LONG_W) VALUES (44,'DENVER','CO',40,105);`
- 3 • `INSERT INTO STATION (ID,CITY,STATE,LAT_N, LONG_W) VALUES (66,'CARIBOU','ME',47,68);`

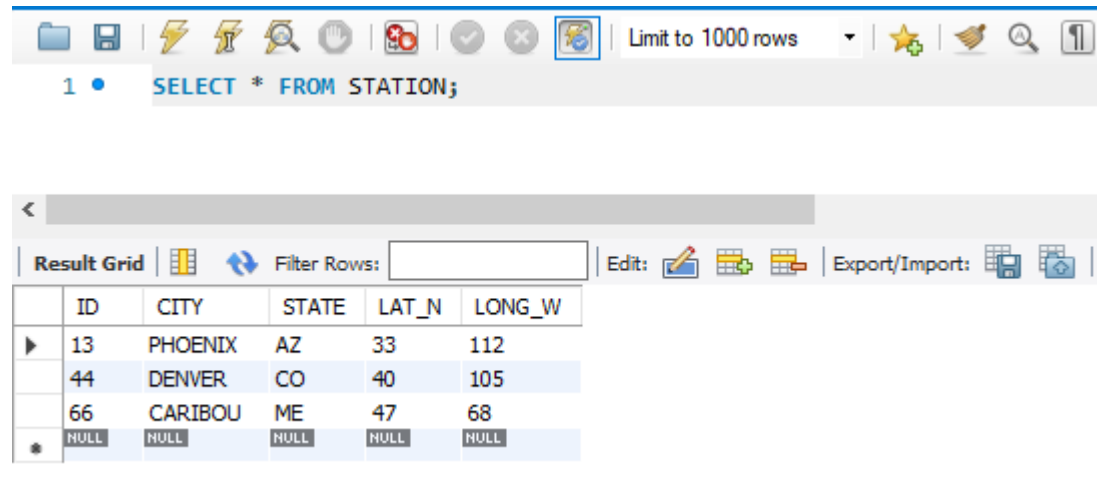
Below the statements, there is a section labeled "Output" with a dropdown menu set to "Action Output". The output table shows the execution results:

#	Time	Action	Message
✓ 1	19:57:40	create table STATION (ID integer Primary key , CITY CHAR(20), STATE CHAR(2), LAT_N integer, ...	0 row(s) affected
✓ 2	19:59:30	INSERT INTO STATION (ID,CITY,STATE,LAT_N, LONG_W) VALUES (13,'PHOENIX','AZ',33,112)	1 row(s) affected
✓ 3	19:59:30	INSERT INTO STATION (ID,CITY,STATE,LAT_N, LONG_W) VALUES (44,'DENVER','CO',40,105)	1 row(s) affected
✓ 4	19:59:30	INSERT INTO STATION (ID,CITY,STATE,LAT_N, LONG_W) VALUES (66,'CARIBOU','ME',47,68)	1 row(s) affected

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

Answer

Page | 3



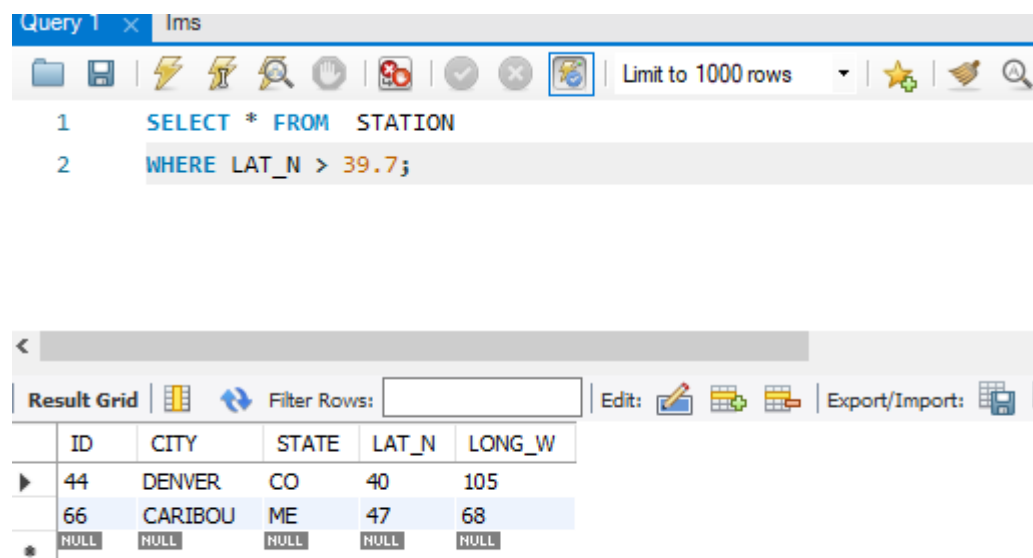
1 • `SELECT * FROM STATION;`

Result Grid

	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

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

Answer



Query 1 x lms

1 `SELECT * FROM STATION`

2 `WHERE LAT_N > 39.7;`

Result Grid

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

Q5: Create another table 'STATS' to store normalized temperature and precipitation data.

Page | 4

Column	Data type	Remark
ID	Number	ID must match with some ID from the STATION table(so name & location will be known).
MONTH	Number	The range of months is between (1 and 12)
TEMP_F	Number	Temperature is in Fahrenheit degrees, Ranging between (-80 and 150)
RAIN_I	Number	Rain is in inches, Ranging between (0 and 100)

There will be no Duplicate **ID** and **MONTH** combination.

Answer

Query 1 x lms

Limit to 1000 rows

```

1 CREATE TABLE STATS (
2   ID integer,
3   MONTH integer CHECK ( MONTH BETWEEN 1 AND 12),
4   TEMP_F decimal(5,2) check(TEMP_F between -80 AND 150),
5   RAIN_I DECIMAL (5,2) check(RAIN_I BETWEEN 0 AND 100)
6 );

```

Output

Action Output

#	Time	Action	Message
✓ 1	19:57:40	create table STATION (ID integer Primary key , CITY CHAR(20), STATE CHAR(2), LAT_N integer...	0 row(s) affected
✓ 2	19:59:30	INSERT INTO STATION (ID,CITY,STATE,LAT_N, LONG_W) VALUES (13,PHOE	create table STATION (ID integer Primary key , CITY CHAR(20), STATE CHAR(2), LAT_N integer, LONG_W integer) affected
✓ 3	19:59:30	INSERT INTO STATION (ID,CITY,STATE,LAT_N, LONG_W) VALUES (44,'DENV	ID integer Primary key , CITY CHAR(20), STATE CHAR(2), LAT_N integer, LONG_W integer affected
✓ 4	19:59:30	INSERT INTO STATION (ID,CITY,STATE,LAT_N, LONG_W) VALUES (66,'CARIB	affected
✓ 5	20:01:13	SELECT * FROM STATION LIMIT 0, 1000	returned
✓ 6	20:01:46	SELECT * FROM STATION LIMIT 0, 1000	returned
✓ 7	20:04:25	SELECT * FROM STATION WHERE LAT_N > 39.7 LIMIT 0, 1000	2 row(s) returned
✓ 8	20:19:48	CREATE TABLE STATS (ID integer, MONTH integer CHECK (MONTH BETWEEN 1 AND 12), ...	0 row(s) affected

Q6: Populate the table STATS with some statistics for January and July

Page | 5

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

Answer

Query 1 x lms

Limit to 1000 rows

```

1 • INSERT INTO STATS (ID,MONTH,TEMP_F,RAIN_I) VALUES (13,1,57.4,.31);
2 • INSERT INTO STATS (ID,MONTH,TEMP_F,RAIN_I) VALUES (13,7,91.7,5.15);
3 • INSERT INTO STATS (ID,MONTH,TEMP_F,RAIN_I) VALUES (44,1,27.3,.18);
4 • INSERT INTO STATS (ID,MONTH,TEMP_F,RAIN_I) VALUES (44,7,74.8,2.11);
5 • INSERT INTO STATS (ID,MONTH,TEMP_F,RAIN_I) VALUES (66,1,6.7,2.1);
6 • INSERT INTO STATS (ID,MONTH,TEMP_F,RAIN_I) VALUES (66,7,65.8,4.52);

```

Output

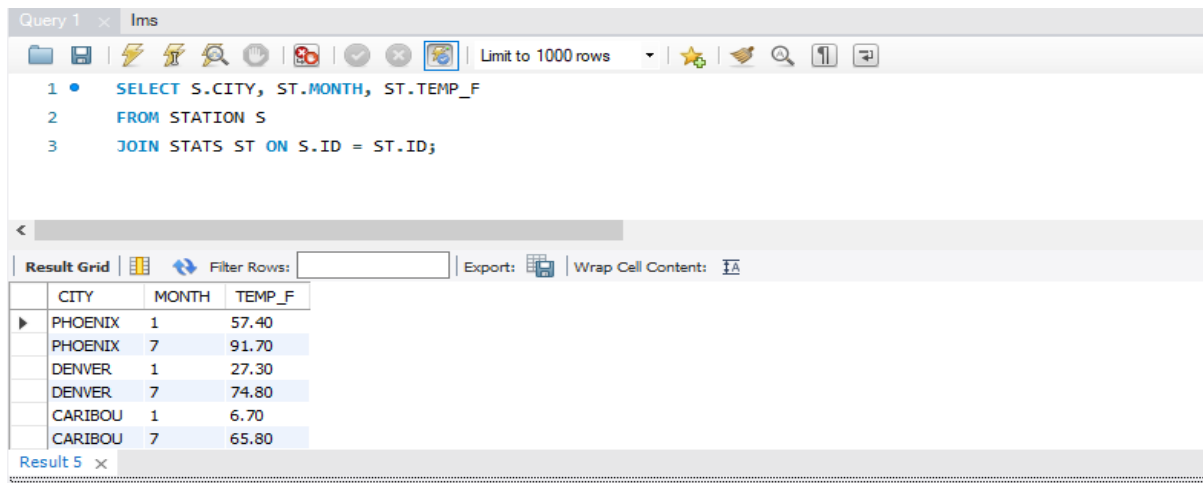
Action Output

#	Time	Action	Message
✓ 1	19:57:40	create table STATION (ID integer Primary key , CITY CHAR(20), STATE CHAR(2), LAT_N integer...	0 row(s) affected
✓ 2	19:59:30	INSERT INTO STATION (ID,CITY,STATE,LAT_N, LONG_W) VALUES (13,'PHOENIX','AZ',33,112)	1 row(s) affected
✓ 3	19:59:30	INSERT INTO STATION (ID,CITY,STATE,LAT_N, LONG_W) VALUES (44,'DENVER','CO',40,105)	1 row(s) affected
✓ 4	19:59:30	INSERT INTO STATION (ID,CITY,STATE,LAT_N, LONG_W) VALUES (66,'CARIBOU','ME',47,68)	1 row(s) affected
✓ 5	20:01:13	SELECT * FROM STATION LIMIT 0, 1000	3 row(s) returned
✓ 6	20:01:46	SELECT * FROM STATION LIMIT 0, 1000	3 row(s) returned
✓ 7	20:04:25	SELECT * FROM STATION WHERE LAT_N > 39.7 LIMIT 0, 1000	2 row(s) returned
✓ 8	20:19:48	CREATE TABLE STATS (ID integer, MONTH integer CHECK (MONTH BETWEEN 1 AND 12), ...	0 row(s) affected
✓ 9	20:23:51	INSERT INTO STATS (ID,MONTH,TEMP_F,RAIN_I) VALUES (13,1,57.4,.31)	1 row(s) affected
✓ 10	20:23:51	INSERT INTO STATS (ID,MONTH,TEMP_F,RAIN_I) VALUES (13,7,91.7,5.15)	1 row(s) affected
✓ 11	20:23:51	INSERT INTO STATS (ID,MONTH,TEMP_F,RAIN_I) VALUES (44,1,27.3,.18)	1 row(s) affected
✓ 12	20:23:51	INSERT INTO STATS (ID,MONTH,TEMP_F,RAIN_I) VALUES (44,7,74.8,2.11)	1 row(s) affected
✓ 13	20:23:51	INSERT INTO STATS (ID,MONTH,TEMP_F,RAIN_I) VALUES (66,1,6.7,2.1)	1 row(s) affected
✓ 14	20:23:51	INSERT INTO STATS (ID,MONTH,TEMP_F,RAIN_I) VALUES (66,7,65.8,4.52)	1 row(s) affected

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

Answer

Page | 6



Query 1 x Ims

Limit to 1000 rows

```
1 • SELECT S.CITY, ST.MONTH, ST.TEMP_F
2 FROM STATION S
3 JOIN STATS ST ON S.ID = ST.ID;
```

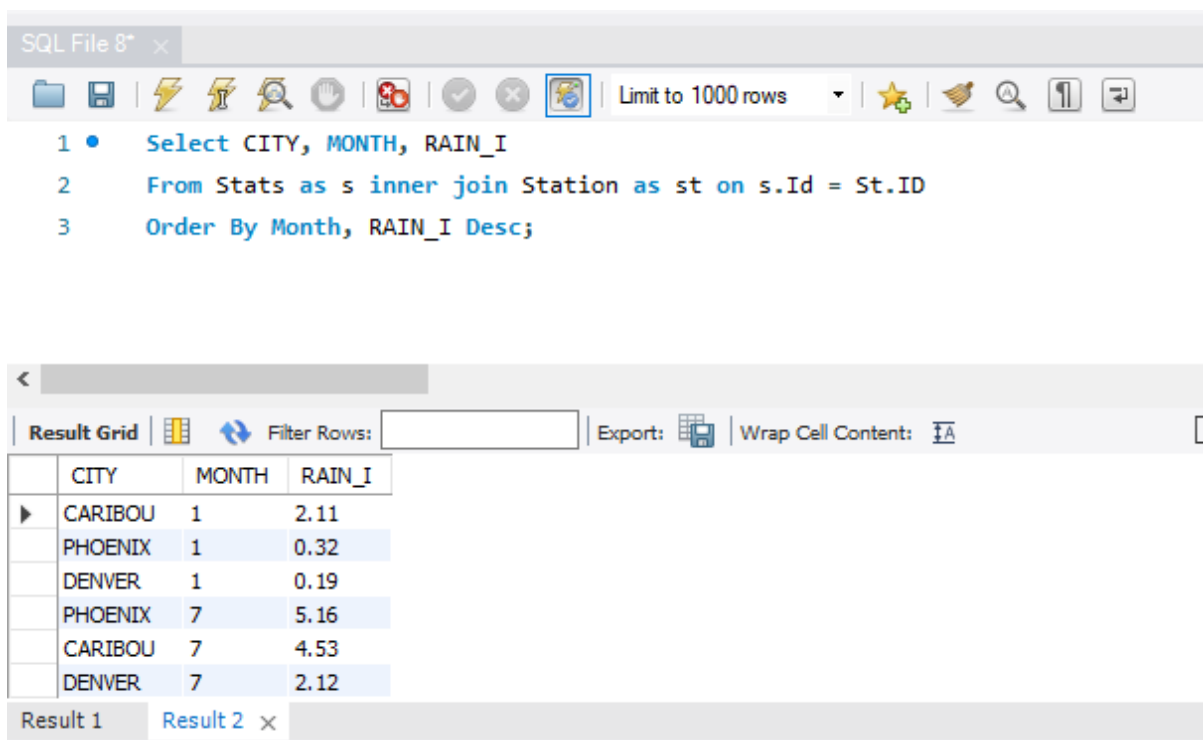
Result Grid

	CITY	MONTH	TEMP_F
▶	PHOENIX	1	57.40
	PHOENIX	7	91.70
	DENVER	1	27.30
	DENVER	7	74.80
	CARIBOU	1	6.70
	CARIBOU	7	65.80

Result 5 x

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.

Answer



SQL File 8* x

Limit to 1000 rows

```
1 • Select CITY, MONTH, RAIN_I
2 From Stats as s inner join Station as st on s.Id = St.ID
3 Order By Month, RAIN_I Desc;
```

Result Grid

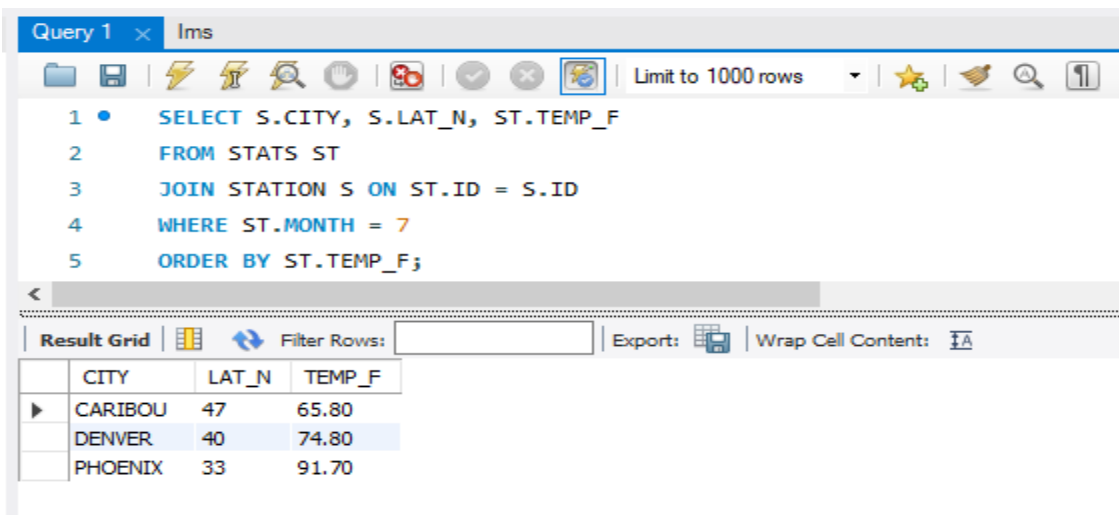
	CITY	MONTH	RAIN_I
▶	CARIBOU	1	2.11
	PHOENIX	1	0.32
	DENVER	1	0.19
	PHOENIX	7	5.16
	CARIBOU	7	4.53
	DENVER	7	2.12

Result 1 Result 2 x

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

Answer

Page | 7



Query 1 x lms

Limit to 1000 rows

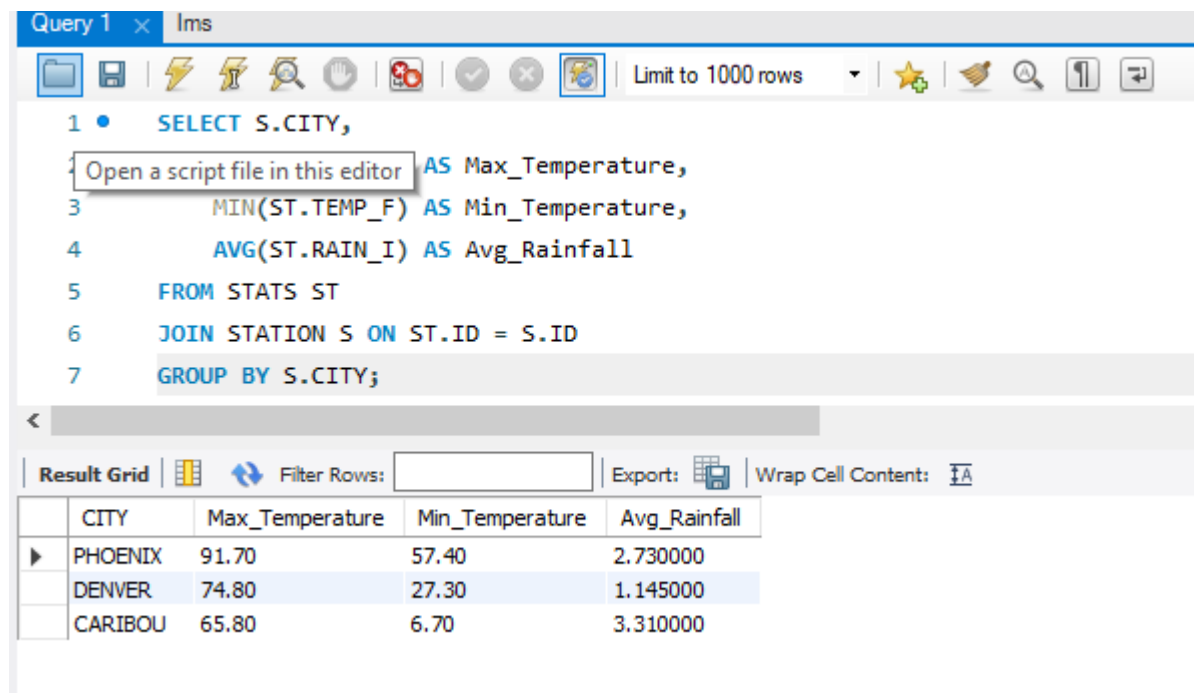
```
1 • SELECT S.CITY, S.LAT_N, ST.TEMP_F
2 FROM STATS ST
3 JOIN STATION S ON ST.ID = S.ID
4 WHERE ST.MONTH = 7
5 ORDER BY ST.TEMP_F;
```

Result Grid

	CITY	LAT_N	TEMP_F
▶	CARIBOU	47	65.80
	DENVER	40	74.80
	PHOENIX	33	91.70

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

Answer



Query 1 x lms

Limit to 1000 rows

```
1 • SELECT S.CITY,
2 AS Max_Temperature,
3 MIN(ST.TEMP_F) AS Min_Temperature,
4 AVG(ST.RAIN_I) AS Avg_Rainfall
5 FROM STATS ST
6 JOIN STATION S ON ST.ID = S.ID
7 GROUP BY S.CITY;
```

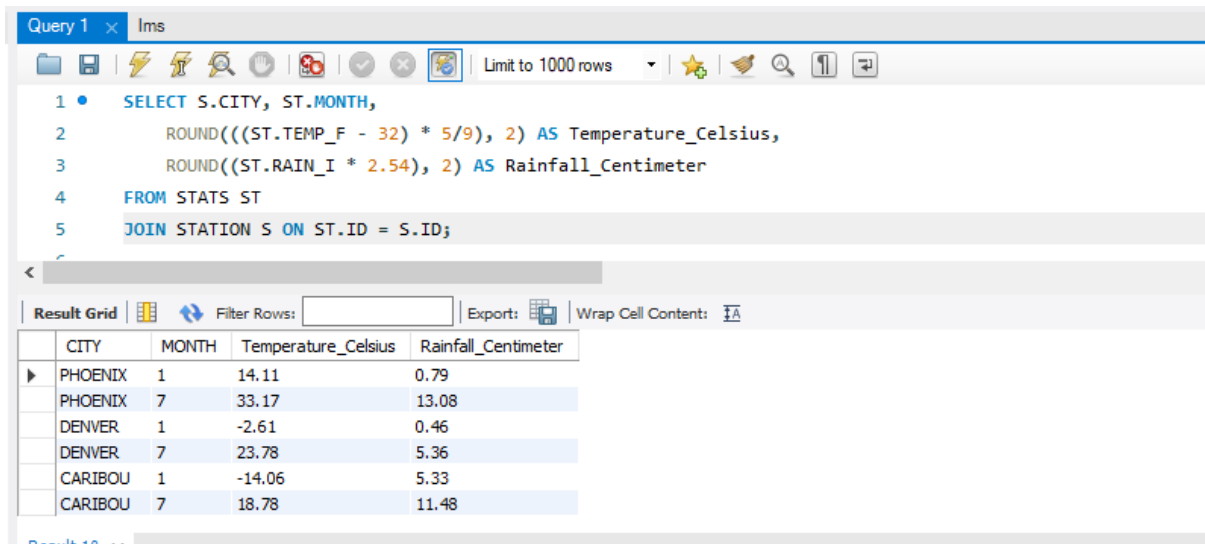
Result Grid

	CITY	Max_Temperature	Min_Temperature	Avg_Rainfall
▶	PHOENIX	91.70	57.40	2.730000
	DENVER	74.80	27.30	1.145000
	CARIBOU	65.80	6.70	3.310000

Q11: Execute a query to display each city's monthly temperature in **Celsius** and rainfall in Centimetres.

Answer

Page | 8



The screenshot shows a SQL query editor with a query named 'Query 1'. The query is as follows:

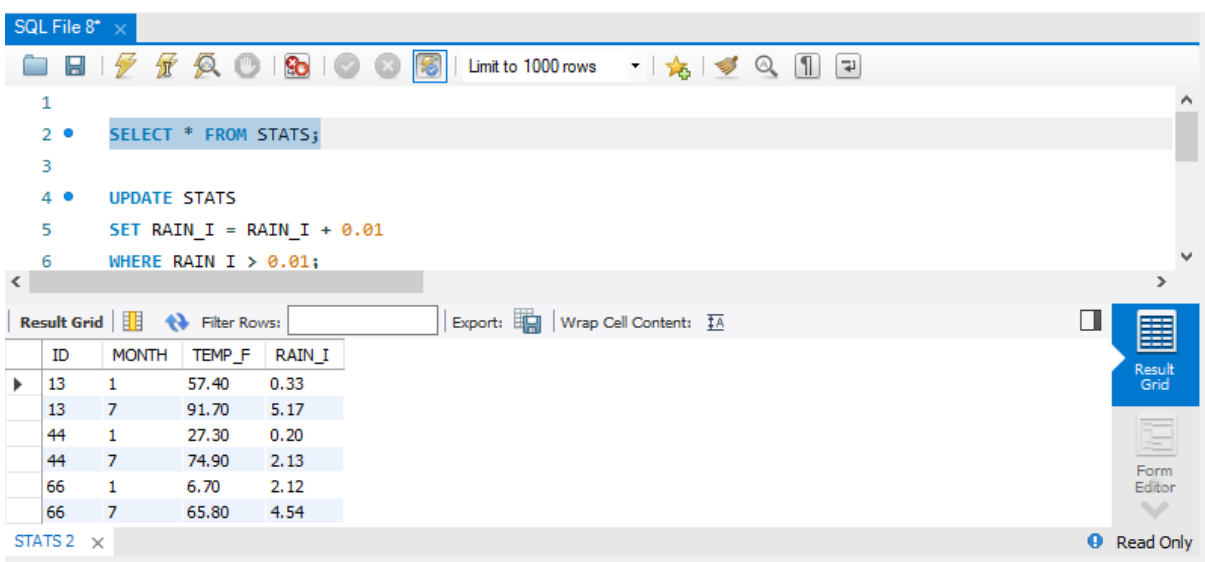
```
1 • SELECT S.CITY, ST.MONTH,
2     ROUND(((ST.TEMP_F - 32) * 5/9), 2) AS Temperature_Celsius,
3     ROUND((ST.RAIN_I * 2.54), 2) AS Rainfall_Centimeter
4 FROM STATS ST
5 JOIN STATION S ON ST.ID = S.ID;
```

Below the query, the 'Result Grid' is displayed, showing the results of the query. The table has four columns: CITY, MONTH, Temperature_Celsius, and Rainfall_Centimeter. The results are as follows:

	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

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

Answer



The screenshot shows a SQL query editor with a query named 'SQL File 8*'. The query is as follows:

```
1
2 • SELECT * FROM STATS;
3
4 • UPDATE STATS
5 SET RAIN_I = RAIN_I + 0.01
6 WHERE RAIN_I > 0.01;
```

Below the query, the 'Result Grid' is displayed, showing the results of the query. The table has five columns: ID, MONTH, TEMP_F, and RAIN_I. The results are as follows:

	ID	MONTH	TEMP_F	RAIN_I
▶	13	1	57.40	0.33
	13	7	91.70	5.17
	44	1	27.30	0.20
	44	7	74.90	2.13
	66	1	6.70	2.12
	66	7	65.80	4.54

Q13: Update **Denver's July** temperature reading as **74.9**.

Answer

Page | 9

The screenshot shows the SQL File 8+ interface. The SQL editor contains the following queries:

```
1  
2 • SELECT * FROM STATS;  
3  
4 • UPDATE STATS  
5   SET TEMP_F = 74.9  
6   WHERE ID = 44 AND MONTH = 7;
```

The Result Grid shows the following data:

	ID	MONTH	TEMP_F	RAIN_I
▶	13	1	57.40	0.32
	13	7	91.70	5.16
	44	1	27.30	0.19
	44	7	74.90	2.12
	66	1	6.70	2.11
	66	7	65.80	4.53

The Output section shows the Action Output:

#	Time	Action	Message
✓ 1	23:12:08	SELECT * FROM STATS LIMIT 0, 1000	6 row(s) returned

THANK YOU