

# **Assignment – Advance SQL [Major]**

**Mojo Assignment -01**

**By-RAJESH KUMAR**

## Assignment - SQL [Major]

**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	

**ANS.1:-**

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

SQL Worksheet

```

1 CREATE TABLE STATION
2 (ID NUMBER PRIMARY KEY,
3 CITY CHAR(20),
4 STATE CHAR(2),
5 LAT_N REAL,
6 LONG_W REAL);
```

Table created.

**Q2)** 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

**ANS.2:-**

```
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);
```

**SQL Worksheet**

```
1 INSERT INTO STATION VALUES(13, 'Phoenix', 'AZ', 33, 112);
2 INSERT INTO STATION VALUES(44, 'Denver', 'CO', 40, 105);
3 INSERT INTO STATION VALUES(66, 'Caribou', 'ME', 47, 68);
4
5
```

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

**Q3)** Execute a query to look at table "STATION" in undefined order:

**ANS.3:-** `SELECT * FROM STATION`

**SQL Worksheet**

```
1 SELECT * FROM STATION
2
```

ID	CITY	STATE	LAT_N	LONG_W
13	Phoenix	AZ	33	112
44	Denver	CO	40	105
66	Caribou	ME	47	68

[Download CSV](#)

3 rows selected.

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

**ANS.4:-**

```
SELECT * FROM STATION
WHERE LAT_N > 39.7;
```

### SQL Worksheet

```

1  SELECT * FROM STATION
2  WHERE LAT_N > 39.7;
3

```

ID	CITY	STATE	LAT_N	LONG_W
44	Denver	CO	40	105
66	Caribou	ME	47	68

Download CSV

2 rows selected.

**Q5)** Create another table, "STATS", to store normalized temperature and precipitation data:

Column	Data type	Remark
ID	Number	must match some STATION table ID (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

**(Note:-** There will be no Duplicate ID and MONTH combination.)

**ANS.5:-**

```
CREATE TABLE STATS
(ID NUMBER REFERENCES STATION(ID),
MONTH NUMBER 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));
```

SQL Worksheet

```

1 CREATE TABLE STATS
2 (ID NUMBER REFERENCES STATION(ID),
3 MONTH NUMBER CHECK(MONTH BETWEEN 1 AND 12),
4 TEMP_F REAL CHECK(TEMP_F BETWEEN -80 AND 150),
5 RAIN_I REAL CHECK(RAIN_I BETWEEN 0 AND 100),
6 PRIMARY KEY(ID, MONTH));

```

Table created.

**Q6)** 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

**ANS.6:-**

```
INSERT INTO STATS VALUES(13, 1, 57.4, 0.31);
INSERT INTO STATS VALUES(13, 7, 91.7, 5.15);
INSERT INTO STATS VALUES(44, 1, 27.3, 0.18);
INSERT INTO STATS VALUES(44, 7, 74.8, 2.11);
INSERT INTO STATS VALUES(66, 1, 6.7, 2.10);
INSERT INTO STATS VALUES(66, 7, 65.8, 4.52);
```

SQL Worksheet

```

1  INSERT INTO STATS VALUES(13, 1, 57.4, 0.31);
2  INSERT INTO STATS VALUES(13, 7, 91.7, 5.15);
3  INSERT INTO STATS VALUES(44, 1, 27.3, 0.18);
4  INSERT INTO STATS VALUES(44, 7, 74.8, 2.11);
5  INSERT INTO STATS VALUES(66, 1, 6.7, 2.10);
6  INSERT INTO STATS VALUES(66, 7, 65.8, 4.52);
7  |

```

```

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

```

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

**ANS.7:-**

1) Display Temperature in Fahrenheit For each city.

(Temp\_F Stands for Temperature in Fahrenheit.)

**SQL Worksheet**

```

1 SELECT CITY, TEMP_F FROM STATION, STATS
2 WHERE STATION.ID = STATS.ID;

```

CITY	TEMP_F
Phoenix	57.4
Phoenix	91.7
Denver	27.3
Denver	74.8
Caribou	6.7
Caribou	65.8

[Download CSV](#)

6 rows selected.

2) Display Temperature in Fahrenheit For each city Monthwise.

**SQL Worksheet**

```

1 SELECT CITY, MONTH, TEMP_F FROM STATION, STATS
2 WHERE STATION.ID = STATS.ID;

```

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

[Download CSV](#)

6 rows selected.

**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.

**ANS.8:-**

**SQL Worksheet**

```

1 SELECT MONTH, STATS.ID, RAIN_I, CITY FROM STATS, STATION
2 WHERE STATS.ID = STATION.ID
3 ORDER BY MONTH, RAIN_I DESC;

```

MONTH	ID	RAIN_I	CITY
1	66	2.1	Caribou
1	13	.31	Phoenix
1	44	.18	Denver
7	13	5.15	Phoenix
7	66	4.52	Caribou
7	44	2.11	Denver

Download CSV

6 rows selected.

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



**ANS.9:-**

**SQL Worksheet**

```

1  SELECT LAT_N, CITY, TEMP_F FROM STATS, STATION
2  WHERE MONTH=7 AND STATS.ID= STATION.ID
3  ORDER BY TEMP_F;
4

```

LAT_N	CITY	TEMP_F
47	Caribou	65.8
40	Denver	74.8
33	Phoenix	91.7

Download CSV

3 rows selected.

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

**ANS.10:-** 1)

**SQL Worksheet**

```

1 SELECT MAX(TEMP_F), MIN(TEMP_F),
2 AVG(RAIN_I), CITY FROM STATS, STATION
3 WHERE STATS.ID = STATION.ID
4 GROUP BY CITY;
5

```

MAX(TEMP_F)	MIN(TEMP_F)	AVG(RAIN_I)	CITY
65.8	6.7	3.31	Caribou
91.7	57.4	2.73	Phoenix
74.8	27.3	1.145	Denver

Download CSV

3 rows selected.

- 1) Max and Min Temperatures as well as Average Rainfall for each city in proper manner.

SQL Worksheet

```

1 SELECT MAX(TEMP_F) AS Max_Temperature, MIN(TEMP_F) AS Min_Temperature,
2 AVG(RAIN_I) AS Average_Rainfall, CITY FROM STATS, STATION
3 WHERE STATS.ID = STATION.ID
4 GROUP BY CITY
5 ORDER BY CITY DESC;
6

```

MAX_TEMPERATURE	MIN_TEMPERATURE	AVERAGE_RAINFALL	CITY
91.7	57.4	2.73	Phoenix
74.8	27.3	1.145	Denver
65.8	6.7	3.31	Caribou

Download CSV

3 rows selected.

**Q11)** Execute a query to display each city's monthly temperature in Celcius and rainfall in Centimeter.

**ANS.11:-**

1)Create a view (derived table or persistent query) to convert Fahrenheit to Celsius and inches to centimeters.



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

**ANS.12:-**

- 1) Update all rows of table 'STATS' to compensate for faulty rain gauges.

SQL Worksheet

1

UPDATE STATS SET RAIN\_I = RAIN\_I+0.01;

6 row(s) updated.

2) Now take a look:

SQL Worksheet

```

1  UPDATE STATS SET RAIN_I = RAIN_I+0.01;
2
3  SELECT * FROM STATS;

```

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

Download CSV

6 rows selected.

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

**ANS.13:-**

1)Update one row, Denver's July Temperature reading.

### SQL Worksheet

```

1 UPDATE STATS SET TEMP_F = 74.9
2 WHERE ID = 44
3 AND MONTH = 7;
4

```

1 row(s) updated.

2)Now take a look:

### SQL Worksheet

```

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

```

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

[Download CSV](#)

6 rows selected.