

“SQL Assignment Submission- (Vikas Kumar Maurya) “

VIKAS KUMAR MAURYA
VIKASM16794@GMAIL.COM

BUSINESS ANALYST
Assignment – SQL [Major]

CREATE database `weather_observation`;

USE `weather_observation`;

-- Q1) Create a table **“STATION”** to store information about weather observation stations:

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

#	Time	Action	Message	Duration / Fetch
13	16:12:34	USE weather_observation	0 row(s) affected	0.047 sec
14	16:14:04	CREATE TABLE STATION (ID INT PRIMARY KEY, CITY CHAR(20), STATE CHAR(2), LAT...	0 row(s) affected	0.531 sec

```
SELECT * FROM STATION;
```

ID	CITY	STATE	LAT_N	LONG_W
NULL	NULL	NULL	NULL	NULL

-- Q2) Insert the following records into the table:

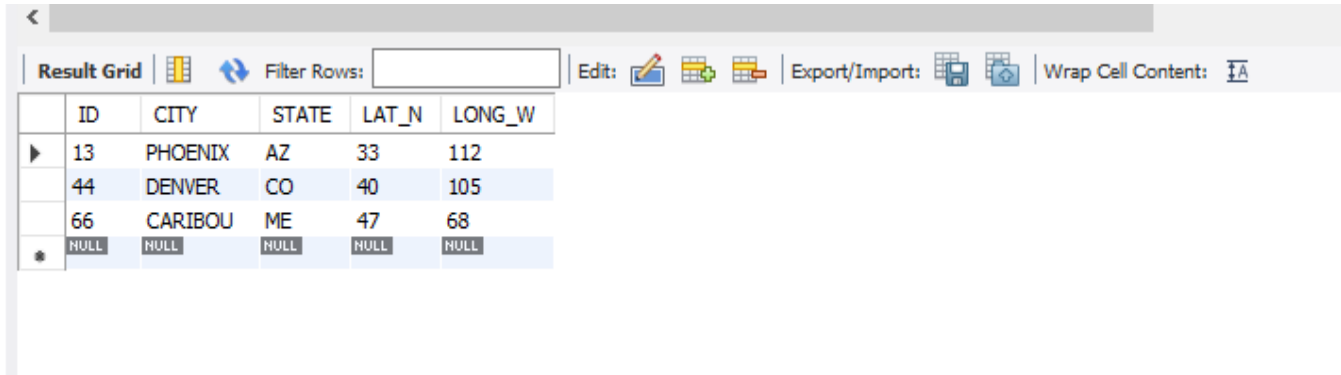
```
INSERT INTO STATION (ID, CITY, STATE, LAT_N, LONG_W) VALUES (13,'PHOENIX','AZ',33,112),  
(44,'DENVER','CO',40,105), (66,'CARIBOU','ME',47,68);
```

13				
14	•	INSERT INTO STATION (ID,CITY,STATE,LAT_N, LONG_W)		
15		VALUES (13,'PHOENIX','AZ',33,112),		
16		(44,'DENVER','CO',40,105),		
17		(66,'CARIBOU','ME',47,68);		
18				
19		-- Q3) Execute a query to look at table STATION in undefined order.		
20	•	SELECT * FROM STATION;		
21				

#	Time	Action	Message
15	16:15:51	SELECT * FROM STATION LIMIT 0, 1000	0 row(s) returned
16	16:17:01	INSERT INTO STATION (ID,CITY,STATE,LAT_N, LONG_W) VALUES (13,'PHOENIX','AZ',...	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0

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

```
SELECT * FROM STATION;
```

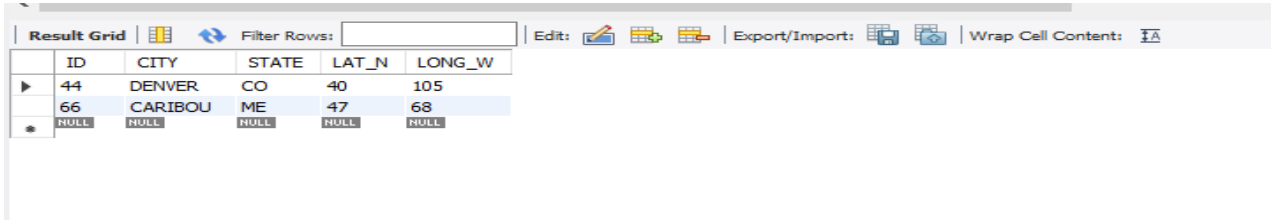


	ID	CITY	STATE	LAT_N	LONG_W
▶	13	PHOENIX	AZ	33	112
	44	DENVER	CO	40	105
	66	CARIBOU	ME	47	68
*	NULL	NULL	NULL	NULL	NULL

-- Q4) 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

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

```
CREATE TABLE STATS
```

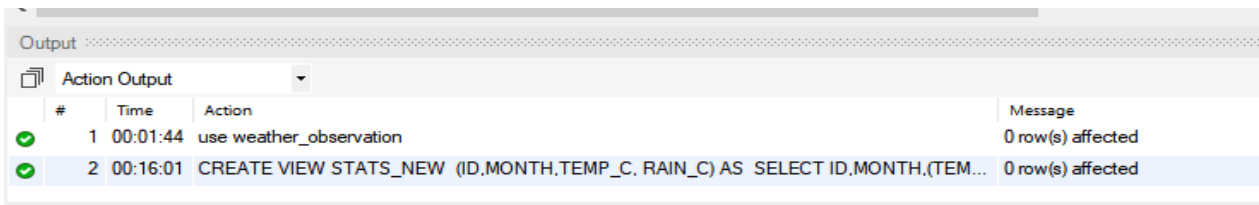
```
(ID INT references STATION (ID),
```

```
MONTH INT CHECK (MONTH BETWEEN 1 AND 12),
```

```
TEMP_F REAL CHECK (TEMP_F BETWEEN -80 AND 150),
```

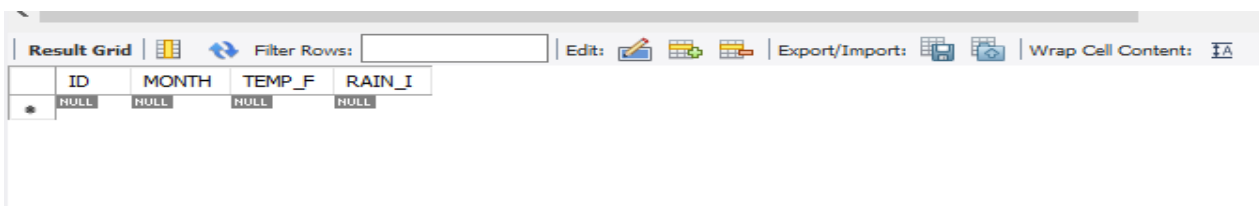
```
RAIN_I REAL CHECK (RAIN_I BETWEEN 0 AND 100),
```

```
primary KEY(ID,MONTH));
```



#	Time	Action	Message
✓ 1	00:01:44	use weather_observation	0 row(s) affected
✓ 2	00:16:01	CREATE VIEW STATS_NEW (ID,MONTH,TEMP_C, RAIN_C) AS SELECT ID,MONTH,(TEM...	0 row(s) affected

```
SELECT * FROM STATS;
```



	ID	MONTH	TEMP_F	RAIN_I
*	NULL	NULL	NULL	NULL

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

```
INSERT INTO STATS (ID,MONTH,TEMP_F, RAIN_I) VALUES
```

```
(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
21	16:19:19	SELECT * FROM STATS LIMIT 0, 1000	0 row(s) returned
22	16:19:35	INSERT INTO STATS (ID,MONTH,TEMP_F, RAIN_I) VALUES (13,1,57.4,0.31), (13,7,91.7,...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0
			Duration / Fetch
			0.047 sec / 0.000 se
			0.109 sec

```
SELECT * FROM STATS;
```

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

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

```
SELECT A.CITY , B.TEMP_F FROM STATION A JOIN STATS B ON A.ID = B.ID;
```

Result Grid		
Filter Rows:		
Export: Wrap Cell Content:		
	CITY	TEMP_F
▶	PHOENIX	57.4
	PHOENIX	91.7
	DENVER	27.3
	DENVER	74.8
	CARIBOU	6.7
	CARIBOU	65.8

-- 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 A JOIN STATION B ON A.ID = B.ID ORDER BY MONTH, RAIN_I DESC;
```

“Or”

```
SELECT A.ID, A.MONTH, A.TEMP_F, A.RAIN_I, B.CITY FROM STATS A, STATION B WHERE A.ID=B.ID ORDER BY A.MONTH, A.RAIN_I DESC;
```

Result Grid					
		Filter Rows:			
		Export:	Wrap Cell Content:		
	ID	MONTH	TEMP_F	RAIN_I	CITY
▶	66	1	6.7	2.1	CARIBOU
	13	1	57.4	0.31	PHOENIX
	44	1	27.3	0.18	DENVER
	13	7	91.7	5.15	PHOENIX
	66	7	65.8	4.52	CARIBOU
	44	7	74.8	2.11	DENVER

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

```
SELECT A.TEMP_F, B.CITY, B.LAT_N FROM STATS A JOIN STATION B ON A.ID=B.ID WHERE MONTH =7 ORDER BY TEMP_F;
```

Result Grid			
		Filter Rows:	
		Export:	Wrap Cell Content:
	TEMP_F	CITY	LAT_N
▶	65.8	CARIBOU	47
	74.8	DENVER	40
	91.7	PHOENIX	33

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

```
SELECT MAX(A.TEMP_F), MIN(A.TEMP_F), ROUND(avg(A.RAIN_I),2) AS AVG_RAIN, B.CITY FROM STATS A
JOIN STATION B ON A.ID = B.ID group by B.CITY ;
```

Result Grid				
		Filter Rows:		
		Export:	Wrap Cell Content:	
	MAX(A.TEMP_F)	MIN(A.TEMP_F)	AVG_RAIN	CITY
▶	91.7	57.4	2.73	PHOENIX
	74.8	27.3	1.14	DENVER
	65.8	6.7	3.31	CARIBOU

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

```
CREATE VIEW STATS_NEW (ID, MONTH, TEMP_C, RAIN_C) AS
SELECT ID, MONTH, (TEMP_F-32)*5/9, RAIN_I*0.3937
FROM STATS;
```

SELECT * FROM STATS_NEW;

Result Grid				
		Filter Rows:		
		Export:		
		Wrap Cell Content:		
	ID	MONTH	TEMP_C	RAIN_C
▶	13	1	14.11111111111111	0.122047
	13	7	33.16666666666666	2.027555
	44	1	-2.61111111111111	0.070866
	44	7	23.77777777777778	0.830707
	66	1	-14.05555555555555	0.82677
	66	7	18.77777777777778	1.779523999999999

-- Q12) 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;

Output				
Action Output				
#	Time	Action	Message	
5	00:29:29	UPDATE STATS SET RAIN_I = RAIN_I+0.01	Error Code: 1054. Unknown column '0.01' in 'field list'	
6	00:30:11	UPDATE STATS SET RAIN_I = RAIN_I+0.01	6 row(s) affected Rows matched: 6 Changed: 6 Warnings: 0	

SELECT * FROM STATS;

Result Grid				
		Filter Rows:		
		Edit:		
		Export/Import:		
		Wrap Cell C		
	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

-- Q13) Update Denver's July temperature reading as 74.9.

UPDATE STATS SET TEMP_F = 74.9 WHERE MONTH= 7 AND ID=44;

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 7 ×

Output				
Action Output				
#	Time	Action	Message	
15	00:38:55	UPDATE STATS SET TEMP_F = 74.9 WHERE MONTH= 7 AND ID=44	1 row(s) affected Rows m	
16	00:39:15	SELECT * FROM STATS LIMIT 0, 1000	6 row(s) returned	