**“SQL Assignment Submission- (Vikas Kumar Maurya) “**

**VIKAS KUMAR MAURYA**

[**VIKASM16794@GMAIL.COM**](mailto:VIKASM16794@GMAIL.COM)

**BUSINESS ANALYST**

**Assignment – SQL [Major]**

**CREATE database weather\_observation;**

**USE weather\_observation;**

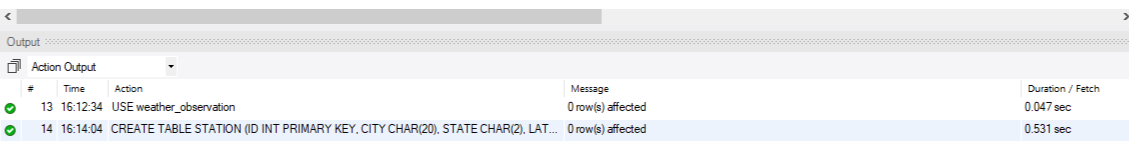
***-- Q1) Create a table “STATION” to store information about weather observation stations:***

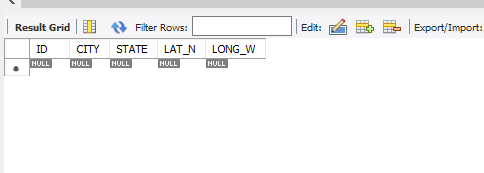
CREATE TABLE STATION (ID INT PRIMARY KEY,

CITY CHAR (20),

STATE CHAR (2),

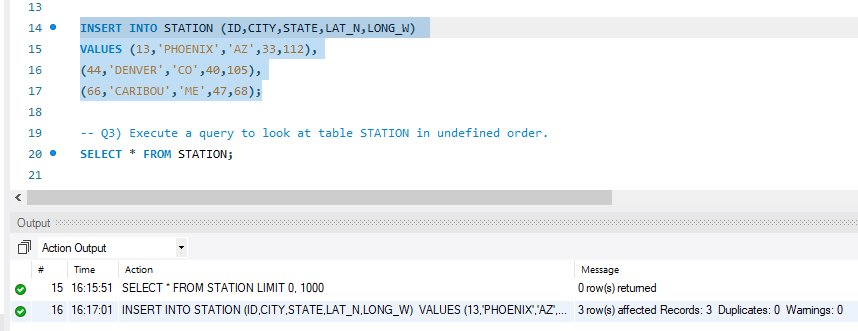
LAT\_N INT,

LONG\_W INT); 

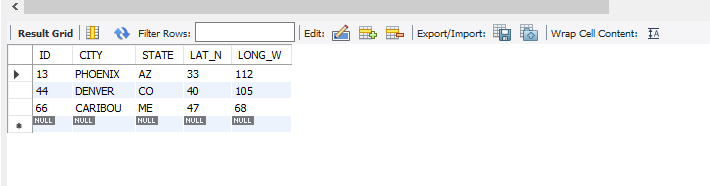
SELECT \* FROM STATION; 

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

INSERT INTO STATION (ID, CITY, STATE, LAT\_N, LONG\_W) VALUES (13,'PHOENIX','AZ',33,112),

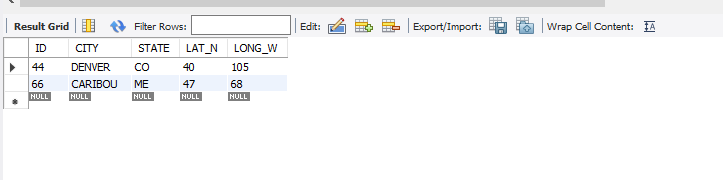
(44,'DENVER','CO',40,105), (66,'CARIBOU','ME',47,68); 

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

SELECT \* FROM STATION; 

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

SELECT \* FROM STATION

WHERE LAT\_N>39.7; 

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

CREATE TABLE STATS

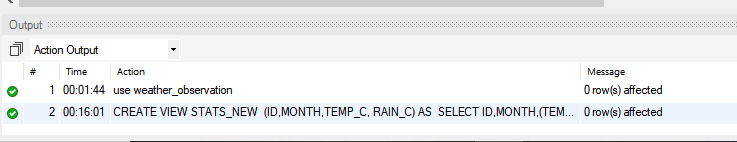
(ID INT references STATION (ID),

MONTH INT CHECK (MONTH BETWEEN 1 AND 12),

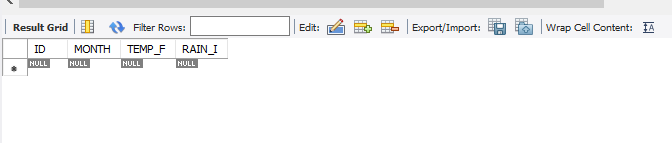
TEMP\_F REAL CHECK (TEMP\_F BETWEEN -80 AND 150),

RAIN\_I REAL CHECK ( RAIN\_I BETWEEN 0 AND 100),

primary KEY(ID,MONTH));



SELECT \* FROM STATS;



***-- 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),

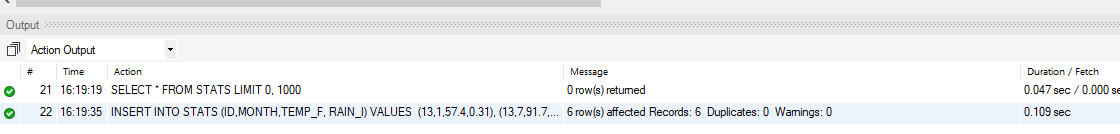
(13,7,91.7,5.15),

(44,1,27.3,0.18),

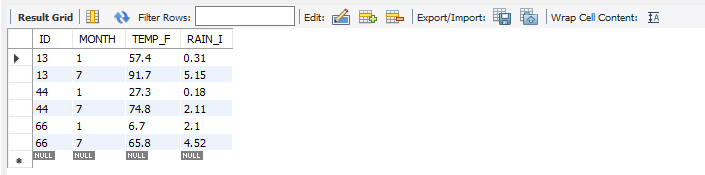
(44,7,74.8,2.11),

(66,1,6.7,2.1),

(66,7,65.8,4.52);

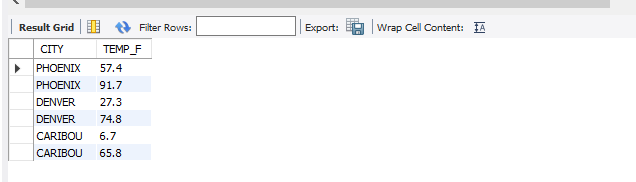


SELECT \* FROM STATS;



***-- 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;

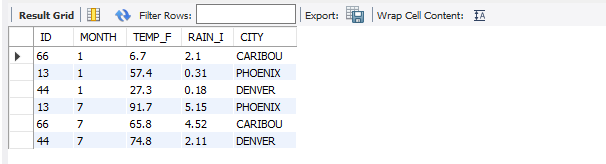


***-- Q8) Execute a query to look at the table STATS, ordered by month and greatest rainfall, with columns rearranged. It should also show the corresponding cities.***

SELECT \* FROM STATS A JOIN STATION B ON A.ID = B.ID ORDER BY MONTH, RAIN\_I DESC;

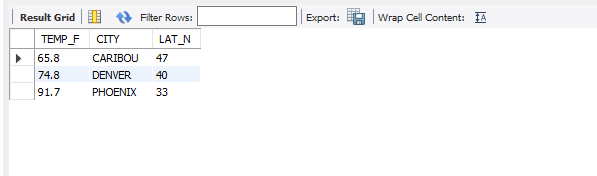
**“Or”**

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;



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

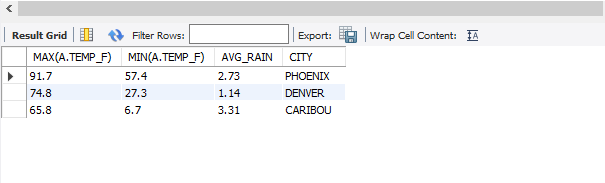
SELECT A.TEMP\_F,B.CITY,B.LAT\_N FROM STATS A JOIN STATION B ON A.ID=B.ID WHERE MONTH =7 ORDER BY TEMP\_F;



***-- 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 ;

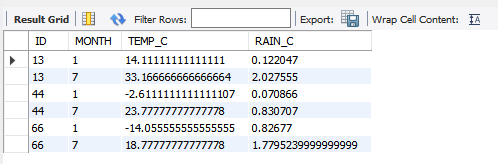


***-- Q11) Execute a query to display each city’s monthly temperature in Celcius and rainfall in Centimeter.***

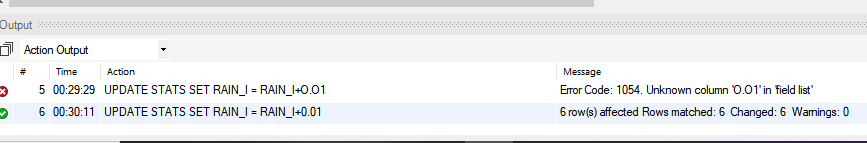
CREATE VIEW STATS\_NEW (ID,MONTH,TEMP\_C, RAIN\_C) AS

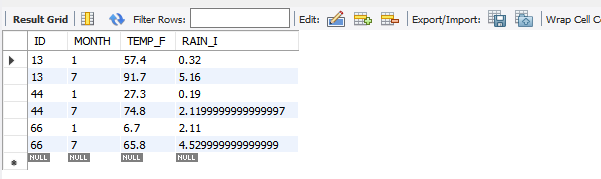
SELECT ID, MONTH,(TEMP\_F-32)\*5/9, RAIN\_I\*0.3937

FROM STATS;

SELECT \* FROM STATS\_NEW; 

***-- 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; 

SELECT \* FROM STATS; 

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

UPDATE STATS SET TEMP\_F = 74.9 WHERE MONTH= 7 AND ID=44; 