SEPTEMBER 3, 2024

LAB HOMEWORK 1 A

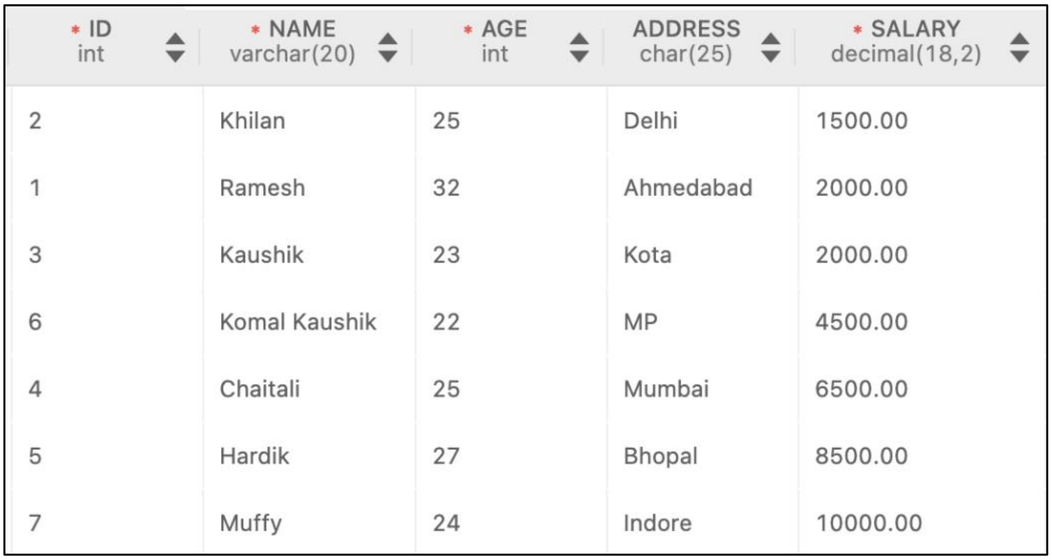
CS 457 LAB B

UMAIR DADA
20281 - SFBU

1. 3 Examples of Order and Group By

a. Order By: The `Order By` command is used for sorting the data in ascending or descending direction – Example: In an e-commerce store the price high to low / low to high can be via the order by ASC or DESC for price column.

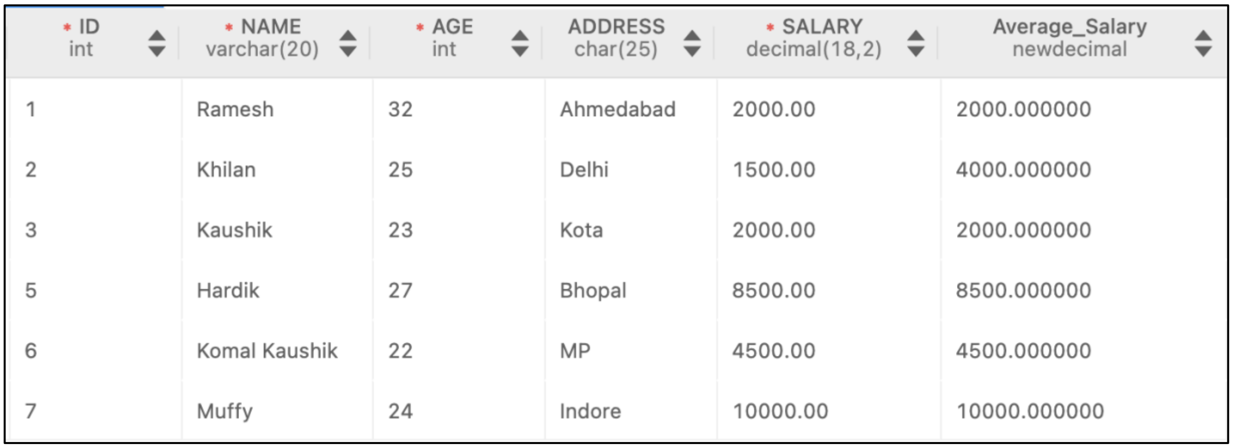
i. QUERY - SELECT * FROM CUSTOMERS ORDER BY `SALARY` ASC;



* ID int	* NAME varchar(20)	* AGE int	ADDRESS char(25)	* SALARY decimal(18,2)
2	Khilan	25	Delhi	1500.00
1	Ramesh	32	Ahmedabad	2000.00
3	Kaushik	23	Kota	2000.00
6	Komal Kaushik	22	MP	4500.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
7	Muffy	24	Indore	10000.00

b. Group By: This command is used for grouping same the value of column in a single column. This helps to find same values quicker – Example: To find how many people are the age of 30.

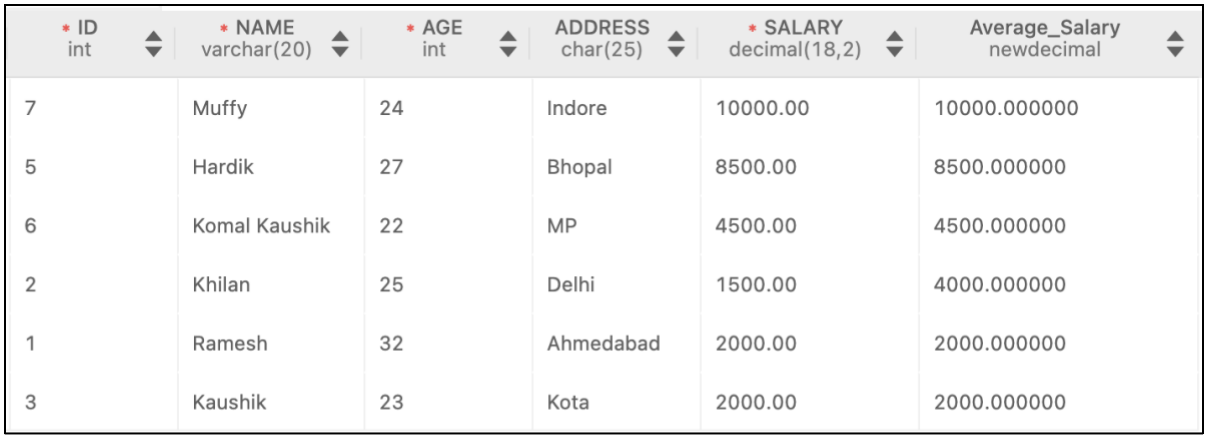
i. QUERY – SELECT *, AVG(SALARY) AS Average_Salary FROM CUSTOMERS GROUP BY AGE;



* ID int	* NAME varchar(20)	* AGE int	ADDRESS char(25)	* SALARY decimal(18,2)	Average_Salary newdecimal
1	Ramesh	32	Ahmedabad	2000.00	2000.000000
2	Khilan	25	Delhi	1500.00	4000.000000
3	Kaushik	23	Kota	2000.00	2000.000000
5	Hardik	27	Bhopal	8500.00	8500.000000
6	Komal Kaushik	22	MP	4500.00	4500.000000
7	Muffy	24	Indore	10000.00	10000.000000

c. Group By and Order By Combined: Using this can group the same values and order the those values or any other values in Ascending or Descending manner.

   i. QUERY - SELECT *, AVG(SALARY) AS Average_Salary FROM CUSTOMERS GROUP BY AGE ORDER BY Average_Salary DESC;

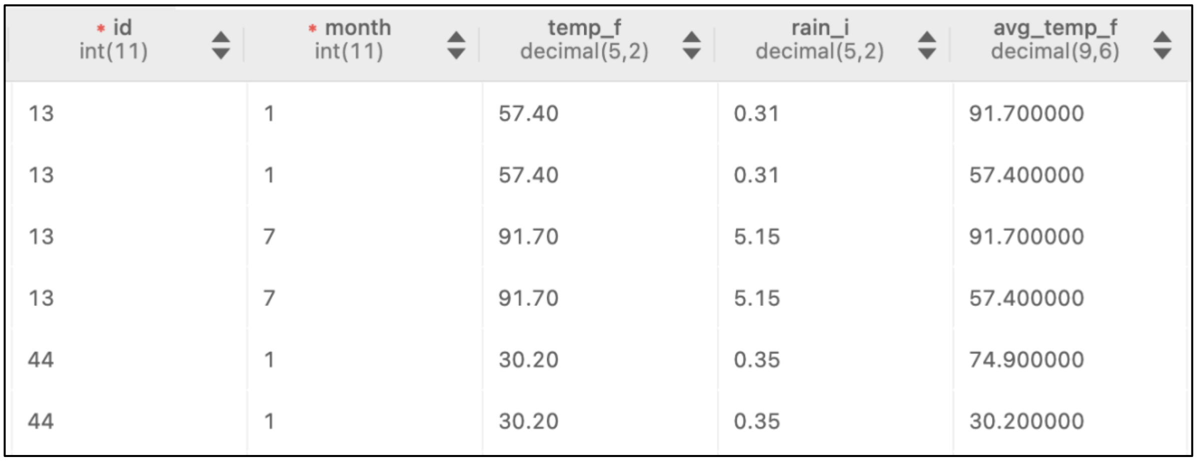


* ID int	* NAME varchar(20)	* AGE int	ADDRESS char(25)	* SALARY decimal(18,2)	Average_Salary newdecimal
7	Muffy	24	Indore	10000.00	10000.000000
5	Hardik	27	Bhopal	8500.00	8500.000000
6	Komal Kaushik	22	MP	4500.00	4500.000000
2	Khilan	25	Delhi	1500.00	4000.000000
1	Ramesh	32	Ahmedabad	2000.00	2000.000000
3	Kaushik	23	Kota	2000.00	2000.000000

2. Temp F

a. QUERY (No Table to be shown as it's a create command)- CREATE VIEW metric_stats AS SELECT s.id,s.month, s.temp_f, s.rain_i, avg_stats.avg_temp_f FROM stats s JOIN (SELECT id, AVG(temp_f) AS avg_temp_f FROM stats GROUP BY id, month) avg_stats ON s.id = avg_stats.id;

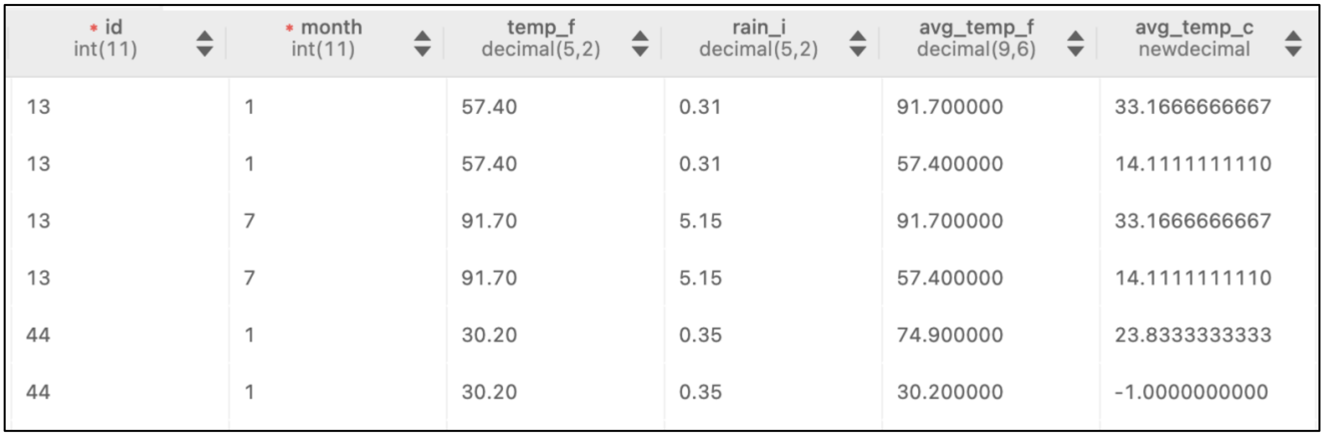
b. QUERY – SELECT * FROM metric_stats;



* id int(11)	* month int(11)	temp_f decimal(5,2)	rain_i decimal(5,2)	avg_temp_f decimal(9,6)
13	1	57.40	0.31	91.700000
13	1	57.40	0.31	57.400000
13	7	91.70	5.15	91.700000
13	7	91.70	5.15	57.400000
44	1	30.20	0.35	74.900000
44	1	30.20	0.35	30.200000

c. QUERY (As example this query will show the average temp in both Farenheit and Celsius) - SELECT * , (avg_temp_f - 32) * 5 / 9 AS avg_temp_c FROM metric_stats;

i.



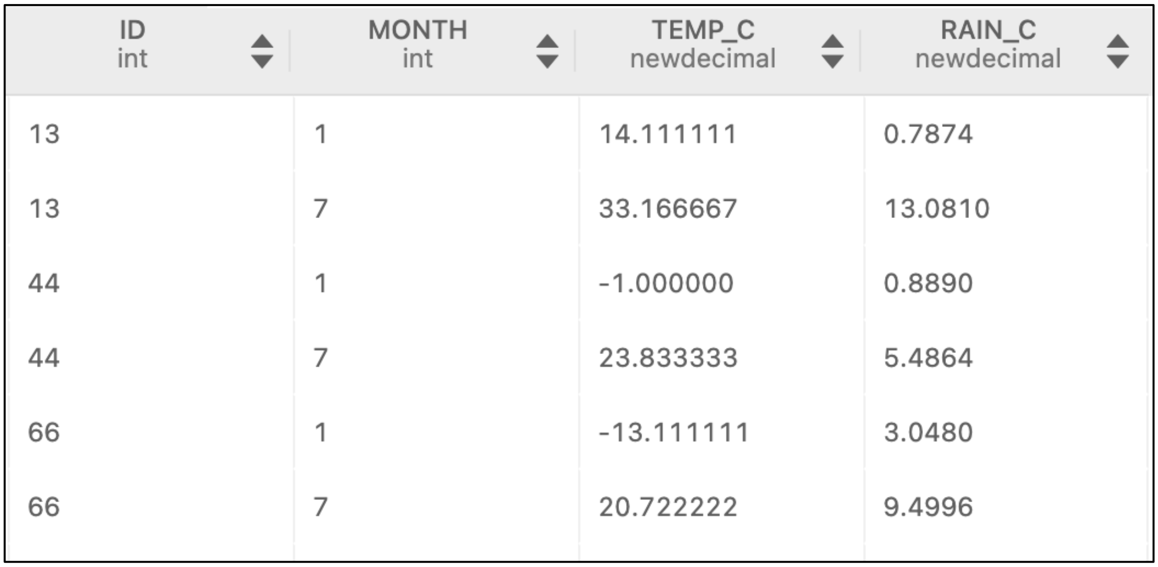
* id int(11)	* month int(11)	temp_f decimal(5,2)	rain_i decimal(5,2)	avg_temp_f decimal(9,6)	avg_temp_c newdecimal
13	1	57.40	0.31	91.700000	33.1666666667
13	1	57.40	0.31	57.400000	14.1111111110
13	7	91.70	5.15	91.700000	33.1666666667
13	7	91.70	5.15	57.400000	14.1111111110
44	1	30.20	0.35	74.900000	23.8333333333
44	1	30.20	0.35	30.200000	-1.0000000000

3. More Examples:

a. Rain conversion from inches to cm & temperature from farenheit to celsius.

i. QUERY - CREATE VIEW METRIC_STATS (ID, MONTH, TEMP_C, RAIN_C) AS SELECT ID, MONTH,(TEMP_F - 32) * 5 /9, RAIN_I * 2.54 FROM stats;

ii. QUERY – SELECT * FROM METRIC_STATS;

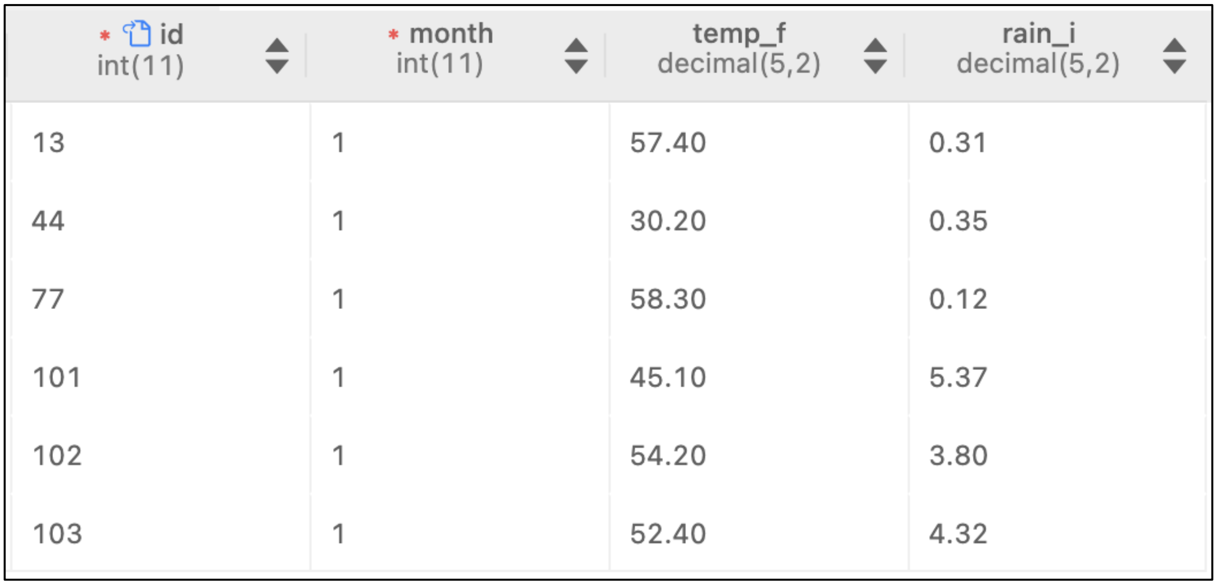


ID int	MONTH int	TEMP_C newdecimal	RAIN_C newdecimal
13	1	14.111111	0.7874
13	7	33.166667	13.0810
44	1	-1.000000	0.8890
44	7	23.833333	5.4864
66	1	-13.111111	3.0480
66	7	20.722222	9.4996

b. DELETE

i. QUERY (Created a savepoint sp1 to revert the output prior to this query) - DELETE FROM stats WHERE MONTH = 7 OR ID IN (SELECT ID FROM station WHERE LONG_W < 90);

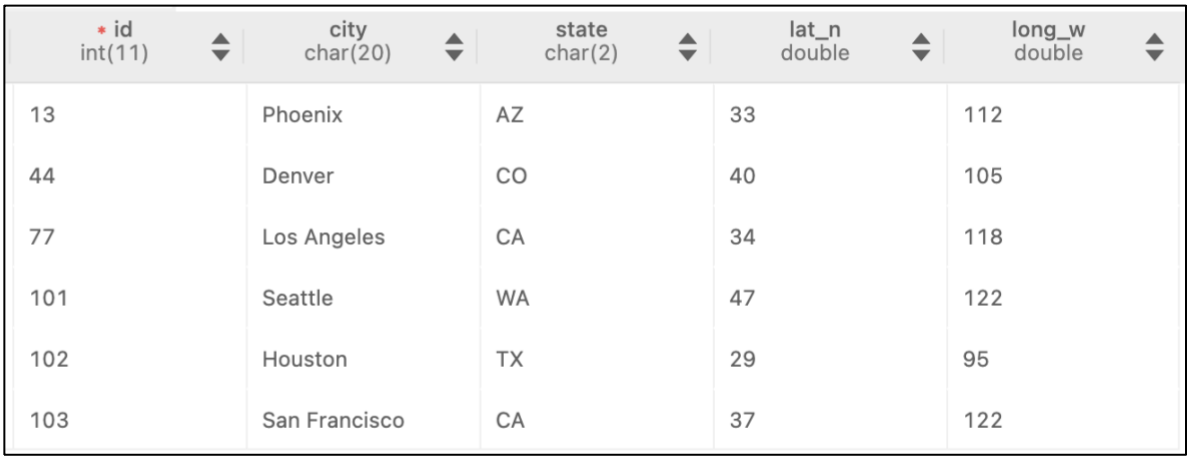
ii. QUERY - SELECT * FROM stats;



* id int(11)	* month int(11)	temp_f decimal(5,2)	rain_i decimal(5,2)
13	1	57.40	0.31
44	1	30.20	0.35
77	1	58.30	0.12
101	1	45.10	5.37
102	1	54.20	3.80
103	1	52.40	4.32

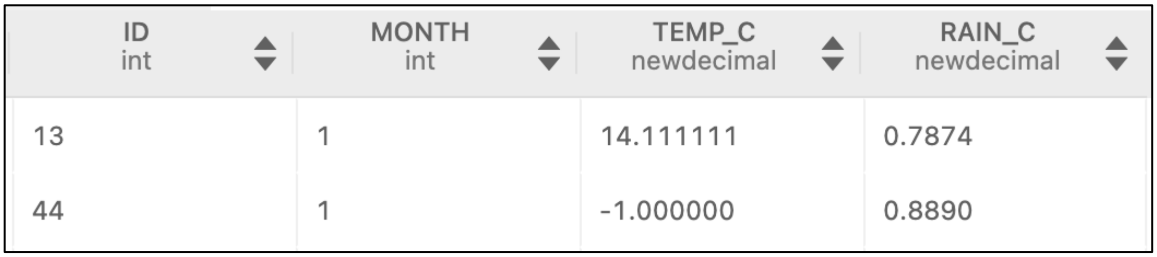
iii. QUERY (Created a savepoint sp2 to revert the output prior to this query) - DELETE FROM station WHERE LONG_W < 90;

iv. QUERY - SELECT * FROM station;



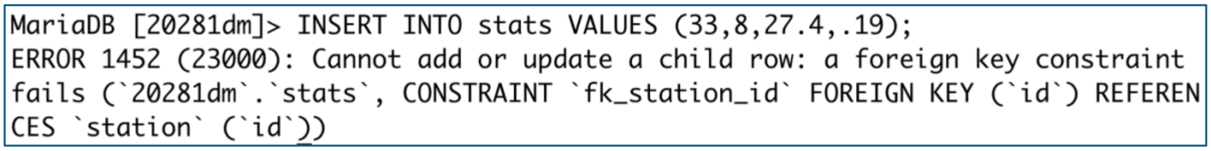
* id int(11)	city char(20)	state char(2)	lat_n double	long_w double
13	Phoenix	AZ	33	112
44	Denver	CO	40	105
77	Los Angeles	CA	34	118
101	Seattle	WA	47	122
102	Houston	TX	29	95
103	San Francisco	CA	37	122

v. QUERY - SELECT * FROM METRIC_STATS;



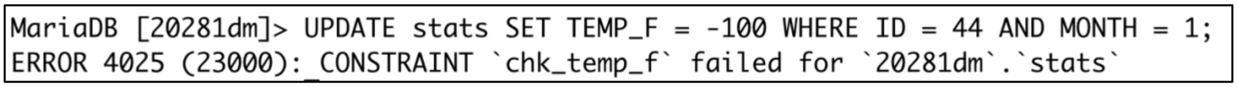
ID int	MONTH int	TEMP_C newdecimal	RAIN_C newdecimal
13	1	14.111111	0.7874
44	1	-1.000000	0.8890

vi. QUERY - INSERT INTO stats VALUES (33,8,27.4,.19);



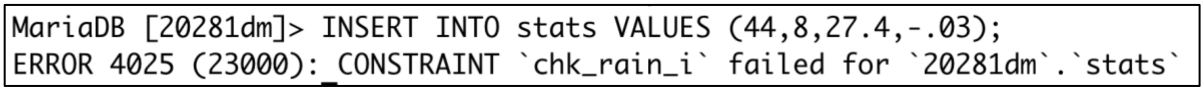
```
MariaDB [20281dm]> INSERT INTO stats VALUES (33,8,27.4,.19);
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint
fails (`20281dm`.`stats`, CONSTRAINT `fk_station_id` FOREIGN KEY (`id`) REFEREN
CES `station` (`id`))
```

vii. QUERY - UPDATE stats SET TEMP_F = -100 WHERE ID = 44 AND MONTH = 1;



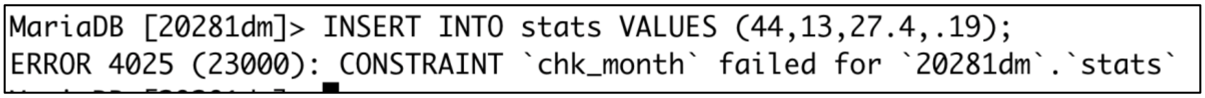
```
MariaDB [20281dm]> UPDATE stats SET TEMP_F = -100 WHERE ID = 44 AND MONTH = 1;
ERROR 4025 (23000): CONSTRAINT `chk_temp_f` failed for `20281dm`.`stats`
```

viii. QUERY - INSERT INTO stats VALUES (44,8,27.4,-.03);



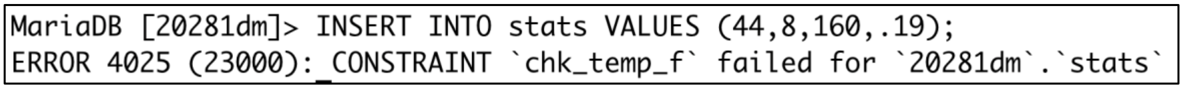
```
MariaDB [20281dm]> INSERT INTO stats VALUES (44,8,27.4,-.03);
ERROR 4025 (23000): CONSTRAINT `chk_rain_i` failed for `20281dm`.`stats`
```

ix. QUERY - INSERT INTO stats VALUES (44,13,27.4,.19);



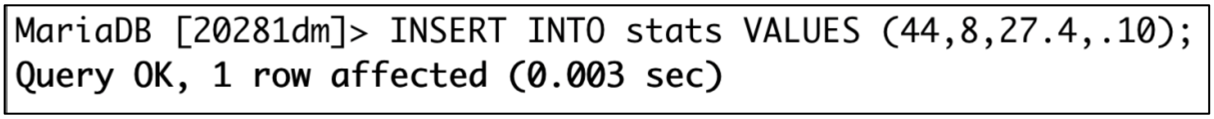
```
MariaDB [20281dm]> INSERT INTO stats VALUES (44,13,27.4,.19);
ERROR 4025 (23000): CONSTRAINT `chk_month` failed for `20281dm`.`stats`
```

x. QUERY - INSERT INTO stats VALUES (44,8,160,.19);



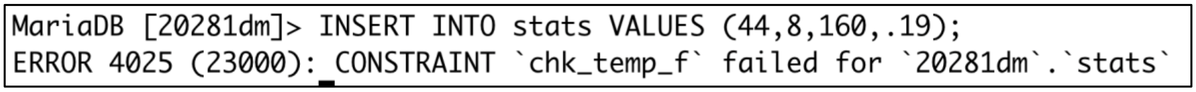
```
MariaDB [20281dm]> INSERT INTO stats VALUES (44,8,160,.19);
ERROR 4025 (23000): CONSTRAINT `chk_temp_f` failed for `20281dm`.`stats`
```

xi. QUERY - INSERT INTO stats VALUES (44,8,27.4,.10);



```
MariaDB [20281dm]> INSERT INTO stats VALUES (44,8,27.4,.10);
Query OK, 1 row affected (0.003 sec)
```

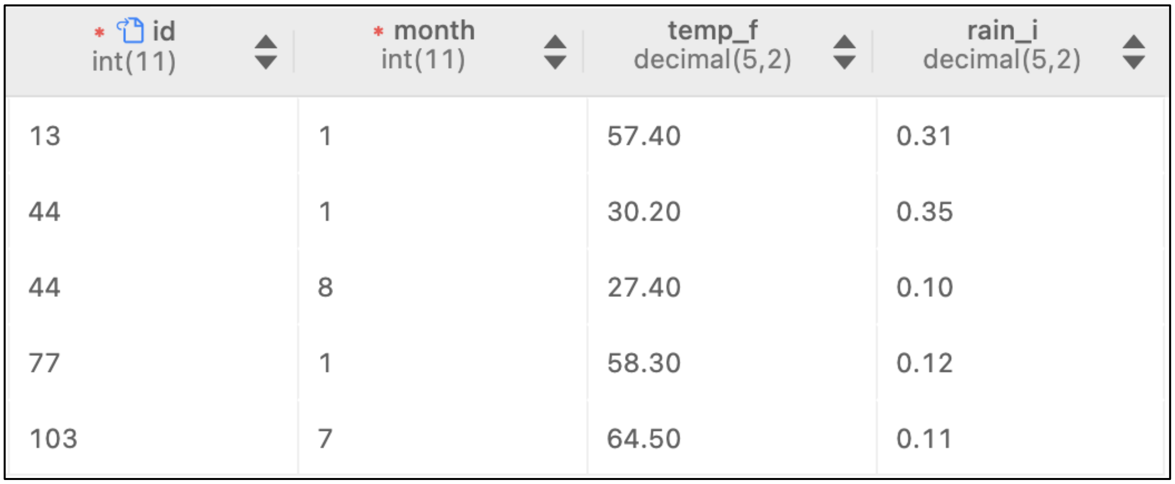
xii. QUERY - INSERT INTO stats VALUES (44,8,160,.19);



```
MariaDB [20281dm]> INSERT INTO stats VALUES (44,8,160,.19);
ERROR 4025 (23000): CONSTRAINT `chk_temp_f` failed for `20281dm`.`stats`
```

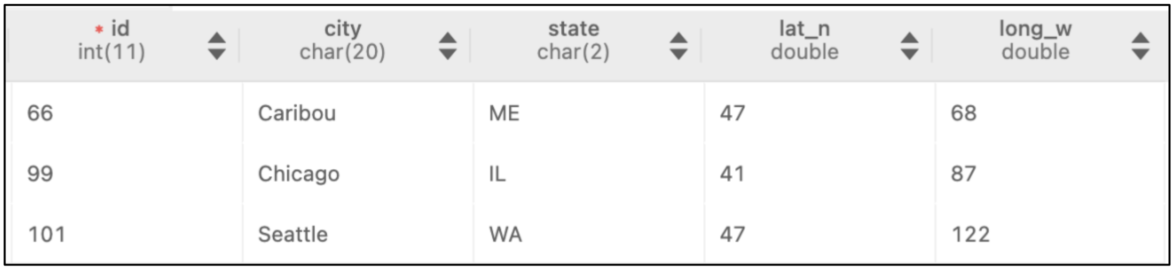
4. LAB 1A Question 3 Extra Examples.

a. QUERY - SELECT * FROM stats WHERE (temp_f > 80 OR rain_i < 1) AND temp_f < 70;



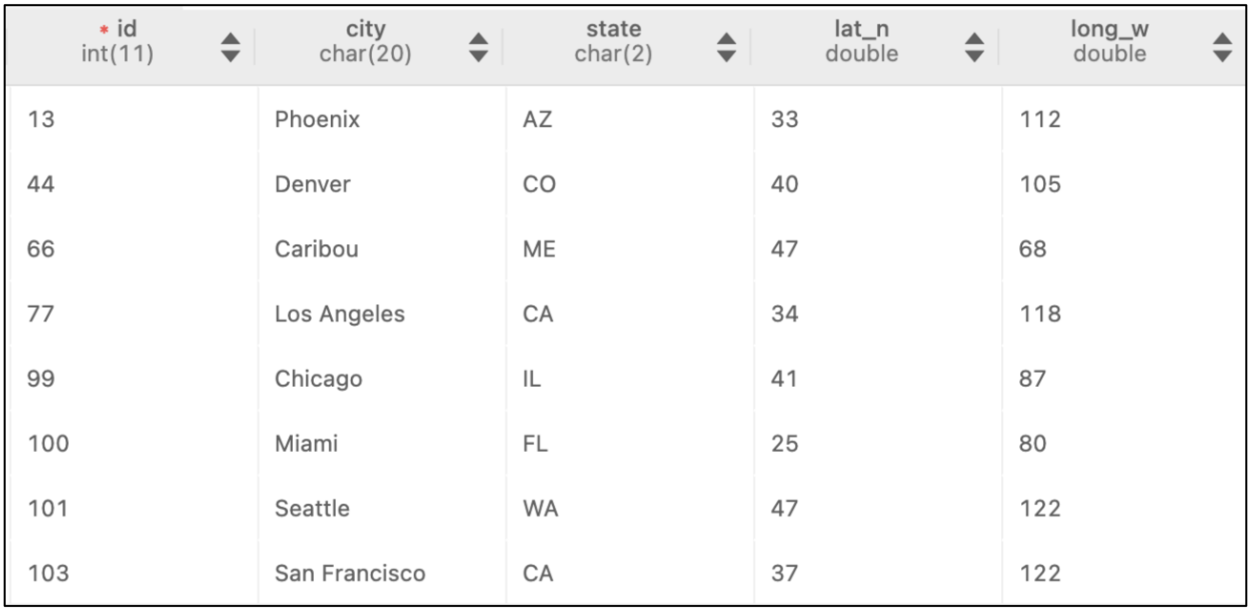
* id int(11)	* month int(11)	temp_f decimal(5,2)	rain_i decimal(5,2)
13	1	57.40	0.31
44	1	30.20	0.35
44	8	27.40	0.10
77	1	58.30	0.12
103	7	64.50	0.11

b. QUERY – SELECT * FROM station WHERE lat_n > 40 OR long_w < 100 AND state = 'CA';



* id int(11)	city char(20)	state char(2)	lat_n double	long_w double
66	Caribou	ME	47	68
99	Chicago	IL	41	87
101	Seattle	WA	47	122

c. QUERY - SELECT * FROM station WHERE NOT (city = 'New York' OR (lat_n < 30 AND long_w > 80));



* id int(11)	city char(20)	state char(2)	lat_n double	long_w double
13	Phoenix	AZ	33	112
44	Denver	CO	40	105
66	Caribou	ME	47	68
77	Los Angeles	CA	34	118
99	Chicago	IL	41	87
100	Miami	FL	25	80
101	Seattle	WA	47	122
103	San Francisco	CA	37	122