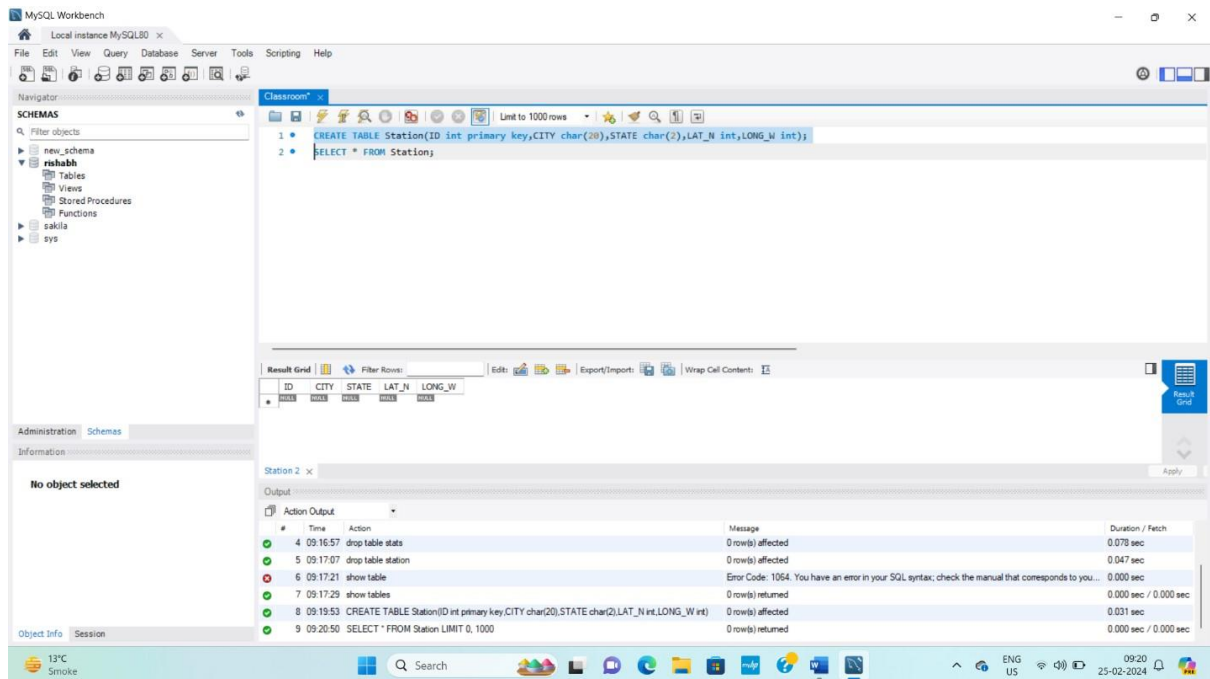


# Assignment Submission

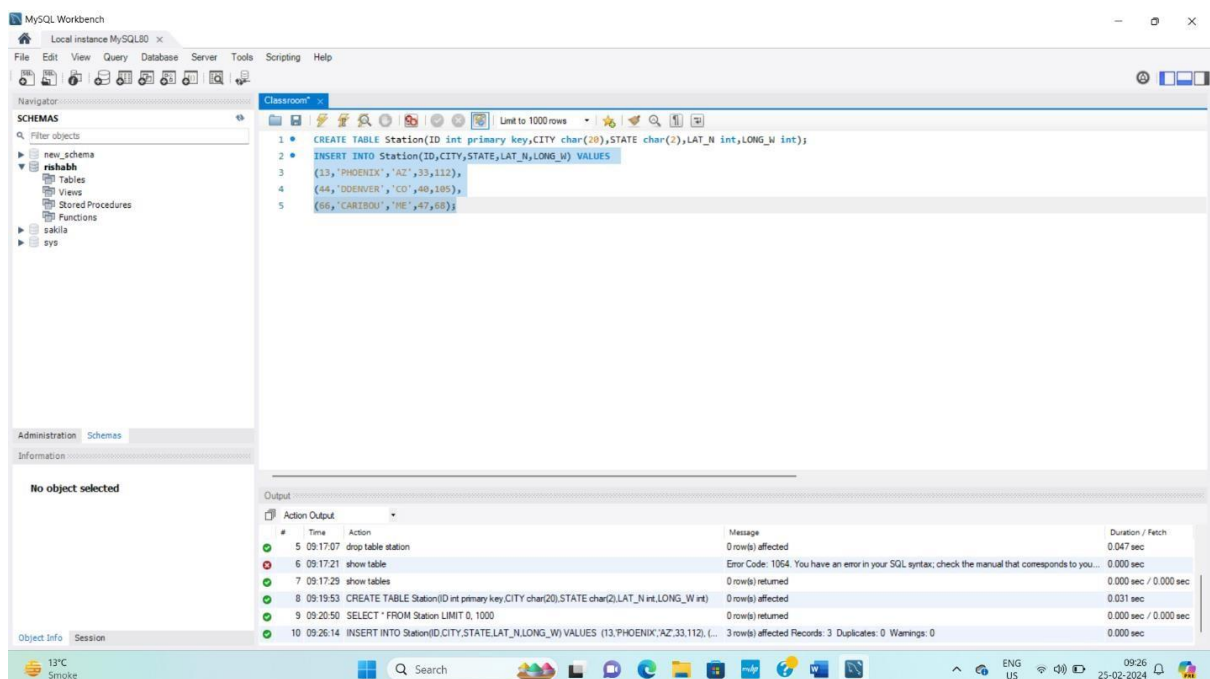
Name - Rishabh Saxena

Email - Errishabhsaxena24@gmail.com

Q1) Create a table "STATION" to store information about weather observation stations:



Q2) Insert the following records into the table:



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

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```

1 CREATE TABLE Station(ID int primary key,CITY char(20),STATE char(2),LAT_N int,LONG_W int);
2 INSERT INTO Station(ID,CITY,STATE,LAT_N,LONG_W) VALUES
3 (13,"PHOENIX","AZ",33,112),
4 (44,"DENVER","CO",40,105),
5 (66,"CARIBOU","ME",47,68);
6 SELECT * FROM Station;

```

The Result Grid shows the output of the query:

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

The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
6	09:17:21	show table	Error Code: 1054. You have an error in your SQL syntax; check the manual that corresponds to your...	0.000 sec
7	09:17:29	show tables	0 row(s) returned	0.000 sec / 0.000 sec
8	09:19:53	CREATE TABLE Station(ID int primary key,CITY char(20),STATE char(2),LAT_N int,LONG_W int)	0 row(s) affected	0.031 sec
9	09:20:50	SELECT * FROM Station LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
10	09:26:14	INSERT INTO Station(ID,CITY,STATE,LAT_N,LONG_W) VALUES (13,PHOENIX,AZ,33,112), (...)	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
11	09:27:26	SELECT * FROM Station LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

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

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following queries:

```

1 CREATE TABLE Station(ID int primary key,CITY char(20),STATE char(2),LAT_N int,LONG_W int);
2 INSERT INTO Station(ID,CITY,STATE,LAT_N,LONG_W) VALUES
3 (13,"PHOENIX","AZ",33,112),
4 (44,"DENVER","CO",40,105),
5 (66,"CARIBOU","ME",47,68);
6 SELECT * FROM Station;
7 SELECT * FROM Station
8 WHERE LAT_N > 39.7;

```

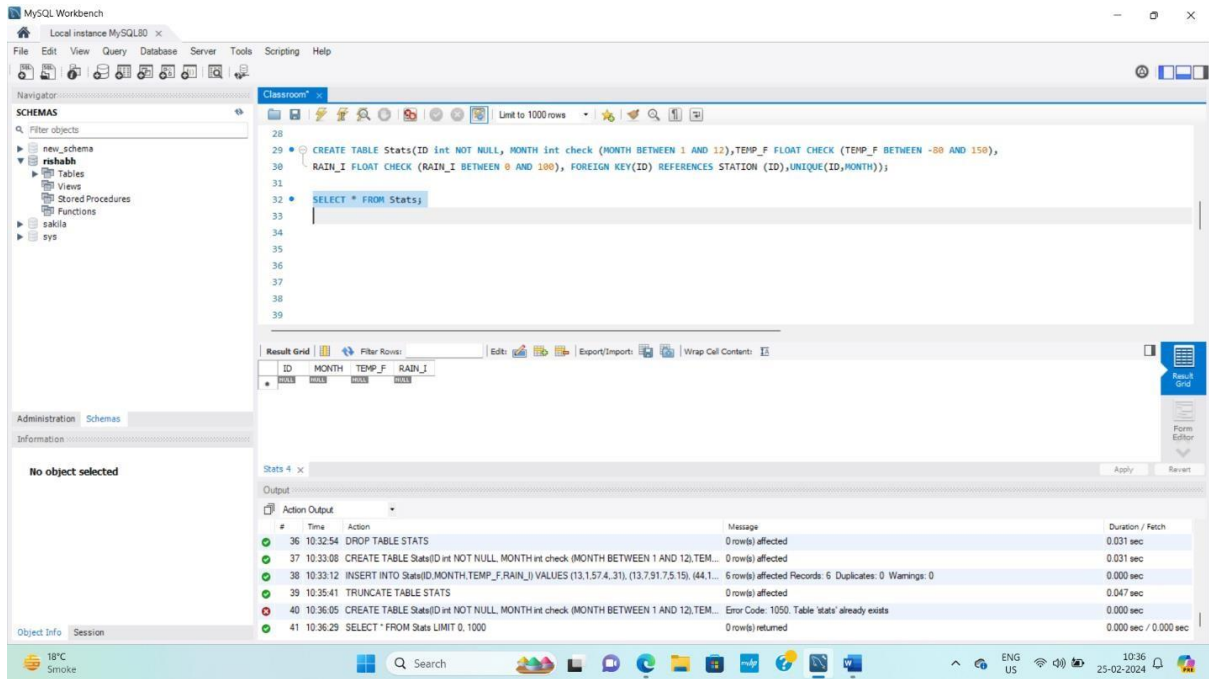
The Result Grid shows the output of the query:

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

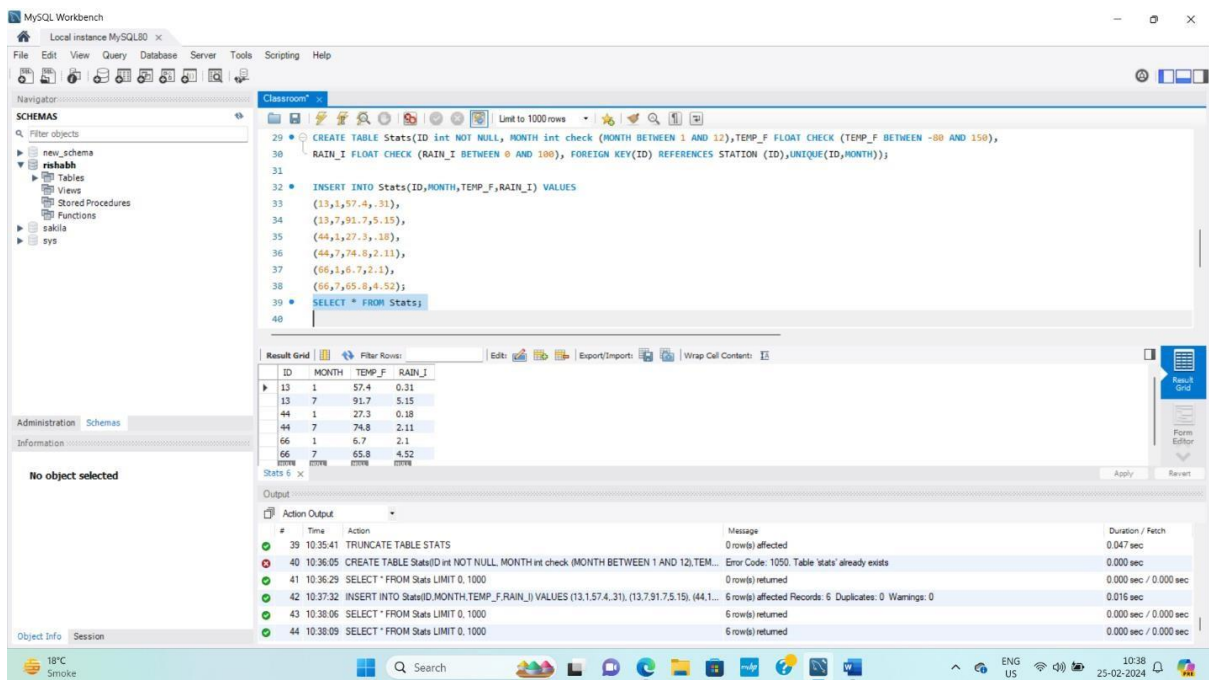
The Output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
7	09:17:29	show tables	0 row(s) returned	0.000 sec / 0.000 sec
8	09:19:53	CREATE TABLE Station(ID int primary key,CITY char(20),STATE char(2),LAT_N int,LONG_W int)	0 row(s) affected	0.031 sec
9	09:20:50	SELECT * FROM Station LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
10	09:26:14	INSERT INTO Station(ID,CITY,STATE,LAT_N,LONG_W) VALUES (13,PHOENIX,AZ,33,112), (...)	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
11	09:27:26	SELECT * FROM Station LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
12	09:29:17	SELECT * FROM Station WHERE LAT_N > 39.7 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

5. Create another table, '`STATS`', to store normalized temperature and precipitation data:  
There will be no Duplicate `ID` and `MONTH` combination.



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



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

MySQL Workbench interface showing a query execution. The query is:

```

SELECT TEMP_F, CITY FROM STATS
INNER JOIN STATION
ON STATION.ID = STATS.ID;

```

The result grid displays the following data:

TEMP_F	CITY
57.4	PHOENIX
91.7	PHOENIX
27.3	DENVER
74.8	DENVER
6.7	CARIBOU
65.8	CARIBOU

The output pane shows the execution log with the following actions:

- 40 10:36:05 CREATE TABLE Stats(ID INT NOT NULL, MONTH INT CHECK (MONTH BETWEEN 1 AND 12), TEMP\_F FLOAT, RAIN\_I FLOAT, CITY VARCHAR(255))
- 41 10:36:29 SELECT \* FROM Stats LIMIT 0, 1000
- 42 10:37:32 INSERT INTO Stats(ID, MONTH, TEMP\_F, RAIN\_I) VALUES (13, 1, 57.4, 31), (13, 7, 91.7, 5.15), (44, 1, 27.3, 0.31), (44, 7, 74.8, 2.11), (66, 1, 6.7, 2.1), (66, 7, 65.8, 4.52);
- 43 10:38:06 SELECT \* FROM Stats LIMIT 0, 1000
- 44 10:38:09 SELECT \* FROM Stats LIMIT 0, 1000
- 45 10:41:52 SELECT TEMP\_F, CITY FROM STATS INNER JOIN STATION ON STATION.ID = STATS.ID LIMIT 0, 1000

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.

MySQL Workbench interface showing a query execution. The query is:

```

SELECT MONTH, RAIN_I, CITY FROM STATS
INNER JOIN STATION
ON STATS.ID = STATION.ID
ORDER BY MONTH, RAIN_I DESC;

```

The result grid displays the following data:

MONTH	RAIN_I	CITY
1	2.1	CARIBOU
1	0.31	PHOENIX
1	0.38	DENVER
7	5.15	PHOENIX
7	4.52	CARIBOU
7	2.11	DENVER

The output pane shows the execution log with the following actions:

- 42 10:37:32 INSERT INTO Stats(ID, MONTH, TEMP\_F, RAIN\_I) VALUES (13, 1, 57.4, 31), (13, 7, 91.7, 5.15), (44, 1, 27.3, 0.31), (44, 7, 74.8, 2.11), (66, 1, 6.7, 2.1), (66, 7, 65.8, 4.52);
- 43 10:38:06 SELECT \* FROM Stats LIMIT 0, 1000
- 44 10:38:09 SELECT \* FROM Stats LIMIT 0, 1000
- 45 10:41:52 SELECT TEMP\_F, CITY FROM STATS INNER JOIN STATION ON STATION.ID = STATS.ID LIMIT 0, 1000
- 46 10:45:40 SELECT MONTH, RAIN\_I, CITY FROM STATS INNER JOIN STATION ON STATION.ID = STATS.ID ORDER BY MONTH, RAIN\_I DESC; LIMIT 0, 1000
- 47 10:46:14 SELECT MONTH, RAIN\_I, CITY FROM STATS INNER JOIN STATION ON STATION.ID = STATS.ID ORDER BY MONTH, RAIN\_I DESC; LIMIT 0, 1000

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

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- new\_schema
- rhshabb
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys

Administration Schemas

Information

No object selected

Object Info Session

Classroom

Limit to 1000 rows

```

49 SELECT MONTH,TEMP_F,CITY,LAT_N FROM STATION
50 INNER JOIN STATION
51 ON STATION.ID = STATION.ID
52 WHERE MONTH = 7
53 ORDER BY TEMP_F

```

Result Grid

MONTH	TEMP_F	CITY	LAT_N
7	65.8	CARIBOU	47
7	74.8	DOENVER	40
7	91.7	PHOENX	33

Result 10

Output

Action Output

#	Time	Action	Message	Duration / Fetch
43	10:38:06	SELECT * FROM Stats LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
44	10:38:09	SELECT * FROM Stats LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
45	10:41:52	SELECT TEMP_F,CITY FROM STATION INNER JOIN STATION ON STATION.ID = STATION.ID	6 row(s) returned	0.000 sec / 0.000 sec
46	10:45:40	SELECT MONTH,RAIN_I,CITY FROM STATION INNER JOIN STATION ON STATION.ID = STATION.ID	6 row(s) returned	0.016 sec / 0.000 sec
47	10:46:14	SELECT MONTH,RAIN_I,CITY FROM STATION INNER JOIN STATION ON STATION.ID = STATION.ID	6 row(s) returned	0.015 sec / 0.000 sec
48	10:51:05	SELECT MONTH,TEMP_F,CITY,LAT_N FROM STATION INNER JOIN STATION ON STATION.ID = STATION.ID	3 row(s) returned	0.000 sec / 0.000 sec

18°C Smoke

Search

ENG US

10:51

25-02-2024

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

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- new\_schema
- rhshabb
  - Tables
  - Views
  - Stored Procedures
  - Functions
- sakila
- sys

Administration Schemas

Information

No object selected

Object Info Session

Classroom

Limit to 1000 rows

```

49 SELECT CITY,MAX(TEMP_F),MIN(TEMP_F),
50 AVG(RAIN_I) FROM STATION
51 INNER JOIN STATION
52 ON STATION.ID = STATION.ID
53 GROUP BY CITY;

```

Result Grid

CITY	MAX(TEMP_F)	MIN(TEMP_F)	AVG(RAIN_I)
PHOENX	91.7	57.4	2.7300000488758087
DOENVER	74.8	27.3	1.1499999511241913
CARIBOU	65.8	6.7	3.309999942779541

Result 11

Output

Action Output

#	Time	Action	Message	Duration / Fetch
47	10:46:14	SELECT MONTH,RAIN_I,CITY FROM STATION INNER JOIN STATION ON STATION.ID = STATION.ID	6 row(s) returned	0.015 sec / 0.000 sec
48	10:51:05	SELECT MONTH,TEMP_F,CITY,LAT_N FROM STATION INNER JOIN STATION ON STATION.ID = STATION.ID	3 row(s) returned	0.000 sec / 0.000 sec
49	10:55:05	SELECT CITY,MAX(TEMP_F),MIN(TEMP_F),AVG(RAIN_I) FROM STATION INNER JOIN STATION ON STATION.ID = STATION.ID	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'MAX(TEMP_F),MIN(TEMP_F),AVG(RAIN_I) FROM STATION INNER JOIN STATION ON STATION.ID = STATION.ID' at line 1	0.000 sec
50	10:55:20	SELECT CITY,MAX(TEMP_F),MIN(TEMP_F),AVG(RAIN_I) FROM STATION INNER JOIN STATION ON STATION.ID = STATION.ID	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'MAX(TEMP_F),MIN(TEMP_F),AVG(RAIN_I) FROM STATION INNER JOIN STATION ON STATION.ID = STATION.ID' at line 1	0.000 sec
51	10:55:58	SELECT CITY,MAX(TEMP_F),MIN(TEMP_F),AVG(RAIN_I) FROM STATION INNER JOIN STATION ON STATION.ID = STATION.ID	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'MAX(TEMP_F),MIN(TEMP_F),AVG(RAIN_I) FROM STATION INNER JOIN STATION ON STATION.ID = STATION.ID' at line 1	0.000 sec
52	10:57:14	SELECT CITY,MAX(TEMP_F),MIN(TEMP_F),AVG(RAIN_I) FROM STATION INNER JOIN STATION ON STATION.ID = STATION.ID	3 row(s) returned	0.000 sec / 0.000 sec

19°C Sunny

Search

ENG US

10:57

25-02-2024

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



The screenshot shows the MySQL Workbench interface. The 'Classroom' schema is selected in the Navigator. A query is entered in the SQL editor:

```

SELECT CITY, (TEMP_F-32)*5/9 AS TEMP_F, RAIN_I*2.54 AS RAIN_I
FROM STATS
INNER JOIN STATION
ON STATION.ID = STATS.ID;

```

The 'Result Grid' shows the following data:

CITY	TEMP_F	RAIN_I
PHOENIX	14.11111958821615	0.787400060558319
PHOENIX	33.16666497124566	13.081000242233277
DENVER	-2.61111534966363	0.4572000181674957
DENVER	23.77779473198784	5.3593997335433965
CARIBOU	-14.05555661519369	5.333999757672235
CARIBOU	18.77779473198784	11.480799951553346

The 'Output' pane shows the execution log with messages for each row returned.

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

The screenshot shows the MySQL Workbench interface. The 'Classroom' schema is selected. A query is entered in the SQL editor:

```

UPDATE STATS
SET RAIN_I = RAIN_I - 0.01;
SET SQL_SAFE_UPDATES = 0;
SELECT * FROM STATS;

```

The 'Result Grid' shows the following data:

ID	MONTH	TEMP_F	RAIN_I
13	1	57.4	0.3
13	7	91.7	5.14
44	1	27.3	0.17
44	7	74.8	2.1
66	1	6.7	2.09
66	7	65.8	4.51

The 'Output' pane shows the execution log with messages for each row returned and the update operation.

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

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

Filter objects

SCHEMAS

- new\_schema
- rihabbb
  - Tables
    - station
    - stats
  - Views
  - Stored Procedures
  - Functions
  - sakila
  - sys

Administration Schemas

Information

No object selected

Object Info Session

Classroom\*

Limit to 1000 rows

```
75
76 select * from stats;
77
78
79 select city,temp_f,month from stats
80 inner join station
81 on stats.id = station.id
82 where month = 7 And city = 'Denver';
83
84 update stats
85 set temp_f = 74.9
86 where ID = 44 AND month = 7;
```

Result Grid

city	temp_f	month
Denver	74.9	7

Output

Action Output

#	Time	Action	Message	Duration / Fetch
82	15:10:53	update stats set temp_f = 74.9 where month = 7 And city = 'Denver'	Error Code: 1054. Unknown column 'city' in 'where clause'	0.000 sec
83	15:12:47	update stats set temp_f = 74.9 where ID = 44, month = 7 And city = 'Denver'	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to you...	0.000 sec
84	15:13:05	update stats set temp_f = 74.9 where ID = 44 AND month = 7	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
85	15:13:11	select city,temp_f,month from stats inner join station on stats.id = station.id where month = 7 And ci...	1 row(s) returned	0.000 sec / 0.000 sec
86	15:13:23	update stats set temp_f = 74.9 where ID = 44 AND month = 7	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
87	15:13:33	select city,temp_f,month from stats inner join station on stats.id = station.id where month = 7 And ci...	1 row(s) returned	0.016 sec / 0.000 sec

24°C Mostly cloudy

Search

ENG IN

15:15 25-02-2024