

# Assignment 4 - Normalization

Team name : errorFree

Group Members:

Ritik Bhandari(002738904)

Rishi Shelly(002771020)

Dhiral Mayavanshi(002706357 )

Table Name: New York Weather\_Nov22

The Table satisfies 1st Normal form which includes a primary key with minimal set of attributes.

The values in each column of a table are atomic and not multi-valued attributes.  
Lastly, two columns do not store similar information in the same table.

## SQL Views:

1. Creating and calling view for clothing preference of people in Boston

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema with tables like 'weather' and 'boston\_form\_response'. The central pane shows a SQL editor with the following code:

```
1  use weather;
2  #Creating view for clothing preference for people in Boston
3  ●  create view clothing_pref_of_people_in_boston as
4      select bw.avg_temperature, bw.avg_humidity,bw.avg_wind_speed,bf.clothing
5          from boston_weather_nov22 bw right join boston_form_response bf on
6          bw.avg_temperature = bf.avg_temperature;
7  ##Calling view for clothing preference for people in Boston
8  ●  select * from clothing_pref_of_people_in_boston;
```

The 'Result Grid' pane below the SQL editor displays the results of the query:

	avg_temperature	avg_humidity	avg_wind_speed	clothing
61	88.3	7.7		t-shirt
61	88.3	7.7		full-sleeves
54.1	67.4	6.6		full-sleeves
33.7	39.1	16.5		jackets
66.2	82.2	11.6		t-shirt
69.9	79	15.8		t-shirt
67.4	53.8	14.7		t-shirt
48.8	36.2	13.8		full-sleeves
43	59.9	8.3		jackets
44.4	44.4	14.9		full-sleeves

The bottom pane shows the 'Output' window with the following log entries:

Action	Time	Message	Duration / Fetch
Action	94 20:36:46	select bw.avg_temperature, bw.avg_humidity,bw.avg_wind_speed,bf.clothing from boston_weather_nov22 bw right join boston_form_response bf on bw.avg_temperature = bf.avg_temperature;	0.000 sec
Changes applied	95 20:37:18	CREATE VIEW `clothing_pref_of_people_in_boston` AS SELECT * FROM `weather.boston_weather_nov22` LIMIT 0, 1000	0.015 sec / 0.000 sec
Changes applied	96 20:37:39	ALTER TABLE `boston_form_response` ADD INDEX `avg_temperature`(`avg_temperature`);	
Changes applied	97 20:37:49	select bw.avg_temperature, bw.avg_humidity,bw.avg_wind_speed,bf.clothing from boston_weather_nov22 bw right join boston_form_response bf on bw.avg_temperature = bf.avg_temperature;	0.000 sec / 0.000 sec
Changes applied	98 20:37:52	CREATE VIEW `clothing_pref_of_people_in_boston` AS SELECT * FROM `weather.boston_weather_nov22` LIMIT 0, 1000	0.016 sec
Changes applied	99 20:37:56	SELECT * FROM `clothing_pref_of_people_in_boston` LIMIT 0, 1000	0.000 sec / 0.000 sec

2. Creating and calling View for which is the best city for generating Wind energy.

The screenshot shows the MySQL Workbench interface with the following details:

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Schemas:** Schemas pane showing databases: sv5, testing, tps, weather. The testing database is selected.
- Query Editor:** SQL tab containing the following code:

```

6   Join Boston_Form_Response bf
7   on bw.avg_temperature = bf.avg_temperature
8   where Max_Wind_Speed > '20' group by Max_Wind_Speed
9   UNION
10  select nw.Max_Wind_Speed, count(nw.Max_Wind_Speed), nf.city
11  from Newyork_Weather_Nov22 nw
12  Join Newyork_Form_Response nf
13  on nw.avg_temperature = nf.avg_temperature
14  where Max_Wind_Speed > '20' group by Max_Wind_Speed
15  UNION
16  select cw.Max_Wind_Speed, count(cw.Max_Wind_Speed), cf.city
17  from Colorado_Weather_Nov22 cw
18  Join Colorado_Form_Response cf
19  on cw.avg_temperature = cf.avg_temperature
20  where Max_Wind_Speed > '20' group by Max_Wind_Speed;
21
22  • select*from Best_city_for_generating_wind_energy;
23
24

```
- Result Grid:** Shows the results of the query:

Max_Wind_Speed	count(bw.Max_Wind_Speed)	city
23	4	Boston
26	4	Boston
25	5	Boston
21	2	Boston
29	1	Boston
- Action Output:** Shows the execution log:

#	Time	Action	Message	Duration / Fetch
4	20:01:57	select*from Best_city_for_generating_wind_energy LIMIT 0, 1000	21 row(s) returned	0.546 sec / 0.000 sec
5	20:50:46	select*from Best_city_for_generating_wind_energy LIMIT 0, 1000	21 row(s) returned	0.062 sec / 0.000 sec
- System Status:** Shows system status: 32°F Clear, ENG, 20:51, 13-12-2022.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

testing

Tables

- boston\_form\_response
- boston\_weather\_nov22
- colorado\_form\_response
- colorado\_weather\_nov22
- newyork\_form\_response
- newyork\_weather\_nov22

Views

Stored Procedures

tp

weather

Administration Schemas

Information

Schema: testing

```

abc* Limit to 1000 rows SQLAdditions
6 Join Boston_Form_Response bf
7 on bw.avg_temperature = bf.avg_temperature
8 where Max_Wind_Speed > '20' group by Max_Wind_Speed
9 UNION
10 select nw.Max_Wind_Speed, count(nw.Max_Wind_Speed),nf.city
11 from Newyork_Weather_Nov22 nw
12 join Newyork_Form_Response nf
13 on nw.avg_temperature = nf.avg_temperature
14 where Max_Wind_Speed > '20' group by Max_Wind_Speed
15 UNION
16 select cw.Max_Wind_Speed, count(cw.Max_Wind_Speed),cf.city
17 from Colorado_Weather_Nov22 cw
18 join Colorado_Form_Response cf
19 on cw.avg_temperature = cf.avg_temperature
20 where Max_Wind_Speed > '20' group by Max_Wind_Speed;
21
22 • select*from Best_city_for_generating_wind_energy;
23
24

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Max_Wind_Speed	count(bw.Max_Wind_Speed)	city
23	4	Boston
26	5	Boston
25	2	Boston
29	1	Boston

Best\_city\_for\_generating\_wind\_... x

Output

Action Output

- 4 20:01:57 select\*from Best\_city\_for\_generating\_wind\_energy LIMIT 0, 1000 21 row(s) returned 0.546 sec / 0.000 sec
- 5 20:50:46 select\*from Best\_city\_for\_generating\_wind\_energy LIMIT 0, 1000 21 row(s) returned 0.062 sec / 0.000 sec

Object Info Session

Type here to search

32°F Clear ENG 20:51 13-12-2022

### 3. Creating and calling View for rate of illness in New York due to low temperature and Dew Point

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

weather

Tables

- boston\_form\_response
- boston\_weather\_nov22
- colorado\_form\_response
- colorado\_weather\_nov22
- newyork\_form\_response
- newyork\_weather\_nov22

Columns

- Date\_Now
- Max\_Temperature
- Avg\_Temperature
- Min\_Temperature
- Max\_Dew\_Point
- Avg\_Dew\_Point
- Min\_Dew\_Point
- Avg\_Humidity
- Min\_Humidity
- Min\_Humid\_Pct

Administration Schemas

Information

Table: newyork\_weather\_nov22

Columns:

Date_Now	int
Max_Temperature	int
Avg_Temperature	double
Min_Temperature	int
Max_Dew_Point	int
Avg_Dew_Point	double
Min_Dew_Point	double
Avg_Humidity	float
Min_Humidity	float
Min_Humid_Pct	int

Query 1 | boston\_weather\_nov22 - Table newyork\_weather\_nov22 - Table

```

abc* Limit to 1000 rows SQLAdditions
1 use weather;
2 #Creating view for rate of illness in NewYork due to low temperature and dewpoint
3 • create view rate_of_illness_in_NewYork_due_to_low_temperature_and_dew_points as
4 select nw.Min_Temperature, nw.Avg_Dew_Point, nf.frequency_illness from newyork_weather_nov22 nw
5 right join newyork_form_response nf
6 on nw.avg_temperature = nf.avg_temperature;
7 ##Calling view for clothing preference for people in Boston
8 • select * from rate_of_illness_in_NewYork_due_to_low_temperature_and_dew_points;
9
10

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Min_Temperature	Avg_Dew_Point	frequency_illness
55.8	normal	
57	46.2	normal
54	46.9	feverish
53	53.2	normal
63	59	normal
63	60.6	feverish
56	45.3	normal
47	25	normal
41	29.5	feverish
40	45.6	feverish

rate\_of\_illness\_in\_NewYork\_du... x

Output

Action Output

- 109 21:03:29 selected nw.Min\_Temperature, nw.Avg\_Dew\_Point, nf.frequency\_illness from newyork\_weather\_nov22 nw right.. Error Code: 1054. Unknown column 'nw.Avg\_Dew\_Point' in field list' 0.000 sec
- 110 21:04:34 Apply changes to newyork\_weather\_nov22 Changes applied
- 111 21:05:21 Apply changes to newyork\_weather\_nov22 Changes applied
- 112 21:05:40 create view rate\_of\_illness\_in\_NewYork\_due\_to\_low\_temperature\_and\_dew\_point as select nw.Min\_Temper.. Error Code: 1050. Table 'rate\_of\_illness\_in\_NewYork\_due\_to\_low\_temperature\_and\_dew\_point' already exists 0.000 sec
- 113 21:06:15 create view rate\_of\_illness\_in\_NewYork\_due\_to\_low\_temperature\_and\_dew\_points as select nw.Min\_Temper.. 0 row(s) affected 0.000 sec
- 114 21:06:16 select \*from rate\_of\_illness\_in\_NewYork\_due\_to\_low\_temperature\_and\_dew\_points LIMIT 0, 1000 44 row(s) returned 0.000 sec / 0.000 sec

Object Info Session

32°F Clear ENG IN 9:06 PM 12/13/2022

4. Creating and calling View for Heater requirement by the people of Boston according to its weather conditions

The screenshot shows the MySQL Workbench interface. In the top query editor, a script is being run to create a view named 'heater\_requirement\_by\_people\_of\_boston'. The script includes joins between 'boston\_weather\_nov22' and 'newyork\_form\_response' tables based on average temperature. The resulting view is then queried to show data.

```

use weather;
#Creating view for heater requirement by people of boston according to weather
create view heater_requirement_by_people_of_boston as
select * from boston_weather_nov22 bw
right join newyork_form_response bf
on bw.avg_temperature = bf.avg_temperature;
##Creating view for heater requirement by people of boston according to weather
select * from heater_requirement_by_people_of_boston ;

```

The result grid displays the following data:

avg_temperature	Total_Precipitation	Avg_Wind_Speed	heater_req
61	0.04	7.7	no
61	0.04	7.7	yes
54.1	0	6.6	no
33.7	0	16.5	yes
66.2	0	11.6	no
69.9	0	15.8	no
67.4	0	14.7	yes
48.8	0	13.8	yes
43	0	8.3	yes
43	0	14.8	yes

The output pane shows the history of actions taken:

#	Time	Action	Message	Duration / Fetch
111	21:05:21	Apply changes to newyork_weather_nov22	Changes applied	0.000 sec
112	21:05:40	create view rate_of_illness_in_NewYork_due_to_low_temperature_and_dew_point as select nw.Max_Temperature, nw.Total_Precipitation, nw.Avg_Wind_Speed, nw.Min_Humidity, nw.Max_Humidity, nw.Avg_Temperature, nw.Date_Now, nw.Max_Wind_Points, nw.Mean_Humidity, nw.Mean_Temperature, nw.Mean_Wind_Points	Error Code: 1050. Table 'rate_of_illness_in_NewYork_due_to_low_temperature_and_dew_point' already exists	0.000 sec
113	21:06:16	create view rate_of_illness_in_NewYork_due_to_low_temperature_and_dew_points as select nw.Max_Temperature, nw.Total_Precipitation, nw.Avg_Wind_Speed, nw.Min_Humidity, nw.Max_Humidity, nw.Avg_Temperature, nw.Date_Now, nw.Max_Wind_Points, nw.Mean_Humidity, nw.Mean_Temperature, nw.Mean_Wind_Points	0 row(s) affected	0.000 sec
114	21:06:16	select *from rate_of_illness_in_NewYork_due_to_low_temperature_and_dew_points LIMIT 0, 1000	44 row(s) returned	0.000 sec / 0.000 sec
115	21:13:03	create view heater_requirement_by_people_of_boston as select bw.avg_temperature, bw.Total_Precipitation, bw.Avg_Wind_Speed, bw.heater_req from boston_weather_nov22 bw right join newyork_form_response bf on bw.avg_temperature = bf.avg_temperature;	0 row(s) affected	0.016 sec
116	21:13:03	select * from heater_requirement_by_people_of_boston LIMIT 0, 1000	39 row(s) returned	0.000 sec / 0.000 sec

5. Creating and calling view for public opinion about the weather conditions in New York

The screenshot shows the MySQL Workbench interface. A script is being run to create a view named 'public\_opinion\_about\_weather\_conditions\_in\_NewYork'. The view selects data from multiple tables including 'newyork\_form\_response' and 'boston\_weather\_nov22'. The resulting view is then queried to show data.

```

use weather;
#Creating view for public opinion about weather conditions in NewYork
create view public_opinion_about_weather_conditions_in_NewYork as
select nw.Max_Temperature, nw.Max_Humidity, nw.Max_Wind_Speed, nw.public_opinion
from newyork_form_response nf
join boston_weather_nov22 bw
on nw.avg_temperature = bw.avg_temperature;
##Calling view for public opinion about weather conditions in NewYork
select * from public_opinion_about_weather_conditions_in_NewYork ;

```

The result grid displays the following data:

Max_Temperature	Max_Humidity	Max_Wind_Speed	public_opinion
67	90	15	warm
71	78	14	warm
67	81	10	warm
71	97	13	warm
75	84	18	warm
73	87	12	warm
77	84	21	warm
58	50	17	moderate
53	68	13	warm
67	76	10	warm
68	93	21	warm
69	94	21	warm
61	77	23	moderate
46	57	23	cold

The output pane shows the history of actions taken:

#	Time	Action	Message	Duration / Fetch
116	21:13:03	select * from heater_requirement_by_people_of_boston LIMIT 0, 1000	39 row(s) returned	0.000 sec / 0.000 sec
117	21:16:31	create view public_opinion_about_weather_conditions_in_NewYork as select nw.Max_Temperature, nw.Max_Humidity, nw.Max_Wind_Speed, nw.public_opinion from newyork_form_response nf join boston_weather_nov22 bw on nw.avg_temperature = bw.avg_temperature;	Error Code: 1054. Unknown column 'nw.Max_Temperature' in 'field list'	0.000 sec

## 6. Creating and Calling View for Heater, gas and electricity usage/requirement by the people Of Colorado

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** sys, weather, colorado\_weather\_nov22, colorado\_weather\_nov22.
- Query Editor (Query 1):**

```

1 use weather;
2 #Creating view for Heater, gas and electricity usage/requirement by the people of Colorado
3 create view Heater_gas_electricity_usage_by_the_people_of_Colorado as
4 select cw.Avg_Temperature, cw.Avg_Dew_Point, cw.Max_Wind_Speed, cf.heater_req, cf.gas_electricity_usage
5 from colorado_weather_nov22 cw
6 join colorado_form_response cf
7 on cw.avg_temperature = cf.avg_temperature;
8
9 #Calling view for Heater, gas and electricity usage/requirement by the people of Colorado

```
- Result Grid:**

Avg_Temperature	Avg_Dew_Point	Max_Wind_Speed	heater_req	gas_electricity_usage
44.8	17.8	18.4	no	low
48.3	13.8	24.2	no	low
28.6	18.8	11.5	yes	low
24	13	28.8	yes	high
36.2	23.2	35.7	yes	low
32.4	21.4	13.8	yes	high
33.9	23.2	11.5	yes	high
36.3	20	13.8	yes	high
37	16.8	24.2	yes	high
27.4	9.7	32.2	yes	high
21.2	7.2	16.1	yes	high
27.8	4.2	10.4	yes	high
- Output:**

#	Time	Action	Message	Duration / Fetch
136	21:35:38	Apply changes to colorado_weather_nov22	Changes applied	0.000 sec
137	21:35:53	use weather	0 row(s) affected	0.000 sec
138	21:35:53	create view Heater_gas_electricity_usage_by_the_people_of_Colorado as select cw.Avg_Temperature, cw...	0 row(s) affected	0.000 sec
139	21:35:53	select * from Heater_gas_electricity_usage_by_the_people_of_Colorado LIMIT 0, 1000	39 row(s) returned	0.000 sec / 0.000 sec

## 7. Creating and Calling View for Colorado's public opinion and rate of illness during different weather conditions

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** sys, weather, colorado\_weather\_nov22, colorado\_weather\_nov22.
- Query Editor (Query 1):**

```

1 use weather;
2 #Creating view for Colorado's public opinion and rate of illness during different weather conditions
3 create view Colorado_public_opinion_and_rate_of_illness_different_weathers as
4 select cw.Min_Temperature, cw.Min_Humidity, cf.public_opinion, cf.frequency_illness
5 from colorado_weather_nov22 cw
6 join colorado_form_response cf
7 on cw.avg_temperature = cf.avg_temperature;
8
9 #Calling view Colorado's public opinion and rate of illness during different weather conditions
10 select * from Colorado_public_opinion_and_rate_of_illness_different_weathers ;
11

```
- Result Grid:**

Min_Temperature	Min_Humidity	public_opinion	frequency_illness
27.8	17	moderate	normal
26.9	3	moderate	feverish
19.7	33	cold	extreme cough and cold
9.4	39	cold	extreme cough and cold
28.7	47	cold	extreme cough and cold
23.7	50	cold	extreme cough and cold
19.2	30	cold	extreme cough and cold
22.3	19	cold	extreme cough and cold
19.8	9	cold	extreme cough and cold
27	27	mid	extreme cough and cold
- Output:**

#	Time	Action	Message	Duration / Fetch
145	21:42:24	selected * from Heater_gas_electricity_usage_by_the_people_of_Colorado LIMIT 0, 1000	39 row(s) returned	0.000 sec / 0.000 sec
146	21:42:52	create view Colorado_public_opinion_and_rate_of_illness_different_weather as select cw.Min_Temperature...	Error Code: 1050. Table 'Colorado_public_opinion_and_rate_of_illness_different_weather' already exists	0.000 sec
147	21:43:09	create view Colorado_public_opinion_and_rate_of_illness_different_weathers as select cw.Min_Temperatur...	0 row(s) affected	0.000 sec
148	21:43:09	select * from Colorado_public_opinion_and_rate_of_illness_different_weathers LIMIT 0, 1000	39 row(s) returned	0.016 sec / 0.000 sec

## 8. Creating and Calling View for Heater requirement with Average Temperature for people of Boston

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The current database is 'weather'.
- Tables:** The 'weather' schema contains tables like 'boston\_form\_response', 'colorado\_form\_response', 'colorado\_weather\_nov22', 'newyork\_form\_response', and 'newyork\_weather\_nov22'.
- Query Editor:** The query being run is:
 

```

1 use weather;
2 #Creating view
3 create view self_join as
4 select boston_form_response.heater_req,boston_weather_nov22.avg_temperature
5 from boston_form_response,boston_weather_nov22 where boston_form_response.avg_temperature > boston_weather_nov22.avg_temperature;
6
7
8
9 * select * from self_join;
10
11
      
```
- Result Grid:** The results show a single column 'avg\_temperature' with values ranging from 39.4 to 50.4.
- Output:** The log shows the creation of the view and the execution of the select statement.

## 9. Creating and Calling View for electricity usage in boston according average temperature

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The current database is 'weather'.
- Tables:** The 'weather' schema contains tables like 'boston\_form\_response', 'colorado\_form\_response', 'colorado\_weather\_nov22', 'newyork\_form\_response', and 'newyork\_weather\_nov22'.
- Query Editor:** The query being run is:
 

```

1 use weather;
2 #Creating view
3 create view Boston_electricity_usage_avg_temp as
4 select boston_form_response.gas_electricity_usage,boston_weather_nov22.avg_temperature
5 from boston_form_response,boston_weather_nov22 where boston_form_response.avg_temperature > boston_weather_nov22.avg_temperature;
6
7
8
9 * select * from Boston_electricity_usage_avg_temp;
10
11
      
```
- Result Grid:** The results show a single column 'avg\_temperature' with values ranging from 39.4 to 50.4.
- Output:** The log shows the creation of the view and the execution of the select statement.

## 10. Creating and Calling View for what is the maximum and minimum humidity in New York during the month of Nov 22

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the database structure under the schema `weather`, including tables like `boston_form_response`, `boston_weather_nov22`, `colorado_form_response`, `colorado_weather_nov22`, `newyork_form_response`, and `newyork_weather_nov22`.
- Query Editor (Query 1):**

```

1 use weather;
2 #Creating view for What is the Maximum and Minimum Humidity in NewYork during the month of Nov 22
3 create view Maximum_and_Minimum_Humidity_in_NewYork as
4 select max(Avg_Humidity) as Max_Avg_Humidity, min(Avg_Humidity) as Minimum_Avg_Humidity
From NewYork_Weather_Nov22;
6
8
9 #Calling view What is the Maximum and Minimum Humidity in NewYork during the month of Nov 22
10 * select * from Maximum_and_Minimum_Humidity_in_NewYork ;
11
    
```
- Result Grid:** Displays the results of the query:

Max_Avg_Humidity	Minimum_Avg_Humidity
87.3	35.5
- Output:** Shows the history of actions:

#	Time	Action	Message	Duration / Fetch
180	22:22:07	Apply changes to newyork_weather_nov22	No changes detected	0.016 sec
181	22:23:12	create view Maximum_and_Minimum_Humidity_in_NewYork as select max(Avg_Humidity) as Max... Error Code: 1060. Duplicate column name 'Maximum_Avg_Humidity'	1 row(s) returned	0.000 sec / 0.000 sec
182	22:23:50	select max(Avg_Humidity) as Max_Avg_Humidity, min(Avg_Humidity) as Minimum_Avg_Humidity Fr... Error Code: 1146. Table 'weather.maximum_and_minimum_humidity_in_newyork' doesn't exist		0.000 sec
183	22:23:50	select * from Maximum_and_Minimum_Humidity_in_NewYork LIMIT 0, 1000		
- System Bar:** Shows the system status including temperature (31°F), battery level, signal strength, and system time (10:24 PM, 12/13/2022).

## 11. Creating and calling view which is the best city to visit in the month of November

The screenshot shows the MySQL Workbench interface. In the top-left pane, the 'Schemas' tree shows the 'testing' database selected. In the main query editor window, the SQL code for creating the view 'BestCity\_ToVisit' is pasted:

```

CREATE view BestCity_ToVisit as
Select city, count(public_opinion), public_opinion, frequency_illness
from newyork_form_response
group by public_opinion
union
Select city, count(public_opinion), public_opinion, frequency_illness
from colorado_form_response
group by public_opinion;
select * from BestCity_ToVisit;
    
```

The 'Result Grid' tab shows the results of the query:

city	count(public_opinion)	public_opinion	frequency_illness
NewYork	16	warm	normal
NewYork	15	moderate	normal
NewYork	13	cold	normal
Colorado	6	moderate	normal
Colorado	33	cold	extreme cough and cold

In the bottom right pane, the 'Output' tab displays the log of actions taken:

- Action 83: CREATE view BestCity\_ToVisit as Select city, count(public\_opinion), public\_opinion, frequency\_illness from ... Duration / Fetch 0.468 sec
- Action 84: select \* from BestCity\_ToVisit LIMIT 0, 1000 Duration / Fetch 0.000 sec / 0.000 sec

## 11. Creating and Calling View for how many times the maximum temperature was greater than 70F.

The screenshot shows the MySQL Workbench interface. In the top-left pane, the 'Schemas' tree shows the 'weather' database selected. In the main query editor window, the SQL code for creating the view 'times\_Maximum\_Temperature\_was\_greater\_than\_70F' is pasted:

```

use weather;
#Creating view for How many times Maximum Temperature was greater than 70F in Boston
create view times_Maximum_Temperature_was_greater_than_70F as
select count(Max_Temperature) from Boston_Weather_Nov22
where Max_Temperature > 70;

#Calling view How many times Maximum Temperature was greater than 70F in Boston
select * from times_Maximum_Temperature_was_greater_than_70F;
    
```

The 'Result Grid' tab shows the results of the query:

count(Max_Temperature)
5

In the bottom right pane, the 'Output' tab displays the log of actions taken:

- Action 182: select max(Avg\_Humidity) as Max\_Avg\_Humidity, min(Avg\_Humidity) as Minimum\_Avg\_Humidity From ... Duration / Fetch 0.000 sec / 0.000 sec
- Action 183: select \* from Maximum\_and\_Minimum\_Humidity\_in\_NewYork LIMIT 0, 1000 Duration / Fetch 0.000 sec / 0.000 sec
- Action 184: create view times\_Maximum\_Temperature\_was\_greater\_than\_70F as select count(Max\_Temperature) Duration / Fetch 0.016 sec
- Action 185: select \* from times\_Maximum\_Temperature\_was\_greater\_than\_70F LIMIT 0, 1000 Duration / Fetch 0.000 sec / 0.000 sec

## 12. Creating and Calling View frequency of illness in new York and Colorado

MySQL Workbench

Schemas: DmddAs2

File Edit View Query Database Server Tools Scripting Help

Navigator: abc\*

SQL Editor:

```
1 • use testing;
2
3 • CREATE view Comparison_Freq_Illness as
4   Select city, count(public_opinion),public_opinion, frequency_Illness
5   from newyork_form_response
6   group by public_opinion union
7   select city, count(public_opinion), public_opinion, frequency_illness
8   from colorado_form_response
9   group by public_opinion;
10
11 • select * from Comparison_Freq_Illness;
```

Result Grid:

city	count(public_opinion)	public_opinion	frequency_Illness
NewYork	16	warm	normal
NewYork	15	moderate	normal
NewYork	13	cold	normal
Colorado	6	moderate	normal
Colorado	33	cold	extreme cough and cold

Information:

View: best\_city\_for\_generating\_wind\_en

Columns:

Max_Wind_Speed	count(bw.Max_Wind_Speed)	city
int	bigint	varchar

Action Output:

#	Time	Action	Message	Duration / Fetch
66	23:41:39	CREATE view Comparison_Freq_Illness as Select city, count(public_opinion),public.opinion, frequency_Illness from newyork_form_response group by public.opinion union select city, count(public.opinion), public.opinion, frequency_illness from colorado_form_response group by public.opinion;	0 row(s) affected	0.578 sec
67	23:41:39	select * from Comparison_Freq_Illness LIMIT 0, 1000	5 row(s) returned	0.016 sec / 0.000 sec

Type here to search

System Tray: 30°F Mostly clear, ENG, 23:41, 13-12-2022

### 13. Creating and Calling view comparison between the preferred clothing in New York, Boston and Colorado

The screenshot shows the MySQL Workbench interface with a query editor window titled 'abc\*' containing the following SQL code:

```
1 • use testing;
2
3 • CREATE view Comparison_Preff_Clothing as
4   select city, count(clothing), clothing from boston_form_response group by clothing
5   UNION
6   select city, count(clothing), clothing from newyork_form_response group by clothing
7   UNION
8   select city, count(clothing), clothing from colorado_form_response group by clothing;
9 • select * from Comparison_Preff_Clothing;
```

The 'Result Grid' tab shows the following data:

city	count(clothing)	clothing
Boston	8	t-shirt
Boston	15	jackets
NewYork	14	t-shirt
NewYork	16	full-sleeves
NewYork	14	jackets
Colorado	1	t-shirt
Colorado	5	full-sleeves
Colorado	33	jacket

The 'Output' tab shows the following log entries:

#	Time	Action	Message	Duration / Fetch
75	23:45:13	CREATE view Comparison_Preff_Clothing as select city, count(clothing), clothing from boston_form_respon...	0 row(s) affected	0.187 sec
76	23:45:17	select * from Comparison_Preff_Clothing LIMIT 0, 1000	9 row(s) returned	0.016 sec / 0.000 sec

#### 14. Creating and Calling View on which date did the city experience this particular weather conditions

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `weather` with tables `boston\_form\_response`, `boston\_weather\_nov22`, `colorado\_form\_response`, `colorado\_weather\_nov22`, and `newyork\_form\_response`.
- Query Editor (Query 1):**

```

1 use weather;
2 #Creating view for On which date did the city experience this particular weather conditions
3 create view date_particular_weathers as
4 select * from newyork_weather_nov22
5 where max_Temperature = '60' and max_dew_point = '63' and max_humidity = '93' and max_wind_speed = '21';
6
7
8 #Calling view On which date did the city experience this particular weather conditions
9 * select * from date_particular_weathers ;
10
11

```
- Result Grid:** Displays the columns: Date\_Nov, Max\_Temperature, Avg\_Temperature, Min\_Temperature, Max\_Dew\_Point, Avg\_Dew\_Point, Min\_Dew\_Point, Max\_Humidity, Avg\_Humidity, Min\_Humidity, Max\_Wind\_Speed, and Avg\_Wind\_Speed.
- Action Output:**

#	Time	Action	Message	Duration / Fetch
188	22:32:38	select * from newyork_weather_nov22 where max_Temperature = '60' and max_dew_point = '63' and ...	0 rows/returned	0.000 sec / 0.000 sec
189	22:32:38	select * from date_particular_weather LIMIT 0, 1000	Error Code: 1146. Table 'weather.date_particular_weather' doesn't exist	0.000 sec
190	22:33:00	create view date_particular_weathers as select * from newyork_weather_nov22 where max_Temperature ...	0 rows/affected	0.016 sec
191	22:33:00	select * from date_particular_weathers LIMIT 0, 1000	0 rows/returned	0.000 sec / 0.000 sec
- System Bar:** Shows the date and time as 12/3/2022 10:35 PM.

#### 15. Creating and calling view on how many times the average temperature of Colorado was around 33.

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `weather` with tables `boston\_form\_response`, `boston\_weather\_nov22`, `colorado\_form\_response`, `colorado\_weather\_nov22`, and `newyork\_form\_response`.
- Query Editor (Query 1):**

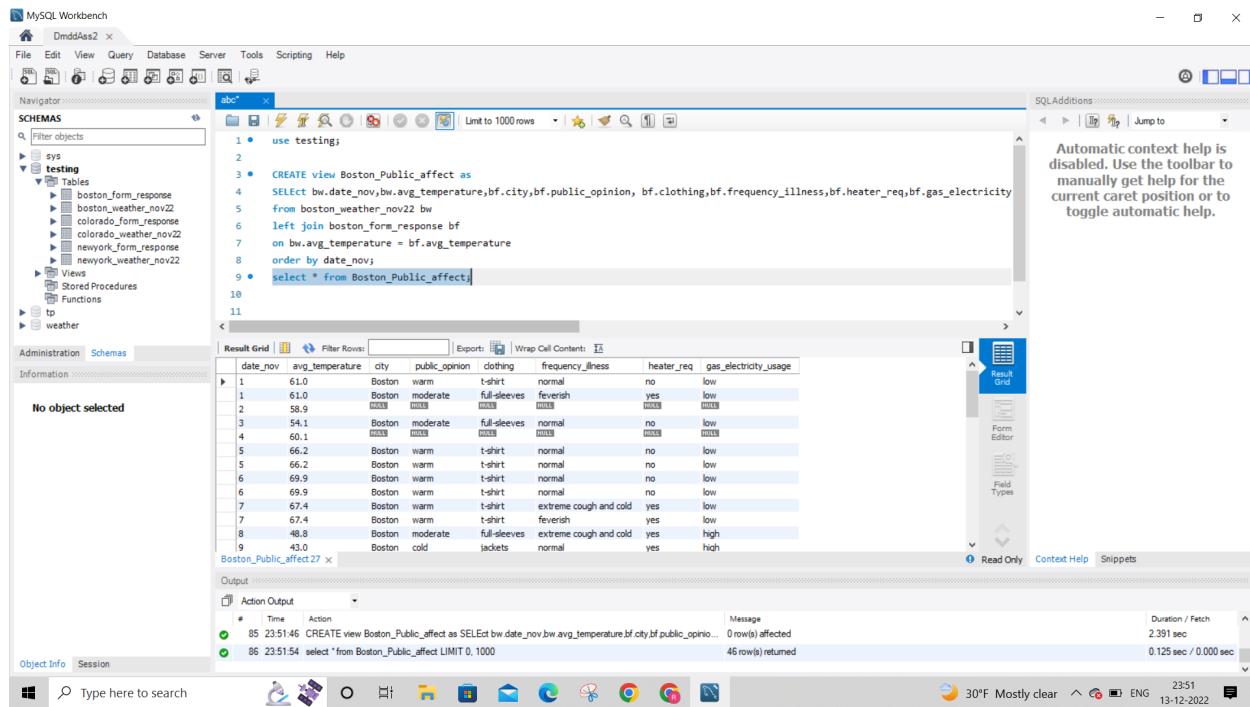
```

1 use weather;
2 #Creating view for how many times the Average Temperature of Colorado was around 33
3 create view Average_Temperature_of_Colorado_was_around_33 as
4 select avg_temperature from colorado_weather_nov22
5 where avg_temperature like '35.%';
6
7
8 #Calling view On which date did the city experience this particular weather conditions
9 * select * from Average_Temperature_of_Colorado_was_around_33;
10
11

```
- Result Grid:** Displays the column `avg\_temperature` with values 35.4 and 35.8.
- Action Output:**

#	Time	Action	Message	Duration / Fetch
190	22:33:00	create view date_particular_weathers as select * from newyork_weather_nov22 where max_Temperature ...	0 rows/affected	0.016 sec
191	22:33:00	select * from date_particular_weathers LIMIT 0, 1000	0 rows/returned	0.000 sec / 0.000 sec
192	22:39:51	create view Average_Temperature_of_Colorado_was_around_33 as select avg_temperature from colo... 0 rows/affected	0.016 sec	
193	22:39:51	select * from Average_Temperature_of_Colorado_was_around_33 LIMIT 0, 1000	2 rows/returned	0.000 sec / 0.000 sec
- System Bar:** Shows the date and time as 12/3/2022 10:39 PM.

## 16. Creating view How does the Average Temperature of Boston in month of Nov affect the public



The screenshot shows the MySQL Workbench interface with a query editor window titled 'abc\*' containing the following SQL code:

```
1 • use testing;
2
3 • CREATE view Boston_Public_affect as
4   SELECT bw.date_nov,bw.avg_temperature,bf.city,bf.public_opinion, bf.clothing,bf.frequency_illness,bf.heater_req,bf.gas_electricity
5   from boston_weather_nov22 bw
6   left join boston_form_response bf
7   on bw.avg_temperature = bf.avg_temperature
8   order by date_nov;
9 • select * from Boston_Public_affect;
```

The results grid displays 46 rows of data from the 'Boston\_Public\_affect' view. The columns are: date\_nov, avg\_temperature, city, public\_opinion, clothing, frequency\_illness, heater\_req, gas\_electricity\_usage. The data includes various weather and clothing items for Boston in November.

date_nov	avg_temperature	city	public_opinion	clothing	frequency_illness	heater_req	gas_electricity_usage
1	61.0	Boston	warm	t-shirt	normal	no	low
1	61.0	Boston	moderate	full-sleeves	feverish	yes	low
2	58.9						
3	54.1	Boston	moderate	full-sleeves	normal	no	low
4	60.1						
5	66.2	Boston	warm	t-shirt	normal	no	low
5	66.2	Boston	warm	t-shirt	normal	no	low
6	69.9	Boston	warm	t-shirt	normal	no	low
6	69.9	Boston	warm	t-shirt	normal	no	low
7	67.4	Boston	warm	t-shirt	extreme cough and cold	yes	low
7	67.4	Boston	warm	t-shirt	feverish	yes	low
8	48.8	Boston	moderate	full-sleeves	extreme cough and cold	yes	high
9	43.0	Boston	cold	jackets	normal	yes	high

The status bar at the bottom right shows the system is 30°F Mostly clear, with a duration of 0.125 sec / 0.000 sec.

Git URL: [Dhiral25/Weather-Analysis \(github.com\)](https://github.com/Dhiral25/Weather-Analysis)

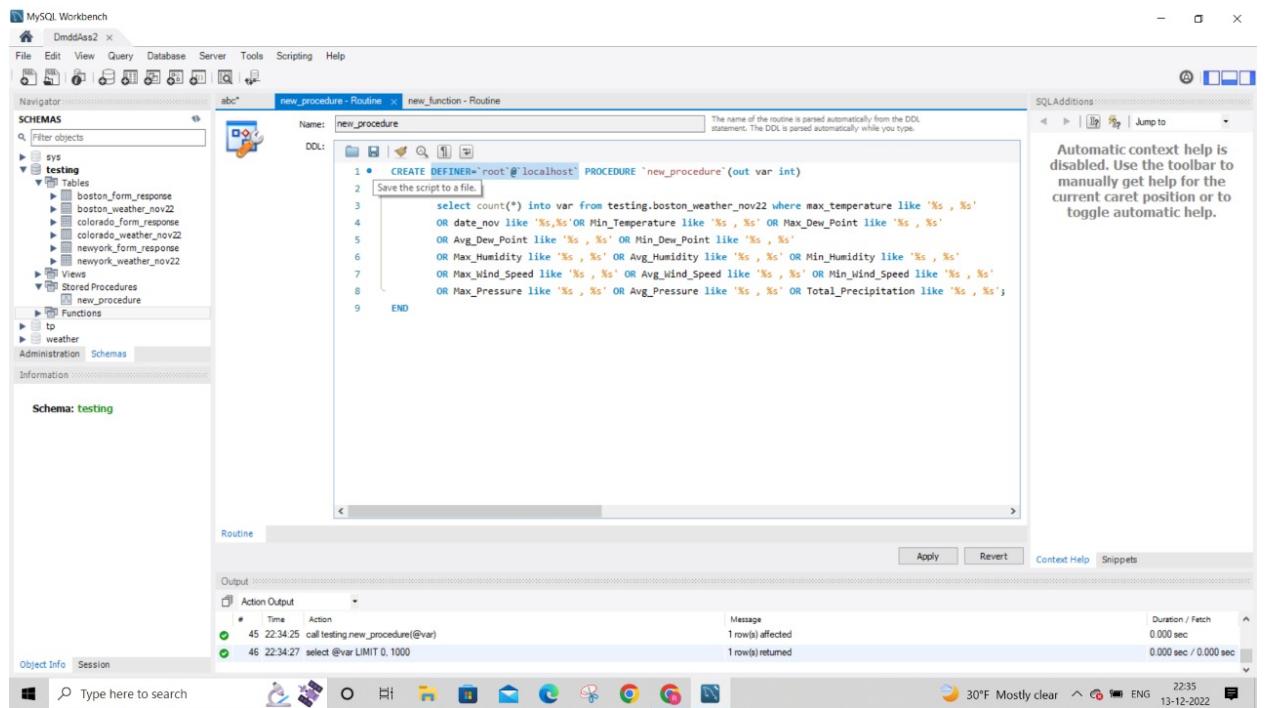






## Normalization :

Checking if table Boston\_Weather\_Nov22 is in 1NF



MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SCHEMAS Filter objects

- testing
  - Tables
    - boston\_form\_response
    - boston\_weather\_nov22
    - colorado\_form\_response
    - colorado\_weather\_nov22
    - newyork\_form\_response
    - newyork\_weather\_nov22
  - Views
  - Stored Procedures
    - new\_procedure
  - Functions
- tp
- weather

Administration Schemas

Information Schema: testing

DQL Name: new\_function

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

```

1 • CREATE DEFINER='root'@'localhost' FUNCTION `new_function`(`a` int) RETURNS varchar(20) CHARSET utf8
2     READS SQL DATA
3     DETERMINISTIC
4     BEGIN
5         declare result varchar (20);
6         SET global log_bin_trust_function_creators = 1;
7
8
9         IF `a`>0 THEN
10             SET result = 'table is not in INF';
11         ELSE
12             SET result = 'table is in INF';
13         END IF;
14         RETURN result;
15     END
  
```

Routine

Output

Action Output

#	Time	Action	Message	Duration / Fetch
45	22:34:25	call testing.new_procedure(@var)	1 row(s) affected	0.000 sec
46	22:34:27	select @var LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Context Help Snippets

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SCHEMAS Filter objects

- sys
- testing
  - Tables
    - boston\_form\_response
    - boston\_weather\_nov22
    - colorado\_form\_response
    - colorado\_weather\_nov22
    - newyork\_form\_response
    - newyork\_weather\_nov22
  - Views
  - Stored Procedures
    - new\_procedure
  - Functions
- tp
- weather

Administration Schemas

Information Schema: testing

DQL Name: new\_procedure

Result Grid

Limit to 1000 rows

```

1 • use testing;
2
3 • set @var = 0;
4 • call testing.new_procedure(@var);
5 • select @var;
6
7
8 • select testing.new_function(@var);
9
10
11
  
```

Result Grid

Filter Rows

Exports

Wrap Cell Content

@var

Result 11

Output

Action Output

#	Time	Action	Message	Duration / Fetch
45	22:34:25	call testing.new_procedure(@var)	1 row(s) affected	0.000 sec
46	22:34:27	select @var LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Read Only Context Help Snippets

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Navigator Schemas

SCHEMAS Filter objects

- sys
- testing
  - Tables
    - boston\_form\_response
    - boston\_weather\_nov22
    - colorado\_form\_response
    - colorado\_weather\_nov22
    - newyork\_form\_response
    - newyork\_weather\_nov22
  - Views
  - Stored Procedures
    - new\_procedure
  - Functions
- tp
- weather

Administration Schemas

Information Schema: testing

DQL Name: new\_function

Result Grid

Limit to 1000 rows

```

1 • use testing;
2
3 • set @var = 0;
4 • call testing.new_procedure(@var);
5 • select @var;
6
7
8 • select testing.new_function(@var);
9
10
11
  
```

Result Grid

Filter Rows

Exports

Wrap Cell Content

@var

Result 11

Output

Action Output

#	Time	Action	Message	Duration / Fetch
45	22:34:25	call testing.new_procedure(@var)	1 row(s) affected	0.000 sec
46	22:34:27	select @var LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Read Only Context Help Snippets

SQL Additions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

The screenshot shows the MySQL Workbench interface. In the top-left pane, the 'Schemas' tree is visible, showing the 'testing' schema which contains tables like 'boston\_form\_response', 'boston\_weather\_nov22', etc., and stored procedures like 'new\_procedure'. The central pane displays a query editor with the following SQL code:

```

abc* new_procedure - Routine      new_function - Routine
1 • use testing;
2
3 • set @var = 0;
4 • call testing.new_procedure(@var);
5 • select @var;
6
7
8 • select testing.new_function(@var);
9
10
11

```

The bottom pane shows the 'Result Grid' with the output of the last query:

	testing.new_function(@var)
	table is in INN

The status bar at the bottom right indicates the system is 30°F, Mostly clear, and the date is 13-12-2022.

- Satisfies 1NF requirements
- The table has a primary key attribute identified as Date\_Nov
- No multi-value attributes present
- No repeating groups

## Checking if Boston\_weather\_nov22 table is in 1NF

MySQL Workbench

DmddAs2 X

File Edit View Query Database Server Tools Scripting Help

abc\* new\_procedure - Routine new\_function - Routine

Schemas: testing

Tables: boston\_form\_response, boston\_weather\_nov22, colorado\_form\_response, colorado\_weather\_nov22, newyork\_form\_response, newyork\_weather\_nov22

Views: new\_procedure

Stored Procedures: new\_procedure

Functions: new\_function

tp weather

Administration Schemas

Information: newyork\_weather\_nov22

Columns:

- Date\_Nov int
- Max\_Temperature int
- Avg\_Temperature** varchar(10) PK
- Min\_Temperature int
- Max\_Dew\_Point int
- Avg\_Dew\_Point int
- Min\_Dew\_Point int
- Max\_Humidity int
- Min\_Humidity int
- Max\_Humidity int
- Max\_Wind\_Speed int
- Avg\_Wind\_Speed int
- Min\_Wind\_Speed int
- Avg\_Pressure int
- Min\_Pressure int
- Total\_Precipitation int

Name: new\_procedure

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DQL:

```

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `new_procedure`(out var int)
2   BEGIN
3     select count(*) into var from testing.newyork_weather_nov22 where max_temperature like '%s , %s'
4     OR date_nov like '%s,%s' OR Min_Temperature like '%s , %s' OR Max_Dew_Point like '%s , %s'
5     OR Avg_Dew_Point like '%s , %s' OR Min_Dew_Point like '%s , %s'
6     OR Max_Humidity like '%s , %s' OR Avg_Humidity like '%s , %s' OR Min_Humidity like '%s , %s'
7     OR Max_Wind_Speed like '%s , %s' OR Avg_Wind_Speed like '%s , %s' OR Min_Wind_Speed like '%s , %s'
8     OR Max_Pressure like '%s , %s' OR Avg_Pressure like '%s , %s' OR Total_Precipitation like '%s , %s';
9   END

```

Routine

Output

Action Output

#	Time	Action	Message	Duration / Fetch
55	22:37:14	select @var LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
56	22:37:16	select testing.new_function(@var) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Apply Revert Context Help Snippets

Object Info Session

Type here to search

30°F Mostly clear 22:37 13-12-2022

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

DmddAs2 X

File Edit View Query Database Server Tools Scripting Help

abc\* new\_procedure - Routine new\_function - Routine

Schemas: testing

Tables: boston\_form\_response, boston\_weather\_nov22, colorado\_form\_response, colorado\_weather\_nov22, newyork\_form\_response, newyork\_weather\_nov22

Views: new\_procedure

Stored Procedures: new\_procedure

Functions: new\_function

tp weather

Administration Schemas

Information: newyork\_weather\_nov22

Schema: testing

Name: new\_function

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DQL:

```

1 • CREATE DEFINER='root'@'localhost' FUNCTION `new_function`(a int) RETURNS varchar(20) CHARSET utf8
2   READS SQL DATA
3   DETERMINISTIC
4   BEGIN
5     declare result varchar (20);
6     SET global log_bin_trust_function_creators = 1;
7
8     IF a>0 THEN
9       SET result = 'table is not in INF';
10    ELSE
11      SET result = 'table is in INF';
12    END IF;
13    RETURN result;
14  END

```

Routine

Output

Action Output

#	Time	Action	Message	Duration / Fetch
45	22:34:25	call testing.new_procedure(@var)	1 row(s) affected	0.000 sec
46	22:34:27	select @var LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Apply Revert Context Help Snippets

Object Info Session

Type here to search

30°F Mostly clear 22:35 13-12-2022

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

abc* new_procedure - Routine      new_function - Routine
1 • use testing;
2
3 • set @var = 0;
4 • call testing.new_procedure(@var);
5 • select @var;
6
7
8 • select testing.new_function(@var);
9
10
11

Result Grid | Filter Rows | Export | Wrap Cell Content | @var | 0

```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

Action Output
# Time Action
1 45 22:34:25 call testing.new_procedure(@var)
2 46 22:34:27 select @var LIMIT 0, 1000

Message
1 row(s) affected
1 row(s) returned

Duration / Fetch
0.000 sec
0.000 sec / 0.000 sec

```

30°F Mostly clear 22:35 13-12-2022

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

- Satisfies 1NF requirements
- The table has a primary key attribute identified as Date\_Nov
- No multi-value attributes present
- No repeating groups

## checking if table Colorado\_Weather\_Nov22 is in 1NF

MySQL Workbench - DmddAss2

**Routine Editor**

**DDL:**

```

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `new_procedure`(out var int)
2   BEGIN
3     select count(*) into var from testing.colorado_weather_nov22 where max_temperature like '%s , %s'
4     OR date_nov like '%s,%s' OR Min_Temperature like '%s , %s' OR Max_Dew_Point like '%s , %s'
5     OR Avg_Dew_Point like '%s , %s' OR Min_Dew_Point like '%s , %s'
6     OR Max_Humidity like '%s , %s' OR Avg_Humidity like '%s , %s' OR Min_Humidity like '%s , %s'
7     OR Max_Wind_Speed like '%s , %s' OR Avg_Wind_Speed like '%s , %s' OR Min_Wind_Speed like '%s , %s'
8     OR Max_Pressure like '%s , %s' OR Avg_Pressure like '%s , %s' OR Total_Precipitation like '%s , %s';
9   END

```

**Output:**

Action Output	Time	Action	Message	Duration / Fetch
55 22:37:14	select @var LIMIT 0, 1000		1 row(s) returned	0.000 sec / 0.000 sec
56 22:37:16	select testing.new_procedure(@var) LIMIT 0, 1000		1 row(s) returned	0.000 sec / 0.000 sec

**Routine Editor**

**DLL:**

```

1 • CREATE DEFINER='root'@'localhost' FUNCTION `new_function`(a int) RETURNS varchar(20) CHARSET utf8
2   READS SQL DATA
3   DETERMINISTIC
4   BEGIN
5     declare result varchar (20);
6     SET global log_bin_trust_function_creators = 1;
7
8
9     IF a>0 THEN
10       SET result = 'table is not in 1NF';
11     ELSE
12       SET result = 'table is in 1NF';
13     END IF;
14     RETURN result;
15   END

```

**Output:**

Action Output	Time	Action	Message	Duration / Fetch
45 22:34:25	call testing.new_procedure(@var)		1 row(s) affected	0.000 sec
46 22:34:27	select @var LIMIT 0, 1000		1 row(s) returned	0.000 sec / 0.000 sec

```

abc* new_procedure - Routine      new_function - Routine
1 • use testing;
2
3 • set @var = 0;
4 • call testing.new_procedure(@var);
5 • select @var;
6
7
8 • select testing.new_function(@var);

Result Grid | Filter Rows | Export | Wrap Cell Content: 15
@var
0

Result 11 x
Output
Action Output
# Time Action
45 22:34:25 call testing.new_procedure(@var)
46 22:34:27 select @var LIMIT 0, 1000
Message
1 row(s) affected
1 row(s) returned
Duration / Fetch
0.000 sec
0.000 sec / 0.000 sec

Read Only Context Help Snippets

```

```

abc* new_procedure - Routine      new_function - Routine
1 • use testing;
2
3 • set @var = 0;
4 • call testing.new_procedure(@var);
5 • select @var;
6
7
8 • select testing.new_function(@var);

Result Grid | Filter Rows | Export | Wrap Cell Content: 15
testing.new_function(@var)
table is in INN

Result 15 x
Output
Action Output
# Time Action
49 22:35:38 select @var LIMIT 0, 1000
50 22:35:55 select testing.new_function(@var) LIMIT 0, 1000
Message
1 row(s) returned
1 row(s) returned
Duration / Fetch
0.000 sec / 0.000 sec
0.000 sec / 0.000 sec

Read Only Context Help Snippets

```

- Satisfies 1NF requirements
- The table has a primary key attribute identified as Date\_Nov
- No multi-value attributes present
- No repeating groups

# Checking if Boston\_Form\_response table is in 1NF

MySQL Workbench screenshot showing the creation of a stored procedure named 'new\_procedure'.

```

CREATE DEFINER='root'@'localhost' PROCEDURE `new_procedure` (out var int)
BEGIN
    select count(*) into var from testing.boston_form_response where city like '%s , %s'
    OR Avg_Temperature like '%s,%s' OR public_opinion like '%s , %s' OR clothing like '%s , %s'
    OR frequency_illness like '%s , %s' OR heater_req like '%s , %s'
    OR gas_electricity_usage like '%s , %s';
END
  
```

MySQL Workbench screenshot showing the creation of a function named 'new\_function'.

```

CREATE DEFINER='root'@'localhost' FUNCTION `new_function` (a int) RETURNS varchar(20) CHARSET utf8
READS SQL DATA
DETERMINISTIC
BEGIN
    declare result varchar (20);
    SET global log_bin_trust_function_creators = 1;

    IF a>0 THEN
        SET result = 'table is not in 1NF';
    ELSE
        SET result = 'table is in 1NF';
    END IF;
    RETURN result;
END
  
```

```

abc* new_procedure - Routine      new_function - Routine
1 • use testing;
2
3 • set @var = 0;
4 • call testing.new_procedure(@var);
5 • select @var;
6
7
8 • select testing.new_function(@var);
9
10
11

Result Grid | Filter Rows | Export | Wrap Cell Content | @var | 0

```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

Action Output
# Time Action
1 45 22:34:25 call testing.new_procedure(@var)
2 46 22:34:27 select @var LIMIT 0, 1000

Message
1 row(s) affected
1 row(s) returned

Duration / Fetch
0.000 sec
0.000 sec / 0.000 sec

```

30°F Mostly clear 22:35 13-12-2022

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

- Satisfies 1NF requirements
- The table has a primary key attribute identified as Date\_Nov
- No multi-value attributes present
- No repeating groups

# Checking if Colorado\_Form\_response table is in 1NF

MySQL Workbench

Dmddas2 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

testing

Tables

boston\_form\_response

Columns

Indexes

Foreign Keys

Triggers

boston\_weather\_nov22

colorado\_form\_response

colorado\_weather\_nov22

newyork\_form\_response

newyork\_weather\_nov22

Views

Stored Procedures

Administration Schemas

Information

Table: boston\_form\_response

Columns:

city	varchar(255)
Avg_Temperature	varchar(255)
public_opinion	varchar(255)
clothing	varchar(255)
frequency_illness	varchar(255)
heater_req	varchar(255)
gas_electricity_usage	varchar(255)

Name: new\_procedure

DOL:

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `new_procedure` (out var int)
2   BEGIN
3     select count(*) into var from testing.Colorado_form_response where city like '%s , %s'
4     OR Avg_Temperature like '%s,%s' OR public_opinion like '%s , %s' OR clothing like '%s , %s'
5     OR frequency_illness like '%s , %s' OR heater_req like '%s , %s'
6     OR gas_electricity_usage like '%s , %s';
7   END
```

Routine

Output

Action Output

#	Time	Action	Message	Duration / Fetch
61	22:40:58	select @var LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
62	22:40:02	select testing.new_function(@var) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Apply Revert Context Help Snippets

MySQL Workbench

Dmddas2 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

testing

Tables

boston\_form\_response

boston\_weather\_nov22

colorado\_form\_response

colorado\_weather\_nov22

newyork\_form\_response

newyork\_weather\_nov22

Views

Stored Procedures

new\_procedure

Functions

tp

weather

Administration Schemas

Information

Schema: testing

Name: new\_function

DOL:

```
1 • CREATE DEFINER='root'@'localhost' FUNCTION `new_function`(a int) RETURNS varchar(20) CHARSET utf8
2   READS SQL DATA
3   DETERMINISTIC
4   BEGIN
5     declare result varchar (20);
6     SET global log_bin_trust_function_creators = 1;
7
8
9     IF a>0 THEN
10       SET result = 'table is not in 1NF';
11     ELSE
12       SET result = 'table is in 1NF';
13     END IF;
14     RETURN result;
15   END
```

Routine

Output

Action Output

#	Time	Action	Message	Duration / Fetch
45	22:34:25	call testing.new_procedure(@var)	1 row(s) affected	0.000 sec
46	22:34:27	select @var LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Apply Revert Context Help Snippets

The screenshot shows the MySQL Workbench interface with two tabs open: 'new\_procedure - Routine' and 'new\_function - Routine'. The 'new\_procedure - Routine' tab contains the following SQL code:

```

1 • use testing;
2
3 • set @var = 0;
4 • call testing.new_procedure(@var);
5 • select @var;
6
7
8 • select testing.new_function(@var);
9
10
11

```

The 'new\_function - Routine' tab shows the result of the query 'select testing.new\_function(@var);' which returns the value 0.

The 'Result Grid' pane shows the output of the last query:

testing.new_function(@var)
table is in INN

The 'Action Output' pane shows the execution log:

Time	Action	Message	Duration / Fetch
49 22:35:38	select @var LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
50 22:35:55	select testing.new_function(@var) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

The screenshot shows the MySQL Workbench interface with two tabs open: 'new\_procedure - Routine' and 'new\_function - Routine'. The 'new\_procedure - Routine' tab contains the same SQL code as the previous screenshot:

```

1 • use testing;
2
3 • set @var = 0;
4 • call testing.new_procedure(@var);
5 • select @var;
6
7
8 • select testing.new_function(@var);
9
10
11

```

The 'new\_function - Routine' tab shows the result of the query 'select testing.new\_function(@var);' which returns the value 0.

The 'Result Grid' pane shows the output of the last query:

testing.new_function(@var)
0

The 'Action Output' pane shows the execution log:

Time	Action	Message	Duration / Fetch
45 22:34:25	call testing.new_procedure(@var)	1 row(s) affected	0.000 sec
46 22:34:27	select @var LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

- Satisfies 1NF requirements
- The table has a primary key attribute identified as Date\_Nov
- No multi-value attributes present
- No repeating groups

# Checking if NewYork\_Form\_response table is in 1NF

MySQL Workbench screenshot showing the creation of a stored procedure named new\_procedure.

**DDL:**

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `new_procedure` (out var int)
2
3     BEGIN
4         select count(*) into var from testing.newyork_form_response where city like '%s , %s'
5         OR Avg_Temperature like '%s,%s' OR public_opinion like '%s , %s' OR clothing like '%s , %s'
6         OR frequency_ilness like '%s , %s' OR heater_req like '%s , %s'
7         OR gas_electricity_usage like '%s , %s';
8     END
```

**Output:**

#	Time	Action	Message	Duration / Fetch
61	22:40:58	select @var LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
62	22:41:02	select testing.new_function(@var) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench screenshot showing the creation of a function named new\_function.

**DLL:**

```
1 • CREATE DEFINER='root'@'localhost' FUNCTION `new_function`(a int) RETURNS varchar(20) CHARSET utf8
2     READS SQL DATA
3     DETERMINISTIC
4
5     BEGIN
6         declare result varchar (20);
7         SET global log_bin_trust_function_creators = 1;
8
9         IF a>0 THEN
10             SET result = 'table is not in 1NF';
11         ELSE
12             SET result = 'table is in 1NF';
13         END IF;
14         RETURN result;
15     END
```

**Output:**

#	Time	Action	Message	Duration / Fetch
45	22:34:25	call testing.new_procedure(@var)	1 row(s) affected	0.000 sec
46	22:34:27	select @var LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench

Schemas: testing

```
abc* new_procedure - Routine new_function - Routine
1 • use testing;
2
3 • set @var = 0;
4 • call testing.new_procedure(@var);
5 • select @var;
6
7
8 • select testing.new_function(@var);
9
10
11
```

Result Grid | Filter Rows | Export | Wrap Cell Content | Result Grid | Form Editor | Field Types

testing.new\_function(@var)

table is in INN

Action Output

#	Time	Action	Message	Duration / Fetch
49	22:35:38	select @var LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
50	22:35:55	select testing.new_function(@var) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Result 15 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
49	22:35:38	select @var LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
50	22:35:55	select testing.new_function(@var) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

30°F Mostly clear 22:35 13-12-2022

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

MySQL Workbench

Schemas: testing

```
abc* new_procedure - Routine new_function - Routine
1 • use testing;
2
3 • set @var = 0;
4 • call testing.new_procedure(@var);
5 • select @var;
6
7
8 • select testing.new_function(@var);
9
10
11
```

Result Grid | Filter Rows | Export | Wrap Cell Content | Result Grid | Form Editor | Field Types

@var

0

Action Output

#	Time	Action	Message	Duration / Fetch
45	22:34:25	call testing.new_procedure(@var)	1 row(s) affected	0.000 sec
46	22:34:27	select @var LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Result 11 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
45	22:34:25	call testing.new_procedure(@var)	1 row(s) affected	0.000 sec
46	22:34:27	select @var LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Type here to search

30°F Mostly clear 22:35 13-12-2022

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

## Checking if table

Boston\_Weather\_Nov22, Newyork\_Weather\_Nov22, Colorado\_Weather\_Nov22 is in 2NF:

The screenshot shows a Microsoft Excel spreadsheet titled "NewYork\_Weather\_Nov22.csv - Excel". The table has 26 columns and 26 rows. The columns are labeled from A to Z. Row 1 contains the column headers: Date\_Nov, Max\_Temp, Avg\_Temp, Min\_Temp, Max\_Dew, Avg\_Dew, I\_Min\_Dew, Max\_Hum, Avg\_Humid, Min\_Humi, Max\_Winc, Avg\_Wind, Min\_Winc, Max\_Press, Avg\_Press, Min\_Press, Total\_Precipitation. The data starts from row 2, with values such as 1, 67, 62, 59, 58, 55.8, 54, 90, 80.8, 63, 15, 7.7, 3, 30.1, 30, 29.9, 0.23. The table is located in the "NewYork\_Weather\_Nov22" sheet. The status bar at the bottom right shows "30°F Clear 23:00 13-12-2022".

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W		
1	Date_Nov	Max_Temp	Avg_Temp	Min_Temp	Max_Dew	Avg_Dew	I_Min_Dew	Max_Hum	Avg_Humid	Min_Humi	Max_Winc	Avg_Wind	Min_Winc	Max_Press	Avg_Press	Min_Press	Total_Precipitation							
2	1	67	62	59	58	55.8	54	90	80.8	63	15	7.7	3	30.1	30	29.9	0.23							
3	2	71	63.3	57	54	46.2	37	78	56.1	30	14	6.1	0	30.3	30.2	30.1	0							
4	3	67	59.5	54	51	46.9	42	81	64.7	40	10	5.1	0	30.4	30.4	30.3	0							
5	4	71	59	52	58	53.2	50	97	82.3	53	13	5.7	0	30.4	30.3	30.3	0							
6	5	75	67.5	63	60	59	58	84	74.9	57	18	10.5	5	30.3	30.2	30.1	0							
7	6	73	68.1	63	63	60.6	59	87	77.3	66	12	8.1	5	30.2	30.2	30.1	0							
8	7	77	69.5	56	62	45.3	20	84	48.1	18	21	13	7	30.4	30.2	30.1	0.03							
9	8	58	52.5	47	32	25	17	50	35.5	23	17	13.6	10	30.6	30.5	30.4	0							
10	9	53	47.5	41	40	29.5	21	68	50.6	37	13	7.3	0	30.6	30.6	30.5	0							
11	10	67	57.3	48	53	45.8	39	78	66.2	47	10	5.8	0	30.4	30.3	30.2	0							
12	11	68	65	58	65	61.1	53	93	87.3	80	21	12.1	0	30.2	29.8	29.6	0							
13	12	69	66.8	62	65	57.2	47	94	72.8	47	21	13.2	5	29.7	29.5	29.4	1.27							
14	13	61	50.9	43	50	38	27	77	62.6	39	23	15.4	6	30.1	29.9	29.7	0.05							
15	14	46	42	39	27	23.8	21	57	48.8	39	23	13.3	6	30.4	30.3	30.1	0.02							
16	15	47	41.7	37	38	29.1	21	82	62	49	28	13.6	7	30.5	30.3	30.1	0							
17	16	50	45	41	43	36.3	26	93	73.5	47	28	17	9	30.1	29.9	28.9	0.68							
18	17	47	41.5	39	25	22.7	19	57	47.7	37	22	15.6	7	30.1	29.9	29.9	0.01							
19	18	44	39.9	37	27	19	13	62	44.3	28	23	16	12	30.1	30.1	30	0							
20	19	40	36.9	33	22	15.4	11	53	42.2	31	18	11.2	5	30.2	30.1	30	0							
21	20	38	35.4	31	23	10.3	6	55	36	29	26	18.3	7	30.3	30.1	29.9	0							
22	21	42	36	29	25	11.6	7	53	37.5	24	16	11.3	7	30.5	30.4	30.2	0							
23	22	51	45.4	40	28	21.3	18	53	38.8	29	14	8.9	3	30.4	30.3	30.2	0							
24	23	55	49.1	45	29	23.8	17	53	37.4	30	13	7.7	0	30.3	30.2	30.1	0							
25	24	53	46.9	41	39	33.5	28	69	60	47	9	6.3	0	30.4	30.3	30.1	0							
26	25	55	49.8	47	45	38	26	83	65.3	44	23	13	6	30.1	29.9	29.7	0							

BostonWeatherNov22.csv - Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	Date_Nov	Max_Temp	Avg_Temp	Min_Temp	Max_Dew	Avg_Dew	Min_Dew	Max_Hum	Avg_Hum	Min_Hum	Max_Winc	Avg_Winc	Wind_Min	Wind_Max	Press_Avg	Press_Min	Press_Max	Total_Precipitation					
2	1	64	61	57	60	57.5	54	100	88.3	80	12	7.7	5	30	29.9	29.9	0.04						
3	2	69	58.9	53	56	48.3	38	90	70.2	38	13	6.9	0	30.4	30.2	30	0						
4	3	60	54.1	47	51	43.3	35	86	67.4	49	14	6.6	0	30.4	30.4	30.3	0						
5	4	71	60.1	51	58	52.4	48	97	77.6	47	22	12.6	6	30.4	30.3	30.3	0						
6	5	76	66.2	58	64	60.2	56	97	82.2	60	17	11.6	5	30.3	30.2	30.2	0						
7	6	76	69.9	66	65	62.7	59	93	79	58	24	15.8	12	30.2	30.2	30.1	0						
8	7	75	67.4	55	64	45.9	17	90	53.8	16	23	14.7	9	30.3	30.1	30	0						
9	8	54	48.8	40	31	22.4	16	49	36.2	24	21	13.8	8	30.6	30.5	30.3	0						
10	9	48	43	36	38	29.7	22	79	59.9	49	13	8.3	0	30.6	30.6	30.5	0						
11	10	69	56.4	44	52	44	39	89	66.1	38	22	14.8	10	30.4	30.2	30.1	0						
12	11	70	63.3	55	68	59.2	52	100	87.2	63	23	12.3	0	30.1	29.9	29.6	0						
13	12	76	67.6	56	68	60.9	46	96	80.2	55	24	16.1	7	29.8	29.5	29.3	0.41						
14	13	56	50.3	42	50	43.2	27	93	77.2	55	20	11.9	3	30	29.8	29.7	0.04						
15	14	44	39.5	35	25	19.9	17	55	46	37	18	11.3	5	30.4	30.2	30	0.1						
16	15	43	38.2	33	33	21.9	19	86	52.6	40	9	6.4	0	30.5	30.4	30.3	0						
17	16	57	46.9	37	56	42.6	32	97	85.4	65	25	15.8	3	30.3	29.8	29.6	0.45						
18	17	46	41	36	33	27	20	83	58.6	42	25	15	9	29.9	29.8	29.8	0.34						
19	18	45	38.5	33	23	20.2	16	64	48.7	37	16	12.3	7	30.1	30	29.9	0						
20	19	43	36.8	32	24	19.6	16	69	50.5	36	17	11.3	7	30.2	30.1	30	0						
21	20	38	33.7	29	20	9.9	0	59	39.1	20	26	16.5	5	30.2	30	29.9	0						
22	21	39	33.4	26	25	13.8	4	60	45.1	35	23	14.7	6	30.4	30.3	30.1	0						
23	22	48	41.5	35	25	21.4	16	55	45.2	36	17	10	6	30.3	30.2	30.1	0						
24	23	50	44.1	36	27	25.1	21	65	48.2	40	15	9.9	5	30.4	30.2	30.1	0						
25	24	47	39.4	32	38	28.3	23	82	64.9	48	12	7.3	0	30.4	30.3	30.2	0						
26	25	52	47.2	42	47	40	28	89	76.8	56	20	12.4	7	30.1	29.8	29.6	0						

- All requirements for 1NF met
- No partial dependency on any column
- None of the fields have data calculated from other fields

Checking if table Boston\_Form\_Response , Colorado\_Form\_Response, Newyork\_Form\_Response :

Boston\_form\_response.csv - Excel

Avg_Temp	public_clothing	frequency	heater_req	gas_electricity_usage
61	warm	t-shirt	normal	no
61	moderate	full-sleeve	feverish	yes
54.1	moderate	full-sleeve	normal	no
33.7	cold	jackets	extreme	yes
66.2	warm	t-shirt	normal	no
69.9	warm	t-shirt	normal	no
67.4	warm	t-shirt	extreme	yes
48.8	moderate	full-sleeve	extreme	yes
43	cold	jackets	normal	yes
56.4	moderate	full-sleeve	extreme	yes
66.2	warm	t-shirt	normal	no
41	cold	jackets	extreme	yes
50.3	warm	t-shirt	normal	no
39.5	cold	jackets	extreme	yes
38.2	cold	jackets	extreme	yes
46.9	moderate	full-sleeve	extreme	yes
41	cold	jackets	feverish	yes
38.5	cold	jackets	extreme	yes
38.5	moderate	full-sleeve	feverish	no
33.7	cold	jackets	extreme	yes
33.4	cold	jackets	extreme	yes
44.1	cold	jackets	extreme	no
39.5	cold	jackets	extreme	yes
47.2	moderate	full-sleeve	feverish	yes

Colorado\_form\_response.csv - Excel

Avg_Temp	public_clothing	frequency	heater_req	gas_electricity_usage
44.8	moderate	t-shirt	normal	no
48.3	moderate	full-sleeve	feverish	no
28.6	cold	jacket	extreme	yes
24	cold	jacket	extreme	yes
36.2	cold	jacket	extreme	yes
32.4	cold	jacket	extreme	yes
33.9	cold	jacket	extreme	yes
36.3	cold	jacket	extreme	yes
37	cold	jacket	extreme	yes
27.4	cold	jacket	extreme	yes
21.2	cold	jacket	extreme	yes
27.8	cold	jacket	extreme	yes
31.6	cold	jacket	extreme	yes
18.9	cold	jacket	extreme	yes
19.8	cold	jacket	extreme	yes
25.4	cold	jacket	extreme	yes
16.6	cold	jacket	extreme	yes
10.4	cold	jacket	extreme	yes
22.5	cold	jacket	extreme	yes
35.4	cold	jacket	extreme	yes
32	cold	jacket	extreme	yes
35.8	cold	jacket	extreme	yes
33.8	cold	jacket	extreme	yes
24.9	cold	jacket	extreme	yes
29.8	cold	jacket	extreme	yes

NewYork\_form\_response.csv - Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	City	Avg_Temp	public_otp	clothing	frequency	heater_req	gas_electricity_usage																
2	NewYork	62	warm	t-shirt	normal	no	low																
3	NewYork	63.3	warm	t-shirt	normal	no	low																
4	NewYork	59.5	warm	t-shirt	feverish	yes	high																
5	NewYork	59	warm	t-shirt	normal	no	low																
6	NewYork	67.5	warm	t-shirt	normal	no	low																
7	NewYork	68.1	warm	full-sleeve	feverish	yes	low																
8	NewYork	69.5	warm	t-shirt	normal	no	low																
9	NewYork	52.5	moderate	t-shirt	normal	no	low																
10	NewYork	47.5	warm	full-sleeve	feverish	yes	high																
11	NewYork	57.3	warm	full-sleeve	feverish	yes	high																
12	NewYork	65	warm	t-shirt	normal	no	low																
13	NewYork	66.8	warm	t-shirt	normal	no	low																
14	NewYork	50.9	moderate	full-sleeve	feverish	yes	high																
15	NewYork	42	cold	jackets	normal	yes	high																
16	NewYork	41.7	cold	jackets	normal	yes	high																
17	NewYork	45	moderate	full-sleeve	feverish	no	low																
18	NewYork	41.5	cold	jackets	extreme	yes	high																
19	NewYork	39.9	cold	jackets	extreme	yes	high																
20	NewYork	36.9	cold	jackets	extreme	yes	high																
21	NewYork	35.4	cold	jackets	extreme	yes	high																
22	NewYork	36	cold	jackets	extreme	yes	high																
23	NewYork	45.4	moderate	full-sleeve	feverish	yes	low																
24	NewYork	49.1	moderate	full-sleeve	feverish	yes	high																
25	NewYork	46.9	moderate	full-sleeve	normal	no	low																
26	NewYork	49.8	moderate	full-sleeve	normal	yes	high																

- 2NF requirements satisfied
- No transitive dependency

