

Normalization

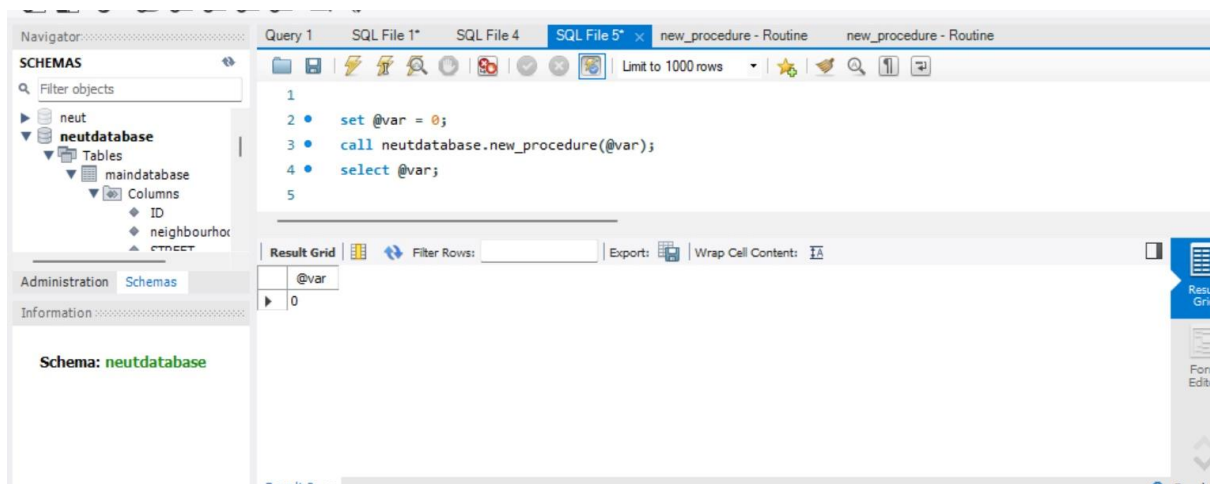
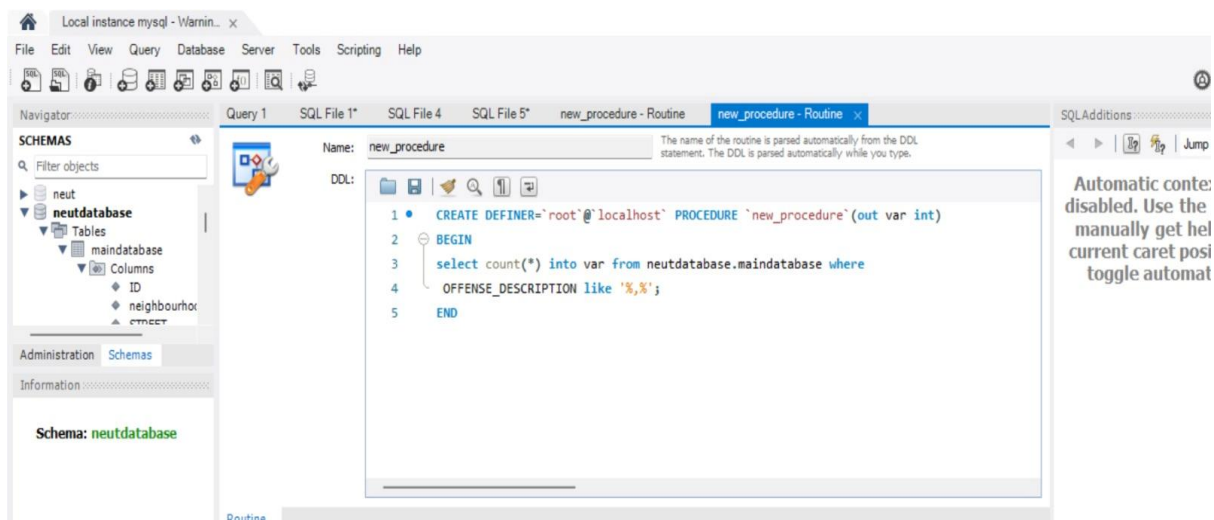
Topic Name- NEU Transportation System

In the below process, we have checked whether our tables are in 1NF, 2NF and 3NF.

1NF:

Here, in order to check whether our table is in 1NF or not we have performed the following steps –

Step 1 – We have created a stored procedure named “new _procedure” in order to check whether the values in each column are atomic and there is no repeating groups.



Step 2- Then we have created a function named “new_function” to check and display the result .

The screenshot displays a MySQL IDE interface. The top pane shows the SQL code for creating a function named 'new_function'. The code is as follows:

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

The bottom pane shows the execution of the function. The SQL statement is:

```
5
6 select neutdatabase.new_function(@var);
7
8
```

The 'Result Grid' shows the output of the function call:

neutdatabase.new_function(@var)
table in 1 NF

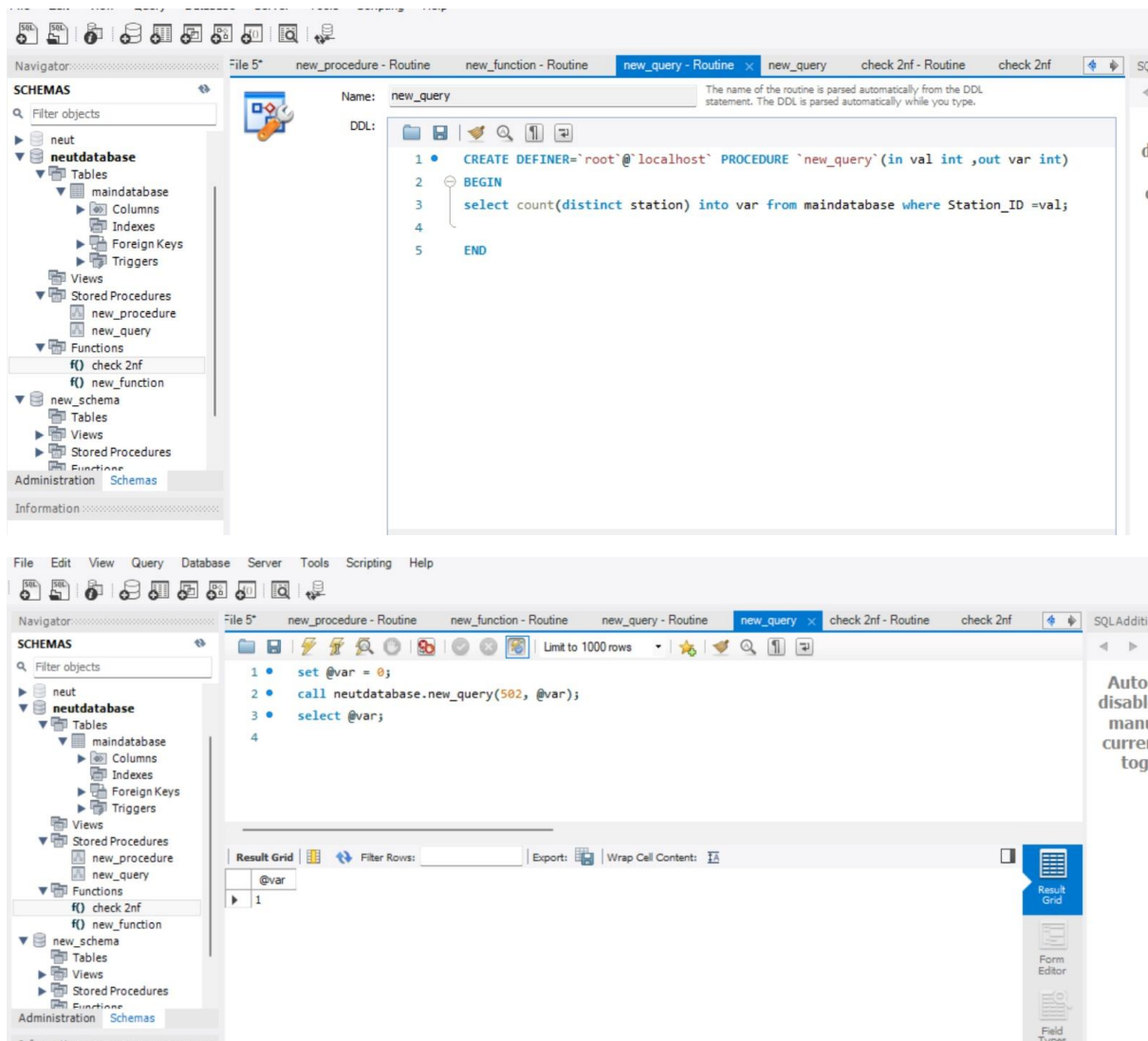
The left sidebar shows the database structure, including 'maindatabase', 'new_schema', 'phpmyadmin', and 'test'. The 'new_schema' is expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'Functions' folder is selected, and the 'new_function' is listed. The 'Administration' tab is active, and the 'Schema: neutdatabase' is displayed at the bottom.

Auto-disable maint current tog

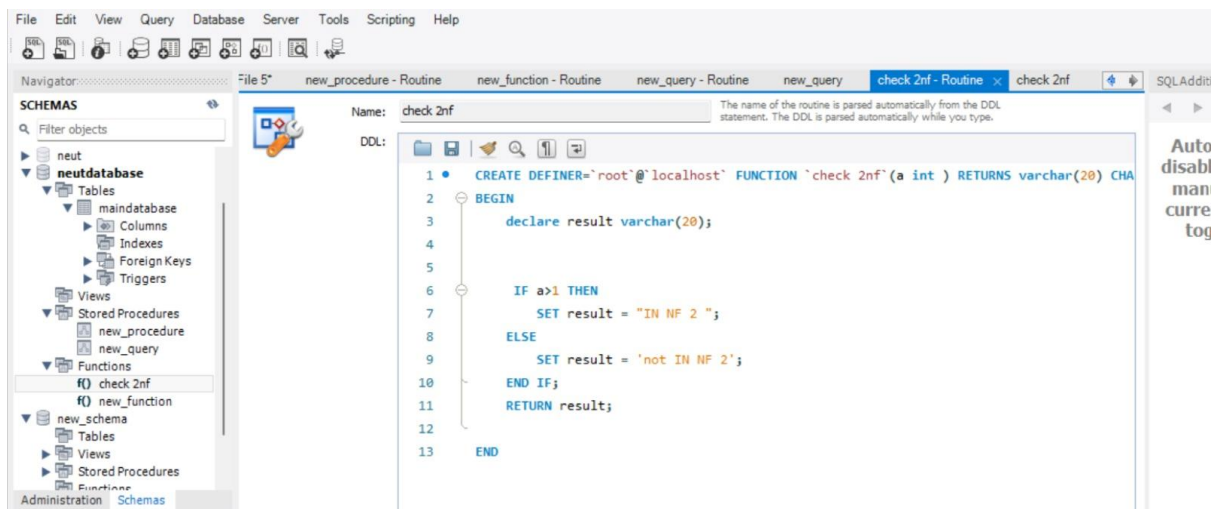
2NF:

Here, in order to check whether our table is in 2NF or not we have performed the following steps –

Step 1 – We have created a stored procedure named “new_query” in order to check that there is no partial dependencies.



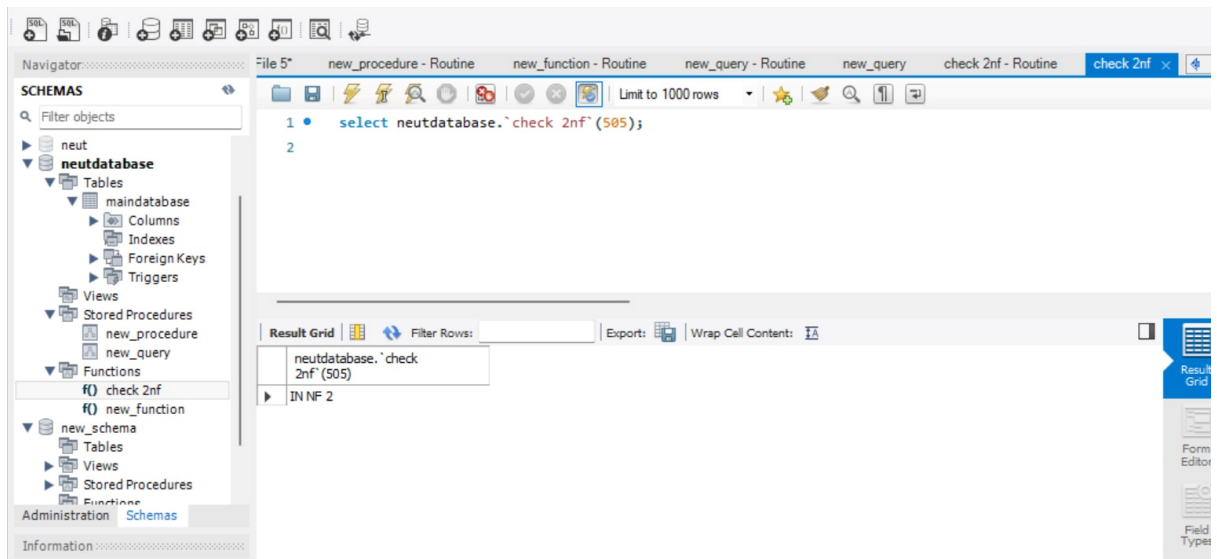
Step 2- Then we have created a function named “check 2nf” to check and display the result .



The screenshot shows the SQL Studio interface with the 'check 2nf' routine selected. The DDL editor displays the following SQL code:

```
1 CREATE DEFINER='root'@'localhost' FUNCTION `check 2nf`(a int ) RETURNS varchar(20) CHARACTER SET utf8 COLLATE utf8_general_ci
2 BEGIN
3     declare result varchar(20);
4
5
6     IF a>1 THEN
7         SET result = "IN NF 2 ";
8     ELSE
9         SET result = 'not IN NF 2';
10    END IF;
11    RETURN result;
12
13 END
```

The left sidebar shows the 'neutdatabase' schema with various objects, including the 'check 2nf' function.



The screenshot shows the SQL Studio interface with the 'check 2nf' routine selected. The query editor displays the following SQL code:

```
1 select neutdatabase.`check 2nf`(505);
2
```

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

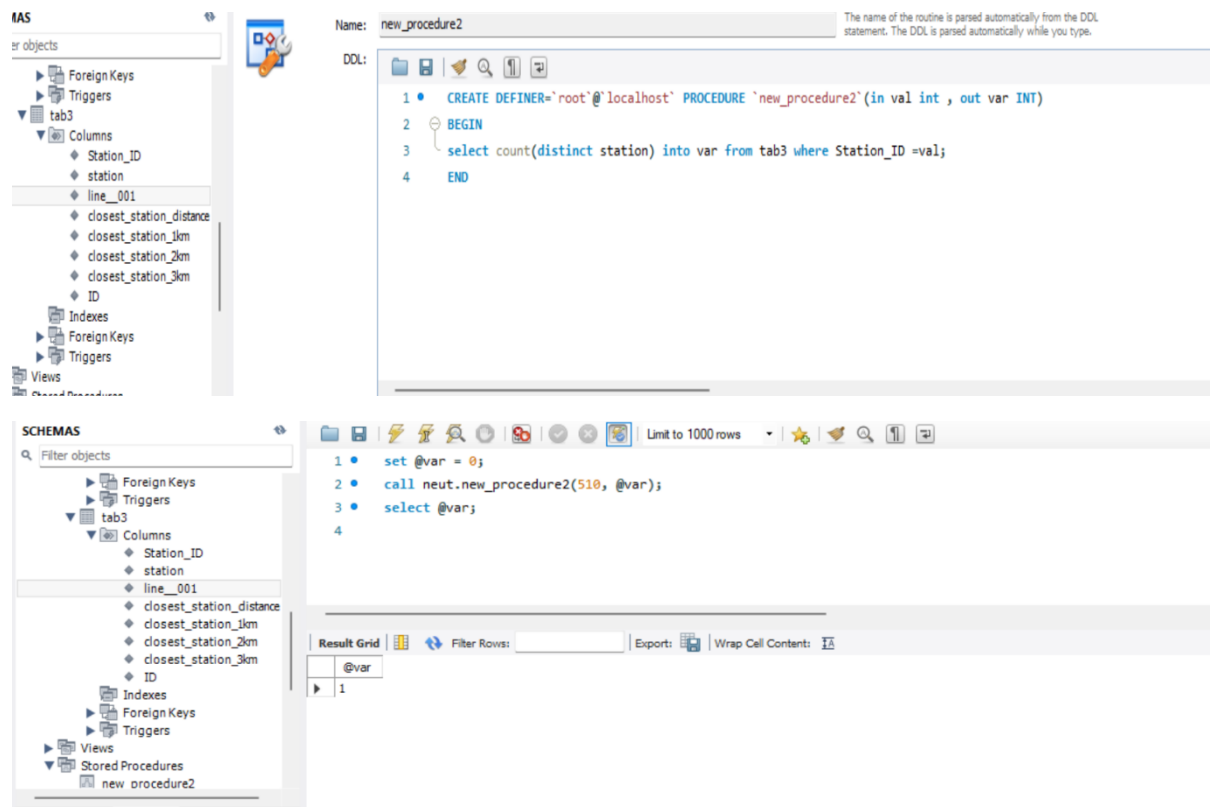
neutdatabase.`check 2nf` (505)
IN NF 2

The left sidebar shows the 'neutdatabase' schema with various objects, including the 'check 2nf' function.

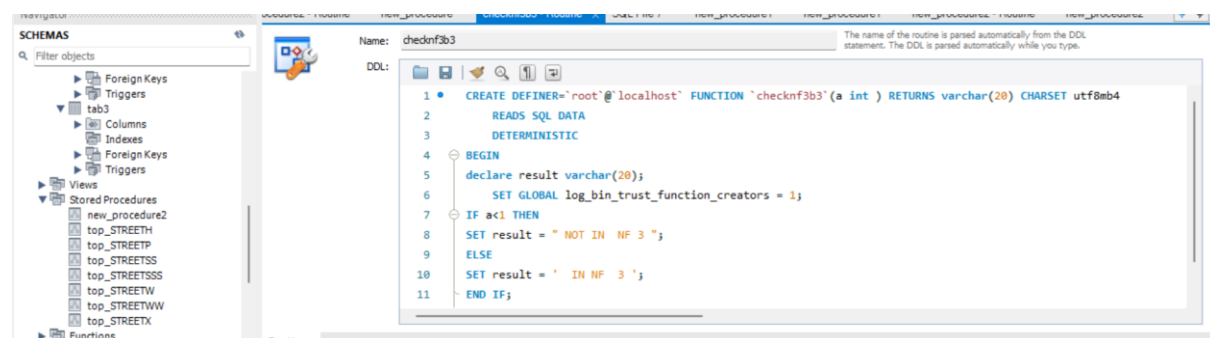
3NF:

Here, in order to check whether our table is in 3NF or not we have performed the following steps –

Step 1 – We have created a stored procedure named “new_procedure2” in order to check that there is no transitive dependencies.



Step 2- Then we have created a function named “checknf3b3” to check and display the result .



IMAS

Iter objects

Foreign Keys

Triggers

tab3

Columns

Station_ID

station

line__001

closest_station_distance

closest_station_1km

closest_station_2km

closest_station_3km

ID

Indexes

Foreign Keys

Triggers

Views

Stored Procedures

new_procedure2

Limit to 1000 rows

1 • call neut.new_procedure(510,@v1);

2 • select neut.checknf3b3(@v1)

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

neut.checknf3b3(@v1)
INF 3