CS3095D DBMS LAB

EXERCISE NO:5

Normalisation:

The process of organizing the data in database. This avoids data redundancy, insertion anomaly, update anomaly and deletion anomaly.

In this , we apply normalization theory and normalize the tables into 1NF, 2NF, 3NF.

(a)First normal form(1NF)

(b)Second normal form(2NF)

(c)Third normal form(3NF)

First normal form(1NF):

The first normal form (1NF) is a property of a relation in a relational database. A relation is in first normal form if and only if the domain of each attribute contains only atomic values.As per the rule of first normal form, each attribute contains only a single value from that domain.

Second Normal form(2NF):

A table is said to be in 2NF if both the following conditions hold:

a. Table is in First normal form (1NF)

b. Table should not contain any Partial dependencies.

Third Normal form(3NF):

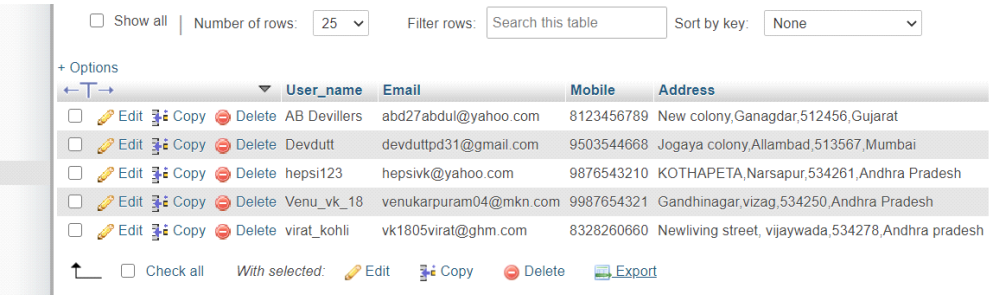
A table design is said to be in 3NF if both the following conditions hold:

a. Table must be in the second normal form (2NF)

b. Table should not hold any transitive dependencies.

TABLE 1:Users table

****1NF:**



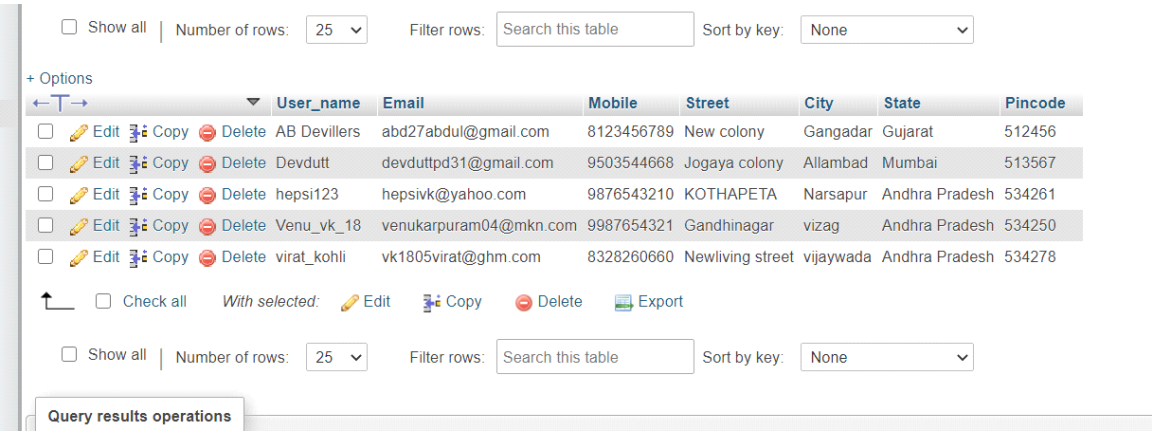
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	User_name	Email	Mobile	Address
Edit Copy Delete	AB Devillers	abd27abdul@yahoo.com	8123456789	New colony,Ganagdar,512456,Gujarat
Edit Copy Delete	Devdutt	devduttpd31@gmail.com	9503544668	Jogaya colony,Allambad,513567,Mumbai
Edit Copy Delete	hepsi123	hepsivk@yahoo.com	9876543210	KOTHAPETA,Narsapur,534261,Andhra Pradesh
Edit Copy Delete	Venu_vk_18	venukarpuram04@mkn.com	9987654321	Gandhinagar,vizag,534250,Andhra Pradesh
Edit Copy Delete	virat_kohli	vk1805virat@ghm.com	8328260660	Newliving street, vijaywada,534278,Andhra pradesh

Check all With selected: Edit Copy Delete Export

In this Users table column named Address holds multiple/composite values which is a violation to 1NF. So reduce the column Address into atomic values.



Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	User_name	Email	Mobile	Street	City	State	Pincode
Edit Copy Delete	AB Devillers	abd27abdul@gmail.com	8123456789	New colony	Gangadar	Gujarat	512456
Edit Copy Delete	Devdutt	devduttpd31@gmail.com	9503544668	Jogaya colony	Allambad	Mumbai	513567
Edit Copy Delete	hepsi123	hepsivk@yahoo.com	9876543210	KOTHAPETA	Narsapur	Andhra Pradesh	534261
Edit Copy Delete	Venu_vk_18	venukarpuram04@mkn.com	9987654321	Gandhinagar	vizag	Andhra Pradesh	534250
Edit Copy Delete	virat_kohli	vk1805virat@ghm.com	8328260660	Newliving street	vijaywada	Andhra Pradesh	534278

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

The above table satisfies 1NF condition since all the columns holds only atomic values.

****2NF:**

Here there are no partial dependencies of the primary key. Every non-prime attribuite is either or fully dependent. So,it satisfies the 2NF Form.

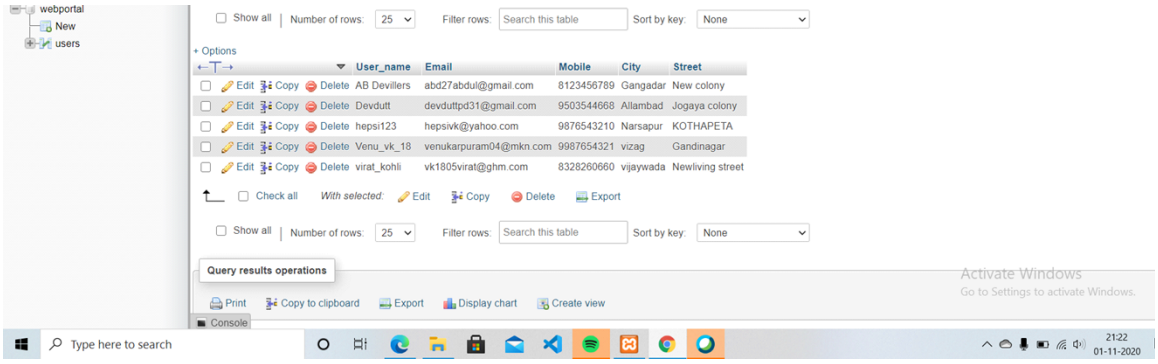
****3NF:**

Now our table follows 2NF and we need to check for the existence of transitive dependencies to claim our table follows 3NF.

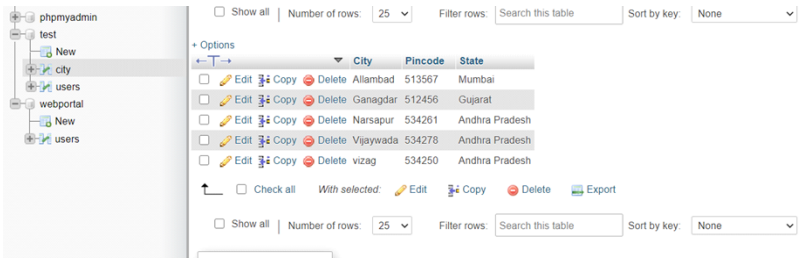
If we observe the above table, we can clearly notice that the attributes Pincode,state depends on the attribute city which is a non-prime attribute. Thus our table violates 3NF.

To remove this redundancy we need to remove all the transitive

dependents from our table and place them in a new table called City_Table in which the primary key is city which is referenced by the foreign key city in Users table.



Users table

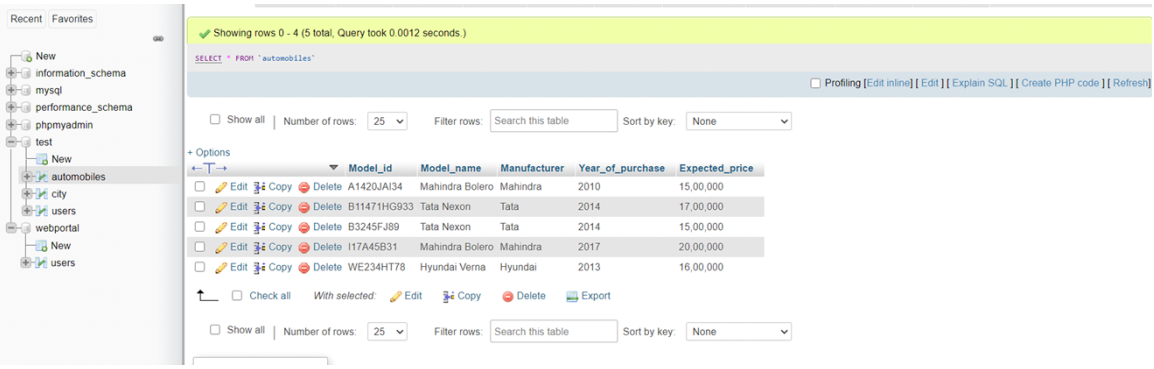


City table

With this all the transitive dependents are also removed from our table and it still obeys 2NF.

So now we can say that our table satisfies 3NF

TABLE 2: Vehicle Table



****1NF**:

Table Vehicle satisfies 1NF since all the columns in the table are

holding only atomic values.

****2NF**:

Table Vehicle violates 2NF since there exists a partial dependency in our table.

In the table vehicle primary key is composite key which is composed

of {Model_iD ,Model_name} and it is clear from the table that the attribute

Manufacturer depends only on ModelName ,so there araises a partial dependency in the table which is a violation for 2NF.

To remove this redundancy, remove the partial dependent from the vehicle table and add it in a new table called models in which the primary key is Model_name.

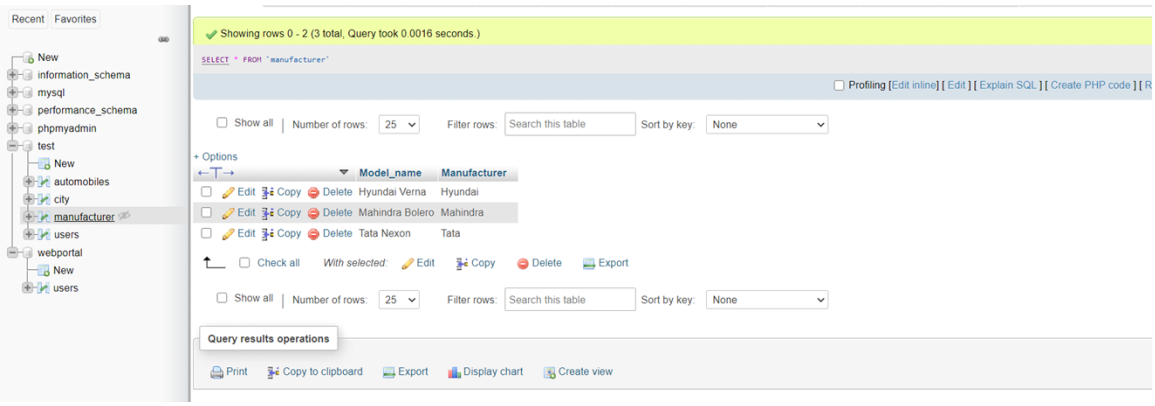


Table: ModelName

****3NF:**

Now the table vehicle follows 2NF and also there doesn't exist any transitive dependency in the table vehicle.

So we can say that table vehicle satisfies 3NF.