

Name	Hatim Yusuf Sawai
UID no.	2021300108
Experiment No.	10

AIM:	Normalization Forms in dbms
PROBLEM STATEMENT:	Apply Normalization forms: 1NF, 2NF, 3NF & BCNF to a given table in Mysql
THEORY:	<p>What Is Normalization in SQL?</p> <p>Normalization is the process to eliminate data redundancy and enhance data integrity in the table. Normalization also helps to organize the data in the database. It is a multi-step process that sets the data into tabular form and removes the duplicated data from the relational tables.</p> <p>Normalization organizes the columns and tables of a database to ensure that database integrity constraints properly execute their dependencies. It is a systematic technique of decomposing tables to eliminate data redundancy (repetition) and undesirable characteristics like Insertion, Update, and Deletion anomalies.</p> <p>1st Normal Form (1NF):</p> <p>A table is referred to as being in its First Normal Form if atomicity of the table is 1. Here, atomicity states that a single cell cannot hold multiple values.</p> <p>1. It must hold only a single-valued attribute. The First normal form disallows the multi-valued attribute, composite attribute, and their combinations.</p> <p>By applying the First Normal Form, you achieve atomicity, and also every column has unique values.</p>

Second Normal Form (2NF):

1. The first condition for the table to be in Second Normal Form is that the table must be in First Normal Form.
2. The table should not possess partial dependency. The partial dependency here means the proper subset of the candidate key should give a non-prime attribute.

Third Normal Form (3NF):

1. The first condition for the table to be in Third Normal Form is that the table should be in the Second Normal Form.
2. The second condition is that there should be no transitive dependency for non-prime attributes, which indicates that non-prime attributes (which are not a part of the candidate key) should not depend on other non-prime attributes in a table. Therefore, a transitive dependency is a functional dependency in which $A \rightarrow C$ (A determines C) indirectly, because of $A \rightarrow B$ and $B \rightarrow C$ (where it is not the case that $B \rightarrow A$).
3. The third Normal Form ensures the reduction of data duplication. It is also used to achieve data integrity.

Boyce Codd Normal Form (BCNF):

Boyce Codd Normal Form is also known as 3.5 NF. It is the superior version of 3NF and was developed by Raymond F. Boyce and Edgar F. Codd to tackle certain types of anomalies which were not resolved with 3NF.






1. The first condition for the table to be in Boyce Codd Normal Form is that the table should be in the third normal form.
2. Every Right-Hand Side (RHS) attribute of the functional dependencies should depend on the super key of that particular table.

QUERIES:

Without Normalization:

Hospital table is not normalized i.e. it has multivalued attributes and is not decomposed:

```
CREATE TABLE hospital (  
    P_id int primary key,  
    Pname varchar(30) NOT NULL,  
    Age int NOT NULL,  
    Ph_no varchar(255) NOT NULL,  
    Address varchar(30) NOT NULL  
);
```

P_id	Pname	Age	Ph_no	Address
 Filter...	 Filter...	 Filte	 Filter...	 Filter...
1	Rahul	20	7236473256,3528475869	Delhi
2	Rohan	25	7236473256,3528475869	Mumbai
3	Tanvi	30	7236473256,3528475869	Kolkata
4	Shubham	35	7236473256,3528475869	gurgoan
5	Suhail	40	7236473256,3528475869	Delhi

1NF Form:

To make the table into 1NF form we remove the multivalued attributes and make separate records for each value:

```
CREATE TABLE hospital1nf (  
    P_id int NOT NULL,  
    Pname varchar(30) NOT NULL,  
    Age int NOT NULL,  
    Ph_no bigint NOT NULL,  
    Address varchar(30) NOT NULL  
);
```

P_id	Pname	Age	Ph_no	Address
abc Filter...	abc Filter...	abc Filter...	abc Filter...	abc Filter...
1	Akshat	20	7236473256	Delhi
1	Akshat	20	9652502364	Delhi
2	Rohan	25	7236473256	Mumbai
2	Rohan	25	9854726638	Mumbai
3	Tanvi	30	7236473256	Kolkata
3	Tanvi	30	8694627846	Kolkata
4	Shubham	35	7236473256	gurgoan
5	Suhail	40	7236473256	Delhi

2NF Form:

Below table is not in 2NF form as it has partial dependency of names on ids:

```
CREATE TABLE hospital2nf (
  P_id int NOT NULL,
  D_id int NOT NULL,
  Pname varchar(30) NOT NULL,
  Dname varchar(30) NOT NULL
);
```

P_id	D_id	Pname	Dname
abc Filter..	abc Filter...	abc Filter...	abc Filter...
1	1	Akshat	Dr. Askhat
2	2	Rohan	Dr. Rohan
3	3	Tanvi	Dr. Tanvi
4	4	Shubham	Dr. Shubham
5	5	Suhail	Dr. Suhail

Therefore, the above table is decomposed into 2 tables:

```
CREATE TABLE patient2nf (  
    P_id int PRIMARY KEY NOT NULL,  
    Pname varchar(30) NOT NULL  
);  
  
CREATE TABLE doctor2nf (  
    D_id int NOT NULL PRIMARY KEY,  
    Dname varchar(30) NOT NULL  
);
```

D_id	Dname
abc Filter	abc Filter...
1	Dr. Askhat
2	Dr. Rohan
3	Dr. Tanvi
4	Dr. Shubham
5	Dr. Suhail

P_id	Pname
abc Filter	abc Filter...
1	Akshat
2	Rohan
3	Tanvi
4	Shubham
5	Suhail

3NF Form:

Below table is not in 3NF form because it contains transitive dependency where dname is dependent on d_id which is dependent on p_id, therefore, dname is dependent on p_id directly:

```
CREATE TABLE hospital3nf (  
    P_id int NOT NULL,  
    Pname varchar(30) NOT NULL,  
    D_id int NOT NULL,  
    Dname varchar(30) NOT NULL,  
    Ph_no bigint NOT NULL,  
    Address varchar(30) NOT NULL  
);
```

P_id	Pname	D_id	Dname	Ph_no	Address
1	Akshat	1	Dr. Askhat	7236473256	Delhi
2	Rohan	2	Dr. Rohan	7236473256	Mumbai
3	Tanvi	3	Dr. Tanvi	7236473256	Kolkata
4	Shubham	4	Dr. Shubham	7236473256	gurgoan
5	Suhail	5	Dr. Suhail	7236473256	Delhi

To convert the table to 3NF, we decompose the table into the following:

```
1. CREATE TABLE patient3nf (
    P_id int NOT NULL PRIMARY KEY,
    Pname varchar(30) NOT NULL,
    D_id int NOT NULL,
    Ph_no bigint NOT NULL,
    Address varchar(30) NOT NULL
);
```

P_id	Pname	D_id	Ph_no	Address
1	Akshat	1	7236473256	Delhi
2	Rohan	2	7236473256	Mumbai
3	Tanvi	3	7236473256	Kolkata
4	Shubham	4	7236473256	gurgoan
5	Suhail	5	7236473256	Delhi

```
2. CREATE TABLE doctor3nf (
    D_id int NOT NULL PRIMARY KEY,
    Dname varchar(30) NOT NULL
);
```

	<table> <tr> <th>D_id</th><th>Dname</th></tr> <tr> <td>abc Filte</td><td>abc Filter...</td></tr> <tr> <td>1</td><td>Dr. Askhat</td></tr> <tr> <td>2</td><td>Dr. Rohan</td></tr> <tr> <td>3</td><td>Dr. Tanvi</td></tr> <tr> <td>4</td><td>Dr. Shubham</td></tr> <tr> <td>5</td><td>Dr. Suhail</td></tr> </table>	D_id	Dname	abc Filte	abc Filter...	1	Dr. Askhat	2	Dr. Rohan	3	Dr. Tanvi	4	Dr. Shubham	5	Dr. Suhail
D_id	Dname														
abc Filte	abc Filter...														
1	Dr. Askhat														
2	Dr. Rohan														
3	Dr. Tanvi														
4	Dr. Shubham														
5	Dr. Suhail														
CONCLUSION:	In this experiment, we learned how to normalize a given table into the various normal forms: 1nf, 2nf & 3nf to store data with more efficiency & consistency in the database.														