DBMS PROJECT  
NORMALISATION OF TABLES



**Fall Semester 2016-2017**

**TOPIC:**

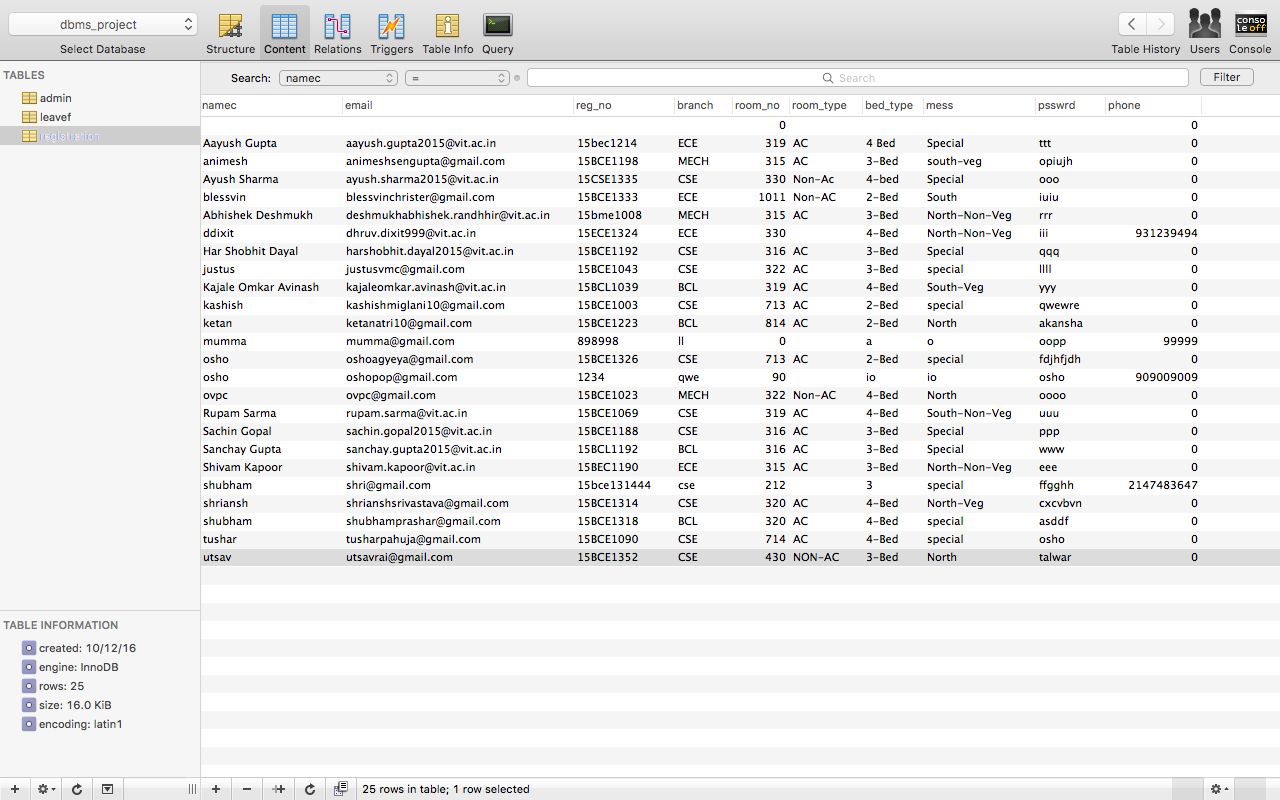
HOSTEL MANAGEMENT SYSTEM

**Teacher:**

GRACELINE JASMINE

**STUDENTS:**

* Osho Agyeya(15BCE1326)
* Kashish Miglani(15BCE1003)
* Shriansh Srivastava(15BCE1314)
* Shubham Parashar(15BCE1318)



**Given functional dependencies for table registration:**

**reg\_no->namec, email, branch, psswrd, phone, room\_no,mess**

**room\_no->room\_type,bed\_type**

**1NF:**

For this table:

1. values of each attribute are atomic
2. no composite values
3. all entries are of same kind
4. each column has unique name
5. no 2 rows are identical

Therefore it is in 1st normal form.

**2NF:**

A relation is in 2NF if it is in 1NF and all non-prime attributes are fully functionally dependant on each candidate key of R. Here, candidate key is reg\_no. Therefore non-prime attributes are namec, email, branch, psswrd, phone, room\_no, mess ,room\_type, bed\_type. All of them are fully functionally dependent on the candidate key. Therefore, the table is in 2NF.

**3NF:**

A relation is in 3NF if it is in 2NF and no non prime attribute is transitively dependant on candidate key or in other words, there should not be the case that a non-prime attribute is determined by another non-prime attribute. Here, non prime attributes are namec, email, branch, psswrd, phone, room\_no, mess ,room\_type, bed\_type, but room\_type,bed\_type are determined by room\_no . Therefore the table is not in 3NF. Now, to convert it into 3NF, we break the table registration as follows:

registration=

registration1(reg\_no,namec,email,branch,psswrd,phone,room\_no,mess)

Its functional dependencies are:

reg\_no->namec, email, branch, psswrd, phone, room\_no,mess

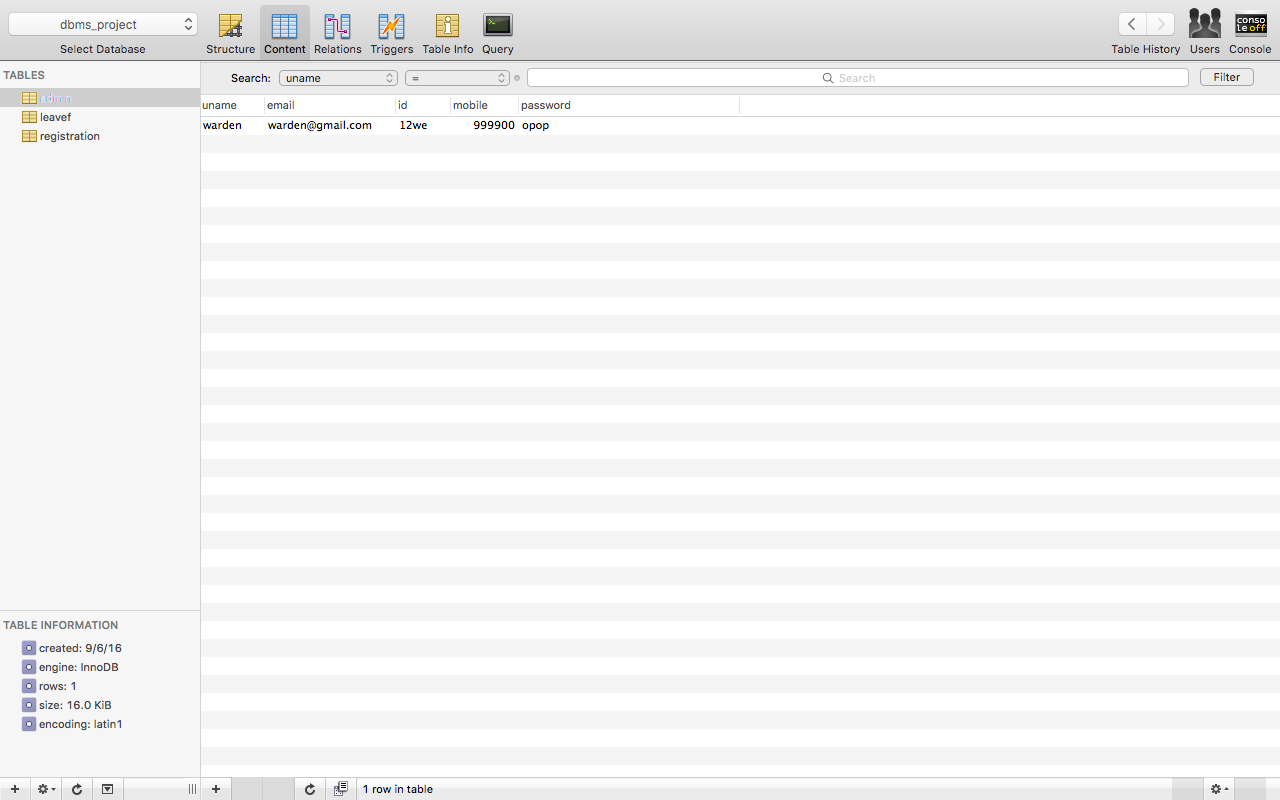
+

registration2(room\_no,room\_type,bed\_type)

Its functional dependencies are:

room\_no->room\_type,bed\_type

Now the table is in 3NF.  
  
**BCNF:**  
A relation is in BCNF if it is in 3NF and for each X->Y , X is a super key.  
For table registration1(reg\_no,namec,email,branch,psswrd,phone,room\_no,mess)  
Its functional dependencies are:  
reg\_no->namec, email, branch, psswrd, phone, room\_no,mess  
Here reg\_no is a super key. Therefore this table is in BCNF.   
For table registration2(room\_no,room\_type,bed\_type)  
Its functional dependencies are:  
room\_no->room\_type,bed\_type  
Here room\_no is the superkey. Therefore this table is also in BCNF.



**Given functional dependencies for table admin:**

**id->uname,email,mobile,password**

**1NF:**

For this table:

1. values of each attribute are atomic
2. no composite values
3. all entries are of same kind
4. each column has unique name
5. no 2 rows are identical

Therefore in 1st normal form.

**2NF:**

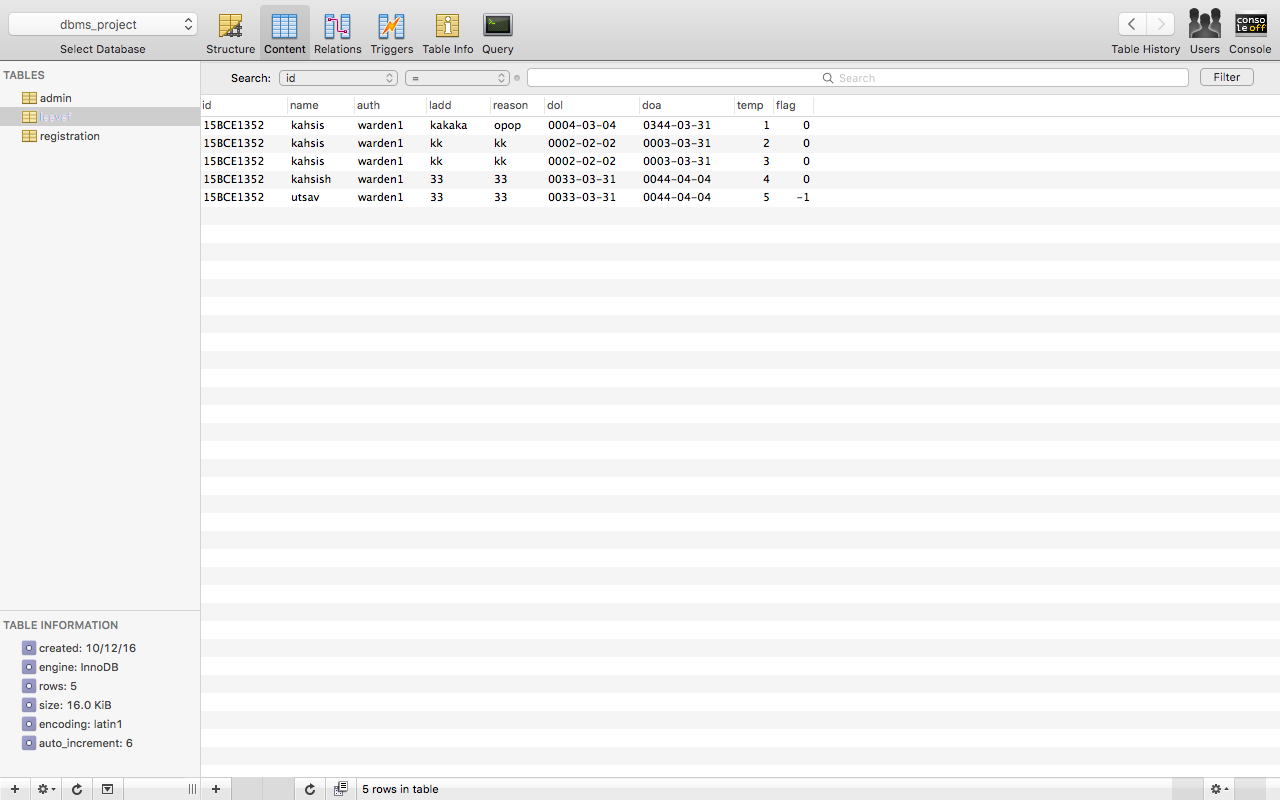
A relation is in 2NF if it is in 1NF and all non-prime attributes are fully functionally dependant on each candidate key of R. Here, candidate key is id. Therefore non-prime attributes are uname, email,mobile,password. All of them are fully functionally dependent on the candidate key. Therefore, the table is in 2NF.

**3NF:**

A relation is in 3NF if it is in 2NF and no non prime attribute is transitively dependant on candidate key or in other words, there should not be the case that a non-prime attribute is determined by another non-prime attribute. Here, non prime attributes are uname,email,mobile,password and all are determined by the candidate key only. Therefore the table is in 3NF.

**BCNF:**

A relation is in BCNF if it is in 3NF and for each X->Y , X is a super key. Here for table admin, id is a super key. Therefore the table is in BCNF.



**Given functional dependencies for table leavef:**

**id,temp->name,reason,ladd,auth**

**reason->dol,doa**

**1NF:**

For this table:

1. values of each attribute are atomic
2. no composite values
3. all entries are of same kind
4. each column has unique name
5. no 2 rows are identical

Therefore in 1st normal form.

**2NF:**

A relation is in 2NF if it is in 1NF and all non-prime attributes are fully functionally dependant on each candidate key of R. Here, candidate key is (id,temp). Therefore non-prime attributes are name, reason,ladd,auth,reason,dol,doa. All of them are fully functionally dependent on the candidate key. Therefore, the table is in 2NF.

**3NF:**

A relation is in 3NF if it is in 2NF and no non prime attribute is transitively dependant on candidate key or in other words, there should not be the case that a non-prime attribute is determined by another non-prime attribute. Here, non prime attributes are name, reason,ladd,auth,reason,dol,doa but dol,doa is determined by reason. Therefore the table is not in 3NF. Now, to convert it into 3NF, we break the table leavef as follows:

leavef=leavef1(id,temp,name,reason,ladd,auth)

Its functional dependencies are:

id,temp->name,reason,ladd,auth

+

leavef2(reason,dol,doa)

Its functional dependencies are:

reason->dol,doa

Now the table is in 3NF.

**BCNF:**

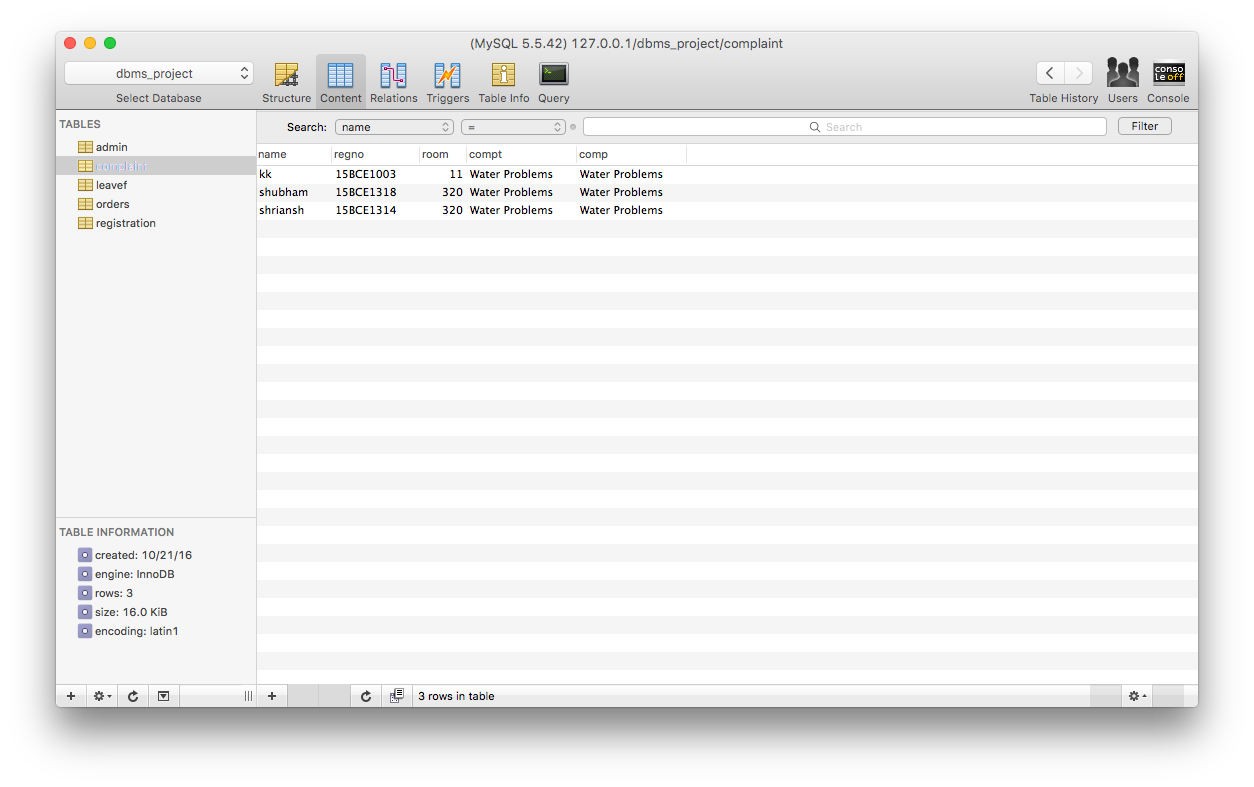
A relation is in BCNF if it is in 3NF and for each X->Y , X is a super key. Here for table

leavef=leavef1(id,temp,name,reason,ladd,auth)

Its functional dependencies are:

id,temp->name,reason,ladd,auth  
(id,temp) is a super key. Therefore, the table is in BCNF.  
For table   
leavef2(reason,dol,doa)  
Its functional dependencies are:  
reason->dol,doa  
Here , reason is a super key. Therefore, the table is in BCNF.

**Given functional dependencies for table complaint( assuming one student from a room can lodge only one complaint)**



**reg\_no->name,room**

**name->compt**

**compt->comp**

**1NF:**

For this table:

1. values of each attribute are atomic
2. no composite values
3. all entries are of same kind
4. each column has unique name
5. no 2 rows are identical

Therefore in 1st normal form.

**2NF:**

A relation is in 2NF if it is in 1NF and all non-prime attributes are fully functionally dependant on each candidate key of R. Here, candidate key is (reg\_no). Therefore non-prime attributes are name, room,compt,comp. All of them are fully functionally dependent on the candidate key. Therefore, the table is in 2NF.

**3NF:**

A relation is in 3NF if it is in 2NF and no non prime attribute is transitively dependant on candidate key or in other words, there should not be the case that a non-prime attribute is determined by another non-prime attribute. Here, non prime attributes are name, room,compt,comp but compt is determined by name and comp is determined by compt. Therefore the table is not in 3NF. Now, to convert it into 3NF, we break the table leavef as follows:

complaint=complaint1(reg\_no,name,room)

Its functional dependencies are:

reg\_no->name,room

+

complaint2(name,compt)

Its functional dependencies are:

name->compt

+

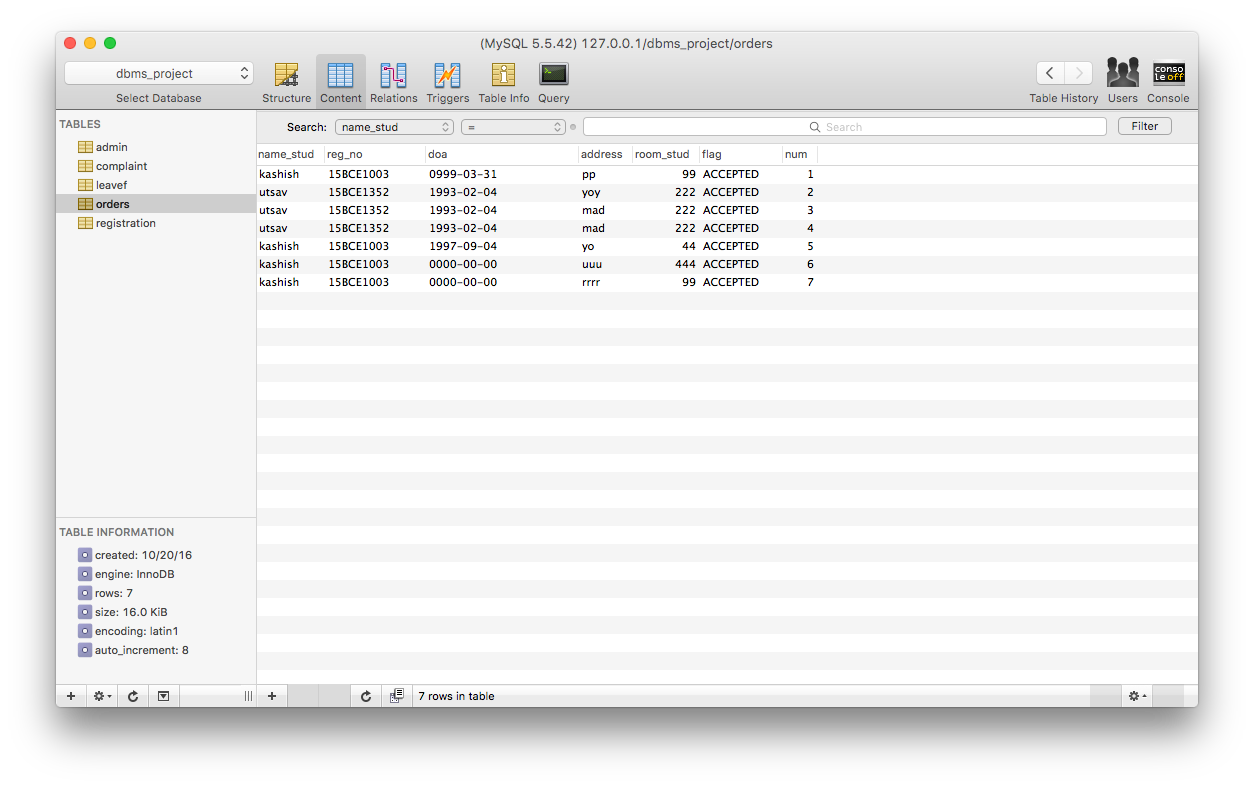
complaint3(compt.comp)

Its functional dependencies are:

compt->comp

Now the table is in 3NF.

**BCNF:**A relation is in BCNF if it is in 3NF and for each X->Y , X is a super key. Here for table complaint1(reg\_no,name,room)  
Its functional dependencies are:  
reg\_no->name,room  
reg\_no is a super key. Therefore, this table is in BCNF.  
For table  
complaint2(name,compt)  
Its functional dependencies are:  
name->compt  
name is super key. . Therefore, this table is in BCNF.  
For table  
complaint3(compt,comp)  
Its functional dependencies are:  
compt->comp  
Here, compt is a super key. Therefore, the table is in BCNF.



**Given functional dependencies for table orders**

**num->name\_stud,reg\_no,doa,address,room\_stud**

**reg\_no->name\_stud,room\_stud**

**address->doa**

**1NF:**

For this table:

1. values of each attribute are atomic
2. no composite values
3. all entries are of same kind
4. each column has unique name
5. no 2 rows are identical

Therefore in 1st normal form.

**2NF:**

A relation is in 2NF if it is in 1NF and all non-prime attributes are fully functionally dependant on each candidate key of R. Here, candidate key is (num). Therefore non-prime attributes are name\_stud, reg\_no,doa,address,room\_stud. All of them are fully functionally dependent on the candidate key. Therefore, the table is in 2NF.

**3NF:**

A relation is in 3NF if it is in 2NF and no non prime attribute is transitively dependant on candidate key or in other words, there should not be the case that a non-prime attribute is determined by another non-prime attribute. Here, non prime attributes are name\_stud, reg\_no,doa,address,room\_stud.But name\_stud,room\_stud are determined by reg\_no. Also, doa is determined by address. Therefore the table is not in 3NF. Now, to convert it into 3NF, we break the table orders as follows:

orders=orders1(num,reg\_no,address)

Its functional dependencies are:

**num->,reg\_no,address**

+

order2(reg\_no,name\_stud,room\_stud)

Its functional dependencies are:

**reg\_no->name\_stud,room\_stud**

+

orders3(address,doa)

Its functional dependencies are:

**address->doa**

Now the table is in 3NF.

**BCNF:**A relation is in BCNF if it is in 3NF and for each X->Y , X is a super key. Here for table

orders=orders1(num,reg\_no,address)

Its functional dependencies are:

num->,reg\_no,address

here,num is super key. Therefore the table is in BCNF.

For table

order2(reg\_no,name\_stud,room\_stud)

Its functional dependencies are:

reg\_no->name\_stud,room\_stud

reg\_no is super key. Therefore the table is in BCNF.  
  
  
For table  
orders3(address,doa)  
Its functional dependencies are:

address->doa

Here, address is super key. Therefore the table is in BCNF.

**ALL NORMALISED TABLES AND THEIR SCREENSHOTS:**

