

DATABASE MANAGEMENT SYSTEMS

PROJECT REPORT

TOPIC: MEDICAL CENTRE DATABASE

SUBMITTED BY:

MANSI BREJA

2016UCO1569

KAVITA MAURYA

2016UCO1579

COE-2

THIRD SEMESTER

INDEX

<u>S NO.</u>	<u>TOPIC</u>	<u>PAGE NO.</u>
1.	CERTIFICATE	3
2.	ACKNOWLEDGEMENT	4
3.	PROBLEM STATEMENT	5
4.	E-R DIAGRAM	6
5.	ENTITIES	7
6.	ENTITIES AND THEIR RELATIONSHIPS	7
7.	ATTRIBUTES I) ENTITY SETS II) RELATIONSHIP SETS	7
8.	CONVERSION OF ER MODEL TO RELATIONAL MODEL	10
9.	THE RELATIONAL MODEL	11
10.	INTEGRITY CONSTRAINTS	16
11.	TRIGGERS	18
12.	VIEWS	19
13.	NORMALISATION	21
14.	WORKING AND OUTPUTS	23

CERTIFICATE

This is to certify that the DATABASE MANAGEMENT SYSTEMS PROJECT entitled, "MEDICAL CENTRE DATABASE" submitted by MANSI BREJA AND KAVITA MAURYA in partial fulfillment of the requirements for the award of BACHELOR OF ENGINEERING in COMPUTER SCIENCE at the NETAJI SUBHAS INSTITUTE OF TECHNOLOGY is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the project has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Date:

MS. SAVITA YADAV
(Project Guide)

ACKNOWLEDGEMENT

In the accomplishment of this project successfully, many people have best owned upon me their heart pledged support and we would like to thank all of them.

Primarily we would like to thank our teacher, ma'am Savita Yadav whose valuable guidance has helped us patch this project and complete it successfully.

We would also like to thank our parents and friends for their valuable suggestions and guidance.

PROBLEM STATEMENT

A medical centre requires to store,update and delete huge chunks of data almost everyday. Hence it requires a proper organization of information so that it can be easily retrieved, updated, inserted or deleted.

Files do not turn out to be an easy option for retrieving data easily and quickly and besides it also leads to several other problems including redundancy and inconsistency.

Hence to avoid all these problem, we store all the data related to the database in the form of tables and use Database Management Systems to handle it.

Databases have ACID properties (atomicity, consistency, isolation and durability) and hence they provide an extremely efficient way for storing and managing data.

Here, in this project, we have aimed at depicting the various aspects involved while creating a medical centre's database and developed the front end using JDBC.

The project provides a menu to the user which primarily provides options to work on the admin's side and on a user's side. The menu provides several options to view update and delete records from the tables involved such as DOCTOR, PATIENT, NURSE, RECEPTIONIST ,etc.

We have also explained and implemented the key concepts of DBMS including views,normal forms and integrity constraints.

E-R DIAGRAM

ENTITIES

- Patient
- Medicine
- Equipment
- Room
- Employee
- Doctor
- Nurse
- Receptionist
- Record

ENTITIES AND THEIR RELATIONSHIPS:

- 1) Employee "IS A"
 - Doctor
 - Nurse
 - Receptionist
- 2) Receptionist "MAINTAINS" Record
- 3) Nurse "GOVERNS" Room
- 4) Doctor "ATTENDS" Patient
- 5) Patient "TAKES" Medicine
- 6) Patient "IS OPERATED WITH" Equipment
- 7) Patient "IS ASSIGNED" Room

ATTRIBUTES

1) Entity Sets

Here, all the entity sets except EQUIPMENT are strong entity sets whereas EQUIPMENT is a weak entity set.

i. EMPLOYEE

PRIMARY KEY- e_id

SIMPLE ATTRIBUTES- e_name, e_sex, e_salary, e_DOB, e_experience

MULTIVALUED ATTRIBUTE- emp_contact_no

DERIVED ATTRIBUTE- avg_salary, e_age

EMPLOYEE "IS A"

ii. DOCTOR

SIMPLE ATTRIBUTES- d_type, d_day, d_time, degree, speciality

iii. NURSE

SIMPLE ATTRIBUTES- n_work_shift

iv. RECEPTIONIST

SIMPLE ATTRIBUTES- work_shift

v. PATIENT

PRIMARY KEY- p_id

SIMPLE ATTRIBUTES- p_name, p_sex, p_age, p_date_admitted, p_date_discharged

MULTIVALUED ATTRIBUTE- p_contact_no

COMPOSITE ATTRIBUTE- p_address consisting of: house_no, street, city

vi. RECORD

PRIMARY KEY- rec_id

SIMPLE ATTRIBUTES- rec_name, rec_month, rec_year

vii. MEDICINE

PRIMARY KEY- m_code

SIMPLE ATTRIBUTES- m_name, m_price

viii. EQUIPMENT

PARTIAL KEY OR DISCRIMINATOR- equip_name

SIMPLE ATTRIBUTES- equip_price

ix. ROOM

PRIMARY KEY- room_id

SIMPLE ATTRIBUTES- r_type, r_charge

2) Relationship Sets

i. IS A

An employee can either be a doctor, or a nurse, or a receptionist.

ii. MAINTAINS

A receptionist maintains a record.

In this relationship, the participation of the entity record is total participation represented by a double line as every record needs a receptionist for its maintenance.

ii. ATTENDS

A doctor attends a patient.

In this relationship, the participation of the entity patient is total participation represented by a double line as every patient admitted needs a doctor to attend him/her.

iii. GOVERNS

A nurse governs a room.

The relationship governs consists of a descriptive attribute "START_DATE".

There is no total participation of any entity in this relationship because there can be a nurse who does not govern any room as well as there can be a room which is empty and hence needs no nurse for governing it.

iv. TAKES

A patient takes a medicine.

Since every admitted patient needs a medicine to cure, there is total participation of the entity patient in this relationship represented by a double line.

v. IS OPERATED WITH

A patient is operated with an equipment.

This relationship is represented by a double diagonal as equipment is a weak entity and the relationship is called an identifying relationship and patient is called as the owner or identifying entity set.

Also, there is total participation of entity equipment in this relationship represented by a double line because every equipment is used for some patient's operation.

vii. IS ASSIGNED

A patient is assigned a room.

Here, the participation of the entity patient is total participation represented by a double line because every patient admitted needs a room.

CONVERSION OF E-R MODEL TO RELATIONAL MODEL

- The e-r model consists of 9 entity sets and 6 relationship sets excluding the “is a” relationship and hence $9+6=15$ tables at max need to be created for the relational model representation of this database.
- Also, the e-r diagram consists of 2 multivalued attributes i.e emp_contact_no and p_contact_no which need to be represented by separate tables.
- Derived attributes are not generally represented in the actual relational schemas.
- Composite attributes are represented as separate attributes in the tables of the relational model as every attribute in the table needs to be atomic.
- A weak entity set is converted to a relational schema by including the primary key of the strong entity set in the identifying relationship as its attribute.
- There are some cases when either the relational schemas become redundant or can be combined with other schemas to reduce the number of tables and hence ensure simplicity:

1) COMBINATION OF SCHEMAS

- Entity-sets A and B, relationship-set AB – Many-to-one mapping – A’s participation in AB is total
- Generates relation schemas A, B, AB – Primary key of AB is primary_key (A) • (A is on “many” side of mapping) – AB has foreign key constraints on both A and B
- Combine A and AB relation schemas – Primary key of combined schema still primary_key (A) – Only need one foreign-key constraint, to B

2) REDUNDANCY OF SCHEMAS

- Identifying relationship is many-to-one, with no descriptive attributes
- Relational schema for weak entity-set includes primary key for strong entity-set – Foreign key constraint imposed, too
- No need to create relational schema for identifying relationship – Would be redundant to weak entity-set’s relational schema.

THE RELATIONAL MODEL

1. The relationships "takes", "is assigned", "maintains" and "attends" are combined with the entity sets having total participation in these relationships.
2. The relationship "is operated with" becomes redundant as it is an identifying relationship for a weak entity set.
3. The relationship "governs" is converted to a relational schema by taking the primary key of room and nurse entities as its attributes.
4. All the 9 entities are converted to relational schema by simply creating a table for each of them.
5. There are 2 tables which are formed by converting multi valued attributes to relational model namely "p_contact_no" and "emp_contact_no".
6. Hence there are $1+9+2=12$ tables in the relational model.

After applying the above stated points, the shown e-r diagram is converted to a relational model consisting of the following tables with their primary and foreign keys:

TABLES:

1) create table Employee (e_id numeric ,e_name varchar(20) NOT NULL ,e_sex varchar(10) NOT NULL ,e_DOB date NOT NULL, e_salary decimal(7,2) NOT NULL ,e_experience varchar(20),primary key(e_id));

2)create table emp_contact_no(e_id numeric NOT NULL,e_contact varchar(10) ,primary key(e_contact),foreign key (e_id) references Employee(e_id));

3)create table Nurse(n_id numeric, n_work_shift varchar(10) NOT NULL, primary key(n_id),foreign key(n_id) references Employee(e_id));

4)create table Room(r_id numeric,r_type varchar(20) NOT NULL,r_charge decimal(7,2) NOT NULL ,primary key(r_id));

5)Create table Governs(n_id numeric,r_id numeric,start_date date NOT NULL,primary key(n_id,r_id),UNIQUE INDEX(n_id,r_id),foreign key (n_id) references nurse(n_id),foreign key (r_id) references room(r_id));

6)create table Receptionist(recep_id numeric,work_shift varchar(20) NOT NULL,primary key(recep_id),foreign key (recep_id) references Employee(e_id));

1) EMPLOYEE

```
mysql> describe employee;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| e_id       | decimal(10,0) | NO   | PRI | NULL    |       |
| e_name     | varchar(20)   | NO   |     | NULL    |       |
| e_sex      | varchar(10)   | NO   |     | NULL    |       |
| e_DOB      | date          | NO   |     | NULL    |       |
| e_salary   | decimal(7,2)  | NO   |     | NULL    |       |
| e_experience | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

2) EMP_CONTACT_NO

```
mysql> describe emp_contact_no;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| e_id       | decimal(10,0) | NO   | MUL | NULL    |       |
| e_contact  | varchar(10)   | NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

3) DOCTOR

```
mysql> describe doctor;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| d_id       | decimal(10,0) | NO   | PRI | NULL    |       |
| d_type     | varchar(20)   | NO   |     | NULL    |       |
| d_day      | varchar(20)   | NO   |     | NULL    |       |
| d_time     | time          | NO   |     | NULL    |       |
| speciality | varchar(20)   | NO   |     | NULL    |       |
| degree     | varchar(20)   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

4) NURSE

```
mysql> describe nurse;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| n_id       | decimal(10,0) | NO   | PRI | NULL    |       |
| n_work_shift | varchar(10)   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

5) RECEPTIONIST

```
mysql> describe receptionist;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| recep_id   | decimal(10,0) | NO   | PRI | NULL    |       |
| work_shift | varchar(20)   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

6) RECORD

```
mysql> describe record;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| rec_id     | decimal(10,0) | NO   | PRI | NULL    |       |
| rec_name   | varchar(1000) | NO   |     | NULL    |       |
| rec_month  | varchar(20)   | NO   |     | NULL    |       |
| rec_year   | decimal(10,0) | NO   |     | NULL    |       |
| recep_id   | decimal(10,0) | NO   | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

7) GOVERNS

```
mysql> describe governs;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| n_id       | decimal(10,0) | NO   | PRI | NULL    |       |
| r_id       | decimal(10,0) | NO   | PRI | NULL    |       |
| start_date | date          | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

8) ROOM

```
mysql> describe room;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| r_id       | decimal(10,0) | NO   | PRI | NULL    |       |
| r_type     | varchar(20)   | NO   |     | NULL    |       |
| r_charge   | decimal(7,2)  | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

9) PATIENT

```
mysql> describe patient;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| p_id       | decimal(10,0) | NO   | PRI | NULL    |       |
| p_name     | varchar(20)   | NO   |     | NULL    |       |
| p_address_house_no | varchar(20) | YES  |     | NULL    |       |
| p_address_street | varchar(50) | YES  |     | NULL    |       |
| p_address_city | varchar(50) | YES  |     | NULL    |       |
| p_sex      | varchar(10)   | NO   |     | NULL    |       |
| p_age      | decimal(10,0) | NO   |     | NULL    |       |
| p_date_admitted | date        | NO   |     | NULL    |       |
| p_date_discharged | date        | YES  |     | NULL    |       |
| p_d_id     | decimal(10,0) | NO   | MUL | NULL    |       |
| p_r_id     | decimal(10,0) | NO   | MUL | NULL    |       |
| p_m_code   | decimal(10,0) | NO   | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)
```

10) PATIENT_CONTACT_NO

```
mysql> describe patient_contact_no;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| p_id       | decimal(10,0) | NO   | MUL | NULL    |       |
| p_contact_no | varchar(10)   | NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

11) MEDICINE

```
mysql> describe medicine;
```

Field	Type	Null	Key	Default	Extra
m_code	decimal(10,0)	NO	PRI	NULL	
m_name	varchar(20)	NO		NULL	
m_price	decimal(7,2)	NO		NULL	

3 rows in set (0.00 sec)

12) EQUIPMENT

```
mysql> describe equipment;
```

Field	Type	Null	Key	Default	Extra
equip_name	varchar(20)	NO	PRI	NULL	
equip_price	decimal(7,2)	NO		NULL	
p_id	decimal(10,0)	NO	PRI	NULL	

3 rows in set (0.00 sec)

INTEGRITY CONSTRAINTS

Database constraints are restrictions on the contents of the database or on database operations. It is a condition specified on a database schema that restricts the data to be inserted in an instance of the database.

■ DOMAIN CONSTRAINTS

Domain Constraints specifies that what set of values an attribute can take. Value of each attribute X must be an atomic value from the domain of X.

The data type associated with domains include integer, character, string, date, time, currency etc. An attribute value must be available in the corresponding domain.

In all the tables that have been created in the relational models, domain constraints have been implemented by specifying proper data types for every attribute.

■ PRIMARY KEY CONSTRAINTS

Keys are attributes or sets of attributes that uniquely identify an entity within its entity set. An Entity set E can have multiple keys out of which one key will be designated as the primary key. Primary Key must have unique and not null values in the relational table.

In all the tables created here, primary keys have been specified as either a single attribute such as (e_id) for EMPLOYEE or a group of attributes such as (n_id,r_id) for GOVERNS.

■ NOT NULL CONSTRAINT

The NOT NULL constraint enforces a column to NOT accept NULL values. This enforces a field to always contain a value, which means that you cannot insert a new record, or update a record without adding a value to this field.

Here, the not null constraint has been enforced on several attributes of tables.

For instance, the work_shift attribute for a NURSE cannot be null, the name of a MEDICINE cannot be null, the name of an EMPLOYEE cannot be null and so on.

■ REFERENCIAL INTEGRITY

A foreign key is an identifier in a table that matches the primary key of a different table.

The foreign key creates the relationship with a different table, and referential integrity refers to the relationship between these tables.

When one table has a foreign key to another table, the concept of referential integrity states that you may not add a record to the table that contains the foreign key unless there is a corresponding record in the linked table.

Here, several tables have used this concept.

List of all the foreign keys used:

- EMP_CONTACT_NO: foreign key (e_id) references Employee(e_id)
- NURSE: foreign key(n_id) references Employee(e_id)
- GOVERNS: foreign key (n_id) references nurse(n_id), foreign key (r_id) references room(r_id)
- RECEPTIONIST: foreign key (recep_id) references Employee(e_id)
- RECORD: foreign key (recep_id) references Receptionist(recep_id)
- DOCTOR: foreign key (d_id) references Employee(e_id)
- PATIENT: foreign key(p_d_id) references Doctor(d_id), foreign key (p_r_id) references room(r_id), foreign key(p_m_code) references Medicine(m_code)
- PATIENT_CONTACT_NO: foreign key(p_id) references Patient(p_id))
- EQUIPMENT: foreign key (p_id) references patient(p_id));

TRIGGERS

The MySQL trigger is a database object that is associated with a table. It will be activated when a defined action is executed for the table. The trigger can be executed when you run one of the following MySQL statements on the table: INSERT, UPDATE and DELETE and it can be invoked before or after the event.

We have used 2 triggers here:

In table employee and in table room to ensure that if the values of e_salary of employee and that of room_charge of room are below a certain value, then an error message is displayed on the screen and no insertion takes place in the tables.

TRIGGER-1

```
mysql> delimiter //
mysql> create trigger emp_sal_check before insert on employee
-> for each row
-> begin
-> if new.e_salary<30000 then
-> signal sqlstate '45001' set message_text="MINIMUM SALARY IS 30000! MISTAKE!";
-> end if;
-> end;//
Query OK, 0 rows affected (0.13 sec)

mysql> delimiter ;

mysql> INSERT INTO EMPLOYEE VALUES(150,'HEERA JAIN','MALE','1989-03-03',200,'2 YEARS');
ERROR 1644 (45001): MINIMUM SALARY IS 30000! MISTAKE!
mysql>
```

TRIGGER-2

```
mysql> delimiter //
mysql> create trigger room_charge before insert on room
-> for each row
-> begin
-> if new.r_charge<1000 then
-> signal sqlstate '45001' set message_text="MINIMUM ROOM CHARGE IS RS.1000! MISTAKE!";
-> end if;
-> end;//
Query OK, 0 rows affected (0.11 sec)

mysql> delimiter ;
mysql> insert into room values(380,'AC II',20);
ERROR 1644 (45001): MINIMUM ROOM CHARGE IS RS.1000! MISTAKE!
mysql>
```

VIEWS

VIEWS are virtual tables .By virtual, we mean, the tables do not store any data of their own but display data stored in other tables.

The Purpose of Views are to reduce code complexity and take the advantage of Code Re-usability.

1) For getting doctor's name using employee id from the table employee,

```
mysql> CREATE VIEW DOCTOR_NAME AS
-> select d_id, e_name from employee natural join doctor where e_id=d_id;
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> select * from doctor_name;
+-----+-----+
| d_id | e_name |
+-----+-----+
| 102 | MEENA KUMARI |
| 104 | ISHAN AHLAWAT |
| 105 | SHAILENDRA SHAH |
| 106 | LAKSHITA DUGGAL |
| 108 | SHAILESH MISHRA |
| 110 | PARAMVEER SINGH |
| 112 | HEERA AGGARWAL |
| 114 | JAY GUPTA |
| 115 | PRAKRITI BANSAL |
| 116 | ANJALI AGARWAL |
| 117 | ANUPAM SEN |
| 120 | RIYA AHLUWALIA |
| 123 | AMIT TRIVEDI |
| 125 | PIYUSH WADHWA |
| 128 | RIYA JAIN |
| 129 | V.P GUPTA |
+-----+-----+
16 rows in set (0.04 sec)
```

2)For getting nurse's name using employee id from the table employee,

```
mysql> CREATE VIEW NURSE_NAME AS
-> ^C
mysql> CREATE VIEW NURSE_NAME AS
-> select n_id , e_name from employee natural join nurse where n_id=e_id;
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> select * from nurse_name;
+-----+-----+
| n_id | e_name |
+-----+-----+
| 101 | SHAILESH MISHRA |
| 103 | PRIYA YADAV |
| 109 | RIYA JAIN |
| 113 | REEMA SETHI |
| 121 | SHRUTI BATRA |
| 122 | REKHA VERMA |
| 127 | AKANKSHA JAIN |
| 130 | PRIYANSHU BATRA |
+-----+-----+
8 rows in set (0.00 sec)
```

3) For getting receptionist's name using employee id from the table employee,

```
mysql> CREATE VIEW RECEP_NAME AS  
-> select recep_id , e_name from employee natural join receptionist where recep_id=e_id;  
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> select * from recep_name;  
+-----+-----+  
| recep_id | e_name  
+-----+-----+  
|      107 | RAMESH KUMAR SHAH  
|      111 | REETA KHURANA  
|      118 | R.K MISHRA  
|      119 | KRITI KHARBANDA  
|      124 | DEEPSHIKHAR KHATRI  
|      126 | SHREYA MITTAL  
+-----+-----+  
6 rows in set (0.00 sec)
```

NORMALISATION

TABLE-1: PATIENT_DOCTOR

P_id	P_name	P_sex	P_age	D_id	D_salary
801	JIYA	FEMALE	24	101	45000
802	GAURAV	MALE	35	101	45000
...

TABLE-2: DOCTOR

D_id	D_salary
101	45000
...	...

TABLE-3: PATIENT

P_id	P_name	P_sex	P_age	D_id
801	JIYA	FEMALE	24	101
802	GAURAV	MALE	35	101
...

■ 1NF

The First normal form (1NF) sets basic rules for an organized database :

- Define the data items required, because they become the columns in a table.
- Place the related data items in a table.
- Ensure that there are no repeating groups of data.
- Ensure that there is a primary key.

The table-1 has repeating groups of data and hence is not in 1NF whereas table-2 and 3 satisfy all the above given rules and are in 1NF.

■ 2NF

A database is in second normal form if it satisfies the following conditions:

- It is in first normal form
- All non-key attributes are fully functional dependent on the primary key

In a table, if attribute B is functionally dependent on A, but is not functionally dependent on a proper subset of A, then B is considered fully functional dependent on A. Hence, in a 2NF table, all non-key attributes cannot be dependent on a subset of the primary key. Note that if the primary key is not a composite key, all non-key attributes are always fully functional dependent on the primary key. A table that is in 1st normal form and contains only a single key as the primary key is automatically in 2nd normal form.

In the tables given above, table-1 is not in 1NF and hence it cannot be in 2NF also.

But tables-2 and 3 satisfy the above rule and hence are in 2NF.

■ 3NF

A database is in third normal form if it satisfies the following conditions:

- It is in second normal form
- There is no transitive functional dependency

By transitive functional dependency, we mean we have the following relationships in the table:

A is functionally dependent on B, and B is functionally dependent on C. In this case, C is transitively dependent on A via B.

In the above formed table, we can very easily make out that since the functional dependencies, $p_id \rightarrow d_id$,

$D_id \rightarrow d_salary$ hold true,

We have a transitive functional dependency and hence it is not in 3NF.

But, the 2 tables formed after decomposition perfectly follow the given rules for being in 3NF.

■ BCNF

A relational schema R is considered to be in Boyce–Codd normal form (BCNF) if, for every one of its dependencies $X \rightarrow Y$, one of the following conditions holds true:

- $X \rightarrow Y$ is a trivial functional dependency (i.e., Y is a subset of X)
- X is a superkey for schema R

Consider the table-1:

Clearly we can observe here that this table is not in BCNF.

The functional dependency,

$D_id \rightarrow D_salary$ is a non trivial functional dependency because D_id is not a superkey for the above relation.

The superkey is- P_id

Hence, we need to decompose this table into further 2 tables.

DECOMPOSITION

- Find a nontrivial functional dependency $X \rightarrow Y$ which violates the BCNF condition (where the X is not a superkey)
- Split your table in two tables:
one with attributes XY (all attributes from the dependency),
one with X attributes together with the remaining attributes from the original relation.

In tables-2 and 3, only the trivial functional dependencies hold and hence both are in BCNF.

In this project, we have implemented this concept on a bigger scale by forming 2 separate tables from patient and doctor but consisting of several other attributes as well.

WORKING AND OUTPUTS

BASIC MENU

1. Admin

What task would you like to carry out?

1. View information

Details related to:

1. Patients
2. Doctors
3. Nurse
4. Receptionist
5. Room
6. Records
0. Exit

2. Insert new Information

Select your task:

1. Insert new Employee
CHOOSE AN OPTION:
 1. Doctor
 2. Nurse
 3. Receptionist
 0. Return
2. Insert new medicine
3. Insert new Equipment
0. Exit

3. Update existing information

Select your task:

1. Update information of Employee
CHOOSE AN OPTION:
 1. Doctor
 2. Nurse
 3. Receptionist
 0. Return
2. Update information of medicine
3. Update room details
0. Exit

4. Delete some existing information

Select your task:

1. Delete some Employee
 1. Doctor
 2. Nurse
 3. Receptionist
 0. Return
2. Delete some medicine
3. Delete some Equipment
0. Exit

0. Exit Admin Page

2. User

CHOOSE AN OPTION TO WORK AND PROCEED:

1. Doctor

Select your task:

1. See all patient details
2. See all nurse details
3. See all records
0. Exit

1. Nurse

Select your task:

- 1. See all medicine details
- 2. See all equipment details
- 0. Exit

2. Receptionist

Select your task:

1. View information

CHOOSE AN OPTION:

- 1. Doctors
- 2. Nurse
- 3. Records
- 4. Patients
- 0. Return

2. Patient related

CHOOSE AN OPTION:

- 1. Add
- 2. Update
- 3. Delete
- 0. Return

3. Record Related

CHOOSE AN OPTION:

- 1. Add
- 2. Update
- 3. Delete
- 0. Return

0. Exit

0. Exit

0. Exit

OUTPUTS

```
<terminated> Medical [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe
WELCOME TO THE

CARE AND CURE MEDICAL CENTRE
...Because we care for you :)

HELLO SIR/MA'AM!

Kindly choose an option to work and proceed:
1. Admin
2. User

0. Exit
1

WELCOME ADMIN!

What task would you like to carry out?
1. View information
2. Insert new information
3. Update existing information
4. Delete some existing information

0. Exit Admin Page
1

What information would you like to view?

Details related to:
1. Patients
<
```

```
<terminated> Medical [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe
Details related to:
1. Patients
2. Doctors
3. Nurses
4. Receptionists
5. Rooms
6. Records
0. Exit
2

ID      Name
102     MEENA KUMARI
104     ISHAN AHLAWAT
105     SHAILENDRA SHAH
106     LAKSHITA DUGGAL
108     SHAILESH MISHRA
110     PARAMVEER SINGH
112     HEERA AGGARWAL
114     JAY GUPTA
115     PRAKRITI BANSAL
116     ANJALI AGARWAL
117     ANUPAM SEN
120     RIYA AHLUWALIA
123     AMIT TRIVEDI
125     PIYUSH WADHWA
128     RIYA JAIN
129     V.P GUPTA

Enter a doctor's ID to view complete details :
114

*****
<
```

```

<terminated> Medical [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (28-Oct-2017, 12:15:38 am)
114
*****
Id : 114
Name : JAY GUPTA
Gender : MALE
Doctor is : PERMANENT
Speciality : GYNECOLOGIST
Degree : PHD
Days of Visit : MON,TUES,SAT
Time : 01:00:00
Salary : 78000.0
Experience : 5.6 YEARS
Doctor's Contact No/No.s : 3448142341
*****

Do you wish to view details of any other doctor ?(Enter 1 for yes and 0 for no):
0

What information would you like to view?

Details related to:
1. Patients
2. Doctors
3. Nurses
4. Receptionists
5. Rooms
6. Records
0. Exit
0

```

```

<terminated> Medical [Java Application] C:\Program Files\Java\jre
What information would you like to view?

Details related to:
1. Patients
2. Doctors
3. Nurses
4. Receptionists
5. Rooms
6. Records
0. Exit
0

WELCOME ADMIN!

What task would you like to carry out?
1. View information
2. Insert new information
3. Update existing information
4. Delete some existing information

0. Exit Admin Page
3

Select your task:

1. Update information of Employee
2. Update information of medicine
3. Update room details
0. Exit
1

```

```
<terminated> Medical [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe
```

CHOOSE AN OPTION:

- ```
1. Doctor
2. Nurse
3. Receptionist
0. Return
```

2

|     |                 |
|-----|-----------------|
| 101 | SHAILESH MISHRA |
| 103 | PRIYA YADAV     |
| 109 | RIYA JAIN       |
| 113 | REEMA SETHI     |
| 121 | SHRUTI BATRA    |
| 122 | REKHA VERMA     |
| 127 | AKANKSHA JAIN   |
| 130 | PRIYANSHU BATRA |

```
Enter the id of nurse whose details you want to modify:
```

121

Old Details:->->->->->....

\*\*\*\*\*

Id : 121

Name : SHRUTI BATRA

Work Shift : EVENING

Gender : FEMALE

```
Salary : 88000.0
```

Experience : 5.2 YEARS

Incharge of Room Nos. :306 , 307 , END

Nurse's Contact No/No.s :3443545141 3443545641

\*\*\*\*\*

Choose the field you wish to update:



```
<terminated> Medical [Java Application] C:\Program Files\
```

Choose the field you wish to update:

1. Name
2. Salary
3. Experience
4. Contact Nos.
5. Work shift of nurse
0. Return

2

Enter salary:

40000

1 record Updated

Choose the field you wish to update:

1. Name
2. Salary
3. Experience
4. Contact Nos.
5. Work shift of nurse
0. Return

Q

CHOOSE AN OPTION:

- ```
1. Doctor
2. Nurse
3. Receptionist
0. Return
```

0

Select your task:

^

```

<terminated> Medical [Java Application] C:\Program Files\Java\jre1.8
Select your task:

1. Update information of Employee
2. Update information of medicine
3. Update room details
0. Exit
0

WELCOME ADMIN!

What task would you like to carry out?
1. View information
2. Insert new information
3. Update existing information
4. Delete some existing information

0. Exit Admin Page
0

Kindly choose an option to work and proceed:
1. Admin
2. User

0. Exit
2

WELCOME USER!

```

```

<terminated> Medical [Java Application] C:\Program Files\Java\jre1.8
WELCOME USER!

CHOOSE AN OPTION TO WORK AND PROCEED:
1. Doctor
2. Nurse
3. Receptionist
0. Exit
3

Select your task:

1. View information
2. Patient related
3. Record Related
0. Exit
1

CHOOSE AN OPTION:

1. Doctors
2. Nurses
3. Records
4. Patients
0. Return
3

Record ID

902
908
910
916

```

<terminated> Medical [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (28-Oct-2017, 12:15:38 am)

903
918
904
912
905
909
914
906
907
913

Enter a record's ID to view complete details :

904

Details of the record are as follows

Record Id : 904

Record of : MEDICINE STOCK ORDER PLACED

Record Month : JANUARY

Record Year : 2016

Handled By : KRITI KHARBANDA

Do you wish to view details of any other record ?(Enter 1 for yes and 0 for no):

0

CHOOSE AN OPTION:

1. Doctors
2. Nurses
3. Records

<terminated> Medical [Java Application] C:\Program Files

3. Records
4. Patients
0. Return

4

ID	NAME
801	KAMLA VYAS
802	MEERA JAIN
803	FATIMA VYAS
804	REEMA KAUR
805	MOHIT KHATRI
806	PRATAP GARG
807	MEERA AGARWAL
808	RATAN DEEP
809	GEETA MIDHA
810	KIRAN BINDAL
811	RASHMI DESAI
812	FARUQ SHAH
813	KARAN KATARIA
814	KATRINA KUMAR
815	MOHAN VYAS
816	KAMLA MEHTA
817	PEERA GANDHI
818	MEENAL KUMARI
819	PURAN MEHTA
820	KIRAN DAYAL
821	MANKIRAT DUA
822	INDIRA SINGH
823	KAMAL KUMAR
824	LALIT YADAV
825	PRIYA CHUGH

```

<terminated> Medical [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (28-Oct-2017, 12:15:38 am)
824      LALIT YADAV
825      PRIYA CHUGH

Enter a patient's ID to view complete details :
823

*****
Patient id : 823
Name : KAMAL KUMAR
ADDRESS :
House No. : FLAT NO-32
Street : MEENA BAZAR
City : DELHI
Gender : MALE
Age : 44
Admitted Date : 2015-12-04
Discharged Date : still admitted
Room No. : 120
Room Type : AC I
Doctor assigned : RIYA AHLUWALIA
Patient's Contact No/No.s : 5657712877
*****

Do you wish to view details of any other patient ?(Enter 1 for yes and 0 for no):
0

CHOOSE AN OPTION:

1. Doctors
2. Nurses
3. Records
4. Patients

```

```

<terminated> Medical [Java Application] C:\Program Files\Java\jre1.8.0_144\bin\javaw.exe (28-Oct-2017, 12:15:38 am)
0

Select your task:

1. View information
2. Patient related
3. Record Related
0. Exit
0
Oops!!!
Not a valid Choice

WELCOME USER!

CHOOSE AN OPTION TO WORK AND PROCEED:

1. Doctor
2. Nurse
3. Receptionist
0. Exit
0

Kindly choose an option to work and proceed:

1. Admin
2. User

0. Exit
0
Thank you :)
For Visiting

```