KLE TECH KLE Technological University
Creating Value
Leveraging Knowledge

School of Computer Science and Engineering

DBMS Course project Report

On

Hospital Management Database

List of Team Members:

NAME	*ROLL NO*	*USN*
Srishti S Eligar	137	01FE19BCS042
Ankita A Mane	146	01FE19BCS052
Swathi Pardeshi	153	01FE19BCS060
Muskan R Havaldar	165	01FE19BCS007

Content

1. Introduction

1.1. Hospital Management

The purpose of the project entitled as “HOSPITAL MANAGEMENT SYSTEM” is to computerize the Front Office Management of Hospital to develop software which is user friendly simple, fast, and cost– effective. It deals with the collection of patient’s information, diagnosis details, etc. Traditionally, it was done manually. The main function of the system is register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully System input contains patient details, diagnosis details, while system output is to get these details on to the screen. Only they can add data into the database. The data can be retrieved easily. The data are well protected for personal use and makes the data processing very fast.

1.2. Database Management System

The project Hospital Management system includes registration of patients, storing their details into the system, and also computerized billing in the labs. The software has the facility to give a unique id for every patient and stores the details of every patient and the staff automatically. It includes a search facility to know the current status of each room. User can search availability of a doctor and the details of a patient using the id.

Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

Hospital Management System is designed for multi-speciality hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration and critical financial accounting, in a seamless flow.

Hospital Management System is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing. Hospital Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes.

1.3. Problem Statement

- A Database is being created for storing Hospital Management information. This project aims to design and implement the database for hospital management system.

- Traditionally, it was done manually. The main function of the system is register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully.

- System input contains patient details, diagnosis details, while system output is to get these details on to the screen.

- The hospital has staff, rooms and patients.

- Each staff has a staff id(unique), shift, name, year of experience, date of joining, address, salary, phone number and each staff is assigned to a job type.

- Every job type has a job id(unique), name and qualification.

- Staff is assigned to room. Each room has a room no(unique), type, phone number, number of beds.

- Staff performs test and each test has an unique test id, name, date of test, date of result, amount, staff id of staff who performs it and prescription id.

- Each patient has a patient id(unique), name, gender, age, phone number, email, address, type (inpatient or outpatient).

- Patients receive prescription. Each prescription has a prescription id(unique), date, diagnosis, medicine name, quantity, dose and details of the staff who prescribes it and patient who receives it.

- Patient can use insurance. Each insurance has an id(unique), name, amount, phone number and address of the insurance company.

- In-Patients have patient record. Each patient record has a record id(unique), admit date, discharge date, in status, amount, bed number, patient's id and room number.

1.4. Objectives

1. Define hospital
2. Recording information about the Patients that come.
3. Recording information related to diagnosis given to patients.
4. Recording information related to tests the patients are prescribed.
5. Keeping information about various diseases and medicines available to cure them.

These are the various jobs that need to be done in a hospital by the operational staff and Doctors. All these works are done on papers.

1.5. Motivation

1. Information about Patients is done by just writing a few Patient details . Whenever the Patient comes up his information is stored freshly.
2. Diagnosis information to patients is generally recorded on the document, which contains Patient information. It is destroyed after some time period to decrease the paper load in the office.
3. Test records of patients are maintained in pre-formatted sheets, which are kept in a file.

4. Information about various diseases is not kept as any document. Doctors themselves do this job by remembering various medicines.

All this work is done manually by the receptionist and other operational staff and lot of papers are needed to be handled and taken care of. Doctors have to remember various medicines available for diagnosis and sometimes miss better alternatives as they can't remember them at that time.

2. Requirement Collection and Analysis

2.1. Introduction

The most critical aspect of specification is the gathering and compilation of system and user requirements. This process is normally done in conjunction with managers and users. The major goal in requirements gathering process is to:

- Collect the data used by the organization,
- Identify relationships/conditions to be applied on the data,
- Identify future data needs, and
- Determine how the data is used and generated.
- Identify the functions that are performed on the data

The starting place for data collection is gathering existing forms and reviewing policies and systems. Then, ask users what the data means, and determine their daily processes.

Following subsections discusses the data requirements and functional & non functional requirements identified based on the following activities collected from the hospital management system.

- The hospital has many jobtypes, each jobtype is uniquely identified by its jobtype ID, name of the jobtype and the qualification required for the same. Each jobtype is uniquely identified by its jobtype ID

• The hospital keeps all staffs' unique ID, first, middle and last name, address, phone number, salary, date of joining, years of experience, shift and the jobtype ID. Each staff is uniquely identified by his/her staff ID.

• The hospital also keeps many types of insurances for the patients and each insurance is identified by its ID, name, insurance company contact number and the amount required to b that insurance each insurance here is uniquely identified by its insurance ID.

• The hospital keeps a track of all patients where each patient is uniquely identified by

a patient ID, patient's first, middle and last name, whether he/she is an inpatient or an outpatient, contact number, email ID, age, his/her address, gender, insurance ID. Each patient here is uniquely identified by its patient ID.

• The hospital maintains all the prescriptions given to the patients which is identified by the prescription ID, medicine name, its dosage, quantity, diagnosis information, the staff ID and the patient Id. Each prescription here is uniquely identified by its prescription ID.

• The hospital keeps a track of all the tests conducted which is identified by test ID, name, date of the test conducted, the amount required for the test to be conducted, the staff ID and the prescription Id. Each test here is uniquely identified by its test ID.

• The hospital maintains all the information about the rooms/wards present which in turn is

identified by its room number, phone number, room type, floor number at which the room is located, number of beds in that room and the staff Id assigned to that room. Each room here is uniquely identified by its room ID.

• The hospital maintains all the information about the patient record information which is identified b its ID the date when the patient was identified, the discharge date, his/her IN patient, the amount the patient paid for the hospital treatments, the bed number, room number and the patient ID. Each patient-record here is uniquely identified by its patient-record ID.

2.2. Data requirement

Data requirement describes the data to be stored in the database pertaining to activities of the hospital database mangement system as described in section 2.1. Details of the data stored in the database is shown in the table 2.1 and Table 2.2.

Table 2.1

S.NO	**GROUP**	**Data related to the group**
1	Job Type	Job type Id, name, qualification
2	Staff	Staff Id, first, middle and last name, phone number, date of joining, years of experience, address, salary, shift
3	Insurance	Insurance Id, name, contact number of the company of the selected insurance, address,
4	Patient	Patient ID, first, middle and last Name, type, contact number, email ID, age, gender, insurance Id.
5	Prescription	Prescription ID, medicine name, dosage, date, quantity, diagnosis, staff ID, patient ID
6	Test	Test ID, Test name, date, amount, staff ID, Prescription ID
7	Room	Room number, phone number, Type, floor number, number of beds, staff ID
8	Patient Record	Record ID, admit date, discharge ID, in_status, amount, bed number, room number, patient ID

Table 2.2

S.NO	**CONDITIONS**
1	Many staff members can have the same jobtype but each staff member can have only one jobtype
2	Each staff member can conduct more than one tests

3	Each staff member can be assigned to many numbers
4	Many tests can result in the same prescription
5	Each staff member can prescribe multiple prescrpitions
6	Each patient can be prescribed more than one medicine
7	Each patient can only have one insurance
8	Each room can have many patient records

2.3. Functional Requirement

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks. So, it’s important to make them clear both for the development team and the stakeholders (clients). Table 2.3 shows the different types users of driving school database application and their respective responsibilities (tasks). Table 2.4 shows the different functions and user can perform on the database.

Table 2.3

S.NO	**USERS**	**TASKS**
1	Administrator	Data administration functionalities
2	Doctors	Responsible for Viewing and modifying every branch data
3	Nurses	Responsible for Viewing every Branch data
4	Ward managers	Responsible for Viewing and modifying respective rooms of a particular floor data

Table 2.4

S.NO	FUNCTIONS	USERS
1	Insert the records into the database	Administrator, Doctors Ward managers

2	Delete the records from the database	Administrator, Doctors Ward managers

2.4. Non-Functional Requirement

Non-functional Requirements (NFRs) for our hospital management system include attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

3. Database Design

3.1. Introduction

The requirements gathering and specification provides you with a high-level understanding of the organization, its data, and the processes that you must model in the database. Database design involves constructing a suitable model of for the information. Since the design process is complicated, especially for large databases, database design is divided into three phases:

- Conceptual database design
- Logical database design
- Physical database design.

In our project work we are addressing the conceptual database design using ER modelling and logical database design using the implementation data model called Relational model.

3.2. Conceptual Design

Conceptual database design involves modelling the collected information at a high-level of abstraction without using a particular data model or DBMS. This model allows for easy communication between end-users and database developers and has a clear method to convert from high-level model to relational model. The most popular model for conceptual database design is the Entity Relationship model which describes data as attribute, entity and relationship.

Table 3.1 shows the list of attributes Table 3.2 shows the list of entity types and table 3.3 shows the list of relationship types identified for the requirement discussed in the section 2.1.

Table 3.1: List of attributes

Sl.NO	Attribute Name	Attribute Type	Justification	Entity Type
1	J_ID	Simple		Jobtype
2	J_NAME	Simple		Jobtype
3	Qualification	Simple		Jobtype
4	S_ID	Simple		Staff
5	SHIFT	Simple		Staff
6	S_Fname	Simple		Staff
7	S_Mname	Simple		Staff
8	S_Lname	Simple		Staff
9	S_YOE	Simple		Staff
10	S_ADDRESS	Simple		Staff
11	S_SALARY	Simple		Staff
12	S_PH_NO	Simple		Staff
13	S_DOJ	Simple		Staff
14	PR_ID	Simple		Prescription
15	PR_DATE	Simple		Prescription
16	MED_name	Simple		Prescription
17	PR_DOSE	Simple		Prescription
18	PR_Quantity	Simple		Prescription

19	Diagnosis	Simple		Prescription
20	P_ID	Simple		Patient
21	P_Fname	Simple		Patient
22	P_Mname	Simple		Patient
23	P_Lname	Simple		Patient
24	P_GENDER	Simple		Patient
25	P_AGE	Simple		Patient
26	P_TYPE	Simple		Patient
27	P_PH_NO	Simple		Patient
28	P_ADDRESS	Simple		Patient
29	P_EMAIL_ID	Simple		Patient
30	I_ID	Simple		Insurance
31	I_NAME	Simple		Insurance
32	I_ADDRESS	Simple		Insurance
33	I_PH_NO	Simple		Insurance
34	I_AMT	Simple		Insurance
35	REC_ID	Simple		Patient_record
36	Admit_DATE	Simple		Patient_record
37	Discharge_DATE	Simple		Patient_record
38	IN_STATUS	Simple		Patient_record
39	AMOUNT	Simple		Patient_record
40	BED_NO	Simple		Patient_record
41	R_NO	Simple		Room
42	R_TYPE	Simple		Room
43	R_FLOORNO	Simple		Room

44	NO_OF_BEDS	Simple		Room
45	R_PH_NO	Simple		Room
46	T_ID	Simple		Test
47	T_NAME	Simple		Test
48	T_DATE	Simple		Test
49	T_ResultDATE	Simple		Test
50	T_AMT	Simple		Test

Table 3.2: List of Entity Types

Sl.NO	Entity Type Name	Type Of Entity Type	Justification
1	Jobtype	Strong	It has key attribute
2	Staff	Strong	It has key attribute
3	Patient	Strong	It has key attribute
4	Prescription	Strong	It has key attribute
5	Test	Strong	It has key attribute
6	Room	Strong	It has key attribute
7	Patient_record	Strong	It has key attribute
8	Insurance	Strong	It has key attribute

Table 3.3: List of Relationship Types

Sl.NO	Relationship Type Name	Type of Relationship Type	Justification	Participating Entity Type with Cardinality ratio	Participation

1	Has	Simple		Jobtype	Partial
				Staff	Total
2	Performs	Simple		Staff	Partial
				Test	Toatal
3	Assigned to	Simple		Staff	Partial
				Room	Total
4	Prescribes	Simple		Staff	Partial
				Prescription	Total
5	Directs	Simple		Prescription	Partial
				Test	Total
6	Is connected to	Simple		Room	Total
				Patient_record	Total
7	Receives	Simple		Patient	Total
				Prescription	Total
8	Uses	Simple		Patient	Partial
				Insuarance	Total
9	Has	Simple		Patient	Partial
				Patient_record	Total

3.2.1 E-R Diagram, Schema Diagram and Normalization

Entity relationship diagram of the proposed system as described in the requirement analysis is shown in the figure 3.1and figure 3.2 shows the schema diagram obtained after converting ER diagram to relational model.

HOSPITAL MANAGMENT ER DIAGRAM

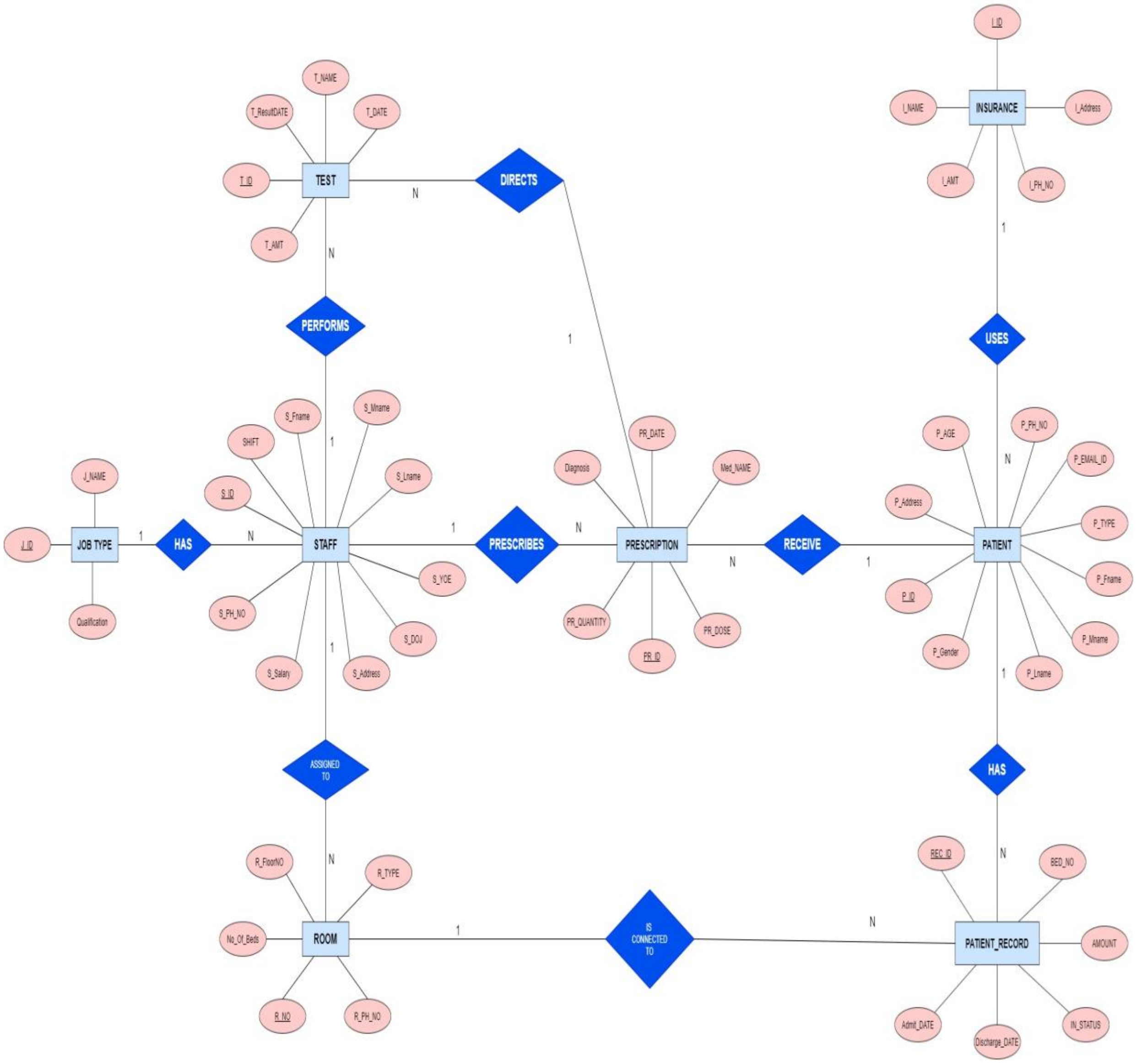
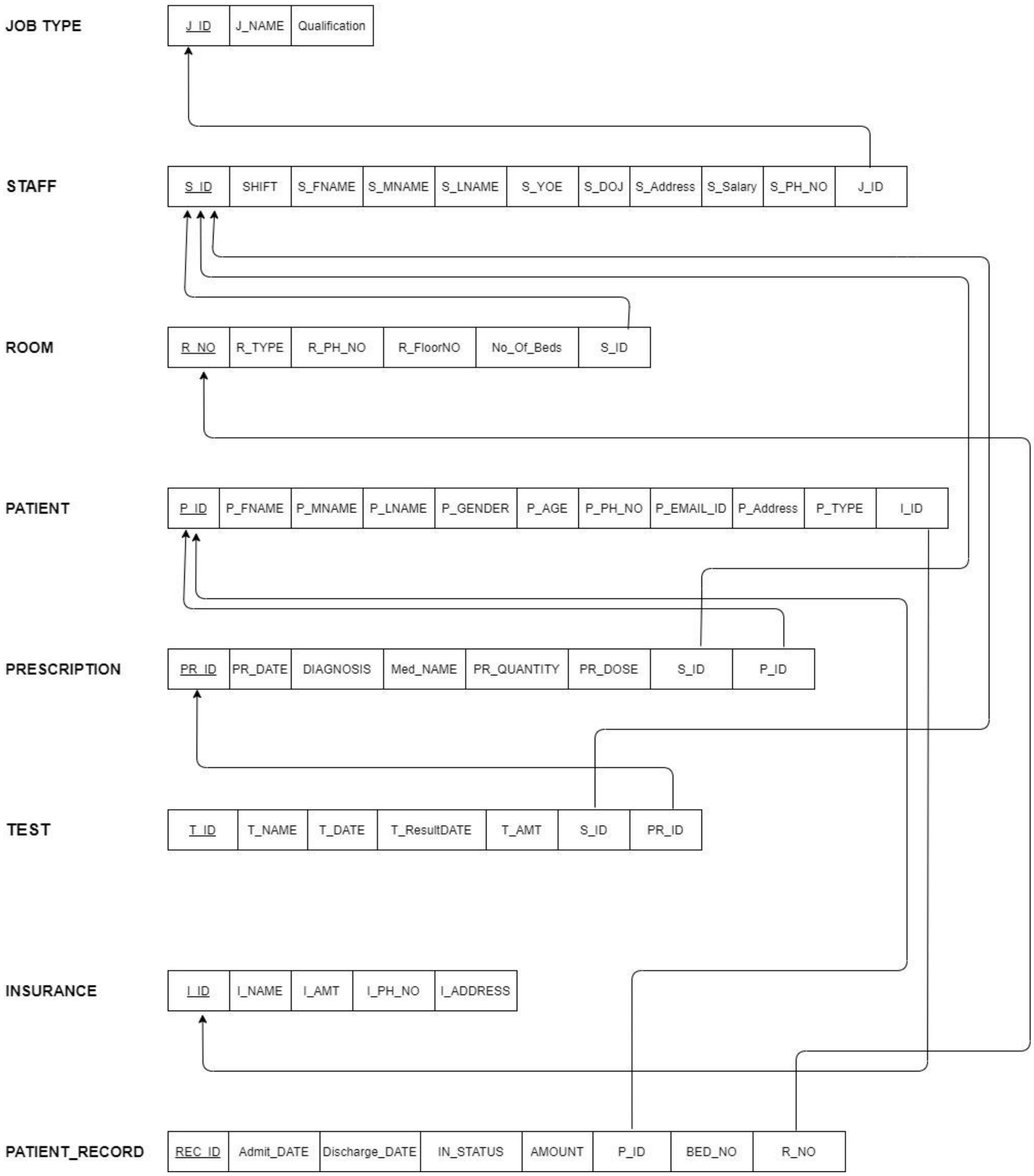


Fig.3.1: ER Diagram

HOSPITAL MANAGMENT SCHEMA



JOB TYPE

J_ID	J_NAME	Qualification

STAFF

S_ID	SHIFT	S_FNAME	S_MNAME	S_LNAME	S_YOE	S_DOJ	S_Address	S_Salary	S_PH_NO	J_ID

ROOM

R_NO	R_TYPE	R_PH_NO	R_FloorNO	No_Of_Beds	S_ID

PATIENT

P_ID	P_FNAME	P_MNAME	P_LNAME	P_GENDER	P_AGE	P_PH_NO	P_EMAIL_ID	P_Address	P_TYPE	I_ID

PRESCRIPTION

PR_ID	PR_DATE	DIAGNOSIS	Med_NAME	PR_QUANTITY	PR_DOSE	S_ID	P_ID

TEST

T_ID	T_NAME	T_DATE	T_ResultDATE	T_AMT	S_ID	PR_ID

INSURANCE

I_ID	I_NAME	I_AMT	I_PH_NO	I_ADDRESS

PATIENT_RECORD

REC_ID	Admit_DATE	Discharge_DATE	IN_STATUS	AMOUNT	P_ID	BED_NO	R_NO

Fig 3.2: Schema diagram of the Fig.3.1

JOBTYPE

(J_id, J_name, Qualification)

- The Relation is in 1NF as it has atomic valued attributes.
- The Relation is in 2NF since, every attribute is fully functionally dependent on the key.
- We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

STAFF

(S_id, S_Fname, S_Mname, S_Lname,S_PH_NO, S_DOJ, S_YOE,S_Address,S_Salary,SHIFT,J_id)

- The Relation is in 1NF as it has atomic valued attributes.
- The Relation is in 2NF since, every attribute is fully functionally dependent on the key.
- We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

INSURANCE

(i_id, i_name, i_ph_no, i_address, i_amount)

- The Relation is in 1NF as it has atomic valued attributes.
- The Relation is in 2NF since, every attribute is fully functionally dependent on the key.
- We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

PATIENT

(P_ID, P_Fname, P_Mname, P_Lname, P_Type, P_PH_NO, P_EMAIL_ID, P_AGE, P_Address, P_Gender, i_id)

- The Relation is in 1NF as it has atomic valued attributes.
- The Relation is in 2NF since, every attribute is fully functionally dependent on the key.
- We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

PRESCRIPTION

(PR_ID, Med_name, PR_DOSE, PR_DATE, PR_QUANTITY, Diagnosis, S_id, P_ID)

- The Relation is in 1NF as it has atomic valued attributes.
- The Relation is in 2NF since, every attribute is fully functionally dependent on the key.
- We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

TEST

(T_id, T_name, T_date, TR_date, T_amt, S_id, PR_ID)

- The Relation is in 1NF as it has atomic valued attributes.
- The Relation is in 2NF since, every attribute is fully functionally dependent on the key.
- We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

ROOM

(R_NO, R_PH_NO, R_TYPE, R_FloorNO, No_Of_Beds, S_id)

- The Relation is in 1NF as it has atomic valued attributes.
- The Relation is in 2NF since, every attribute is fully functionally dependent on the key.
- We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

PATIENT_RECORD

(rec_id, admit_date, discharge_date, in_status, amount, bed_no, R_NO, P_ID)

- The Relation is in 1NF as it has atomic valued attributes.
- The Relation is in 2NF since, every attribute is fully functionally dependent on the key.
- We observe that there is no transitivity in functional dependencies for the given relation. Hence the relation is in 3NF.

4. Implementation and Results

4.1. Introduction

Implementation involves the construction of a database according to the specification of a logical schema. This will include the specification of an appropriate storage schema, security enforcement, external schema and so on. Implementation is influenced by the choice of available DBMSs, database tools and operating environment. There are additional tasks beyond simply creating a database schema and implementing the constraints such as data must be entered into the tables, issues relating to the users and user processes need to be addressed, and the management activities associated with wider aspects of corporate data management need to be supported. In practice, implementation of the logical schema in a given DBMS requires a very detailed knowledge of the specific features and facilities that the DBMS has to offer. In an ideal world, and in keeping with good software engineering practice, the first stage of implementation would involve matching the design requirements with the best available implementing tools and then using those tools for the implementation. In database terms, this might involve choosing vendor products with DBMS and SQL variants most suited to the database which is to be implemented. There are many relational DBMSs, available such as Oracle Database, Microsoft SQL Server , MySQL, IBM DB2, IBM Informix and Microsoft Access, use SQL. In this project we used Oracle SQL developer create the following tables of Hospital Database.

4.2. Database Tables

Following tables table 4.1 to table 4.8 are the tables created for the schema diagram shown in figure 3.2.

Table 4.1: Job Type table description

Attribute	Type	Constraints
J_id	varchar2(20)	Primary key not null
J_name	varchar2(25)	not null
Qualification	varchar2(40)	

Table 4.2: Staff table description

Attribute	Type	Constraints
S_id	varchar2(20)	Primary key
S_Fname	varchar2(20)	
S_Mname	varchar2(20)	
S_Lname	varchar2(20)	
S_PH_NO	varchar2(20)	
S_DOJ	Date	
S_YOE	Int	
S_Address	varchar2(60)	
S_Salary	varchar2(10)	
SHIFT	varchar2(20)	

J_id	varchar2(20)	Foreign Key reference to Jobtype

Table 4.3: Room table description

Attribute	Type	Constraints
R_NO	Number	Primary key
R_PH_NO	varchar2(100)	
R_TYPE	varchar2(255)	
R_FloorNO	varchar2(50)	
No_Of_Beds	varchar2(50)	
S_id	varchar2(20)	Foreign Key reference to staff

Table 4.4: Patient table description

Attribute	Type	Constraints
P_ID	int	Primary key
P_Fname	varchar2(255)	
P_Mname	varchar2(255)	
P_Lname	varchar2(255)	
P_Type	varchar2(255)	
P_PH_NO	varchar2(100)	
P_EMAIL_ID	varchar2(255)	
P_AGE	varchar2(50)	
P_Address	varchar2(255)	
P_Gender	varchar2(255)	
i_id	number	Foreign Key reference to insurance

Table 4.5: Prescription table description

Attribute	Type	Constraints
PR_ID	Number	Primary key

Med_name	varchar2(255)	
PR_DOSE	varchar2(50)	
PR_DATE	varchar2(255)	
PR_QUANTITY	varchar2(50)	
Diagnosis	varchar2(255)	
S_id	varchar2(20)	Foreign Key reference to staff
P_ID	int	Foreign Key reference to patient

Table 4.6: Test table description

Attribute	Type	Constraints
T_id	Int	Primary key
T_name	varchar2(120)	
T_date	Date	
TR_date	Date	
T_amt	Int	
S_id	varchar2(20)	Foreign Key reference to staff
PR_ID	number	Foreign Key reference to prescription

Table 4.7: Insurance table description

Attribute	Type	Constraints
I_id	Number	Primary key
I_name	varchar2(120)	
I_ph_no	varchar2(120)	
I_address	varchar2(140)	
I_amount	int	

Table 4.8: Patient Record table description

Attribute	Type	Constraints
Rec_id	varchar2(20)	Primary key
Admit_date	Date	
Discharge_date	Date	
In_status	varchar2(20)	
Amount	Int	
Bed_no	Int	
R_NO	Number	Foreign Key reference to room
P_ID	Int	Foreign Key reference to patient

Following syntax shows for creating database table shown with an example for the Table 4.1

Database Creation:

```
drop table jobtype cascade constraints;
drop table staff cascade constraints;
drop table patient cascade constraints;
drop table Room cascade constraints;
drop table prescription cascade constraints;
drop table test cascade constraints;
drop table insurance cascade constraints;
drop table patient_record cascade constraints;
create table jobtype
(
J_id varchar2(20) primary key not null,
J_name varchar(25) not null,
Qualification varchar2(40)
);
create table staff
(
S_id varchar2(20) primary key,
S_Fname varchar2(20),
S_Mname varchar2(20),
S_Lname varchar2(20),
S_PH_NO varchar2(20),
S_DOJ date,
S_YOE int,
S_Adrress varchar2(60),
S_Salary varchar2(10),
SHIFT varchar2(10),
J_id varchar2(20),
foreign key (j_id) references jobtype(j_id)
);
create table insurance
(
i_id number primary key,
i_name varchar2(120),
i_ph_no varchar2(120),
i_address varchar2(140),
i_amount int
);
CREATE TABLE Patient(
   P_ID int primary key,
   P_Fname varchar2(255),
   P_Mname varchar2(255),
   P_Lname varchar2(255),
   P_Type varchar2(255),
   P_PH_NO varchar2(100),
   P_EMAIL_ID varchar2(255),
   P_AGE varchar2(50),
   P_Address  varchar2(255),
   P_Gender varchar2(255),
   i_id number,
   foreign key (i_id) references insurance(i_id)
);
CREATE TABLE prescription(
  PR_ID number primary key,
  Med_name varchar2(255) default NULL,
  PR_DOSE varchar2(50) default NULL,
  PR_DATE varchar2(255),
  PR_QUANTITY varchar2(50) default NULL,
  Diagnosis varchar2(255) default NULL,
  S_id varchar2(20),
  P_ID int,
  foreign key (S_id) references staff(S_id),
  foreign key (P_ID) references patient(P_ID)
);
create table test(
T_id int,
T_name varchar(120),
T_date date,
TR_date date,
T_amt int,
primary key(T_id),
S_id varchar2(20),
foreign key (S_id) references staff(S_id),
PR_ID number,
foreign key (PR_ID) references prescription(PR_ID)
);
CREATE TABLE Room (
  R_NO number PRIMARY KEY,
  R_PH_NO varchar2(100),
  R_TYPE varchar2(255) ,
  R_FloorNO varchar2(50) ,
  No_Of_Beds varchar2(50) ,
  S_id varchar2(20),
  foreign key (S_id) references staff(S_id)
);
create table patient_record
(
rec_id varchar2(20) primary key,
admit_date date,
discharge_date date,
in_status varchar2(20),
amount int,
bed_no int,
R_NO number,
foreign key (R_NO) references Room(R_NO),
P_ID int,
foreign key (P_ID) references Patient(P_ID)
);
```

4.3. Results

With this report we have attached two word document which contain all sql and pl/sql queries and their outputs.

References

[1] https://www.freeprojectz.com/entity-relationship/hospital-management-system-er-diagram

[2]https://www.google.com/url?sa=t&source=web&rct=j&url=https://www.scribd.com/document/349571448/Hospital-Management-System&ved=2ahUKEwiowb7Gv6rwAhW0H7cAHZ8ID28QFjAGegQICRAC&usg=AOvVaw09ecP5phsv6nFtKrZfR_t_&cshid=1619941431701

[3] https://github.com/vanshikadarbari/Hospital-database-management-system/blob/master/SQL%20Queries.txt