

Table of Contents

Table of Figures	3
Table Of Tables.....	4
Acknowledgement.....	5
Introduction	6
Part 1.1: Poster Presentation	8
Part 1.2: Introduction to Database Design & Development	9
Types of Databases	9
Functions of DBMS.....	9
Database Design	10
Keys.....	11
Relationships in Database	13
Normalization	15
Entity Relationship Diagram (ERD)	22
.....	23
ER Diagram of Norman Hospital.....	24
Enhanced Entity Relational Diagram	27
EER Diagram.....	29
ER Diagram	30
Part 1.3 Mockups and Report.	31
Report About the Prototype System	31
Part 1.4 Evaluation Report	38
User Requirements.....	38
Part 2.1 Physical Relational Database.....	40
Part 2.2 Privilege Matrix	54
How To Grant Different User Permissions	54

.....	58
Part 2.2 Inner Join, Outer Join, Full Join.....	59
Trigger	79
Before Update Trigger	79
Before Update Trigger	80
Part 3.1: Explain the Data Validation & Verification Process.	83
Data Verification.....	84
Steps of Verification	84
Outputs of Verification	85
Data Validation	86
Steps of Validation	87
Outputs of Validation	88
Part 3.2: Error Report	90
Common Data Errors.....	90
Minimum Error.....	92
Alter Table Error.....	93
Database Error	93
Sum Error	94
Delete Error.....	94
Part 4.1: Produce a user and Technical Manual.....	95
Technical Manual.....	95
User Manual	96
Part 4.2: Future Improvements.	102
Conclusion	103

Table of Figures

Figure 1:Sample Database Design	10
Figure 2:Primary Key in Database	11
Figure 3:Foreign Key Sample Image	12
Figure 4:One to One Relationship	13
Figure 5:One to Many Relation	13
Figure 6:Many to Many Relationship	14
Figure 7:Symbols of ER Diagram	22
Figure 8:Sample ER Diagram	23
Figure 9:Create a Database	27
Figure 10:Inserting Table in to Database	27
Figure 11:Inserting Details to Table	28
Figure 12:Sample Database	28
Figure 13:EER Diagram	29
Figure 14:Entity Relationship Diagram	30
Figure 15:Mockups User Id	31
Figure 16:Main Page of DBMS	32
Figure 17:Patient Information Page	32
Figure 18:To Make Appointments	33
Figure 19:Norman Hospital Doctors Information	33
Figure 20:In this picture the doctors will know about their patient	34
Figure 21:This about the house officers	34
Figure 22:Patient visit time information	35
Figure 23:Ward information of patients	35
Figure 24:Admission of every patient	36
Figure 25:ICU patients Information	36
Figure 26:Discharge information of patients	37
Figure 27:Pharmacy details	37
Figure 28:before Forward Engineering	40
Figure 29:Example	40
Figure 30:Previleges matrix	55

Figure 31:Account limits	56
Figure 32:Privilege's user and password	56
Figure 33:Previlege roles	57
Figure 34:Password previleges	57
Figure 35:Previlege actions	58
Figure 36:Previlege access denied	58

Table Of Tables

Table 1:Table of User Information	15
Table 2:Patient Information	17
Table 3:1st normalization Doctor table	17
Table 4:Prescription	17
Table 5:Hospital Admit	17
Table 6:Patient Visit	18
Table 7:Patient Info	19
Table 8:Patient Appointment	19
Table 9:Doctor Info	19
Table 10:Consultation Prescription	19
Table 11:Admission Appointment	20
Table 12:Admission Counter	20
Table 13:Ward Information	20
Table 14:Doctor patient visit	20
Table 15:House officer info	20
Table 16:Discharge info	21
Table 17:ICU Info	21
Table 18:Pharmacy information	21

Acknowledgement

The success of this assignment required a lot of guidance and assistance from my classmates, lectures and I am extremely thankful to get this all along the completion of my assignment work.

Whatever I have done is only due to such guidance and assistance and I would not forget to thank them. I respect and thank Mr. M. Kartheeban for giving me an opportunity to do this assignment on time, I extremely grateful to him for providing such a nice support and guidance.

I am really grateful because I managed to complete this assignment within the time given by my lecturer. This assignment cannot be completed without the effort and co-operation from my class mates. I would like to express my gratitude to my friends and respondents for support and willingness to spend some time with me.

A. Stany Nirojan

Introduction

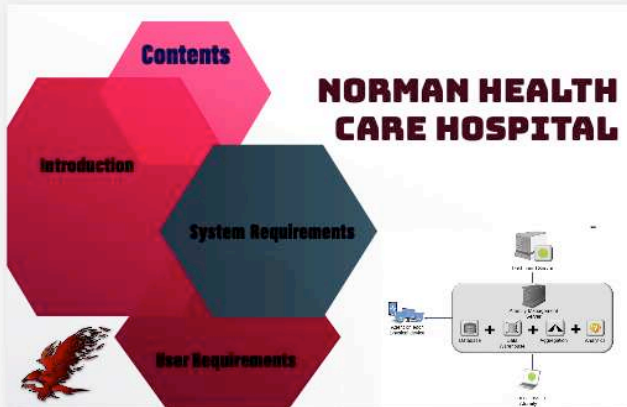
Databases are at the Centre of most information systems in everyday use, therefore it is important that they are designed and built using appropriate methods to ensure that they meet users' requirements while being robust and maintainable.

A database system is usually regarded as the database which contains related tables of data maintained by a database management system (DBMS), along with applications that provide controlled access to the database.

And able to identify how we qualified in database development and how to solve the problems in easy ways and implement the solution. Finally, I have prepared the documentation to define what we achieve.

L01

Part 1.1: Poster Presentation



Norman Hospital is a public hospital providing treatment since 2000. Hospital uses manual documentation and manual charge entry for all works. So, we need to produce a Database System for their data entry.

User Requirements For a Hospital Database System

- To computerize all details regarding patient and hospital.
- To schedule patient appointments with doctors making it convenient for both.
- To schedule specialized and emergency services to ensure complete and effective utilization.
- To maintain patient records and vitals

System Requirements

- At least 2 physical drives (3 or more recommended).
- Initial disk space of 160 MB (required for creating the TADDIM schema).
- Disk space for discovery data. Use either of these formulas to estimate disk space requirement (assuming Level 3 discovery)
- Additional disk space for ongoing growth. Plan for 10% growth weekly.

THANKS!

Part 1.2: Introduction to Database Design & Development

Databases are the main in the most of the system in the modern world. It is designed and buildup with the user requirements, software requirements and hardware requirements. A database which contains databases, tables of data and maintain by the Database Management System (DBMS) with applications controlled that databases.

Databases are like Backend work process because the Frontend (Applications) are direct the databases.

Types of Databases

- **Transactional Databases** – Main function of this database is to add new data, change existing data, delete existing data. Eg: Client – Server Database
- **Decision Support System (DSS) Database**– They support decisions.
- **Hybrid** – It is a mixture of OLTP (On -line transaction processing) and data warehouse throughout requirements.

Functions of DBMS

- **Data Storage** – Database can access by many users so they can storage, retrieve and update data.
- **Retrieve Data**
- **Update Data**
- **Data Communication Interfaces** – Telecommunications can allow to transactions of data through remote so, DBMS must provide an interface to assist the transactions.

Database Design

- Collecting Information
- Strategy & Planning Process
- Design & Implement Process

Data collecting is the main purpose in creating a database system because users will tell about their requirements (They can have a imagination that is how our system would be like). As a Database designer we need to consider about the software requirements and hardware requirements.

Planning will make the system more accurate and we will find the new needs to fulfill the customers(users).

Designing will allow to create a logical and physical design and able to identify what are additional use to make the system effectively and after that the development phase will start.

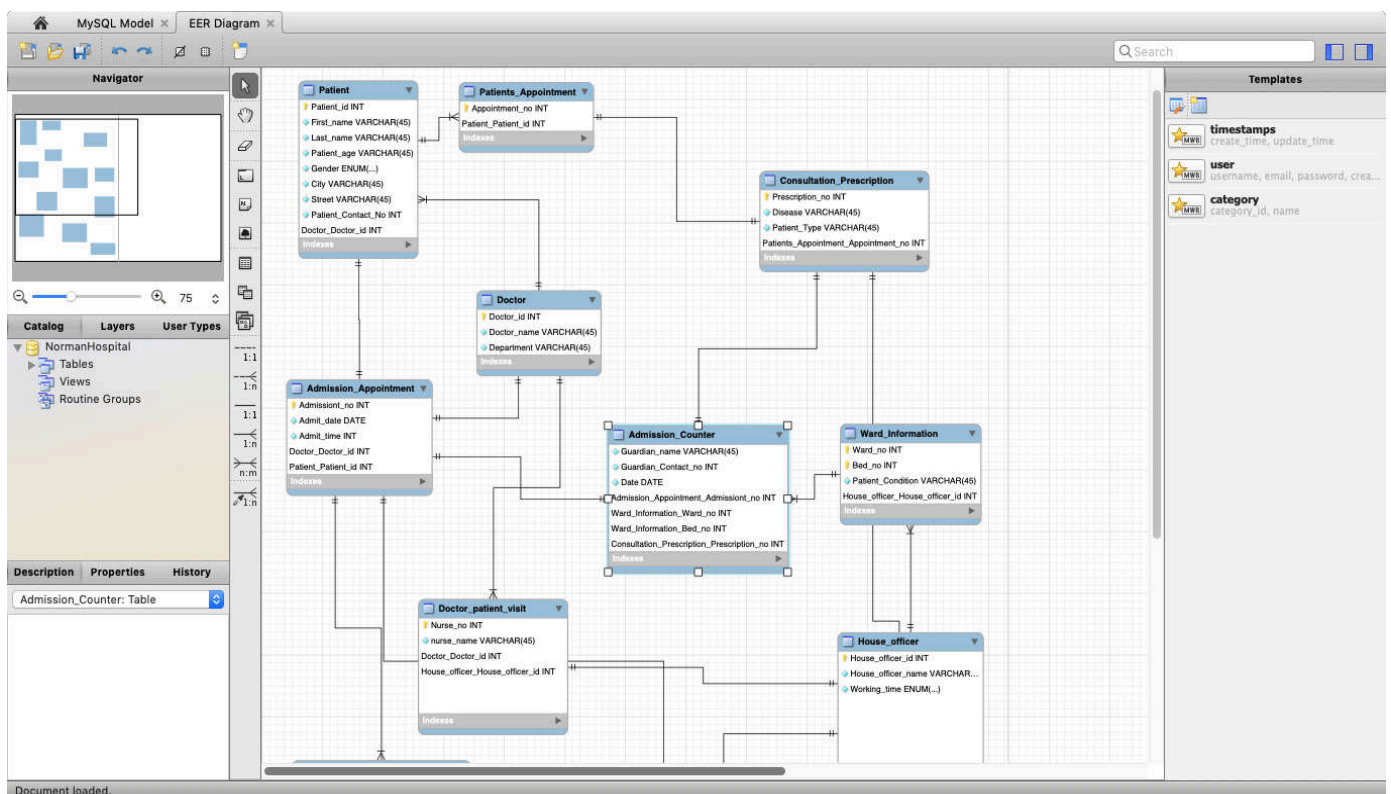


Figure 1: Sample Database Design

Keys

Primary Key

Primary key is used to uniquely identify the record in the tables. A table wants a unique identification because there is no way to find a record without possible of finding more than one record. So, primary key is a heart of a table.

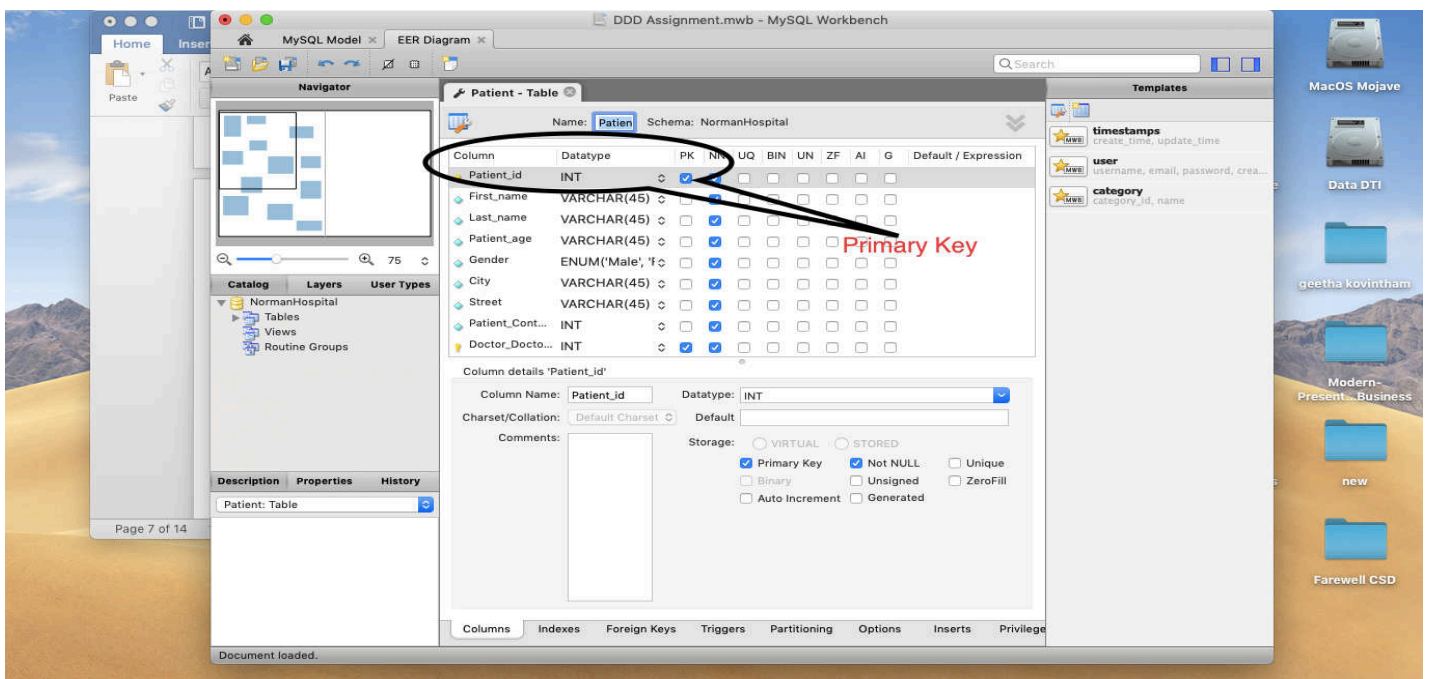


Figure 2:Primary Key in Database

Foreign Key

Foreign key is the copy of the primary keys. If you get a primary key from a table and when you put that key into another table it will be as foreign key. After that we will derive all data to that primary key located table to foreign key located table.

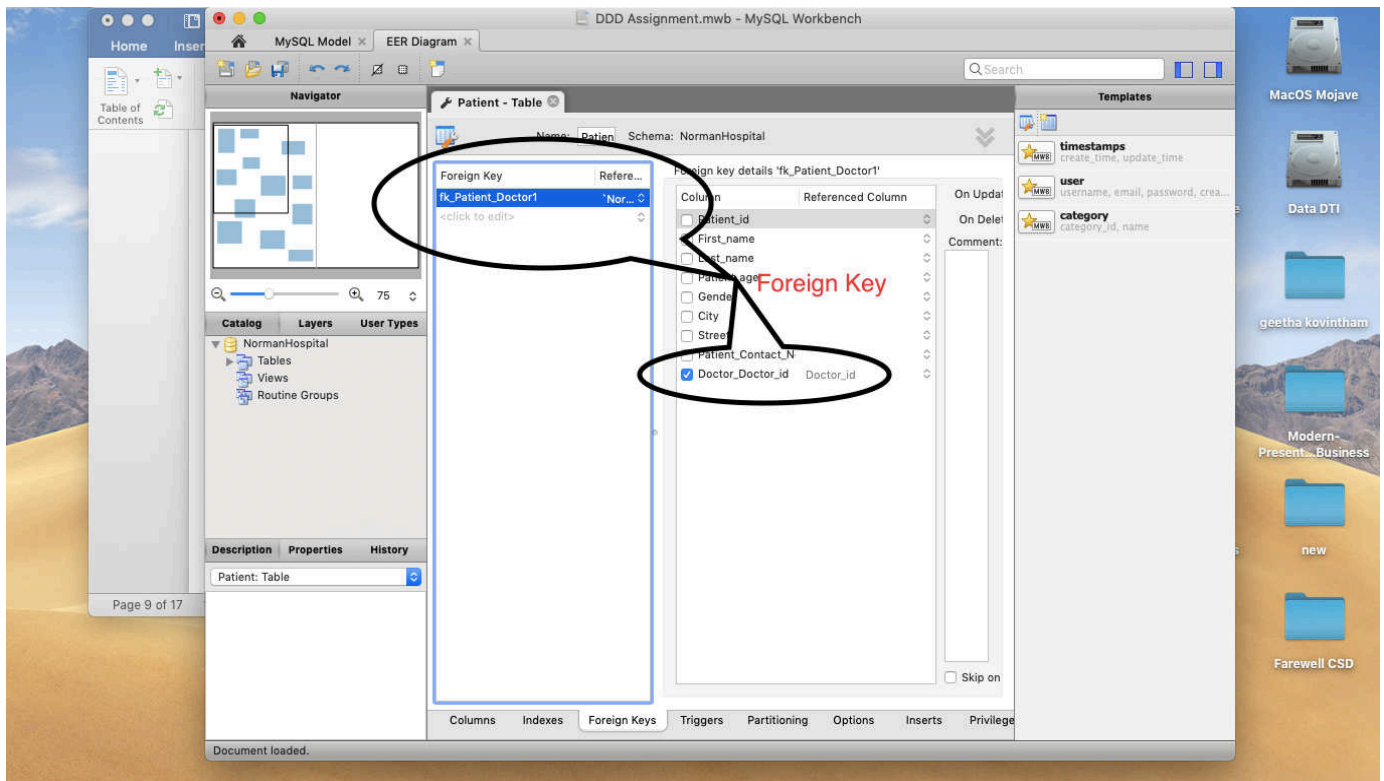


Figure 3: Foreign Key Sample Image

Relationships in Database

All entities in a database will be connected through the relationship. They are:

One – to – One

This relationship implements exactly one entry in both tables. One to one relationship is a type of 4th Normalization.

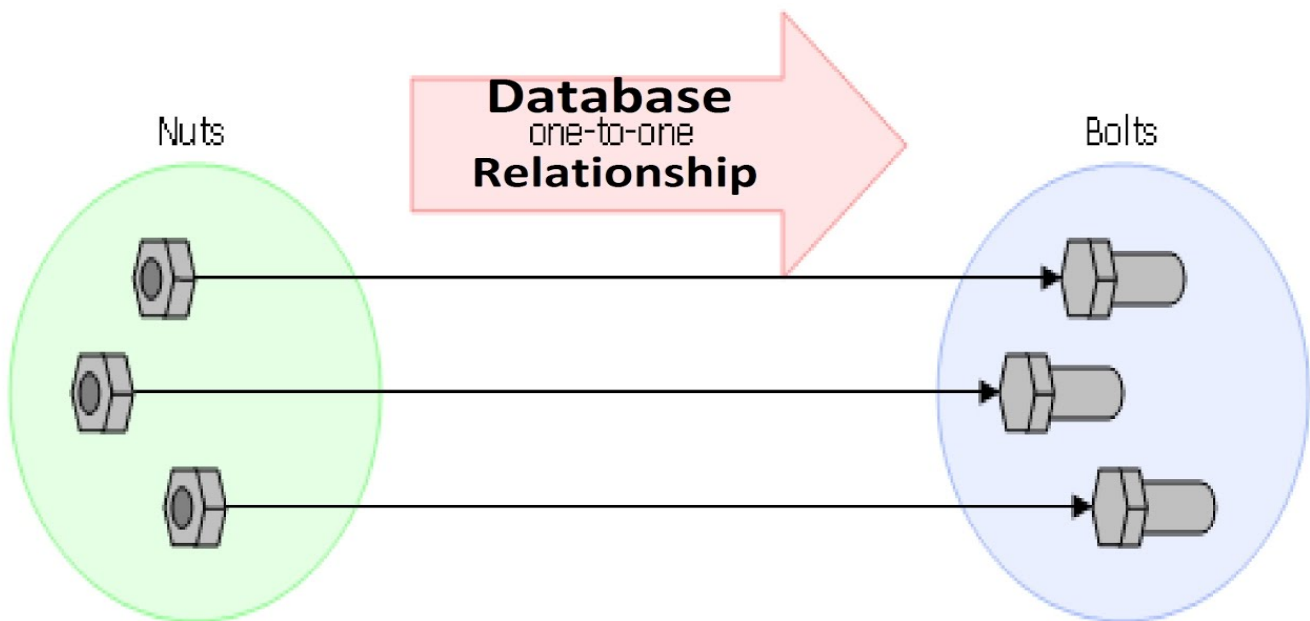


Figure 4: One to One Relationship

One – to – Many

One to many implements one entry to many entries between tables. If you take doctors and patients, one doctor will treat many patients so doctor is in one and patients will act as many.

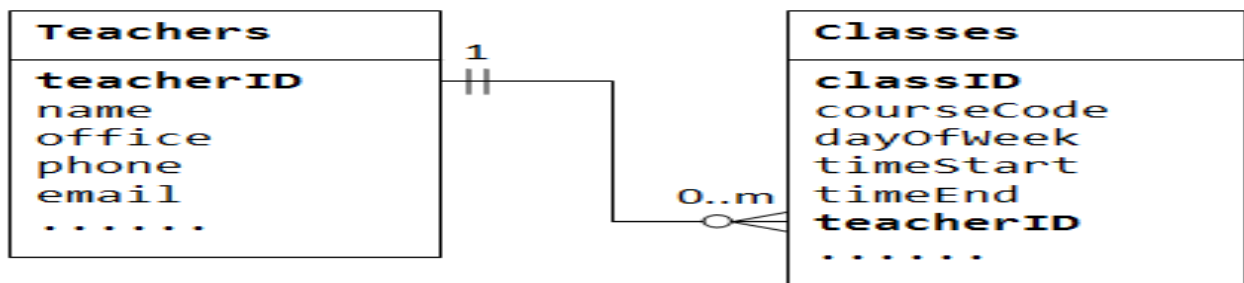


Figure 5: One to Many Relation

Many – to – Many

Every record in a table there are many possible records in another related table. If we take patients entity and ward entity many patients admit in many wards at Norman hospital so they have a many to many relationships.

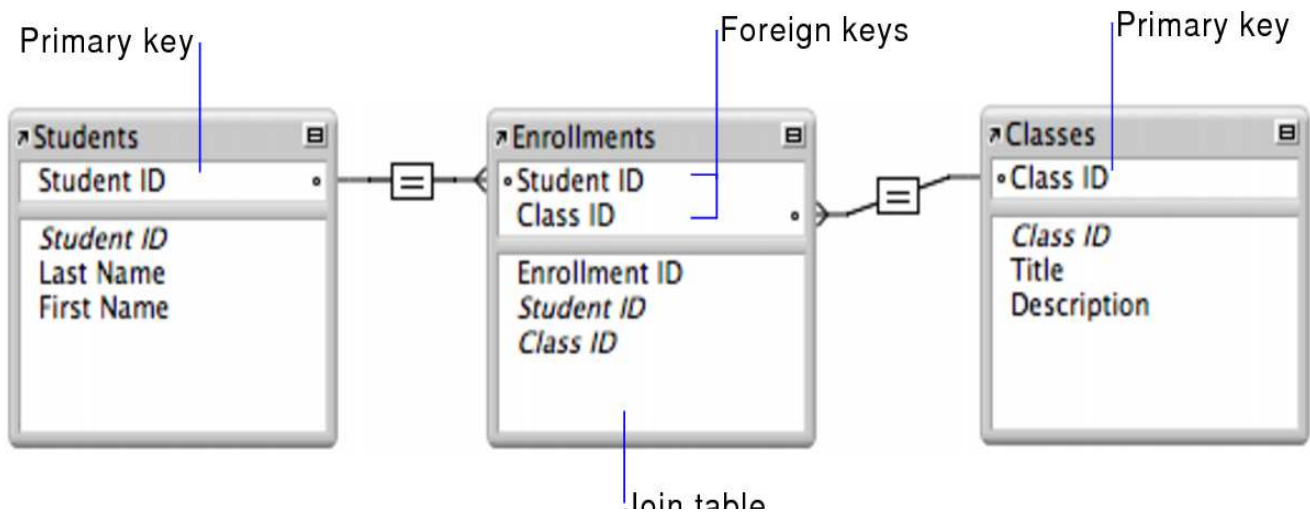


Figure 6: Many to Many Relationship

Normalization

Normalization is a process of determining the redundancy exist in the tables (multiple values). If we want to reduce the redundancy, the main is Primary Key to Foreign Key. If we assign a primary key in one table and put that same primary key as foreign key in other table then, that table will derive the table details.

Benefits of Normalization

- Storage data will reduce
- The data will be organized correctly
- Easy to search the fields.

Types of Normalization

- 1st Normal Form
- 2nd Normal Form
- 3rd Normal Form
- BCNF

0 Normal Form

Details Collected from Norman Hospital For Database System

Table 1:Table of User Information

Patient_id	Doctor department
Patient name	File no
Patient age	Date visited
Patient address	Description
Patient contact no	Disease
Appointment no	Drugs
Doctor_id	Gender
Doctor name	Parent/guardian name
House officer id	Guardian Contact no
House officer name	Address
Working time	Ward number
Reference no	Bed no

1st Normal Form

Eliminate the repeating groups such as similar records in all tables will be become unique by primary key in each of the tables and the other tables are depend on the primary key. Rules are:

- Eliminate the repeating groups in individual tables.
- Create a separate table for each set of related data.
- Identify each set of related data with the primary key.
- Primary key will uniquely identify attribute value (Patient_id, Doctor_id).

Patient

- Patients id
 - Patient name
 - Patient Address
 - Patient age
 - Patient Contact no
 - Appointment no
 - Doctor id
 - Doctor name
 - Doctor Department
 - Patient id
 - Patient name
 - Doctor name
 - Disease
 - Drugs
- We will see the first normalization in this tables. In the Patients entity for every patient the doctor id, name, doctor name will be repeat. So, we want to create an entity names doctor and include these columns:

Patient

Table 2:Patient Information

Patient name
Patient age
Patient Address
Patient Contact no
Appointment no
Doctor name
Doctor department

Doctor

Table 3:1st normalization Doctor table

Doctor name
Doctor Department

Prescription

Table 4:Prescription

Patient name
Patient age
Doctor name
disease
Name of the drugs

Hospital Admit

Table 5:Hospital Admit

Patient name
Patient age
Patient gender
Doctor name
Parent/guardian name
Parent/guardian contact no
address
Date and time
Ward number

Bed number
House officer

Patient Visit

Table 6:Patient Visit

Date
Time
Patient condition
Drugs
House officer
Nurse no

2nd Normal Form

All non-key values fully function on the primary key. No partial dependencies are allowed. A partial dependency exists when a field is fully dependent on a part of a composite primary key.

Composite primary key is a candidate key, that is when two primary keys in a table so one is primary key and another one is composite primary key.

Patient

Table 7:Patient Info

Patient id <<PK>>
First name
Last name
Patient age
Gender
City
Street
Patient Contact no
Doctor id <<FK>>

Patient Appointment

Table 8:Patient Appointment

Appointment no <<PK>>
Patient id <<FK>>

Doctor

Table 9:Doctor Info

Doctor id <<PK>>
Doctor name
Department

Consultation Prescription

Table 10:Consultation Prescription

Prescription no <<PK>>
Disease
Patient Type
Patient Appointment no <<FK>>

Admission Appointment

Table 11: Admission Appointment

Admission no <<PK>>
Admit Date
Admit Time
Doctor id <<FK>>
Patient id <<FK>>

Admission Counter

Table 12: Admission Counter

Guardian name
Guardian Contact no
Date
Admission no <<FK>>
Ward no <<FK>>
Bed no <<FK>>
Prescription no <<FK>>

Ward Information

Table 13: Ward Information

Ward no <<PK>>
Bed no <<PK>>
Patient Condition
House officer id <<FK>>

Doctor Patient Visit

Table 14: Doctor patient visit

Nurse no <<PK>>
Nurse name
Doctor id <<FK>>
House officer id <<FK>>

House officer Information

Table 15: House officer info

House officer id <<PK>>
House officer name
Working time

Discharge Information

Table 16:Discharge info

Reference no <<PK>>
Date
Time
Admission no <<FK>>

ICU Information

Table 17:ICU Info

ICU officer id <<PK>>
Record no <<PK>>
Patient Condition
Admission no <<FK>>
House officer id <<FK>>

Pharmacy Information

Table 18:Pharmacy information

Pharmacy customer no <<PK>>
Drug Description
Reference no <<FK>>
Prescription no <<FK>>

Entity Relationship Diagram (ERD)

Entity Relationship Diagram is a visual representation of data and which shows how the data are connected or related to each other. ER diagrams are frequently used during the design stage of a development process in order to identify different system elements and their relationships with each other.

ER Diagram is defined by the set of entities and an entity is represented by a set of attributes.

In the Norman Hospital system, we already did the normalization up to 3rd and made a plan of 12 entities. Before that we will see about the Entity Relational Diagram Symbols and their meanings.

ER Diagram Symbols and Notations



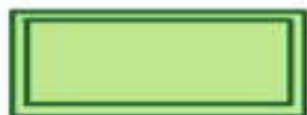
Entity



Attribute



Relationship



**Weak
Entity**



**Multivalued
Attribute**



**Weak
Relationship**

Figure 7: Symbols of ER Diagram

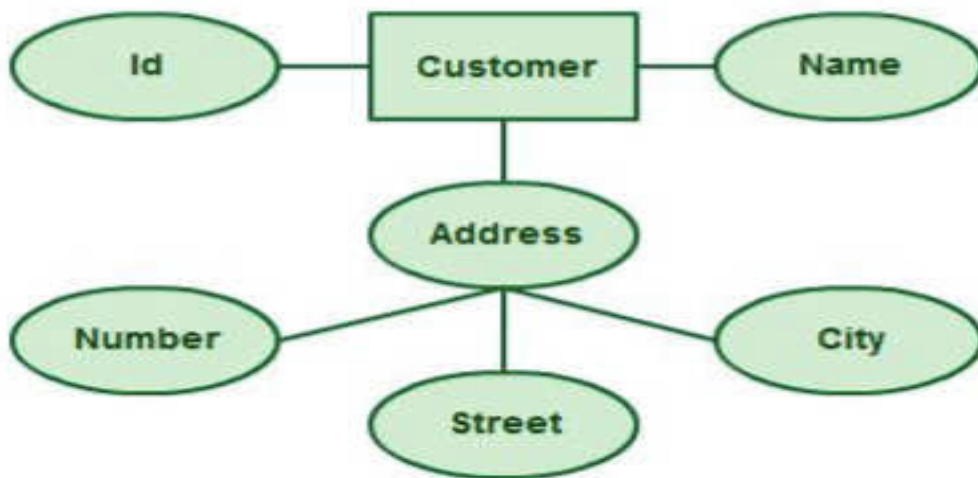
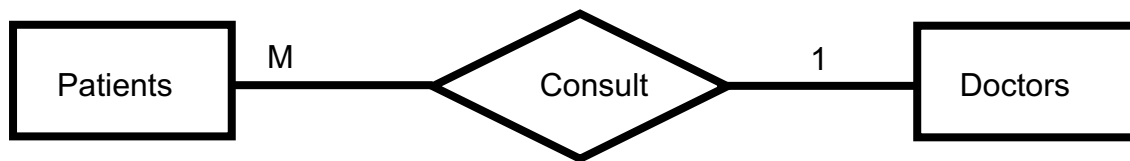


Figure 8: Sample ER Diagram

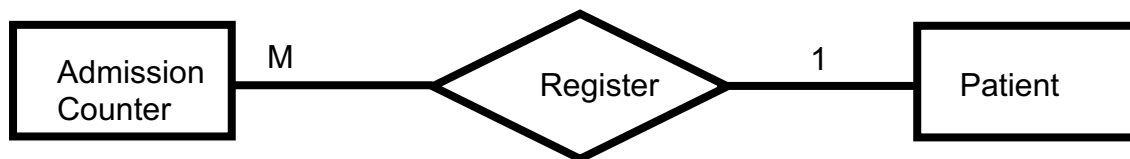
ER Diagram of Norman Hospital

We have separated the Norman hospital system informations using the Normalizations so easy to make the ER diagram.

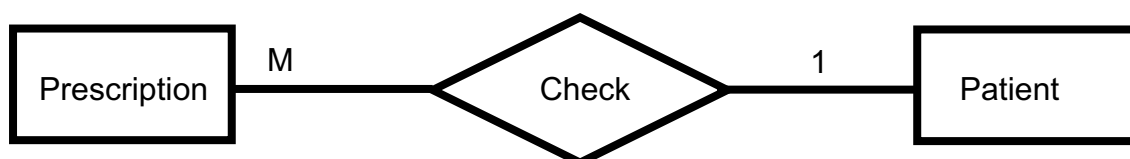
First, we need to prepare the entity diagrams for the informations we have made. So, I have 13 entities and these entities are patient, doctor, patient appointment, ward admission, ward information, visit time, file information, house officer, pharmacy, discharge, ICU, admission counter and prescription.



The relationship between doctor and patients is one to many because one doctor consults many patients.



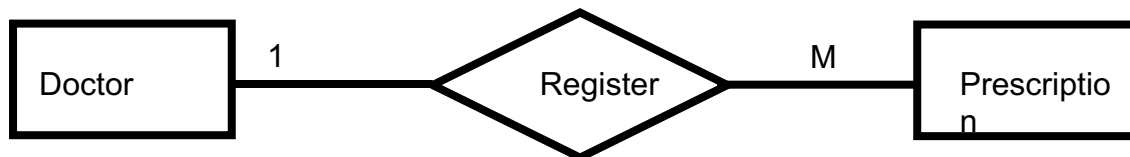
The relationship between admission counter and patients is many to one because many patients register in one admission counter.



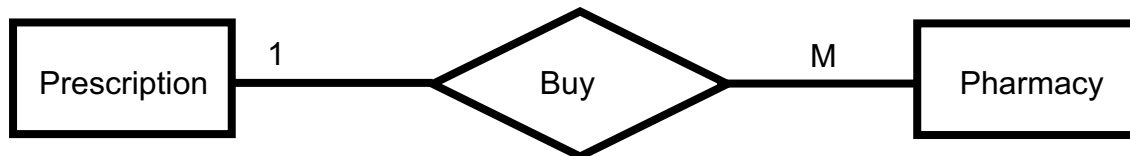
The relationship between patients and prescription is many to one because one patient will have many prescriptions.



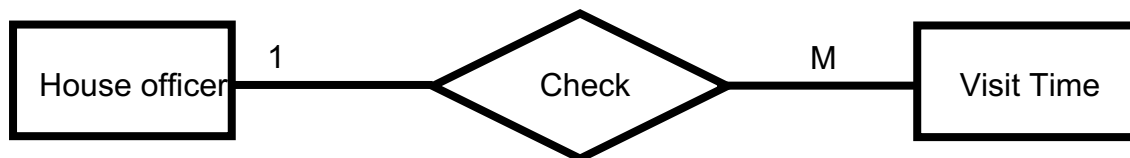
The relationship between patient appointment and patient is many to one because one patient will have many appointments.



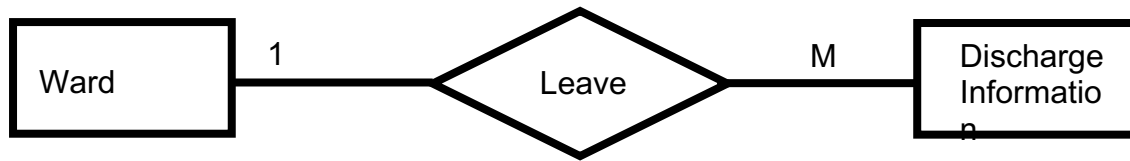
The relationship between doctor and prescription is one to many because one doctor will have many prescriptions.



The relationship between prescription and pharmacy is one to many because one prescription will accept in many pharmacies.



The relationship between House officer and doctor visit time is one to many because one house officer will visit in many wards.



The relationship between Ward and discharge is one to many because one ward will have many discharges.

Enhanced Entity Relational Diagram

Enhanced entity-relationship models, also known as extended entity-relationship models, are advanced database diagrams very similar to regular ER diagrams. Enhanced ERDs are high-level models that represent the requirements and complexities of complex databases.

A well-designed EER will help you build storage systems that are long-lasting and useful.

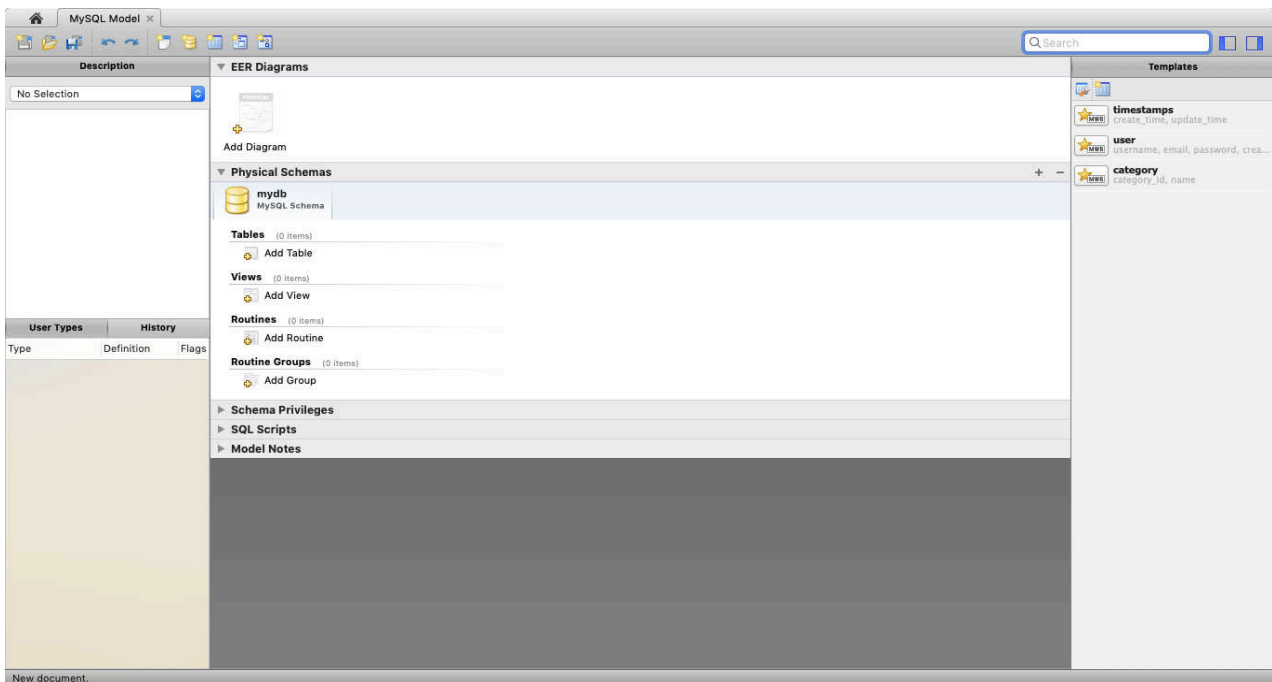


Figure 9: Create a Database

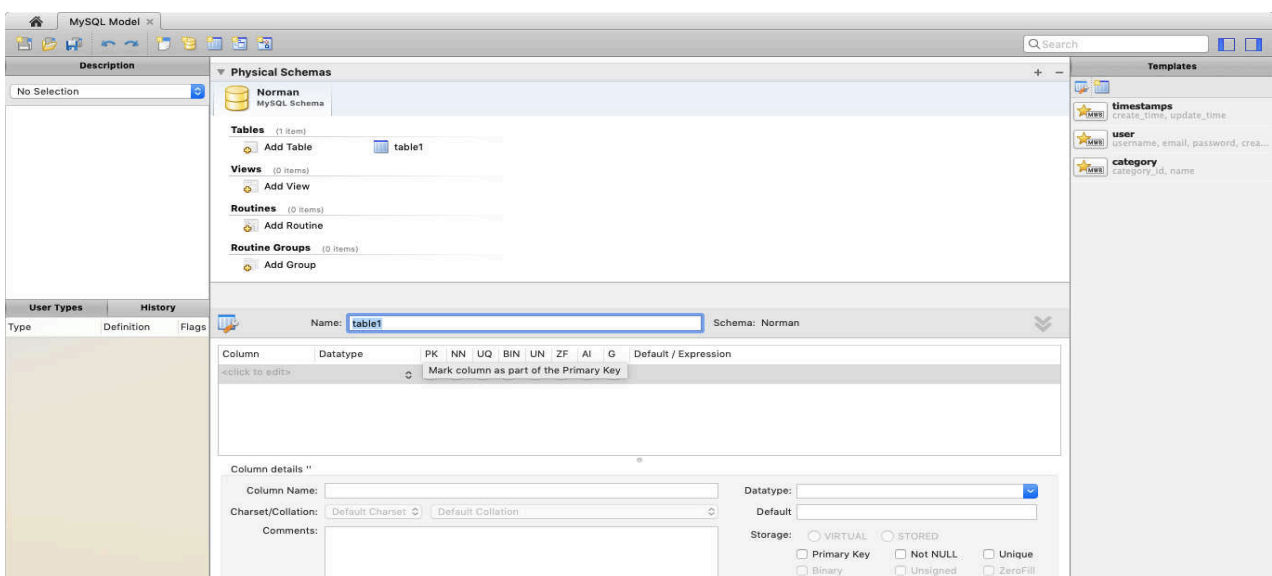


Figure 10: Inserting Table in to Database

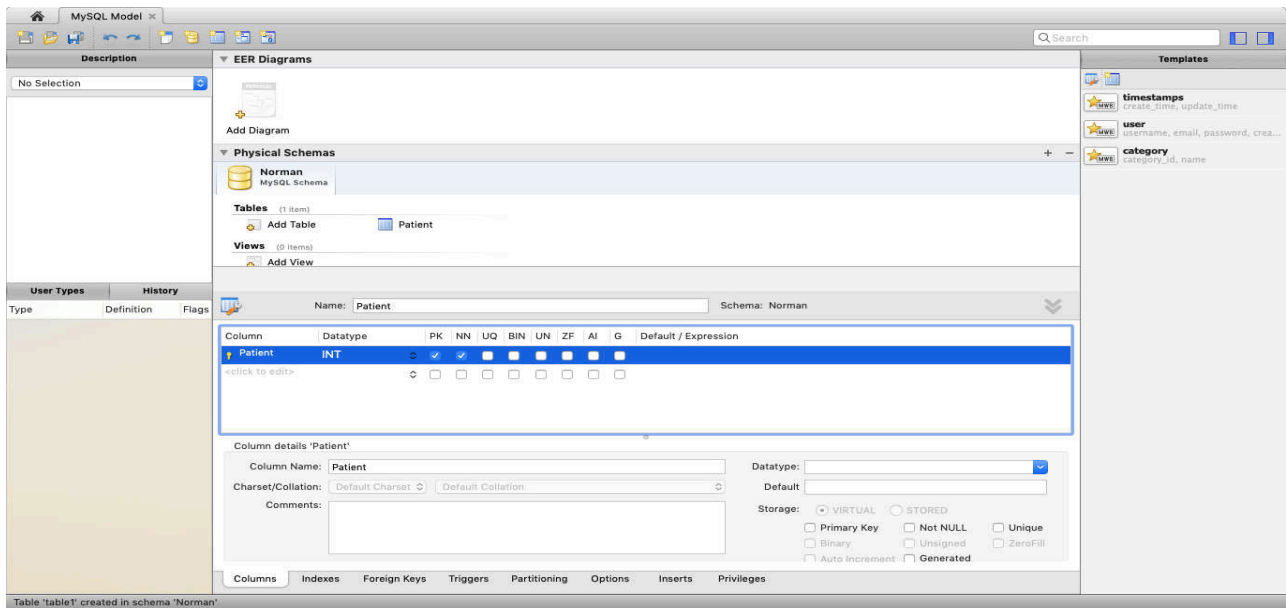


Figure 11: Inserting Details to Table

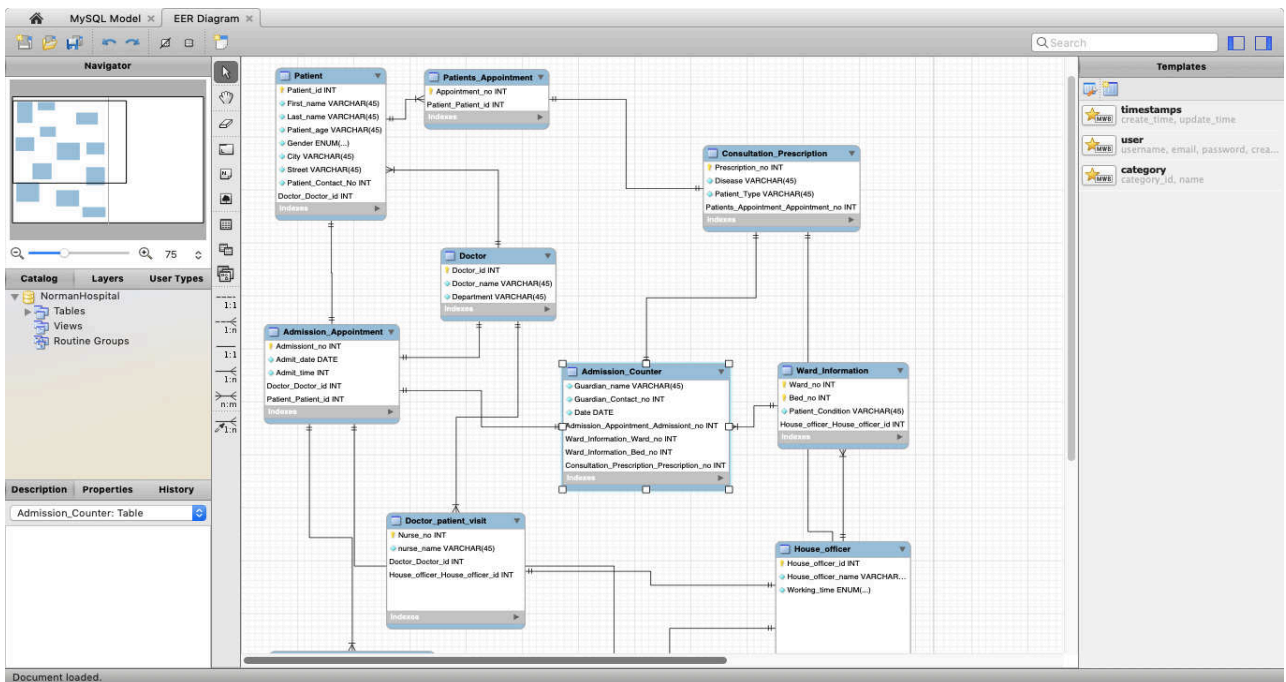


Figure 12: Sample Database

EER Diagram

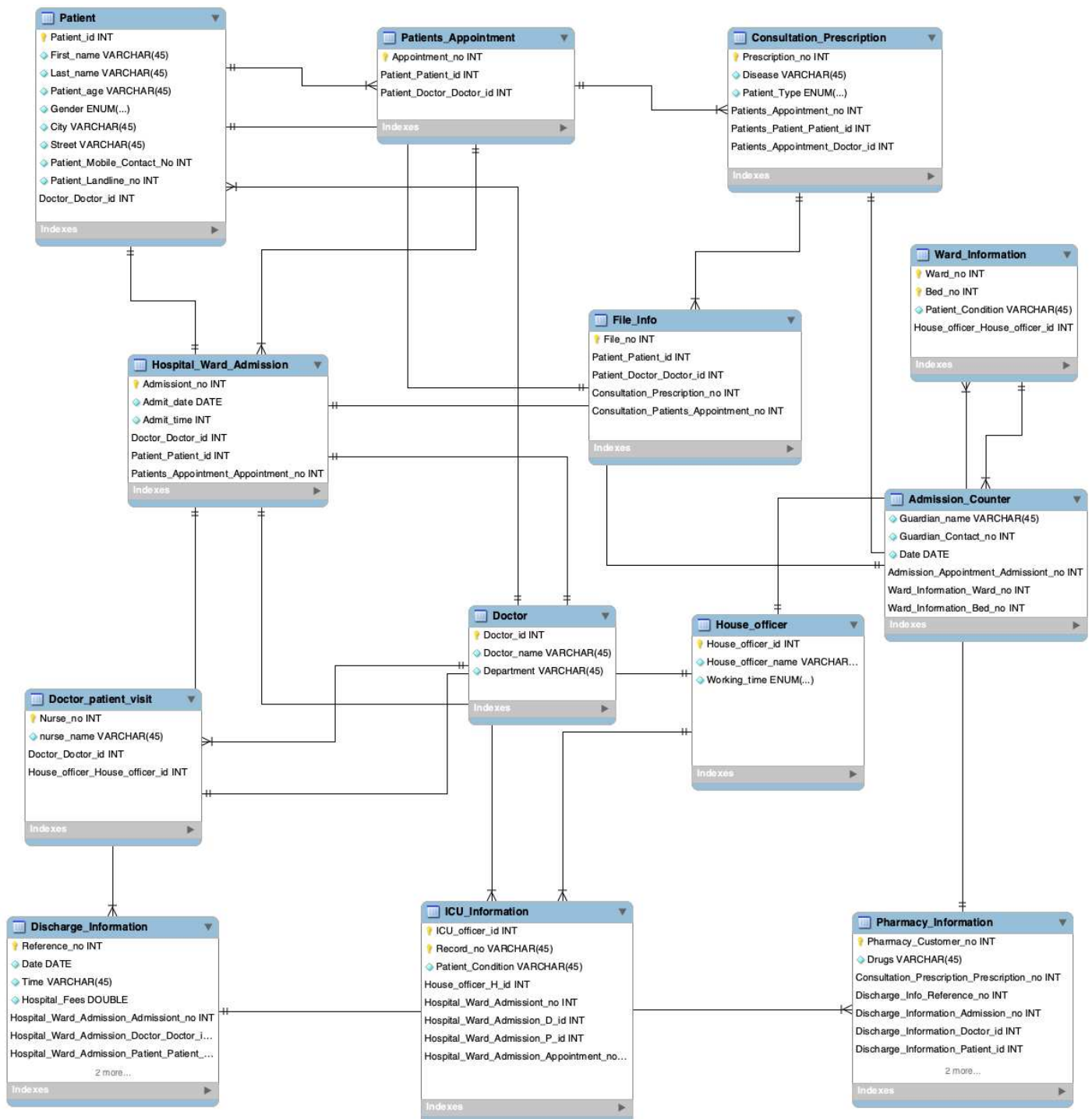


Figure 13:EER Diagram

ER Diagram

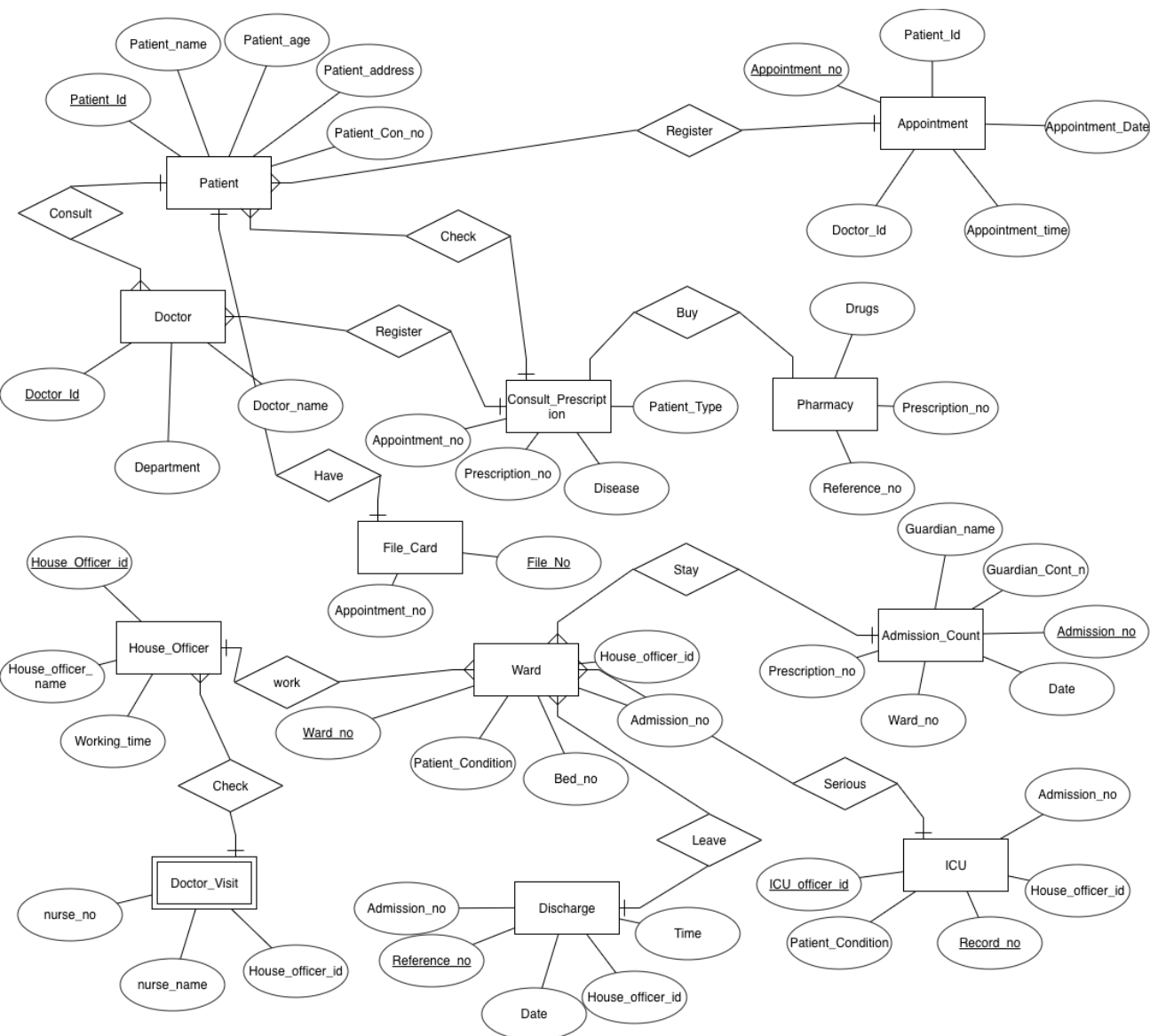


Figure 14: Entity Relationship Diagram

Part 1.3 Mockups and Report.

Report About the Prototype System

Prototype is used to demonstrate the system like an original view for the users. So, the users will easily understand how the real system will look like. Adobe XD is a software tool used to create the prototype of this Database management System

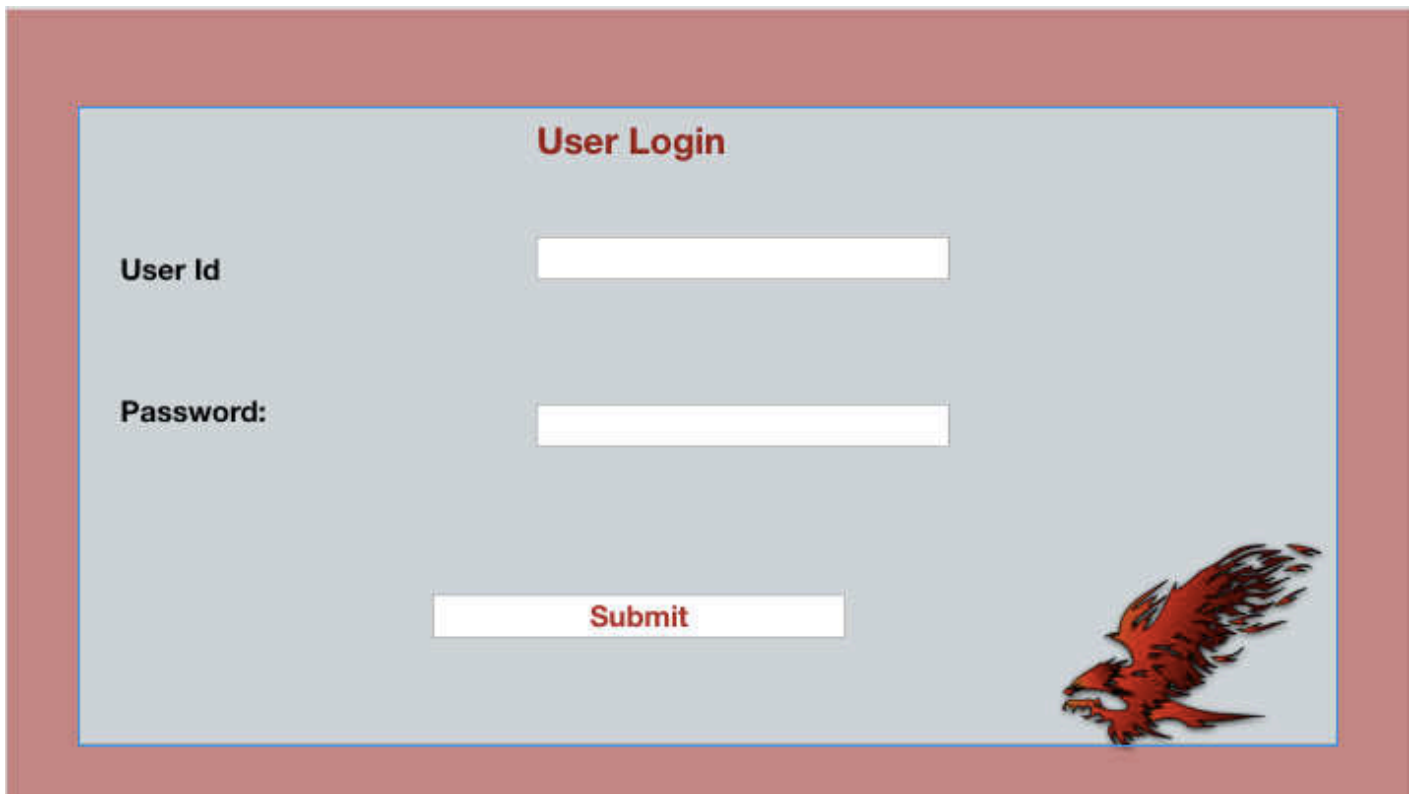
A mockup of a user login interface. It features a light blue rectangular area centered on a reddish-brown background. At the top of the light blue area, the text "User Login" is written in a bold, dark red font. Below this, there are two white input fields. The first field is preceded by the label "User Id" in a bold black font. The second field is preceded by the label "Password:" in a bold black font. Below the input fields, there is a white rectangular button with the word "Submit" written in a bold, dark red font. In the bottom right corner of the light blue area, there is a stylized red eagle with its wings spread, facing left.

Figure 15:Mockups User Id

In this page the user of the Norman hospital should enter their user id and password to go through the database system.

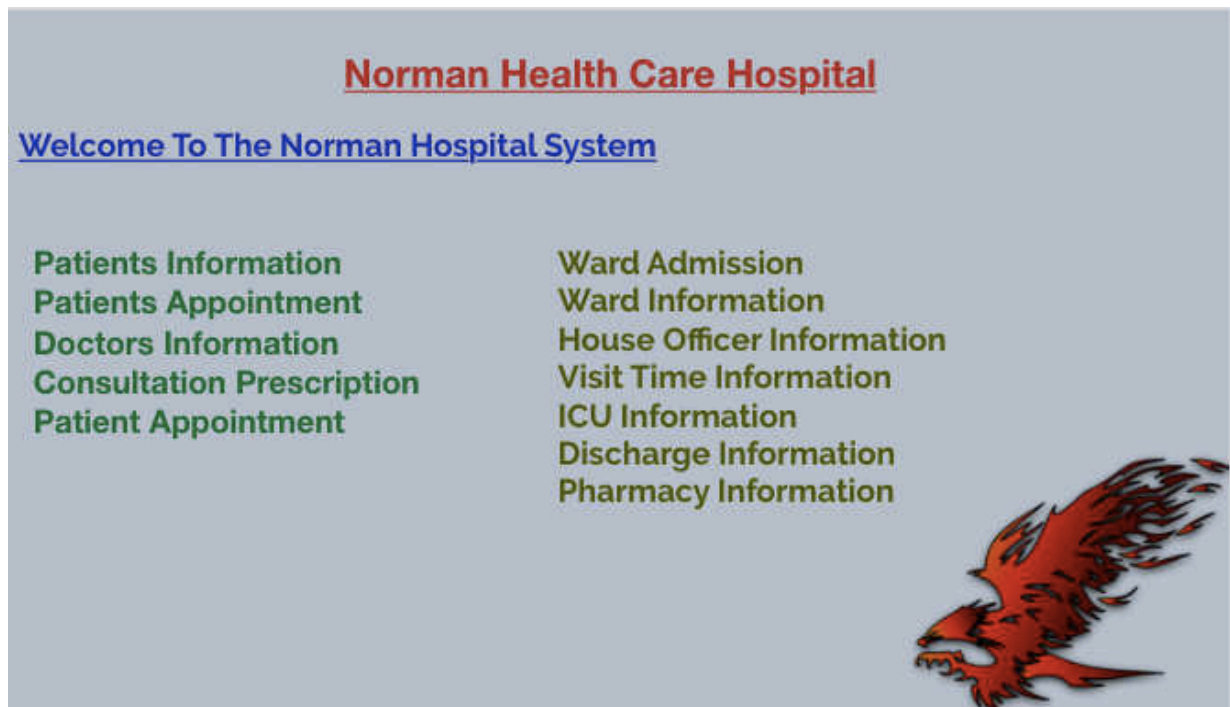
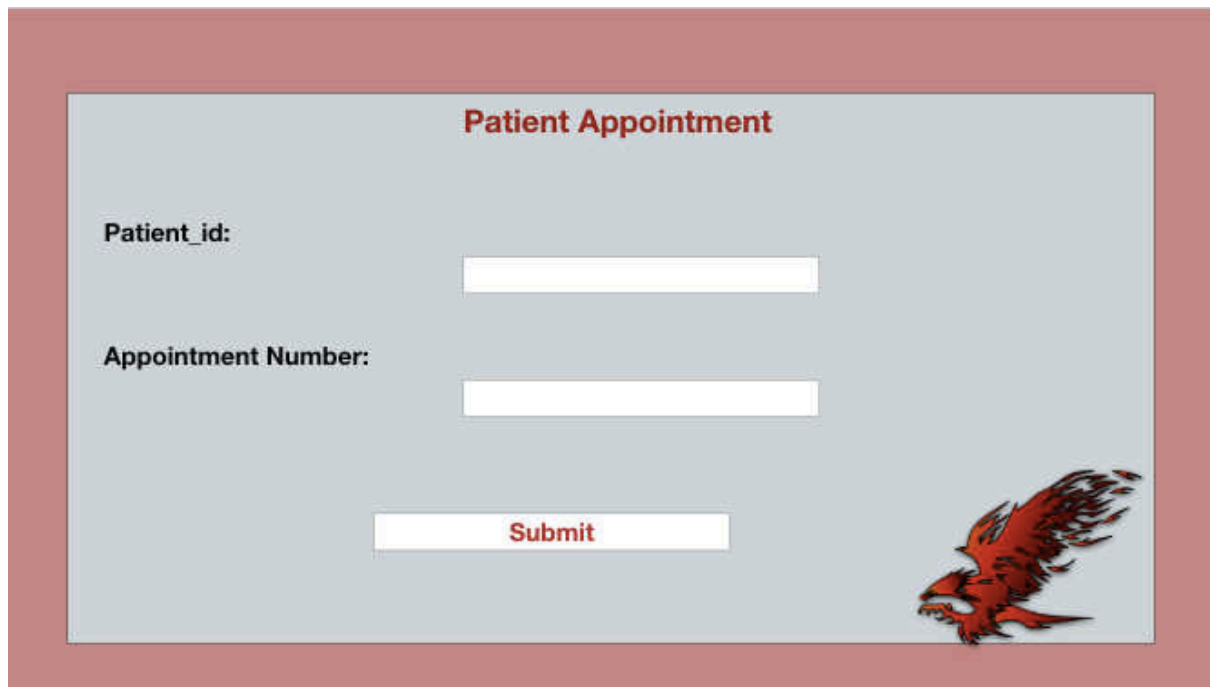


Figure 16:Main Page of DBMS

In this page the users of the Norman Hospital will have different options so they will select to go to separate pages for that.

Figure 17:Patient Information Page

In this page the new patient's will be added and after added press the submit button so, the details will be go to the database so the patient's information will be available through it.



Patient Appointment

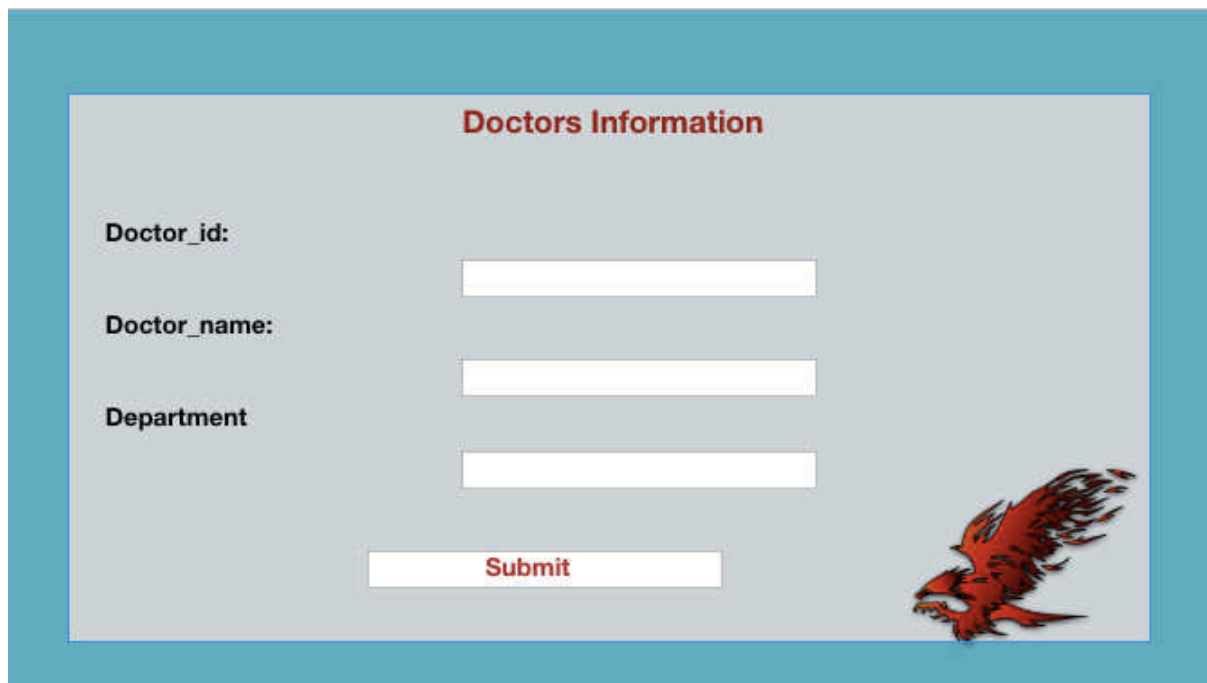
Patient_id:

Appointment Number:



Figure 18: To Make Appointments

Patients will be able to make appointments to come for checkups to doctors. If you are a new patient, our details will be added to the patient information page after that they make appointments.



Doctors Information

Doctor_id:

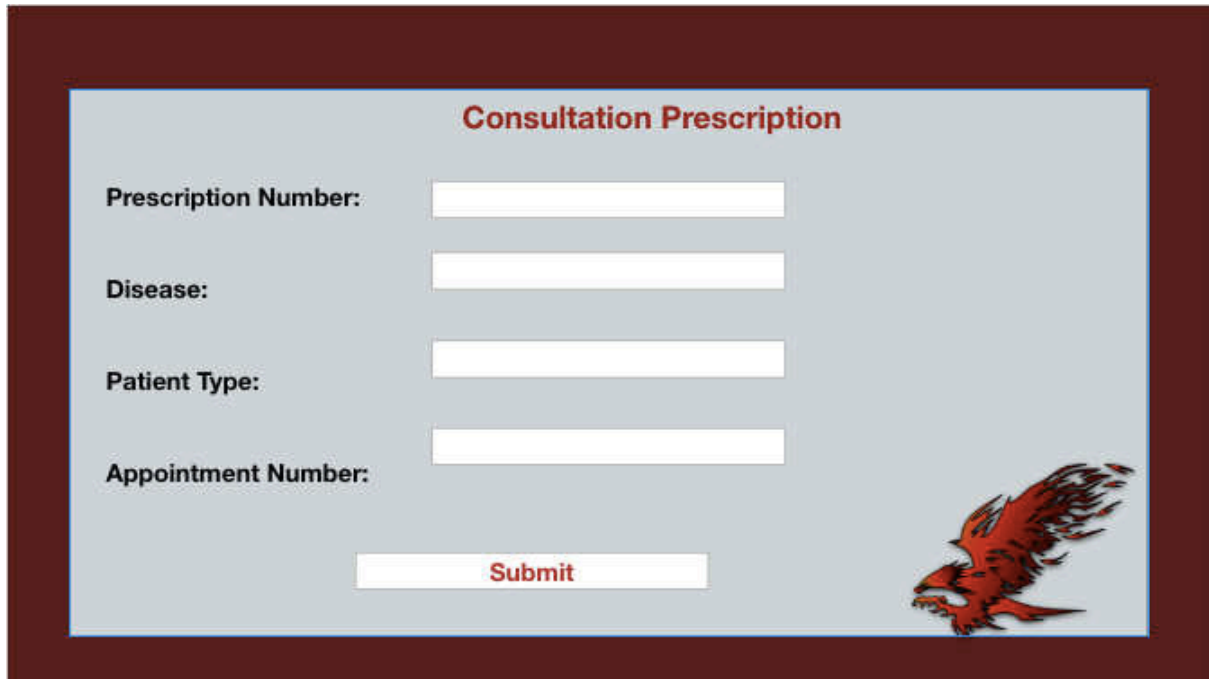
Doctor_name:

Department:



Figure 19: Norman Hospital Doctors Information

In this page, we have doctors' details so all doctors in the Norman Hospital will be in this page so we can access the doctor details by their doctor's ID.



Consultation Prescription

Prescription Number:

Disease:

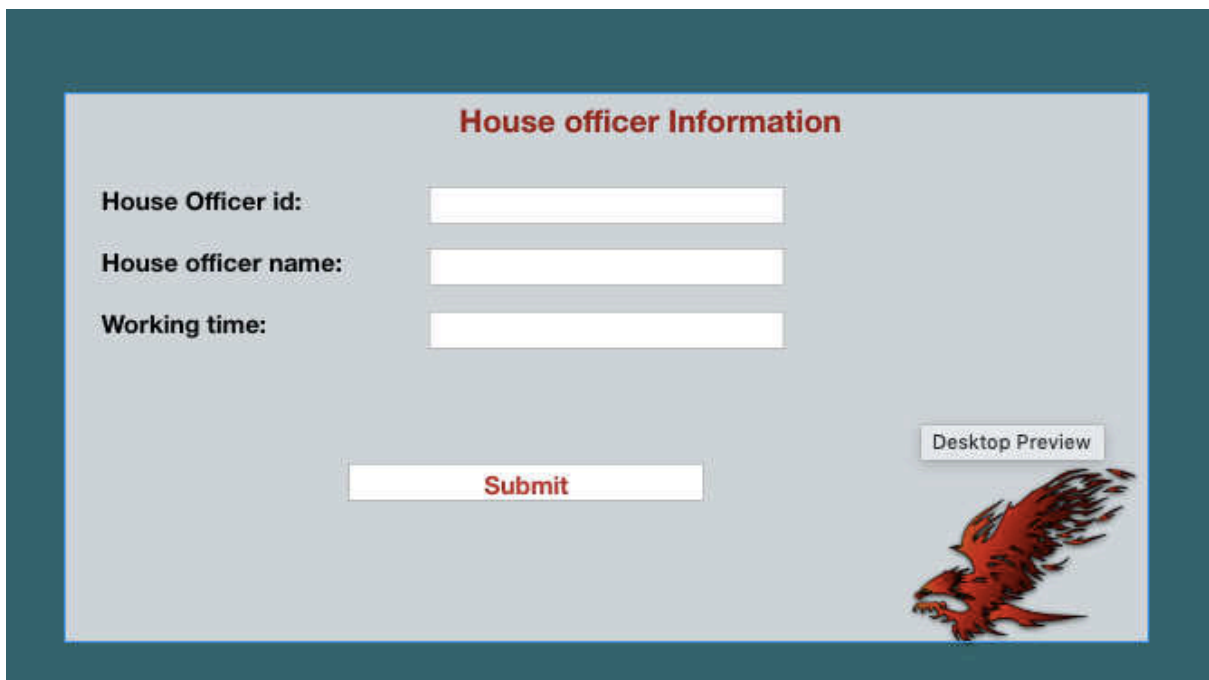
Patient Type:

Appointment Number:



Figure 20: In this picture the doctors will know about their patient

In this page we have all patient's prescription information and it will be accessed by the prescription numbers.



House officer Information

House Officer id:

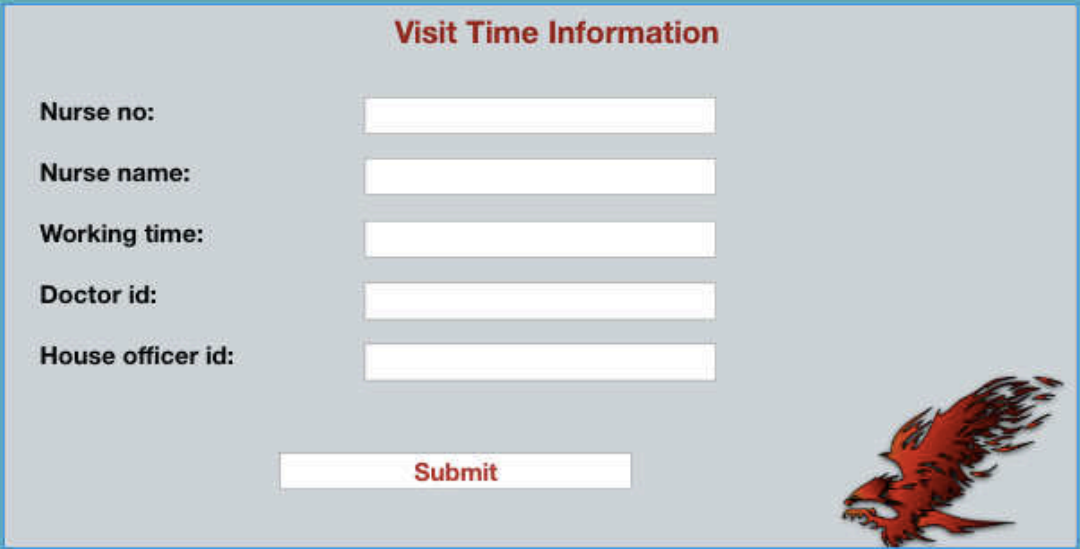
House officer name:

Working time:



Figure 21: This about the house officers

In this page we have all the house officers who are worked in Norman hospital and their information's are accessed by their id.



Visit Time Information

Nurse no:

Nurse name:

Working time:

Doctor id:

House officer id:

Figure 22: Patient visit time information

In this page the visit time of the doctors to the patient and which house officer and nurse are hire for that specified time are included.



Ward Information

Ward no:

House Officer id:

Bed no:

Admission no:

Patient Condition:

Figure 23: Ward information of patients

In this page the ward, bed, admission, and house officer information of every patients are included.



Admission Counter

Admission no:

Prescription no:

Guardian name:

Guardian Contact no:

Date:

Admission no:

Ward no:

Submit



Figure 24: Admission of every patient

If the patient wants to admit in the ward the following informations are collected through this page.



ICU Information

ICU officer no:

Admission no:

Patient Condition:

Record no:

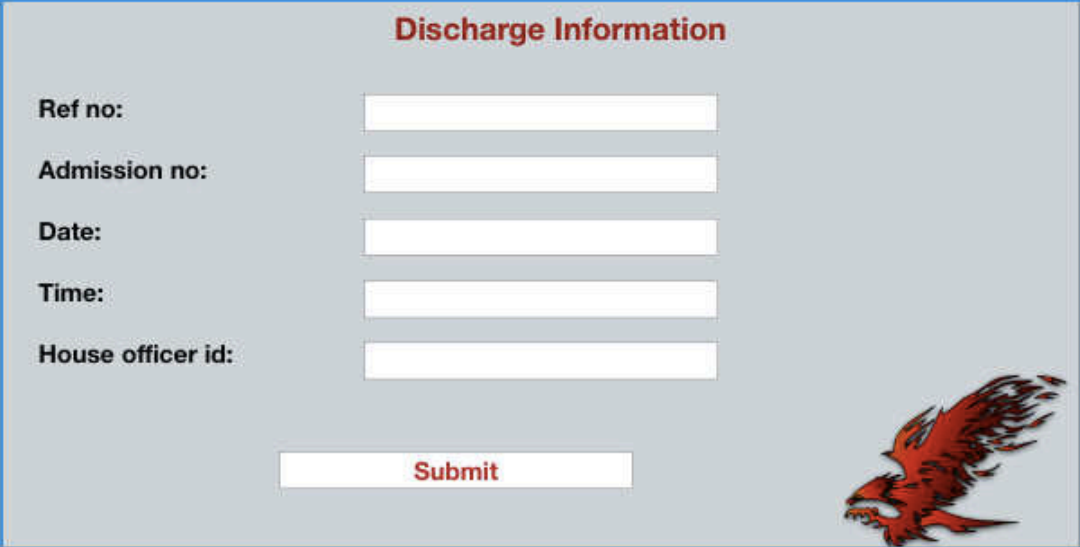
House officer id:

Submit



Figure 25: ICU patients Information

In this page the patient's who are admit to the ICU ward will fill up in this page.



Discharge Information

Ref no:

Admission no:

Date:

Time:

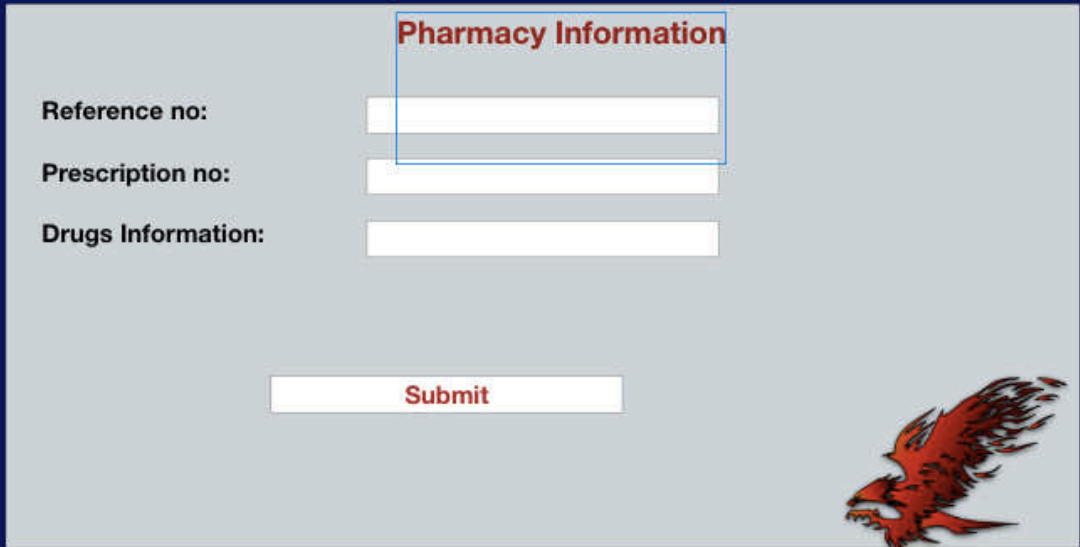
House officer id:

Submit



Figure 26:Discharge information of patients

In this page the patients who are discharged from the hospital will be included in this page.



Pharmacy Information

Reference no:

Prescription no:

Drugs Information:

Submit



Figure 27:Pharmacy details

This page will include the patient's drugs information.

Part 1.4 Evaluation Report

Evaluation is a report containing pros and cons of that specific software. So, the developers will be able to make the program or system more accurate and perfect.

In this Database Management System, the details are given by the Norman Hospital users so we need to make it perfect and accurate.

Norman Health Care Hospital is a public hospital providing treatments for the patients live in around the city of Galle. This hospital has been started early in 2000 and all the works are maintained manually up to now. So, I created this database system to make the work easily.

The details in the database system is more accurate because there is no repeating values or details. All details in this system are connected each other. If we take the patients table all new patients will added to that entity (Table) and in prescription the patient id is as a foreign key so the patient entity details are available or connected to the prescription entity. Like that all entities are connected so we have fulfilled user requirements and system requirement like Processor, Ram, Operating System.

User Requirements

Patients

All patient's have appointment numbers so, we have two entities such as patient, appointment and patient entity has first name, last name, age, gender, address, contact number etc.

Doctor

In a hospital system doctors are the most important and this table should have the doctor_id, doctor name, department and doctor information accessed by the doctor_id and this attribute act as foreign key in other tables to derive the doctor details.

Admission Counter

If a patient wants to admit in the hospital the patient information should be added to this table and this table has admission_no, guardian name, guardian contact number, bed no, ward no and lots of details.

These are some informations of data required by the users so we need to fulfill their needs and get their satisfaction.

L02

Part 2.1 Physical Relational Database

I created all the entities or tables needed for the Norman hospital system then going to forward engineer the schema to get a physical relational database.

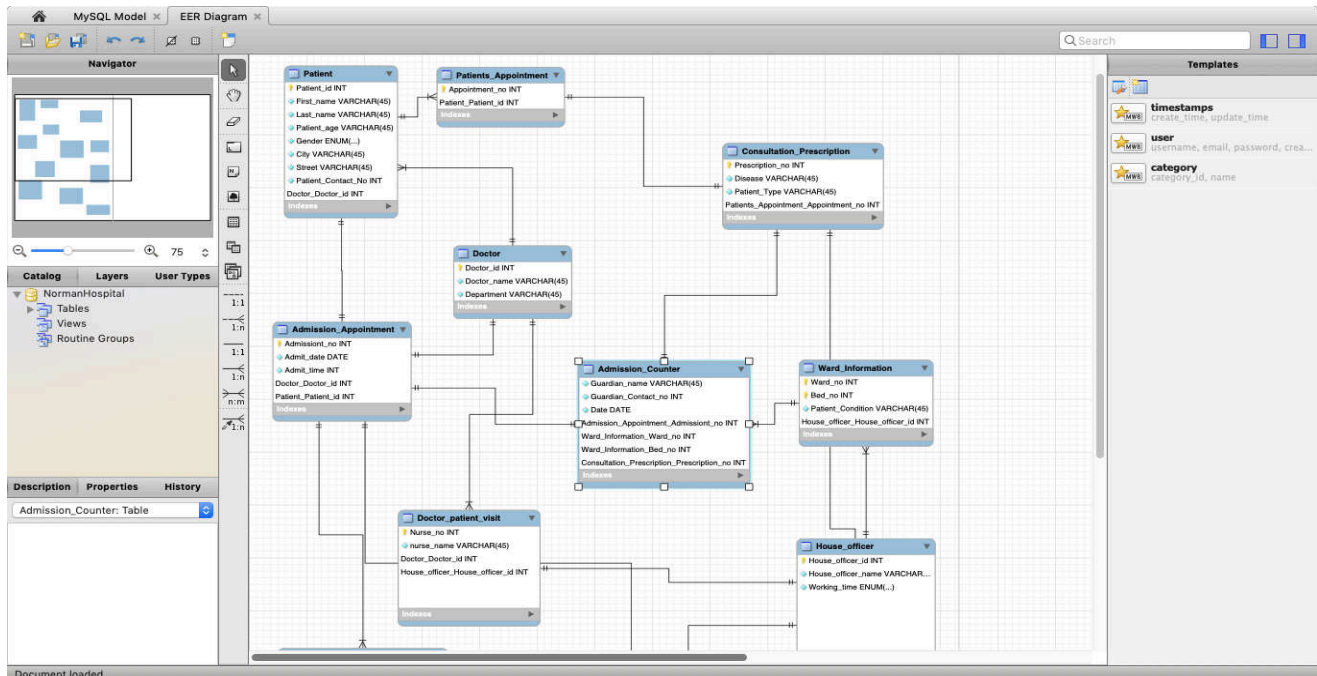


Figure 28: before Forward Engineering

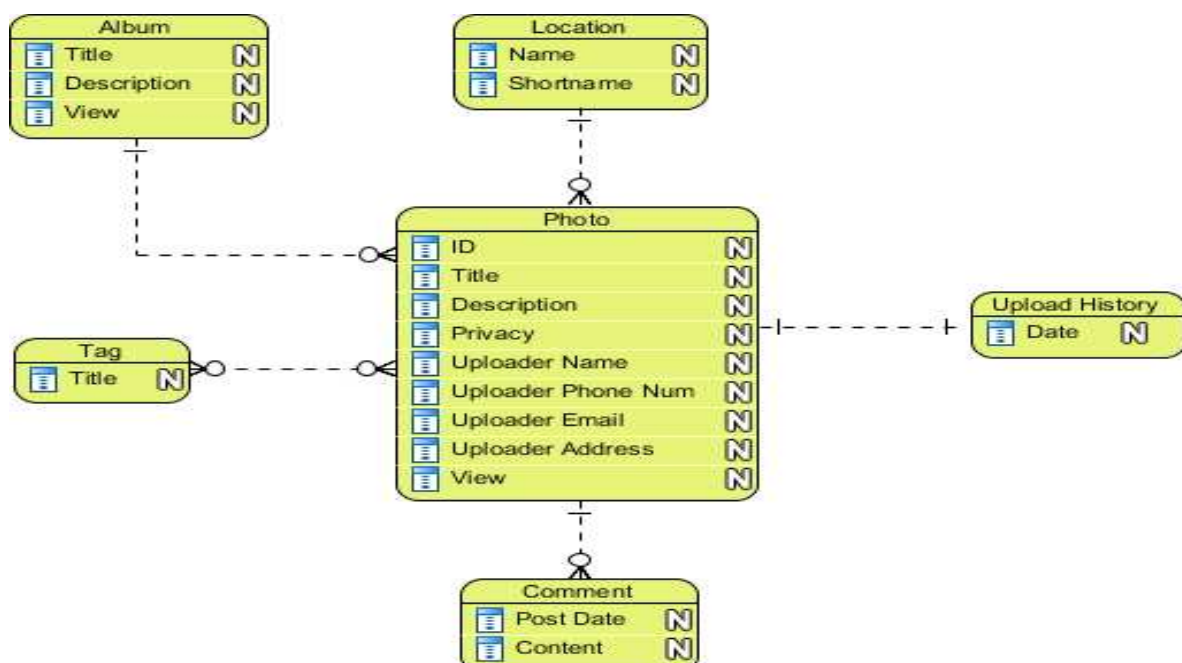


Figure 29: Example

Some Examples

Table name: Patient

```
CREATE TABLE IF NOT EXISTS `NormanHospital`.`Patient` (
  `Patient_id` INT NOT NULL,
  `First_name` VARCHAR (45) NOT NULL,
  `Last_name` VARCHAR (45) NOT NULL,
  `Patient_age` VARCHAR (45) NOT NULL,
  `Gender` ENUM ('Male', 'Female') NOT NULL,
  `City` VARCHAR (45) NOT NULL,
  `Street` VARCHAR (45) NOT NULL,
  `Patient_Contact_No` INT NOT NULL,
  `Doctor_Doctor_id` INT NOT NULL,
  PRIMARY KEY (`Patient_id`, `Doctor_Doctor_id`),
  INDEX `fk_Patient_Doctor1_idx` (`Doctor_Doctor_id` ASC) VISIBLE,
  CONSTRAINT `fk_Patient_Doctor1`
  FOREIGN KEY (`Doctor_Doctor_id`)
  REFERENCES `NormanHospital`.`Doctor` (`Doctor_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

In this I created a table patient using workbench to input all the patient's details. So, I put patient id attribute as primary key to access all the patient's detail to everywhere in the database system. In this Doctor table Doctor id is become as a foreign key to view the doctor details.

```
mysql> desc patient;
```

Field	Type	Null	Key	Default	Extra
Patient_id	int(11)	NO	PRI	NULL	
First_name	varchar(45)	NO		NULL	
Last_name	varchar(45)	NO		NULL	
Patient_age	varchar(45)	NO		NULL	
Gender	enum('Male','Female')	NO		NULL	
City	varchar(45)	NO		NULL	
Street	varchar(45)	NO		NULL	
Patient_Mobile_Contact_No	int(11)	NO		NULL	
Patient_Landline_no	int(11)	NO		NULL	
Doctor_Doctor_id	int(11)	NO	PRI	NULL	

```
10 rows in set (0.01 sec)
```

```
mysql> █
```

Table name: Patient Appointment

```
CREATE TABLE IF NOT EXISTS `NormanHospital`.`Patients_Appointment` (
  `Appointment_no` INT NOT NULL,
  `Patient_Patient_id` INT NOT NULL,
  PRIMARY KEY (`Patient_Patient_id`, `Appointment_no`),
  CONSTRAINT `fk_Patients_Appointment_Patient`
    FOREIGN KEY (`Patient_Patient_id`)
    REFERENCES `NormanHospital`.`Patient` (`Patient_id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

In this table we have appointment information they make their appointments. If they are new patients first their information fills up in the patient table after that come to this table and make their information. I assign appointment no as a primary key and put patient's id as foreign key to access the patient details in the appointment.

```
[mysql> desc patients_appointment;
```

Field	Type	Null	Key	Default	Extra
Appointment_no	int(11)	NO	PRI	NULL	
Patient_Patient_id	int(11)	NO	PRI	NULL	
Patient_Doctor_Doctor_id	int(11)	NO	PRI	NULL	

```
3 rows in set (0.03 sec)
```

```
mysql> █
```

Table name: Doctor

```
CREATE TABLE IF NOT EXISTS `NormanHospital`.`Doctor` (
  `Doctor_id` INT NOT NULL,
  `Doctor_name` VARCHAR (45) NOT NULL,
  `Department` VARCHAR (45) NOT NULL,
  PRIMARY KEY (`Doctor_id`))
ENGINE = InnoDB;
```

The table contains one primary key name as doctor id and by this primary key the doctor's details will be available throughout the database system. It has some int values and varchar values.

```
[mysql> desc doctor;
```

Field	Type	Null	Key	Default	Extra
Doctor_id	int(11)	NO	PRI	NULL	
Doctor_name	varchar(45)	NO		NULL	
Department	varchar(45)	NO		NULL	

```
3 rows in set (0.10 sec)
```

Table name: Consultation Prescription

```

CREATETABLEIFNOT EXISTS `NormanHospital`.`Consultation_Prescription`
(
  `Prescription_no` INT NOT NULL,
  `Disease` VARCHAR (45) NOT NULL,
  `Patient_Type` VARCHAR (45) NOT NULL,
  `Patients_Appointment_Appointment_no` INT NOT NULL,
  PRIMARYKEY (`Prescription_no`,
  `Patients_Appointment_Appointment_no`),
  INDEX `fk_Consultation_Prescription_Patients_Appointment1_idx`
  (`Patients_Appointment_Appointment_no` ASC) VISIBLE,
  CONSTRAINT `fk_Consultation_Prescription_Patients_Appointment1`
  FOREIGN KEY (`Patients_Appointment_Appointment_no`)
  REFERENCES `NormanHospital`.`Patients_Appointment`
  (`Appointment_no`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

In this table the disease of the patient, patient type out patient or not they are specified and we have a specified primary key for this table named as prescription no and this table access two foreign key named Appointment no and table has int, varchar values.

```
mysql> desc consultation_prescription;
```

Field	Type	Null	Key	Default	Extra
Prescription_no	int(11)	NO	PRI	NULL	
Disease	varchar(45)	NO		NULL	
Patient_Type	enum('In Patient','Out Patient')	NO		NULL	
Patients_Appointment_no	int(11)	NO	PRI	NULL	
Patients_Patient_Patient_id	int(11)	NO	PRI	NULL	
Patients_Appointment_Doctor_id	int(11)	NO	PRI	NULL	

```
6 rows in set (0.00 sec)
```

```
mysql> █
```

Table name: Hospital Ward Admission

```
CREATE TABLE IF NOT EXISTS `NormanHospital`.`Hospital_ward_Admission` (
  `Admission_no` INT NOT NULL,
  `Admit_date` DATE NOT NULL,
  `Admit_time` INT NOT NULL,
  `Doctor_Doctor_id` INT NOT NULL,
  `Patient_Patient_id` INT NOT NULL,
  PRIMARY KEY (`Admission_no`, `Doctor_Doctor_id`, `Patient_Patient_id`),
  INDEX `fk_Admission_Appointment_Doctor1_idx` (`Doctor_Doctor_id` ASC) VISIBLE,
  INDEX `fk_Admission_Appointment_Patient1_idx` (`Patient_Patient_id` ASC) VISIBLE,
  CONSTRAINT `fk_Admission_Appointment_Doctor1` FOREIGN KEY (`Doctor_Doctor_id`) REFERENCES `NormanHospital`.`Doctor` (`Doctor_id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_Admission_Appointment_Patient1` FOREIGN KEY (`Patient_Patient_id`) REFERENCES `NormanHospital`.`Patient` (`Patient_id`) ON DELETE NO ACTION ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

In this table a primary key named admission no and this table used for the hospital admission to admit the patient to ward. And this table has two foreign keys named patient Id and doctor id and has int and varchar values.

```
mysql> desc hospital_ward_admission;
```

Field	Type	Null	Key	Default	Extra
Admission_no	int(11)	NO	PRI	NULL	
Admit_date	date	NO		NULL	
Admit_time	int(11)	NO		NULL	
Doctor_Doctor_id	int(11)	NO	PRI	NULL	
Patient_Patient_id	int(11)	NO	PRI	NULL	
Patients_Appointment_Appointment_no	int(11)	NO	PRI	NULL	

```
6 rows in set (0.00 sec)
```

Table name: House officer

```
CREATE TABLE IF NOT EXISTS `NormanHospital`.`House_officer` (
  `House_officer_id` INT NOT NULL,
  `House_officer_name` VARCHAR (45) NOT NULL,
  `Working_time` ENUM ('Morning', 'Afternoon', 'Evening') NOT NULL,
  PRIMARY KEY (`House_officer_id`))
ENGINE = InnoDB;
```

For each ward there are some house officers to take care of their patients and this table has their details. All the house officers' details will be available in this table and this table has some int, varchar and enum values.

```
mysql> desc house_officer;
```

Field	Type	Null	Key	Default	Extra
House_officer_id	int(11)	NO	PRI	NULL	
House_officer_name	varchar(45)	NO		NULL	
Working_time	enum('Morning','Afternoon','Evening')	NO		NULL	

```
3 rows in set (0.09 sec)
```

```
[mysql> ]
```

Table name: Ward Information

```
CREATE TABLE IF NOT EXISTS `NormanHospital`.`Ward Information` (
  `Ward_no` INT NOT NULL,
  `Bed_no` INT NOT NULL,
  `Patient_Condition` VARCHAR (45) NOT NULL,
  `House_officer_House_officer_id` INT NOT NULL,
  PRIMARY KEY (`Ward_no`, `Bed_no`,
  `House_officer_House_officer_id`),
  INDEX `fk_Ward_Information_House_officer1_idx`
  (`House_officer_House_officer_id` ASC) VISIBLE,
  CONSTRAINT `fk_Ward_Information_House_officer1`
  FOREIGN KEY (`House_officer_House_officer_id`)
  REFERENCES `NormanHospital`.`House_officer` (`House_officer_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

This table is used to describe the ward information for separate patients. If the person who admit in this hospital their ward information bed information and all the information's will be available in this table.

```
[mysql> desc ward_information;
```

Field	Type	Null	Key	Default	Extra
Ward_no	int(11)	NO	PRI	NULL	
Bed_no	int(11)	NO	PRI	NULL	
Patient_Condition	varchar(45)	NO		NULL	
House_officer_House_officer_id	int(11)	NO	PRI	NULL	

```
4 rows in set (0.00 sec)
```

Table name: Admission Counter

```
CREATE TABLE IF NOT EXISTS `NormanHospital`.`Admission_Counter` (
  `Guardian_name` VARCHAR (45) NOT NULL,
  `Guardian_Contact_no` INT NOT NULL,
  `Date` DATE NOT NULL,
  `Admission_Appointment_Admissiонт_no` INT NOT NULL,
  `Ward_Information_Ward_no` INT NOT NULL,
  `Ward_Information_Bed_no` INT NOT NULL,
  `Consultation_Prescription_Prescription_no` INT NOT NULL,
  PRIMARY KEY (`Admission_Appointment_Admissiонт_no`,
  `Ward_Information_Ward_no`, `Ward_Information_Bed_no`,
  `Consultation_Prescription_Prescription_no`),
  INDEX `fk_Admission_Counter_Ward_Information1_idx`
  (`Ward_Information_Ward_no` ASC, `Ward_Information_Bed_no` ASC)
  VISIBLE,
  INDEX `fk_Admission_Counter_Consultation_Prescription1_idx`
  (`Consultation_Prescription_Prescription_no` ASC) VISIBLE,
  CONSTRAINT `fk_Admission_Counter_Admission_Appointment1`
  FOREIGN KEY (`Admission_Appointment_Admissiонт_no`)
  REFERENCES `NormanHospital`.`Admission_Appointment`
  (`Admissiонт_no`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `fk_Admission_Counter_Ward_Information1`
  FOREIGN KEY (`Ward_Information_Ward_no`,
  `Ward_Information_Bed_no`)
  REFERENCES `NormanHospital`.`Ward_Information` (`Ward_no`,
```



```

`Bed_no`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
    CONSTRAINT `fk_Admission_Counter_Consultation_Prescription1`
    FOREIGN KEY (`Consultation_Prescription_Prescription_no`)
    REFERENCES `NormanHospital`.`Consultation_Prescription`
    (`Prescription_no`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

In this table before the going to admit in the ward the patient's information should be fill up in this table and all the patients have a specific admission no so, they can be easily identified by the doctors, house officers and nurse.

```
mysql> desc admission_counter;
```

Field	Type	Null	Key	Default	Extra
Guardian_name	varchar(45)	NO		NULL	
Guardian_Contact_no	int(11)	NO		NULL	
Date	date	NO		NULL	
Admission_Appointment_Admissiонт_no	int(11)	NO	PRI	NULL	
Ward_Information_Ward_no	int(11)	NO	PRI	NULL	
Ward_Information_Bed_no	int(11)	NO	PRI	NULL	

```
6 rows in set (0.00 sec)
```

```
mysql> █
```

Table name: Doctor Patient Visit

```
CREATE TABLE IF NOT EXISTS `NormanHospital`.`Doctor_patient_visit` (
  `Nurse_no` INT NOT NULL,
  `nurse_name` VARCHAR (45) NOT NULL,
  `Doctor_Doctor_id` INT NOT NULL,
  `House_officer_House_officer_id` INT NOT NULL,
  PRIMARY KEY (`Nurse_no`, `Doctor_Doctor_id`,
  `House_officer_House_officer_id`),
  INDEX `fk_Doctor_patient_visit_Doctor1_idx` (`Doctor_Doctor_id`
ASC) VISIBLE,
  INDEX `fk_Doctor_patient_visit_House_officer1_idx`
(`House_officer_House_officer_id` ASC) VISIBLE,
  CONSTRAINT `fk_Doctor_patient_visit_Doctor1`
  FOREIGN KEY (`Doctor_Doctor_id`)
  REFERENCES `NormanHospital`.`Doctor` (`Doctor_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `fk_Doctor_patient_visit_House_officer1`
  FOREIGN KEY (`House_officer_House_officer_id`)
  REFERENCES `NormanHospital`.`House_officer` (`House_officer_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

When the doctor's visit the patients there will be a nurse and house officer so in this table the doctor visit will be added.

```
mysql> desc doctor_patient_visit;
```

Field	Type	Null	Key	Default	Extra
Nurse_no	int(11)	NO	PRI	NULL	
nurse_name	varchar(45)	NO		NULL	
Doctor_Doctor_id	int(11)	NO	PRI	NULL	
House_officer_House_officer_id	int(11)	NO	PRI	NULL	

```
4 rows in set (0.00 sec)
```

```
mysql> █
```

Table name: Discharge information

```
CREATE TABLE IF NOT EXISTS `NormanHospital`.`Discharge_Information`
(
  `Reference_no` INT NOT NULL,
  `Date` DATE NOT NULL,
  `Time` VARCHAR (45) NOT NULL,
  `Admission_Appointment_Admissiонт_no` INT NOT NULL,
  PRIMARY KEY (`Reference_no`,
  `Admission_Appointment_Admissiонт_no`),
  INDEX `fk_Discharge_Information_Admission_Appointment1_idx`
  (`Admission_Appointment_Admissiонт_no` ASC) VISIBLE,
  CONSTRAINT `fk_Discharge_Information_Admission_Appointment1`
  FOREIGN KEY (`Admission_Appointment_Admissiонт_no`)
  REFERENCES `NormanHospital`.`Admission_Appointment`
  (`Admissiонт_no`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

When the patients discharge from the hospital this table will be fill up and after that they will be discharged.

```
mysql> desc discharge_information;
```

Field	Type	Null	Key	Default	Extra
Reference_no	int(11)	NO	PRI	NULL	
Date	date	NO		NULL	
Time	varchar(45)	NO		NULL	
Hospital_Fees	double	NO		NULL	
Hospital_Ward_Admission_Admissiонт_no	int(11)	NO	PRI	NULL	
Hospital_Ward_Admission_Doctor_Doctor_id	int(11)	NO	PRI	NULL	
Hospital_Ward_Admission_Patient_Patient_id	int(11)	NO	PRI	NULL	
Hospital_Ward_Admission_Patients_Appointment_Appointment_no	int(11)	NO	PRI	NULL	
Doctors_charge	double	NO		NULL	

```
9 rows in set (0.10 sec)
```

```
mysql> █
```

Table name: ICU information

```
CREATE TABLE IF NOT EXISTS `NormanHospital`.`ICU_Information` (
  `ICU_officer_id` INT NOT NULL,
  `Record_no` VARCHAR (45) NOT NULL,
  `Patient_Condition` VARCHAR (45) NOT NULL,
  `Admission_Appointment_Admissiонт_no` INT NOT NULL,
  `House_officer_House_officer_id` INT NOT NULL,
  PRIMARY KEY (`ICU_officer_id`, `Record_no`,
  `Admission_Appointment_Admissiонт_no`,
  `House_officer_House_officer_id`),
  INDEX `fk_ICU_Information_Admission_Appointment1_idx`
  (`Admission_Appointment_Admissiонт_no` ASC) VISIBLE,
  INDEX `fk_ICU_Information_House_officer1_idx`
  (`House_officer_House_officer_id` ASC) VISIBLE,
  CONSTRAINT `fk_ICU_Information_Admission_Appointment1`
  FOREIGN KEY (`Admission_Appointment_Admissiонт_no`)
  REFERENCES `NormanHospital`.`Admission_Appointment`
  (`Admissiонт_no`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `fk_ICU_Information_House_officer1`
  FOREIGN KEY (`House_officer_House_officer_id`)
  REFERENCES `NormanHospital`.`House_officer` (`House_officer_id`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

If the patient admits to ICU the following information's will be fill up and for the ICU there is an officer head for the ICU department and this table has foreign keys like admission no, house officer id and has a primary key named ICU officer id and has

```
mysql> desc icu_information;
```

Field	Type	Null	Key	Default	Extra
ICU_officer_id	int(11)	NO	PRI	NULL	
Record_no	varchar(45)	NO	PRI	NULL	
Patient_Condition	varchar(45)	NO		NULL	
House_officer_H_id	int(11)	NO	PRI	NULL	
Hospital_Ward_Admissiонт_no	int(11)	NO	PRI	NULL	
Hospital_Ward_Admission_D_id	int(11)	NO	PRI	NULL	
Hospital_Ward_Admission_P_id	int(11)	NO	PRI	NULL	
Hospital_Ward_Admission_Appointment_no	int(11)	NO	PRI	NULL	

```
8 rows in set (0.10 sec)
```

```
mysql> █
```

int, varchar values

Table name: Pharmacy information

```
CREATE TABLE IF NOT EXISTS `NormanHospital`.`Pharmacy_Information` (
  `Pharmacy_Customer_no` INT NOT NULL,
  `Drugs` VARCHAR (45) NOT NULL,
  `Discharge_Information_Reference_no` INT NOT NULL,
  `Consultation_Prescription_Prescription_no` INT NOT NULL,
  PRIMARY KEY (`Pharmacy_Customer_no`,
  `Discharge_Information_Reference_no`,
  `Consultation_Prescription_Prescription_no`),
  INDEX `fk_Pharmacy_Information_Consultation_Prescription1_idx`
  (`Consultation_Prescription_Prescription_no` ASC) VISIBLE,
  CONSTRAINT `fk_Pharmacy_Information_Discharge_Information1`
  FOREIGN KEY (`Discharge_Information_Reference_no`)
  REFERENCES `NormanHospital`.`Discharge_Information`
  (`Reference_no`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `fk_Pharmacy_Information_Consultation_Prescription1`
  FOREIGN KEY (`Consultation_Prescription_Prescription_no`)
  REFERENCES `NormanHospital`.`Consultation_Prescription`
  (`Prescription_no`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

After discharge the patient need to take medicines so that when the patient go to pharmacy the table will be fill up.

```
mysql> desc pharmacy_information;
```

Field	Type	Null	Key	Default	Extra
Pharmacy_Customer_no	int(11)	NO	PRI	NULL	
Drugs	varchar(45)	NO		NULL	
Consultation_Prescription_Prescription_no	int(11)	NO	PRI	NULL	
Discharge_Info_Reference_no	int(11)	NO	PRI	NULL	
Discharge_Information_Admission_no	int(11)	NO	PRI	NULL	
Discharge_Information_Doctor_id	int(11)	NO	PRI	NULL	
Discharge_Information_Patient_id	int(11)	NO	PRI	NULL	
Discharge_Information_Appointment_no	int(11)	NO	PRI	NULL	
Pharmacy_Fees	double	NO		NULL	

```
9 rows in set (0.10 sec)
```

```
mysql> █
```

Part 2.2 Privilege Matrix

MySQL is an open source database management software that helps users to store, organize and retrieve data. It has lot of options to grant specific users nuanced permissions within the tables and databases.

How To Grant Different User Permissions

Here is a short list of other common possible permissions that users can enjoy.

- ALL PRIVILEGES- as we saw previously, this would allow a MySQL user full access to a designated database (or if no database is selected, global access across the system).
- CREATE- allows them to create new tables or databases.
- DROP- allows them to delete tables or databases.
- DELETE- allows them to delete rows from tables.
- INSERT- allows them to insert rows into tables.
- SELECT- allows them to use the `SELECT` command to read through databases.
- UPDATE- allow them to update table rows.
- GRANT OPTION- allows them to grant or remove other users' privileges.

The users and privileges are used to administrate the privileges and if you are users in the Norman hospital, you have several permissions such as delete, update and drop so you can only do these permissions you can't access other permissions like insert etc. The user privileges will not allow you to do other privileges.

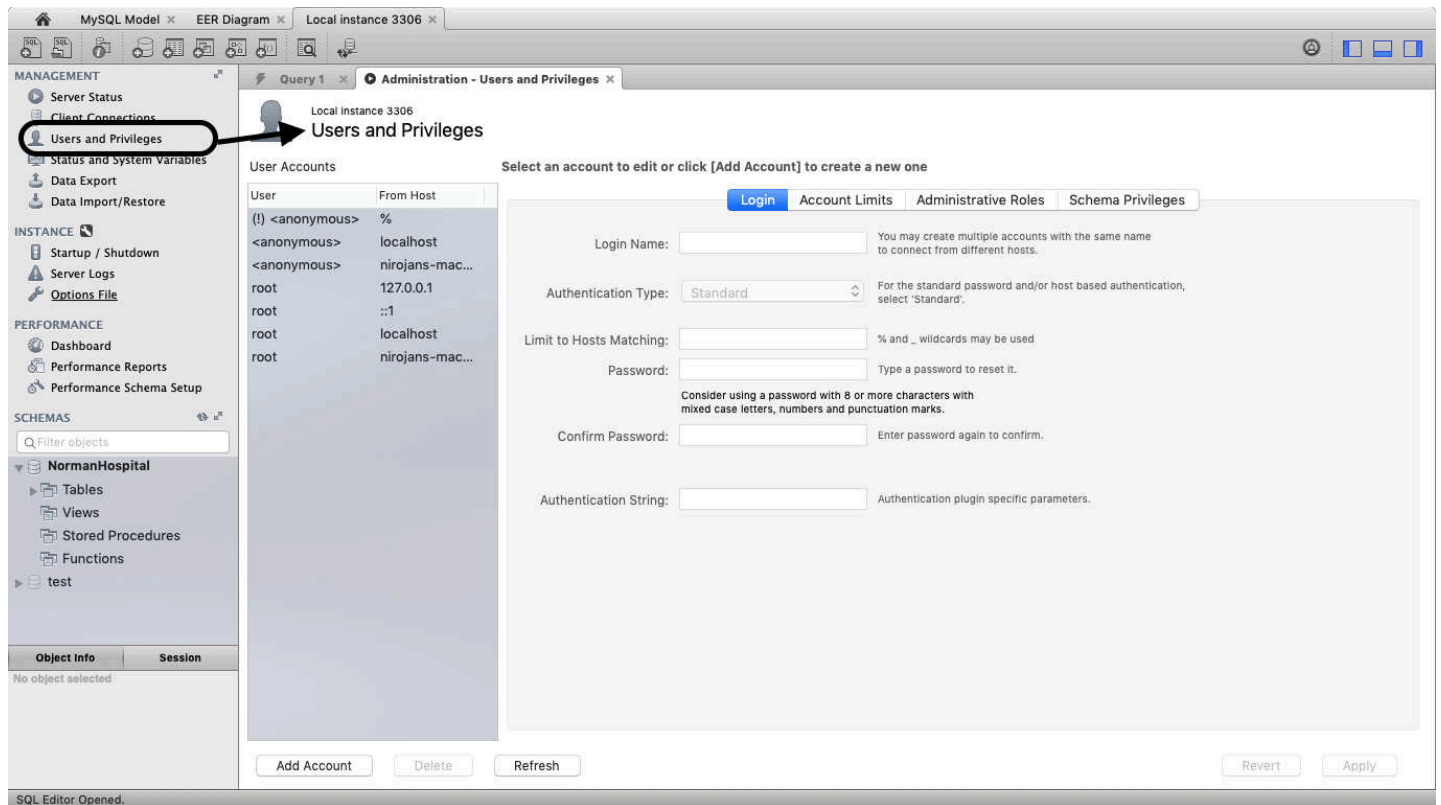


Figure 30:Privileges matrix

This the user's privileges tab you can click this to create a new user and give that user their privileges.

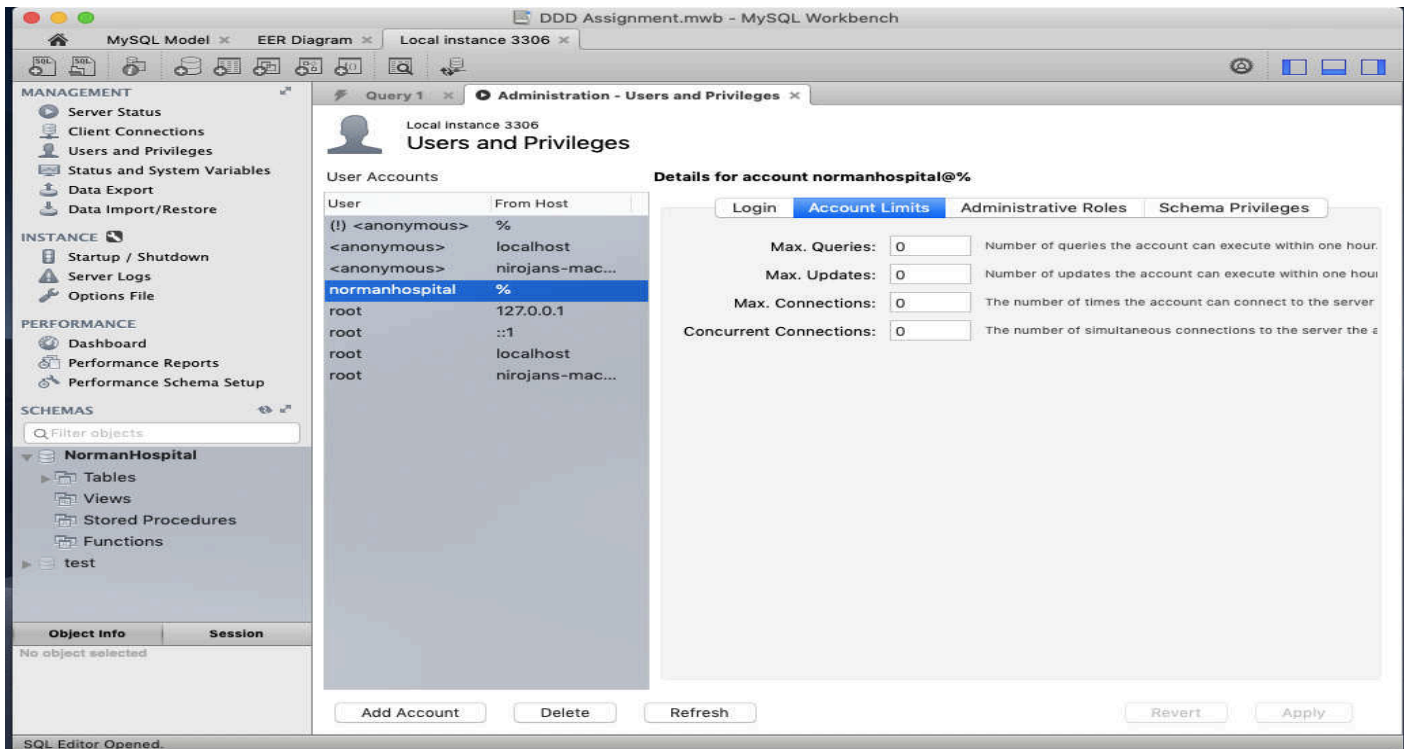


Figure 31:Account limits

This tab will allow the users how many queries they can do their job and we can set the account limits.

In the administration level of a system there are roles apply for every users and DBA is the main role which means it can perform all works and other roles are applied for specific tasks. It will make our system more accurate because never grant more access to user accounts and the privileges are added according to their needs.

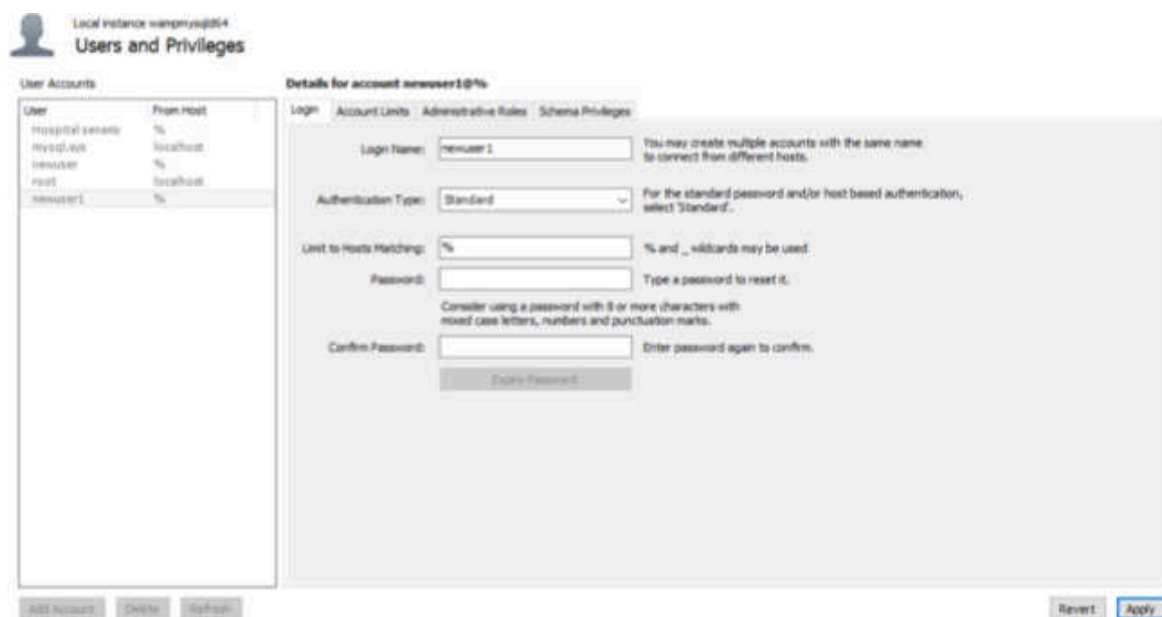


Figure 32:Privilege's user and password

Sample picture of workbench when create user privileges.

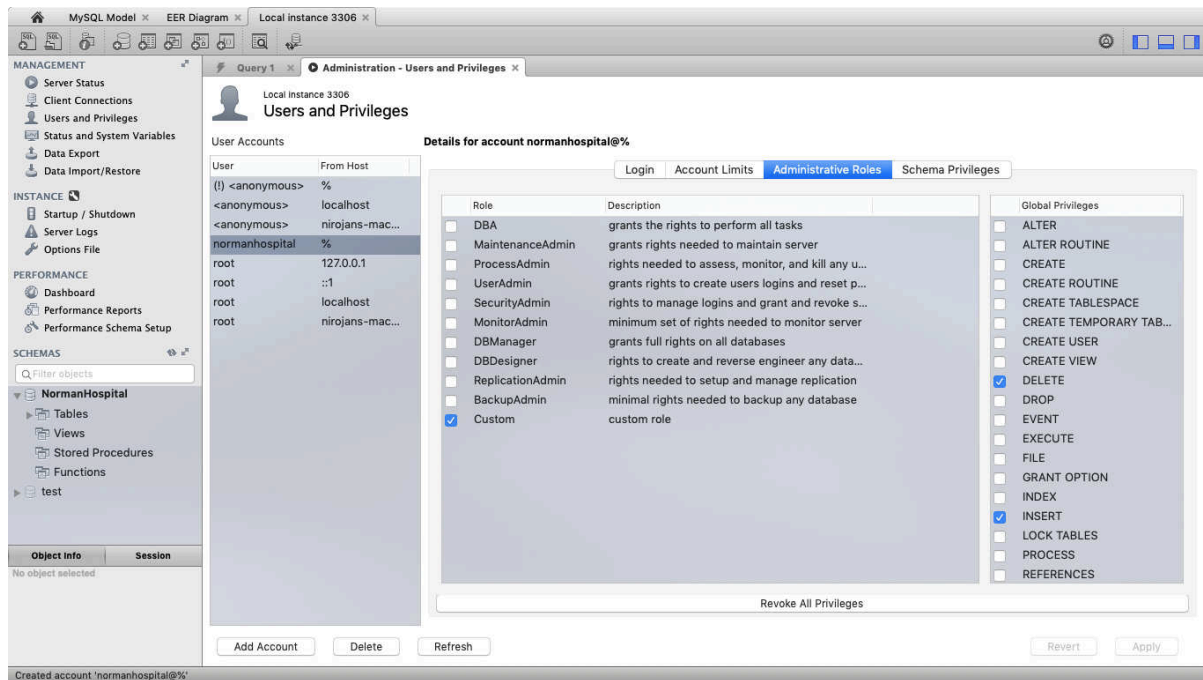


Figure 33:Privilege roles

This tab will allow to select the other users these are some privileges and decline others when user select update privileges and use the insert privileges So, that they can't access other details.

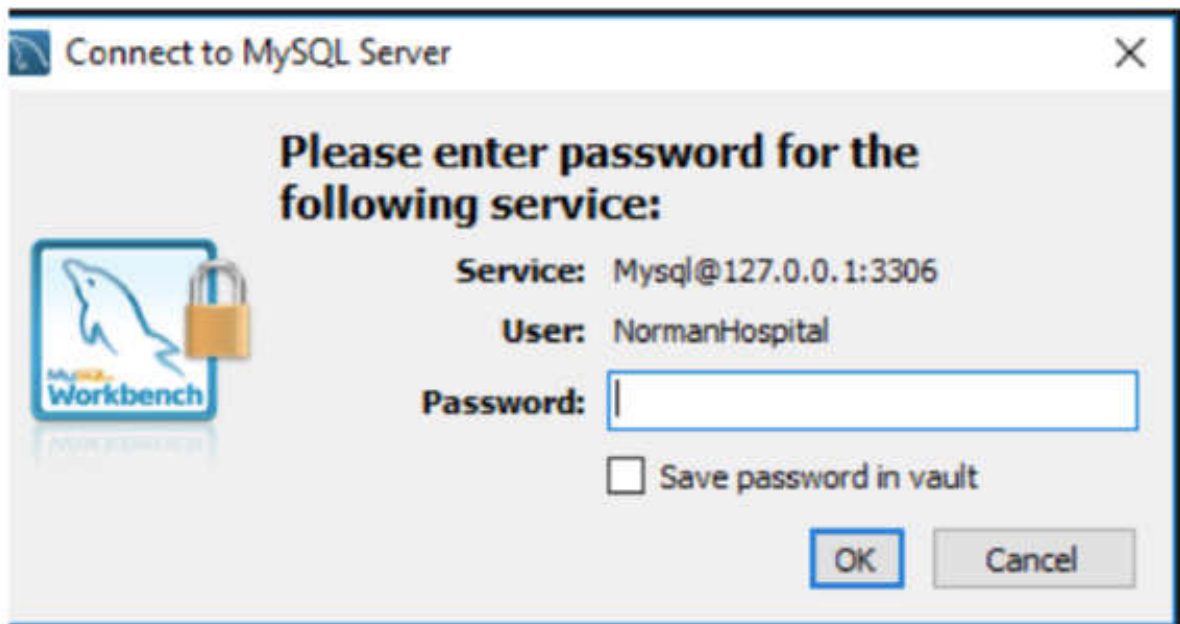


Figure 34:Password privileges

This a user of Norman hospital and we use the password to access the database.

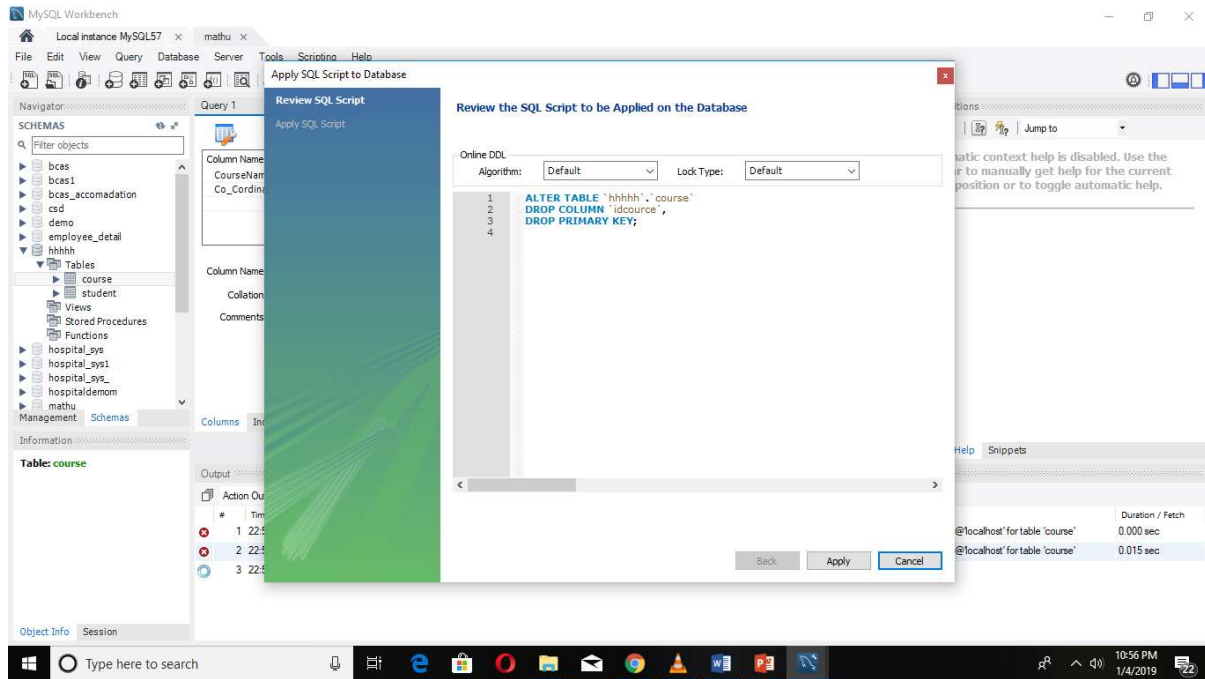


Figure 35:Privilege actions

This privilege are allow to alter, drop but I need to change the table details it will have some errors.

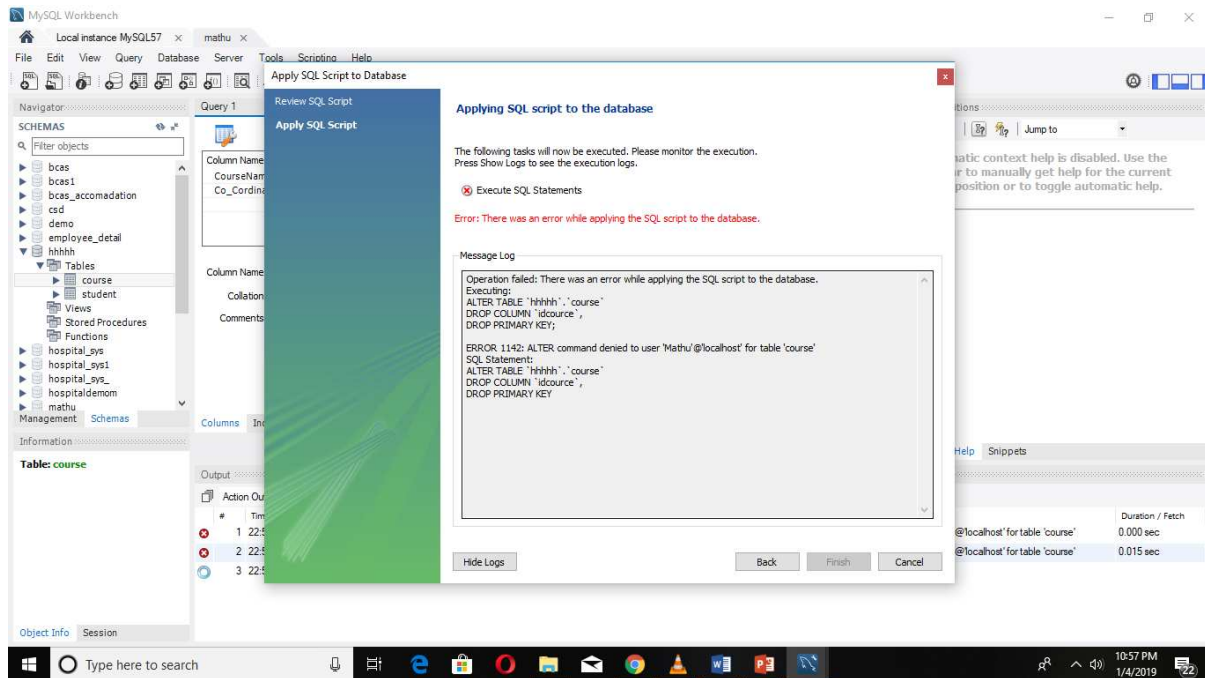


Figure 36:Privilege access denied

This is an error when alter drop column because drop column is not in my privileges.

Part 2.2 Inner Join, Outer Join, Full Join

Inner Join

```
mysql> select A.first_name,B.drugs,C.doctor_name from patient as A
inner join doctor as C on A.doctor_id=C.doctor_id inner join
pharmacy_information as B on
B.discharge_information_doctor_id=C.doctor_id;
```

first_name	drugs	doctor_name
Nirojan	Vitamin C	Logetha
Anton	Vitamin C	Logetha
Aravinthan	Vitamin C	Logetha
Saranya	Vitamin C	Logetha
Partheeban	Vitamin C	Logetha
Sanjsi	Calcium	Mathunujan
Thaas	Calcium	Mathunujan
Thiviyan	Calcium	Mathunujan
king	Calcium	Mathunujan
Antony	Polio Tablets	Vijay
Kunam	Polio Tablets	Vijay
Mayooran	Polio Tablets	Vijay
Delaselp	Polio Tablets	Vijay
Mathu	Multi Vitamin	Ajith
Mohammad	Multi Vitamin	Ajith
Prathap	Multi Vitamin	Ajith
Prana	Iodine Tablets	Simran
Mohammad	Iodine Tablets	Simran
Shankavi	Iodine Tablets	Simran
Anistan	Vitamin A	Samantha
Thananji	Vitamin A	Samantha
Samitha	Vitamin A	Samantha

22 rows in set (0.00 sec)

mysql>

In this I select first name, drugs, doctor name from patient entity, doctor entity and pharmacy entity.

Left Outer Join

```
mysql> select
patient.first_name,patient.gender,consultation_prescription.disease
from patient left join patients_appointment on
patient.patient_id=patients_appointment.patient_patient_id left join
consultation_prescription on
consultation_prescription.patients_appointment_doctor_id=patients_ap
pointment.patient_doctor_doctor_id;
```

I used patient, consultation_prescription and patients_appointment tables to make left outer join.

first_name	gender	disease
Nirojan	Male	Chikunkuniya
Sanjsi	Male	Malaria
Antony	Male	Dengu
Mathu	Male	Fever
Prana	Male	LooseMotion
Anistan	Male	Aids
Anton	Male	NULL
Thaas	Male	NULL
Kunam	Male	NULL
Mohammad	Male	NULL
Mohammad	Male	NULL
Thananji	Male	NULL
Aravinthan	Male	NULL
Thiviyan	Male	NULL
Mayooran	Male	NULL
Prathap	Male	NULL
Shankavi	Female	NULL
Samitha	Female	NULL
Saranya	Female	NULL
king	Male	NULL
Delaselp	Male	NULL

21 rows in set (0.11 sec)

```
mysql>
```

Right Join

```
mysql> select
patient.first_name,patient.gender,consultation_prescription.disease
from patient right join patients_appointment on
patient.patient_id=patients_appointment.patient_patient_id right
join consultation_prescription on
consultation_prescription.patients_appointment_doctor_id=patients_ap
pointment.patient_doctor_doctor_id;
```

In this also I have chooses 3 tables and get this result.

first_name	gender	disease
Nirojan	Male	Chikunkuniya
Sanjsi	Male	Malaria
Antony	Male	Dengu
Mathu	Male	Fever
Prana	Male	LooseMotion
Anistan	Male	Aids

6 rows in set (0.00 sec)

```
mysql>
```

Full Join

```

select      patient.first_name,consultation_prescription.disease
->          from      patient
->      left      join      consultation_prescription      on
consultation_prescription.Patients_Patient_Patient_id
=patient.patient_id
-> union all
-> select  patient.first_name,consultation_prescription.disease
->          from      patient
->      left      join      consultation_prescription      on
consultation_prescription.Patients_Patient_Patient_id=patient.patien
t_id;

```

Full join is a combination of right join and the left join and here is the output of full join.

first_name	disease
Nirojan	Chikunkuniya
Sanjsi	Malaria
Antony	Dengu
Mathu	Fever
Prana	LooseMotion
Anistan	Aids
Anton	NULL
Thaas	NULL
Kunam	NULL
Mohammad	NULL
Mohammad	NULL
Thananji	NULL
Aravinthan	NULL
Thiviyan	NULL
Mayooran	NULL
Prathap	NULL
Shankavi	NULL
Samitha	NULL
Saranya	NULL
king	NULL
Delaselp	NULL
Nirojan	Chikunkuniya
Sanjsi	Malaria
Antony	Dengu
Mathu	Fever
Prana	LooseMotion
Anistan	Aids
Anton	NULL

Thaas	NULL
Kunam	NULL
Mohammad	NULL
Mohammad	NULL
Thananji	NULL
Aravinthan	NULL
Thiviyan	NULL
Mayooran	NULL
Prathap	NULL
Shankavi	NULL
Samitha	NULL
Saranya	NULL
king	NULL
Delaselp	NULL

+-----+

42 rows in set (0.10 sec)

Insert, Update, Delete, Alter, Modify

Insert

```
mysql> select * from patient;
```

Patient_id	First_name	Last_name	Patient_age	Gender	City	Street	Patient_Mobile_Contact_No	Patient_Landline_no	Doctor_Doctor_id
1	Nirojan	Stany	24	Male	Jaffna	Main Street	769545654	212234567	1
2	Sanjsi	Thiru	23	Male	Vavuniya	Vipulanathan road	765432123	212224470	2
3	Antony	Pethurshan	19	Male	Galle	Kandhi Road	765497345	212250646	3
4	Mathu	Sukumar	43	Male	Kandy	1st Cross Street	769456345	212051945	4
5	Prana	Nadaraja	55	Male	Thincomalee	3rd cross Street	712393256	213737878	5
6	Anistan	Lillian	33	Male	Mannar	Kandy road	728745090	213336543	6
7	Anton	Nirojan	51	Male	Galle	Church road	722212334	212229910	1
8	Thaas	Jegan	21	Male	Mannar	bastian road	779345128	213456712	2
9	Kunam	Anshalon	22	Male	Vavuniya	main street	776523491	212349456	3
10	Mohammad	Hussain	51	Male	Jaffna	irupalai road	745634812	312349434	4
11	Mohammad	Niyas	30	Male	Galle	Tellipalai road	723476410	212226603	5
12	Thananji	Anthoni	32	Male	Galle	kumankulam road	779933561	213217710	6
13	Aravinthan	Alex	19	Male	Galle	Pandarikulam road	764523995	213437345	1
14	Thiviyan	Satha	23	Male	Kandy	Vanuniya main street	779323456	212235523	2
15	Mayooran	Mani	25	Male	Colombo	Chunnakam road	712366124	217774523	3
16	Prathap	Theivam	60	Male	Jaffna	Main street	754577012	212250567	4
17	Shankavi	Paramsoothy	59	Female	Mannar	Lanka road	777716300	216674377	5
18	Samitha	Vaithi	18	Female	Vavuniya	sri road	777882782	315612753	6
19	Saranya	Vaithi	14	Female	Galle	mullai road	777107099	212223464	1
20	king	Maker	18	Male	Kandy	buddha road	777107098	212222210	2
21	Delaselp	Ramesh	39	Male	Galle	Keerimalai road	777118530	212222222	3

21 rows in set (0.00 sec)

```
mysql> insert into
patient(Patient_id,first_name,last_name,patient_age,gender,city,street,patient_mobile_contact_no,patient_landline_no,doctor_doctor_id)
values(22,"Partheeban","Sinthu","31","Female","Chunnakam","Road
Street",07770343223,0215561201,1);
```


Query OK, 1 row affected (0.10 sec)

mysql> select * from patient;

Patient_id	First_name	Last_name	Patient_age	Gender	City	Street	Patient_Mobile_Contact_No	Patient_Landline_no	Doctor_Doctor_id
1	Nirojan	Stany	24	Male	Jaffna	Main Street	769545654	212234567	1
2	Sanjsi	Thiru	23	Male	Vavuniya	Vipulanathan road	765432123	212224470	2
3	Antony	Pethurshan	19	Male	Galle	Kandhi Road	765497345	212250646	3
4	Mathu	Sukumar	43	Male	Kandy	1st Cross Street	769456345	212051945	4
5	Prana	Nadaraja	55	Male	Thincomalee	3rd cross Street	712393256	213737878	5
6	Anistan	Lillian	33	Male	Mannar	Kandy road	728745090	213336543	6
7	Anton	Nirojan	51	Male	Galle	Church road	722212334	212229910	1
8	Thaas	Jegan	21	Male	Mannar	``bastian road	779345128	213456712	2
9	Kunam	Anshalon	22	Male	Vavuniya	main street	776523491	212349456	3
10	Mohammad	Hussain	51	Male	Jaffna	irupalai road	745634812	312349434	4
11	Mohammad	Niyas	30	Male	Galle	Tellipalai road	723476410	212226603	5
12	Thananji	Anthoni	32	Male	Galle	kumankulam road	779933561	213217710	6
13	Aravinthan	Alex	19	Male	Galle	Pandarikulam road	764523995	213437345	1
14	Thiviyan	Satha	23	Male	Kandy	Vanuniya main street	779323456	212235523	2
15	Mayooran	Mani	25	Male	Colombo	Chunnakam road	712366124	217774523	3
16	Prathap	Theivam	60	Male	Jaffna	Main street	754577012	212250567	4
17	Shankavi	Paramsoothy	59	Female	Mannar	Lanka road	777716300	216674377	5
18	Samitha	Vaithi	18	Female	Vavuniya	sri road	777882782	315612753	6
19	Saranya	Vaithi	14	Female	Galle	mullai road	777107099	212223464	1
20	king	Maker	18	Male	Kandy	buddha road	777107098	212222210	2
21	Delaselp	Ramesh	39	Male	Galle	Keerimalai road	777118530	212222222	3
22	Partheeban	Sinthu	31	Female	Chunnakam	Road Street	770343223	215561201	1

22 rows in set (0.00 sec)

mysql>

In this patient table I inserted an entry as 22nd Patient_id you can see in the last row/

Update

```
mysql> update house_officer set
house_officer_name="Gnanan",Working_time="Afternoon" where
house_officer_id=1111;
Query OK, 1 row affected (0.11 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> select * from house_officer;
```

House_officer_id	House_officer_name	Working_time
1111	Gnanan	Afternoon
1112	Smitha	Evening
1113	Sakila	Afternoon
1114	Vairamuththu	Morning
1115	Karky	Morning
1116	Kushi	Afternoon

```
6 rows in set (0.00 sec)
```

In this table I update the name and working time of a worker holding id "1111".

Delete

```
mysql> delete from file_info where file_no=001;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from file_info;
```

File_no	Patient_Patient_id	Patient_Doctor_Doctor_id	Consultation_Prescription_no	Consultation_Patients_Appointment_no
111	1	1	121	11
112	2	2	122	12
113	3	3	123	13
114	4	4	124	14
115	5	5	125	15
116	6	6	126	16

```
6 rows in set (0.00 sec)
```

```
mysql>
```

Alter

```
mysql> alter table patient add column status enum("Single","Married")
after last_name;
```

Query OK, 22 rows affected (0.13 sec)

Records: 22 Duplicates: 0 Warnings: 0

```
mysql> select * from patient;
```

Patient_id	First_name	Last_name	status	Patient_age	Gender	City	Street	Patient_Mobile_Contact_No	Patient_Landline_no	Doctor_Doctor_id
1	Nirojan	Stany	NULL	24	Male	Jaffna	Main Street	769545654	212234567	1
2	Sanjsi	Thiru	NULL	23	Male	Vavuniya	Vipulanathan road	765432123	212224470	2
3	Antony	Pethurshan	NULL	19	Male	Galle	Kandhi Road	765497345	212250646	3
4	Mathu	Sukumar	NULL	43	Male	Kandy	1st Cross Street	769456345	212051945	4
5	Prana	Nadaraja	NULL	55	Male	Thincomalee	3rd cross Street	712393256	213737878	5
6	Anistan	Lillian	NULL	33	Male	Mannar	Kandy road	728745090	213336543	6
7	Anton	Nirojan	NULL	51	Male	Galle	Church road	722212334	212229910	1
8	Thaas	Jegan	NULL	21	Male	Mannar	``bastian road	779345128	213456712	2
9	Kunam	Anshalon	NULL	22	Male	Vavuniya	main street	776523491	212349456	3
10	Mohammad	Hussain	NULL	51	Male	Jaffna	irupalai road	745634812	312349434	4
11	Mohammad	Niyas	NULL	30	Male	Galle	Tellipalai road	723476410	212226603	5
12	Thananji	Anthoni	NULL	32	Male	Galle	kumankulam road	779933561	213217710	6
13	Aravinthan	Alex	NULL	19	Male	Galle	Pandarikulam road	764523995	213437345	1
14	Thiviyan	Satha	NULL	23	Male	Kandy	Vanuniya main street	779323456	212235523	2
15	Mayooran	Mani	NULL	25	Male	Colombo	Chunnakam road	712366124	217774523	3
16	Prathap	Theivam	NULL	60	Male	Jaffna	Main street	754577012	212250567	4
17	Shankavi	Paramssothy	NULL	59	Female	Mannar	Lanka road	777716300	216674377	5
18	Samitha	Vaithi	NULL	18	Female	Vavuniya	sri road	777882782	315612753	6
19	Saranya	Vaithi	NULL	14	Female	Galle	mullai road	777107099	212223464	1
20	king	Maker	NULL	18	Male	Kandy	buddha road	777107098	212222210	2
21	Delaselp	Ramesh	NULL	39	Male	Galle	Keerimalai road	777118530	212222222	3
22	Partheeban	Sinthu	NULL	31	Female	Chunnakam	Road Street	770343223	215561201	1

22 rows in set (0.00 sec)

```
mysql>
```

In this table I alter patient table and add a column name status.

Modify

```
mysql> alter table patient MODIFY Patient_age INT(15);
Query OK, 21 rows affected (0.13 sec)
Records: 21 Duplicates: 0 Warnings: 0
```

In this I have modified the patient table and change the varchar data type of age into integer type.

```
mysql> desc patient;
```

Field	Type	Null	Key	Default	Extra
Patient_id	int(11)	NO	PRI	NULL	
First_name	varchar(45)	NO		NULL	
Last_name	varchar(45)	NO		NULL	
Patient_age	int(15)	YES		NULL	
Gender	enum('Male','Female')	NO		NULL	
City	varchar(45)	NO		NULL	
Street	varchar(45)	NO		NULL	
Patient_Mobile_Contact_No	int(11)	NO		NULL	
Patient_Landline_no	int(11)	NO		NULL	
Doctor_Doctor_id	int(11)	NO	PRI	NULL	

10 rows in set (0.10 sec)

```
mysql>
```

Minimum, Maximum, Average, Sum, Count

Minimum

```
mysql> select reference_no as Min_details_referenceno_drugs,
doctors_charge as OUTPUT_Min from discharge_information where
hospital_fees=(select Min(hospital_fees)from discharge_information);
```

```
+-----+-----+
| Min_details_referenceno_drugs | OUTPUT_Min |
+-----+-----+
|                               24 |          3500 |
+-----+-----+
1 row in set (0.00 sec)
```

```
mysql>
```

I found the minimum reference numbers that discharges and minimum hospital fees of Norman Hospital.

Maximum

```
mysql> select reference_no as MAX_details_referenceno_drugs,
doctors_charge as OUTPUT_MAX from discharge_information where
hospital_fees=(select Max(hospital_fees)from discharge_information);
```

MAX_details_referenceno_drugs	OUTPUT_MAX
21	2000

1 row in set (0.10 sec)

Average

```
mysql> select avg(hospital_fees) from discharge_information;
```

avg(hospital_fees)
11916.666666666666

1 row in set (0.00 sec)

```
mysql> select avg(hospital_fees) and avg(pharmacy_fees) from
discharge_information,pharmacy_information;
```

In this I find the average hospital fees of patients in the Norman Hospital.

```
+-----+-----+
| avg(hospital_fees) and avg(pharmacy_fees) |
+-----+-----+
|                                           1 |
+-----+-----+
1 row in set (0.00 sec)
```

Sum

```
mysql> select reference_no, format (sum(hospital_fees *
doctors_charge),2) total from discharge_information group by
reference_no order by sum(hospital_fees * doctors_charge)desc;
```

```
+-----+-----+
| reference_no | total |
+-----+-----+
|           22 | 45,000,000.00 |
|           21 | 40,000,000.00 |
|           24 | 26,250,000.00 |
|           25 | 17,000,000.00 |
|           26 | 15,750,000.00 |
|           23 | 15,000,000.00 |
+-----+-----+
6 rows in set (0.00 sec)
```

In this table I find the discharge patients reference numbers and calculate the fees of a person when discharged from the hospital.

Count

```
mysql> select (select count (appointment_no) from patients_appointment) as
count_appointment, (select count(pharmacy_customer_no) from
pharmacy_information) as count_pharmacy;
```

Count_appoin tment	Count_pharmacy
20	11

1 row in set (0.14 sec)

Display Ages Between 10 and 50.

```
mysql> select
first name,last name,dateofbirth,timestampdiff(year,dateofbirth,curdate())as age, city,patient mobile contact no from patient where
timestampdiff(year,dateofbirth,curdate())between 10 and 50;
```

first_name	last_name	dateofbirth	age	city	patient_mobile_contact_no
Nirojan	Stany	1995-03-21	23	Jaffna	769545654
Sanjsi	Thiru	1996-01-21	22	Vavuniya	765432123
Antony	Pethurshan	2000-05-30	18	Galle	765497345
Mathu	Sukumar	1976-10-12	42	Kandy	769456345
Anistan	Lillian	1986-05-20	32	Mannar	728745090
Anton	Nirojan	1968-10-12	50	Galle	722212334
Thaas	Jegan	1998-12-18	20	Mannar	779345128
Kunam	Anshalon	1997-10-23	21	Vavuniya	776523491
Mohammad	Niyas	1989-01-01	30	Galle	723476410
Thananji	Anthoni	1987-12-14	31	Galle	779933561
Aravinthan	Alex	2000-06-22	18	Galle	764523995
Thiviyan	Satha	1996-12-03	22	Kandy	779323456
Mayooran	Mani	1994-09-18	24	Colombo	712366124
Samitha	Vaithi	2001-12-01	17	Vavuniya	777882782
Saranya	Vaithi	2005-09-21	13	Galle	777107099
king	Maker	2001-12-18	17	Kandy	777107098
Delaselp	Ramesh	1980-12-31	38	Galle	777118530

17 rows in set (0.00 sec)

In this I derive the details and age from 10 and 50 from a single table.

```
mysql> select
first_name,last_name,dateofbirth,timestampdiff(year,dateofbirth,curdate())as age,city,
doctor.doctor_name,patients_appointment.appointment_no
from patient,doctor,patients_appointment
where timestampdiff(year,dateofbirth,curdate())between 10 and 50;
```

first_name	last_name	dateofbirth	age	city	doctor_name	appointment_no
Nirojan	Stany	1995-03-21	23	Jaffna	Logetha	11
Nirojan	Stany	1995-03-21	23	Jaffna	Logetha	12
Nirojan	Stany	1995-03-21	23	Jaffna	Logetha	13
Nirojan	Stany	1995-03-21	23	Jaffna	Logetha	14
Nirojan	Stany	1995-03-21	23	Jaffna	Logetha	15
Nirojan	Stany	1995-03-21	23	Jaffna	Logetha	16
Nirojan	Stany	1995-03-21	23	Jaffna	Mathunujan	11
Nirojan	Stany	1995-03-21	23	Jaffna	Mathunujan	12
Nirojan	Stany	1995-03-21	23	Jaffna	Mathunujan	13
Nirojan	Stany	1995-03-21	23	Jaffna	Mathunujan	14
Nirojan	Stany	1995-03-21	23	Jaffna	Mathunujan	15
Nirojan	Stany	1995-03-21	23	Jaffna	Mathunujan	16
Nirojan	Stany	1995-03-21	23	Jaffna	Vijay	11
Nirojan	Stany	1995-03-21	23	Jaffna	Vijay	12
Nirojan	Stany	1995-03-21	23	Jaffna	Vijay	13
Nirojan	Stany	1995-03-21	23	Jaffna	Vijay	14
Nirojan	Stany	1995-03-21	23	Jaffna	Vijay	15
Nirojan	Stany	1995-03-21	23	Jaffna	Vijay	16
Nirojan	Stany	1995-03-21	23	Jaffna	Ajith	11
Nirojan	Stany	1995-03-21	23	Jaffna	Ajith	12
Nirojan	Stany	1995-03-21	23	Jaffna	Ajith	13
Nirojan	Stany	1995-03-21	23	Jaffna	Ajith	14
Nirojan	Stany	1995-03-21	23	Jaffna	Ajith	15
Nirojan	Stany	1995-03-21	23	Jaffna	Ajith	16
Nirojan	Stany	1995-03-21	23	Jaffna	Simran	11
Nirojan	Stany	1995-03-21	23	Jaffna	Simran	12
Nirojan	Stany	1995-03-21	23	Jaffna	Simran	13
Nirojan	Stany	1995-03-21	23	Jaffna	Simran	14
Nirojan	Stany	1995-03-21	23	Jaffna	Simran	15
Nirojan	Stany	1995-03-21	23	Jaffna	Simran	16
Nirojan	Stany	1995-03-21	23	Jaffna	Samantha	11
Nirojan	Stany	1995-03-21	23	Jaffna	Samantha	12
Nirojan	Stany	1995-03-21	23	Jaffna	Samantha	13
Nirojan	Stany	1995-03-21	23	Jaffna	Samantha	14
Nirojan	Stany	1995-03-21	23	Jaffna	Samantha	15
Nirojan	Stany	1995-03-21	23	Jaffna	Samantha	16
Nirojan	Stany	1995-03-21	23	Jaffna	Kushpu	11
Nirojan	Stany	1995-03-21	23	Jaffna	Kushpu	12
Nirojan	Stany	1995-03-21	23	Jaffna	Kushpu	13
Nirojan	Stany	1995-03-21	23	Jaffna	Kushpu	14
Nirojan	Stany	1995-03-21	23	Jaffna	Kushpu	15
Nirojan	Stany	1995-03-21	23	Jaffna	Kushpu	16
Sanjsi	Thiru	1996-01-21	22	Vavuniya	Logetha	11
Sanjsi	Thiru	1996-01-21	22	Vavuniya	Logetha	12
Sanjsi	Thiru	1996-01-21	22	Vavuniya	Logetha	13
Sanjsi	Thiru	1996-01-21	22	Vavuniya	Logetha	14

In this table I derive the age information from patient, doctor and patient appointment.

king	Maker	2001-12-18	17	Kandy	Samantha	11
king	Maker	2001-12-18	17	Kandy	Samantha	12
king	Maker	2001-12-18	17	Kandy	Samantha	13
king	Maker	2001-12-18	17	Kandy	Samantha	14
king	Maker	2001-12-18	17	Kandy	Samantha	15
king	Maker	2001-12-18	17	Kandy	Samantha	16
king	Maker	2001-12-18	17	Kandy	Kushpu	11
king	Maker	2001-12-18	17	Kandy	Kushpu	12
king	Maker	2001-12-18	17	Kandy	Kushpu	13
king	Maker	2001-12-18	17	Kandy	Kushpu	14
king	Maker	2001-12-18	17	Kandy	Kushpu	15
king	Maker	2001-12-18	17	Kandy	Kushpu	16
Delaselp	Ramesh	1980-12-31	38	Galle	Logetha	11
Delaselp	Ramesh	1980-12-31	38	Galle	Logetha	12
Delaselp	Ramesh	1980-12-31	38	Galle	Logetha	13
Delaselp	Ramesh	1980-12-31	38	Galle	Logetha	14
Delaselp	Ramesh	1980-12-31	38	Galle	Logetha	15
Delaselp	Ramesh	1980-12-31	38	Galle	Logetha	16
Delaselp	Ramesh	1980-12-31	38	Galle	Mathunujan	11
Delaselp	Ramesh	1980-12-31	38	Galle	Mathunujan	12
Delaselp	Ramesh	1980-12-31	38	Galle	Mathunujan	13
Delaselp	Ramesh	1980-12-31	38	Galle	Mathunujan	14
Delaselp	Ramesh	1980-12-31	38	Galle	Mathunujan	15
Delaselp	Ramesh	1980-12-31	38	Galle	Mathunujan	16
Delaselp	Ramesh	1980-12-31	38	Galle	Vijay	11
Delaselp	Ramesh	1980-12-31	38	Galle	Vijay	12
Delaselp	Ramesh	1980-12-31	38	Galle	Vijay	13
Delaselp	Ramesh	1980-12-31	38	Galle	Vijay	14
Delaselp	Ramesh	1980-12-31	38	Galle	Vijay	15
Delaselp	Ramesh	1980-12-31	38	Galle	Vijay	16
Delaselp	Ramesh	1980-12-31	38	Galle	Ajith	11
Delaselp	Ramesh	1980-12-31	38	Galle	Ajith	12
Delaselp	Ramesh	1980-12-31	38	Galle	Ajith	13
Delaselp	Ramesh	1980-12-31	38	Galle	Ajith	14
Delaselp	Ramesh	1980-12-31	38	Galle	Ajith	15
Delaselp	Ramesh	1980-12-31	38	Galle	Ajith	16
Delaselp	Ramesh	1980-12-31	38	Galle	Simran	11
Delaselp	Ramesh	1980-12-31	38	Galle	Simran	12
Delaselp	Ramesh	1980-12-31	38	Galle	Simran	13
Delaselp	Ramesh	1980-12-31	38	Galle	Simran	14
Delaselp	Ramesh	1980-12-31	38	Galle	Simran	15
Delaselp	Ramesh	1980-12-31	38	Galle	Simran	16
Delaselp	Ramesh	1980-12-31	38	Galle	Samantha	11
Delaselp	Ramesh	1980-12-31	38	Galle	Samantha	12
Delaselp	Ramesh	1980-12-31	38	Galle	Samantha	13
Delaselp	Ramesh	1980-12-31	38	Galle	Samantha	14
Delaselp	Ramesh	1980-12-31	38	Galle	Samantha	15
Delaselp	Ramesh	1980-12-31	38	Galle	Samantha	16
Delaselp	Ramesh	1980-12-31	38	Galle	Kushpu	11
Delaselp	Ramesh	1980-12-31	38	Galle	Kushpu	12
Delaselp	Ramesh	1980-12-31	38	Galle	Kushpu	13
Delaselp	Ramesh	1980-12-31	38	Galle	Kushpu	14
Delaselp	Ramesh	1980-12-31	38	Galle	Kushpu	15
Delaselp	Ramesh	1980-12-31	38	Galle	Kushpu	16

714 rows in set (0.01 sec)

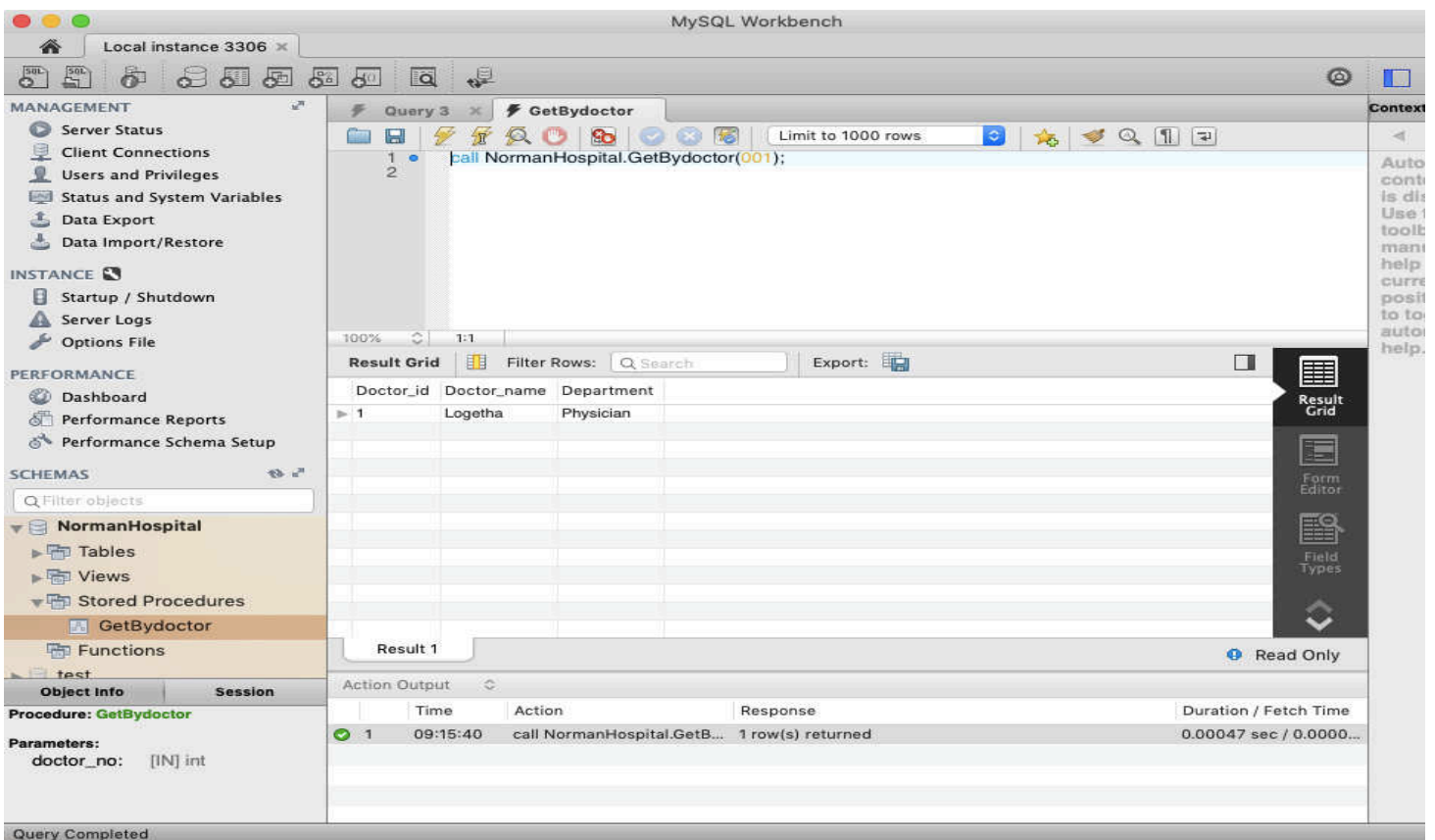
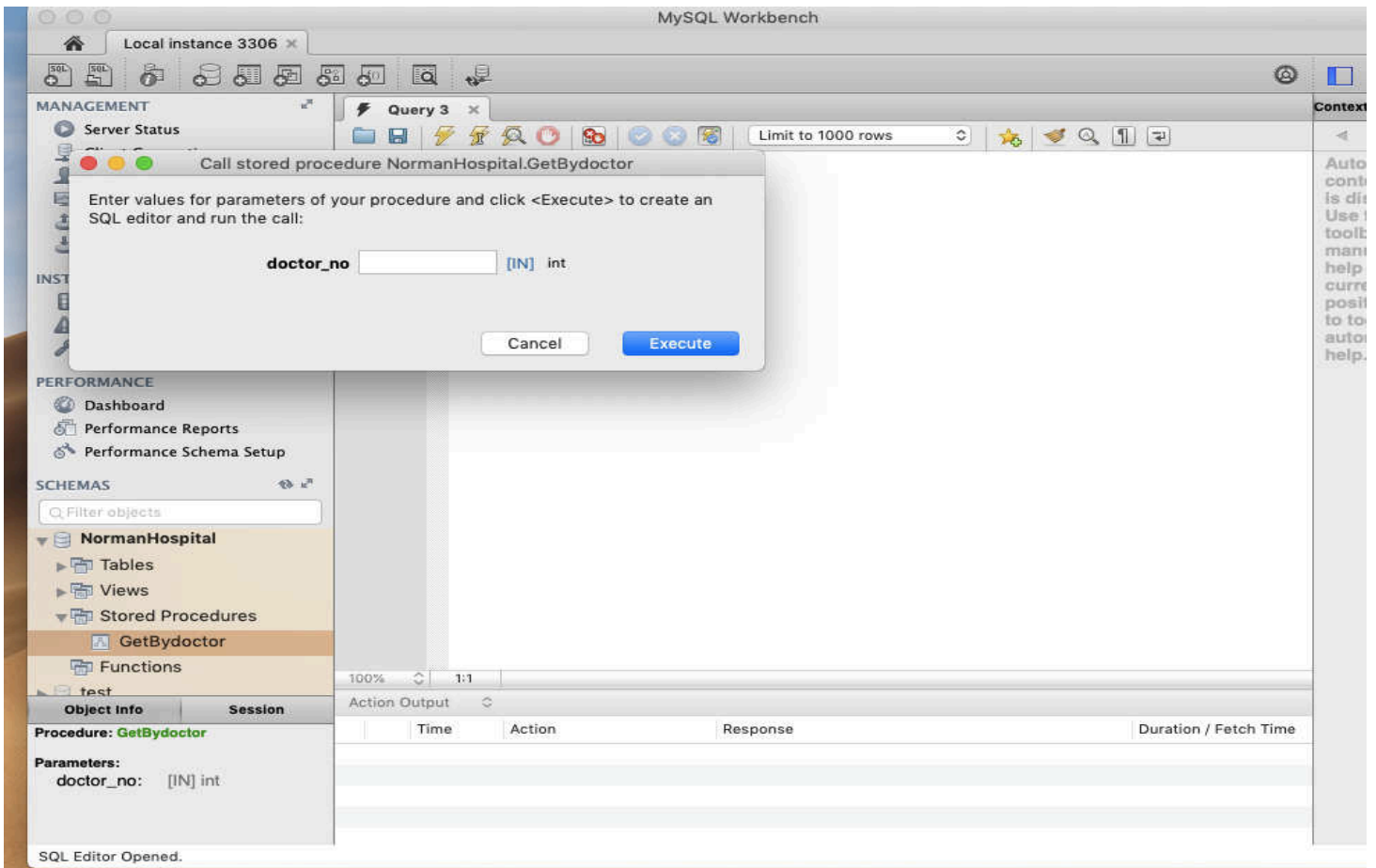
mysql>

Stored Procedures

A stored procedure is a subroutine available to applications that access a relational database management system. Such procedures are stored in the database data dictionary. Uses for stored procedures include data-validation or access-control mechanisms



In this I created Stored procedure for Doctors information. And these images are clearly defined by the steps.



View

I created a view on Norman view and it show some details.

```
mysql> create view Normanview as select first_name,gender,patient_age from patient where patient_id=01;
Query OK, 0 rows affected (0.12 sec)
```

```
mysql> select * from normanview;
+-----+-----+-----+
| first_name | gender | patient_age |
+-----+-----+-----+
| Nirojan   | Male   | 24          |
+-----+-----+-----+
1 row in set (0.10 sec)
```

```
mysql>
```

And I created another view but there is no values and empty set.

```
mysql> create view norman_view as select
patient.first_name,consultation_prescription.prescription_no,doctor.
doctor_name from patient,consultation_prescription inner join doctor
on doctor.doctor_name=consultation_prescription.
Patients_Appointment_Doctor_id;
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> select * from norman_view;
Empty set, 42 warnings (0.00 sec)
```

```
mysql>
```

Trigger

Before Update Trigger

```
mysql> create table patient_audit(id int auto_increment primary key,
first_name varchar(30), doc_id int, changedat Datetime default null,
action varchar(50) default null);
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> desc patient_audit;
```

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	YES		NULL	
doc_id	int(11)	YES		NULL	
changedat	datetime	YES		NULL	
action	varchar(50)	YES		NULL	

```
5 rows in set (0.00 sec)
```

```
mysql> delimiter $$
mysql> create trigger Before_Update_Patients
-> before update on patient
-> for each row
-> begin
-> insert into patient_audit
-> set action="update",
-> id=old.patient_id,
-> first_name=old.first_name,
-> changedat=now();
-> end$$
```

```
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> desc patient;
-> desc patient;
-> end$$
```

```
mysql> update patient set first_name="Albert" where patient_id=01;
-> end$$
Query OK, 1 row affected (0.11 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from patient_audit;
-> end$$
+----+-----+-----+-----+-----+
| id | first_name | doc_id | changedat | action |
+----+-----+-----+-----+-----+
| 1 | Nirojan | NULL | 2019-01-05 09:50:25 | update |
+----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Before Update Trigger

```
mysql> create table patient_afteraudit(id int auto_increment primary
key, last_name varchar(50),doc_id int, changedat Datetime default
null,action varchar(50) default null);
-> end$$
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> create trigger After_Update_Patients
-> after update on patient
-> for each row
-> begin
-> insert into patient_afteraudit
-> set action="update",
-> id=old.patient_id,
-> last_name=old.first_name,
-> changedat=now();
-> end$$
Query OK, 0 rows affected (0.15 sec)
```



```
mysql> update patient set first_name="Anton" where patient_id=02;
-> end$$
Query OK, 1 row affected (0.10 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from patient_afteraudit;
-> end$$
```

id	last_name	doc_id	changedat	action
2	Sanjsi	NULL	2019-01-05 10:07:19	update

1 row in set (0.00 sec)

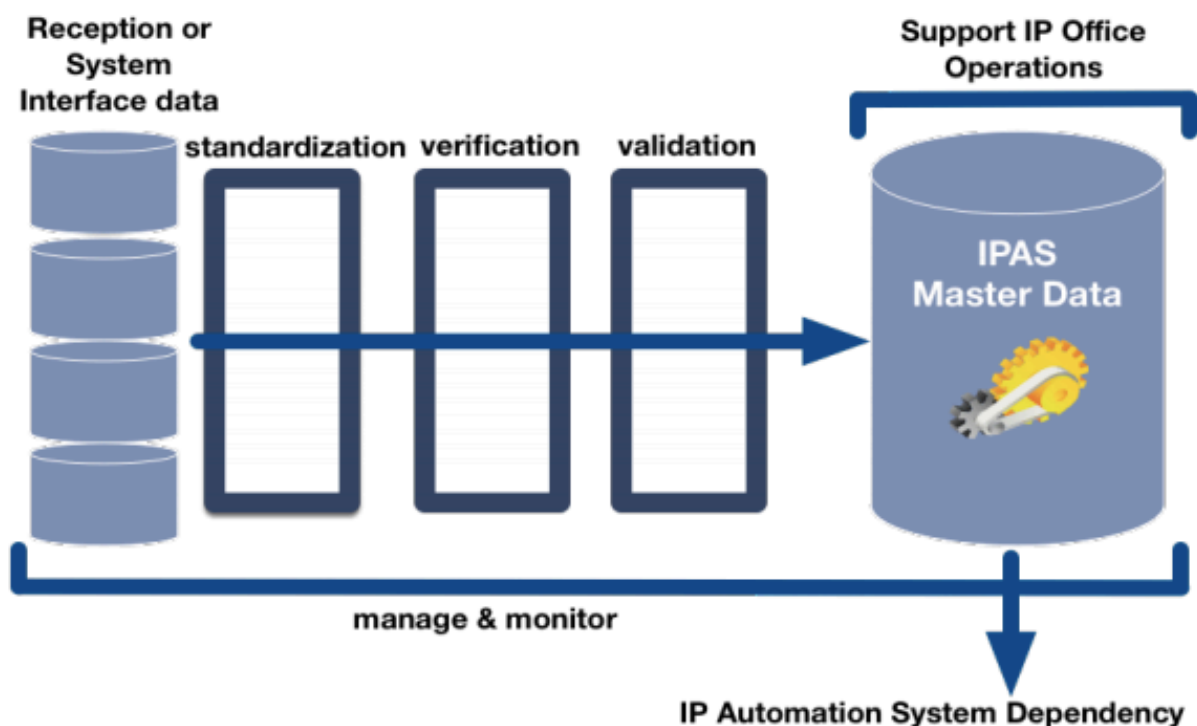
L03

Part 3.1: Explain the Data Validation & Verification Process.

Data validation and verification are very important in the software industry. It's mainly used to avoid the duplicate and wrong data entry in the data basis. The software systems basically maintained to use in the organizations, companies and shops to schedule the jobs in the easier manner. Verification and validation identify the important errors or flaws.

The main concept of verifications is "Are we building the product, right?" and validation is "are we building the right product?". The software industries carried out many projects like management systems and these management related systems are implemented as web based and window based. In this application data validation part is very important and there are so many validation methods are carried out by the developers.

Building the Best & Reliable IP Database



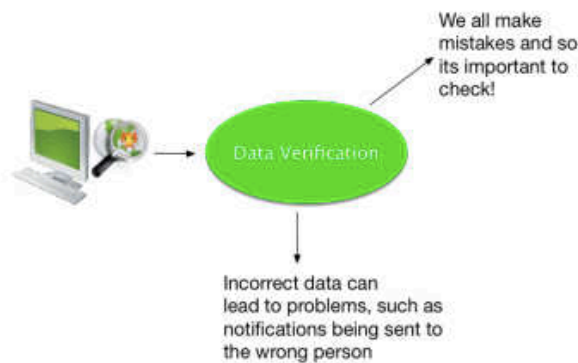
Data Verification

Data verification is the process of evaluating the completeness, correct and conformance of a specific data set by various methods.

The project leaders, QA manager, laboratory director, air monitoring personal assist and the everyone who are participating in the production of data.

Steps of Verification

Why do you check the data?



- Identify the needs, records and technical specifications – Logbooks, Electronic data.
- Compare records and data documentation against the method and requirements – quality assurance project plan and standard operating procedure.

Outputs of Verification

- Verified Data.
- Data Verified Record – certification statement that should be signed by the responsible person and it also identify the non-compliance issues.

The data verification process involves the inspection, analysis, and acceptance of the field data or samples and these inspections can take the form of technical systems audits that is internal or external or frequent inspections by field operators and lab

Ozone Criteria – Alliance	Yes	No	Comments
1-point check done every 2 weeks?	☒	☐	
Zero/span check done every 2 weeks?	☒	☐	
QC points within +/- 7% of std value?	☒	☐	
Shelter temp maintained within 20-30 Degrees C?	☒	☐	
Shelter temp < +/- 2 Degrees C SD over 24 hrs?	☐	☒	6/3 & 6/4, temp varied by >2°
Maintenance performed as scheduled? (see maintenance checklist)	☒	☐	
Other comments:			

CO Criteria	Yes	No	Comments
1-point check done every 2 weeks?	☒	☐	
Zero/span check done every 2 weeks?	☒	☐	
QC points within +/- 10% of std value?	☒	☐	
Shelter temp maintained within 20-30 Degrees C?	☒	☐	
Shelter temp < +/- 2 Degrees C SD over 24 hrs?	☒	☐	
Maintenance performed as scheduled? (see maintenance checklist)	☒	☐	
Other comments:			
Other comments:			

Signature: XXXXXXXX	Date: 7/20/2016
--	-----------------

technicians.

These are some of the sample questions ask during the process of verification.

01. Were the environmental data operations performed according to the governing those operations?
02. Were the environmental data operations performed on the correct time and date originally specified?
03. Did the sampler or monitor perform correctly.

Data Validation

Data validation is the routine process designed to ensure that reported values meet the goals of the environmental data operations and data validation is further defined as examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled.

The systematic approach of data validation is used to ensure and assess the quality of data. The purpose of data validation is to detect and then verify any data values that may not represent actual air quality conditions at the sampling station. The effective data validation procedures usually are handled completely independently from the procedures of initial data collection.

Flags are used alone or in combination to invalidate samples. Specially the appropriate null data flags be used in place of any routine values that are invalidated. This provides some indications to data users and data quality assessors to the reasons why data that was expected to be collected was missing.

There are so many methods such as

- Required field validation
- Range validation
- Compare validation
- Regular expression validation
- Custom validation
- Field validation
- Property validation

These validation methods are very important to develop a software and but mainly managed systems focus on following validations such as

- Empty field validations
- Numeric field validations
- Text field validation
- Email address validation
- Range validation
- Compare validation

Data Field	Data Type	Validation
Name	Varchar	Text field validation
Mobile Number	Integer	Numeric field validation
Email	Varchar	Email field validation
Duration	Date	Range validation
Before insert data	All data type	Empty field validation

Most important point is Data validator must not be the person who developed or producing the data.

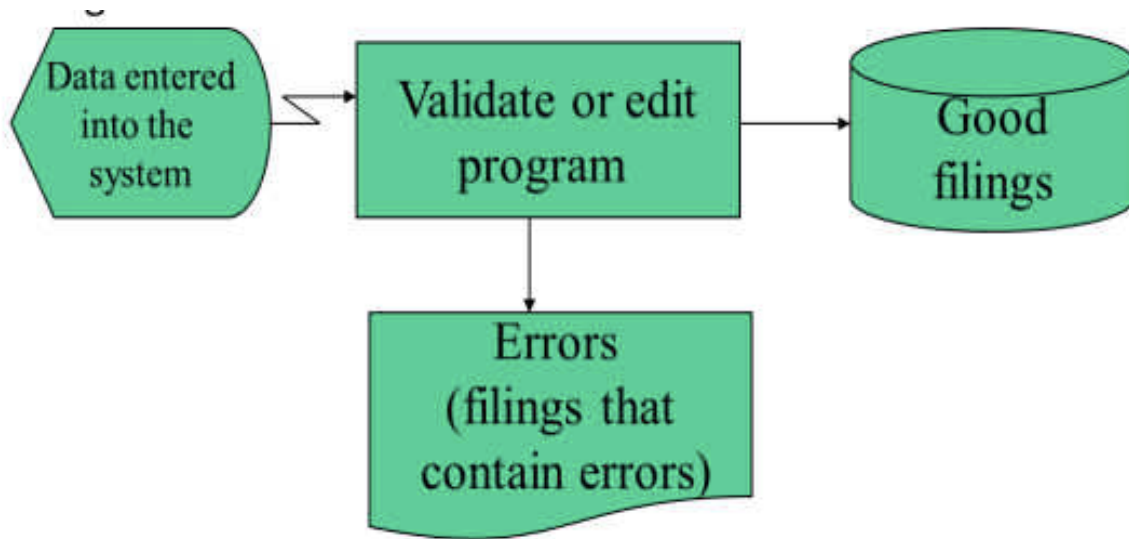
Steps of Validation

- Collect verification records and other needed records – Instrument and Site logbooks.
- Review of the records to determine the quality of data – were the project meet the needs.

CO Criteria	Yes	No	Comments
Flow cells certified in last 12 months?	☒	☐	
In past 6 months, was monitor cal'd?	☒	☐	
Does data fall within expected range of values? Address any outliers.	☒	☐	
Are any trends noticed in performance checks?	☐	☒	
Are any trends noticed in data?	☐	☒	
Was recent audit within acceptable range? Address trends.	☒	☐	
Other comments:			
BGI Criteria – 17D	Yes	No	Comments
In past year, was temp multi-point verification or calibration done?	☐	☒	not multi-pt
In past year, was pressure verified or calibrated?	☒	☐	
In past year or after transport, was flow rate multi-point verification or calibration done?	☒	☐	
Does data fall within expected range of values? Address any outliers.	☒	☐	
Are any trends noticed in performance checks?	☐	☒	
Are any trends noticed in data?	☐	☒	
Was recent audit within acceptable range? Address trends.	☒	☐	
Other comments:			
Signature: XXXXXXXX Date: 6/17/2016			

Outputs of Validation

- Validated Data.
- Data Validation Report.



All data when it initially entered into the system should be checked for errors so that bad data does not get put into permanent disk files. “Garbage In”, “Garbage Out” this process of error checking is called validating or editing.

- Validation and. Verification were most important to maintain the proper database and the most important validation method needs to improve the database management system. The login form was very important in the system for the security of the entire system and this contains so many validations such as empty field, range and compare validations.
- Login boxes are given text box for the users for the user id, password and button for login and cancellation. The user will access the data when the correct information given.

T.NO	Action	Input	Expected Result	Actual Output	Test Result	Test Command
01	Insert	Patient table Patient id in INT value	Value Should be add	Value added and show the column	Value is added.	Pass
02	Update	Update the date in the table discharge information as 2018.03.12	Value should not be added.	Error message shown in date added format	Value is not added.	Invalid Test
03	Insert	Insert double values in discharge table.	Value Should be added.	Shows the values.	Value added.	Pass
04	Delete	Delete a foreign key in doctors visit.	Value should be deleted.	Value should not delete.	Error result.	Error
05	update	Update only doctor id value in doctor table.	Value should not be added.	Value should not add.	No value added.	Invalid Test
06	Delete Row	Delete a row in patient table	Row should be deleted	No values deleted.	Get unexpected result.	Invalid Test
07	Insert	Insert ward number in table	Value should be inserted.	Value added	Get expected result.	Pass

Part 3.2: Error Report

Error report is used to indicate the problems that are faced during the made up of database system and used to make the system more reliable and accurate.

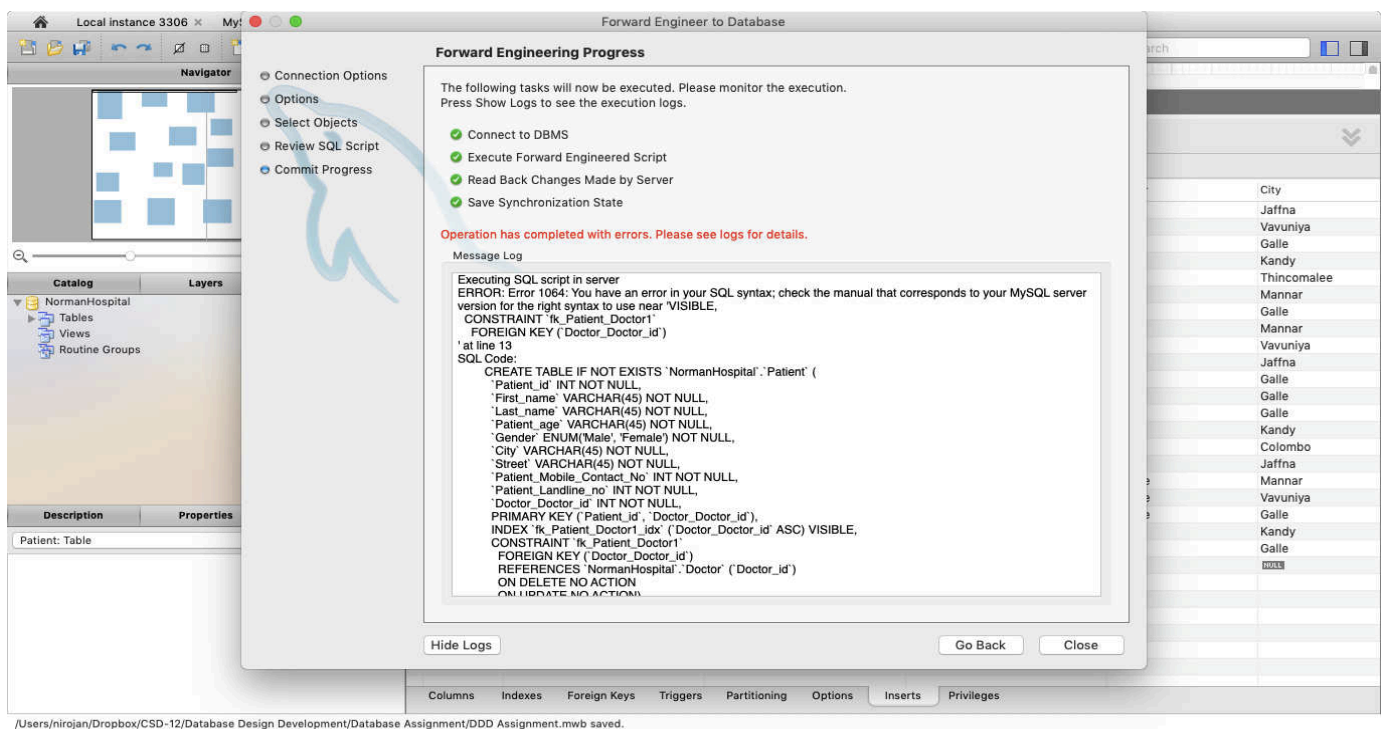
Common Data Errors

We need to consider about some common errors because that are make the system difficult. Examples of some common errors in database system.

Name

- First and Last name are not normally the same.
- No numerals in name fields.
- Suffixes not in last name field.
- First name is not “Husband,” “Wife,” “Man,” “Woman,” “Boy,” “Girl,” “Child”, “Baby,” “Baby Girl,” “Baby Boy” or similar.
- Always confirm the spelling of client first and last names.
- Legal first name (do not add nicknames in “quotes” – those are not searchable elements, add a nickname to the Alias field).

I have faced so many problems when forward engineer. First, I created the tables and details through the workbench and forward engineer that but it was showing the message completed with the errors.



This error was found lots of time because I'm using workbench in MacOS so that some syntax like Visible error appear so after that I delete visible in all my coding's after that is worked.

Minimum Error

```
mysql> select * pharmacy_customer_no,drugs,pharmacy_fees from
pharmacy_information where pharmacy_fees=(select min(pharmacy_fees) from
pharmacy_information);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near
'pharmacy_customer_no,drugs,pharmacy_fees from pharmacy_information where
pharmac' at line 1
mysql> select * pharmacy_customer_no,drugs,pharmacy_fees from
pharmacy_information where pharmacy_fees=(select min(pharmacy_fees) from
pharmacy_information);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near
'pharmacy_customer_no,drugs,pharmacy_fees from pharmacy_information where
pharmac' at line 1
```

This is a small error found when I get the minimum pharmacy fees then I solved out it. The reason for the error is I put * symbol near the select.

```
mysql> select pharmacy_customer_no,drugs,pharmacy_fees from
pharmacy_information where pharmacy_fees=(select min(pharmacy_fees) from
pharmacy_information);
```

```
+-----+-----+-----+
| pharmacy_customer_no | drugs    | pharmacy_fees |
+-----+-----+-----+
|          101 | Vitamin C |          1200 |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Alter Table Error

```
mysql> alter table patient MODIFY Patient_age=int(15);
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near '=int(15)'
at line 1
```

This error I found when I change the details by modifying the patient table and the reason for the error is I put the = sign in between the datatype and name. After that I solved out that.

```
mysql> alter table patient MODIFY Patient_age INT(15);
Query OK, 21 rows affected (0.13 sec)
Records: 21 Duplicates: 0 Warnings: 0
```

Database Error

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| NormanHospital |
| mysql |
| performance_schema |
| test |
+-----+
5 rows in set (0.00 sec)
```

```
mysql> use normanhospitl;
ERROR 1049 (42000): Unknown database 'normanhospitl'
```

This error when I choose the database to implement the database and error comes out because I made some spelling mistakes for the database name “normanhospital” as “normanhospitl”. After that I solved out it.

```
mysql> mysql> use normanhospital;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
```

Sum Error

```
mysql> select reference_no, format (sum(hospital_fees * doctors_charge),2) total
from fees group by reference_no order by sum(hospital_fees * doctors_charge)desc;
ERROR 1146 (42S02): Table 'normanhospital.fees' doesn't exist
```

When I want to find the sum of two calculations there is an error appear that because the fees table does not exist. After that I solved out it.

```
mysql> select reference_no, format (sum(hospital_fees * doctors_charge),2) total
from discharge_information group by reference_no order by sum(hospital_fees *
doctors_charge)desc;
+-----+-----+
| reference_no | total      |
+-----+-----+
| 22 | 45,000,000.00 |
| 21 | 40,000,000.00 |
| 24 | 26,250,000.00 |
6 rows in set (0.00 sec)
```

Delete Error

```
mysql> delete from doctor where Doctor_id=001;
ERROR 1451 (23000): Cannot delete or update a parent row: a foreign key constraint
fails      ('normanhospital`.`hospital_ward_admission`,      CONSTRAINT
`fk_Admission_Appointment_Doctor1` FOREIGN KEY (`Doctor_Doctor_id`)
REFERENCES `Doctor` (`Doctor_id`) ON DELETE NO ACTION ON UPDATE NO
AC)
```

This error when I want to delete a detail from the doctor table the error says that can't delete the parent row. So I solved it.

```
mysql> delete from file_info where file_no=001;
Query OK, 0 rows affected (0.00 sec)
```

These are the error I faced when producing the databases and these errors are make the database system more reliable and comfort to use.

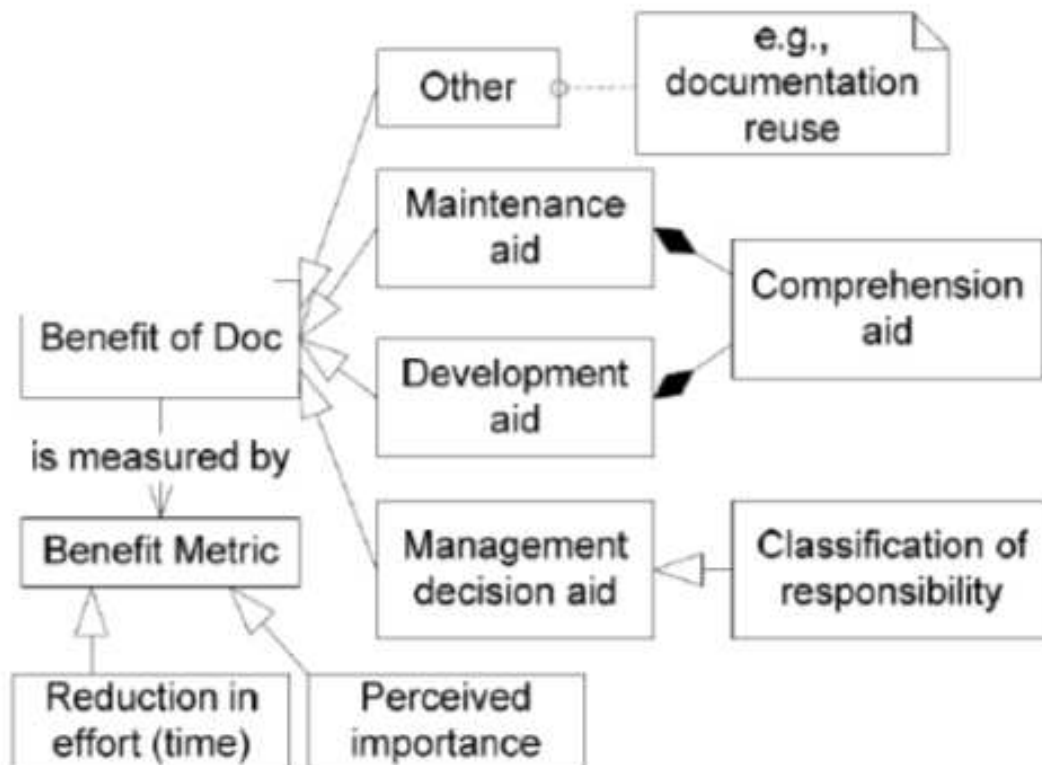
Part 4.1: Produce a user and Technical Manual.

Documentation is about documents, which communicate information. Those documents

provide information for and about a certain object, process or topic. Documentation can be presented in an interactive manner, which increases ease of use.

Technical Manual

technical software documentation includes: requirement specifications, design (architectural) documents, source code comments, test documents and defect (bug) reports.



Data Types and Primary Key, Foreign Keys

In this program there are lots of data types like integer, varchar, float, double, date, time, enum and etc. The foreign keys are used in lot of tables to derive information from one table to another table. All tables has a primary key and some primary key like Patient_id, doctor_id are act as foreign keys in many tables. Because most of the tables want to derive informations from the tables by foreign keys.

Primary Key Symbol – 

Foreign Key Symbol - 

Procedures & Triggers

I created some stored procedures like for doctor table you can view in database and for the protection of the database I have created user privilege named as NormanHospital.

I created the before update and after update triggers to keep the security of the system more flexible. So, if you update any information on the tables it will be added after update audit table and before update trigger table by triggers.

User Manual

Software user documentation is as old as software development itself. The actual users of software systems need information on how to interact with the software system and know how to accomplish their tasks efficiently. These aspects and the fact that documentation often varies in quality and structure have leads to research and also the development of standards around documentation.

It is essential for user documentation to fit the user's needs. An audience analysis has to be done. It is obviously important to capture the purpose of the database system.

Furthermore, the documentation has to capture the work flows which accomplish the various tasks of the domain and what to do in case of errors (troubleshooting). It should

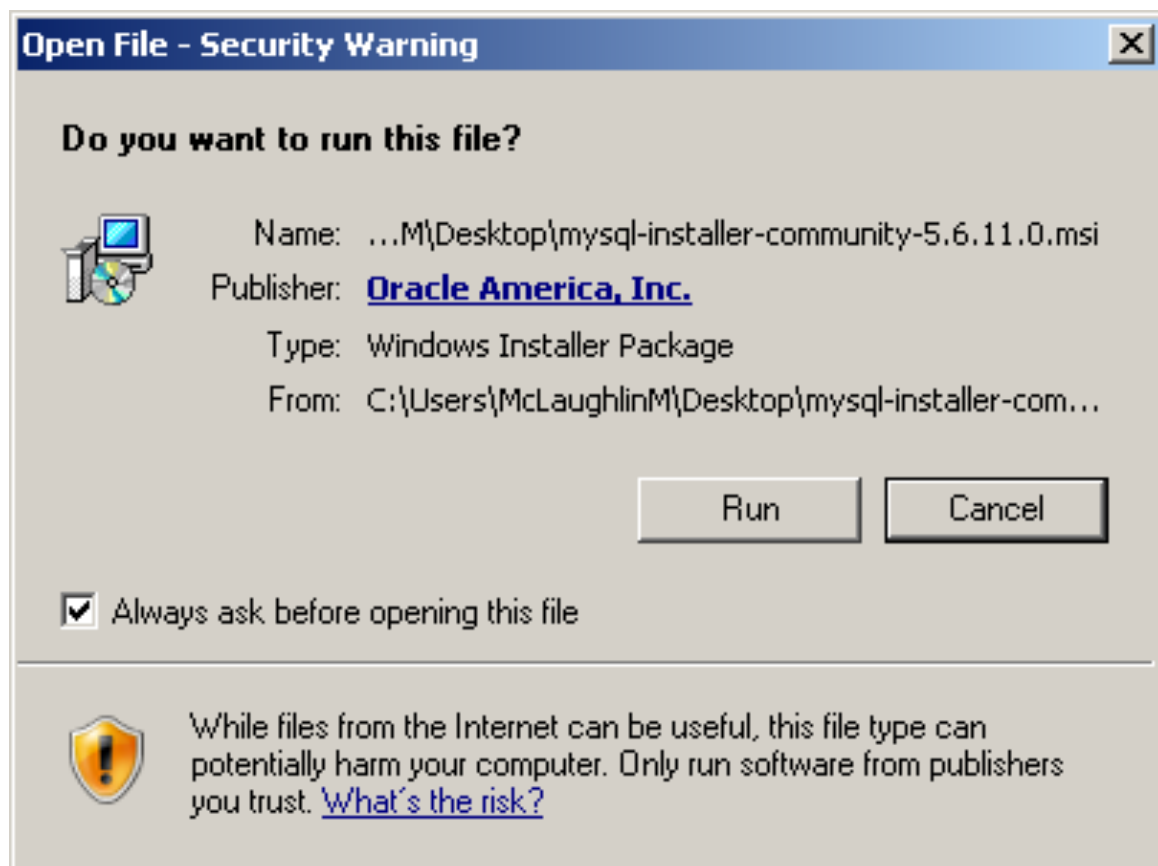
guide the user on the way of accomplishing certain tasks with the database system. Another approach is to explain the database system itself.

Steps of Installing MySQL

Step 1: Download MySQL

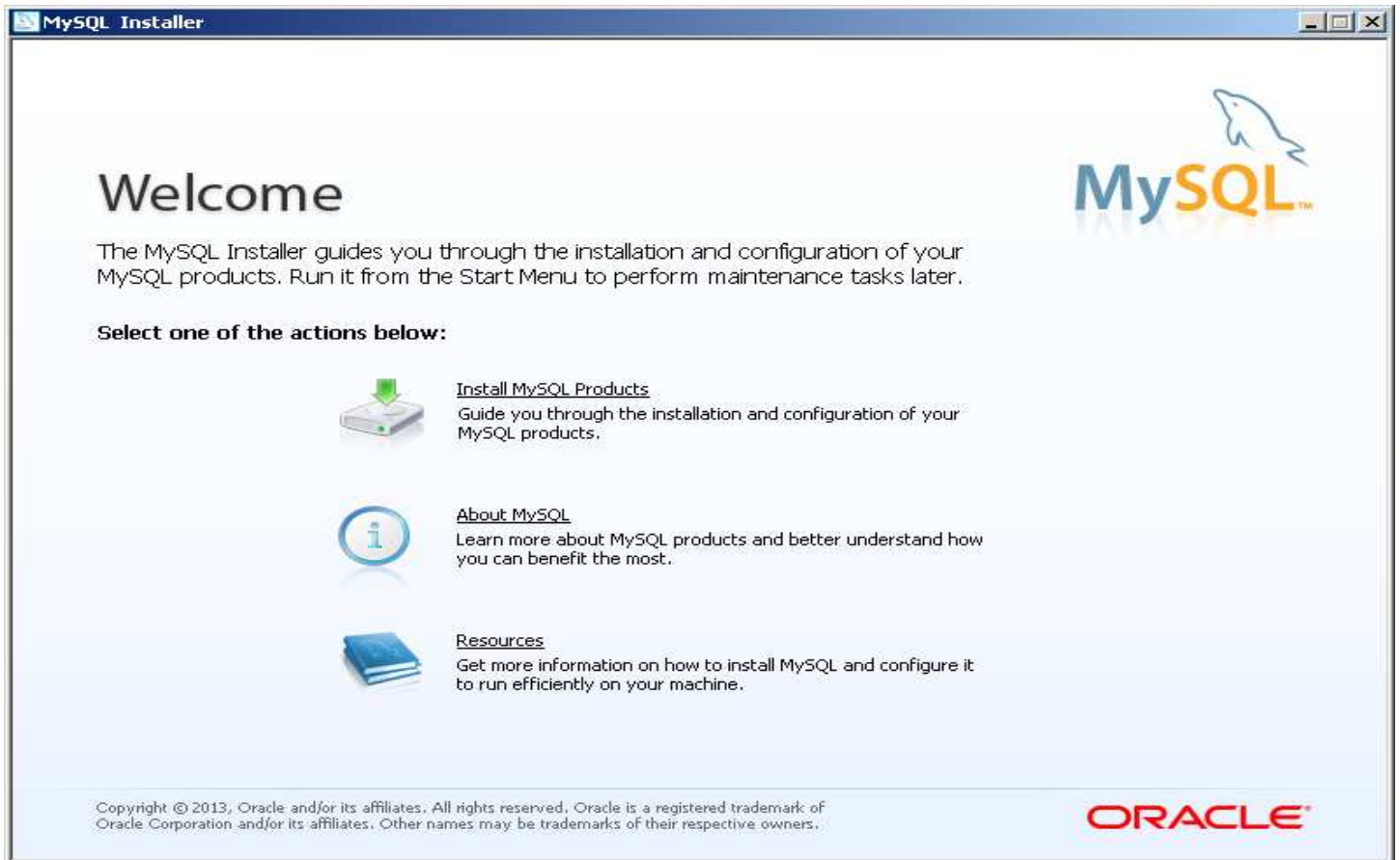
Download MySQL from dev.mysql.com/downloads/.

Step 2: Open and Run MySQL



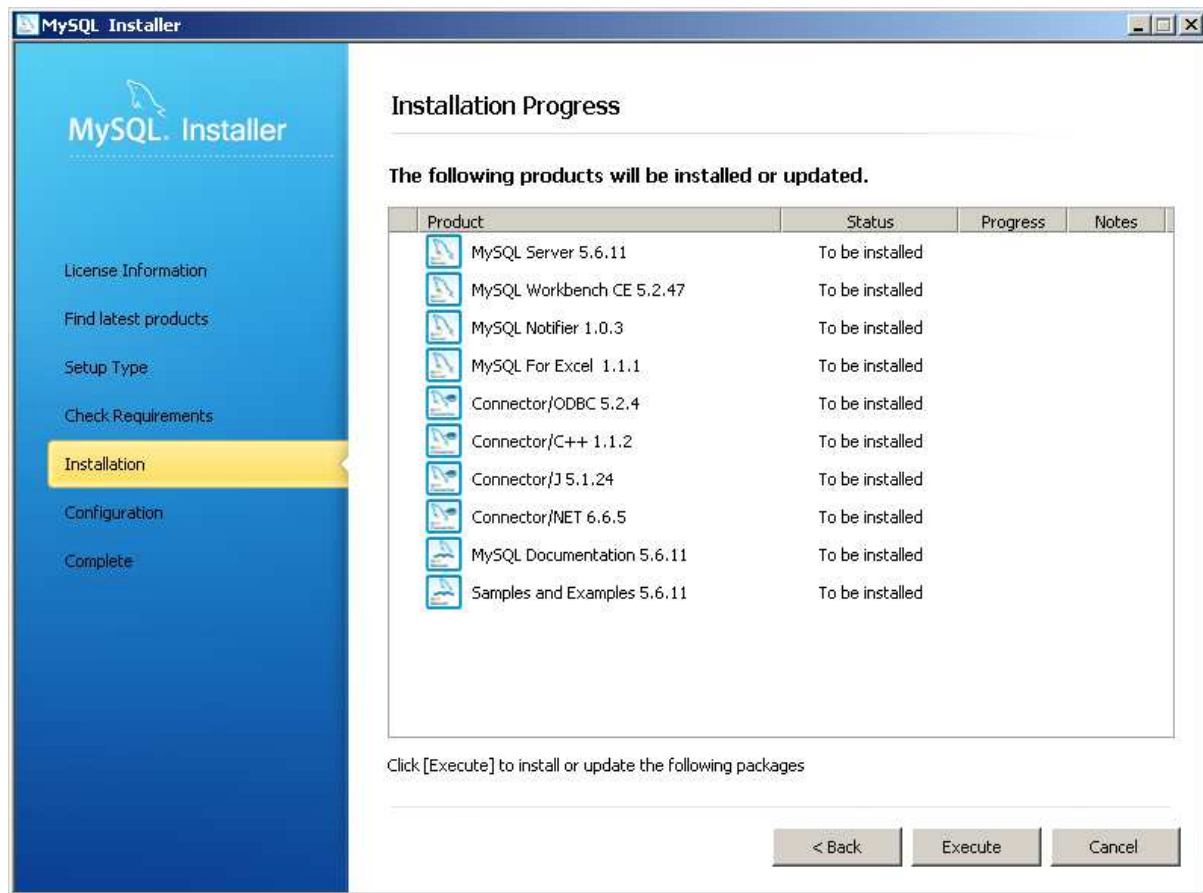
Press Run to Start the configuration and this popup show the location where the mysql will install.

Step 3: Check Physical Require



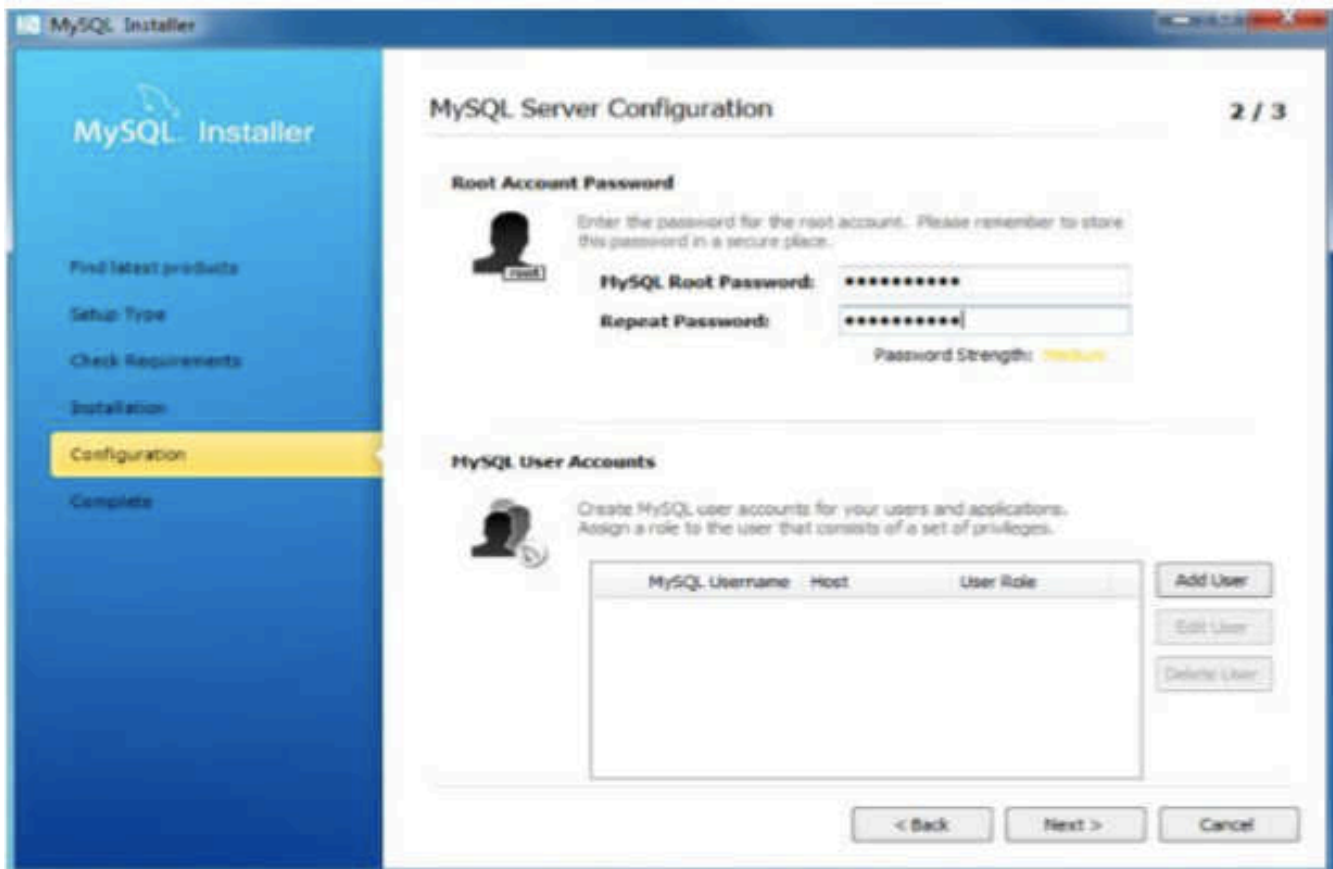
Here you choose what you want to do. You can install MySQL products, inquire about MySQL, or check physical resource components. Provided you installed the pre requisites listed above, you should be prepared to install MySQL Products. Click the Install MySQL Products link to proceed or one of the others to explore.

Step 4: MySQL Servers and Products Installation



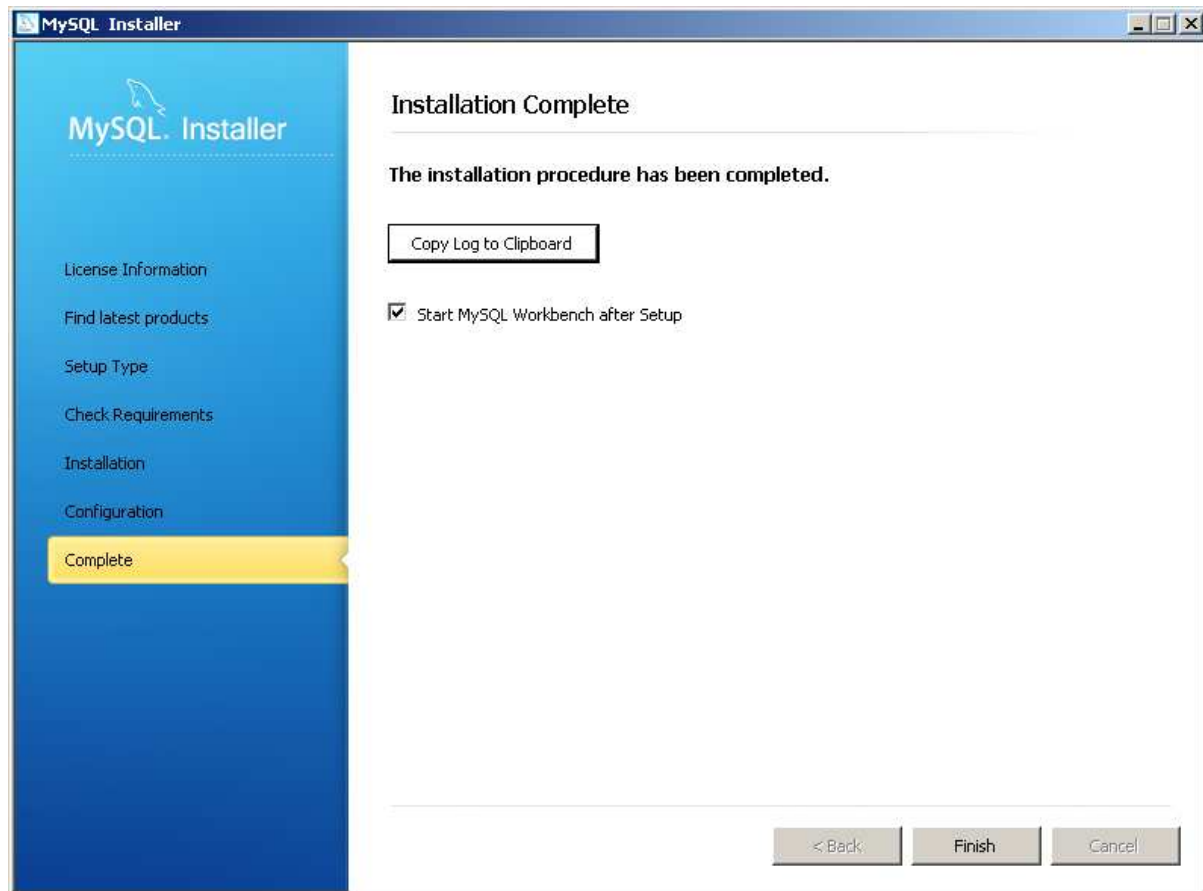
Some of the are we need to press next only and after that this step will appear. Installer screen performs displays the products that it'll install, which are listed below and available in the full image to the left. Click the Execute button to install the products.

Step 5: User name and Password



In this page we can set the specific account password and keep our account secure.

Step 5: Completion



After the execution of MySQL products some steps appear press only next and after that the installation will complete and you can start using it.

My SQL Table:

```
[mysql> desc patients_appointment;
```

Field	Type	Null	Key	Default	Extra
Appointment_no	int(11)	NO	PRI	NULL	
Patient_Patient_id	int(11)	NO	PRI	NULL	
Patient_Doctor_Doctor_id	int(11)	NO	PRI	NULL	

```
3 rows in set (0.03 sec)
```

```
mysql> █
```

Part 4.2: Future Improvements.

The improvement on the database system will be more flexible and data will be easier to access and use. More processing will be done on the edge to facilitate real time computations and decision making.

It's very important for our database community to understand these changes and embrace them, because otherwise they are swept away in the future.

The database systems in the future are going to be very different than what we have today.

- **Making data easier to use.** Data allows us to learn faster. Gather data implicitly by watching what people do.\
- **Real-time computation,** less off-line computation. Rather than pre-computing a recommendation off line there will be a lot more real-time responsiveness.
- **Big Data Linked to Existing Data** - Companies generate big data when pursuing strategies such as the Internet of Things, or tracking web clickstream data for customer trends. But the deep insights come from blending that new big data with data companies already have.
- **Complete Data Protection** - When disk-based database backups came on the scene about a decade ago, they brought two big advances over tape: more accessibility to data and deduplication that reduced the volume of data and that's where it stopped in the last decade. The expectation will be that data backups happen constantly, in real time, so that data never gets lost.

Conclusion

I got the knowledge about the database systems, future improvements and some methods in mysql like Creating Prototypes and some exercises like example minimum, finding age, average, create trigger, join many tables etc. finally all above I have got some ideas of creating one database management and Adobe XD. I think I have got expected output in this assignment. thank you.

A. Stany Nirojan.

J/IT/18/12/01

References

dev.mysql, 2018. [https://dev.mysql..](https://dev.mysql.com/doc/relnotes/workbench/en/wb-news-5-2-10.html) [Online] Available at: <https://dev.mysql.com/doc/relnotes/workbench/en/wb-news-5-2-10.html> [Accessed 4 01 2018].

google, Available at: [Accessed 04 01 2018].

2018. [/www.researchgate.net.](https://www.researchgate.net/publication/318204712_Data_Verification_and_Validation_Process_in_the_Management_System_Development) [Online] [https://www.researchgate.net/publication/318204712 Data Verification and Validation Process in the Management System Development](https://www.researchgate.net/publication/318204712_Data_Verification_and_Validation_Process_in_the_Management_System_Development)

google, 2018. [blog.stibosystems.](https://blog.stibosystems.com/5-ways-to-improve-your-data-management) [Online] Available at: <https://blog.stibosystems.com/5-ways-to-improve-your-data-management> [Accessed 04 01 2018].

google, 2018. [Epa.gov.](https://www.epa.gov/sites/production/files/2016-10/documents/approach_to_data.pdf) [Online] Available at: https://www.epa.gov/sites/production/files/2016-10/documents/approach_to_data.pdf [Accessed 04 01 2018].

<http://www.dbta.com/>, 2018. BigDataQuarterly. [Online] Available at: <http://www.dbta.com/BigDataQuarterly/Articles/The-Database-Technologies-of-the-Future-109659.aspx> [Accessed 04 01 2018].

google, 2018. [sitepoint.com.](https://www.sitepoint.com/how-to-install-mysql/) [Online] Available at: <https://www.sitepoint.com/how-to-install-mysql/> [Accessed 04 01 2018].