IN PARTNERSHIP WITH PLYMOUTH UNIVERSITY

Name: **Perera, L. Pasindu.**	
Student Reference Number: **10568999**	

Module Code: ISAD253SL	Module Name: Databases
Coursework Title: Hospital Management System	
Deadline Date: 3rd January,2016	Member of staff responsible for coursework: Ms. Dileeka Alwis.
Programme: BSc (Honours) Software Engineering, Computer Networks, Computer Security.	

Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook.

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

Vidhanahena, I.P. Oshajith	**10569207**
Perera, L. Pasindu (Team Leader)	**10568999**
Chanuka, K. Imalsha	**10569083**
Rathnayaka, R.M.K.S.B	**10569071**
Wijesekara, J. Chanath Rajindra	**10569206**

We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.

Signed on behalf of the group:

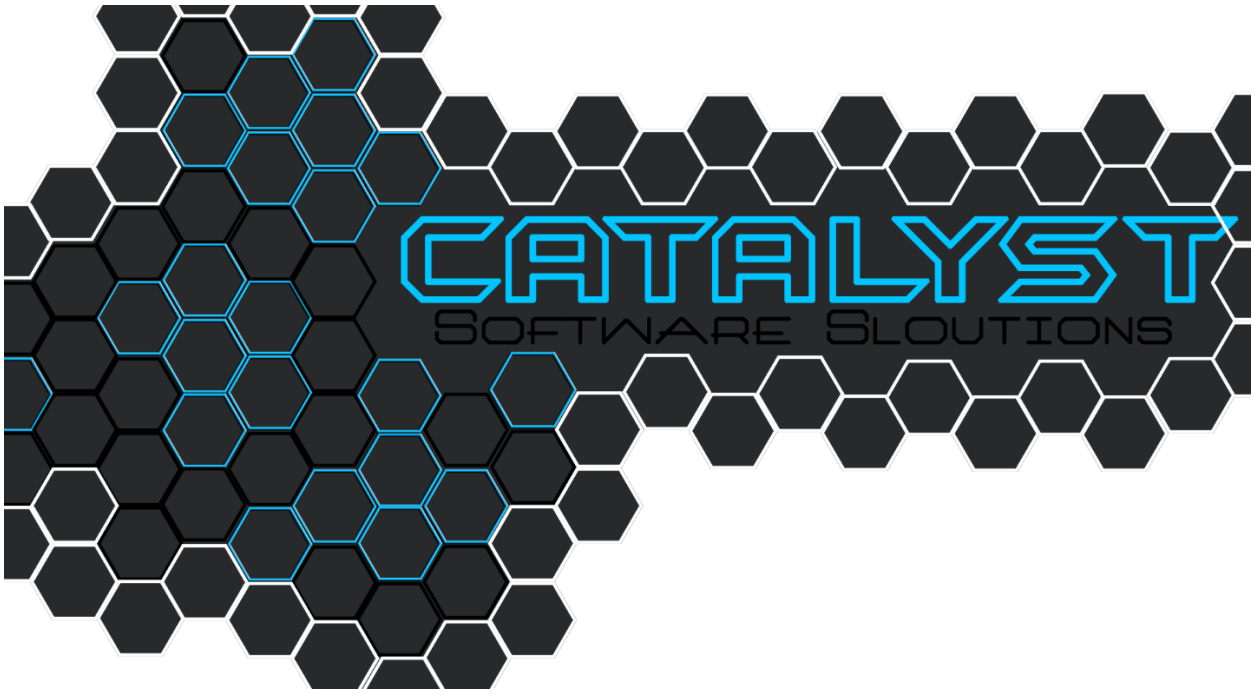
Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***
Signed:

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.
I *have used/not used translation software.

If used, please state name of software……

Overall mark _____% **Assessors Initials _____** **Date_____**

PROJECT REPORT



HOSPITAL MANAGEMENT SYSTEM

Project By: Catalyst Software Solutions

Team

Name	Index
Vidhanahena, I.P. Oshajith	10569207
Perera, L. Pasindu (Team Leader)	10568999
Chanuka, K. Imalsha	10569083
Rathnayaka, R.M.K.S.B.	10569071
Wijesekara, J. Chanath Rajindra	10569206

Contents

Abstract

This is the Project Report document for the coursework of the module ISAD253SL (Databases) for the 2nd year 1st semester of the programmes BSc (Honours) Software Engineering, Computer Networks and Computer Security of the Plymouth University, U.K. which are conducted in National School of Business Management, Sri Lanka. The focus of the coursework of the module ISAD253SL is to analyse a real-time scenario with problematic environment and develop the best solution for the problematic scenario by applying the database design concepts and using a database development tool like SQL.

Used Software Tools

Several software tools were used to complete this project.

1. Microsoft SQL Server 2014

   This software was used to create tables, add constraints, add triggers, stored procedures, user defined functions, to generate views which are necessary and to enter data into the database.

2. Microsoft Visio 2013

   This software was the key software that we used to create digital Enhanced Entity Relationship Diagram and Relational Mapping Diagram.

3. Microsoft Word 2016

   This software was used to create the Project Report for the course work.

4. Git

   A git online repository was used to manage the work and also to secure the coursework content of each team member in case of emergency. GitHub was the Git client we used and the GitBash was the Git shell we used.

Team and Workload

Index Number	Name	Work Load
10569207	Vidhanahena, I.P. Oshajith	Stored Procedures, EER, Create tables and constraints, Mapping, Triggers and Views
10569206	Wijesekara, J. Chanath Rajindra	EER
10569083	Chanuka, K. Imalsha	Create tables and constraints, Mapping
10569071	Rathnayaka, R.M.K.S.B.	Mapping, Create tables and constraints
10568999	Perera, L. Pasindu (Team Leader)	EER, Stored Procedures

Introduction

Project Report Introduction

This document is a to describe requirements for a Hospital Management System which keeps records and maintain all the information regarding every aspect of a usual Hospital. This report also contains the constraints, assumptions and system implantation details and such information about the suggested system.

Project Introduction

This project covers a Hospital Management System which is developed by applying database development concepts and using SQL technology. The system is developed to keep records of each element of the hospital such as departments, rooms, staff and patients. The system is also capable of maintain the information about the mentioned elements and can manipulate data accordingly. The system will make most of the management processes of the Hospital autonomous and will reduces the workload of the staff and minimize the use of resources.

Scope

Project cover the most of the aspects of the Hospital such and Human Resources Management, Departments, Room allocation, Patient Examination, Patient Admitting, Patient Discharging, Treatments, Surgeries and Drug Issuing.

References

No references used.

General Description

Scenario

Organization – City Central Hospital

Specialties - Multi-Specialty Hospital

Identified Elements – Departments, rooms, doctors, nurses, attendants, other staff, patients, Operations, Check-ups, Drugs, Treatments.

Scenario Description

Departments

City Central Hospital has few department like Orthopaedic, Pathology, Emergency, Dental, Gynaecology, Anaesthetics, I.C.U., Blood Bank, Operation Theatre, Laboratory, M.R.I., Neurology, Cardiology, Cancer Department, Corpse, etc.

OPD

Hospital has a OPD (Outpatient Department) where patients visits for check-ups with doctors.

Patients

Information about the patient is collected when patient arrives at the hospital.

Non-Admitting Patients

An Entry card for a patient is generated and sent to doctor.

Every patient has unique patient number.

Admitting Patients

Patient can choose a private of general room when admitting according to his/her preferring.

Before admitting patient has to fulfil some formalities such as room charges and etc.

When admitted patient number, payment method, advanced payment condition, diagnosis, admitted date, treatment and such information are recorded.

When discharging, patient has to go through some formalities like balance charges, test charges, operation charges, doctor charges and etc.

When patient is discharged, entry is recorded in the database with patient number, treatment given, treatment advice, payment made, mode of payment date of discharged and etc.

Regular Patients

Details about regular patients like visit, diagnosis, treatment, medicine recommended and status of treatment should also be recorded. There can be multiple entries of one patient for patient's each visit. Operation details such as patient number, date of admission, date of operation, number of the doctor who conducted the operation, number of the operation theatre in which operation was carried out, type of operation, patient's condition before and after operation and treatment advice, should also be stored in the database, if the patient face any operation in the hospital.

Doctors

Each doctor's information should also be included in the database. Doctors are assigned to Departments. The database should store doctors' data like name, qualification, address, phone number and etc. There are two types of the doctors in the City Central Hospital.

- Regular Doctors who work in the hospital and come to the hospital daily. Database should store following information about Regular Doctors. Pre-defined salary, date of joining and etc.
- Call on Doctors who are called by the hospital if the concerned doctor is not available or when additional doctors are required. Database must have information about Call on Doctors like fees per call, payment due and etc.

Rooms

Database should keep records like room number, room type (general or private), status, if occupied then patient number, patient name, charges per day and etc. Room number should

be unique and room type can only be ‘G’ for General Room or ‘P’ for Private Room and status can only be ‘Y’ or ‘N’.

Drugs, Treatments and Operations

Basic details about drugs, treatments and operations should also be included in the database separately. And also, the database should be aware of the details about customer payments whenever they prescribe medicine, undergo operations, admit in the hospital and discharge from the hospital etc.

Other Hospital Employees

Information about the other hospital staff like nurses, attendants, laboratory staff should be included in the database.

Requirements

Functional Requirements

INSERT data.

UPDATE records.

DELETE records.

Validate inserted records.

Generate reports.

Non – Functional Requirements

Identifying entities.

Creating functions.

Creating stored procedures.

Triggers.

Users

The main user of this system will be System Administrator, who is literate with computers and can use a SQL database correctly.

Hospital management can also have privileges to DELETE, UPDATE or ADD records to employee tables, department, room and drugs tables.

The doctors and nurses have certain privileges and attendant can also use the system with under several conditions.

Assumptions

All the employees are assigned to departments. One employee only belongs in a certain department.

All the employees in the hospital are doctors or nurses or attendants or other staff. There are no other employees in the organization.

All the employees have a employee_id with different prefixes for each type of employees.

One Check-up can have only one treatment.

Patient can admit in the hospital only after a Check-up.

Treatments and drugs are issued only after a Check-up.

One doctor engages in many operations.

Many doctors can engage in one operation.

Admitted patient can also go to another Check-up.

System Design

Identifying Entities

When analysing case study, we can identify several entities.

- Department
- Rooms
- Employee
- Patient
- Check-ups
- Treatments
- Drugs
- Patient admission
- Operation
- Discharge record
- Payment

Further we can divide Rooms, Employee and Patient entities into sub entities.

Employee

- Nurse
- Attendant
- Doctor
- Other Staff

Patient

- Regular patient
- Admitted patient

Rooms

- General rooms
- Private rooms

Further we can divide Doctor sub entity in to 2 sub entities.

Doctor

- Regular doctor
- Call on doctor

Identifying Attributes for pre-identified identities

1. **department**
   (department_id, department_name, location, facilities)

2. **rooms**
   (room_no, type, status)
   a. **general_rooms**
   b. **private_rooms**

3. **employee**
   (employee_id, name, gender, address, NIC_no, phone_no, employee_type, department_id)

   a. **nurse**
      (nurse_id, qualification)

   b. **attendant**
      (employee_id)

   c. **other_staff**
      (employee_id)

   d. **doctor**
      (employee_id, specialized_field)

      i. **regular_doctor**
         (employee_id, basic_salary, date_of_joining)

      ii. **callon_doctor**
         (employee_id, call_on_fee, payment_due)

4. **patient**
   (patient_id, first_name, last_name, entry_date, dob, gender, address, phone_number)
   a. **regular_patient**
      (patient_id)

   b. **admitted_patient**
      (patient_id)

5. **checkups**
   (checkup_id, operation_id, date, check_up_fee, diagnosis, patient_condition, patient_id, employee_id)

6. **treatments**
   (treatment_id,type,description, checkup_id)

7. **drugs**
   (drug_id,price,Alternative_drug,name,dose_description, checkup_id)

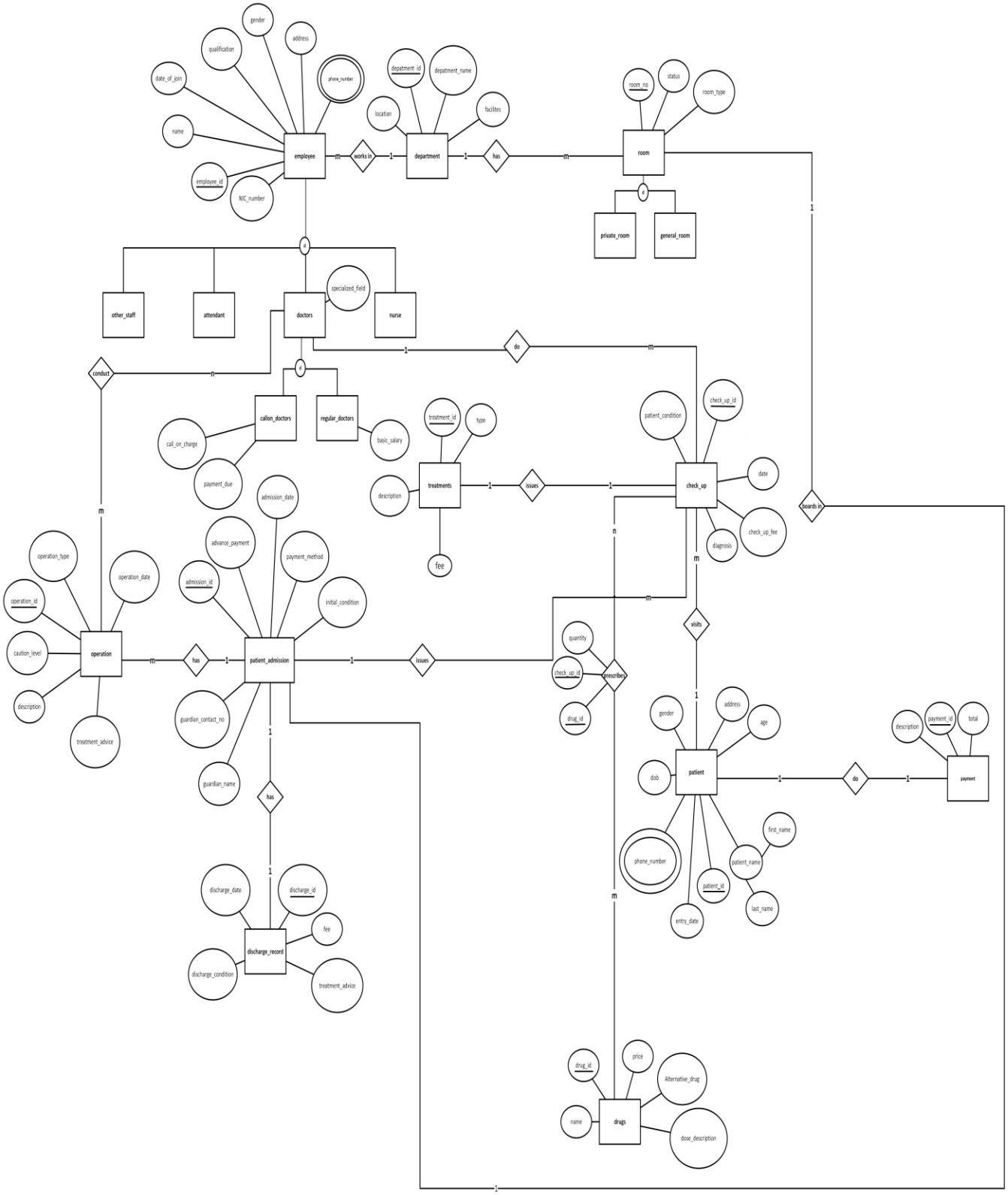
8. **patient_admisson**
   (admission_id, advance_payment, payment_method, initial_condition, guardian_name, guardian_contact_no, checkup_id, admission_date)

9. **operation**
   (operation_id, operation_type, operation_date, treatment_advice, description, caution_level, admission_id, doctor_id)

10. **discharge_record**
    (discharge_id, fee, treatment_advice, discharge_condition, discharge_date, admission_id)

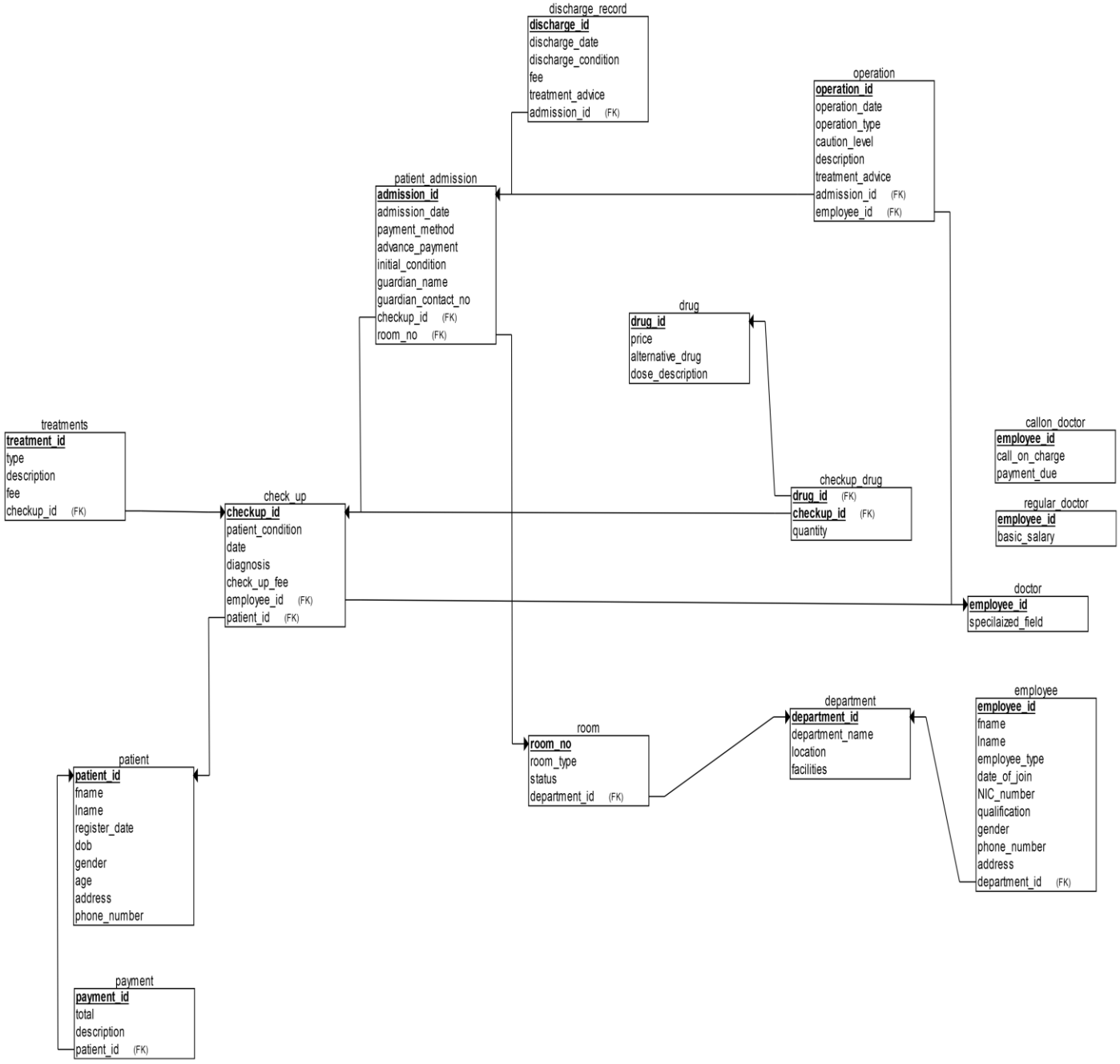
11. **payment**
    (bill_id,total,description,patient_id)

ER and EER Diagrams



An enlarged and clear version of the above EER diagram of the database is available in the folder.

Relational Mapping



An enlarged and clear version of the above Relational Schema of the database is available in the folder.

Normalization

1st Normalization Form

If a relational schema is in First Normal Form table should not have composite attribute, multi-valued attributes and nested relations.

Above relational schema has not completed the requirements to be in "1st Normalized Form".

In tables employee and patient table, **phone_number** is a multi-valued attribute. After normalizing employee and patient tables these are the results.

employee

[employee_id, fname, lname, employee_type, date_of_join, NIC_number, qualification, gender, phone_number, address, department_id]

employee_phone

[employee_id, phone_number]

patient

[patient_id, fname, lname, register_date, dob, gender, age , address, phone_nuumber]

patient_phone

[patient_id, phone_number]

2nd Normalization Form

If a relational schema is in Second Normalization Form,
The relational schema should be in 1NF and every non-primary key attribute should be fully functionally dependent on the primary key.

In the callon_doctor table, attribute "payment_due" does not fully functionally depend on the primary key "employee_id". So, the attribute "payment_due" does not belong in the "callon_doctor" table.

After the above normalization process, this relational schema follows the 2nd normalization Form. So, this relational schema is in 2nd Normalization.

3rd Normalization Form

If a relational schema is in Third Normalization Form,
The relational schema should be in 2NF and every non-key attribute should be non-transitively dependent on the Primary Key.

In the "patient" table attribute "age" depends on the non-key attribute "dob"(date of birth) and "dob" is fully functionally depend on the primary key "patient_id". So, attribute "age" transitively depends on primary-key "patient_id". Since attribute "age" does not follow 3rd Normalization form and attribute "age" is not a required attribute for the table we can omit the attribute "age" from the "patient" table.

After the above normalization process, this relational schema does not possess any transitive dependencies. This Relational Schema is now in 3rd Normalization Form.

SQL Development

Creating tables

```
CREATE DATABASE city_central
use city_central

/*Creating Tables*/

--department table
CREATE TABLE department(
  department_id VARCHAR(6) NOT NULL PRIMARY KEY,
  department_name VARCHAR(100) NOT NULL,
  location VARCHAR(50) NOT NULL,
  facilities VARCHAR(400) NOT NULL,
)

--employee table
CREATE TABLE employee(
       employee_id VARCHAR(5) NOT NULL PRIMARY KEY,
       fname VARCHAR(25) NOT NULL,
       lname VARCHAR(25) NOT NULL,
       date_of_join DATE NOT NULL,
       NIC_number VARCHAR(10) NOT NULL,
       qualification VARCHAR(100) NOT NULL,
       gender CHAR(1) NOT NULL,
       address VARCHAR(400) NOT NULL,
       employee_type VARCHAR(15) NOT NULL,
       department_id VARCHAR(6) NOT NULL
)

--employee_phone table
CREATE TABLE employee_phone(
       employee_id VARCHAR(5) NOT NULL PRIMARY KEY,
       phone_number VARCHAR(10) NOT NULL
)

/*ALTER TABLE patient
ADD CONSTRAINT pk_patient_id PRIMARY KEY(patient_id)*/

--doctor table
CREATE TABLE doctor(
       employee_id VARCHAR(5) NOT NULL PRIMARY KEY,
       specilaized_field VARCHAR(50) NOT NULL
)

--callon_doctor table
CREATE TABLE callon_doctor(
       employee_id VARCHAR(5) NOT NULL PRIMARY KEY,
       call_on_charge money NOT NULL,
)

--regular_doctor table
CREATE TABLE regular_doctor(
       employee_id VARCHAR(5) NOT NULL PRIMARY KEY,
       basic_salary money NOT NULL
)
```

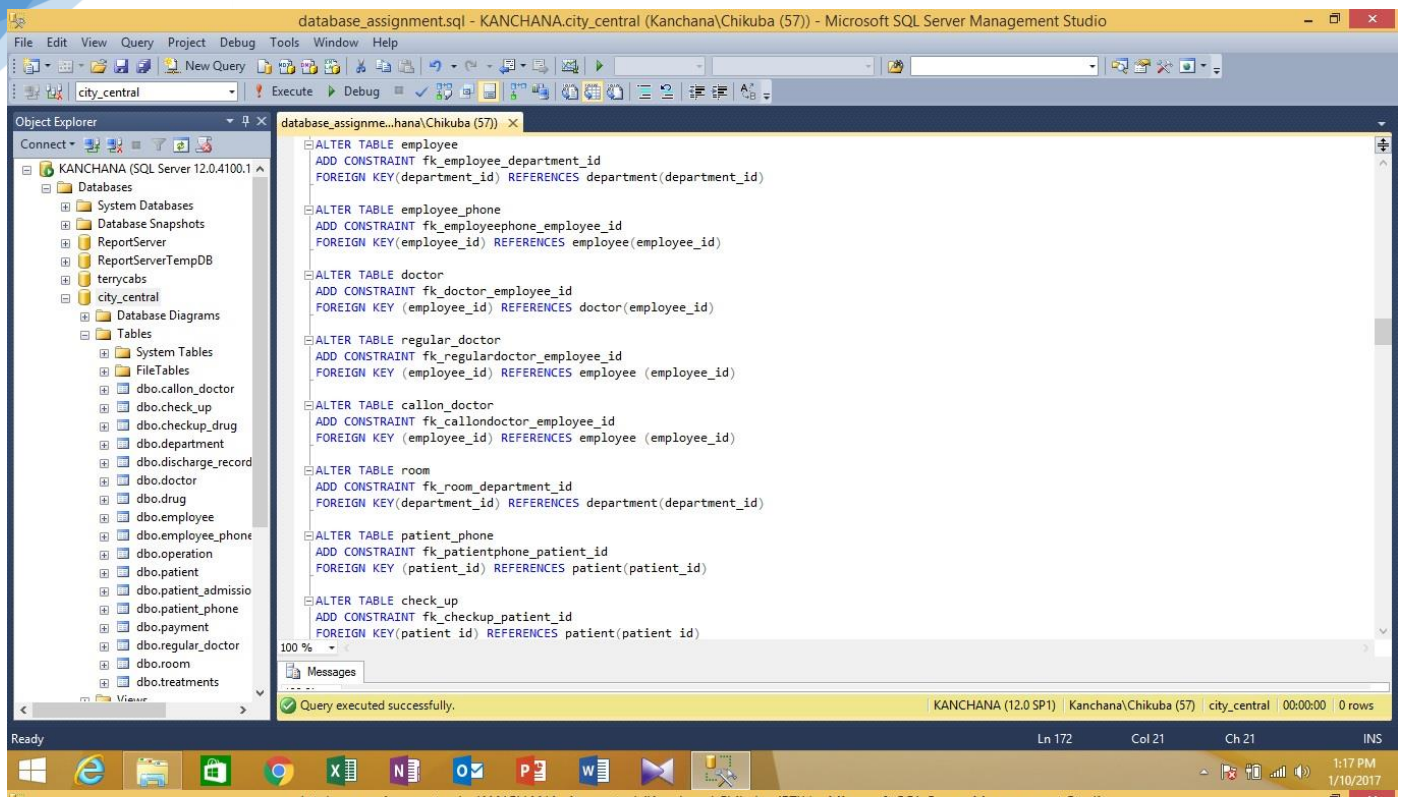
```
--room table
CREATE TABLE room(
  room_no VARCHAR(4) NOT NULL PRIMARY KEY,
  status CHAR(1) NOT NULL,
  room_type VARCHAR(10) NOT NULL,
  room_fee money NOT NULL,
  department_id VARCHAR(6) NOT NULL,
)

--patient table
CREATE TABLE patient(
  patient_id VARCHAR(6) NOT NULL PRIMARY KEY,
  fname VARCHAR(25) NOT NULL,
  lname VARCHAR(25) NOT NULL,
  register_date DATE NOT NULL,
  dob DATE NOT NULL,
  address VARCHAR(200) NOT NULL,
  gender CHAR(1) NOT NULL
)

--patient_phone table
CREATE TABLE patient_phone(
       patient_id VARCHAR(6) NOT NULL PRIMARY KEY,
       phone_number VARCHAR(10)
)

--check_up table
CREATE TABLE check_up(
       checkup_id INT NOT NULL PRIMARY KEY,
       patient_condition VARCHAR(100) NOT NULL,
       date DATE NOT NULL,
       diagnosis CHAR(100) NOT NULL,
       check_up_fee money NOT NULL,
       patient_id VARCHAR(6) NOT NULL,
       employee_id VARCHAR(5) NOT NULL
)

--patient_admission
CREATE TABLE patient_admission(
       admission_id INT NOT NULL PRIMARY KEY,
       admission_date DATE NOT NULL,
       payment_method VARCHAR(10) NOT NULL,
       advance_payment money NOT NULL,
       initial_condition VARCHAR(200) NOT NULL,
       guardian_name VARCHAR(100) NOT NULL,
       guardian_contact_no VARCHAR(10) NOT NULL,
       checkup_id INT NOT NULL,
       room_no VARCHAR(4) NOT NULL
)

--operation table
CREATE TABLE operation(
       operation_id INT NOT NULL PRIMARY KEY,
       operation_date DATE NOT NULL,
       operation_type VARCHAR(50) NOT NULL,
       caution_level VARCHAR(25) NOT NULL,
       description VARCHAR(500) NOT NULL,
       treatment_advice VARCHAR(400) NOT NULL,
       operation_fee money NOT NULL,
       admission_id INT NOT NULL,
       employee_id VARCHAR(5) NOT NULL
)

--discharge_record
```

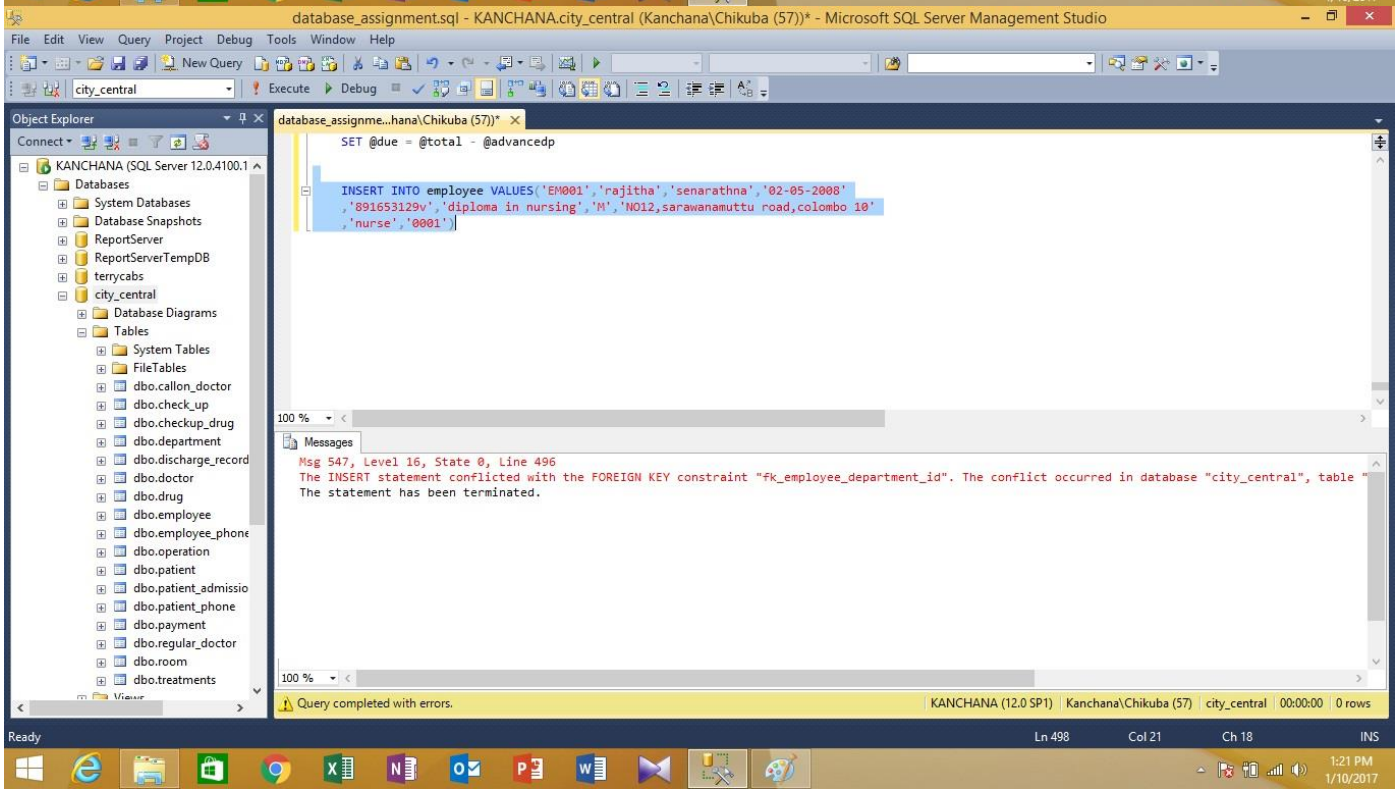
```
CREATE TABLE discharge_record(
	discharge_id INT NOT NULL PRIMARY KEY,
	discharge_date DATE NOT NULL,
	discharge_condition VARCHAR(200) NOT NULL,
	fee money NOT NULL,
	treatment_advice VARCHAR(400) NOT NULL,
	admission_id INT NOT NULL
)

--treatment table
CREATE TABLE treatments(
	treatment_id INT NOT NULL PRIMARY KEY,
	type VARCHAR(50) NOT NULL,
	description VARCHAR(200) NOT NULL,
	fee money NOT NULL,
	checkup_id INT NOT NULL
)

--drug table
CREATE TABLE drug(
	drug_id VARCHAR(6) NOT NULL PRIMARY KEY,
	drug_name VARCHAR(50) NOT NULL,
	price money NOT NULL,
	alternative_drug VARCHAR(50) NOT NULL,
	dose_description VARCHAR(200) NOT NULL
)

--payment table
CREATE TABLE payment(
	payment_id INT NOT NULL PRIMARY KEY,
	total INT,
	description VARCHAR(200) NOT NULL,
	patient_id VARCHAR(6) NOT NULL
)

--checkup_drug table
CREATE TABLE checkup_drug(
	quantity INT NOT NULL,
	drug_id VARCHAR(6) NOT NULL,
	checkup_id INT NOT NULL,
	PRIMARY KEY (drug_id, checkup_id)
```

Constraints

```
--FOREIGN KEY CONSTRAINTS

ALTER TABLE employee
ADD CONSTRAINT fk_employee_department_id
FOREIGN KEY(department_id) REFERENCES department(department_id)

ALTER TABLE employee_phone
ADD CONSTRAINT fk_employeephone_employee_id
FOREIGN KEY(employee_id) REFERENCES employee(employee_id)

ALTER TABLE doctor
ADD CONSTRAINT fk_doctor_employee_id
FOREIGN KEY (employee_id) REFERENCES doctor(employee_id)

ALTER TABLE regular_doctor
ADD CONSTRAINT fk_regulardoctor_employee_id
FOREIGN KEY (employee_id) REFERENCES employee (employee_id)
```

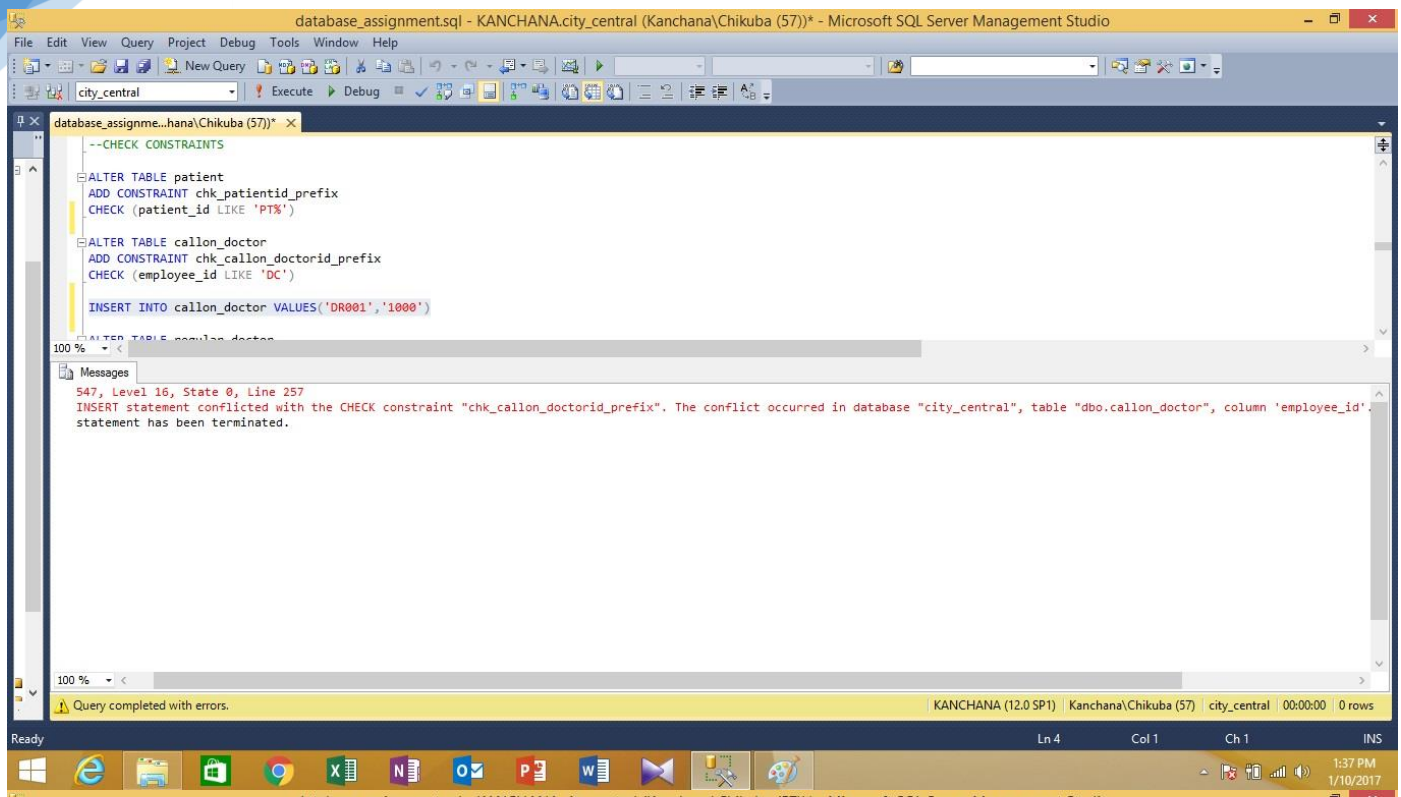
```
ALTER TABLE callon_doctor
ADD CONSTRAINT fk_callondoctor_employee_id
FOREIGN KEY (employee_id) REFERENCES employee (employee_id)

ALTER TABLE room
ADD CONSTRAINT fk_room_department_id
FOREIGN KEY(department_id) REFERENCES department(department_id)

ALTER TABLE patient_phone
ADD CONSTRAINT fk_patientphone_patient_id
FOREIGN KEY (patient_id) REFERENCES patient(patient_id)

ALTER TABLE check_up
ADD CONSTRAINT fk_checkup_patient_id
FOREIGN KEY(patient_id) REFERENCES patient(patient_id)

ALTER TABLE check_up
ADD CONSTRAINT fk_checkup_employee_id
FOREIGN KEY(employee_id) REFERENCES employee(employee_id)

ALTER TABLE patient_admission
ADD CONSTRAINT fk_patientadmission_checkup_id
FOREIGN KEY(checkup_id) REFERENCES check_up(checkup_id)

ALTER TABLE patient_admission
ADD CONSTRAINT fk_patient_admission_room_no
FOREIGN KEY (room_no) REFERENCES room(room_no)

ALTER TABLE operation
ADD CONSTRAINT fk_operation_admission_id
FOREIGN KEY(admission_id) REFERENCES patient_admission(admission_id)


ALTER TABLE operation
ADD CONSTRAINT fk_operation_employee_id
FOREIGN KEY(employee_id) REFERENCES employee(employee_id)

ALTER TABLE discharge_record
ADD CONSTRAINT fk_dischargerecord_admission_id
FOREIGN KEY(admission_id) REFERENCES patient_admission(admission_id)

ALTER TABLE treatments
ADD CONSTRAINT fk_treatments_checkup_id
FOREIGN KEY(checkup_id) REFERENCES check_up(checkup_id)

ALTER TABLE payment
ADD CONSTRAINT fk_payment_patient_id
FOREIGN KEY(patient_id) REFERENCES patient(patient_id)

ALTER TABLE checkup_drug
ADD CONSTRAINT fk_checkupdrug_checkup_id
FOREIGN KEY(checkup_id) REFERENCES check_up(checkup_id)

ALTER TABLE checkup_drug
ADD CONSTRAINT fk_checkupdrug_drug_id
FOREIGN KEY(drug_id) REFERENCES drug(drug_id)
```

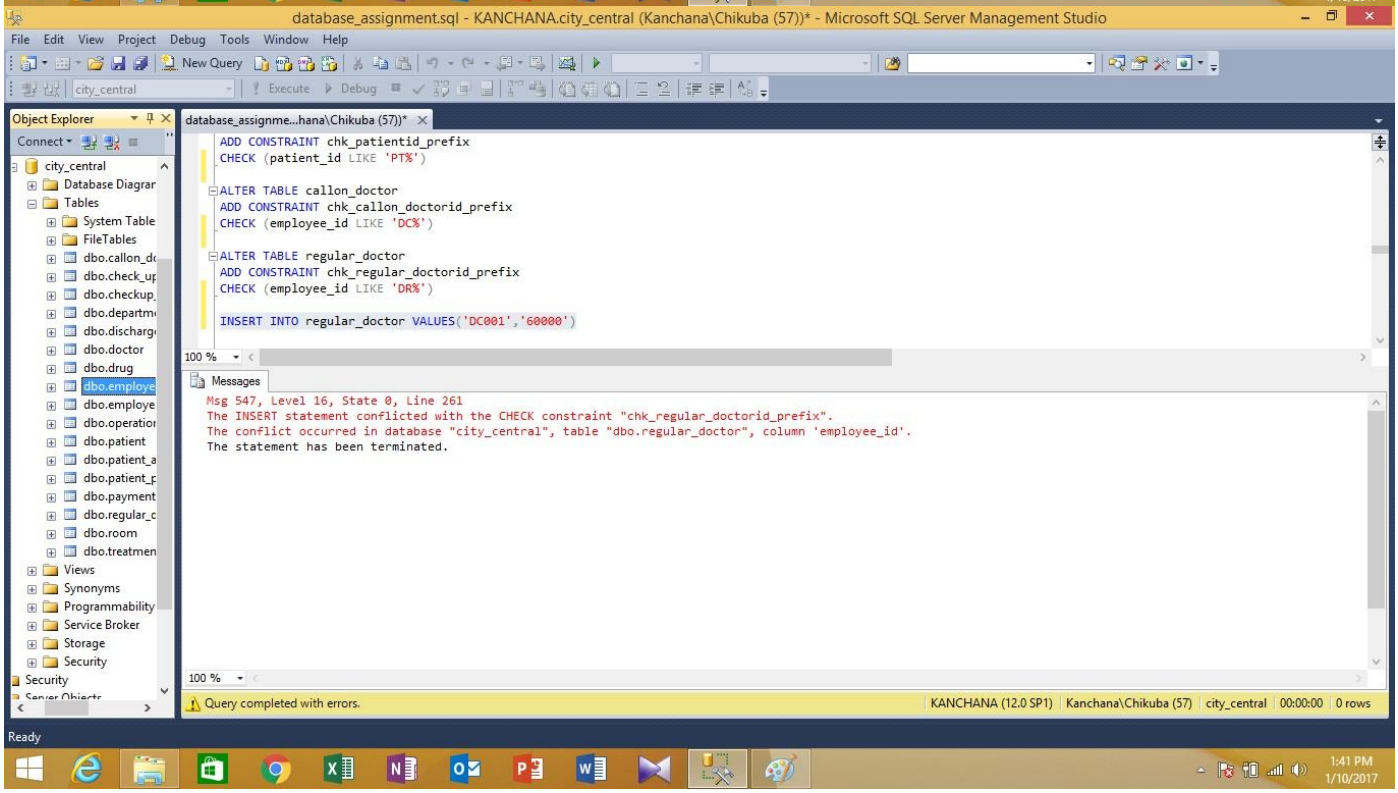


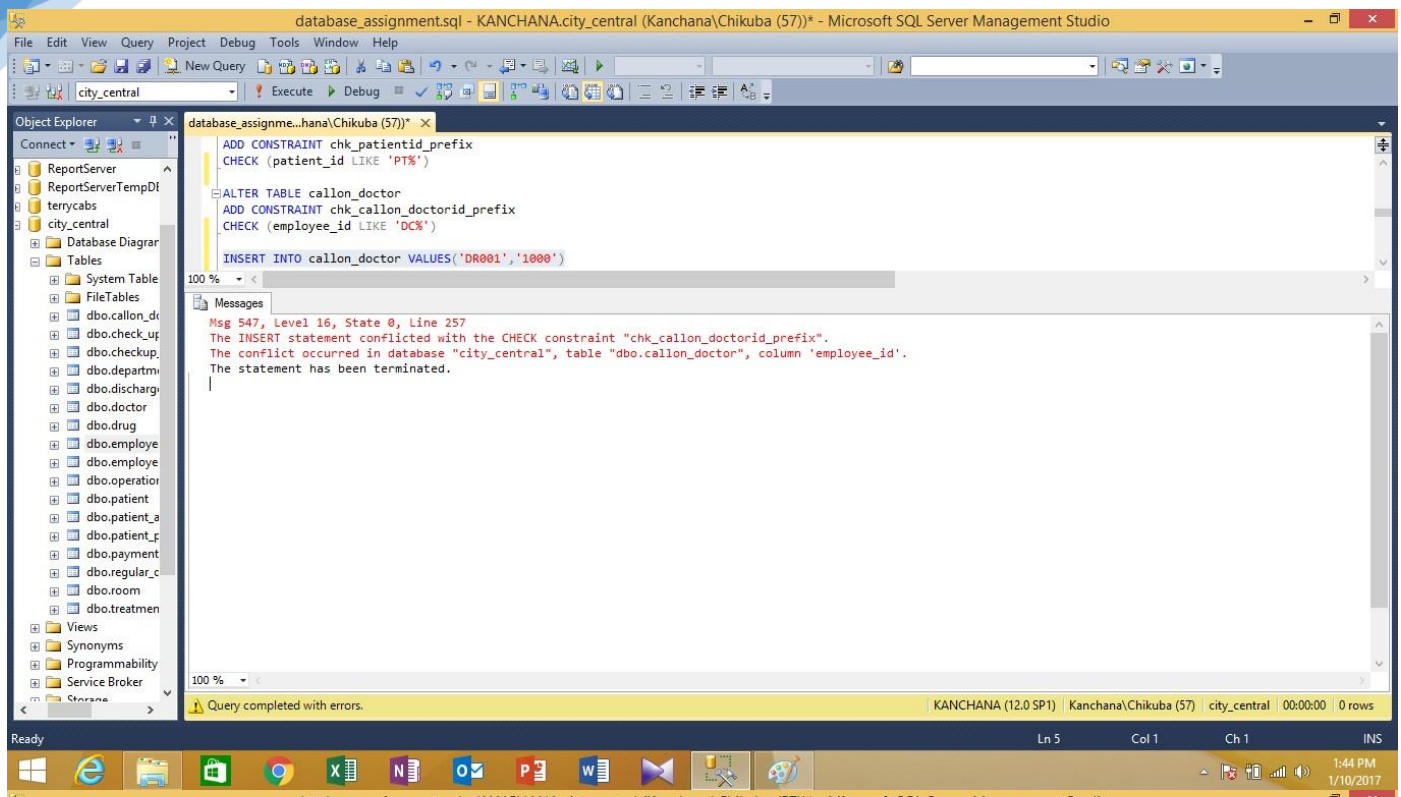


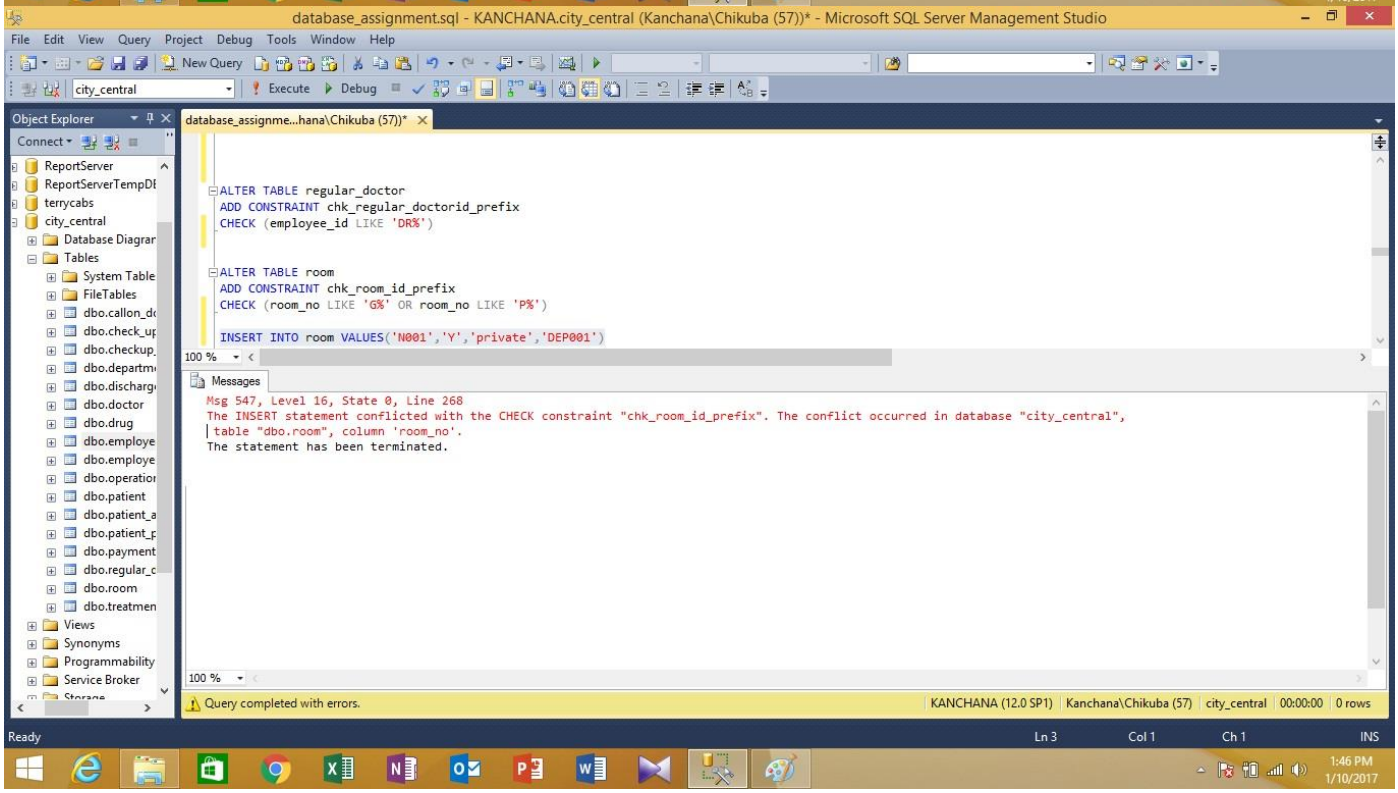

```
--CHECK CONSTRAINTS

ALTER TABLE patient
ADD CONSTRAINT chk_patientid_prefix
CHECK (patient_id LIKE 'PT%')

ALTER TABLE callon_doctor
ADD CONSTRAINT chk_callon_doctorid_prefix
CHECK (employee_id LIKE 'DC%')

ALTER TABLE regular_doctor
```

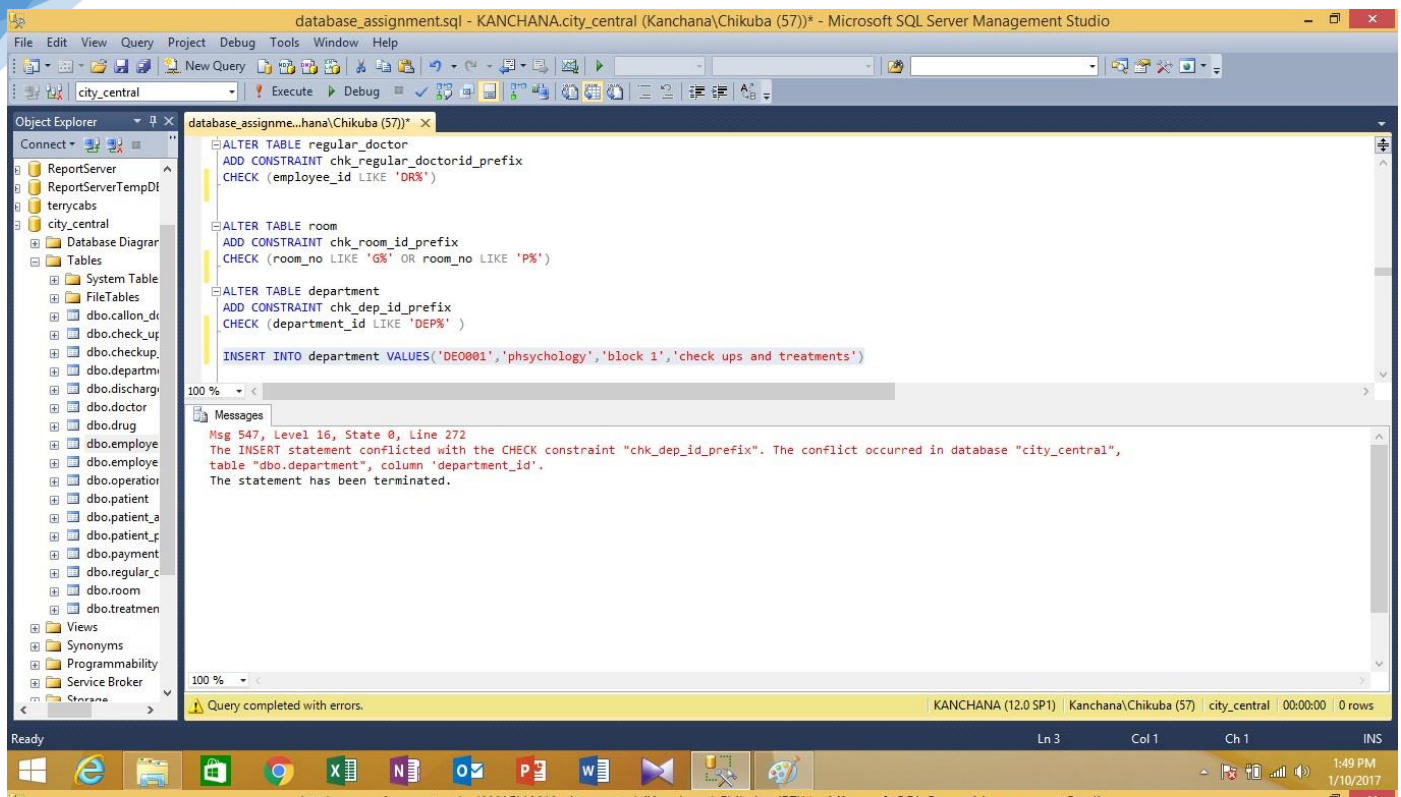
```
ADD CONSTRAINT chk_regular_doctorid_prefix
CHECK (employee_id LIKE 'DR%')

ALTER TABLE room
ADD CONSTRAINT chk_room_id_prefix
CHECK (room_no LIKE 'G%' OR room_no LIKE 'P%')

ALTER TABLE department
ADD CONSTRAINT chk_dep_id_prefix
CHECK (department_id LIKE 'DEP%' )

ALTER TABLE employee
ADD CONSTRAINT chk_employee_id_prefix
CHECK (employee_id LIKE 'EM%' OR employee_id LIKE 'DC%' OR employee_id LIKE 'DR%')
```

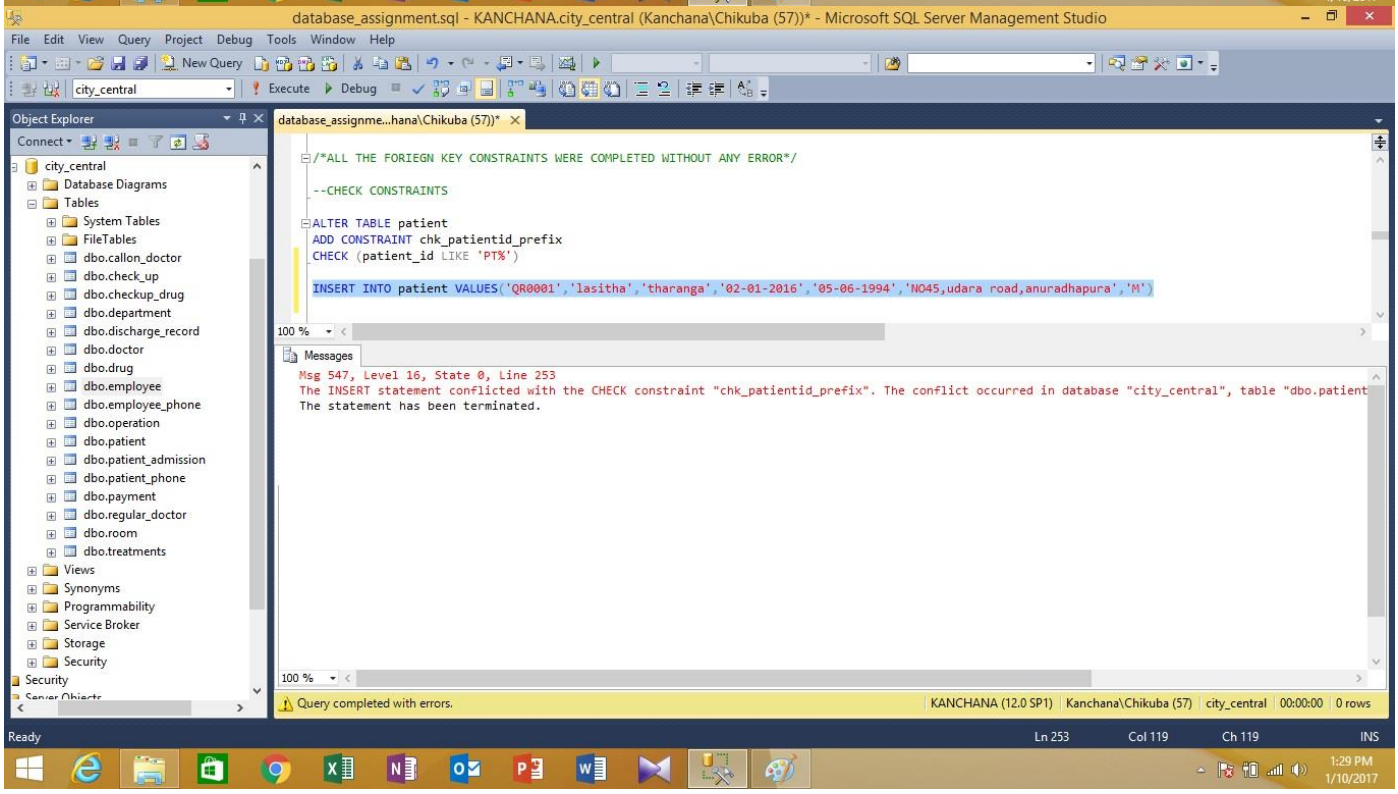

database_assignment.sql - KANCHANA.city_central (Kanchana\Chikuba (57))* - Microsoft SQL Server Management Studio
File Edit View Query Project Debug Tools Window Help
New Query
city_central
Execute Debug
database_assignme...hana\Chikuba (57))*
--CHECK CONSTRAINTS
ALTER TABLE patient
ADD CONSTRAINT chk_patientid_prefix
CHECK (patient_id LIKE 'PT%')
ALTER TABLE callon_doctor
ADD CONSTRAINT chk_callon_doctorid_prefix
CHECK (employee_id LIKE 'DC')
INSERT INTO callon_doctor VALUES('DR001','1000')
100 %
Messages
547, Level 16, State 0, Line 257
INSERT statement conflicted with the CHECK constraint "chk_callon_doctorid_prefix". The conflict occurred in database "city_central", table "dbo.callon_doctor", column 'employee_id'.
statement has been terminated.
100 %
Query completed with errors.
KANCHANA (12.0 SP1) Kanchana\Chikuba (57) city_central 00:00:00 0 rows
Ready
Ln 4 Col 1 Ch 1 INS
1:37 PM
1/10/2017


database_assignment.sql - KANCHANA.city_central (Kanchana\Chikuba (57))* - Microsoft SQL Server Management Studio
File Edit View Project Debug Tools Window Help
New Query
city_central
Execute Debug
Object Explorer
Connect
city_central
Database Diagrar
Tables
System Table
FileTables
dbo.callon_d
dbo.check_up
dbo.checkup_
dbo.departm
dbo.dischargi
dbo.doctor
dbo.drug
dbo.employe
dbo.employe
dbo.operatio
dbo.patient
dbo.patient_a
dbo.patient_p
dbo.payment
dbo.regular_c
dbo.room
dbo.treatmen
Views
Synonyms
Programmability
Service Broker
Storage
Security
Security
database_assignme...hana\Chikuba (57))*
ADD CONSTRAINT chk_patientid_prefix
CHECK (patient_id LIKE 'PT%')
ALTER TABLE callon_doctor
ADD CONSTRAINT chk_callon_doctorid_prefix
CHECK (employee_id LIKE 'DC%')
ALTER TABLE regular_doctor
ADD CONSTRAINT chk_regular_doctorid_prefix
CHECK (employee_id LIKE 'DR%')
INSERT INTO regular_doctor VALUES('DC001','60000')
100 %
Messages
Msg 547, Level 16, State 0, Line 261
The INSERT statement conflicted with the CHECK constraint "chk_regular_doctorid_prefix".
The conflict occurred in database "city_central", table "dbo.regular_doctor", column 'employee_id'.
The statement has been terminated.
100 %
Query completed with errors.
KANCHANA (12.0 SP1) Kanchana\Chikuba (57) city_central 00:00:00 0 rows
Ready
1:41 PM
1/10/2017


database_assignment.sql - KANCHANA.city_central (Kanchana\Chikuba (57))* - Microsoft SQL Server Management Studio
File Edit View Query Project Debug Tools Window Help
New Query
city_central
Execute Debug
Object Explorer
Connect
ReportServer
ReportServerTempDB
terrycabs
city_central
Database Diagrams
Tables
System Tables
FileTables
dbo.callon_doctor
dbo.check_up
dbo.checkup
dbo.department
dbo.discharge
dbo.doctor
dbo.drug
dbo.employee
dbo.employee
dbo.operation
dbo.patient
dbo.patient_a
dbo.patient_p
dbo.payment
dbo.regular_d
dbo.room
dbo.treatmen
Views
Synonyms
Programmability
Service Broker
database_assignme...hana\Chikuba (57))*
ADD CONSTRAINT chk_patientid_prefix
CHECK (patient_id LIKE 'PT%')
ALTER TABLE callon_doctor
ADD CONSTRAINT chk_callon_doctorid_prefix
CHECK (employee_id LIKE 'DC%')
INSERT INTO callon_doctor VALUES('DR001','1000')
100 %
Messages
Msg 547, Level 16, State 0, Line 257
The INSERT statement conflicted with the CHECK constraint "chk_callon_doctorid_prefix".
The conflict occurred in database "city_central", table "dbo.callon_doctor", column 'employee_id'.
The statement has been terminated.
100 %
Query completed with errors.
KANCHANA (12.0 SP1) Kanchana\Chikuba (57) city_central 00:00:00 0 rows
Ready
Ln 5 Col 1 Ch 1 INS
1:44 PM
1/10/2017


database_assignment.sql - KANCHANA.city_central (Kanchana\Chikuba (57))* - Microsoft SQL Server Management Studio
File Edit View Query Project Debug Tools Window Help
New Query
city_central
Execute Debug
Object Explorer
Connect
ReportServer
ReportServerTempDB
terrycabs
city_central
Database Diagrams
Tables
System Tables
FileTables
dbo.callon_doctor
dbo.check_up
dbo.checkup
dbo.department
dbo.discharge
dbo.doctor
dbo.drug
dbo.employee
dbo.employee
dbo.operation
dbo.patient
dbo.patient_a
dbo.patient_p
dbo.payment
dbo.regular_d
dbo.room
dbo.treatmen
Views
Synonyms
Programmability
Service Broker
database_assignme...hana\Chikuba (57))*
ALTER TABLE regular_doctor
ADD CONSTRAINT chk_regular_doctorid_prefix
CHECK (employee_id LIKE 'DR%')
ALTER TABLE room
ADD CONSTRAINT chk_room_id_prefix
CHECK (room_no LIKE 'G%' OR room_no LIKE 'P%')
INSERT INTO room VALUES('N001','Y','private','DEP001')
100 %
Messages
Msg 547, Level 16, State 0, Line 268
The INSERT statement conflicted with the CHECK constraint "chk_room_id_prefix". The conflict occurred in database "city_central",
table "dbo.room", column 'room_no'.
The statement has been terminated.
100 %
Query completed with errors.
KANCHANA (12.0 SP1) Kanchana\Chikuba (57) city_central 00:00:00 0 rows
Ready
Ln 3 Col 1 Ch 1 INS
1:46 PM
1/10/2017

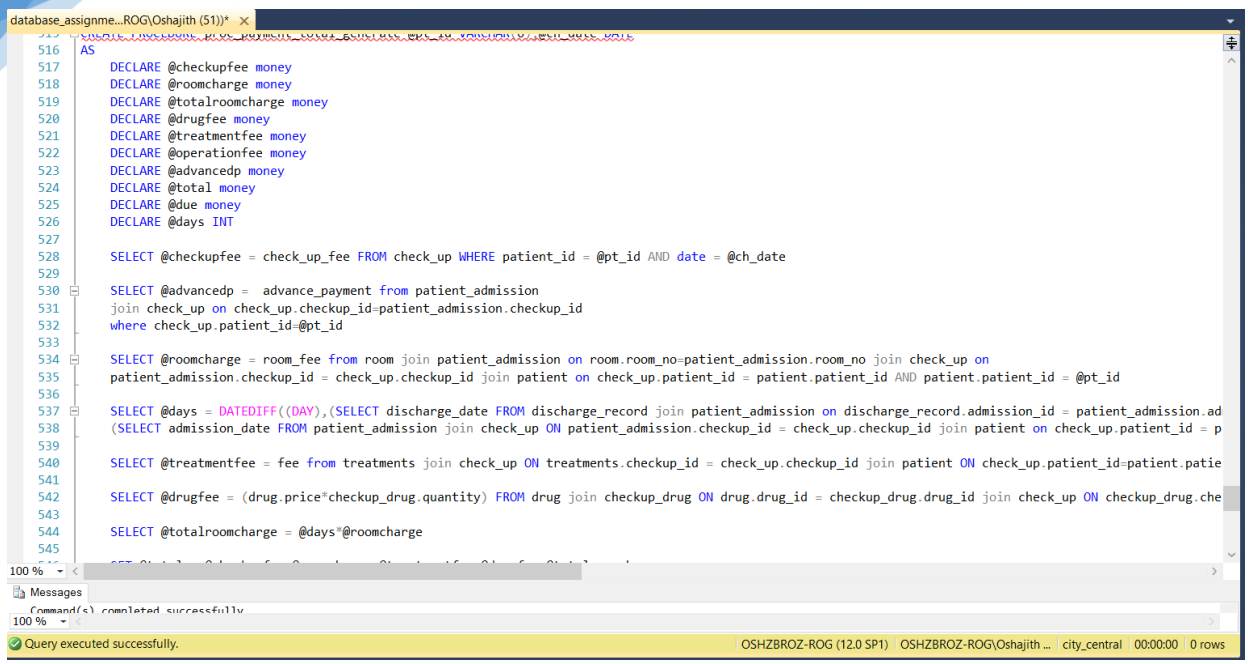




Stored Procedures

```
CREATE PROCEDURE proc_payment_total_generate @pt_id VARCHAR(6),@ch_date DATE
AS
        DECLARE @checkupfee money
        DECLARE @roomcharge money
        DECLARE @totalroomcharge money
        DECLARE @drugfee money
        DECLARE @treatmentfee money
```

```
DECLARE @operationfee money
DECLARE @advancedp money
DECLARE @total money
DECLARE @due money
DECLARE @days INT
SELECT @checkupfee = check_up_fee FROM check_up WHERE patient_id = @pt_id AND date = @ch_date
SELECT @advancedp =  advance_payment from patient_admission
join check_up on check_up.checkup_id=patient_admission.checkup_id
where check_up.patient_id=@pt_id
SELECT @roomcharge = room_fee from room join patient_admission on
room.room_no=patient_admission.room_no join check_up on
patient_admission.checkup_id = check_up.checkup_id join patient on check_up.patient_id =
patient.patient_id AND patient.patient_id = @pt_id
SELECT @days = DATEDIFF((DAY),(SELECT discharge_date FROM discharge_record join
patient_admission on discharge_record.admission_id = patient_admission.admission_id join
check_up ON patient_admission.checkup_id = check_up.checkup_id join patient on
check_up.patient_id = patient.patient_id AND patient.patient_id = @pt_id),
(SELECT admission_date FROM patient_admission join check_up ON patient_admission.checkup_id =
check_up.checkup_id join patient on check_up.patient_id = patient.patient_id AND
patient.patient_id = @pt_id))
SELECT @treatmentfee = fee from treatments join check_up ON treatments.checkup_id =
check_up.checkup_id join patient ON check_up.patient_id=patient.patient_id AND
patient.patient_id = @pt_id
SELECT @drugfee = (drug.price*checkup_drug.quantity) FROM drug join checkup_drug ON
drug.drug_id = checkup_drug.drug_id join check_up ON checkup_drug.checkup_id =
check_up.checkup_id join patient on check_up.patient_id = patient.patient_id AND
patient.patient_id = @pt_id
SELECT @totalroomcharge = @days*@roomcharge
SET @total = @checkupfee+@roomcharge+@treatmentfee+@drugfee+@totalroomcharge
SET @due = @total-@advancedp
UPDATE payment
SET total= @total WHERE patient_id = @pt_id
PRINT('Total Bill Generated.')


EXEC proc_payment_total_generate 'PT0003','2016-03-08'
```



Triggers

```
/*TRIGGER*/
--room status update trigger
CREATE TRIGGER tg_discharge_record_roomstat
ON discharge_record
AFTER INSERT
	AS
		DECLARE @status CHAR(1)
		SET @status='Y'

		DECLARE @discharge_id INT
		SELECT @discharge_id = i.discharge_id FROM inserted i

		DECLARE @add_id INT
		SELECT @add_id = pa.admission_id FROM patient_admission pa JOIN discharge_record dr ON
pa.admission_id=dr.admission_id

		DECLARE @room_no VARCHAR(4)
		SELECT @room_no = room_no FROM patient_admission WHERE admission_id =@add_id
```

```
		UPDATE room
		SET status = @status WHERE room.room_no = @room_no
		PRINT('Room'+@room_no+' is now free.')


--discharge_check_in_operations
CREATE TRIGGER tg_check_discharge_for_operations
ON operation
AFTER INSERT

	AS
	DECLARE @opadrec_id INT
	SELECT @opadrec_id = i.admission_id FROM inserted i
	IF EXISTS (SELECT d.admission_id FROM discharge_record d WHERE @opadrec_id = d.admission_id)
		BEGIN
			PRINT('The Patient Already Discharged')
			ROLLBACK
		END

--emp_id_prfix_check trigger
CREATE TRIGGER emp_id_prefix_check
ON employee
FOR INSERT
AS
BEGIN

	DECLARE @emptype varchar(10)
	DECLARE @empid varchar(5)
	SELECT @emptype=employee_type,@empid=employee_id FROM employee

	IF @emptype='doctor'
	BEGIN
		IF @empid NOT LIKE 'DR%'
		BEGIN
			PRINT 'wrong index format'
		END

		ELSE IF @empid NOT LIKE 'DC%'
		BEGIN
			PRINT 'wrong index format'
		END
	END
	ELSE
	BEGIN
		IF @empid NOT LIKE 'EM%'
		BEGIN
			PRINT 'wrong index format'
		END
	END
END
```

database_assignme...ROG\Oshajith (51))*

```
442  --room status update trigger
443  CREATE TRIGGER tg_discharge_record_roomstat
444  ON discharge_record
445  AFTER INSERT
446      AS
447          DECLARE @status CHAR(1)
448          SET @status='Y'
449
450          DECLARE @discharge_id INT
451          SELECT @discharge_id = i.discharge_id FROM inserted i
452
453          DECLARE @add_id INT
454          SELECT @add_id = pa.admission_id FROM patient_admission pa JOIN discharge_record dr ON pa.admission_id=dr.admission_id
455
456          DECLARE @room_no VARCHAR(4)
457          SELECT @room_no = room_no FROM patient_admission WHERE admission_id =@add_id
458
459          UPDATE room
460          SET status = @status WHERE room.room_no = @room_no
461          PRINT('Room'+@room_no+' is now free.')
462
463
```

Data Entry

```
--INSERTING SAMPLE DATA

INSERT INTO department VALUES('DEP001','phsychology','block 1','check ups and treatments')
INSERT INTO department VALUES('DEP002','cardiology','block2','operations and treatmens')
INSERT INTO department VALUES('DEP003','neurology','block3','check ups and treatments')
INSERT INTO department VALUES('DEP004','fertility clinic','block4','check ups and operations')
INSERT INTO department VALUES('DEP005','dental clinic','block5','treatments and surgery')



INSERT INTO employee VALUES('EM001','rajitha','senarathna','02-05-2008','891653129v','diploma in
nursing','M','NO12,sarawanamuttu road,colombo 10','nurse','DEP001')
INSERT INTO employee VALUES('EM002','amali','perera','03-04-2014','785641296v','diploma in
pharmaceutical','F','NO35,navam mawatha,colombo 08','other','DEP002')
INSERT INTO employee VALUES('EM003','mangalika','kumarihami','09-09-2002','562314789v','higher
diploma in nursing','F','NO63,buthgamuwa road,kandy','attendent','DEP003')
INSERT INTO employee VALUES('DR001','pasindu','perera','07-03-
2015','944759635v','pshychology','M','NO811,thilakarathana mawatha,colombo 10','doctor','DEP001')
INSERT INTO employee VALUES('DC001','oshadi','wijesingha','09-08-
2010','923614578v','neurolgy','F','NO578,suboda road,homagama','doctor','DEP003')
INSERT INTO employee VALUES('EM004','kumaran','padmanadan','11-05-
2009','831245796v','technician','M','NO36,pillayan road,jaffna','other','DEP004')
INSERT INTO employee VALUES('EM005','sumana','widisingha','02-05-2001','591397846v','diploma in
nursing','M','NO213,koggala road,galle','nurse','DEP005')
INSERT INTO employee VALUES('DR002','kanchana','rathnayaka','03-08-
2015','967826375v','cardiology','M','NO212,grand garden,colombo 07','doctor','DEP002')
INSERT INTO employee VALUES('DC002','kameshi','ganegama','11-04-
2011','764139651v','fertility','F','NO555,nihal road,colombo 12','doctor','DEP004')

INSERT INTO employee_phone VALUES('EM001','0773583258')
INSERT INTO employee_phone VALUES('EM002','0715655321')
INSERT INTO employee_phone VALUES('EM003','0789655231')
INSERT INTO employee_phone VALUES('DR001','0728936455')
INSERT INTO employee_phone VALUES('DC001','0779653865')
INSERT INTO employee_phone VALUES('EM004','0714563821')
INSERT INTO employee_phone VALUES('EM005','0764156333')
INSERT INTO employee_phone VALUES('DR002','0770294518')
INSERT INTO employee_phone VALUES('DC002','0724512378')

INSERT INTO doctor VALUES('DR001','psychology')
INSERT INTO doctor VALUES('DC001','neurology')
INSERT INTO doctor VALUES('DC002','fertility')
INSERT INTO doctor VALUES('DR002','cardiology')

INSERT INTO callon_doctor VALUES('DC001','1000')
```

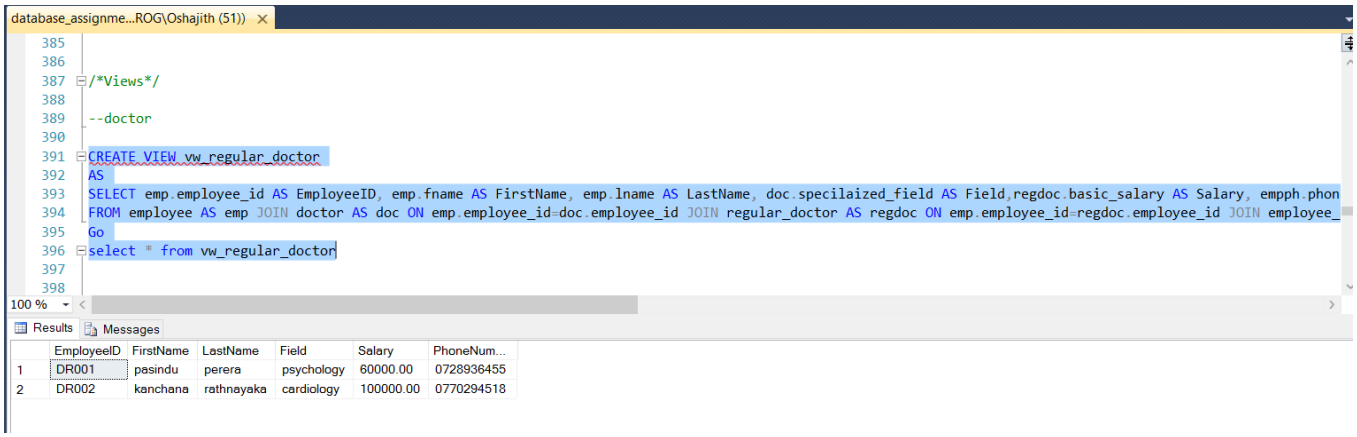
```
INSERT INTO callon_doctor VALUES('DC002','2500')

INSERT INTO regular_doctor VALUES('DR001','60000')
INSERT INTO regular_doctor VALUES('DR002','100000')

INSERT INTO room VALUES('P001','Y','private','500','DEP001')
INSERT INTO room VALUES('P002','N','private','500','DEP002')
INSERT INTO room VALUES('P003','Y','private','500','DEP003')
INSERT INTO room VALUES('P004','Y','private','500','DEP004')
INSERT INTO room VALUES('P005','N','private','500','DEP005')
INSERT INTO room VALUES('G001','Y','general','100','DEP001')
INSERT INTO room VALUES('G002','N','general','100','DEP002')
INSERT INTO room VALUES('G003','N','general','100','DEP003')
INSERT INTO room VALUES('G004','Y','general','100','DEP004')
INSERT INTO room VALUES('G005','N','general','100','DEP005')


INSERT INTO patient VALUES('PT0001','lasitha','tharanga','02-01-2016','05-06-1994','NO45,udara
road,anuradhapura','M')
INSERT INTO patient VALUES('PT0002','ishara','oshajith','06-08-2012','11-03-1993','NO63,mahawa
road,mahawa','M')
INSERT INTO patient VALUES('PT0003','devaki','nayanahari','08-10-2009','01-01-
1990','NO789,wedamulla road,wattala','F')
INSERT INTO patient VALUES('PT0004','kausha','thathsarani','02-03-2014','02-09-
1994','NO777,walasmulla road,matara','F')
INSERT INTO patient VALUES('PT0005','kamesh','ganegama','03-03-2003','06-08-1997','NO478,udugamuwa
road,kotte','M')

INSERT INTO patient_phone VALUES('PT0001','0778569412')
INSERT INTO patient_phone VALUES('PT0002','0721568496')
INSERT INTO patient_phone VALUES('PT0003','0718965243')
INSERT INTO patient_phone VALUES('PT0004','0789531249')
INSERT INTO patient_phone VALUES('PT0005','0112560139')

INSERT INTO check_up VALUES('1','normal','02-02-2016','mental depression','500','PT0001','DR001')
INSERT INTO check_up VALUES('2','bad','03-02-2016','High Cholestarol','500','PT0002','DR002')
INSERT INTO check_up VALUES('3','bad','03-02-2016','nurve damage in arm','500','PT0003','DC001')
INSERT INTO check_up VALUES('4','normal','03-02-2016','baby scan','500','PT0004','DC002')
INSERT INTO check_up VALUES('5','bad','04-02-2016','high blood pressure and
cholestarol','500','PT0005','DR002')

INSERT INTO patient_admission VALUES('1','08-02-
2016','cash','25000','weak','liyanage','0778945163','2','P002')
INSERT INTO patient_admission VALUES('2','08-03-
2016','creditcard','25000','weak','chanuka','0729645789','5','P005')
INSERT INTO patient_admission VALUES('3','08-04-
2016','insuarance','15000','weak','kanthi','0785693125','3','G002')

INSERT INTO operation VALUES('1','08-05-2016','cardiactric surgery','high','clearing the blocked
blood vain in the heart','diet control','20000','1','DR002')
INSERT INTO operation VALUES('2','08-06-2016','cardiactric surgery','high','clearing the blocked
blood vain in the heart','diet control','20000','2','DR002')
INSERT INTO operation VALUES('3','08-07-2016','neurologic surgery','medium','repairing damaged
nurves','physio theropy','15000','3','DC001')

INSERT INTO discharge_record VALUES('1','08-21-2016','good','','diet control and excercise','1')
INSERT INTO discharge_record VALUES('2','08-25-2016','good','','diet control and excercise','2')
INSERT INTO discharge_record VALUES('3','08-29-2016','good','','physio theropy','3')

INSERT INTO treatments VALUES('1','medicine course','use medicine and do meditation','','1')
INSERT INTO treatments VALUES('2','medicine course','use medicine and do excercise','','2')
INSERT INTO treatments VALUES('3','medicine course','use medicine and do meditation','','3')
INSERT INTO treatments VALUES('4','medicine course','use medicine ','','4')
INSERT INTO treatments VALUES('5','medicine course','use medicine and do excercise','','5')
```

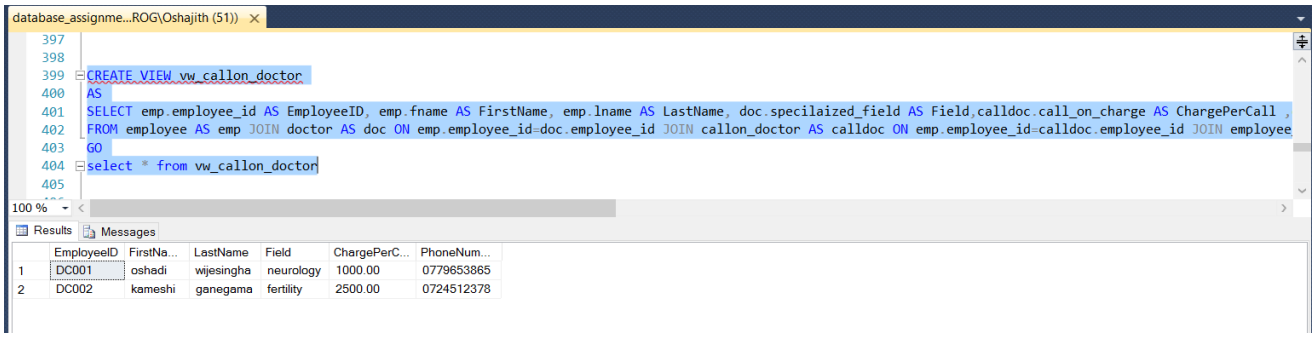
```
INSERT INTO drug VALUES('DRU001','paracetamol',2,'disprine','according to doctors prescription')
INSERT INTO drug VALUES('DRU002','methfomine',15,'glycomatt','according to doctors prescription')
INSERT INTO drug VALUES('DRU003','diagin',5,'milk of magnizium','according to doctors
prescription')
INSERT INTO drug VALUES('DRU004','amoxilin',10,'augmantine','according to doctors prescription')
INSERT INTO drug VALUES('DRU005','piriton',5,' Chlorphenamine','according to doctors prescription')

INSERT INTO payment VALUES('1','','all payments done','PT0001')
INSERT INTO payment VALUES('2','','all payments done','PT0002')
INSERT INTO payment VALUES('3','','all payments done','PT0003')
INSERT INTO payment VALUES('4','','all payments done','PT0004')
INSERT INTO payment VALUES('5','','all payments done','PT0005')

INSERT INTO checkup_drug VALUES('15','DRU001','1')
INSERT INTO checkup_drug VALUES('10','DRU002','2')
INSERT INTO checkup_drug VALUES('8','DRU003','3')
INSERT INTO checkup_drug VALUES('12','DRU004','4')
INSERT INTO checkup_drug VALUES('20','DRU001','5')
```

Views

```
/*Views*/

--doctor

CREATE VIEW vw_regular_doctor
AS
SELECT emp.employee_id AS EmployeeID, emp.fname AS FirstName, emp.lname AS LastName,
doc.specilaized_field AS Field,regdoc.basic_salary AS Salary, empph.phone_number AS PhoneNumber
FROM employee AS emp JOIN doctor AS doc ON emp.employee_id=doc.employee_id JOIN regular_doctor AS
regdoc ON emp.employee_id=regdoc.employee_id JOIN employee_phone AS empph ON
emp.employee_id=empph.employee_id
Go
select * from vw_regular_doctor


CREATE VIEW vw_callon_doctor
AS
SELECT emp.employee_id AS EmployeeID, emp.fname AS FirstName, emp.lname AS LastName,
doc.specilaized_field AS Field,calldoc.call_on_charge AS ChargePerCall ,empph.phone_number AS
PhoneNumber
FROM employee AS emp JOIN doctor AS doc ON emp.employee_id=doc.employee_id JOIN callon_doctor AS
calldoc ON emp.employee_id=calldoc.employee_id JOIN employee_phone AS empph ON
emp.employee_id=empph.employee_id
GO
select * from vw_callon_doctor


--nurse

CREATE VIEW vw_nurse
AS
SELECT emp.employee_id AS EmployeeID,emp.fname AS FirstName,emp.lname AS
LastName,ephone.phone_number AS PhoneNumber
From employee AS emp JOIN employee_phone As ephone
ON emp.employee_id=ephone.employee_id AND emp.employee_type='nurse'
GO
select * from vw_nurse


--attendant
```

```
CREATE VIEW vw_attendent
AS
SELECT emp.employee_id AS EmployeeID,emp.fname AS FirstName,emp.lname AS
LastName,ephone.phone_number AS PhoneNumber
From employee AS emp JOIN employee_phone As ephone
ON emp.employee_id=ephone.employee_id AND emp.employee_type='attendent'
GO
select * from vw_attendent


--other_staff

CREATE VIEW vw_other_staff
AS
SELECT emp.employee_id AS EmployeeID,emp.fname AS FirstName,emp.lname AS
LastName,ephone.phone_number AS PhoneNumber
From employee AS emp JOIN employee_phone As ephone
ON emp.employee_id=ephone.employee_id AND emp.employee_type='other'
GO
select * from vw_other_staff
```





database_assignme...ROG\Oshajith (51))

```
418  --attendant
419
420  CREATE VIEW vw_attendent
421  AS
422  SELECT emp.employee_id AS EmployeeID,emp.fname AS FirstName,emp.lname AS LastName,ephone.phone_number AS PhoneNumber
423  From employee AS emp JOIN employee_phone As ephone
424  ON emp.employee_id=ephone.employee_id AND emp.employee_type='attendent'
425  GO
426  select * from vw_attendent
427
428
```

100 %

Results | Messages

	EmployeeID	FirstName	LastName	PhoneNum...
1	EM003	mangalika	kumarihami	0789655231

database_assignme...ROG\Oshajith (51))

```
427
428
429  --other_staff
430
431  CREATE VIEW vw_other_staff
432  AS
433  SELECT emp.employee_id AS EmployeeID,emp.fname AS FirstName,emp.lname AS LastName,ephone.phone_number AS PhoneNumber
434  From employee AS emp JOIN employee_phone As ephone
435  ON emp.employee_id=ephone.employee_id AND emp.employee_type='other'
436  GO
437  select * from vw_other_staff
438
```

100 %

Results | Messages

	EmployeeID	FirstNa...	LastName	PhoneNum...
1	EM002	amali	perera	0715655321
2	EM004	kumaran	padmanadan	0714563821

database_assignme...ROG\Oshajith (51))

```
406
407  --nurse
408
409  CREATE VIEW vw_nurse
410  AS
411  SELECT emp.employee_id AS EmployeeID,emp.fname AS FirstName,emp.lname AS LastName,ephone.phone_number AS PhoneNumber
412  From employee AS emp JOIN employee_phone As ephone
413  ON emp.employee_id=ephone.employee_id AND emp.employee_type='nurse'
414  GO
415  select * from vw_nurse
416
```

100 %

Results | Messages

	EmployeeID	FirstNa...	LastName	PhoneNum...
1	EM001	rajitha	senarathna	0773583258
2	EM005	sumana	widisingha	0764156333

Student Workload Matrix

Index Number	ER/EER Diagram	Relational Mapping, Data Normalisation, Data Dictionary	Tables, Constraints	Views, Triggers	Stored Procedures, User Defined Functions
10569207	X		X	X	X
10568999		X	X	X	X
10569206	X				
10569083		X	X		
10569071	X	X	X		

Peer evaluation form for group work

(1=strongly disagree; 2=disagree; 3=agree; 4=strongly agree)

Evaluation Criteria	**10569207**	**10568999**	**10569206**	**10569206**	**10569071**
Attends group meetings regularly and arrives on time.	04	04	04	04	04
Contributes meaningfully to group discussions.	04	04	04	04	04
Completes the tasks on time.	03	03	03	03	03
Prepares work in a quality manner.	04	04	03	03	03
Contributes significantly to the success of the project in a cooperative and supportive attitude.	04	04	03	03	03
TOTALS	19	19	17	17	17

Conclusion

- This hospital management system is make most of the management processes of the Hospital autonomous and will reduces the workload of the staff and minimize the use of resources.
- This hospital management system is also capable of maintain the information about the mentioned elements and can manipulate data accordingly.