

**Module Title Database Design and Development**

**Assignment Title Asia Royal Hospital’s Patients Management System**

**Examination Cycle WINTER 2021**

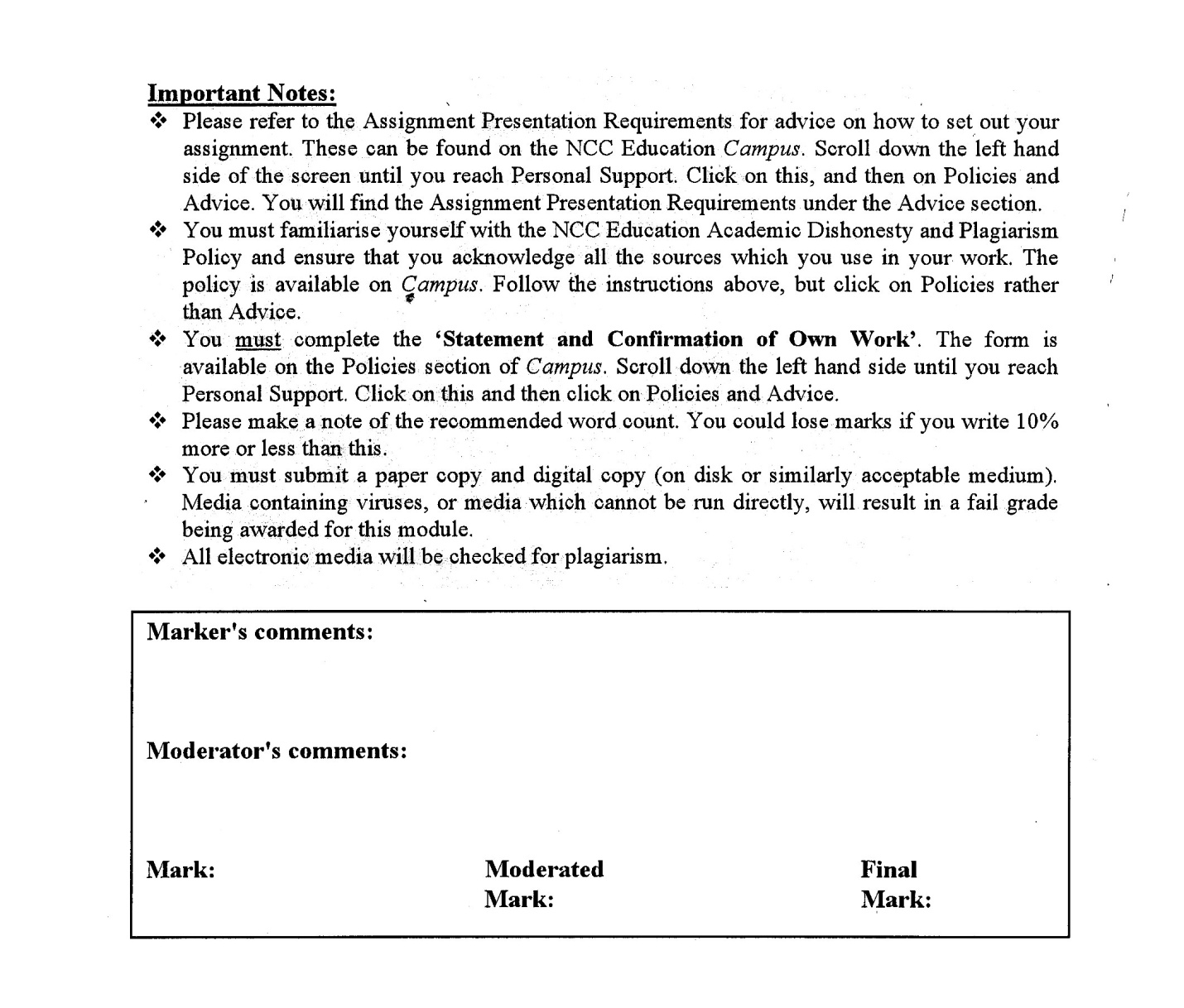
**Candidate Name HTET YAMIN EI**

**Candidate No**

**00000000**

**Centre Name KMD INSTITUTE (YANGON)**

**Submission Date: 30.10.2021**



****

**Statement and Confirmation of Own Work**

**Programmed/Qualification name: Level 5 DBIT Business**

*All NCC Education assessed assignments submitted by students must have this statement as the cover page or it will not be accepted for marking. Please ensure that this statement is either firmly attached to the cover of the assignment or electronically inserted into the front of the assignment.*

**Student declaration**

I have read and understood NCC Education’s Policy on Academic Dishonesty and Plagiarism.

I can confirm the following details:

**Student ID/Registration number : 00000000**

**Name : HTET YAMIN EI**

**Centre Name : KMD Computer Centre (Yangon)**

**Module Name : Database Design and Development**

**Module Leader : KAY THWE WIN**

**Number of words : (2145) words excluding from tables and headings**

I confirm that this is my own work and that I have not plagiarized any part of it. I have also noted the assessment criteria and pass mark for assignments.

**Due Date : 30-10-2021**

**Student Signature : Ei**

**Submitted Date : 30-10-2021**

Table of Contents

[Introduction 6](#_Toc86582790)

[TASK – 1 7](#_Toc86582791)

[Description of Business 7](#_Toc86582792)

[1.1 Scenario 8](#_Toc86582793)

[TASK – 2 14](#_Toc86582794)

[ER and Data Dictionary 14](#_Toc86582795)

[2.1. Entity Relationship Diagram (ERD) for Asia Royal’s Patient Management System 15](#_Toc86582796)

[2.2. Data Dictionary 15](#_Toc86582797)

[TASK – 3 22](#_Toc86582798)

[Normalisation 22](#_Toc86582799)

[3.1. Purpose of Normalisation 23](#_Toc86582800)

[3.2. Problems of Update Anomalies 29](#_Toc86582801)

[TASK - 4 30](#_Toc86582802)

[Assessment of Design 30](#_Toc86582803)

[4.1. Mapping Logical Database Design to Physical Database Design 31](#_Toc86582804)

[4.2. Designed Tables for Target DBMS 31](#_Toc86582805)

[4.3. Derived Data 31](#_Toc86582806)

[TASK 5 32](#_Toc86582807)

[Scripts to Create Table Structures 32](#_Toc86582808)

[5.1. PatientType 33](#_Toc86582809)

[5.2. Patient 33](#_Toc86582810)

[5.3. Doctor 34](#_Toc86582811)

[5.4. Appointment 34](#_Toc86582812)

[5.5. Nurse 35](#_Toc86582813)

[5.6. RoomType 35](#_Toc86582814)

[5.7. Room 36](#_Toc86582815)

[5.8. Admission 36](#_Toc86582816)

[5.9. Specialist 37](#_Toc86582817)

[5.10. Test 37](#_Toc86582818)

[5.11. TestResult 38](#_Toc86582819)

[5.12. Medicine 38](#_Toc86582820)

[5.13. Treatment 39](#_Toc86582821)

[5.14. Enhancing Database by the Use of SQL 40](#_Toc86582822)

[Task – 6 41](#_Toc86582823)

[Data Population 41](#_Toc86582824)

[6.1. PatientType 42](#_Toc86582825)

[6.2. Patient 42](#_Toc86582826)

[6.3. Doctor 43](#_Toc86582827)

[6.4. Nurse 45](#_Toc86582828)

[6.5. Specialist 46](#_Toc86582829)

[6.6. Test 48](#_Toc86582830)

[6.7. Medicine 50](#_Toc86582831)

[6.8. RoomType 52](#_Toc86582832)

[6.9. Room 53](#_Toc86582833)

[6.10. Appointment 54](#_Toc86582834)

[6.11. TestResult 56](#_Toc86582835)

[6.12. Admission 57](#_Toc86582836)

[6.13. Treatment 59](#_Toc86582837)

[TASK – 7 61](#_Toc86582838)

[SQL Reports 61](#_Toc86582839)

[7.1. Changing Drug’s Name and Price 62](#_Toc86582840)

[7.2. Deleting Unavailable Test from Test Table 62](#_Toc86582841)

[7.3. Checking Which Room is Occupied by Which Patient 63](#_Toc86582842)

[7.4. Checking Average Age of All Patients 63](#_Toc86582843)

[7.5. Adding a New Column in Treatment Table 64](#_Toc86582844)

[7.6. Deriving Data (Calculating Total Cost of Drugs Bought) 65](#_Toc86582845)

[7.7. Retrieving the Patients Who Had Taken the Most Expensive Test 65](#_Toc86582846)

[7.8. Checking Patients Who Had Both Appointments and Tests Using Inner Join 66](#_Toc86582847)

[7.9. Retrieving No. of Patients Taken in Each Test 66](#_Toc86582848)

[7.10. Calculating the Total Cost of All Drugs Bought by Patient P0008 67](#_Toc86582849)

[TASK – 8 68](#_Toc86582850)

[Future Development of Distributed Database 68](#_Toc86582851)

[8.1. Distributed Database System 69](#_Toc86582852)

[8.2. Components of a Distributed Database Management System 69](#_Toc86582853)

[8.3. Factors Considered to Implement a Distributed Database 69](#_Toc86582854)

[8.4. Implementation of Distributed Database 70](#_Toc86582855)

[8.5. Adaptation to Potential Future Expansion 70](#_Toc86582856)

[TASK – 9 71](#_Toc86582857)

[Learning Undertaken to Complete This Assignment 71](#_Toc86582858)

[9.1. Description 72](#_Toc86582859)

[9.2. Feelings 72](#_Toc86582860)

[9.3. Evaluation 72](#_Toc86582861)

[9.4. Analysis 72](#_Toc86582862)

[9.5. Conclusion 72](#_Toc86582863)

[9.6. Action Plan 72](#_Toc86582864)

[References 73](#_Toc86582865)

# Introduction

Asia Royal Hospital is a 250-bedded private hospital opened on 18th March 2000. It’s an eleven-story twin building in No.14, Baho Street, Sanchaung Township, Yangon, Myanmar. Multidisciplinary medical care service is provided with modern medical equipments. Prompt and efficient treatment is given to patients by well-trained residential doctors, nurses and aided by professional medical staffs at every specific department.

# TASK – 1

# Description of Business

## 1.1 Scenario

Asia Royal Hospital is a 250-bedded private hospital with multidisciplinary medical care service in Yangon, Myanmar. A database is needed for managing appointments, admissions, test results and treatments of patients in the hospital. This database will not be focused on the details of medicine, insurance or payment made by patients and staff managements of the hospital.

A patient can have many appointments with one or more doctors and a doctor will have several appointments in a day. Doctors are defined by their specializations. Patients are defined by inpatient, outpatient and emergency patient. A doctor may prescribe at least one medication for the patient as treatment.

If the patient is in severe conditions, admission is required for close monitoring overnight. They are called inpatients. One doctor can admit many patients. A patient may be admitted more than once. A room is used for each admission and at least one nurse is stand-by for each room, one nurse may work in many rooms.

When the patient requires to take tests like lab tests or x-ray scan tests, each test result refers to one patient and one specialist is responsible for at least one test result.

Examples of data records are shown below.

**Document 1. Appointments That Patient ID P0009 Had**

|  |  |
| --- | --- |
| Patient ID: P0009  First Name: Shannon  Last Name: Williams | Appointment Date: 12.10.2021 |
| Gender: Female **Doctor ID**: D005  Age: 23 **Doctor Name**: Dr. Peter Evans  Marital Status: Single  Phone: 09254134469  Address: Dagon, Yangon  Patient Type: IP | |
|  | |
| Patient ID: P0009  First Name: Shannon  Last Name: Williams | *Appointment Date*: 18.10.2021 |
| Gender: Female **Doctor ID**: D005  Age: 23 **Doctor Name**: Dr. Peter Evans  Marital Status: Single  Phone: 09254134469  Address: Dagon, Yangon  Patient Type: IP | |

**Document 2. Patients’ Admissions**

|  |  |
| --- | --- |
| Patient ID: P0001  First Name: James  Last Name: Smith | Admission Date: 19.10.2021 |
| Gender: Male  Age: 68  Email:jamessmith@gmail.com  Marital Status: Married  Phone: 09254012356  Guardian: Ricky Smith  Address: Kyauktada, Yangon  Patient Type: IP | *Admitted by*  *Doctor ID:* D001  *Doctor Name:* Dr. Ben Stokes  *Room No.:* R003  *Nurse ID:* N002  *Nurse Name:*Lily Jackson |
| **Agreement** | |
| I give permission to health professionals in Asia Royal Hospital involved in my care for this admission to hospital and to access health information about me which is related to my current treatment. I comprehend that I am responsible for any remaining balance if my procedure is not completely covered by insurance.  **Signed by**  **J.Smith** | |

|  |  |
| --- | --- |
| Patient ID: P0006  First Name: David  Last Name: Menkin | Admission Date: 25.10.2021 |
| Gender: Male  Age:45  Email: davidmenkin@gmail.com  Marital Status: Single  Phone: 09773315624  Guardian: Gordon Menkin  Address: Kyauktada, Yangon  Patient Type: IP | *Admitted by*  *Doctor ID:* D007  *Doctor Name:* Dr. Jane Stacy  *Room No.:* R010  *Nurse ID:* N006  *Nurse Name:*Sophie Thompson |
| **Agreement** | |
| I give permission to health professionals in Asia Royal Hospital involved in my care for this admission to hospital and to access health information about me which is related to my current treatment. I comprehend that I am responsible for any remaining balance if my procedure is not completely covered by insurance.  **Signed by**  **D.Menkin** | |

**Document 3. Patient’s Test Results**

|  |
| --- |
| *Patient ID:* P0004  *First Name:* Samuel  *Last Name:* Cardoza  *Patient Type:* OP  *Test ID:* CBC  *Specialist ID:* S01  *Specialist Name:* Susan Ward |
|  |
| *Total Cost: $ 140.00* |

|  |
| --- |
| *Patient ID:* P0001  *First Name:* James  *Last Name:* Smith  *Patient Type:* IP  *Test ID:* ECG  *Specialist ID:* S04  *Specialist Name:* Tom Curran |
|  |
| *Total Cost: $ 160.00* |

**Document 4. Patient’s Treatments**

|  |  |  |  |
| --- | --- | --- | --- |
| *Patient ID:* P0008  *First Name:* Carolina  *Last Name:* Ravassa  *Patient Type:* IP  *Doctor ID*: D009 | | | |
| **Medicine Name** | **Quantity** | **Cost** | **Amount** |
| Neomycin | 3 | $ 86.47 | $ 259.41 |
| Botox | 2 | $ 15.00 | $ 30.00 |
| Total:  Paid:  Change: | | | $ 289.41 |
| $ 290 |
| $ 0.59 |
| Goods sold are not returnable. Thanks! | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| *Patient ID:* P0010  *First Name:* Annie  *Last Name:* Frank  *Patient Type:* IP  *Doctor ID:* D004 | | | |
| **Medicine Name** | **Quantity** | **Cost** | **Amount** |
| Ralista | 10 | $ 29.97 | $ 299.70 |
| Folic Acid | 10 | $ 11.06 | $ 110.60 |
| Total:  Paid:  Change: | | | $ 410.30 |
| $ 415 |
| $ 4.70 |
| Goods sold are not returnable. Thanks! | | | |

# TASK – 2

# ER and Data Dictionary

## **2.1. Entity Relationship Diagram (ERD) for Asia Royal’s Patient Management** System

The relation between each table is as explained in the scenario of task 1.

## 2.2. Data Dictionary

Integrity constraints are a set of rules which is used to sustain the qualities of information. They ensure that inserting, updating data and other processes have to be executed in such a way that data integrity is not altered. Therefore, integrity constraints are used to act as a safeguard against accidental impairment to the database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity: **PatientType**  Primary Key: PatientTypeID  Foreign Key: - | | | | |
| Attribute | Datatype | Size | Constraints | Description |
| PatientTypeID | Varchar | 10 | Entity Integrity,  Domain Constraint | Unique identification for patient types |
| TypeName | Varchar | 20 | Domain Constraint | Types of Patients |

Domain Constraints on PatientType Table  
Check (PatientTypeID in (‘IP, ‘OP’, ‘EP’)),  
Check (TypeName in (‘Inpatient’, ‘Outpatient’, ‘Emergency Patient’))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity: **Patient**  Primary Key: PatientID  Foreign Key: PatientTypeID | | | | |
| Attribute | Datatype | Size | Constraints | Description |
| PatientID | Varchar | 10 | Entity Integrity,  Domain Constraint | Identification number of the patient |
| FirstName | Varchar | 30 | - | First Name of the patient |
| LastName | Varchar | 30 | - | Last Name of the patient |
| Gender | Varchar | 10 | Domain Constraint | Gender of the patient |
| Age | Integer | - | - | Patient’s age |
| MaritalStatus | Varchar | 10 | Domain Constraint | Love life of patient |
| Address | Varchar | 100 | - | The location where the patient lives |
| Phone | Varchar | 15 | - | Contact number of patient |
| Email | Varchar | 50 | - | Patient’s email |
| Guardian | Varchar | 50 | - | Name of patient’s guardian |
| PatientTypeID | Varchar | 10 | Referential Integrity,  Domain Constraint | Inpatient or Outpatient |

Referential Integrity on Patient Table  
Foreign Key (PatientTypeID) references PatientType (PatientTypeID)

Domain Constraints on Patient Table  
Check (PatientID like (‘P%’)),

Check (Gender in (‘Male’, ’Female’)),

Check (MaritalStatus in (‘Single’, ‘Married’, ‘Divorced’, ‘Widowed’))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity: **Doctor**  Primary Key: DocID  Foreign Key: - | | | | |
| Attribute | Datatype | Size | Constraints | Description |
| DocID | Varchar | 10 | Entity Integrity,  Domain Constraint | Identification number of the doctor |
| DocName | Varchar | 80 | - | Name of the doctor |
| Specialization | Varchar | 30 | - | Doctor’s specialization field |
| ConsultantFees | Decimal | (10,2) | - | Doctor’s consultation fees |
| AvailableTime | Varchar | 50 | - | The time when Doctors sit in the hospital to see patients |

Domain Constraint on Doctor Table  
Check (DocID like (‘D%’)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity: **Appointment**  Primary Key: ApptID  Foreign Key: PatientID, DocID | | | | |
| Attribute | Datatype | Size | Constraints | Description |
| ApptID | Varchar | 10 | Entity Integrity, Domain Constraint | Appointment number |
| ApptDate | Date |  | - | Date scheduled for the appointment |
| PatientID | Varchar | 10 | Referential Integrity, Propagation | Identification number of the patient |
| DocID | Varchar | 10 | Referential Integrity,  Propagation | Identification number of the doctor |

Domain Constraint on the Appointment Table  
Check (ApptID like (‘Appt%’))

Propagation Constraint on the Appointment Table

Foreign Key (PatientID) references Patient (PatientID),

Foreign Key (DocID) references Doctor (DocID)

On delete no action

On update cascade

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity: **Nurse**  Primary Key: NurseID  Foreign Key: - | | | | |
| Attribute | Datatype | Size | Constraints | Description |
| NurseID | Varchar | 10 | Entity Integrity, Domain Constraint | Identification number of the nurse |
| NurseName | Varchar | 50 | - | Name of the nurse |

Domain Constraint on the Nurse Table  
Check (NurseID like (‘N%’))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity: **RoomType**  Primary Key: RoomType  Foreign Key: - | | | | |
| Attribute | Datatype | Size | Constraints | Description |
| RoomType | Varchar | 20 | Entity Integrity | Type of Bedroom |
| Cost | Decimal | (10,2) | - | Price of each room |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity: **Room**  Primary Key: RoomNo  Foreign Key: NurseID, RoomType | | | | |
| Attribute | Datatype | Size | Constraints | Description |
| RoomNo | Varchar | 10 | Entity Integrity, Domain Constraint | Room Number |
| RoomType | Varchar | 20 | Domain Constraint | Types of room (E.g. – VIP room) |
| NurseID | Varchar | 10 | Referential Integrity,  Propagation | ID of the ward where the room is located |

Propagation Constraint on Room Table

Foreign Key (RoomType) references RoomType (RoomType),

Foreign Key (NurseID) references Nurse (NurseID)

On delete set null

On update cascade

Domain Constraint on Room Table  
Check (RoomNo like (‘R%’)),

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity: **Admission**  Primary Key: AdmID  Foreign Key: PatientID, DocID, RoomNo | | | | |
| Attribute | Datatype | Size | Constraints | Description |
| AdmID | Varchar | 10 | Entity Integrity,  Domain Constraint | Identification number of admission |
| AdmDate | Date |  | - | Date of admission |
| PatientID | Varchar | 10 | Referential Integrity, Propagation | ID number of the admitted patient |
| DocID | Varchar | 10 | Referential Integrity, Propagation | ID number of the doctor who admitted the patient |
| RoomNo | Varchar | 10 | Referential Integrity, Propagation | The room where admitted patient stays |

Propagation Constraint on Admission Table

Foreign Key (PatientID) references Patient (PatientID),

Foreign Key (RoomNo) references Room (RoomNo),

Foreign Key (DocID) references Doctor (DocID)

On delete no action

On update cascade

Domain Constraint on Admission Table  
Check (AdmID like (‘Adm%’))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity: **Specialist**  Primary Key: SpecialistID  Foreign Key: - | | | | |
| Attribute | Datatype | Size | Constraints | Description |
| SpecialistID | Varchar | 10 | Entity Integrity, Domain Constraint | ID number of the Laboratorian |
| SpecialistName | Varchar | 80 | - | Name of the Laboratorian |

Domain Constraint on Specialist Table  
Check (SpecialistID like (‘S%’))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity: **Test**  Primary Key: TestID  Foreign Key: - | | | | |
| **Attribute** | **Datatype** | **Size** | **Constraint** | **Description** |
| TestID | Varchar | 10 | Entity Integrity | ID of the lab test |
| TestName | Varchar | 80 | - | Name of the test |
| Cost | Decimal | (10,2) | - | Cost of each test |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity: **TestResult**  Primary Key: LabResultID  Foreign Key: PatientID, SpecialistID, TestID | | | | |
| Attribute | Datatype | Size | Constraints | Description |
| TestResultID | Varchar | 10 | Entity Integrity,  Domain Constraint | Lab result of the patient |
| SpecialistID | Varchar | 10 | Referential Integrity, Propagation | ID of the Laboratorian who ran the test |
| PatientID | Varchar | 10 | Referential Integrity, Propagation | Identification number of the patient |
| TestID | Varchar | 10 | Referential Integrity, Propagation | ID of the lab test |

Propagation Constraint on TestResult Table

Foreign Key (PatientID) references Patient (PatientID),

Foreign Key (TestID) references LabTest (TestID),

Foreign Key (SpecialistID) references Specialist (SpecialistID)

On delete no action

On update cascade

Domain Constraint on TestResult Table  
Check (TestResultID like (‘T%’))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity: **Medicine**  Primary Key: MedID  Foreign Key: - | | | | |
| Attribute | Datatype | Size | Constraints | Description |
| MedID | Varchar | 10 | Entity Integrity,  Domain Constraint | ID number of the medicine |
| MedName | Varchar | 50 | - | Name of the medicine |
| Cost | Decimal | (10,2) | - | Cost of each medicine per card |

Domain Constraint on Medicine Table  
Check (MedID like (‘M%’))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entity: **Treatment**  Primary Key: TreatmentID  Foreign Key: PatientID, DocID, MedID | | | | |
| Attribute | Datatype | Size | Constraints | Description |
| TreatmentID | Varchar | 10 | Entity Integrity,  Domain Constraint | Unique number of Treatment |
| PatientID | Varchar | 10 | Referential Integrity, Propagation | ID of Patient who had treatment |
| DocID | Varchar | 10 | Referential Integrity, Propagation | Doctor’s ID who gave treatment |
| MedID | Varchar | 10 | Referential Integrity, Propagation | ID of the drug given |
| Quantity | Integer | - | - | Quantity of each drug bought |

Domain Constraint on Treatment Table

Check (TreatmentID like (‘TM%’))

Propagation Constraint on Treatment Table

Foreign Key (PatientID) references Patient (PatientID),

Foreign Key (MedID) references Medicine (MedID),

Foreign Key (DocID) references Doctor (DocID)

On delete no action

On update cascade

# TASK – 3

# Normalisation

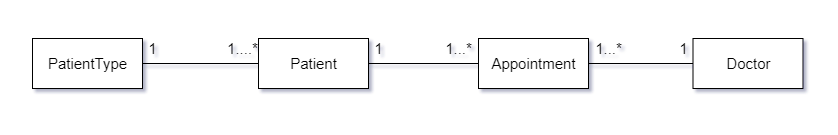
## 3.1. Purpose of Normalisation

Normalisation is the process of systematizing data in a database. The purpose of normalisation is to remove redundancy and inconsistent dependency and to eliminate insert, update, delete anomalies by dividing larger tables into smaller ones and establishing relationships between them. (Jayaram, 2018)

**For Appointments That Patient ID P0009 Had,**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UNF** | **Level** | **1NF** | **2NF** | **3NF** | **Entity Name** |
| PatientID  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID  TypeName  ApptDate  DocID  DocName  Specialization  ConsultantFees  AvailableTime | 1  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2 | PatientID(PK)  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID  TypeName  PatientID(FK)  ApptDate  DocID  DocName  Specialization  ConsultantFees  AvailableTime | PatientID(PK)  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID  TypeName  PatientID(FK)  DocID(FK)  ApptDate  DocID(PK)  DocName  Specialization  ConsultantFees  AvailableTime | PatientID(PK)  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID(FK)  PatientTypeID(PK)  TypeName  ApptID(PK)  ApptDate  PatientID(FK)  DocID(FK)  DocID(PK)  DocName  Specialization  ConsultantFees  AvailableTime | Patient  PatientType  Appointment  Doctor |

PK = Primary Key

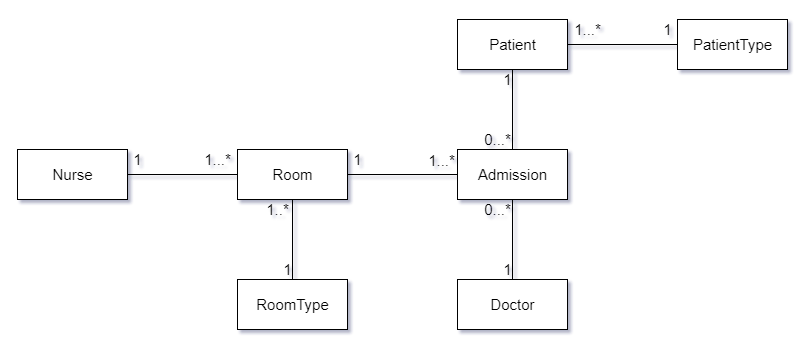
FK = Foreign Key

In unnormalised form, some attributes are added as required for detailed information. In first normal form, single group (PatientID, PatientName and etc.) and repeating group (ApptDate, DocID and etc.) are differentiated from un-normalised form. In second normal form, partial key (DocID) is removed and a new table is made. In third normal form, a new attribute “ApptID” is added as primary key for appointment table and the result is 4 tables.

**For Patients’ Admissions,**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UNF** | **Level** | **1NF** | **2NF** | **3NF** | **Entity Name** |
| PatientID  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID  TypeName  DocID  DocName  Specialization  ConsultantFees  AvailableTime  AdmDate  RoomNo  RoomType  Cost  NurseID  NurseName | 1  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2  2  2  2 | PatientID(PK)  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID  TypeName  PatientID(FK)  DocID  DocName  Specialization  ConsultantFees  AvailableTime  AdmDate  RoomNo  RoomType  Cost  NurseID  NurseName | PatientID(PK)  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID  TypeName  PatientID(FK)  RoomNo(FK)  DocID  DocName  Specialization  ConsultantFees  AvailableTime  AdmDate  RoomNo(PK)  RoomType  Cost  NurseID  NurseName | PatientID(PK)  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID(FK)  PatientTypeID(PK)  TypeName  AdmID(PK)  PatientID(FK)  RoomNo(FK)  DocID(FK)  AdmDate  DocID(PK)  DocName  Specialization  ConsultantFees  AvailableTime  RoomNo(PK)  RoomType(FK)  NurseID(FK)  RoomType(PK)  Cost  NurseID(PK)  NurseName | Patient  PatientType  Admission  Doctor  Room  RoomType  Nurse |

PK = Primary Key  
FK = Foreign Key

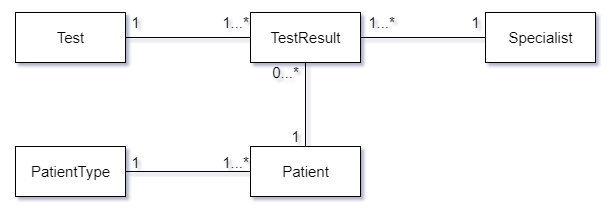


Normalisation is done the same as explained in above. In third normal form, “AdmID” is added as primary key for the dummy table and Patient, PatientType, Admission, Room, RoomType, Nurse and Doctor tables are obtained as final result.

**For Patient’s Test Results,**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UNF** | **Level** | **1NF** | **2NF** | **3NF** | **Entity Name** |
| PatientID  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID  TypeName  TestID  TestName  Cost  SpecialistID  SpecilalistName | 1  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2 | PatientID(PK)  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID  TypeName  PatientID(FK)  TestID  TestName  Cost  SpecialistID  SpecilalistName | PatientID(PK)  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID  TypeName  PatientID(FK)  TestID(FK)  SpecialistID  SpecilalistName  TestID(PK)  TestName  Cost | PatientID(PK)  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID(FK)  PatientTypeID(PK)  TypeName  TestResultID(PK)  PatientID(FK)  TestID(FK)  SpecialistID(FK)  SpecialistID(PK)  SpecilalistName  TestID(PK)  TestName  Cost | Patient  PatientType  TestResult  Specialist  Test |

PK = Primary Key

FK = Foreign Key

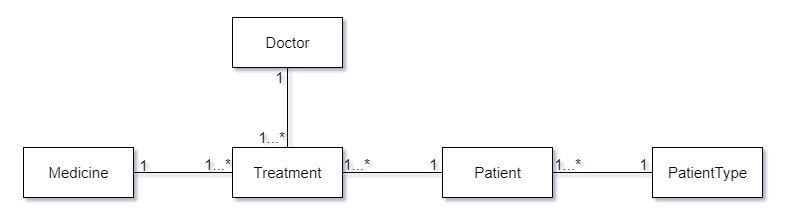
The steps involved are the same as explained in the first normalisation. In third normal form, “TestResultID” is added as primary key for the dummy table and Patient, PatientType, Test, TestResult and Specialist tables are acquired as final outcome.

**For Patient’s Treatments,**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UNF** | **Level** | **1NF** | **2NF** | **3NF** | **Entity Name** |
| PatientID  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID  TypeName  DocID  DocName  Specialization  ConsultantFees  AvailableTime  Quantity  MedicineID  MedicineName  Cost | 1  1  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2  2  2 | PatientID(PK)  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID  TypeName  PatientID(FK)  DocID  DocName  Specialization  ConsultantFees  AvailableTime  Quantity  MedicineID  MedicineName  Cost | PatientID(PK)  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID  TypeName  PatientID(FK)  DocID(FK)  Quantity  MedicineID  MedicineName  Cost  DocID(PK)  DocName  Specialization  ConsultantFees  AvailableTime | PatientID(PK)  PatientName  Gender  Age  MaritalStatus  Address  Phone  Email  Guardian  PatientTypeID(FK)  PatientTypeID(PK)  TypeName  TreatmentID(PK)  PatientID(FK)  DocID(FK)  MedicineID(FK)  Quantity  MedicineID(PK)  MedicineName  Cost  DocID(PK)  DocName  Specialization  ConsultantFees  AvailableTime | Patient  PatientType  Treatment  Medicine  Doctor |

PK = Primary Key

FK = Foreign Key

The steps included are similar with as explained in the first normalisation. In third normal form, “TreatmentID” is added as the primary key for the dummy table and Patient, PatientType, Treatment and Medicine tables are obtained as final product.

## 3.2. Problems of Update Anomalies

Update anomalies are data inconsistencies which arises from partial update and data redundancy. For example, consider the TestResult table isn’t normalised recording the TestID, TestName and Cost. If we want to update the cost of a test, we would have to update on every row in which the test is done. To overcome this, table is split as required so that TestID, TestName and Cost are stored in a separate table.

# TASK - 4

# Assessment of Design

## 4.1. Mapping Logical Database Design to Physical Database Design

Logical database design is mapped to physical database design by 3 types of mapping: one-to-many mapping, many-to-many mapping, and domain mapping. For one-to-many relations, only two tables are created. For many-to-many relations, a dummy table is created. For domain mapping, some columns are created as new tables. (E.g. PatientType column is separated from the Patient table as a new table). DBMS (SQL Server 2012) and SQL language (DDL, DML, DCL, DIL) are used for creating this database.

## 4.2. Designed Tables for Target DBMS

The logical data model is reviewed table by table and column by column to check what is actually required, what can be removed, which table should be broken into smaller tables and which data elements should be merged to reduce response time and query and enhance ease of use. After consideration using iterative approach, 13 tables including the dummy tables are derived in physical database design.

## 4.3. Derived Data

A variety of optimizations are evaluated for their applicability to the logical data model before the database is generated. These include trade-offs such as splitting one entity to multiple tables (E.g. Patient table and PatientType table), storing derived data which gives summary information in a column based on calculated attributes and so on. Deriving data will be done in the following tasks below.

In Treatment table, total cost of drug that the patient bought will be calculated and stored in a new column named “TotalCost”. See Task – 7 for codes.



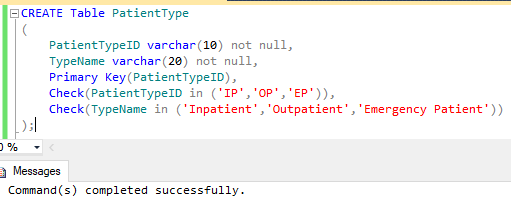
Treatment  
TreatmentID(PK)  
PatientID(FK)  
MedID(FK)  
Quantity  
/TotalCost (t.Quantity\*m.Cost   
 FROM Treatment tm, Medicine m   
 WHERE tm.MedID = m.MedID)

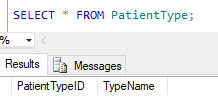
# TASK 5

# Scripts to Create Table Structures

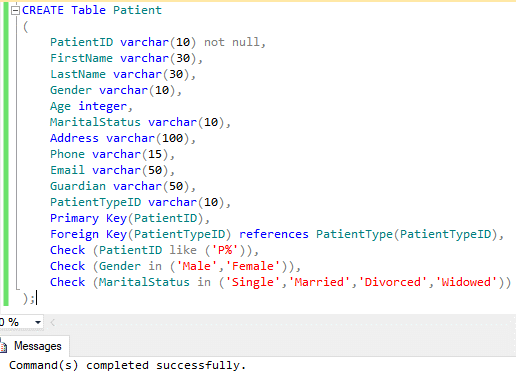
The scripts used to create the database are developed in the following order.

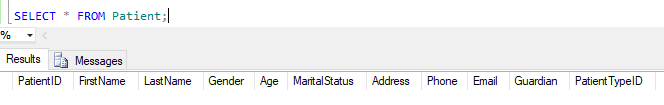
## 5.1. PatientType



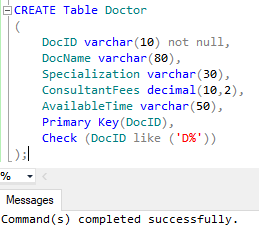


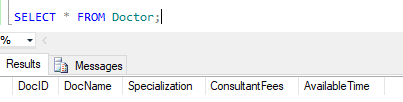
## 5.2. Patient



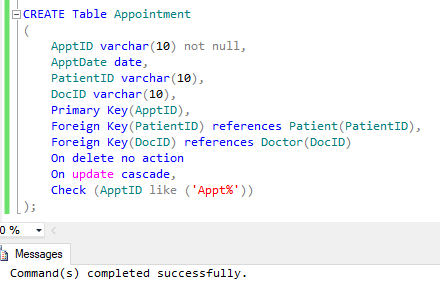


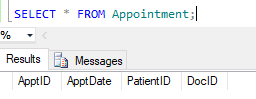
## 5.3. Doctor



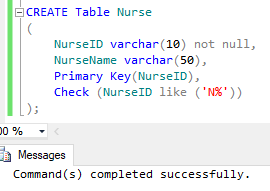


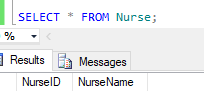
## 5.4. Appointment



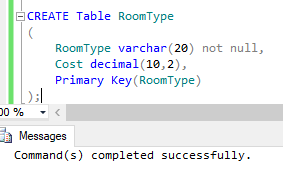


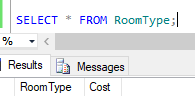
## 5.5. Nurse



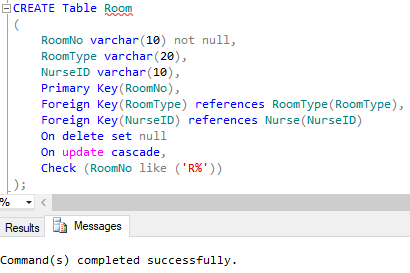


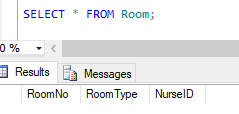
## 5.6. RoomType



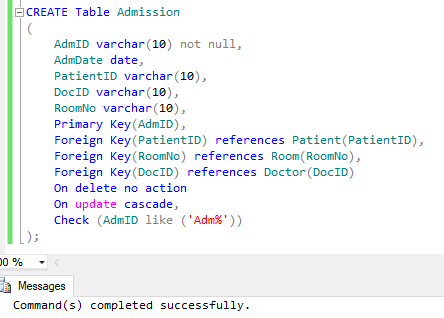


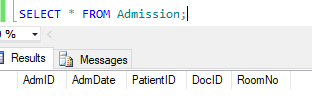
## 5.7. Room



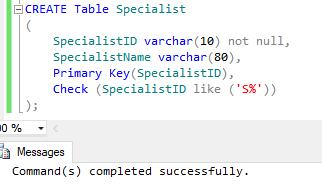


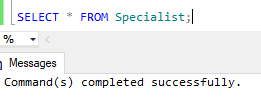
## 5.8. Admission



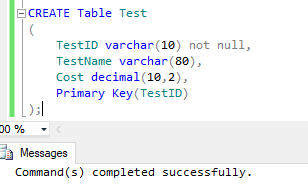


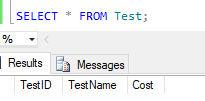
## 5.9. Specialist



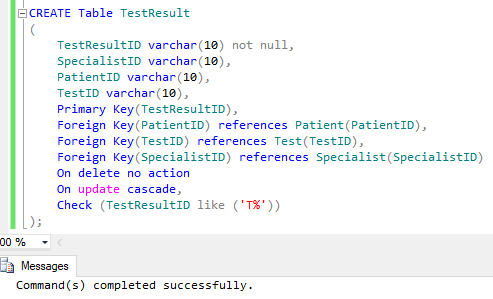


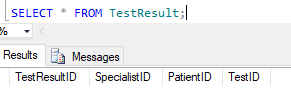
## 5.10. Test



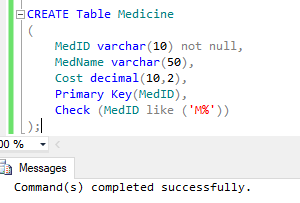


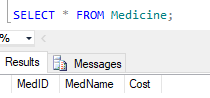
## 5.11. TestResult



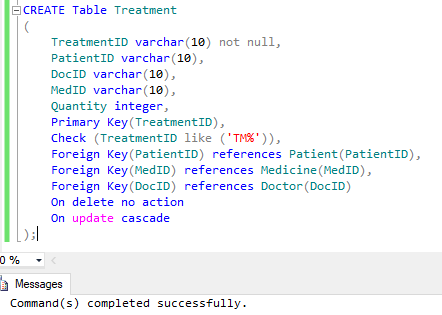


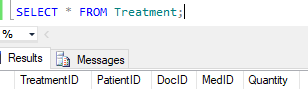
## 5.12. Medicine





## 5.13. Treatment





## 5.14. Enhancing Database by the Use of SQL

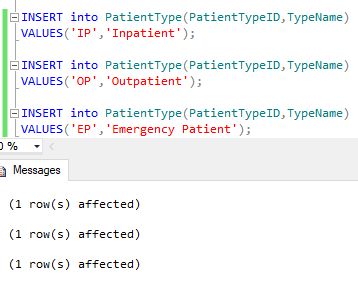
The above stated SQL statements are used for creating the database. Decimal datatype is utilized for further calculations in tables which include price column. The other columns use varchar datatype and for quantity column in Medicine table, integer datatype is used for calculating derived data. Detailed information can be stored, retrieved and managed instantly because of the usage of SQL.

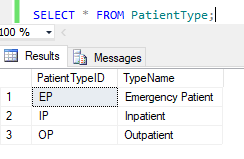
# Task – 6

# Data Population

Data inserted in this database are fictional. They’re inserted in the following order table by table.

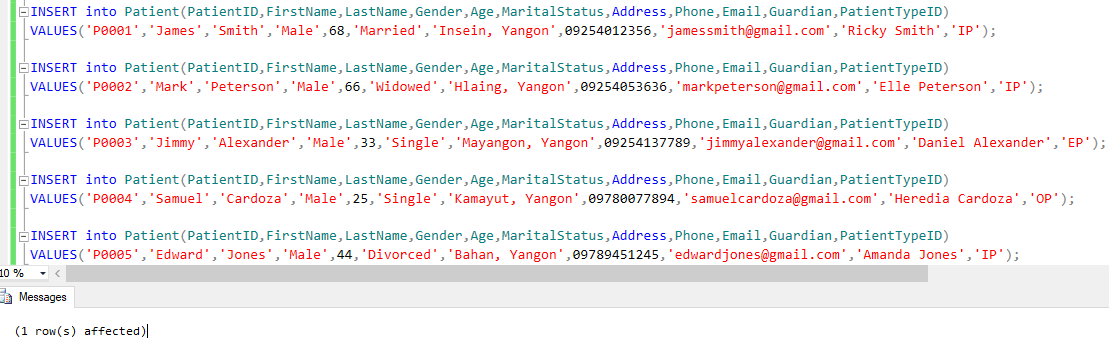
## 6.1. PatientType

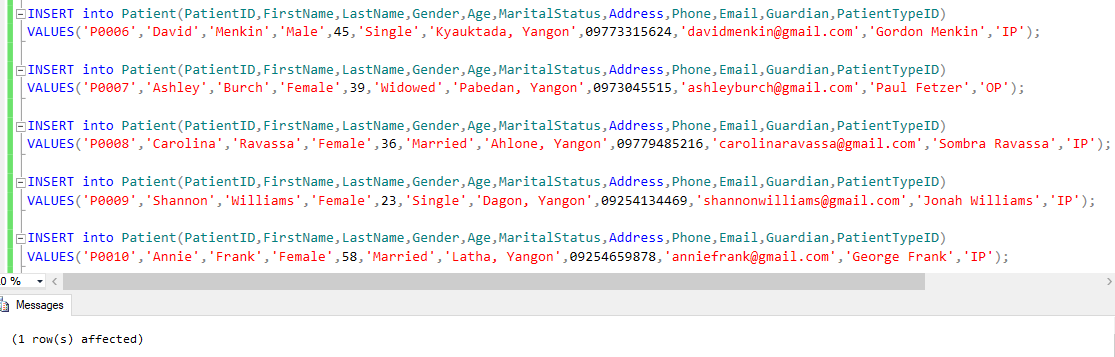


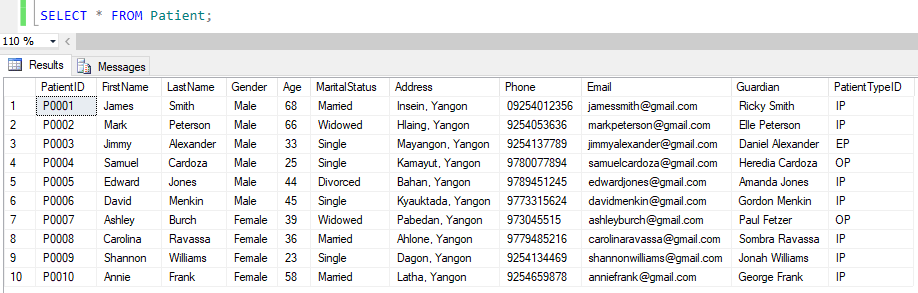


## 6.2. Patient

The screenshots are quite small because of many columns in the table.

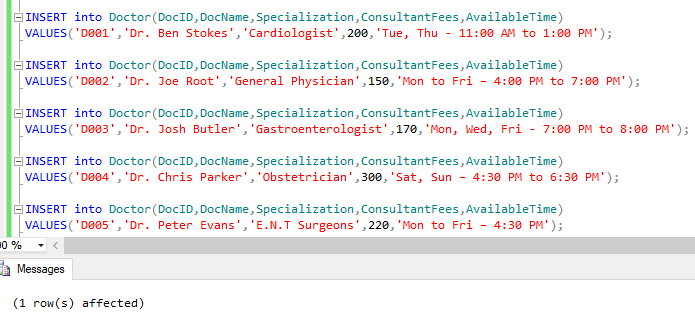


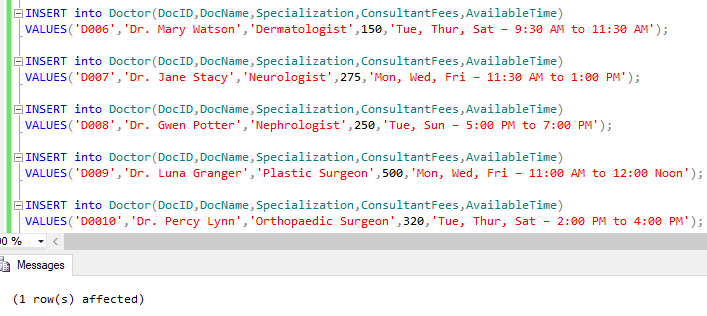




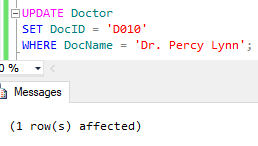
## 6.3. Doctor

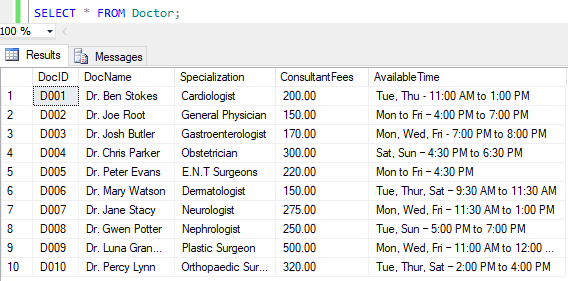
The consultation fees are in dollars.



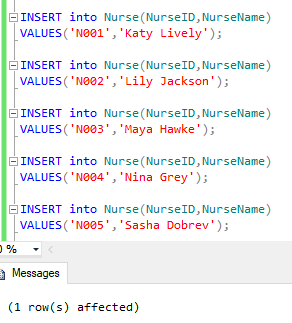


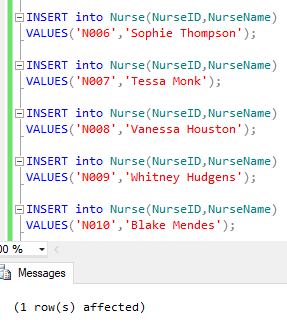
I mistakenly added Dr. Lynn’s ID so I had to update his ID.

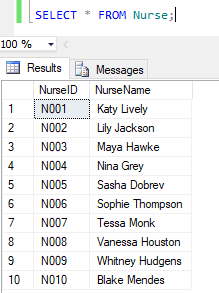




## 6.4. Nurse

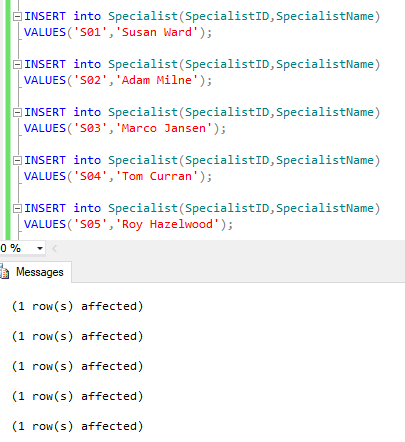


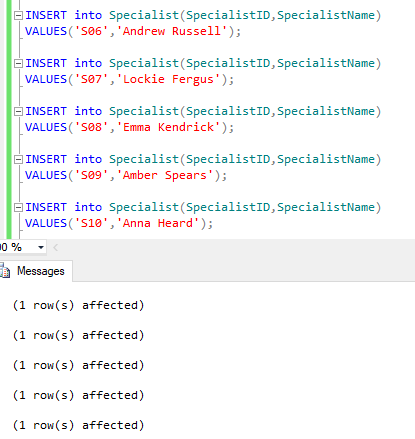


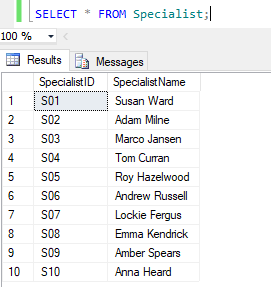


## 6.5. Specialist

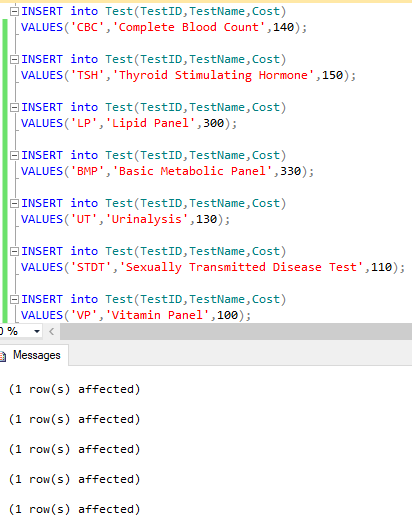
Specialists include radiologists and pathologists.

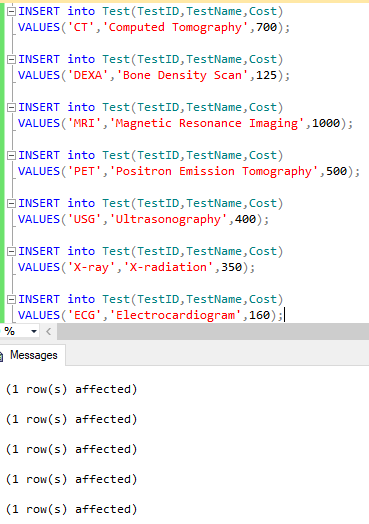


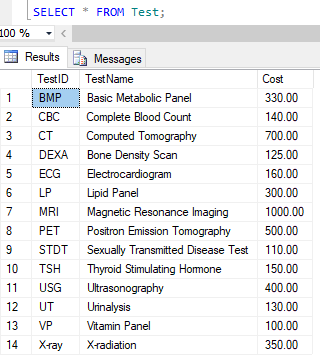




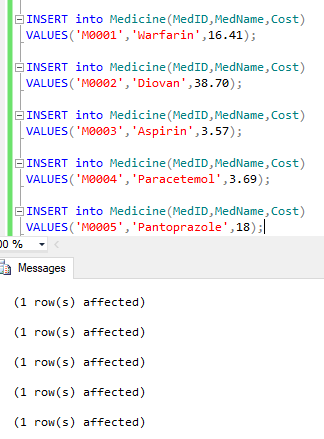
## 6.6. Test

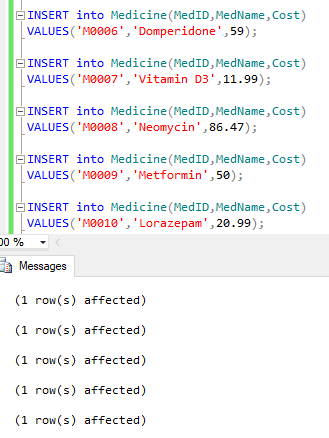




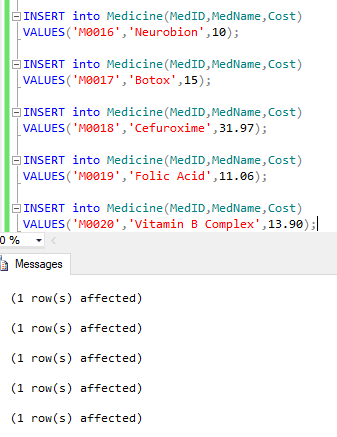


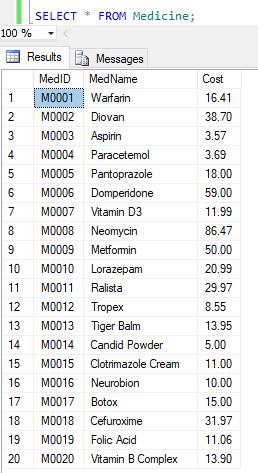
## 6.7. Medicine



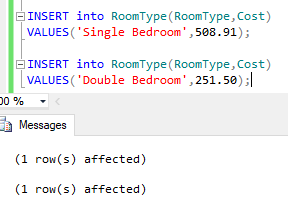


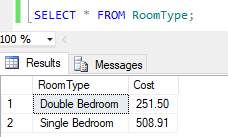






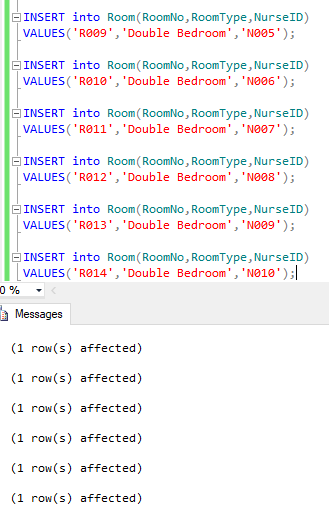
## 6.8. RoomType

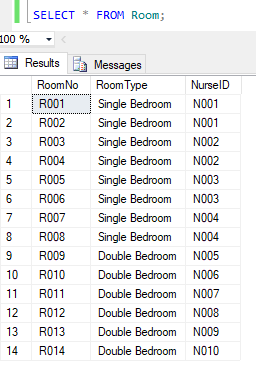




## 6.9. Room



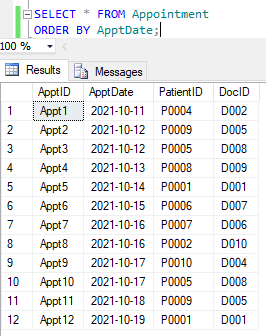


****

## 6.10. Appointment

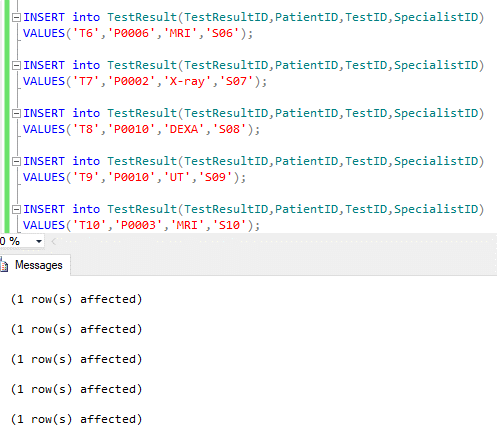


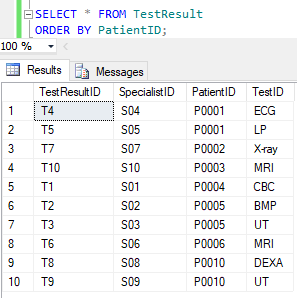




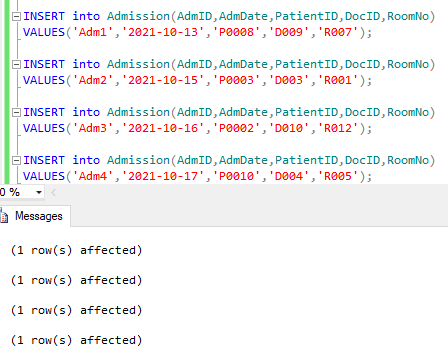
## TestResult

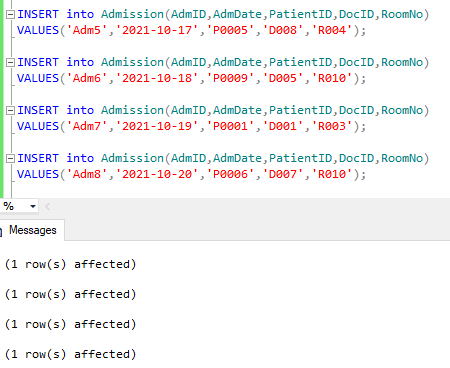


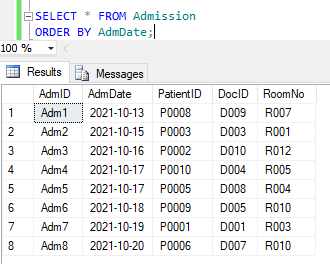




## Admission



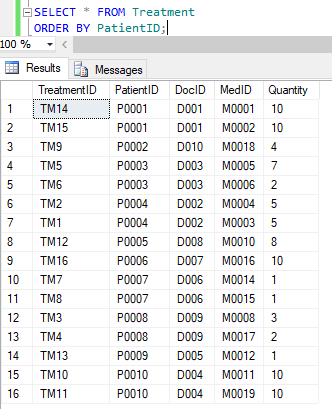




## Treatment



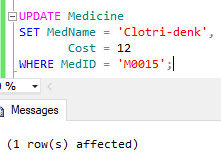


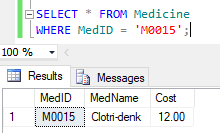


# TASK – 7

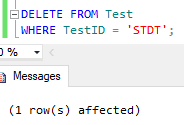
# SQL Reports

## 7.1. Changing Drug’s Name and Price

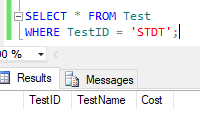




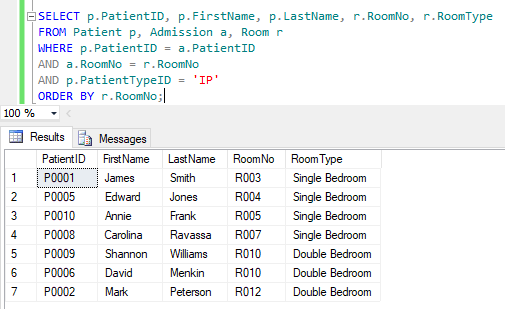
## 7.2. Deleting Unavailable Test from Test Table



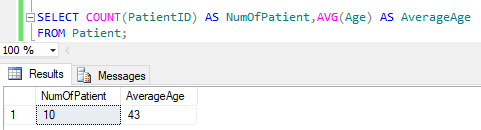
Since it’s deleted, the data is gone.



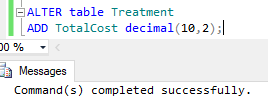
## 7.3. Checking Which Room is Occupied by Which Patient

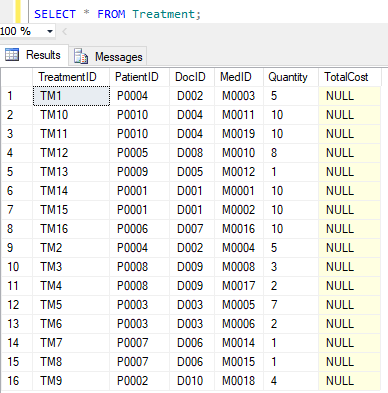


## 7.4. Checking Average Age of All Patients

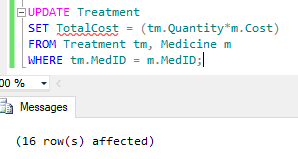


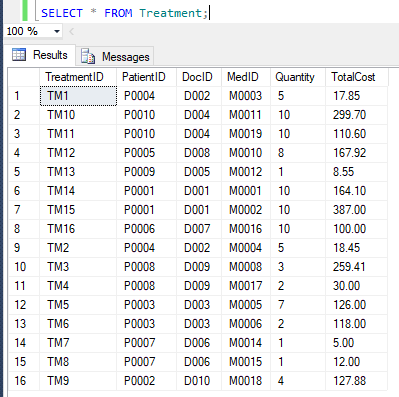
## 7.5. Adding a New Column in Treatment Table



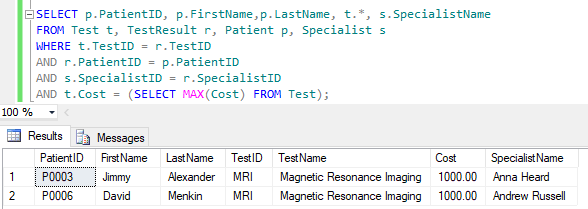


## 7.6. Deriving Data (Calculating Total Cost of Drugs Bought)

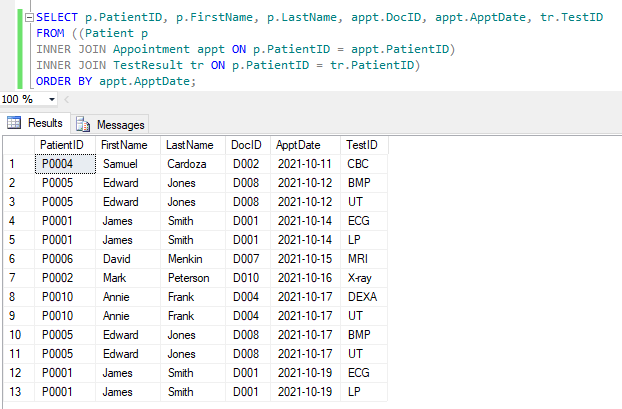




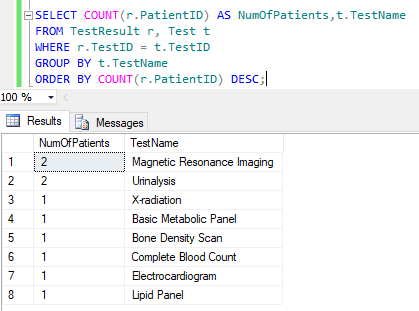
## 7.7. Retrieving the Patients Who Had Taken the Most Expensive Test



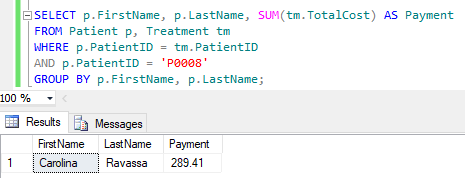
## 7.8. Checking Patients Who Had Both Appointments and Tests Using Inner Join



## 7.9. Retrieving No. of Patients Taken in Each Test



## 7.10. Calculating the Total Cost of All Drugs Bought by Patient P0008



# TASK – 8

# Future Development of Distributed Database

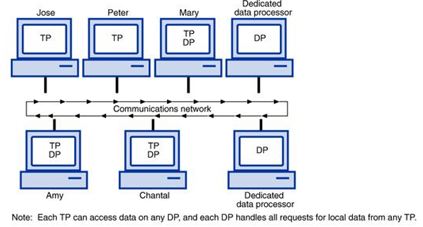
## 8.1. Distributed Database System

A distributed database is a collection of several interrelated databases where data is stored in more than one site. It is not limited to one system. DBMS can manage data in each site independent of other sites. Fragmentation and replication make sure that the distributed databases remain contemporary.

## 8.2. Components of a Distributed Database Management System

They include computer workstations (sites or nodes) forming the network system, network components which reside in each workstation, communications media which carries data from one node to another, transaction processor (TP) known as the transaction manager (TM) or the application processor (AP) that requests data remotely and locally and data processor (DP), known as the data manager (DM) which stores and retrieves data situated at the site.

The interaction among TPs and DPs illustrated in the figure is generated through a specific set of rules, used by DDBMS. (THIRU, n.d.)



(THIRU, n.d.)

## 8.3. Factors Considered to Implement a Distributed Database

Modular development can be done where systems are expanded by adding local data and new computers to the new site and connecting them to the distributed system without getting interrupted. The system will stop completely when failures arise in centralized databases. However, the system will still function at reduced performance until the error is fixed when a part fails in distributed database systems. If the data is localized near where it's most used, lower communication costs is needed for distributed database systems which is impossible in centralized systems. DDBMS support both OLTP and OLAP. (Moore, n.d.)

## 8.4. Implementation of Distributed Database

Replication and fragmentation ensure that distributed databases remain up-to-date and current. By using replication, identical data can be accessed locally, thus avoiding traffic. Less complex fragmentation divides the whole database into various sub-tables. Replication seems to be more suitable for hospital information system. It is especially effective for reading only the generated data in one organization unit by other units. Nonetheless, asynchronous replication appears to be the only practical solution which must be used methodically because it can hide a network failure preventing access to the most up-to-date information. Of 3 types of distributed database: Homogenous, Heterogeneous, Federated, heterogeneous type is more preferable for a hospital in my opinion. It allows separate site to develop without an overall central plan in case of potential future expansion. (Anon., n.d.) (Wojciechowski, n.d.)

## 8.5. Adaptation to Potential Future Expansion

The hospital processes huge volumes of data where even a powerful server might not be able to handle efficiently. Data distribution allows to improve performance. In the case of hospital information system, site autonomy is important which is an another advantage of data distribution. Each unit has its own server controlling its data and can decide which data should be made accessible for other units. The biggest drawback of a distributed system is a likelihood of losing access to remote data in case of remote network failure but replication can minimize consequences of such failures.

# TASK – 9

# Learning Undertaken to Complete This Assignment

## 9.1. Description

While working on the assignment, electricity got cut frequently and WIFI doesn’t work sometimes. Moreover, during the period, my whole family including me got infected with Covid-19. I had to take 2 weeks break from doing the assignment to take care of them.

## 9.2. Feelings

I get frustrated every time I think about the problems I’ve gone through. I had little motivation to do the assignment after what had happened. But I changed the way I think and I somehow managed to motivate myself and got the work done. Looking back, I’m feeling quite satisfied that I completed the assignment in time even with the issues I had.

## 9.3. Evaluation

The fact that we got Covid-19 gave me the idea to write about patient management in hospital for the assignment. I was kind of confused what would be the assignment title. The only bad thing about all these incidents was that I didn’t have the same amount of time as other students to work on the assignment.

## 9.4. Analysis

I think the reason that the assignment worked out well is because I was in close contact with the hospital system and I kept on observing how things work and this assignment is the result of my observations. I also did some research on my own on how the hospital information system works. After completing the assignment, I realised there’s some weak points in my database like how I assumed and established the “Treatment” and “Appointment” tables. The current database allows treatments to be added without appointments. Besides, nurses are assigned to rooms, not to patients. Some rooms may be empty, which means no work for those nurses.

## 9.5. Conclusion

Doing this assignment made me think that anything is possible if u try hard. Besides, I also learned that health is the most important thing in life. I also comprehend that my skills aren’t enough to actually handle a real hospital database system.

## 9.6. Action Plan

I have changed my WIFI with a new ISP not to get interrupted during the work. I will take additional care of my health and improve my researching skills to understand more about patient database management system by reading more articles. In conclusion, If I have to do projects like this in the future, I guarantee that I will try my best to be better than what I did.

(Anon., 2020)

# References

Anon., 2020. *Gibbs' Reflective Cycle.* [Online]   
Available at: https://www.ed.ac.uk/reflection/reflectors-toolkit/reflecting-on-experience/gibbs-reflective-cycle  
[Accessed 18 October 2021].

Anon., n.d. *Distributed Database.* [Online]   
Available at: https://www.networxsecurity.org/members-area/glossary/d/distributed-database.html  
[Accessed 17 October 2021].

Jayaram, P., 2018. *What is Database Normalization in SQL Server?.* [Online]   
Available at: https://www.sqlshack.com/what-is-database-normalization-in-sql-server/  
[Accessed 5 October 2021].

Moore, L., n.d. *Distributed Database.* [Online]   
Available at: https://searchoracle.techtarget.com/definition/distributed-database  
[Accessed 16 October 2021].

THIRU, n.d. *Components of DDBMS.* [Online]   
Available at: http://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/551-components-of-ddbms.html  
[Accessed 16 October 2021].

Wojciechowski, M., n.d. *DISTRIBUTING AND REPLICATING DATA IN HOSPITAL INFORMATION SYSTEMS.* [Online]   
Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.80.4695&rep=rep1&type=pdf  
[Accessed 17 October 2021].

Candidate Checklist

Please use the following checklist to ensure that your work is ready for submission.



Have you read the NCC Education documents 'What is Academic Misconduct? Guidance for Candidates' and 'Avoiding Plagiarism and Collusion: Guidance for Candidates' and ensured that you have acknowledge all the sources that you have used in your work?

Have you completed the 'Statement and Confirmation of Own Work' form and attached it to your assignment? You must do this.



Have you ensured that your work has not gone over or under the recommended word count by more than 10%?



Have you ensured that your work does not contain viruses and can be run directly?

