Module Code & Module Title

CC5051NI- Database

Year and Semester

2020-21 Autumn

Student Name: Pratik Amatya

Group: C1

London Met ID: 19031389

College ID: NP01CP4A190024

Assignment Due Date: Sunday 20 December 2020

Assignment Submission Date: Sunday 20 December 2020

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Introduction of the College

Islington college is an educational institution located in Kathmandu, Nepal. The college provides degrees in IT and Business Fields. The college offers bachelor's degrees in multimedia technologies, computing and computer networking & IT security for an IT degree. Likewise, they offer BBA degree in International Business, Finance and Marketing for a Business Degree. All the programmes provided by the college is affiliated with London Metropolitan University.

There is a level system where in each level, students are taught different modules. The college provides facilities such as cafeterias, library, lecture halls, computer labs, networking labs, multimedia lab, discussion rooms, tutorial rooms and seminar rooms to their students. The college are always listening to the feedback from their students and help them as much as they can to make the study environment more enjoyable and productive.

Being the most prestigious private college in Nepal is the vision of the college. Their aim is to produce IT and business professionals who are competitive and industry ready by teaching them the latest and up-to-date skills and knowledge. (Islington, 2018)

1.2 Current Business Activities and Operations

Islington college has been providing quality education to its students through operation of the following activities:

1. The college provides courses such as BBA and BIT.
2. Each course has a course leader to whom students can ask clarify their doubt any time.
3. Student needs to choose a course in other to be in enrolled to the college. Then, the student needs to select a specification from the course to specialize at. The specifications for each of the bachelor's course are provided below:
    a. If the student chooses BIT course, he/she can choose between specifications such as multimedia technologies, computing and computer networking & IT security.
    b. Similarly, if the student chooses BBA course, he/she can choose between specifications such as International Business, Finance and Marketing.
4. Each specifications of the course have multiple modules. Different specifications have some common modules. Different modules are taught dependent on specification and the student level.
5. The different specifications include modules such as database, Information system, Network and operating systems, etc.
6. All the person related to the college are categorized as a Person having Instructor and Student as its type.
7. Each person is required to provide at least one address designated as the mailing address. The address should contain details such as house number, street, city, province, country, fax number and landline number. The person is not required to provide the fax number or the landline number of the address.
8. Each instructor may have a role of course leader, module leader or tutor. Each instructor is associated with one course only.
9. Each module is taught by multiple instructors. The students are allowed to ask their queries in any class.
10. There are 6 classes where the class sessions are held. They each have their own name such as Pokhara room, Lumbini room, Patan room, Everest room, Bhaktapur room and Lukla room.

11. The students check the total marks obtained from all modules they are enrolled in after two weeks of having given tests by logging in the online website of the college using their ID. The records will be empty for the one level student.
12. The college increases the monthly salary of instructors for every year from the joined date to appreciate their work and dedication to the college.
13. The students as well employees can check their details such as their first name, last name, address, contact info, and fax number kept on the college database using their ID. They can contact the college if the details have to be changed or updated.
14. Every student can apply for instructor role after graduation for their respective course.
15. The students can pay for the course they have chosen through the online payment as well. The total fee amount is different depending on the course and not the specification.

1.3 Business Rules

The college stores many details in its database for many purposes. The college keeps records of each person engaged with them so as to monitor their progress and guide them to be more productive. The following business rules define how the data is stored in the database:

- A person can be either instructor or student. They cannot be both.
- A person is associated with only one course and specification. This is because a student is allowed to choose only one course and one specification of it and an instructor is associated with only one course and specification of it.
- Each course has multiple specifications but each specification is associated with one course.
- A person can have single or multiple addresses. But it is necessary to have at least one address.
- A person can be associated with multiple modules. This case occurs as if the person is an instructor then they may teach multiple modules and if the person is a student, then the student is enrolled in multiple modules.
- Each specification has multiple modules and each module is associated with multiple specifications but only one course.

- Each course is associated with multiple specification and each specification is associated with a course only.

1.4 Identification of Entities and Attributes

A real-world object that is easily distinguishable from the others is called an entity. For example, in a school database, teachers employed by the school, the students enrolled in the school and the courses offered by the school can be considered as entities. All these entities have some attributes that give them their identity.

For example, a teacher entity may have name and course as attributes.

There are some cases when some common attributes are shared among a few entities in a data model. Based on the attributes, these entities are categorized into supertype and sub types.

- Super type

  Supertype is a generic entity that forms relationship with one or more sub types and contains attributes that its shares with it subtypes. (Oracle, 2020)

  A subtype discriminator is an attribute in the supertype entity that determines to which subtype the supertype occurrence is related. (Palmer, 2020)

- Sub type

  A subtype is sub-groups of the subtype entity that inherit all the supertype attributes and have unique attributes that are different from other subtypes. (Oracle, 2020)

  There are mainly two types of sub type. They are:

- Disjoint subtypes

  The subtypes contain a unique subset of the supertype entity set. In disjoint subtypes, each entity instance of the supertype may only exist as one of the subtypes. (Palmer, 2020)

  It is represented by letter d in an ERD.

- Overlapping subtypes

  These subtypes contain non-unique subsets of the supertype entity set. In overlapping subtypes, each entity instance of the supertype may appear in more than one subtype. (Palmer, 2020)

  It is represented by letter ‘o’ in an ERD.

- Completeness constraint in sub-type

"The completeness constraint specifies whether each entity supertype occurrence must also be a member of at least one subtype. The completeness constraint can be partial or total.
Partial completeness (symbolized by a circle over a single line) means that not every supertype occurrence is a member of a subtype; that is, there may be some supertype occurrences that are not members of any subtype. On the other hand, Total completeness (symbolized by a circle over a double line) means that every supertype occurrence must be a member of at least one subtype." (My Reading Room, 2016)

The different types of attributes are:

- Simple Attribute

  The attributes that have atomic values and cannot be divided further are called simple attributes. For example, a student's gender is an atomic value of a single word.

- Composite Attribute

  Those attributes formed by the combination of two or more attributes are called composite attributes. For example, a student's full name may have first_name and last_name.

Key is an attribute or collection of attributes that uniquely identifies an entity among entity set. (Tutorialspoint, 2020)

For example, the roll_number of a student uniquely identifies him/her among students.

The different types of keys are:

- Super Key – A set of attributes (single or multiple attributes) that collectively identifies an entity in an entity set. (Tutorialspoint, 2020)
- Candidate Key – A minimal key is called a candidate key. An entity set may have more than one candidate key. (Tutorialspoint, 2020)
- Primary key – A primary key is one of the candidate keys that uniquely identifies the entity set. (Tutorialspoint, 2020)
- Foreign key – A foreign key is a column or a combination of columns whose values match a Primary key in a different table. It is used to link two tables together. (Tutorialspoint, 2020)

Entities	Attributes
Person	Person_ID , Module_ID (FK), Address_ID (FK) , First_Name, Last_Name, Phone_Number, Email_Address, Person_Type
Student	Person_ID(PK, FK), Gender, DOB, Total_Marks, Student_Level, Enrolled_Date
Instructor	Person_ID (PK, FK), Role, Salary, Appointed_Date
Specification	Specification_ID (PK), Specification_Name, Course_ID , Course_Name , Course_Fee
Module	Module_ID (PK), Specification_ID (FK), Module_Name, Class_Name, Module_Level
Address	Address_ID (PK), House_Number, Street, Province, City, Country, Landline_Number, Fax_Number, Address_Type

Table 1: Table showing the initial entity and attributes

1.5 Initial ER Diagram

ER Diagram also called Entity Relationship Diagram, is a diagram which shows the relationship between the entity sets stored in a database and helps in the clarification of the logical structure of databases. ER diagrams are created based on the three basic concepts: entities, attributes and relationships.

In ER Diagrams, entities are represented by different symbols such as rectangles, attributes are represented by ovals and represent relationships are represented by diamond symbol.

The ER Diagram include many specialized symbols which has their specialized meaning and differentiates itself from flowchart. The purpose of ER Diagram is to represent the entity framework infrastructure. (Guru99, 2020)

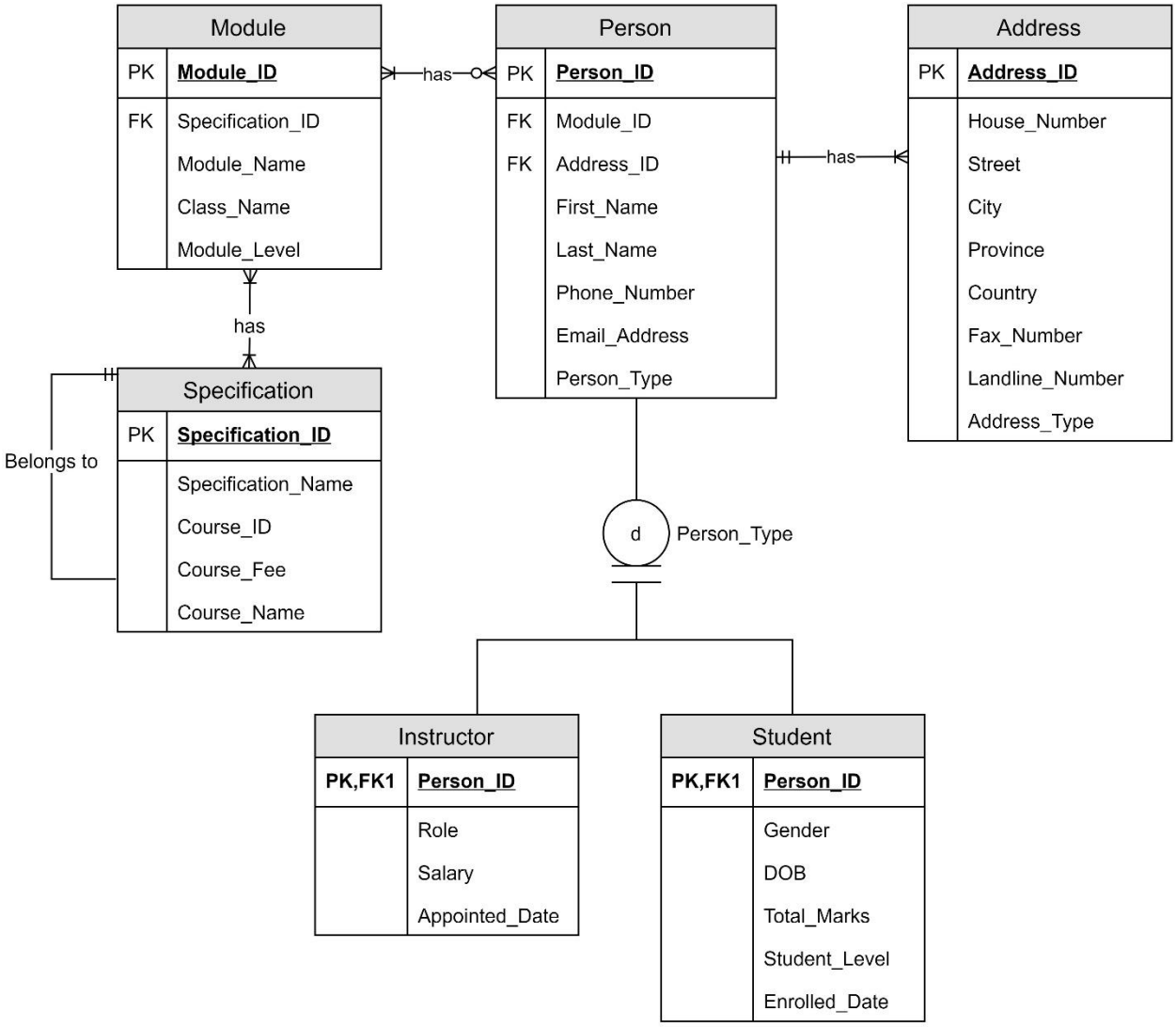


Figure 1: Initial ER Diagram

Every initial ER diagram tends to have a lot of issues. In the ERD above, the subtype discriminator is the attribute Person_Type and the two subtypes are disjoint having total completeness. The ER diagram present above is not an exception. Some of the major issues in the ER diagram above are listed below:

- There occurs many-to-many relationship between the entities such as specification and module, person and module. In a many-to-many relationship, one column has to store multiple values which is very hard for maintenance and querying. (Brumm, 2017)

Due to the issues in the ER Diagram, the data integrity is negatively affected. Hence, to solve the issues and to decrease data redundancy, Normalization needs to be implemented.

2. Normalization

2.1 Assumptions

- Student can be enrolled in only one course and select only one specification from the course. Similarly, each instructor can be associated with only one course and specification.
- Each person has at least one address having address type as mailing address.
- Each student and instructor may have multiple modules they are enrolled in and required to teach respectively.
- The role of the instructor may be module leader, course leader or instructor.
- Each person may or may not provide a fax number and a landline number for his/her address.
- Each module is in the same class only. But in a class, many modules may be taught.
- Each course should have only one course leader.
- The total marks of all the modules obtained by the student is stored.
- Each instructor can be associated to a single course. Whereas, a single course may have multiple instructors.
- Each specification has multiple modules. And each module may or may not be in different modules.
- The fee the student has to pay is dependent on the course and not its specifications.

2.2 Normalization

"Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships." (Guru99, 2020)

2.2.1 UNF (Un-Normalized Form)

Scenario for UNF:

- Each person should provide his/her first name, last name, phone number, email address.
- The person may either be instructor or student. They cannot be both.
- Each person should provide at least one address to be designated as mailing address which contains the house number, street, city, province, country, fax number, landline number. It is not necessary for the person to provide the landline number or the fax number.
- Each person (instructor or student) is associated with only one course and one specification of it.
- Each person with multiple modules of the specification.
- Each course, specification and module have their own ID uniquely identifying them.

Now, the next step is to identify and list all the attributes into a single entity.

Showing repeating groups:

Person(<u>Person_ID</u>, First_Name, Last_Name, Phone_Number, Email_Address, Person_Type , Course_ID, Course_Name,Course_Fee, {Address_ID, House_Number, Street, City, Province, Country, Fax_Number, Landline_Number, Address_Type}, Specification_ID, Specification_Name,{Module_ID, Module_Name, Class_Name, Module_Level , Course_ID, {Specification_ID}}, Gender, DOB, Total_Marks, Enrolled_Date, Student_Level, Role, Salary, Appointed_Date)

Here, all the attributes have been listed and the repeating groups have been separated. The attributes which are considered as repeating groups are kept inside curly brackets. Each person can have multiple addresses and be associated in multiple modules. Hence, all the attributes relating to address and module are kept in repeating groups respectively. Since, each module may be taught in multiple specification, the Specification_ID attribute is kept as repeating group inside the repeating group i.e., all the attributes relating to the module. And since each person is a student or an instructor, the attributes relating to student and instructor is not kept in the repeating group.

2.2.2 1NF

For a table to be in First Normal form (1NF), each column should have a single valued attribute. Hence, all the repeating groups form a separate entity. After the repeating groups and the non-repeating groups are form as separated entities, the primary key as well the foreign keys in each entity are identified.

In the 1NF below, the underlined attributes are the primary key and the attributes that has * symbol are the foreign keys. This step is easier as the repeating groups were already identified in the UNF.

Entities:

Person(Person_ID, First_Name, Last_Name, Phone_Number, Email_Address, Person_Type , Course_ID, Course_Name,Course_Fee, Specification_ID, Specification_Name, Gender, DOB, Total_Marks, Enrolled_Date, Student_Level, Role, Salary, Appointed_Date)

Address(Address_ID , Person_ID* , House_Number, Street, City, Province, Country, Fax_Number, Landline_Number, Address_Type)

Module(Module_ID, Person_ID*, Module_Name, Class_Name, Module_Level, Course_ID*)

Person_Module_Specification (Person_ID*, Module_ID* , Specification_ID*)

2.2.3 2NF

In second normal form, the partial dependencies are removed and a new entity is formed. The removing of partial dependencies which results minimizing data redundancy.

In Person Table,

Person(Person_ID, First_Name, Last_Name, Phone_Number, Email_Address, Person_Type , Course_ID, Course_Name,Course_Fee, Specification_ID, Specification_Name, Gender, DOB, Total_Marks, Enrolled_Date, Student_Level,Role, Salary, Appointed_Date)

In Address Table,

Address_ID → House_Number, Street, City, Province, Country, Fax_Number, Landline_Number, Address_Type

Address_ID, Person_ID →

Person_Address(Address_ID*, Person_ID*)

Address(Address_ID, House_Number, Street, City, Province, Country, Fax_Number, Landline_Number, Address_Type)

In Module Table,

Module_ID →Module_Name, Class_Name, Module_Level, Course_ID*

Module_ID, Person_ID →

Module(Module_ID , Module_Name, Class_Name, Module_Level, Course_ID*)

Person_Module (Person_ID* , Module_ID*)

Person_Module_Specification (Person_ID*, Module_ID* , Specification_ID*)

Hence, Total Tables formed:

Person(Person_ID, First_Name, Last_Name, Phone_Number, Email_Address, Person_Type , Course_ID, Course_Name,Course_Fee, Specification_ID, Specification_Name, Gender, DOB, Total_Marks, Enrolled_Date, Student_Level,Role, Salary, Appointed_Date)

Person_Address(Address_ID*, Person_ID*)

Address(Address_ID, House_Number, Street, City, Province, Country, Fax_Number, Landline_Number, Address_Type)

Module(Module_ID , Module_Name, Class_Name, Module_Level, Course_ID*)

Person_Module (Person_ID* , Module_ID*)

Person_Module_Specification (Person_ID*, Module_ID* , Specification_ID*)

In the steps above, the attributes dependent on Address_ID is separated to form a new entity. And a bridge entitity Person_Address is formed which is used to link Person and Address entity. The Person_Address contains the foreign keys Address_ID and Person_ID from the address and person entities respectively forming a composite key.

2.2.4 3NF

In the third normal form (3NF), the transitive dependency has to be eliminated.

“When an indirect relationship causes functional dependency, it is called Transitive Dependency. If P -> Q and Q -> R is true, then P-> R is a transitive dependency.” (Onsman, 2018)

In Person Table:

Person_ID →First_Name , Last_Name, Phone_Number, Email_Address, Person_Type , Course_ID, Course_Name, Course_Fee, Specification_ID, Specification_Name

In Student Table :

Person_ID → Gender, DOB, Total_Marks, Student_Level, Enrolled_Date

In Instructor Table:

Person_ID → Role, Salary, Appointed_Date

Also, In Person Table:

Person_ID → Course_ID → Course_Name, Course_Fee

Person_ID →Course_ID

Course_ID→Course_Name , Course_Fee

And,

Person_ID → Specification_ID→ Specification_Name, Course_ID*

Person_ID →Specification_ID

Specification_ID→ Specification_Name, Course_ID*

Person(Person_ID, First_Name , Last_Name, Phone_Number, Email_Address, Person_Type , Course_ID*, Specification_ID*)

Course(Course_ID , Course_Name, Course_Fee)

Specification(Specification_ID, Specification_Name, Course_ID*)

Hence, **Total Table formed** :

Person(Person_ID, First_Name , Last_Name, Phone_Number, Email_Address, Person_Type , Course_ID*, Specification_ID*)

Course(Course_ID , Course_Name, Course_Fee)

Specification(Specification_ID, Specification_Name, Course_ID*)

Student(Person_ID*, Gender, DOB, Total_Marks, Student_Level, Enrolled_Date)

Instructor(Person_ID*, Role, Salary, Appointed_Date)

Person_Address(Address_ID*, Person_ID*)

Address(Address_ID, House_Number, Street, City, Province, Country, Fax_Number, Landline_Number, Address_Type)

Person_Module (Person_ID* , Module_ID*)

Module(Module_ID , Module_Name, Class_Name, Module_Level, Course_ID*)

Person_Module_Specification (Person_ID*, Module_ID* , Specification_ID*)

We know that the Person entity is a Subertype and Student and Instructor entity are sub types.

Depending on the Person type, Student and Instructor Entity inherits the Primary key of the Person entity that is Person_ID and the inherited foreign key i.e. Person_ID act as primary key at their respective tables. Hence in the steps above, the attributes relating to student and instructor are separated from the Person entity and two new entities, Student and Instructor are formed.

In the Person entity, Specification_ID gives the value of all the attributes relating to specification such as specification name and the course id it belongs to. Hence, the attributes relating to the specification is separated from the person entity and a new entity called Specification is formed and the attribute, Specification_ID is stored in the Person entity as foreign key.

Similarly, in the Person entity, Course_ID gives the value of all the attributes such as the course name and the course fee. Hence, the attributes relating to the course is separated from the person entity and a new entity called Course is formed and the attribute, Course_ID is stored in the Person entity as foreign key.

After 3NF:

Among the tables formed, in the table Person_Module_Specification, Person_ID is redundant. This table's main purpose is to form relationship between Module and Specification. Since, the Person entity has already formed relation with the Specification and Module table, the Person_ID is not needed in the Person_Module_Specification table. Hence, the foreign key Person_ID is removed from the table and the remaining keys that are Module_ID and Specfication_ID form a composite primary key. And the table is renamed to Module_Specification as well.

Hence, the total tables formed:

Person(Person_ID, First_Name , Last_Name, Phone_Number, Email_Address, Person_Type , Course_ID*, Specification_ID*)

Course(Course_ID , Course_Name, Course_Fee)

Specification(Specification_ID, Specification_Name, Course_ID*)

Student(Person_ID*, Gender, DOB, Total_Marks, Student_Level, Enrolled_Date)

Instructor(Person_ID*, Role, Salary, Appointed_Date)

Person_Address(Address_ID*, Person_ID*)

Address(Address_ID, House_Number, Street, City, Province, Country, Fax_Number, Landline_Number, Address_Type)

Person_Module (Person_ID* , Module_ID*)

Module(Module_ID , Module_Name, Class_Name, Module_Level, Course_ID*)

Module_Specification (Module_ID* , Specification_ID*)

2.2.5 ER Diagram after carrying out normalization

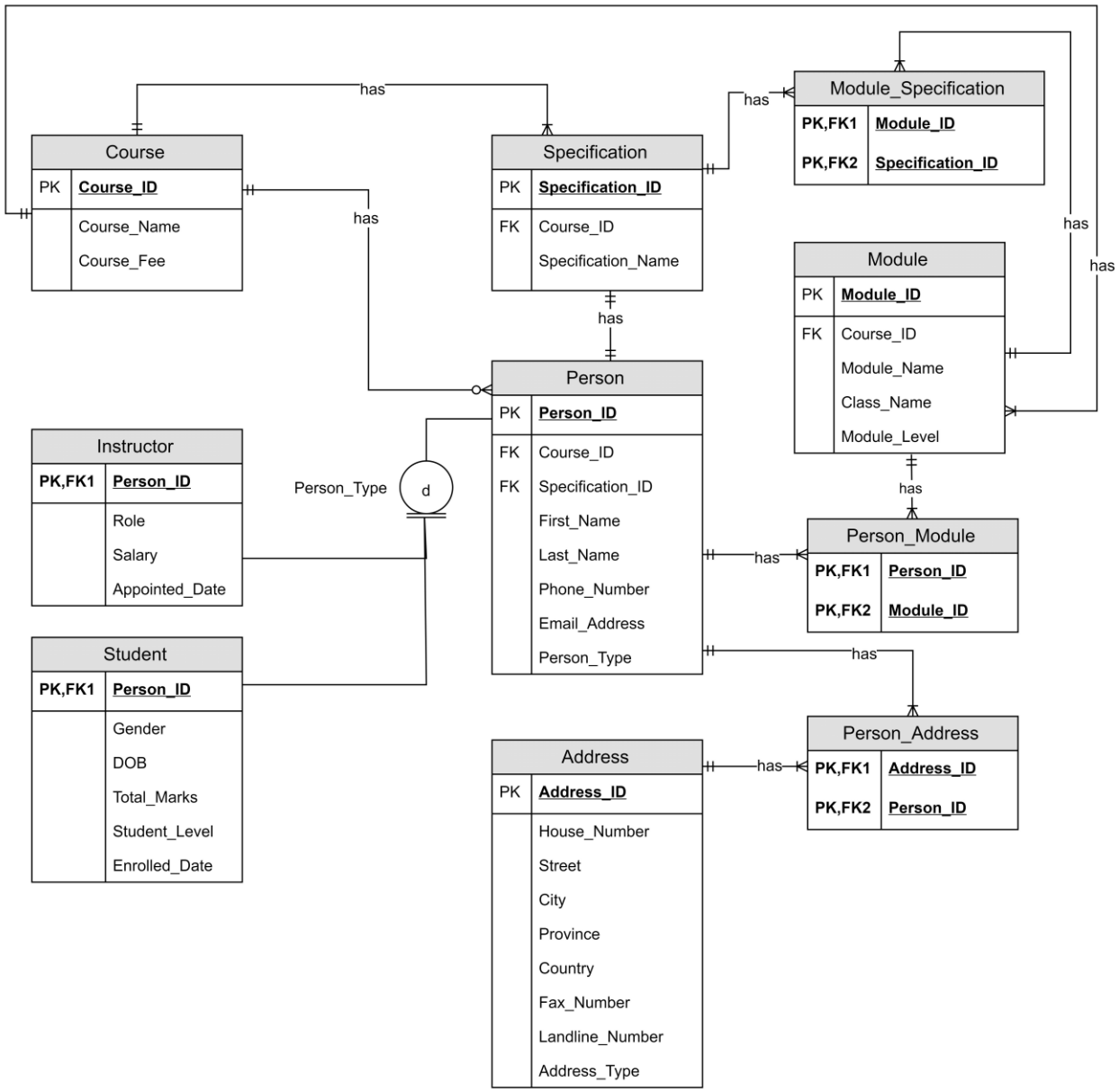


Figure 2: Final ERD

The above ERD is obtained after normalization. The problems faced in the initial ERD has been removed. The problem in the initial ERD that is many to many relationships has been removed with the implementation of bridge entities. In the initial ERD, the Person and the module entity has many-to-many relationships as each Person can be associated with many modules and each module can be associated with many persons. To solve the problem, a bridge entity i.e.

Person_Module has been implemented through normalisation where the foreign keys Person_ID and Module_ID forms a composite primary key in which the combination of two keys uniquely identifies each record. Hence, it makes it easier and less problematic to retrieve the data and establish relation between the Person and Module entity.

Similarly, though the normalization, the bridge entities such Person_Address and Module_Specification has also been implemented where the composite keys are Person_ID, Address_ID and Module_ID , Specification_ID respectively.

Similarly, the course and the specification has been separated into individual entities. This has reduced the data redundancy as, at each entry of a new specification, only the Course_ID has to be entered and not the remaining attributes relating to course every time. And the separation of Course into its own entity has made it easier to make changes in the course details as the changes can be made in a single record instead of multiple records of the same course.

3. Implementation

3.1 Creating User

Before the creation of the tables, a new user is created. The CREATE_USER command has been used to create a new MYSQL account that is DatabaseCW which can be accessed through the password i.e. CW. And GRANT command has been used to make the DatabaseCW user a database administer through which they are given basic database administration permission such as CREATE, DROP, DELETE, INSERT , SELECT , etc.

The query used for creating use is given below:

CREATE USER PratikCW identified by CW;

GRANT dba TO PratikCW;

```
SQL> CONNECT System;
Enter password:
Connected.
SQL> CREATE USER PratikCW IDENTIFIED BY CW;

User created.

SQL> GRANT DBA TO PratikCW;

Grant succeeded.
```

Figure 3 Creating User

3.2 Creating of Tables

For the creation of tables , CREATE command is used. It falls under the Data Definition Language (DDL). The DDL consists of the SQL commands that is used for the purpose of defining the database schema.

3.2.1 Course Table

```
SQL> CREATE TABLE Course
  2  (Course_ID VARCHAR2(5) PRIMARY KEY,
  3  Course_Name VARCHAR2(20) NOT NULL UNIQUE,
  4  Course_Fee NUMBER(9,2) NOT NULL);

Table created.

SQL> DESCRIBE Course;
 Name                                      Null?    Type
 ----------------------------------------- -------- -------------
 COURSE_ID                                 NOT NULL VARCHAR2(5)
 COURSE_NAME                               NOT NULL VARCHAR2(20)
 COURSE_FEE                                NOT NULL NUMBER(9,2)
```

Figure 4: Creating the Course Table

The course table consists of data relating to the courses provided by the college. In the course table, Course_ID attribute is the primary key which has Varchar datatype. The remaining attributes are Course_Name and Course_Fee which has data type Varchar and Number data type. It is defined that all the attributes should not be null and the course name should be unique.

The query used to create the course table is listed below:

CREATE TABLE Course

(Course_ID VARCHAR2(5) PRIMARY KEY,

Course_Name VARCHAR2(20) NOT NULL UNIQUE,

Course_Fee NUMBER(9,2) NOT NULL);

3.2.2 Specification Table

```
SQL> CREATE TABLE Specification
  2  (Specification_ID VARCHAR2(5) PRIMARY KEY,
  3  Specification_Name VARCHAR2(20) NOT NULL UNIQUE,
  4  Course_ID VARCHAR2(5)  NOT NULL ,
  5  FOREIGN KEY (Course_ID) REFERENCES Course(Course_ID));

Table created.

SQL> DESCRIBE Specification;
 Name                                      Null?    Type
 ----------------------------------------- -------- -------------
 SPECIFICATION_ID                          NOT NULL VARCHAR2(5)
 SPECIFICATION_NAME                        NOT NULL VARCHAR2(20)
 COURSE_ID                                 NOT NULL VARCHAR2(5)
```

Figure 5: Creating the Specification Table

The specification table consists of data relating to all the specifications provided by the college. In the specification table, Specification_ID attribute is the primary key which has Varchar datatype. In the table, Course_ID is the foreign key that references the Course_ID from the Course table which has Varchar datatype. The remaining attribute is Specification_Name which has Varchar data type. It is defined that all the attributes should not be null.

The query used to create the specification table is listed below:

CREATE TABLE Specification
(Specification_ID VARCHAR2(5) PRIMARY KEY,
Specification_Name VARCHAR2(20) NOT NULL UNIQUE,
Course_ID VARCHAR2(5) NOT NULL ,
FOREIGN KEY (Course_ID) REFERENCES Course(Course_ID));

3.2.3 Module Table

```
SQL> CREATE TABLE Module
  2  (Module_ID VARCHAR2(5) PRIMARY KEY,
  3  Course_ID VARCHAR2(5) NOT NULL,
  4  Module_Name VARCHAR2(50) NOT NULL,
  5  Class_Name VARCHAR2(15)  NOT NULL,
  6  Module_Level VARCHAR2(10)  NOT NULL,
  7  FOREIGN KEY (Course_ID) REFERENCES Course(Course_ID));

Table created.

SQL> DESCRIBE Module;
 Name                                      Null?    Type
 ----------------------------------------- -------- --------------
 MODULE_ID                                 NOT NULL VARCHAR2(5)
 COURSE_ID                                 NOT NULL VARCHAR2(5)
 MODULE_NAME                               NOT NULL VARCHAR2(50)
 CLASS_NAME                                NOT NULL VARCHAR2(15)
 MODULE_LEVEL                              NOT NULL VARCHAR2(10)
```

Figure 6: Creating the Module Table

The module table consists of data relating to all the modules of different specifications provided by the college. In the module table, Module_ID attribute is the primary key which has Varchar datatype. In the table, Course_ID is the foreign key that references the Course_ID from the Course table which has Varchar datatype. The remaining attributes are Module_Name, Class_Name and Module_Level which all have Varchar data type. It is defined that all the attributes should not be null.

The query used to create the module table is listed below:

```
CREATE TABLE Module
(Module_ID VARCHAR2(5) PRIMARY KEY,
Course_ID VARCHAR2(5) NOT NULL,
Module_Name VARCHAR2(50) NOT NULL,
Class_Name VARCHAR2(15)  NOT NULL,
Module_Level VARCHAR2(10)  NOT NULL,
FOREIGN KEY (Course_ID) REFERENCES Course(Course_ID));
```

3.2.4 Person Table

```
SQL> CREATE TABLE Person
  2  (Person_ID VARCHAR2(5) PRIMARY KEY,
  3  Course_ID VARCHAR2(5) NOT NULL,
  4  Specification_ID VARCHAR2(5) NOT NULL,
  5  First_Name VARCHAR2(15) NOT NULL,
  6  Last_Name VARCHAR2(15)  NOT NULL,
  7  Phone_Number VARCHAR2(15)  NOT NULL,
  8  Email_Address VARCHAR2(35)  NOT NULL,
  9  Person_Type VARCHAR2(10)  NOT NULL,
 10  FOREIGN KEY (Course_ID) REFERENCES Course(Course_ID),
 11  FOREIGN KEY (Specification_ID) REFERENCES Specification(Specification_ID));

Table created.

SQL> DESCRIBE Person;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PERSON_ID                                 NOT NULL VARCHAR2(5)
 COURSE_ID                                 NOT NULL VARCHAR2(5)
 SPECIFICATION_ID                          NOT NULL VARCHAR2(5)
 FIRST_NAME                                NOT NULL VARCHAR2(15)
 LAST_NAME                                 NOT NULL VARCHAR2(15)
 PHONE_NUMBER                              NOT NULL VARCHAR2(15)
 EMAIL_ADDRESS                             NOT NULL VARCHAR2(35)
 PERSON_TYPE                               NOT NULL VARCHAR2(10)
```

Figure 7: Creating the Person table

The person table consists of data relating to all the individuals associated to the college. In the Person table, Person_ID attribute is the primary key which has Varchar datatype. In the table, Course_ID and Specification_ID are the foreign keys that references the Course_ID and Specification_ID from the Course table and Specification table respectively which both has Varchar datatype. The remaining attributes are First_Name, Last_Name, Phone_Number. Email_Address and Person_Type which all have Varchar data type. It is defined that all the attributes should not be null.

The query used to create the Person table is listed below:

CREATE TABLE Person

(Person_ID VARCHAR2(5) PRIMARY KEY,

Course_ID VARCHAR2(5) NOT NULL,

Specification_ID VARCHAR2(5) NOT NULL,

First_Name VARCHAR2(15) NOT NULL,

Last_Name VARCHAR2(15) NOT NULL,

Phone_Number VARCHAR2(15) NOT NULL,

Email_Address VARCHAR2(35) NOT NULL,

Person_Type VARCHAR2(10) NOT NULL,

FOREIGN KEY (Course_ID) REFERENCES Course(Course_ID),

FOREIGN KEY (Specification_ID) REFERENCES Specification(Specification_ID));

3.2.5 Instructor Table

```
SQL> CREATE TABLE Instructor
  2  (Person_ID VARCHAR2(5) PRIMARY KEY,
  3  Role VARCHAR2(30) NOT NULL,
  4  Salary Number(9,2) NOT NULL,
  5  Appointed_Date DATE  NOT NULL,
  6  FOREIGN KEY (Person_ID) REFERENCES Person(Person_ID));

Table created.

SQL> DESCRIBE Instructor;
 Name                                      Null?    Type
 ----------------------------------------- -------- -------------
 PERSON_ID                                 NOT NULL VARCHAR2(5)
 ROLE                                      NOT NULL VARCHAR2(30)
 SALARY                                    NOT NULL NUMBER(9,2)
 APPOINTED_DATE                            NOT NULL DATE
```

Figure 8: Creating the Instructor Table

The instructor table consists of data of the instructors associated with the college. In the Instructor table, Person_ID attribute is the primary key as well as the foreign key which references Person_ID from the person table which has Varchar datatype. The remaining attributes are role, salary and appointed_date which has varchar, number and date data type respectively. It is defined that all the attributes should not be null.

The query used to create the Instructor table is listed below:

CREATE TABLE Instructor

(Person_ID VARCHAR2(5) PRIMARY KEY,

Role VARCHAR2(30) NOT NULL,

Salary Number(9,2) NOT NULL,

Appointed_Date DATE NOT NULL,

FOREIGN KEY (Person_ID) REFERENCES Person(Person_ID));

3.2.6 Student Table

```
SQL> CREATE TABLE Student
  2  (Person_ID VARCHAR2(5) PRIMARY KEY,
  3  Gender VARCHAR2(20) NOT NULL,
  4  DOB DATE NOT NULL,
  5  Total_Marks Number(7,2) NOT NULL,
  6  Student_Level VARCHAR2(10) NOT NULL,
  7  Enrolled_Date DATE  NOT NULL,
  8  FOREIGN KEY (Person_ID) REFERENCES Person(Person_ID));

Table created.

SQL> DESCRIBE Student;
 Name                                      Null?    Type
 ----------------------------------------- -------- -------------
 PERSON_ID                                 NOT NULL VARCHAR2(5)
 GENDER                                    NOT NULL VARCHAR2(20)
 DOB                                       NOT NULL DATE
 TOTAL_MARKS                               NOT NULL NUMBER(7,2)
 STUDENT_LEVEL                             NOT NULL VARCHAR2(10)
 ENROLLED_DATE                             NOT NULL DATE
```

Figure 9: Creating the Student table

The Student table consists of data of the students associated with the college. In the Person table, Person_ID attribute is the primary key as well as the foreign key which references Person_ID from the person table which has Varchar datatype. The remaining attributes are gender, DOB, total_marks, student_level and enrolled_date which has varchar, date, number , varchar and date data type respectively. It is defined that all the attributes should not be null.

The query used to create the Student table is listed below:

CREATE TABLE Student

(Person_ID VARCHAR2(5) PRIMARY KEY,

Gender VARCHAR2(20) NOT NULL,

DOB DATE NOT NULL,

Total_Marks Number(7,2) NOT NULL,

Student_Level VARCHAR2(10) NOT NULL,

Enrolled_Date DATE NOT NULL,

FOREIGN KEY (Person_ID) REFERENCES Person(Person_ID));

3.2.7 Address Table

```
SQL> CREATE TABLE Address
  2  (Address_ID VARCHAR2(5) PRIMARY KEY,
  3  House_Number NUMBER(5) NOT NULL,
  4  Street VARCHAR2(20) NOT NULL,
  5  City VARCHAR2(15) NOT NULL,
  6  Province VARCHAR2(15)  NOT NULL,
  7  Country VARCHAR2(15)  NOT NULL,
  8  Fax_Number VARCHAR2(15),
  9  Landline_Number VARCHAR2(18),
 10  Address_Type VARCHAR2(30)  NOT NULL);

Table created.

SQL> DESCRIBE Address;
 Name                                      Null?    Type
 ----------------------------------------- -------- -------------
 ADDRESS_ID                                NOT NULL VARCHAR2(5)
 HOUSE_NUMBER                              NOT NULL NUMBER(5)
 STREET                                    NOT NULL VARCHAR2(20)
 CITY                                      NOT NULL VARCHAR2(15)
 PROVINCE                                  NOT NULL VARCHAR2(15)
 COUNTRY                                   NOT NULL VARCHAR2(15)
 FAX_NUMBER                                         VARCHAR2(15)
 LANDLINE_NUMBER                                    VARCHAR2(18)
 ADDRESS_TYPE                              NOT NULL VARCHAR2(30)
```

Figure 10: Creating the Address table

The adress table consists of data relating to the addresses of individuals associated with the college. In the Address table, Address_ID attribute is the primary key which has Varchar datatype. In the table, the remaining attributes are house_number, street, city, province, country, fax_number, landline_number and address_type which has Varchar or Number data type. It is defined that all the attributes except the fax_number and the landline_number attributes should not be null.

The query used to create the address table is listed below:

CREATE TABLE Address

(Address_ID VARCHAR2(5) PRIMARY KEY,

House_Number NUMBER(5) NOT NULL,

Street VARCHAR2(20) NOT NULL,

City VARCHAR2(15) NOT NULL,

Province VARCHAR2(15) NOT NULL,

Country VARCHAR2(15) NOT NULL,

Fax_Number VARCHAR2(15),

Landline_Number VARCHAR2(18),

Address_Type VARCHAR2(30) NOT NULL);

3.2.8 Person_Module Table

```
SQL> CREATE TABLE Person_Module
  2  (Person_ID VARCHAR2(5) NOT NULL,
  3  Module_ID VARCHAR2(5) NOT NULL,
  4  PRIMARY KEY(Person_ID, Module_ID),
  5  FOREIGN KEY (Person_ID) REFERENCES Person(Person_ID),
  6  FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID));

Table created.

SQL> DESCRIBE Person_Module;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 PERSON_ID                                 NOT NULL VARCHAR2(5)
 MODULE_ID                                 NOT NULL VARCHAR2(5)
```

Figure 11: Creating the Person-Module Table

The Person_Module is a bridge table which consists of composite primary key. The Person_ID and Module_ID are foreign keys which references the Person_ID and Module_ID from the Person and Module table respectively. The combination of the Person_ID and the Module_ID forms a composite primary key which uniquely identifies each record in the table. Both the attributes have Varchar datatype.

The query used to create the Person_Module table is listed below:

CREATE TABLE Person_Module

(Person_ID VARCHAR2(5) NOT NULL,

Module_ID VARCHAR2(5) NOT NULL,

PRIMARY KEY(Person_ID, Module_ID),

FOREIGN KEY (Person_ID) REFERENCES Person(Person_ID),

FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID));

3.2.9 Person_Address Table

```
SQL> CREATE TABLE Person_Address
  2  (Person_ID VARCHAR2(5) NOT NULL,
  3  Address_ID VARCHAR2(5) NOT NULL,
  4  PRIMARY KEY(Person_ID, Address_ID),
  5  FOREIGN KEY (Person_ID) REFERENCES Person(Person_ID),
  6  FOREIGN KEY (Address_ID) REFERENCES Address(Address_ID));

Table created.

SQL> DESCRIBE Person_Address;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
 PERSON_ID                                 NOT NULL VARCHAR2(5)
 ADDRESS_ID                                NOT NULL VARCHAR2(5)
```

Figure 12: Creating the Person_Address Table

The Person_Address is a bridge table which consists of composite primary key. The Person_ID and Address_ID are foreign keys which references the Person_ID and Address_ID from the Person and Address table respectively. The combination of the Person_ID and the Address_ID forms a composite primary key which uniquely identifies each record in the table. Both the attributes have Varchar datatype.

The query used to create the Person_Address table is listed below:

CREATE TABLE Person_Address
(Person_ID VARCHAR2(5) NOT NULL,
Address_ID VARCHAR2(5) NOT NULL,
PRIMARY KEY(Person_ID, Address_ID),
FOREIGN KEY (Person_ID) REFERENCES Person(Person_ID),
FOREIGN KEY (Address_ID) REFERENCES Address(Address_ID));

3.2.10 Module_Specification Table

```
SQL> CREATE TABLE Module_Specification
  2  (Module_ID VARCHAR2(5) NOT NULL,
  3  Specification_ID VARCHAR2(5) NOT NULL,
  4  PRIMARY KEY(Module_ID, Specification_ID),
  5  FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID),
  6  FOREIGN KEY (Specification_ID) REFERENCES Specification(Specification_ID));

Table created.

SQL> DESCRIBE Module_Specification;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MODULE_ID                                 NOT NULL VARCHAR2(5)
 SPECIFICATION_ID                          NOT NULL VARCHAR2(5)
```

Figure 13: Creating the Module_Specification Table

The Module_Specification is a bridge table which consists of composite primary key. The Module_ID and Specification_ID are foreign keys which references the Module_ID and Specfication_ID from the Module and Specification table respectively. The combination of the Module_ID and the Specification_ID forms a composite primary key which uniquely identifies each record in the table. Both the attributes have Varchar datatype.

The query used to create the Module_Specification table is listed below:

CREATE TABLE Module_Specification
(Module_ID VARCHAR2(5) NOT NULL,
Specification_ID VARCHAR2(5) NOT NULL,
PRIMARY KEY(Module_ID, Specification_ID),
FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID),
FOREIGN KEY (Specification_ID) REFERENCES Specification(Specification_ID));

3.3 Populating DB Tables

To insert data into the tables, the INSERT command is used which is a Data Manipulation Language (DML). DML consists of commands that deals with the manipulation of the data present in the database. (GeeksforGeeks, 2019)

"The COMMIT command is the transactional command used to save changes invoked by a transaction to the database. The COMMIT command saves all the transactions to the database since the last COMMIT or ROLLBACK command.

The syntax for the COMMIT command is as follows.

COMMIT;" (Tutorialspoint, 2020)

After the creation of the required tables of the database, the data is inserted into each table.

3.3.1 Inserting values into Course table

Since, the College provides only two courses, there are only two entries in the course table. The query for inserting the course table is:

```
INSERT ALL
INTO Course VALUES ('C1','BIT', 1096500)
INTO Course VALUES ('C2','BBA',1033500)
SELECT * FROM dual;
```

```
SQL> INSERT ALL
  2  INTO Course VALUES ('C1','BIT', 1096500)
  3  INTO Course VALUES ('C2','BBA',1033500)
  4  SELECT * FROM dual;

2 rows created.
```

Figure 14: Course Insertion Statement

```
SQL> SELECT * FROM Course;

COURS COURSE_NAME          COURSE_FEE
----- -------------------- ----------
C1    BIT                     1096500
C2    BBA                     1033500
```

Figure 15: Course Selection Statement

3.3.2 Inserting values into Specification table

```
INSERT ALL
INTO Specification VALUES ('S1', 'Computing', 'C1')
INTO Specification VALUES ('S2', 'Multimedia' ,'C1')
INTO Specification VALUES ('S3', 'Networking', 'C1')
INTO Specification VALUES ('S4', 'Intl Business', 'C2')
```

INTO Specification VALUES ('S5', 'Finance', 'C2')

INTO Specification VALUES ('S6', 'Marketing', 'C2')

SELECT * FROM dual;

```
SQL> INSERT ALL
  2  INTO Specification VALUES ('S1', 'Computing', 'C1')
  3  INTO Specification VALUES ('S2', 'Multimedia' ,'C1')
  4  INTO Specification VALUES ('S3', 'Networking', 'C1')
  5  INTO Specification VALUES ('S4', 'Intl Business', 'C2')
  6  INTO Specification VALUES ('S5', 'Finance', 'C2')
  7  INTO Specification VALUES ('S6', 'Marketing', 'C2')
  8  SELECT * FROM dual;

6 rows created.
```

Figure 16: Specification Insertion Statement

```
SQL> SELECT * FROM Specification;

SPECI SPECIFICATION_NAME   COURS
----- -------------------- -----
S1    Computing            C1
S2    Multimedia           C1
S3    Networking           C1
S4    Intl Business        C2
S5    Finance              C2
S6    Marketing            C2

6 rows selected.
```

Figure 17: Specification Selection Statement

3.3.3 Inserting values into Module table

INSERT ALL

INTO Module VALUES ('M1','C1','Digital Design and Image Making', 'Lumbini room','Second')

INTO Module VALUES ('M2','C1','Programming','Patan room','First')

INTO Module VALUES ('M3','C1','3D Modelling','Everest room','Third')

INTO Module VALUES ('M4','C1','Emerging Programming Platforms and Technologies','Patan room','Second')

INTO Module VALUES ('M5','C1','Databases','Lukla room','Third')

INTO Module VALUES ('M6','C1','Information Systems','Bhaktapur room','First')

INTO Module VALUES ('M7','C1','Networks and Operating Systems','Patan room','Second')

INTO Module VALUES ('M8','C1','Ethical Hacking','Lukla room','Third')

INTO Module VALUES ('M9','C1','Communications Engineering','Bhaktapur room','First')

INTO Module VALUES ('M10','C2','Understanding Business Information','Patan room','Second')

INTO Module VALUES ('M11','C2','Business Research and Decision-Making','Everest room','Third')

INTO Module VALUES ('M12','C2','Business Without Borders','Lukla room','Fourth')

INTO Module VALUES ('M13','C2','International Finance and Trade','Bhaktapur room','Third')

INTO Module VALUES ('M14','C2','Services Marketing','Patan room','Third')

INTO Module VALUES ('M15','C2','Digital Marketing','Lumbini room','Fourth')

INTO Module VALUES ('M16','C2','Brand Management','Lukla room','Fourth')

INTO Module VALUES ('M17','C2','Consumer Public Relations','Bhaktapur room','Fourth')

INTO Module VALUES ('M18','C2','Leading Innovation and Entrepreneurship','Lumbini room','Second')

SELECT * FROM dual;

```
SQL> INSERT ALL
  2  INTO Module VALUES ('M1','C1','Digital Design and Image Making', 'Lumbini room','Second')
  3  INTO Module VALUES ('M2','C1','Programming','Patan room','First')
  4  INTO Module VALUES ('M3','C1','3D Modelling','Everest room','Third')
  5  INTO Module VALUES ('M4','C1','Emerging Programming Platforms and Technologies','Patan room','Second')
  6  INTO Module VALUES ('M5','C1','Databases','Lukla room','Third')
  7  INTO Module VALUES ('M6','C1','Information Systems','Bhaktapur room','First')
  8  INTO Module VALUES ('M7','C1','Networks and Operating Systems','Patan room','Second')
  9  INTO Module VALUES ('M8','C1','Ethical Hacking','Lukla room','Third')
 10  INTO Module VALUES ('M9','C1','Communications Engineering','Bhaktapur room','First')
 11  INTO Module VALUES ('M10','C2','Understanding Business Information','Patan room','Second')
 12  INTO Module VALUES ('M11','C2','Business Research and Decision-Making','Everest room','Third')
 13  INTO Module VALUES ('M12','C2','Business Without Borders','Lukla room','Fourth')
 14  INTO Module VALUES ('M13','C2','International Finance and Trade','Bhaktapur room','Third')
 15  INTO Module VALUES ('M14','C2','Services Marketing','Patan room','Third')
 16  INTO Module VALUES ('M15','C2','Digital Marketing','Lumbini room','Fourth')
 17  INTO Module VALUES ('M16','C2','Brand Management','Lukla room','Fourth')
 18  INTO Module VALUES ('M17','C2','Consumer Public Relations','Bhaktapur room','Fourth')
 19  INTO Module VALUES ('M18','C2','Leading Innovation and Entrepreneurship','Lumbini room','Second')
 20  INTO Module VALUES ('M19','C1','Drawing and Character Design','Patan room','First')
 21  INTO Module VALUES ('M20','C1','Project','Patan room','Third')
 22  INTO Module VALUES ('M21','C1','Moving Image and VFX','Patan room','Second')
 23  SELECT * FROM dual;

21 rows created.
```

Figure 18: Module Insertion Statement

```
SQL> SELECT * FROM Module;

MODUL COURS MODULE_NAME                                        CLASS_NAME      MODULE_LEV
----- ----- -------------------------------------------------- --------------- ----------
M1    C1    Digital Design and Image Making                    Lumbini room    Second
M2    C1    Programming                                        Patan room      First
M3    C1    3D Modelling                                       Everest room    Third
M4    C1    Emerging Programming Platforms and Technologies    Patan room      Second
M5    C1    Databases                                          Lukla room      Third
M6    C1    Information Systems                                Bhaktapur room  First
M7    C1    Networks and Operating Systems                     Patan room      Second
M8    C1    Ethical Hacking                                    Lukla room      Third
M9    C1    Communications Engineering                         Bhaktapur room  First
M10   C2    Understanding Business Information                 Patan room      Second
M11   C2    Business Research and Decision-Making              Everest room    Third
M12   C2    Business Without Borders                            Lukla room      Fourth
M13   C2    International Finance and Trade                     Bhaktapur room  Third
M14   C2    Services Marketing                                 Patan room      Third
M15   C2    Digital Marketing                                  Lumbini room    Fourth
M16   C2    Brand Management                                   Lukla room      Fourth
M17   C2    Consumer Public Relations                          Bhaktapur room  Fourth
M18   C2    Leading Innovation and Entrepreneurship            Lumbini room    Second
M19   C1    Drawing and Character Design                       Patan room      First
M20   C1    Project                                            Patan room      Third
M21   C1    Moving Image and VFX                               Patan room      Second

21 rows selected.
```

Figure 19: Module Selection Statement

3.3.4 Inserting values into Person table

INSERT ALL

INTO Person VALUES ('P1', 'C1','S1','Badrinath','Moktan','9855541708','badrinathmoktan@gmail.com','Student')

INTO Person VALUES ('P2','C1','S2','Kishor','Shilakar','9855539888','kishorshilakar@gmail.com','Student')

INTO Person VALUES ('P3','C2','S5','Arpana','Rawat','9855568466','arpanarawat@gmail.com','Student')

INTO Person VALUES ('P4','C2','S4','Jaya','Dhungana','9855540557','jayadhungana@gmail.com','Student')

INTO Person VALUES ('P5','C1','S3','Naina','Pun','9855536427','nainapun@gmail.com','Student')

INTO Person VALUES ('P6','C2','S5','Karuna','Dhamala','9855554739','karunadhamala@gmail.com','Student')

INTO Person VALUES ('P7','C1','S2','Raju','Sinha','9855578467','rajusinha@gmail.com','Student')

INTO Person VALUES ('P8','C2','S4','Paras','Koirala','9855517927','paraskoirala@gmail.com','Instructor')

INTO Person VALUES ('P9','C1','S3','Prabhu','Aryal','9855548829','prabhuaryal@gmail.com','Instructor')

INTO Person VALUES ('P10','C1','S1','Sanjita','Mali','9855502333','sanjitamali@gmail.com','Instructor')

INTO Person VALUES ('P11','C2','S6','Mahavir', 'Thapaliya','9855577520','mahavirthapaliya@gmail.com','Instructor')

INTO Person VALUES ('P12','C1','S3','Manu','Choudhary','9855558609','manuchoudhary@gmail.com','Instructor')

INTO Person VALUES ('P13','C2','S4','Shanti','Gartaula','9855573753','shantigartaula@gmail.com','Instructor')

INTO Person VALUES ('P14','C2','S6','Kiran','Panday','9855524409','kiranpanday@gmail.com','Instructor')

SELECT * FROM dual;

```
SQL> INSERT ALL
  2  INTO  Person VALUES ('P1', 'C1','S1','Badrinath','Moktan','9855541708','badrinathmoktan@gmail.com','Student')
  3  INTO  Person VALUES ('P2','C1','S2','Kishor','Shilakar','9855539888','kishorshilakar@gmail.com','Student')
  4  INTO  Person VALUES ('P3','C2','S5','Arpana','Rawat','9855568466','arpanarawat@gmail.com','Student')
  5  INTO  Person VALUES ('P4','C2','S4','Jaya','Dhungana','9855540557','jayadhungana@gmail.com','Student')
  6  INTO  Person VALUES ('P5','C1','S3','Naina','Pun','9855536427','nainapun@gmail.com','Student')
  7  INTO  Person VALUES ('P6','C2','S5','Karuna','Dhamala','9855554739','karunadhamala@gmail.com','Student')
  8  INTO  Person VALUES ('P7','C1','S2','Raju','Sinha','9855578467','rajusinha@gmail.com','Student')
  9  INTO  Person VALUES ('P8','C2','S4','Siddhartha','Koirala','9855517927','siddharthakoirala@gmail.com','Instructor')
 10  INTO  Person VALUES ('P9','C1','S3','Saurya','Silwal','9855548829','sauryaaryal@gmail.com','Instructor')
 11  INTO  Person VALUES ('P10','C1','S1','Simba','Mali','9855502333','simbamali@gmail.com','Instructor')
 12  INTO  Person VALUES ('P11','C2','S6','Sanjita', 'Thapaliya','9855577520','sanjitathapaliya@gmail.com','Instructor')
 13  INTO  Person VALUES ('P12','C1','S3','Siva','Choudhary','9855558609','sivachoudhary@gmail.com','Instructor')
 14  INTO  Person VALUES ('P13','C2','S4','Soorya','Gartaula','9855573753','sooryagartaula@gmail.com','Instructor')
 15  INTO  Person VALUES ('P14','C2','S6','Kiran','Panday','9855524409','kiranpanday@gmail.com','Instructor')
 16  INTO Person VALUES ('P15','C1','S2','Aavash','Aryal','9855524499','aavash@gmail.com','Instructor')
 17  INTO Person VALUES ('P16','C1','S2','Birat','Shah','9855521109','birat@gmail.com','Instructor')
 18  INTO Person VALUES ('P17','C1','S2','Sitka','Candice','9855524411','sitka@gmail.com','Instructor')
 19  INTO  Person VALUES ('P18','C1','S3','Dandi','Dahal','9818536427','dandi@gmail.com','Student')
 20  INTO  Person VALUES ('P19','C1','S3','Pratik','Amatya','9870536427','pratik@gmail.com','Student')
 21  INTO  Person VALUES ('P20','C1','S3','Hari','Shrestha','9845536427','hari@gmail.com','Student')
 22  INTO  Person VALUES ('P21','C1','S3','Nirajan','Thapa','9846536427','nirajan@gmail.com','Student')
 23  INTO  Person VALUES ('P22','C1','S3','Pratima','Adhikari','9895536427','pratima@gmail.com','Student')
 24  INTO  Person VALUES ('P23','C1','S3','Divya','KC','9855999427','divya@gmail.com','Student')
 25  SELECT * FROM DUAL;

23 rows created.
```

Figure 20: Person Insertion Statement

```
SQL> SELECT * FROM Person;

PERSO COURS SPECI FIRST_NAME      LAST_NAME       PHONE_NUMBER    EMAIL_ADDRESS                       PERSON_TYP
----- ----- ----- --------------- --------------- --------------- ----------------------------------- ----------
P1    C1    S1    Badrinath       Moktan          9855541708      badrinathmoktan@gmail.com           Student
P2    C1    S2    Kishor          Shilakar        9855539888      kishorshilakar@gmail.com            Student
P3    C2    S5    Arpana          Rawat           9855568466      arpanarawat@gmail.com               Student
P4    C2    S4    Jaya            Dhungana        9855540557      jayadhungana@gmail.com              Student
P5    C1    S3    Naina           Pun             9855536427      nainapun@gmail.com                  Student
P6    C2    S5    Karuna          Dhamala         9855554739      karunadhamala@gmail.com             Student
P7    C1    S2    Raju            Sinha           9855578467      rajusinha@gmail.com                 Student
P8    C2    S4    Siddhartha      Koirala         9855517927      siddharthakoirala@gmail.com         Instructor
P9    C1    S3    Saurya          Silwal          9855548829      sauryaaryal@gmail.com               Instructor
P10   C1    S1    Simba           Mali            9855502333      simbamali@gmail.com                 Instructor
P11   C2    S6    Sanjita         Thapaliya       9855577520      sanjitathapaliya@gmail.com          Instructor
P12   C1    S3    Siva            Choudhary       9855558609      sivachoudhary@gmail.com             Instructor
P13   C2    S4    Soorya          Gartaula        9855573753      sooryagartaula@gmail.com            Instructor
P14   C2    S6    Kiran           Panday          9855524409      kiranpanday@gmail.com               Instructor
P15   C1    S2    Aavash          Aryal           9855524499      aavash@gmail.com                    Instructor
P16   C1    S2    Birat           Shah            9855521109      birat@gmail.com                     Instructor
P17   C1    S2    Sitka           Candice         9855524411      sitka@gmail.com                     Instructor
P18   C1    S3    Dandi           Dahal           9818536427      dandi@gmail.com                     Student
P19   C1    S3    Pratik          Amatya          9870536427      pratik@gmail.com                    Student
P20   C1    S3    Hari            Shrestha        9845536427      hari@gmail.com                      Student
P21   C1    S3    Nirajan         Thapa           9846536427      nirajan@gmail.com                   Student
P22   C1    S3    Pratima         Adhikari        9895536427      pratima@gmail.com                   Student
P23   C1    S3    Divya           KC              9855999427      divya@gmail.com                     Student

23 rows selected.
```

Figure 21: Person Selection Statement

3.3.5 Inserting values into Instructor table

```
INSERT ALL
INTO Instructor VALUES ('P8', 'Module Leader', 55000,'22-OCT-20')
INTO Instructor VALUES ('P9',  'Course Leader', 60000,'22-OCT-20')
INTO Instructor VALUES ('P10', 'Tutor', 40000,'22-OCT-19')
INTO Instructor VALUES ('P11', 'Module Leader' ,52000,'22-OCT-19')
INTO Instructor VALUES ('P12', 'Module Leader', 57000,'22-OCT-19')
INTO Instructor VALUES ('P13', 'Course Leader', 54000,'22-OCT-18')
INTO Instructor VALUES ('P14', 'Tutor',45000,'22-OCT-18')
SELECT * FROM dual;
```

```
SQL> INSERT ALL
  2  INTO Instructor VALUES ('P8', 'Module Leader', 55000,'22-OCT-20')
  3  INTO Instructor VALUES ('P9',  'Course Leader', 60000,'22-OCT-20')
  4  INTO Instructor VALUES ('P10', 'Tutor', 40000,'22-OCT-19')
  5  INTO Instructor VALUES ('P11', 'Module Leader' ,52000,'22-OCT-19')
  6  INTO Instructor VALUES ('P12', 'Module Leader', 57000,'22-OCT-19')
  7  INTO Instructor VALUES ('P13', 'Course Leader', 54000,'22-OCT-18')
  8  INTO Instructor VALUES ('P14', 'Module Leader',45000,'22-OCT-18')
  9  INTO Instructor VALUES ('P15','Module Leader','58000','12-DEC-2018')
 10  INTO Instructor VALUES ('P16','Module Leader','58000','10-JAN-2018')
 11  INTO Instructor VALUES ('P17','Module Leader','58000','22-MAR-2018')
 12  SELECT * FROM DUAl;

10 rows created.
```

Figure 22: Instructor Insertion Statement

```
SQL> SELECT * FROM Instructor;

PERSO ROLE                              SALARY APPOINTED
----- ------------------------------ ---------- ---------
P8    Module Leader                     55000 22-OCT-20
P9    Course Leader                     60000 22-OCT-20
P10   Tutor                             40000 22-OCT-19
P11   Module Leader                     52000 22-OCT-19
P12   Module Leader                     57000 22-OCT-19
P13   Course Leader                     54000 22-OCT-18
P14   Module Leader                     45000 22-OCT-18
P15   Module Leader                     58000 12-DEC-18
P16   Module Leader                     58000 10-JAN-18
P17   Module Leader                     58000 22-MAR-18

10 rows selected.
```

Figure 23: Instructor Selection Staement

3.3.6 Inserting values into Student table

INSERT ALL

INTO Student VALUES ('P1', 'Male', '22-OCT-01', 350,'First','22-OCT-20')

INTO Student VALUES ('P2', 'Male', '22-NOV-00', 330,'First','22-OCT-20')

INTO Student VALUES ('P3', 'Female', '22-JAN-00', 320,'Second','22-OCT-19')

INTO Student VALUES ('P4', 'Male', '22-FEB-01', 310,'Second','22-OCT-19')

INTO Student VALUES ('P5', 'Female', '22-SEP-01', 340,'Second','22-OCT-19')

INTO Student VALUES ('P6', 'Female', '22-JAN-00', 360,'Third','22-OCT-18')

INTO Student VALUES ('P7', 'Male', '22-SEP-01', 370,'Third','22-OCT-18')

SELECT * FROM dual;

```
SQL> INSERT ALL
  2  INTO Student VALUES ('P1', 'Male', '22-OCT-01', 350,'First','22-OCT-20')
  3  INTO Student VALUES ('P2', 'Male', '22-NOV-00', 330,'First','22-OCT-20')
  4  INTO Student VALUES ('P3', 'Female', '22-JAN-00', 320,'Second','22-OCT-19')
  5  INTO Student VALUES ('P4', 'Male', '22-FEB-01', 310,'Second','22-OCT-19')
  6  INTO Student VALUES ('P5', 'Female', '22-SEP-01', 340,'Second','22-OCT-19')
  7  INTO Student VALUES ('P6', 'Female', '22-JAN-00', 360,'Third','22-OCT-18')
  8  INTO Student VALUES ('P7', 'Male', '22-SEP-01', 370,'Third','22-OCT-18')
  9  INTO Student VALUES ('P18', 'Female', '21-SEP-00', 370,'Third','22-OCT-18')
 10  INTO Student VALUES ('P19', 'Male', '10-SEP-01', 380,'Third','22-OCT-18')
 11  INTO Student VALUES ('P20', 'Male', '11-SEP-01', 370,'Third','22-OCT-18')
 12  INTO Student VALUES ('P21', 'Male', '22-JAN-00', 375,'Third','22-OCT-18')
 13  INTO Student VALUES ('P22', 'Female', '21-SEP-01', 378,'Third','22-OCT-18')
 14  INTO Student VALUES ('P23', 'Female', '25-SEP-01', 371,'Third','22-OCT-18')
 15  SELECT * FROM dual;

13 rows created.
```

Figure 24: Student Insertion Statement

```
SQL> SELECT * FROM Student;

PERSO GENDER               DOB       TOTAL_MARKS STUDENT_LE ENROLLED_
----- -------------------- --------- ----------- ---------- ---------
P1    Male                 22-OCT-01         350 First      22-OCT-20
P2    Male                 22-NOV-00         330 First      22-OCT-20
P3    Female               22-JAN-00         320 Second     22-OCT-19
P4    Male                 22-FEB-01         310 Second     22-OCT-19
P5    Female               22-SEP-01         340 Second     22-OCT-19
P6    Female               22-JAN-00         360 Third      22-OCT-18
P7    Male                 22-SEP-01         370 Third      22-OCT-18
P18   Female               21-SEP-00         370 Third      22-OCT-18
P19   Male                 10-SEP-01         380 Third      22-OCT-18
P20   Male                 11-SEP-01         370 Third      22-OCT-18
P21   Male                 22-JAN-00         375 Third      22-OCT-18
P22   Female               21-SEP-01         378 Third      22-OCT-18
P23   Female               25-SEP-01         371 Third      22-OCT-18

13 rows selected.
```

Figure 25: Student Selection Statement

3.3.7 Inserting values into Address table

INSERT ALL

INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)

VALUES ('AD1','1204','Birat chowk','Biratnagar','Province-1','Nepal','712348749','4416278','Mailing Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Address_Type)
VALUES ('AD2','3405' , 'Panchytar', 'Nawalparasi','Province-2','Nepal','Temporary Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Landline_Number,Address_Type)
VALUES ('AD3','3453','New Road','Kathmandu','Province-4','Nepal','4183842','Mailing Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
VALUES ('AD4','5677','Baniyatar chowk','Kathmandu','Province-4','Nepal','712392749','4183842','Mailing Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Address_Type)
VALUES ('AD5','4356','Tangal','Kathmandu','Province-4','Nepal','796392749','Temporary Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Landline_Number,Address_Type)
VALUES ('AD6','5665','Jorpati','Kathmandu','Province-4','Nepal','4182942','Mailing Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
VALUES ('AD7','5674','Bhaktapur chowk','Bhaktapur','Province-2','Nepal','729392749','4193842','Mailing Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Address_Type)
VALUES ('AD8','6784','Prithivi chowk','Pokhara','Province-6','Nepal','Temporary Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Landline_Number,Address_Type)
VALUES ('AD9','4345','Swayambu','Kathmandu','Province-4','Nepal','41323842','Mailing Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Address_Type)

```
VALUES ('AD10','3459','Myagdi chowk','Kathmandu','Province-4','Nepal','712392778','Mailing Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
VALUES ('AD11','9239','Jorpati','Kathmandu','Province-4','Nepal','717892749','4182942','Mailing Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Landline_Number,Address_Type)
VALUES ('AD12','4954','Myagdi chowk','Kathmandu','Province-4','Nepal','4183823','Mailing Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Address_Type)
VALUES ('AD13','4589','Baniyatar chowk','Kathmandu','Province-4','Nepal','712392749','Mailing Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
VALUES ('AD14','4562','Baniyatar chowk','Kathmandu','Province-4','Nepal','712392749','4183842','Mailing Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Landline_Number,Address_Type)
VALUES ('AD15','3459','Swayambu','Kathmandu','Province-4','Nepal','41323842','Mailing Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Address_Type)
VALUES ('AD16','3451','New Road','Kathmandu','Province-4','Nepal','712392749','Mailing Address')
INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
VALUES ('AD17','3999','Myagdi chowk','Kathmandu','Province-4','Nepal','712392778','4183823','Mailing Address')
SELECT * FROM dual;
```

```
Run SQL Command Line

SQL> INSERT ALL
  2  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
  3  VALUES ('AD1','1204','Birat chowk','Biratnagar','1','Nepal','712348749','4416278','Mailing Address')
  4  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Address_Type)
  5  VALUES ('AD2','3405' , 'Panchytar', 'Nawalparasi','2','Nepal','Temporary Address')
  6  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Landline_Number,Address_Type)
  7  VALUES ('AD3','3453','New Road','Kathmandu','4','Nepal','4183842','Mailing Address')
  8  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
  9  VALUES ('AD4','5677','Baniyatar chowk','Kathmandu','4','Nepal','712392749','4183842','Mailing Address')
 10  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Address_Type)
 11  VALUES ('AD5','4356','Tangal','Kathmandu','4','Nepal','796392749','Temporary Address')
 12  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Landline_Number,Address_Type)
 13  VALUES ('AD6','5665','Jorpati','Kathmandu','4','Nepal','4182942','Mailing Address')
 14  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
 15  VALUES ('AD7','5674','Bhaktapur chowk','Bhaktapur','2','Nepal','729392749','4193842','Mailing Address')
 16  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Address_Type)
 17  VALUES ('AD8','6784','Prithivi chowk','Pokhara','6','Nepal','Temporary Address')
 18  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Landline_Number,Address_Type)
 19  VALUES ('AD9','4345','Swayambu','Kathmandu','4','Nepal','41323842','Mailing Address')
 20  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Address_Type)
 21  VALUES ('AD10','3459','Myagdi chowk','Kathmandu','4','Nepal','712392778','Mailing Address')
 22  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
 23  VALUES ('AD11','9239','Jorpati','Kathmandu','4','Nepal','717892749','4182942','Mailing Address')
 24  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Landline_Number,Address_Type)
 25  VALUES ('AD12','4954','Myagdi chowk','Kathmandu','4','Nepal','4183823','Mailing Address')
 26  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Address_Type)
 27  VALUES ('AD13','4589','Baniyatar chowk','Kathmandu','4','Nepal','712392749','Mailing Address')
 28  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
 29  VALUES ('AD14','4562','Baniyatar chowk','Kathmandu','4','Nepal','712392749','4183842','Mailing Address')
 30  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Landline_Number,Address_Type)
 31  VALUES ('AD15','3459','Swayambu','Kathmandu','4','Nepal','41323842','Mailing Address')
 32  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Address_Type)
 33  VALUES ('AD16','3451','New Road','Kathmandu','4','Nepal','712392749','Mailing Address')
 34  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
 35  VALUES ('AD17','3999','Myagdi chowk','Kathmandu','4','Nepal','712392778','4183823','Mailing Address')
 36  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
 37  VALUES ('AD18','3129','Myagdi chowk','Kathmandu','4','Nepal','712993778','4183423','Mailing Address')
 38  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
 39  VALUES ('AD19','3469','Myagdi chowk','Kathmandu','4','Nepal','712358778','4134823','Mailing Address')
 40  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
 41  VALUES ('AD20','1204','Birat chowk','Biratnagar','1','Nepal','722348749','4423278','Mailing Address')
 42  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
 43  VALUES ('AD21','1344','Birat chowk','Biratnagar','1','Nepal','732348749','4415278','Mailing Address')
 44  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
 45  VALUES ('AD22','1344','Birat chowk','Biratnagar','1','Nepal','742348749','4414278','Mailing Address')
 46  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
 47  VALUES ('AD23','1894','Birat chowk','Biratnagar','1','Nepal','752348749','4412278','Mailing Address')
 48  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
 49  VALUES ('AD24','1124','Birat chowk','Biratnagar','1','Nepal','762348749','4411278','Mailing Address')
 50  INTO Address (Address_ID, House_Number, Street, City, Province, Country, Fax_Number,Landline_Number,Address_Type)
 51  VALUES ('AD25','1994','Birat chowk','Biratnagar','1','Nepal','772348749','4499278','Mailing Address')
 52  SELECT * FROM dual;

25 rows created.
```

Figure 26: Address Insertion Statement

```
SQL> SELECT * FROM Address;

ADDRE HOUSE_NUMBER STREET               CITY           PROVINCE       COUNTRY        FAX_NUMBER     LANDLINE_NUMBER    ADDRESS_TYPE
----- ------------ -------------------- -------------- -------------- -------------- -------------- ------------------ ------------------
AD1           1204 Birat chowk          Biratnagar     1              Nepal          712348749      4416278            Mailing Address
AD2           3405 Panchytar            Nawalparasi    2              Nepal                                            Temporary Address
AD3           3453 New Road             Kathmandu      4              Nepal                         4183842            Mailing Address
AD4           5677 Baniyatar chowk      Kathmandu      4              Nepal          712392749      4183842            Mailing Address
AD5           4356 Tangal               Kathmandu      4              Nepal          796392749                         Temporary Address
AD6           5665 Jorpati              Kathmandu      4              Nepal                         4182942            Mailing Address
AD7           5674 Bhaktapur chowk      Bhaktapur      2              Nepal          729392749      4193842            Mailing Address
AD8           6784 Prithivi chowk       Pokhara        6              Nepal                                            Temporary Address
AD9           4345 Swayambu             Kathmandu      4              Nepal                         41323842           Mailing Address
AD10          3459 Myagdi chowk         Kathmandu      4              Nepal          712392778                         Mailing Address
AD11          9239 Jorpati              Kathmandu      4              Nepal          717892749      4182942            Mailing Address
AD12          4954 Myagdi chowk         Kathmandu      4              Nepal                         4183823            Mailing Address
AD13          4589 Baniyatar chowk      Kathmandu      4              Nepal          712392749                         Mailing Address
AD14          4562 Baniyatar chowk      Kathmandu      4              Nepal          712392749      4183842            Mailing Address
AD15          3459 Swayambu             Kathmandu      4              Nepal                         41323842           Mailing Address
AD16          3451 New Road             Kathmandu      4              Nepal          712392749                         Mailing Address
AD17          3999 Myagdi chowk         Kathmandu      4              Nepal          712392778      4183823            Mailing Address
AD18          3129 Myagdi chowk         Kathmandu      4              Nepal          712993778      4183423            Mailing Address
AD19          3469 Myagdi chowk         Kathmandu      4              Nepal          712358778      4134823            Mailing Address
AD20          1204 Birat chowk          Biratnagar     1              Nepal          722348749      4423278            Mailing Address
AD21          1344 Birat chowk          Biratnagar     1              Nepal          732348749      4415278            Mailing Address
AD22          1344 Birat chowk          Biratnagar     1              Nepal          742348749      4414278            Mailing Address
AD23          1894 Birat chowk          Biratnagar     1              Nepal          752348749      4412278            Mailing Address
AD24          1124 Birat chowk          Biratnagar     1              Nepal          762348749      4411278            Mailing Address
AD25          1994 Birat chowk          Biratnagar     1              Nepal          772348749      4499278            Mailing Address

25 rows selected.
```

Figure 27: Address Selection Statement

3.3.8 Inserting values into Person_Module table

```
INSERT ALL
INTO  PERSON_MODULE VALUES ('P1','M2')
INTO  PERSON_MODULE VALUES ('P1','M4')
INTO  PERSON_MODULE VALUES ('P1','M5')
INTO  PERSON_MODULE VALUES ('P1','M6')
INTO  PERSON_MODULE VALUES ('P1','M7')
INTO  PERSON_MODULE VALUES ('P2','M1')
INTO  PERSON_MODULE VALUES ('P2','M2')
INTO  PERSON_MODULE VALUES ('P2','M3')
INTO  PERSON_MODULE VALUES ('P2','M4')
INTO  PERSON_MODULE VALUES ('P5','M2')
INTO  PERSON_MODULE VALUES ('P5','M6')
INTO  PERSON_MODULE VALUES ('P5','M7')
INTO  PERSON_MODULE VALUES ('P5','M8')
INTO  PERSON_MODULE VALUES ('P5','M9')
INTO  PERSON_MODULE VALUES ('P7','M1')
INTO  PERSON_MODULE VALUES ('P7','M2')
INTO  PERSON_MODULE VALUES ('P7','M3')
```

```
INTO  PERSON_MODULE VALUES ('P7','M4')
INTO  PERSON_MODULE VALUES ('P9','M2')
INTO  PERSON_MODULE VALUES ('P9','M6')
INTO  PERSON_MODULE VALUES ('P9','M7')
INTO  PERSON_MODULE VALUES ('P9','M8')
INTO  PERSON_MODULE VALUES ('P9','M9')
INTO  PERSON_MODULE VALUES ('P10','M2')
INTO  PERSON_MODULE VALUES ('P10','M4')
INTO  PERSON_MODULE VALUES ('P10','M5')
INTO  PERSON_MODULE VALUES ('P10','M6')
INTO  PERSON_MODULE VALUES ('P10','M7')
INTO  PERSON_MODULE VALUES ('P12','M2')
INTO  PERSON_MODULE VALUES ('P12','M6')
INTO  PERSON_MODULE VALUES ('P12','M7')
INTO  PERSON_MODULE VALUES ('P12','M8')
INTO  PERSON_MODULE VALUES ('P12','M9')
INTO  PERSON_MODULE VALUES ('P3','M10')
INTO  PERSON_MODULE VALUES ('P3','M11')
INTO  PERSON_MODULE VALUES ('P3','M12')
INTO  PERSON_MODULE VALUES ('P3','M13')
INTO  PERSON_MODULE VALUES ('P3','M14')
INTO  PERSON_MODULE VALUES ('P3','M18')
INTO  PERSON_MODULE VALUES ('P4','M10')
INTO  PERSON_MODULE VALUES ('P4','M11')
INTO  PERSON_MODULE VALUES ('P4','M12')
INTO  PERSON_MODULE VALUES ('P4','M13')
INTO  PERSON_MODULE VALUES ('P4','M14')
INTO  PERSON_MODULE VALUES ('P4','M18')
INTO  PERSON_MODULE VALUES ('P6','M10')
INTO  PERSON_MODULE VALUES ('P6','M11')
INTO  PERSON_MODULE VALUES ('P6','M12')
```

```
INTO  PERSON_MODULE VALUES ('P6','M13')
INTO  PERSON_MODULE VALUES ('P6','M14')
INTO  PERSON_MODULE VALUES ('P6','M18')
INTO  PERSON_MODULE VALUES ('P8','M10')
INTO  PERSON_MODULE VALUES ('P8','M11')
INTO  PERSON_MODULE VALUES ('P8','M12')
INTO  PERSON_MODULE VALUES ('P8','M13')
INTO  PERSON_MODULE VALUES ('P8','M14')
INTO  PERSON_MODULE VALUES ('P8','M18')
INTO  PERSON_MODULE VALUES ('P11','M10')
INTO  PERSON_MODULE VALUES ('P11','M11')
INTO  PERSON_MODULE VALUES ('P11','M12')
INTO  PERSON_MODULE VALUES ('P11','M13')
INTO  PERSON_MODULE VALUES ('P11','M14')
INTO  PERSON_MODULE VALUES ('P13','M11')
INTO  PERSON_MODULE VALUES ('P13','M12')
INTO  PERSON_MODULE VALUES ('P13','M13')
INTO  PERSON_MODULE VALUES ('P14','M10')
INTO  PERSON_MODULE VALUES ('P14','M11')
INTO  PERSON_MODULE VALUES ('P14','M12')
INTO  PERSON_MODULE VALUES ('P14','M13')
SELECT * FROM dual;
```

```
SQL> INSERT ALL
  2  INTO  PERSON_MODULE VALUES ('P1','M2')
  3  INTO  PERSON_MODULE VALUES ('P1','M4')
  4  INTO  PERSON_MODULE VALUES ('P1','M5')
  5  INTO  PERSON_MODULE VALUES ('P1','M6')
  6  INTO  PERSON_MODULE VALUES ('P1','M7')
  7  INTO  PERSON_MODULE VALUES ('P2','M1')
  8  INTO  PERSON_MODULE VALUES ('P2','M2')
  9  INTO  PERSON_MODULE VALUES ('P2','M3')
 10  INTO  PERSON_MODULE VALUES ('P2','M4')
 11  INTO  PERSON_MODULE VALUES ('P5','M2')
 12  INTO  PERSON_MODULE VALUES ('P5','M6')
 13  INTO  PERSON_MODULE VALUES ('P5','M7')
 14  INTO  PERSON_MODULE VALUES ('P5','M8')
 15  INTO  PERSON_MODULE VALUES ('P5','M9')
 16  INTO  PERSON_MODULE VALUES ('P7','M1')
 17  INTO  PERSON_MODULE VALUES ('P7','M2')
 18  INTO  PERSON_MODULE VALUES ('P7','M3')
 19  INTO  PERSON_MODULE VALUES ('P7','M4')
 20  INTO  PERSON_MODULE VALUES ('P9','M2')
 21  INTO  PERSON_MODULE VALUES ('P9','M6')
 22  INTO  PERSON_MODULE VALUES ('P9','M7')
 23  INTO  PERSON_MODULE VALUES ('P9','M8')
 24  INTO  PERSON_MODULE VALUES ('P9','M9')
 25  INTO  PERSON_MODULE VALUES ('P10','M2')
 26  INTO  PERSON_MODULE VALUES ('P10','M4')
 27  INTO  PERSON_MODULE VALUES ('P10','M5')
 28  INTO  PERSON_MODULE VALUES ('P10','M6')
 29  INTO  PERSON_MODULE VALUES ('P10','M7')
 30  INTO  PERSON_MODULE VALUES ('P12','M2')
 31  INTO  PERSON_MODULE VALUES ('P12','M6')
 32  INTO  PERSON_MODULE VALUES ('P12','M7')
 33  INTO  PERSON_MODULE VALUES ('P12','M8')
 34  INTO  PERSON_MODULE VALUES ('P12','M9')
 35  INTO  PERSON_MODULE VALUES ('P3','M10')
 36  INTO  PERSON_MODULE VALUES ('P3','M11')
 37  INTO  PERSON_MODULE VALUES ('P3','M12')
 38  INTO  PERSON_MODULE VALUES ('P3','M13')
 39  INTO  PERSON_MODULE VALUES ('P3','M14')
 40  INTO  PERSON_MODULE VALUES ('P3','M18')
 41  INTO  PERSON_MODULE VALUES ('P4','M10')
 42  INTO  PERSON_MODULE VALUES ('P4','M11')
 43  INTO  PERSON_MODULE VALUES ('P4','M12')
 44  INTO  PERSON_MODULE VALUES ('P4','M13')
 45  INTO  PERSON_MODULE VALUES ('P4','M14')
 46  INTO  PERSON_MODULE VALUES ('P4','M18')
 47  INTO  PERSON_MODULE VALUES ('P6','M10')
 48  INTO  PERSON_MODULE VALUES ('P6','M11')
 49  INTO  PERSON_MODULE VALUES ('P6','M12')
 50  INTO  PERSON_MODULE VALUES ('P6','M13')
 51  INTO  PERSON_MODULE VALUES ('P6','M14')
 52  INTO  PERSON_MODULE VALUES ('P6','M18')
 53  INTO  PERSON_MODULE VALUES ('P8','M10')
 54  INTO  PERSON_MODULE VALUES ('P8','M11')
 55  INTO  PERSON_MODULE VALUES ('P8','M12')
 56  INTO  PERSON_MODULE VALUES ('P8','M13')
 57  INTO  PERSON_MODULE VALUES ('P8','M14')
 58  INTO  PERSON_MODULE VALUES ('P8','M18')
 59  INTO  PERSON_MODULE VALUES ('P11','M10')
 60  INTO  PERSON_MODULE VALUES ('P11','M11')
 61  INTO  PERSON_MODULE VALUES ('P11','M12')
 62  INTO  PERSON_MODULE VALUES ('P11','M13')
 63  INTO  PERSON_MODULE VALUES ('P11','M14')
 64  INTO  PERSON_MODULE VALUES ('P13','M11')
 65  INTO  PERSON_MODULE VALUES ('P13','M12')
 66  INTO  PERSON_MODULE VALUES ('P13','M13')
 67  INTO  PERSON_MODULE VALUES ('P14','M10')
 68  INTO  PERSON_MODULE VALUES ('P14','M11')
 69  INTO  PERSON_MODULE VALUES ('P14','M12')
 70  INTO  PERSON_MODULE VALUES ('P14','M13')
 71  INTO PERSON_MODULE VALUES ('P15','M5')
 72  INTO PERSON_MODULE VALUES ('P16','M5')
 73  INTO PERSON_MODULE VALUES ('P17','M5')
 74  INTO PERSON_MODULE VALUES ('P18','M8')
 75  INTO PERSON_MODULE VALUES ('P18','M9')
 76  INTO PERSON_MODULE VALUES ('P19','M8')
 77  INTO PERSON_MODULE VALUES ('P19','M9')
 78  INTO PERSON_MODULE VALUES ('P20','M8')
 79  INTO PERSON_MODULE VALUES ('P20','M9')
 80  INTO PERSON_MODULE VALUES ('P21','M9')
 81  INTO PERSON_MODULE VALUES ('P22','M8')
 82  INTO PERSON_MODULE VALUES ('P23','M8')
 83  SELECT * FROM dual;

81 rows created.
```

Figure 28: Person_Module Insertion Statement

```
SQL> SELECT * FROM PERSON_MODULE;

PERSO MODUL
----- -----
P1    M2
P1    M4
P1    M5
P1    M6
P1    M7
P2    M1
P2    M2
P2    M3
P2    M4
P5    M2
P5    M6
P5    M7
P5    M8
P5    M9
P7    M1
P7    M2
P7    M3
P7    M4
P9    M2
P9    M6
P9    M7
P9    M8
P9    M9
P10   M2
P10   M4
P10   M5
P10   M6
P10   M7
P12   M2
P12   M6
P12   M7
P12   M8
P12   M9
P3    M10
P3    M11
P3    M12
P3    M13
P3    M14
P3    M18
P4    M10
P4    M11
P4    M12
P4    M13
P4    M14
P4    M18
P6    M10
P6    M11

PERSO MODUL
----- -----
P6    M12
P6    M13
P6    M14
P6    M18
P8    M10
P8    M11
P8    M12
P8    M13
P8    M14
P8    M18
P11   M10
P11   M11
P11   M12
P11   M13
P11   M14
P13   M11
P13   M12
P13   M13
P14   M10
P14   M11
P14   M12
P14   M13
P15   M5
P16   M5
P17   M5
P18   M8
P18   M9
P19   M8
P19   M9
P20   M8
P20   M9
P21   M9
P22   M8
P23   M8

81 rows selected.
```

Figure 29: Person_Module Selection Statement

3.3.9 Inserting values into Person_Address table

```
INSERT ALL
INTO  Person_Address VALUES ('P1','AD1')
INTO Person_Address VALUES ('P1','AD2')
INTO Person_Address VALUES ('P2','AD3')
INTO Person_Address VALUES ('P3','AD4')
INTO Person_Address VALUES ('P3','AD5')
INTO Person_Address VALUES ('P4','AD6')
INTO Person_Address VALUES ('P5','AD7')
INTO Person_Address VALUES ('P6','AD8')
INTO Person_Address VALUES ('P7','AD9')
INTO Person_Address VALUES ('P8','AD10')
INTO Person_Address VALUES ('P9','AD11')
INTO Person_Address VALUES ('P10','AD12')
INTO Person_Address VALUES ('P11','AD13')
INTO Person_Address VALUES ('P12','AD14')
INTO Person_Address VALUES ('P13','AD15')
INTO Person_Address VALUES ('P14','AD16')
SELECT * FROM dual;
```

```
SQL> INSERT ALL
  2  INTO  Person_Address VALUES ('P1','AD1')
  3  INTO Person_Address VALUES ('P1','AD2')
  4  INTO Person_Address VALUES ('P2','AD3')
  5  INTO Person_Address VALUES ('P3','AD4')
  6  INTO Person_Address VALUES ('P3','AD5')
  7  INTO Person_Address VALUES ('P4','AD6')
  8  INTO Person_Address VALUES ('P5','AD7')
  9  INTO Person_Address VALUES ('P6','AD8')
 10  INTO Person_Address VALUES ('P7','AD9')
 11  INTO Person_Address VALUES ('P8','AD10')
 12  INTO Person_Address VALUES ('P9','AD11')
 13  INTO Person_Address VALUES ('P10','AD12')
 14  INTO Person_Address VALUES ('P11','AD13')
 15  INTO Person_Address VALUES ('P12','AD14')
 16  INTO Person_Address VALUES ('P13','AD15')
 17  INTO Person_Address VALUES ('P14','AD16')
 18  INTO Person_Address VALUES ('P15','AD17')
 19  INTO Person_Address VALUES ('P16','AD18')
 20  INTO Person_Address VALUES ('P17','AD19')
 21  INTO Person_Address VALUES ('P18','AD20')
 22  INTO Person_Address VALUES ('P19','AD21')
 23  INTO Person_Address VALUES ('P20','AD22')
 24  INTO Person_Address VALUES ('P21','AD23')
 25  INTO Person_Address VALUES ('P22','AD24')
 26  INTO Person_Address VALUES ('P23','AD25')
 27  SELECT * FROM dual;

25 rows created.
```

Figure 30: Person_Address Insertion Statement

```
SQL> SELECT * FROM Person_Address;

PERSO ADDRE
----- -----
P1    AD1
P1    AD2
P2    AD3
P3    AD4
P3    AD5
P4    AD6
P5    AD7
P6    AD8
P7    AD9
P8    AD10
P9    AD11
P10   AD12
P11   AD13
P12   AD14
P13   AD15
P14   AD16
P15   AD17
P16   AD18
P17   AD19
P18   AD20
P19   AD21
P20   AD22
P21   AD23
P22   AD24
P23   AD25

25 rows selected.
```

Figure 31: Person_Address Selection Statement

3.3.10 Inserting values into Module_Specification table

```
INSERT ALL
INTO   MODULE_SPECIFICATION VALUES ('M1','S2')
INTO   MODULE_SPECIFICATION VALUES ('M2','S2')
INTO   MODULE_SPECIFICATION VALUES ('M2','S1')
INTO   MODULE_SPECIFICATION VALUES ('M2','S3')
INTO   MODULE_SPECIFICATION VALUES ('M3','S2')
INTO   MODULE_SPECIFICATION VALUES ('M4','S2')
INTO   MODULE_SPECIFICATION VALUES ('M4','S1')
```

```
INTO   MODULE_SPECIFICATION VALUES ('M5','S1')
INTO   MODULE_SPECIFICATION VALUES ('M6','S1')
INTO   MODULE_SPECIFICATION VALUES ('M6','S3')
INTO   MODULE_SPECIFICATION VALUES ('M7','S1')
INTO   MODULE_SPECIFICATION VALUES ('M7','S3')
INTO   MODULE_SPECIFICATION VALUES ('M8','S3')
INTO   MODULE_SPECIFICATION VALUES ('M9','S3')
INTO   MODULE_SPECIFICATION VALUES ('M10','S4')
INTO   MODULE_SPECIFICATION VALUES ('M10','S5')
INTO   MODULE_SPECIFICATION VALUES ('M10','S6')
INTO   MODULE_SPECIFICATION VALUES ('M11','S4')
INTO   MODULE_SPECIFICATION VALUES ('M11','S5')
INTO   MODULE_SPECIFICATION VALUES ('M11','S6')
INTO   MODULE_SPECIFICATION VALUES ('M12','S4')
INTO   MODULE_SPECIFICATION VALUES ('M12','S5')
INTO   MODULE_SPECIFICATION VALUES ('M12','S6')
INTO   MODULE_SPECIFICATION VALUES ('M13','S4')
INTO   MODULE_SPECIFICATION VALUES ('M13','S5')
INTO   MODULE_SPECIFICATION VALUES ('M13','S6')
INTO   MODULE_SPECIFICATION VALUES ('M14','S4')
INTO   MODULE_SPECIFICATION VALUES ('M14','S5')
INTO   MODULE_SPECIFICATION VALUES ('M14','S6')
INTO   MODULE_SPECIFICATION VALUES ('M15','S6')
INTO   MODULE_SPECIFICATION VALUES ('M16','S6')
INTO   MODULE_SPECIFICATION VALUES ('M17','S6')
INTO   MODULE_SPECIFICATION VALUES ('M18','S4')
INTO   MODULE_SPECIFICATION VALUES ('M18','S5')
INTO   MODULE_SPECIFICATION VALUES ('M18','S6')
SELECT * FROM dual;
```

```
SQL> INSERT ALL
  2  INTO   MODULE_SPECIFICATION VALUES ('M1','S2')
  3  INTO   MODULE_SPECIFICATION VALUES ('M2','S2')
  4  INTO   MODULE_SPECIFICATION VALUES ('M2','S1')
  5  INTO   MODULE_SPECIFICATION VALUES ('M2','S3')
  6  INTO   MODULE_SPECIFICATION VALUES ('M3','S2')
  7  INTO   MODULE_SPECIFICATION VALUES ('M4','S2')
  8  INTO   MODULE_SPECIFICATION VALUES ('M4','S1')
  9  INTO   MODULE_SPECIFICATION VALUES ('M5','S1')
 10  INTO   MODULE_SPECIFICATION VALUES ('M6','S1')
 11  INTO   MODULE_SPECIFICATION VALUES ('M6','S3')
 12  INTO   MODULE_SPECIFICATION VALUES ('M7','S1')
 13  INTO   MODULE_SPECIFICATION VALUES ('M7','S3')
 14  INTO   MODULE_SPECIFICATION VALUES ('M8','S3')
 15  INTO   MODULE_SPECIFICATION VALUES ('M9','S3')
 16  INTO   MODULE_SPECIFICATION VALUES ('M10','S4')
 17  INTO   MODULE_SPECIFICATION VALUES ('M10','S5')
 18  INTO   MODULE_SPECIFICATION VALUES ('M10','S6')
 19  INTO   MODULE_SPECIFICATION VALUES ('M11','S4')
 20  INTO   MODULE_SPECIFICATION VALUES ('M11','S5')
 21  INTO   MODULE_SPECIFICATION VALUES ('M11','S6')
 22  INTO   MODULE_SPECIFICATION VALUES ('M12','S4')
 23  INTO   MODULE_SPECIFICATION VALUES ('M12','S5')
 24  INTO   MODULE_SPECIFICATION VALUES ('M12','S6')
 25  INTO   MODULE_SPECIFICATION VALUES ('M13','S4')
 26  INTO   MODULE_SPECIFICATION VALUES ('M13','S5')
 27  INTO   MODULE_SPECIFICATION VALUES ('M13','S6')
 28  INTO   MODULE_SPECIFICATION VALUES ('M14','S4')
 29  INTO   MODULE_SPECIFICATION VALUES ('M14','S5')
 30  INTO   MODULE_SPECIFICATION VALUES ('M14','S6')
 31  INTO   MODULE_SPECIFICATION VALUES ('M15','S6')
 32  INTO   MODULE_SPECIFICATION VALUES ('M16','S6')
 33  INTO   MODULE_SPECIFICATION VALUES ('M17','S6')
 34  INTO   MODULE_SPECIFICATION VALUES ('M18','S4')
 35  INTO   MODULE_SPECIFICATION VALUES ('M18','S5')
 36  INTO   MODULE_SPECIFICATION VALUES ('M18','S6')
 37  INTO   MODULE_SPECIFICATION VALUES ('M19','S2')
 38  INTO   MODULE_SPECIFICATION VALUES ('M20','S2')
 39  INTO   MODULE_SPECIFICATION VALUES ('M21','S2')
 40  SELECT * FROM dual;

38 rows created.
```

Figure 32: Module_Specfication Insertion Statement

```
SQL> SELECT * FROM MODULE_SPECIFICATION;

MODUL SPECI
----- -----
M1    S2
M2    S2
M2    S1
M2    S3
M3    S2
M4    S2
M4    S1
M5    S1
M6    S1
M6    S3
M7    S1
M7    S3
M8    S3
M9    S3
M10   S4
M10   S5
M10   S6
M11   S4
M11   S5
M11   S6
M12   S4
M12   S5
M12   S6
M13   S4
M13   S5
M13   S6
M14   S4
M14   S5
M14   S6
M15   S6
M16   S6
M17   S6
M18   S4
M18   S5
M18   S6
M19   S2
M20   S2
M21   S2

38 rows selected.
```

Figure 33: Module_Specification Selection Statement

4. Information and Transaction Queries

4.1 Information queries

4.1.1 List all the students with all their addresses with their phone numbers.

```
SELECT
P.Person_ID , P.First_Name || ' ' || P.Last_Name "Full Name",
P.Phone_Number, A.Address_ID, A.House_Number, A.Street, A.City,
A.Province, A.Country, A.Fax_Number,
A.Landline_Number, A.Address_Type
FROM Student S
JOIN Person P
        ON P.Person_ID = S.Person_ID
JOIN Person_Address PA
        ON P.Person_ID = PA.Person_ID
JOIN Address A
        ON PA.Address_ID = A.Address_ID;
```

In the query above, the attributes such as Person_ID, First_Name, Last_Name, Phone_Number , Address_ID, House_Number, Street, City, Province, Country, Fax_Number, Landline_Number, Address_Type are selected from the Person, Address, Person_Address tables using INNER JOIN.

```
SQL> SELECT
  2  P.Person_ID , P.First_Name || ' ' || P.Last_Name "Full Name",
  3  P.Phone_Number, A.Address_ID, A.House_Number, A.Street, A.City,
  4  A.Province, A.Country, A.Fax_Number,
  5  A.Landline_Number, A.Address_Type
  6  FROM Student S
  7  JOIN Person P
  8  ON P.Person_ID = S.Person_ID
  9  JOIN Person_Address PA
 10  ON P.Person_ID = PA.Person_ID
 11  JOIN Address A
 12  ON PA.Address_ID = A.Address_ID;
```

```
PERSO Full Name                       PHONE_NUMBER    ADDRE HOUSE_NUMBER STREET               CITY           PROVINCE       COUNTRY
     FAX_NUMBER      LANDLINE_NUMBER
----- ------------------------------- --------------- ----- ------------ -------------------- -------------- -------------- --------
--------
ADDRESS_TYPE
-----------------------------
P1    Badrinath Moktan                9855541708      AD1           1204 Birat chowk          Biratnagar     1              Nepal
     712348749       4416278
Mailing Address

P1    Badrinath Moktan                9855541708      AD2           3405 Panchytar            Nawalparasi    2              Nepal
Temporary Address

P2    Kishor Shilakar                 9855539888      AD3           3453 New Road             Kathmandu      4              Nepal
               4183842
Mailing Address

P3    Arpana Rawat                    9855568466      AD4           5677 Baniyatar chowk      Kathmandu      4              Nepal
     712392749       4183842
Mailing Address

P3    Arpana Rawat                    9855568466      AD5           4356 Tangal               Kathmandu      4              Nepal
     796392749
Temporary Address

P4    Jaya Dhungana                   9855540557      AD6           5665 Jorpati              Kathmandu      4              Nepal
               4182942
Mailing Address

P5    Naina Pun                       9855536427      AD7           5674 Bhaktapur chowk      Bhaktapur      2              Nepal
     729392749       4193842
Mailing Address

P6    Karuna Dhamala                  9855554739      AD8           6784 Prithivi chowk       Pokhara        6              Nepal
Temporary Address

P7    Raju Sinha                      9855578467      AD9           4345 Swayambu             Kathmandu      4              Nepal
               41323842
Mailing Address

P18   Dandi Dahal                     9818536427      AD20          1204 Birat chowk          Biratnagar     1              Nepal
     722348749       4423278
Mailing Address

P19   Pratik Amatya                   9870536427      AD21          1344 Birat chowk          Biratnagar     1              Nepal
     732348749       4415278
Mailing Address

P20   Hari Shrestha                   9845536427      AD22          1344 Birat chowk          Biratnagar     1              Nepal
     742348749       4414278
Mailing Address

P21   Nirajan Thapa                   9846536427      AD23          1894 Birat chowk          Biratnagar     1              Nepal
     752348749       4412278
Mailing Address

P22   Pratima Adhikari                9895536427      AD24          1124 Birat chowk          Biratnagar     1              Nepal
     762348749       4411278
Mailing Address

P23   Divya KC                        9855999427      AD25          1994 Birat chowk          Biratnagar     1              Nepal
     772348749       4499278
Mailing Address


15 rows selected.
```

Figure 34: Information Query 1

4.1.2 List all the modules which are taught by more than one instructor.

SELECT

PM.Module_ID, MAX(M.Module_Name) "Module Name", COUNT(DISTINCT PM.Person_ID) "No of Instructors"

FROM Person_Module PM

JOIN Instructor I

```
        ON PM.Person_ID = I.Person_ID
JOIN Module M
        ON PM.Module_ID = M.Module_ID
GROUP BY PM.Module_ID
HAVING COUNT(PM.Person_ID)>1 ;
```

In the query above, the attributes such as Module_Name, Module_ID, Person_ID are selected from the Person, Instructor, Person_Module and Module table using INNER JOIN. Group functions such as MAX, COUNT and GROUP BY function has also been used. The query above shows the Module names which have instructors greater than 1.

```
SQL> SELECT
  2  PM.Module_ID, MAX(M.Module_Name) "Module Name", COUNT(DISTINCT PM.Person_ID) "No of Instructors"
  3  FROM Person_Module PM
  4  JOIN Instructor I
  5  ON PM.Person_ID = I.Person_ID
  6  JOIN Module M
  7  ON PM.Module_ID = M.Module_ID
  8  GROUP BY PM.Module_ID
  9  HAVING COUNT(PM.Person_ID)>1 ;

MODUL Module Name                                        No of Instructors
----- -------------------------------------------------- -----------------
M10   Understanding Business Information                                 3
M13   International Finance and Trade                                    4
M11   Business Research and Decision-Making                              4
M7    Networks and Operating Systems                                     3
M2    Programming                                                        3
M12   Business Without Borders                                           4
M14   Services Marketing                                                 2
M8    Ethical Hacking                                                    2
M9    Communications Engineering                                         2
M6    Information Systems                                                3
M5    Databases                                                          4

11 rows selected.
```

Figure 35: Information Query 2

4.1.3 List the name of all the instructors whose name contains ‘s’ and salary is above 50,000.

```
SELECT
        P.Person_ID, P.First_Name || ' ' || P.Last_Name "Full Name", I.Salary
FROM Person P
JOIN Instructor I
        ON P.Person_ID = I.Person_ID
WHERE(LOWER(P.First_Name) LIKE '%s%' OR LOWER(P.Last_Name) LIKE '%s%')
```

AND I.Salary > 50000;

In the query above, the attributes such as Person_ID, First_Name, Last_Name, Salary are selected from the Person and Instructor table using INNER JOIN. With the use of the LIKE function, the query above shows the Person_ID, First_Name, Last_Name and Salary of the instructors who have ‘s’ in their name and has salary above 50000.

```
SQL> SELECT
  2  P.Person_ID, P.First_Name || ' ' || P.Last_Name "Full Name", I.Salary
  3  FROM Person P
  4  JOIN Instructor I
  5  ON P.Person_ID = I.Person_ID
  6  WHERE(LOWER(P.First_Name) LIKE '%s%' OR LOWER(P.Last_Name) LIKE '%s%')
  7  AND I.Salary > 50000;

PERSO Full Name                          SALARY
----- ------------------------------ ----------
P8    Siddhartha Koirala                  55000
P9    Saurya Silwal                       60000
P11   Sanjita Thapaliya                   52000
P12   Siva Choudhary                      57000
P13   Soorya Gartaula                     54000
P15   Aavash Aryal                        58000
P16   Birat Shah                          58000
P17   Sitka Candice                       58000

8 rows selected.
```

Figure 36: Information Query 3

4.1.4 List the modules comes under the ‘Multimedia’ specification.

```
SELECT
        M.Module_ID, M.Module_Name, M.Class_Name, M.Module_Level
FROM Module M
JOIN Module_Specification MS
        ON M.Module_ID = MS.Module_ID
JOIN Specification S
        ON MS.Specification_ID = S.Specification_ID
WHERE LOWER(S.Specification_Name) = 'multimedia';
```

In the query above, the attributes such Module_ID, Module_Name, Class_Name, Module_Level are selected from the Module, Module_Specification and Specification table using INNER JOIN. With the use of HAVING function, the query above shows the details of the modules of specification ‘multimedia’.

```
SQL> SELECT
  2  M.Module_ID, M.Module_Name, M.Class_Name, M.Module_Level
  3  FROM Module M
  4  JOIN Module_Specification MS
  5  ON M.Module_ID = MS.Module_ID
  6  JOIN Specification S
  7  ON MS.Specification_ID = S.Specification_ID
  8  WHERE LOWER(S.Specification_Name) = 'multimedia';

MODUL MODULE_NAME                                        CLASS_NAME      MODULE_LEV
----- ------------------------------------------------- --------------- ----------
M1    Digital Design and Image Making                    Lumbini room    Second
M2    Programming                                        Patan room      First
M3    3D Modelling                                       Everest room    Third
M4    Emerging Programming Platforms and Technologies    Patan room      Second
M19   Drawing and Character Design                       Patan room      First
M20   Project                                            Patan room      Third
M21   Moving Image and VFX                               Patan room      Second

7 rows selected.
```

Figure 37: Information Query 4

4.1.5 List the name of the head of modules with the list of his phone number.

```
SELECT
        P.Person_ID,P.First_Name, P.Last_Name, P.Phone_Number,
        I.Role
FROM Person P
JOIN Instructor I
        ON P.Person_ID = I.Person_ID
WHERE Role= 'Module Leader';
```

In the query above, the attributes such Person_ID, First_Name, Last_Name, Phone_Number, Role are selected from the Person and Instructor table using INNER JOIN. Specifying the value of Role column should be ‘Module Leader’, the query above shows the details of the Instructors who have the role ‘Module Leader’.

```
SQL> SELECT
  2  P.Person_ID,P.First_Name, P.Last_Name, P.Phone_Number,
  3  I.Role
  4  FROM Person P
  5  JOIN Instructor I
  6  ON P.Person_ID = I.Person_ID
  7  WHERE Role= 'Module Leader';

PERSO FIRST_NAME      LAST_NAME       PHONE_NUMBER    ROLE
----- --------------- --------------- --------------- ---------------
P8    Siddhartha      Koirala         9855517927      Module Leader
P11   Sanjita         Thapaliya       9855577520      Module Leader
P12   Siva            Choudhary       9855558609      Module Leader
P14   Kiran           Panday          9855524409      Module Leader
P15   Aavash          Aryal           9855524499      Module Leader
P16   Birat           Shah            9855521109      Module Leader
P17   Sitka           Candice         9855524411      Module Leader

7 rows selected.
```

Figure 38: Information Query 5

4.1.6 List all Students who have enrolled in ‘networking’ specifications.

```
SELECT
        P.Person_ID, P.First_Name, P.Last_Name,
        S.Specification_Name
FROM Person P
JOIN Specification S
        ON P.Specification_ID = S.Specification_ID
WHERE LOWER(P.Person_Type) = 'student' AND LOWER(S.Specification_Name) =
'networking';
```

In the query above, the attributes Person_ID, First_Name, Last_Name, Specification_Name are selected from the Person and Specification table using INNER JOIN. Specifying that the value of Person_Type and Specification_Name should be ‘student’ and ‘networking’ respectively, the query above shows the details of the students who are enrolled in networking specification.

```
SQL> SELECT
  2  P.Person_ID, P.First_Name, P.Last_Name,
  3  S.Specification_Name
  4  FROM Person P
  5  JOIN Specification S
  6  ON P.Specification_ID = S.Specification_ID
  7  WHERE LOWER(P.Person_Type) = 'student' AND LOWER(S.Specification_Name) = 'networking';

PERSO FIRST_NAME      LAST_NAME       SPECIFICATION_NAME
----- --------------- --------------- --------------------
P5    Naina           Pun             Networking
P18   Dandi           Dahal           Networking
P19   Pratik          Amatya          Networking
P20   Hari            Shrestha        Networking
P21   Nirajan         Thapa           Networking
P22   Pratima         Adhikari        Networking
P23   Divya           KC              Networking

7 rows selected.
```

Figure 39: Information Query 6

4.1.7 List the fax number of the instructor who teaches the 'database' module.

SELECT

P.Person_ID, P.First_Name || ' ' || P.Last_Name "Full Name", A.Fax_Number, M.Module_Name

FROM Person P

JOIN Person_Address PA

ON P.Person_ID = PA.Person_ID

JOIN Address A

ON PA.Address_ID = A.Address_ID

JOIN Person_Module PM

ON PM.Person_ID = P.Person_ID

JOIN Module M

ON PM.Module_ID = M.Module_ID

WHERE M.Module_Name = 'Databases' AND LOWER(P.Person_Type)='instructor';

In the query above, the attributes Person_ID, First_Name, Last_Name, Fax_Number and Module_Name are selected from the Person, Person_Address, Address, Person_Module and Module table using INNER JOIN. Specifying that the value of Person_Type and Module_Name

should be ‘instructor’ and ‘Databases’ respectively, the query above shows the details of the instructors who teaches the databases module.

```
SQL> SELECT
  2  P.Person_ID, P.First_Name || ' ' || P.Last_Name "Full Name", A.Fax_Number, M.Module_Name
  3  FROM Person P
  4  JOIN Person_Address PA
  5  ON P.Person_ID = PA.Person_ID
  6  JOIN Address A
  7  ON PA.Address_ID = A.Address_ID
  8  JOIN Person_Module PM
  9  ON PM.Person_ID = P.Person_ID
 10  JOIN Module M
 11  ON PM.Module_ID = M.Module_ID
 12  WHERE M.Module_Name = 'Databases' AND LOWER(P.Person_Type)='instructor';

PERSO Full Name                      FAX_NUMBER      MODULE_NAME
----- ------------------------------ --------------- ----------------------------------------
P10   Simba Mali                                     Databases
P15   Aavash Aryal                   712392778       Databases
P16   Birat Shah                     712993778       Databases
P17   Sitka Candice                  712358778       Databases
```

Figure 40: Information Query 7

4.1.8 List the specification falls under the BIT course.

```
SELECT
        S.SPECIFICATION_ID, S.Specification_Name,
        C.Course_ID, C.Course_Name
FROM Specification S
JOIN Course C
        ON S.Course_ID = C.Course_ID
WHERE UPPER(C.Course_Name) = 'BIT';
```

In the query above, the attributes Course_ID, Course_Name, Specification_ID and Specification_Name are selected from the Course and Specification table using INNER JOIN. Specifying that the value of Course_Name should be ‘BIT’, the query above shows the details of the specfications under the Course BIT.

```
SQL> SELECT
  2  S.SPECIFICATION_ID, S.Specification_Name,
  3  C.Course_ID, C.Course_Name
  4  FROM Specification S
  5  JOIN Course C
  6  ON S.Course_ID = C.Course_ID
  7  WHERE UPPER(C.Course_Name) = 'BIT';

SPECI SPECIFICATION_NAME   COURS COURSE_NAME
----- -------------------- ----- --------------------
S1    Computing            C1    BIT
S2    Multimedia           C1    BIT
S3    Networking           C1    BIT
```

Figure 41: Information Query 8

4.1.9 List all the modules taught in any one particular class.

```
SELECT
        M.Module_ID, M.Module_Name, M.Class_Name
FROM Module M
Where M.Class_Name = 'Patan room';
```

In the query above, the attributes Module_ID, Module_Name and Class_Name are selected from the Module. The query above shows the details of the modules which are taught in the Patan room.

```
SQL> Spool 'D:\Downloads\Islington\Database\CW\Queries\IQ9.sql'
SQL> SELECT
  2  M.Module_ID, M.Module_Name, M.Class_Name
  3  FROM Module M
  4  Where M.Class_Name = 'Patan room';

MODUL MODULE_NAME                                        CLASS_NAME
----- ------------------------------------------------ -----------
M2    Programming                                        Patan room
M4    Emerging Programming Platforms and Technologies    Patan room
M7    Networks and Operating Systems                     Patan room
M10   Understanding Business Information                 Patan room
M14   Services Marketing                                 Patan room
M19   Drawing and Character Design                       Patan room
M20   Project                                            Patan room
M21   Moving Image and VFX                               Patan room

8 rows selected.
```

Figure 42: Information Query 9

4.1.10 List all the teachers with all their addresses who have ‘a’ at the end of their first names.

```
SELECT
        P.Person_ID, P.First_Name || ' ' || P.Last_Name "Full Name",
        A.Address_ID, A.House_Number, A.Street, A.City,
        A.Province, A.Country, A.Fax_Number,
        A.Landline_Number, A.Address_Type
FROM Person P
JOIN Person_Address PA
        ON P.Person_ID = PA.Person_ID
JOIN Address A
        ON PA.Address_ID = A.Address_ID
WHERE P.Person_Type='Instructor' AND P.First_Name LIKE '%a';
```

In the query above, the attributes Person_ID, First_Name, Last_Name, Address_ID, House_Number, Street, City, Province, Country, Fax_Number, Landline_Number and Address_Type are selected from the Person and Address table using INNER JOIN. Specifying that the value of Person_Type should be ‘Instructor’ and the First_Name has letter a at the end of the word using LIKE ‘%a’, the query above shows the address and other basic details of the Instructor whose last letter of the first name is the letter a.

```
SQL> SELECT
  2  P.Person_ID, P.First_Name || ' ' || P.Last_Name "Full Name",
  3  A.Address_ID, A.House_Number, A.Street, A.City,
  4  A.Province, A.Country, A.Fax_Number,
  5  A.Landline_Number, A.Address_Type
  6  FROM Person P
  7  JOIN Person_Address PA
  8  ON P.Person_ID = PA.Person_ID
  9  JOIN Address A
 10  ON PA.Address_ID = A.Address_ID
 11  WHERE P.Person_Type='Instructor' AND P.First_Name LIKE '%a';

PERSO Full Name                      ADDRE HOUSE_NUMBER STREET              CITY           PROVINCE       COUNTRY        FAX_NUMBER      LANDLINE_NUMBER
ADDRESS_TYPE
----- ------------------------------ ----- ------------ -------------------- -------------- -------------- -------------- --------------- ------------------ --
----------------------
P8    Siddhartha Koirala             AD10          3459 Myagdi chowk        Kathmandu      4              Nepal          712392778
         Mailing Address
P9    Saurya Silwal                  AD11          9239 Jorpati             Kathmandu      4              Nepal          717892749       4182942
 Mailing Address
P10   Simba Mali                     AD12          4954 Myagdi chowk        Kathmandu      4              Nepal
      4183823          Mailing Address
P11   Sanjita Thapaliya              AD13          4589 Baniyatar chowk     Kathmandu      4              Nepal          712392749
         Mailing Address
P12   Siva Choudhary                 AD14          4562 Baniyatar chowk     Kathmandu      4              Nepal          712392749       4183842
 Mailing Address
P13   Soorya Gartaula                AD15          3459 Swayambu            Kathmandu      4              Nepal
      41323842         Mailing Address
P17   Sitka Candice                  AD19          3469 Myagdi chowk        Kathmandu      4              Nepal          712358778       4134823
 Mailing Address

7 rows selected.
```

Figure 43: Information Query 10

4.2 Transaction queries

4.2.1 Show the students, course they enroll in and their fees. Reduce 10% of the fees if they are enrolled in a computing course.

```
SELECT
        P.Person_ID, P.First_Name || ' ' || P.Last_Name "Full Name",
        C.Course_Name,C.Course_Fee, DECODE(S.Specification_Name,
'Computing',(C.Course_Fee - (C.Course_Fee*.1)),C.Course_Fee) "REVISED_COURSE_FEE"
FROM Person P
JOIN Course C
        ON C.Course_ID = P.Course_ID
JOIN Specification S
        ON P.Specification_ID = S.Specification_ID
WHERE LOWER(P.Person_Type)='student';
```

In the query above, the attributes Person_ID, First_Name, Last_Name, Course_Name and Course_Fee are selected from the Person, Specification and Course table using INNER JOIN if the value of the Person_Type is ‘student’. The DECODE function has also been used which checks the value of the Specification_Name attribute and reduces the values of the attribute Course_Fee

by 10% if the value of the Specfication_Name is ‘Computing’ or shows the default value of the Course_Fee.

```
SQL> SELECT
  2  P.Person_ID, P.First_Name || ' ' || P.Last_Name "Full Name",
  3  C.Course_Name,C.Course_Fee, DECODE(S.Specification_Name, 'Computing',(C.Course_Fee - (C.Course_Fee*.1)),C.Course_Fee) "REVISED_COURSE_FEE"
  4  FROM Person P
  5  JOIN Course C
  6  ON C.Course_ID = P.Course_ID
  7  JOIN Specification S
  8  ON P.Specification_ID = S.Specification_ID
  9  WHERE LOWER(P.Person_Type)='student';

PERSO Full Name                      COURSE_NAME          COURSE_FEE REVISED_COURSE_FEE
----- ------------------------------ -------------------- ---------- ------------------
P1    Badrinath Moktan               BIT                     1096500             986850
P2    Kishor Shilakar                BIT                     1096500            1096500
P3    Arpana Rawat                   BBA                     1033500            1033500
P4    Jaya Dhungana                  BBA                     1033500            1033500
P5    Naina Pun                      BIT                     1096500            1096500
P6    Karuna Dhamala                 BBA                     1033500            1033500
P7    Raju Sinha                     BIT                     1096500            1096500
P18   Dandi Dahal                    BIT                     1096500            1096500
P19   Pratik Amatya                  BIT                     1096500            1096500
P20   Hari Shrestha                  BIT                     1096500            1096500
P21   Nirajan Thapa                  BIT                     1096500            1096500
P22   Pratima Adhikari               BIT                     1096500            1096500
P23   Divya KC                       BIT                     1096500            1096500

13 rows selected.
```

Figure 44: Transaction Query 1

4.2.2 Place the default Number 1234567890 if the list of phone numbers to the location of the address is empty and give the column name as ‘Contact details.

```
SELECT P.Person_ID, P.First_Name || ' '|| P.Last_Name "Full Name",
        NVL(A.Landline_Number,'1234567890') "Contact details"
FROM Person P
JOIN Person_Address PA
        ON PA.Person_ID = P.Person_ID
JOIN Address A
        ON PA.Address_ID = A.Address_ID;
```

In the query above, the attributes Person_ID, First_Name, Last_Name and Landline_Number are selected from the Person, Person_Address and Address table using INNER JOIN. The NVL function has also been used which checks if the Landline_Number attribute is Null and replaces the Null value with ‘1234567890’ if the Landline_Number is Null or shows the default value of the Landline_Number.

```
SQL> SELECT P.Person_ID, P.First_Name || ' '|| P.Last_Name "Full Name",
  2  NVL(A.Landline_Number,'1234567890') "Contact details"
  3  FROM Person P
  4  JOIN Person_Address PA
  5  ON PA.Person_ID = P.Person_ID
  6  JOIN Address A
  7  ON PA.Address_ID = A.Address_ID;

PERSO Full Name                      Contact details
----- ------------------------------ ------------------
P1    Badrinath Moktan               4416278
P1    Badrinath Moktan               1234567890
P2    Kishor Shilakar                4183842
P3    Arpana Rawat                   4183842
P3    Arpana Rawat                   1234567890
P4    Jaya Dhungana                  4182942
P5    Naina Pun                      4193842
P6    Karuna Dhamala                 1234567890
P7    Raju Sinha                     41323842
P8    Siddhartha Koirala             1234567890
P9    Saurya Silwal                  4182942
P10   Simba Mali                     4183823
P11   Sanjita Thapaliya              1234567890
P12   Siva Choudhary                 4183842
P13   Soorya Gartaula                41323842
P14   Kiran Panday                   1234567890
P15   Aavash Aryal                   4183823
P16   Birat Shah                     4183423
P17   Sitka Candice                  4134823
P18   Dandi Dahal                    4423278
P19   Pratik Amatya                  4415278
P20   Hari Shrestha                  4414278
P21   Nirajan Thapa                  4412278
P22   Pratima Adhikari               4411278
P23   Divya KC                       4499278

25 rows selected.
```

Figure 45: Transaction Query 2

4.2.3 Show the name of all the students with the number of weeks since they have enrolled in the course.

```
SELECT P.Person_ID, P.First_Name, P.Last_Name , ROUND((SYSDATE-S.Enrolled_Date)/7)
"Enrolled Weeks"
FROM Person P
JOIN Student S
        ON S.Person_ID = P.Person_ID
WHERE LOWER(P.Person_Type)='student';
```

In the query above, the attributes Person_ID, First_Name, Last_Name and Enorlled_Date are selected from the Person and Student table using INNER JOIN if the value of the Person_Type is ‘student’. The SYSDATE accesses the current date of the system and ROUND rounds the number. In the query above, the SYSDATE is substracted from the Enrolled_Date and then divided by 7 to get the value in number of weeks. The Alias has been used to temporarily give the name of the column name as “Enrolled weeks”.

```
SQL> SELECT P.Person_ID, P.First_Name, P.Last_Name , ROUND((SYSDATE-S.Enrolled_Date)/7) "Enrolled Weeks"
  2  FROM Person P
  3  JOIN Student S
  4  ON S.Person_ID = P.Person_ID
  5  WHERE LOWER(P.Person_Type)='student';

PERSO FIRST_NAME      LAST_NAME       Enrolled Weeks
----- --------------- --------------- --------------
P1    Badrinath       Moktan                       8
P2    Kishor          Shilakar                     8
P3    Arpana          Rawat                       60
P4    Jaya            Dhungana                    60
P5    Naina           Pun                         60
P6    Karuna          Dhamala                    113
P7    Raju            Sinha                      113
P18   Dandi           Dahal                      113
P19   Pratik          Amatya                     113
P20   Hari            Shrestha                   113
P21   Nirajan         Thapa                      113
P22   Pratima         Adhikari                   113
P23   Divya           KC                         113

13 rows selected.
```

Figure 46: Transaction Query 3

4.2.4 Show the name of the instructors who got equal salary and work in the same specification.

SELECT P.Person_ID, P.First_Name || ' ' || P.Last_Name "Full Name", I.Salary, S.Specification_Name FROM Person P

JOIN Instructor I

ON P.Person_ID = I.Person_ID

JOIN Specification S

ON S.Specification_ID = P.Specification_ID

WHERE P.Person_ID IN

(SELECT DISTINCT a.Person_ID FROM Instructor a JOIN Person Pa ON a.Person_ID = Pa.Person_ID,

Instructor b JOIN Person Pb ON Pb.Person_ID = b.Person_ID WHERE a.salary = b.salary AND Pa.Specification_ID=Pb.Specification_ID AND (a.Person_ID <> b.Person_ID)) ;

In the query above, the attributes Person_ID, First_Name, Last_Name Salary and Specfication_Name are selected from the Person , Instructor and Specfication table using INNER JOIN. The IN operator checks all the values returns from the subquery which returns the distinct values of the Person_ID (since the DISTINCT function is used) whose salary and specification are the same.

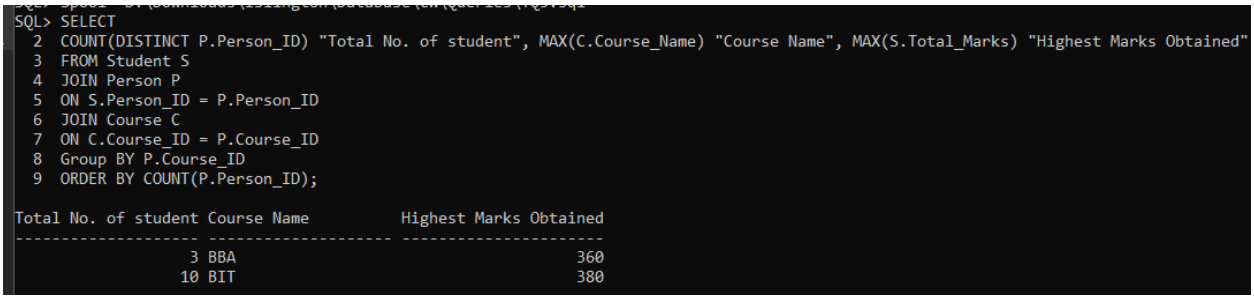
```
SQL> SELECT P.Person_ID, P.First_Name || ' ' || P.Last_Name "Full Name", I.Salary,  S.Specification_Name FROM Person P
  2  JOIN Instructor I
  3  ON P.Person_ID = I.Person_ID
  4  JOIN Specification S
  5  ON S.Specification_ID = P.Specification_ID
  6  WHERE P.Person_ID IN
  7  (SELECT DISTINCT a.Person_ID FROM Instructor a JOIN Person Pa ON a.Person_ID = Pa.Person_ID,
  8   Instructor b JOIN Person Pb ON Pb.Person_ID = b.Person_ID WHERE a.salary = b.salary AND  Pa.Specification_ID=Pb.Specification_ID AND (a.Person_ID <> b.Person_ID))
 ;

PERSO Full Name                          SALARY SPECIFICATION_NAME
----- ------------------------------ ---------- --------------------
P17   Sitka Candice                       58000 Multimedia
P16   Birat Shah                          58000 Multimedia
P15   Aavash Aryal                        58000 Multimedia
```

Figure 47: Transaction Query 4

4.2.5 List all the courses with the total number of students enrolled course name and the highest marks obtained.

```
SELECT
        COUNT(DISTINCT P.Person_ID) "Total No. of student", MAX(C.Course_Name) "Course Name", MAX(S.Total_Marks) "Highest Marks Obtained"
FROM Student S
JOIN Person P
        ON S.Person_ID = P.Person_ID
JOIN Course C
        ON C.Course_ID = P.Course_ID
Group BY P.Course_ID
ORDER BY COUNT(P.Person_ID);
```

In the query above, the attributes Person_ID, Course_Name, Total_Marks are selected from the Person, Student and Course table using INNER JOIN. The COUNT function counts the total number of record of the Person_ID in each Course_ID as GROUP BY statement is used. Since, the DISTINCT is used, COUNT counts the total number of record where Person_ID does not have a Null value. The MAX function is used which displays the highest value of the Total_Marks per the Course_ID.



```
SQL> SELECT
  2  COUNT(DISTINCT P.Person_ID) "Total No. of student", MAX(C.Course_Name) "Course Name", MAX(S.Total_Marks) "Highest Marks Obtained"
  3  FROM Student S
  4  JOIN Person P
  5  ON S.Person_ID = P.Person_ID
  6  JOIN Course C
  7  ON C.Course_ID = P.Course_ID
  8  Group BY P.Course_ID
  9  ORDER BY COUNT(P.Person_ID);

Total No. of student Course Name          Highest Marks Obtained
-------------------- -------------------- ----------------------
                   3 BBA                                     360
                  10 BIT                                     380
```

Figure 48: Transaction Query 5

4.2.6 List all the instructors who are also a course leader.

```
SELECT
        P.Person_ID, P.First_Name || ' '|| P.Last_Name "Full Name", P.Phone_Number,
        I.Role, C.Course_Name
FROM Person P
JOIN Instructor I
        ON P.Person_ID = I.Person_ID
JOIN Course C
        ON P.Course_ID = C.Course_ID
WHERE Role= 'Course Leader';
```

In the query above, the attributes Person_ID, First_Name, Last_Name, Phone_Number, Role and Course_Name are selected from the Person, Instructor and Course table using INNER JOIN. The above query shows the phone number and basic details the instructor who has the role of ‘Course Leader’.

```
SQL> SELECT
  2  P.Person_ID, P.First_Name || ' '|| P.Last_Name "Full Name", P.Phone_Number,
  3  I.Role, C.Course_Name
  4  FROM Person P
  5  JOIN Instructor I
  6  ON P.Person_ID = I.Person_ID
  7  JOIN Course C
  8  ON P.Course_ID = C.Course_ID
  9  WHERE Role= 'Course Leader';

PERSO Full Name                      PHONE_NUMBER    ROLE                           COURSE_NAME
----- ------------------------------ --------------- ------------------------------ --------------------
P9    Saurya Silwal                  9855548829      Course Leader                  BIT
P13   Soorya Gartaula                9855573753      Course Leader                  BBA
```

Figure 49: Transaction Query 6

4.3 Creation of Dump File

```
Command Prompt

C:\Users\Pratik>D:

D:\Downloads\Islington\Database\CW\Queries>EXP PratikCW/CW file = coursework.dmp

Export: Release 11.2.0.2.0 - Production on Thu Dec 17 21:03:21 2020

Copyright (c) 1982, 2009, Oracle and/or its affiliates.  All rights reserved.


Connected to: Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
Export done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set
server uses AL32UTF8 character set (possible charset conversion)

About to export specified users ...
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user PRATIKCW
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user PRATIKCW
About to export PRATIKCW's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export PRATIKCW's tables via Conventional Path ...
. . exporting table                       ADDRESS         25 rows exported
. . exporting table                        COURSE          2 rows exported
. . exporting table                    INSTRUCTOR         10 rows exported
. . exporting table                        MODULE         21 rows exported
. . exporting table           MODULE_SPECIFICATION         38 rows exported
. . exporting table                        PERSON         23 rows exported
. . exporting table                 PERSON_ADDRESS         25 rows exported
. . exporting table                  PERSON_MODULE         81 rows exported
. . exporting table                  SPECIFICATION          6 rows exported
. . exporting table                        STUDENT         13 rows exported
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting operators
. exporting referential integrity constraints
. exporting triggers
. exporting indextypes
. exporting bitmap, functional and extensible indexes
. exporting posttables actions
. exporting materialized views
. exporting snapshot logs
. exporting job queues
. exporting refresh groups and children
. exporting dimensions
. exporting post-schema procedural objects and actions
. exporting statistics
Export terminated successfully without warnings.

D:\Downloads\Islington\Database\CW\Queries>
```

Figure 50: Screenshot of Dump File Created

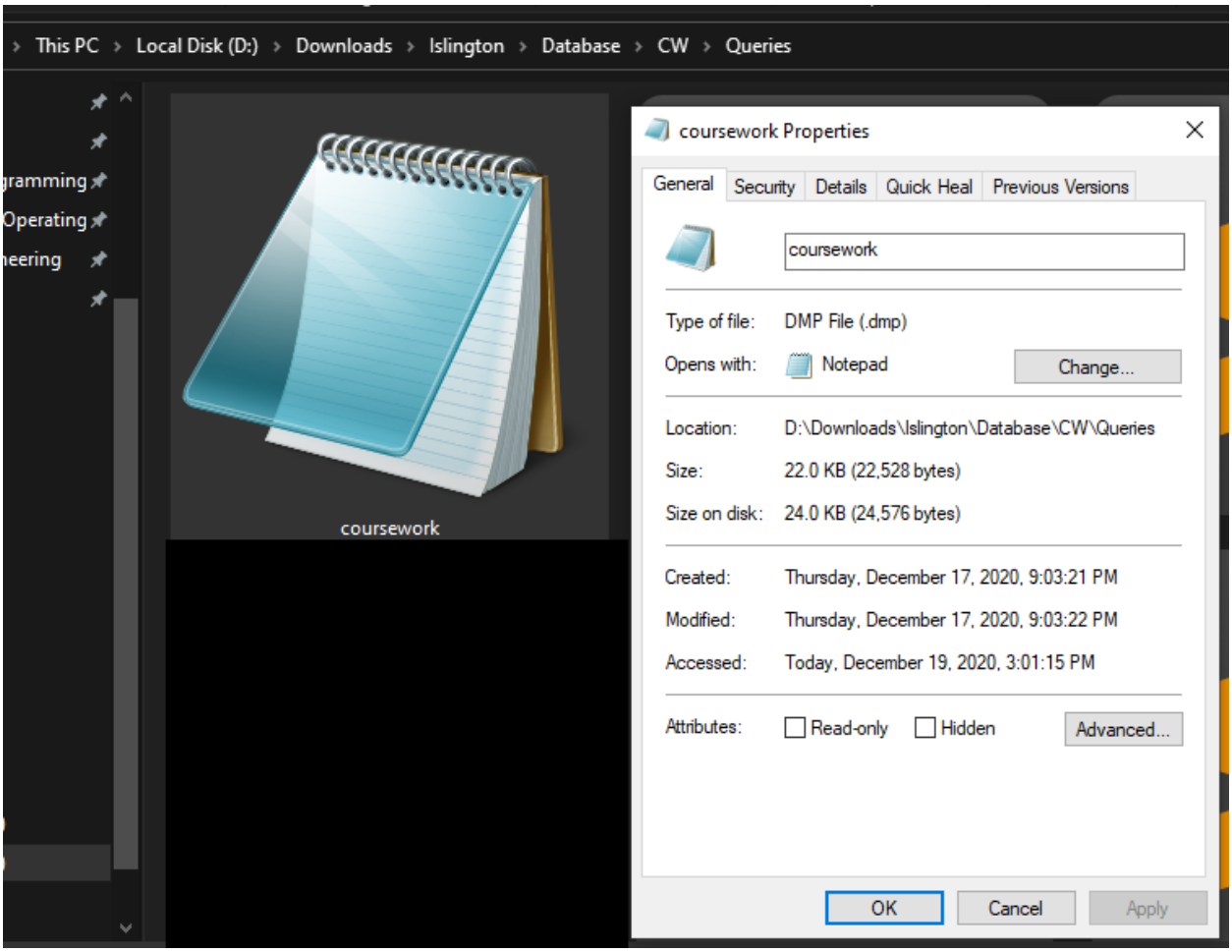


Figure 51: Screenshot of the actual Dump File

4.4 Drop tables

All the tables in the user are dropped in the following order:

Drop Table Person_Module;

Drop Table Person_Address;

Drop Table Module_Specification;

Drop Table Instructor;

Drop Table Student;

Drop Table Person;

Drop Table Specification;

Drop Table Module;

Drop Table Address;

Drop Table Course;

```
SQL> Drop Table Person_Module;

Table dropped.

SQL> Drop Table Person_Address;

Table dropped.

SQL> Drop Table Module_Specification;

Table dropped.
```

```
SQL> Drop Table Instructor;

Table dropped.

SQL> Drop Table Student;

Table dropped.

SQL> Drop Table Person;

Table dropped.

SQL> Drop Table Specification;

Table dropped.

SQL> Drop Table Module;

Table dropped.

SQL> Drop Table Address;

Table dropped.

SQL> Drop Table Course;

Table dropped.
```

Figure 52: Dropping Tables

5. Conclusion

The coursework was completed in time through proper dedication and research work. The objective of the coursework was to design and implement a database for a private college. It was not that challenging to identity the entities and its attributes. The tasks done till initial ERD was not that troublesome. But, the most challenging part of the coursework was the normalization part. There were many problems in the implementation of the supertype and subtype in my case. And I had many misconceptions in the 1NF, 2NF and 3NF of normalization. But after many attempts and back and forth of mails, the normalization was finally approved by our lecturer Yunisha ma'am who is also our person academic tutor.

Through the normalization, the keys such as Primary keys, Foreign keys and unique keys were identified. The normalization helped in the reduction of data redundancy and establish clear relationships between the entities. The normalization helped in the understanding of the partial and transitive dependencies. The data insertion and creation of tables was not difficult compare to the queries which required a lot of knowledge gained though research about SQL statements. The lab and lecture slides were read many times to solve the queries as some of the queries were quite difficult to solve.

I am very thankful to the effort from our tutors who answered all our queries whether mail or through video meet. The coursework couldn't have been done without their feedback. This coursework helped me learn the proper implementation of normalization through practise.

During the completion of the coursework, it was difficult to maintain proper time management as there were other three coursework of other modules as well. But through the course work, a lot of knowledge on the designing and implementation of the database, the normalization process and the SQL statements were gained. The ideas gained from this coursework was used to design the ER diagram of the Software Engineering module. And the experience gained from this coursework will help in the future projects as well.

6. Bibliography

Brumm, B. (2017) *How to Handle a Many-to-Many Relationship in Database Design - DZone Database* [Online]. Available from: https://dzone.com/articles/how-to-handle-a-many-to-many-relationship-in-datab [Accessed 1 December 2020].

Castro, K. (2018) *Many-to-Many Relationship in DBMS* [Online]. Available from: https://www.tutorialspoint.com/Many-to-Many-Relationship-in-DBMS [Accessed 1 December 2020].

FreeTutes. (n.d.) *E-R Model concept* [Online]. Available from: https://www.freetutes.com/systemanalysis/sa7-e-r-model-concept.html [Accessed 1 December 2020].

GeeksforGeeks. (2019) *SQL _ DDL, DQL, DML, DCL and TCL Commands - GeeksforGeeks* [Online]. Available from: https://www.geeksforgeeks.org/sql-ddl-dql-dml-dcl-tcl-commands/ [Accessed 17 December 2020].

Guru99. (2020) *ER Diagram_ Entity Relationship Diagram Model _ DBMS Example* [Online]. Available from: https://www.guru99.com/er-diagram-tutorial-dbms.html [Accessed 1 December 2020].

Guru99. (2020) *What is Normalization 1NF, 2NF, 3NF, BCNF Database Example* [Online]. Available from: https://www.guru99.com/database-normalization.html [Accessed 3 December 2020].

Islington. (2018) *Vision, Mission, Values - Islington College* [Online]. Available from: https://islington.edu.np/about/vision-mission-values/ [Accessed 7 December 2020].

My Reading Room. (2016) *Completeness Constraint* [Online]. Available from: http://www.myreadingroom.co.in/notes-and-studymaterial/65-dbms/493-completeness-constraint.html#:~:text=Partial%20completeness%20(symbolized%20by%20a,not%20members%20of%20any%20subtype.&text=A%20double%20horizontal%20line%20under%20the%20circle%20represen [Accessed 12 December 2020].

Onsman, A. (2018) *Transitive dependency in DBMS* [Online]. Available from: https://www.tutorialspoint.com/Transitive-dependency-in-DBMS [Accessed 13 December 2020].

Oracle. (2020) *Tables for Supertype and Subtype Entities in Oracle Communications Data Model* [Online]. Available from: https://docs.oracle.com/cd/E29633_01/CDMOG/GUID-80902404-F909-4884-B5AE-

B0BE6EEE0781.htm#:~:text=A%20supertype%20is%20a%20generic,relationships%20distinct%20from%20other%20subgroups. [Accessed 11 December 2020].

Palmer, C.C. (2020) *Lecture 05* [Online]. Available from: https://www.cs.dartmouth.edu/~cs61/Lectures/05%20-%20Advanced%20Data%20Modeling/05%20-%20Advanced%20Data%20Modeling.html#:~:text=A%20subtype%20discriminator%20is%20an,as%20one%20of%20the%20subtypes. [Accessed 11 December 2020].

Tutorialspoint. (2020) *ER Model - Basic Concepts - Tutorialspoint* [Online]. Available from: https://www.tutorialspoint.com/dbms/er_model_basic_concepts.htm [Accessed 1 December 2020].

Tutorialspoint. (2020) *SQL - Foreign Key - Tutorialspoint* [Online]. Available from: https://www.tutorialspoint.com/sql/sql-foreign-key.htm [Accessed 1 December 2020].

Tutorialspoint. (2020) *SQL - Transactions - Tutorialspoint* [Online]. Available from: https://www.tutorialspoint.com/sql/sql-transactions.htm#:~:text=to%20the%20database.-,The%20COMMIT%20command%20is%20the%20transactional%20command%20used%20to%20save,last%20COMMIT%20or%20ROLLBACK%20command. [Accessed 17 December 2020].