LONDON METROPOLITAN UNIVERSITY

Module Code & Module Title

CC4057NT Introduction to Information System

Assessment Weightage & Type

30% Individual Coursework

Year and Semester

2019-20 Autumn

Student Name: Sujan Chaudhary

London Met ID: 19031885

College ID:

Assignment Due Date:

Assignment Submission Date: 2020/11/21

Title (Where Required):

Word Count (Where Required):

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.

Contents

1.0. INTRODUCTION

The name is COBASS college which stands for College Of business and social studies which has to offer courses like computer BBA, BIT, MSc, BHM, BBS, CSit, etc. We believe in providing quality education not quantity we only hire limited student by taking entrance exam student who pass entrance are allowed to study in college. All professors are highly qualified and experienced who have been teaching for 20-30 years.

The college is of 7 story building well furniture. The school is set in spacious, landscaped grounds with modern buildings which includes basketball court, volleyball court and also badminton court we only not focus on education but also to extra-curriculum activities each year from our college participate in all type of extra curriculum activities like football, volleyball, table-tennis first of all we qualify best students for the competition to compete in a higher level. We have well maintained library for students to read. Since 2010 our college have topped in BIT course stream all over the Nepal till the date, we produce many best students who become successful in professional life there are many students who pass out from our college work in many international and national companies at best post and many have started their own startup and become successful. In other course also our student tops all years.

To create college a put where a group of energetic, experienced and qualified teacher and speakers will empower the understudies of all levels to charge the way they think by making them realize their position within the universally competitive world through the application of their learned expertise and competence.

1.1. AIM

- Provide the highest quality teaching and learning.
- Challenge the boundaries of knowledge, research, and discipline.
- Give proper guidance for the future studies and present studies to achieve their goal.
- supporting students financially by providing scholarship, prizes and as well as assisting students in particular financial hardship.
- providing library, computing, cultural, sporting and social facilities to enable students to achieve their full potential both academically and otherwise
- provide equal education without any type of discrimination to all student without regards to gender, marital status, color, race, religion, national origin or disability.
- Provide additional courses that have a practical and vocational orientation to improve skills of students.

1.2. Objectives

It is the objective of the college to impart four-year Under Graduate Honors and General Courses in many streams like BIT, BBA, BBS, etc. to the students admitted through proper counselling and thus prepare them as really human resources with knowledge and skill so that they can keep pace with the modern society and enter into the vast arena of higher education. Our college is simply a temple of knowledge that specially to develop in its student's modern temper, computerized work force, scientific outlook, respectful attitude with selfless behavior and freedom from superstitions which in long run will enable them to go step by step with the modern world. As a typical Nepali we believe in superstitions to get rid out of that our college conduct awareness programs.

1.3. Description of Current Business Activities and Operation

As you all know, the traditional functions of universities are teaching & mostly basic research. These activities are carried out by the academic staff like lab teacher, tutor, etc. The administrative staff including the human resources, finance department, IT department, etc. role is to support the academic departments in carrying out their duties in the best optimum manner. All are highly qualified and experienced. Human resource is responsible for conducting interviews, hiring applicants, dealing with interpersonal conflicts and determining the benefit packages employees should receive. Working papers, drafts, duplicate copies of records stored elsewhere, short-term facilitative records (such as phone messages), and unimportant records such as unsolicited 'junk mail' which may be destroyed without approval once administrative use has ended. Advertising the college for boosting the exposure of college and for development. We offer two shifts day and morning for students so that they can have their class in there favorable time.

1.4. Business rule

- Student should maintain discipline according to the rules book of the college if they don't follow, they will get punished.
- All student data is recorded and handled by IT department.
- Many teachers can teach one course but one teacher can only one course
- In one class only one module should teach

2.0. Identification of entities and attribute

2.1. Entities

Entity is the thing, person, place, unit, object or any item about which the data should be captured and stored in the form of properties, workflow and tables. While workflow and tables are optional for database entity, properties are required because entity without properties is not an entity

(study, © copyright 2003-2020)

Course, address, student, module, specification, instructor, class, course_leader

2.2. Attribute

Attribute is a property or characteristic. All attributes have values. For example, a student entity may have name, joining-date, student_id as attributes.

(study, © copyright 2003-2020)

Course
Course_id
Course_name
Course_fee
credit

address
Address_id
City
Province
country
Fax
Phone_number
House_number

Instructor
Instructo_id
Instructor_name
Experience
Instructor

specification
Specification_id
Specification_name

Module
Module_id
Module_name

Class
Class_id
Class_name
Block

Cours_leader
qaulification

Student
Student_id
Name
Gender
Dob
Marks
Joining date

2.3. Initial Entity Relational Diagram ERD

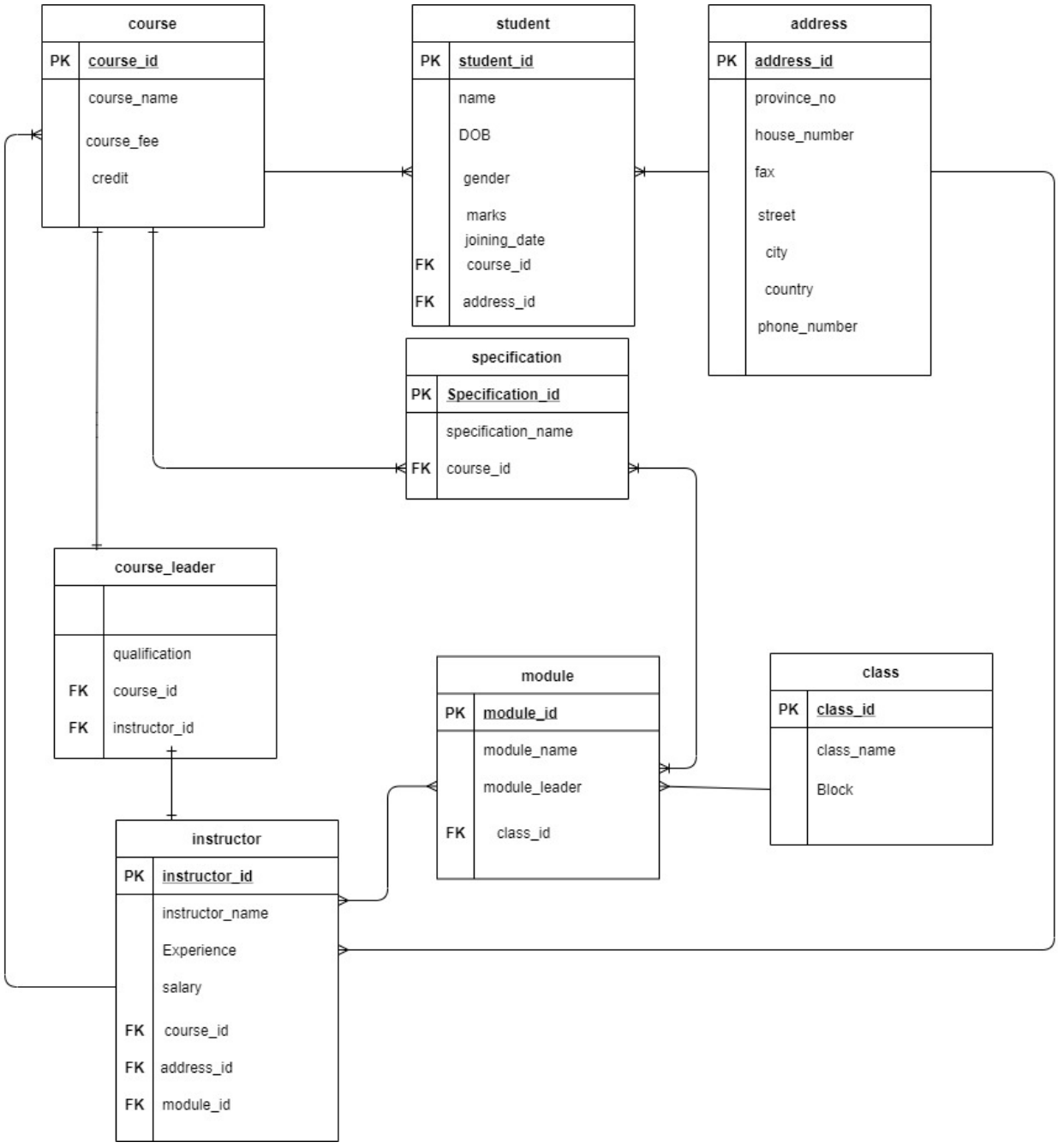

Figure 1:initial ERD

ERD full form is entity relationship diagram which shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties.

(smart draw, ©1994-2020 SmartDraw, LLC)

Since it is initial ERD it unnormalized there is different error like anomalies, repetition of data which should normalized. Primary key and foreign keys are shown in the ERD as it just a initial ERD final ERD is made after normalization completion.

3.0. Data base design

3.1. Assumption

- Student and instructor can't have same phone number and address
- Student and instructor both can only have one permanent address and many temporary addresses, if they have multiple temporary addresses, they should list down
- One instructor can teach multiple courses.
- Student can't enroll multiple courses.

3.2. Normalization

Normalization is the method of organizing information in a database. This incorporates making tables and setting up connections between those tables concurring to rules designed both to protect the information and to form the database more adaptable by eliminating redundancy and inconsistent dependency. It has three steps 1NF, 2NF, 3NF. Every column have unique key in 1NF, partial dependency is checked of every tables in 2NF, transitive dependency is checked of every tables in 3NF.

(microsoft, © Microsoft 2020)

3.3. UNF OF COLLEGE

course (course_id(PK), course_name, fee, credit, {student_id, name, marks, gender, DOB, joining-date, {address_id, location, city, fax, country, street, house_number, province_no, phone_number}}, {specification_id, specification_name, {modules_id, modules_name, module_leader,{ class_id, class_name,Block}}}, {instructor_id, instructor_name}, {qualification, course-leader_name})

- **Scenario for UNF as how the records are maintained**
    1. Student can enroll in only one course and each course can have any number of students.
    2. Each instructor can teach any one or many modules at a time and a module can teach by any instructor.
    3. Each address consists of country, province, city, street, house number, phone numbers and fax.

4. Each course can offer any number of specifications computing, networking, marketing.
5. Each person, it records all his / her address, of which exactly one is designated as the mailing address.
6. Each module is taught in any given particular class, but in each class a number of modules are taught.
7. A student can enroll for any one course and each course can have any number of students

1NF OF COLLEGE

Course (course_id(PK), course_name, course_fee,credit)

Student (student_id(PK), name, DOB, gender, marks, joining_date, course_id(FK), address_id(FK))

Address(address_id(PK),province_no,house_number,fax,street,city,country, phone_number)

Fax-address (fax(PK), address_id(FK))

phone_number_address (phone_number(PK), address_id(FK))

Specification (specification_id(PK), specification_name, course_id(FK))

Module (modules_id(PK), modules_name, module_leader(FK), specifaction_id(pk, FK), class_id(FK))

Instructor(instructor_id(PK),firstname,lastname,experience,salary,course_id(FK),address_id(FK), module_id(PK,FK))

Course-leader (qualification, courseleader_id(PK, FK), instructor_id(FK))

Class detail-1 (class_id(PK), class_name, Block)

2NF FORM FOR COLLEGE

Course-detail-2(course_id(PK), course_name, course_fee,credit)

Checking Partial dependency for student table

Student-2 (student_id(PK), name, DOB, gender, marks, joining_date, course_id(FK), address_id(FK))

Checking Partial dependency for ADDRESS table

Address(address_id(PK),province_no,house_number,fax,street,city, country,phone_number)

Fax-address (fax(PK), address_id(FK))

phone_number_address (phone_number(PK), address_id(FK))

Checking Partial dependency for specification table

Specification(specification_id(PK),specification_name, course_id(FK))

Checking partial dependency for module table

Module (modules_id(PK), modules_name, module_leader(FK), specifaction_id(pk, FK), class_id(FK))

Module (modules_id(PK), modules_name, module_leader(FK), class_id(FK)

Specification_module (modules_id(PK,FK), specifaction_id(PK, FK)

Checking partial dependency for instructor table

Instructor (instructor_id(PK), firstname, lastname, experience, salary, module_id(PK,FK), course_id(FK), address_id(FK))

Instructor_id->experience, salary, firstname,lastname, course_id(FK), address_id(FK)

Instructor_id-module_id->

Instructor-2 (Instructor_id(PK),firstname,lastname,experience, salary, course_id(FK), address_id(FK))

Module_instructor (Instructor_id(PK,FK), module_id(PK,FK))

Checking partial dependency for course-leader table

Course-leader (qualification, course_id(FK), instructor_id(FK))

Checking partial dependency for class table

Class (class_id(PK), class_name, Block)

Final 2NF

Course-detail-2 (course_id(PK), course_name, course_fee,credit)

Student-2 (student_id(PK), name, DOB, gender,marks, joining_date, course_id(FK), address_id(FK))

Address(address_id(PK),province_no,house_number,fax,street,city, country,phone_number)

Fax-address (fax(PK), address_id(FK))

Address-phone_number (phone_number,(PK)address_id(FK))

Specification(specification_id(PK),specification_name, course_id(FK))

Module (modules_id(PK), modules_name, module_leader(FK), class_id(FK)

Specification_module (modules_id(PK,FK), specifaction_id(PK, FK)

Class-2 (class_id(PK), class_name, Block)

Instructor-2 (Instructor_id(PK), firstname,lastname, experience, salary, course_id(FK), address_id(FK))

Instructor-module (Instructor_id(PK,FK), module_id(PK,FK))

Course-leader-2 (qualification, course_id(FK), instructor_id(FK))

Class-2 (class_id(PK), class_name, Block)

3NF form

Checking transitive dependency for course table

Course-2(course_id(PK), course_name, course_fee, credit)

Course_id-> course_name>course_fee>credit

Course-3 (Course_id,(PK) course_name, course_fee, credit)

Checking transitive dependency for student table

Student-2 (student_id(PK), name, DOB, gender, marks, joining_date, course_id(FK), address_id(FK))

Student_id->name >DOB> gender>marks>joining_date>course_id>address_id

Student-3 (Student_id(PK), name, DOB, gender,marks, course_id(FK), address_id(FK))

Checking transitive dependency for address table

Address(address_id(PK),province_no,house_number,fax,street,city,country, phone_number)

Address_id-> province_no> house_number>fax>street>city>country>phone_number

address-3(address_id(PK),province_no,house_number,fax,street,city, country,phone_number)

Fax_address (fax(PK), address_id(FK))

phone_number_address (phone_number(PK),address_id(FK))

Checking transitive dependency for specification table

Specification(specification_id(PK),specification_name,course_id(FK))

Specification_id-> specification_name>course_id(FK)

Specification-3 (Specification_id(PK), specification_name, course_id(FK))

Checking transitive dependency for module table

Module (modules_id(PK), modules_name, module_leader(FK), class_id(FK))

Module_id> modules_name> module_leader(FK)> class_id(FK)

Module-3 (modules_id(PK), modules_name, module_leader(FK), class_id(FK)

Specification_module (modules_id(PK,FK), specifaction_id(PK, FK)

Checking transitive dependency for instructor table

Instructor-2 (Instructor_id(PK), firstname, lastname, experience, salary, course_id(FK), address_id(FK))

Instructor_id->firstname>lastname> experience>salary>course_id(FK)> address_id(FK)

Module_instructor (Instructor_id(PK,FK), module_id(PK,FK))

Instructor-3 (Instructor_id(PK), instructor_name, experience, salary, course_id(FK), address_id(FK))

Checking transitive dependency for course leader table

Course-leader-3 (qualification, course_id(FK), instructor_id(FK))

Checking transitive dependency for class table

Class-3 (class_id(PK), class_name, Block)

Final 3NF

Course-3 (Course_id,(PK) course_name,course_fee, credit)

Student-3 (Student_id(PK), name,DOB, gender, marks, joinig_date, course_id(FK), address_id(FK))

address-3(address_id(PK),province_no,house_number,fax,street,city, country,phone_number)

Fax_address (fax(PK), address_id(FK))

phone_number_address (phone_number(PK), address_id(FK))

Specification-3 (Specification_id(PK), specification_name, course_id(FK))

Module-3 (modules_id(PK), modules_name, module_leader(FK), class_id(FK)

Specification_module (modules_id(PK,FK), specifaction_id(PK, FK)

Instructor-3 (Instructor_id(PK), firstname,lastname, experience, salary, course_id(FK), address_id(FK))

Module_instructor (Instructor_id(PK,FK), module_id(PK,FK))

Course_leader-3 (qualification, course_id(FK), instructor_id(FK))

Class-3 (class_id(PK), class_name,Block)

4.0. Final ERD

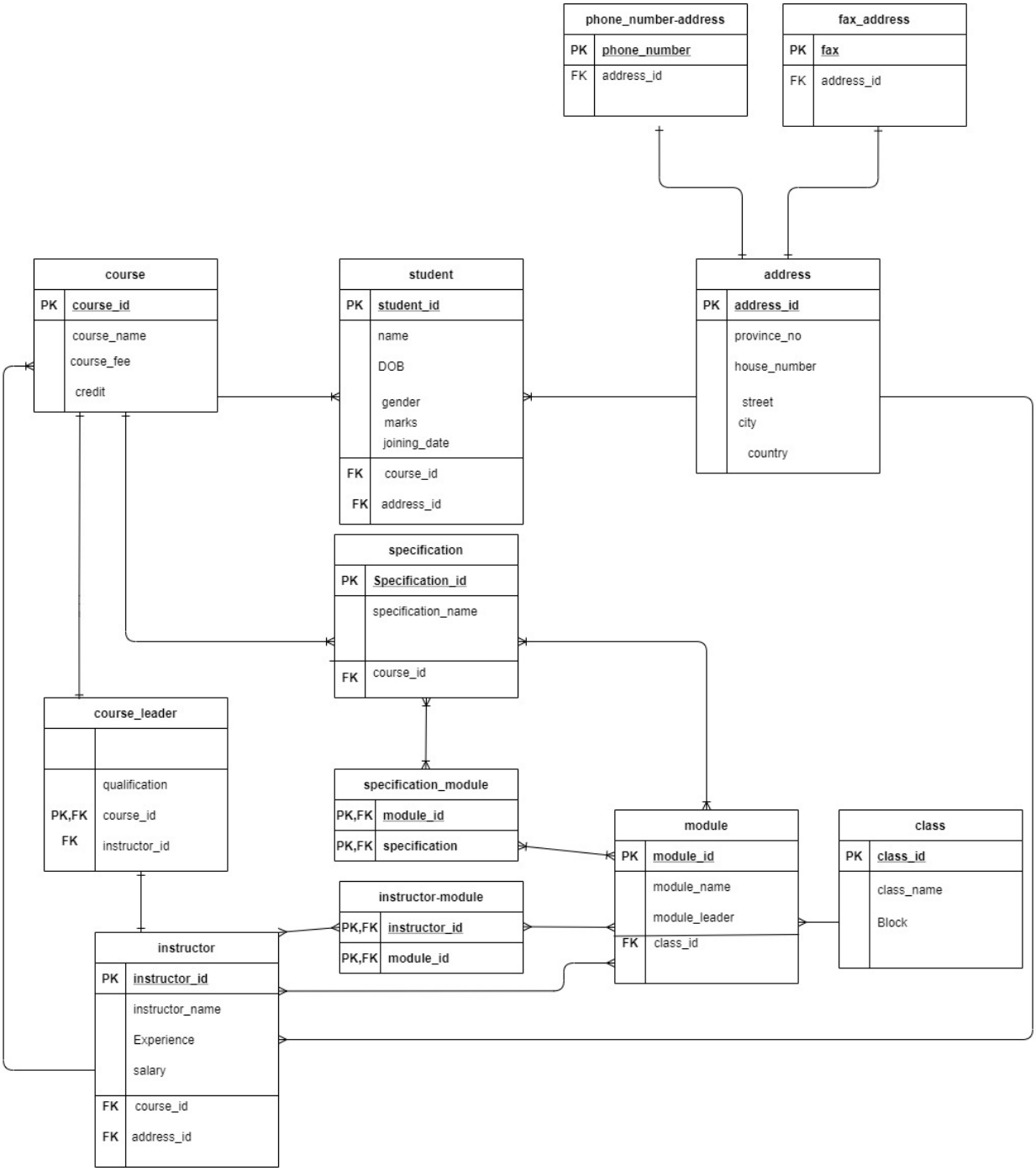

Figure 2:FInal ERD

5.0. Implementation

5.1. Creation of tables

1. Course

```
create table Course(
course_id int not null,
course_name varchar(25),
fee int not null,
duration varchar(25) not null,
constraint course_id_pk primary key(course_id));
```

```
SQL> create table Course(
  2    course_id int not null,
  3    course_name varchar(30) not null,
  4    course_fee int not null,
  5    credit int not null,
  6    constraint course_id_pk primary key(course_id));

Table created.

SQL> desc course
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COURSE_ID                                 NOT NULL NUMBER(38)
 COURSE_NAME                               NOT NULL VARCHAR2(30)
 COURSE_FEE                                NOT NULL NUMBER(38)
 CREDIT                                    NOT NULL NUMBER(38)
```

Figure 3:create table course

2. Address

```
create table Address(
address_id int not null,
province_no int not null,
house_number int not null,
```

street varchar(30) not null,

city varchar(30) not null,

country varchar(30) not null,

constraint address_id_pk primary key(address_id));

```
SQL> create table Address(
  2  address_id int not null,
  3  province_no int not null,
  4  house_number varchar(30) not null,
  5  street varchar(30) not null,
  6  city varchar(30) not null,
  7  country varchar(30) not null,
  8  constraint address_id_pk primary key(address_id));

Table created.

SQL> desc address
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ADDRESS_ID                                NOT NULL NUMBER(38)
 PROVINCE_NO                               NOT NULL NUMBER(38)
 HOUSE_NUMBER                              NOT NULL VARCHAR2(30)
 STREET                                    NOT NULL VARCHAR2(30)
 CITY                                      NOT NULL VARCHAR2(30)
 COUNTRY                                   NOT NULL VARCHAR2(30)
```

Figure 4:create table address

3. Class

create table Class(

class_id int not null,

class_name varchar(30) not null,

Block varchar(30) not null,

constraint class_id_pk primary key(class_id));

```
SQL> create table Class(
  2  class_id int not null,
  3  class_name varchar(30) not null,
  4  Block varchar(30) not null,
  5  constraint class_id_pk primary key(class_id));

Table created.

SQL> desc class
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLASS_ID                                  NOT NULL NUMBER(38)
 CLASS_NAME                                NOT NULL VARCHAR2(30)
 BLOCK                                     NOT NULL VARCHAR2(30)
```

Figure 5:create table class

4. Student

create table Student(

student_id int not null,

name varchar(30) not null,

DOB date not null,

gender varchar(30) not null,

marks int not null,

Joining_date date not null,

course_id int not null,

address_id int not null,

constraint student_id_pk primary key(student_id),

constraint course_id_fk1 foreign key(course_id) references Course(course_id),

constraint address_id_fk4 foreign key(address_id) references Address(address_id));

```
SQL> create table Student(
  2  student_id int not null,
  3  name varchar(30) not null,
  4  DOB date not null,
  5  gender varchar(30) not null,
  6  marks int not null,
  7  Joining_date date not null,
  8  course_id int not null,
  9  address_id int not null,
 10  constraint student_id_pk primary key(student_id),
 11  constraint course_id_fk1 foreign key(course_id) references Course(course_id),
 12  constraint address_id_fk4 foreign key(address_id) references Address(address_id));

Table created.

SQL> desc student
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STUDENT_ID                                NOT NULL NUMBER(38)
 NAME                                      NOT NULL VARCHAR2(30)
 DOB                                       NOT NULL DATE
 GENDER                                    NOT NULL VARCHAR2(30)
 MARKS                                     NOT NULL NUMBER(38)
 JOINING_DATE                              NOT NULL DATE
 COURSE_ID                                 NOT NULL NUMBER(38)
 ADDRESS_ID                                NOT NULL NUMBER(38)
```

Figure 6:create table student

5. Module

create table Module(

module_id int not null,

module_name varchar(30) not null,

module_leader int not null,

class_id int not null,

constraints module_id_pk primary key(module_id),

constraints class_id_fk1 foreign key(class_id) references class(class_id),

constraint module_leader_fk foreign key(module_leader) references instructor(instructor_id));

```
SQL> create table Module(
  2  module_id int not null,
  3   module_name varchar(30) not null,
  4  module_leader int not null,
  5  class_id int not null,
  6  constraints module_id_pk primary key(module_id),
  7  constraints class_id_fk1 foreign key(class_id) references class(class_id),
  8  constraint module_leader_fk foreign key(module_leader) references instructor(instructor_id));

Table created.

SQL> desc module
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MODULE_ID                                 NOT NULL NUMBER(38)
 MODULE_NAME                               NOT NULL VARCHAR2(30)
 MODULE_LEADER                             NOT NULL NUMBER(38)
 CLASS_ID                                  NOT NULL NUMBER(38)
```

Figure 7:create table module

6. Instructor

instructor_id int not null,

first_name varchar(30) not null,

last_name varchar(30) not null,

experience varchar(30) not null,

salary int not null,

course_id int not null,

address_id int not null,

constraint instructor_id_pk primary key(instructor_id),

constraint course_id_fk foreign key(course_id) references course(course_id),

constraint address_id_fk3 foreign key(address_id) references address(address_id))

```
SQL> create table instructor(
  2    instructor_id int not null,
  3    firstname varchar(30) not null,
  4    lastname varchar(30) not null,
  5    experience varchar(30) not null,
  6    salary int not null,
  7    course_id int not null,
  8    address_id int not null,
  9    constraint instructor_id_pk primary key(instructor_id),
 10    constraint course_id_fk foreign key(course_id) references course(course_id),
 11    constraint address_id_fk3 foreign key(address_id) references address(address_id));

Table created.

SQL> desc instructor
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 INSTRUCTOR_ID                             NOT NULL NUMBER(38)
 FIRSTNAME                                 NOT NULL VARCHAR2(30)
 LASTNAME                                  NOT NULL VARCHAR2(30)
 EXPERIENCE                                NOT NULL VARCHAR2(30)
 SALARY                                    NOT NULL NUMBER(38)
 COURSE_ID                                 NOT NULL NUMBER(38)
 ADDRESS_ID                                NOT NULL NUMBER(38)
```

Figure 8:create table instructor

7. Fax_address

Create table fax_addres(

fax int not null,

address_id int not null,

constraint fax_pk primary key(fax),

constraint address_id_fk foreign key(address_id) references Address(address_id));

```
SQL> create table fax_address(
  2    fax int not null,
  3    address_id int not null,
  4    constraint fax_pk primary key(fax),
  5    constraint address_id_fk foreign key(address_id) references Address(address_id));

Table created.

SQL> desc fax_address
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 FAX                                       NOT NULL NUMBER(38)
 ADDRESS_ID                                NOT NULL NUMBER(38)

SQL> create table phonenumber_address(
  2    phone_number int not null,
  3    address_id int not null,
  4    constraint phone_number_pk primary key(phone_number),
  5    constraint address_id_fk2 foreign key(address_id) references Address(address_id));

Table created.

SQL> desc phonenumber_address
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PHONE_NUMBER                              NOT NULL NUMBER(38)
 ADDRESS_ID                                NOT NULL NUMBER(38)
```

Figure 9:create table fax_address

8. Phonenumber_address

Create table phonenumber_address

phone_number int not null,

address_id int not null,

constraint phone_number_pk primary key(phone_number),

constraint address_id_fk2 foreign key(address_id) references Address(address_id));

```
SQL> create table phonenumber_address(
  2    phone_number int not null,
  3    address_id int not null,
  4    constraint phone_number_pk primary key(phone_number),
  5    constraint address_id_fk2 foreign key(address_id) references Address(address_id));

Table created.

SQL> desc phonenumber_address
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PHONE_NUMBER                              NOT NULL NUMBER(38)
 ADDRESS_ID                                NOT NULL NUMBER(38)
```

Figure 10:create table phonenumber_address

9. Specification

create table specification(

specification_id int not null,

specification_name varchar(30),

course_id int not null,

constraint specification_id_pk primary key(specification_id),

constraint course_id_fk3 foreign key(course_id) references course(course_id));

```
SQL> create table specification(
  2    specification_id int not null,
  3    specification_name varchar(30),
  4    course_id int not null,
  5    constraint specification_id_pk primary key(specification_id),
  6    constraint course_id_fk3 foreign key(course_id) references course(course_id));

Table created.

SQL> desc specification
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SPECIFICATION_ID                          NOT NULL NUMBER(38)
 SPECIFICATION_NAME                                 VARCHAR2(30)
 COURSE_ID                                 NOT NULL NUMBER(38)
```

Figure 11:create table specification

10. Course_leader

```
SQL> create table course_leader(
  2    qualification varchar(30) not null,
  3    course_id int not null,
  4    instructor_id int not null,
  5    constraint course_id_pk1 primary key(course_id),
  6    constraint course_id_fk2 foreign key(course_id) references course(course_id),
  7    constraint instructor_id_fk1 foreign key(instructor_id) references instructor(instructor_id));

Table created.

SQL> desc course_leader
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 QUALIFICATION                             NOT NULL VARCHAR2(30)
 COURSE_ID                                 NOT NULL NUMBER(38)
 INSTRUCTOR_ID                             NOT NULL NUMBER(38)
```

Figure 12:create table course_leader

11. Instructor_module

create table instructor_module(

instructor_id int not null,

module_id int not null,

constraint instructor_module_pk primary key(module_id,instructor_id),

constraint instructor_id_fk4 foreign key(instructor_id) references instructor(instructor_id),

constraint module_id_fk foreign key(module_id) references Module(module_id));

```
SQL> create table instructor_module(
  2  instructor_id int not null,
  3  module_id int not null,
  4  constraint instructor_module_pk primary key(module_id,instructor_id),
  5  constraint instructor_id_fk4 foreign key(instructor_id) references instructor(instructor_id),
  6  constraint module_id_fk foreign key(module_id) references Module(module_id));

Table created.

SQL> desc instructor_module
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 INSTRUCTOR_ID                             NOT NULL NUMBER(38)
 MODULE_ID                                 NOT NULL NUMBER(38)
```

Figure 13:create table insructor_module

12. Specificationmodule

create table instructor_module(

module_id int not null,

specification_id int not null,

constraint specification_module_pk primary key(module_id,instructor_id),

constraint module_id_fk2 foreign key(module_id) references Module(module_id),

constraint specification_id_fk1 foreign key(specification_id) references

Specification(specification_id));

```
SQL> create table specificationmodule(
  2  module_id int not null,
  3  specification_id int not null,
  4   constraint specification_module_pk primary key(module_id,specification_id),
  5  constraint module_id_fk2 foreign key(module_id) references Module(module_id),
  6   constraint specification_id_fk1 foreign key(specification_id) references Specification(specificati
on_id));

Table created.

SQL> desc specificationmodule
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MODULE_ID                                 NOT NULL NUMBER(38)
 SPECIFICATION_ID                          NOT NULL NUMBER(38)

SQL> select* from tab;
```

Figure 14:create table specificationmodule

6.0. Populating data into the tables

6.1. Commit

Commit is command is used save the tables that we created in SQL.

Inserting values in table

1. Course

into course(course_id,course_name,course_fee,credit)values(1,'BIT',8000000,30)

into course(course_id,course_name,course_fee,credit)values(2,’BBA’,7000000,30)

into course(course_id,course_name,course_fee,credit)values(3,’BBS’,1000000,15)

into course(course_id,course_name,course_fee,credit)values(4,’BSC.CSIT’,1000000,15)

into course(course_id,course_name,course_fee,credit)values(5,’MSc’,4000000,60)

into course(course_id,course_name,course_fee,credit)values(6,’MIT’,9000000,30)

into course(course_id,course_name,course_fee,credit)values(7,’Bhm’,15000000,30)

select*from dual;

```
SQL> insert all
  2    into course(course_id,course_name,course_fee,credit)values(1,’BIT’,8000000,30)
  3    into course(course_id,course_name,course_fee,credit)values(2,’BBA’,7000000,30)
  4    into course(course_id,course_name,course_fee,credit)values(3,’BBS’,1000000,15)
  5    into course(course_id,course_name,course_fee,credit)values(4,’BSC.CSIT’,1000000,15)
  6    into course(course_id,course_name,course_fee,credit)values(5,’MSc’,4000000,60)
  7    into course(course_id,course_name,course_fee,credit)values(6,’MIT’,9000000,30)
  8    into course(course_id,course_name,course_fee,credit)values(7,’Bhm’,15000000,30)
  9    select*from dual;

7 rows created.

SQL> select * from course;

 COURSE_ID COURSE_NAME                    COURSE_FEE     CREDIT
---------- ------------------------------ ---------- ----------
         1 BIT                             8000000         30
         2 BBA                             7000000         30
         3 BBS                             1000000         15
         4 BSC.CSIT                        1000000         15
         5 MSc                             4000000         60
         6 MIT                             9000000         30
         7 Bhm                            15000000         30

7 rows selected.
```

Figure 15:course values

2. Address

```
SQL> insert all
  2   into Address(address_id,province_no,street,city,country,house_number)values(105011,2,'Tarahara','Itahari','Nepal',10)
  3  into Address(address_id,province_no,street,city,country,house_number)values(100232,1,'Dharan','Homes','Nepal',20)
  4  into Address(address_id,province_no,street,city,country,house_number)values(104506,6,'shanti-road','biratnagar','Nepal',50)
  5  into Address(address_id,province_no,street,city,country,house_number)values(102473,3,'mahendra','Damak','Nepal',60)
  6  into Address(address_id,province_no,street,city,country,house_number)values(103784,4,'lekhnath','pathri','Nepal',70)
  7  into Address(address_id,province_no,street,city,country,house_number)values(100945,2,'bhawanipur','Birat-chowk','Nepal',15)
  8  into Address(address_id,province_no,street,city,country,house_number)values(101210,2,'buddha_chowk','dulari','Nepal',16)
  9  select * from dual;

7 rows created.
```

```
SQL> insert all
  2  into Address(address_id,province_no,street,city,country,house_number)values(102245,5,'mahuliya','guugau','Nepal',45)
  3  into Address(address_id,province_no,street,city,country,house_number)values(102256,7,'panmara','guugau','Nepal',45)
  4
SQL> insert all
  2  into Address(address_id,province_no,street,city,country,house_number)values(102245,5,'mahuliya','gudgau','Nepal',45)
  3  into Address(address_id,province_no,street,city,country,house_number)values(102256,7,'panmara','Dhankuta','Nepal',80)
  4  into Address(address_id,province_no,street,city,country,house_number)values(102278,1,'chausathi','Bhojpur','Nepal',101)
  5  into Address(address_id,province_no,street,city,country,house_number)values(102365,2,'khanchi-chowk','sankhasawa','Nepal',102)
  6  into Address(address_id,province_no,street,city,country,house_number)values(102367,4,'jhumka-road','jhumka','Nepal',103)
  7  into Address(address_id,province_no,street,city,country,house_number)values(102370,3,'pakali-road','pakali','Nepal',104)
  8  into Address(address_id,province_no,street,city,country,house_number)values(102375,1,'rangeli-road','rangeli','Nepal',108)
  9  into Address(address_id,province_no,street,city,country,house_number)values(102371,5,'khanar-road','khanar','Nepal',107)
 10  select*from dual;

8 rows created.
```

```
ADDRESS_ID PROVINCE_NO HOUSE_NUMBER                 STREET                         CITY                           COUNTRY
---------- ----------- ------------------------------ ------------------------------ ------------------------------ ------------------------------
    105011           2 10                             Tarahara                       Itahari                        Nepal
    100232           1 20                             Dharan                         Homes                          Nepal
    104506           6 50                             shanti-road                    biratnagar                     Nepal
    102473           3 60                             mahendra                       Damak                          Nepal
    103784           4 70                             lekhnath                       pathri                         Nepal
    100945           2 15                             bhawanipur                     Birat-chowk                    Nepal
    101210           2 16                             buddha_chowk                   dulari                         Nepal
    102245           5 45                             mahuliya                       gudgau                         Nepal
    102256           7 80                             panmara                        Dhankuta                       Nepal
    102278           1 101                            chausathi                      Bhojpur                        Nepal
    102365           2 102                            khanchi-chowk                  sankhasawa                     Nepal

ADDRESS_ID PROVINCE_NO HOUSE_NUMBER                 STREET                         CITY                           COUNTRY
---------- ----------- ------------------------------ ------------------------------ ------------------------------ ------------------------------
    102367           4 103                            jhumka-road                    jhumka                         Nepal
    102370           3 104                            pakali-road                    pakali                         Nepal
    102375           1 108                            rangeli-road                   rangeli                        Nepal
    102371           5 107                            khanar-road                    khanar                         Nepal

15 rows selected.
```

Figure 16:address values

3. Student

insert all

into Student(student_id,name,DOB,gender,marks,joining_date,course_id,address_id)values (301,'Sujan khatri',TO_DATE('13.09.2001','DD.MM.YYYY'),'Male',80,TO_DATE('01.01.2019','DD.MM.YYYY'),6,105011)

into Student(student_id,name,DOB,gender,marks,joining_date,course_id,address_id)values (302,'Jenisha rai',TO_DATE('17.08.2001','DD.MM.YYYY'),'Female',70,TO_DATE('10.01.2019','DD.MM.YYYY'),1,100232)

into
Student(student_id,name,DOB,gender,marks,joining_date,course_id,address_id)values
(303,'Rejina
limbu',TO_DATE('07.08.2000','DD.MM.YYYY'),'Female',70,TO_DATE('15.01.2019','DD.
MM.YYYY'),5,104506)

into
Student(student_id,name,DOB,gender,marks,joining_date,course_id,address_id)values
(304,'Sapana
ghimire',TO_DATE('05.07.2000','DD.MM.YYYY'),'Female',90,TO_DATE('10.02.2019','D
D.MM.YYYY'),7,103784)

into
Student(student_id,name,DOB,gender,marks,joining_date,course_id,address_id)values
(305,'Athartha
giri',TO_DATE('07.08.2000','DD.MM.YYYY'),'Female',80,TO_DATE('12.02.2019','DD.M
M.YYYY'),3,102473)

into
Student(student_id,name,DOB,gender,marks,joining_date,course_id,address_id)values
(306,'Avash
chy',TO_DATE('03.09.2001','DD.MM.YYYY'),'Male',88,TO_DATE('11.02.2020','DD.MM.
YYYY'),1,105011)

into
Student(student_id,name,DOB,gender,marks,joining_date,course_id,address_id)values
(307,'Ganesh
dhakal',TO_DATE('25.08.2003','DD.MM.YYYY'),'Male',99,TO_DATE('01.01.2020','DD.M
M.YYYY'),2,104506)

select* from dual;

```
SQL> insert all
  2  into Student(student_id,name,DOB,gender,marks,joining_date,course_id,address_id)values(301,'Sujan khatri',TO_DATE('13.09.2001','DD.MM.YYYY'),'Male',80,TO_DATE('01.01.2019','DD.MM.YYYY'),6,105011)
  3  into Student(student_id,name,DOB,gender,marks,joining_date,course_id,address_id)values(302,'Jenisha rai',TO_DATE('17.08.2001','DD.MM.YYYY'),'Female',70,TO_DATE('10.01.2019','DD.MM.YYYY'),1,100232)
  4  into Student(student_id,name,DOB,gender,marks,joining_date,course_id,address_id)values(303,'Rejina limbu',TO_DATE('07.08.2000','DD.MM.YYYY'),'Female',70,TO_DATE('15.01.2019','DD.MM.YYYY'),5,104506)
  5  into Student(student_id,name,DOB,gender,marks,joining_date,course_id,address_id)values(304,'Sapana ghimire',TO_DATE('05.07.2000','DD.MM.YYYY'),'Female',90,TO_DATE('10.02.2019','DD.MM.YYYY'),7,103784)
  6  into Student(student_id,name,DOB,gender,marks,joining_date,course_id,address_id)values(305,'Athartha giri',TO_DATE('07.08.2000','DD.MM.YYYY'),'Female',80,TO_DATE('12.02.2019','DD.MM.YYYY'),3,102473)
  7  into Student(student_id,name,DOB,gender,marks,joining_date,course_id,address_id)values(306,'Avash chy',TO_DATE('03.09.2001','DD.MM.YYYY'),'Male',88,TO_DATE('11.02.2020','DD.MM.YYYY'),1,105011)
  8  into Student(student_id,name,DOB,gender,marks,joining_date,course_id,address_id)values(307,'Ganesh dhakal',TO_DATE('25.08.2003','DD.MM.YYYY'),'Male',99,TO_DATE('01.01.2020','DD.MM.YYYY'),2,104506)
  9  select* from dual;

7 rows created.
```

```
SQL> select *from student;

STUDENT_ID NAME                           DOB       GENDER                           MARKS JOINING_D  COURSE_ID ADDRESS_ID
---------- ------------------------------ --------- ------------------------------ ---------- --------- ---------- ----------
       301 Sujan khatri                   13-SEP-01 Male                               80 01-JAN-19          6     105011
       302 Jenisha rai                    17-AUG-01 Female                             70 10-JAN-19          1     100232
       303 Rejina limbu                   07-AUG-00 Female                             70 15-JAN-19          5     104506
       304 Sapana ghimire                 05-JUL-00 Female                             90 10-FEB-19          7     103784
       305 Athartha giri                  07-AUG-00 Female                             80 12-FEB-19          3     102473
       306 Avash chy                      03-SEP-01 Male                               88 11-FEB-20          1     105011
       307 Ganesh dhakal                  25-AUG-03 Male                               99 01-JAN-20          2     104506

7 rows selected.
```

Figure 17:Student values

4. Class

```
insert all
into Class(class_id,class_name,Block)values(501,'Aristotle',A)
into Class(class_id,class_name,Block)values(502,'Newton',B)
into Class(class_id,class_name,Block)values(503,'Hiler',B)
into Class(class_id,class_name,Block)values(504,'Islington',C)
into Class(class_id,class_name,Block)values(505,Cavandish',A)
into Class(class_id,class_name,Block)values(506,'Borom',D)
into Class(class_id,class_name,Block)values(507,'Marshall',B)
select*from dual;
```

```
SQL> insert all
  2  into Class(class_id,class_name,Block)values(501,'Aristotle','A')
  3  into Class(class_id,class_name,Block)values(502,'Newton','B')
  4  into Class(class_id,class_name,Block)values(503,'Hitler','B')
  5  into Class(class_id,class_name,Block)values(504,'Islington','C')
  6  into Class(class_id,class_name,Block)values(505,'Cavendish','A')
  7  into Class(class_id,class_name,Block)values(506,'Borom','D')
  8  into Class(class_id,class_name,Block)values(507,'Marshall','B')
  9  select*from dual;

7 rows created.

SQL> select *from class;

  CLASS_ID CLASS_NAME                     BLOCK
---------- ------------------------------ ------------------------------
       501 Aristotle                      A
       502 Newton                         B
       503 Hitler                         B
       504 Islington                      C
       505 Cavendish                      A
       506 Borom                          D
       507 Marshall                       B

7 rows selected.
```

Figure 18:class values

5. Module

```
SQL> insert all
  2  into Module(module_id,module_name,module_leader,class_id)values(701,'programming',202,503)
  3  into Module(module_id,module_name,module_leader,class_id)values(702,'Modelling and texturing',206,505)
  4  into Module(module_id,module_name,module_leader,class_id)values(703,'Moving image and vfx',206,504)
  5  into Module(module_id,module_name,module_leader,class_id)values(704,'Traveling and Tourism',203,502)
  6  into Module(module_id,module_name,module_leader,class_id)values(705,'Software Engineering',202,504)
  7  into Module(module_id,module_name,module_leader,class_id)values(706,'Thermodynamics',205,501)
  8  into Module(module_id,module_name,module_leader,class_id)values(707,'Economics',204,507)
  9  select*from dual;

7 rows created.

SQL> select *from Module;

 MODULE_ID MODULE_NAME                    MODULE_LEADER   CLASS_ID
---------- ------------------------------ ------------- ----------
       701 programming                               202        503
       702 Modelling and texturing                   206        505
       703 Moving image and vfx                      206        504
       704 Traveling and Tourism                     203        502
       705 Software Engineering                      202        504
       706 Thermodynamics                            205        501
       707 Economics                                 204        507

7 rows selected.

SQL>
```

Figure 19:module values

6. Instructor

```
SQL> insert all
  2  into instructor(instructor_id,firstname,lastname,experience,salary,course_id,address_id)values(201,'Suman','Giri','7 years', 55000,1, 102245)
  3  into instructor(instructor_id,firstname,lastname,experience,salary,course_id,address_id)values(202,'Prajwal','Limbu','2 years', 70000,3,102256)
  4  into instructor(instructor_id,firstname,lastname,experience,salary,course_id,address_id)values(203,'Hemraj','dhakal','20 years', 700000,4,102278)
  5  into instructor(instructor_id,firstname,lastname,experience,salary,course_id,address_id)values(204,'Ganesh','Shrestha','10 years', 50000,6,102365)
  6  into instructor(instructor_id,firstname,lastname,experience,salary,course_id,address_id)values(205,'Manish','Chaudhary','15 years', 90000,2,102367)
  7  into instructor(instructor_id,firstname,lastname,experience,salary,course_id,address_id)values(206,'Aakriti','Chauhan','5 years', 150000,2,102370)
  8  into instructor(instructor_id,firstname,lastname,experience,salary,course_id,address_id)values(207,'Sapana','Chaudhary','7 years', 110000,7,102371)
  9  into instructor(instructor_id,firstname,lastname,experience,salary,course_id,address_id)values(208,'Utsav','madal','7 years', 100000,5,102375)
 10  select*from dual;

8 rows created.

SQL> select *from instructor;

INSTRUCTOR_ID FIRSTNAME                      LASTNAME                       EXPERIENCE                       SALARY  COURSE_ID ADDRESS_ID
------------- ------------------------------ ------------------------------ ------------------------------ ---------- ---------- ----------
          201 Suman                          Giri                           7 years                             55000          1     102245
          202 Prajwal                        Limbu                          2 years                             70000          3     102256
          203 Hemraj                         dhakal                         20 years                           700000          4     102278
          204 Ganesh                         Shrestha                       10 years                            50000          6     102365
          205 Manish                         Chaudhary                      15 years                            90000          2     102367
          206 Aakriti                        Chauhan                        5 years                            150000          2     102370
          207 Sapana                         Chaudhary                      7 years                            110000          7     102371
          208 Utsav                          madal                          7 years                            100000          5     102375

8 rows selected.

SQL>
```

Figure 20:instructor values

7. Fax_address

```
insert all
  into fax_address(fax,address_id)values(2512342,105011)
  into fax_address(fax,address_id)values(3300000,104506)
  into fax_address(fax,address_id)values(6000230,105011)
  into fax_address(fax,address_id)values(0251569,100945)
  into fax_address(fax,address_id)values(5421678,100232)
  into fax_address(fax,address_id)values(0212876,101210)
  into fax_address(fax,address_id)values(1111111,102245)
  into fax_address(fax,address_id)values(8888888,102256)
  into fax_address(fax,address_id)values(7777777,102278)
  into fax_address(fax,address_id)values(5555555,102365)
  into fax_address(fax,address_id)values(8885555,102367)
  into fax_address(fax,address_id)values(7778855,102370)
  into fax_address(fax,address_id)values(4445556,102375)
  into fax_address(fax,address_id)values(1222312,102371)
  select * from dual;
```

```
SQL> insert all
  2  into fax_address(fax,address_id)values(2512342,105011)
  3  into fax_address(fax,address_id)values(3300000,104506)
  4  into fax_address(fax,address_id)values(6000230,105011)
  5  into fax_address(fax,address_id)values(0251569,100945)
  6  into fax_address(fax,address_id)values(5421678,100232)
  7  into fax_address(fax,address_id)values(0212876,101210)
  8  into fax_address(fax,address_id)values(4217648,103784)
  9  select * from dual;

7 rows created.
```

```
SQL> insert all
  2  into fax_address(fax,address_id)values(1111111,102245)
  3  into fax_address(fax,address_id)values(8888888,102256)
  4  into fax_address(fax,address_id)values(7777777,102278)
  5  into fax_address(fax,address_id)values(5555555,102365)
  6  into fax_address(fax,address_id)values(8885555,102367)
  7  into fax_address(fax,address_id)values(7778855,102370)
  8  into fax_address(fax,address_id)values(4445556,102375)
  9  into fax_address(fax,address_id)values(1222312,102371)
 10  select*from dual;

8 rows created.
```

```
SQL> select*from fax_address;

       FAX ADDRESS_ID
---------- ----------
   2512342     105011
   3300000     104506
   6000230     105011
    251569     100945
   5421678     100232
    212876     101210
   4217648     103784
   1111111     102245
   8888888     102256
   7777777     102278
   5555555     102365

       FAX ADDRESS_ID
---------- ----------
   8885555     102367
   7778855     102370
   4445556     102375
   1222312     102371

15 rows selected.
```

Figure 21:fax_address values

8. Phonenumber_address

```
insert all
  into phonenumber_address(phone_number,address_id)values(9815303132,105011)
  into phonenumber_address(phone_number,address_id)values(9845445455,100232)
  into phonenumber_address(phone_number,address_id)values(9800000001,101210)
  into phonenumber_address(phone_number,address_id)values(9800900792,103784)
  into phonenumber_address(phone_number,address_id)values(9810151617,100945)
  into phonenumber_address(phone_number,address_id)values(9845565645,100232)
  into phonenumber_address(phone_number,address_id)values(9878784545,104506)
  into phonenumber_address(phone_number,address_id)values(9800000101,102245)
  into phonenumber_address(phone_number,address_id)values(9810101010,102256)
  into phonenumber_address(phone_number,address_id)values(9820202020,102278)
  into phonenumber_address(phone_number,address_id)values(9814554544,102365)
  into phonenumber_address(phone_number,address_id)values(9877755566,102367)
```

into phonenumber_address(phone_number,address_id)values(9822555555,102370)

into phonenumber_address(phone_number,address_id)values(9812121212,102375)

into phonenumber_address(phone_number,address_id)values(9856565665,102371)

select*from dual;

```
SQL> insert all
  2  into phonenumber_address(phone_number,address_id)values(9815303132,105011)
  3  into phonenumber_address(phone_number,address_id)values(9845445455,100232)
  4  into phonenumber_address(phone_number,address_id)values(9800000001,101210)
  5  into phonenumber_address(phone_number,address_id)values(9800900792,103784)
  6  into phonenumber_address(phone_number,address_id)values(9810151617,100945)
  7  into phonenumber_address(phone_number,address_id)values(9845565645,100232)
  8  into phonenumber_address(phone_number,address_id)values(9878784545,104506)
  9  select*from dual;

7 rows created.
```

```
SQL> insert all
  2  into phonenumber_address(phone_number,address_id)values(9800000101,102245)
  3  into phonenumber_address(phone_number,address_id)values(9810101010,102256)
  4  into phonenumber_address(phone_number,address_id)values(9820202020,102278)
  5  into phonenumber_address(phone_number,address_id)values(9814554544,102365)
  6  into phonenumber_address(phone_number,address_id)values(9877755566,102367)
  7  into phonenumber_address(phone_number,address_id)values(9822555555,102370)
  8  into phonenumber_address(phone_number,address_id)values(9812121212,102375)
  9  into phonenumber_address(phone_number,address_id)values(9856565665,102371)
 10  select*from dual;

8 rows created.

SQL> select*from phonenumber_address;

PHONE_NUMBER ADDRESS_ID
------------ ----------
  9815303132     105011
  9845445455     100232
  9800000001     101210
  9800900792     103784
  9810151617     100945
  9845565645     100232
  9878784545     104506
  9800000101     102245
  9810101010     102256
  9820202020     102278
  9814554544     102365

PHONE_NUMBER ADDRESS_ID
------------ ----------
  9877755566     102367
  9822555555     102370
  9812121212     102375
  9856565665     102371

15 rows selected.

SQL>
```

Figure 22:phonenumber_address values

9. Specification

Insert all
Into
specification(specification_id,specification_name,course_id)values(601,'Acounting',2)
Into
specification(specification_id,specification_name,course_id)values(602,'Masterchef',7)
Into specification(specification_id,specification_name,course_id)values(603,'Finance',3)
Into
specification(specification_id,specification_name,course_id)values(604,'Database',6)

Into
specification(specification_id,specification_name,course_id)values(605,'Neetworking',1)
Into specification(specification_id,specification_name,course_id)values(606,'Physics',5)
Into
specification(specification_id,specification_name,course_id)values(607,'Multimedia',1)
Select*from dual;

```
SQL> Insert all
  2  Into specification(specification_id,specification_name,course_id)values(601,'Acounting',2)
  3  Into specification(specification_id,specification_name,course_id)values(602,'Masterchef',7)
  4  Into specification(specification_id,specification_name,course_id)values(603,'Finance',3)
  5  Into specification(specification_id,specification_name,course_id)values(604,'Database',6)
  6  Into specification(specification_id,specification_name,course_id)values(605,'Neetworking',1)
  7  Into specification(specification_id,specification_name,course_id)values(606,'Physics',5)
  8  Into specification(specification_id,specification_name,course_id)values(607,'Multimedia',1)
  9  Select*from dual;

7 rows created.
```

```
SQL> select*from specification;

SPECIFICATION_ID SPECIFICATION_NAME             COURSE_ID
---------------- ------------------------------ ----------
             601 Acounting                               2
             602 Masterchef                              7
             603 Finance                                 3
             604 Database                                6
             605 Neetworking                             1
             606 Physics                                 5
             607 Multimedia                              1
             608 computing                               1

8 rows selected.

SQL>
```

Figure 23:specification values

10. Specificationmodule

insert all

into specificationmodule(module_id,specification_id)values(702,607)

into specificationmodule(module_id,specification_id)values(706,606)

into specificationmodule(module_id,specification_id)values(707,601)

into specificationmodule(module_id,specification_id)values(703,607)

into specificationmodule(module_id,specification_id)values(705,604)

into specificationmodule(module_id,specification_id)values(701,605)

into specificationmodule(module_id,specification_id)values(704,603)

select*from dual;

```
SQL> insert all
  2  into specificationmodule(module_id,specification_id)values(702,607)
  3  into specificationmodule(module_id,specification_id)values(706,606)
  4  into specificationmodule(module_id,specification_id)values(707,601)
  5  into specificationmodule(module_id,specification_id)values(703,607)
  6  into specificationmodule(module_id,specification_id)values(705,604)
  7  into specificationmodule(module_id,specification_id)values(701,605)
  8  into specificationmodule(module_id,specification_id)values(704,603)
  9  select*from dual;

7 rows created.

SQL> select* from specificationmodule;

 MODULE_ID SPECIFICATION_ID
---------- ----------------
       701              605
       702              607
       703              607
       704              603
       705              604
       706              606
       707              601

7 rows selected.
```

Figure 24:specificationmodule values

11. Instructor_module

Insert all

Into instructor_module(instructor_id,module_id)values(206,701)

Into instructor_module(instructor_id,module_id)values(206,703)

Into instructor_module(instructor_id,module_id)values(203,704)

Into instructor_module(instructor_id,module_id)values(202,705)

Into instructor_module(instructor_id,module_id)values(205,706)

Into instructor_module(instructor_id,module_id)values(204,707)

Into instructor_module(instructor_id,module_id)values(202,701)

Select*from dual;

```
SQL> insert all
  2  into instructor_module(instructor_id,module_id)values(206,701)
  3  into instructor_module(instructor_id,module_id)values(206,703)
  4  into instructor_module(instructor_id,module_id)values(203,704)
  5  into instructor_module(instructor_id,module_id)values(202,705)
  6  into instructor_module(instructor_id,module_id)values(205,706)
  7  into instructor_module(instructor_id,module_id)values(204,707)
  8  into instructor_module(instructor_id,module_id)values(202,701)
  9  select*from dual;

7 rows created.

SQL> select* from instructor_module;

INSTRUCTOR_ID  MODULE_ID
------------- ----------
          202        701
          206        701
          206        703
          203        704
          202        705
          205        706
          204        707

7 rows selected.
```

Figure 25:instructor_module values

12. Course_leader

```
insert all
  into course_leader(qualification,course_id,instructor_id)values('MSc.IT',1,202)
  into course_leader(qualification,course_id,instructor_id)values('Master in commerce',2,201)
  into course_leader(qualification,course_id,instructor_id)values('PHD.IT',3,206)
  into course_leader(qualification,course_id,instructor_id)values('BHM',7,203)
  into course_leader(qualification,course_id,instructor_id)values('PHD.Management',6,204)
  into course_leader(qualification,course_id,instructor_id)values('PHD.IT',4,203)
  into course_leader(qualification,course_id,instructor_id)values('PHD.physics',5,205)
  select*from dual;
```

```
SQL> insert all
  2    into course_leader(qualification,course_id,instructor_id)values('MSc.IT',1,202)
  3    into course_leader(qualification,course_id,instructor_id)values('Master in commerce',2,201)
  4    into course_leader(qualification,course_id,instructor_id)values('PHD.IT',3,206)
  5    into course_leader(qualification,course_id,instructor_id)values('BHM',7,203)
  6    into course_leader(qualification,course_id,instructor_id)values('PHD.Management',6,204)
  7    into course_leader(qualification,course_id,instructor_id)values('PHD.IT',4,203)
  8    into course_leader(qualification,course_id,instructor_id)values('PHD.physics',5,205)
  9    select*from dual;

7 rows created.
```

```
SQL> select *from course_leader;

QUALIFICATION                  COURSE_ID INSTRUCTOR_ID
------------------------------ ---------- -------------
MSc.IT                                  1           202
Master in commerce                      2           201
PHD.IT                                  3           206
BHM                                     7           203
PHD.Management                          6           204
PHD.IT                                  4           203
PHD.physics                             5           205

7 rows selected.

SQL>
```

Figure 26:course_leader values

7.0. Information Queries

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

select student.name, address.*, phonenumber_address.phone_number from student join address on student.address_id = address.address_id join phonenumber_address on phonephonenumber_address.address_id = address.address_id;

```
SQL> select student.name, address.*, phonenumber_address.phone_number from student join address on student.address_id = address.address_id join phonenumber_address on phonenumber_address.address_id = address.add
ress_id;

NAME                      ADDRESS_ID PROVINCE_NO HOUSE_NUMBER              STREET                    CITY                      COUNTRY                   PHONE_NUMBER
------------------------- ---------- ---------- ------------------------- ------------------------- ------------------------- ------------------------- ------------
Sujan khatri                  105011          2 10                        Tarahara                  Itahari                   Nepal                       9815303132
Avash chy                     105011          2 10                        Tarahara                  Itahari                   Nepal                       9815303132
Jenisha rai                   100232          1 20                        Dharan                    Homes                     Nepal                       9845445455
Sapana ghimire                103784          4 70                        lekhnath                  pathri                    Nepal                       9800900792
Jenisha rai                   100232          1 20                        Dharan                    Homes                     Nepal                       9845565645
Rejina limbu                  104506          6 50                        shanti-road               biratnagar                Nepal                       9878784545
Ganesh dhakal                 104506          6 50                        shanti-road               biratnagar                Nepal                       9878784545

7 rows selected.
```

Figure 27:QUERY 1

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

```
select instructor_module.module_id, module.module_name from instructor_module join module on instructor.module_id = module.module_id group by instructor_module.module_id,module.module_name having count(instructor_module.module_id) > 1;
```

```
SQL> select instructor_module.module_id, module.module_name from instructor_module join module on instructor_module.module_id= module.module_id group by instructor_module.module_id,module.module_name having coun
t(instructor_module.module_id) > 1;

 MODULE_ID MODULE_NAME
---------- ------------------------------
       701 programming

SQL>
```

Figure 28:QUERY 2

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

```
select firstname, lastname, salary from instructor where lower(firstname) like '%s%' and salary > 50000;
```

```
SQL> select firstname, lastname, salary from instructor where lower(firstname) like '%s%' and salary > 50000;

FIRSTNAME                     LASTNAME                       SALARY
----------------------------- ------------------------------ ----------
Suman                         Giri                            55000
Manish                        Chaudhary                       90000
Sapana                        Chaudhary                      110000
Utsav                         madal                          100000

SQL>
```

Figure 29:QUERY 3

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

```
select specificationmodule.specification_id, specificationmodule.module_id,
module.module_name from specificationmodule join specification on
specificationmodule.specification_id = specification.specification_id join module on
specificationmodule.module_id = module.module_Id where
lower(specification.specification_name) = 'multimedia';
```

```
SQL> select specificationmodule.specification_id, specificationmodule.module_id, module.module_name from specificationmodule join specification on specificationmodule.specification_id = specification.specificati
on_id join module on specificationmodule.module_id = module.module_Id where lower(specification.specification_name) = 'multimedia';

SPECIFICATION_ID  MODULE_ID MODULE_NAME
---------------- ---------- ------------------------------
             607        702 Modelling and texturing
             607        703 Moving image and vfx

SQL>
```

Figure 30:QUERY 4

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

```
select module.module_leader, instructor.firstname,
phonenumber_address.phone_number from module join instructor on
module.module_leader = instructor.instructor_id join
phonenumber_address on phonenumber_address.address_id =
instructor.address_id  group by module.module_leader,
instructor.firstname, phonenumber_address.phone_number;
```

```
SQL> select module.module_leader, instructor.firstname, phonenumber_address.phone_number from module join instructor on module.module_leader = instructor.instructor_id join phonenumber_address on phonenumber_add
ress.address_id = instructor.address_id  group by module.module_leader, instructor.firstname, phonenumber_address.phone_number;

MODULE_LEADER FIRSTNAME                    PHONE_NUMBER
------------- ------------------------------ ------------
          206 Aakriti                        9800000001
          204 Ganesh                         9800900792
          205 Manish                         9810151617
          202 Prajwal                        9845565645
          202 Prajwal                        9845445455
```

Figure 31:QUERY 5

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

```
select student.name, specification.specification_name from student join course on
student.course_id = course.course_id join specification on specification.course_id =
course.course_id where lower(specification.specification_name) = 'neetworking';
```

```
SQL> select student.name, specification.specification_name from student join course on student.course_id = course.course_id join specification on specification.course_id = course.course_id where lower(specificat
ion.specification_name) = 'neetworking';

NAME                          SPECIFICATION_NAME
----------------------------- ------------------------------
Avash chy                     Neetworking
Jenisha rai                   Neetworking

SQL>
```

Figure 32:QUERY 6

vii. List the fax number of the instructor who teaches the ‘database’ module.

```
select instructor.firstname, fax_address.fax_number from instructor join fax_address on fax_address.address_id= instructor_id.address_id join module on module.module_leader = instructor.instructor_id where lower(module.module_name) = 'database';
```

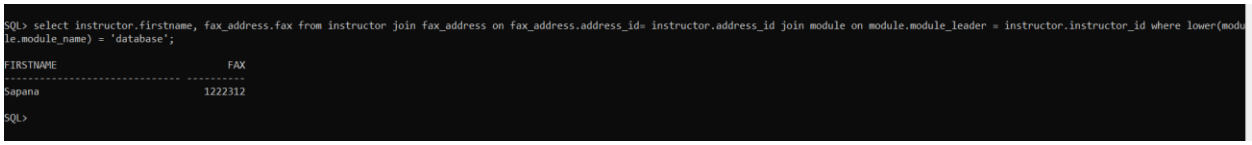

Figure 33:QUERY 7

viii. List the specification falls under the BIT course.

```
select course.course_name, specification.specification_name from specification join course on course.course_id = specification.course_id where upper(course.course_name) = 'BIT';
```

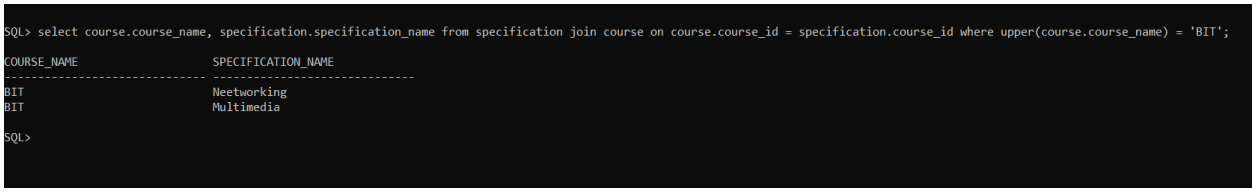

Figure 34:QUERY 8

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

```
select Module.Module_name, Class.Class_name from Module join class on Module.class_id = class.class_id where lower(class.class_name)=’islington’;
```

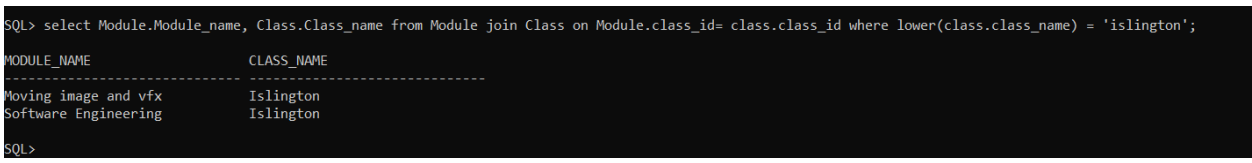

Figure 35:QUERY 9

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

```
select instructor.firstname, address.* from instructor join address on address.address_id = instructor.address_id where lower(instructor.firstname) like '%a';
```

```
SQL> select instructor.firstname, address.* from instructor join address on address.address_id = instructor.address_id where lower(instructor.firstname) like '%a';

FIRSTNAME                      ADDRESS_ID PROVINCE_NO HOUSE_NUMBER                 STREET                         CITY                           COUNTRY
------------------------------ ---------- ---------- ------------------------------ ------------------------------ ------------------------------
Sapana                             102473          3 60                             mahendra                       Damak                          Nepal

SQL>
```

Figure 36:QUERY 10

8.0. Transaction Queries:

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

select student.name, student.course_id, course.course_fees, (course.course_fees * 0.1) as "Discount Amount" from student join course on course.course_id = student.course_id join specification on specification.course_id = course.course_id where lower(specification.specification_name) = 'computing';

```
SQL> select student.name, student.course_id, course.course_fee, (course.course_fee * 0.1) as "Discount Amount" from student join course on course.course_id = student.course_id join specification on specification
.course_id = course.course_id where lower(specification.specification_name) = 'computing';

NAME                       COURSE_ID COURSE_FEE Discount Amount
-------------------------- ---------- ---------- ---------------
Jenisha rai                         1    8000000          800000
Avash chy                           1    8000000          800000

SQL>
```

Figure 37:QUERY 11

ii. 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.

ALTER TABLE phonenumber_address add Contact int default ‘0123456789’;

```
SQL> ALTER TABLE phonenumber_address add Contact int default '0123456789';

Table altered.

SQL> select* from phonenumber_address;

PHONE_NUMBER ADDRESS_ID    CONTACT
------------ ---------- ----------
  9815303132     105011  123456789
  9845445455     100232  123456789
  9800000001     101210  123456789
  9800900792     103784  123456789
  9810151617     100945  123456789
  9845565645     100232  123456789
  9878784545     104506  123456789
  9800000101     102245  123456789
  9810101010     102256  123456789
  9820202020     102278  123456789
  9814554544     102365  123456789

PHONE_NUMBER ADDRESS_ID    CONTACT
------------ ---------- ----------
  9877755566     102367  123456789
  9822555555     102370  123456789
  9812121212     102375  123456789
  9856565665     102371  123456789

15 rows selected.

SQL>
```

Figure 38:QUERY 12

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

select student.name,((sysdate - student.joining_date)/7) as "Enrolled Weeks" from student join course on student.course_id = course.course_id;

```
SQL> select student.name,((sysdate - student.joining_date)/7 ) as "Enrolled Weeks" from student join course on student.course_id = course.course_id;

NAME                         Enrolled Weeks
---------------------------- --------------
Sujan khatri                     102.426718
Jenisha rai                      101.141004
Rejina limbu                     100.426718
Sapana ghimire                   96.7124322
Athartha giri                    96.4267179
Avash chy                        44.4267179
Ganesh dhakal                    50.2838608

7 rows selected.
```

Figure 39:QUERY 13

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

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

select course.course_name, max(student.marks) as "Highest Marks", count(student.course_id) as "No of students" from course join student on student.course_id = course.course_id group by course.course_name;

```
SQL> select course.course_name, max(student.marks) as "Highest Marks", count(student.course_id) as "No of students" from course join student on student.course_id = course.course_id group by course.course_name;

COURSE_NAME                    Highest Marks No of students
------------------------------ ------------- --------------
BIT                                       88              2
BBS                                       80              1
Bhm                                       90              1
MIT                                       80              1
MSc                                       70              1
BBA                                       99              1

6 rows selected.
```

Figure 40:query 15

vi. List all the instructors who are also a course leader**.**

select course_leader.*, instructor.firstname from course_leader join instructor on instructor.instructor_id = course_leader.instructor_id;

```
SQL> select course_leader.*, instructor.firstname from course_leader join instructor on instructor.instructor_id = course_leader.instructor_id;

QUALIFICATION                  COURSE_ID INSTRUCTOR_ID FIRSTNAME
------------------------------ ---------- ------------- ------------------------------
Master in commerce                      2           201 Suman
MSc.IT                                  1           202 Prajwal
PHD.IT                                  4           203 Hemraj
BHM                                     7           203 Hemraj
PHD.Management                          6           204 Ganesh
PHD.physics                             5           205 Manish
PHD.IT                                  3           206 Aakriti

7 rows selected.

SQL>
```

Figure 41:query 16

9.0. Dump file screenshots

```
F:\>EXP sujanchyL2C7/sujancc file = database.dmp

Export: Release 11.2.0.2.0 - Production on Mon Dec 21 11:36:05 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 - Production
Export done in WE8MSWIN1252 character set and AL16UTF16 NCHAR character set
server uses AL32UTF8 character set (possible charset conversion)
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user SUJANCHYL2C7
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user SUJANCHYL2C7
About to export SUJANCHYL2C7's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export SUJANCHYL2C7's tables via Conventional Path ...
. . exporting table                        ADDRESS         15 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                          CLASS          7 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                         COURSE          7 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                  COURSE_LEADER          7 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                   FAX_ADDRESS         15 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                    INSTRUCTOR          8 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table             INSTRUCTOR_MODULE          7 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                         MODULE          8 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table           PHONENUMBER_ADDRESS         15 rows exported
```

```
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                     INSTRUCTOR          8 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table              INSTRUCTOR_MODULE          7 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                        MODULE          8 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table            PHONENUMBER_ADDRESS         15 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                  SPECIFICATION          8 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table            SPECIFICATIONMODULE          7 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                        STUDENT          7 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. 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 with warnings.

F:\>
```

Figure 42:dump file

10.0. Conclusion

It was very wonderful and learning experience while working on this coursework. I tried my best to include all necessary point required for this coursework by researching gorm website, documents, books. While researching it allows me to get extra knowledge for my self-improvement. This project allows me to know about database and how to implement commands on SQL. Many problems arise while doing coursework by help of sir and my research I was able to tackle all the problems. Draw Entity relation Diagram (ERD) in draw.io which I had used before in previous coursework so, I was familiar to the GUI of draw.io no problem comes while drawing ERD. We should have to make two diagrams initial ERD which is drawn before normalization and final ERD is drawn after normalization. After completion of ERD and normalization. Create user in SQL my user name is sujanchyL2C7 and password is Sujancc than create table according to the normalization. In my case before normalization 8 tables and after normalization it become 12 table. And then insert all the data in table which I created according to the queries given in the coursework. Showed table in SQL and then all queries are done during doing queries many problems come by the help of my class fellow I am able to complete only 15 queries and 1 is remaining. During typing commands in SQL if simple comma is not typed it show error due to this problem we should have to retype again. So, to tackle that problem I copy all command in word file if there a mistake than I can direct compy and paste it on SQL. After completion of queries given in the coursework is dump by using command line EXP sujanchyL2C7/sujancc file = database.dmp. At last, it was a great experience doing this coursework it allows me to improve research skills and also interactive skills due to lockdown we are not able to do physical classes so in online class it not that easy to convey exact message what we are saying so there was a many problem but by tackling it I am able to succeed the coursework completely.

11.0. Bibliography

microsoft. (© Microsoft 2020). Retrieved from https://docs.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description

smart draw. (©1994-2020 SmartDraw, LLC). Retrieved from https://www.smartdraw.com/entity-relationship-diagram/

study. (© copyright 2003-2020). Retrieved from study: https://study.com/academy/lesson/what-is-an-entity-in-a-database.html