Module Code & Module Title

CC5051NT Databases

Assessment Weightage & Type

50% Individual Coursework

Year and Semester

2019-20 Autumn

Student Name: Alisha Shrestha

London Met ID: 19033571

Assignment Due Date: 2020-12-21

Assignment Submission Date: 2020-12-21

Table of content

1. Introduction 1
1.1 Introduction of college 1
1.2 Aims and objectives 1
1.3 Current Business Activities and Operations 2
1.4 Current Business Rules 3
1.5 Identification of Entities and Attributes 4
1.5.1. List of the created objects: 4
1.5.2 Identification and representation of the Primary Keys and Foreign Keys 7
1.6 Initial ER-Diagram 10
2. Normalization 11
2.1 UNF: 11
2.2 1NF: 11
2.3 2NF: 12
2.4 3NF 15
3. ER Diagram after Normalization 19
4. Database Implementation 20
4.1 Table Generation 20
4.2 Populating database tables: 32
4.3 Final Tables: 52
5. Database Querying 61
5.1. Information Queries 61
5.2 Transaction Queries 64
5.3 Creation of Dump file 67
5.4 Drop tables 68

6. Conclusion 69

7. Bibliography 71

Table of figure

Figure 1 Initial Entity Relational Diagram 10
Figure 2 ER Diagram after Normalization.......... 19
Figure 3 Creating username, password and successfully connected 20
Figure 4 Creating Address Table.......... 21
Figure 5 Describe table Address 21
Figure 6 Creating table for Person 22
Figure 7 Describe table Person.......... 22
Figure 8 Creating table for Contact.......... 23
Figure 9 Describe table Contact.......... 23
Figure 10 Creating table for Fax 23
Figure 11 Describe table Fax.......... 24
Figure 12 Creating table for Course 24
Figure 13 Describe table Course 24
Figure 14 Creating table for Specification.......... 24
Figure 15 Describe table Specification.......... 25
Figure 16 Creating table for Module 25
Figure 17 Describe table Module 25
Figure 18 4 Creating table for Instructor 25
Figure 19 Describe table Instructor 26
Figure 20 4 Creating table for Student.......... 26
Figure 21 Describe table Student.......... 27
Figure 22 4 Creating table for Instructor_detail 27
Figure 23 Describe table Instructors_detal.......... 27
Figure 24 Creating table for Course_specification 28
Figure 25 Describe table Course_specification.......... 28
Figure 26 4 Creating table for Module_specification.......... 29
Figure 27 Describe table Module_specification 29
Figure 28 4 Creating table for Class.......... 29
Figure 29 Describe table Class 29
Figure 30 4 Creating table for Class_module 30

Figure 31 Describe table Class_module 30
Figure 32 Creating table for Module_head 30
Figure 33 Describe table Module_head 30
Figure 34 Creating table for Course_leader 31
Figure 35 Describe table Course_leader 31
Figure 36 4 Creating table for Personal_info 31
Figure 37 Describe table Personal_info 31
Figure 38 Inserting values into Person table 32
Figure 39 Inserting values into Person table 33
Figure 40 Inserting values into Address table 34
Figure 41 Inserting values into Personal_info table 35
Figure 42 Inserting values into Personal_info table 36
Figure 43 Inserting values into Contact table 37
Figure 44 Inserting values into Contact table 38
Figure 45 Inserting values into Fax table 39
Figure 46 Inserting values into Course table 40
Figure 47 Inserting values into Specification table 41
Figure 48 Inserting values into Module table 42
Figure 49 Inserting values into Module_specification table 43
Figure 50 Inserting values into Class table 44
Figure 51 Inserting values into Class_module table 45
Figure 52 Inserting values into Instructor table 46
Figure 53 Inserting values into Student table 47
Figure 54 Inserting values into Instructor_detail table 48
Figure 55 Inserting values into Course_specification table 49
Figure 56 Inserting values into Course_leader table 50
Figure 57 Inserting values into Module_head table 51
Figure 58 Address table 52
Figure 59 Person table 52
Figure 60 Contact table 53
Figure 61 Personal_info table 53

Figure 62 Course table 54
Figure 63 Specification table 54
Figure 64 Module table 55
Figure 65 Instructor table 55
Figure 66 Student table 56
Figure 67 Instructor_detail table 56
Figure 68 Module_specification table 57
Figure 69 Class table 57
Figure 70 Class_module table 58
Figure 71 Module_head table 58
Figure 72 Course_leader table 59
Figure 73 Course_specification table 59
Figure 74 Fax tab 60
Figure 75 Information Query 1 61
Figure 76 Information Query 3 62
Figure 77 Information Query 4 62
Figure 78 Information Query 5 63
Figure 79 Information Query 6 63
Figure 80 Information Query 7 64
Figure 81 Transaction Query 1 65
Figure 82 Transaction Query 2 65
Figure 83 Transaction Query 3 66
Figure 84 Transaction Query 6 66
Figure 85 Transaction Query 6 67

Table of table

Table 1 Entities and attributes of Person 4
Table 2 Entities and attributes of Address 4
Table 3 Entities and attributes of Contact 5
Table 4 Entities and attributes of Fax 5
Table 5 Entities and attributes of Students 5
Table 6 Entities and attributes of Instructor 6
Table 7 Entities and attributes of Class 6
Table 8 Entities and attributes of Course 6
Table 9 Entities and attributes of Specificatiom 7
Table 10 Entities and attributes of Module 7
Table 11 Representation of Primary key and Foreign Key for Person table 7
Table 12 Representation of Primary key for Address Table 7
Table 13 Representation of Primary key and Foreign Key for Contact 8
Table 14 Representation of Primary key and Foreign Key for Fax 8
Table 15 Representation of Primary key and Foreign Key for Student 8
Table 16 Representation of Primary key and Foreign Key for Instructor 8
Table 17 Representation of Primary key for Class 8
Table 18 Representation of Primary key and Foreign Key for Course 9
Table 19 Representation of Primary key and Foreign Key for Specification 9
Table 20 Representation of Primary key and Foreign Key for Module 9

1. Introduction

1.1 Introduction of college

College is a place which holds one of the most memorable years of one's life and is entirely different from school. Itahari International College was established in 2017 which was the first college to provide UK university degree in Itahari as it has directly partnered with London Metropolitan University and it is also one of the leading institute of Innovate Nepal Group (ING). The college provides degrees like BSc (Hons) Computing and BA (Hons) Business Administration. This college is known for its different method of teaching unlike other colleges which provides same education. IIC had a teaching style which was based on LTW where 'L' stands for lecture, 'T' stands for tutorial and 'W' stands for workshop.

The college does not only focus on written exams but it rather gives more emphasis to particle assignments or presentation which directly and indirectly helps the student to work in group, gain confidence and interact easily. The college manages different activities during the college time for instance, sports, talk series, Hult prize etc., which makes the student active and motivated one way or the other.

The college mainly focuses on the students' future and prepares the students to be able to compete in their sectors as well as making the students independent with all the skills. The other fascinating factor about the college is the teachers, the tutors of the college are academically very qualified and they are always free to help without denying any students. This college is fully based on digitalized education system so it no longer provides boring learning experiences which made the study more interesting.

1.2 Aims and objectives

i Aims

- To challenge every students to reach the highest level of achievement possible.
- To encourage each and every student to have a knowledge and research regarding the study.
- To develop creative thinking and positive attitudes towards people.
- To introduce students with valuable finding and learning experiences.
- To provide peaceful and knowledgeable environment for the students.

ii Objective

The long term goals are always followed by certain objectives in an institute where objectives are the small ways and steps of fulfilling the goals. The college has set number of objectives which leads towards the achievement of the final goal. Some of the objectives are given below:

- To increase student involvement in academic as well as extra activities.
- To improve and upgrade faculty when required by providing special training events to teachers.
- To provide the students with better library facilities for their research.
- To facilitate students and teachers with national and international seminars for assessment techniques.
- To provide students with extra opportunities for their better learning and gaining experience.

1.3 Current Business Activities and Operations

This is a well-known institute with a very advance technology of teaching environment so, why would this college follow the traditional way of keeping the records of their institute therefore, the data and information related to either finance or personal details of the students are stored in an organized way in a database.

The college entirely follows the digitalized form of teaching and carrying out their own business activities. Managing an educational system requires careful planning and time management so college moving from a traditional paper-based data management system to digital and automated system allows to saves a lot of time and hardcopies. It maintains records related to every students and instructors from fees, financial records, salary, Degrees etc.

Using database for storing all sorts of data allows college to stay updated which makes decision making process faster for the college administration. The college provides four courses where two provides IT degree and the other two provides Business Degree. The courses available are BBA and BIT which has further more specifications like computing, marketing, networking etc. Here in this college the course specifications contains several module like in other college there are subjects so each module are taught in classes by the instructors and the other factor about this college is that it’s based on LTW where the same module is taught in three different ways in each class of the week. There are number of classes for Lecture, tutorial and a Lab for workshop.

An instructor of the college not only teaches the module of the course but is able to be a course leader too. Currently the college is not only focusing on the studies but it is also letting the students to participate in different national and international competition.

1.4 Current Business Rules

The college with huge number of students must have important set of rules for maintaining the large number of records without any complications, so for that reason college has set number of rules for proper management of records and the rules are:

- The college system must keep track of address of all the associated people.
- The data must also contain one mailing address of the people.
- Address detail of student and instructor must contain country, province, city, street, house number, phone number, fax number.
- Each student and instructors must provide personal details.
- Instructor must provide valid degree certificate.
- Each student can only be enrolled in one course.
- One instructor can teach only one course.
- One instructor is able to take classes of different module at a time.
- Courses of the college have credit hour.
- Each class has limited number of students according to LTW.
- Among each course only one instructor of the same course can be the course leader.

1.5 Identification of Entities and Attributes

1.5.1. List of the created objects:

i Person:

Table 1 Entities and attributes of Person

Person
Id_number
First_name
Last_name
Gender
DOB
Address_no

ii Address:

Table 2 Entities and attributes of Address

Address
Address_no
House_no
Country
Province
Street
City

iii Contact:

Table 3 Entities and attributes of Contact

Contact
<u>Phone_number</u>
Address_no*

iv Fax:

Table 4 Entities and attributes of Fax

Fax
<u>Fax_no</u>
Address_no*

v Student:

Table 5 Entities and attributes of Students

Student
<u>Enrollment_id</u>
Age
Enrollment_date
Email_id
Id_number*
Mark_obt

vi Instructor:

Table 6 Entities and attributes of Instructor

Instructor
<u>Instructor_id</u>
Salary
Degree
Course_id*
Id_number*

vii Class:

Table 7 Entities and attributes of Class

Class
<u>Classroom_no</u>
LTW
No_of_Std

viii Course:

Table 8 Entities and attributes of Course

Course
<u>Course_id</u>
Course_name
No_of_specification
Credit_hour
Course_leader*

ix Specification:

Table 9 Entities and attributes of Specificatiom

Specification
<u>Specification_code</u>
Specific_name
Course_id*
Fee

x Module

Table 10 Entities and attributes of Module

Module
<u>Module_id</u>
Module_name
Specification_code*
Module_leader*
Classromm_no*

1.5.2 Identification and representation of the Primary Keys and Foreign Keys

I. Person:

Table 11 Representation of Primary key and Foreign Key for Person table

Entity Name	Primary Key	Foreign Key	Reference Table
Person	Id_number	Address_no	Address

II. Address:

Table 12 Representation of Primary key for Address Table

.

Entity Name	Primary Key
Address	Address_no

III. Contact:

Table 13 Representation of Primary key and Foreign Key for Contact

Entity Name	Primary Key	Foreign Key	Reference Table
Contact	Phone_no	Address_no	Address

IV. Fax:

Table 14 Representation of Primary key and Foreign Key for Fax

Entity Name	Primary Key	Foreign Key	Reference Table
Fax	Fax_no	Address_no	Address

V. Student:

Table 15 Representation of Primary key and Foreign Key for Student

Entity Name	Primary Key	Foreign Key	Reference Table
Student	Enrollment_id	Id_number	Person

VI. Instructor:

Table 16 Representation of Primary key and Foreign Key for Instructor

Entity Name	Primary Key	Foreign Key	Reference Table
Instructor	Instructor_id	Id_number, Course_id	Person Course

VII. Class:

Table 17 Representation of Primary key for Class

Entity Name	Primary Key
Class	Classroom_no

VIII. Course:

Table 18 Representation of Primary key and Foreign Key for Course

Entity Name	Primary Key	Foreign Key	Reference Table
Course	Course_id	Course_leader	Instructor

IX. Specification:

Table 19 Representation of Primary key and Foreign Key for Specification

Entity Name	Primary Key	Foreign Key	Reference Table
Specification	Specification_code	Course_id	Course

X. Module:

Table 20 Representation of Primary key and Foreign Key for Module

Entity Name	Primary Key	Foreign Key	Reference Table
Module	Module_code	Specificaction_code Module_leader Classroom_no	Specification Instructor Class

1.6 Initial ER-Diagram

Entity Relation Diagram is a graphical representation of structural diagram for using it in a database design. ERD shows entities sets the stored data in a database and relation between them. An entity set is a collection of similar entities. ERD provide visual starting point of databases design that can also be used to help determined information system requirements throughout an organization. ERD is useful for organizing data that can be represented by a relation structural. Database helps in creating relation database diagram in a simple way. (Silberschatz, Silberchatz, & Korth, 2005)

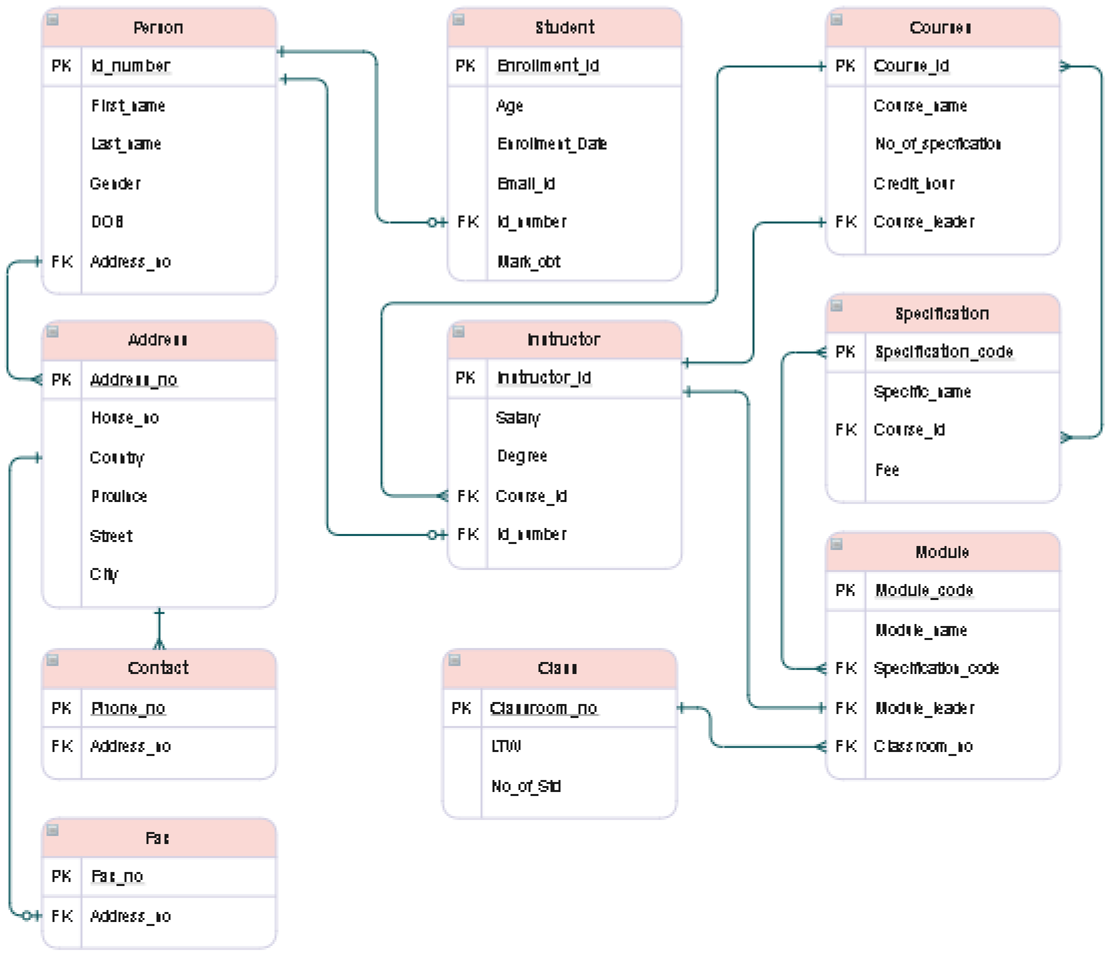


Figure 1 Initial Entity Relational Diagram

The above illustration of ER-diagram has many data anomalies and redundancies were the anomalies are update, deletion and insertion and data redundancies is a condition created in a database when same data of the single person is stored in multiple location. The relationships between entities are also not properly managed. In the above ERD the course entity has many specifications and within the specification there are many modules so it creates a data redundancy and to avoid the anomalies and data redundancies normalization is done.

2. Normalization

Normalization is the process of organizing the data in the database to reduce the redundancy from a relation and is used to eliminate anomalies. In the process of normalization it divides the table into smaller table and link them using relationship. (Guru99)

The types of normalization are:

- **UNF:** UNF stands for un-normalized form of the normalization where data are arranged by separating the repeating group in a single table.
- **1NF:** It stands for first normal form where attributes of the table cannot hold multiple values.
- **2NF:** It stands for second normal form of the normalization where all the non-primary attributes must be fully functionally dependent on prime key attributes.
- **3NF:** It stands for third normal form where it must be second normal form and no non-prime attributes is transitively dependent on primary key attribute. (Guru99)

2.1 UNF:

Person{Id_number, First_name, last_name, Gender, DOB, {Address_no, House_no, Country, Province, Street, City, {Phone_number} , {Fax_number}} , {Enrollment_id, Age, Email_id, Enrollment_day, Mark_obt} , {Instructor_id, Salary, Degree, {Course_id, Course_name, No_of_specification, Credit_hours {Specification_code, Specfic_name, Fee, {Module_code, Module_name, {Classroom_no, LTW, No_of_std}}}}} }.

In UNF the data are arranged in group in a single table by separating

2.2 1NF:

Separating repeating and non-repeating group

Person{Id_number, First_name, Last_name, Gender, DOB}

Address{Address_no, House_no, Country, Province, Street, City, Id_number *}

Contact{Phone_no, Address_no* }

Fax{Fax_no, Address_no*}

Student{Enrollment_id, Age , Email_id, Id_number*, Enrollment_day, Mark_obt}

Instructor{Instructor_id, Id_number*, Salary, Degree, Module_code*}

Course{Course_id, Course_name, No_of_specification, Credit_hour, Course_leader*}

Specification{Specification_code, Course_id*, Specific_name, Fee}

Module{Module_code, Specification_code*, Module_name}

Class{Classroom_no, Module_code*, LTW, No_of_std}

2.3 2NF:

Checking Full Functional and Partial Dependency

Person{Id_number, First_name, Last_name, Gender, DOB}

Student{Enrollment_id, Age , Email_id, Id_number*, Enrollment_day, Mark_obt}

Course{Course_id, Course_name, No_of_specification, Credit_hour, Course_leader*}

Contact{Phone_no, Address_no* }

Fax{Fax_no, Address_no*}

For Address

Address_no ------> House_no, Country, Province, Street, City

Id_number ------>

Address_no, Id_number ------>

Address{Address_no, House_no, Country, Province, Street, City}

Person_info{Mail_address*, Id_number*}

For Instructor

Instructor_id ------> Salary, Degree, Id_number

Module_code ------>

Instructor_id, Module_code ------>

Instructor{Instructor_id,Id_number*, Salary, Degree}

Instructor_details{Instructor_id*, Module_code*}

For Specification:

Specification_code ------> Specific_name, Fee

Course_id ------>

Specification_code, Course_id ------>

Specification{Specification_code, Specific_name, Fee}

Course_Specification{Specification_code*, Course_id*}

For Module

Module_code ------> Module_name

Specification_code ------>

Module_code, Specification_code ------>

Module{Module_code, Module_name}

Module_specification{Module_code, Specification_code*}

For Class

Classroom_no ------> LTW, No_no_std

Module_code ------>

Classroom_no, Module_code ------>

Class{Classroom_no, LTW, No_no_std}

Class_module{Classroom_no, Module_code*}

2NF

Person{Id_number, First_name, Last_name, Gender, DOB}

Student{Enrollment_id, Age , Email_id, Id_number*, Enrollment_day, Mark_obt}

Course{Course_id, Course_name, No_of_specification, Credit_hour, Course_leader*}

Contact{Phone_no, Address_no* }

Fax{Fax_no, Address_no*}

Address{Address_no, House_no, Country, Province, Street, City}

Person_info{Mail_address*, Id_number*}

Instructor{Instructor_id, Id_number*, Salary, Degree}

Instructor_details{Instructor_id*, Module_code*}

Specification{Specification_code, Specific_name, Fee}

Course_Specification{Specification_code*, Course_id*}

Module{Module_code, Module_name}

Module_specification{Module_code, Specification_code*}

Class{Classroom_no, LTW, No_no_std}

Class_module{Classroom_no, Module_code*}

Module_head{Module_code*, Head*}

2.4 3NF

Checking Transitive Dependency

Contact{Phone_no, Address_no* }

Fax{Fax_no, Address_no*}

Person_info{Mail_address*, Id_number*}

Instructor_details{Instructor_id*, Module_code*}

Course_Specification{Specification_code*, Course_id*}

Module_specification{Module_code, Specification_code*}

Class_module{Classroom_no, Module_code*}

Specification{Specification_code, Specific_name, Fee}

Module{Module_code, Module_name}

Class{Classroom_no, LTW, No_no_std}

Module_head{Module_code*, Head*}

For Instructor

Instructor_id ------> id_number ------>

Instructor_id ------> Salary------>

Instructor_id ------> Degree------>

Instructor{Instructor_id, Id_number* ,Salary, Degree}

For Person

Id_number ------> First_name ------>

Id_number ------> Last_name ------>

Id_number ------> Gender ------>

Id_number ------> DOB ------>

Person{Id_number, First_name, Last_name, Gender, DOB}

For Student

Enrollment_id ------> Age ------>

Enrollment_id ------> Email_id ------>

Enrollment_id ------> Id_number ------>

Enrollment_id ------> Enrollement_day ------>

Enrollement_id ------> Mark_obt ------>

Student{Enrollment_id, Age , Email_id, Id_number*, Enrollment_day, Mark_obt}

For Course

Course_id ------> Course_name ------>

Course_id ------> No_of_specification ------>

Course_id ------> Credit_hour ------>

Course_id ------> Course_leader ------>

Course{Course_id, Course_name, No_of_specification, Credit_hour, Course_leader*}

For Address

Address_no ------> House_no ------>

Address_no ------> Country ------>

Address_no ------> Province ------>

Address_no ------> Street ------>

Address_no ------> City ------>

Address{Address_no, House_no, Country, Province, Street, City}

3NF

Contact{Phone_no, Address_no* }

Fax{Fax_no, Address_no*}

Person_info{Mail_address*, Id_number*}

Instructor_details{Instructor_id*, Module_code*}

Course_Specification{Specification_code*, Course_id*}

Module_specification{Module_code, Specification_code*}

Class_module{Classroom_no, Module_code*}

Instructor{Instructor_id, Id_number*, Salary, Degree, Course_id*, Specification_code*}

Specification{Specification_code, Specific_name, Fee}

Module{Module_code, Module_name}

Class{Classroom_no, LTW, No_no_std}

Module_head{Module_code*, Head*}

Person{Id_number, First_name, Last_name, Gender, DOB}

Student{Enrollment_id, Age , Email_id, Id_number*, Course_id*, Specification_code*, Enrollment_day, Mark_obt}

Course{Course_id, Course_name, No_of_specification, Credit_hour}

Address{Address_no, House_no, Country, Province, Street, City}

Course_leader{Course_id*, Course_leader*}

Assumptions:

- One Address can have multiple Contact number.
- Address can have either one or no Fax.
- Person can either be instructor or student but not both.
- One instructor can teach many modules and many modules can be taught by one Instructor.
- One instructor can teach only one course.
- One instructor can be the course leader of only one course.
- One course can have many specifications.
- Specification contains many Modules.
- One module has one module head and instructor can be the module head of any one module.
- One class can have many modules.

3. ER Diagram after Normalization

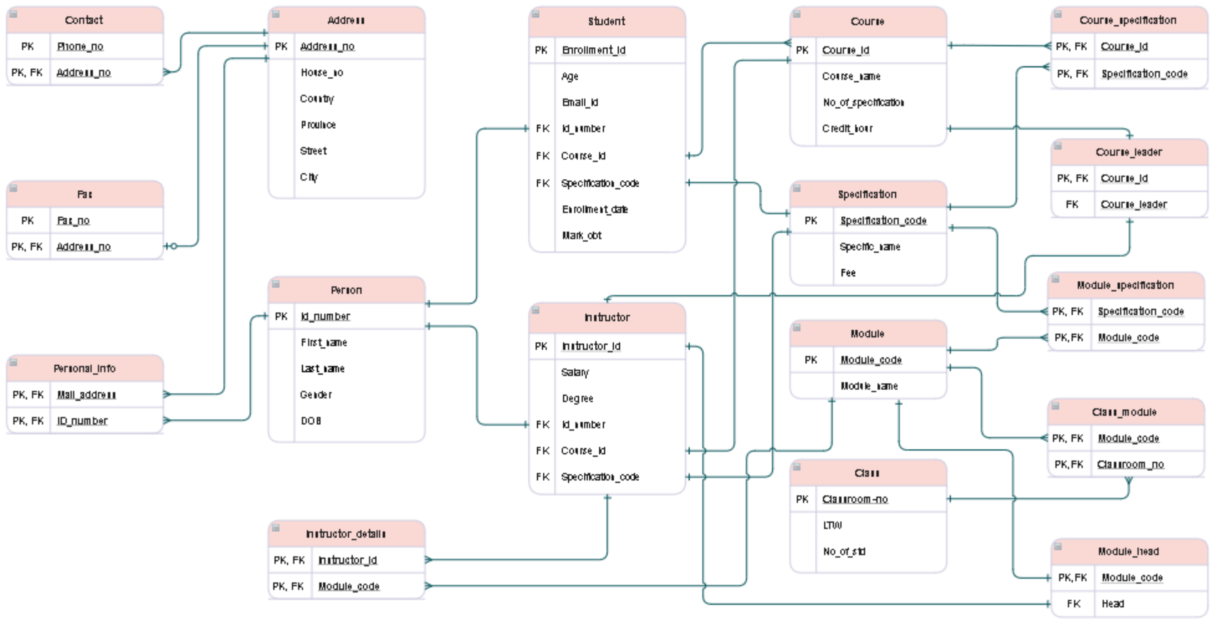
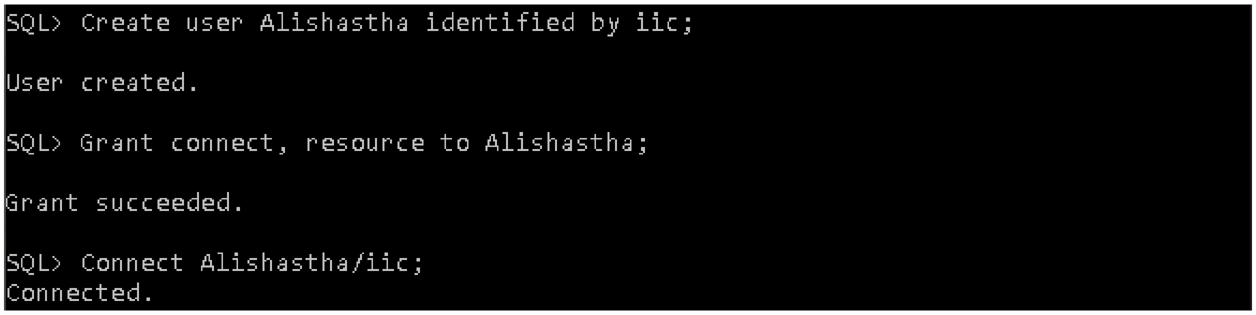


Figure 2 ER Diagram after Normalization

4. Database Implementation

4.1 Table Generation

For creating a new table in database, 'CREATE TABLE' command is used and it requires details like name of the table, data type etc.

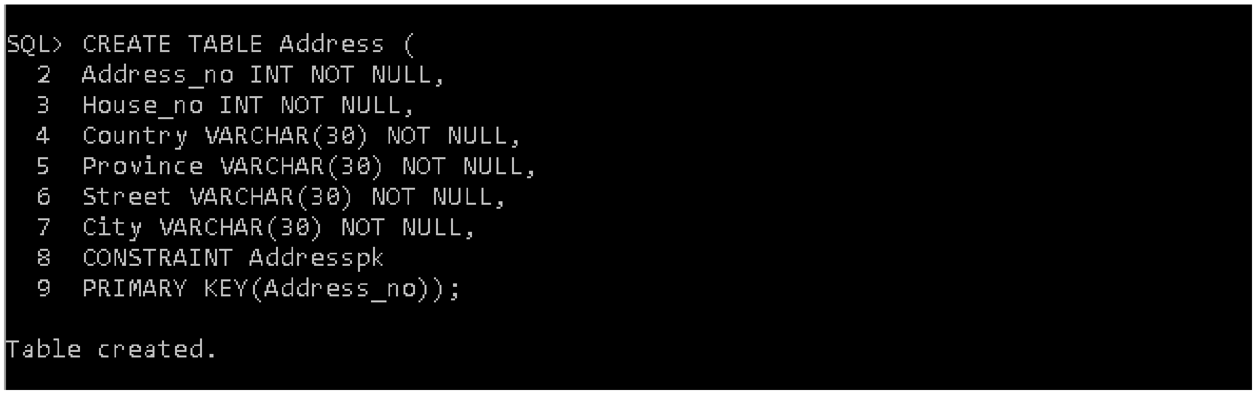


```
SQL> Create user Alishastha identified by iic;

User created.

SQL> Grant connect, resource to Alishastha;

Grant succeeded.

SQL> Connect Alishastha/iic;
Connected.
```

Figure 3 Creating username, password and successfully connected

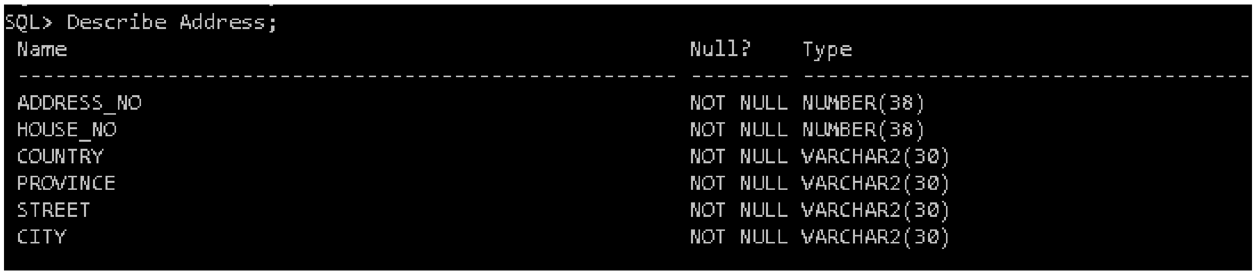
- **Creating table for Address:**

```
CREATE TABLE Address (
Address_no INT NOT NULL,
House_no INT NOT NULL,
Country VARCHAR(30) NOT NULL,
Province VARCHAR(30) NOT NULL,
Street VARCHAR(30) NOT NULL,
City VARCHAR(30) NOT NULL,
CONSTRAINT Addresspk
PRIMARY KEY(Address_no));
```



```
SQL> CREATE TABLE Address (
  2  Address_no INT NOT NULL,
  3  House_no INT NOT NULL,
  4  Country VARCHAR(30) NOT NULL,
  5  Province VARCHAR(30) NOT NULL,
  6  Street VARCHAR(30) NOT NULL,
  7  City VARCHAR(30) NOT NULL,
  8  CONSTRAINT Addresspk
  9  PRIMARY KEY(Address_no));

Table created.
```

Figure 4 Creating Address Table

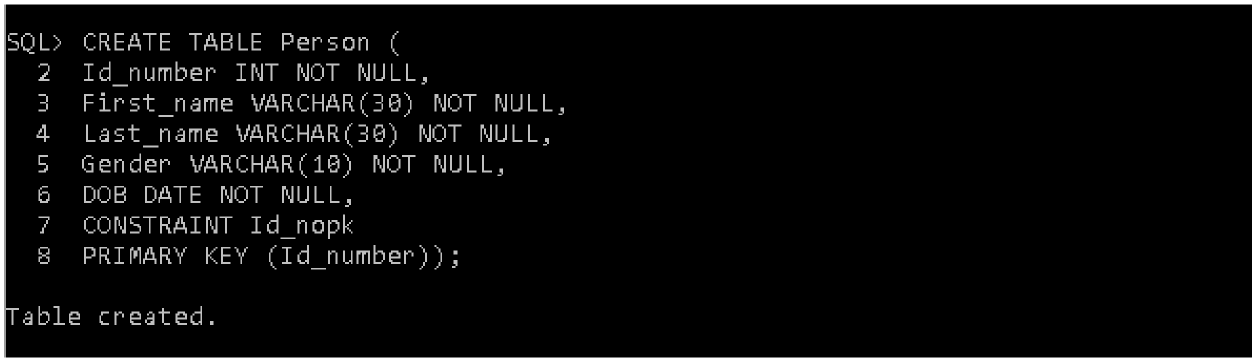


```
SQL> Describe Address;
 Name                                              Null?    Type
 ------------------------------------------------- -------- ----------------------------------
 ADDRESS_NO                                        NOT NULL NUMBER(38)
 HOUSE_NO                                          NOT NULL NUMBER(38)
 COUNTRY                                           NOT NULL VARCHAR2(30)
 PROVINCE                                          NOT NULL VARCHAR2(30)
 STREET                                            NOT NULL VARCHAR2(30)
 CITY                                              NOT NULL VARCHAR2(30)
```

Figure 5 Describe table Address

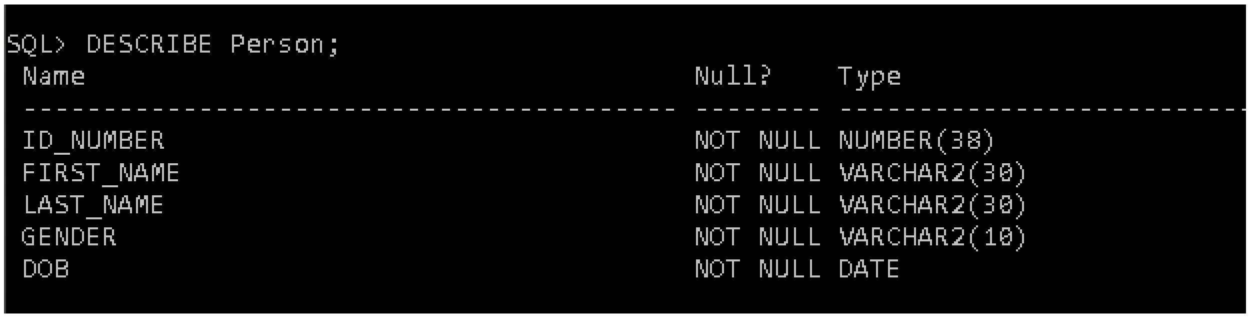
- **Creating table for Person:**

```
Id_number INT NOT NULL,
First_name VARCHAR(30) NOT NULL,
Last_name VARCHAR(30) NOT NULL,
Gender VARCHAR(10) NOT NULL,
DOB DATE NOT NULL,
CONSTRAINT Id_nopk
PRIMARY KEY (Id_number));
```



```
SQL> CREATE TABLE Person (
  2  Id_number INT NOT NULL,
  3  First_name VARCHAR(30) NOT NULL,
  4  Last_name VARCHAR(30) NOT NULL,
  5  Gender VARCHAR(10) NOT NULL,
  6  DOB DATE NOT NULL,
  7  CONSTRAINT Id_nopk
  8  PRIMARY KEY (Id_number));

Table created.
```

Figure 6 Creating table for Person

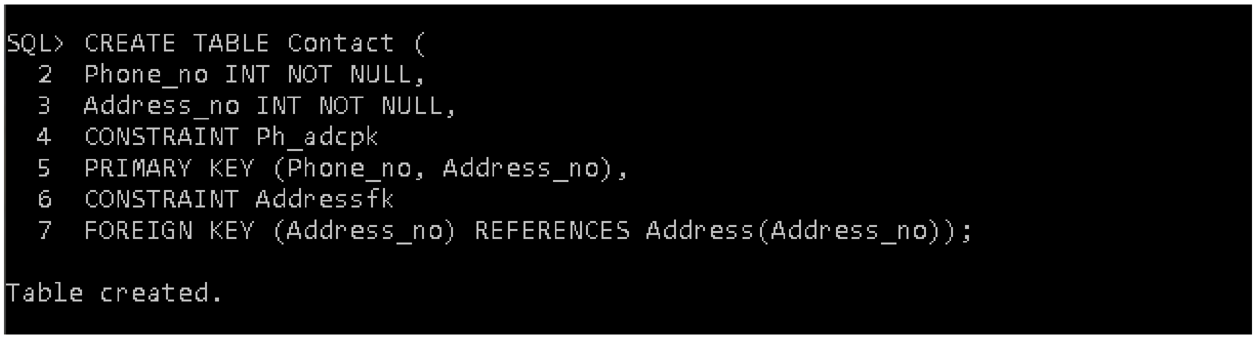


```
SQL> DESCRIBE Person;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ID_NUMBER                                 NOT NULL NUMBER(38)
 FIRST_NAME                                NOT NULL VARCHAR2(30)
 LAST_NAME                                 NOT NULL VARCHAR2(30)
 GENDER                                    NOT NULL VARCHAR2(10)
 DOB                                       NOT NULL DATE
```

Figure 7 Describe table Person

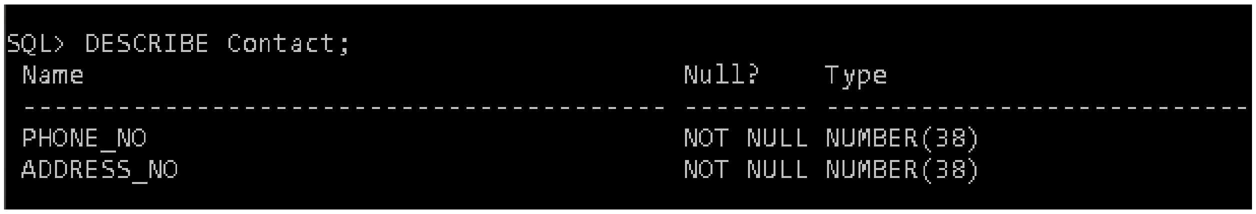
- **Creating table for Contact:**

```
CREATE TABLE Contact (

Phone_no INT NOT NULL,

Address_no INT NOT NULL,

CONSTRAINT Ph_adcpk

PRIMARY KEY (Phone_no, Address_no),

CONSTRAINT Addressfk

FOREIGN KEY (Address_no) REFERENCES Address(Address_no));
```



```
SQL> CREATE TABLE Contact (
  2  Phone_no INT NOT NULL,
  3  Address_no INT NOT NULL,
  4  CONSTRAINT Ph_adcpk
  5  PRIMARY KEY (Phone_no, Address_no),
  6  CONSTRAINT Addressfk
  7  FOREIGN KEY (Address_no) REFERENCES Address(Address_no));

Table created.
```

Figure 8 Creating table for Contact

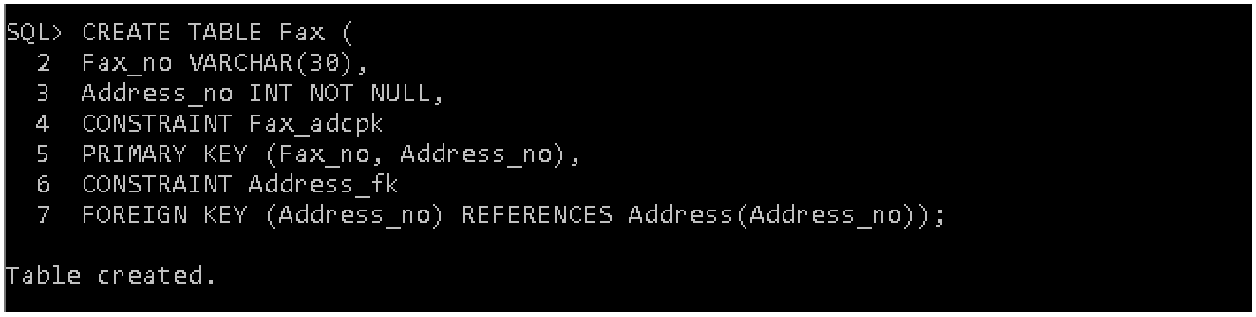


```
SQL> DESCRIBE Contact;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PHONE_NO                                  NOT NULL NUMBER(38)
 ADDRESS_NO                                NOT NULL NUMBER(38)
```

Figure 9 Describe table Contact

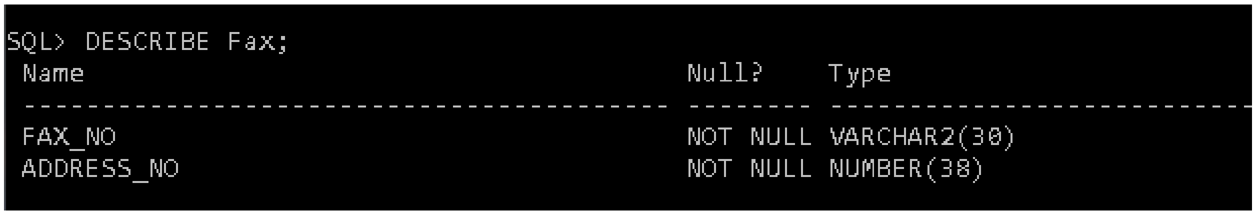
- **Creating table for Fax:**

```
CREATE TABLE Fax (

Fax_no VARCHAR(30),

Address_no INT NOT NULL,

CONSTRAINT Fax_adcpk

PRIMARY KEY (Fax_no, Address_no),

CONSTRAINT Address_fk

FOREIGN KEY (Address_no) REFERENCES Address(Address_no));
```



```
SQL> CREATE TABLE Fax (
  2  Fax_no VARCHAR(30),
  3  Address_no INT NOT NULL,
  4  CONSTRAINT Fax_adcpk
  5  PRIMARY KEY (Fax_no, Address_no),
  6  CONSTRAINT Address_fk
  7  FOREIGN KEY (Address_no) REFERENCES Address(Address_no));

Table created.
```

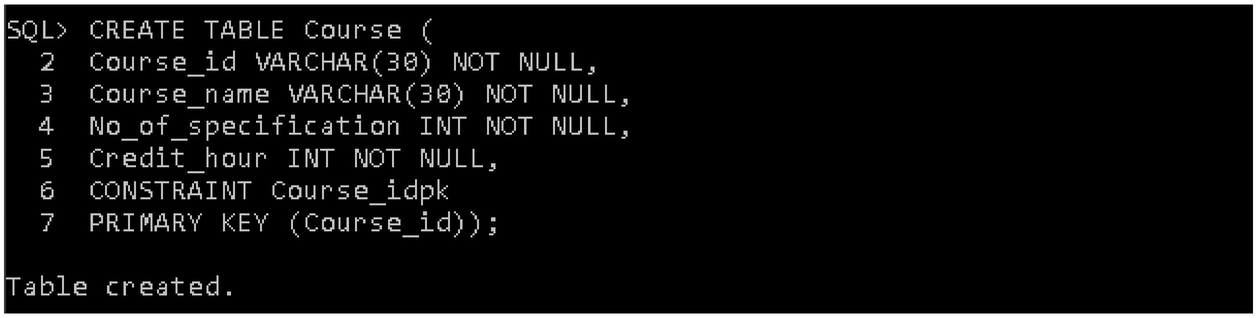
Figure 10 Creating table for Fax



```
SQL> DESCRIBE Fax;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 FAX_NO                                    NOT NULL VARCHAR2(30)
 ADDRESS_NO                                NOT NULL NUMBER(38)
```

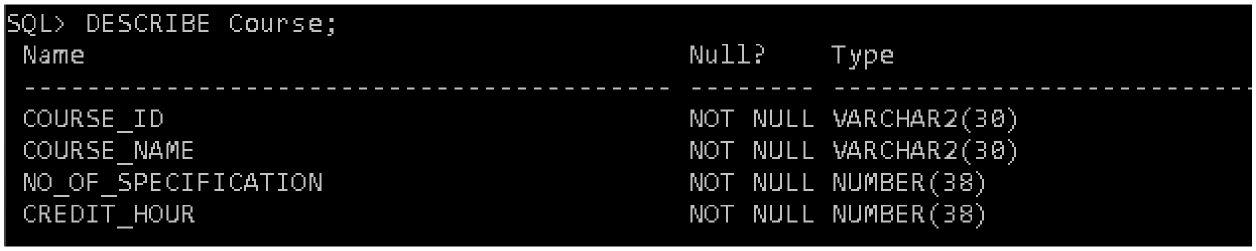
Figure 11 Describe table Fax

- **Creating table for Course**



```
SQL> CREATE TABLE Course (
  2  Course_id VARCHAR(30) NOT NULL,
  3  Course_name VARCHAR(30) NOT NULL,
  4  No_of_specification INT NOT NULL,
  5  Credit_hour INT NOT NULL,
  6  CONSTRAINT Course_idpk
  7  PRIMARY KEY (Course_id));

Table created.
```

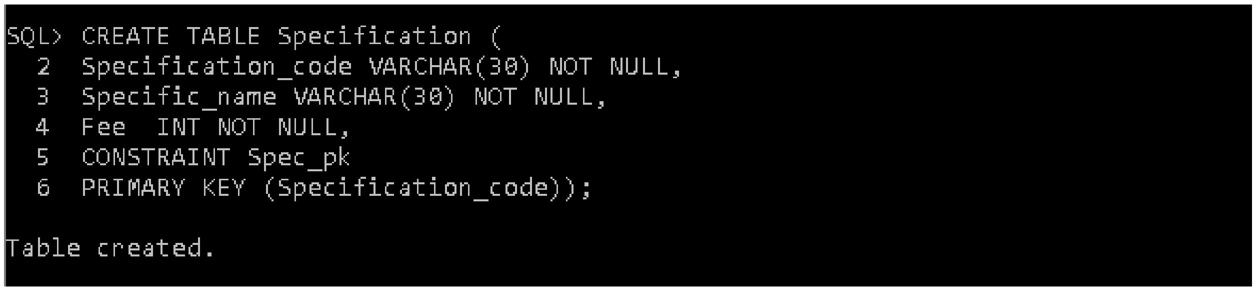
Figure 12 Creating table for Course



```
SQL> DESCRIBE Course;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COURSE_ID                                 NOT NULL VARCHAR2(30)
 COURSE_NAME                               NOT NULL VARCHAR2(30)
 NO_OF_SPECIFICATION                       NOT NULL NUMBER(38)
 CREDIT_HOUR                               NOT NULL NUMBER(38)
```

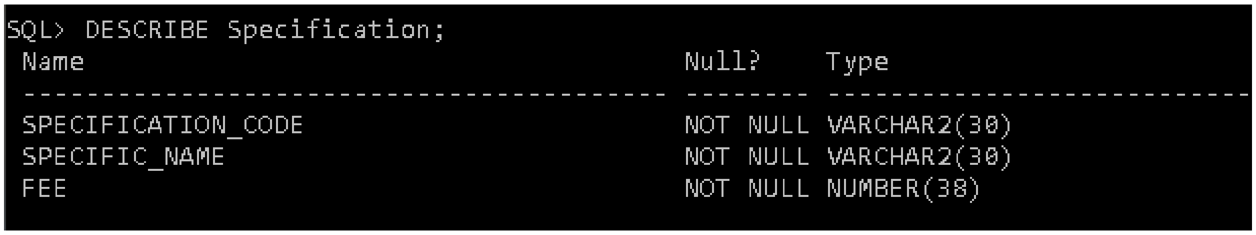
Figure 13 Describe table Course

- **Creating table for Specification:**



```
SQL> CREATE TABLE Specification (
  2  Specification_code VARCHAR(30) NOT NULL,
  3  Specific_name VARCHAR(30) NOT NULL,
  4  Fee  INT NOT NULL,
  5  CONSTRAINT Spec_pk
  6  PRIMARY KEY (Specification_code));

Table created.
```

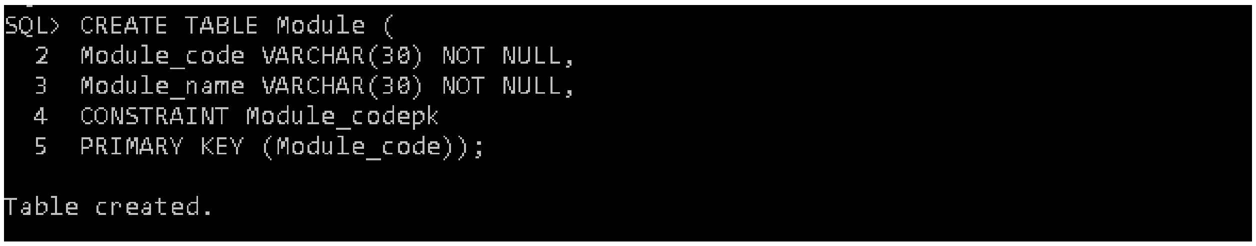
Figure 14 Creating table for Specification



```
SQL> DESCRIBE Specification;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SPECIFICATION_CODE                        NOT NULL VARCHAR2(30)
 SPECIFIC_NAME                             NOT NULL VARCHAR2(30)
 FEE                                       NOT NULL NUMBER(38)
```

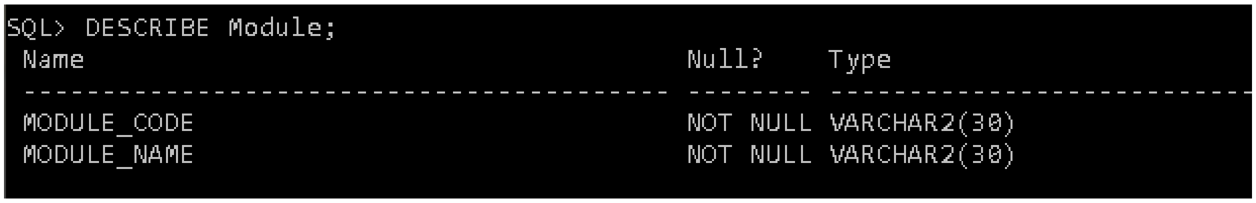
Figure 15 Describe table Specification

- **Creating table for Module:**



```
SQL> CREATE TABLE Module (
     Module_code VARCHAR(30) NOT NULL,
     Module_name VARCHAR(30) NOT NULL,
     CONSTRAINT Module_codepk
     PRIMARY KEY (Module_code));

Table created.
```

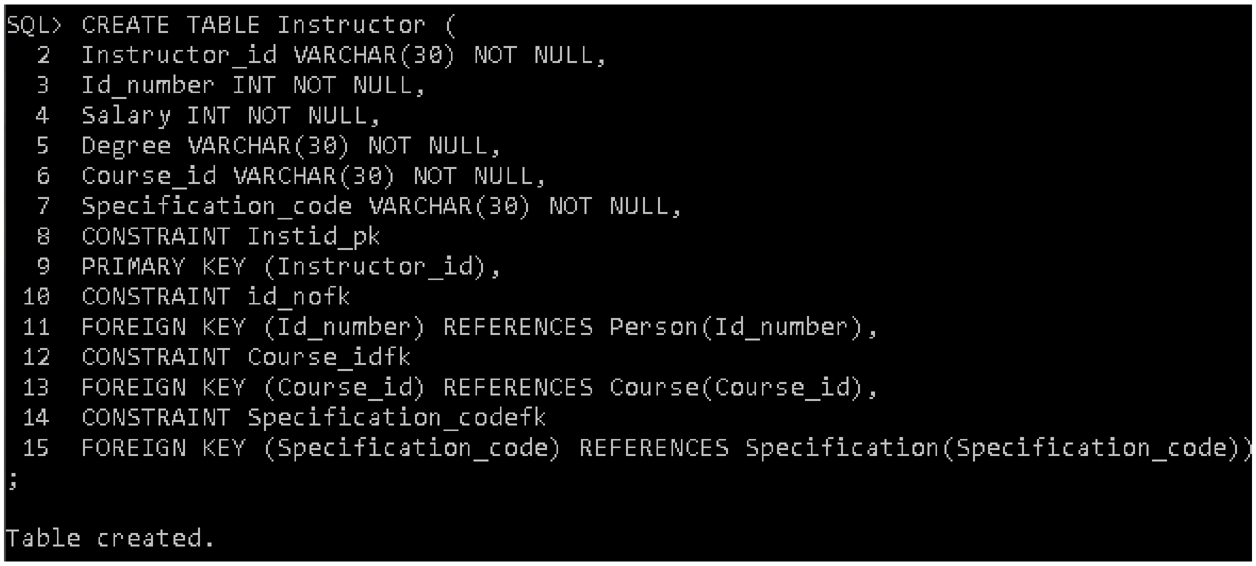
Figure 16 Creating table for Module



```
SQL> DESCRIBE Module;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MODULE_CODE                               NOT NULL VARCHAR2(30)
 MODULE_NAME                               NOT NULL VARCHAR2(30)
```

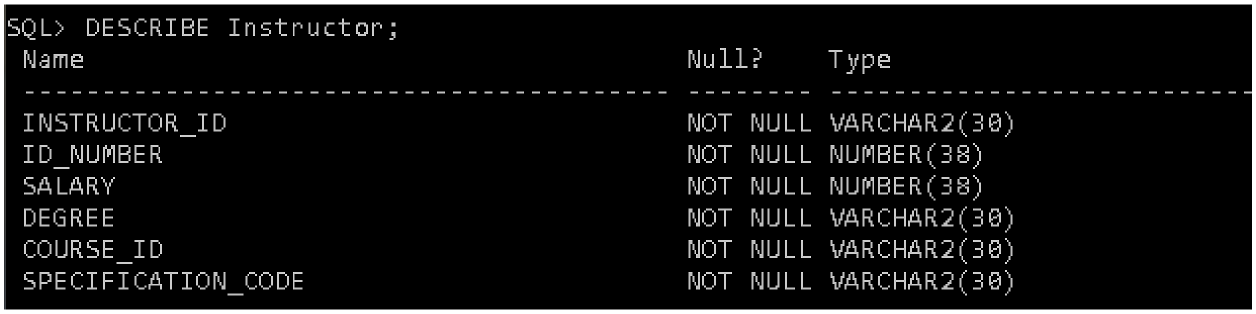
Figure 17 Describe table Module

- **Creating table for Instructor:**



```
SQL> CREATE TABLE Instructor (
     Instructor_id VARCHAR(30) NOT NULL,
     Id_number INT NOT NULL,
     Salary INT NOT NULL,
     Degree VARCHAR(30) NOT NULL,
     Course_id VARCHAR(30) NOT NULL,
     Specification_code VARCHAR(30) NOT NULL,
     CONSTRAINT Instid_pk
     PRIMARY KEY (Instructor_id),
     CONSTRAINT id_nofk
     FOREIGN KEY (Id_number) REFERENCES Person(Id_number),
     CONSTRAINT Course_idfk
     FOREIGN KEY (Course_id) REFERENCES Course(Course_id),
     CONSTRAINT Specification_codefk
     FOREIGN KEY (Specification_code) REFERENCES Specification(Specification_code))
;

Table created.
```

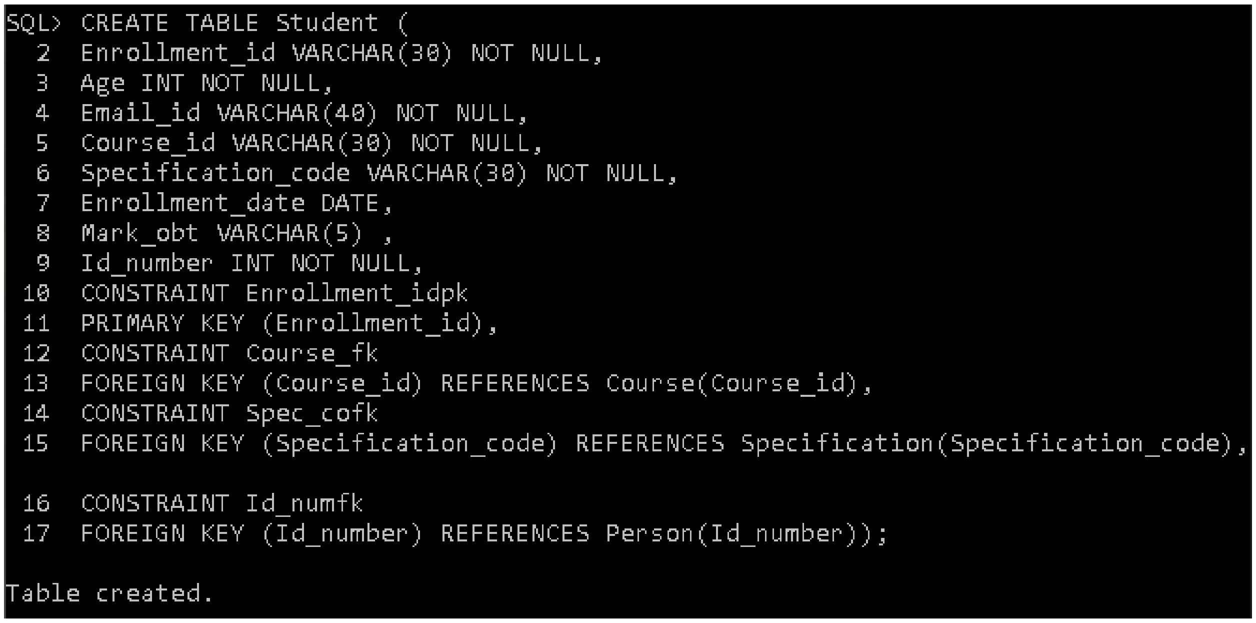
Figure 18 4 Creating table for Instructor



```
SQL> DESCRIBE Instructor;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 INSTRUCTOR_ID                             NOT NULL VARCHAR2(30)
 ID_NUMBER                                 NOT NULL NUMBER(38)
 SALARY                                    NOT NULL NUMBER(38)
 DEGREE                                    NOT NULL VARCHAR2(30)
 COURSE_ID                                 NOT NULL VARCHAR2(30)
 SPECIFICATION_CODE                        NOT NULL VARCHAR2(30)
```

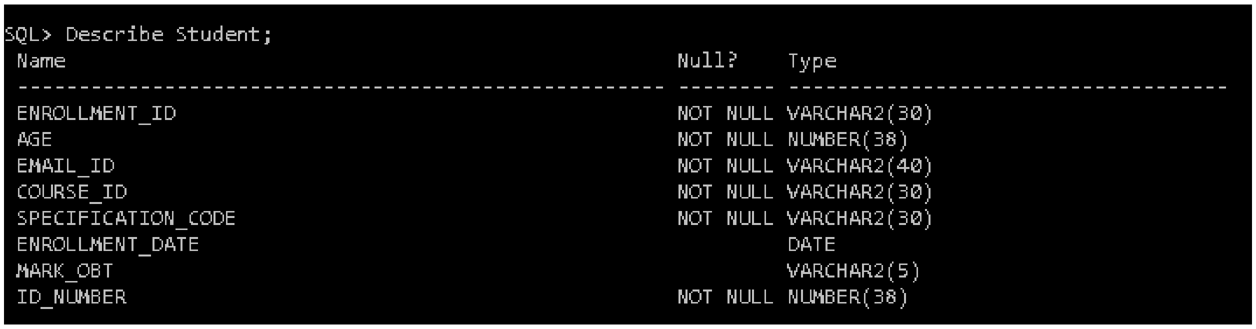
Figure 19 Describe table Instructor

- **Creating table for Student:**



```
SQL> CREATE TABLE Student (
  2  Enrollment_id VARCHAR(30) NOT NULL,
  3  Age INT NOT NULL,
  4  Email_id VARCHAR(40) NOT NULL,
  5  Course_id VARCHAR(30) NOT NULL,
  6  Specification_code VARCHAR(30) NOT NULL,
  7  Enrollment_date DATE,
  8  Mark_obt VARCHAR(5) ,
  9  Id_number INT NOT NULL,
 10  CONSTRAINT Enrollment_idpk
 11  PRIMARY KEY (Enrollment_id),
 12  CONSTRAINT Course_fk
 13  FOREIGN KEY (Course_id) REFERENCES Course(Course_id),
 14  CONSTRAINT Spec_cofk
 15  FOREIGN KEY (Specification_code) REFERENCES Specification(Specification_code),

 16  CONSTRAINT Id_numfk
 17  FOREIGN KEY (Id_number) REFERENCES Person(Id_number));

Table created.
```

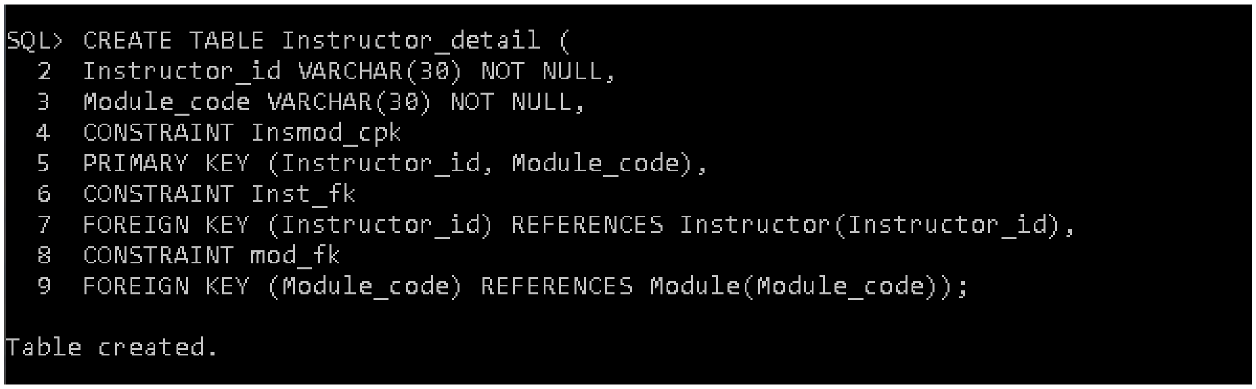
Figure 20 4 Creating table for Student



```
SQL> Describe Student;
 Name                                                  Null?    Type
 ----------------------------------------------------- -------- ------------------------------------
 ENROLLMENT_ID                                         NOT NULL VARCHAR2(30)
 AGE                                                   NOT NULL NUMBER(38)
 EMAIL_ID                                              NOT NULL VARCHAR2(40)
 COURSE_ID                                             NOT NULL VARCHAR2(30)
 SPECIFICATION_CODE                                    NOT NULL VARCHAR2(30)
 ENROLLMENT_DATE                                                DATE
 MARK_OBT                                                       VARCHAR2(5)
 ID_NUMBER                                             NOT NULL NUMBER(38)
```

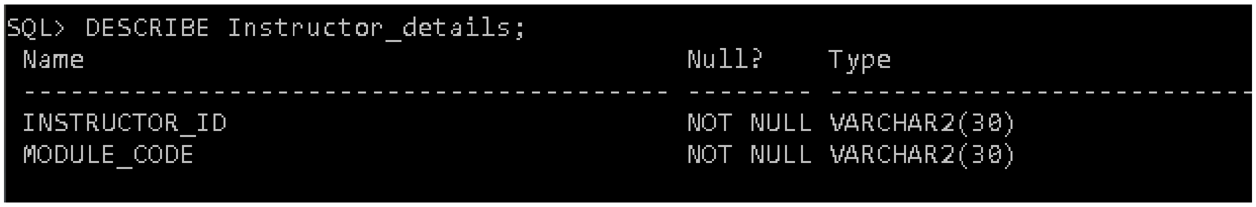
Figure 21 Describe table Student

- **Creating table for Instructor_detail:**



```
SQL> CREATE TABLE Instructor_detail (
     Instructor_id VARCHAR(30) NOT NULL,
     Module_code VARCHAR(30) NOT NULL,
     CONSTRAINT Insmod_cpk
     PRIMARY KEY (Instructor_id, Module_code),
     CONSTRAINT Inst_fk
     FOREIGN KEY (Instructor_id) REFERENCES Instructor(Instructor_id),
     CONSTRAINT mod_fk
     FOREIGN KEY (Module_code) REFERENCES Module(Module_code));

Table created.
```

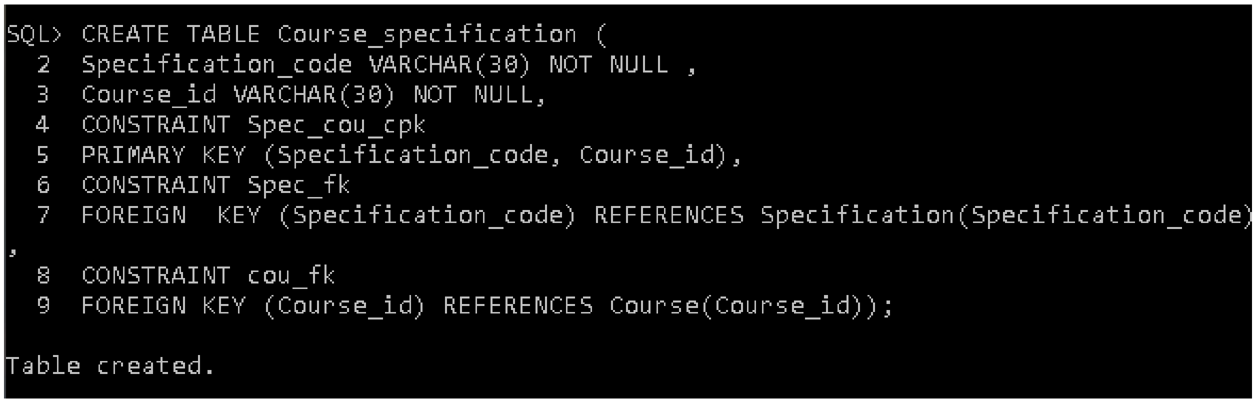
Figure 22 4 Creating table for Instructor_detail



```
SQL> DESCRIBE Instructor_details;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 INSTRUCTOR_ID                             NOT NULL VARCHAR2(30)
 MODULE_CODE                               NOT NULL VARCHAR2(30)
```

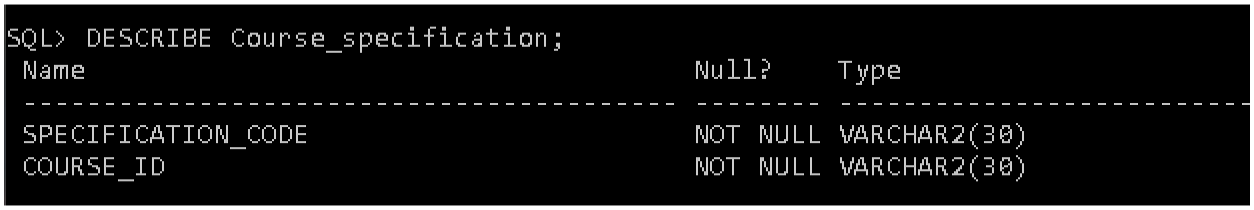
Figure 23 Describe table Instructors_detal

- **Creating table for Course_specification:**



```
SQL> CREATE TABLE Course_specification (
  2  Specification_code VARCHAR(30) NOT NULL ,
  3  Course_id VARCHAR(30) NOT NULL,
  4  CONSTRAINT Spec_cou_cpk
  5  PRIMARY KEY (Specification_code, Course_id),
  6  CONSTRAINT Spec_fk
  7  FOREIGN  KEY (Specification_code) REFERENCES Specification(Specification_code)
,
  8  CONSTRAINT cou_fk
  9  FOREIGN KEY (Course_id) REFERENCES Course(Course_id));

Table created.
```

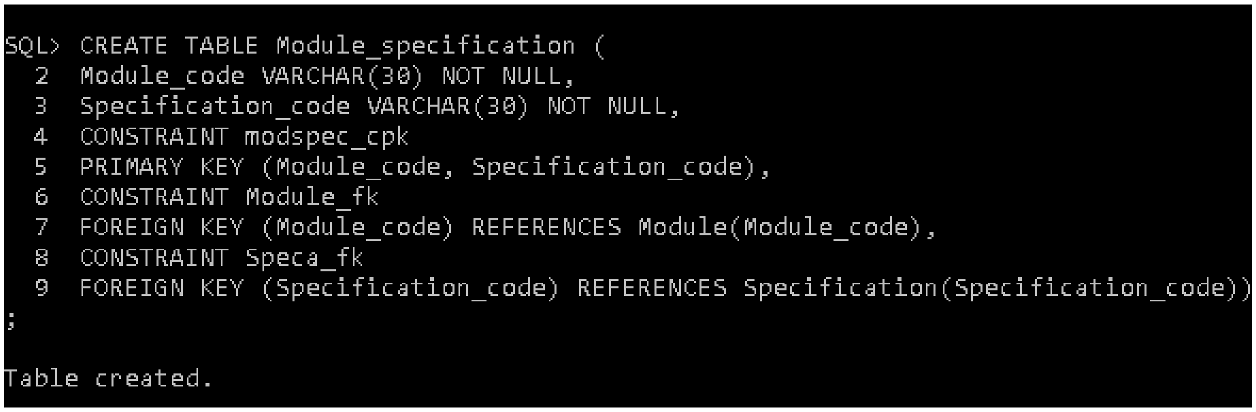
Figure 24 Creating table for Course_specification



```
SQL> DESCRIBE Course_specification;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SPECIFICATION_CODE                        NOT NULL VARCHAR2(30)
 COURSE_ID                                 NOT NULL VARCHAR2(30)
```

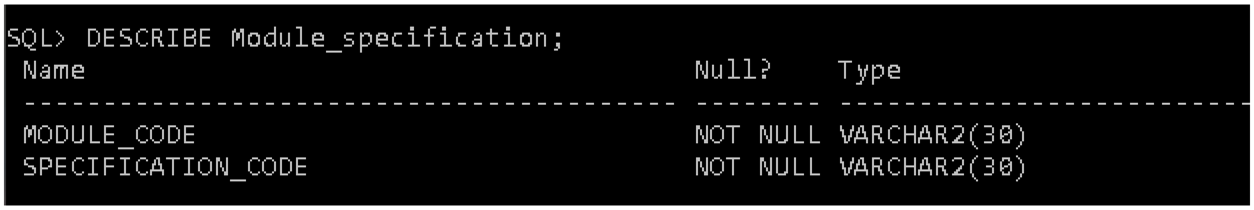
Figure 25 Describe table Course_specification

- **Creating table for Module_specification:**



```
SQL> CREATE TABLE Module_specification (
  2  Module_code VARCHAR(30) NOT NULL,
  3  Specification_code VARCHAR(30) NOT NULL,
  4  CONSTRAINT modspec_cpk
  5  PRIMARY KEY (Module_code, Specification_code),
  6  CONSTRAINT Module_fk
  7  FOREIGN KEY (Module_code) REFERENCES Module(Module_code),
  8  CONSTRAINT Speca_fk
  9  FOREIGN KEY (Specification_code) REFERENCES Specification(Specification_code))
;

Table created.
```

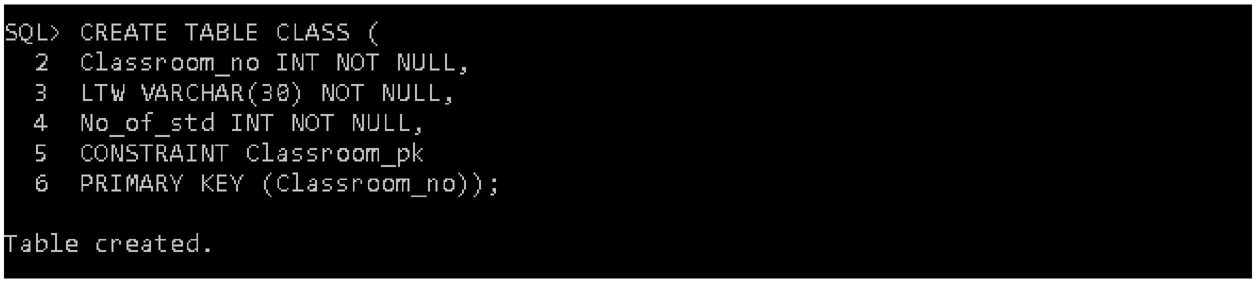
Figure 26 4 Creating table for Module_specification



```
SQL> DESCRIBE Module_specification;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MODULE_CODE                               NOT NULL VARCHAR2(30)
 SPECIFICATION_CODE                        NOT NULL VARCHAR2(30)
```

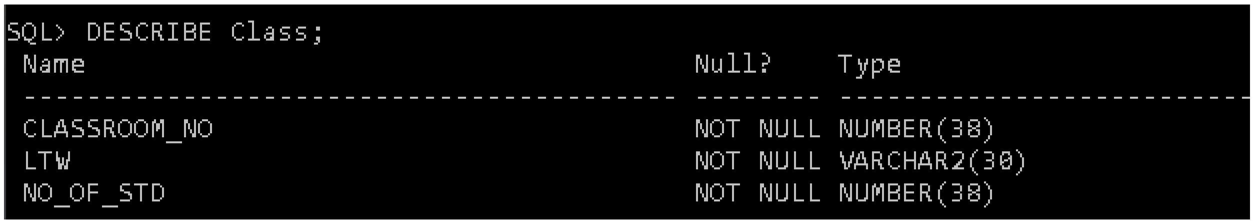
Figure 27 Describe table Module_specification

- **Creating table for Class:**



```
SQL> CREATE TABLE CLASS (
  2  Classroom_no INT NOT NULL,
  3  LTW VARCHAR(30) NOT NULL,
  4  No_of_std INT NOT NULL,
  5  CONSTRAINT Classroom_pk
  6  PRIMARY KEY (Classroom_no));

Table created.
```

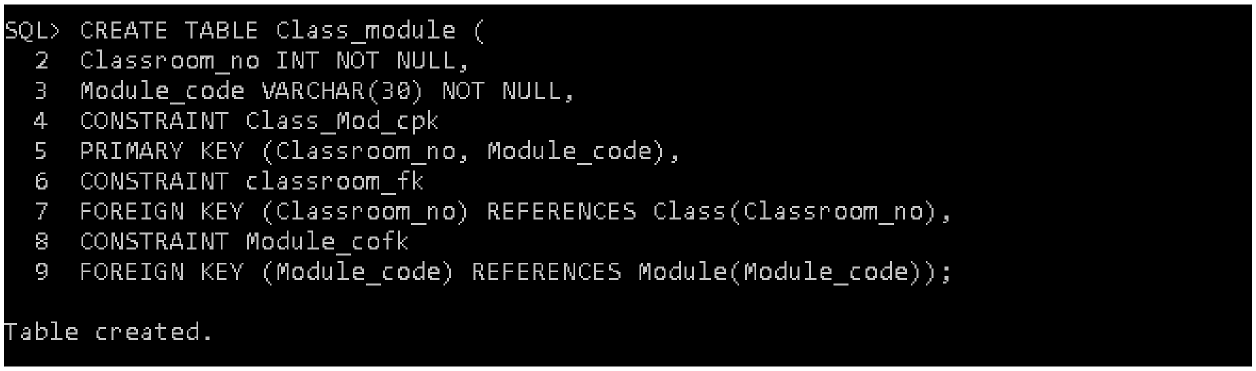
Figure 28 4 Creating table for Class



```
SQL> DESCRIBE Class;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CLASSROOM_NO                              NOT NULL NUMBER(38)
 LTW                                       NOT NULL VARCHAR2(30)
 NO_OF_STD                                 NOT NULL NUMBER(38)
```

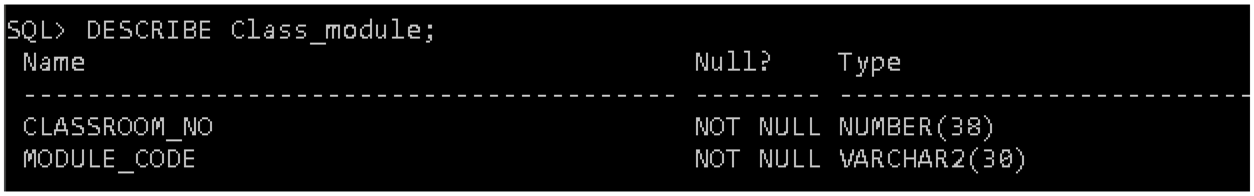
Figure 29 Describe table Class

- **Creating table for Class_module:**



```
SQL> CREATE TABLE Class_module (
     Classroom_no INT NOT NULL,
     Module_code VARCHAR(30) NOT NULL,
     CONSTRAINT Class_Mod_cpk
     PRIMARY KEY (Classroom_no, Module_code),
     CONSTRAINT classroom_fk
     FOREIGN KEY (Classroom_no) REFERENCES Class(Classroom_no),
     CONSTRAINT Module_cofk
     FOREIGN KEY (Module_code) REFERENCES Module(Module_code));

Table created.
```

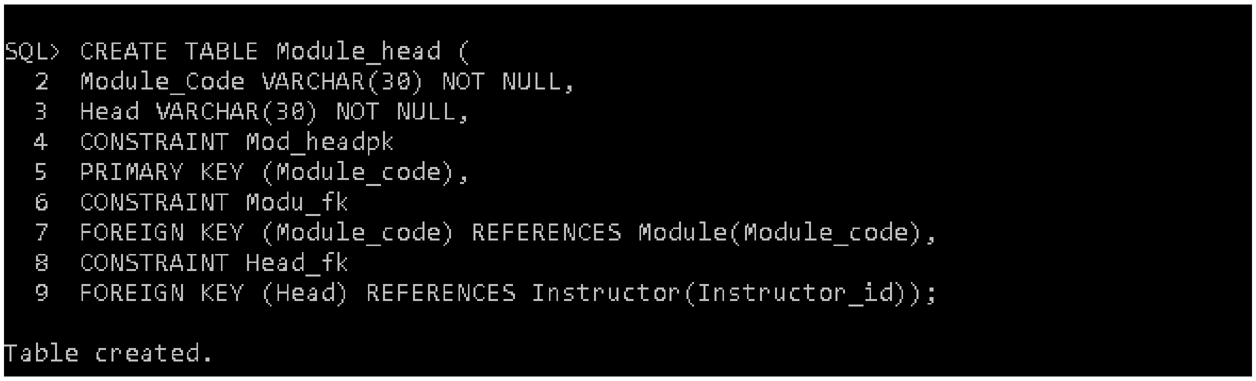
Figure 30 4 Creating table for Class_module



```
SQL> DESCRIBE Class_module;
 Name                                      Null?    Type
 ----------------------------------------- -------- -------------------------
 CLASSROOM_NO                              NOT NULL NUMBER(38)
 MODULE_CODE                               NOT NULL VARCHAR2(30)
```

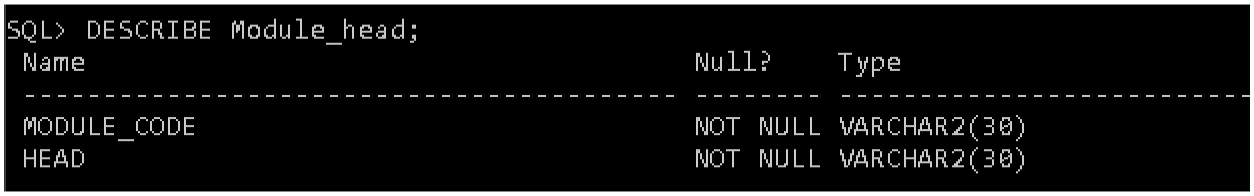
Figure 31 Describe table Class_module

- **Creating table for Module_head:**



```
SQL> CREATE TABLE Module_head (
     Module_Code VARCHAR(30) NOT NULL,
     Head VARCHAR(30) NOT NULL,
     CONSTRAINT Mod_headpk
     PRIMARY KEY (Module_code),
     CONSTRAINT Modu_fk
     FOREIGN KEY (Module_code) REFERENCES Module(Module_code),
     CONSTRAINT Head_fk
     FOREIGN KEY (Head) REFERENCES Instructor(Instructor_id));

Table created.
```

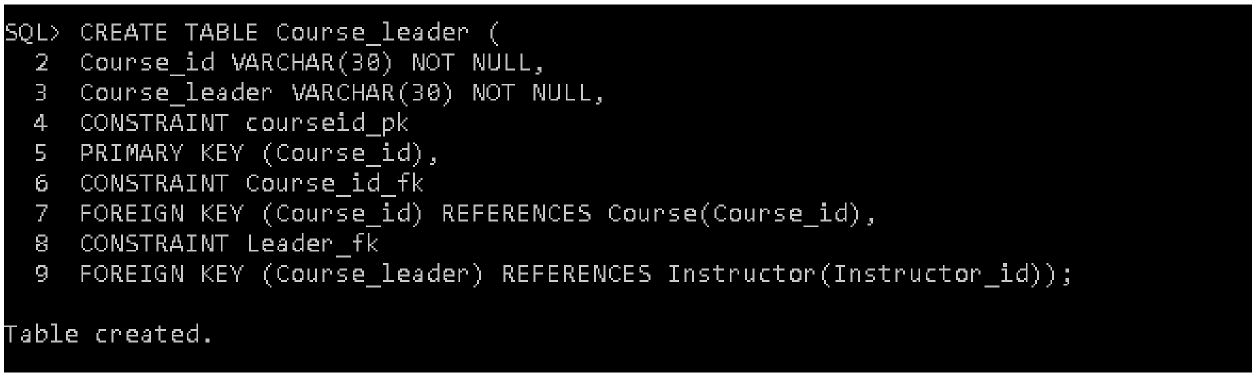
Figure 32 Creating table for Module_head



```
SQL> DESCRIBE Module_head;
 Name                                      Null?    Type
 ----------------------------------------- -------- -------------------------
 MODULE_CODE                               NOT NULL VARCHAR2(30)
 HEAD                                      NOT NULL VARCHAR2(30)
```

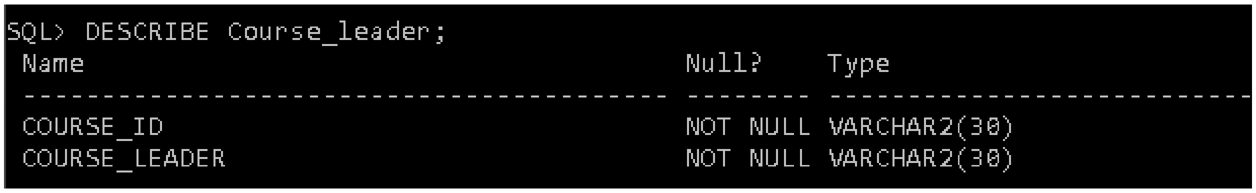
Figure 33 Describe table Module_head

- **Creating table for Course_leader:**



```
SQL> CREATE TABLE Course_leader (
  2  Course_id VARCHAR(30) NOT NULL,
  3  Course_leader VARCHAR(30) NOT NULL,
  4  CONSTRAINT courseid_pk
  5  PRIMARY KEY (Course_id),
  6  CONSTRAINT Course_id_fk
  7  FOREIGN KEY (Course_id) REFERENCES Course(Course_id),
  8  CONSTRAINT Leader_fk
  9  FOREIGN KEY (Course_leader) REFERENCES Instructor(Instructor_id));

Table created.
```

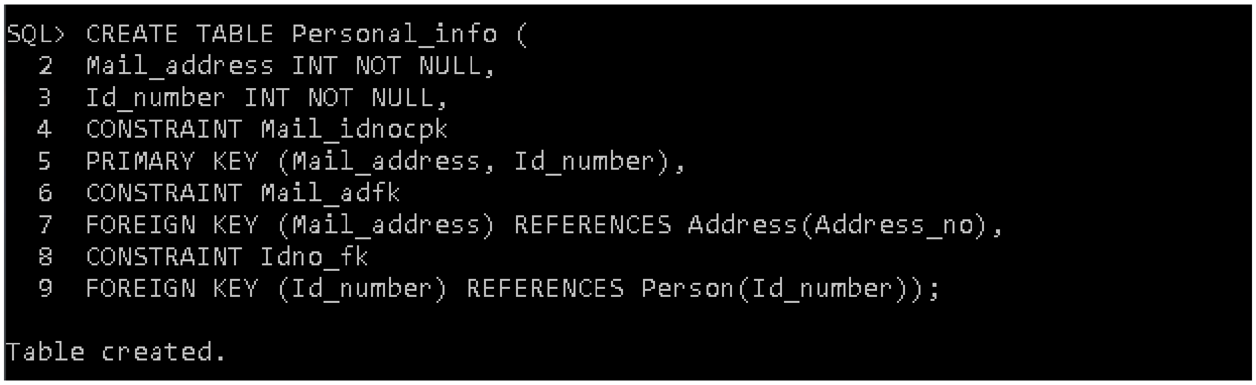
Figure 34 Creating table for Course_leader



```
SQL> DESCRIBE Course_leader;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 COURSE_ID                                 NOT NULL VARCHAR2(30)
 COURSE_LEADER                             NOT NULL VARCHAR2(30)
```

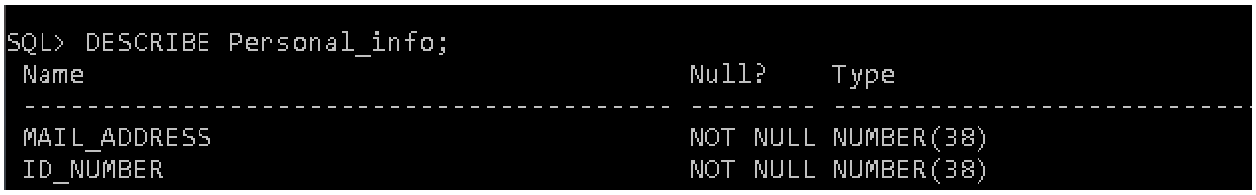
Figure 35 Describe table Course_leader

- **Creating table for Personal_info:**



```
SQL> CREATE TABLE Personal_info (
  2  Mail_address INT NOT NULL,
  3  Id_number INT NOT NULL,
  4  CONSTRAINT Mail_idnocpk
  5  PRIMARY KEY (Mail_address, Id_number),
  6  CONSTRAINT Mail_adfk
  7  FOREIGN KEY (Mail_address) REFERENCES Address(Address_no),
  8  CONSTRAINT Idno_fk
  9  FOREIGN KEY (Id_number) REFERENCES Person(Id_number));

Table created.
```

Figure 36 4 Creating table for Personal_info



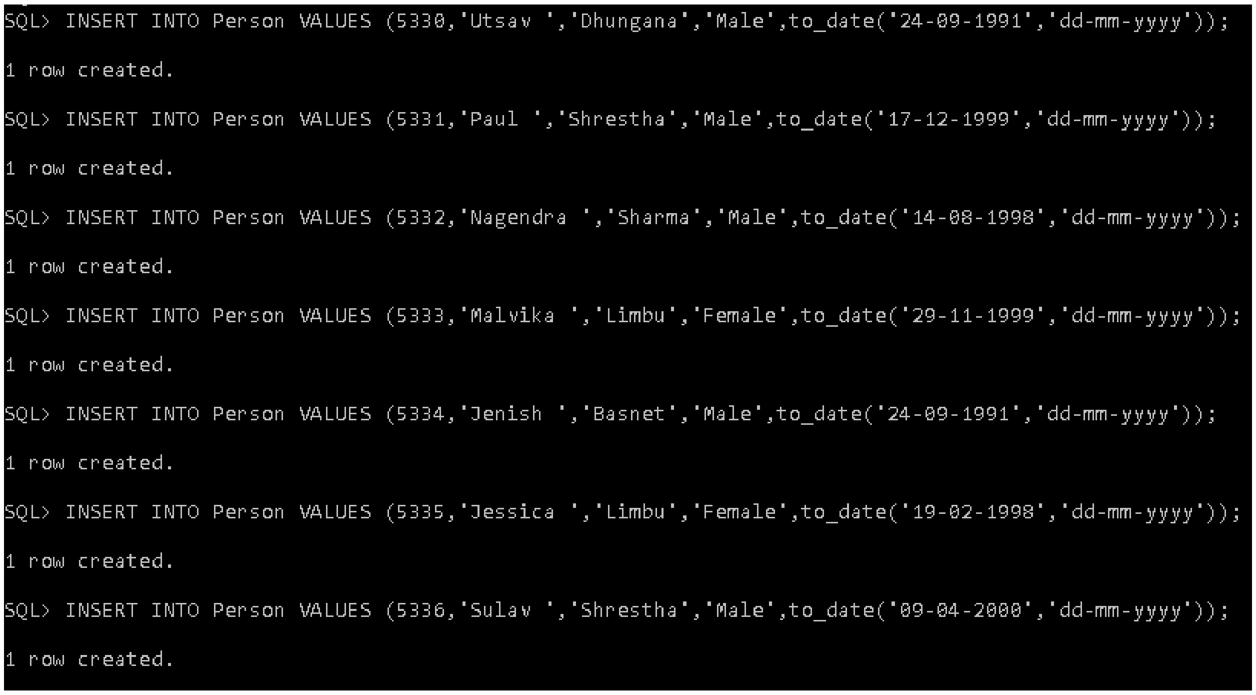
```
SQL> DESCRIBE Personal_info;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MAIL_ADDRESS                              NOT NULL NUMBER(38)
 ID_NUMBER                                 NOT NULL NUMBER(38)
```

Figure 37 Describe table Personal_info

4.2 Populating database tables:

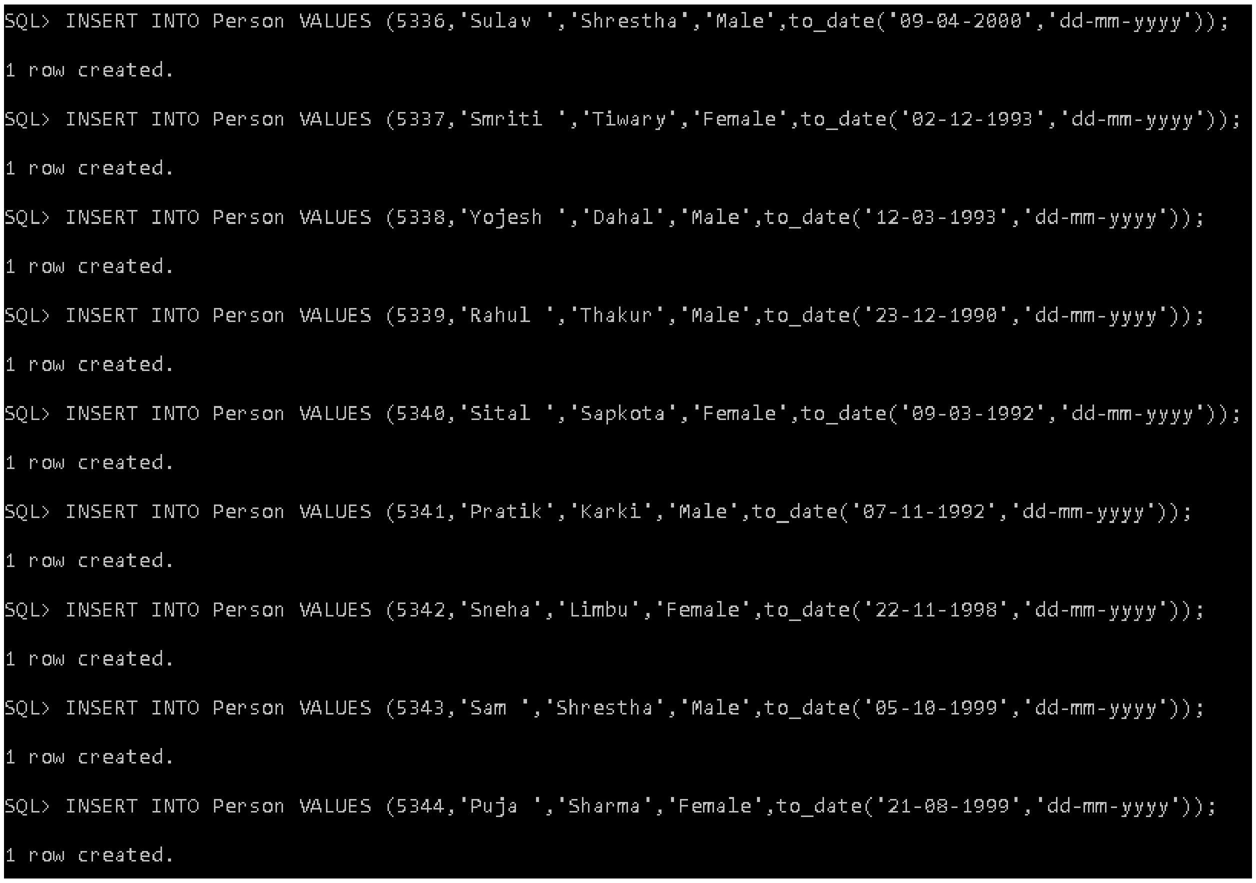
For inserting data into table ‘INSERT INTO’ command is used.

- **Inserting values into Person table:**



```
SQL> INSERT INTO Person VALUES (5330,'Utsav ','Dhungana','Male',to_date('24-09-1991','dd-mm-yyyy'));

1 row created.

SQL> INSERT INTO Person VALUES (5331,'Paul ','Shrestha','Male',to_date('17-12-1999','dd-mm-yyyy'));

1 row created.

SQL> INSERT INTO Person VALUES (5332,'Nagendra ','Sharma','Male',to_date('14-08-1998','dd-mm-yyyy'));

1 row created.

SQL> INSERT INTO Person VALUES (5333,'Malvika ','Limbu','Female',to_date('29-11-1999','dd-mm-yyyy'));

1 row created.

SQL> INSERT INTO Person VALUES (5334,'Jenish ','Basnet','Male',to_date('24-09-1991','dd-mm-yyyy'));

1 row created.

SQL> INSERT INTO Person VALUES (5335,'Jessica ','Limbu','Female',to_date('19-02-1998','dd-mm-yyyy'));

1 row created.

SQL> INSERT INTO Person VALUES (5336,'Sulav ','Shrestha','Male',to_date('09-04-2000','dd-mm-yyyy'));

1 row created.
```

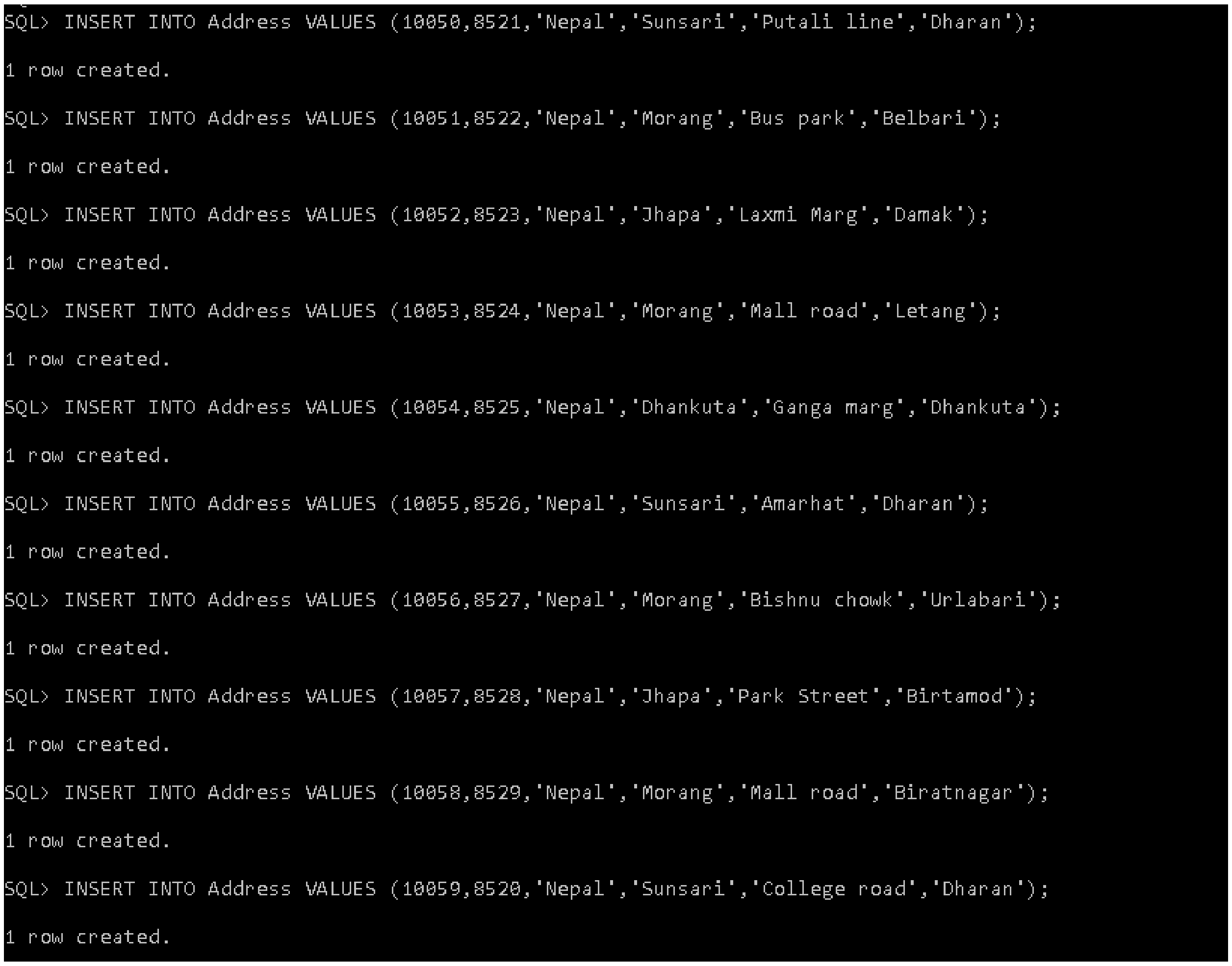
Figure 38 Inserting values into Person table



```
SQL> INSERT INTO Person VALUES (5336,'Sulav ','Shrestha','Male',to_date('09-04-2000','dd-mm-yyyy'));

1 row created.

SQL> INSERT INTO Person VALUES (5337,'Smriti ','Tiwary','Female',to_date('02-12-1993','dd-mm-yyyy'));

1 row created.

SQL> INSERT INTO Person VALUES (5338,'Yojesh ','Dahal','Male',to_date('12-03-1993','dd-mm-yyyy'));

1 row created.

SQL> INSERT INTO Person VALUES (5339,'Rahul ','Thakur','Male',to_date('23-12-1990','dd-mm-yyyy'));

1 row created.

SQL> INSERT INTO Person VALUES (5340,'Sital ','Sapkota','Female',to_date('09-03-1992','dd-mm-yyyy'));

1 row created.

SQL> INSERT INTO Person VALUES (5341,'Pratik','Karki','Male',to_date('07-11-1992','dd-mm-yyyy'));

1 row created.

SQL> INSERT INTO Person VALUES (5342,'Sneha','Limbu','Female',to_date('22-11-1998','dd-mm-yyyy'));

1 row created.

SQL> INSERT INTO Person VALUES (5343,'Sam ','Shrestha','Male',to_date('05-10-1999','dd-mm-yyyy'));

1 row created.

SQL> INSERT INTO Person VALUES (5344,'Puja ','Sharma','Female',to_date('21-08-1999','dd-mm-yyyy'));

1 row created.
```

Figure 39 Inserting values into Person table

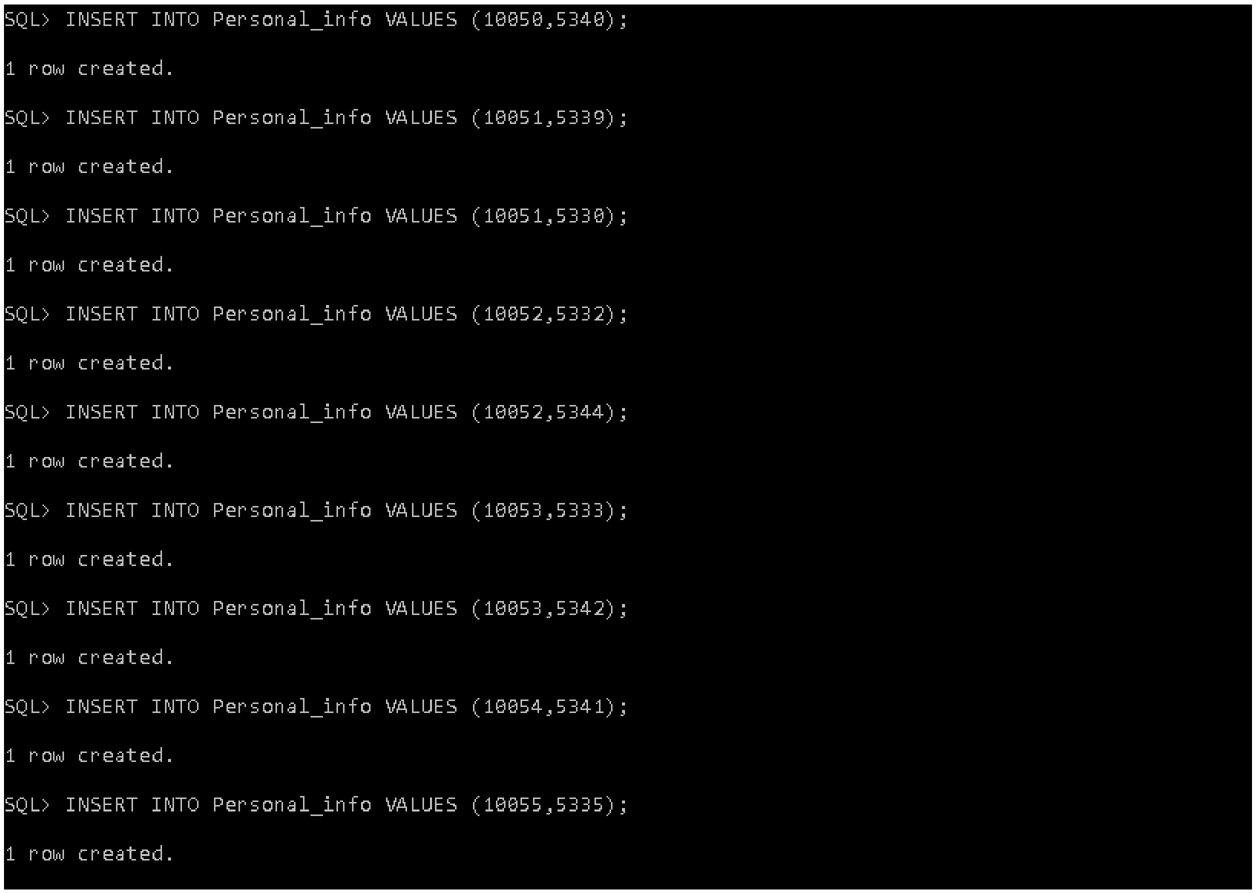
- **Inserting values into Address:**



```
SQL> INSERT INTO Address VALUES (10050,8521,'Nepal','Sunsari','Putali line','Dharan');

1 row created.

SQL> INSERT INTO Address VALUES (10051,8522,'Nepal','Morang','Bus park','Belbari');

1 row created.

SQL> INSERT INTO Address VALUES (10052,8523,'Nepal','Jhapa','Laxmi Marg','Damak');

1 row created.

SQL> INSERT INTO Address VALUES (10053,8524,'Nepal','Morang','Mall road','Letang');

1 row created.

SQL> INSERT INTO Address VALUES (10054,8525,'Nepal','Dhankuta','Ganga marg','Dhankuta');

1 row created.

SQL> INSERT INTO Address VALUES (10055,8526,'Nepal','Sunsari','Amarhat','Dharan');

1 row created.

SQL> INSERT INTO Address VALUES (10056,8527,'Nepal','Morang','Bishnu chowk','Urlabari');

1 row created.

SQL> INSERT INTO Address VALUES (10057,8528,'Nepal','Jhapa','Park Street','Birtamod');

1 row created.

SQL> INSERT INTO Address VALUES (10058,8529,'Nepal','Morang','Mall road','Biratnagar');

1 row created.

SQL> INSERT INTO Address VALUES (10059,8520,'Nepal','Sunsari','College road','Dharan');

1 row created.
```

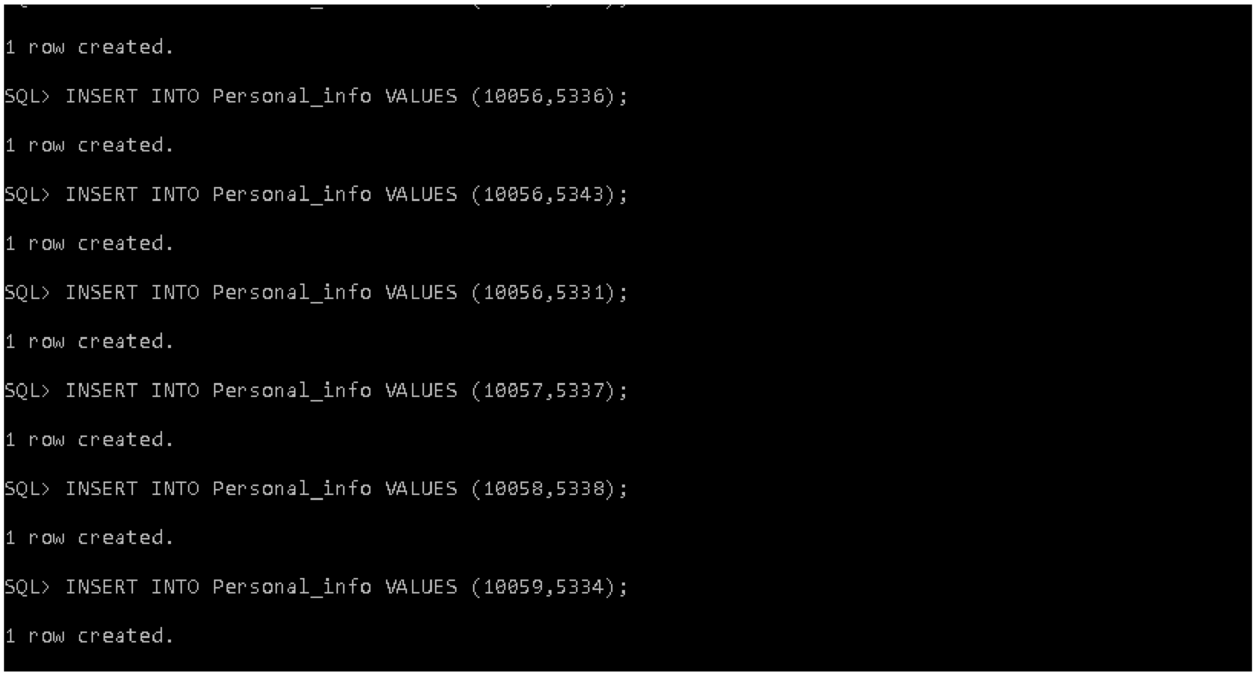
Figure 40 Inserting values into Address table

- **Inserting values into Personal_info:**



```
SQL> INSERT INTO Personal_info VALUES (10050,5340);

1 row created.

SQL> INSERT INTO Personal_info VALUES (10051,5339);

1 row created.

SQL> INSERT INTO Personal_info VALUES (10051,5330);

1 row created.

SQL> INSERT INTO Personal_info VALUES (10052,5332);

1 row created.

SQL> INSERT INTO Personal_info VALUES (10052,5344);

1 row created.

SQL> INSERT INTO Personal_info VALUES (10053,5333);

1 row created.

SQL> INSERT INTO Personal_info VALUES (10053,5342);

1 row created.

SQL> INSERT INTO Personal_info VALUES (10054,5341);

1 row created.

SQL> INSERT INTO Personal_info VALUES (10055,5335);

1 row created.
```

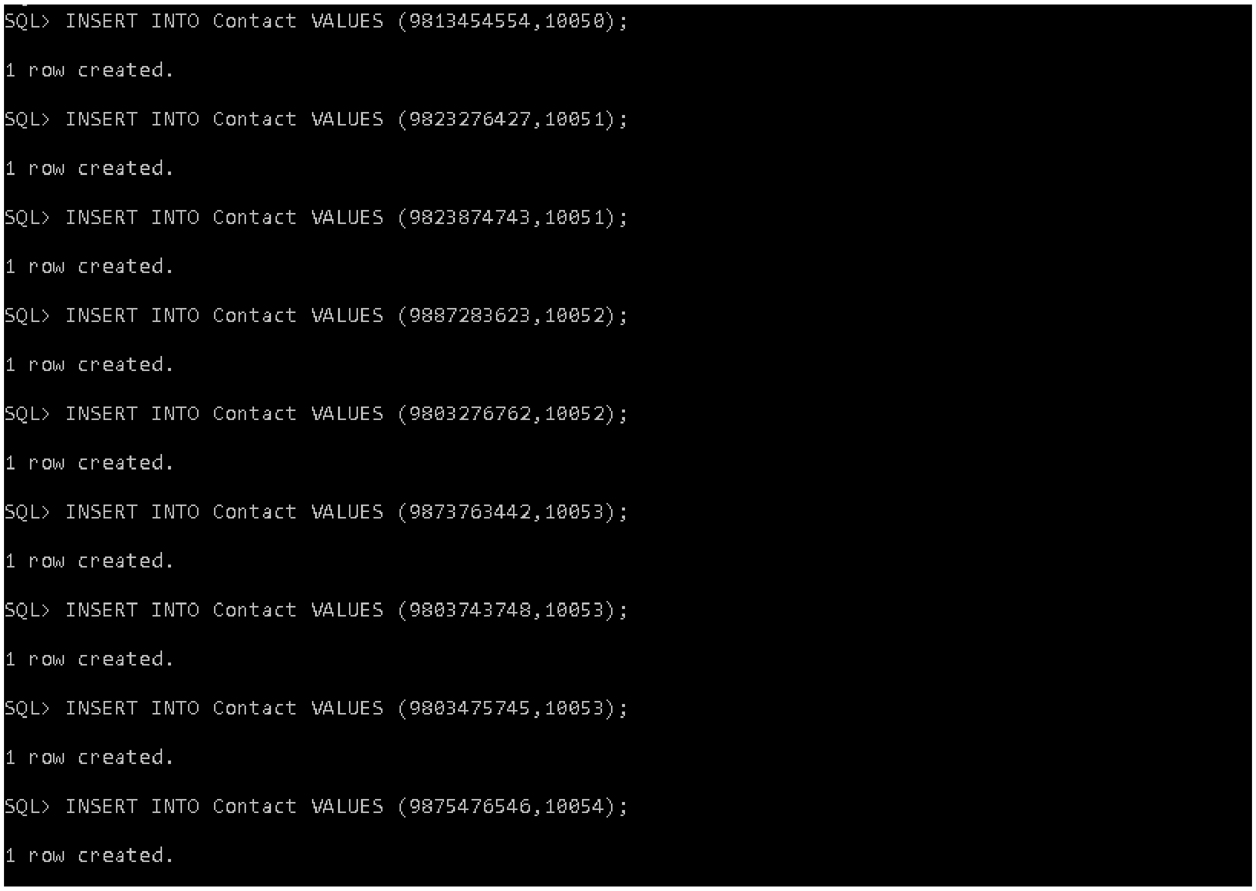
Figure 41 Inserting values into Personal_info table



```
1 row created.

SQL> INSERT INTO Personal_info VALUES (10056,5336);

1 row created.

SQL> INSERT INTO Personal_info VALUES (10056,5343);

1 row created.

SQL> INSERT INTO Personal_info VALUES (10056,5331);

1 row created.

SQL> INSERT INTO Personal_info VALUES (10057,5337);

1 row created.

SQL> INSERT INTO Personal_info VALUES (10058,5338);

1 row created.

SQL> INSERT INTO Personal_info VALUES (10059,5334);

1 row created.
```

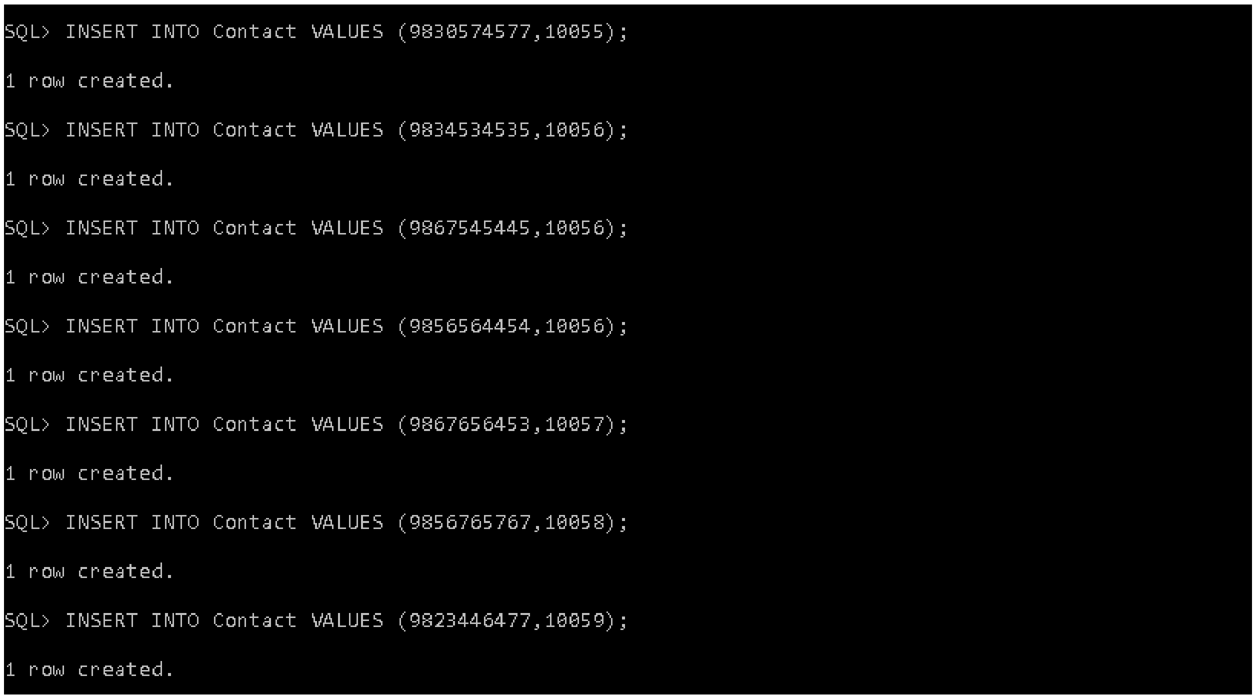
Figure 42 Inserting values into Personal_info table

- **Inserting values into Contact:**



```
SQL> INSERT INTO Contact VALUES (9813454554,10050);

1 row created.

SQL> INSERT INTO Contact VALUES (9823276427,10051);

1 row created.

SQL> INSERT INTO Contact VALUES (9823874743,10051);

1 row created.

SQL> INSERT INTO Contact VALUES (9887283623,10052);

1 row created.

SQL> INSERT INTO Contact VALUES (9803276762,10052);

1 row created.

SQL> INSERT INTO Contact VALUES (9873763442,10053);

1 row created.

SQL> INSERT INTO Contact VALUES (9803743748,10053);

1 row created.

SQL> INSERT INTO Contact VALUES (9803475745,10053);

1 row created.

SQL> INSERT INTO Contact VALUES (9875476546,10054);

1 row created.
```

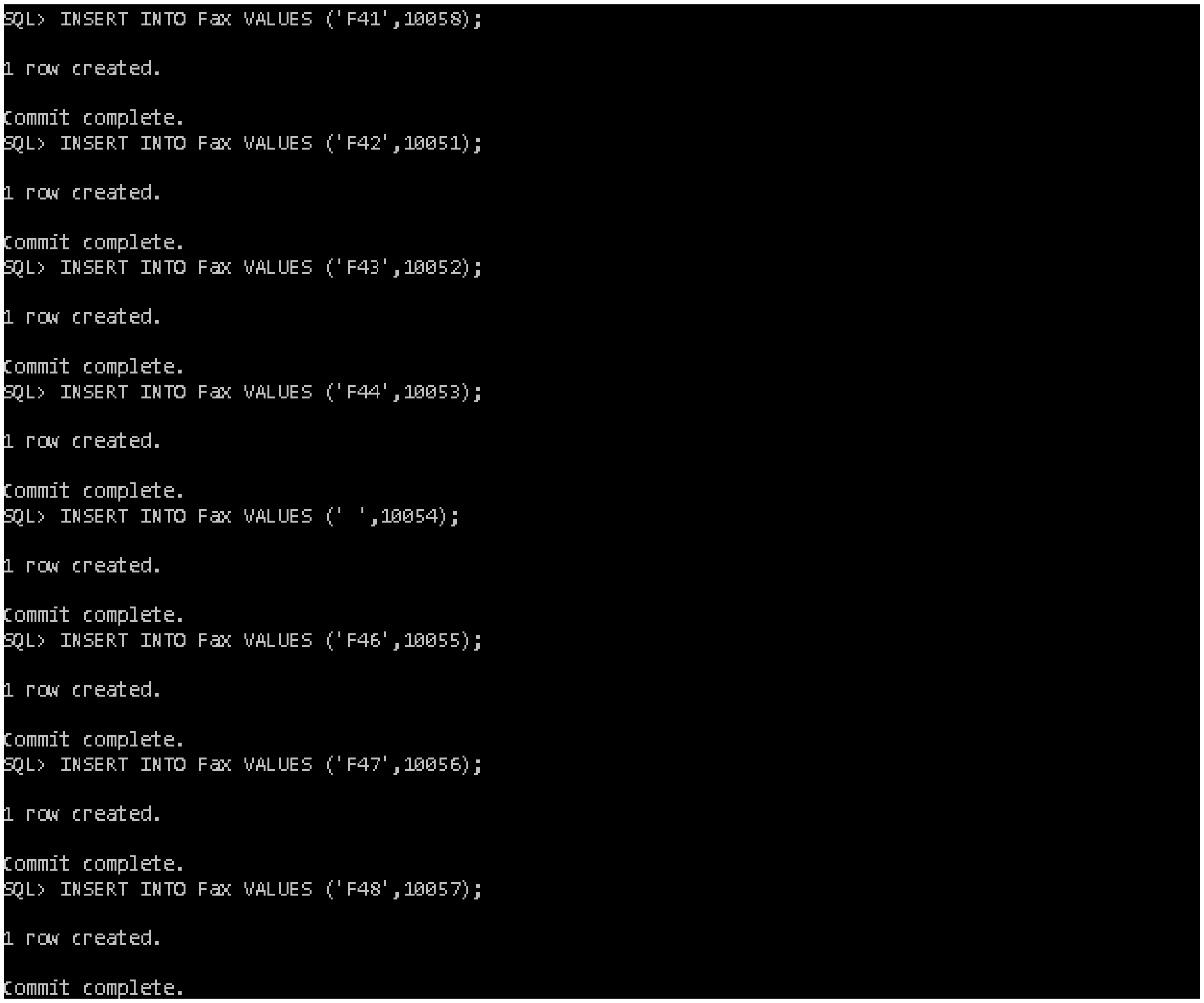
Figure 43 Inserting values into Contact table



```
SQL> INSERT INTO Contact VALUES (9830574577,10055);

1 row created.

SQL> INSERT INTO Contact VALUES (9834534535,10056);

1 row created.

SQL> INSERT INTO Contact VALUES (9867545445,10056);

1 row created.

SQL> INSERT INTO Contact VALUES (9856564454,10056);

1 row created.

SQL> INSERT INTO Contact VALUES (9867656453,10057);

1 row created.

SQL> INSERT INTO Contact VALUES (9856765767,10058);

1 row created.

SQL> INSERT INTO Contact VALUES (9823446477,10059);

1 row created.
```

Figure 44 Inserting values into Contact table

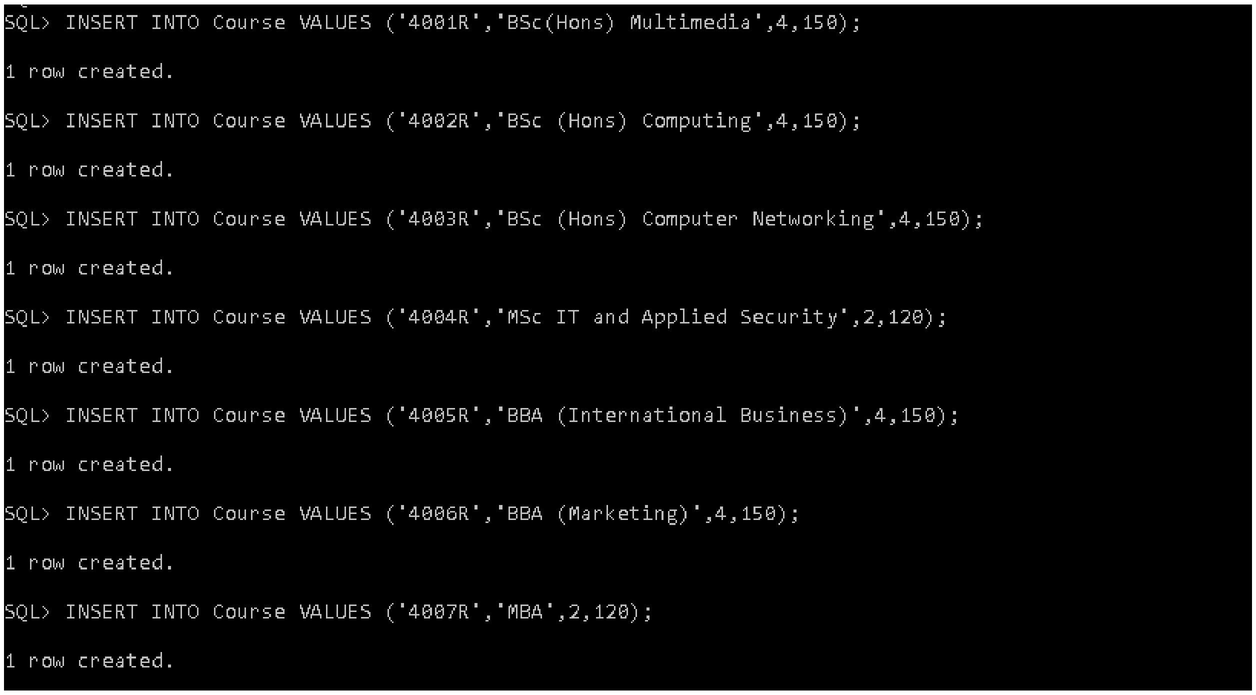
- **Inserting values into Fax:**



```
SQL> INSERT INTO Fax VALUES ('F41',10058);

1 row created.

Commit complete.
SQL> INSERT INTO Fax VALUES ('F42',10051);

1 row created.

Commit complete.
SQL> INSERT INTO Fax VALUES ('F43',10052);

1 row created.

Commit complete.
SQL> INSERT INTO Fax VALUES ('F44',10053);

1 row created.

Commit complete.
SQL> INSERT INTO Fax VALUES (' ',10054);

1 row created.

Commit complete.
SQL> INSERT INTO Fax VALUES ('F46',10055);

1 row created.

Commit complete.
SQL> INSERT INTO Fax VALUES ('F47',10056);

1 row created.

Commit complete.
SQL> INSERT INTO Fax VALUES ('F48',10057);

1 row created.

Commit complete.
```

Figure 45 Inserting values into Fax table

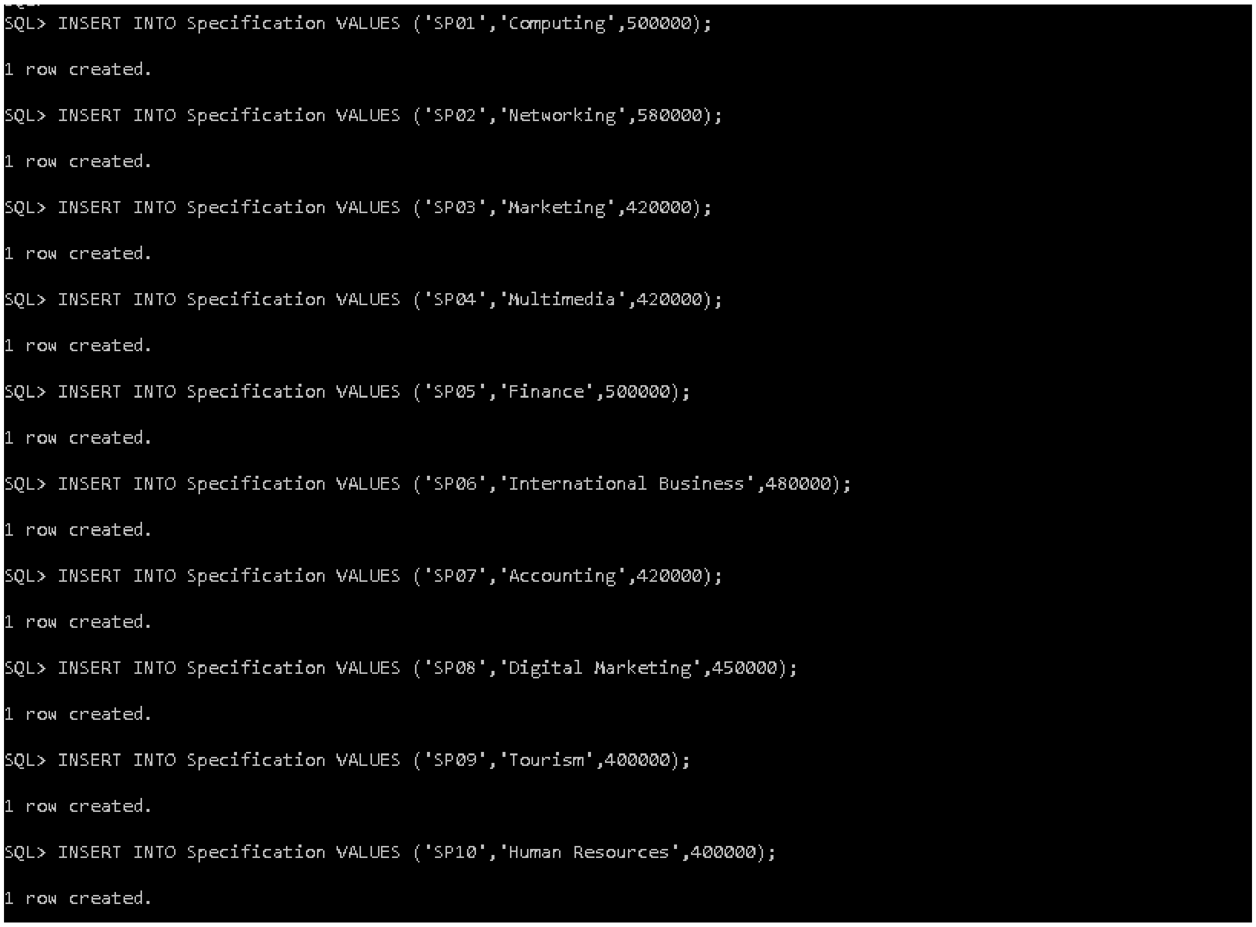
- **Inserting values into Course:**



```
SQL> INSERT INTO Course VALUES ('4001R','BSc(Hons) Multimedia',4,150);

1 row created.

SQL> INSERT INTO Course VALUES ('4002R','BSc (Hons) Computing',4,150);

1 row created.

SQL> INSERT INTO Course VALUES ('4003R','BSc (Hons) Computer Networking',4,150);

1 row created.

SQL> INSERT INTO Course VALUES ('4004R','MSc IT and Applied Security',2,120);

1 row created.

SQL> INSERT INTO Course VALUES ('4005R','BBA (International Business)',4,150);

1 row created.

SQL> INSERT INTO Course VALUES ('4006R','BBA (Marketing)',4,150);

1 row created.

SQL> INSERT INTO Course VALUES ('4007R','MBA',2,120);

1 row created.
```

Figure 46 Inserting values into Course table

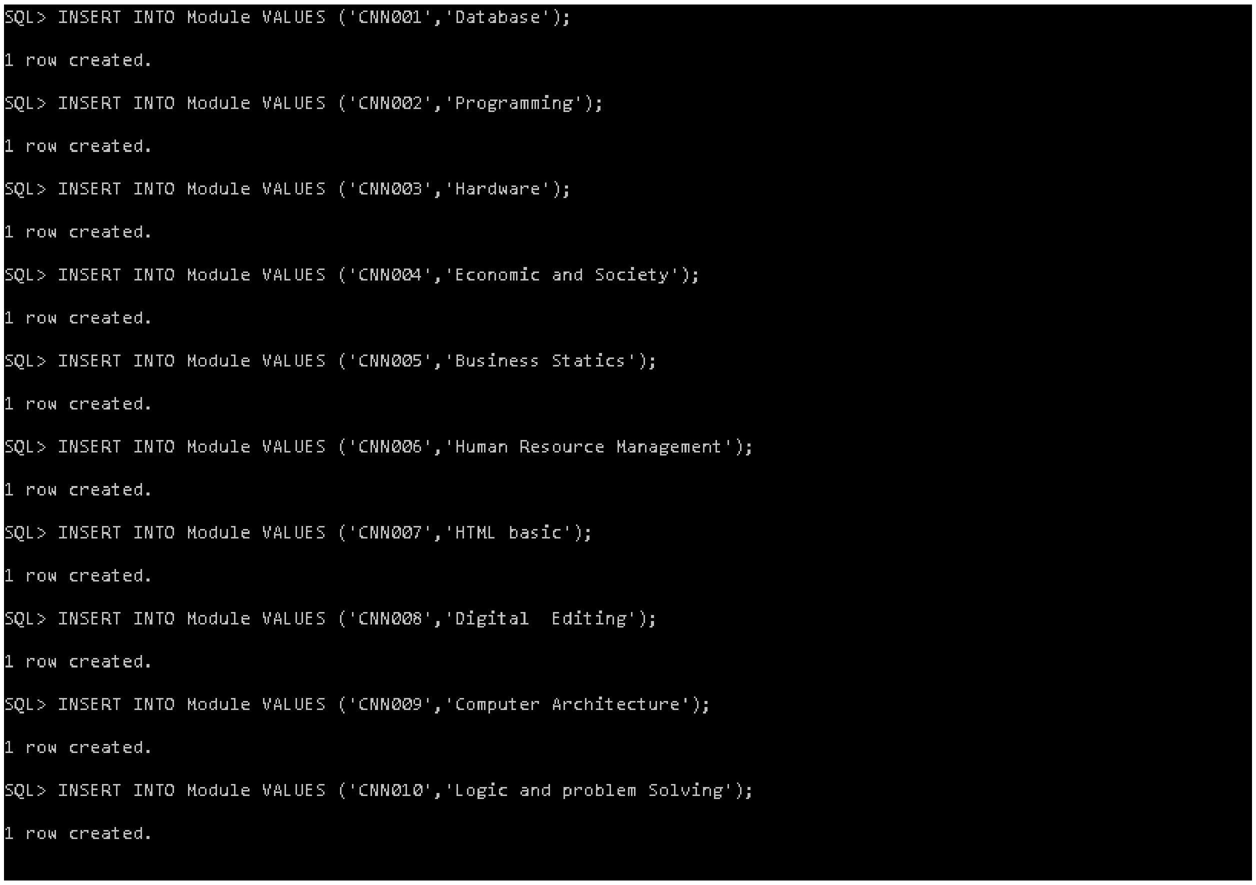
- **Inserting values into Specification:**



```
SQL> INSERT INTO Specification VALUES ('SP01','Computing',500000);

1 row created.

SQL> INSERT INTO Specification VALUES ('SP02','Networking',580000);

1 row created.

SQL> INSERT INTO Specification VALUES ('SP03','Marketing',420000);

1 row created.

SQL> INSERT INTO Specification VALUES ('SP04','Multimedia',420000);

1 row created.

SQL> INSERT INTO Specification VALUES ('SP05','Finance',500000);

1 row created.

SQL> INSERT INTO Specification VALUES ('SP06','International Business',480000);

1 row created.

SQL> INSERT INTO Specification VALUES ('SP07','Accounting',420000);

1 row created.

SQL> INSERT INTO Specification VALUES ('SP08','Digital Marketing',450000);

1 row created.

SQL> INSERT INTO Specification VALUES ('SP09','Tourism',400000);

1 row created.

SQL> INSERT INTO Specification VALUES ('SP10','Human Resources',400000);

1 row created.
```

Figure 47 Inserting values into Specification table

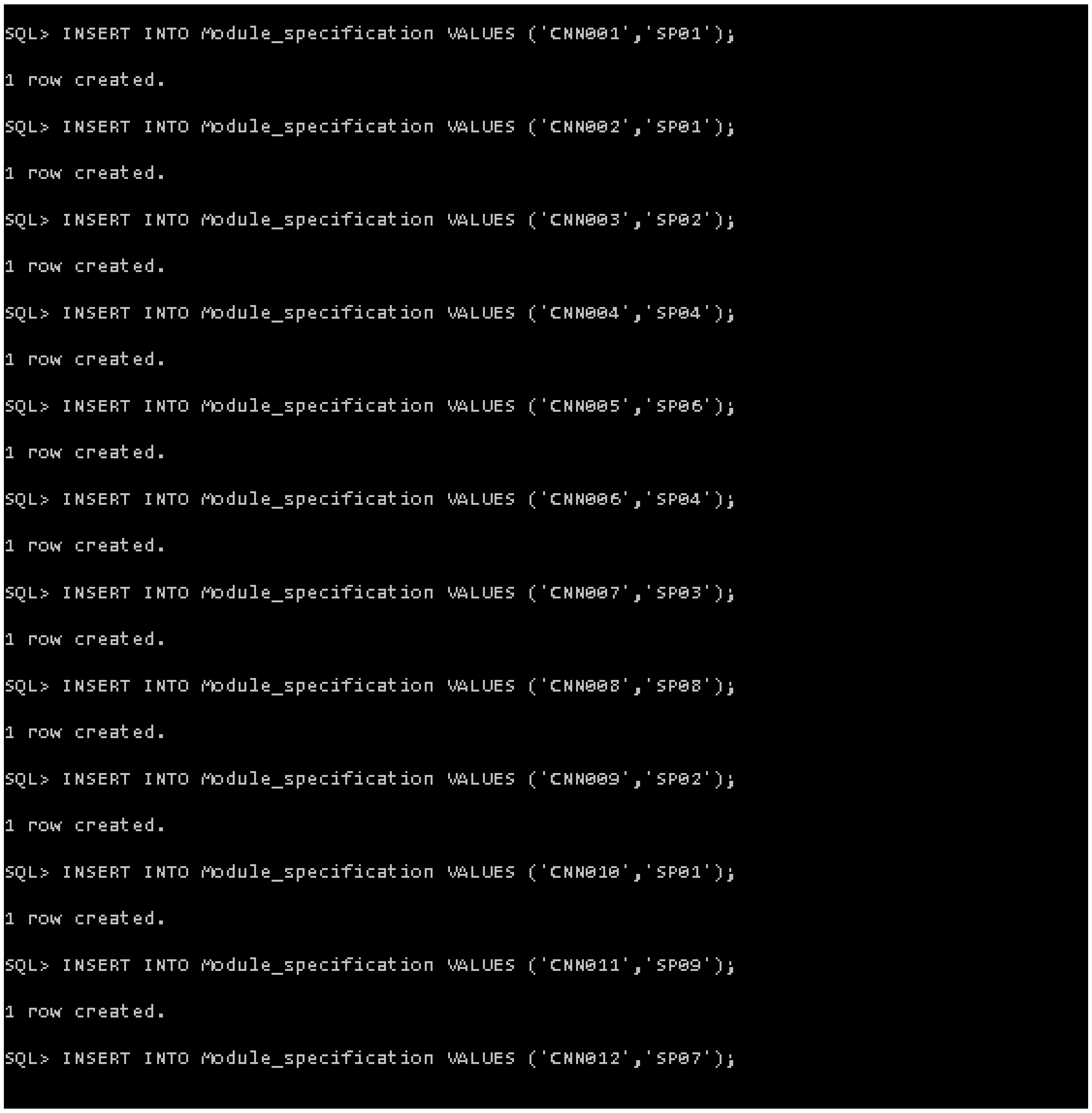
- **Inserting values into Module:**



```
SQL> INSERT INTO Module VALUES ('CNN001','Database');

1 row created.

SQL> INSERT INTO Module VALUES ('CNN002','Programming');

1 row created.

SQL> INSERT INTO Module VALUES ('CNN003','Hardware');

1 row created.

SQL> INSERT INTO Module VALUES ('CNN004','Economic and Society');

1 row created.

SQL> INSERT INTO Module VALUES ('CNN005','Business Statics');

1 row created.

SQL> INSERT INTO Module VALUES ('CNN006','Human Resource Management');

1 row created.

SQL> INSERT INTO Module VALUES ('CNN007','HTML basic');

1 row created.

SQL> INSERT INTO Module VALUES ('CNN008','Digital  Editing');

1 row created.

SQL> INSERT INTO Module VALUES ('CNN009','Computer Architecture');

1 row created.

SQL> INSERT INTO Module VALUES ('CNN010','Logic and problem Solving');

1 row created.
```

Figure 48 Inserting values into Module table

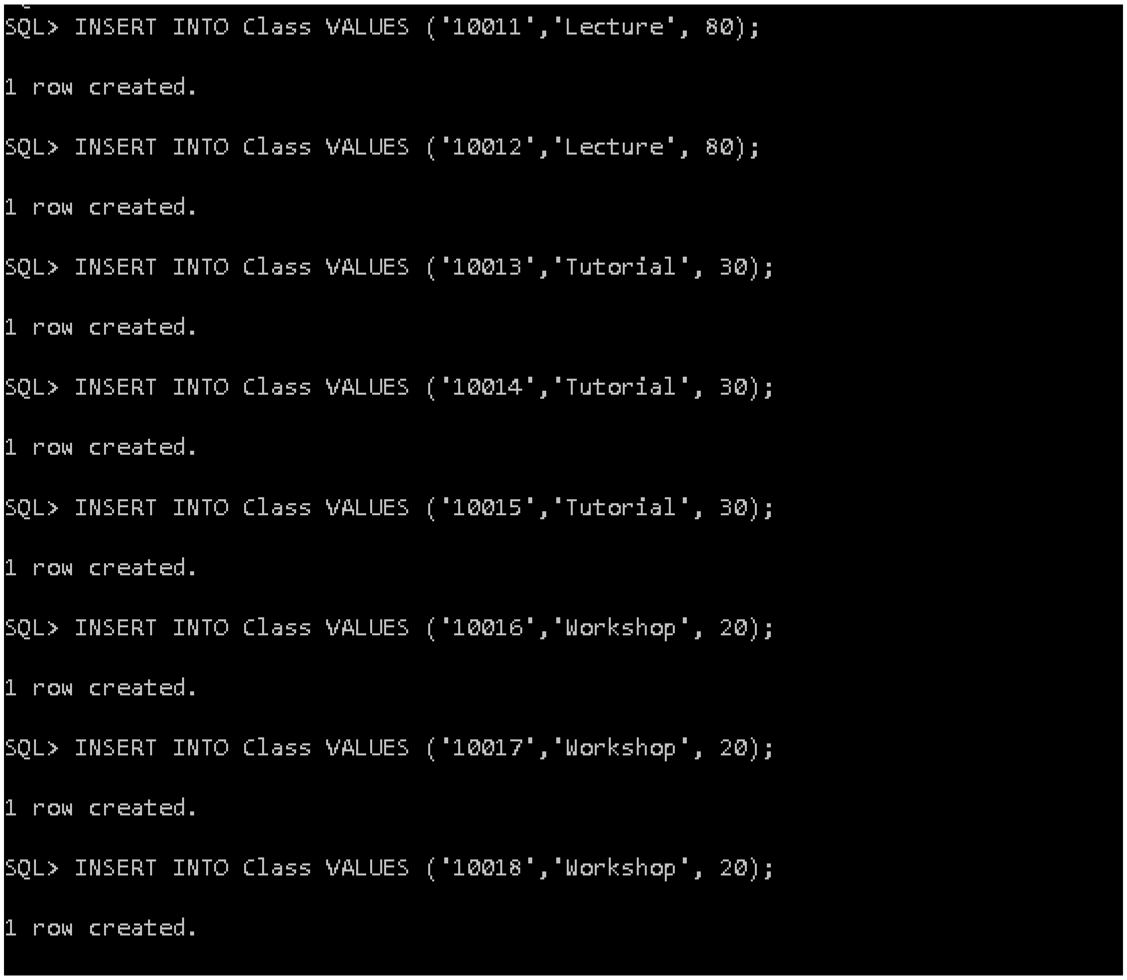
- **Inserting values into Module_specification:**



```
SQL> INSERT INTO Module_specification VALUES ('CNN001','SP01');

1 row created.

SQL> INSERT INTO Module_specification VALUES ('CNN002','SP01');

1 row created.

SQL> INSERT INTO Module_specification VALUES ('CNN003','SP02');

1 row created.

SQL> INSERT INTO Module_specification VALUES ('CNN004','SP04');

1 row created.

SQL> INSERT INTO Module_specification VALUES ('CNN005','SP06');

1 row created.

SQL> INSERT INTO Module_specification VALUES ('CNN006','SP04');

1 row created.

SQL> INSERT INTO Module_specification VALUES ('CNN007','SP03');

1 row created.

SQL> INSERT INTO Module_specification VALUES ('CNN008','SP08');

1 row created.

SQL> INSERT INTO Module_specification VALUES ('CNN009','SP02');

1 row created.

SQL> INSERT INTO Module_specification VALUES ('CNN010','SP01');

1 row created.

SQL> INSERT INTO Module_specification VALUES ('CNN011','SP09');

1 row created.

SQL> INSERT INTO Module_specification VALUES ('CNN012','SP07');
```

Figure 49 Inserting values into Module_specification table

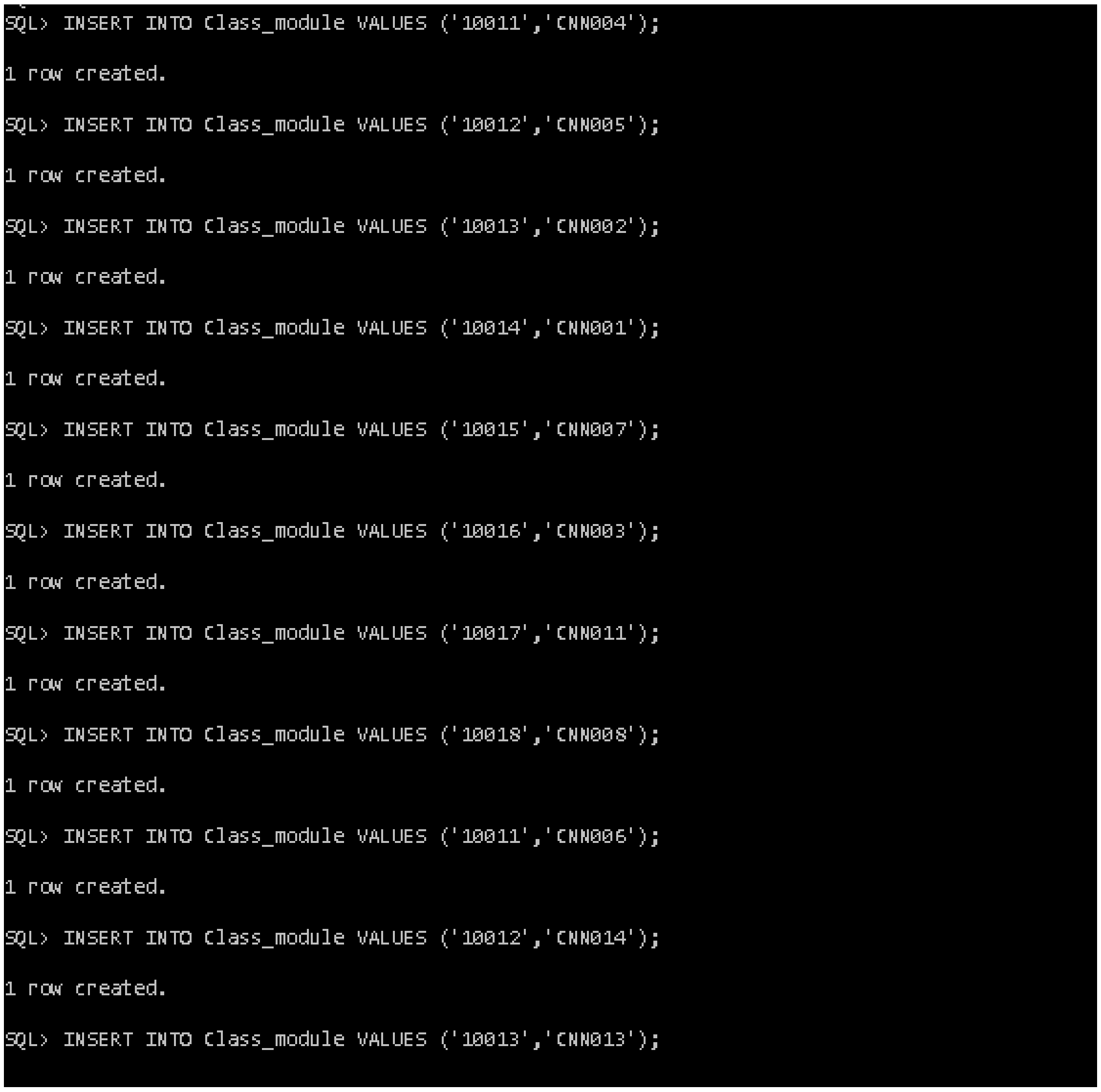
- **Inserting values into Class:**



```
SQL> INSERT INTO Class VALUES ('10011','Lecture', 80);

1 row created.

SQL> INSERT INTO Class VALUES ('10012','Lecture', 80);

1 row created.

SQL> INSERT INTO Class VALUES ('10013','Tutorial', 30);

1 row created.

SQL> INSERT INTO Class VALUES ('10014','Tutorial', 30);

1 row created.

SQL> INSERT INTO Class VALUES ('10015','Tutorial', 30);

1 row created.

SQL> INSERT INTO Class VALUES ('10016','Workshop', 20);

1 row created.

SQL> INSERT INTO Class VALUES ('10017','Workshop', 20);

1 row created.

SQL> INSERT INTO Class VALUES ('10018','Workshop', 20);

1 row created.
```

Figure 50 Inserting values into Class table

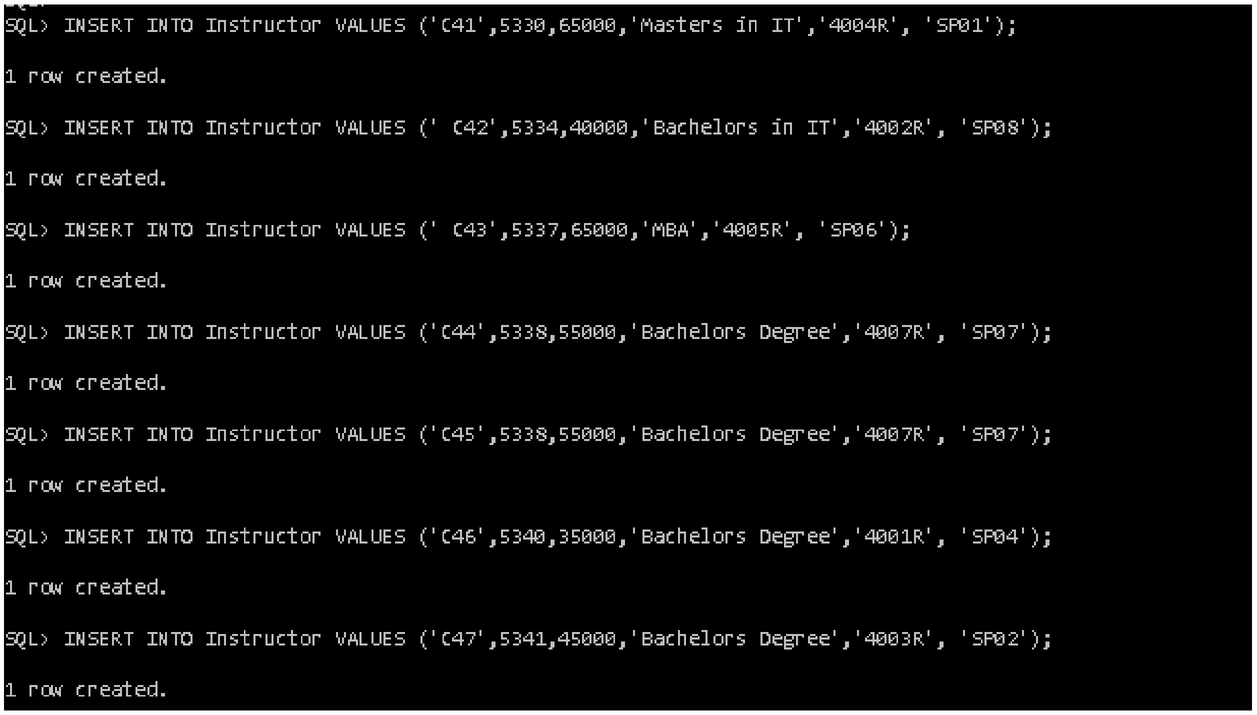
- **Inserting values into Class_module:**



```
SQL> INSERT INTO Class_module VALUES ('10011','CNN004');

1 row created.

SQL> INSERT INTO Class_module VALUES ('10012','CNN005');

1 row created.

SQL> INSERT INTO Class_module VALUES ('10013','CNN002');

1 row created.

SQL> INSERT INTO Class_module VALUES ('10014','CNN001');

1 row created.

SQL> INSERT INTO Class_module VALUES ('10015','CNN007');

1 row created.

SQL> INSERT INTO Class_module VALUES ('10016','CNN003');

1 row created.

SQL> INSERT INTO Class_module VALUES ('10017','CNN011');

1 row created.

SQL> INSERT INTO Class_module VALUES ('10018','CNN008');

1 row created.

SQL> INSERT INTO Class_module VALUES ('10011','CNN006');

1 row created.

SQL> INSERT INTO Class_module VALUES ('10012','CNN014');

1 row created.

SQL> INSERT INTO Class_module VALUES ('10013','CNN013');
```

Figure 51 Inserting values into Class_module table

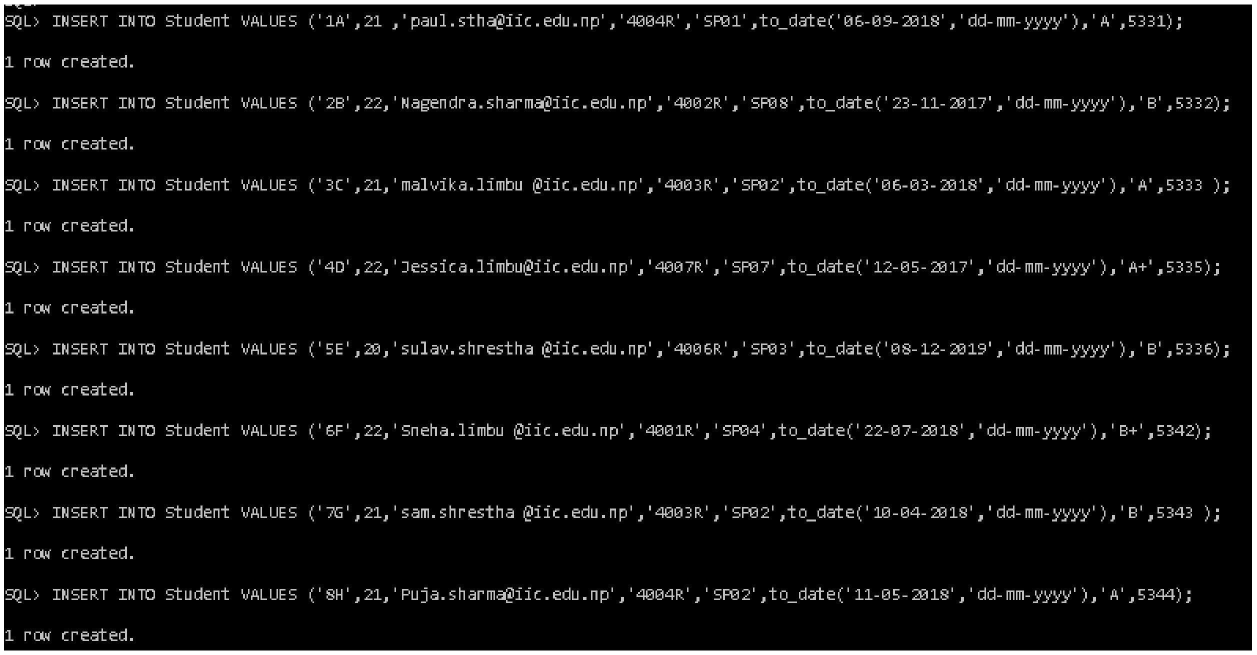
- **Inserting values into Instructor:**



```
SQL> INSERT INTO Instructor VALUES ('C41',5330,65000,'Masters in IT','4004R', 'SP01');

1 row created.

SQL> INSERT INTO Instructor VALUES (' C42',5334,40000,'Bachelors in IT','4002R', 'SP08');

1 row created.

SQL> INSERT INTO Instructor VALUES (' C43',5337,65000,'MBA','4005R', 'SP06');

1 row created.

SQL> INSERT INTO Instructor VALUES ('C44',5338,55000,'Bachelors Degree','4007R', 'SP07');

1 row created.

SQL> INSERT INTO Instructor VALUES ('C45',5338,55000,'Bachelors Degree','4007R', 'SP07');

1 row created.

SQL> INSERT INTO Instructor VALUES ('C46',5340,35000,'Bachelors Degree','4001R', 'SP04');

1 row created.

SQL> INSERT INTO Instructor VALUES ('C47',5341,45000,'Bachelors Degree','4003R', 'SP02');

1 row created.
```

Figure 52 Inserting values into Instructor table

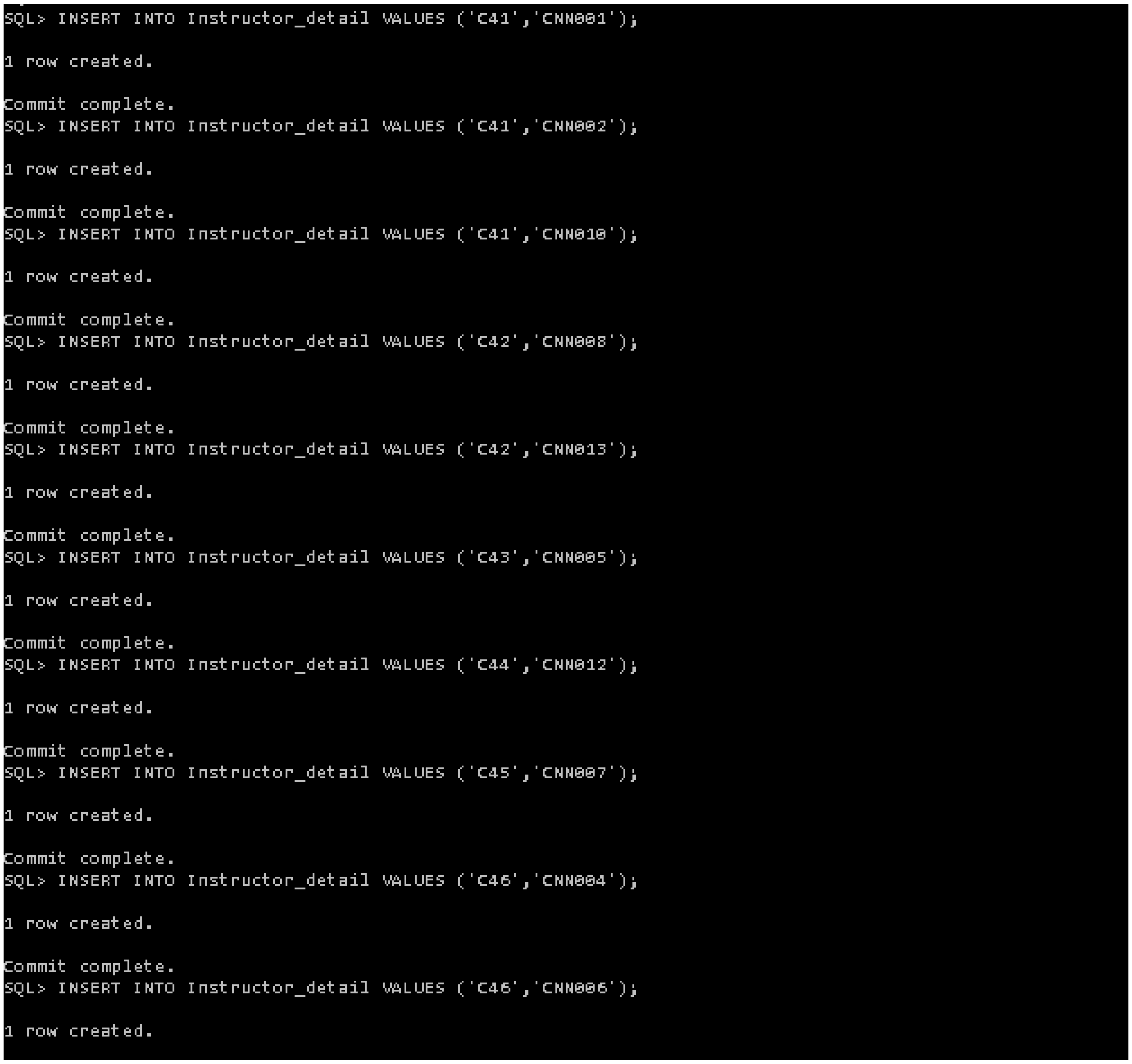
- **Inserting values into Student:**



```
SQL> INSERT INTO Student VALUES ('1A',21 ,'paul.stha@iic.edu.np','4004R','SP01',to_date('06-09-2018','dd-mm-yyyy'),'A',5331);

1 row created.

SQL> INSERT INTO Student VALUES ('2B',22,'Nagendra.sharma@iic.edu.np','4002R','SP08',to_date('23-11-2017','dd-mm-yyyy'),'B',5332);

1 row created.

SQL> INSERT INTO Student VALUES ('3C',21,'malvika.limbu @iic.edu.np','4003R','SP02',to_date('06-03-2018','dd-mm-yyyy'),'A',5333 );

1 row created.

SQL> INSERT INTO Student VALUES ('4D',22,'Jessica.limbu@iic.edu.np','4007R','SP07',to_date('12-05-2017','dd-mm-yyyy'),'A+',5335);

1 row created.

SQL> INSERT INTO Student VALUES ('5E',20,'sulav.shrestha @iic.edu.np','4006R','SP03',to_date('08-12-2019','dd-mm-yyyy'),'B',5336);

1 row created.

SQL> INSERT INTO Student VALUES ('6F',22,'Sneha.limbu @iic.edu.np','4001R','SP04',to_date('22-07-2018','dd-mm-yyyy'),'B+',5342);

1 row created.

SQL> INSERT INTO Student VALUES ('7G',21,'sam.shrestha @iic.edu.np','4003R','SP02',to_date('10-04-2018','dd-mm-yyyy'),'B',5343 );

1 row created.

SQL> INSERT INTO Student VALUES ('8H',21,'Puja.sharma@iic.edu.np','4004R','SP02',to_date('11-05-2018','dd-mm-yyyy'),'A',5344);

1 row created.
```

Figure 53 Inserting values into Student table

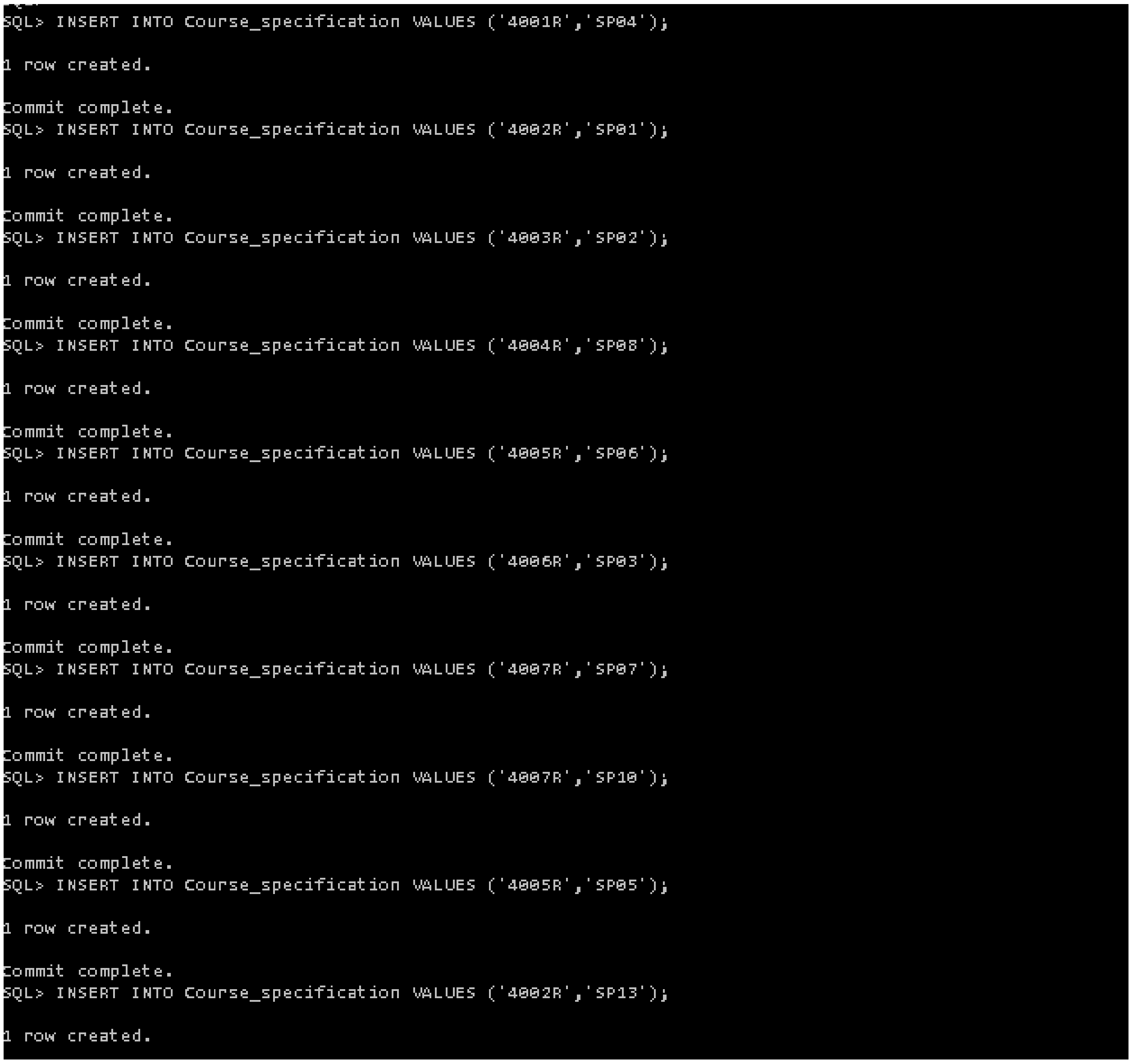
- **Inserting values into Instructor_detail:**



```
SQL> INSERT INTO Instructor_detail VALUES ('C41','CNN001');

1 row created.

Commit complete.
SQL> INSERT INTO Instructor_detail VALUES ('C41','CNN002');

1 row created.

Commit complete.
SQL> INSERT INTO Instructor_detail VALUES ('C41','CNN010');

1 row created.

Commit complete.
SQL> INSERT INTO Instructor_detail VALUES ('C42','CNN008');

1 row created.

Commit complete.
SQL> INSERT INTO Instructor_detail VALUES ('C42','CNN013');

1 row created.

Commit complete.
SQL> INSERT INTO Instructor_detail VALUES ('C43','CNN005');

1 row created.

Commit complete.
SQL> INSERT INTO Instructor_detail VALUES ('C44','CNN012');

1 row created.

Commit complete.
SQL> INSERT INTO Instructor_detail VALUES ('C45','CNN007');

1 row created.

Commit complete.
SQL> INSERT INTO Instructor_detail VALUES ('C46','CNN004');

1 row created.

Commit complete.
SQL> INSERT INTO Instructor_detail VALUES ('C46','CNN006');

1 row created.
```

Figure 54 Inserting values into Instructor_detail table

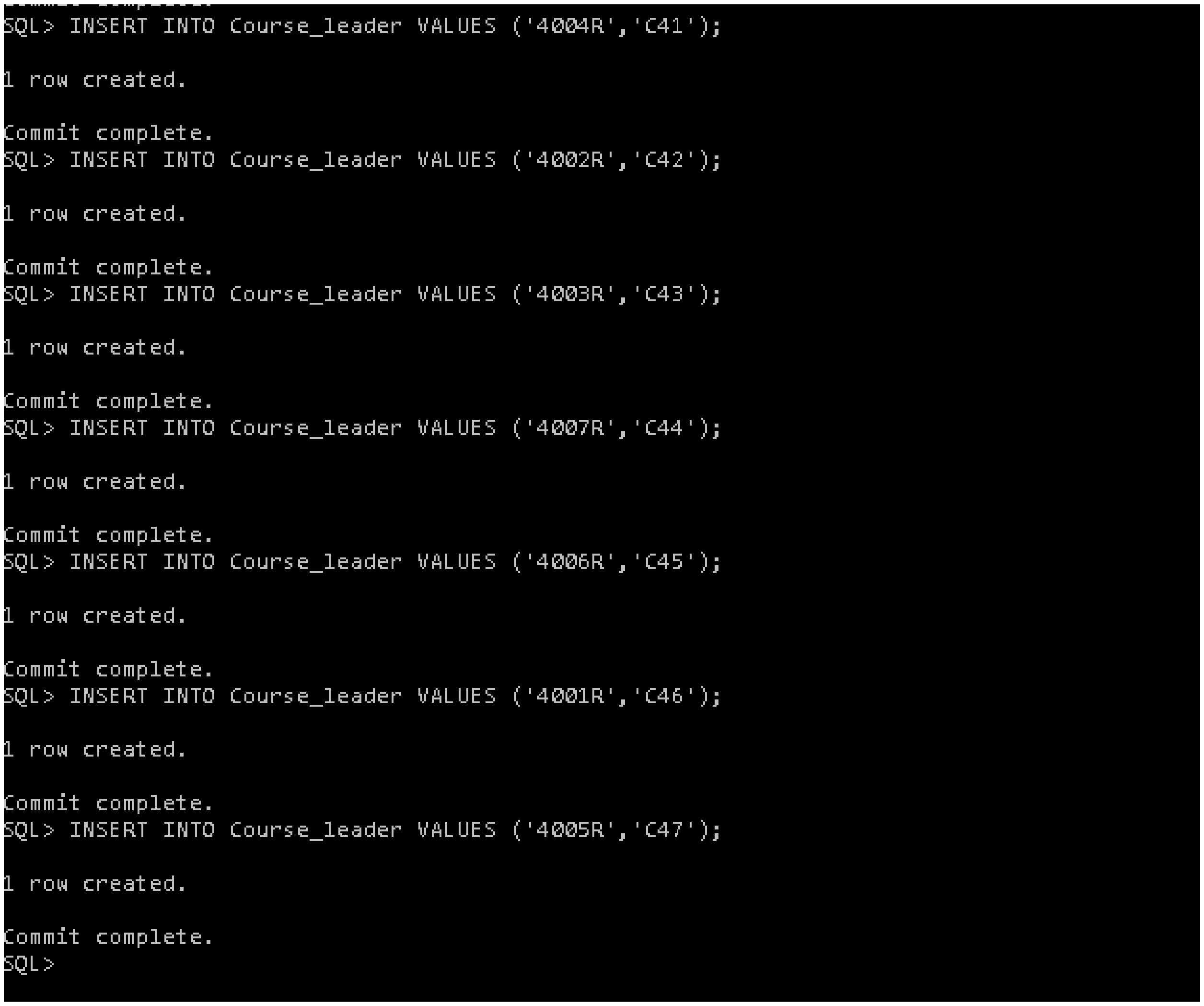
- **Inserting values into Course_specification:**



```
SQL> INSERT INTO Course_specification VALUES ('4001R','SP04');

1 row created.

Commit complete.
SQL> INSERT INTO Course_specification VALUES ('4002R','SP01');

1 row created.

Commit complete.
SQL> INSERT INTO Course_specification VALUES ('4003R','SP02');

1 row created.

Commit complete.
SQL> INSERT INTO Course_specification VALUES ('4004R','SP08');

1 row created.

Commit complete.
SQL> INSERT INTO Course_specification VALUES ('4005R','SP06');

1 row created.

Commit complete.
SQL> INSERT INTO Course_specification VALUES ('4006R','SP03');

1 row created.

Commit complete.
SQL> INSERT INTO Course_specification VALUES ('4007R','SP07');

1 row created.

Commit complete.
SQL> INSERT INTO Course_specification VALUES ('4007R','SP10');

1 row created.

Commit complete.
SQL> INSERT INTO Course_specification VALUES ('4005R','SP05');

1 row created.

Commit complete.
SQL> INSERT INTO Course_specification VALUES ('4002R','SP13');

1 row created.
```

Figure 55 Inserting values into Course_specification table

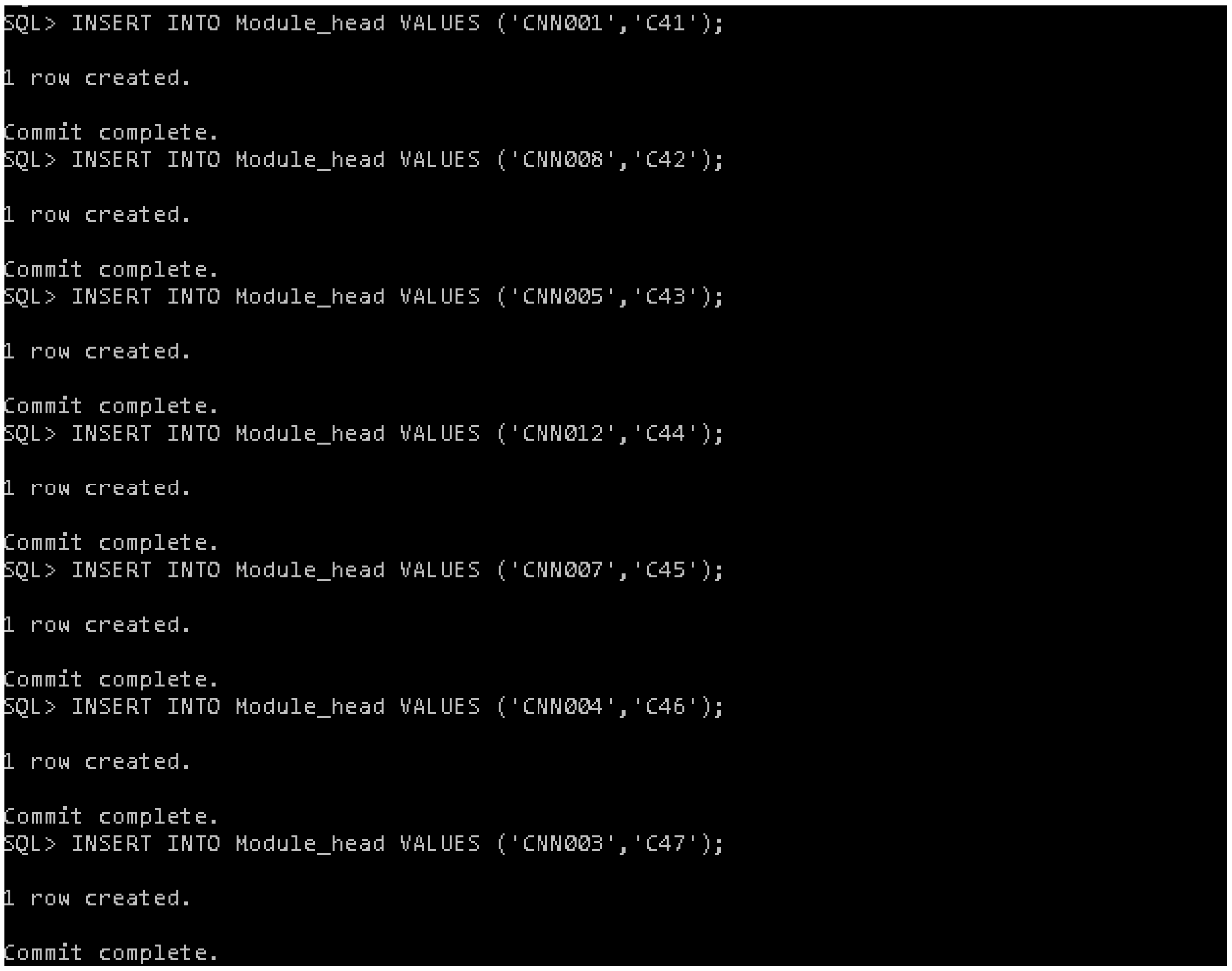
- **Inserting values into Course_leader:**



```
SQL> INSERT INTO Course_leader VALUES ('4004R','C41');

1 row created.

Commit complete.
SQL> INSERT INTO Course_leader VALUES ('4002R','C42');

1 row created.

Commit complete.
SQL> INSERT INTO Course_leader VALUES ('4003R','C43');

1 row created.

Commit complete.
SQL> INSERT INTO Course_leader VALUES ('4007R','C44');

1 row created.

Commit complete.
SQL> INSERT INTO Course_leader VALUES ('4006R','C45');

1 row created.

Commit complete.
SQL> INSERT INTO Course_leader VALUES ('4001R','C46');

1 row created.

Commit complete.
SQL> INSERT INTO Course_leader VALUES ('4005R','C47');

1 row created.

Commit complete.
SQL>
```

Figure 56 Inserting values into Course_leader table

- **Inserting values into Module_head:**



```
SQL> INSERT INTO Module_head VALUES ('CNN001','C41');

1 row created.

Commit complete.
SQL> INSERT INTO Module_head VALUES ('CNN008','C42');

1 row created.

Commit complete.
SQL> INSERT INTO Module_head VALUES ('CNN005','C43');

1 row created.

Commit complete.
SQL> INSERT INTO Module_head VALUES ('CNN012','C44');

1 row created.

Commit complete.
SQL> INSERT INTO Module_head VALUES ('CNN007','C45');

1 row created.

Commit complete.
SQL> INSERT INTO Module_head VALUES ('CNN004','C46');

1 row created.

Commit complete.
SQL> INSERT INTO Module_head VALUES ('CNN003','C47');

1 row created.

Commit complete.
```

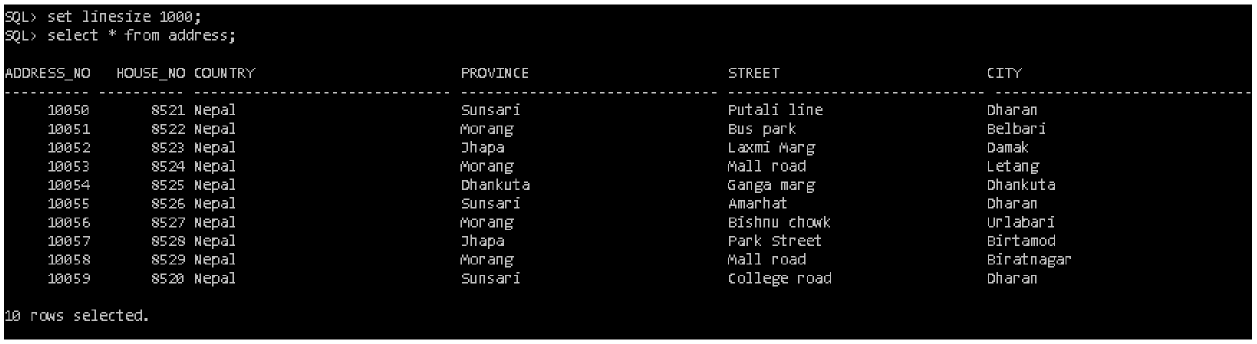
Figure 57 Inserting values into Module_head table

4.3 Final Tables:

For Fetching data in database ‘SELECT’ command is used.

- Address table:

  SELECT * FROM Address;

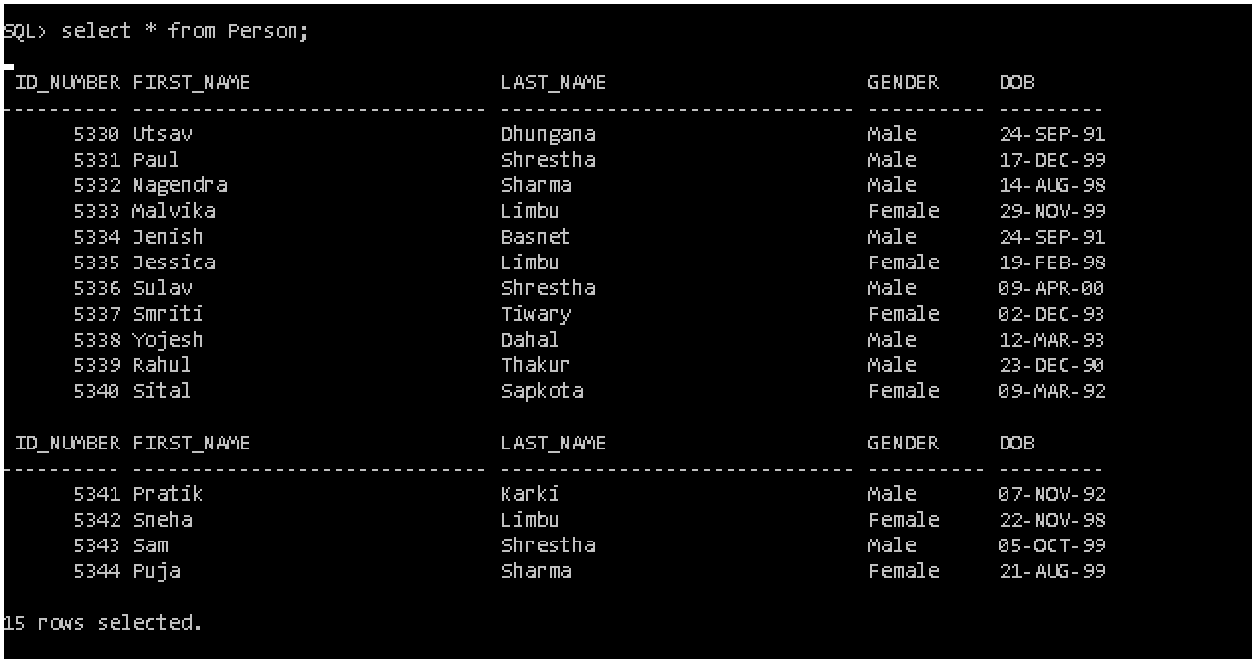


```
SQL> set linesize 1000;
SQL> select * from address;

ADDRESS_NO   HOUSE_NO COUNTRY                        PROVINCE                       STREET                         CITY
---------- ---------- ------------------------------ ------------------------------ ------------------------------ ------------------------------
     10050       8521 Nepal                          Sunsari                        Putali line                    Dharan
     10051       8522 Nepal                          Morang                         Bus park                       Belbari
     10052       8523 Nepal                          Jhapa                          Laxmi Marg                     Damak
     10053       8524 Nepal                          Morang                         Mall road                      Letang
     10054       8525 Nepal                          Dhankuta                       Ganga marg                     Dhankuta
     10055       8526 Nepal                          Sunsari                        Amarhat                        Dharan
     10056       8527 Nepal                          Morang                         Bishnu chowk                   Urlabari
     10057       8528 Nepal                          Jhapa                          Park Street                    Birtamod
     10058       8529 Nepal                          Morang                         Mall road                      Biratnagar
     10059       8520 Nepal                          Sunsari                        College road                   Dharan

10 rows selected.
```

Figure 58 Address table

- Person table:

  SELECT * FROM Person;

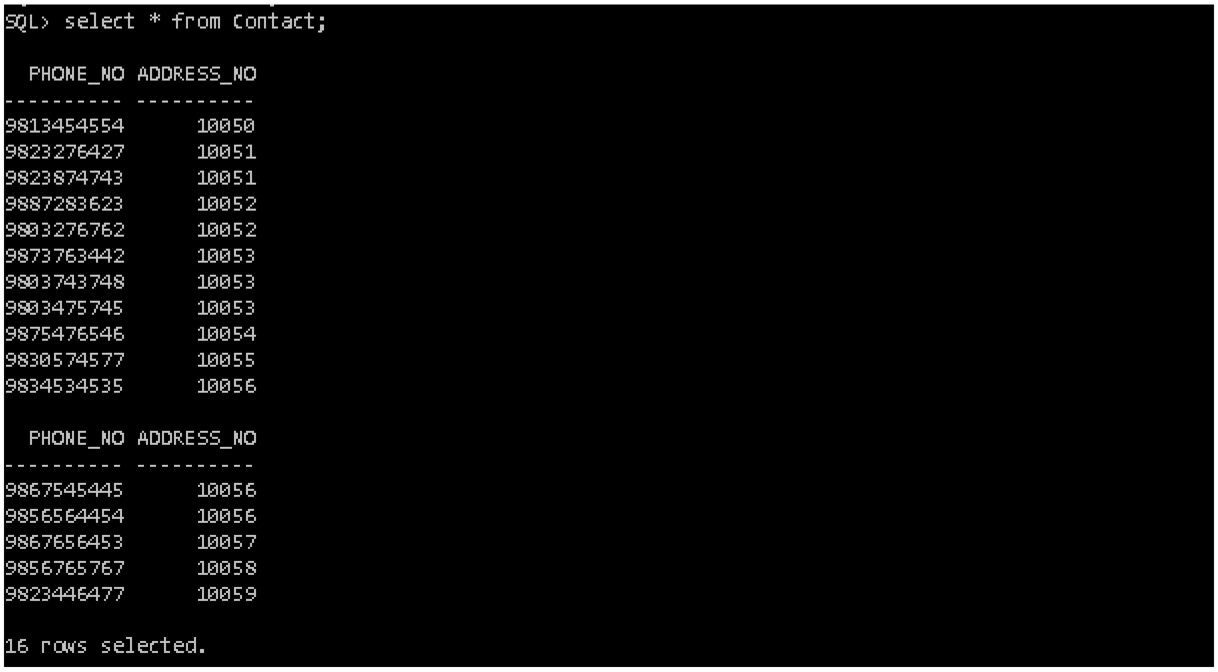


```
SQL> select * from Person;

 ID_NUMBER FIRST_NAME                     LAST_NAME                      GENDER     DOB
---------- ------------------------------ ------------------------------ ---------- ---------
      5330 Utsav                          Dhungana                       Male       24-SEP-91
      5331 Paul                           Shrestha                       Male       17-DEC-99
      5332 Nagendra                       Sharma                         Male       14-AUG-98
      5333 Malvika                        Limbu                          Female     29-NOV-99
      5334 Jenish                         Basnet                         Male       24-SEP-91
      5335 Jessica                        Limbu                          Female     19-FEB-98
      5336 Sulav                          Shrestha                       Male       09-APR-00
      5337 Smriti                         Tiwary                         Female     02-DEC-93
      5338 Yojesh                         Dahal                          Male       12-MAR-93
      5339 Rahul                          Thakur                         Male       23-DEC-90
      5340 Sital                          Sapkota                        Female     09-MAR-92

 ID_NUMBER FIRST_NAME                     LAST_NAME                      GENDER     DOB
---------- ------------------------------ ------------------------------ ---------- ---------
      5341 Pratik                         Karki                          Male       07-NOV-92
      5342 Sneha                          Limbu                          Female     22-NOV-98
      5343 Sam                            Shrestha                       Male       05-OCT-99
      5344 Puja                           Sharma                         Female     21-AUG-99

15 rows selected.
```

Figure 59 Person table

- Contact table:

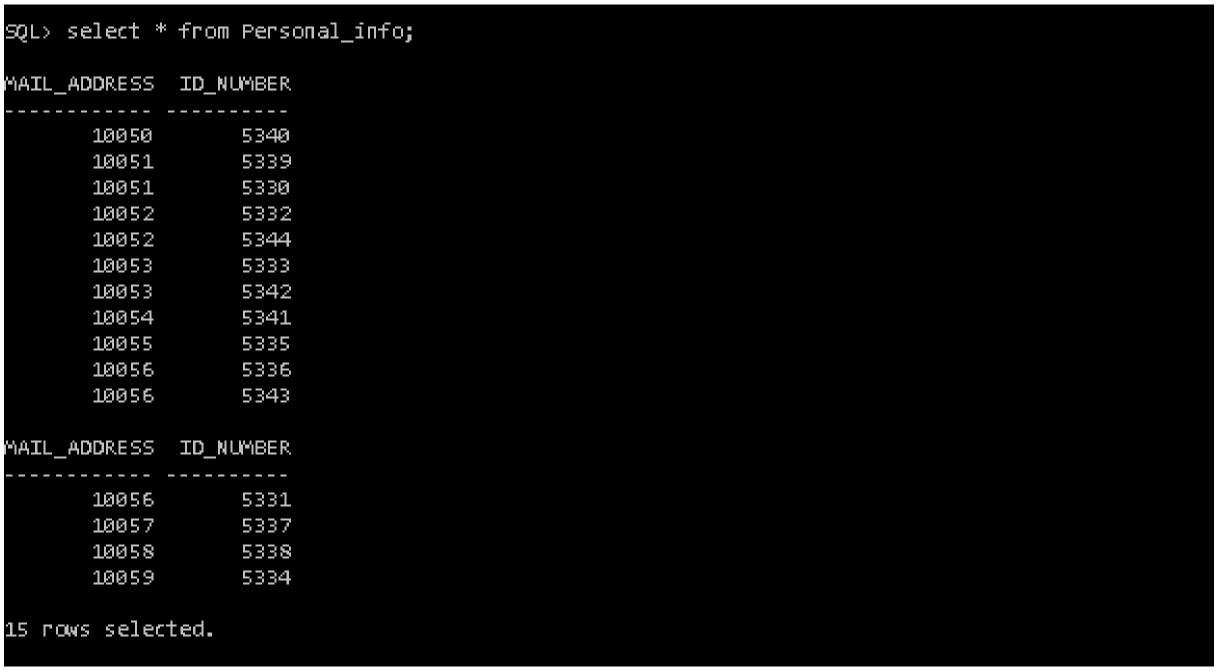
  SELECT * FROM Contact;



```
SQL> select * from Contact;

  PHONE_NO ADDRESS_NO
---------- ----------
9813454554      10050
9823276427      10051
9823874743      10051
9887283623      10052
9803276762      10052
9873763442      10053
9803743748      10053
9803475745      10053
9875476546      10054
9830574577      10055
9834534535      10056

  PHONE_NO ADDRESS_NO
---------- ----------
9867545445      10056
9856564454      10056
9867656453      10057
9856765767      10058
9823446477      10059

16 rows selected.
```

Figure 60 Contact table

- Personal_info table:
  SELECT * FROM Personal_info;

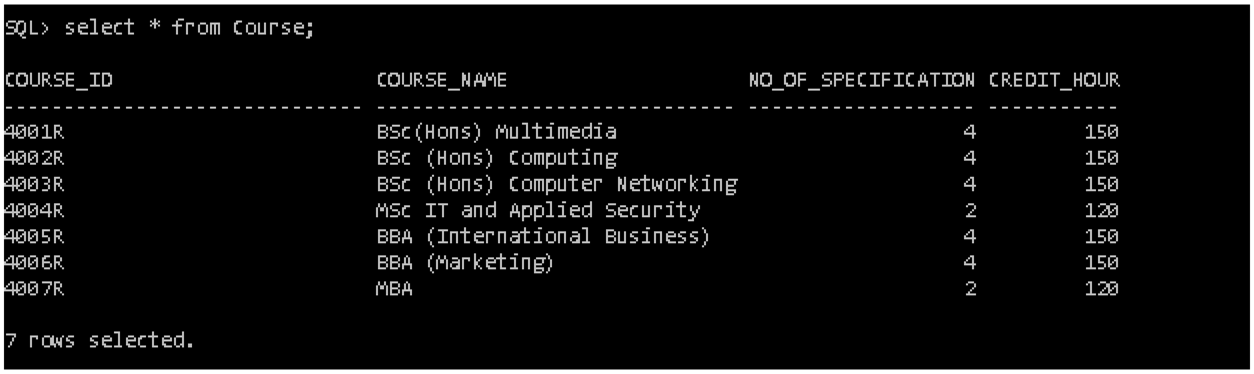


```
SQL> select * from Personal_info;

MAIL_ADDRESS  ID_NUMBER
------------ ----------
       10050       5340
       10051       5339
       10051       5330
       10052       5332
       10052       5344
       10053       5333
       10053       5342
       10054       5341
       10055       5335
       10056       5336
       10056       5343

MAIL_ADDRESS  ID_NUMBER
------------ ----------
       10056       5331
       10057       5337
       10058       5338
       10059       5334

15 rows selected.
```

Figure 61 Personal_info table

- Course table:

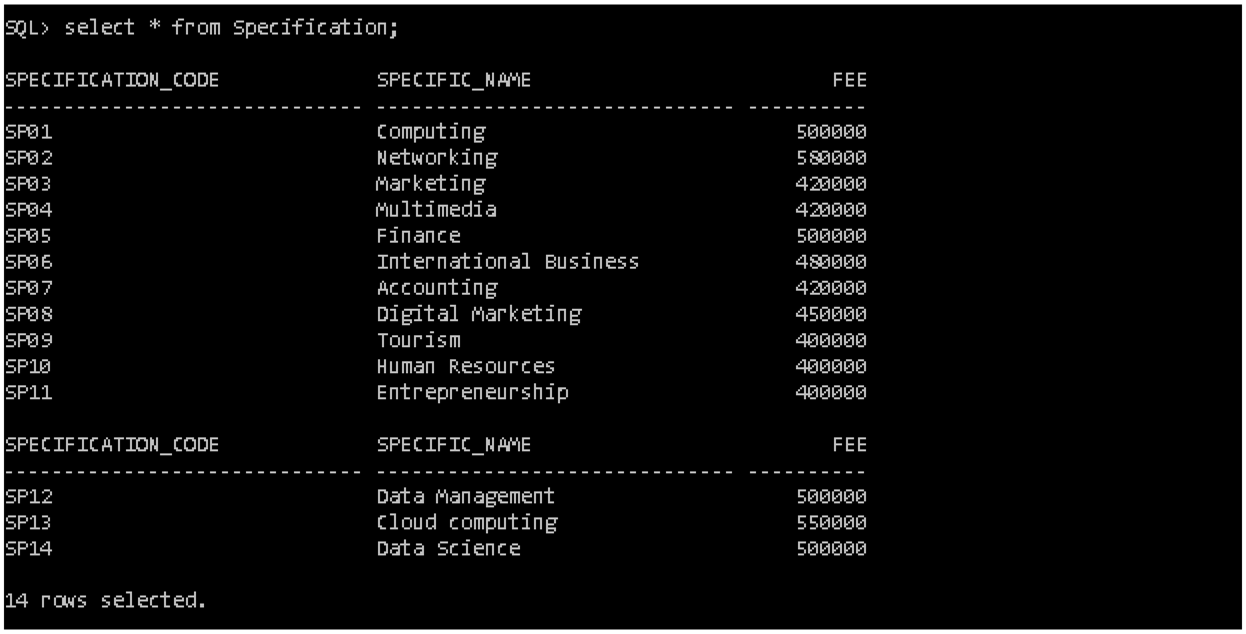
SELECT * FROM Course;



```
SQL> select * from Course;

COURSE_ID                      COURSE_NAME                    NO_OF_SPECIFICATION CREDIT_HOUR
------------------------------ ------------------------------ ------------------- -----------
4001R                          BSc(Hons) Multimedia                             4         150
4002R                          BSc (Hons) Computing                             4         150
4003R                          BSc (Hons) Computer Networking                   4         150
4004R                          MSc IT and Applied Security                      2         120
4005R                          BBA (International Business)                     4         150
4006R                          BBA (Marketing)                                  4         150
4007R                          MBA                                              2         120

7 rows selected.
```

Figure 62 Course table

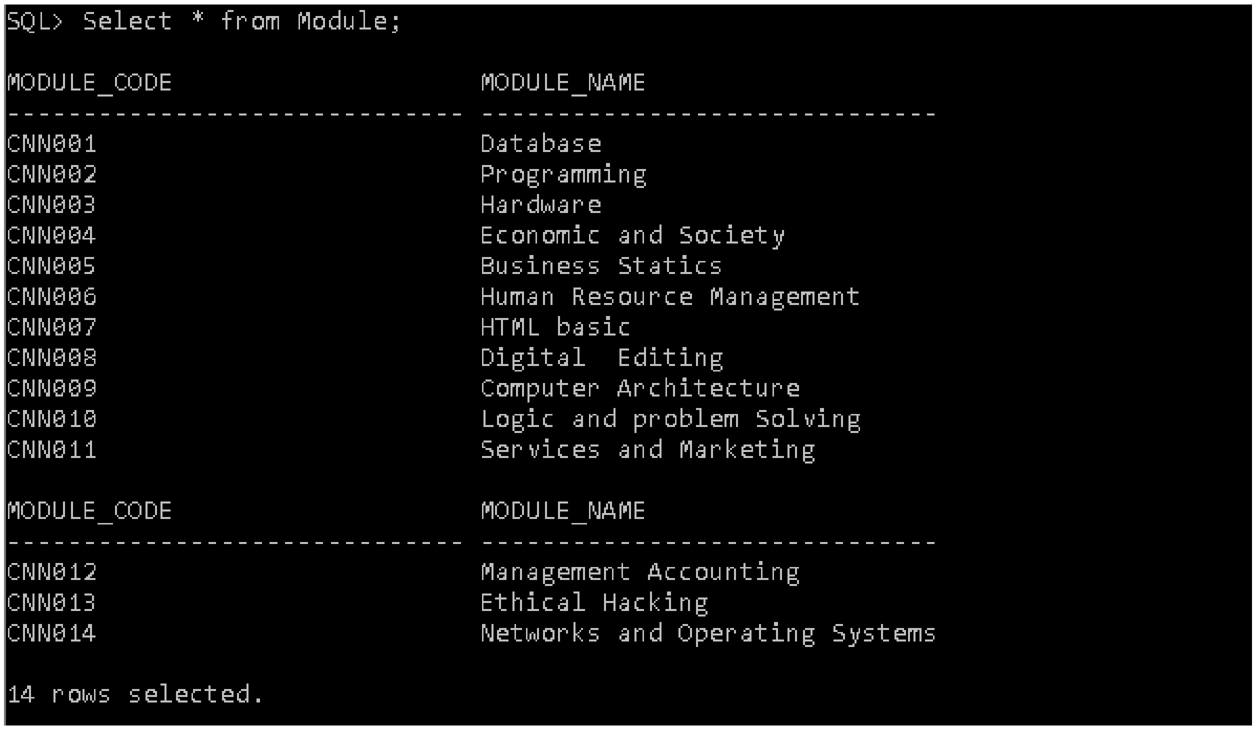
- Specification Table:
  SELECT * FROM Specification;



```
SQL> select * from Specification;

SPECIFICATION_CODE             SPECIFIC_NAME                         FEE
------------------------------ ------------------------------ ----------
SP01                           Computing                          500000
SP02                           Networking                         580000
SP03                           Marketing                          420000
SP04                           Multimedia                         420000
SP05                           Finance                            500000
SP06                           International Business             480000
SP07                           Accounting                         420000
SP08                           Digital Marketing                  450000
SP09                           Tourism                            400000
SP10                           Human Resources                    400000
SP11                           Entrepreneurship                   400000

SPECIFICATION_CODE             SPECIFIC_NAME                         FEE
------------------------------ ------------------------------ ----------
SP12                           Data Management                    500000
SP13                           Cloud computing                    550000
SP14                           Data Science                       500000

14 rows selected.
```

Figure 63 Specification table

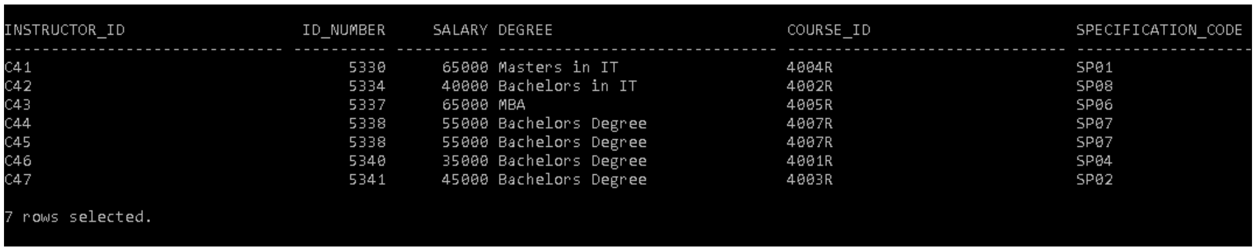
- Module table
  SELECT * FROM Module;



```
SQL> Select * from Module;

MODULE_CODE                    MODULE_NAME
------------------------------ ------------------------------
CNN001                         Database
CNN002                         Programming
CNN003                         Hardware
CNN004                         Economic and Society
CNN005                         Business Statics
CNN006                         Human Resource Management
CNN007                         HTML basic
CNN008                         Digital  Editing
CNN009                         Computer Architecture
CNN010                         Logic and problem Solving
CNN011                         Services and Marketing

MODULE_CODE                    MODULE_NAME
------------------------------ ------------------------------
CNN012                         Management Accounting
CNN013                         Ethical Hacking
CNN014                         Networks and Operating Systems

14 rows selected.
```

Figure 64 Module table

- Instructor table:
  SELECT * FROM Instructor;

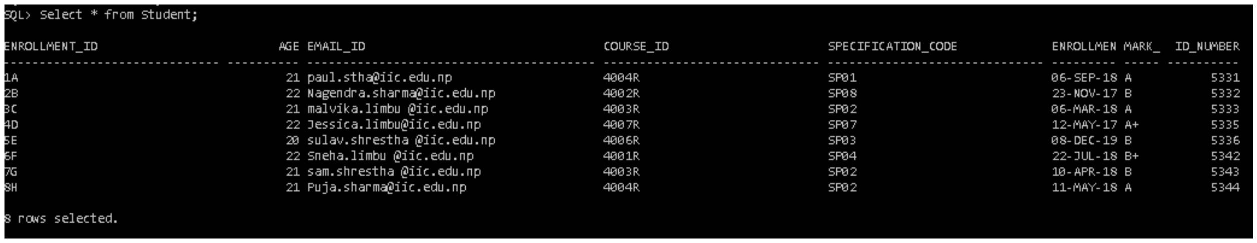


```
INSTRUCTOR_ID                   ID_NUMBER     SALARY DEGREE                       COURSE_ID                      SPECIFICATION_CODE
------------------------------ ---------- ---------- ---------------------------- ------------------------------ ------------------
C41                                  5330      65000 Masters in IT                4004R                          SP01
C42                                  5334      40000 Bachelors in IT              4002R                          SP08
C43                                  5337      65000 MBA                          4005R                          SP06
C44                                  5338      55000 Bachelors Degree             4007R                          SP07
C45                                  5338      55000 Bachelors Degree             4007R                          SP07
C46                                  5340      35000 Bachelors Degree             4001R                          SP04
C47                                  5341      45000 Bachelors Degree             4003R                          SP02

7 rows selected.
```

Figure 65 Instructor table

- Student table:

  SELECT * FROM Student;

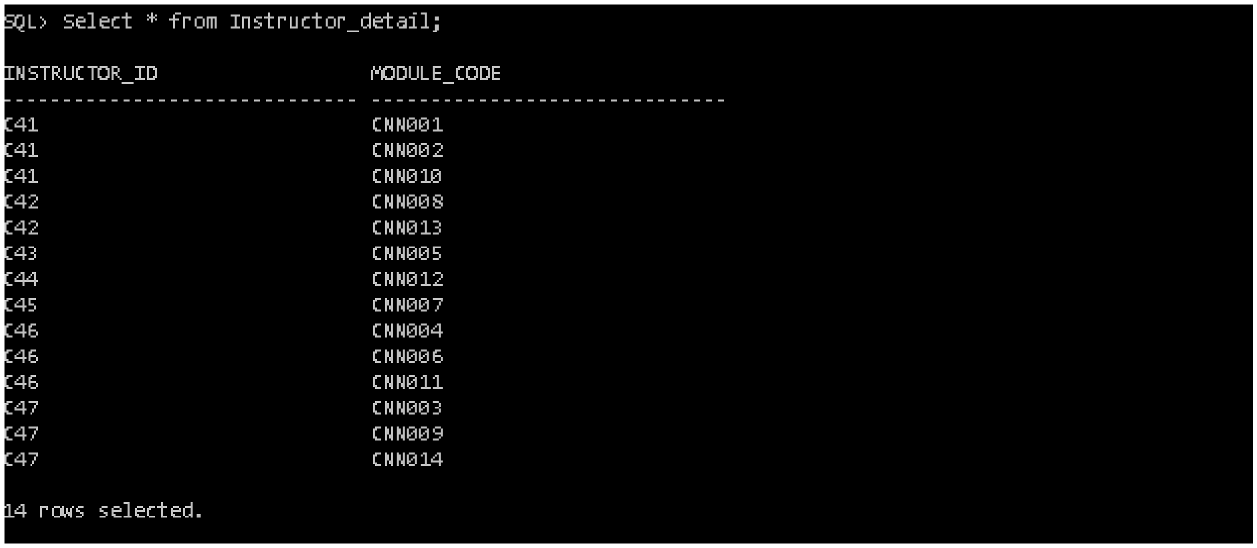


```
SQL> Select * from Student;

ENROLLMENT_ID                        AGE EMAIL_ID                            COURSE_ID                    SPECIFICATION_CODE           ENROLLMEN MARK_  ID_NUMBER
---------------------------- ---------- ------------------------------------ ---------------------------- ---------------------------- --------- ----- ----------
1A                                    21 paul.stha@iic.edu.np                4004R                        SP01                         06-SEP-18 A           5331
2B                                    22 Nagendra.sharma@iic.edu.np          4002R                        SP08                         23-NOV-17 B           5332
3C                                    21 malvika.limbu @iic.edu.np           4003R                        SP02                         06-MAR-18 A           5333
4D                                    22 Jessica.limbu@iic.edu.np            4007R                        SP07                         12-MAY-17 A+          5335
5E                                    20 sulav.shrestha @iic.edu.np          4006R                        SP03                         08-DEC-19 B           5336
6F                                    22 Sneha.limbu @iic.edu.np             4001R                        SP04                         22-JUL-18 B+          5342
7G                                    21 sam.shrestha @iic.edu.np            4003R                        SP02                         10-APR-18 B           5343
8H                                    21 Puja.sharma@iic.edu.np              4004R                        SP02                         11-MAY-18 A           5344

8 rows selected.
```

Figure 66 Student table

- Instructor_detail:

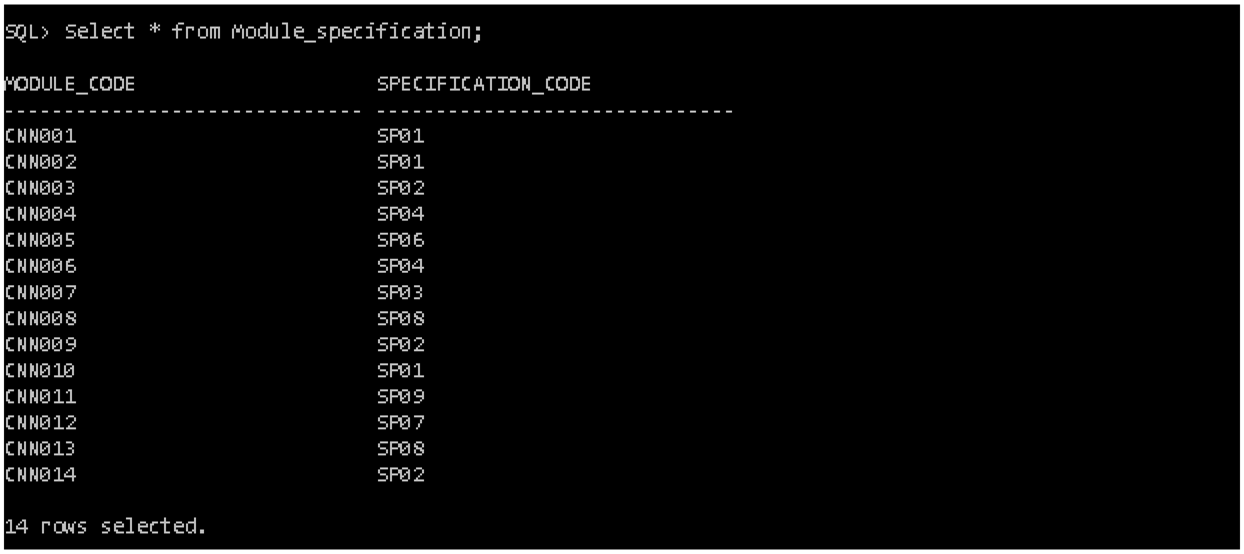
  SELECT * FROM Instructor_detail;



```
SQL> Select * from Instructor_detail;

INSTRUCTOR_ID                  MODULE_CODE
------------------------------ ------------------------------
C41                            CNN001
C41                            CNN002
C41                            CNN010
C42                            CNN008
C42                            CNN013
C43                            CNN005
C44                            CNN012
C45                            CNN007
C46                            CNN004
C46                            CNN006
C46                            CNN011
C47                            CNN003
C47                            CNN009
C47                            CNN014

14 rows selected.
```

Figure 67 Instructor_detail table

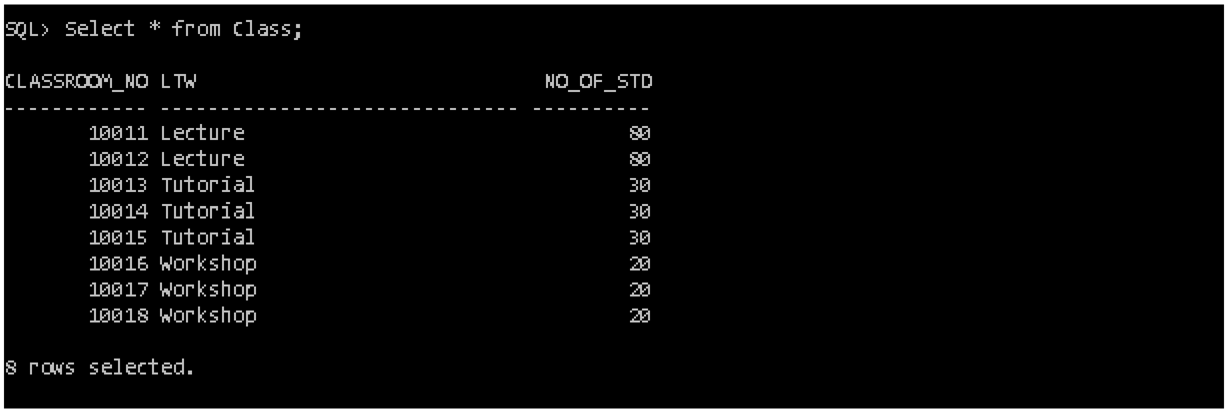
- Module_specification table:
  SELECT * FROM Module_specification;



```
SQL> Select * from Module_specification;

MODULE_CODE                    SPECIFICATION_CODE
------------------------------ ------------------------------
CNN001                         SP01
CNN002                         SP01
CNN003                         SP02
CNN004                         SP04
CNN005                         SP06
CNN006                         SP04
CNN007                         SP03
CNN008                         SP08
CNN009                         SP02
CNN010                         SP01
CNN011                         SP09
CNN012                         SP07
CNN013                         SP08
CNN014                         SP02

14 rows selected.
```

Figure 68 Module_specification table

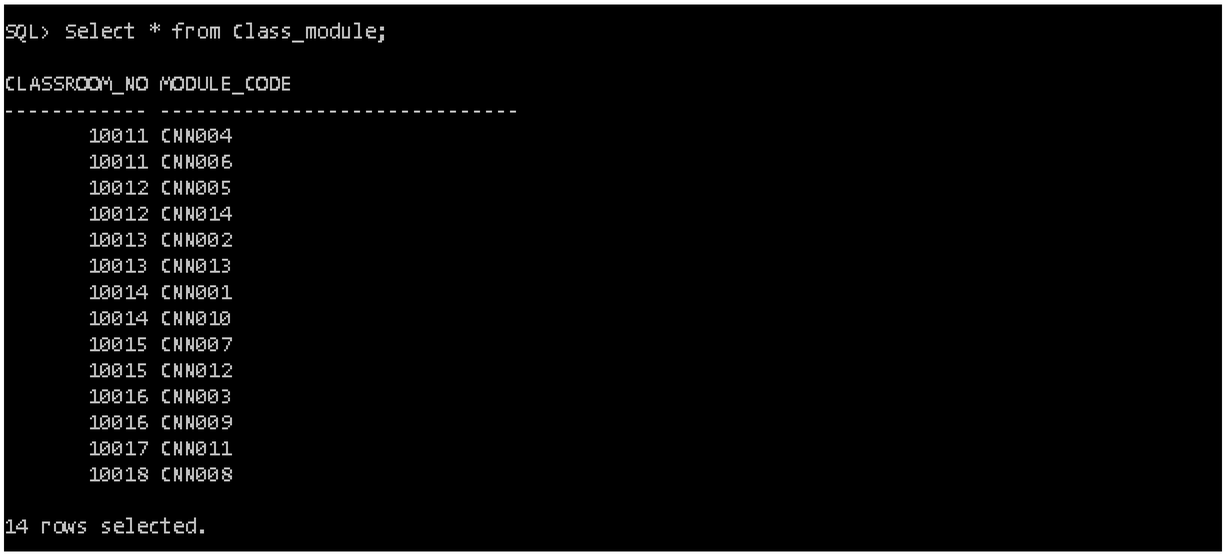
- Class table:
  SELECT * FROM Class;



```
SQL> Select * from Class;

CLASSROOM_NO LTW                              NO_OF_STD
------------ ------------------------------ ----------
       10011 Lecture                                80
       10012 Lecture                                80
       10013 Tutorial                               30
       10014 Tutorial                               30
       10015 Tutorial                               30
       10016 Workshop                               20
       10017 Workshop                               20
       10018 Workshop                               20

8 rows selected.
```

Figure 69 Class table

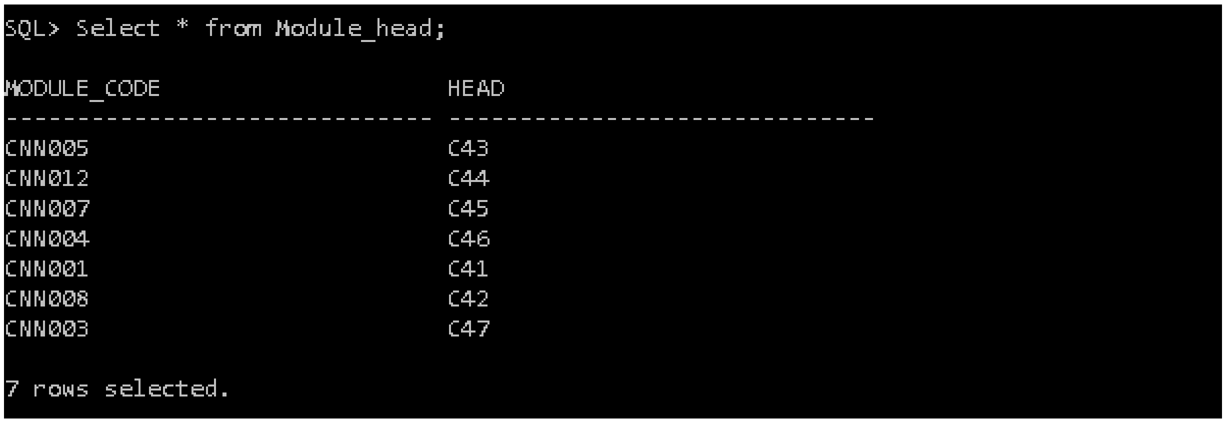
- Class_module:
  SELECT * FROM Class_module;



```
SQL> Select * from Class_module;

CLASSROOM_NO MODULE_CODE
------------ ------------------------------
       10011 CNN004
       10011 CNN006
       10012 CNN005
       10012 CNN014
       10013 CNN002
       10013 CNN013
       10014 CNN001
       10014 CNN010
       10015 CNN007
       10015 CNN012
       10016 CNN003
       10016 CNN009
       10017 CNN011
       10018 CNN008

14 rows selected.
```

Figure 70 Class_module table

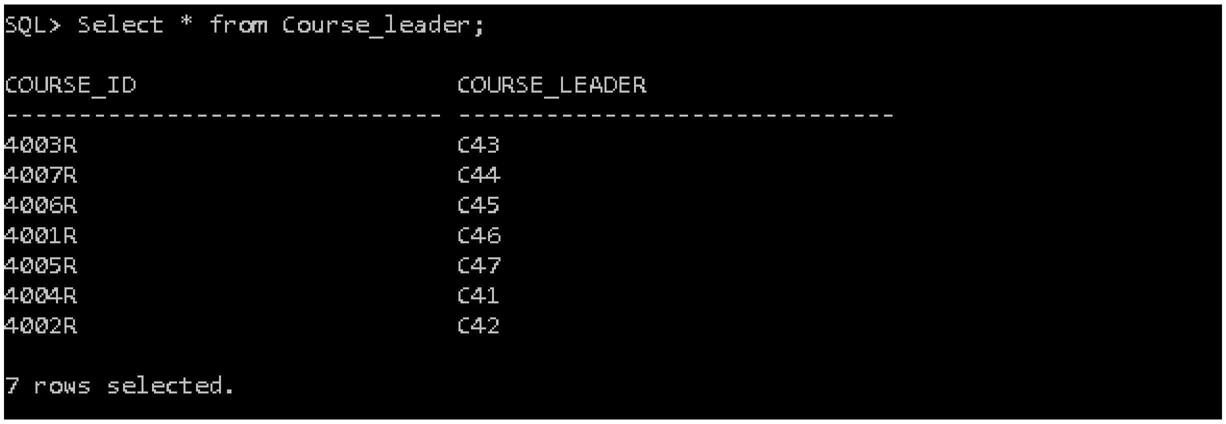
- Module_head table:
  SELECT * FROM Module_head;



```
SQL> Select * from Module_head;

MODULE_CODE                    HEAD
------------------------------ ------------------------------
CNN005                         C43
CNN012                         C44
CNN007                         C45
CNN004                         C46
CNN001                         C41
CNN008                         C42
CNN003                         C47

7 rows selected.
```

Figure 71 Module_head table

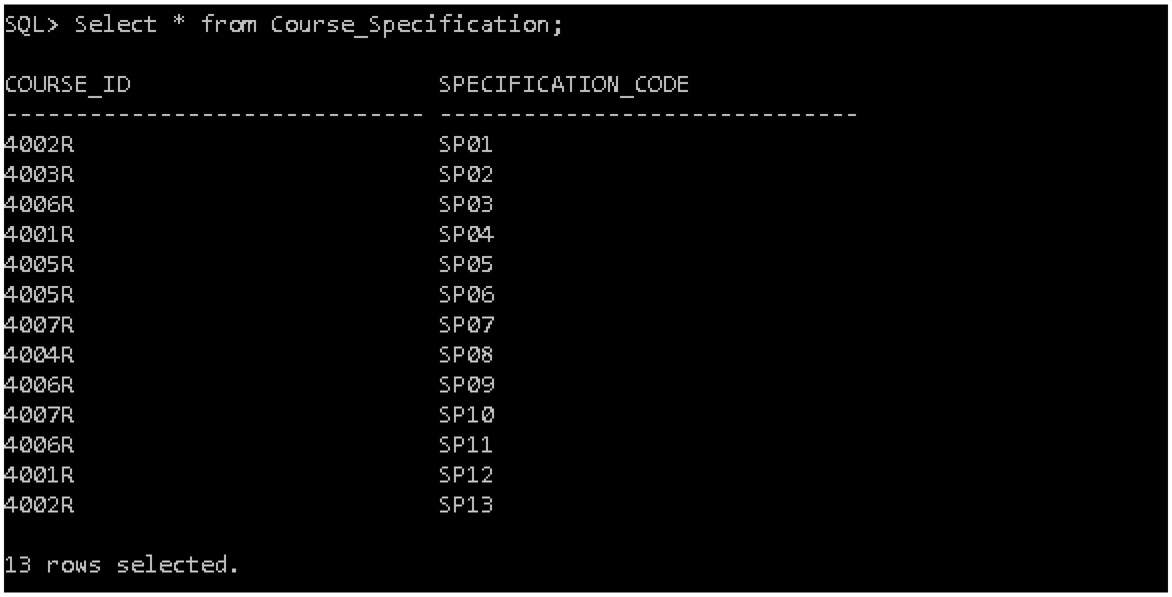
- Course_leader table:
  SELECT * FROM Course_leader;



```
SQL> Select * from Course_leader;

COURSE_ID                      COURSE_LEADER
------------------------------ ------------------------------
4003R                          C43
4007R                          C44
4006R                          C45
4001R                          C46
4005R                          C47
4004R                          C41
4002R                          C42

7 rows selected.
```

Figure 72 Course_leader table

- Course_specification table;
  SELECT * FROM Course_specification;

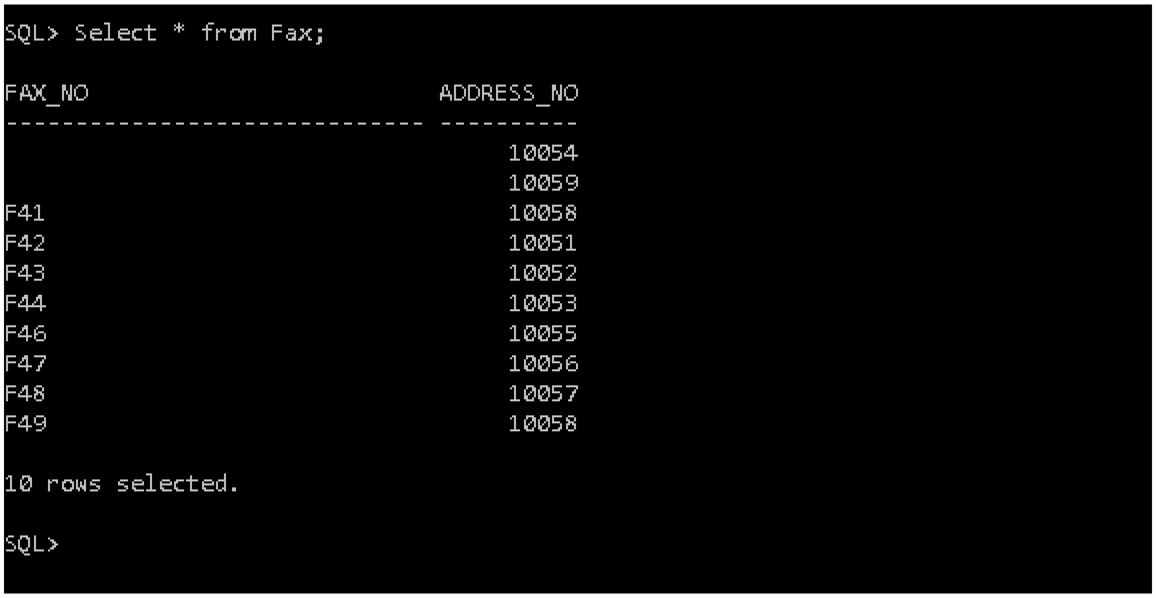


```
SQL> Select * from Course_Specification;

COURSE_ID                      SPECIFICATION_CODE
------------------------------ ------------------------------
4002R                          SP01
4003R                          SP02
4006R                          SP03
4001R                          SP04
4005R                          SP05
4005R                          SP06
4007R                          SP07
4004R                          SP08
4006R                          SP09
4007R                          SP10
4006R                          SP11
4001R                          SP12
4002R                          SP13

13 rows selected.
```

Figure 73 Course_specification table

- Fax table:

SELECT * FROM Fax;



```
SQL> Select * from Fax;

FAX_NO                         ADDRESS_NO
------------------------------ ----------
                                    10054
                                    10059
F41                                 10058
F42                                 10051
F43                                 10052
F44                                 10053
F46                                 10055
F47                                 10056
F48                                 10057
F49                                 10058

10 rows selected.

SQL>
```

Figure 74 Fax tab

5. Database Querying

5.1. Information Queries

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

SELECT Person.First_name, Person.Last_name, Person.Gender,

Address.Address_no, Address.House_no, Address.Country,Address.Province, Address.Street, Address.City,

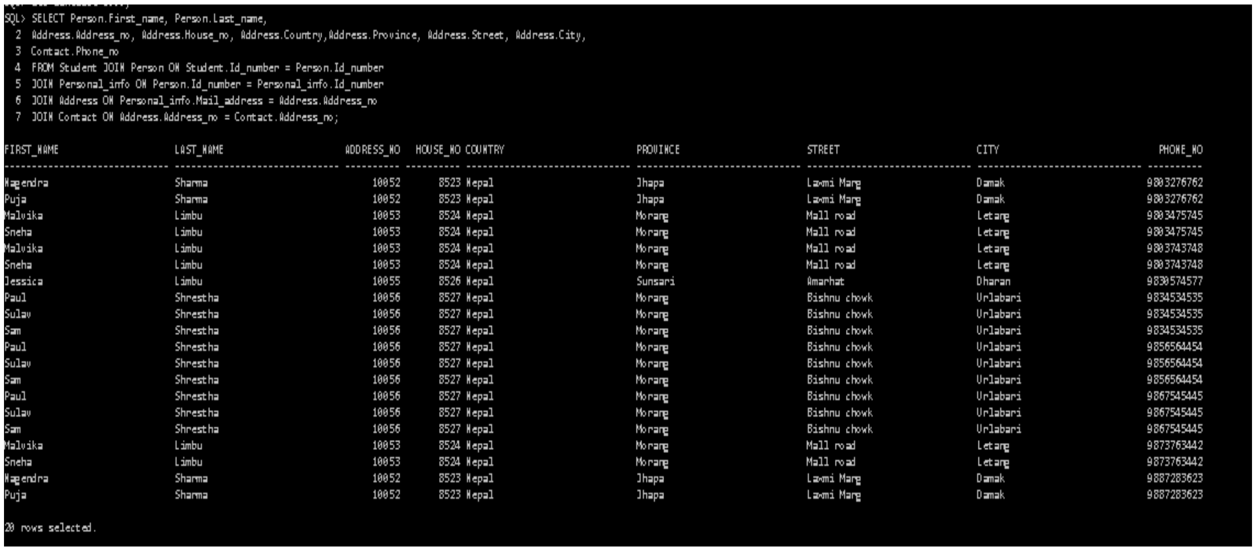
Contact.Phone_no

FROM Student JOIN Person ON Student.Id_number = Person.Id_number

JOIN Personal_info ON Person.Id_number = Personal_info.Id_number

JOIN Address ON Personal_info.Mail_address = Address.Address_no

JOIN Contact ON Address.Address_no = Contact.Address_no;



```
SQL> SELECT Person.First_name, Person.Last_name,
  2  Address.Address_no, Address.House_no, Address.Country,Address.Province, Address.Street, Address.City,
  3  Contact.Phone_no
  4  FROM Student JOIN Person ON Student.Id_number = Person.Id_number
  5  JOIN Personal_info ON Person.Id_number = Personal_info.Id_number
  6  JOIN Address ON Personal_info.Mail_address = Address.Address_no
  7  JOIN Contact ON Address.Address_no = Contact.Address_no;

FIRST_NAME                   LAST_NAME                    ADDRESS_NO   HOUSE_NO COUNTRY                      PROVINCE                     STREET                       CITY                           PHONE_NO
---------------------------- ---------------------------- ---------- ---------- ---------------------------- ---------------------------- ---------------------------- ---------------------------- ----------
Nagendra                     Sharma                            10052       8523 Nepal                        Jhapa                        Laxmi Marg                   Damak                        9803276762
Puja                         Sharma                            10052       8523 Nepal                        Jhapa                        Laxmi Marg                   Damak                        9803276762
Malvika                      Limbu                             10053       8524 Nepal                        Morang                       Mall road                    Letang                       9803475745
Sneha                        Limbu                             10053       8524 Nepal                        Morang                       Mall road                    Letang                       9803475745
Malvika                      Limbu                             10053       8524 Nepal                        Morang                       Mall road                    Letang                       9803743748
Sneha                        Limbu                             10053       8524 Nepal                        Morang                       Mall road                    Letang                       9803743748
Jessica                      Limbu                             10055       8526 Nepal                        Sunsari                      Amarhat                      Dharan                       9830574577
Paul                         Shrestha                          10056       8527 Nepal                        Morang                       Bishnu chowk                 Urlabari                     9834534535
Sulav                        Shrestha                          10056       8527 Nepal                        Morang                       Bishnu chowk                 Urlabari                     9834534535
Sam                          Shrestha                          10056       8527 Nepal                        Morang                       Bishnu chowk                 Urlabari                     9834534535
Paul                         Shrestha                          10056       8527 Nepal                        Morang                       Bishnu chowk                 Urlabari                     9856564454
Sulav                        Shrestha                          10056       8527 Nepal                        Morang                       Bishnu chowk                 Urlabari                     9856564454
Sam                          Shrestha                          10056       8527 Nepal                        Morang                       Bishnu chowk                 Urlabari                     9856564454
Paul                         Shrestha                          10056       8527 Nepal                        Morang                       Bishnu chowk                 Urlabari                     9867545445
Sulav                        Shrestha                          10056       8527 Nepal                        Morang                       Bishnu chowk                 Urlabari                     9867545445
Sam                          Shrestha                          10056       8527 Nepal                        Morang                       Bishnu chowk                 Urlabari                     9867545445
Malvika                      Limbu                             10053       8524 Nepal                        Morang                       Mall road                    Letang                       9873763442
Sneha                        Limbu                             10053       8524 Nepal                        Morang                       Mall road                    Letang                       9873763442
Nagendra                     Sharma                            10052       8523 Nepal                        Jhapa                        Laxmi Marg                   Damak                        9887283623
Puja                         Sharma                            10052       8523 Nepal                        Jhapa                        Laxmi Marg                   Damak                        9887283623

20 rows selected.
```

Figure 75 Information Query 1

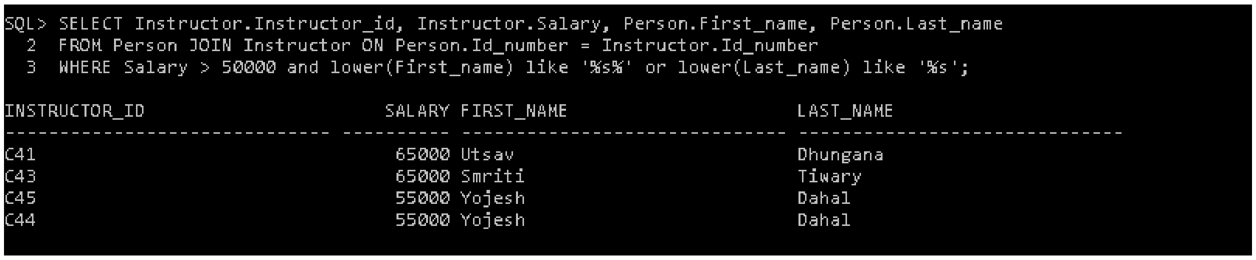
ii List all the modules which are taught by more than one instructor

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

SELECT Instructor.Instructor_id, Instructor.Salary, Person.First_name, Person.Last_name

FROM Person JOIN Instructor ON Person.Id_number = Instructor.Id_number

WHERE Salary > 50000 and lower(First_name) like '%s%' or lower(Last_name) like '%s';



```
SQL> SELECT Instructor.Instructor_id, Instructor.Salary, Person.First_name, Person.Last_name
  2  FROM Person JOIN Instructor ON Person.Id_number = Instructor.Id_number
  3  WHERE Salary > 50000 and lower(First_name) like '%s%' or lower(Last_name) like '%s';

INSTRUCTOR_ID                      SALARY FIRST_NAME                     LAST_NAME
------------------------------ ---------- ------------------------------ ------------------------------
C41                                 65000 Utsav                          Dhungana
C43                                 65000 Smriti                         Tiwary
C45                                 55000 Yojesh                         Dahal
C44                                 55000 Yojesh                         Dahal
```

Figure 76 Information Query 3

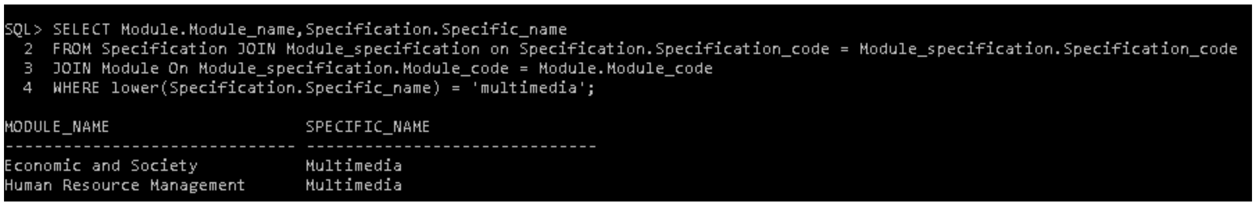
iv List the modules comes under the ‘Multimedia’ specification.

SELECT Module.Module_name,Specification.Specific_name

FROM Specification JOIN Module_specification on Specification.Specification_code = Module_specification.Specification_code

JOIN Module On Module_specification.Module_code = Module.Module_code

WHERE lower(Specification.Specific_name) = 'multimedia';



```
SQL> SELECT Module.Module_name,Specification.Specific_name
  2  FROM Specification JOIN Module_specification on Specification.Specification_code = Module_specification.Specification_code
  3  JOIN Module On Module_specification.Module_code = Module.Module_code
  4  WHERE lower(Specification.Specific_name) = 'multimedia';

MODULE_NAME                    SPECIFIC_NAME
------------------------------ ------------------------------
Economic and Society           Multimedia
Human Resource Management      Multimedia
```

Figure 77 Information Query 4

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

SELECT Person.First_name, Person.Last_name, Module.Module_name, Contact.Phone_no, Module_head.Head

FROM Contact JOIN Address ON Contact.Address_no = Address.Address_no

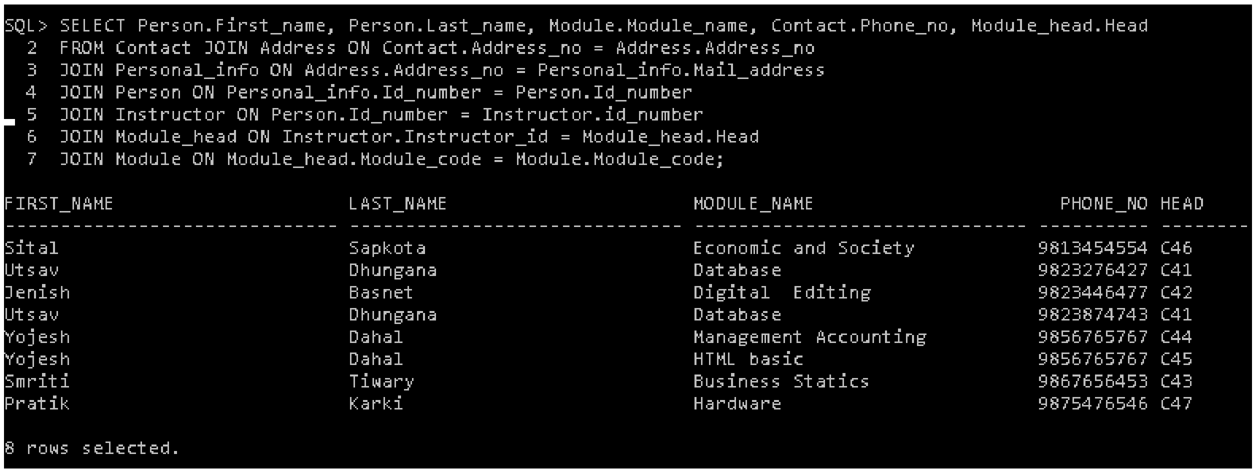
JOIN Personal_info ON Address.Address_no = Personal_info.Mail_address

JOIN Person ON Personal_info.Id_number = Person.Id_number

JOIN Instructor ON Person.Id_number = Instructor.id_number

JOIN Module_head ON Instructor.Instructor_id = Module_head.Head

JOIN Module ON Module_head.Module_code = Module.Module_code;



```
SQL> SELECT Person.First_name, Person.Last_name, Module.Module_name, Contact.Phone_no, Module_head.Head
  2  FROM Contact JOIN Address ON Contact.Address_no = Address.Address_no
  3  JOIN Personal_info ON Address.Address_no = Personal_info.Mail_address
  4  JOIN Person ON Personal_info.Id_number = Person.Id_number
  5  JOIN Instructor ON Person.Id_number = Instructor.id_number
  6  JOIN Module_head ON Instructor.Instructor_id = Module_head.Head
  7  JOIN Module ON Module_head.Module_code = Module.Module_code;

FIRST_NAME                     LAST_NAME                      MODULE_NAME                      PHONE_NO HEAD
------------------------------ ------------------------------ ------------------------------ ---------- --------
Sital                          Sapkota                        Economic and Society           9813454554 C46
Utsav                          Dhungana                       Database                       9823276427 C41
Jenish                         Basnet                         Digital  Editing               9823446477 C42
Utsav                          Dhungana                       Database                       9823874743 C41
Yojesh                         Dahal                          Management Accounting          9856765767 C44
Yojesh                         Dahal                          HTML basic                     9856765767 C45
Smriti                         Tiwary                         Business Statics               9867656453 C43
Pratik                         Karki                          Hardware                       9875476546 C47

8 rows selected.
```

Figure 78 Information Query 5

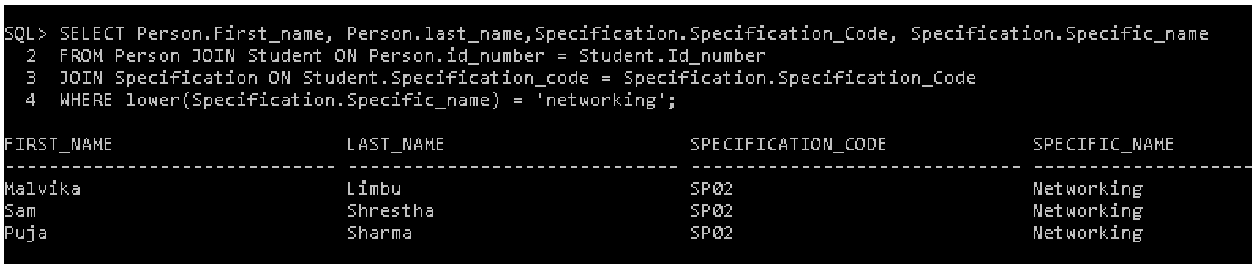
vi List all Students who have enrolled in 'networking' specifications.

SELECT Person.First_name, Person.last_name,Specification.Specification_Code, Specification.Specific_name

FROM Person JOIN Student ON Person.id_number = Student.Id_number

JOIN Specification ON Student.Specification_code = Specification.Specification_Code

WHERE lower(Specification.Specific_name) = 'networking';



```
SQL> SELECT Person.First_name, Person.last_name,Specification.Specification_Code, Specification.Specific_name
  2  FROM Person JOIN Student ON Person.id_number = Student.Id_number
  3  JOIN Specification ON Student.Specification_code = Specification.Specification_Code
  4  WHERE lower(Specification.Specific_name) = 'networking';

FIRST_NAME                     LAST_NAME                      SPECIFICATION_CODE             SPECIFIC_NAME
------------------------------ ------------------------------ ------------------------------ ------------------
Malvika                        Limbu                          SP02                           Networking
Sam                            Shrestha                       SP02                           Networking
Puja                           Sharma                         SP02                           Networking
```

Figure 79 Information Query 6

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

SELECT Fax.Fax_no, Person.First_name, Person.Last_name, Module.Module_name

FROM Module JOIN Module_head ON Module.Module_code = Module_head.Module_code

JOIN Instructor ON Module_head.Head= Instructor.Instructor_id

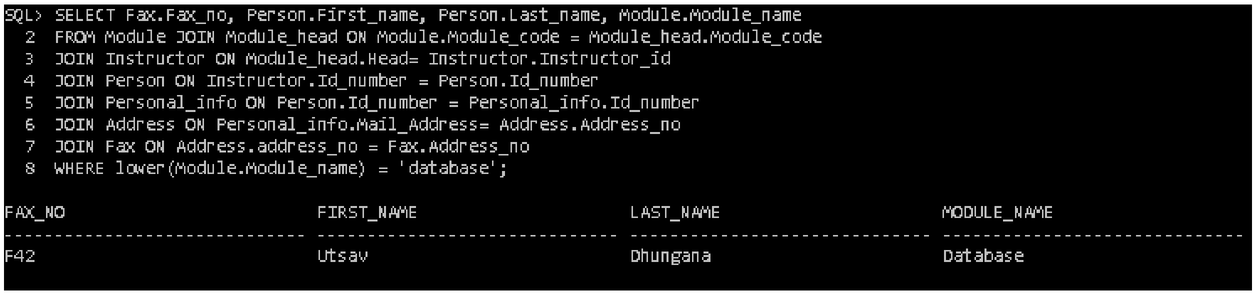
JOIN Person ON Instructor.Id_number = Person.Id_number

JOIN Personal_info ON Person.Id_number = Personal_info.Id_number

JOIN Address ON Personal_info.Mail_Address= Address.Address_no

JOIN Fax ON Address.address_no = Fax.Address_no

WHERE lower(Module.Module_name) = 'database';



```
SQL> SELECT Fax.Fax_no, Person.First_name, Person.Last_name, Module.Module_name
  2  FROM Module JOIN Module_head ON Module.Module_code = Module_head.Module_code
  3  JOIN Instructor ON Module_head.Head= Instructor.Instructor_id
  4  JOIN Person ON Instructor.Id_number = Person.Id_number
  5  JOIN Personal_info ON Person.Id_number = Personal_info.Id_number
  6  JOIN Address ON Personal_info.Mail_Address= Address.Address_no
  7  JOIN Fax ON Address.address_no = Fax.Address_no
  8  WHERE lower(Module.Module_name) = 'database';

FAX_NO                         FIRST_NAME                     LAST_NAME                      MODULE_NAME
------------------------------ ------------------------------ ------------------------------ ------------------------------
F42                            Utsav                          Dhungana                       Database
```

Figure 80 Information Query 7

viii List the specification falls under the BIT course.
ix List all the modules taught in any one particular class.
x List all the teachers with all their addresses who have ‘a’ at the end of their first names.

5.2 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 Person.First_name, Person.Last_name,Specification.Specific_name,Specification.Fee,

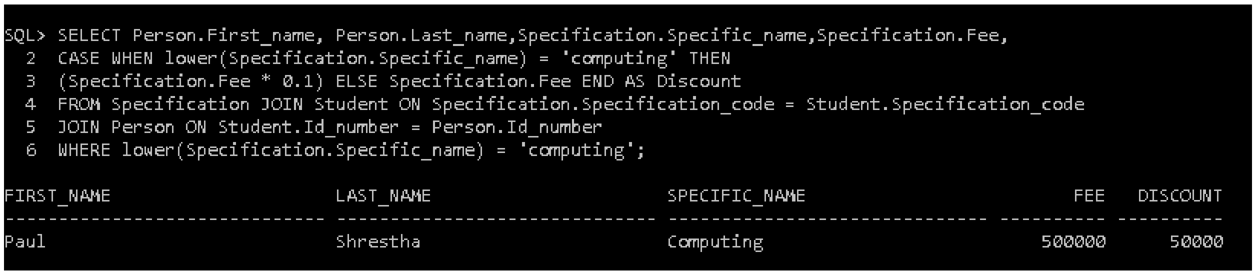
CASE WHEN lower(Specification.Specific_name) = 'computing' THEN

(Specification.Fee * 0.1) ELSE Specification.Fee END AS Discount

FROM Specification JOIN Student ON Specification.Specification_code = Student.Specification_code

JOIN Person ON Student.Id_number = Person.Id_number

WHERE lower(Specification.Specific_name) = 'computing';

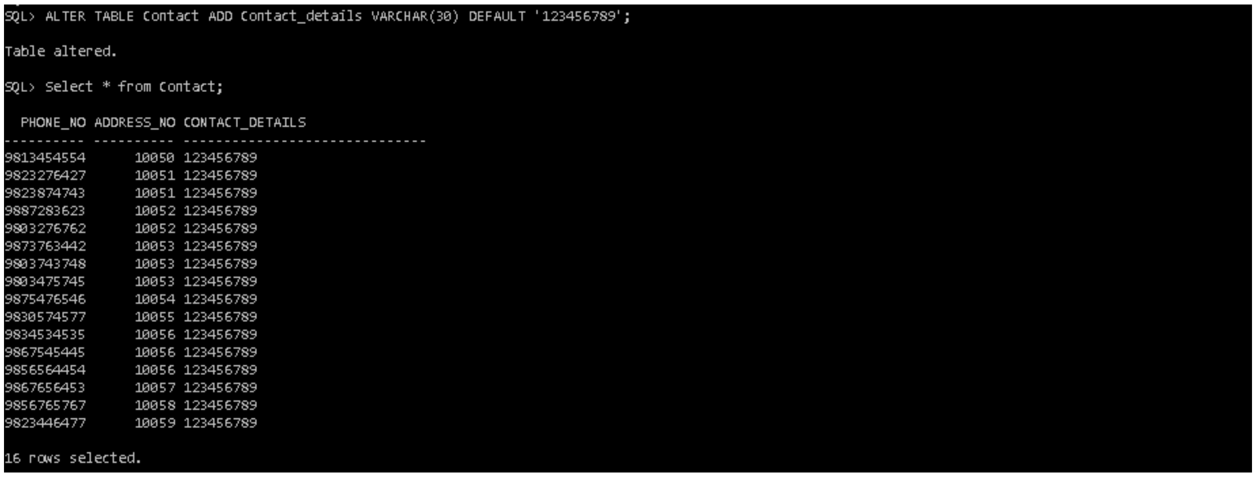


```
SQL> SELECT Person.First_name, Person.Last_name,Specification.Specific_name,Specification.Fee,
     CASE WHEN lower(Specification.Specific_name) = 'computing' THEN
     (Specification.Fee * 0.1) ELSE Specification.Fee END AS Discount
     FROM Specification JOIN Student ON Specification.Specification_code = Student.Specification_code
     JOIN Person ON Student.Id_number = Person.Id_number
     WHERE lower(Specification.Specific_name) = 'computing';

FIRST_NAME                     LAST_NAME                      SPECIFIC_NAME                         FEE   DISCOUNT
------------------------------ ------------------------------ ------------------------------ ---------- ----------
Paul                           Shrestha                       Computing                          500000      50000
```

Figure 81 Transaction Query 1

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 Contact ADD Contact_details VARCHAR(30) DEFAULT '123456789';



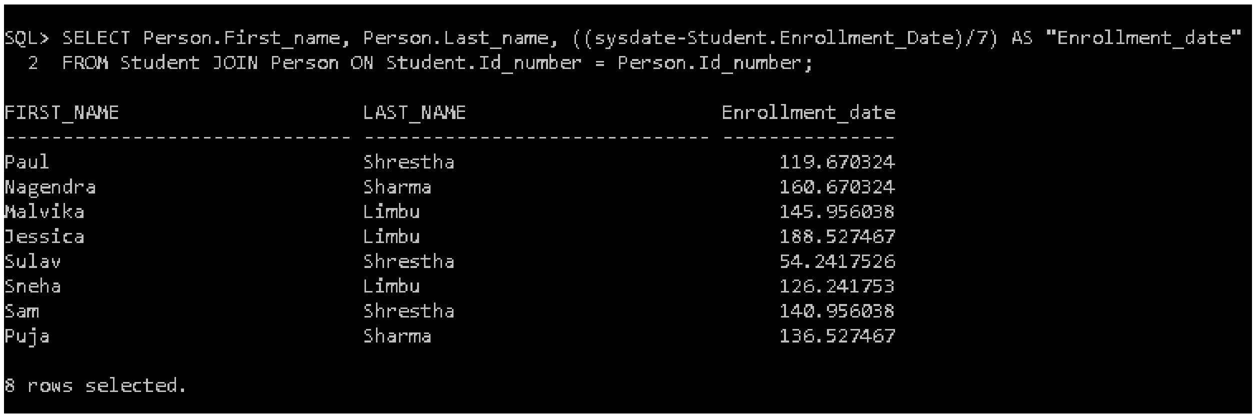
```
SQL> ALTER TABLE Contact ADD Contact_details VARCHAR(30) DEFAULT '123456789';

Table altered.

SQL> Select * from Contact;

  PHONE_NO ADDRESS_NO CONTACT_DETAILS
---------- ---------- ------------------------------
9813454554      10050 123456789
9823276427      10051 123456789
9823874743      10051 123456789
9887283623      10052 123456789
9803276762      10052 123456789
9873763442      10053 123456789
9803743748      10053 123456789
9803475745      10053 123456789
9875476546      10054 123456789
9830574577      10055 123456789
9834534535      10056 123456789
9867545445      10056 123456789
9856564454      10056 123456789
9867656453      10057 123456789
9856765767      10058 123456789
9823446477      10059 123456789

16 rows selected.
```

Figure 82 Transaction Query 2

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

SELECT Person.First_name, Person.Last_name, ((sysdate-Student.Enrollment_Date)/7) AS "Enrollment_date"

FROM Student JOIN Person ON Student.Id_number = Person.Id_number;



```
SQL> SELECT Person.First_name, Person.Last_name, ((sysdate-Student.Enrollment_Date)/7) AS "Enrollment_date"
  2  FROM Student JOIN Person ON Student.Id_number = Person.Id_number;

FIRST_NAME                     LAST_NAME                      Enrollment_date
------------------------------ ------------------------------ ---------------
Paul                           Shrestha                            119.670324
Nagendra                       Sharma                              160.670324
Malvika                        Limbu                               145.956038
Jessica                        Limbu                               188.527467
Sulav                          Shrestha                            54.2417526
Sneha                          Limbu                               126.241753
Sam                            Shrestha                            140.956038
Puja                           Sharma                              136.527467

8 rows selected.
```

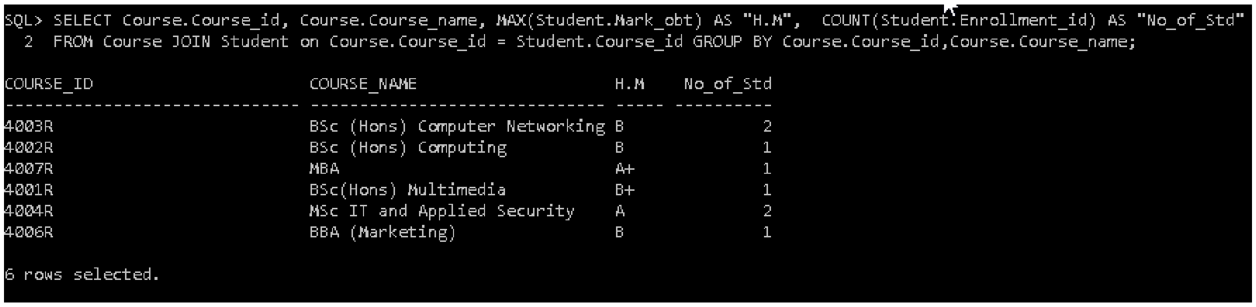
Figure 83 Transaction Query 3

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_id, Course.Course_name, MAX(Student.Mark_obt) AS "H.M", COUNT(Student.Enrollment_id) AS "No_of_Std"

FROM Course JOIN Student on Course.Course_id = Student.Course_id GROUP BY Course.Course_id,Course.Course_name;



```
SQL> SELECT Course.Course_id, Course.Course_name, MAX(Student.Mark_obt) AS "H.M",  COUNT(Student.Enrollment_id) AS "No_of_Std"
  2  FROM Course JOIN Student on Course.Course_id = Student.Course_id GROUP BY Course.Course_id,Course.Course_name;

COURSE_ID                      COURSE_NAME                    H.M    No_of_Std
------------------------------ ------------------------------ ----- ----------
4003R                          BSc (Hons) Computer Networking B              2
4002R                          BSc (Hons) Computing           B              1
4007R                          MBA                            A+             1
4001R                          BSc(Hons) Multimedia           B+             1
4004R                          MSc IT and Applied Security    A              2
4006R                          BBA (Marketing)                B              1

6 rows selected.
```

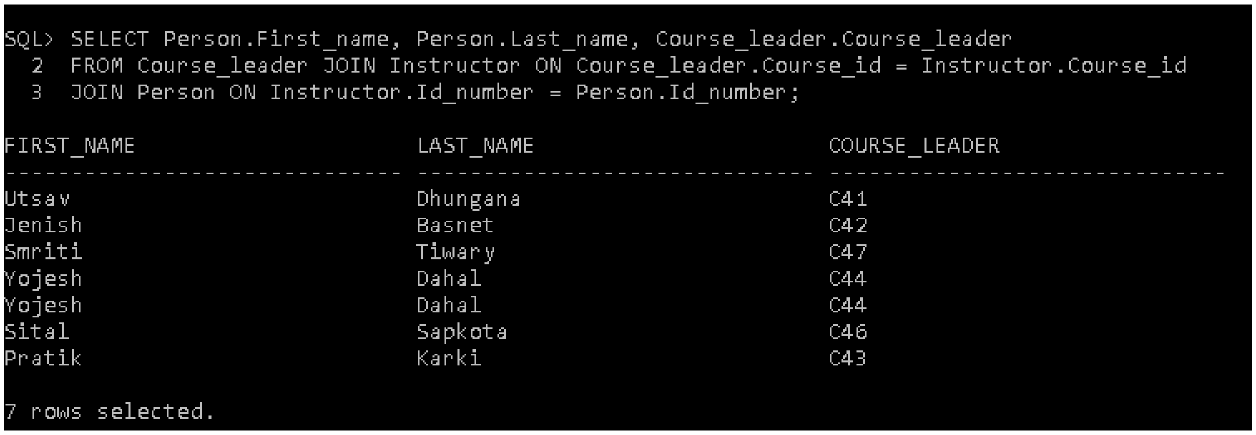
Figure 84 Transaction Query 6

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

SELECT Person.First_name, Person.Last_name, Course_leader.Course_leader

FROM Course_leader JOIN Instructor ON Course_leader.Course_id = Instructor.Course_id

JOIN Person ON Instructor.Id_number = Person.Id_number;

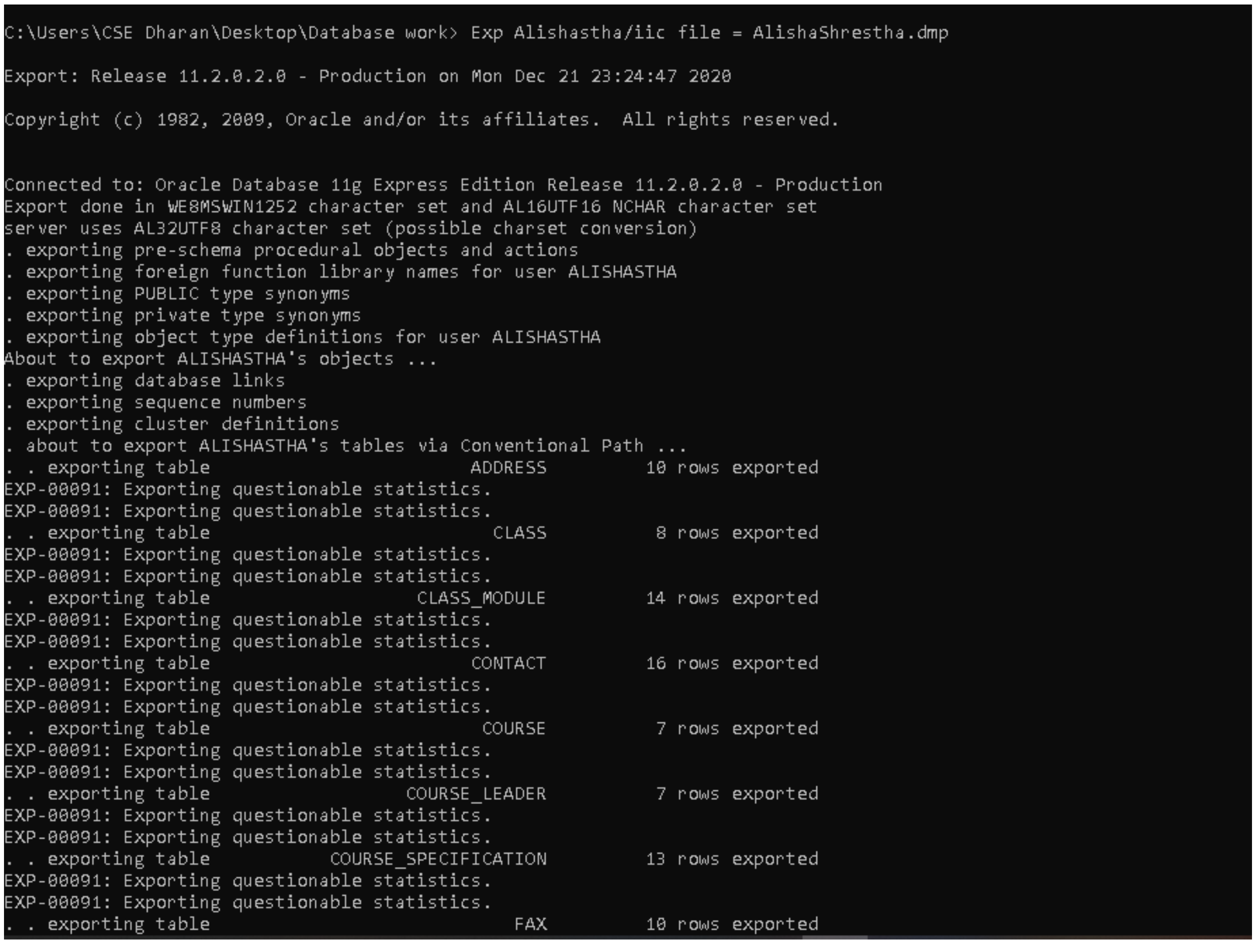


```
SQL> SELECT Person.First_name, Person.Last_name, Course_leader.Course_leader
  2  FROM Course_leader JOIN Instructor ON Course_leader.Course_id = Instructor.Course_id
  3  JOIN Person ON Instructor.Id_number = Person.Id_number;

FIRST_NAME                     LAST_NAME                      COURSE_LEADER
------------------------------ ------------------------------ ------------------------------
Utsav                          Dhungana                       C41
Jenish                         Basnet                         C42
Smriti                         Tiwary                         C47
Yojesh                         Dahal                          C44
Yojesh                         Dahal                          C44
Sital                          Sapkota                        C46
Pratik                         Karki                          C43

7 rows selected.
```

Figure 85 Transaction Query 6

5.3 Creation of Dump file

Exp Alishastha/iic file = AlishaShrestha.dmp



```
C:\Users\CSE Dharan\Desktop\Database work> Exp Alishastha/iic file = AlishaShrestha.dmp

Export: Release 11.2.0.2.0 - Production on Mon Dec 21 23:24:47 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 ALISHASTHA
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user ALISHASTHA
About to export ALISHASTHA's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export ALISHASTHA's tables via Conventional Path ...
. . exporting table                        ADDRESS         10 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                          CLASS          8 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                   CLASS_MODULE         14 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                        CONTACT         16 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           COURSE_SPECIFICATION         13 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                            FAX         10 rows exported
```

5.4 Drop tables

Syntax for Dropping table:

Drop table Address;

Drop table Person;

Drop table Contact;

Drop table Personal_infot;

Drop table Course;

Drop table Specification;

Drop table Module;

Drop table Instructor;

Drop table Student;

Drop table Instructor_detail;

Drop table Module_specification;

Drop table Class;

Drop table Class_module;

Drop table Module_head;

Drop table Course_leader;

Drop table Course_Specification;

Drop table Fax;

6. Conclusion

This coursework was entirely focused on the creation of database management system for the college. Database helps in storing data and information in a systematic manner which helps in easy access to the data and information with in no time. It is a very important part in a big organization like this college which holds numerous numbers of data related to its people. The entire task assigned in the coursework was finally completed with lots of efforts and mistakes as well. The assigned tasks in the coursework were not an easy task which required a lots labor and research. This coursework was not only about the completion of task in time but was also helpful in developing various skills and gaining experience in this topic. Getting chance to involve practically in this project has helped to gain sound knowledge of database and its mechanism for handling huge number of records.

Moreover, lots of study and research were carried out on the relevant topics such as normalization, the working of SQL, construction of ERD, etc. although this project was done properly with great labor which was very beneficial as this was not just a task but it had its practical impact in real life. The task was difficult and was done step by step with proper planning, research on the related topic and mainly about the normalization process. The most difficult part in this task was normalization which was also the most time consuming section compared to other part of work.

The project was carried out by preparing the initial ERD and to avoid data redundancy and anomalies normalizations were practiced. After the completion of the normalization, final ERD was constructed based on 3NF and data were populated for solving the given queries. The queries part was all solved along with the creation of dump file which was the end of the course work.

This project has taught a proper way of using a database management system in any organization which has huge number of people involved. Now talking about the most difficult part which was the process of normalization where numerous problems and errors were face. The process of normalization was not as easy as it seemed because it was very complicated part of the whole coursework and after various research and assistance from the module leader the work was finally completed. The main difficulty faced during this coursework was due to the virtual classes where the students lacked the physical interaction with their module leaders. The other difficulty faced during the coursework was the queries section, though it was the most interesting part of the task but still it came out to be more complicated.

The successful completion of the entire task assigned in the coursework was only possible with lots of hard work and mostly patience. For completing this coursework successfully lots of learning and study were done as this task was not very easy to be done only based on online classes. The project was very important to test the ability of the own skills regarding database which was very fruitful for the improvement of own knowledge. This task would not be possible without regular practice and proper guidance of the module leader. This project has helped in overall understanding of the importance, usages and merits of database in any field. Through this project the functions and features of software related to database were examined. This project made knowledgeable enough to be able to work with the management of data for small business organizations due to its effectiveness function.

7. Bibliography

Guru99. (n.d.). *Guru99.com*. Retrieved from Guru99: https://www.guru99.com/database-normalization.html

Silberschatz, A., Silberchatz, A., & Korth, H. f. (2005). Database System Concept. In A. SIlberschartz, *Database System Concepts 5th Edition* (p. 1024 pages). McGraw-Hill Publishing Co.