CC5051NI – DATABASES SYSTEMS

50% Individual Coursework

2020-21 Autumn

Student Name: Azan Ahmed Siddique

London Met ID: 19030779

College ID: NP01CP4A190120

Assignment Due Date: 20th December 2020

Assignment Submission Date: 20th December 2020

Table of Contents

TABLE OF FIGURES

1. Introduction

1.1. Introduction of College

Islington College is an educational institution based in Kathmandu, Nepal. Established in 1996, it was previously also known as Informatics College. The college underwent a name change in 2011 when it partnered with UK based London Metropolitan University to provide international degrees to students living in Nepal.

Considered to be one the finest IT & Business College in Nepal. Islington College focuses on providing excellent IT and Business education to its students. The college provides bachelor's degrees in Computing, Computer Networking & IT Security, Multimedia Technologies, and Business Administration and master’s degree in IT & Applied security, and business administration. The vision of Islington College is to be the best private college in Nepal by continuing to provide excellent education and further improving upon them.

Islington College aims to provide international degree programs to student living in Nepal and make them industry ready graduates by making sure that every student is technically proficient and possess the required skills to achieve success in their career.

1.2. Current Business Activities and Operations

Islington College uses the following business activities:

I. Keep track of all people, i.e. students and staff members.
II. All address, temporary and permanent, of students and staff will be recorded and mailing address will also be designated.
III. The address will consist of country, province, city, street, house number and a list of phone number to the address and a list of fax numbers to the location of the address.
IV. The college contains many courses, like BBA, IT, MBA etc.
V. Each course contains several specifications.
VI. Each specification contains several modules.
VII. Same modules can fall under different specification as well. For example, database module can fall under both computing and networking specification.
VIII. A course can have many instructors associated with it but an instructor can be associated in only one course.
IX. Each course will have a course leader and an instructor can be the leader of only one course.
X. Each instructor can teach one of many module at a time and a module can be taught by many instructors.
XI. A student can enroll for any one course and each course can have any number of students.

XII. Each module is taught in any given class but in each class a number of modules are taught.

1.3. Business Rules

There are various rules that a college must follow, some of them are:

- The college database should keep track of addresses and phone numbers of all the people associated with the college, like the students and instructors. Of the addresses provided, one needs to be a mailing address.
- The name, age, gender and date of birth of all the students and instructors should be stored.
- The phone number, email and fax numbers of the students and instructors should also be stored.
- The modules should have module id, as well as module name and class that it's taught in.
- All the modules under a specification (computing, networking, multimedia etc.) need to be stored.
- All the specification under a course (BIT, BBA, MBA, etc.) need to be stored as well.
- An instructor can teach multiple modules.
- A module will have multiple students.
- There can be multiple modules taught in one particular class.
- College can use the given information to inform students about routines, fee payment, etc.
- College should keep record of when the student has enrolled in a particular course.
- Students cannot enroll in multiple courses and specifications at once.
- There can be many modules under a specification.
- A course can have many different specifications. Each course will have a course leader.

1.4. Identification of Entities and Attributes

Entity is a single unique object in the real world that is being mastered (IBM, 2020). Examples of an entity are a single person, single product, or single organization (IBM, 2020).

Attribute is a characteristic or trait of an entity type that describes the entity, for example, the Person entity type has the Date of Birth attribute (IBM, 2020).

Entities	**Attributes**
Courses	Course_ID(PK), Course_Name, Course_Fees, Specification_Name
Modules	Module_ID(PK), Course_ID(FK), Module_Name, Class
Instructor	Instructor_ID(PK), Course_ID(FK), Ins_First_Name, Ins_Last_Name Ins_Age, Ins_Gender, Ins_Date_of_Birth, Joining_date, Ins_Type, Ins_Salary, Ins_E-mail
Student	Student_ID(PK), Course_ID(FK), Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_date, Std_Marks, Std_E-mail
Instructor_Address	InsAddress_ID(PK), Instructor_ID(FK), Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_House_no, Ins_Mailing_Address, Ins_Phone_no, Ins_Fax_no
Student_Address	StdAddress_ID(PK), Student_ID(FK), Std_Country, Std_Province, Std_City, Std_Street, Std_House_no, Std_Mailing_Address, Std_Phone_no, Std_Fax_no,

1.5. Initial ERD

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system (Lucid Chart, 2020). ER diagrams most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research (Lucid Chart, 2020). Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes (Lucid Chart, 2020). They mirror grammatical structure, with entities as nouns and relationships as verbs (Lucid Chart, 2020).

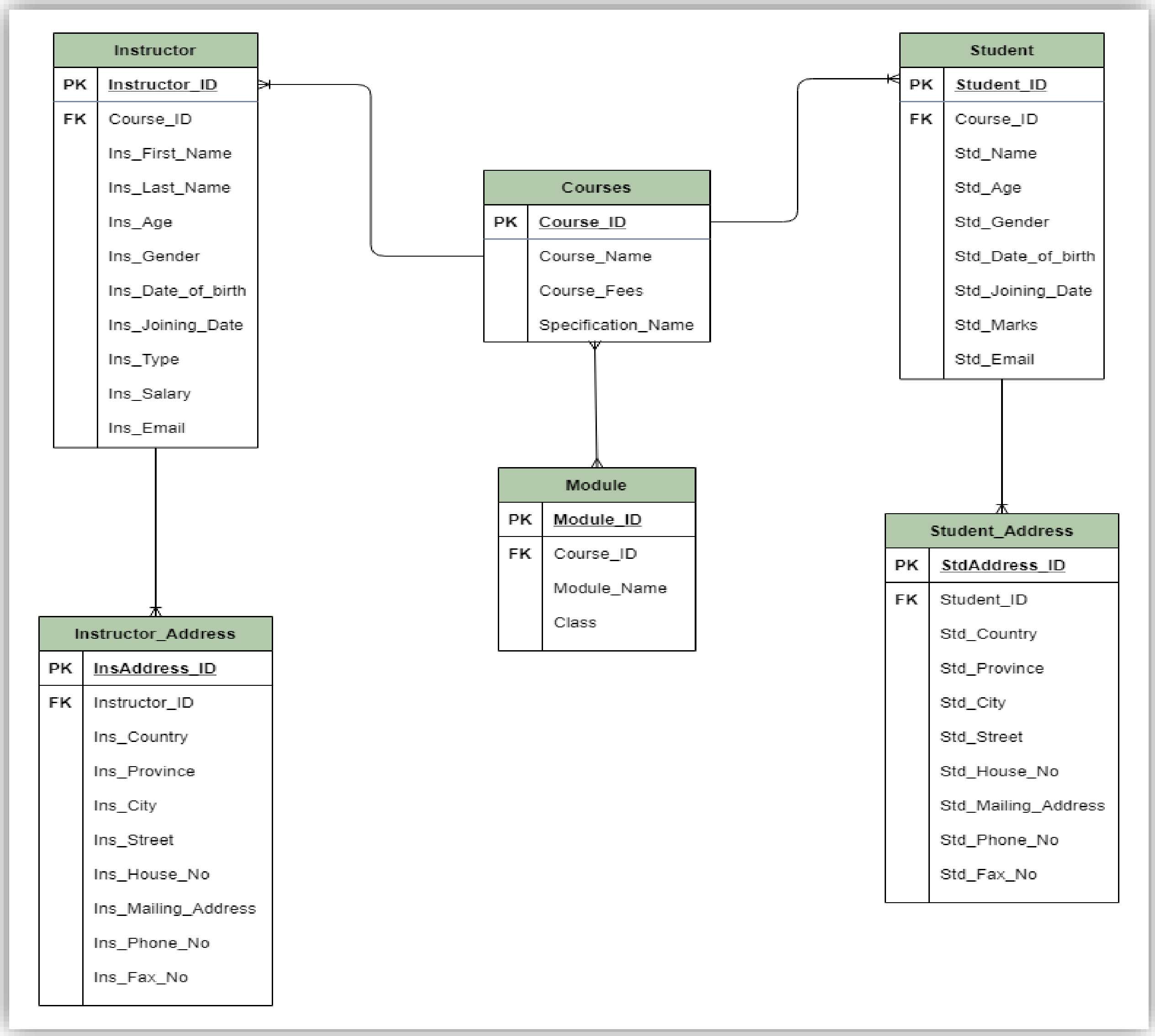


Figure 1: Initial ERD

There are several problems that may arise when designing initial ERD. The above ERD has many to many relationship between course and module. In relation database, it is impossible to create many to many relationship between two tables (File Maker, 2020). The ERD also contains data redundancy and anomalies. In order to fix these problems and make the database design efficient, the ER diagram needs to be normalized.

2. Normalization

2.1. Assumptions

- Each Instructor and Student will have a name, age, gender, joining date, date of birth, Address, house number, phone number, mobile number and fax number.
- Each course has its course leader.
- Every students and instructors will have their own student ID and instructor ID respectively.
- Every course will have a course ID, course name, course fees, and specification name.
- A course can have same name but different course_ID depending on the specification (Computing, Multimedia, Networking, etc.).
- Each class can have any number of module taught in it but a module can be taught in any one particular class.
- Each module has a Module_ID, Module_name and class.
- Every Instructors will also have an instructor ID, instructor type and salary.
- Every student will also have a student ID and marks.

2.2. Normalization

NORMALIZATION is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion (Guru99, 2020), Update and Deletion Anomalies (Guru99, 2020). (Guru99, 2020). Normalization rules divides larger tables into smaller tables and links them using relationships (Guru99, 2020). The purpose of Normalization in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically (Guru99, 2020).

2.2.1. UNF (Un-Normalized Form)

Scenario for UNF:

- Record of student and instructor is stored.

- The address (Country, Province, City, Street, House_no, Mailing_Address, phone number and fax number.) of students and instructors should be registered.
- Each student and instructor will need to provide their Name, Age, Gender, Date of Birth, Joining date, mobile number, email address.
- Student can enroll in only one course.
- Instructor can teach in only one course but a course can have multiple instructors.
- Each course (BBA, BIT, MBA, etc.) has one course leader and an instructor can be the leader of only one course.
- Each course offer any number of specifications (computing, multimedia, marketing, etc.).
- Student can choose only one specification.
- Each specification has different modules (Databases, Network and OS, Programming etc.) in it.
- Each module is taught in one particular class but in each class any number of modules can be taught. Each module will have a module head.
- An instructor can teach any one or many modules at a time and a module can be taught by many instructors.
- Salary of instructor will be determined by the type of instructor.
- Each address will have an Address ID.

Showing Repeating Groups

Course (Course_ID(PK), Course_Name, Course_Fees, Specification_Name, {Module_ID, Module_Name, Class}}, {Instructor_ID(PK), Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Joining_date, Ins_Type, Ins_Salary, Ins_Email, {InsAddress_ID, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_House_no, Ins_Mailing_Address, Ins_Phone_no, Ins_Fax_no }},{Student_ID(PK), Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_date, Std_Marks, Std_Email,{StdAddress_ID, Std_Country, Std_Province, Std_City, Std_Street, Std_House_no, Std_Mailing_Address, Std_Phone_no, Std_Fax_no}})

2.2.2. 1NF (First Normal Form)

First normal form (1NF) sets the fundamental rules for database normalization and relates to a single table within a relational database system (TechoPedia, 2020). Normalization follows three basic steps, each building on the last (TechoPedia, 2020). The first of these is the first normal form (TechoPedia, 2020).

The first normal form states that every column in the table must be unique, separate tables must be created for each set of related data, each table must be identified with a unique column or concatenated columns called the primary key, no rows may be duplicated, no columns may be duplicated, no row/column intersections contain a null value, no row/column intersections contain multivalued field (TechoPedia, 2020).

Scenario for 1NF:

In the UNF, the repeating groups have been identified and separated into entities after which composite primary keys have been formed. Primary key made up of two attributes is called a composite primary key.

Entities:

Course-1 (Course_ID(PK), Course_Name, Course_Fees, Specification_Name)

Module-1 (Module_ID(PK), Course_ID(FK), Module_Name, Class)

Instructor-1 (Instructor_ID(PK), Course_ID(FK), Ins_First_Name, Ins_Last_Name,Ins_Age, Ins_Gender, Ins_Date_of_Birth, Joining_date, Ins_Type, Ins_Salary, Ins_Email)

Student-1 (Student_ID(PK), Course_ID(FK), Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_date, Std_Marks, Std_Email)

Instructor_Address-1 (InsAddress_ID(PK), Instructor_ID(FK), Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_House_no, Ins_Mailing_Address, Ins_Phone_no, Ins_Fax_no)

Student_Address-1 (StdAddress_ID(PK), Student_ID(FK), Std_Country, Std_Province, Std_City, Std_Street, Std_House_no, Std_Mailing_Address, Std_Phone_no, Std_Fax_no)

2.2.3. 2NF (Second Normal Form)

Second Normal Form (2NF) is based on the concept of full functional dependency (GeeksforGeeks, 2020). Second Normal Form applies to relations with composite keys, that is, relations with a primary key composed of two or more attributes (GeeksforGeeks, 2020). A relation with a single-attribute primary key is automatically in at least 2NF (GeeksforGeeks, 2020). A relation that is not in 2NF may suffer from the update anomalies (GeeksforGeeks, 2020).

To be in second normal form, a relation must be in first normal form and relation must not contain any partial dependency (GeeksforGeeks, 2020). A relation is in 2NF if it has no Partial Dependency (attributes which are not part of any candidate key), is dependent on any proper subset of any candidate key of the table (GeeksforGeeks, 2020).

Scenario for 2NF:

The normalization of 1NF relations to 2NF involves the removal of partial dependencies (GeeksforGeeks, 2020). If a partial dependency exists, we remove the partially dependent attribute(s) from the relation by placing them in a new relation along with a copy of their determinant (GeeksforGeeks, 2020).

Showing Partial Dependency

For Module:

- Module_ID determines the module_name and class.
- Composite primary key Module_ID, Course_ID do not determine any attributes.

Module_ID -> Module_Name, Class

Module_ID, Course_ID ->

For Instructor:

- Instructor_ID determine the instructor name, age, gender, date of birth, joining date, instructor type, salary and E-mail.
- Composite primary key Instructor_ID, Course_ID do not determine any attributes.

Instructor_ID -> Ins_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Joining_date, Ins_Type, Ins_Salary, Ins_Email

Instructor_ID, Course_ID ->

For Student:

- Student_ID determine the instructor name, age, gender, date of birth, joining date, marks and E-mail.
- Composite primary key Student_ID, Course_ID do not determine any attributes.

Student_ID -> Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_date, Std_Marks, Std_Email

Student_ID, Course_ID ->

For Instructor_Address:

- InsAddress_ID determine the country, province city, street, house_no, mailing address, phone no and fax no of the instructor
- Composite primary key InsAddress_ID, Instructor_ID do not determine show attribute.

InsAddress_ID -> Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_House_no, Ins_Mailing_Address, Ins_Phone_no, Ins_Fax_no

InsAddress_ID, Instructor_ID ->

For Student_Address:

- StdAddress_ID determine the country, province city, street, house_no, mailing address, phone no and fax no of the instructor
- Composite primary key StdAddress_ID, Student_ID do not determine show attribute.

StdAddress_ID -> Std_Country, Std_Province, Std_City, Std_Street, Std_House_no, Std_Mailing_Address, Std_Phone_no, Std_Fax_no

StdAddress_ID, Student_ID ->

Entities:

Course-2 (Course_ID(PK), Course_Name, Course_Fees, Specification_Name)

Module-2 (Module_ID(PK), Module_Name, Class)

Module_Info-2 (Module_ID(FK), Course_ID(FK))

Instructor-2 (Instructor_ID(PK), Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Joining_date, Ins_Type, Ins_Salary, Ins_Email)

Instructor_Info-2 (Instructor_ID(FK), Course_ID(FK))

Student-2 (Student_ID(PK), Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_date, Std_Marks, Std_Email)

Student_Info-2 (Student_ID(FK), Course_ID(FK))

Instructor_Address-2 (InsAddress_ID(PK), Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_House_no, Ins_Mailing_Address, Ins_Phone_no, Ins_Fax_no)

Instructor_Address_Info-2 (InsAddress_ID(FK), Instructor_ID(FK))

Student_Address-2 (StdAddress_ID(PK), Std_Country, Std_Province, Std_City, Std_Street, Std_House_no, Std_Mailing_Address, Std_Phone_no, Std_Fax_no)

Student_Address_Info-2 (StdAddress_ID(FK), Student_ID(FK))

2.2.4. 3NF (Third Normal Form)

If a relation is in 2NF and no non key attribute is transitively dependent on the primary key then it is in 3NF (Third Normal Form). A transitive dependency is an indirect relationship between data elements in a database (Xspdf, 2020). A relation is in 3NF if at least one of the following condition holds in every non-trivial function dependency X –> Y (GeeksforGeeks, 2020):

1. X is a super key (GeeksforGeeks, 2020).
2. Y is a prime attribute (each element of Y is part of some candidate key) (GeeksforGeeks, 2020).

Scenario for 3NF:

The normalization of 2NF relations to 3NF involves the removal of transitive dependencies (GeeksforGeeks, 2020). To remove a transitive dependency (if it exists), the attribute that are transitively dependent need to be place into a new relation along with the copy of the determinant.

<u>Showing transitive dependency</u>

For Instructor_Address:

- Instructor Address ID determines the country, province city, street, house_no, mailing address and house_no determines the phone_no and fax_no of the instructor.

InsAddressID -> Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_House_no, Ins_Mailing_Address,

Ins_House_no -> Phone_No, Fax_No

For Student_Address:

- Student Address ID determines the country, province city, street, house_no, mailing address and house_no determines the phone_no and fax_no of the student.

StdAddressID -> Std_Country, Std_Province, Std_City, Std_Street, Std_House_no, Std_Mailing_Address, Std_House_no -> Std_Phone_No, Std_Fax_No

Entities:

Course-3 (Course_ID(PK), Course_Name, Course_Fees, Specification_Name)

Module-3 (Module_ID(PK), Module_Name, Class)

Module_Info-3 (Module_ID(FK), Module_ID(FK))

Instructor-3 (Instructor_ID(PK), Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Joining_date, Ins_Type, Ins_Salary, Ins_Email)

Instructor_Info-3 (Instructor_ID(FK), Course_ID(FK))

Student-3 (Student_ID(PK), Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_date, Std_Marks, Std_Email)

Student_Info-3 (Student_ID(FK), Course_ID(FK))

Instructor_Address-3 (InsAddress_ID(PK), Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_House_no(FK), Ins_Mailing_Address)

Instructor_Residency-3 (Ins_House_no(PK), Ins_Phone_no, Ins_Fax_no)

Instructor_Address_Info-3 (InsAddress_ID(FK), Instructor_ID(FK))

Student_Address-3 (StdAddress_ID(PK), Std_Country, Std_Province, Std_City, Std_Street, Std_House_no(FK), Std_Mailing_Address)

Student_Residency-3 (Std_House_no(PK), Std_Phone_no, Std_Fax_no)

Student_Address_Info-3 (StdAddress_ID(FK), Student_ID(FK))

2.3. ER diagram after carrying out normalization

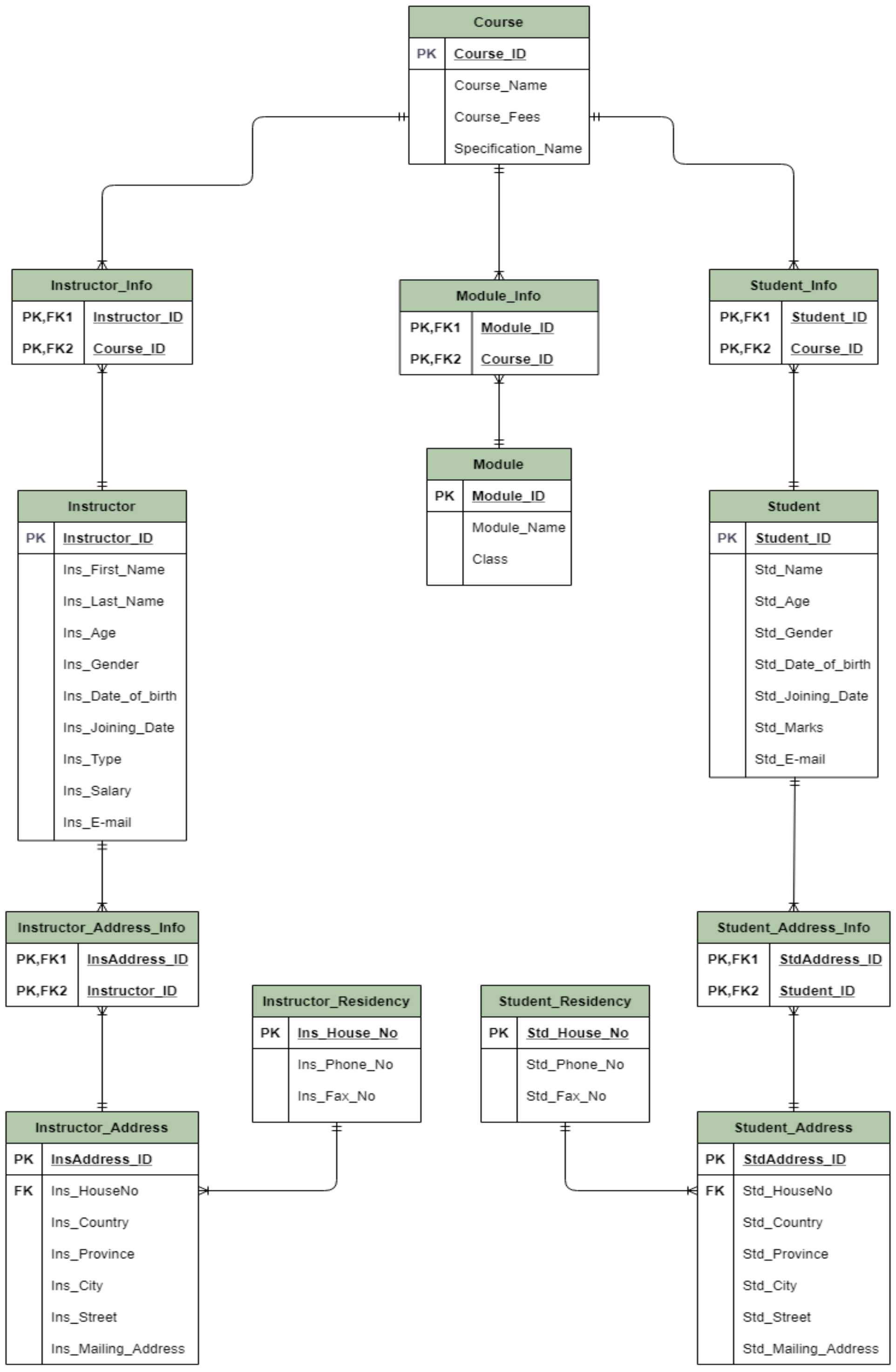


Figure 2: Final ERD

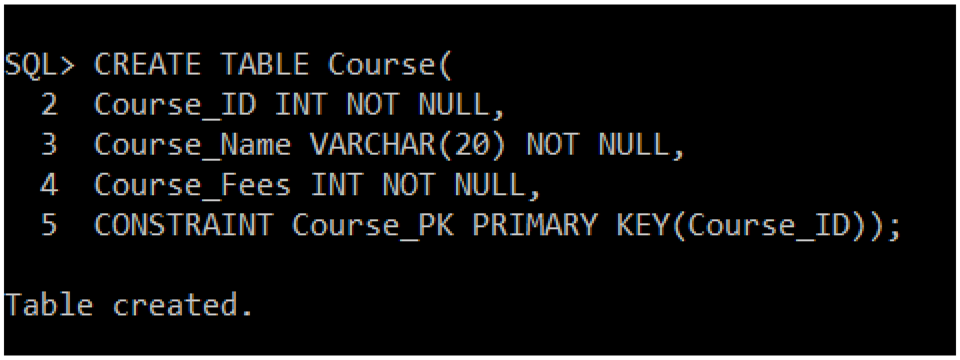
3. Database Implementation

3.1. Table Generation

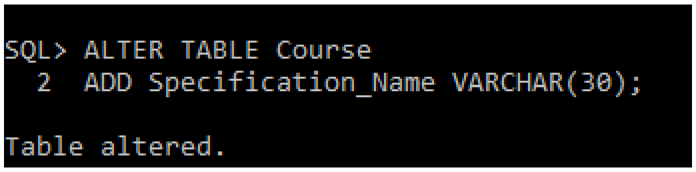
To create a new table in Oracle Database, you use the CREATE TABLE statement (Oracle Tutorial, 2020). Tables are uniquely named within a database and schema (SQL Server, 2020). Each table contains one or more columns (SQL Server, 2020). And each column has an associated data type that defines the kind of data (numbers, strings, or temporal data) it can store (SQL Server, 2020). The Oracle ALTER TABLE statement is used to add, modify, or drop/delete columns in a table (Tech on the net, 2020). The Oracle ALTER TABLE statement is also used to rename a table (Tech on the net, 2020). The PRIMARY KEY constraint uniquely identifies each record in a table (w3school, 2020). Primary keys must contain UNIQUE values, and cannot contain NULL values (w3school, 2020). A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields) (w3school, 2020).

Creating table for Course

```
CREATE TABLE Course(

Course_ID INT NOT NULL,

Course_Name VARCHAR(20) NOT NULL,

Course_Fees INT NOT NULL,

CONSTRAINT Course_PK PRIMARY KEY(Course_ID));
```



```
SQL> CREATE TABLE Course(
  2  Course_ID INT NOT NULL,
  3  Course_Name VARCHAR(20) NOT NULL,
  4  Course_Fees INT NOT NULL,
  5  CONSTRAINT Course_PK PRIMARY KEY(Course_ID));

Table created.
```



```
SQL> ALTER TABLE Course
  2  ADD Specification_Name VARCHAR(30);

Table altered.
```

Figure 3: Creating Course Table

Creating table for Module_Info

CREATE TABLE Module_Info(

Module_ID INT NOT NULL,

Course_ID INT NOT NULL,

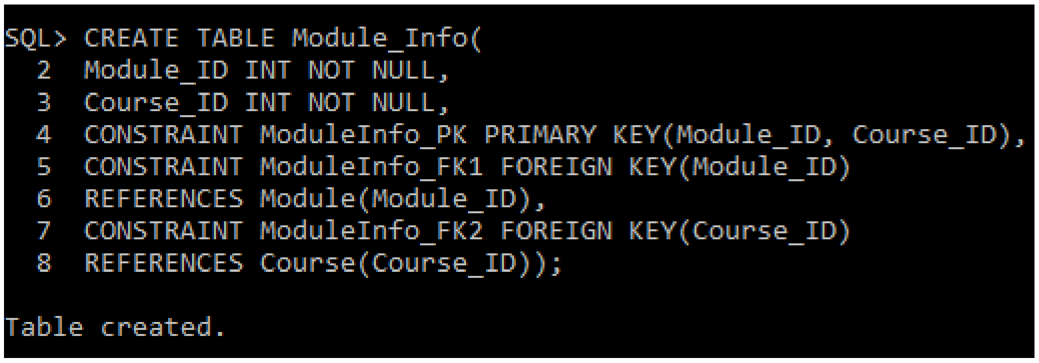
CONSTRAINT ModuleInfo_PK PRIMARY KEY(Module_ID, Course_ID),

CONSTRAINT ModuleInfo_FK1 FOREIGN KEY(Module_ID)

REFERENCES Module(Module_ID),

CONSTRAINT ModuleInfo_FK2 FOREIGN KEY(Course_ID)

REFERENCES Course(Course_ID));



```
SQL> CREATE TABLE Module_Info(
  2  Module_ID INT NOT NULL,
  3  Course_ID INT NOT NULL,
  4  CONSTRAINT ModuleInfo_PK PRIMARY KEY(Module_ID, Course_ID),
  5  CONSTRAINT ModuleInfo_FK1 FOREIGN KEY(Module_ID)
  6  REFERENCES Module(Module_ID),
  7  CONSTRAINT ModuleInfo_FK2 FOREIGN KEY(Course_ID)
  8  REFERENCES Course(Course_ID));

Table created.
```

Figure 4: Creating Module_info table

Creating table for Module

CREATE TABLE Module(

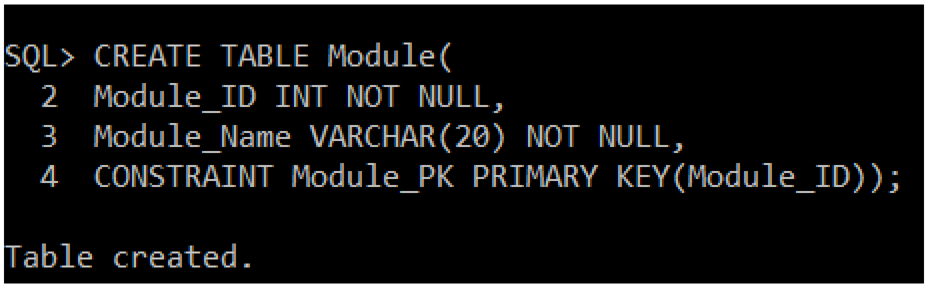
Module_ID INT NOT NULL,

Module_Name VARCHAR(20) NOT NULL,

CONSTRAINT Module_PK PRIMARY KEY(Module_ID));

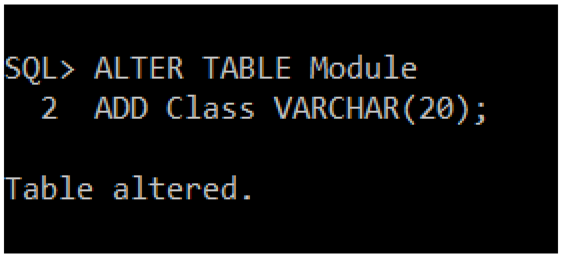
ALTER TABLE Module

ADD Class VARCHAR(20);



```
SQL> CREATE TABLE Module(
  2  Module_ID INT NOT NULL,
  3  Module_Name VARCHAR(20) NOT NULL,
  4  CONSTRAINT Module_PK PRIMARY KEY(Module_ID));

Table created.
```

Figure 5: Create Module table



```
SQL> ALTER TABLE Module
  2  ADD Class VARCHAR(20);

Table altered.
```

Figure 6: Alter table Module

Creating table for Instructor_Info

CREATE TABLE Instructor_Info(

Instructor_ID INT NOT NULL,

Course_ID INT NOT NULL,

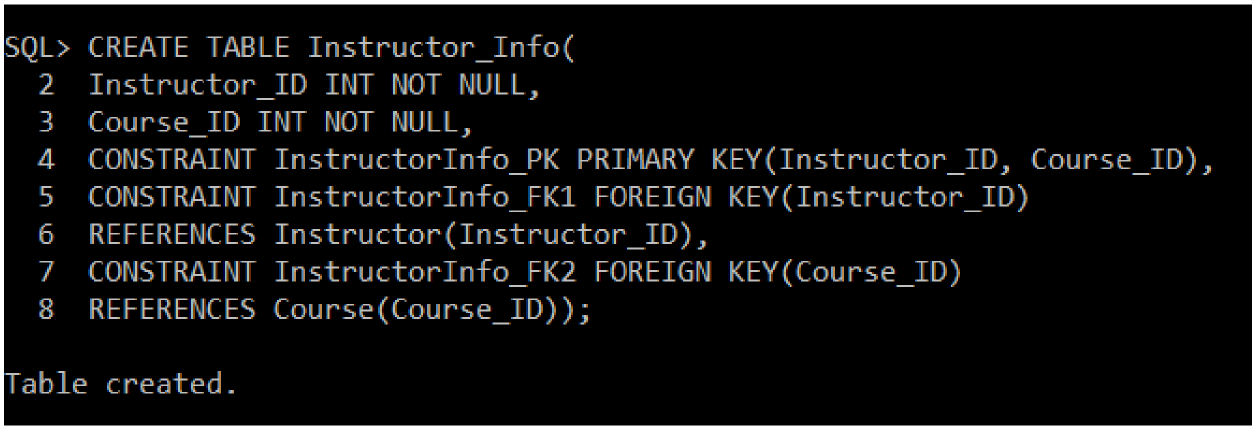
CONSTRAINT InstructorInfo_PK PRIMARY KEY(Instructor_ID, Course_ID),

CONSTRAINT InstructorInfo_FK1 FOREIGN KEY(Instructor_ID)

REFERENCES Instructor(Instructor_ID),

CONSTRAINT InstructorInfo_FK2 FOREIGN KEY(Course_ID)

REFERENCES Course(Course_ID));

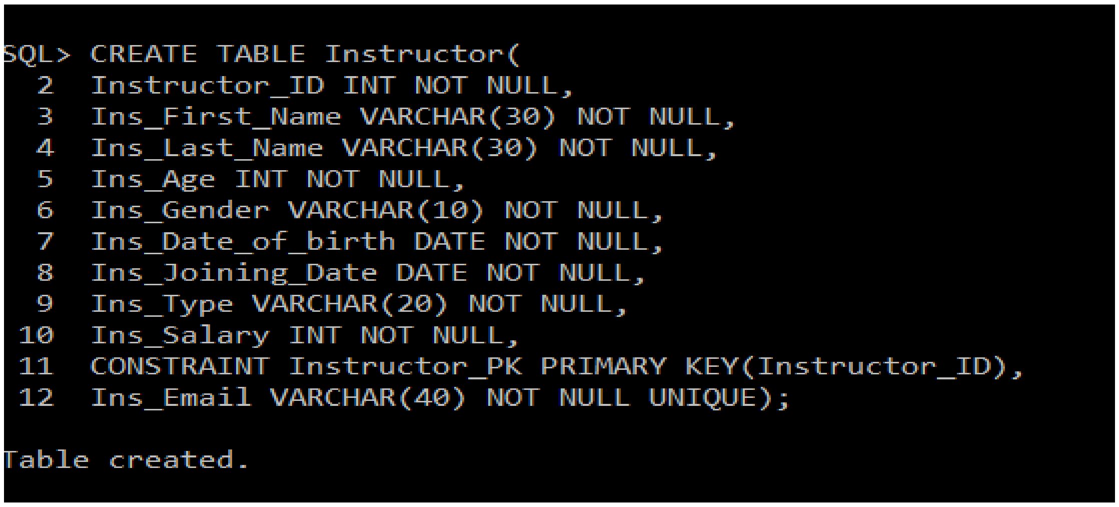


```
SQL> CREATE TABLE Instructor_Info(
  2  Instructor_ID INT NOT NULL,
  3  Course_ID INT NOT NULL,
  4  CONSTRAINT InstructorInfo_PK PRIMARY KEY(Instructor_ID, Course_ID),
  5  CONSTRAINT InstructorInfo_FK1 FOREIGN KEY(Instructor_ID)
  6  REFERENCES Instructor(Instructor_ID),
  7  CONSTRAINT InstructorInfo_FK2 FOREIGN KEY(Course_ID)
  8  REFERENCES Course(Course_ID));

Table created.
```

Figure 7: Creating Instructor_Info table

Creating table for Instructor

```
CREATE TABLE Instructor(
Instructor_ID INT NOT NULL,
Ins_First_Name VARCHAR(30) NOT NULL,
Ins_Last_Name VARCHAR(30) NOT NULL,
Ins_Age INT NOT NULL,
Ins_Gender VARCHAR(10) NOT NULL,
Ins_Date_of_birth DATE NOT NULL,
Ins_Joining_Date DATE NOT NULL,
Ins_Type VARCHAR(20) NOT NULL,
Ins_Salary INT NOT NULL,
CONSTRAINT Instructor_PK PRIMARY KEY(Instructor_ID),
Ins_Email VARCHAR(40) NOT NULL UNIQUE);
```



```
SQL> CREATE TABLE Instructor(
  2  Instructor_ID INT NOT NULL,
  3  Ins_First_Name VARCHAR(30) NOT NULL,
  4  Ins_Last_Name VARCHAR(30) NOT NULL,
  5  Ins_Age INT NOT NULL,
  6  Ins_Gender VARCHAR(10) NOT NULL,
  7  Ins_Date_of_birth DATE NOT NULL,
  8  Ins_Joining_Date DATE NOT NULL,
  9  Ins_Type VARCHAR(20) NOT NULL,
 10  Ins_Salary INT NOT NULL,
 11  CONSTRAINT Instructor_PK PRIMARY KEY(Instructor_ID),
 12  Ins_Email VARCHAR(40) NOT NULL UNIQUE);

Table created.
```

Figure 8: Creating Instructor table

Creating table for Instructor_Address_Info

CREATE TABLE Instructor_Address_Info(

InsAddress_ID INT NOT NULL,

Instructor_ID INT NOT NULL,

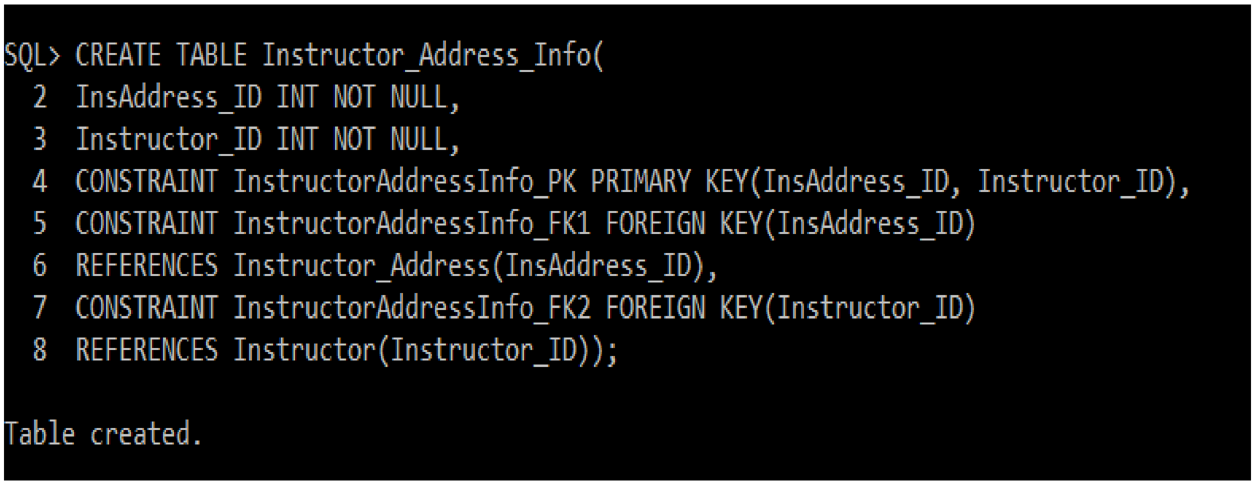
CONSTRAINT InstructorAddressInfo_PK PRIMARY KEY(InsAddress_ID, Instructor_ID),

CONSTRAINT InstructorAddressInfo_FK1 FOREIGN KEY(InsAddress_ID)

REFERENCES Instructor_Address(InsAddress_ID),

CONSTRAINT InstructorAddressInfo_FK2 FOREIGN KEY(Instructor_ID)

REFERENCES Instructor(Instructor_ID));



```
SQL> CREATE TABLE Instructor_Address_Info(
  2  InsAddress_ID INT NOT NULL,
  3  Instructor_ID INT NOT NULL,
  4  CONSTRAINT InstructorAddressInfo_PK PRIMARY KEY(InsAddress_ID, Instructor_ID),
  5  CONSTRAINT InstructorAddressInfo_FK1 FOREIGN KEY(InsAddress_ID)
  6  REFERENCES Instructor_Address(InsAddress_ID),
  7  CONSTRAINT InstructorAddressInfo_FK2 FOREIGN KEY(Instructor_ID)
  8  REFERENCES Instructor(Instructor_ID));

Table created.
```

Figure 9: Creating Instructor_Address_Info table

Creating table for Instructor_Address

CREATE TABLE Instructor_Address(

InsAddress_ID INT NOT NULL,

Ins_HouseNo INT NOT NULL,

Ins_Country VARCHAR(30) NOT NULL,

Ins_Province VARCHAR(30) NOT NULL,

Ins_City VARCHAR(30) NOT NULL,

Ins_Street VARCHAR(30) NOT NULL,

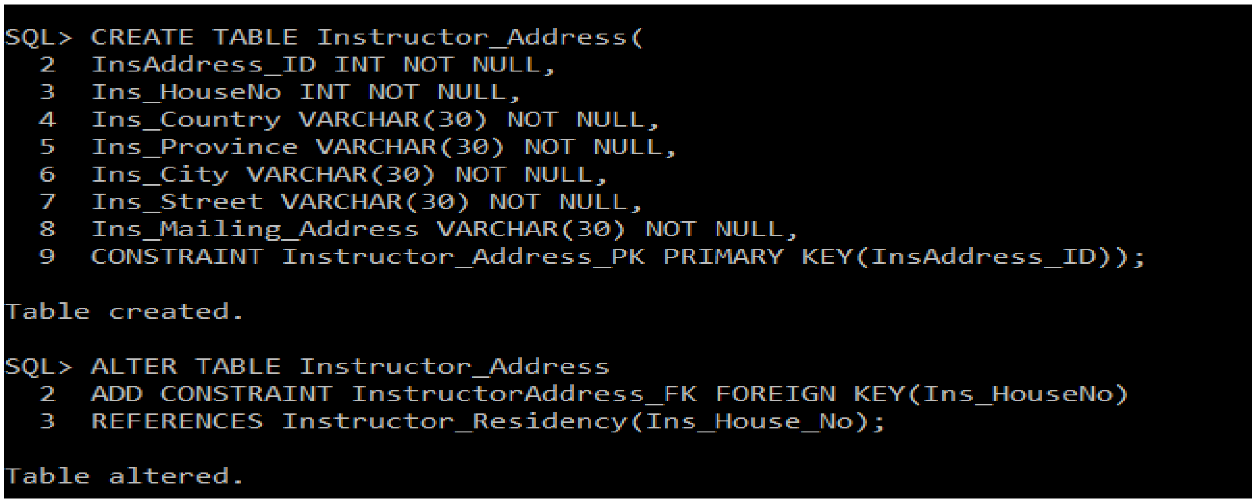
Ins_Mailing_Address VARCHAR(30) NOT NULL,

CONSTRAINT Instructor_Address_PK PRIMARY KEY(InsAddress_ID));

ALTER TABLE Instructor_Address

ADD CONSTRAINT InstructorAddress_FK FOREIGN KEY(Ins_HouseNo)

REFERENCES Instructor_Residency(Ins_House_No);



```
SQL> CREATE TABLE Instructor_Address(
  2  InsAddress_ID INT NOT NULL,
  3  Ins_HouseNo INT NOT NULL,
  4  Ins_Country VARCHAR(30) NOT NULL,
  5  Ins_Province VARCHAR(30) NOT NULL,
  6  Ins_City VARCHAR(30) NOT NULL,
  7  Ins_Street VARCHAR(30) NOT NULL,
  8  Ins_Mailing_Address VARCHAR(30) NOT NULL,
  9  CONSTRAINT Instructor_Address_PK PRIMARY KEY(InsAddress_ID));

Table created.

SQL> ALTER TABLE Instructor_Address
  2  ADD CONSTRAINT InstructorAddress_FK FOREIGN KEY(Ins_HouseNo)
  3  REFERENCES Instructor_Residency(Ins_House_No);

Table altered.
```

Figure 10: Creating Instructor_Address table

Creating table for Instructor_Residency

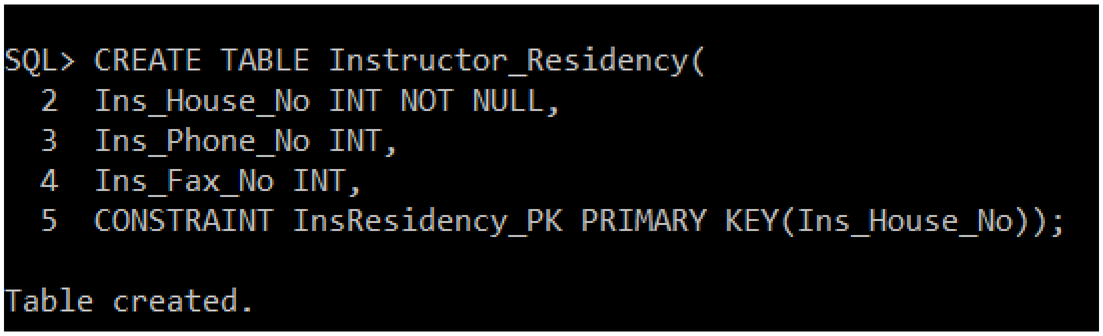
CREATE TABLE Instructor_Residency(

Ins_House_No INT NOT NULL,

Ins_Phone_No INT,

Ins_Fax_No INT,

CONSTRAINT InsResidency_PK PRIMARY KEY(Ins_House_No));

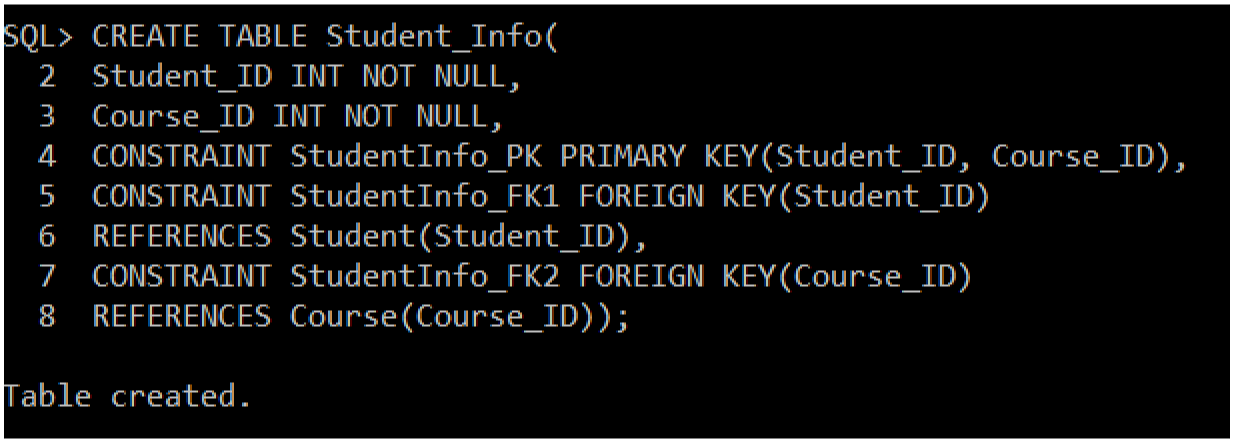


```
SQL> CREATE TABLE Instructor_Residency(
  2  Ins_House_No INT NOT NULL,
  3  Ins_Phone_No INT,
  4  Ins_Fax_No INT,
  5  CONSTRAINT InsResidency_PK PRIMARY KEY(Ins_House_No));

Table created.
```

Figure 11: Creating Instructor_Residency table

Creating table for Student_Info

```
CREATE TABLE Student_Info(

Student_ID INT NOT NULL,

Course_ID INT NOT NULL,

CONSTRAINT StudentInfo_PK PRIMARY KEY(Student_ID, Course_ID),

CONSTRAINT StudentInfo_FK1 FOREIGN KEY(Student_ID)

REFERENCES Student(Student_ID),

CONSTRAINT StudentInfo_FK2 FOREIGN KEY(Course_ID)

REFERENCES Course(Course_ID));
```



```
SQL> CREATE TABLE Student_Info(
  2  Student_ID INT NOT NULL,
  3  Course_ID INT NOT NULL,
  4  CONSTRAINT StudentInfo_PK PRIMARY KEY(Student_ID, Course_ID),
  5  CONSTRAINT StudentInfo_FK1 FOREIGN KEY(Student_ID)
  6  REFERENCES Student(Student_ID),
  7  CONSTRAINT StudentInfo_FK2 FOREIGN KEY(Course_ID)
  8  REFERENCES Course(Course_ID));

Table created.
```

Figure 12: Creating Student_Info table

Creating table for Student

CREATE TABLE Student(

Student_ID INT NOT NULL,

Std_Name VARCHAR(30) NOT NULL,

Std_Age INT NOT NULL,

Std_Gender VARCHAR(10) NOT NULL,

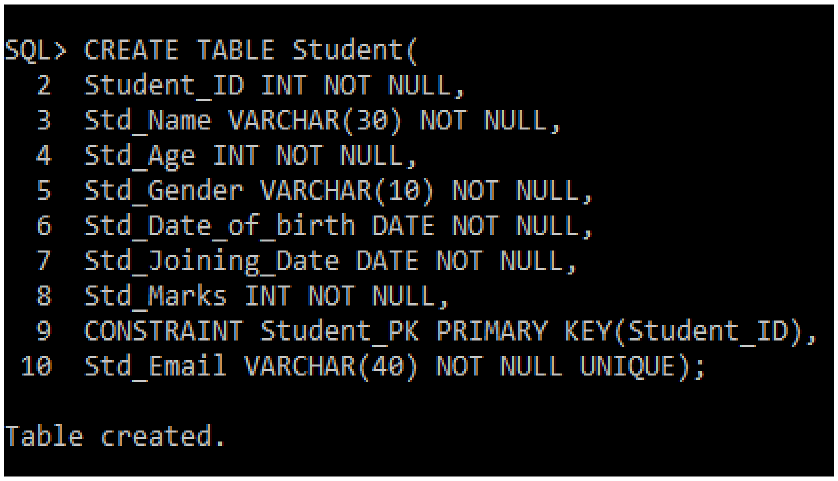
Std_Date_of_birth DATE NOT NULL,

Std_Joining_Date DATE NOT NULL,

Std_Marks INT NOT NULL,

CONSTRAINT Student_PK PRIMARY KEY(Student_ID),

Std_Email VARCHAR(40) NOT NULL UNIQUE);



```
SQL> CREATE TABLE Student(
     Student_ID INT NOT NULL,
     Std_Name VARCHAR(30) NOT NULL,
     Std_Age INT NOT NULL,
     Std_Gender VARCHAR(10) NOT NULL,
     Std_Date_of_birth DATE NOT NULL,
     Std_Joining_Date DATE NOT NULL,
     Std_Marks INT NOT NULL,
     CONSTRAINT Student_PK PRIMARY KEY(Student_ID),
     Std_Email VARCHAR(40) NOT NULL UNIQUE);

Table created.
```

Figure 13: Creating Student table

Creating table for Student_Address_Info

CREATE TABLE Student_Address_Info(

StdAddress_ID INT NOT NULL,

Student_ID INT NOT NULL,

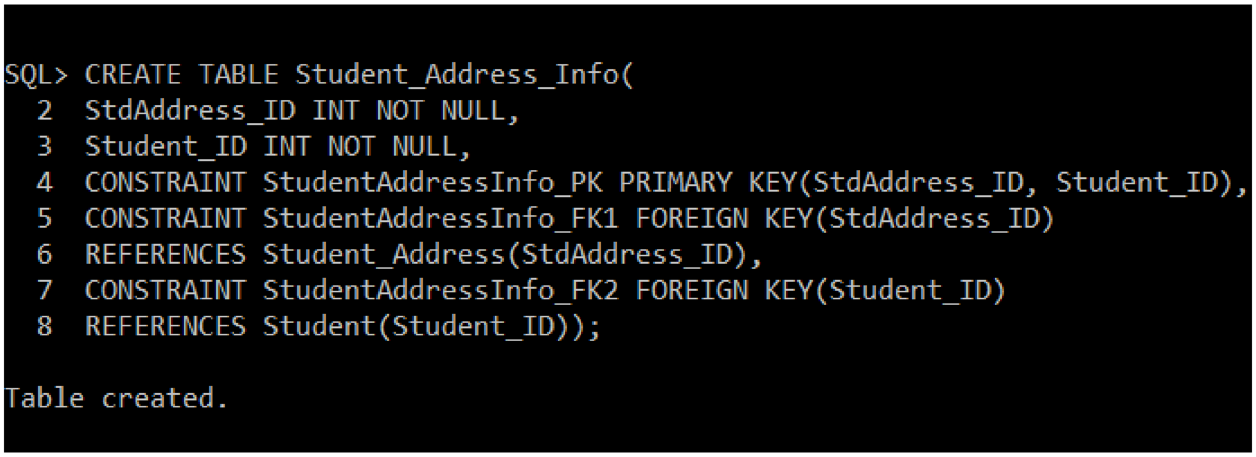
CONSTRAINT StudentAddressInfo_PK PRIMARY KEY(StdAddress_ID, Student_ID),

CONSTRAINT StudentAddressInfo_FK1 FOREIGN KEY(StdAddress_ID)

REFERENCES Student_Address(StdAddress_ID),

CONSTRAINT StudentAddressInfo_FK2 FOREIGN KEY(Student_ID)

REFERENCES Student(Student_ID));



```
SQL> CREATE TABLE Student_Address_Info(
  2  StdAddress_ID INT NOT NULL,
  3  Student_ID INT NOT NULL,
  4  CONSTRAINT StudentAddressInfo_PK PRIMARY KEY(StdAddress_ID, Student_ID),
  5  CONSTRAINT StudentAddressInfo_FK1 FOREIGN KEY(StdAddress_ID)
  6  REFERENCES Student_Address(StdAddress_ID),
  7  CONSTRAINT StudentAddressInfo_FK2 FOREIGN KEY(Student_ID)
  8  REFERENCES Student(Student_ID));

Table created.
```

Figure 14: Creating Student_Address_Info table

Creating table for Student_Address

CREATE TABLE Student_Address(

StdAddress_ID INT NOT NULL,

Std_HouseNo INT NOT NULL,

Std_Country VARCHAR(30) NOT NULL,

Std_Province VARCHAR(30) NOT NULL,

Std_City VARCHAR(30) NOT NULL,

Std_Street VARCHAR(30) NOT NULL,

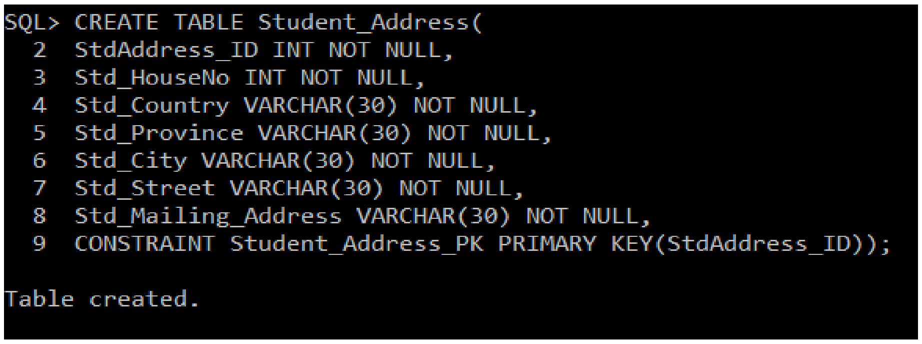
Std_Mailing_Address VARCHAR(30) NOT NULL,

CONSTRAINT Student_Address_PK PRIMARY KEY(StdAddress_ID));

ALTER TABLE Student_Address

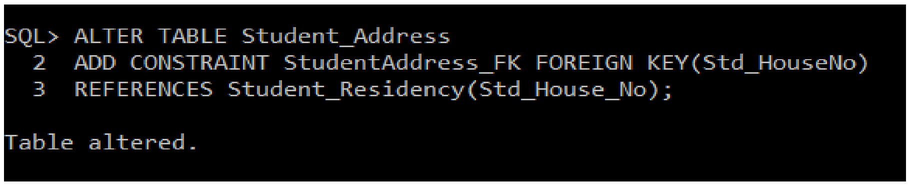
ADD CONSTRAINT StudentAddress_FK FOREIGN KEY(Std_HouseNo)

REFERENCES Student_Residency(Std_House_No);



```
SQL> CREATE TABLE Student_Address(
  2  StdAddress_ID INT NOT NULL,
  3  Std_HouseNo INT NOT NULL,
  4  Std_Country VARCHAR(30) NOT NULL,
  5  Std_Province VARCHAR(30) NOT NULL,
  6  Std_City VARCHAR(30) NOT NULL,
  7  Std_Street VARCHAR(30) NOT NULL,
  8  Std_Mailing_Address VARCHAR(30) NOT NULL,
  9  CONSTRAINT Student_Address_PK PRIMARY KEY(StdAddress_ID));

Table created.
```

Figure 15: Creating Student_Address table



```
SQL> ALTER TABLE Student_Address
  2  ADD CONSTRAINT StudentAddress_FK FOREIGN KEY(Std_HouseNo)
  3  REFERENCES Student_Residency(Std_House_No);

Table altered.
```

Figure 16: ALTER Student_Address table

Creating table for Student_Residency

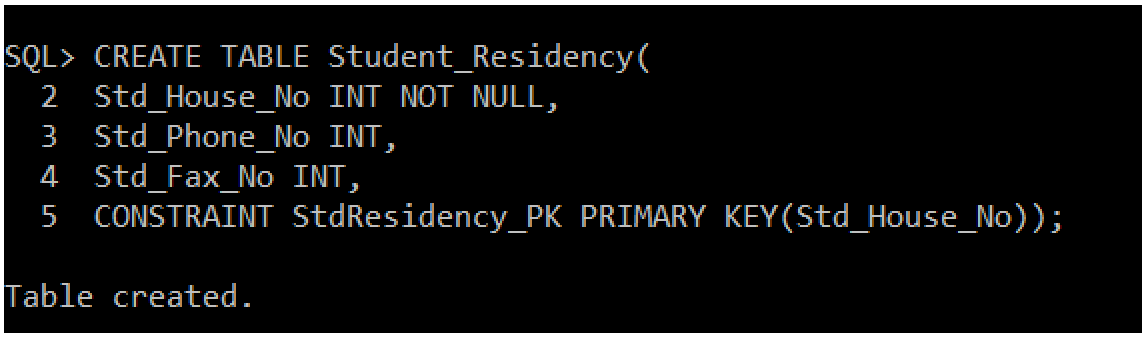
CREATE TABLE Student_Residency(

Std_House_No INT NOT NULL,

Std_Phone_No INT,

Std_Fax_No INT,

CONSTRAINT StdResidency_PK PRIMARY KEY(Std_House_No));



```
SQL> CREATE TABLE Student_Residency(
  2  Std_House_No INT NOT NULL,
  3  Std_Phone_No INT,
  4  Std_Fax_No INT,
  5  CONSTRAINT StdResidency_PK PRIMARY KEY(Std_House_No));

Table created.
```

Figure 17: Creating Student_Residency table

3.2. Populating Database tables

To insert data into tables, SQL INSERT statement has to be used. The SQL INSERT statement is used to insert a one or more records into a table (Tech on the net, 2020). To make the changes made in the current transaction, COMMIT statement has to be used.

Inserting values in Course table

```
INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name)
VALUES
(1001, 'BIT', 114000, 'Computing');


INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name)
VALUES
(1002, 'BIT', 114000, 'Multimedia');


INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name)
VALUES
(1003, 'BIT', 114000, 'Networking');


INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name)
VALUES
(1004, 'BBA', 114000, 'Marketing');
```

INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES

(1005, 'BBA', 114000, 'Finance');

INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES

(1006, 'BCA', 114000, 'Computer Apps');

INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES

(1007, 'BIM', 114000, 'Info. Management');

INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES

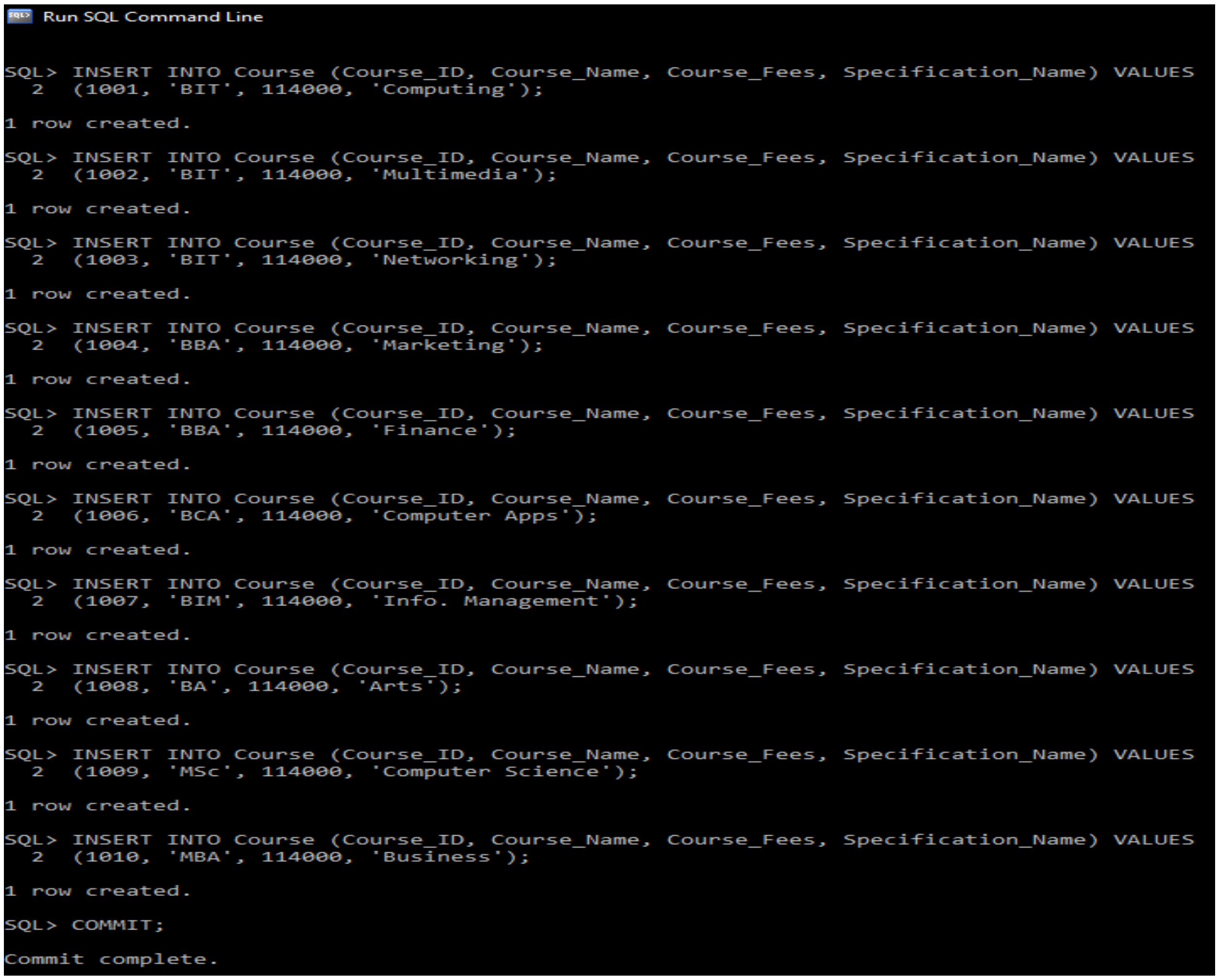
(1008, 'BA', 114000, 'Arts');

INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES

(1009, 'MSc', 114000, 'Computer Science');

INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES

(1010, 'MBA', 114000, 'Business');



Run SQL Command Line

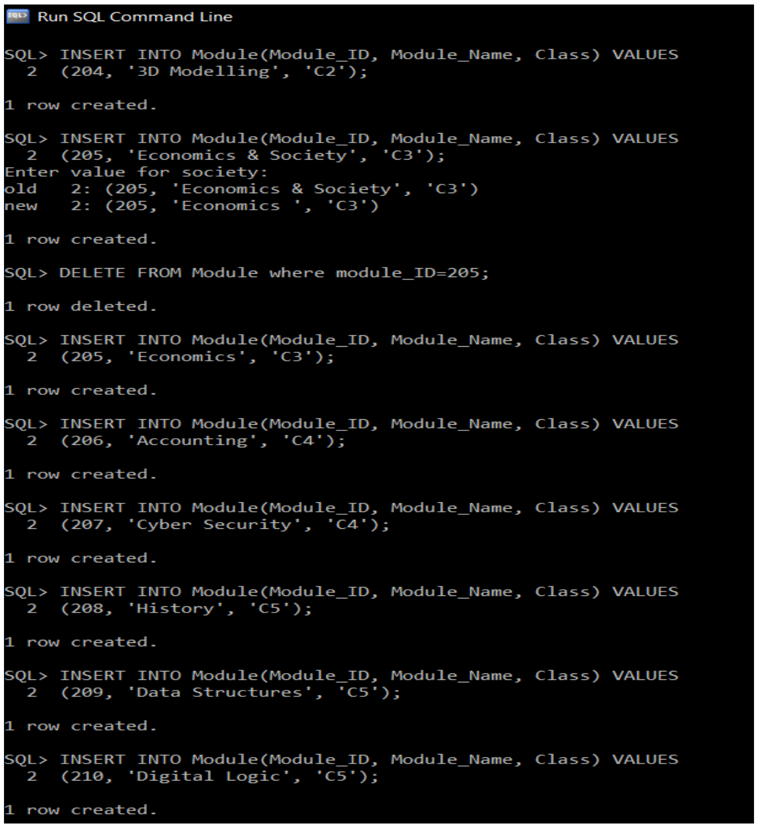
```
SQL> INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES
  2  (1001, 'BIT', 114000, 'Computing');

1 row created.

SQL> INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES
  2  (1002, 'BIT', 114000, 'Multimedia');

1 row created.

SQL> INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES
  2  (1003, 'BIT', 114000, 'Networking');

1 row created.

SQL> INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES
  2  (1004, 'BBA', 114000, 'Marketing');

1 row created.

SQL> INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES
  2  (1005, 'BBA', 114000, 'Finance');

1 row created.

SQL> INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES
  2  (1006, 'BCA', 114000, 'Computer Apps');

1 row created.

SQL> INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES
  2  (1007, 'BIM', 114000, 'Info. Management');

1 row created.

SQL> INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES
  2  (1008, 'BA', 114000, 'Arts');

1 row created.

SQL> INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES
  2  (1009, 'MSc', 114000, 'Computer Science');

1 row created.

SQL> INSERT INTO Course (Course_ID, Course_Name, Course_Fees, Specification_Name) VALUES
  2  (1010, 'MBA', 114000, 'Business');

1 row created.

SQL> COMMIT;

Commit complete.
```

Figure 18: Inserting values in Course table

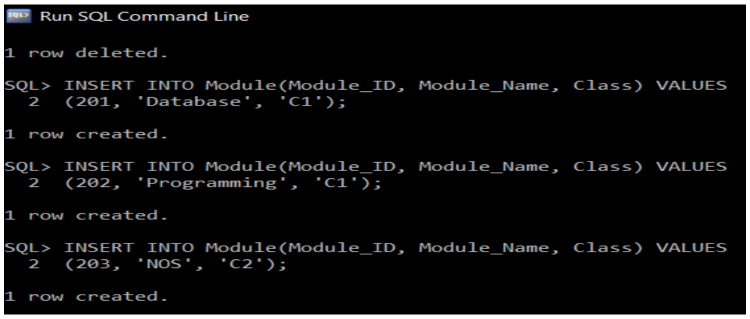
Inserting values in Module table

```
INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
(201, 'Database', 'C1');

INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
(202, 'Programming', 'C1');

INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
(203, 'NOS', 'C2');

INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
(204, '3D Modelling', 'C2');

INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
(205, 'Economics', 'C3');

INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
(206, 'Accounting', 'C4');

INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
(207, 'Cyber Security', 'C4');
```

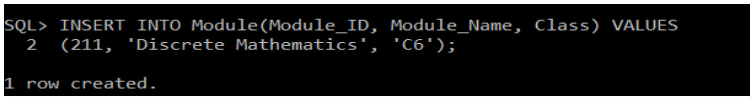
```
INSERT INTO Module(Module_ID, Module_Name, Class) VALUES

(208, 'History', 'C5');


INSERT INTO Module(Module_ID, Module_Name, Class) VALUES

(209, 'Data Structures', 'C5');


INSERT INTO Module(Module_ID, Module_Name, Class) VALUES

(210, 'Digital Logic', 'C5');


INSERT INTO Module(Module_ID, Module_Name, Class) VALUES

(211, 'Discrete Mathematics', 'C6');


INSERT INTO Module(Module_ID, Module_Name, Class) VALUES

(212, 'Digital Design', 'C5');


INSERT INTO Module(Module_ID, Module_Name, Class) VALUES

(213, 'Game Design', 'C5');


INSERT INTO Module(Module_ID, Module_Name, Class) VALUES

(214, 'Animation', 'C5');


INSERT INTO Module(Module_ID, Module_Name, Class) VALUES

(215, 'Image Making', 'C5');
```



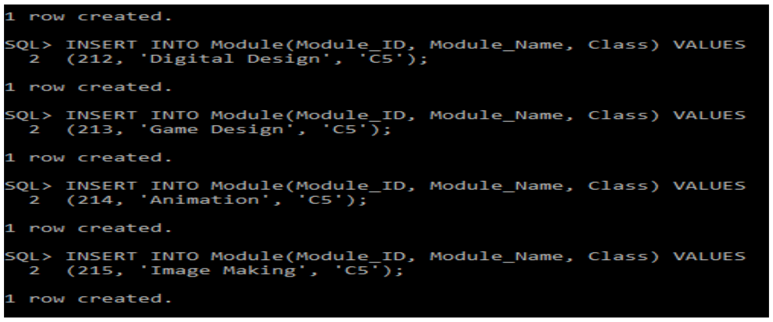
```
Run SQL Command Line

SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (204, '3D Modelling', 'C2');

1 row created.

SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (205, 'Economics & Society', 'C3');
Enter value for society:
old   2: (205, 'Economics & Society', 'C3')
new   2: (205, 'Economics ', 'C3')

1 row created.

SQL> DELETE FROM Module where module_ID=205;

1 row deleted.

SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (205, 'Economics', 'C3');

1 row created.

SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (206, 'Accounting', 'C4');

1 row created.

SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (207, 'Cyber Security', 'C4');

1 row created.

SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (208, 'History', 'C5');

1 row created.

SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (209, 'Data Structures', 'C5');

1 row created.

SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (210, 'Digital Logic', 'C5');

1 row created.
```



```
Run SQL Command Line

1 row deleted.

SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (201, 'Database', 'C1');

1 row created.

SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (202, 'Programming', 'C1');

1 row created.

SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (203, 'NOS', 'C2');

1 row created.
```



```
SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (211, 'Discrete Mathematics', 'C6');

1 row created.
```



```
1 row created.

SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (212, 'Digital Design', 'C5');

1 row created.

SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (213, 'Game Design', 'C5');

1 row created.

SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (214, 'Animation', 'C5');

1 row created.

SQL> INSERT INTO Module(Module_ID, Module_Name, Class) VALUES
  2  (215, 'Image Making', 'C5');

1 row created.
```

Figure 19: Inserting values in Module table

Inserting values in Instructor table

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(51, 'Saroj', 'Thapa', 40, 'Male', '09-Jan-80', '19-Jan-17', 'Course Leader', 60000, 'saroj@gmail.com');

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(52, 'Nidhi', 'Gupta', 32, 'Female', '09-APR-88', '25-Apr-16', 'Course Leader', 60000, 'nidhi@gmail.com');

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(53, 'Ram', 'Gopal', 33, 'Male', '03-JUN-87', '25-Mar-16', 'Course Leader', 60000, 'ram@gmail.com');

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(54, 'Nala', 'Shrestha', 35, 'Female', '04-DEC-85', '20-Mar-19', 'Course Leader', 60000, 'nala@gmail.com');

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(55, 'Mirza', 'Khan', 35, 'Male', '04-NOV-85', '20-Apr-15', 'Course Leader', 60000, 'mirza@gmail.com');

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(56, 'Simon', 'Shrestha', 28, 'Male', '10-DEC-92','5-AUG-18', 'Course Leader', 60000, 'simon@gmail.com');

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(57, 'Season', 'Khadka', 30, 'Male', '02-NOV-90', '19-Apr-19', 'Course Leader', 60000, 'season@gmail.com');

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(58, 'Aman', 'Maharjan', 31, 'Male', '10-DEC-89','5-AUG-19', 'Module Leader', 55000, 'aman@gmail.com');

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(59, 'Siddhartha', 'Ghimire', 25, 'Male', '10-NOV-95','5-AUG-19', 'Module Leader', 55000, 'siddhartha@gmail.com');

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(60, 'Neha', 'Banu', 31, 'Female', '15-DEC-89','20-AUG-19', 'Module Leader', 55000, 'neha@gmail.com');

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(61, 'Lubna', 'Karki', 27, 'Female', '12-MAY-93','20-AUG-18', 'Module Leader', 55000, 'lubna@gmail.com');

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(62, 'Ashish', 'RAI', 31, 'Male', '15-JUL-89','20-MAY-19', 'Module Leader', 55000, 'ashish@gmail.com');

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(63, 'Ashiq', 'RAJ', 26, 'Male', '11-SEP-94','05-OCT-19', 'Module Leader', 55000, 'ashiq@gmail.com');

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(64, 'Smriti', 'Basnet', 24, 'Female', '15-JUL-96','20-MAY-19', 'Module Leader', 55000, 'smriti@gmail.com');

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(65, 'Suman', 'Thapa', 34, 'Female', '11-SEP-86','05-SEP-19', 'Module Leader', 55000, 'suman@gmail.com');

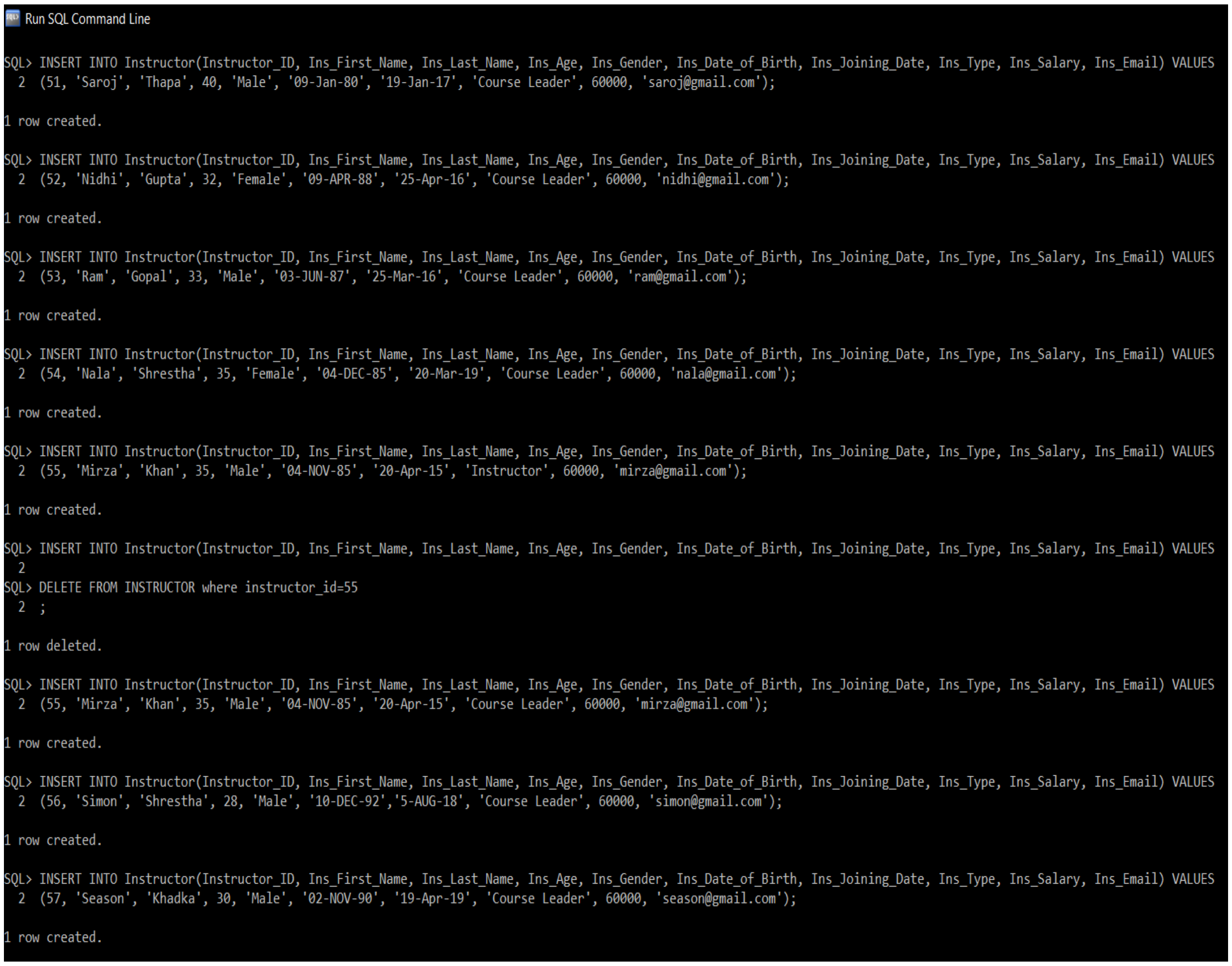
```
INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age,
Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
(66, 'Arun', 'Kumar', 39, 'Male', '15-JUL-81','20-MAY-16', 'Module Leader', 55000,
'arun@gmail.com');


INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age,
Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
(67, 'Kriti', 'Dangol', 39, 'Female', '15-FEB-81','20-MAY-16', 'Module Leader', 55000,
'kriti@gmail.com');


INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age,
Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
(68, 'Nawaraj', 'Kafle', 29, 'Male', '15-FEB-91','20-MAY-17', 'Instructor', 50000,
'nawaraj@gmail.com');


INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age,
Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
(69, 'Sudip', 'Shrestha', 36, 'Male', '15-FEB-84','17-MAY-17', 'Instructor', 50000,
'sudip@gmail.com');


INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age,
Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
(70, 'Rijan', 'Giri', 27, 'Male', '17-JAN-93','19-APR-18', 'Instructor', 50000, 'rijan@gmail.com');
```

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

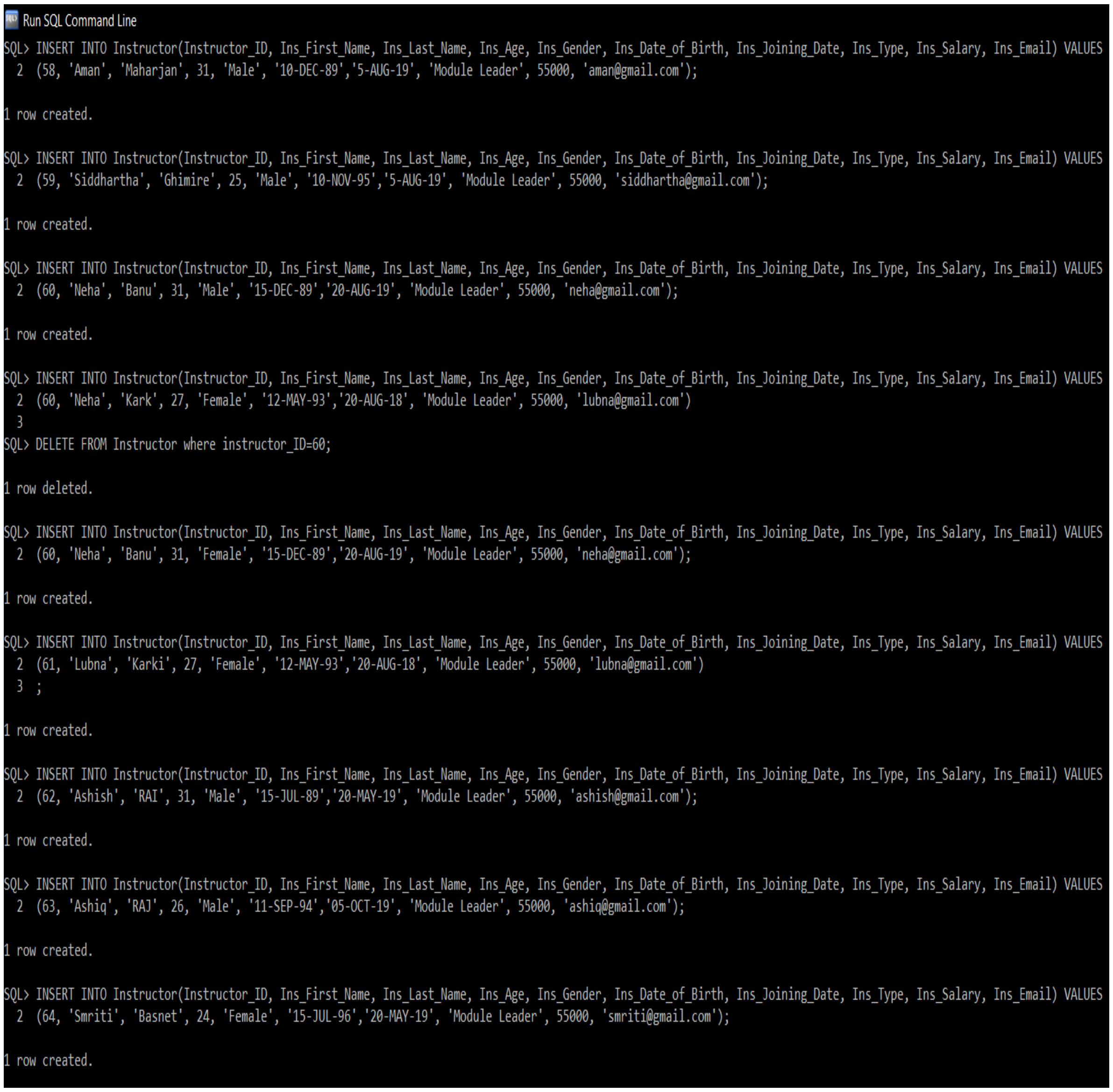
(71, 'Sara', 'Thapa', 40, 'Female', '09-Jan-80', '19-Jan-17', 'Module Leader', 55000, 'sara@gmail.com');

INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES

(72, 'Bajra', 'Bajracharya', 30, 'Male', '09-Apr-90', '19-May-18', 'Instructor', 50000, 'bajra@gmail.com');

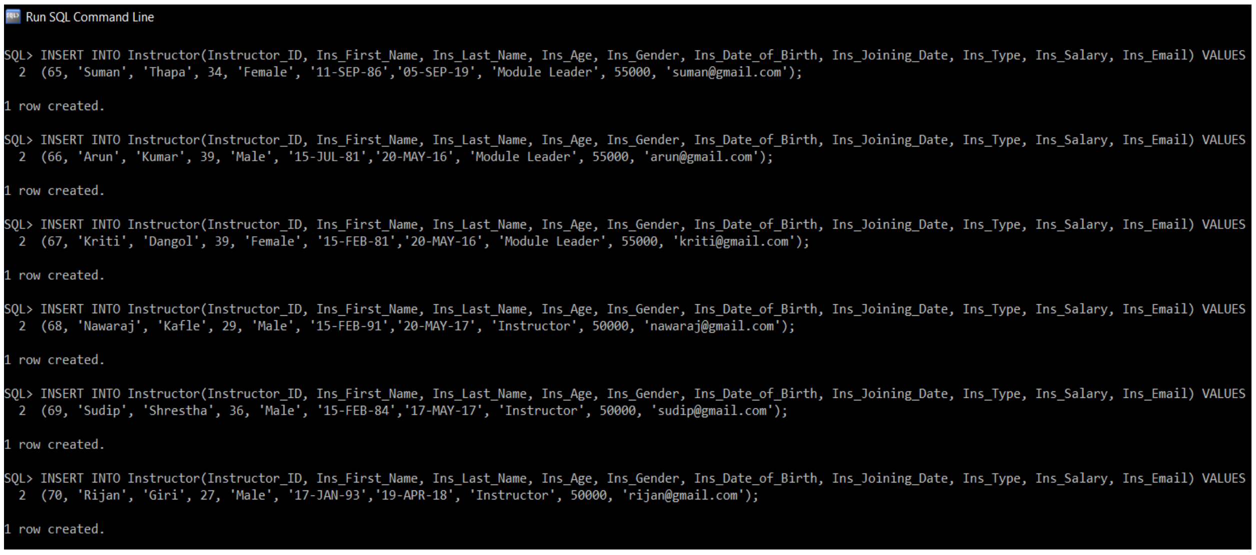

Run SQL Command Line

```
SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (51, 'Saroj', 'Thapa', 40, 'Male', '09-Jan-80', '19-Jan-17', 'Course Leader', 60000, 'saroj@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (52, 'Nidhi', 'Gupta', 32, 'Female', '09-APR-88', '25-Apr-16', 'Course Leader', 60000, 'nidhi@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (53, 'Ram', 'Gopal', 33, 'Male', '03-JUN-87', '25-Mar-16', 'Course Leader', 60000, 'ram@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (54, 'Nala', 'Shrestha', 35, 'Female', '04-DEC-85', '20-Mar-19', 'Course Leader', 60000, 'nala@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (55, 'Mirza', 'Khan', 35, 'Male', '04-NOV-85', '20-Apr-15', 'Instructor', 60000, 'mirza@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2
SQL> DELETE FROM INSTRUCTOR where instructor_id=55
  2  ;

1 row deleted.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (55, 'Mirza', 'Khan', 35, 'Male', '04-NOV-85', '20-Apr-15', 'Course Leader', 60000, 'mirza@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (56, 'Simon', 'Shrestha', 28, 'Male', '10-DEC-92','5-AUG-18', 'Course Leader', 60000, 'simon@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (57, 'Season', 'Khadka', 30, 'Male', '02-NOV-90', '19-Apr-19', 'Course Leader', 60000, 'season@gmail.com');

1 row created.
```



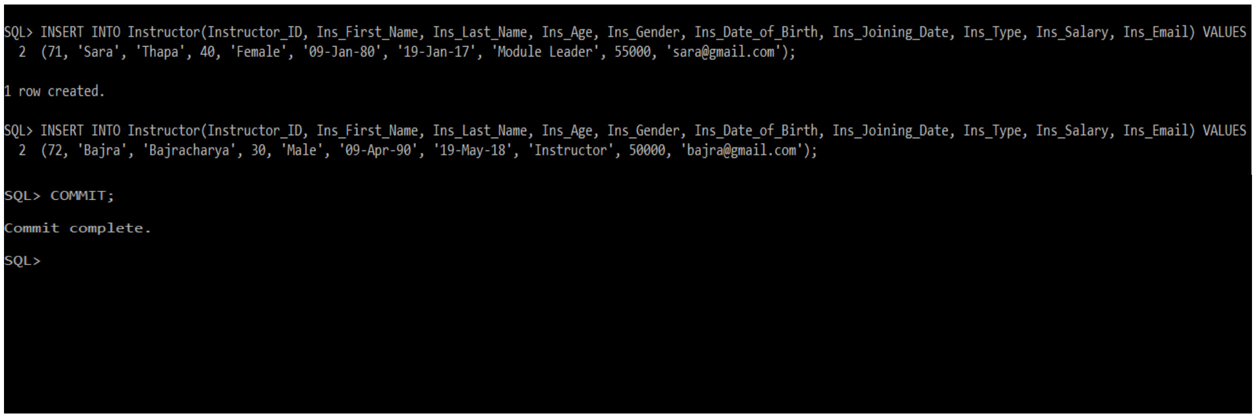
Run SQL Command Line

```
SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (58, 'Aman', 'Maharjan', 31, 'Male', '10-DEC-89','5-AUG-19', 'Module Leader', 55000, 'aman@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (59, 'Siddhartha', 'Ghimire', 25, 'Male', '10-NOV-95','5-AUG-19', 'Module Leader', 55000, 'siddhartha@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (60, 'Neha', 'Banu', 31, 'Male', '15-DEC-89','20-AUG-19', 'Module Leader', 55000, 'neha@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (60, 'Neha', 'Kark', 27, 'Female', '12-MAY-93','20-AUG-18', 'Module Leader', 55000, 'lubna@gmail.com')
  3
SQL> DELETE FROM Instructor where instructor_ID=60;

1 row deleted.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (60, 'Neha', 'Banu', 31, 'Female', '15-DEC-89','20-AUG-19', 'Module Leader', 55000, 'neha@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (61, 'Lubna', 'Karki', 27, 'Female', '12-MAY-93','20-AUG-18', 'Module Leader', 55000, 'lubna@gmail.com')
  3  ;

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (62, 'Ashish', 'RAI', 31, 'Male', '15-JUL-89','20-MAY-19', 'Module Leader', 55000, 'ashish@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (63, 'Ashiq', 'RAJ', 26, 'Male', '11-SEP-94','05-OCT-19', 'Module Leader', 55000, 'ashiq@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (64, 'Smriti', 'Basnet', 24, 'Female', '15-JUL-96','20-MAY-19', 'Module Leader', 55000, 'smriti@gmail.com');

1 row created.
```



Run SQL Command Line

```
SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (65, 'Suman', 'Thapa', 34, 'Female', '11-SEP-86','05-SEP-19', 'Module Leader', 55000, 'suman@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (66, 'Arun', 'Kumar', 39, 'Male', '15-JUL-81','20-MAY-16', 'Module Leader', 55000, 'arun@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (67, 'Kriti', 'Dangol', 39, 'Female', '15-FEB-81','20-MAY-16', 'Module Leader', 55000, 'kriti@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (68, 'Nawaraj', 'Kafle', 29, 'Male', '15-FEB-91','20-MAY-17', 'Instructor', 50000, 'nawaraj@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (69, 'Sudip', 'Shrestha', 36, 'Male', '15-FEB-84','17-MAY-17', 'Instructor', 50000, 'sudip@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (70, 'Rijan', 'Giri', 27, 'Male', '17-JAN-93','19-APR-18', 'Instructor', 50000, 'rijan@gmail.com');

1 row created.
```



```
SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (71, 'Sara', 'Thapa', 40, 'Female', '09-Jan-80', '19-Jan-17', 'Module Leader', 55000, 'sara@gmail.com');

1 row created.

SQL> INSERT INTO Instructor(Instructor_ID, Ins_First_Name, Ins_Last_Name, Ins_Age, Ins_Gender, Ins_Date_of_Birth, Ins_Joining_Date, Ins_Type, Ins_Salary, Ins_Email) VALUES
  2  (72, 'Bajra', 'Bajracharya', 30, 'Male', '09-Apr-90', '19-May-18', 'Instructor', 50000, 'bajra@gmail.com');

SQL> COMMIT;

Commit complete.

SQL>
```

Figure 20: Inserting values in Instructor table

Inserting value in Student table

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(1,'Anup Shrestha', 21, 'Male', '02-JAN-1999', '01-AUG-2019', 80, 'anup@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(2,'Azan Ahmed', 21, 'Male', '09-APR-1999', '10-SEP-2019', 90, 'azan@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(3,'Barsha Das', 22, 'Female', '05-MAR-1998', '10-SEP-2018', 81, 'barsha@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(4,'Bibek Paudyal', 22, 'Male', '06-JUL-1998', '05-SEP-2018', 82, 'bibek@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(5,'Biman Lakhey', 21, 'Male', '07-JUL-1999', '05-JUL-2019', 82, 'biman@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(6,'Dipesh Shrestha', 23, 'Male', '08-DEC-1997', '03-AUG-2018', 76, 'dipesh@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(7,'Gyanu Adhikari', 21, 'Female', '08-DEC-1999', '03-AUG-2019', 88, 'gyanu@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(8,'Rhythm', 21, 'Male', '01-NOV-1999', '03-SEP-2019', 85, 'rhythm@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(9,'Sanket Kadel', 22, 'Male', '26-OCT-1998', '03-SEP-2018', 86, 'sanket@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(10,'Neha Bharati', 22, 'Female', '01-NOV-1998', '03-SEP-2018', 85, 'neha@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(11,'Mimona Karki', 21, 'Female', '02-MAY-1999', '03-AUG-2019', 85, 'mimona@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(12,'Kishu Maharjan', 21, 'Male', '03-FEB-1999', '04-JUL-2019', 85, 'kishu@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(13,'Prajeet Kumar', 23, 'Male', '09-MAR-1997', '10-AUG-2017', 89, 'prajeet@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(14,'Suyogya Luitel', 20, 'Male', '09-MAR-2000', '10-AUG-2020', 84, 'suyogya@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(15,'Utsav Basyal', 20, 'Male', '29-JAN-2000', '5-AUG-2020', 83, 'utsav@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(16,'Prastut Paudel', 20, 'Male', '29-APR-2000', '05-AUG-2020', 84, 'prastut@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(17,'Roshan Adhikari', 20, 'Male', '30-MAY-2000', '11-AUG-2020', 84, 'roshan@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(18,'Sumit Khatri', 21, 'Male', '10-JUN-1999', '12-AUG-2020', 80, 'sumit@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(19,'Sahitya Rauniyar', 21, 'Male', '02-JAN-1999', '01-AUG-2019', 80, 'sahitya@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

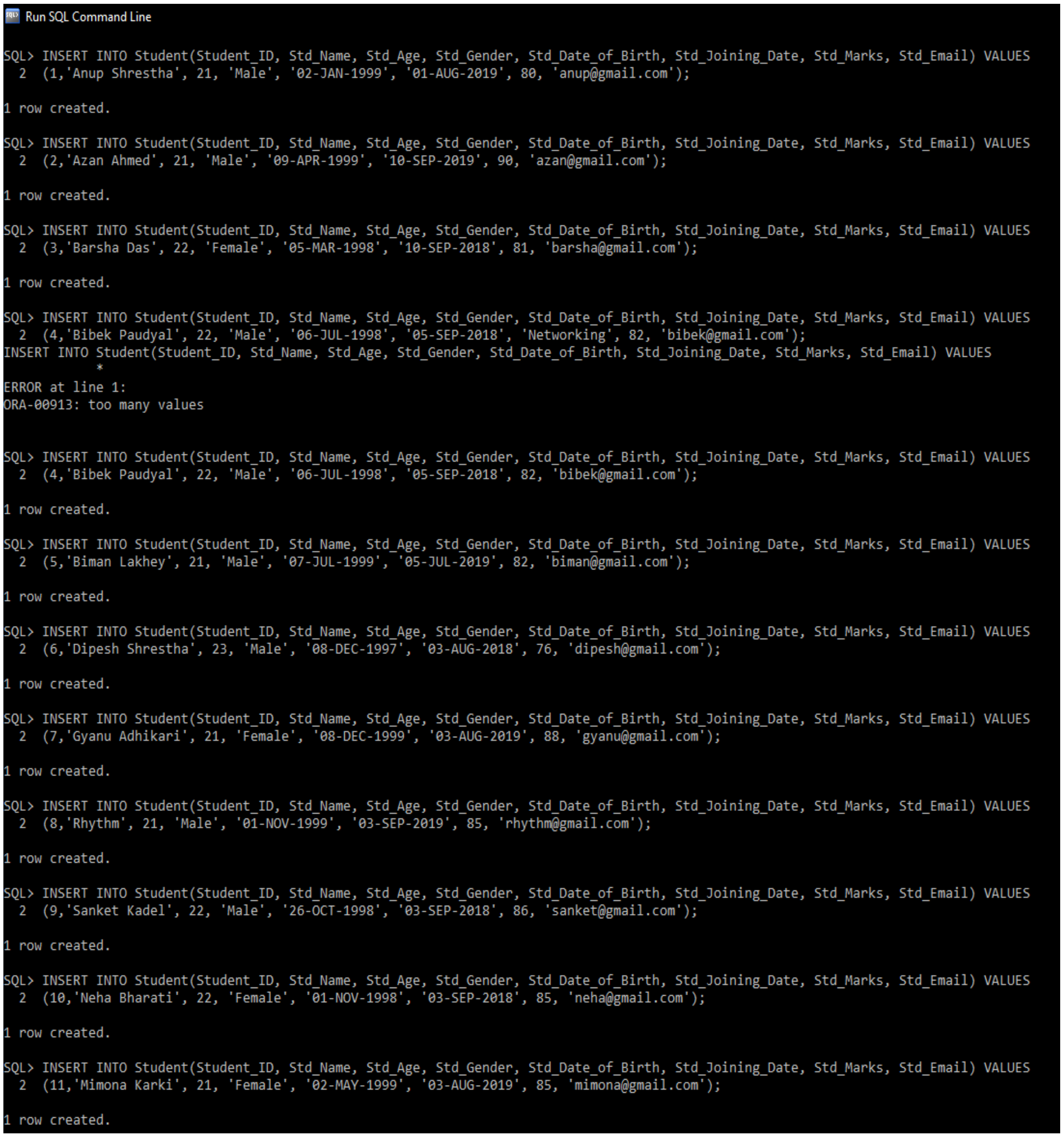
(20,'Amit', 22, 'Male', '02-JAN-1998', '01-AUG-2018', 80, 'amit@gmail.com');

INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(21,'Prism Koirala', 22, 'Male', '02-MAY-1998', '02-AUG-2018', 85, 'prism@gmail.com');

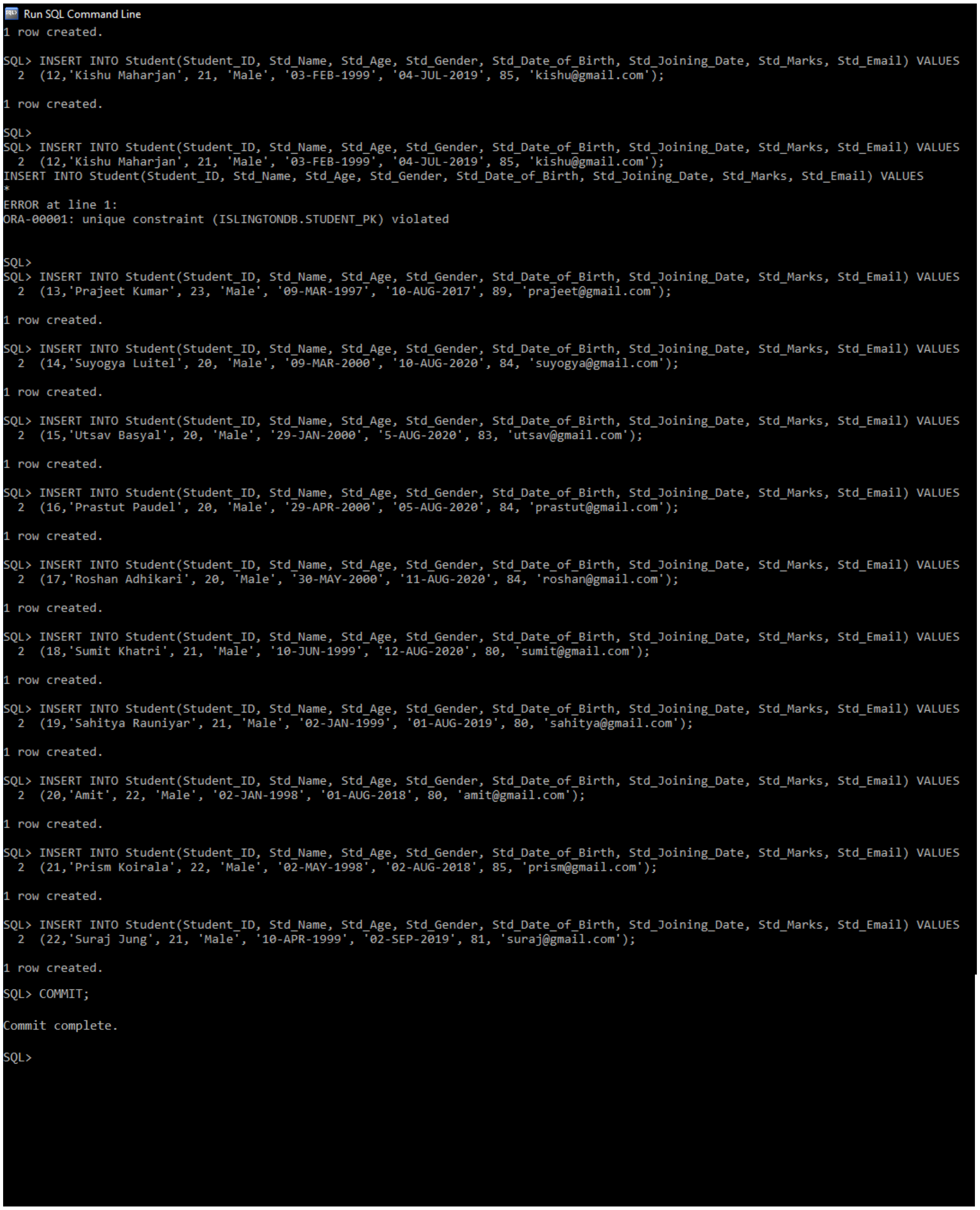
INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES

(22,'Suraj Jung', 21, 'Male', '10-APR-1999', '02-SEP-2019', 81, 'suraj@gmail.com');



Run SQL Command Line

```
SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (1,'Anup Shrestha', 21, 'Male', '02-JAN-1999', '01-AUG-2019', 80, 'anup@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (2,'Azan Ahmed', 21, 'Male', '09-APR-1999', '10-SEP-2019', 90, 'azan@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (3,'Barsha Das', 22, 'Female', '05-MAR-1998', '10-SEP-2018', 81, 'barsha@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (4,'Bibek Paudyal', 22, 'Male', '06-JUL-1998', '05-SEP-2018', 'Networking', 82, 'bibek@gmail.com');
INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
            *
ERROR at line 1:
ORA-00913: too many values


SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (4,'Bibek Paudyal', 22, 'Male', '06-JUL-1998', '05-SEP-2018', 82, 'bibek@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (5,'Biman Lakhey', 21, 'Male', '07-JUL-1999', '05-JUL-2019', 82, 'biman@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (6,'Dipesh Shrestha', 23, 'Male', '08-DEC-1997', '03-AUG-2018', 76, 'dipesh@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (7,'Gyanu Adhikari', 21, 'Female', '08-DEC-1999', '03-AUG-2019', 88, 'gyanu@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (8,'Rhythm', 21, 'Male', '01-NOV-1999', '03-SEP-2019', 85, 'rhythm@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (9,'Sanket Kadel', 22, 'Male', '26-OCT-1998', '03-SEP-2018', 86, 'sanket@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (10,'Neha Bharati', 22, 'Female', '01-NOV-1998', '03-SEP-2018', 85, 'neha@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (11,'Mimona Karki', 21, 'Female', '02-MAY-1999', '03-AUG-2019', 85, 'mimona@gmail.com');

1 row created.
```



Run SQL Command Line

```
1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (12,'Kishu Maharjan', 21, 'Male', '03-FEB-1999', '04-JUL-2019', 85, 'kishu@gmail.com');

1 row created.

SQL>
SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (12,'Kishu Maharjan', 21, 'Male', '03-FEB-1999', '04-JUL-2019', 85, 'kishu@gmail.com');
INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
*
ERROR at line 1:
ORA-00001: unique constraint (ISLINGTONDB.STUDENT_PK) violated


SQL>
SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (13,'Prajeet Kumar', 23, 'Male', '09-MAR-1997', '10-AUG-2017', 89, 'prajeet@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (14,'Suyogya Luitel', 20, 'Male', '09-MAR-2000', '10-AUG-2020', 84, 'suyogya@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (15,'Utsav Basyal', 20, 'Male', '29-JAN-2000', '5-AUG-2020', 83, 'utsav@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (16,'Prastut Paudel', 20, 'Male', '29-APR-2000', '05-AUG-2020', 84, 'prastut@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (17,'Roshan Adhikari', 20, 'Male', '30-MAY-2000', '11-AUG-2020', 84, 'roshan@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (18,'Sumit Khatri', 21, 'Male', '10-JUN-1999', '12-AUG-2020', 80, 'sumit@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (19,'Sahitya Rauniyar', 21, 'Male', '02-JAN-1999', '01-AUG-2019', 80, 'sahitya@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (20,'Amit', 22, 'Male', '02-JAN-1998', '01-AUG-2018', 80, 'amit@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (21,'Prism Koirala', 22, 'Male', '02-MAY-1998', '02-AUG-2018', 85, 'prism@gmail.com');

1 row created.

SQL> INSERT INTO Student(Student_ID, Std_Name, Std_Age, Std_Gender, Std_Date_of_Birth, Std_Joining_Date, Std_Marks, Std_Email) VALUES
  2  (22,'Suraj Jung', 21, 'Male', '10-APR-1999', '02-SEP-2019', 81, 'suraj@gmail.com');

1 row created.

SQL> COMMIT;

Commit complete.

SQL>
```

Figure 21: Inserting values in Student table

Inserting values in Instructor_Residency table

```
INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
(701, 8921121212, 21345);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
(702, NULL, 12345);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
(703, NULL, 32345);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
 (704, 9803712345, NULL);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
(705, 9893712345, NULL);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
(706, 9893712343, 12345);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
(707, NULL, 54321);
```

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES

(708, 9999999999, 54323);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES

(709, 9999999991, 54322);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES

(710, 9888888888, 54325);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES

(711, 9788888888, NULL);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES

(712, 9688888886, 34343);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES

(713, 9988888886, 44343);

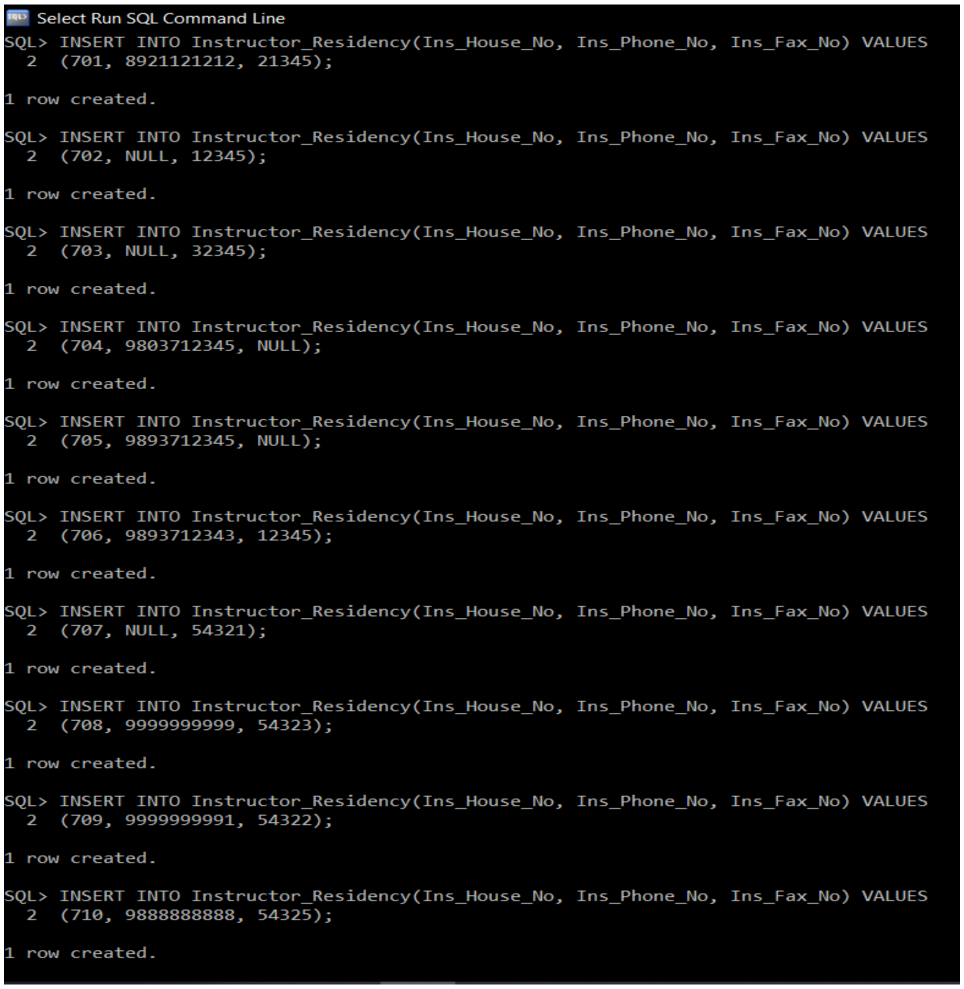
INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES

(714, NULL, 54343);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES

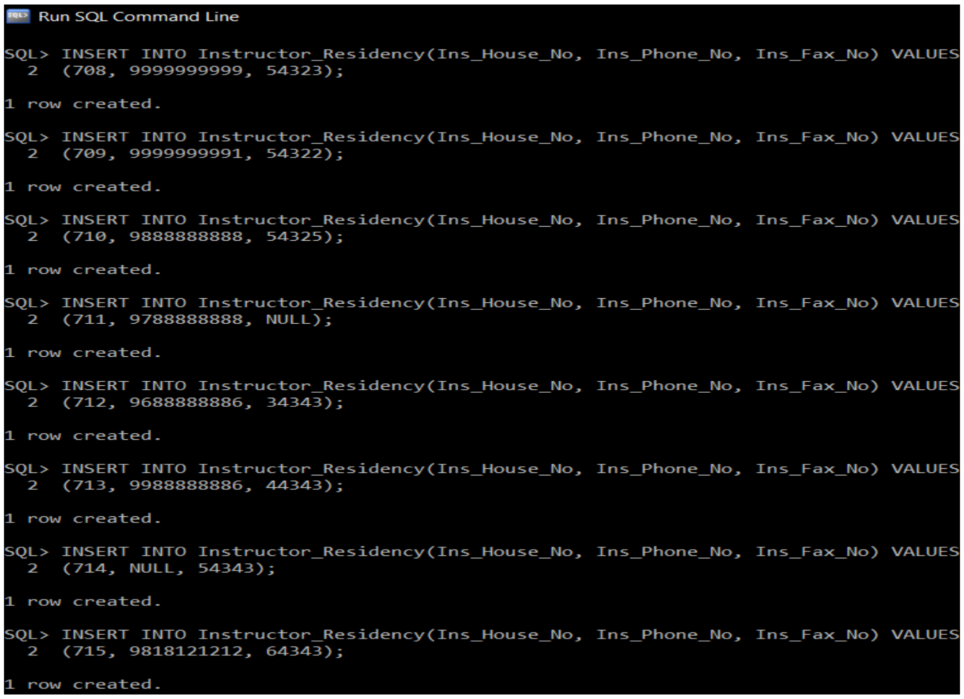
(715, 9818121212, 64343);

```
INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
(716, 9828121213, 63343);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
(717, 9828122222, NULL);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
(718, 9828127772, 71717);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
(719, 9828127773, 71716);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
(720, 9838127775, 72715);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
(721, 9838166775, 72717);

INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
(722, 9838169999, 92719);
```



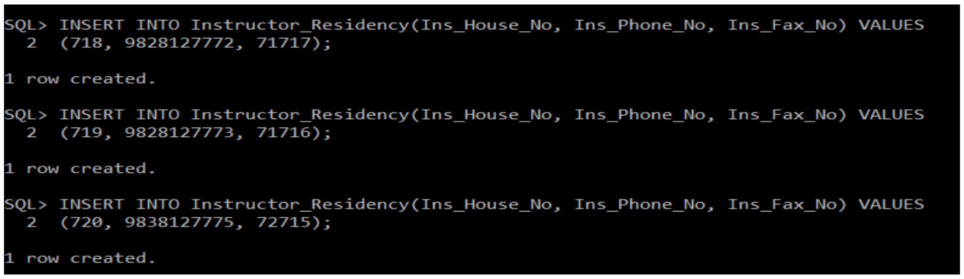
Select Run SQL Command Line

```
SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (701, 8921121212, 21345);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (702, NULL, 12345);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (703, NULL, 32345);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (704, 9803712345, NULL);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (705, 9893712345, NULL);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (706, 9893712343, 12345);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (707, NULL, 54321);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (708, 9999999999, 54323);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (709, 9999999991, 54322);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (710, 9888888888, 54325);

1 row created.
```

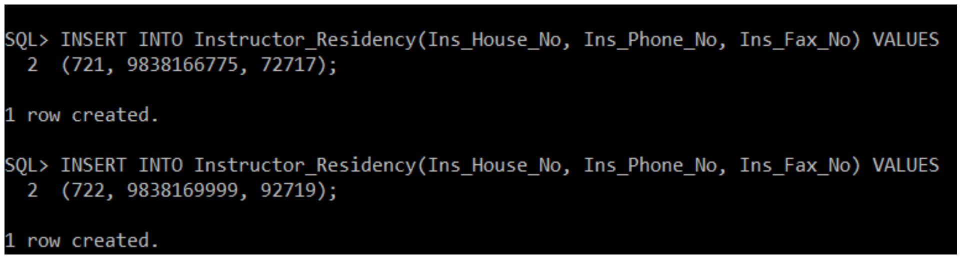


Run SQL Command Line

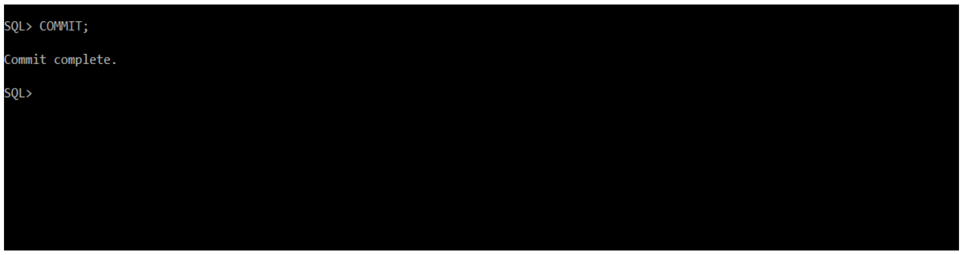
```
SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (708, 9999999999, 54323);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (709, 9999999991, 54322);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (710, 9888888888, 54325);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (711, 9788888888, NULL);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (712, 9688888886, 34343);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (713, 9988888886, 44343);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (714, NULL, 54343);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (715, 9818121212, 64343);

1 row created.
```



```
SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (718, 9828127772, 71717);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (719, 9828127773, 71716);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (720, 9838127775, 72715);

1 row created.
```



```
SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (721, 9838166775, 72717);

1 row created.

SQL> INSERT INTO Instructor_Residency(Ins_House_No, Ins_Phone_No, Ins_Fax_No) VALUES
  2  (722, 9838169999, 92719);

1 row created.
```



```
SQL> COMMIT;

Commit complete.

SQL>
```

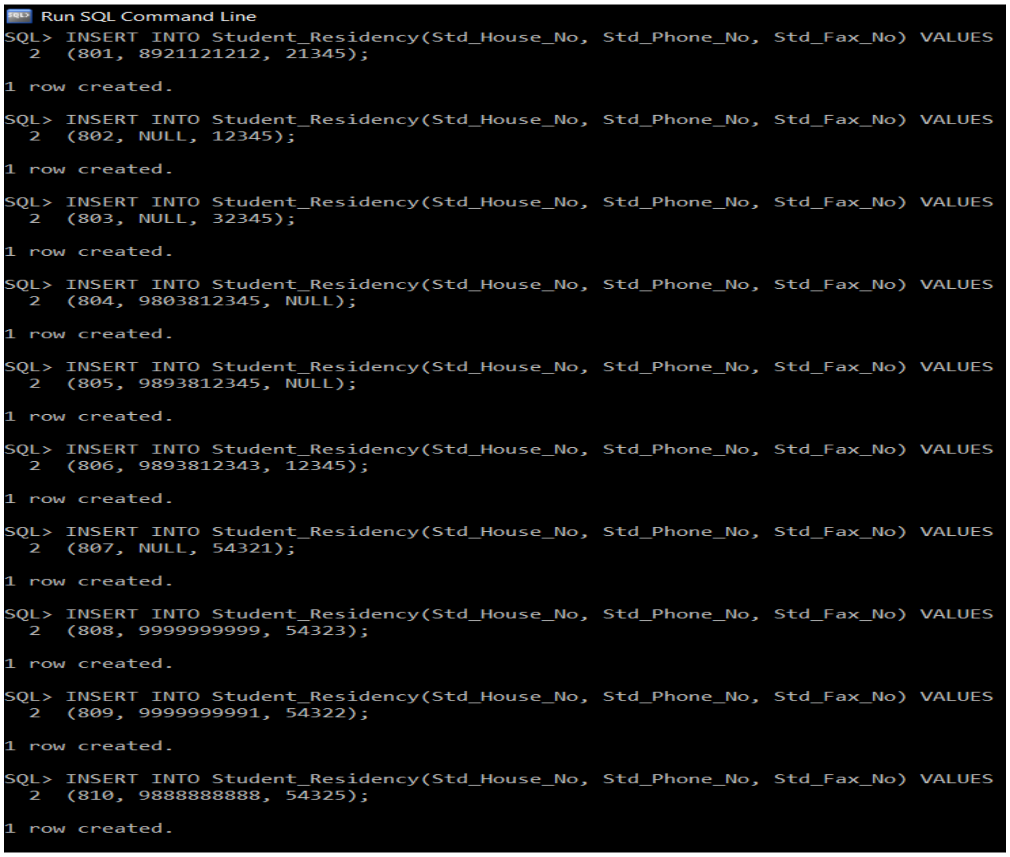
Figure 22: Inserting values in Instructor_Residency table

Inserting values in Student_Residency table

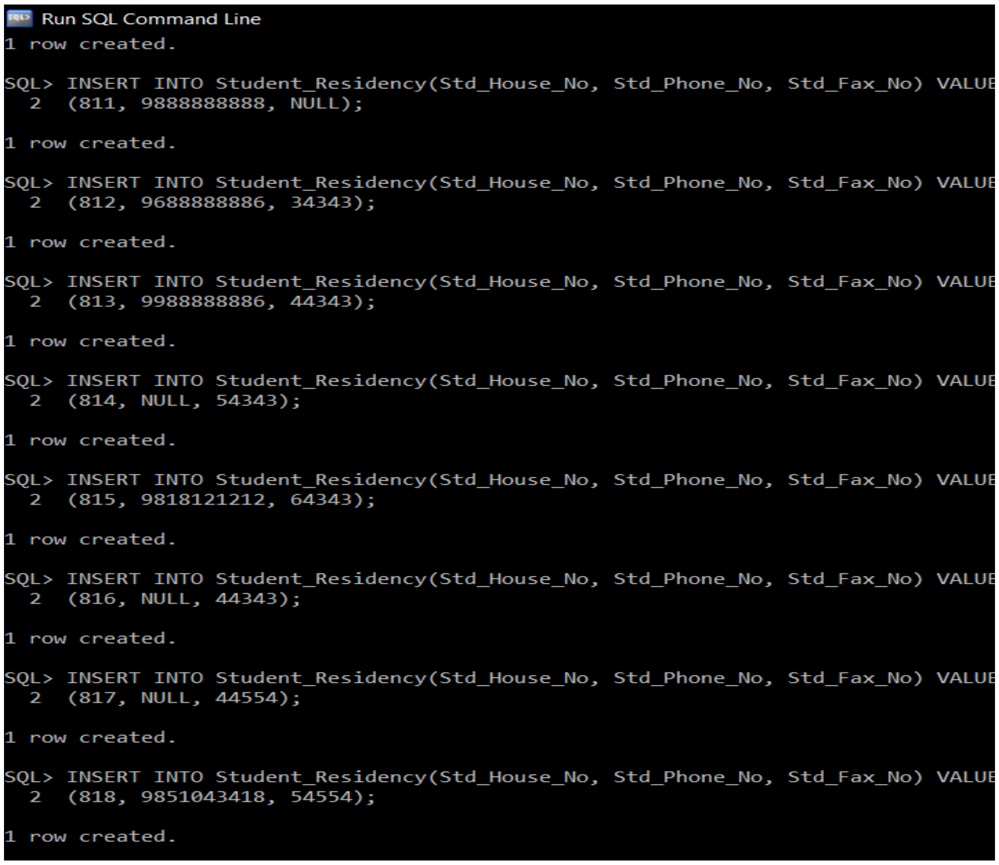
```
INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(801, 8921121212, 21345);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(802, NULL, 12345);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(803, NULL, 32345);
INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(804, 9803812345, NULL);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(805, 9893812345, NULL);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(806, 9893812343, 12345);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(807, NULL, 54321);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
```

```
(808, 9999999999, 54323);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(809, 9999999991, 54322);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(810, 9888888888, 54325);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(811, 9888888888, NULL);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(812, 9688888886, 34343);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(813, 9988888886, 44343);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(814, NULL, 54343);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(815, 9818121212, 64343);
```

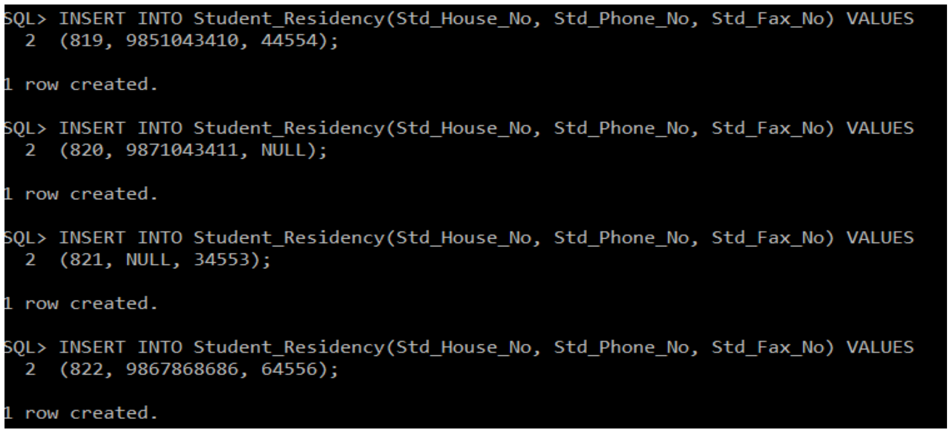
```
INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(816, NULL, 44343);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(817, NULL, 44554);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(818, 9851043418, 54554);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(819, 9851043410, 44554);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(820, 9871043411, NULL);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(821, NULL, 34553);

INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
(822, 9867868686, 64556);
```



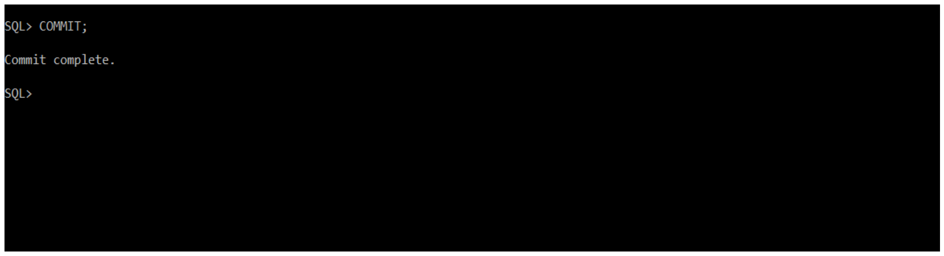
```
Run SQL Command Line
SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
  2  (801, 8921121212, 21345);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
  2  (802, NULL, 12345);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
  2  (803, NULL, 32345);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
  2  (804, 9803812345, NULL);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
  2  (805, 9893812345, NULL);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
  2  (806, 9893812343, 12345);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
  2  (807, NULL, 54321);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
  2  (808, 9999999999, 54323);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
  2  (809, 9999999991, 54322);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
  2  (810, 9888888888, 54325);

1 row created.
```



```
Run SQL Command Line
1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUE
  2  (811, 9888888888, NULL);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUE
  2  (812, 9688888886, 34343);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUE
  2  (813, 9988888886, 44343);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUE
  2  (814, NULL, 54343);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUE
  2  (815, 9818121212, 64343);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUE
  2  (816, NULL, 44343);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUE
  2  (817, NULL, 44554);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUE
  2  (818, 9851043418, 54554);

1 row created.
```



```
SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
  2  (819, 9851043410, 44554);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
  2  (820, 9871043411, NULL);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
  2  (821, NULL, 34553);

1 row created.

SQL> INSERT INTO Student_Residency(Std_House_No, Std_Phone_No, Std_Fax_No) VALUES
  2  (822, 9867868686, 64556);

1 row created.
```



```
SQL> COMMIT;

Commit complete.

SQL>
```

Figure 23: Inserting values in Student_Residency table

Inserting values in Instructor_Address table

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9111, 701, 'Nepal', 'Bagmati', 'Kathmandu', 'Bafal', '701Bafal');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9112, 702, 'Nepal', 'Bagmati', 'Kathmandu', 'Tachal', '702Tachal');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9113, 703, 'Nepal', 'Bagmati', 'Kathmandu', 'Kalimati', '703Kalimati');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9114, 704, 'Nepal', 'Bagmati', 'Kathmandu', 'Salt Trading', '704SaltTrading');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9115, 705, 'Nepal', 'Bagmati', 'Kathmandu', 'Soaltee Mode', '705SoalteeMode');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9116, 706, 'Nepal', 'Bagmati', 'Kathmandu', 'Kalanki', '706Kalanki');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9117, 707, 'Nepal', 'Bagmati', 'Kathmandu', 'Balkhu', '707Balkhu');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9118, 708, 'Nepal', 'Bagmati', 'Kathmandu', 'Chobhar', '708Chobhar');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9119, 709, 'Nepal', 'Bagmati', 'Kathmandu', 'New Road', '709NewRoad');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9120, 710, 'Nepal', 'Bagmati', 'Kathmandu', 'Jawlakhel', '710Jawlakhel');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9121, 711, 'Nepal', 'Bagmati', 'Kathmandu', 'Sitapaila', '711Sitapaila');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9122, 712, 'Nepal', 'Bagmati', 'Kathmandu', 'Chauni', '712Chauni');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9123, 713, 'Nepal', 'Bagmati', 'Kathmandu', 'Baneshwor', '713Baneshwor');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9124, 714, 'Nepal', 'Bagmati', 'Kathmandu', 'Naxal', '714Naxal');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9125, 715, 'Nepal', 'Bagmati', 'Kathmandu', 'Jamal', '715Jamal');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9126, 716, 'Nepal', 'Bagmati', 'Kathmandu', 'Baneshwor', '716Baneshwor');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9127, 717, 'Nepal', 'Bagmati', 'Kathmandu', 'Nakkhu', '717Nakkhu');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9128, 718, 'Nepal', 'Bagmati', 'Kathmandu', 'Kuleshwor', '718Kuleshwor');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9129, 719, 'Nepal', 'Bagmati', 'Kathmandu', 'Sanepa', '719Sanepa');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

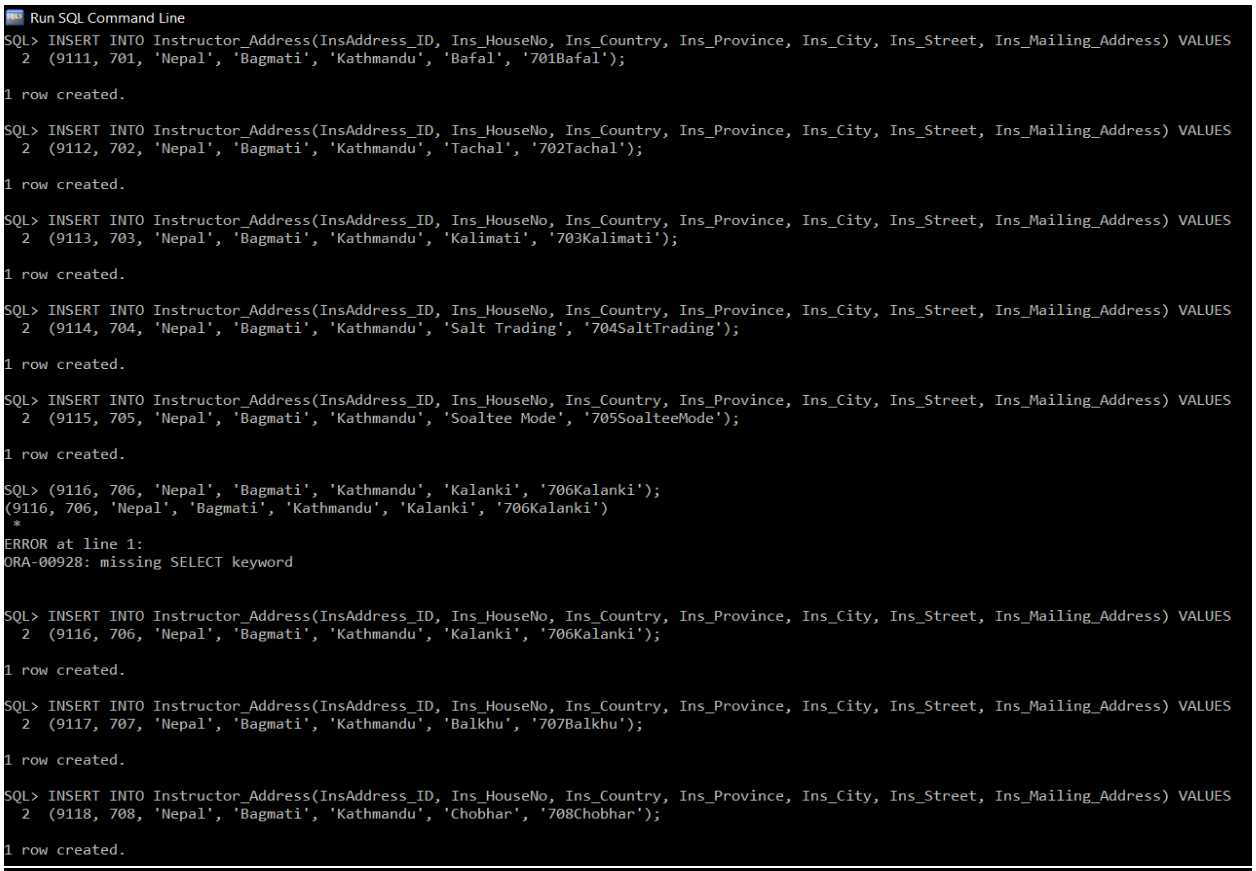
(9130, 720, 'Nepal', 'Bagmati', 'Kathmandu', 'Patan', '720Patan');

INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9131, 721, 'Nepal', 'Bagmati', 'Kathmandu', 'Ravi Bhawan', '721Bhawan');

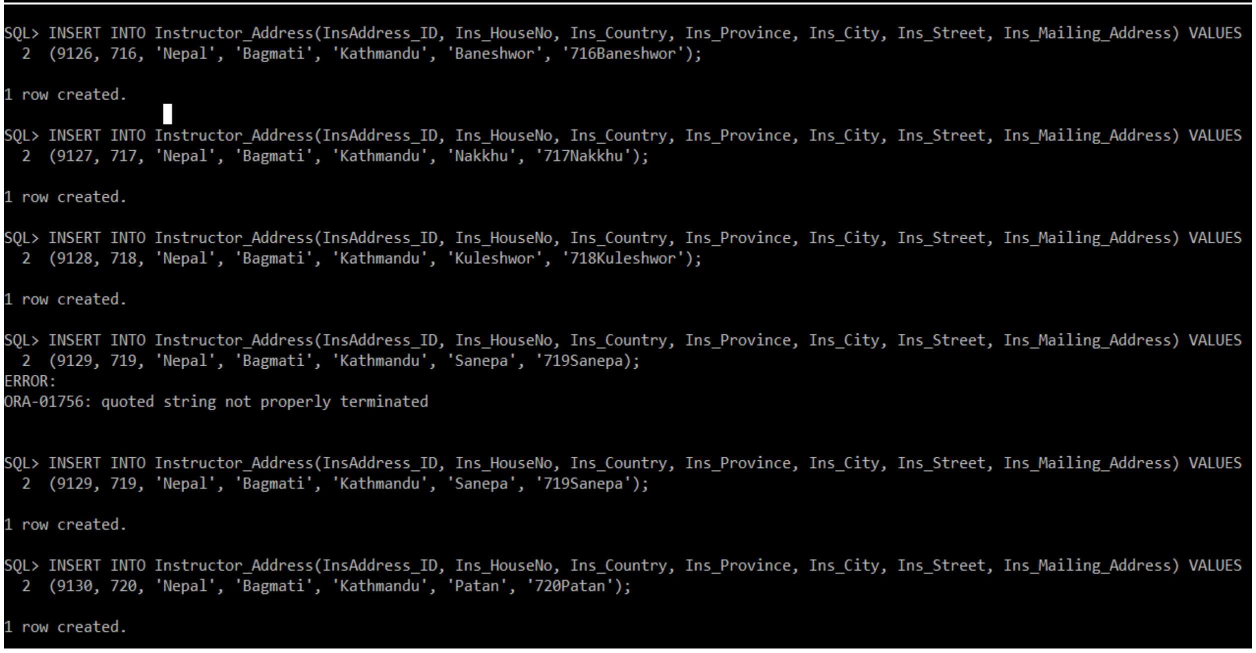
INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES

(9132, 722, 'Nepal', 'Bagmati', 'Kathmandu', 'Bhatbhateni', '722Bhatbhateni');

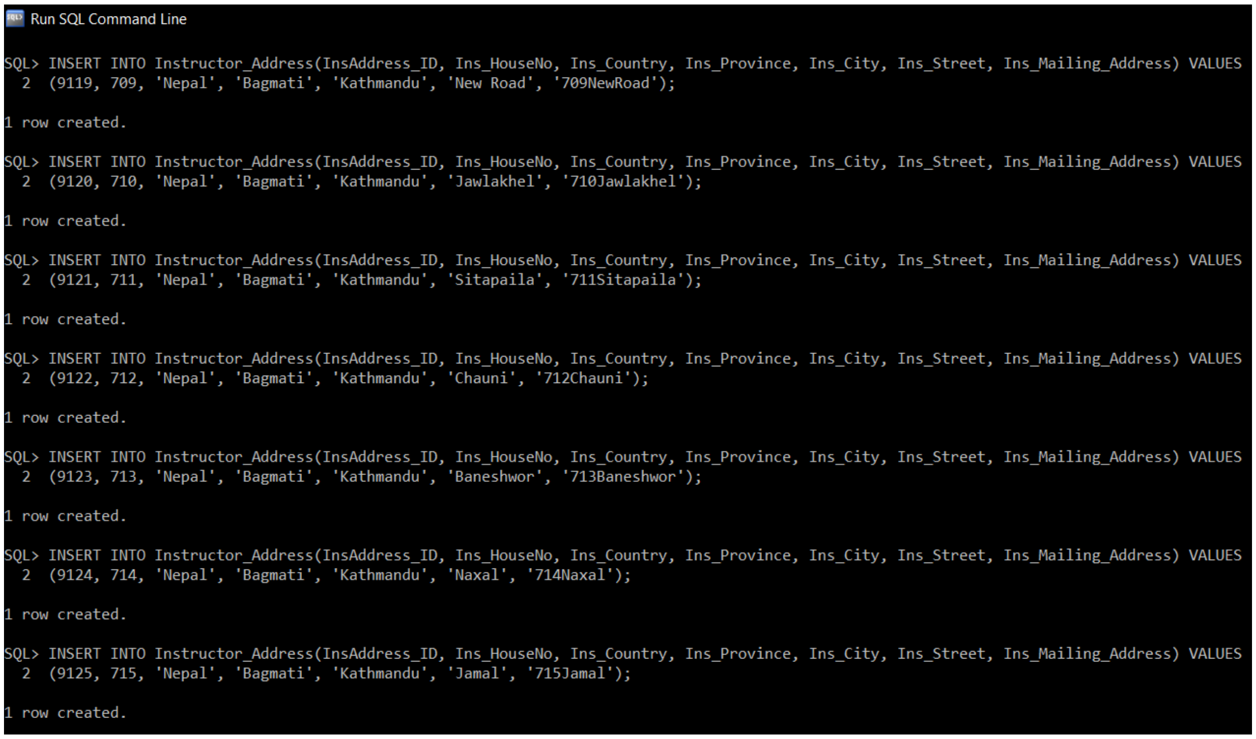


Run SQL Command Line

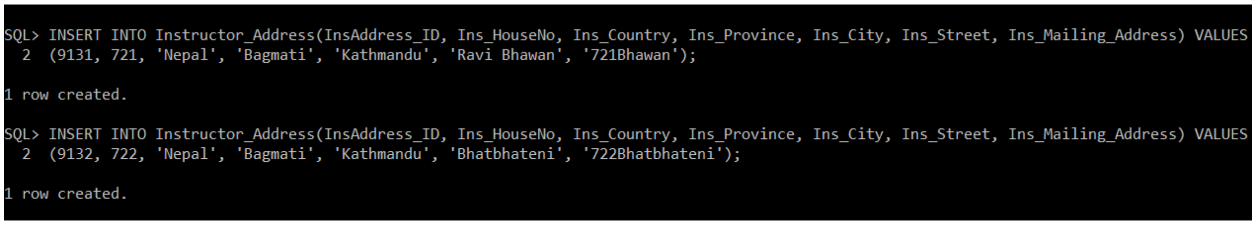
```
SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9111, 701, 'Nepal', 'Bagmati', 'Kathmandu', 'Bafal', '701Bafal');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9112, 702, 'Nepal', 'Bagmati', 'Kathmandu', 'Tachal', '702Tachal');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9113, 703, 'Nepal', 'Bagmati', 'Kathmandu', 'Kalimati', '703Kalimati');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9114, 704, 'Nepal', 'Bagmati', 'Kathmandu', 'Salt Trading', '704SaltTrading');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9115, 705, 'Nepal', 'Bagmati', 'Kathmandu', 'Soaltee Mode', '705SoalteeMode');

1 row created.

SQL> (9116, 706, 'Nepal', 'Bagmati', 'Kathmandu', 'Kalanki', '706Kalanki');
(9116, 706, 'Nepal', 'Bagmati', 'Kathmandu', 'Kalanki', '706Kalanki')
 *
ERROR at line 1:
ORA-00928: missing SELECT keyword


SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9116, 706, 'Nepal', 'Bagmati', 'Kathmandu', 'Kalanki', '706Kalanki');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9117, 707, 'Nepal', 'Bagmati', 'Kathmandu', 'Balkhu', '707Balkhu');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9118, 708, 'Nepal', 'Bagmati', 'Kathmandu', 'Chobhar', '708Chobhar');

1 row created.
```



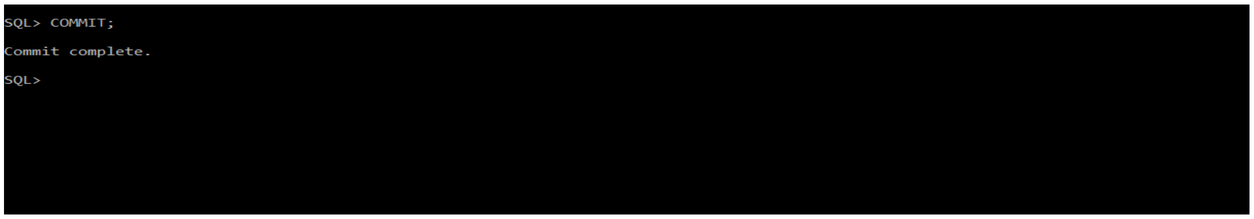
```
SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9126, 716, 'Nepal', 'Bagmati', 'Kathmandu', 'Baneshwor', '716Baneshwor');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9127, 717, 'Nepal', 'Bagmati', 'Kathmandu', 'Nakkhu', '717Nakkhu');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9128, 718, 'Nepal', 'Bagmati', 'Kathmandu', 'Kuleshwor', '718Kuleshwor');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9129, 719, 'Nepal', 'Bagmati', 'Kathmandu', 'Sanepa', '719Sanepa);
ERROR:
ORA-01756: quoted string not properly terminated


SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9129, 719, 'Nepal', 'Bagmati', 'Kathmandu', 'Sanepa', '719Sanepa');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9130, 720, 'Nepal', 'Bagmati', 'Kathmandu', 'Patan', '720Patan');

1 row created.
```



```
Run SQL Command Line

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9119, 709, 'Nepal', 'Bagmati', 'Kathmandu', 'New Road', '709NewRoad');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9120, 710, 'Nepal', 'Bagmati', 'Kathmandu', 'Jawlakhel', '710Jawlakhel');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9121, 711, 'Nepal', 'Bagmati', 'Kathmandu', 'Sitapaila', '711Sitapaila');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9122, 712, 'Nepal', 'Bagmati', 'Kathmandu', 'Chauni', '712Chauni');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9123, 713, 'Nepal', 'Bagmati', 'Kathmandu', 'Baneshwor', '713Baneshwor');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9124, 714, 'Nepal', 'Bagmati', 'Kathmandu', 'Naxal', '714Naxal');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9125, 715, 'Nepal', 'Bagmati', 'Kathmandu', 'Jamal', '715Jamal');

1 row created.
```



```
SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9131, 721, 'Nepal', 'Bagmati', 'Kathmandu', 'Ravi Bhawan', '721Bhawan');

1 row created.

SQL> INSERT INTO Instructor_Address(InsAddress_ID, Ins_HouseNo, Ins_Country, Ins_Province, Ins_City, Ins_Street, Ins_Mailing_Address) VALUES
  2  (9132, 722, 'Nepal', 'Bagmati', 'Kathmandu', 'Bhatbhateni', '722Bhatbhateni');

1 row created.
```



```
SQL> COMMIT;
Commit complete.
SQL>
```

Figure 24: Inserting values in Instructor_Address table

Inserting values in Student_Address table

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

(8111, 801, 'Nepal', 'Bagmati', 'Kathmandu', 'Bafal', '801Bafal');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

(8112, 802, 'Nepal', 'Bagmati', 'Kathmandu', 'Tachal', '802Tachal');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

(8113, 803, 'Nepal', 'Bagmati', 'Kathmandu', 'Kalimati', '803Kalimati');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

(8114, 804, 'Nepal', 'Bagmati', 'Kathmandu', 'Salt Trading', '804SaltTrading');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

(8115, 805, 'Nepal', 'Bagmati', 'Kathmandu', 'Soaltee Mode', '805SoalteeMode');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

```
(8116, 806, 'Nepal', 'Bagmati', 'Kathmandu', 'Kalanki', '806Kalanki');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province,
Std_City, Std_Street, Std_Mailing_Address) VALUES
(8117, 807, 'Nepal', 'Bagmati', 'Kathmandu', 'Balkhu', '807Balkhu');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province,
Std_City, Std_Street, Std_Mailing_Address) VALUES
(8118, 808, 'Nepal', 'Bagmati', 'Kathmandu', 'Chobhar', '808Chobhar');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province,
Std_City, Std_Street, Std_Mailing_Address) VALUES
(8119, 809, 'Nepal', 'Bagmati', 'Kathmandu', 'New Road', '809NewRoad');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province,
Std_City, Std_Street, Std_Mailing_Address) VALUES
(8120, 810, 'Nepal', 'Bagmati', 'Kathmandu', 'Jawlakhel', '810Jawlakhel');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province,
Std_City, Std_Street, Std_Mailing_Address) VALUES
(8121, 811, 'Nepal', 'Bagmati', 'Kathmandu', 'Sitapaila', '811Sitapaila');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province,
Std_City, Std_Street, Std_Mailing_Address) VALUES
```

(8122, 812, 'Nepal', 'Bagmati', 'Kathmandu', 'Chauni', '812Chauni');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

(8123, 813, 'Nepal', 'Bagmati', 'Kathmandu', 'Baneshwor', '813Baneshwor');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

(8124, 814, 'Nepal', 'Bagmati', 'Kathmandu', 'Naxal', '814Naxal');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

(8125, 815, 'Nepal', 'Bagmati', 'Kathmandu', 'Jamal', '815Jamal');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

(8126, 816, 'Nepal', 'Bagmati', 'Kathmandu', 'Ason', '816Ason');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

(8127, 817, 'Nepal', 'Bagmati', 'Kathmandu', 'Jyatha', '817Jyatha');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

(8128, 818, 'Nepal', 'Bagmati', 'Kathmandu', 'Teku', '818Teku');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

(8129, 819, 'Nepal', 'Bagmati', 'Kathmandu', 'Patan', '819Patan');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

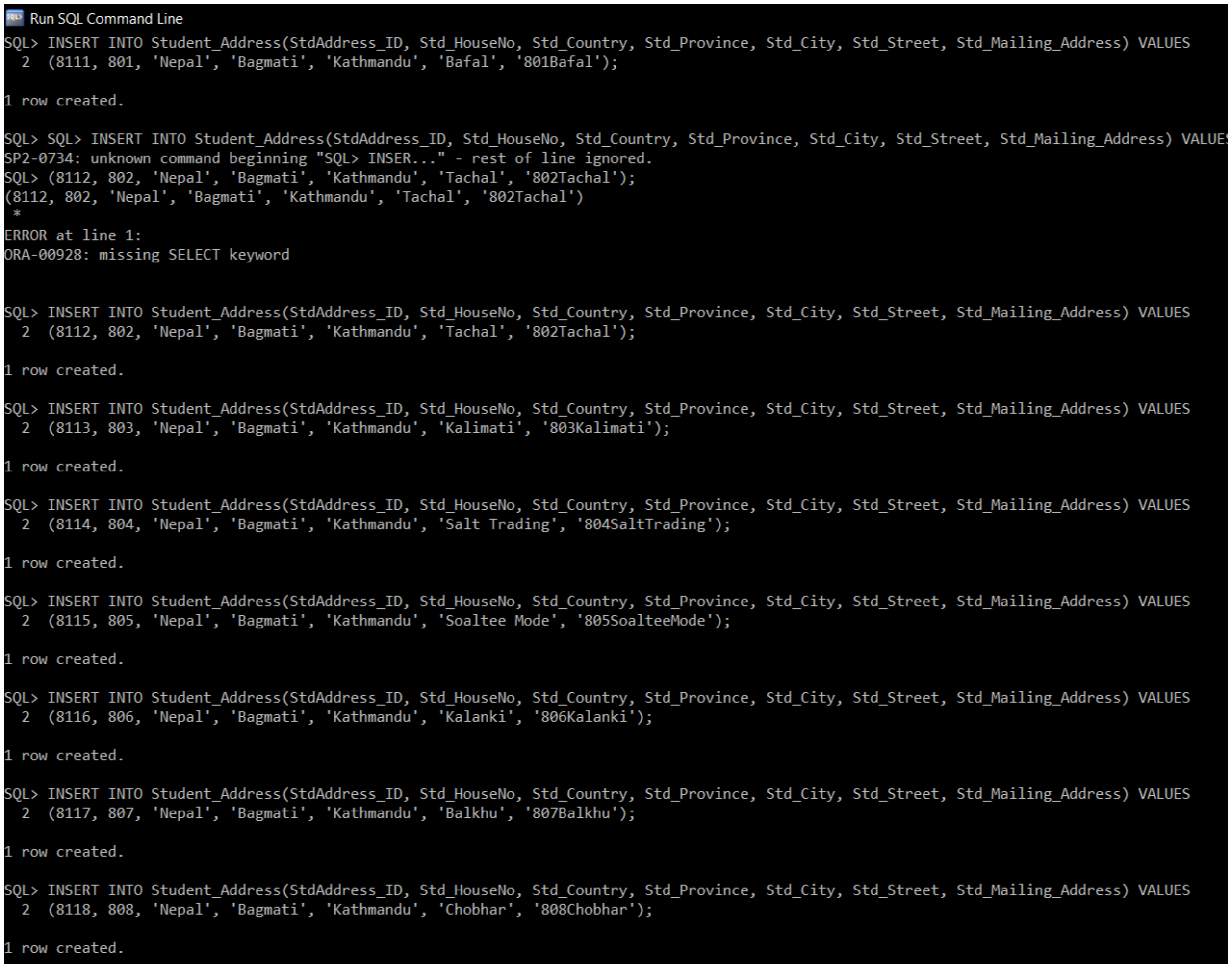
(8130, 820, 'Nepal', 'Bagmati', 'Kathmandu', 'Ravi Bhawan', '820Bhawan');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

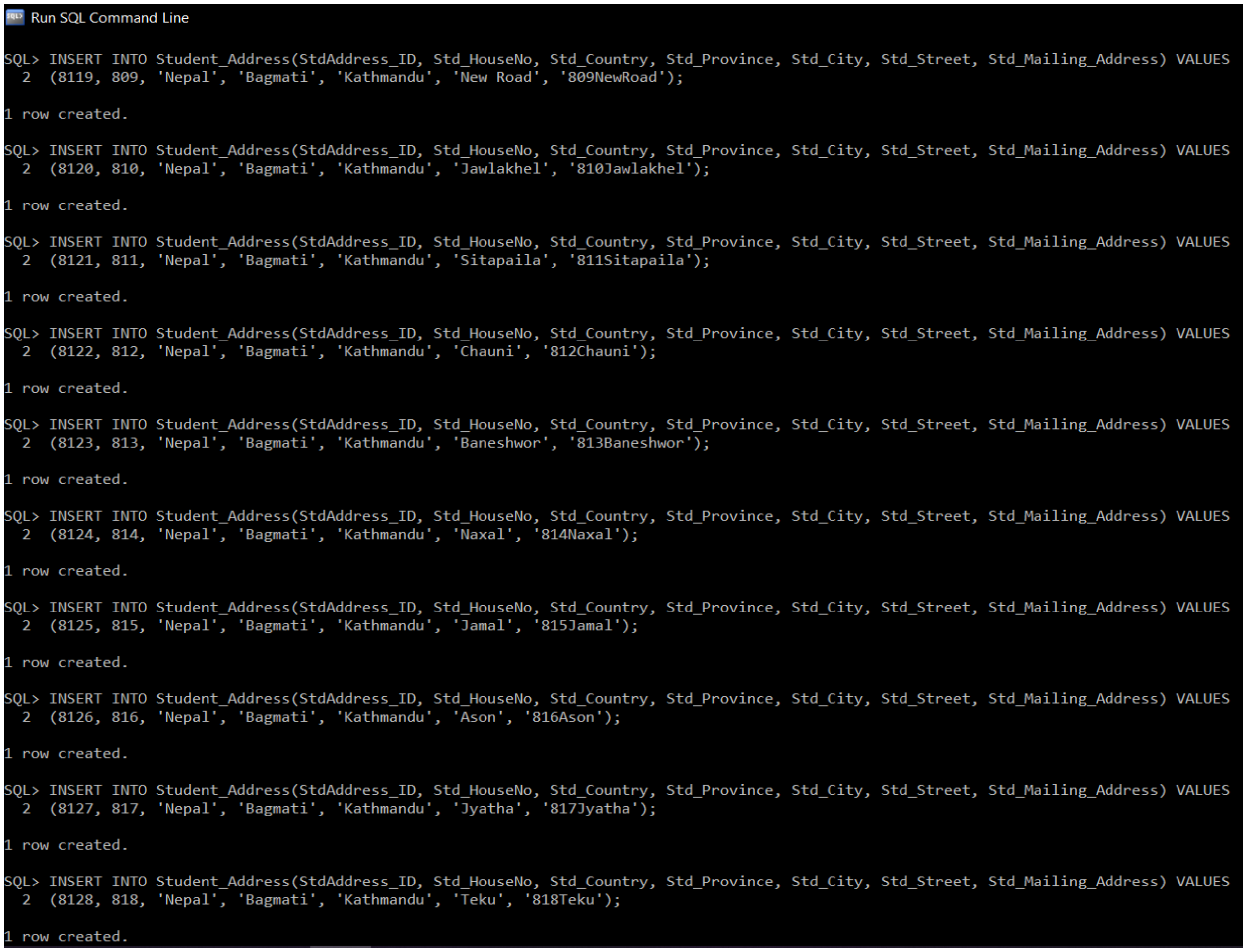
(8131, 821, 'Nepal', 'Bagmati', 'Kathmandu', 'Balkhu', '821Balkhu');

INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES

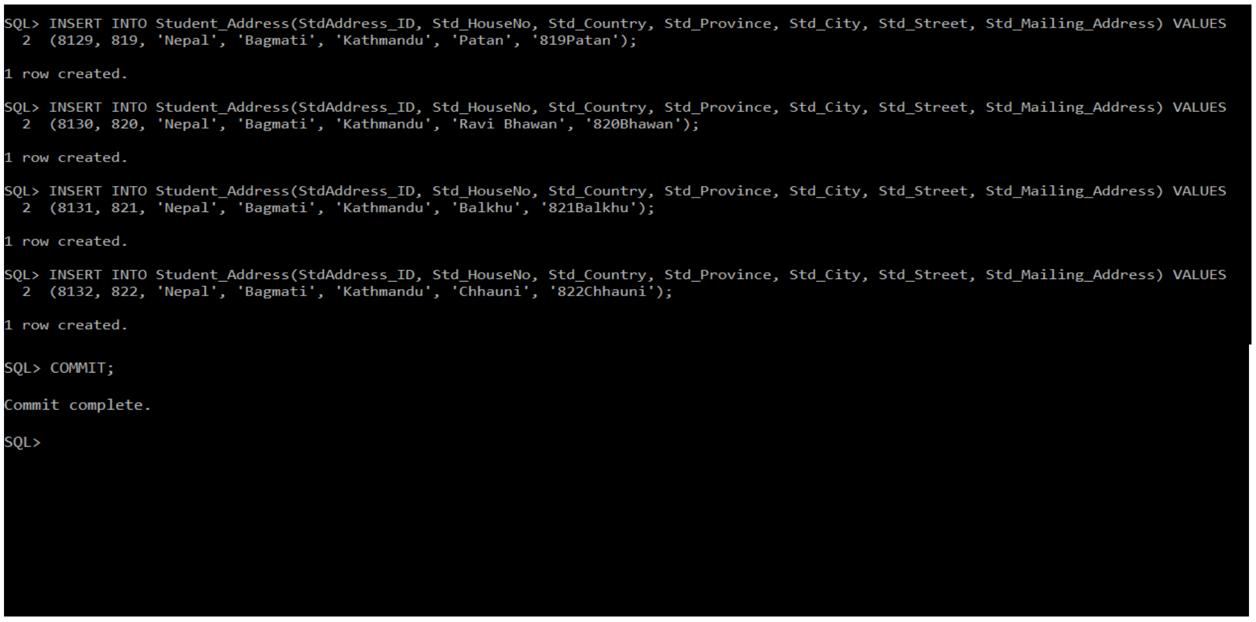
(8132, 822, 'Nepal', 'Bagmati', 'Kathmandu', 'Chhauni', '822Chhauni');



```
Run SQL Command Line
SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8111, 801, 'Nepal', 'Bagmati', 'Kathmandu', 'Bafal', '801Bafal');

1 row created.

SQL> SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUE
SP2-0734: unknown command beginning "SQL> INSER..." - rest of line ignored.
SQL> (8112, 802, 'Nepal', 'Bagmati', 'Kathmandu', 'Tachal', '802Tachal');
(8112, 802, 'Nepal', 'Bagmati', 'Kathmandu', 'Tachal', '802Tachal')
 *
ERROR at line 1:
ORA-00928: missing SELECT keyword


SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8112, 802, 'Nepal', 'Bagmati', 'Kathmandu', 'Tachal', '802Tachal');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8113, 803, 'Nepal', 'Bagmati', 'Kathmandu', 'Kalimati', '803Kalimati');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8114, 804, 'Nepal', 'Bagmati', 'Kathmandu', 'Salt Trading', '804SaltTrading');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8115, 805, 'Nepal', 'Bagmati', 'Kathmandu', 'Soaltee Mode', '805SoalteeMode');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8116, 806, 'Nepal', 'Bagmati', 'Kathmandu', 'Kalanki', '806Kalanki');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8117, 807, 'Nepal', 'Bagmati', 'Kathmandu', 'Balkhu', '807Balkhu');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8118, 808, 'Nepal', 'Bagmati', 'Kathmandu', 'Chobhar', '808Chobhar');

1 row created.
```



```
Run SQL Command Line

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8119, 809, 'Nepal', 'Bagmati', 'Kathmandu', 'New Road', '809NewRoad');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8120, 810, 'Nepal', 'Bagmati', 'Kathmandu', 'Jawlakhel', '810Jawlakhel');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8121, 811, 'Nepal', 'Bagmati', 'Kathmandu', 'Sitapaila', '811Sitapaila');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8122, 812, 'Nepal', 'Bagmati', 'Kathmandu', 'Chauni', '812Chauni');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8123, 813, 'Nepal', 'Bagmati', 'Kathmandu', 'Baneshwor', '813Baneshwor');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8124, 814, 'Nepal', 'Bagmati', 'Kathmandu', 'Naxal', '814Naxal');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8125, 815, 'Nepal', 'Bagmati', 'Kathmandu', 'Jamal', '815Jamal');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8126, 816, 'Nepal', 'Bagmati', 'Kathmandu', 'Ason', '816Ason');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8127, 817, 'Nepal', 'Bagmati', 'Kathmandu', 'Jyatha', '817Jyatha');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8128, 818, 'Nepal', 'Bagmati', 'Kathmandu', 'Teku', '818Teku');

1 row created.
```



```
SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8129, 819, 'Nepal', 'Bagmati', 'Kathmandu', 'Patan', '819Patan');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8130, 820, 'Nepal', 'Bagmati', 'Kathmandu', 'Ravi Bhawan', '820Bhawan');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8131, 821, 'Nepal', 'Bagmati', 'Kathmandu', 'Balkhu', '821Balkhu');

1 row created.

SQL> INSERT INTO Student_Address(StdAddress_ID, Std_HouseNo, Std_Country, Std_Province, Std_City, Std_Street, Std_Mailing_Address) VALUES
  2  (8132, 822, 'Nepal', 'Bagmati', 'Kathmandu', 'Chhauni', '822Chhauni');

1 row created.

SQL> COMMIT;

Commit complete.

SQL>
```

Figure 25: Insert in Student_Address table

Inserting values in Module_Info table

```
INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
(201, 111);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
(202, 111);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
(202, 112);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
(203, 111);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
(203, 113);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
(204, 112);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
(205, 114);
```

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES

(205, 115);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES

(206, 117);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES

(207, 113);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES

(211, 116);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES

(208, 118);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES

(209, 119);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES

(210, 120);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES

(201, 112);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES

(203, 112);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES

(212, 112);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES

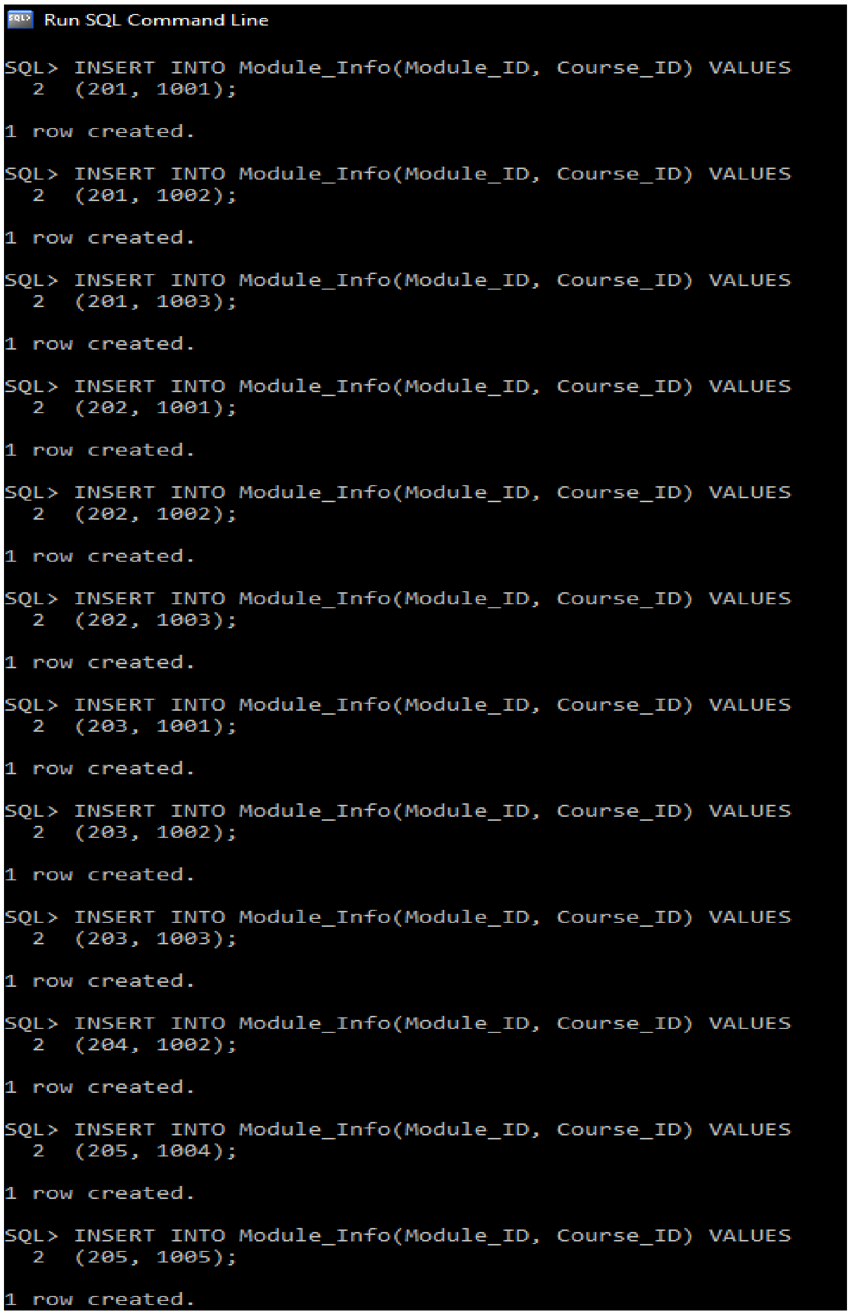
(213, 112);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES

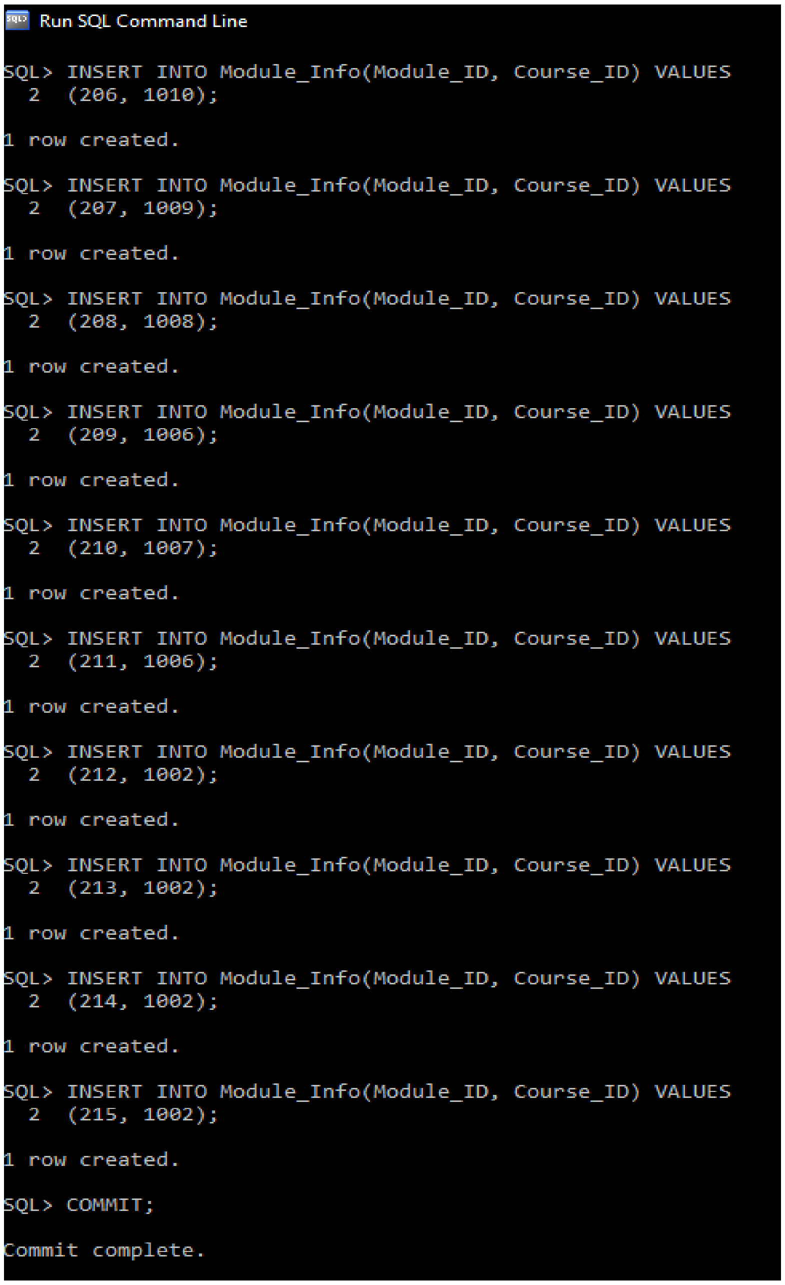
(214, 112);

INSERT INTO Module_Info(Module_ID, Course_ID) VALUES

(215, 112);



```
SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (201, 1001);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (201, 1002);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (201, 1003);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (202, 1001);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (202, 1002);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (202, 1003);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (203, 1001);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (203, 1002);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (203, 1003);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (204, 1002);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (205, 1004);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (205, 1005);

1 row created.
```



```
SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (206, 1010);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (207, 1009);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (208, 1008);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (209, 1006);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (210, 1007);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (211, 1006);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (212, 1002);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (213, 1002);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (214, 1002);

1 row created.

SQL> INSERT INTO Module_Info(Module_ID, Course_ID) VALUES
  2  (215, 1002);

1 row created.

SQL> COMMIT;

Commit complete.
```

Figure 26: Inserting values in Module_Info table

Inserting values in Instructor_Info table

```
INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(51, 1001);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(52, 1004);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(53, 1006);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(52, 1005);
INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(54, 1007);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(51, 1002);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(55, 1008);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
```

```
(56, 1009);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(51, 1003);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(57, 1010);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(58, 1001);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(59, 1002);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(60, 1003);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(61, 1004);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(62, 1005);
```

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES

(63, 1006);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES

(64, 1007);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES

(65, 1008);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES

(66, 1009);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES

(67, 1010);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES

(68, 1001);

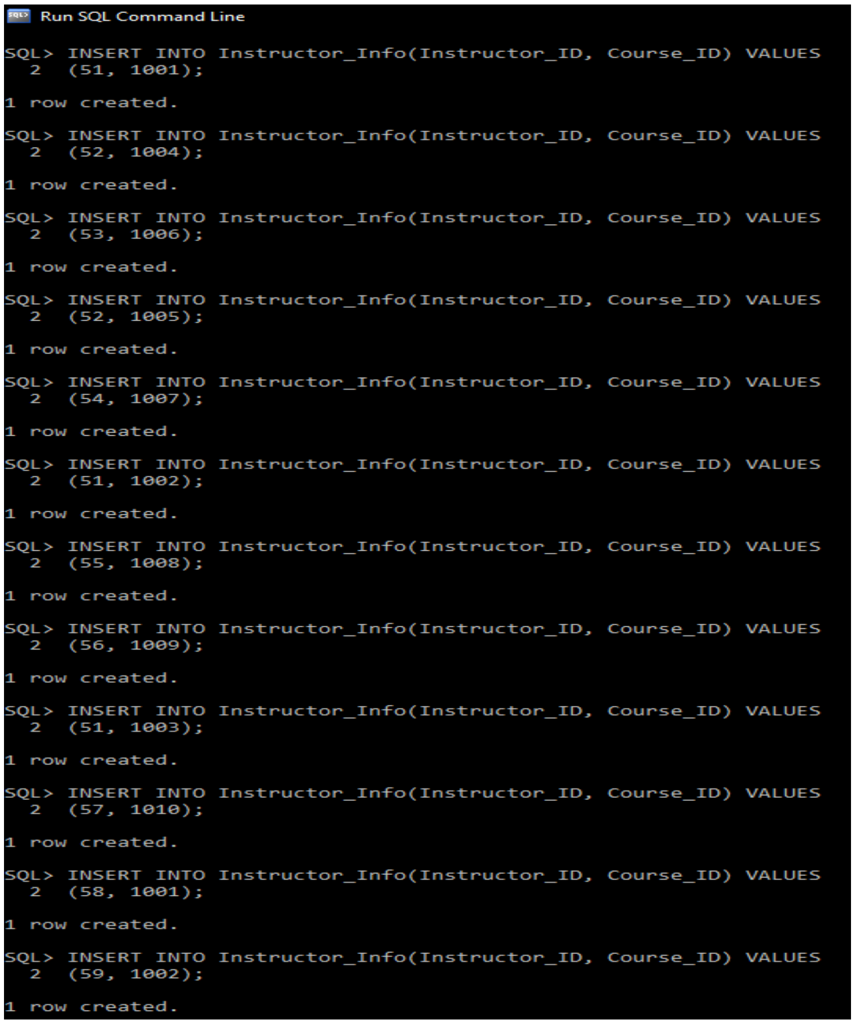
INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES

(69, 1002);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES

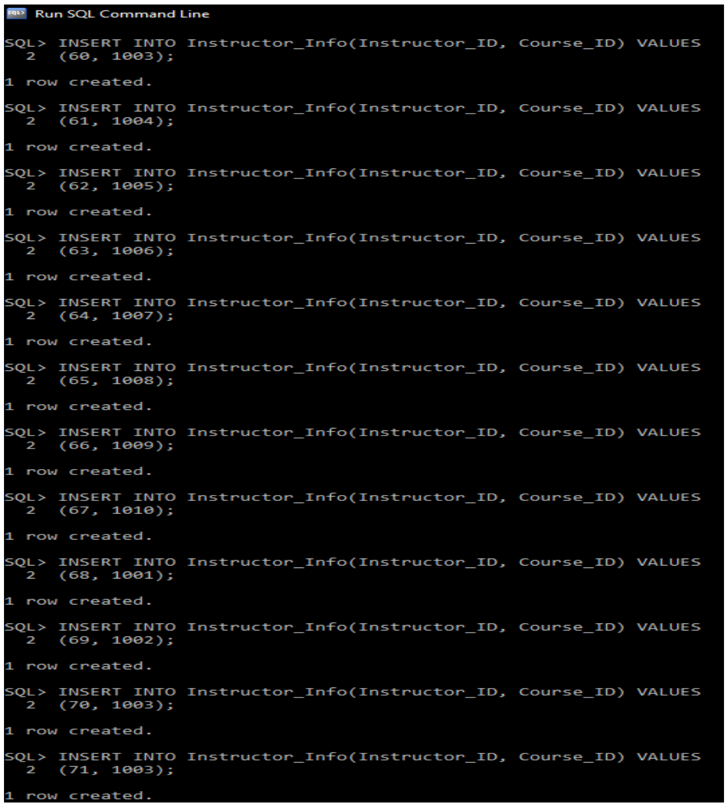
(70, 1003);

```
INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(71, 1003);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(68, 1002);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(70, 1002);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(72, 1002);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(59, 1001);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(59, 1003);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(58, 1002);

INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
(58, 1003);
```

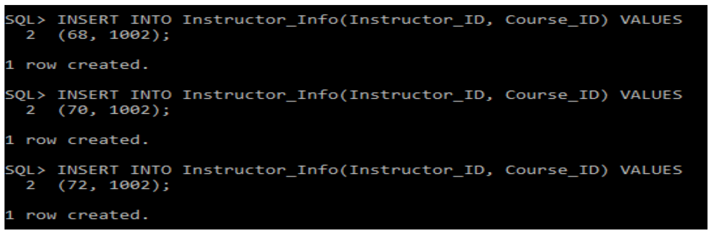


Run SQL Command Line

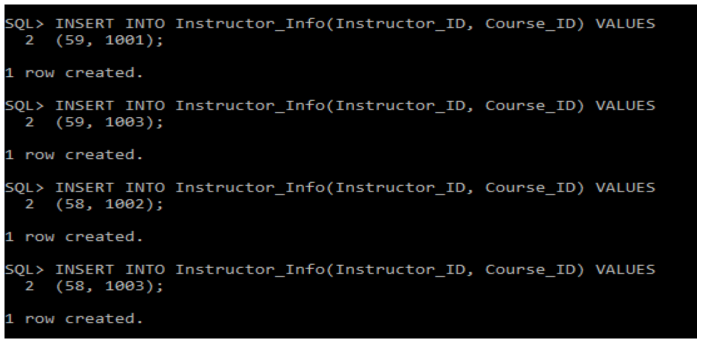
```
SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (51, 1001);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (52, 1004);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (53, 1006);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (52, 1005);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (54, 1007);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (51, 1002);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (55, 1008);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (56, 1009);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (51, 1003);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (57, 1010);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (58, 1001);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (59, 1002);

1 row created.
```



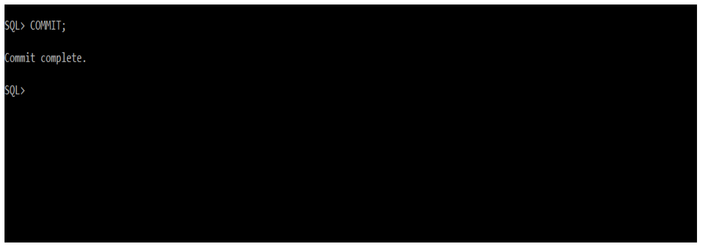
```
Run SQL Command Line

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (60, 1003);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (61, 1004);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (62, 1005);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (63, 1006);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (64, 1007);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (65, 1008);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (66, 1009);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (67, 1010);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (68, 1001);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (69, 1002);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (70, 1003);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (71, 1003);

1 row created.
```



```
SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (68, 1002);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (70, 1002);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (72, 1002);

1 row created.
```



```
SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (59, 1001);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (59, 1003);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (58, 1002);

1 row created.

SQL> INSERT INTO Instructor_Info(Instructor_ID, Course_ID) VALUES
  2  (58, 1003);

1 row created.
```



```
SQL> COMMIT;

Commit complete.

SQL>
```

Figure 27: Inserting value in Instructor_Info

Inserting values in Student_Info table

```
INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
(1, 1001);

INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
(2, 1001);

INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
(3, 1001);

INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
(4, 1001);

INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
(5, 1002);

INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
(6, 1002);

INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
(7, 1002);
```

```
INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
(8, 1002);


INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
(9, 1003);
INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
(10, 1003);


INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
(11, 1003);


INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
(12, 1003);


INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
(13, 1003);


INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
(14, 1003);


INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
(15, 1003);
```

INSERT INTO Student_Info(Student_ID, Course_ID) VALUES

(16, 1004);

INSERT INTO Student_Info(Student_ID, Course_ID) VALUES

(17, 1005);

INSERT INTO Student_Info(Student_ID, Course_ID) VALUES

(18, 1006);

INSERT INTO Student_Info(Student_ID, Course_ID) VALUES

(19, 1007);

INSERT INTO Student_Info(Student_ID, Course_ID) VALUES

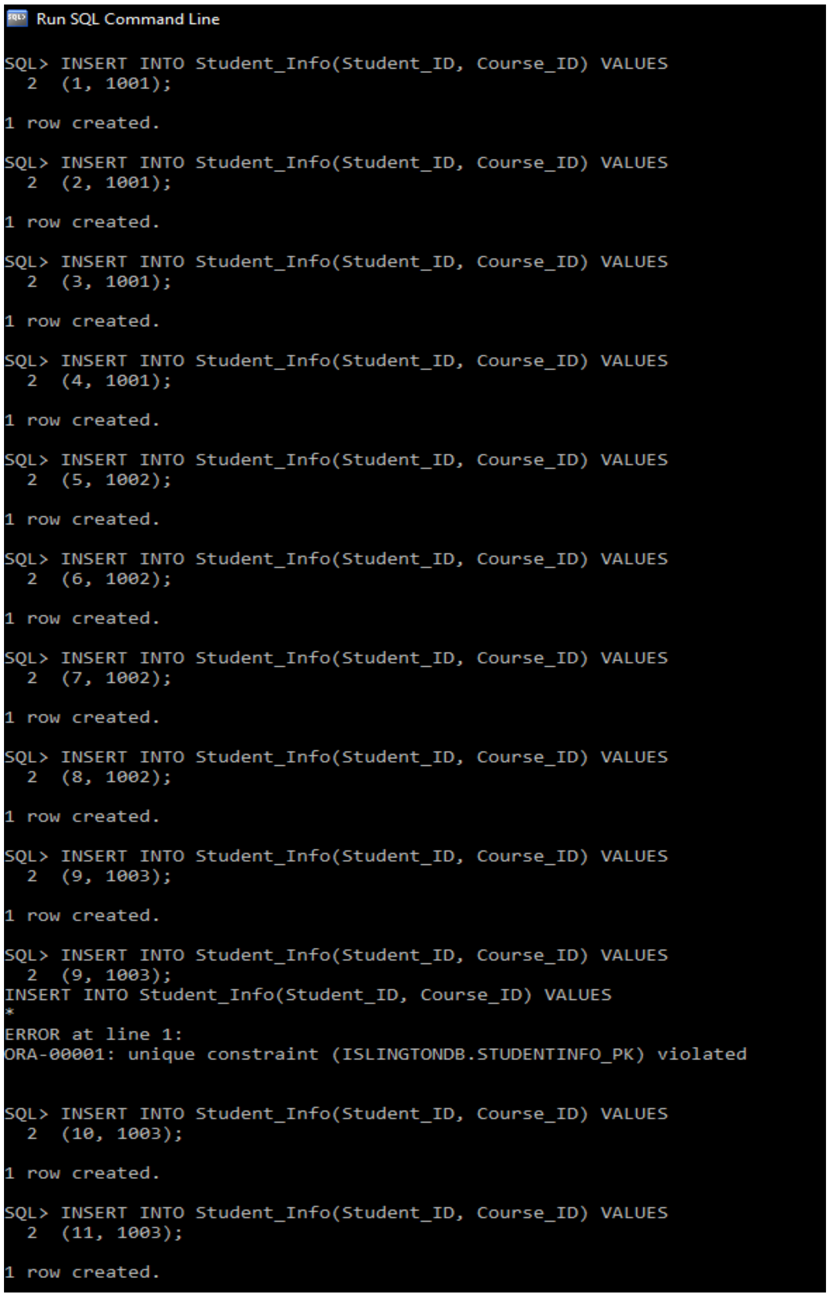
(20, 1008);

INSERT INTO Student_Info(Student_ID, Course_ID) VALUES

(21, 1009);

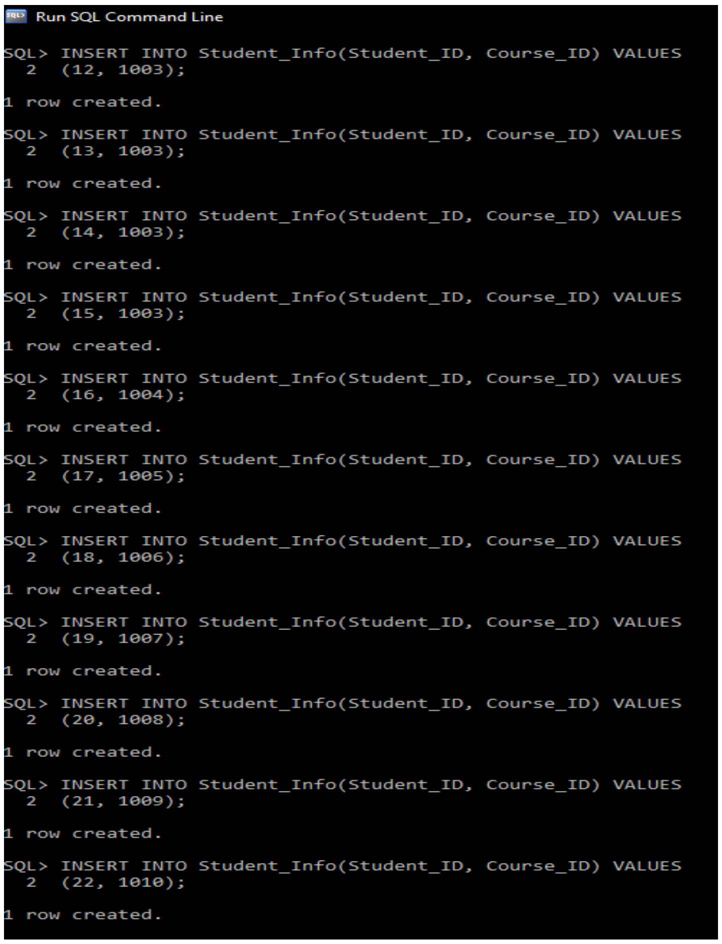
INSERT INTO Student_Info(Student_ID, Course_ID) VALUES

(22, 1010);

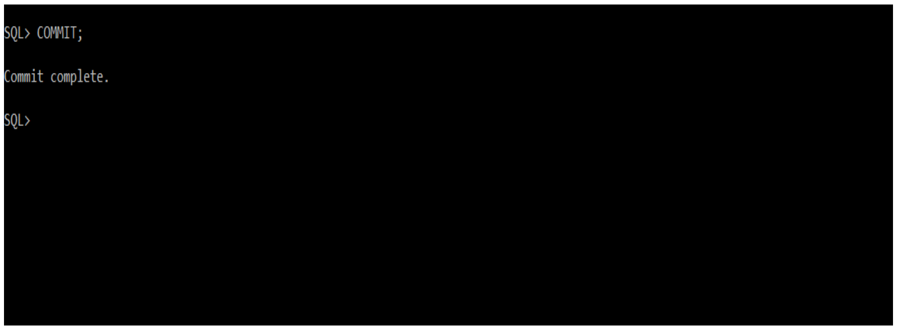


Run SQL Command Line

```
SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (1, 1001);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (2, 1001);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (3, 1001);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (4, 1001);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (5, 1002);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (6, 1002);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (7, 1002);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (8, 1002);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (9, 1003);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (9, 1003);
INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
*
ERROR at line 1:
ORA-00001: unique constraint (ISLINGTONDB.STUDENTINFO_PK) violated


SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (10, 1003);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (11, 1003);

1 row created.
```



```
Run SQL Command Line

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (12, 1003);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (13, 1003);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (14, 1003);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (15, 1003);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (16, 1004);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (17, 1005);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (18, 1006);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (19, 1007);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (20, 1008);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (21, 1009);

1 row created.

SQL> INSERT INTO Student_Info(Student_ID, Course_ID) VALUES
  2  (22, 1010);

1 row created.
```



```
SQL> COMMIT;

Commit complete.

SQL>
```

Figure 28: Inserting value in Student_Info table

Inserting values in Instructor_Address_Info table

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9111, 51);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9112, 52);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9113, 53);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9114, 54);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9115, 55);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9115, 56);

DELETE FROM Instructor_Address_Info WHERE Instructor_ID=56;

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9116, 56);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9117, 57);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9118, 58);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9119, 59);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9120, 60);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9121, 61);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9122, 62);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9123, 63);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9124, 64);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9125, 65);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9126, 66);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9127, 67);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9128, 68);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9129, 69);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

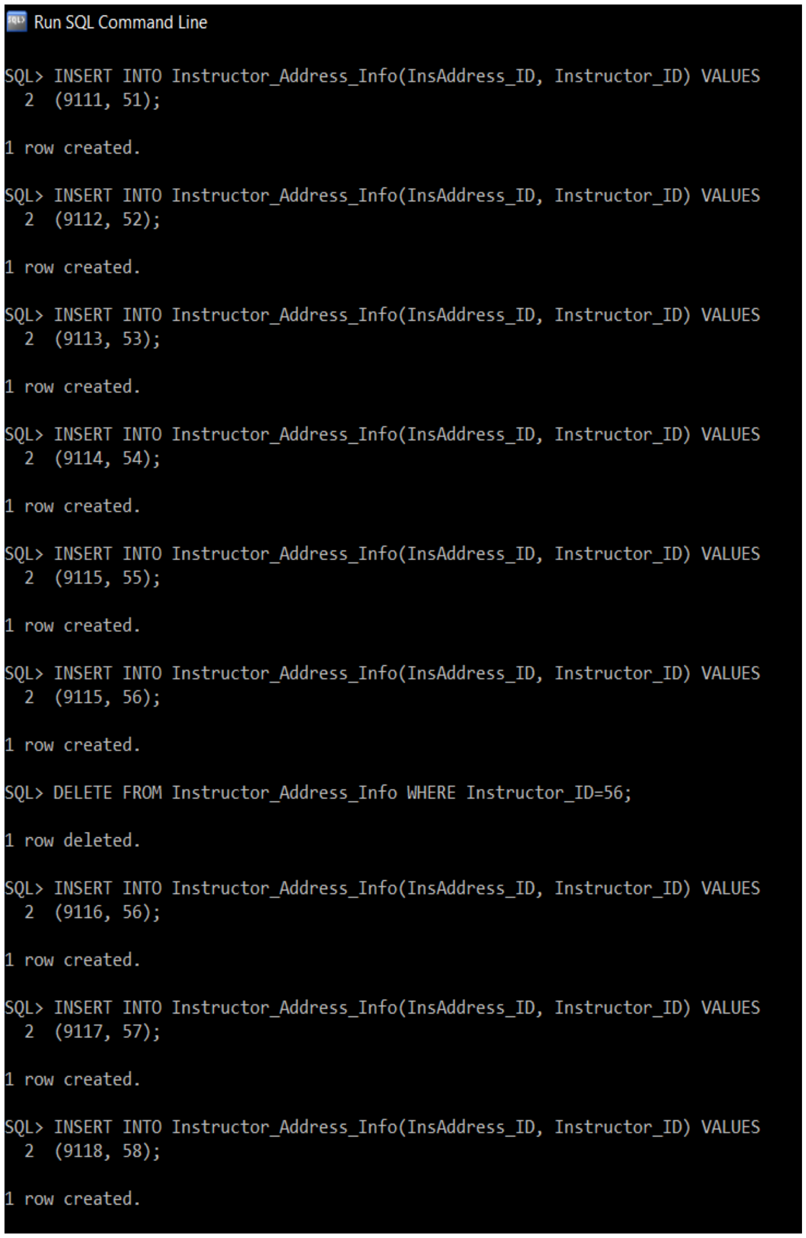
(9130, 70);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

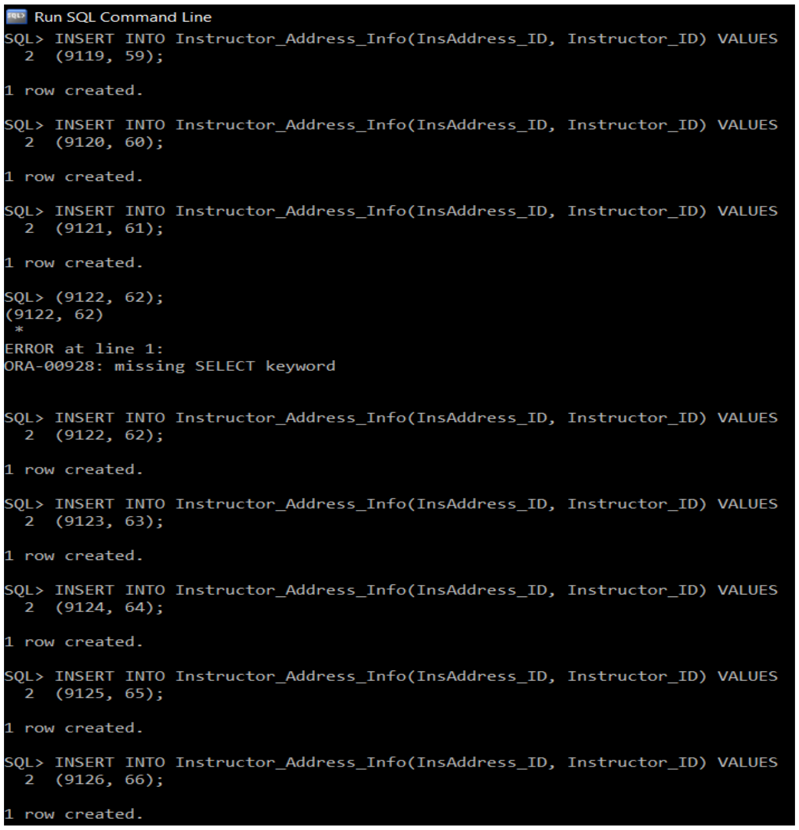
(9131, 71);

INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES

(9132, 72);

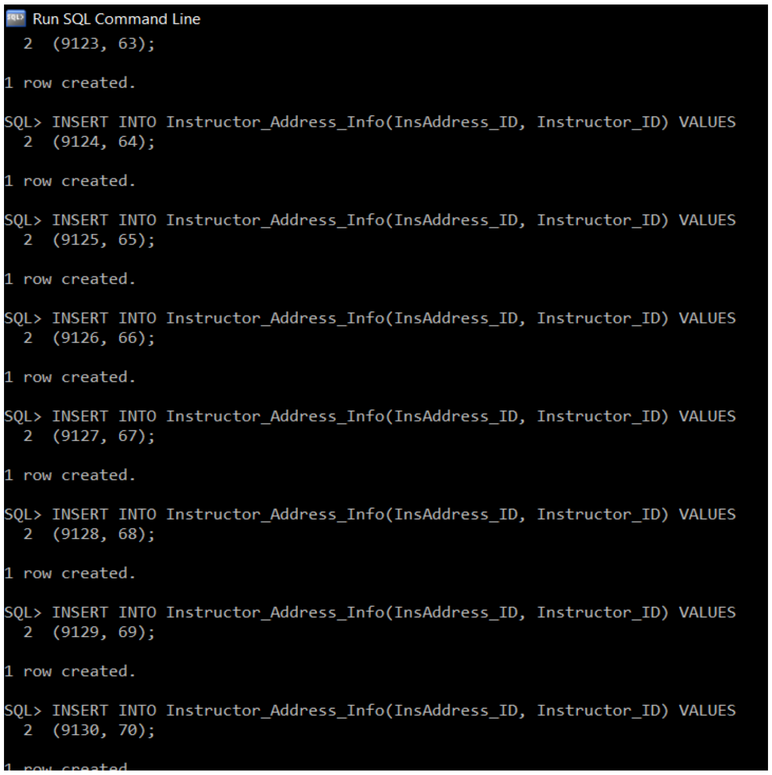


```
Run SQL Command Line

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9111, 51);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9112, 52);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9113, 53);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9114, 54);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9115, 55);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9115, 56);

1 row created.

SQL> DELETE FROM Instructor_Address_Info WHERE Instructor_ID=56;

1 row deleted.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9116, 56);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9117, 57);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9118, 58);

1 row created.
```



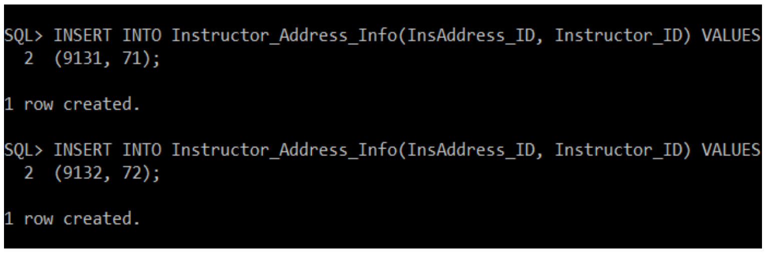
Run SQL Command Line

```
SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9119, 59);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9120, 60);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9121, 61);

1 row created.

SQL> (9122, 62);
(9122, 62)
 *
ERROR at line 1:
ORA-00928: missing SELECT keyword


SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9122, 62);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9123, 63);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9124, 64);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9125, 65);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9126, 66);

1 row created.
```

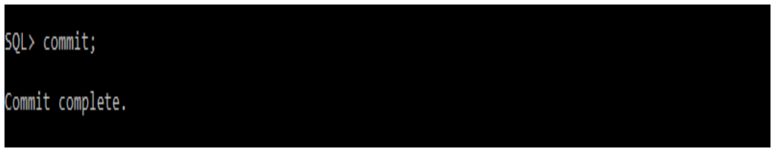


Run SQL Command Line

```
  2  (9123, 63);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9124, 64);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9125, 65);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9126, 66);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9127, 67);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9128, 68);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9129, 69);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9130, 70);

1 row created.
```



```
SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9131, 71);

1 row created.

SQL> INSERT INTO Instructor_Address_Info(InsAddress_ID, Instructor_ID) VALUES
  2  (9132, 72);

1 row created.
```



```
SQL> commit;

Commit complete.
```

Figure 29: Inserting value in Instructor_Address_Info table

Inserting values in Student_Address_Info table

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES

(8111, 1);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES

(8112, 2);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES

(8113, 3);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES

(8114, 4);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES

(8115, 5);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES

(8116, 6);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES

(8117, 7);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES

```
(8118, 8);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
(8119, 9);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
(8120, 10);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
(8121, 11);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
(8122, 12);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
(8123, 13);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
(8124, 14);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
(8125, 15);
```

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES

(8126, 16);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES

(8127, 17);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES

(8128, 18);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES

(8129, 19);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES

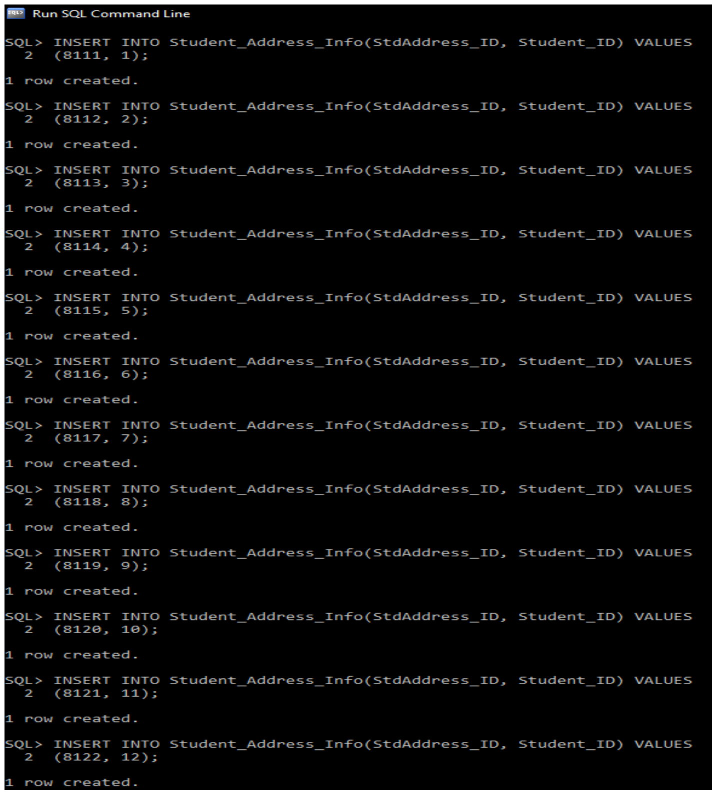
(8130, 20);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES

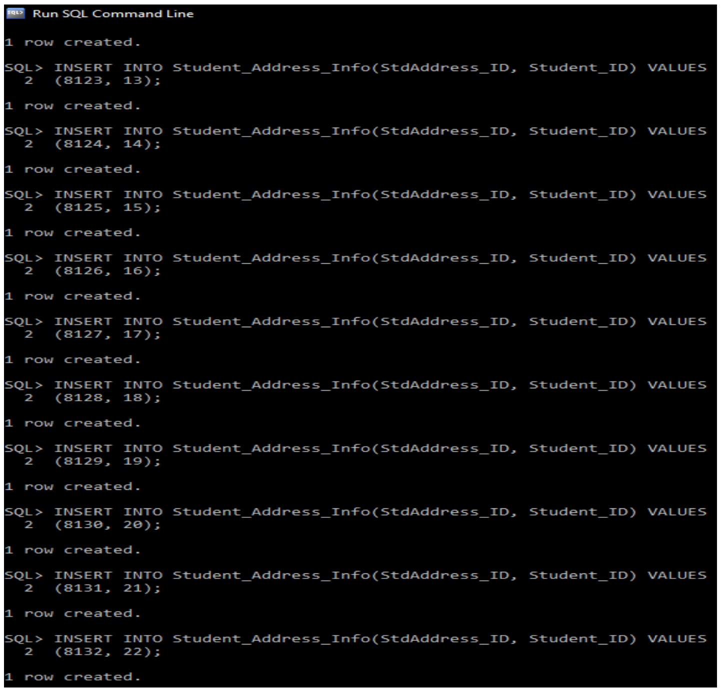
(8131, 21);

INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES

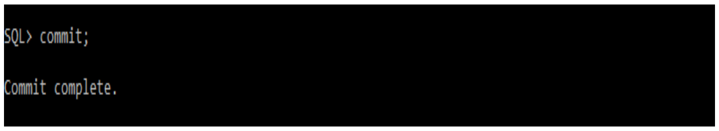
(8132, 22);



```
Run SQL Command Line

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8111, 1);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8112, 2);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8113, 3);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8114, 4);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8115, 5);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8116, 6);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8117, 7);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8118, 8);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8119, 9);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8120, 10);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8121, 11);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8122, 12);

1 row created.
```



```
Run SQL Command Line

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8123, 13);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8124, 14);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8125, 15);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8126, 16);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8127, 17);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8128, 18);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8129, 19);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8130, 20);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8131, 21);

1 row created.

SQL> INSERT INTO Student_Address_Info(StdAddress_ID, Student_ID) VALUES
  2  (8132, 22);

1 row created.
```



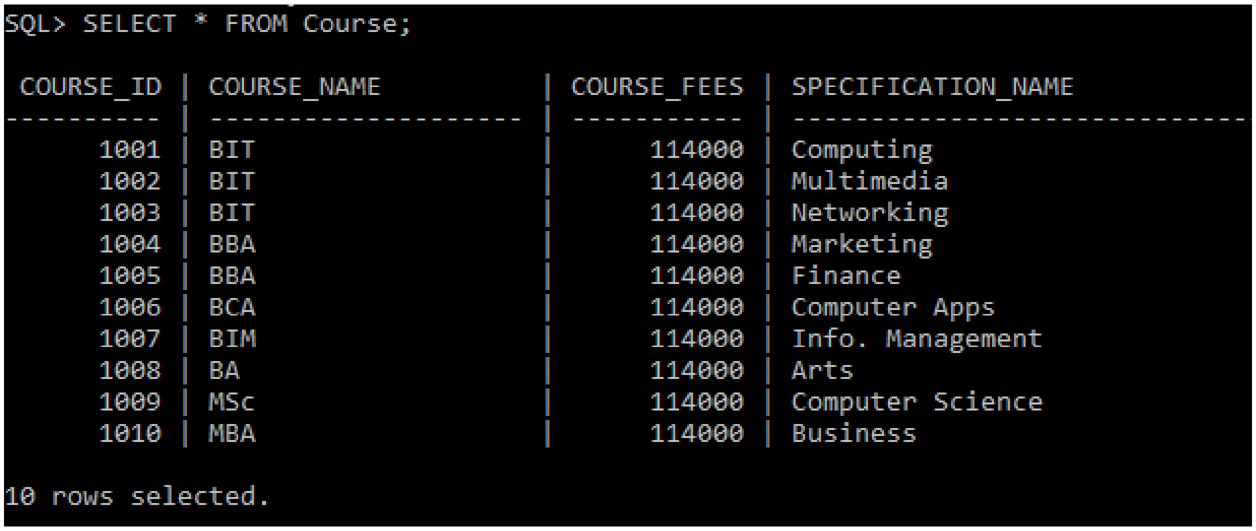
```
SQL> commit;

Commit complete.
```

Figure 30: Inserting values in Student_Address_Info table

3.3. Displaying the tables

To display the MySQL database tables, the SELECT command must be used.

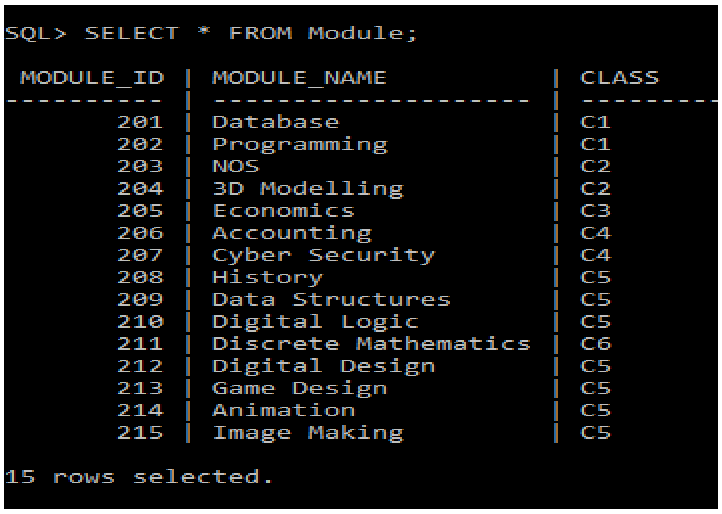
Course table



```
SQL> SELECT * FROM Course;

 COURSE_ID | COURSE_NAME          | COURSE_FEES | SPECIFICATION_NAME
---------- | -------------------- | ----------- | ------------------------------
      1001 | BIT                  |      114000 | Computing
      1002 | BIT                  |      114000 | Multimedia
      1003 | BIT                  |      114000 | Networking
      1004 | BBA                  |      114000 | Marketing
      1005 | BBA                  |      114000 | Finance
      1006 | BCA                  |      114000 | Computer Apps
      1007 | BIM                  |      114000 | Info. Management
      1008 | BA                   |      114000 | Arts
      1009 | MSc                  |      114000 | Computer Science
      1010 | MBA                  |      114000 | Business

10 rows selected.
```

Figure 31: Course Table

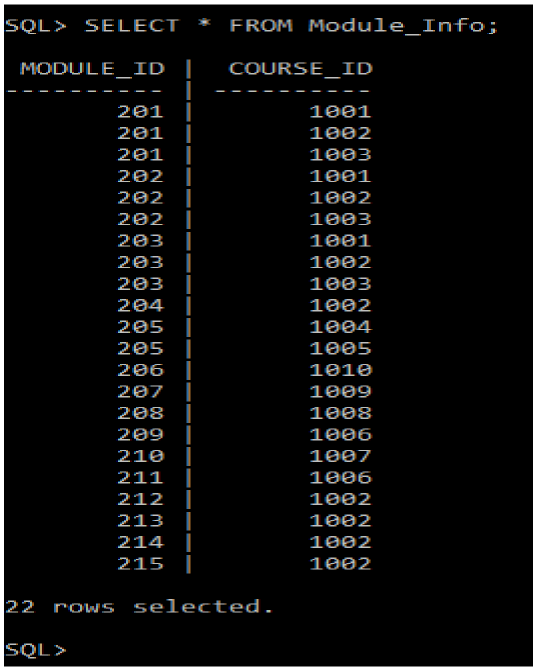
Module table



```
SQL> SELECT * FROM Module;

 MODULE_ID | MODULE_NAME           | CLASS
---------- | --------------------  | ---------
       201 | Database              | C1
       202 | Programming           | C1
       203 | NOS                   | C2
       204 | 3D Modelling          | C2
       205 | Economics             | C3
       206 | Accounting            | C4
       207 | Cyber Security        | C4
       208 | History               | C5
       209 | Data Structures       | C5
       210 | Digital Logic         | C5
       211 | Discrete Mathematics  | C6
       212 | Digital Design        | C5
       213 | Game Design           | C5
       214 | Animation             | C5
       215 | Image Making          | C5

15 rows selected.
```

Figure 32: Module table

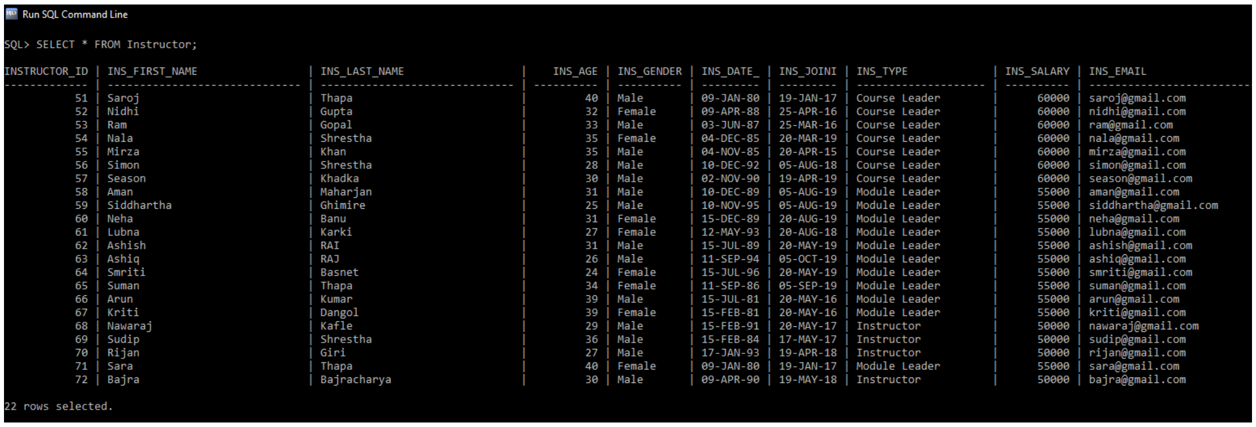
Module_Info table



```
SQL> SELECT * FROM Module_Info;

 MODULE_ID |  COURSE_ID
---------- | ----------
       201 |       1001
       201 |       1002
       201 |       1003
       202 |       1001
       202 |       1002
       202 |       1003
       203 |       1001
       203 |       1002
       203 |       1003
       204 |       1002
       205 |       1004
       205 |       1005
       206 |       1010
       207 |       1009
       208 |       1008
       209 |       1006
       210 |       1007
       211 |       1006
       212 |       1002
       213 |       1002
       214 |       1002
       215 |       1002

22 rows selected.

SQL>
```

Figure 33: Module_Info table

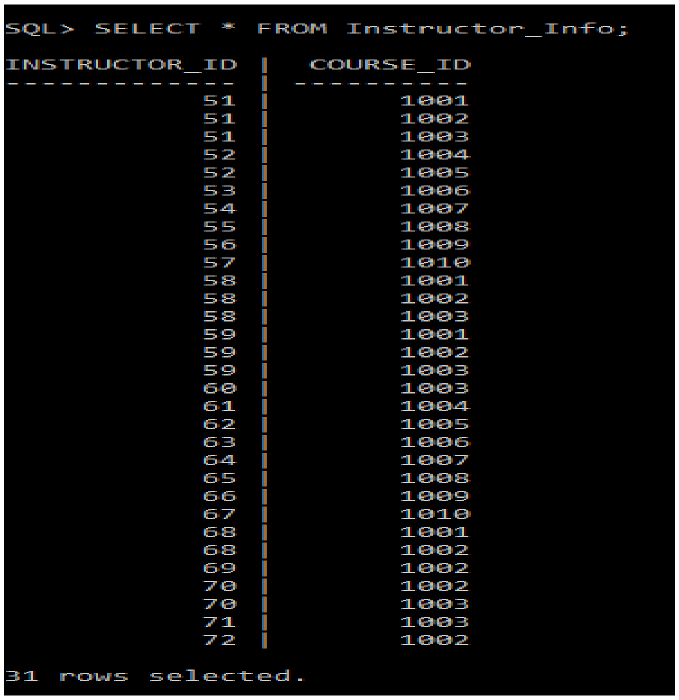
Instructor table



```
Run SQL Command Line

SQL> SELECT * FROM Instructor;

INSTRUCTOR_ID | INS_FIRST_NAME                 | INS_LAST_NAME                  |    INS_AGE | INS_GENDER | INS_DATE_ | INS_JOINI | INS_TYPE             | INS_SALARY | INS_EMAIL
------------- | ------------------------------ | ------------------------------ | ---------- | ---------- | --------- | --------- | -------------------- | ---------- | -------------------------
           51 | Saroj                          | Thapa                          |         40 | Male       | 09-JAN-80 | 19-JAN-17 | Course Leader        |      60000 | saroj@gmail.com
           52 | Nidhi                          | Gupta                          |         32 | Female     | 09-APR-88 | 25-APR-16 | Course Leader        |      60000 | nidhi@gmail.com
           53 | Ram                            | Gopal                          |         33 | Male       | 03-JUN-87 | 25-MAR-16 | Course Leader        |      60000 | ram@gmail.com
           54 | Nala                           | Shrestha                       |         35 | Female     | 04-DEC-85 | 20-MAR-19 | Course Leader        |      60000 | nala@gmail.com
           55 | Mirza                          | Khan                           |         35 | Male       | 04-NOV-85 | 20-APR-15 | Course Leader        |      60000 | mirza@gmail.com
           56 | Simon                          | Shrestha                       |         28 | Male       | 10-DEC-92 | 05-AUG-18 | Course Leader        |      60000 | simon@gmail.com
           57 | Season                         | Khadka                         |         30 | Male       | 02-NOV-90 | 19-APR-19 | Course Leader        |      60000 | season@gmail.com
           58 | Aman                           | Maharjan                       |         31 | Male       | 10-DEC-89 | 05-AUG-19 | Module Leader        |      55000 | aman@gmail.com
           59 | Siddhartha                     | Ghimire                        |         25 | Male       | 10-NOV-95 | 05-AUG-19 | Module Leader        |      55000 | siddhartha@gmail.com
           60 | Neha                           | Banu                           |         31 | Female     | 15-DEC-89 | 20-AUG-19 | Module Leader        |      55000 | neha@gmail.com
           61 | Lubna                          | Karki                          |         27 | Female     | 12-MAY-93 | 20-AUG-18 | Module Leader        |      55000 | lubna@gmail.com
           62 | Ashish                         | RAI                            |         31 | Male       | 15-JUL-89 | 20-MAY-19 | Module Leader        |      55000 | ashish@gmail.com
           63 | Ashiq                          | RAJ                            |         26 | Male       | 11-SEP-94 | 05-OCT-19 | Module Leader        |      55000 | ashiq@gmail.com
           64 | Smriti                         | Basnet                         |         24 | Female     | 15-JUL-96 | 20-MAY-19 | Module Leader        |      55000 | smriti@gmail.com
           65 | Suman                          | Thapa                          |         34 | Female     | 11-SEP-86 | 05-SEP-19 | Module Leader        |      55000 | suman@gmail.com
           66 | Arun                           | Kumar                          |         39 | Male       | 15-JUL-81 | 20-MAY-16 | Module Leader        |      55000 | arun@gmail.com
           67 | Kriti                          | Dangol                         |         39 | Female     | 15-FEB-81 | 20-MAY-16 | Module Leader        |      55000 | kriti@gmail.com
           68 | Nawaraj                        | Kafle                          |         29 | Male       | 15-FEB-91 | 20-MAY-17 | Instructor           |      50000 | nawaraj@gmail.com
           69 | Sudip                          | Shrestha                       |         36 | Male       | 15-FEB-84 | 17-MAY-17 | Instructor           |      50000 | sudip@gmail.com
           70 | Rijan                          | Giri                           |         27 | Male       | 17-JAN-93 | 19-APR-18 | Instructor           |      50000 | rijan@gmail.com
           71 | Sara                           | Thapa                          |         40 | Female     | 09-JAN-80 | 19-JAN-17 | Module Leader        |      55000 | sara@gmail.com
           72 | Bajra                          | Bajracharya                    |         30 | Male       | 09-APR-90 | 19-MAY-18 | Instructor           |      50000 | bajra@gmail.com

22 rows selected.
```

Figure 34: Instructor table

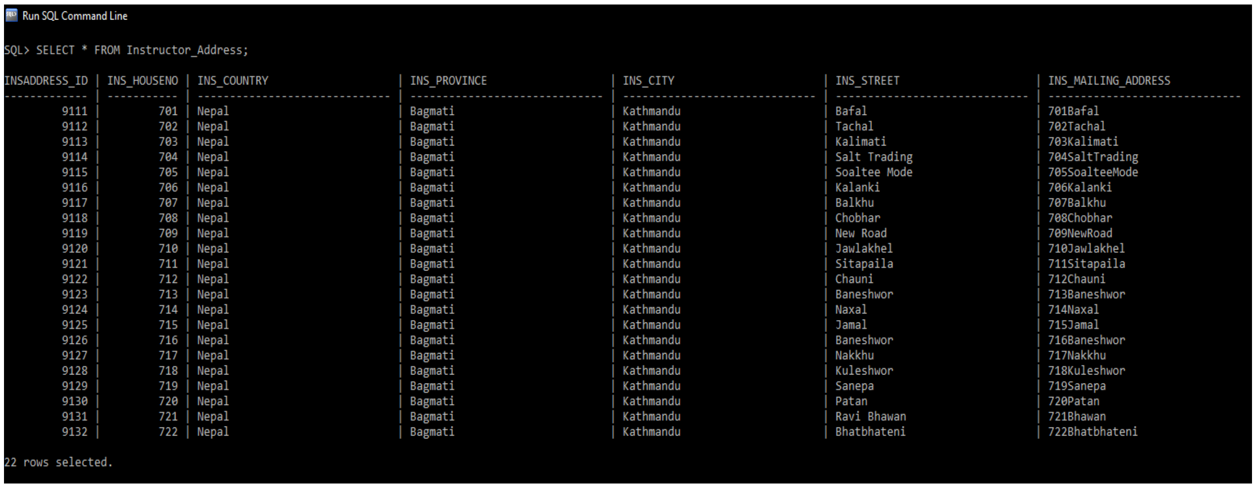
Instructor_Info table



```
SQL> SELECT * FROM Instructor_Info;

INSTRUCTOR_ID |  COURSE_ID
------------- | ----------
           51 |       1001
           51 |       1002
           51 |       1003
           52 |       1004
           52 |       1005
           53 |       1006
           54 |       1007
           55 |       1008
           56 |       1009
           57 |       1010
           58 |       1001
           58 |       1002
           58 |       1003
           59 |       1001
           59 |       1002
           59 |       1003
           60 |       1003
           61 |       1004
           62 |       1005
           63 |       1006
           64 |       1007
           65 |       1008
           66 |       1009
           67 |       1010
           68 |       1001
           68 |       1002
           69 |       1002
           70 |       1002
           70 |       1003
           71 |       1003
           72 |       1002

31 rows selected.
```

Figure 35: Instructor_Info table

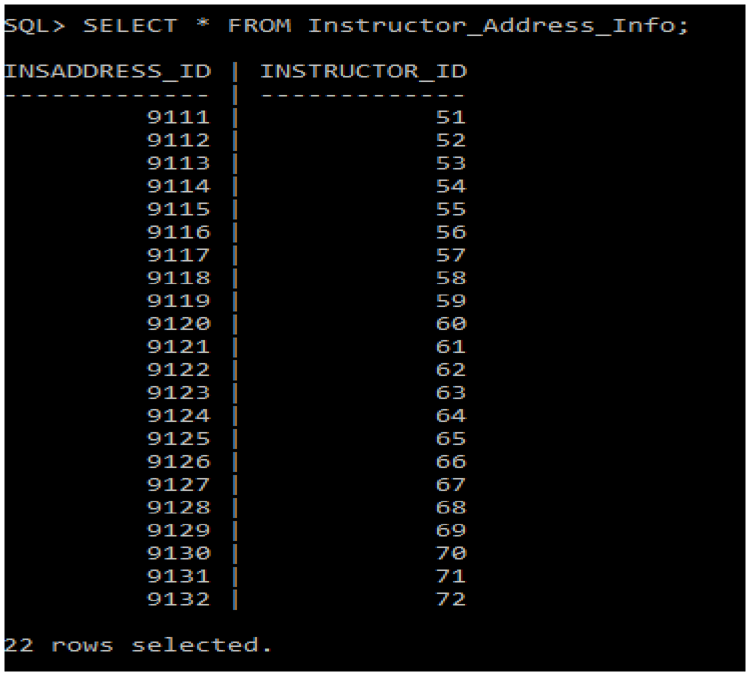
Instructor_Address table



```
Run SQL Command Line

SQL> SELECT * FROM Instructor_Address;

INSADDRESS_ID | INS_HOUSENO | INS_COUNTRY                   | INS_PROVINCE                  | INS_CITY                      | INS_STREET                    | INS_MAILING_ADDRESS
------------- | ----------- | ----------------------------- | ----------------------------- | ----------------------------- | ----------------------------- | -----------------------------
         9111 |         701 | Nepal                         | Bagmati                       | Kathmandu                     | Bafal                         | 701Bafal
         9112 |         702 | Nepal                         | Bagmati                       | Kathmandu                     | Tachal                        | 702Tachal
         9113 |         703 | Nepal                         | Bagmati                       | Kathmandu                     | Kalimati                      | 703Kalimati
         9114 |         704 | Nepal                         | Bagmati                       | Kathmandu                     | Salt Trading                  | 704SaltTrading
         9115 |         705 | Nepal                         | Bagmati                       | Kathmandu                     | Soaltee Mode                  | 705SoalteeMode
         9116 |         706 | Nepal                         | Bagmati                       | Kathmandu                     | Kalanki                       | 706Kalanki
         9117 |         707 | Nepal                         | Bagmati                       | Kathmandu                     | Balkhu                        | 707Balkhu
         9118 |         708 | Nepal                         | Bagmati                       | Kathmandu                     | Chobhar                       | 708Chobhar
         9119 |         709 | Nepal                         | Bagmati                       | Kathmandu                     | New Road                      | 709NewRoad
         9120 |         710 | Nepal                         | Bagmati                       | Kathmandu                     | Jawlakhel                     | 710Jawlakhel
         9121 |         711 | Nepal                         | Bagmati                       | Kathmandu                     | Sitapaila                     | 711Sitapaila
         9122 |         712 | Nepal                         | Bagmati                       | Kathmandu                     | Chauni                        | 712Chauni
         9123 |         713 | Nepal                         | Bagmati                       | Kathmandu                     | Baneshwor                     | 713Baneshwor
         9124 |         714 | Nepal                         | Bagmati                       | Kathmandu                     | Naxal                         | 714Naxal
         9125 |         715 | Nepal                         | Bagmati                       | Kathmandu                     | Jamal                         | 715Jamal
         9126 |         716 | Nepal                         | Bagmati                       | Kathmandu                     | Baneshwor                     | 716Baneshwor
         9127 |         717 | Nepal                         | Bagmati                       | Kathmandu                     | Nakkhu                        | 717Nakkhu
         9128 |         718 | Nepal                         | Bagmati                       | Kathmandu                     | Kuleshwor                     | 718Kuleshwor
         9129 |         719 | Nepal                         | Bagmati                       | Kathmandu                     | Sanepa                        | 719Sanepa
         9130 |         720 | Nepal                         | Bagmati                       | Kathmandu                     | Patan                         | 720Patan
         9131 |         721 | Nepal                         | Bagmati                       | Kathmandu                     | Ravi Bhawan                   | 721Bhawan
         9132 |         722 | Nepal                         | Bagmati                       | Kathmandu                     | Bhatbhateni                   | 722Bhatbhateni

22 rows selected.
```

Figure 36: Instructor_Address table

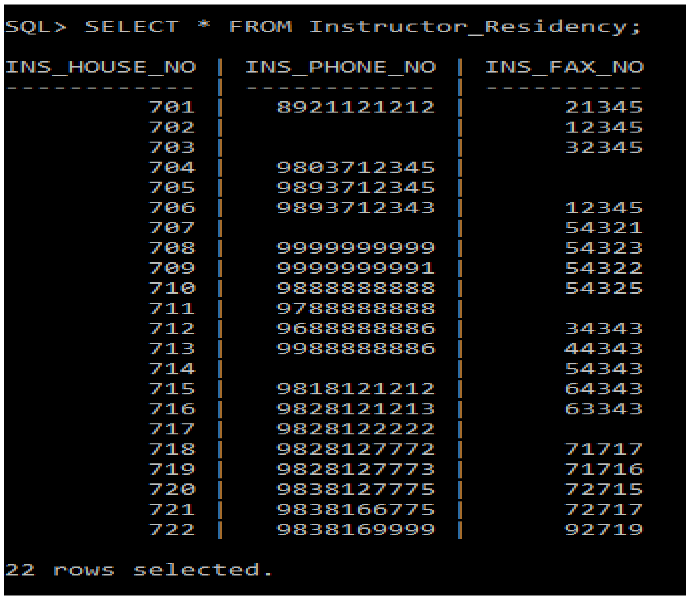
Instructor_Address_Info table



```
SQL> SELECT * FROM Instructor_Address_Info;

INSADDRESS_ID | INSTRUCTOR_ID
------------- | -------------
         9111 |            51
         9112 |            52
         9113 |            53
         9114 |            54
         9115 |            55
         9116 |            56
         9117 |            57
         9118 |            58
         9119 |            59
         9120 |            60
         9121 |            61
         9122 |            62
         9123 |            63
         9124 |            64
         9125 |            65
         9126 |            66
         9127 |            67
         9128 |            68
         9129 |            69
         9130 |            70
         9131 |            71
         9132 |            72

22 rows selected.
```

Figure 37: Instructor_Address_Info table

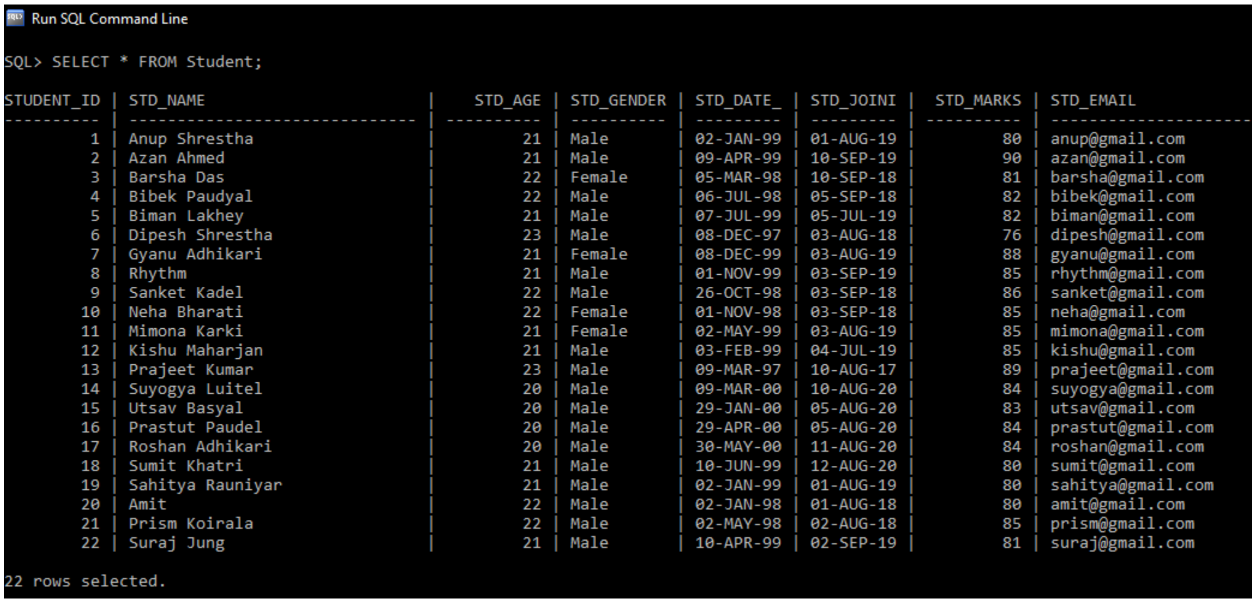
Instructor_Residency table



```
SQL> SELECT * FROM Instructor_Residency;

INS_HOUSE_NO | INS_PHONE_NO | INS_FAX_NO
------------ | ------------ | ----------
         701 |   8921121212 |      21345
         702 |              |      12345
         703 |              |      32345
         704 |   9803712345 |
         705 |   9893712345 |
         706 |   9893712343 |      12345
         707 |              |      54321
         708 |   9999999999 |      54323
         709 |   9999999991 |      54322
         710 |   9888888888 |      54325
         711 |   9788888888 |
         712 |   9688888886 |      34343
         713 |   9988888886 |      44343
         714 |              |      54343
         715 |   9818121212 |      64343
         716 |   9828121213 |      63343
         717 |   9828122222 |
         718 |   9828127772 |      71717
         719 |   9828127773 |      71716
         720 |   9838127775 |      72715
         721 |   9838166775 |      72717
         722 |   9838169999 |      92719

22 rows selected.
```

Figure 38: Instructor_Residency table

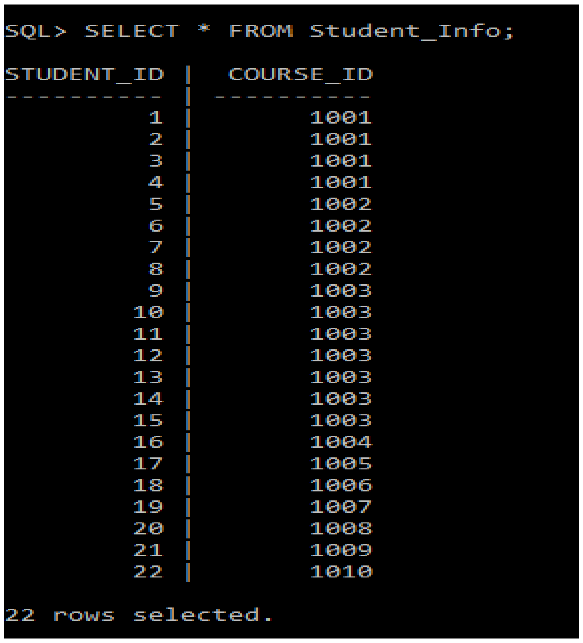
Student table



```
Run SQL Command Line

SQL> SELECT * FROM Student;

STUDENT_ID | STD_NAME                       |    STD_AGE | STD_GENDER | STD_DATE_ | STD_JOINI |  STD_MARKS | STD_EMAIL
---------- | ------------------------------ | ---------- | ---------- | --------- | --------- | ---------- | --------------------
         1 | Anup Shrestha                  |         21 | Male       | 02-JAN-99 | 01-AUG-19 |         80 | anup@gmail.com
         2 | Azan Ahmed                     |         21 | Male       | 09-APR-99 | 10-SEP-19 |         90 | azan@gmail.com
         3 | Barsha Das                     |         22 | Female     | 05-MAR-98 | 10-SEP-18 |         81 | barsha@gmail.com
         4 | Bibek Paudyal                  |         22 | Male       | 06-JUL-98 | 05-SEP-18 |         82 | bibek@gmail.com
         5 | Biman Lakhey                   |         21 | Male       | 07-JUL-99 | 05-JUL-19 |         82 | biman@gmail.com
         6 | Dipesh Shrestha                |         23 | Male       | 08-DEC-97 | 03-AUG-18 |         76 | dipesh@gmail.com
         7 | Gyanu Adhikari                 |         21 | Female     | 08-DEC-99 | 03-AUG-19 |         88 | gyanu@gmail.com
         8 | Rhythm                         |         21 | Male       | 01-NOV-99 | 03-SEP-19 |         85 | rhythm@gmail.com
         9 | Sanket Kadel                   |         22 | Male       | 26-OCT-98 | 03-SEP-18 |         86 | sanket@gmail.com
        10 | Neha Bharati                   |         22 | Female     | 01-NOV-98 | 03-SEP-18 |         85 | neha@gmail.com
        11 | Mimona Karki                   |         21 | Female     | 02-MAY-99 | 03-AUG-19 |         85 | mimona@gmail.com
        12 | Kishu Maharjan                 |         21 | Male       | 03-FEB-99 | 04-JUL-19 |         85 | kishu@gmail.com
        13 | Prajeet Kumar                  |         23 | Male       | 09-MAR-97 | 10-AUG-17 |         89 | prajeet@gmail.com
        14 | Suyogya Luitel                 |         20 | Male       | 09-MAR-00 | 10-AUG-20 |         84 | suyogya@gmail.com
        15 | Utsav Basyal                   |         20 | Male       | 29-JAN-00 | 05-AUG-20 |         83 | utsav@gmail.com
        16 | Prastut Paudel                 |         20 | Male       | 29-APR-00 | 05-AUG-20 |         84 | prastut@gmail.com
        17 | Roshan Adhikari                |         20 | Male       | 30-MAY-00 | 11-AUG-20 |         84 | roshan@gmail.com
        18 | Sumit Khatri                   |         21 | Male       | 10-JUN-99 | 12-AUG-20 |         80 | sumit@gmail.com
        19 | Sahitya Rauniyar               |         21 | Male       | 02-JAN-99 | 01-AUG-19 |         80 | sahitya@gmail.com
        20 | Amit                           |         22 | Male       | 02-JAN-98 | 01-AUG-18 |         80 | amit@gmail.com
        21 | Prism Koirala                  |         22 | Male       | 02-MAY-98 | 02-AUG-18 |         85 | prism@gmail.com
        22 | Suraj Jung                     |         21 | Male       | 10-APR-99 | 02-SEP-19 |         81 | suraj@gmail.com

22 rows selected.
```

Figure 39: Student table

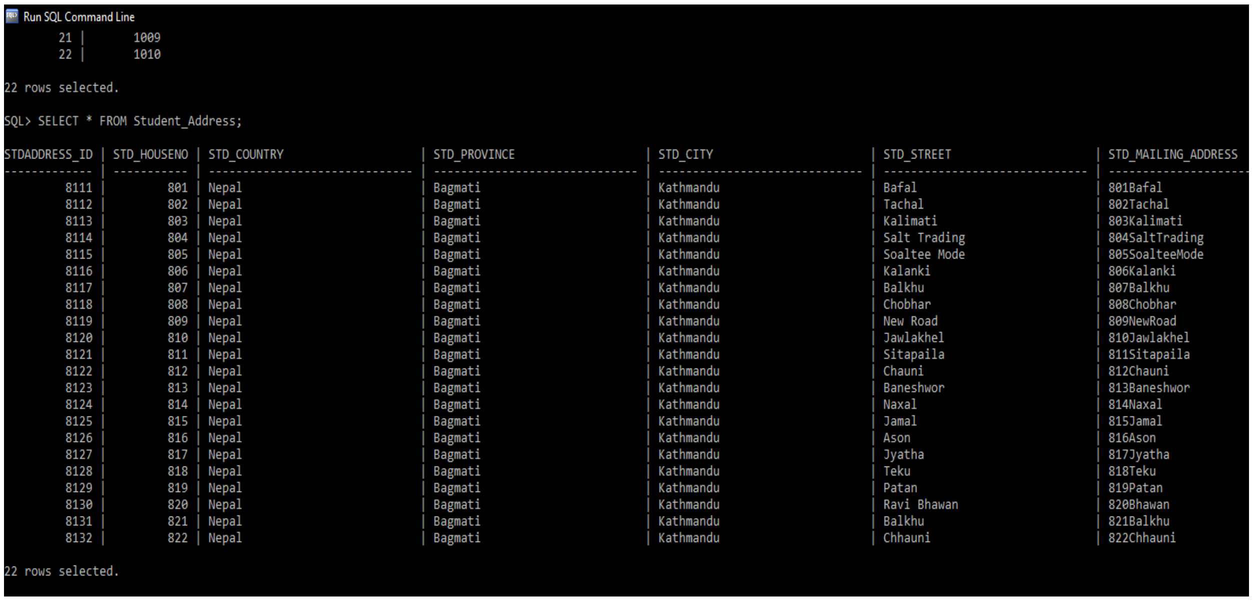
Student_Info table



```
SQL> SELECT * FROM Student_Info;

STUDENT_ID |  COURSE_ID
---------- | ----------
         1 |       1001
         2 |       1001
         3 |       1001
         4 |       1001
         5 |       1002
         6 |       1002
         7 |       1002
         8 |       1002
         9 |       1003
        10 |       1003
        11 |       1003
        12 |       1003
        13 |       1003
        14 |       1003
        15 |       1003
        16 |       1004
        17 |       1005
        18 |       1006
        19 |       1007
        20 |       1008
        21 |       1009
        22 |       1010

22 rows selected.
```

Figure 40: Student_Info table

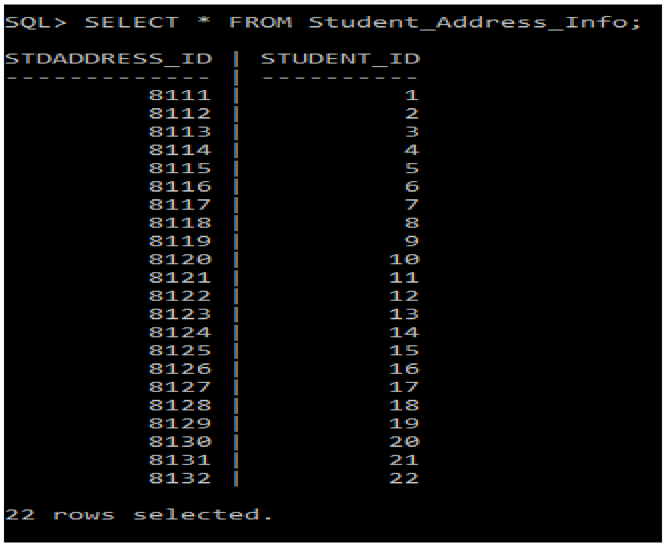
Student_Address table



```
Run SQL Command Line
       21 |       1009
       22 |       1010

22 rows selected.

SQL> SELECT * FROM Student_Address;

STDADDRESS_ID | STD_HOUSENO | STD_COUNTRY                   | STD_PROVINCE                  | STD_CITY                      | STD_STREET                    | STD_MAILING_ADDRESS
------------- | ----------- | ----------------------------- | ----------------------------- | ----------------------------- | ----------------------------- | --------------------
         8111 |         801 | Nepal                         | Bagmati                       | Kathmandu                     | Bafal                         | 801Bafal
         8112 |         802 | Nepal                         | Bagmati                       | Kathmandu                     | Tachal                        | 802Tachal
         8113 |         803 | Nepal                         | Bagmati                       | Kathmandu                     | Kalimati                      | 803Kalimati
         8114 |         804 | Nepal                         | Bagmati                       | Kathmandu                     | Salt Trading                  | 804SaltTrading
         8115 |         805 | Nepal                         | Bagmati                       | Kathmandu                     | Soaltee Mode                  | 805SoalteeMode
         8116 |         806 | Nepal                         | Bagmati                       | Kathmandu                     | Kalanki                       | 806Kalanki
         8117 |         807 | Nepal                         | Bagmati                       | Kathmandu                     | Balkhu                        | 807Balkhu
         8118 |         808 | Nepal                         | Bagmati                       | Kathmandu                     | Chobhar                       | 808Chobhar
         8119 |         809 | Nepal                         | Bagmati                       | Kathmandu                     | New Road                      | 809NewRoad
         8120 |         810 | Nepal                         | Bagmati                       | Kathmandu                     | Jawlakhel                     | 810Jawlakhel
         8121 |         811 | Nepal                         | Bagmati                       | Kathmandu                     | Sitapaila                     | 811Sitapaila
         8122 |         812 | Nepal                         | Bagmati                       | Kathmandu                     | Chauni                        | 812Chauni
         8123 |         813 | Nepal                         | Bagmati                       | Kathmandu                     | Baneshwor                     | 813Baneshwor
         8124 |         814 | Nepal                         | Bagmati                       | Kathmandu                     | Naxal                         | 814Naxal
         8125 |         815 | Nepal                         | Bagmati                       | Kathmandu                     | Jamal                         | 815Jamal
         8126 |         816 | Nepal                         | Bagmati                       | Kathmandu                     | Ason                          | 816Ason
         8127 |         817 | Nepal                         | Bagmati                       | Kathmandu                     | Jyatha                        | 817Jyatha
         8128 |         818 | Nepal                         | Bagmati                       | Kathmandu                     | Teku                          | 818Teku
         8129 |         819 | Nepal                         | Bagmati                       | Kathmandu                     | Patan                         | 819Patan
         8130 |         820 | Nepal                         | Bagmati                       | Kathmandu                     | Ravi Bhawan                   | 820Bhawan
         8131 |         821 | Nepal                         | Bagmati                       | Kathmandu                     | Balkhu                        | 821Balkhu
         8132 |         822 | Nepal                         | Bagmati                       | Kathmandu                     | Chhauni                       | 822Chhauni

22 rows selected.
```

Figure 41: Student_Address table

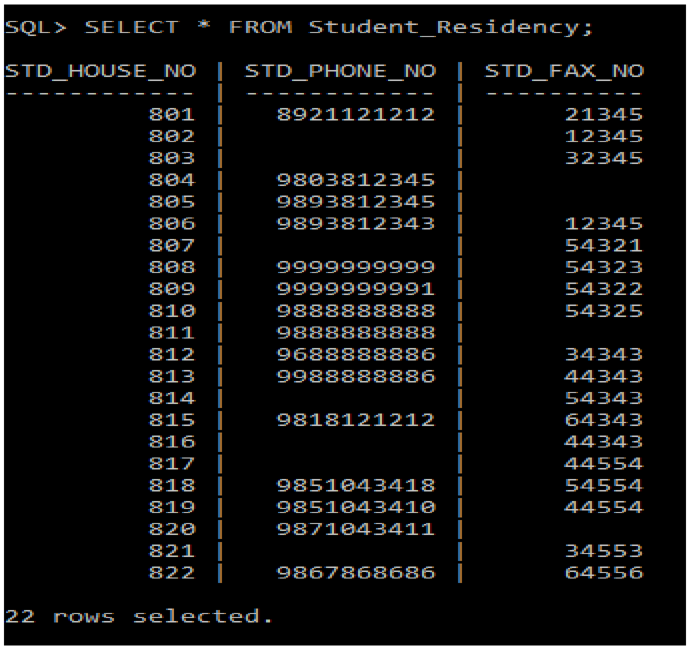
Student_Address_Info table



```
SQL> SELECT * FROM Student_Address_Info;

STDADDRESS_ID | STUDENT_ID
------------- | ----------
         8111 |          1
         8112 |          2
         8113 |          3
         8114 |          4
         8115 |          5
         8116 |          6
         8117 |          7
         8118 |          8
         8119 |          9
         8120 |         10
         8121 |         11
         8122 |         12
         8123 |         13
         8124 |         14
         8125 |         15
         8126 |         16
         8127 |         17
         8128 |         18
         8129 |         19
         8130 |         20
         8131 |         21
         8132 |         22

22 rows selected.
```

Figure 42: Student_Address_Info table

Student_Residency table



```
SQL> SELECT * FROM Student_Residency;

STD_HOUSE_NO | STD_PHONE_NO | STD_FAX_NO
------------ | ------------ | ----------
         801 |   8921121212 |      21345
         802 |              |      12345
         803 |              |      32345
         804 |   9803812345 |
         805 |   9893812345 |
         806 |   9893812343 |      12345
         807 |              |      54321
         808 |   9999999999 |      54323
         809 |   9999999991 |      54322
         810 |   9888888888 |      54325
         811 |   9888888888 |
         812 |   9688888886 |      34343
         813 |   9988888886 |      44343
         814 |              |      54343
         815 |   9818121212 |      64343
         816 |              |      44343
         817 |              |      44554
         818 |   9851043418 |      54554
         819 |   9851043410 |      44554
         820 |   9871043411 |
         821 |              |      34553
         822 |   9867868686 |      64556

22 rows selected.
```

Figure 43: Student_Residency table

4. Information and Transaction Queries

4.1. Information Queries

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

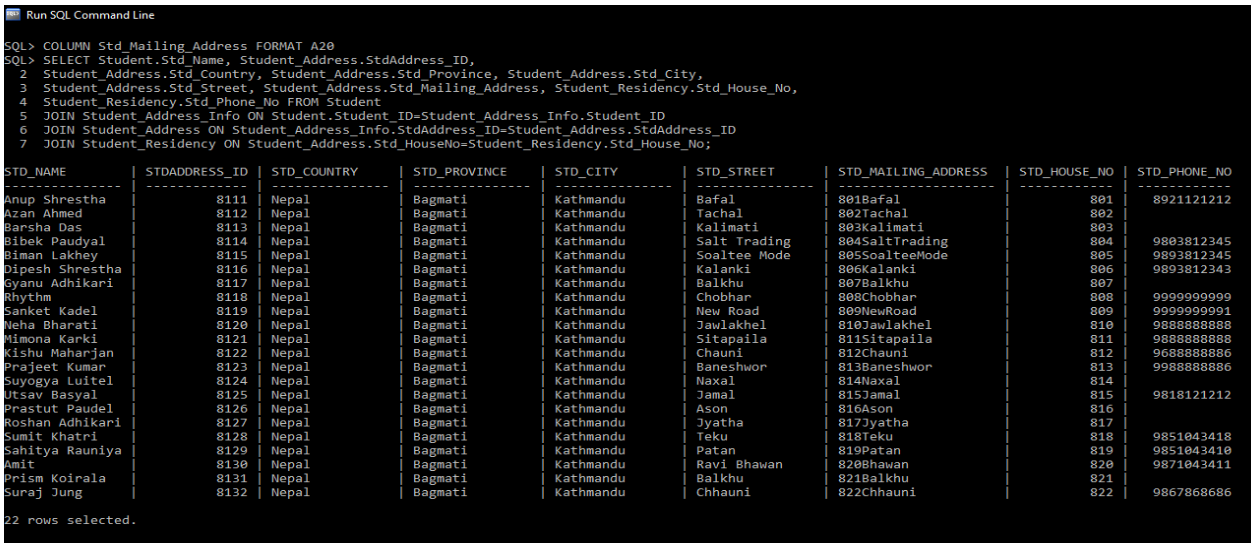
SELECT Student.Std_Name, Student_Address.StdAddress_ID, Student_Address.Std_Country, Student_Address.Std_Province, Student_Address.Std_City, Student_Address.Std_Street, Student_Address.Std_Mailing_Address, Student_Residency.Std_House_No,

Student_Residency.Std_Phone_No FROM Student

JOIN Student_Address_Info ON Student.Student_ID=Student_Address_Info.Student_ID

JOIN Student_Address ON Student_Address_Info.StdAddress_ID=Student_Address.StdAddress_ID

JOIN Student_Residency ON Student_Address.Std_HouseNo=Student_Residency.Std_House_No;



```
SQL> COLUMN Std_Mailing_Address FORMAT A20
SQL> SELECT Student.Std_Name, Student_Address.StdAddress_ID,
     Student_Address.Std_Country, Student_Address.Std_Province, Student_Address.Std_City,
     Student_Address.Std_Street, Student_Address.Std_Mailing_Address, Student_Residency.Std_House_No,
     Student_Residency.Std_Phone_No FROM Student
     JOIN Student_Address_Info ON Student.Student_ID=Student_Address_Info.Student_ID
     JOIN Student_Address ON Student_Address_Info.StdAddress_ID=Student_Address.StdAddress_ID
     JOIN Student_Residency ON Student_Address.Std_HouseNo=Student_Residency.Std_House_No;

STD_NAME        | STDADDRESS_ID | STD_COUNTRY     | STD_PROVINCE    | STD_CITY        | STD_STREET      | STD_MAILING_ADDRESS  | STD_HOUSE_NO | STD_PHONE_NO
--------------- | ------------- | --------------- | --------------- | --------------- | --------------- | -------------------- | ------------ | ------------
Anup Shrestha   |          8111 | Nepal           | Bagmati         | Kathmandu       | Bafal           | 801Bafal             |          801 |   8921121212
Azan Ahmed      |          8112 | Nepal           | Bagmati         | Kathmandu       | Tachal          | 802Tachal            |          802 |
Barsha Das      |          8113 | Nepal           | Bagmati         | Kathmandu       | Kalimati        | 803Kalimati          |          803 |
Bibek Paudyal   |          8114 | Nepal           | Bagmati         | Kathmandu       | Salt Trading    | 804SaltTrading       |          804 |   9803812345
Biman Lakhey    |          8115 | Nepal           | Bagmati         | Kathmandu       | Soaltee Mode    | 805SoalteeMode       |          805 |   9893812345
Dipesh Shrestha |          8116 | Nepal           | Bagmati         | Kathmandu       | Kalanki         | 806Kalanki           |          806 |   9893812343
Gyanu Adhikari  |          8117 | Nepal           | Bagmati         | Kathmandu       | Balkhu          | 807Balkhu            |          807 |
Rhythm          |          8118 | Nepal           | Bagmati         | Kathmandu       | Chobhar         | 808Chobhar           |          808 |   9999999999
Sanket Kadel    |          8119 | Nepal           | Bagmati         | Kathmandu       | New Road        | 809NewRoad           |          809 |   9999999991
Neha Bharati    |          8120 | Nepal           | Bagmati         | Kathmandu       | Jawlakhel       | 810Jawlakhel         |          810 |   9888888888
Mimona Karki    |          8121 | Nepal           | Bagmati         | Kathmandu       | Sitapaila       | 811Sitapaila         |          811 |   9888888888
Kishu Maharjan  |          8122 | Nepal           | Bagmati         | Kathmandu       | Chauni          | 812Chauni            |          812 |   9688888886
Prajeet Kumar   |          8123 | Nepal           | Bagmati         | Kathmandu       | Baneshwor       | 813Baneshwor         |          813 |   9988888886
Suyogya Luitel  |          8124 | Nepal           | Bagmati         | Kathmandu       | Naxal           | 814Naxal             |          814 |
Utsav Basyal    |          8125 | Nepal           | Bagmati         | Kathmandu       | Jamal           | 815Jamal             |          815 |   9818121212
Prastut Paudel  |          8126 | Nepal           | Bagmati         | Kathmandu       | Ason            | 816Ason              |          816 |
Roshan Adhikari |          8127 | Nepal           | Bagmati         | Kathmandu       | Jyatha          | 817Jyatha            |          817 |
Sumit Khatri    |          8128 | Nepal           | Bagmati         | Kathmandu       | Teku            | 818Teku              |          818 |   9851043418
Sahitya Rauniya |          8129 | Nepal           | Bagmati         | Kathmandu       | Patan           | 819Patan             |          819 |   9851043410
Amit            |          8130 | Nepal           | Bagmati         | Kathmandu       | Ravi Bhawan     | 820Bhawan            |          820 |   9871043411
Prism Koirala   |          8131 | Nepal           | Bagmati         | Kathmandu       | Balkhu          | 821Balkhu            |          821 |
Suraj Jung      |          8132 | Nepal           | Bagmati         | Kathmandu       | Chhauni         | 822Chhauni           |          822 |   9867868686

22 rows selected.
```

Figure 44: Information Query 1

In this query, Student name has been selected from student table and all the attributes (except for student fax no from the student residency table) from tablesStudent Address and Student Residency have been selected. The tables have been joined using INNER JOIN.

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

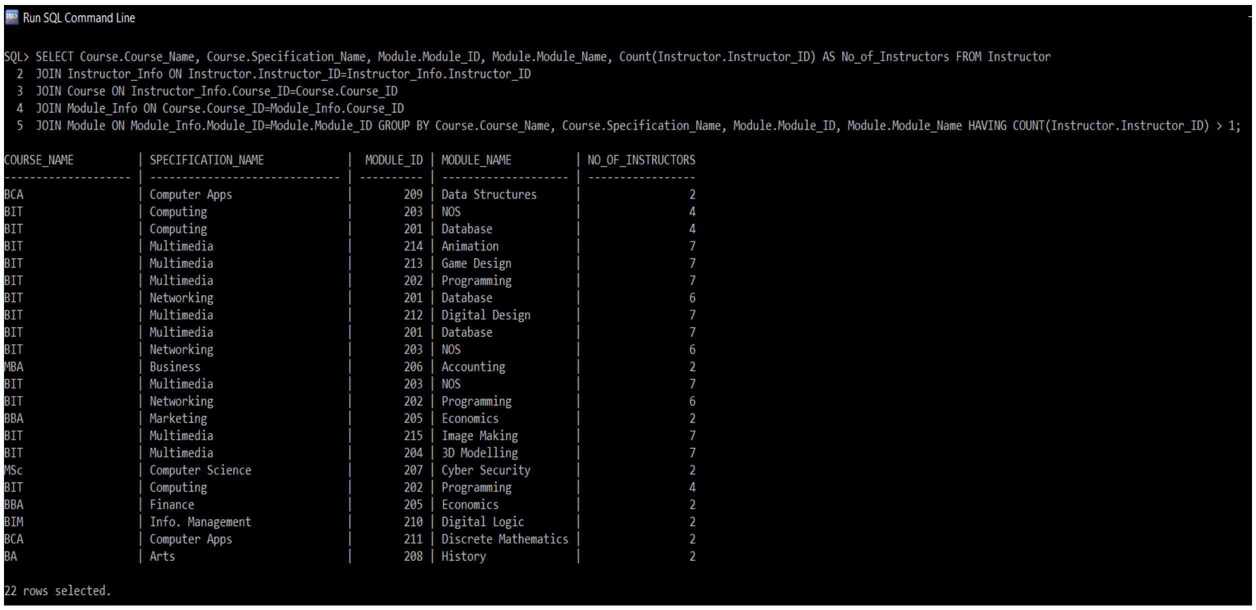
SELECT Course.Course_Name, Course.Specification_Name, Module.Module_ID, Module.Module_Name, Count(Instructor.Instructor_ID) AS No_of_Instructors FROM Instructor

JOIN Instructor_Info ON Instructor.Instructor_ID=Instructor_Info.Instructor_ID

JOIN Course ON Instructor_Info.Course_ID=Course.Course_ID

JOIN Module_Info ON Course.Course_ID=Module_Info.Course_ID

JOIN Module ON Module_Info.Module_ID=Module.Module_ID GROUP BY Course.Course_Name, Course.Specification_Name, Module.Module_ID, Module.Module_Name HAVING COUNT(Instructor.Instructor_ID) > 1;



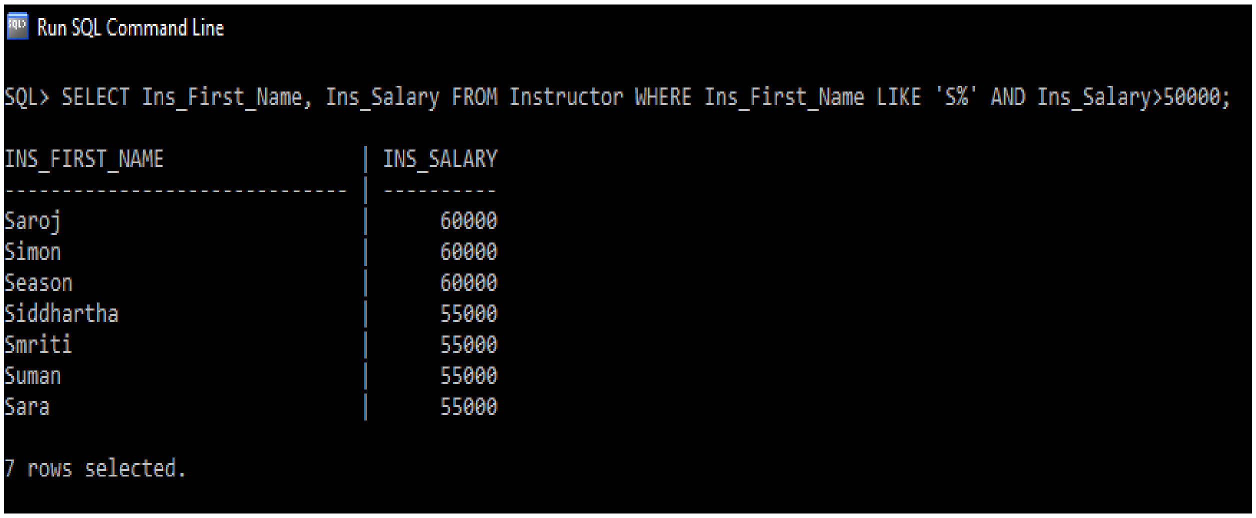
```
SQL> SELECT Course.Course_Name, Course.Specification_Name, Module.Module_ID, Module.Module_Name, Count(Instructor.Instructor_ID) AS No_of_Instructors FROM Instructor
  2  JOIN Instructor_Info ON Instructor.Instructor_ID=Instructor_Info.Instructor_ID
  3  JOIN Course ON Instructor_Info.Course_ID=Course.Course_ID
  4  JOIN Module_Info ON Course.Course_ID=Module_Info.Course_ID
  5  JOIN Module ON Module_Info.Module_ID=Module.Module_ID GROUP BY Course.Course_Name, Course.Specification_Name, Module.Module_ID, Module.Module_Name HAVING COUNT(Instructor.Instructor_ID) > 1;

COURSE_NAME         | SPECIFICATION_NAME            |  MODULE_ID | MODULE_NAME          | NO_OF_INSTRUCTORS
------------------- | ----------------------------- | ---------- | -------------------- | -----------------
BCA                 | Computer Apps                 |        209 | Data Structures      |                 2
BIT                 | Computing                     |        203 | NOS                  |                 4
BIT                 | Computing                     |        201 | Database             |                 4
BIT                 | Multimedia                    |        214 | Animation            |                 7
BIT                 | Multimedia                    |        213 | Game Design          |                 7
BIT                 | Multimedia                    |        202 | Programming          |                 7
BIT                 | Networking                    |        201 | Database             |                 6
BIT                 | Multimedia                    |        212 | Digital Design       |                 7
BIT                 | Multimedia                    |        201 | Database             |                 7
BIT                 | Networking                    |        203 | NOS                  |                 6
MBA                 | Business                      |        206 | Accounting           |                 2
BIT                 | Multimedia                    |        203 | NOS                  |                 7
BIT                 | Networking                    |        202 | Programming          |                 6
BBA                 | Marketing                     |        205 | Economics            |                 2
BIT                 | Multimedia                    |        215 | Image Making         |                 7
BIT                 | Multimedia                    |        204 | 3D Modelling         |                 7
MSc                 | Computer Science              |        207 | Cyber Security       |                 2
BIT                 | Computing                     |        202 | Programming          |                 4
BBA                 | Finance                       |        205 | Economics            |                 2
BIM                 | Info. Management              |        210 | Digital Logic        |                 2
BCA                 | Computer Apps                 |        211 | Discrete Mathematics |                 2
BA                  | Arts                          |        208 | History              |                 2

22 rows selected.
```

Figure 45: Information Query 2

In this query, course name, specification name, module Id, module name, instructor count have been selected from the tables Course, module and instructor info. The tables have been joined using INNER JOIN.

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

SELECT Ins_First_Name, Ins_Salary FROM Instructor WHERE Ins_First_Name LIKE 'S%' AND Ins_Salary>50000;



```
SQL> SELECT Ins_First_Name, Ins_Salary FROM Instructor WHERE Ins_First_Name LIKE 'S%' AND Ins_Salary>50000;

INS_FIRST_NAME                 | INS_SALARY
------------------------------ | ----------
Saroj                          |      60000
Simon                          |      60000
Season                         |      60000
Siddhartha                     |      55000
Smriti                         |      55000
Suman                          |      55000
Sara                           |      55000

7 rows selected.
```

Figure 46: Information Query 3

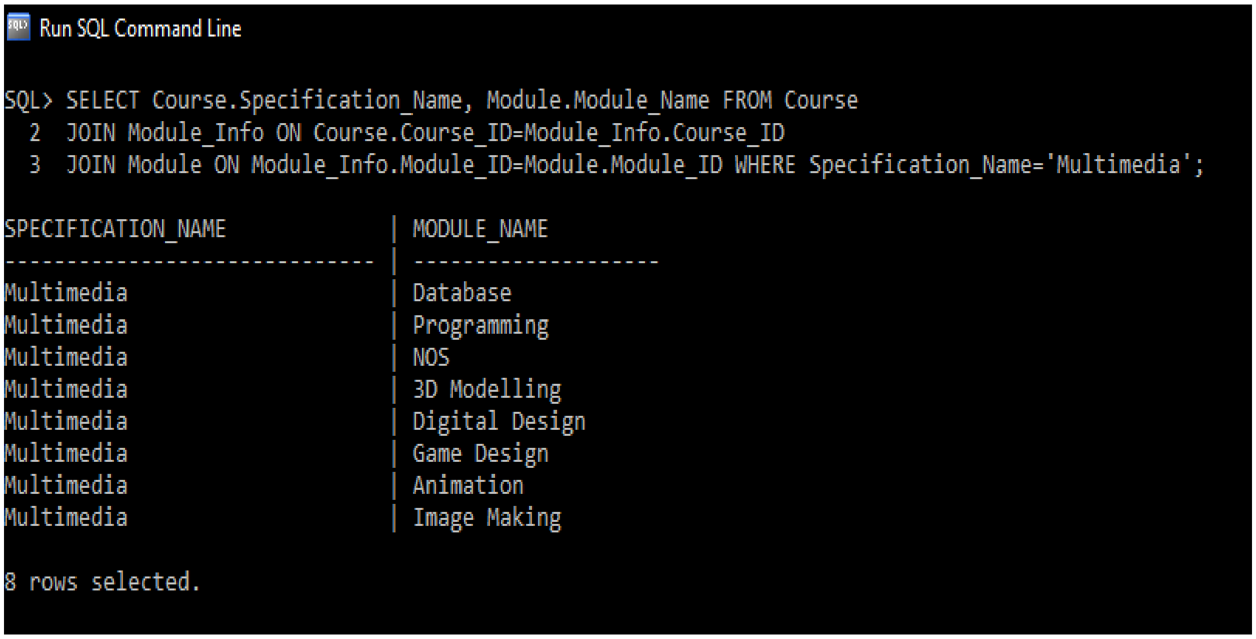
In this query, the details of the instructor whose first name start with ‘s’ and have salary>50000 has been shown. For that, Instructor First name and instructor salary from the table instructor has been selected and WHERE clause and LIKE operator have been used to produce the required output.

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

SELECT Course.Specification_Name, Module.Module_Name FROM Course

JOIN Module_Info ON Course.Course_ID=Module_Info.Course_ID

JOIN Module ON Module_Info.Module_ID=Module.Module_ID WHERE Specification_Name='Multimedia';



Run SQL Command Line

```
SQL> SELECT Course.Specification_Name, Module.Module_Name FROM Course
  2  JOIN Module_Info ON Course.Course_ID=Module_Info.Course_ID
  3  JOIN Module ON Module_Info.Module_ID=Module.Module_ID WHERE Specification_Name='Multimedia';

SPECIFICATION_NAME             | MODULE_NAME
------------------------------ | --------------------
Multimedia                     | Database
Multimedia                     | Programming
Multimedia                     | NOS
Multimedia                     | 3D Modelling
Multimedia                     | Digital Design
Multimedia                     | Game Design
Multimedia                     | Animation
Multimedia                     | Image Making

8 rows selected.
```

Figure 47: Information Query 4

In this query, Specification Name, Module Name have been selected from tables Course and Module to display the modules in the multimedia specification. The tables have been joined using INNER JOIN.

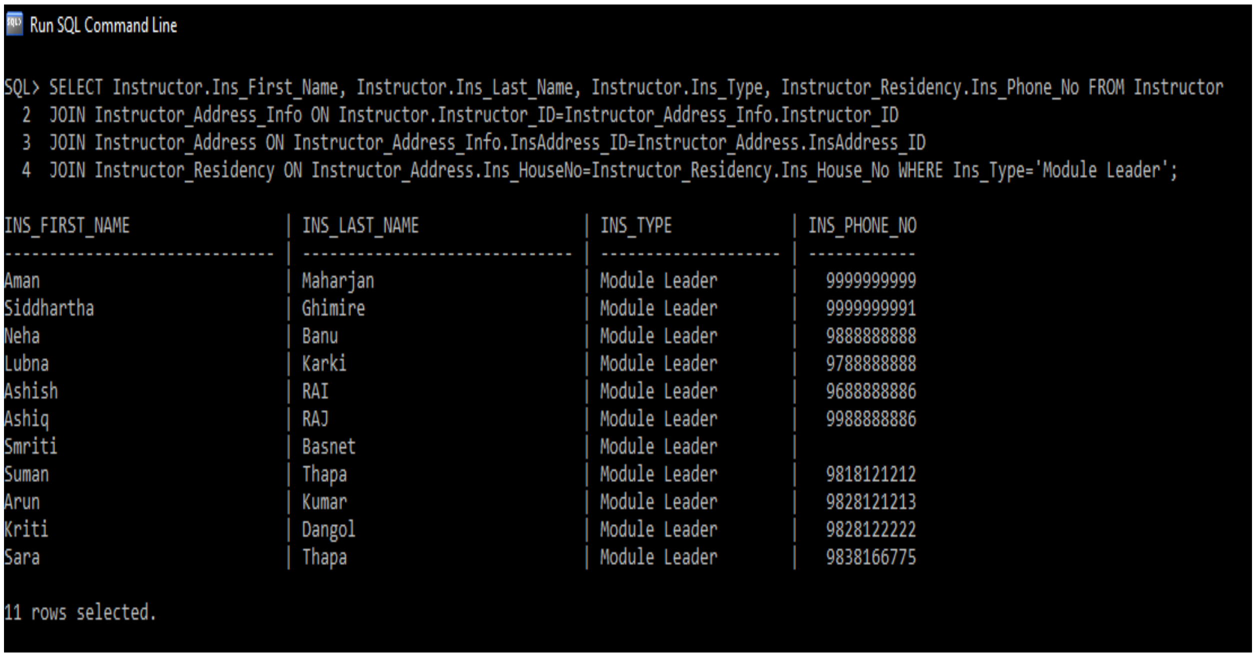
4.1.5. List the name of the head of modules with the list of his phone number.

SELECT Instructor.Ins_First_Name, Instructor.Ins_Last_Name, Instructor.Ins_Type, Instructor_Residency.Ins_Phone_No FROM Instructor

JOIN Instructor_Address_Info ON Instructor.Instructor_ID=Instructor_Address_Info.Instructor_ID

JOIN Instructor_Address ON Instructor_Address_Info.InsAddress_ID=Instructor_Address.InsAddress_ID

JOIN Instructor_Residency ON Instructor_Address.Ins_HouseNo=Instructor_Residency.Ins_House_No WHERE Ins_Type='Module Leader';



```
SQL> SELECT Instructor.Ins_First_Name, Instructor.Ins_Last_Name, Instructor.Ins_Type, Instructor_Residency.Ins_Phone_No FROM Instructor
  2  JOIN Instructor_Address_Info ON Instructor.Instructor_ID=Instructor_Address_Info.Instructor_ID
  3  JOIN Instructor_Address ON Instructor_Address_Info.InsAddress_ID=Instructor_Address.InsAddress_ID
  4  JOIN Instructor_Residency ON Instructor_Address.Ins_HouseNo=Instructor_Residency.Ins_House_No WHERE Ins_Type='Module Leader';

INS_FIRST_NAME                | INS_LAST_NAME                | INS_TYPE           | INS_PHONE_NO
----------------------------- | ---------------------------- | ------------------ | ------------
Aman                          | Maharjan                     | Module Leader      |   9999999999
Siddhartha                    | Ghimire                      | Module Leader      |   9999999991
Neha                          | Banu                         | Module Leader      |   9888888888
Lubna                         | Karki                        | Module Leader      |   9788888888
Ashish                        | RAI                          | Module Leader      |   9688888886
Ashiq                         | RAJ                          | Module Leader      |   9988888886
Smriti                        | Basnet                       | Module Leader      |
Suman                         | Thapa                        | Module Leader      |   9818121212
Arun                          | Kumar                        | Module Leader      |   9828121213
Kriti                         | Dangol                       | Module Leader      |   9828122222
Sara                          | Thapa                        | Module Leader      |   9838166775

11 rows selected.
```

Figure 48: Information Query 5

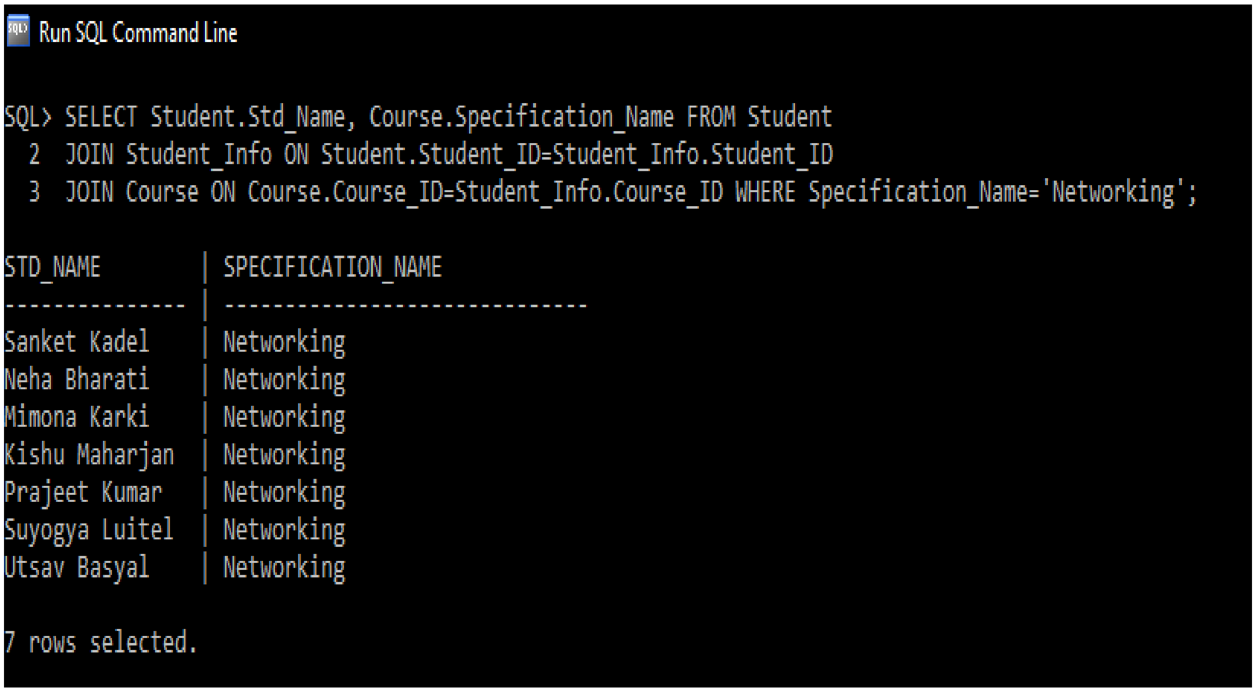
In this query, Instructor First Name, Instructor Last Name, Instructor Type, Instructor Phone No have been selected from tables Instructor and Instructor Residency to show the module leaders along with their phone numbers. The tables have been joined using INNER JOIN.

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

SELECT Student.Std_Name, Course.Specification_Name FROM Student

JOIN Student_Info ON Student.Student_ID=Student_Info.Student_ID

JOIN Course ON Course.Course_ID=Student_Info.Course_ID WHERE Specification_Name='Networking';



```
SQL> SELECT Student.Std_Name, Course.Specification_Name FROM Student
  2  JOIN Student_Info ON Student.Student_ID=Student_Info.Student_ID
  3  JOIN Course ON Course.Course_ID=Student_Info.Course_ID WHERE Specification_Name='Networking';

STD_NAME        | SPECIFICATION_NAME
--------------- | ------------------------------
Sanket Kadel    | Networking
Neha Bharati    | Networking
Mimona Karki    | Networking
Kishu Maharjan  | Networking
Prajeet Kumar   | Networking
Suyogya Luitel  | Networking
Utsav Basyal    | Networking

7 rows selected.
```

Figure 49: Information Query 6

In this query, Student Name, Specification Name have been selected from the tables Student and Course to show the student studying in ‘Networking’ specification along with the student name. The tables have been joined using INNER JOIN.

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

SELECT Instructor.Ins_First_Name, Instructor.Ins_Last_Name, Module.Module_Name, Course.Course_ID, Instructor_Residency.Ins_Fax_No FROM Course

JOIN Module_Info ON Module_Info.Course_ID=Course.Course_ID

JOIN Module ON Module_Info.Module_ID=Module.Module_ID

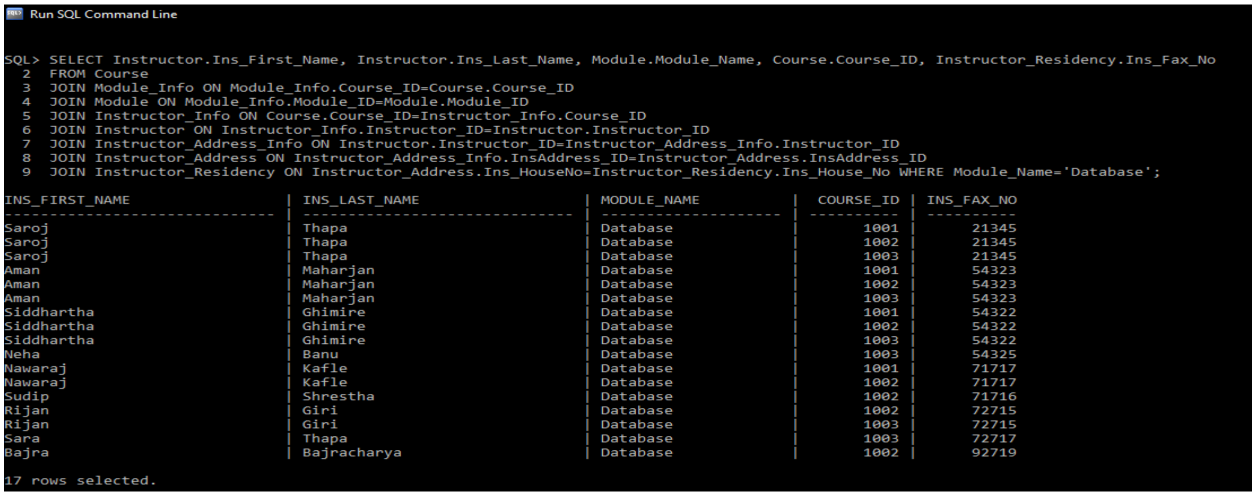
JOIN Instructor_Info ON Course.Course_ID=Instructor_Info.Course_ID

JOIN Instructor ON Instructor_Info.Instructor_ID=Instructor.Instructor_ID

JOIN Instructor_Address_Info ON Instructor.Instructor_ID=Instructor_Address_Info.Instructor_ID

JOIN Instructor_Address ON Instructor_Address_Info.InsAddress_ID=Instructor_Address.InsAddress_ID

JOIN Instructor_Residency ON Instructor_Address.Ins_HouseNo=Instructor_Residency.Ins_House_No WHERE Module_Name='Database';



```
Run SQL Command Line

SQL> SELECT Instructor.Ins_First_Name, Instructor.Ins_Last_Name, Module.Module_Name, Course.Course_ID, Instructor_Residency.Ins_Fax_No
     FROM Course
     JOIN Module_Info ON Module_Info.Course_ID=Course.Course_ID
     JOIN Module ON Module_Info.Module_ID=Module.Module_ID
     JOIN Instructor_Info ON Course.Course_ID=Instructor_Info.Course_ID
     JOIN Instructor ON Instructor_Info.Instructor_ID=Instructor.Instructor_ID
     JOIN Instructor_Address_Info ON Instructor.Instructor_ID=Instructor_Address_Info.Instructor_ID
     JOIN Instructor_Address ON Instructor_Address_Info.InsAddress_ID=Instructor_Address.InsAddress_ID
     JOIN Instructor_Residency ON Instructor_Address.Ins_HouseNo=Instructor_Residency.Ins_House_No WHERE Module_Name='Database';

INS_FIRST_NAME                 | INS_LAST_NAME                  | MODULE_NAME          |  COURSE_ID | INS_FAX_NO
------------------------------ | ------------------------------ | -------------------- | ---------- | ----------
Saroj                          | Thapa                          | Database             |       1001 |      21345
Saroj                          | Thapa                          | Database             |       1002 |      21345
Saroj                          | Thapa                          | Database             |       1003 |      21345
Aman                           | Maharjan                       | Database             |       1001 |      54323
Aman                           | Maharjan                       | Database             |       1002 |      54323
Aman                           | Maharjan                       | Database             |       1003 |      54323
Siddhartha                     | Ghimire                        | Database             |       1001 |      54322
Siddhartha                     | Ghimire                        | Database             |       1002 |      54322
Siddhartha                     | Ghimire                        | Database             |       1003 |      54322
Neha                           | Banu                           | Database             |       1003 |      54325
Nawaraj                        | Kafle                          | Database             |       1001 |      71717
Nawaraj                        | Kafle                          | Database             |       1002 |      71717
Sudip                          | Shrestha                       | Database             |       1002 |      71716
Rijan                          | Giri                           | Database             |       1002 |      72715
Rijan                          | Giri                           | Database             |       1003 |      72715
Sara                           | Thapa                          | Database             |       1003 |      72717
Bajra                          | Bajracharya                    | Database             |       1002 |      92719

17 rows selected.
```

Figure 50: Information Query 7

In this query, Instructor First Name, Instructor Last Name, Module Name, Course Id and Instructor Fax No have been selected from tables Instructor, Module, Course and Instructor Residency to display the desired output. The tables have been joined using INNER JOIN.

4.1.8. List the specification falls under the BIT course.

SELECT Course.Course_Name, Course.Specification_Name FROM Course WHERE Course_Name='BIT';

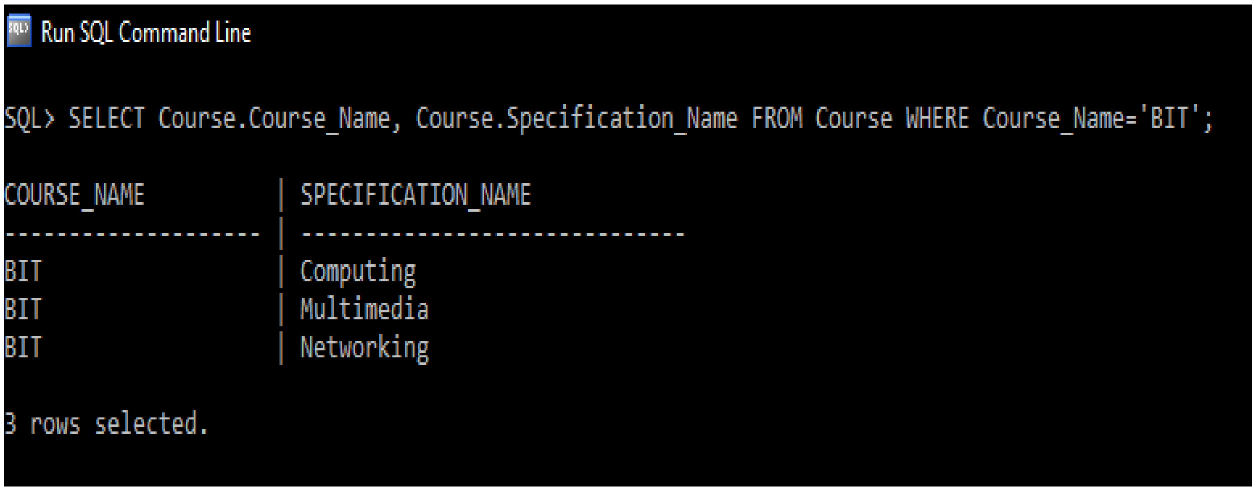
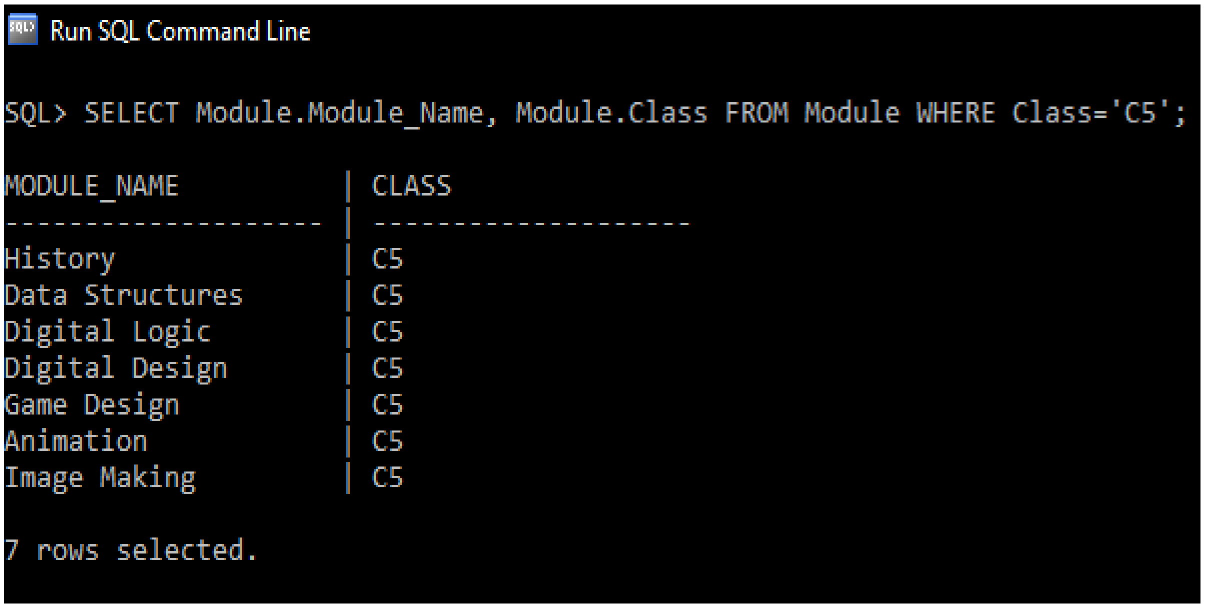

```
Run SQL Command Line

SQL> SELECT Course.Course_Name, Course.Specification_Name FROM Course WHERE Course_Name='BIT';

COURSE_NAME          | SPECIFICATION_NAME
-------------------- | ------------------------------
BIT                  | Computing
BIT                  | Multimedia
BIT                  | Networking

3 rows selected.
```

Figure 51: Information Query 8

In this query, Course Name, Specification Name have been selected from table Course to show the specification under BIT course. To achieve the desired output, WHERE clause has been used.

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

SELECT Module.Module_Name, Module.Class FROM Module WHERE Class='C5';



```
SQL> SELECT Module.Module_Name, Module.Class FROM Module WHERE Class='C5';

MODULE_NAME          | CLASS
-------------------- | --------------------
History              | C5
Data Structures      | C5
Digital Logic        | C5
Digital Design       | C5
Game Design          | C5
Animation            | C5
Image Making         | C5

7 rows selected.
```

Figure 52: Information Query 9

In this query, Module Name, Class have been selected from the table Module and to achieve the desired output which is to show the modules taught in any one particular class, WHERE clause has been used.

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

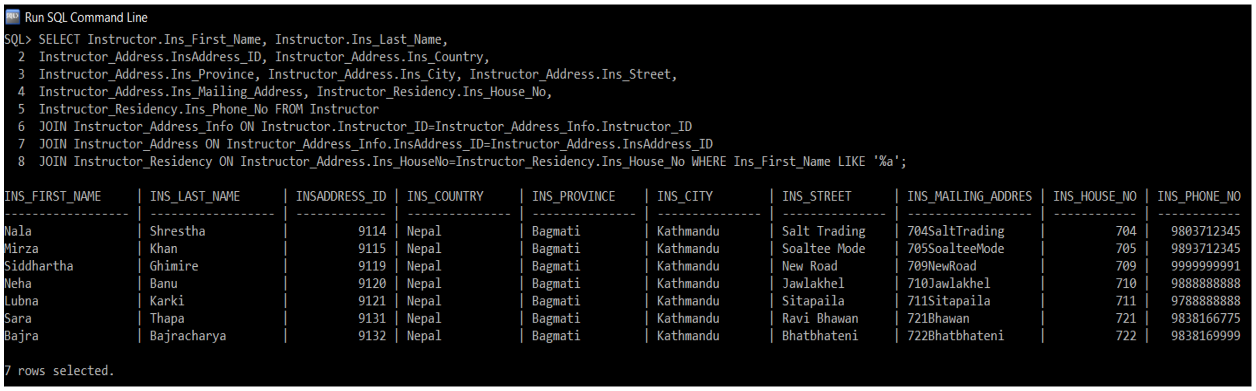
SELECT Instructor.Ins_First_Name, Instructor.Ins_Last_Name, Instructor_Address.InsAddress_ID, Instructor_Address.Ins_Country, Instructor_Address.Ins_Province, Instructor_Address.Ins_City, Instructor_Address.Ins_Street, Instructor_Address.Ins_Mailing_Address, Instructor_Residency.Ins_House_No,

Instructor_Residency.Ins_Phone_No FROM Instructor

JOIN Instructor_Address_Info ON Instructor.Instructor_ID=Instructor_Address_Info.Instructor_ID

JOIN Instructor_Address ON Instructor_Address_Info.InsAddress_ID=Instructor_Address.InsAddress_ID

JOIN Instructor_Residency ON Instructor_Address.Ins_HouseNo=Instructor_Residency.Ins_House_No WHERE Ins_First_Name LIKE '%a';



```
Run SQL Command Line
SQL> SELECT Instructor.Ins_First_Name, Instructor.Ins_Last_Name,
     Instructor_Address.InsAddress_ID, Instructor_Address.Ins_Country,
     Instructor_Address.Ins_Province, Instructor_Address.Ins_City, Instructor_Address.Ins_Street,
     Instructor_Address.Ins_Mailing_Address, Instructor_Residency.Ins_House_No,
     Instructor_Residency.Ins_Phone_No FROM Instructor
     JOIN Instructor_Address_Info ON Instructor.Instructor_ID=Instructor_Address_Info.Instructor_ID
     JOIN Instructor_Address ON Instructor_Address_Info.InsAddress_ID=Instructor_Address.InsAddress_ID
     JOIN Instructor_Residency ON Instructor_Address.Ins_HouseNo=Instructor_Residency.Ins_House_No WHERE Ins_First_Name LIKE '%a';

INS_FIRST_NAME     | INS_LAST_NAME      | INSADDRESS_ID | INS_COUNTRY     | INS_PROVINCE    | INS_CITY        | INS_STREET      | INS_MAILING_ADDRES | INS_HOUSE_NO | INS_PHONE_NO
------------------ | ------------------ | ------------- | --------------- | --------------- | --------------- | --------------- | ------------------ | ------------ | ------------
Nala               | Shrestha           |          9114 | Nepal           | Bagmati         | Kathmandu       | Salt Trading    | 704SaltTrading     |          704 |   9803712345
Mirza              | Khan               |          9115 | Nepal           | Bagmati         | Kathmandu       | Soaltee Mode    | 705SoalteeMode     |          705 |   9893712345
Siddhartha         | Ghimire            |          9119 | Nepal           | Bagmati         | Kathmandu       | New Road        | 709NewRoad         |          709 |   9999999991
Neha               | Banu               |          9120 | Nepal           | Bagmati         | Kathmandu       | Jawlakhel       | 710Jawlakhel       |          710 |   9888888888
Lubna              | Karki              |          9121 | Nepal           | Bagmati         | Kathmandu       | Sitapaila       | 711Sitapaila       |          711 |   9788888888
Sara               | Thapa              |          9131 | Nepal           | Bagmati         | Kathmandu       | Ravi Bhawan     | 721Bhawan          |          721 |   9838166775
Bajra              | Bajracharya        |          9132 | Nepal           | Bagmati         | Kathmandu       | Bhatbhateni     | 722Bhatbhateni     |          722 |   9838169999

7 rows selected.
```

Figure 53: Information Query 10

In this query, Instructor First Name, Instructor Last Name from Instructor table and all the attributes (except for Instructor Fax No) have been selected from tables Instructor Address, Instructor Residency. WHERE clause along with LIKE operator has been used to show only the names of the instructors who have ‘a’ at the end of their first name. The tables have been joined using INNER JOIN.

4.2. Transaction Queries

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

SELECT Student.Student_ID, Student.Std_Name, Course.Course_Name, Course.Specification_Name, Course.Course_Fees, CASE Course.Specification_Name

WHEN 'Computing' THEN Course_Fees-Course_Fees*0.1

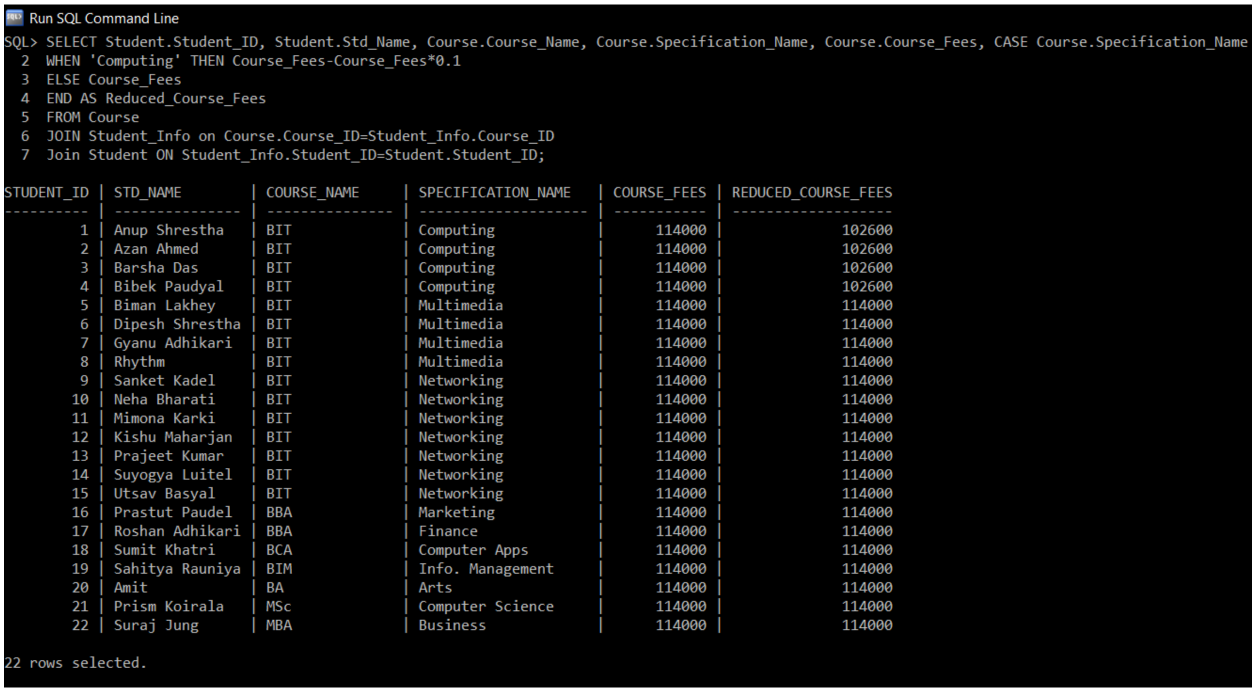
ELSE Course_Fees

END AS Reduced_Course_Fees

FROM Course

JOIN Student_Info on Course.Course_ID=Student_Info.Course_ID

Join Student ON Student_Info.Student_ID=Student.Student_ID;



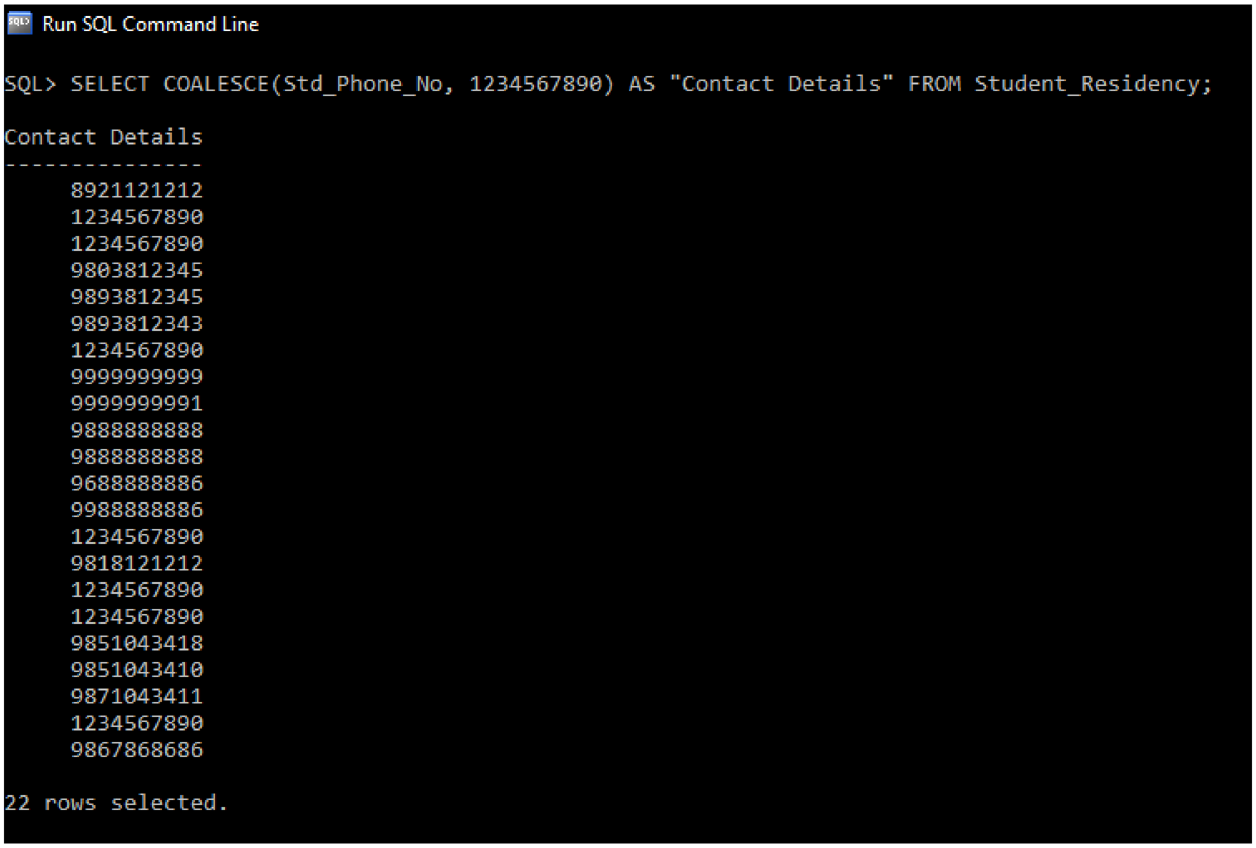
```
Run SQL Command Line
SQL> SELECT Student.Student_ID, Student.Std_Name, Course.Course_Name, Course.Specification_Name, Course.Course_Fees, CASE Course.Specification_Name
     WHEN 'Computing' THEN Course_Fees-Course_Fees*0.1
     ELSE Course_Fees
     END AS Reduced_Course_Fees
     FROM Course
     JOIN Student_Info on Course.Course_ID=Student_Info.Course_ID
     Join Student ON Student_Info.Student_ID=Student.Student_ID;

STUDENT_ID | STD_NAME        | COURSE_NAME     | SPECIFICATION_NAME   | COURSE_FEES | REDUCED_COURSE_FEES
---------- | --------------- | --------------- | -------------------- | ----------- | -------------------
         1 | Anup Shrestha   | BIT             | Computing            |      114000 |              102600
         2 | Azan Ahmed      | BIT             | Computing            |      114000 |              102600
         3 | Barsha Das      | BIT             | Computing            |      114000 |              102600
         4 | Bibek Paudyal   | BIT             | Computing            |      114000 |              102600
         5 | Biman Lakhey    | BIT             | Multimedia           |      114000 |              114000
         6 | Dipesh Shrestha | BIT             | Multimedia           |      114000 |              114000
         7 | Gyanu Adhikari  | BIT             | Multimedia           |      114000 |              114000
         8 | Rhythm          | BIT             | Multimedia           |      114000 |              114000
         9 | Sanket Kadel    | BIT             | Networking           |      114000 |              114000
        10 | Neha Bharati    | BIT             | Networking           |      114000 |              114000
        11 | Mimona Karki    | BIT             | Networking           |      114000 |              114000
        12 | Kishu Maharjan  | BIT             | Networking           |      114000 |              114000
        13 | Prajeet Kumar   | BIT             | Networking           |      114000 |              114000
        14 | Suyogya Luitel  | BIT             | Networking           |      114000 |              114000
        15 | Utsav Basyal    | BIT             | Networking           |      114000 |              114000
        16 | Prastut Paudel  | BBA             | Marketing            |      114000 |              114000
        17 | Roshan Adhikari | BBA             | Finance              |      114000 |              114000
        18 | Sumit Khatri    | BCA             | Computer Apps        |      114000 |              114000
        19 | Sahitya Rauniya | BIM             | Info. Management     |      114000 |              114000
        20 | Amit            | BA              | Arts                 |      114000 |              114000
        21 | Prism Koirala   | MSc             | Computer Science     |      114000 |              114000
        22 | Suraj Jung      | MBA             | Business             |      114000 |              114000

22 rows selected.
```

Figure 54: Transaction Query 1

In this query, Student ID, Student Name, Course Name, Specification Name, Course Fees has been selected for the student and course details from tables Student and Course. These tables were joined using INNER JOIN.

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

SELECT COALESCE(Std_Phone_No, 1234567890) AS "Contact Details" FROM Student_Residency;



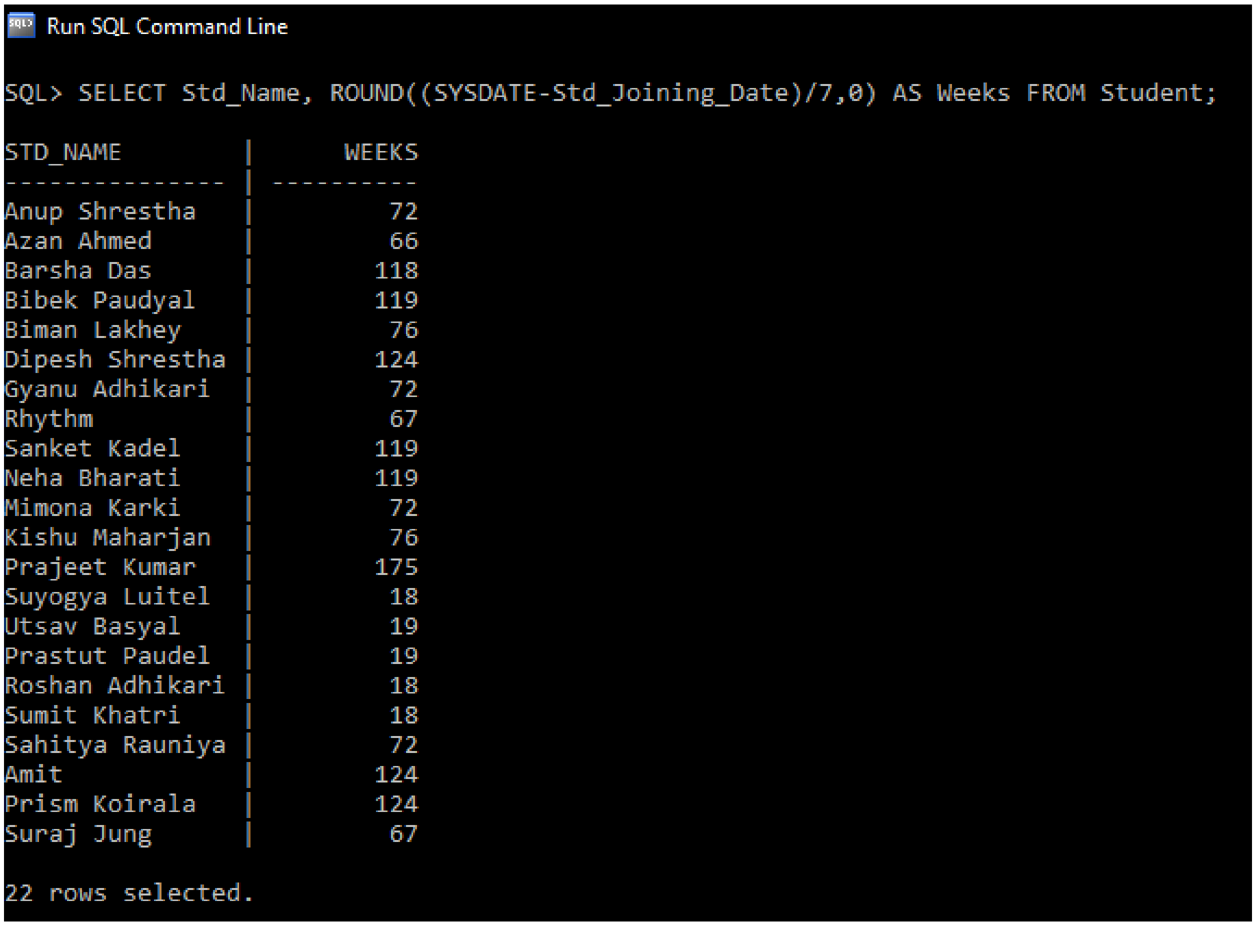
```
SQL> SELECT COALESCE(Std_Phone_No, 1234567890) AS "Contact Details" FROM Student_Residency;

Contact Details
---------------
     8921121212
     1234567890
     1234567890
     9803812345
     9893812345
     9893812343
     1234567890
     9999999999
     9999999991
     9888888888
     9888888888
     9688888886
     9988888886
     1234567890
     9818121212
     1234567890
     1234567890
     9851043418
     9851043410
     9871043411
     1234567890
     9867868686

22 rows selected.
```

Figure 55: Transaction Query 2

In this query, Instructor Phone No has been selected from Instructor Residency table and using the COALESCE() function, default number 1234567890 has been placed wherever the phone number to the location to the address is empty and the column has been named as ‘Contact Details’.

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

SELECT Std_Name, ROUND((SYSDATE-Std_Joining_Date)/7,0) AS Weeks FROM Student;



Run SQL Command Line

```
SQL> SELECT Std_Name, ROUND((SYSDATE-Std_Joining_Date)/7,0) AS Weeks FROM Student;

STD_NAME        |      WEEKS
--------------- | ----------
Anup Shrestha   |         72
Azan Ahmed      |         66
Barsha Das      |        118
Bibek Paudyal   |        119
Biman Lakhey    |         76
Dipesh Shrestha |        124
Gyanu Adhikari  |         72
Rhythm          |         67
Sanket Kadel    |        119
Neha Bharati    |        119
Mimona Karki    |         72
Kishu Maharjan  |         76
Prajeet Kumar   |        175
Suyogya Luitel  |         18
Utsav Basyal    |         19
Prastut Paudel  |         19
Roshan Adhikari |         18
Sumit Khatri    |         18
Sahitya Rauniya |         72
Amit            |        124
Prism Koirala   |        124
Suraj Jung      |         67

22 rows selected.
```

Figure 56: Transaction Query 3

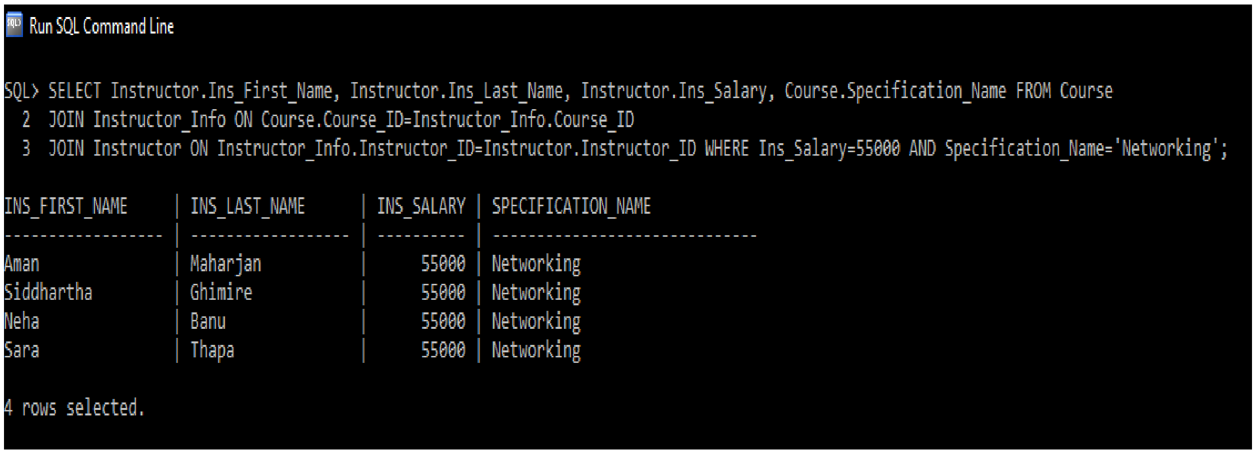
In this query, Student Name has been selected and ROUND() function has been used the calculate the no of weeks since they have enrolled in the course and that column has been names as ‘Weeks’.

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

SELECT Instructor.Ins_First_Name, Instructor.Ins_Last_Name, Instructor.Ins_Salary, Course.Specification_Name FROM Course

JOIN Instructor_Info ON Course.Course_ID=Instructor_Info.Course_ID

JOIN Instructor ON Instructor_Info.Instructor_ID=Instructor.Instructor_ID WHERE Ins_Salary=55000 AND Specification_Name='Networking';



```
Run SQL Command Line

SQL> SELECT Instructor.Ins_First_Name, Instructor.Ins_Last_Name, Instructor.Ins_Salary, Course.Specification_Name FROM Course
  2  JOIN Instructor_Info ON Course.Course_ID=Instructor_Info.Course_ID
  3  JOIN Instructor ON Instructor_Info.Instructor_ID=Instructor.Instructor_ID WHERE Ins_Salary=55000 AND Specification_Name='Networking';

INS_FIRST_NAME    | INS_LAST_NAME     | INS_SALARY | SPECIFICATION_NAME
----------------- | ----------------- | ---------- | ------------------------------
Aman              | Maharjan          |      55000 | Networking
Siddhartha        | Ghimire           |      55000 | Networking
Neha              | Banu              |      55000 | Networking
Sara              | Thapa             |      55000 | Networking

4 rows selected.
```

Figure 57: Transaction Query 4

In this query, Instructor First Name, Last Name, Salary has been selected from Instructor table and Specification name has been selected from Course table. WHERE clause has been used to display the desired output which is to show the name of the instrutors who get equal salary and work in the same specification. The tables have been joined using INNER JOIN.

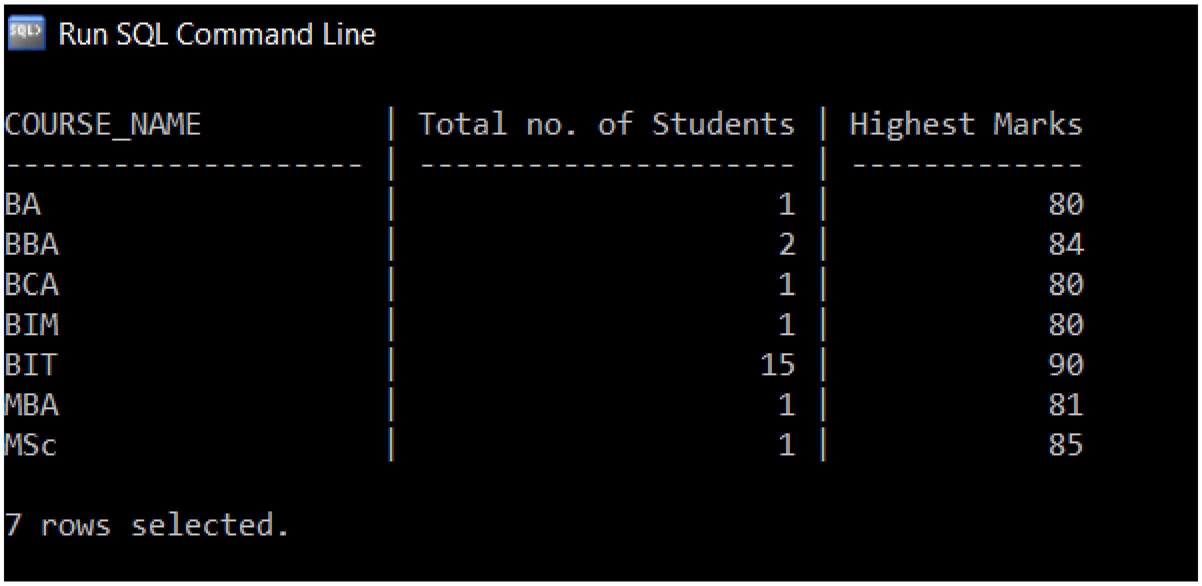
4.2.5. List all the courses with the total number of students enrolled course name and the highest marks obtained.

SELECT Course.Course_Name, COUNT(Student.Student_ID) AS "Total no. of Students", MAX(Student.Std_Marks) AS "Highest Marks" FROM Course

JOIN Student_Info ON Course.Course_ID=Student_Info.Course_ID

JOIN Student ON Student_Info.Student_ID=Student.Student_ID

GROUP BY Course.Course_Name ORDER BY Course.Course_Name;



```
Run SQL Command Line

COURSE_NAME          | Total no. of Students | Highest Marks
-------------------- | --------------------- | -------------
BA                   |                     1 |            80
BBA                  |                     2 |            84
BCA                  |                     1 |            80
BIM                  |                     1 |            80
BIT                  |                    15 |            90
MBA                  |                     1 |            81
MSc                  |                     1 |            85

7 rows selected.
```

Figure 58: Transaction Query 5

In this query, Course Name from has been selected from the Course table and using the COUNT() and MAX() function on the attributes Student ID and Marks respectively, the total no of students in a course and overall highest mark achieved in the course respectively has been displayed. The tables have been joined using INNER JOIN.

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

SELECT Ins_First_Name, Ins_Last_Name, Ins_Type FROM Instructor WHERE Ins_Type='Course Leader';

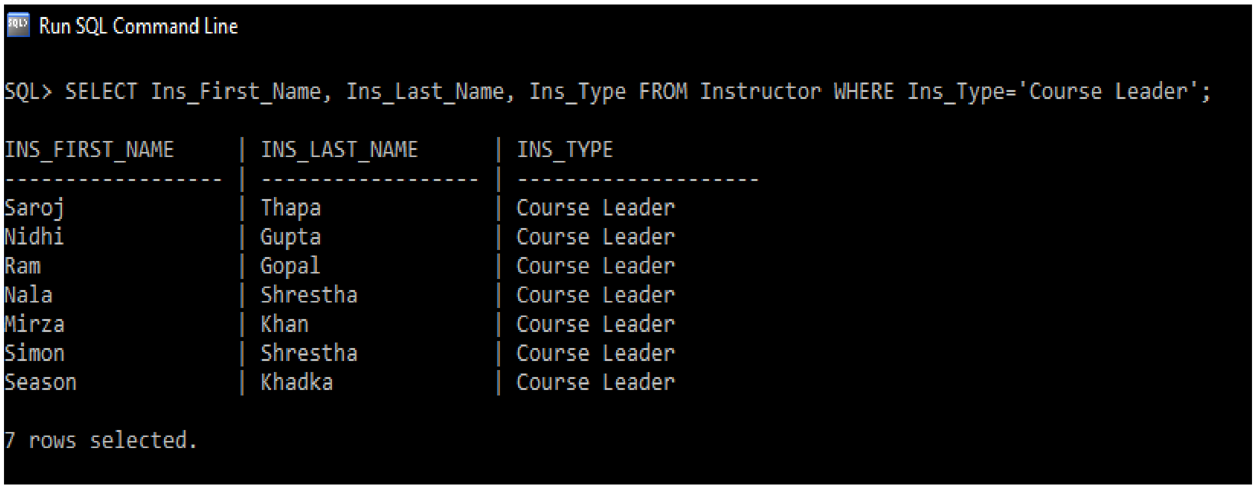
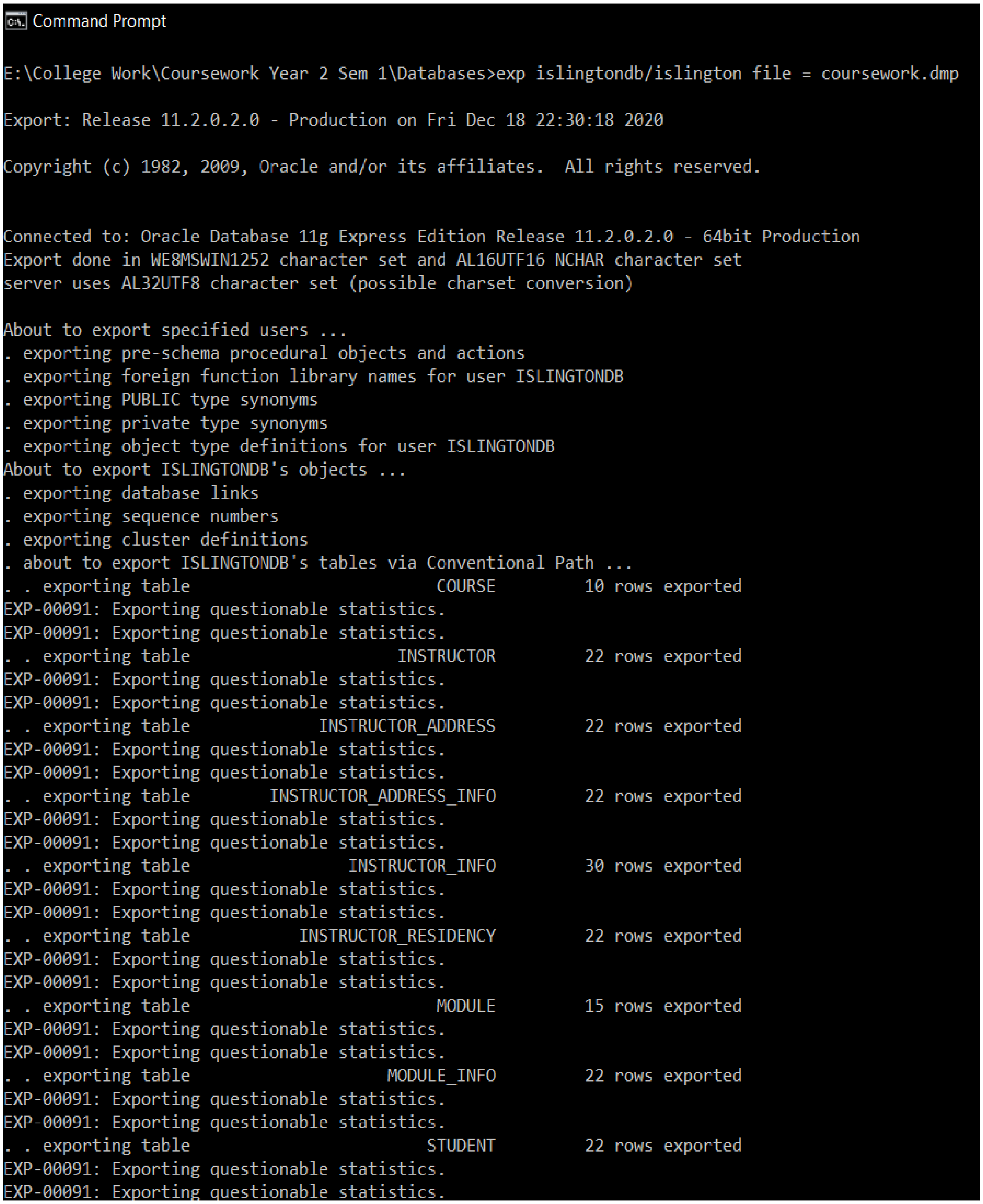

```
Run SQL Command Line

SQL> SELECT Ins_First_Name, Ins_Last_Name, Ins_Type FROM Instructor WHERE Ins_Type='Course Leader';

INS_FIRST_NAME     | INS_LAST_NAME      | INS_TYPE
------------------ | ------------------ | --------------------
Saroj              | Thapa              | Course Leader
Nidhi              | Gupta              | Course Leader
Ram                | Gopal              | Course Leader
Nala               | Shrestha           | Course Leader
Mirza              | Khan               | Course Leader
Simon              | Shrestha           | Course Leader
Season             | Khadka             | Course Leader

7 rows selected.
```

Figure 59: Transaction Query 6

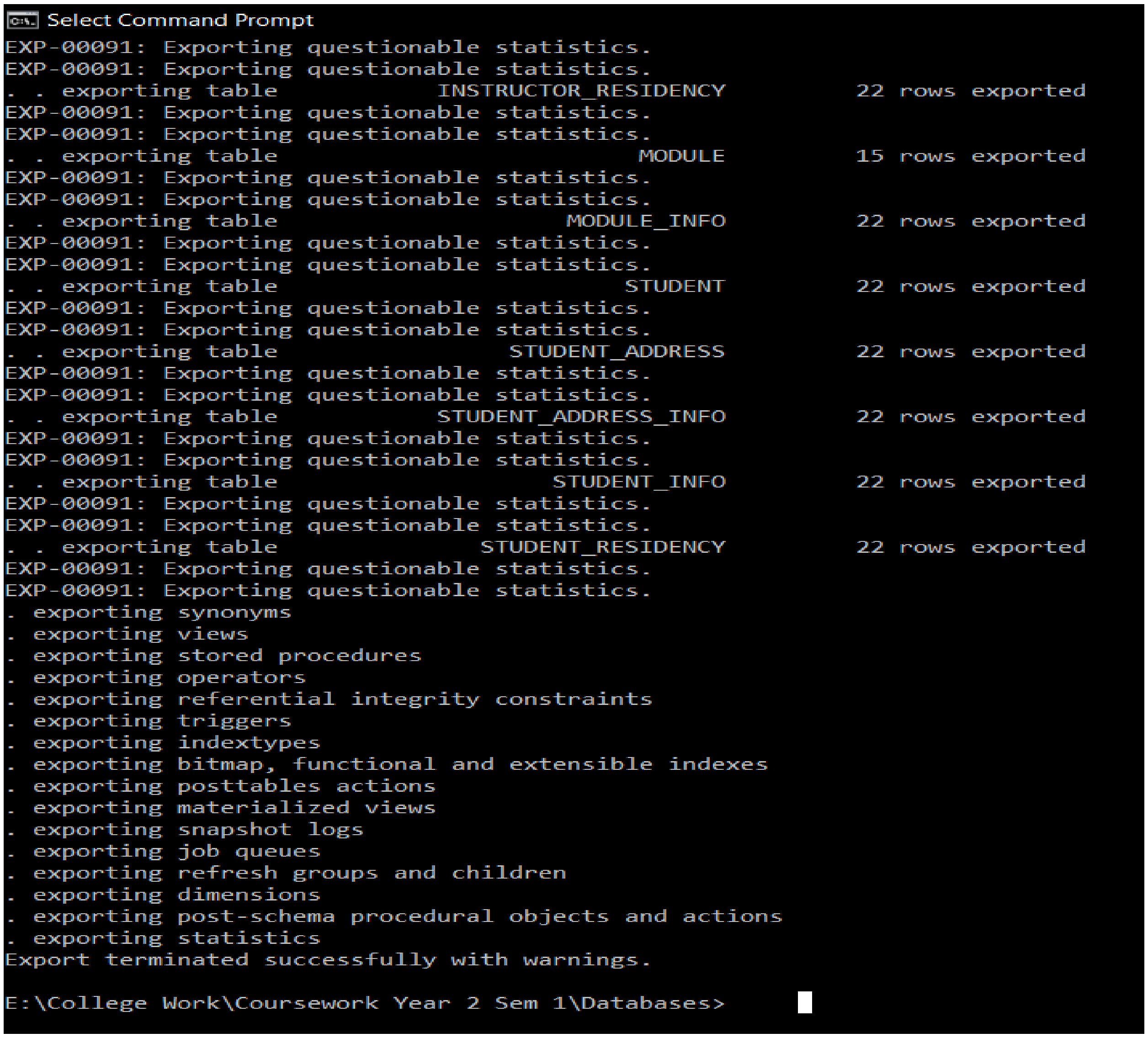
In this query, Instructor First Name, Instructor Last Name and Instructor Type has been selected from the Instructor table. To display the desired output, WHERE clause has been used in which the Instructor type has been set to 'Course Leader' to only display the names of the course leaders.

4.3. Creation of Dump File



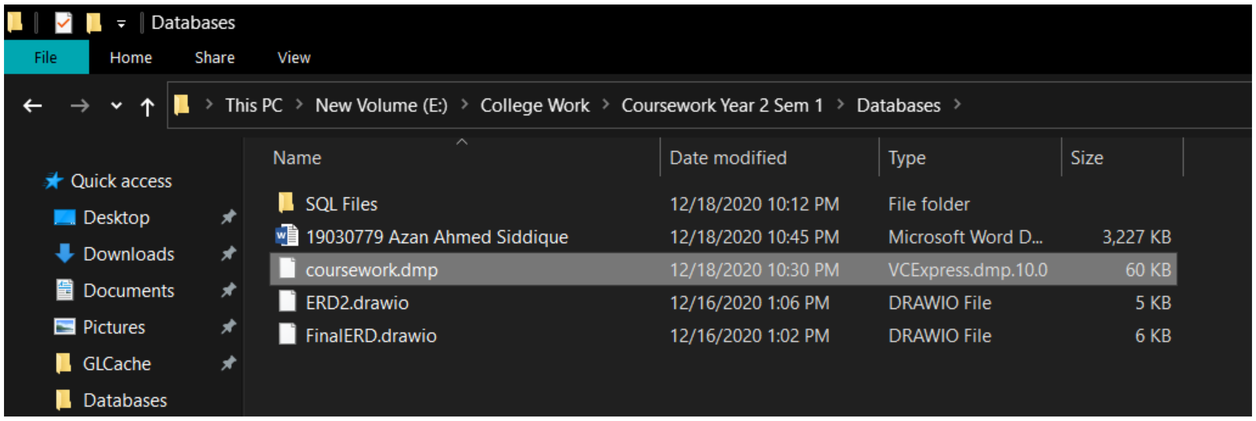
Command Prompt

```
E:\College Work\Coursework Year 2 Sem 1\Databases>exp islingtondb/islington file = coursework.dmp

Export: Release 11.2.0.2.0 - Production on Fri Dec 18 22:30:18 2020

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


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

About to export specified users ...
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user ISLINGTONDB
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user ISLINGTONDB
About to export ISLINGTONDB's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export ISLINGTONDB's tables via Conventional Path ...
. . exporting table                         COURSE         10 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                     INSTRUCTOR         22 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table             INSTRUCTOR_ADDRESS         22 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table        INSTRUCTOR_ADDRESS_INFO         22 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                INSTRUCTOR_INFO         30 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table           INSTRUCTOR_RESIDENCY         22 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                         MODULE         15 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                    MODULE_INFO         22 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                        STUDENT         22 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
```



```
Select Command Prompt
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table            INSTRUCTOR_RESIDENCY         22 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                          MODULE         15 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                     MODULE_INFO         22 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                         STUDENT         22 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                 STUDENT_ADDRESS         22 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table            STUDENT_ADDRESS_INFO         22 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table                    STUDENT_INFO         22 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table               STUDENT_RESIDENCY         22 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting operators
. exporting referential integrity constraints
. exporting triggers
. exporting indextypes
. exporting bitmap, functional and extensible indexes
. exporting posttables actions
. exporting materialized views
. exporting snapshot logs
. exporting job queues
. exporting refresh groups and children
. exporting dimensions
. exporting post-schema procedural objects and actions
. exporting statistics
Export terminated successfully with warnings.

E:\College Work\Coursework Year 2 Sem 1\Databases>
```

Figure 60: Creating the dump file



Name	Date modified	Type	Size
SQL Files	12/18/2020 10:12 PM	File folder	
19030779 Azan Ahmed Siddique	12/18/2020 10:45 PM	Microsoft Word D...	3,227 KB
coursework.dmp	12/18/2020 10:30 PM	VCExpress.dmp.10.0	60 KB
ERD2.drawio	12/16/2020 1:06 PM	DRAWIO File	5 KB
FinalERD.drawio	12/16/2020 1:02 PM	DRAWIO File	6 KB

Figure 61: Screenshot of the actual dump file

4.4. Drop Tables

DROP TABLE Course;

DROP TABLE Module_Info;

DROP TABLE Module;

DROP TABLE Instructor_Info;

DROP TABLE Instructor;

DROP TABLE Student_Info;

DROP TABLE Student;

DROP TABLE Instructor_Address_Info;

DROP TABLE Instructor_Address;

DROP TABLE Instructor_Residency;

DROP TABLE Student_Address_Info;

DROP TABLE Student_Address;

DROP TABLE Student_Residency;

5. Conclusion

While doing this coursework, I faced many difficulties, especially in the early parts of the coursework where I needed to further improve my understanding of Entities and attributes and the relationships that are formed between them to properly understand the question posed in the coursework and identify the entities and attributes given in the coursework case study.

The coursework required the student to first identify the entities and attributes, create an initial ERD and then normalation the relation till 3nf (third normal form), create a final ERD and implement the database and do database querying and then create sql files of the queries and create the dump file of the database.

In the coursework, first I had to do the introduction part where I had to introduce the college, write what current business activities that are carried out in the college and also write the business rules followed the college. After that, I had to identify the entities and attributes based on the scenario given in the coursework and create an initial ER diagram based on that. The initial ER diagram had a lot of data redundancies and a many to many relation which had to be reduced by normalizing the relation till 3nf. After carrying out the normalization, I created a final ER diagram with the the entities and attributes formed in the third normalization form. Both the initial ERD and the final ERD was creating in draw.io. Once the final ERD was created, I moved on to to implementing all of that in a database. For this, I used the Oracle SQL, which is one of the most quickest and secure way to create a database. First, I created all of the tables and inserted values based on the queries provided in the question. After inserting all of the necessary data into the tables, I had to do the information queries and transaction queries provided in the question. Once the information queries were completed, I created the sql files for the information queries and transaction queries using the SPOOL command and then finally created a dump file of the database using the Command Prompt (CMD).

After completing this coursework, I feel confident that I can create a database which stores records data in a structured manner with as few redundancies as possible and extremely easy to manipulate and maintain. Since the coursework was about creating a database record system for a college, it has lots of applicability in the real world as most colleges and schools use a database to record data of everyone and everthing associated with them.

Overall, this was a very challenging but fun coursework to do. I had to do a lot of research and seek guidance from my teachers to understand some of the core concepts that I had difficulties with. My knowledge on SQL and DBMS as a whole has improved significantly. I’ve learned about how to identify entities and attributes, create ER diagrams, identify repeating group in a table and reduce any sort of dependencies as much as possible through normalization, database implementation, creating and altering table and inserting values into a table or updating it or even deleting it, database quering, creating sql files and dump file of the database. I believe everything that I’ve learned while doing this coursework will be a big help in my career.

References

File Maker. (2020, December 13). *File Maker*. Retrieved from File Maker : https://fmhelp.filemaker.com/help/18/fmp/en/index.html#page/FMP_Help/many-to-many-relationships.html

GeeksforGeeks. (2020, December 4). *GeeksforGeeks*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/third-normal-form-3nf/

GeeksforGeeks. (2020, December 4). *GeeksforGeeks*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/second-normal-form-2nf/

GeeksforGeeks. (2020, December 4). *GeeksforGeeks*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/first-normal-form-1nf/

Gitta. (2020, December 4). *Gitta*. Retrieved from Gitta: http://www.gitta.info/LogicModelin/en/html/DataConsiten_Norm3NF.html

Grussells. (2020, December 2). *Grussells*. Retrieved from Grussells: https://db.grussell.org/section005.html#:~:text=A%20fan%20trap%20occurs%20when,out%20from%20a%20single%20entity.&text=The%20fan%20trap%20is%20resolved,to%20represent%20the%20correct%20association.

Guru99. (2020, December 4). *Guru99*. Retrieved from Guru99: https://www.guru99.com/database-normalization.html

IBM. (2020, 12 2). *IBM*. Retrieved from IBM: https://www.ibm.com/support/knowledgecenter/SSWSR9_11.6.0/com.ibm.mdmhs.overview.doc/entityconcepts.html

Lucid Chart. (2020, December 2). *Lucid Chart*. Retrieved from Lucid Chart: https://www.lucidchart.com/pages/er-diagrams

Lucid Chart. (2020, December 12). *Lucid Chart*. Retrieved from Lucid Chart: https://www.lucidchart.com/pages/er-diagrams

Oracle Tutorial. (2020, December 4). *Oracle Tutorial*. Retrieved from Oracle Tutorial: https://www.oracletutorial.com/oracle-basics/oracle-create-table/

PcMag. (2020, December 4). *PcMag*. Retrieved from PcMag: https://www.pcmag.com/encyclopedia/term/transitive-dependency#:~:text=An%20indirect%20relationship%20between%20data,B%20not%2D%3EA).

Programiz. (2020, November 30). *Programiz*. Retrieved from Programiz: https://www.programiz.com/java-programming

SQL Server. (2020, December 12). *SQL Server*. Retrieved from SQL Server: https://www.sqlservertutorial.net/sql-server-basics/sql-server-create-table/

Tech on the net. (2020, December 12). *Tech on the net*. Retrieved from Tech on the net: https://www.techonthenet.com/oracle/tables/alter_table.php

Tech on the net. (2020, December 12). *Tech on the net*. Retrieved from Tech on the net: https://www.techonthenet.com/sql/insert.php

Techopedia. (2020, December 4). *Techopedia*. Retrieved from TechoPedia: https://www.techopedia.com/definition/25955/first-normal-form-1nf

TechoPedia. (2020, December 4). *TechoPedia*. Retrieved from TechoPedia: https://www.techopedia.com/definition/21980/second-normal-form-2nf

Visual Paradigm. (2020, 12 2). *Visual Paradigm*. Retrieved from Visual Paradigm: https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/#:~:text=Entity%20Relationship%20Diagram%2C%20also%20known,inter%2Drelationships%20among%20these%20entities.

w3school. (2020, December 12). *w3school*. Retrieved from w3school: https://www.w3schools.com/sql/sql_primarykey.ASP

Xspdf. (2020, December 5). *Xspdf*. Retrieved from Xspdf: https://www.xspdf.com/resolution/52268859.html#:~:text=A%20transitive%20dependency%20in%20a,must%20eliminate%20any%20transitive%20dependency.