# CC5051NI Databases

## 100% Individual Coursework

## Autumn 2024

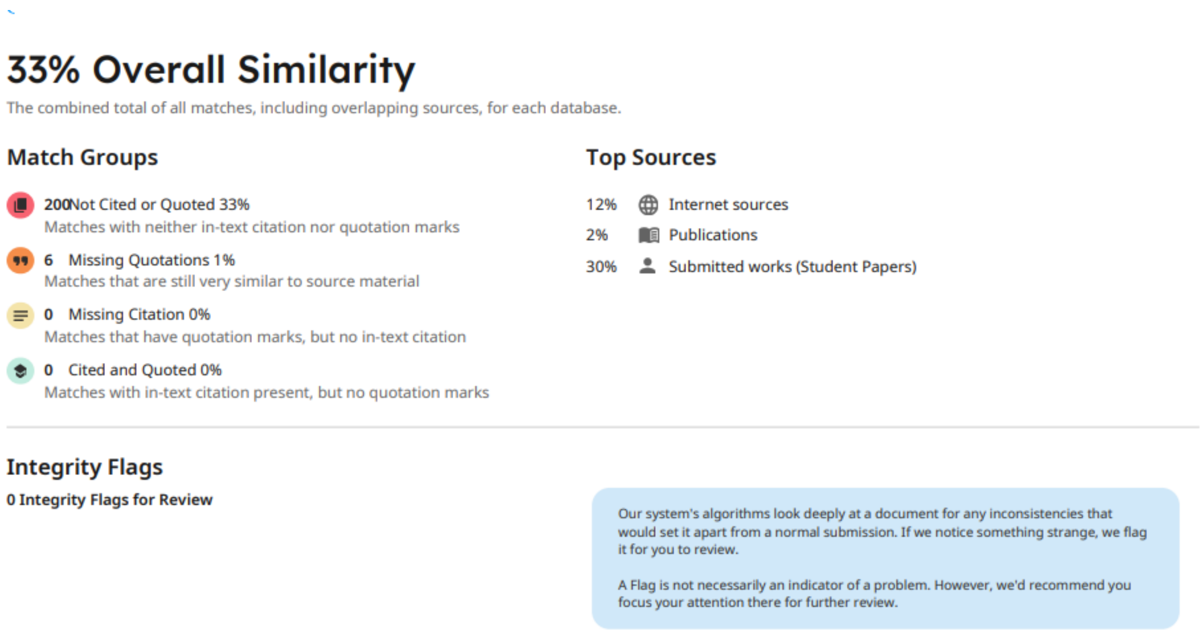
## Credit: 15 Semester Long Module

**Student Name: Saroj Babu Karki**

**London Met ID: 22067792**

**Assignment Submission Date: January 23, 2025**

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



## 33% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

### Match Groups

**200**Not Cited or Quoted 33%
Matches with neither in-text citation nor quotation marks

**6** Missing Quotations 1%
Matches that are still very similar to source material

**0** Missing Citation 0%
Matches that have quotation marks, but no in-text citation

**0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

### Top Sources

| | |
|---|---|
| 12% | Internet sources |
| 2% | Publications |
| 30% | Submitted works (Student Papers) |

### Integrity Flags

**0 Integrity Flags for Review**

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

# Table of Contents

## Table of Figures

# Table of Tables

## 1.0 Introduction

Stark College was established in 2017 with a mission to provide the latest and updated education that aims to foster personal development, intellectual well-being, and global recognition. For the past six years, this institute has built a strong reputation for academic excellence and holistic development which is guided by its core values of innovation, inclusivity, integrity, and community-related services. The college offers various programs related to Information Technology that have been designed to empower students with essential skill sets and knowledge to thrive in an evolving world. With better infrastructures and experienced faculty members dedicated to respective programs continuously explore to create an environment that inspires students in creativity, collaboration, and learning.

Ms. Mary, the visionary entrepreneur, founder, and principal of Stark College has a keen interest in building it as a successful institute. With, years of experience in the field of education and a passion for transformative learning, she has dedicated her career goal to nurture young minds and encourage new innovations in the educational field. Her excellent leadership and commitment have helped the institution adopt a modern approach to education. She reflects it by introducing a new approach to teaching, and learning activities by introducing an “E-classroom platform” for students and teachers. This platform aims to offer a seamless blend of traditional methods of teaching and learning with digital tools to enhance learning outcomes and accessibility.

With Ms. Mary’s dedication, Stark College continues to create new educational experiences and pioneers in embracing the latest teaching and learning methodologies.

## 1.1 Current Business Activities and Operations

Stark College aims to deliver education with a focus on dynamic academic and extracurricular activities, fostering young minds. The institute operates on the basis of a student-centered approach where students are enrolled in various programs according to their interests to focus on skill development and academic excellence. The programs include various categories including Computing, Multimedia, Application Development, Networking and Cybersecurity, Artificial Intelligence and so on. This institute takes pride in providing the latest updated course materials to its students. They also aim to provide an excellent teaching-learning environment to their students for better understanding by conducting interactive lectures, tutorials, and workshop sessions. They allocate teachers according to their specialization to minimize unnecessary conflicts and focus on the primary goal of providing the best education. Each modules are interconnected between different programs. The faculty is responsible for providing teaching and learning resources and making students capable of completing the assessments assigned to them by accessing resources provided by teachers from the respective module. Each resources have to be completed by students thoroughly for progressive learning and students should complete it in a sequential order to access other resources. The teacher announces their announcement through respective channels to provide important information to students. Assessment results are published after proper evaluation. With a focus on emphasizing better education, “Stark College” put its priorities on the e-classroom platform towards modernization of education system.

## 1.2 Business Rules

Business Rules are a set of rules and regulations that are vital for achieving organizational objectives and ensuring consistency in daily enterprise operational activities. While business policies provide general information about guidelines, business rules impose specific constraints on the behavior and structure of the business schema. By focusing on various constraints, conceptual database design can be enhanced to meet system consistency, efficiency, and adaptability. It also sets a framework to create model-based cardinality constraints and integrate them into existing design concept methodologies. It uses formal semantics that define rules for attributes, participation, relationships, and appearance that abides by a database framework. This also aids in system maintenance, operational consistency, and query optimization. (Ram & Khatri, 2005)

Stark College has set its specific business rules to conduct and manage its daily activities and operations which are listed below:

- ✓ A student can enroll exactly in one program and be associated with it, but a program can have multiple students enrolled in one program.

- ✓ A program should consist of multiple modules across different programs for a flexible course structure. Similarly, modules can be associated with multiple programs that are listed in college.

- ✓ Each teacher is responsible for teaching one or more specific modules, but each module is taught by only one teacher.

- ✓ Each module have multiple assessments assigned to students, but each assessments belong to a particular module only.

- ✓ Each assessment can have multiple results. One result for each student who has attempted the assessment. A result is associated with a single assessment.

- ✓ Each student can have multiple results corresponding to various assessments, but a specific result is associated with a specific student.

- ✓ Each module have multiple resources, but each resource is linked to a single module. Other resources are available to students if the students complete previous resources provided for the progressive learning approach.

- ✓ Each teacher from specific module can publish multiple announcements through module-specific channels , but each announcement is linked to a single module.

## 1.3 Assumptions

According to the business rules, following assumptions are listed below:

- ✓ Modules are shared across programs and can be altered or customized as per the policies of the college to align with program specific objectives.

- ✓ A student can enroll in one program at a time. No dual enrollment is allowed in enrollment scenario in this structure.

- ✓ A teacher can be assigned for teaching multiple modules, but one module of a specific program at a time.

- ✓ Assessments within a module is scheduled at specific times that are mandatory for enrolled students in a module.

- ✓ Each result published should include information related to the students and assessments along with obtained marks, grades, status and feedback.

- ✓ Student can access module-specific resources if they are enrolled in the module of a specific program.

- ✓ Announcements are scheduled, categories and visible only to students and teachers associated with specific module.

- ✓ Resources are unlocked sequentially, Student must complete previous resources to gain access over new resources published for the specific module.

- ✓ Student can attempt an assessment once unless permitted by the teacher or guidelines mentioned in program policies of the college.

- ✓ Results have feedback support to provide guidance and clarifications to students for identifying strength and weakness of a particular student.

- ✓ Modules within a program have a requirement for students to complete one module before proceeding to another module.

- ✓ Teacher assigned to modules must be available for module evaluations for the program durations.

## 2.0 Initial Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) is a visual representation of a data model that shows entities, attributes, and relationships between entities in a database modeling phase. ERDs are used mostly in database design and act as a blueprint for understanding the structure and design of a database model. (Chen, 2009)

### Entity

Entities are concepts or objects that can be identified and described in a real-world environment. In a database, entities are represented in the form of tables. Each column is identified as attributes whereas each row is represented as entities. (Visual Paradigm, 2024)

### Attributes

Attributes are characteristics or properties included in an entity. It describes the data which can be stored for each entity. They are typically represented in the form of tables where data will be inserted and stored.

### Relationships

Relationships define relations between entities. Various relationships exist between entities, such as one-to-one, one-to-many, and many-to-many. Relationships are represented by drawing and connecting lines between the related entities and they often have labels to indicate type of relationship.

| Entity | Attributes |
|---|---|
| **Student** | Student_ID(PK), Student_Name, Student_Address, Student_Phone, Student_Email, Student_Date_Of_Birth, Student_Enrollment_Date, Program_ID, Program_Title, Program_Duration, Program_Description, Result_ID, Result_Marks, Result_Grade, Result_Status, Result_Feedback |
| **Module** | Module_ID(PK), Module_Title, Module_Credit_Hours, Module_Duration, Assessment_ID, Assessment_Title, Assessment_Type, Assessment_Weightage, Assessment_Deadline, Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence |
| **Teacher** | Teacher_ID(PK), Teacher_Name, Teacher_Specialization, Teacher_Phone, Teacher_Email, Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date |

*Table 1 List of entities and attributes*

## 2.1 Identification of Entity and Attributes for E-classroom Platform

### 2.1.1 Student

| S.N. | Attribute Name | Data Type | Size | Constraint |
|---|---|---|---|---|
| 1. | Student_ID | Number | 15 | Primary Key |
| 2. | Student_Name | Character | 20 | Not Null |
| 3. | Student_Address | Character | 20 | Not Null |
| 4. | Student_Phone | Character | 20 | Unique |
| 5. | Student_Email | Character | 30 | Unique |
| 6. | Student_Date_Of_Birth | Date | - | Not Null |
| 7. | Student_Enrollment_Date | Date | - | Not Null |
| 8. | Program_ID | Number | 15 | Unique |
| 9. | Program_Title | Character | 35 | Not Null |
| 10. | Program_Duration | Character | 20 | Not Null |
| 11. | Program_Description | Character | 60 | Not Null |
| 12. | Result_ID | Number | 15 | Unique |
| 13. | Result_Marks | Number | 15 | Not Null |
| 14. | Result_Grade | Character | 15 | Not Null |
| 15. | Result_Status | Character | 15 | Not Null |
| 16. | Result_Feedback | Character | 65 | Not Null |

*Table 2 Identification of entities and attributes for Student*

### 2.1.2 Module

| S.N. | Attribute Name | Data Type | Size | Constraint |
|---|---|---|---|---|
| 1. | Module_ID | Number | 15 | Primary Key |
| 2. | Module_Title | Character | 30 | Not Null |
| 3. | Module_Credit_Hours | Number | - | Not Null |
| 4. | Module_Duration | Character | 20 | Not Null |
| 5. | Assessment_ID | Number | 15 | Unique |
| 6. | Assessment_Title | Character | 30 | Not Null |
| 7. | Assessment_Type | Character | 30 | Not Null |
| 8. | Assessment_Weightage | Character | 25 | Not Null |
| 9. | Assessment_Deadline | Date | - | Not Null |
| 10. | Resource_ID | Number | 15 | Unique |
| 11. | Resource_Title | Character | 40 | Not Null |
| 12. | Resource_Type | Character | 20 | Not Null |
| 13. | Resource_Status | Character | 20 | Not Null |
| 14. | Resource_Sequence | Number | - | Not Null |

*Table 3 Identification of entities and attributes for Module*

### 2.1.3 Teacher

| S.N. | Attribute Name | Data Type | Size | Constraint |
|---|---|---|---|---|
| 1. | Teacher_ID | Number | 15 | Primary Key |
| 2. | Teacher_Name | Character | 25 | Not Null |
| 3. | Teacher_Specialization | Character | 40 | Not Null |
| 4. | Teacher_Phone | Character | 20 | Not Null |
| 5. | Teacher_Email | Character | 30 | Not Null |
| 6. | Announcement_ID | Number | 15 | Unique |
| 7. | Announcement_Title | Character | 40 | Not Null |
| 8. | Announcement_Description | Character | 120 | Not Null |
| 9. | Announcement_Post_Date | Date | - | Not Null |

*Table 4 Identification of entities and attributes for Teacher*

## 2.2 Initial Entity Relationship Diagram for E-classroom platform

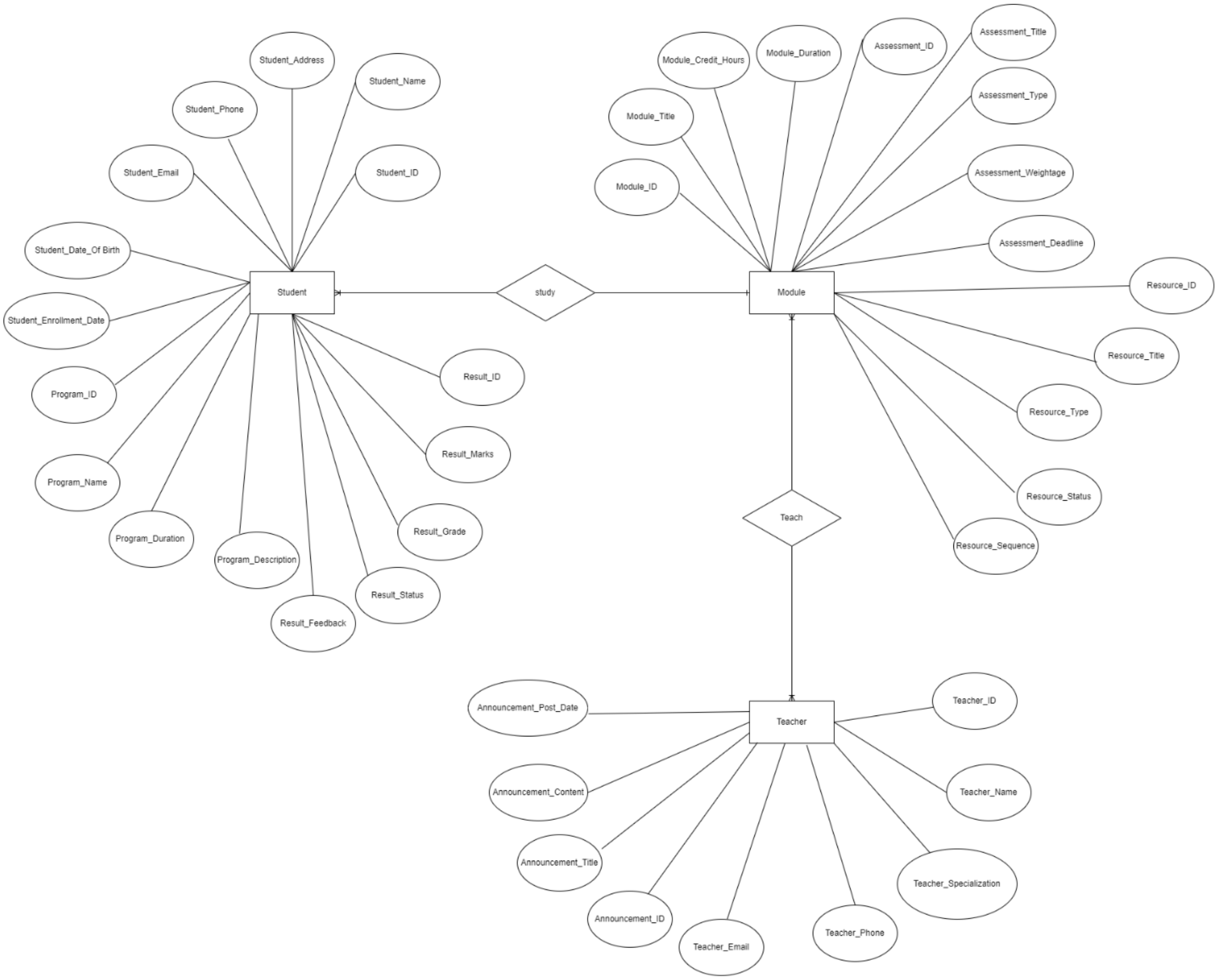


*Figure 1: Initial Entity Relationship Diagram (ERD)*

## 3.0 Normalization

Normalization is a process of breaking down complex tables into simpler forms. It is a formalized set of guidelines that help to reduce data anomalies and redundancies. It also assist in solving problems caused due to unwanted dependencies and redundancies in a database. To tackle and find solutions to these problems normalization in database was used. (Vinita, 30-06-2020)

In our E-classroom platform there were various issues of data redundancies and anomalies. Lack of proper normaliazation have caused various issues while creating tables in a database. So, we have proposed normalization process to tackle hose issues. Furthermore, normalization is categorized to UNF, 1NF, 2NF. 3NF, 4NF and 5NF. But, for our E-classroom database, we are going to do UNF, 1NF, 2NF and 3NF as proposed in case study.

## 3.1 Un-Normalized Form (UNF)

A table is considered to be in Un-Normalized Form (UNF) if it is not organized in a proper relational structure. This creates multivalued attributes with repeating groups. Mutivalued attributes have multiple values stored in a single field. At the same time repeating groups replicate data across columns in a table. This structure leads to various issues such as data inconsistencies, redundancies, and anomalies.

### 3.1.1 Rules for Un-Normalized Form (UNF)

When the tables are listed in Un-Normalized Form (UNF), the following details are enlisted in table that are mentioned below:

- ✓ Data in Un-Normalized Form (UNF) does not follow relational structure and is not organized in the form of proper tables.

- ✓ Attributes in a table contains multivalued data in a single field.

- ✓ Similar data are stored across tables due to the presence of repeating groups.

- ✓ Tables in Un-Normalized form do not have unique identifier.

- ✓ Attributes do not hold indivisible data in a table which causes various insertion, update, and deletion anomalies while storing data in tables.

### 3.1.2 Un-Normalized Form (UNF) Process

To list the tables in Un-Normalized Form (UNF) a group of attributes are listed and repeating groups are identified. Repeating groups are enclosed within a bracket for easier identification of repeating data in tables. Each group created is linked to one entity and tables are listed and enclosed with brackets so that the tables can be separated individually in further process of normaliazation.

**Student** → ( Student_ID, Student_Name, Student_Address Student_Email, Student_Phone, Student_Date_Of_Birth, Student_Enrollment_Date, Program_ID, Program_Title, Program_Duration, Program_Description, { Module_ID, Module_Title, Module_Credit_Hours, Module_Duration, { Teacher_ID, Teacher_Name, Teacher_Specialization, Teacher_Phone, Teacher_Email{Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date} } {Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence } { Assessment_ID, Assessment_Title, Assessment_Type, Assessment_Weightage, Assessment_Deadline, Result_ID, Result_Marks, Result_Grade, Result_Status, Result_Feedback } } )

In the Un-Normalized Form (UNF) structure, Student is the main entity and other details listed in brackets are attributes. The repeating groups include Module details, Teacher details, Announcement details, Resource details, Assessment details, and Result details are enclosed in curly brackets { }. This allows us to identify repeating groups and break into individual tables with unique identifiers.

## 3.2 First Normal Form (1NF)

A table is in First Normal Form (1NF) when the table meets the basic requirements of a relational structure. A unique identifier is defined so that it can identify each row uniquely, ensuring data integrity. The repeating groups in a table is separated and restructured into separate rows and tables with unique identifiers. This makes the tables well-organized and eliminates risks related to redundancy and anomalies.

### 3.2.1 Rules for First Normal Form (1NF)

While separating the Un-Normalized Form (UNF) to First Normal Form (1NF), the following details should be considered in the normalization process:

- ✓ Attributes must contain only one value. Multiple list or values are not allowed to be listed in single row.
- ✓ Each row should be unique and identified with the help of unique identifier primary key.
- ✓ Repeating groups are eliminated by separating them into different tables with its unique identifiers.
- ✓ Each column in a table must have respective data types identified as text, number, date, etc respectively.
- ✓ Duplication of rows and columns in a table is checked and eliminated.

### 3.2.2 First Normal Form (1NF) Process

Tables in First Normal Form (1NF) are listed by separating repeating groups in a table by identifying the unique identifier primary key. Primary Keys are underlined and foreign keys are identified with the '*' symbol. Based on the Un-Normalized Form (UNF) the tables are listed in First Normal Form (1NF) by following the normalization principles of First Normal Form (1NF).

**Final First Normal Form (1NF) Tables**

**Student – 1** → (Student_ID, Program_ID* Student_Name, Student_Address, Student_Email, Student_Phone, Student_Date_Of_Birth)

**Program – 1** → (Program_ID, Program_Title, Program_Duration, Program_Description)

**Module – 1** → (Module_ID, Program_ID*, Module_Title, Module_Credit_Hours, Module_Duration)

**Teacher – 1** → (Teacher_ID, Module_ID*, Teacher_Name, Teacher_Specialization, Teacher_Phone, Teacher_Email)

**Announcement – 1** → (Announcement_ID, Module_ID*, Teacher_ID* Announcement_Title, Announcement_Content, Announcement_Post_Date)

**Resources – 1** → (Resource_ID, Module_ID*, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence)

**Assessment – 1** → (Assessment_ID, Module_ID*, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type)

**Result – 1** → (Result_ID, Student_ID*, Assessment_ID*, Result_Grade, Result_Marks, Result_Status, Result_Feedback)

Here, all the tables follow the principles of the First Normal Form (1NF). Each tables are separated by eliminating repeating columns. The unique identifiers are assigned to each tables as well. This helps to eliminate the issues related to data inconsistencies, redundancies and anomalies making it practical to store data.

## 3.3 Second Normal Form (2NF)

Tables listed in Second Normal Form (2NF) should have full functional dependencies and partial functional dependencies should not exist. If any partial functional dependencies exists then it is removed.

A functional dependency is defined as the relationship between attributes in a relational table. A functionally determines attributes in B, if each value of A is associated exactly with one value B (A → B).

A partial dependency exist when a non-prime attribute functionally dependent on composite primary key instead of whole primary key.

### 3.3.1 Rules for Second Normal Form (2NF)

In Second Normal Form (2NF), the following details should be considered while listing tables:

- ✓ Each tables should already be in First Normal Form (1NF).

- ✓ A non-prime attribute must be fully functionally dependent on whole primary key.

- ✓ If any partial dependencies are found, it should be eliminated by splitting into new tables focusing on full functional dependencies.

### 3.3.3 Second Normal Form (2NF) Process

The tables listed in the First Normal Form are free from repeating groups. But, the tables may include partial functional dependency that violates the principles of Second Normal Form (2NF). To identify and eliminate any partial functional dependency and make sure there is full functional dependency the tables are checked, and dependencies are identified and resolved properly to satisfy the principles of Second Normal Form (2NF).

The tables are checked for full functional dependencies and partial functional dependencies and listed below:

- ✓ **Student_ID → X** (Full Functional Dependency)

  No partial functional dependencies exist in **Student table** since all non-key attributes are fully dependent on primary key **Student_ID**.

- ✓ **Program_ID → X** (Full Functional Dependency)

  No partial functional dependencies exist in **Program table** since all non-key attributes are fully dependent on primary key **Program_ID**.

  **Module_ID → Module_Title** (Partial Functional Dependency)

Partial functional dependency exist since non-key attribute **Module_Title** depends on primary key **Module_ID**.

**Program_ID, Module_ID → X** (Full Functional Dependency)

No non-key attributes are dependent on the combination of composite keys **Program_ID** and **Module_ID**.

✓ **Module_ID → X** (Full Functional Dependency)

No partial dependencies exist since all non-key attributes in **Module Table** are dependent on **Module_ID**.

✓ **Teacher_ID → Teacher_Name** (Partial Functional Dependency)

Partial dependency exist since non-key attribute **Teacher_Name** is dependent on **Teacher_ID** in **Teacher Table**.

**Module_ID → Module_Title** (Partial Functional Dependency)

Partial functional dependency exist since non-key attribute **Module_Title** depends on primary key **Module_ID**.

**Teacher_ID, Module_ID → X** (Full Functional Dependency)

No non-key attributes are dependent on the combination of composite keys **Teacher_ID** and **Module_ID**.

✓ **Announcement_ID → X** (Full Functional Dependency)

No partial dependencies exist since all non-key attributes in **Announcement Table** are dependent on **Announcement_ID**.

✓ **Resource_ID → X** (Full Functional Dependency)

No partial dependencies exist since all non-key attributes in **Resources Table** are dependent on **Rsource_ID**.

- ✓ **Student_ID →  X** (Full Functional Dependency)

  No partial functional dependencies exist in **Student table** since all non-key attributes are fully dependent on primary key **Student_ID**.

  **Assessment_ID →  Assessment_Title** (Partial Functional Depenency)

  Partial functional dependency exist since non-key attribute **Assessment_Title** depends on primary key **Assessment_ID**.

  **Assessment_ID, Student_ID →  X** (Full Functional Dependency)

  No non-key attributes are dependent on the combination of composite keys **Assessment_ID** and **Student_ID**.

- ✓ **Result_ID →  X** (Full Functional Dependency)

  No partial dependencies exist since all non-key attributes in **Result Table** are dependent on **Result_ID**.

**<u>Final Second Normal Form (2NF) Tables</u>**

**Student – 2** → (<u>Student_ID</u>, Student_Name, Student_Address, Student_Email, Student_Phone, Student_Date_Of_Birth, Program_ID*)

**Program – 2** → (<u>Program_ID</u>, Program_Title, Program_Duration, Program_Description)

**Program_Module – 2** → (<u>Program_ID</u>, <u>Module_ID</u>)

**Module – 2** → (<u>Module_ID</u>, Module_Title, Module_Credit_Hours, Module_Duration)

**Teacher – 2** → (Teacher_ID, Teacher_Name, Teacher_Specialization, Teacher_Phone, Teacher_Email)

**Teacher_Module – 2** → (Teacher_ID, Module_ID)

**Announcement – 2** → (Announcement_ID, Announcement_Title, Announcement_Content, Announcement_Post_Date, Module_ID*, Teacher_ID*)

**Resource – 2** → (Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence, Module_ID*)

**Assessment – 2** → (Assessment_ID, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type, Module_ID*)

**Assessment_Result – 2** → (Student_ID, Assessment_ID)

**Result – 2** → (Result_ID, Result_Grade, Result_Marks, Result_Status, Result_Feedback, Student_ID*, Assessment_ID*)

Here, all the full functional dependencies and partial functional dependencies are identified and separated as per the principles of normalization in the Second Normal Form (2NF). The **Program_Module**, **Teacher_Module** and **Assessment_Result** tables are created in order to eliminate partial functional dependencies and satisfy Second Normal Form (2NF).

## 3.4 Third Normal Form (3NF)

Tables listed in Third Normal Form (3NF) should have transitive dependencies. If any transitive dependencies exists then it is removed.

If the attributes of A determines B then, A → B.

If the attributes of B determines C then, B → C.

Then, A → C which means attributes of C is transitively dependent on attributes of A.

Hence, transitively dependency is defined from the above relations.

### 3.4.1 Rules for Third Normal Form (3NF)

In Second Normal Form (2NF), the following details should be considered while listing tables:

- ✓ Each tables should already be in Second Normal Form (2NF).
- ✓ No transitive dependencies should exist in the tables. If any transitive dependencies arise then it should be eliminated by breaking into small relatable tables.
- ✓ The non-prime attributes should be dependent only on the unique identifier primary key.

### 3.4.2 Third Normal Form (3NF) Process

Tables listed in Second Normal Form (2NF) have been checked for any partial functional dependencies and eliminated. But, after checking partial functional dependencies there is possibility of transitive dependencies in tables. To eliminate transitive dependencies we check each and every tables and eliminate it to satisfy the principles of Third Normal Form (3NF).

The tables are checked for any potential partial dependencies and eliminated which are listed below:

- ✓ **Student_ID → X**
  (No transitive dependency since all non-prime attributes depend on **Student_ID**.)

- ✓ **Program_ID → X**
  (No transitive dependency since all non-prime attributes depend on **Program_ID**.)

- ✓ **Program_ID, Module_ID → X**
  (No transitive dependency since there are no non-key attributes.)

- ✓ **Module_ID → X**
  (No transitive dependency since all non-prime attributes depend on **Module_ID**.)

- ✓ **Teacher_ID → X**
  (No transitive dependency since all non-prime attributes depend on **Teacher_ID**.)

- ✓ **Teacher_ID, Module_ID → X**
  (No transitive dependency since there are no non-key attributes.)

- ✓ **Announcement_ID → X**
  (No transitive dependency since all non-prime attributes depend on **Announcement_ID**.)

- ✓ **Resource_ID → X**
  (No transitive dependency since all non-prime attributes depend on **Resource_ID**.)

- ✓ **Assessment_ID → X**
  (No transitive dependency since all non-prime attributes depend on **Assessment_ID**.)

- ✓ **Student_ID, Assessment_ID → Result_ID → Result_Marks, Result_Grade, Result_Status, Result_Feedback** (Transitive Dependency)
  Transitive dependency exists as the non-key attributes of **Result_ID** are transitively dependent on omposite keys **Student_ID** and **Assessment_ID.**

  **(Student_ID, Assessment_ID) → Result_ID**
  **Result_ID** is functionally dependent on combination of composite keys **Student_ID** and **Assessment_ID**

  **Result_ID → Result_Marks, Result_Grade, Result_Status, Result_Feedback**
  The attributes **Result_Marks, Result_Grade, Result_Status, Result_Feedback** are transitively dependent on composite keys **Student_ID** and **Assessment_ID** through **Result_ID.**

**Final Third Normal Form (3NF) Tables**

**Student – 3** → (Student_ID, Student_Name, Student_Address, Student_Email, Student_Phone, Student_Date_Of_Birth, Program_ID*)

**Program – 3** → (Program_ID, Program_Title, Program_Duration, Program_Description)

**Program_Module – 3** → (Program_ID, Module_ID)

**Module – 3** → (Module_ID, Module_Title, Module_Credit_Hours, Module_Duration)

**Teacher – 3** → (Teacher_ID, Teacher_Name, Teacher_Specialization, Teacher_Phone, Teacher_Email)

**Teacher_Module – 3** → (Teacher_ID, Module_ID)

**Announcement – 3** → (Announcement_ID, Announcement_Title, Announcement_Content, Announcement_Post_Date, Module_ID*, Teacher_ID*)

**Resource – 3** → (Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence, Module_ID*)

**Assessment – 3** → (Assessment_ID, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type, Module_ID*)

**Assessment_Result – 3** → (Student_ID, Assessment_ID, Result_ID)

**Result – 3** → (Result_ID, Result_Grade, Result_Marks, Result_Status, Result_Feedback, Student_ID*, Assessment_ID*)

Here, the tables have been normalized to the Third Normal Form (3NF). Possible transitive dependencies are eliminated and the principles of the Third Normal Form (3NF) are also satisfied from the tables that are mentioned above in the list of normalized tables

## 4.0 Data Dictionary and Final Entity Relationship Diagram (ERD)

A set of new tables were listed after normalization was conducted. This ensured that there were no any unnecessary data redundancy, inconsistencies and anomalies. The tables were separated and relationships were defined to connect tables that was used for storing data in those tables in more efficient manner.

## 4.1 Data Dictionary

### 4.1.1 Program

| S.N. | Attribute Name | Data Type | Size | Constraint |
|---|---|---|---|---|
| 1. | Program_ID | Number | 15 | Primary Key |
| 2. | Program_Title | Character | 35 | Not Null |
| 3. | Program_Duration | Character | 20 | Not Null |
| 4, | Program_Description | Character | 60 | Not Null |

*Table 5 Program data dictionary*

### 4.1.2 Student

| S.N. | Attribute Name | Data Type | Size | Constraint |
|---|---|---|---|---|
| 1. | Student_ID | Number | 15 | Primary Key |
| 2. | Student_Name | Character | 20 | Not Null |
| 3. | Student_Address | Character | 20 | Not Null |
| 4. | Student_Phone | Character | 20 | Unique |
| 5. | Student_Email | Character | 30 | Unique |
| 6. | Student_Date_Of_Birth | Date | - | Not Null |
| 7. | Program_ID | Number | 15 | Foreign Key |

*Table 6 Student data dictionary*

### 4.1.3 Module

| S.N. | Attribute Name | Data Type | Size | Constraint |
|---|---|---|---|---|
| 1. | Module_ID | Number | 15 | Primary Key |
| 2. | Module_Title | Character | 30 | Not Null |
| 3. | Module_Credit_Hours | Number | - | Not Null |
| 4. | Module_Duration | Character | 20 | Not Null |

*Table 7 Module data dictionary*

### 4.1.4 Teacher

| S.N. | Attribute Name | Data Type | Size | Constraint |
|---|---|---|---|---|
| 1. | Teacher_ID | Number | 15 | Primary Key |
| 2. | Teacher_Name | Character | 25 | Not Null |
| 3. | Teacher_Specialization | Character | 40 | Not Null |
| 4. | Teacher_Phone | Character | 20 | Not Null |
| 5. | Teacher_Email | Character | 30 | Not Null |

*Table 8 Teacher data dictionary*

### 4.1.5 Program_Module

| S.N. | Attribute Name | Data Type | Size | Constraint | Composite Constraint |
|---|---|---|---|---|---|
| 1. | Program_ID | Number | 15 | Primary Key, Foreign Key | Primary Key |
| 2. | Module_ID | Number | 15 | Primary Key, Foreign Key | |

*Table 9 Program_Module data dictionary*

### 4.1.6 Teacher_Module

| S.N. | Attribute Name | Data Type | Size | Constraint | Composite Constraint |
|---|---|---|---|---|---|
| 1. | Teacher_ID | Number | 15 | Primary Key, Foreign Key | Primary Key |
| 2. | Module_ID | Number | 15 | Primary Key, Foreign Key | |

*Table 10 Teacher_Module data dictionary*

### 4.1.7 Announcement

| S.N. | Attribute Name | Data Type | Size | Constraint |
|---|---|---|---|---|
| 1. | Announcement_ID | Number | 20 | Primary Key |
| 2. | Announcement_Title | Character | 40 | Not Null |
| 3. | Announcement_Content | Character | 120 | Not Null |
| 4. | Announcement_Post_Date | Date | - | Not Null |
| 5. | Module_ID | Number | 15 | Foreign Key |
| 6. | Teacher_ID | Number | 15 | Foreign Key |

*Table 11 Announcement data dictionary*

### 4.1.8 Resources

| S.N. | Attribute Name | Data Type | Size | Constraint |
|---|---|---|---|---|
| 1. | Resource_ID | Number | 15 | Primary Key |
| 2. | Resource_Title | Character | 40 | Not Null |
| 3. | Resource_Type | Character | 20 | Not Null |
| 4. | Resource_Status | Character | 20 | Not Null |
| 5. | Resource_Sequence | Number | - | Not Null |
| 6. | Module_ID | Number | 15 | Foreign Key |

*Table 12 Resources data dictionary*

### 4.2.7 Assessment

| S.N. | Attribute Name | Data Type | Size | Constraint |
|---|---|---|---|---|
| 1. | Assessment_ID | Number | 15 | Primary Key |
| 2. | Assessment_Title | Character | 30 | Not Null |
| 3. | Assessment_Deadline | Date | - | Not Null |
| 4. | Assessment_Weightage | Number | - | Not Null |
| 5. | Assessment_Type | Character | 25 | Not Null |
| 6. | Module_ID | Number | 15 | Foreign Key |

*Table 13 Assessment data dictionar*

### 4.2.8 Result

| S.N. | Attribute Name | Data Type | Size | Constraint |
|---|---|---|---|---|
| 1. | Result_ID | Number | 15 | Primary Key |
| 2. | Result_Grade | Character | 15 | Not Null |
| 3. | Result_Marks | Number | - | Not Null |
| 4. | Result_Status | Character | 15 | Not Null |
| 5. | Result_Feedback | Character | 65 | Not Null |
| 6. | Student_ID | Number | 15 | Foreign Key |
| 7. | Assessment_ID | Number | 15 | Foreign Key |

*Table 14 Result data dictionary*

### 4.2.9 Assessment_Result

| S.N. | Attribute Name | Data Type | Size | Constraint | Composite Constraint |
|---|---|---|---|---|---|
| 1. | Student_ID | Number | 15 | Primary Key, Foreign Key | Primary Key |
| 2. | Assessment_ID | Number | 15 | Primary Key, Foreign Key | |
| 3. | Result_ID | Number | 15 | Primary Key, Foreign Key | |

*Table 15 Assessment_Result data dictionary*

## 4.2 Final Entity Relationship Diagram (ERD)

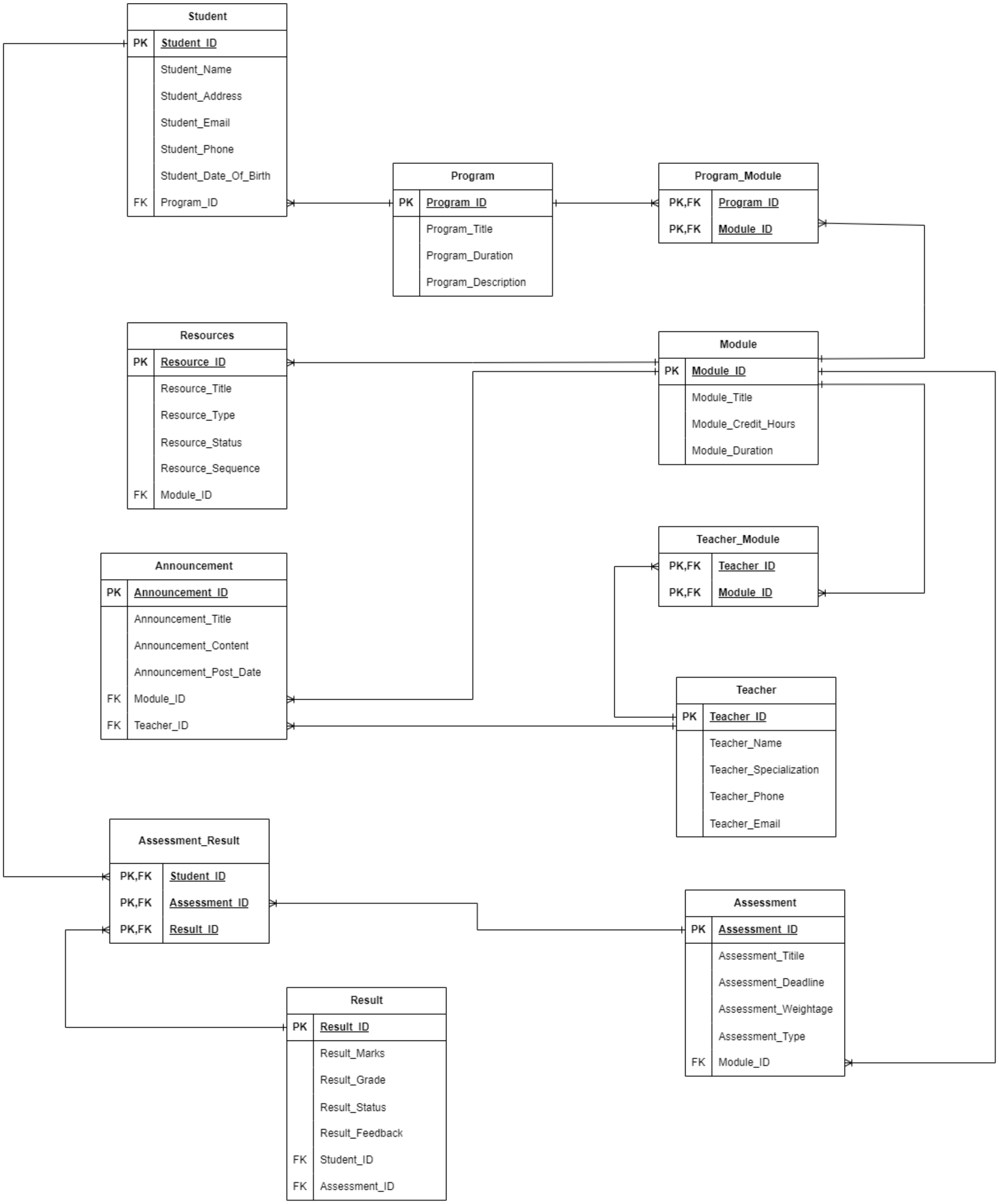
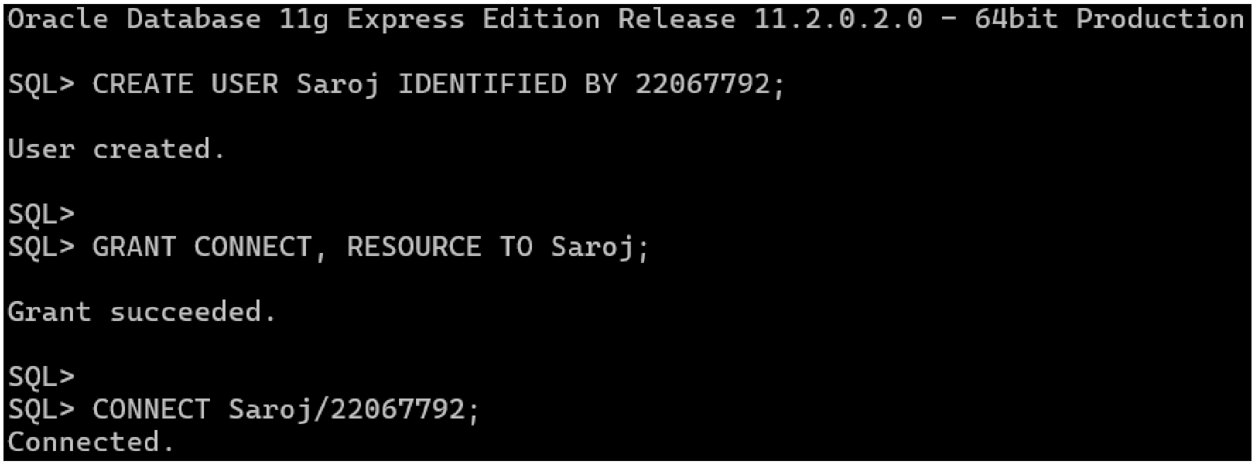


*Figure 2 Final Entity Relationship Diagram (ERD)*

# 5.0 Implementation

## 5.1 User creation and granting privileges

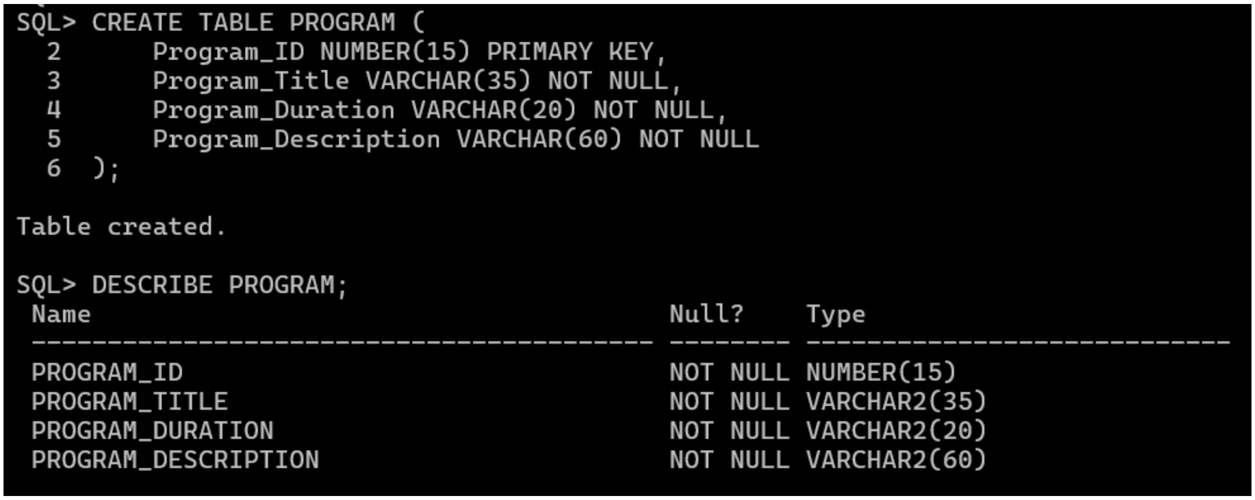


```
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> CREATE USER Saroj IDENTIFIED BY 22067792;

User created.

SQL>
SQL> GRANT CONNECT, RESOURCE TO Saroj;

Grant succeeded.

SQL>
SQL> CONNECT Saroj/22067792;
Connected.
```

*Figure 3 User creation and granting privileges*

## 5.2 Creating and Describing Tables

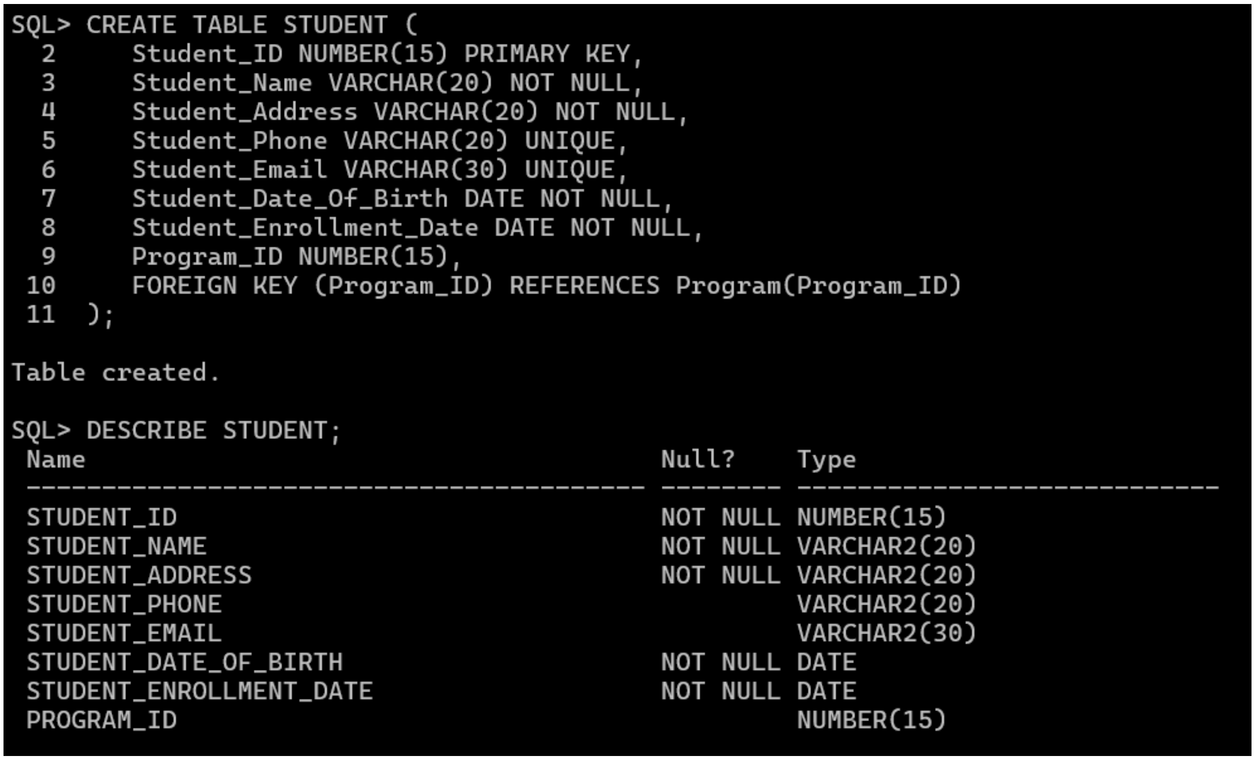
### 5.2.1 Creating and Describing Program Table



```
SQL> CREATE TABLE PROGRAM (
  2      Program_ID NUMBER(15) PRIMARY KEY,
  3      Program_Title VARCHAR(35) NOT NULL,
  4      Program_Duration VARCHAR(20) NOT NULL,
  5      Program_Description VARCHAR(60) NOT NULL
  6  );

Table created.

SQL> DESCRIBE PROGRAM;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PROGRAM_ID                                NOT NULL NUMBER(15)
 PROGRAM_TITLE                             NOT NULL VARCHAR2(35)
 PROGRAM_DURATION                          NOT NULL VARCHAR2(20)
 PROGRAM_DESCRIPTION                       NOT NULL VARCHAR2(60)
```

*Figure 4 Creating and Describing Program Table*

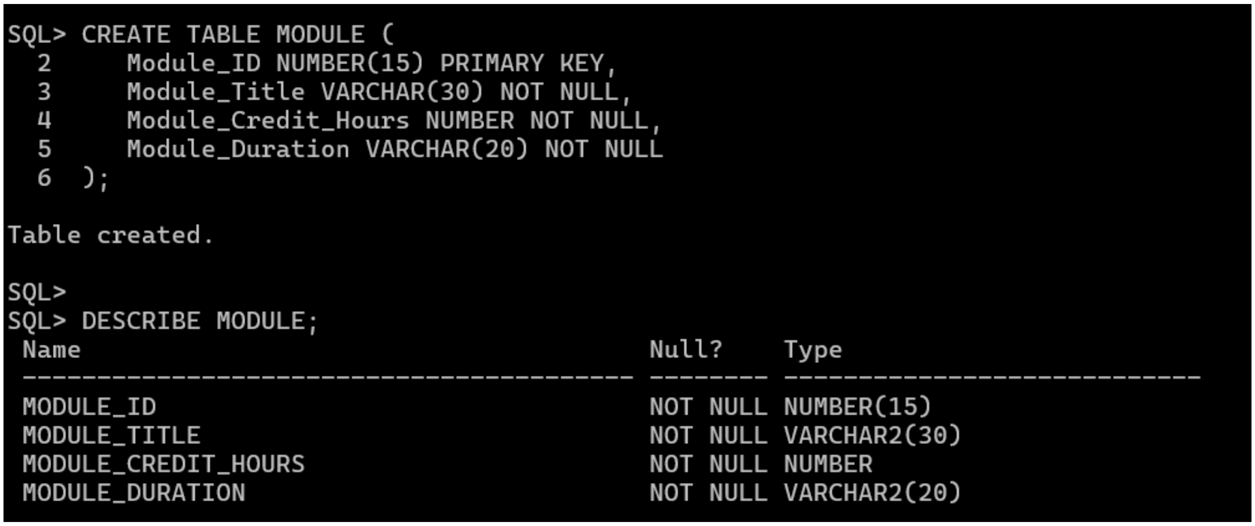
### 5.2.2 Creating and Describing Student Table



```
SQL> CREATE TABLE STUDENT (
  2     Student_ID NUMBER(15) PRIMARY KEY,
  3     Student_Name VARCHAR(20) NOT NULL,
  4     Student_Address VARCHAR(20) NOT NULL,
  5     Student_Phone VARCHAR(20) UNIQUE,
  6     Student_Email VARCHAR(30) UNIQUE,
  7     Student_Date_Of_Birth DATE NOT NULL,
  8     Student_Enrollment_Date DATE NOT NULL,
  9     Program_ID NUMBER(15),
 10     FOREIGN KEY (Program_ID) REFERENCES Program(Program_ID)
 11  );

Table created.

SQL> DESCRIBE STUDENT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STUDENT_ID                                NOT NULL NUMBER(15)
 STUDENT_NAME                              NOT NULL VARCHAR2(20)
 STUDENT_ADDRESS                           NOT NULL VARCHAR2(20)
 STUDENT_PHONE                                      VARCHAR2(20)
 STUDENT_EMAIL                                      VARCHAR2(30)
 STUDENT_DATE_OF_BIRTH                     NOT NULL DATE
 STUDENT_ENROLLMENT_DATE                   NOT NULL DATE
 PROGRAM_ID                                         NUMBER(15)
```

*Figure 5 Creating and Describing Student Table*

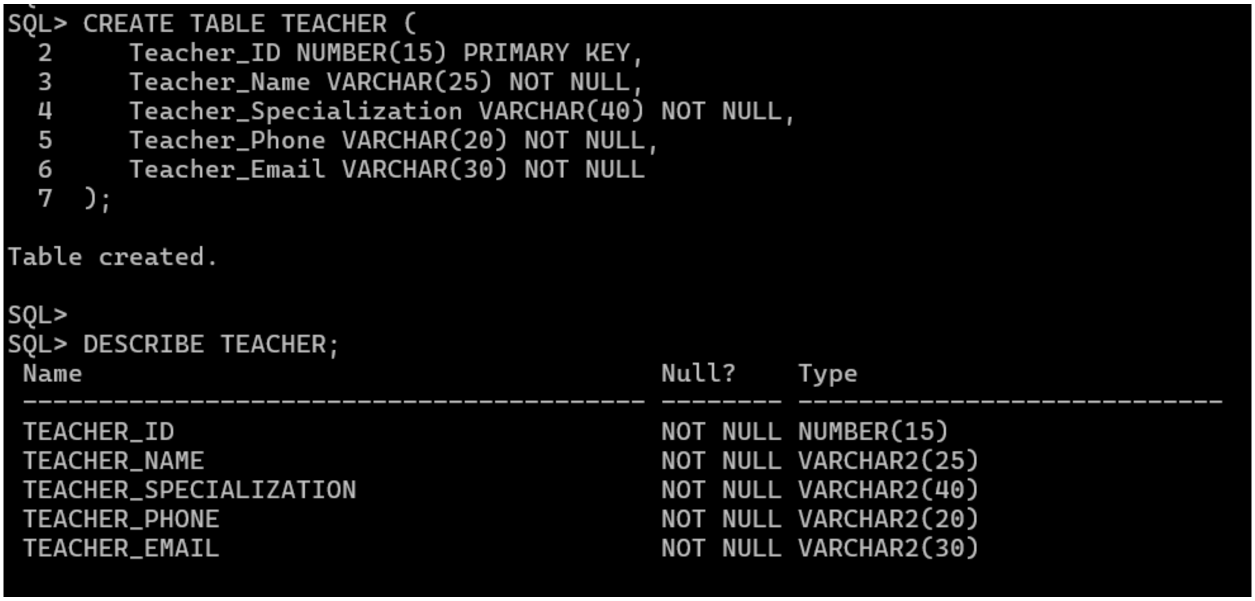
### 5.2.3 Creating and Describing Module Table



```
SQL> CREATE TABLE MODULE (
  2     Module_ID NUMBER(15) PRIMARY KEY,
  3     Module_Title VARCHAR(30) NOT NULL,
  4     Module_Credit_Hours NUMBER NOT NULL,
  5     Module_Duration VARCHAR(20) NOT NULL
  6  );

Table created.

SQL>
SQL> DESCRIBE MODULE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 MODULE_ID                                 NOT NULL NUMBER(15)
 MODULE_TITLE                              NOT NULL VARCHAR2(30)
 MODULE_CREDIT_HOURS                       NOT NULL NUMBER
 MODULE_DURATION                           NOT NULL VARCHAR2(20)
```

*Figure 6 Creating and Describing Module Table*

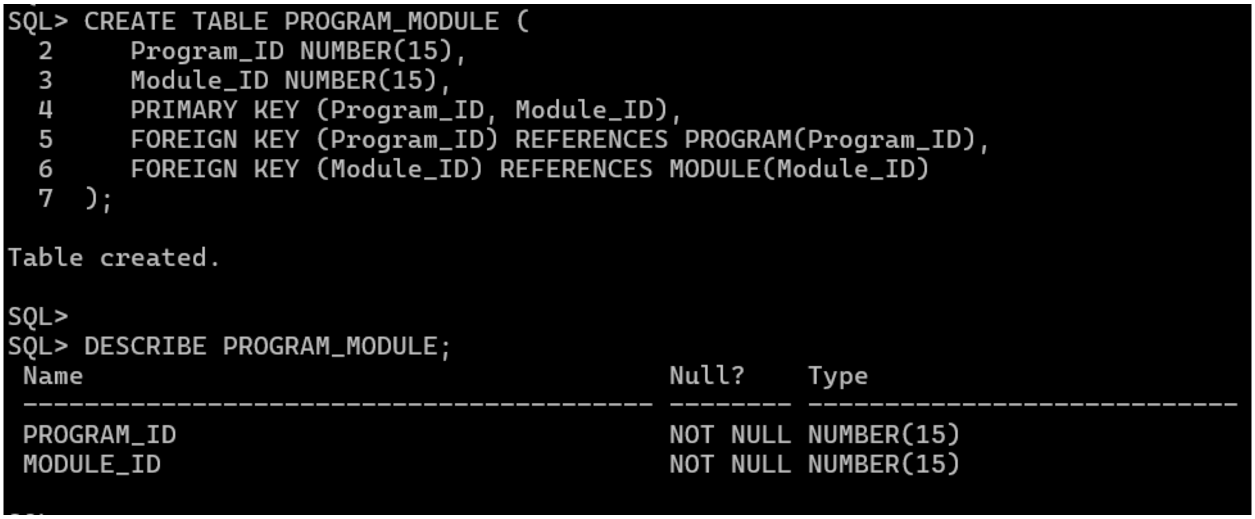
### 5.2.4 Creating and Describing Teacher Table



```
SQL> CREATE TABLE TEACHER (
       Teacher_ID NUMBER(15) PRIMARY KEY,
       Teacher_Name VARCHAR(25) NOT NULL,
       Teacher_Specialization VARCHAR(40) NOT NULL,
       Teacher_Phone VARCHAR(20) NOT NULL,
       Teacher_Email VARCHAR(30) NOT NULL
   );

Table created.

SQL>
SQL> DESCRIBE TEACHER;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TEACHER_ID                                NOT NULL NUMBER(15)
 TEACHER_NAME                              NOT NULL VARCHAR2(25)
 TEACHER_SPECIALIZATION                    NOT NULL VARCHAR2(40)
 TEACHER_PHONE                             NOT NULL VARCHAR2(20)
 TEACHER_EMAIL                             NOT NULL VARCHAR2(30)
```

*Figure 7 Creating and Describing Teacher Table*

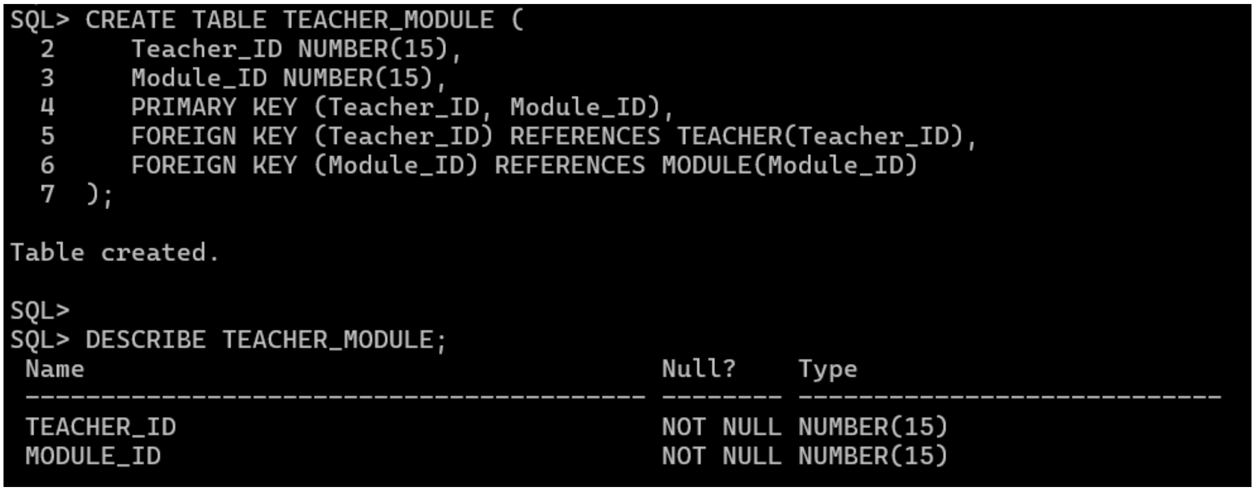
### 5.2.5 Creating and Describing Program_Module Table



```
SQL> CREATE TABLE PROGRAM_MODULE (
       Program_ID NUMBER(15),
       Module_ID NUMBER(15),
       PRIMARY KEY (Program_ID, Module_ID),
       FOREIGN KEY (Program_ID) REFERENCES PROGRAM(Program_ID),
       FOREIGN KEY (Module_ID) REFERENCES MODULE(Module_ID)
   );

Table created.

SQL>
SQL> DESCRIBE PROGRAM_MODULE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 PROGRAM_ID                                NOT NULL NUMBER(15)
 MODULE_ID                                 NOT NULL NUMBER(15)
```

*Figure 8 Creating and Describing Program_Module Table*

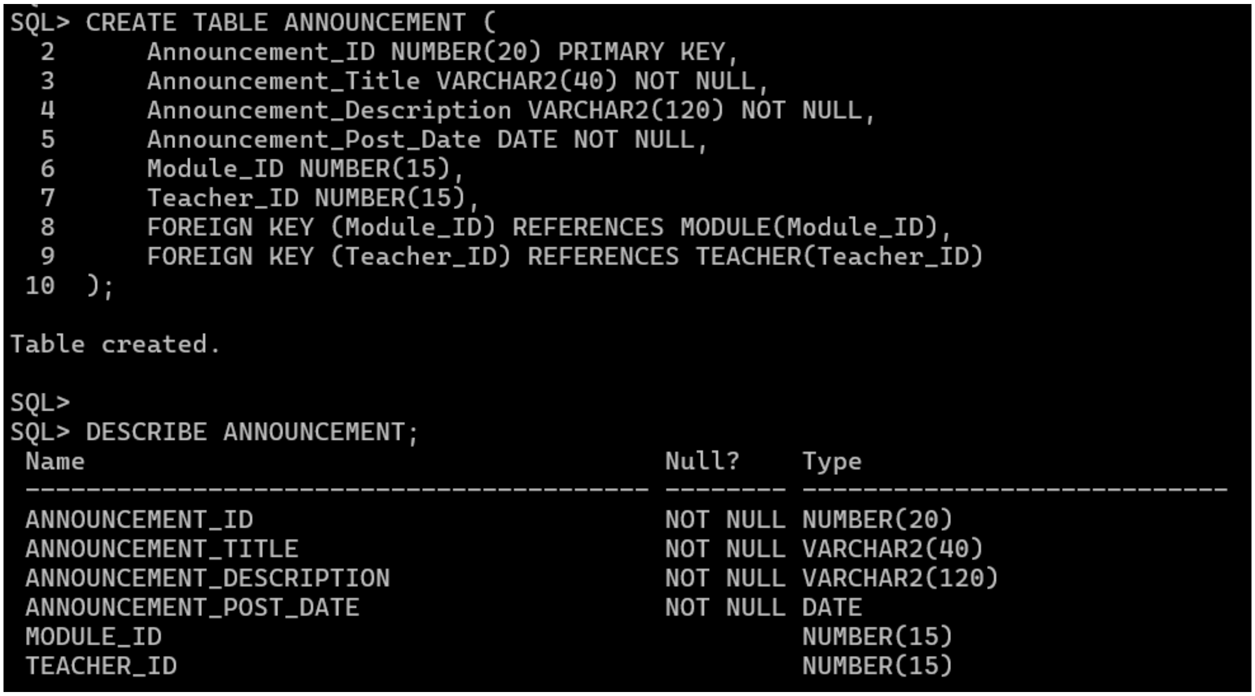
### 5.2.6 Creating and Describing Teacher_Module Table



```
SQL> CREATE TABLE TEACHER_MODULE (
        Teacher_ID NUMBER(15),
        Module_ID NUMBER(15),
        PRIMARY KEY (Teacher_ID, Module_ID),
        FOREIGN KEY (Teacher_ID) REFERENCES TEACHER(Teacher_ID),
        FOREIGN KEY (Module_ID) REFERENCES MODULE(Module_ID)
     );

Table created.

SQL>
SQL> DESCRIBE TEACHER_MODULE;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 TEACHER_ID                                NOT NULL NUMBER(15)
 MODULE_ID                                 NOT NULL NUMBER(15)
```

*Figure 9 Creating and Describing Teacher_Module Table*

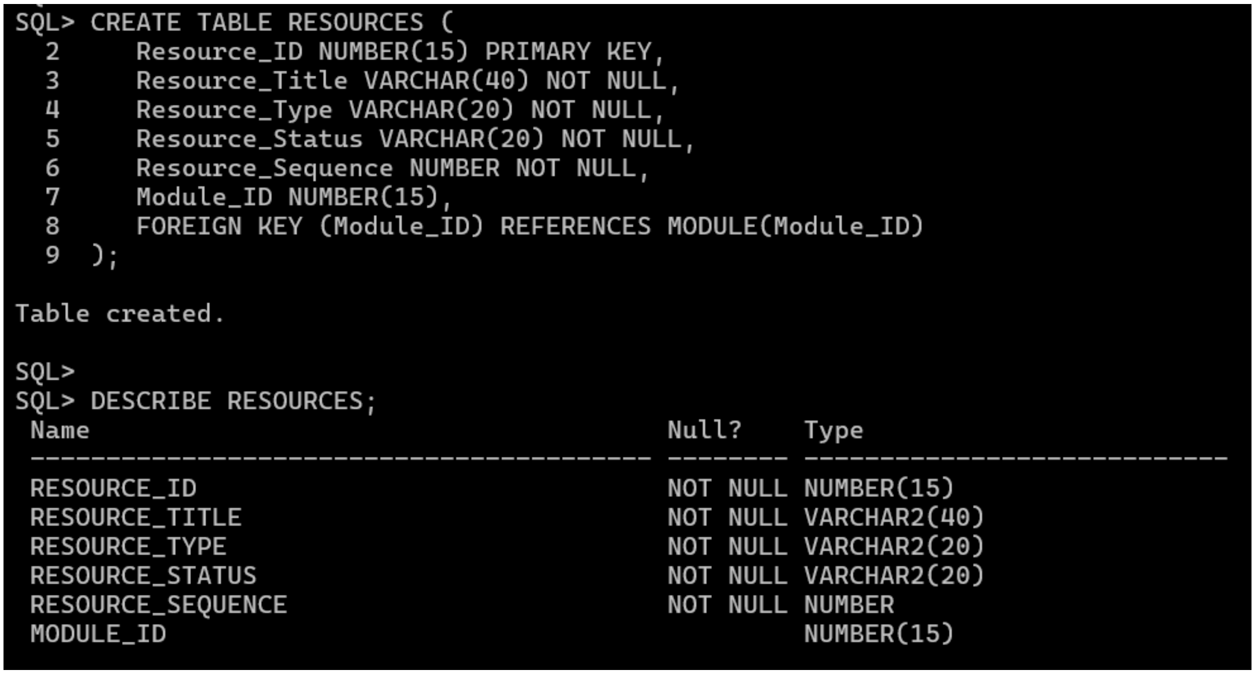
### 5.2.7 Creating and Describing Announcement Table



```
SQL> CREATE TABLE ANNOUNCEMENT (
         Announcement_ID NUMBER(20) PRIMARY KEY,
         Announcement_Title VARCHAR2(40) NOT NULL,
         Announcement_Description VARCHAR2(120) NOT NULL,
         Announcement_Post_Date DATE NOT NULL,
         Module_ID NUMBER(15),
         Teacher_ID NUMBER(15),
         FOREIGN KEY (Module_ID) REFERENCES MODULE(Module_ID),
         FOREIGN KEY (Teacher_ID) REFERENCES TEACHER(Teacher_ID)
     );

Table created.

SQL>
SQL> DESCRIBE ANNOUNCEMENT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ANNOUNCEMENT_ID                           NOT NULL NUMBER(20)
 ANNOUNCEMENT_TITLE                        NOT NULL VARCHAR2(40)
 ANNOUNCEMENT_DESCRIPTION                  NOT NULL VARCHAR2(120)
 ANNOUNCEMENT_POST_DATE                    NOT NULL DATE
 MODULE_ID                                          NUMBER(15)
 TEACHER_ID                                         NUMBER(15)
```

*Figure 10 Creating and Describing Announcement Table*

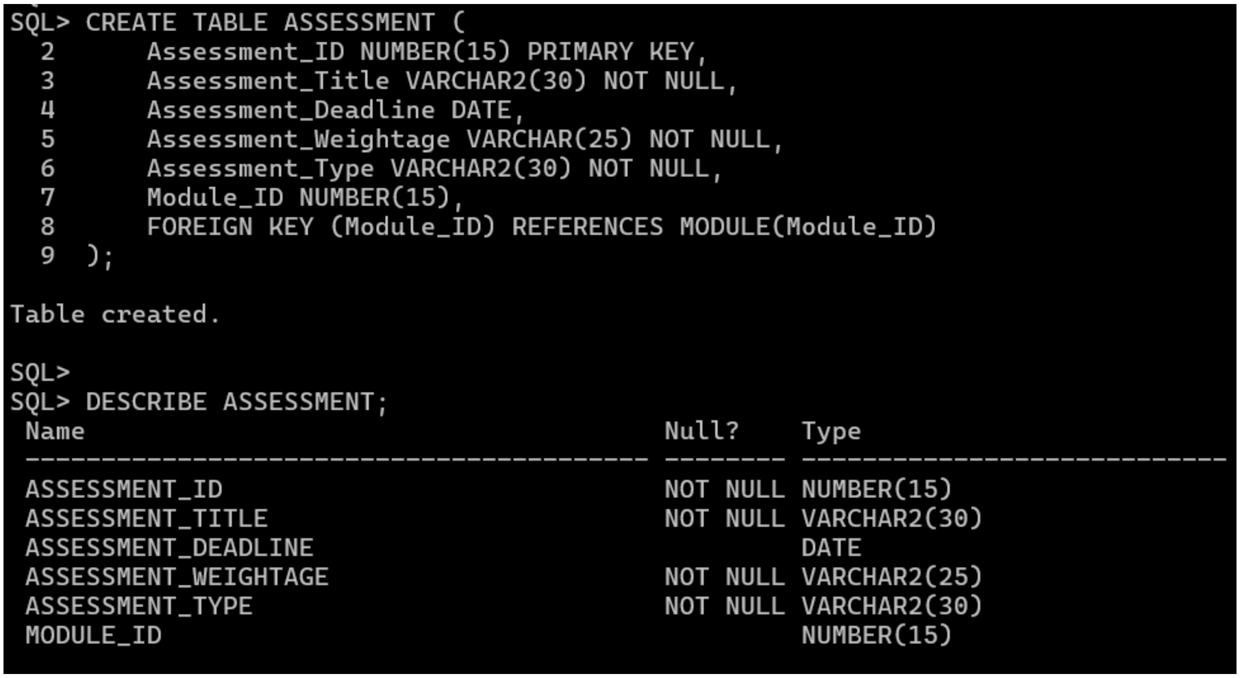
### 5.2.8 Creating and Describing Resources Table



```
SQL> CREATE TABLE RESOURCES (
        Resource_ID NUMBER(15) PRIMARY KEY,
        Resource_Title VARCHAR(40) NOT NULL,
        Resource_Type VARCHAR(20) NOT NULL,
        Resource_Status VARCHAR(20) NOT NULL,
        Resource_Sequence NUMBER NOT NULL,
        Module_ID NUMBER(15),
        FOREIGN KEY (Module_ID) REFERENCES MODULE(Module_ID)
     );

Table created.

SQL>
SQL> DESCRIBE RESOURCES;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 RESOURCE_ID                               NOT NULL NUMBER(15)
 RESOURCE_TITLE                            NOT NULL VARCHAR2(40)
 RESOURCE_TYPE                             NOT NULL VARCHAR2(20)
 RESOURCE_STATUS                           NOT NULL VARCHAR2(20)
 RESOURCE_SEQUENCE                         NOT NULL NUMBER
 MODULE_ID                                          NUMBER(15)
```

*Figure 11 Creating and Describing Resources Table*

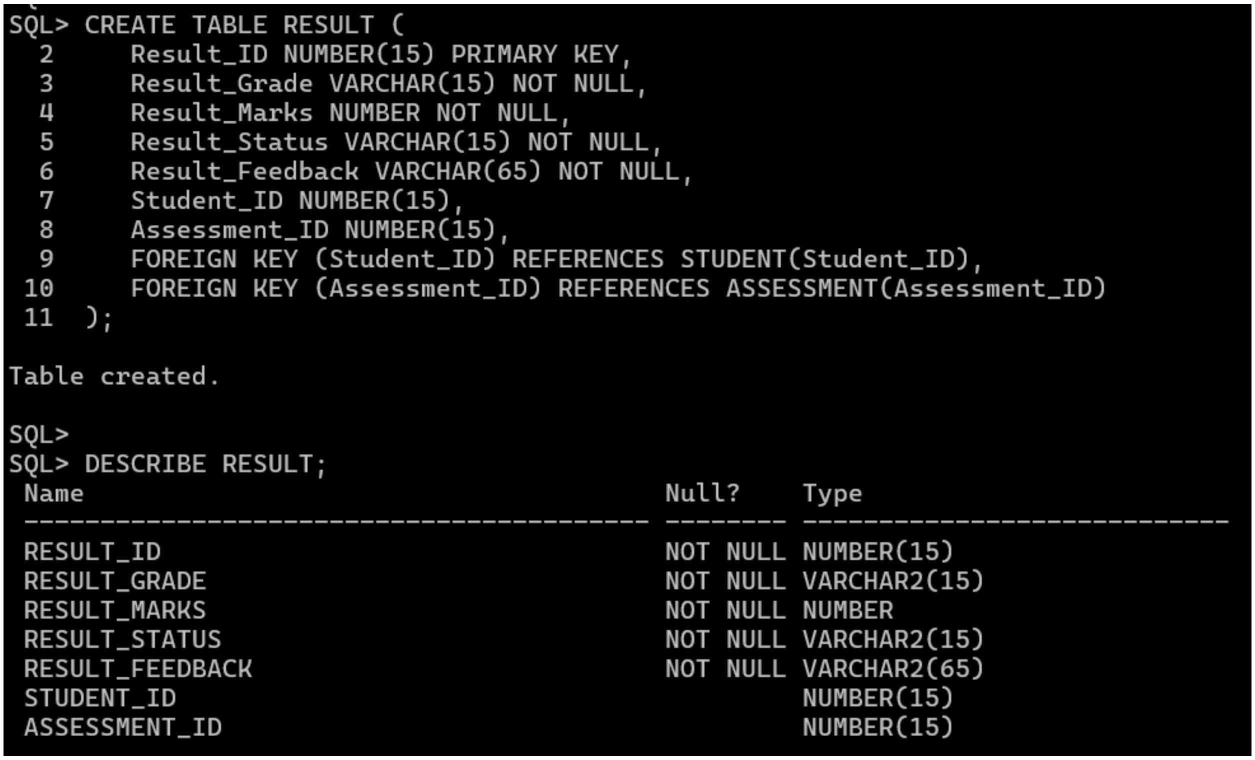
### 5.2.9 Creating and Describing Assessment Table



```
SQL> CREATE TABLE ASSESSMENT (
         Assessment_ID NUMBER(15) PRIMARY KEY,
         Assessment_Title VARCHAR2(30) NOT NULL,
         Assessment_Deadline DATE,
         Assessment_Weightage VARCHAR(25) NOT NULL,
         Assessment_Type VARCHAR2(30) NOT NULL,
         Module_ID NUMBER(15),
         FOREIGN KEY (Module_ID) REFERENCES MODULE(Module_ID)
     );

Table created.

SQL>
SQL> DESCRIBE ASSESSMENT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ASSESSMENT_ID                             NOT NULL NUMBER(15)
 ASSESSMENT_TITLE                          NOT NULL VARCHAR2(30)
 ASSESSMENT_DEADLINE                                DATE
 ASSESSMENT_WEIGHTAGE                      NOT NULL VARCHAR2(25)
 ASSESSMENT_TYPE                           NOT NULL VARCHAR2(30)
 MODULE_ID                                          NUMBER(15)
```

*Figure 12 Creating and Describing Assessment Table*

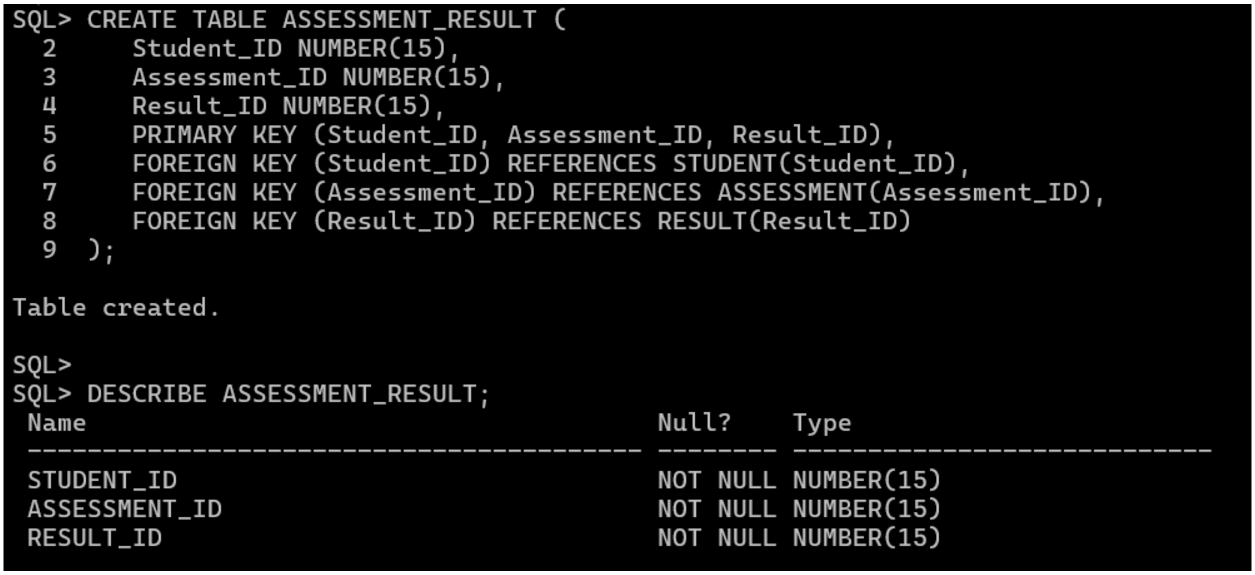
### 5.2.10 Creating and Describing Result Table



```
SQL> CREATE TABLE RESULT (
         Result_ID NUMBER(15) PRIMARY KEY,
         Result_Grade VARCHAR(15) NOT NULL,
         Result_Marks NUMBER NOT NULL,
         Result_Status VARCHAR(15) NOT NULL,
         Result_Feedback VARCHAR(65) NOT NULL,
         Student_ID NUMBER(15),
         Assessment_ID NUMBER(15),
         FOREIGN KEY (Student_ID) REFERENCES STUDENT(Student_ID),
         FOREIGN KEY (Assessment_ID) REFERENCES ASSESSMENT(Assessment_ID)
     );

Table created.

SQL>
SQL> DESCRIBE RESULT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 RESULT_ID                                 NOT NULL NUMBER(15)
 RESULT_GRADE                              NOT NULL VARCHAR2(15)
 RESULT_MARKS                              NOT NULL NUMBER
 RESULT_STATUS                             NOT NULL VARCHAR2(15)
 RESULT_FEEDBACK                           NOT NULL VARCHAR2(65)
 STUDENT_ID                                         NUMBER(15)
 ASSESSMENT_ID                                      NUMBER(15)
```

*Figure 13 Creating and Describing Result Table*

### 5.2.11 Creating and Describing Assessment_Result Table

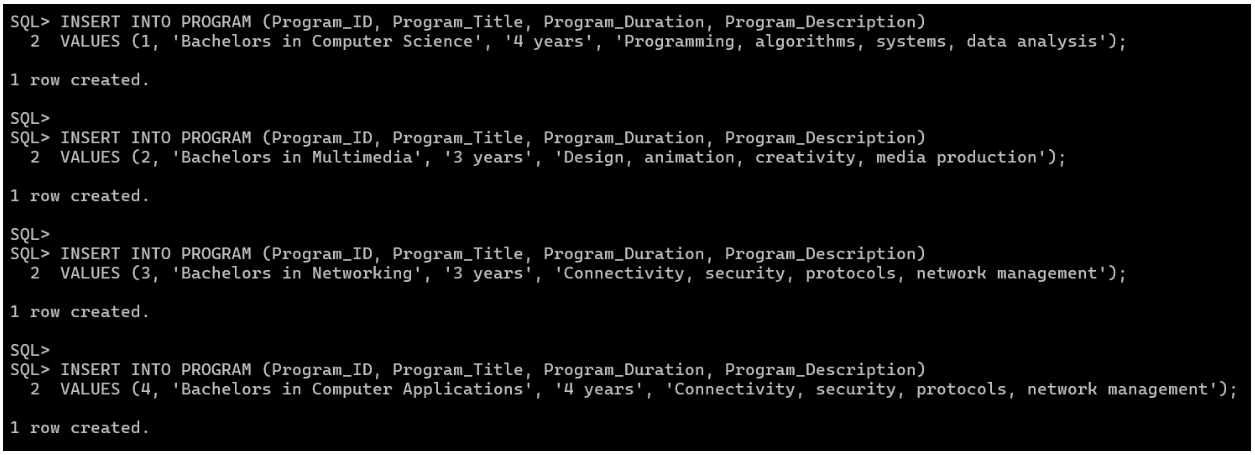


```
SQL> CREATE TABLE ASSESSMENT_RESULT (
         Student_ID NUMBER(15),
         Assessment_ID NUMBER(15),
         Result_ID NUMBER(15),
         PRIMARY KEY (Student_ID, Assessment_ID, Result_ID),
         FOREIGN KEY (Student_ID) REFERENCES STUDENT(Student_ID),
         FOREIGN KEY (Assessment_ID) REFERENCES ASSESSMENT(Assessment_ID),
         FOREIGN KEY (Result_ID) REFERENCES RESULT(Result_ID)
     );

Table created.

SQL>
SQL> DESCRIBE ASSESSMENT_RESULT;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 STUDENT_ID                                NOT NULL NUMBER(15)
 ASSESSMENT_ID                             NOT NULL NUMBER(15)
 RESULT_ID                                 NOT NULL NUMBER(15)
```

*Figure 14 Creating and Describing Assessment_Result*

## 5.3 Inserting data into Tables

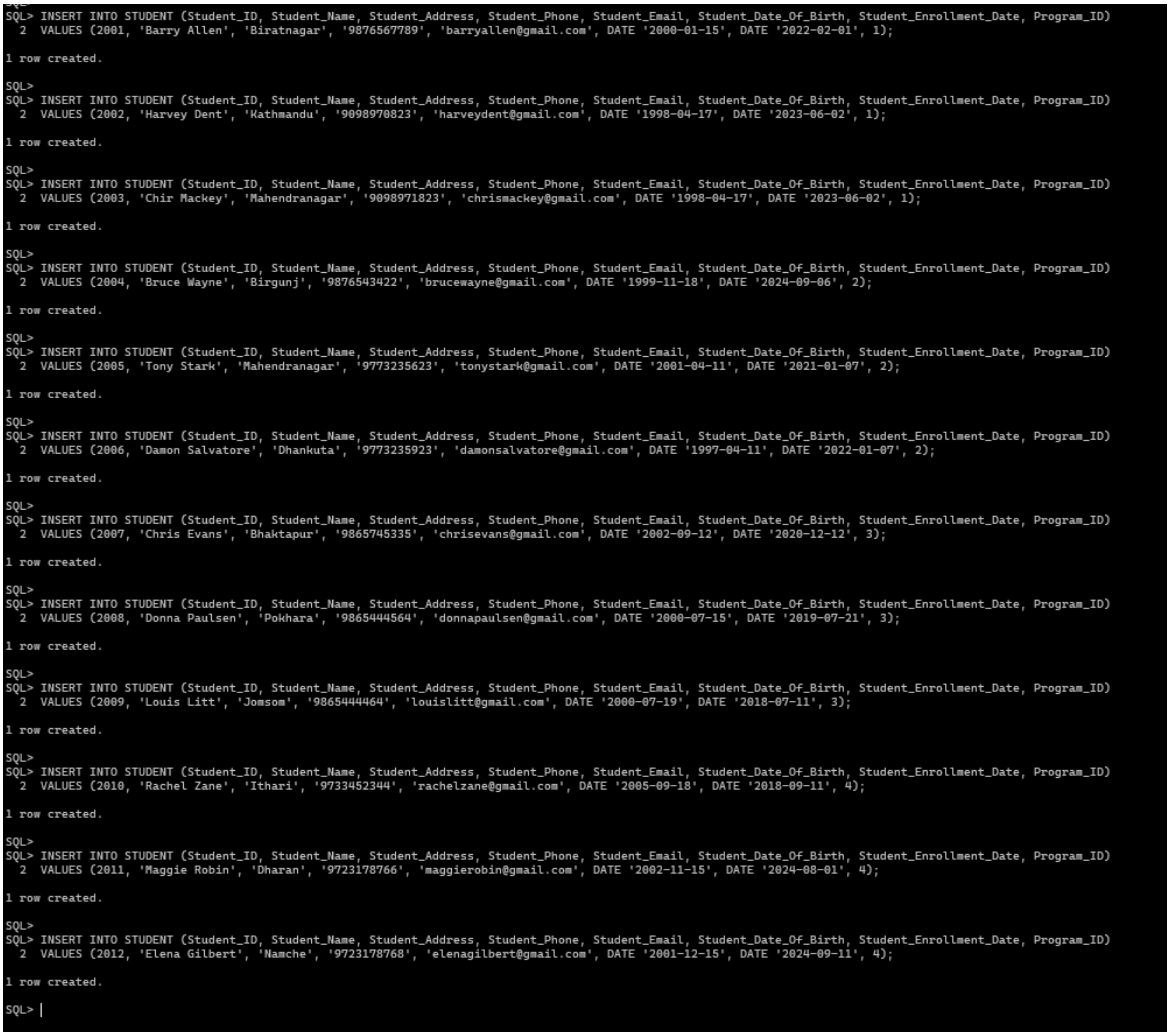
### 5.3.1 Inserting data in Program Table



```
SQL> INSERT INTO PROGRAM (Program_ID, Program_Title, Program_Duration, Program_Description)
  2  VALUES (1, 'Bachelors in Computer Science', '4 years', 'Programming, algorithms, systems, data analysis');

1 row created.

SQL>
SQL> INSERT INTO PROGRAM (Program_ID, Program_Title, Program_Duration, Program_Description)
  2  VALUES (2, 'Bachelors in Multimedia', '3 years', 'Design, animation, creativity, media production');

1 row created.

SQL>
SQL> INSERT INTO PROGRAM (Program_ID, Program_Title, Program_Duration, Program_Description)
  2  VALUES (3, 'Bachelors in Networking', '3 years', 'Connectivity, security, protocols, network management');

1 row created.

SQL>
SQL> INSERT INTO PROGRAM (Program_ID, Program_Title, Program_Duration, Program_Description)
  2  VALUES (4, 'Bachelors in Computer Applications', '4 years', 'Connectivity, security, protocols, network management');

1 row created.
```

*Figure 15 Inserting data in Program Table*

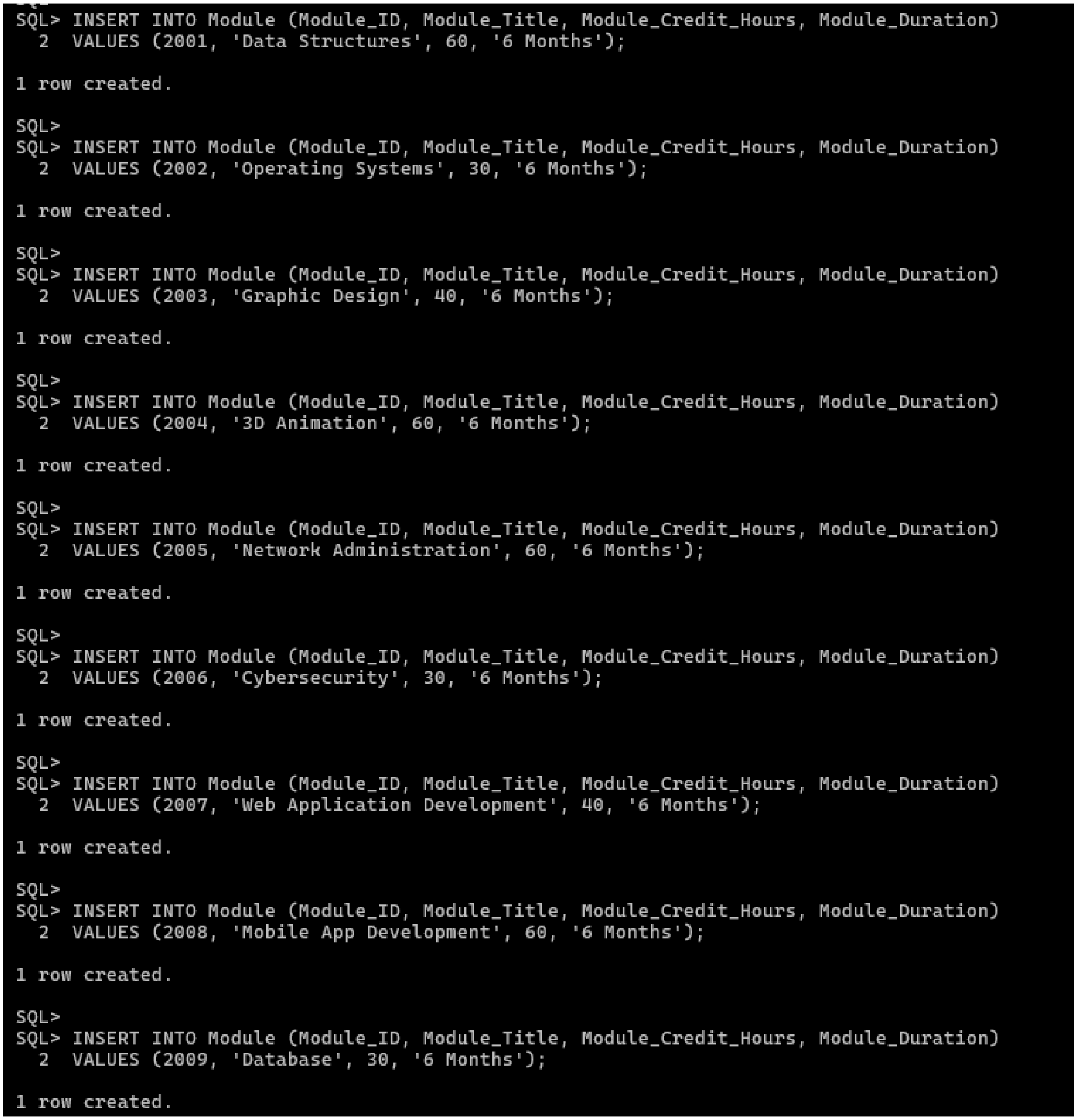
## 5.3.2 Inserting data in Student Table



```
SQL> INSERT INTO STUDENT (Student_ID, Student_Name, Student_Address, Student_Phone, Student_Email, Student_Date_Of_Birth, Student_Enrollment_Date, Program_ID)
  2  VALUES (2001, 'Barry Allen', 'Biratnagar', '9876567789', 'barryallen@gmail.com', DATE '2000-01-15', DATE '2022-02-01', 1);

1 row created.

SQL>
SQL> INSERT INTO STUDENT (Student_ID, Student_Name, Student_Address, Student_Phone, Student_Email, Student_Date_Of_Birth, Student_Enrollment_Date, Program_ID)
  2  VALUES (2002, 'Harvey Dent', 'Kathmandu', '9098970823', 'harveydent@gmail.com', DATE '1998-04-17', DATE '2023-06-02', 1);

1 row created.

SQL>
SQL> INSERT INTO STUDENT (Student_ID, Student_Name, Student_Address, Student_Phone, Student_Email, Student_Date_Of_Birth, Student_Enrollment_Date, Program_ID)
  2  VALUES (2003, 'Chir Mackey', 'Mahendranagar', '9098971823', 'chrismackey@gmail.com', DATE '1998-04-17', DATE '2023-06-02', 1);

1 row created.

SQL>
SQL> INSERT INTO STUDENT (Student_ID, Student_Name, Student_Address, Student_Phone, Student_Email, Student_Date_Of_Birth, Student_Enrollment_Date, Program_ID)
  2  VALUES (2004, 'Bruce Wayne', 'Birgunj', '9876543422', 'brucewayne@gmail.com', DATE '1999-11-18', DATE '2024-09-06', 2);

1 row created.

SQL>
SQL> INSERT INTO STUDENT (Student_ID, Student_Name, Student_Address, Student_Phone, Student_Email, Student_Date_Of_Birth, Student_Enrollment_Date, Program_ID)
  2  VALUES (2005, 'Tony Stark', 'Mahendranagar', '9773235623', 'tonystark@gmail.com', DATE '2001-04-11', DATE '2021-01-07', 2);

1 row created.

SQL>
SQL> INSERT INTO STUDENT (Student_ID, Student_Name, Student_Address, Student_Phone, Student_Email, Student_Date_Of_Birth, Student_Enrollment_Date, Program_ID)
  2  VALUES (2006, 'Damon Salvatore', 'Dhankuta', '9773235923', 'damonsalvatore@gmail.com', DATE '1997-04-11', DATE '2022-01-07', 2);

1 row created.

SQL>
SQL> INSERT INTO STUDENT (Student_ID, Student_Name, Student_Address, Student_Phone, Student_Email, Student_Date_Of_Birth, Student_Enrollment_Date, Program_ID)
  2  VALUES (2007, 'Chris Evans', 'Bhaktapur', '9865745335', 'chrisevans@gmail.com', DATE '2002-09-12', DATE '2020-12-12', 3);

1 row created.

SQL>
SQL> INSERT INTO STUDENT (Student_ID, Student_Name, Student_Address, Student_Phone, Student_Email, Student_Date_Of_Birth, Student_Enrollment_Date, Program_ID)
  2  VALUES (2008, 'Donna Paulsen', 'Pokhara', '9865444564', 'donnapaulsen@gmail.com', DATE '2000-07-15', DATE '2019-07-21', 3);

1 row created.

SQL>
SQL> INSERT INTO STUDENT (Student_ID, Student_Name, Student_Address, Student_Phone, Student_Email, Student_Date_Of_Birth, Student_Enrollment_Date, Program_ID)
  2  VALUES (2009, 'Louis Litt', 'Jomsom', '9865444464', 'louislitt@gmail.com', DATE '2000-07-19', DATE '2018-07-11', 3);

1 row created.

SQL>
SQL> INSERT INTO STUDENT (Student_ID, Student_Name, Student_Address, Student_Phone, Student_Email, Student_Date_Of_Birth, Student_Enrollment_Date, Program_ID)
  2  VALUES (2010, 'Rachel Zane', 'Ithari', '9733452344', 'rachelzane@gmail.com', DATE '2005-09-18', DATE '2018-09-11', 4);

1 row created.

SQL>
SQL> INSERT INTO STUDENT (Student_ID, Student_Name, Student_Address, Student_Phone, Student_Email, Student_Date_Of_Birth, Student_Enrollment_Date, Program_ID)
  2  VALUES (2011, 'Maggie Robin', 'Dharan', '9723178766', 'maggierobin@gmail.com', DATE '2002-11-15', DATE '2024-08-01', 4);

1 row created.

SQL>
SQL> INSERT INTO STUDENT (Student_ID, Student_Name, Student_Address, Student_Phone, Student_Email, Student_Date_Of_Birth, Student_Enrollment_Date, Program_ID)
  2  VALUES (2012, 'Elena Gilbert', 'Namche', '9723178768', 'elenagilbert@gmail.com', DATE '2001-12-15', DATE '2024-09-11', 4);

1 row created.

SQL> |
```

*Figure 16 Inserting data in Student Table*

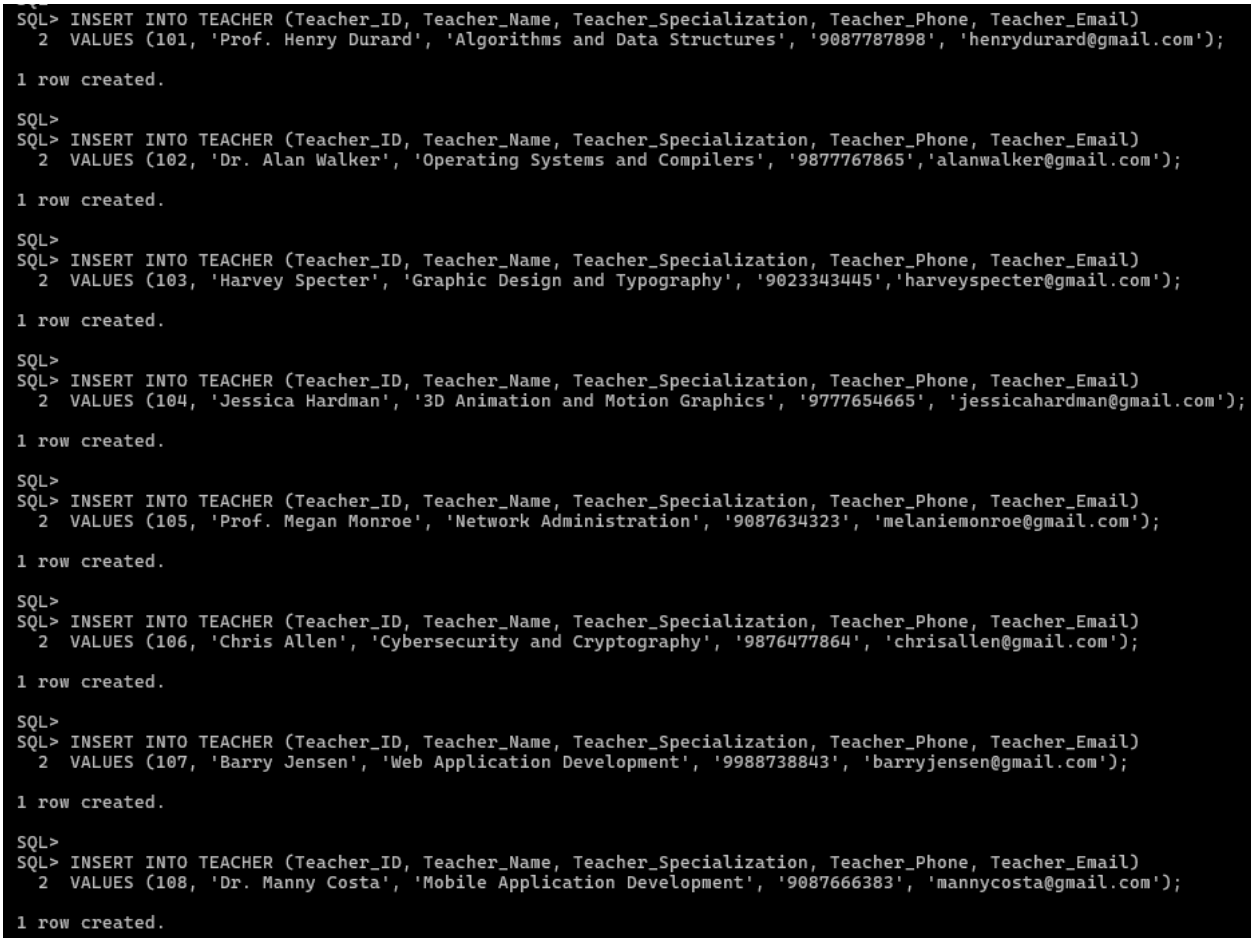
### 5.3.3 Inserting Data in Module Table



```
SQL> INSERT INTO Module (Module_ID, Module_Title, Module_Credit_Hours, Module_Duration)
     VALUES (2001, 'Data Structures', 60, '6 Months');

1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Title, Module_Credit_Hours, Module_Duration)
     VALUES (2002, 'Operating Systems', 30, '6 Months');

1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Title, Module_Credit_Hours, Module_Duration)
     VALUES (2003, 'Graphic Design', 40, '6 Months');

1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Title, Module_Credit_Hours, Module_Duration)
     VALUES (2004, '3D Animation', 60, '6 Months');

1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Title, Module_Credit_Hours, Module_Duration)
     VALUES (2005, 'Network Administration', 60, '6 Months');

1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Title, Module_Credit_Hours, Module_Duration)
     VALUES (2006, 'Cybersecurity', 30, '6 Months');

1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Title, Module_Credit_Hours, Module_Duration)
     VALUES (2007, 'Web Application Development', 40, '6 Months');

1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Title, Module_Credit_Hours, Module_Duration)
     VALUES (2008, 'Mobile App Development', 60, '6 Months');

1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Title, Module_Credit_Hours, Module_Duration)
     VALUES (2009, 'Database', 30, '6 Months');

1 row created.
```

*Figure 17 Inserting Data in Module Table*

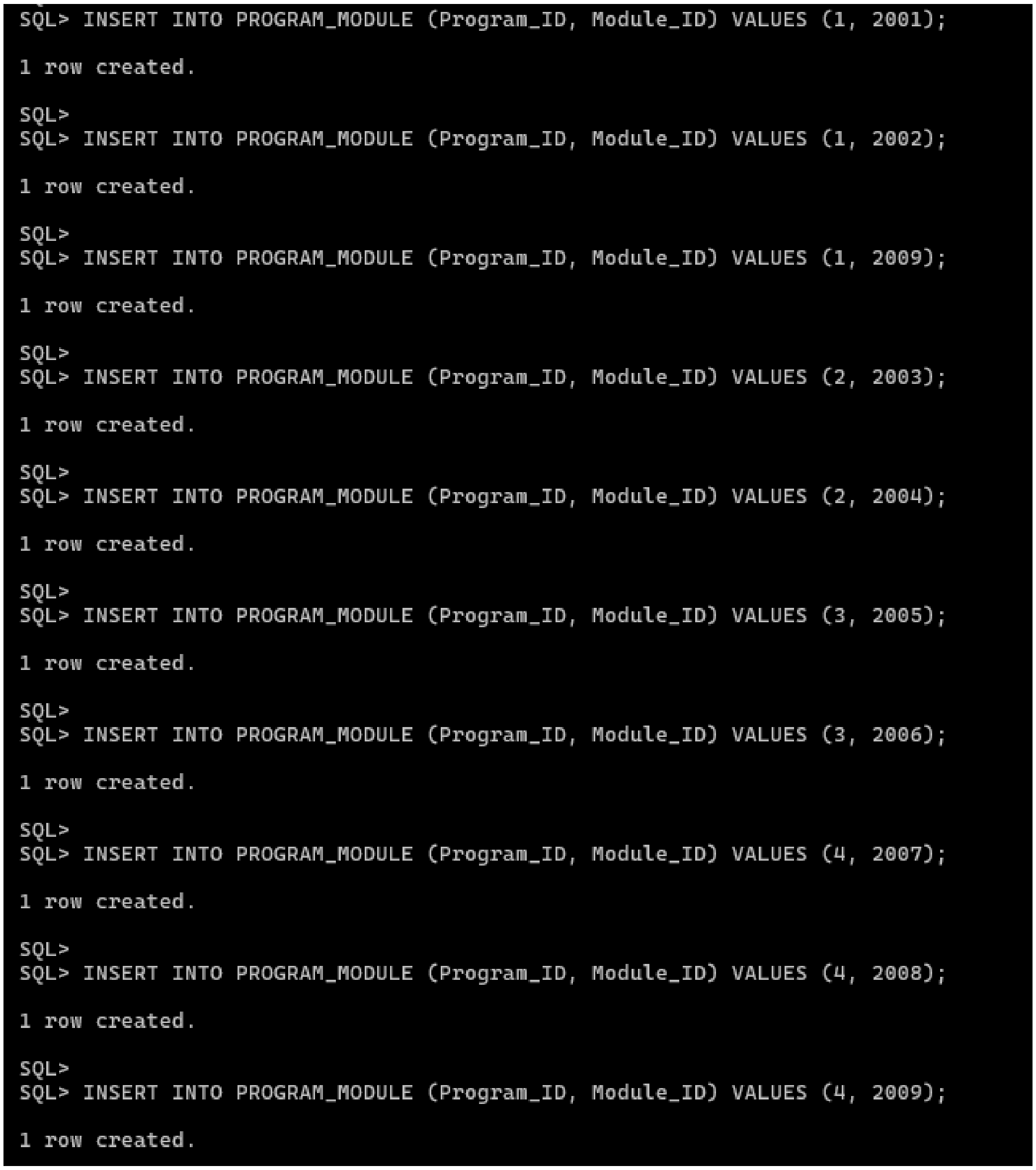
### 5.3.4 Inserting Data in Teacher Table



```
SQL> INSERT INTO TEACHER (Teacher_ID, Teacher_Name, Teacher_Specialization, Teacher_Phone, Teacher_Email)
  2  VALUES (101, 'Prof. Henry Durard', 'Algorithms and Data Structures', '9087787898', 'henrydurard@gmail.com');

1 row created.

SQL>
SQL> INSERT INTO TEACHER (Teacher_ID, Teacher_Name, Teacher_Specialization, Teacher_Phone, Teacher_Email)
  2  VALUES (102, 'Dr. Alan Walker', 'Operating Systems and Compilers', '9877767865','alanwalker@gmail.com');

1 row created.

SQL>
SQL> INSERT INTO TEACHER (Teacher_ID, Teacher_Name, Teacher_Specialization, Teacher_Phone, Teacher_Email)
  2  VALUES (103, 'Harvey Specter', 'Graphic Design and Typography', '9023343445','harveyspecter@gmail.com');

1 row created.

SQL>
SQL> INSERT INTO TEACHER (Teacher_ID, Teacher_Name, Teacher_Specialization, Teacher_Phone, Teacher_Email)
  2  VALUES (104, 'Jessica Hardman', '3D Animation and Motion Graphics', '9777654665', 'jessicahardman@gmail.com');

1 row created.

SQL>
SQL> INSERT INTO TEACHER (Teacher_ID, Teacher_Name, Teacher_Specialization, Teacher_Phone, Teacher_Email)
  2  VALUES (105, 'Prof. Megan Monroe', 'Network Administration', '9087634323', 'melaniemonroe@gmail.com');

1 row created.

SQL>
SQL> INSERT INTO TEACHER (Teacher_ID, Teacher_Name, Teacher_Specialization, Teacher_Phone, Teacher_Email)
  2  VALUES (106, 'Chris Allen', 'Cybersecurity and Cryptography', '9876477864', 'chrisallen@gmail.com');

1 row created.

SQL>
SQL> INSERT INTO TEACHER (Teacher_ID, Teacher_Name, Teacher_Specialization, Teacher_Phone, Teacher_Email)
  2  VALUES (107, 'Barry Jensen', 'Web Application Development', '9988738843', 'barryjensen@gmail.com');

1 row created.

SQL>
SQL> INSERT INTO TEACHER (Teacher_ID, Teacher_Name, Teacher_Specialization, Teacher_Phone, Teacher_Email)
  2  VALUES (108, 'Dr. Manny Costa', 'Mobile Application Development', '9087666383', 'mannycosta@gmail.com');

1 row created.
```

*Figure 18 Inserting Data in Teacher Table*

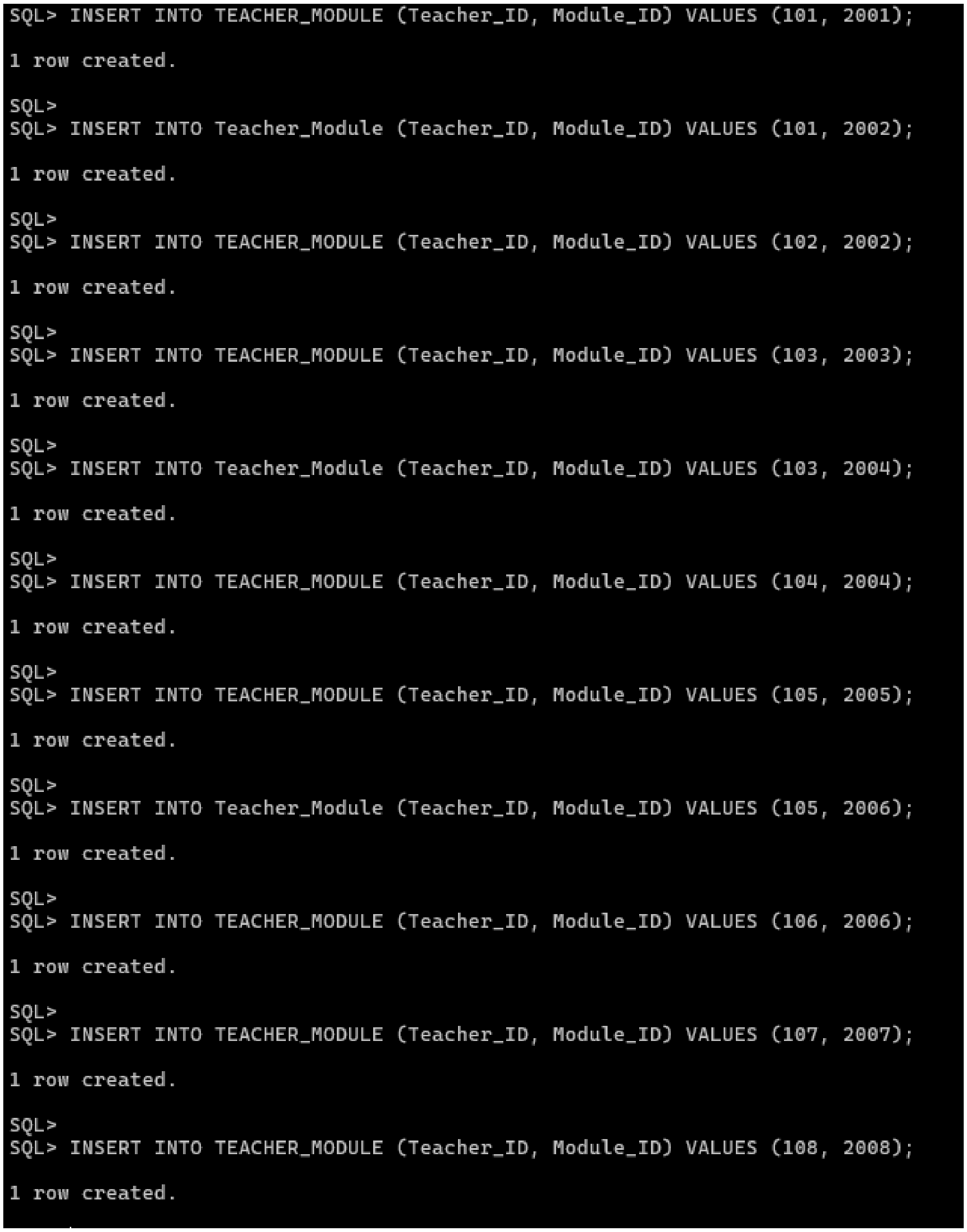
## 5.3.5 Inserting Data in Program_Module Table



```
SQL> INSERT INTO PROGRAM_MODULE (Program_ID, Module_ID) VALUES (1, 2001);

1 row created.

SQL>
SQL> INSERT INTO PROGRAM_MODULE (Program_ID, Module_ID) VALUES (1, 2002);

1 row created.

SQL>
SQL> INSERT INTO PROGRAM_MODULE (Program_ID, Module_ID) VALUES (1, 2009);

1 row created.

SQL>
SQL> INSERT INTO PROGRAM_MODULE (Program_ID, Module_ID) VALUES (2, 2003);

1 row created.

SQL>
SQL> INSERT INTO PROGRAM_MODULE (Program_ID, Module_ID) VALUES (2, 2004);

1 row created.

SQL>
SQL> INSERT INTO PROGRAM_MODULE (Program_ID, Module_ID) VALUES (3, 2005);

1 row created.

SQL>
SQL> INSERT INTO PROGRAM_MODULE (Program_ID, Module_ID) VALUES (3, 2006);

1 row created.

SQL>
SQL> INSERT INTO PROGRAM_MODULE (Program_ID, Module_ID) VALUES (4, 2007);

1 row created.

SQL>
SQL> INSERT INTO PROGRAM_MODULE (Program_ID, Module_ID) VALUES (4, 2008);

1 row created.

SQL>
SQL> INSERT INTO PROGRAM_MODULE (Program_ID, Module_ID) VALUES (4, 2009);

1 row created.
```

*Figure 19 Inserting data in Program_Module Table*

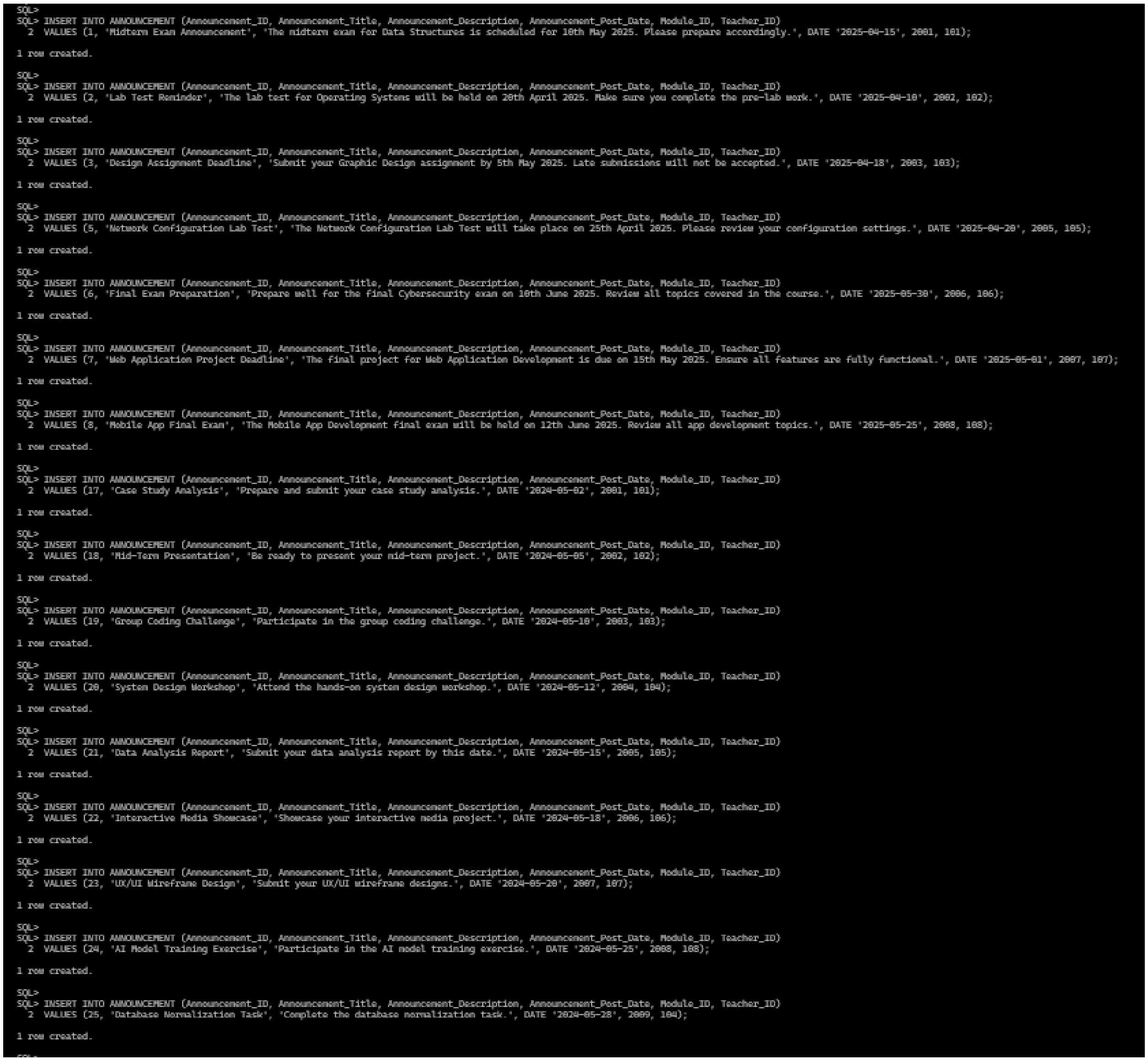
### 5.3.6 Inserting Data in Teacher_Module Table



```
SQL> INSERT INTO TEACHER_MODULE (Teacher_ID, Module_ID) VALUES (101, 2001);

1 row created.

SQL>
SQL> INSERT INTO Teacher_Module (Teacher_ID, Module_ID) VALUES (101, 2002);

1 row created.

SQL>
SQL> INSERT INTO TEACHER_MODULE (Teacher_ID, Module_ID) VALUES (102, 2002);

1 row created.

SQL>
SQL> INSERT INTO TEACHER_MODULE (Teacher_ID, Module_ID) VALUES (103, 2003);

1 row created.

SQL>
SQL> INSERT INTO Teacher_Module (Teacher_ID, Module_ID) VALUES (103, 2004);

1 row created.

SQL>
SQL> INSERT INTO TEACHER_MODULE (Teacher_ID, Module_ID) VALUES (104, 2004);

1 row created.

SQL>
SQL> INSERT INTO TEACHER_MODULE (Teacher_ID, Module_ID) VALUES (105, 2005);

1 row created.

SQL>
SQL> INSERT INTO Teacher_Module (Teacher_ID, Module_ID) VALUES (105, 2006);

1 row created.

SQL>
SQL> INSERT INTO TEACHER_MODULE (Teacher_ID, Module_ID) VALUES (106, 2006);

1 row created.

SQL>
SQL> INSERT INTO TEACHER_MODULE (Teacher_ID, Module_ID) VALUES (107, 2007);

1 row created.

SQL>
SQL> INSERT INTO TEACHER_MODULE (Teacher_ID, Module_ID) VALUES (108, 2008);

1 row created.
```

*Figure 20 Inserting Data in Teacher_Module Table*

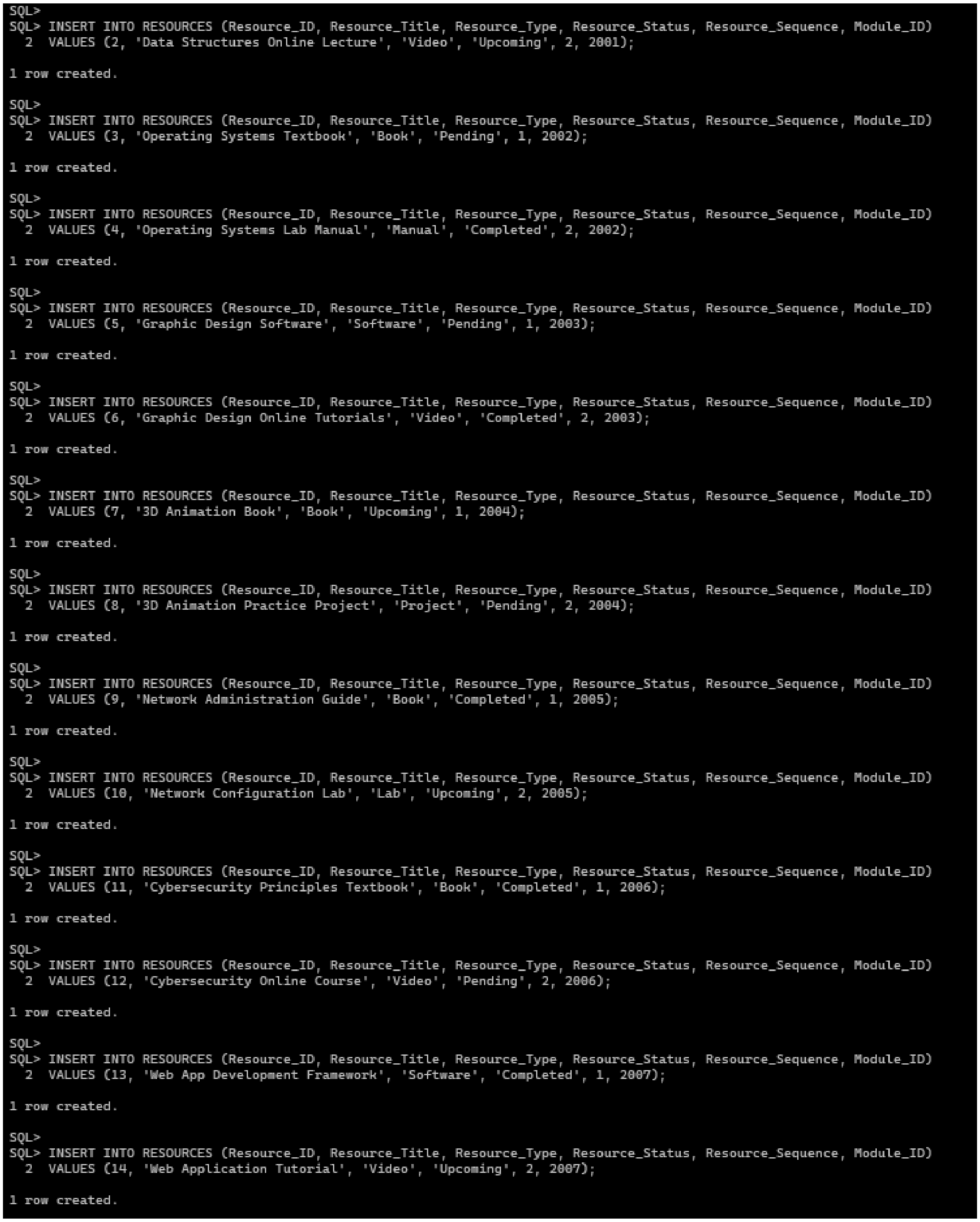
## 5.3.7 Inserting Data in Announcement Table



```
SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (1, 'Midterm Exam Announcement', 'The midterm exam for Data Structures is scheduled for 10th May 2025. Please prepare accordingly.', DATE '2025-04-15', 2001, 101);

1 row created.

SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (2, 'Lab Test Reminder', 'The lab test for Operating Systems will be held on 20th April 2025. Make sure you complete the pre-lab work.', DATE '2025-04-10', 2002, 102);

1 row created.

SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (3, 'Design Assignment Deadline', 'Submit your Graphic Design assignment by 5th May 2025. Late submissions will not be accepted.', DATE '2025-04-18', 2003, 103);

1 row created.

SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (5, 'Network Configuration Lab Test', 'The Network Configuration Lab Test will take place on 25th April 2025. Please review your configuration settings.', DATE '2025-04-20', 2005, 105);

1 row created.

SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (6, 'Final Exam Preparation', 'Prepare well for the final Cybersecurity exam on 10th June 2025. Review all topics covered in the course.', DATE '2025-05-30', 2006, 106);

1 row created.

SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (7, 'Web Application Project Deadline', 'The final project for Web Application Development is due on 15th May 2025. Ensure all features are fully functional.', DATE '2025-05-01', 2007, 107);

1 row created.

SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (8, 'Mobile App Final Exam', 'The Mobile App Development final exam will be held on 12th June 2025. Review all app development topics.', DATE '2025-05-25', 2008, 108);

1 row created.

SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (17, 'Case Study Analysis', 'Prepare and submit your case study analysis.', DATE '2024-05-02', 2001, 101);

1 row created.

SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (18, 'Mid-Term Presentation', 'Be ready to present your mid-term project.', DATE '2024-05-05', 2002, 102);

1 row created.

SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (19, 'Group Coding Challenge', 'Participate in the group coding challenge.', DATE '2024-05-10', 2003, 103);

1 row created.

SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (20, 'System Design Workshop', 'Attend the hands-on system design workshop.', DATE '2024-05-12', 2004, 104);

1 row created.

SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (21, 'Data Analysis Report', 'Submit your data analysis report by this date.', DATE '2024-05-15', 2005, 105);

1 row created.

SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (22, 'Interactive Media Showcase', 'Showcase your interactive media project.', DATE '2024-05-18', 2006, 106);

1 row created.

SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (23, 'UX/UI Wireframe Design', 'Submit your UX/UI wireframe designs.', DATE '2024-05-20', 2007, 107);

1 row created.

SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (24, 'AI Model Training Exercise', 'Participate in the AI model training exercise.', DATE '2024-05-25', 2008, 108);

1 row created.

SQL>
SQL> INSERT INTO ANNOUNCEMENT (Announcement_ID, Announcement_Title, Announcement_Description, Announcement_Post_Date, Module_ID, Teacher_ID)
  2  VALUES (25, 'Database Normalization Task', 'Complete the database normalization task.', DATE '2024-05-28', 2009, 104);

1 row created.

SQL>
```

*Figure 21 Inserting Data in Announcement Table*

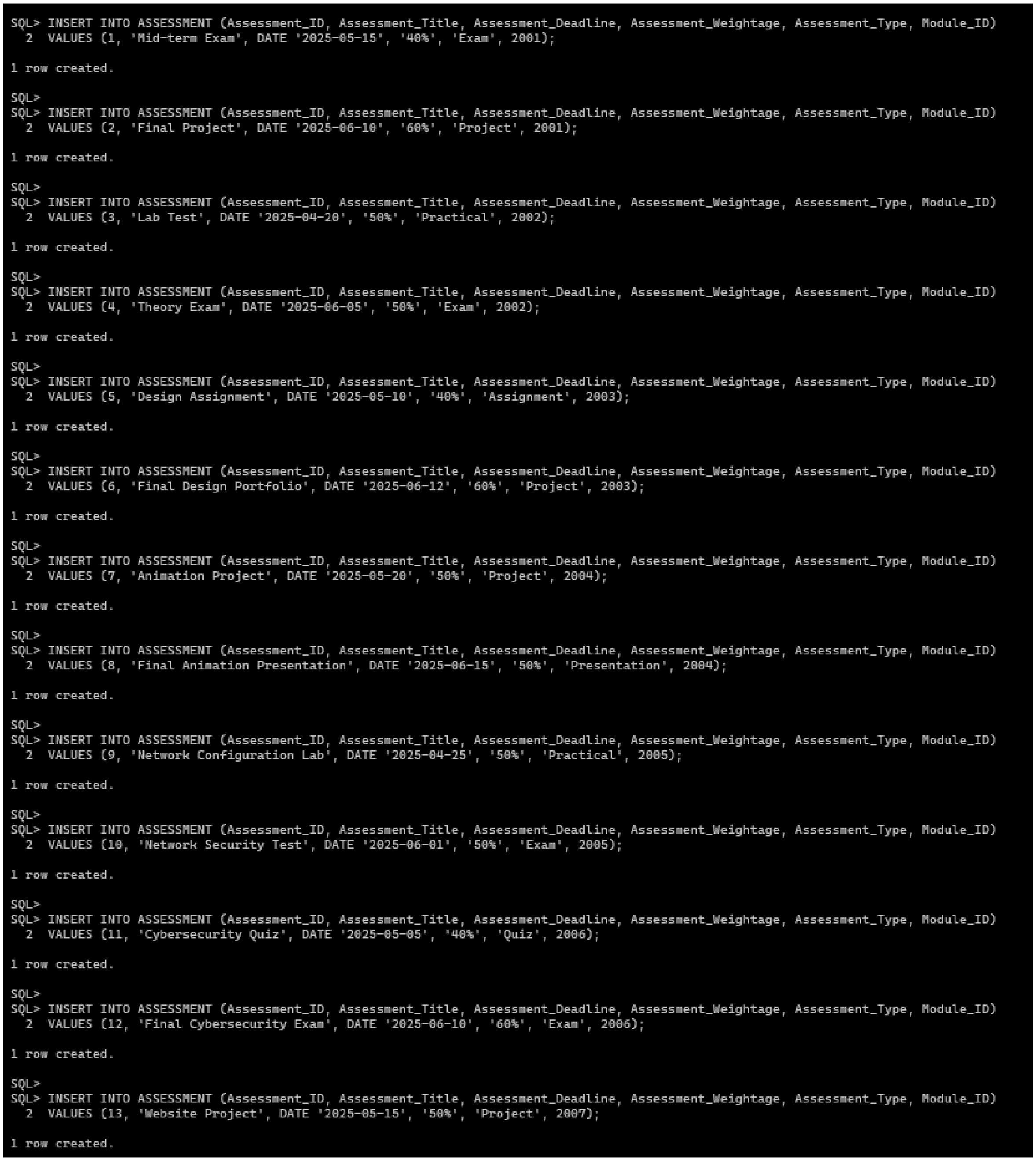
### 5.3.8 Inserting Data in Resources Table



```
SQL>
SQL> INSERT INTO RESOURCES (Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence, Module_ID)
  2  VALUES (2, 'Data Structures Online Lecture', 'Video', 'Upcoming', 2, 2001);

1 row created.

SQL>
SQL> INSERT INTO RESOURCES (Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence, Module_ID)
  2  VALUES (3, 'Operating Systems Textbook', 'Book', 'Pending', 1, 2002);

1 row created.

SQL>
SQL> INSERT INTO RESOURCES (Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence, Module_ID)
  2  VALUES (4, 'Operating Systems Lab Manual', 'Manual', 'Completed', 2, 2002);

1 row created.

SQL>
SQL> INSERT INTO RESOURCES (Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence, Module_ID)
  2  VALUES (5, 'Graphic Design Software', 'Software', 'Pending', 1, 2003);

1 row created.

SQL>
SQL> INSERT INTO RESOURCES (Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence, Module_ID)
  2  VALUES (6, 'Graphic Design Online Tutorials', 'Video', 'Completed', 2, 2003);

1 row created.

SQL>
SQL> INSERT INTO RESOURCES (Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence, Module_ID)
  2  VALUES (7, '3D Animation Book', 'Book', 'Upcoming', 1, 2004);

1 row created.

SQL>
SQL> INSERT INTO RESOURCES (Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence, Module_ID)
  2  VALUES (8, '3D Animation Practice Project', 'Project', 'Pending', 2, 2004);

1 row created.

SQL>
SQL> INSERT INTO RESOURCES (Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence, Module_ID)
  2  VALUES (9, 'Network Administration Guide', 'Book', 'Completed', 1, 2005);

1 row created.

SQL>
SQL> INSERT INTO RESOURCES (Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence, Module_ID)
  2  VALUES (10, 'Network Configuration Lab', 'Lab', 'Upcoming', 2, 2005);

1 row created.

SQL>
SQL> INSERT INTO RESOURCES (Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence, Module_ID)
  2  VALUES (11, 'Cybersecurity Principles Textbook', 'Book', 'Completed', 1, 2006);

1 row created.

SQL>
SQL> INSERT INTO RESOURCES (Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence, Module_ID)
  2  VALUES (12, 'Cybersecurity Online Course', 'Video', 'Pending', 2, 2006);

1 row created.

SQL>
SQL> INSERT INTO RESOURCES (Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence, Module_ID)
  2  VALUES (13, 'Web App Development Framework', 'Software', 'Completed', 1, 2007);

1 row created.

SQL>
SQL> INSERT INTO RESOURCES (Resource_ID, Resource_Title, Resource_Type, Resource_Status, Resource_Sequence, Module_ID)
  2  VALUES (14, 'Web Application Tutorial', 'Video', 'Upcoming', 2, 2007);

1 row created.
```

*Figure 22 Inserting Data in Resources Table*

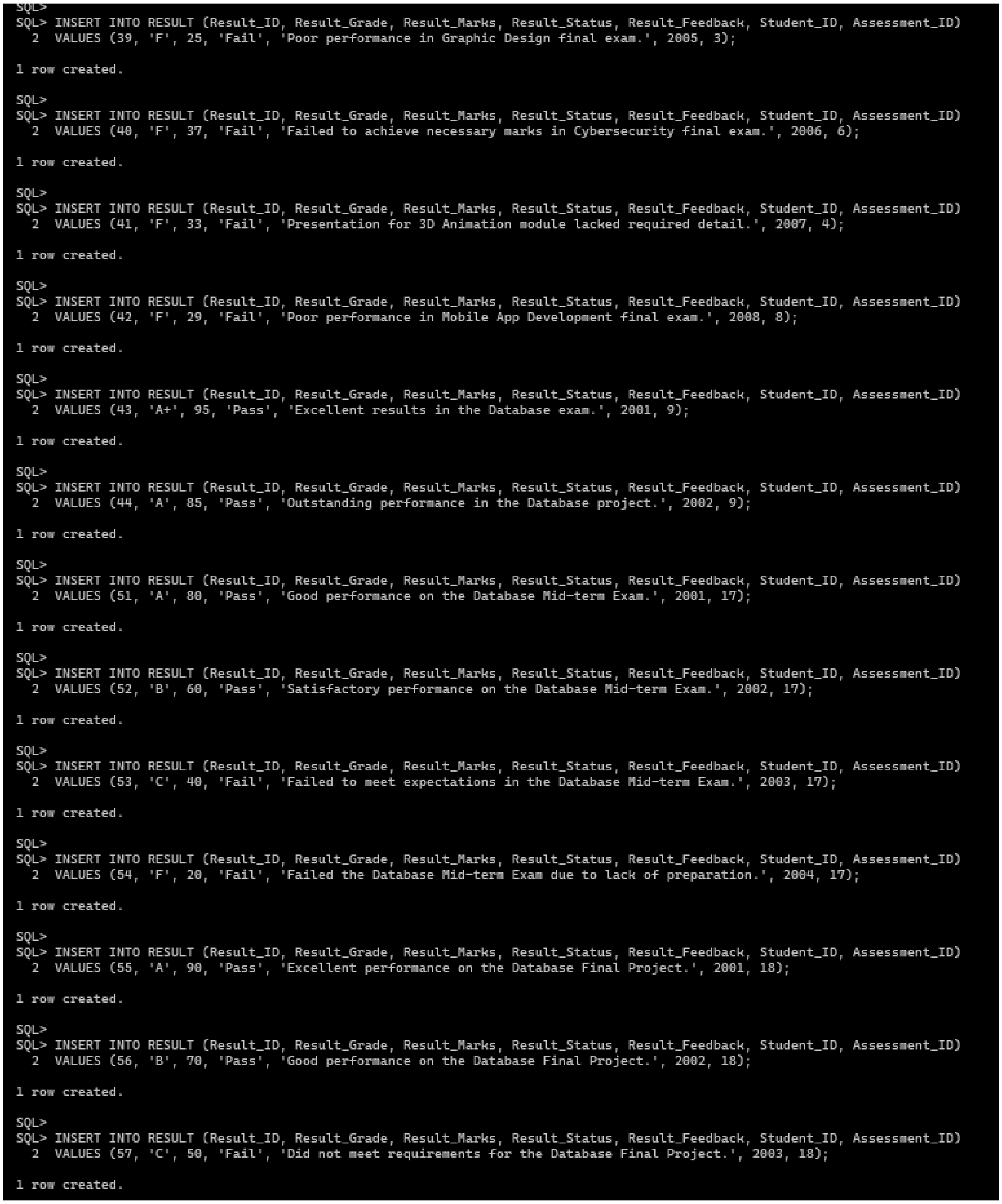
### 5.3.9 Inserting Data in Assessment Table



```
SQL> INSERT INTO ASSESSMENT (Assessment_ID, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type, Module_ID)
  2  VALUES (1, 'Mid-term Exam', DATE '2025-05-15', '40%', 'Exam', 2001);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT (Assessment_ID, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type, Module_ID)
  2  VALUES (2, 'Final Project', DATE '2025-06-10', '60%', 'Project', 2001);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT (Assessment_ID, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type, Module_ID)
  2  VALUES (3, 'Lab Test', DATE '2025-04-20', '50%', 'Practical', 2002);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT (Assessment_ID, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type, Module_ID)
  2  VALUES (4, 'Theory Exam', DATE '2025-06-05', '50%', 'Exam', 2002);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT (Assessment_ID, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type, Module_ID)
  2  VALUES (5, 'Design Assignment', DATE '2025-05-10', '40%', 'Assignment', 2003);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT (Assessment_ID, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type, Module_ID)
  2  VALUES (6, 'Final Design Portfolio', DATE '2025-06-12', '60%', 'Project', 2003);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT (Assessment_ID, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type, Module_ID)
  2  VALUES (7, 'Animation Project', DATE '2025-05-20', '50%', 'Project', 2004);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT (Assessment_ID, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type, Module_ID)
  2  VALUES (8, 'Final Animation Presentation', DATE '2025-06-15', '50%', 'Presentation', 2004);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT (Assessment_ID, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type, Module_ID)
  2  VALUES (9, 'Network Configuration Lab', DATE '2025-04-25', '50%', 'Practical', 2005);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT (Assessment_ID, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type, Module_ID)
  2  VALUES (10, 'Network Security Test', DATE '2025-06-01', '50%', 'Exam', 2005);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT (Assessment_ID, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type, Module_ID)
  2  VALUES (11, 'Cybersecurity Quiz', DATE '2025-05-05', '40%', 'Quiz', 2006);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT (Assessment_ID, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type, Module_ID)
  2  VALUES (12, 'Final Cybersecurity Exam', DATE '2025-06-10', '60%', 'Exam', 2006);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT (Assessment_ID, Assessment_Title, Assessment_Deadline, Assessment_Weightage, Assessment_Type, Module_ID)
  2  VALUES (13, 'Website Project', DATE '2025-05-15', '50%', 'Project', 2007);

1 row created.
```

*Figure 23 Inserting Data in Assessment Table*

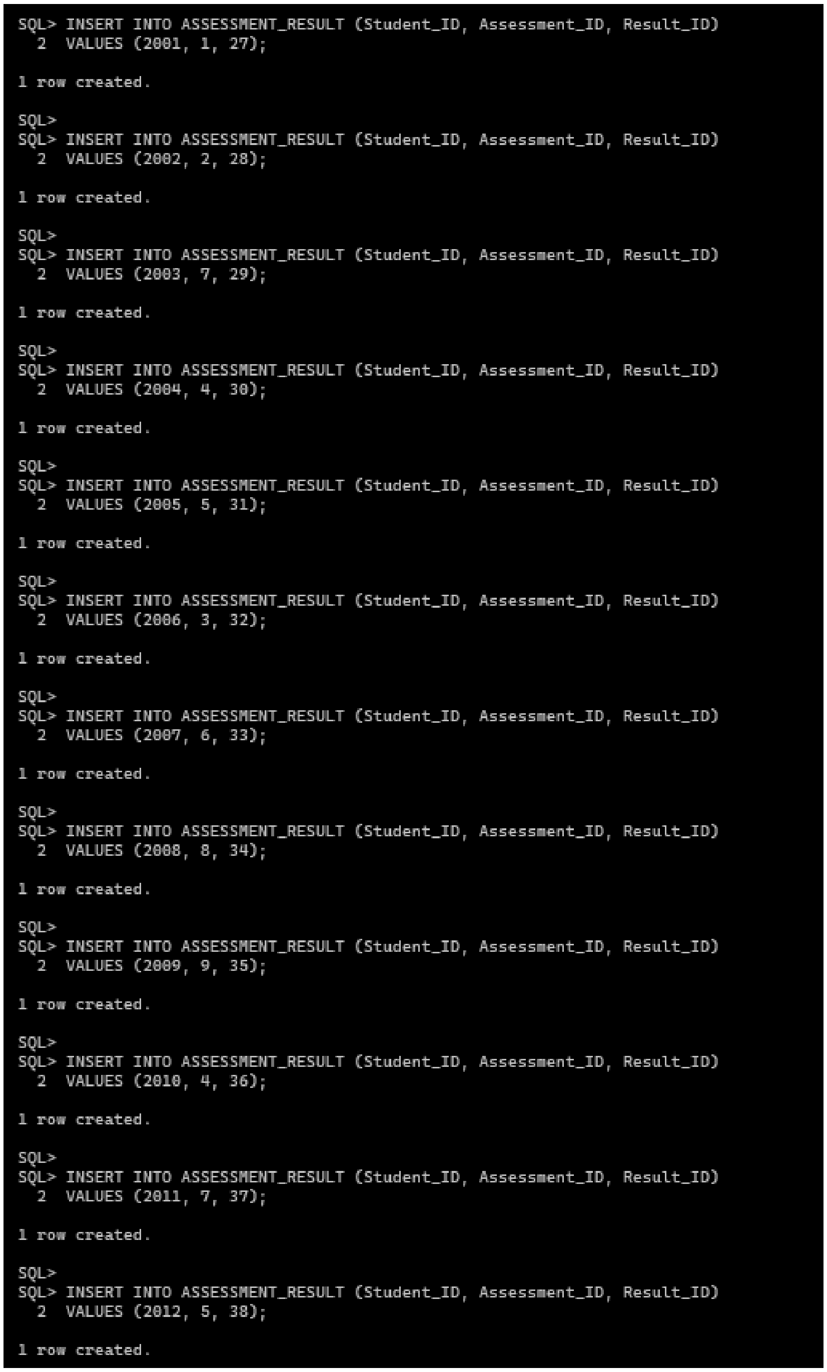
### 5.3.10 Inserting Data in Result Table



```
SQL>
SQL> INSERT INTO RESULT (Result_ID, Result_Grade, Result_Marks, Result_Status, Result_Feedback, Student_ID, Assessment_ID)
  2  VALUES (39, 'F', 25, 'Fail', 'Poor performance in Graphic Design final exam.', 2005, 3);

1 row created.

SQL>
SQL> INSERT INTO RESULT (Result_ID, Result_Grade, Result_Marks, Result_Status, Result_Feedback, Student_ID, Assessment_ID)
  2  VALUES (40, 'F', 37, 'Fail', 'Failed to achieve necessary marks in Cybersecurity final exam.', 2006, 6);

1 row created.

SQL>
SQL> INSERT INTO RESULT (Result_ID, Result_Grade, Result_Marks, Result_Status, Result_Feedback, Student_ID, Assessment_ID)
  2  VALUES (41, 'F', 33, 'Fail', 'Presentation for 3D Animation module lacked required detail.', 2007, 4);

1 row created.

SQL>
SQL> INSERT INTO RESULT (Result_ID, Result_Grade, Result_Marks, Result_Status, Result_Feedback, Student_ID, Assessment_ID)
  2  VALUES (42, 'F', 29, 'Fail', 'Poor performance in Mobile App Development final exam.', 2008, 8);

1 row created.

SQL>
SQL> INSERT INTO RESULT (Result_ID, Result_Grade, Result_Marks, Result_Status, Result_Feedback, Student_ID, Assessment_ID)
  2  VALUES (43, 'A+', 95, 'Pass', 'Excellent results in the Database exam.', 2001, 9);

1 row created.

SQL>
SQL> INSERT INTO RESULT (Result_ID, Result_Grade, Result_Marks, Result_Status, Result_Feedback, Student_ID, Assessment_ID)
  2  VALUES (44, 'A', 85, 'Pass', 'Outstanding performance in the Database project.', 2002, 9);

1 row created.

SQL>
SQL> INSERT INTO RESULT (Result_ID, Result_Grade, Result_Marks, Result_Status, Result_Feedback, Student_ID, Assessment_ID)
  2  VALUES (51, 'A', 80, 'Pass', 'Good performance on the Database Mid-term Exam.', 2001, 17);

1 row created.

SQL>
SQL> INSERT INTO RESULT (Result_ID, Result_Grade, Result_Marks, Result_Status, Result_Feedback, Student_ID, Assessment_ID)
  2  VALUES (52, 'B', 60, 'Pass', 'Satisfactory performance on the Database Mid-term Exam.', 2002, 17);

1 row created.

SQL>
SQL> INSERT INTO RESULT (Result_ID, Result_Grade, Result_Marks, Result_Status, Result_Feedback, Student_ID, Assessment_ID)
  2  VALUES (53, 'C', 40, 'Fail', 'Failed to meet expectations in the Database Mid-term Exam.', 2003, 17);

1 row created.

SQL>
SQL> INSERT INTO RESULT (Result_ID, Result_Grade, Result_Marks, Result_Status, Result_Feedback, Student_ID, Assessment_ID)
  2  VALUES (54, 'F', 20, 'Fail', 'Failed the Database Mid-term Exam due to lack of preparation.', 2004, 17);

1 row created.

SQL>
SQL> INSERT INTO RESULT (Result_ID, Result_Grade, Result_Marks, Result_Status, Result_Feedback, Student_ID, Assessment_ID)
  2  VALUES (55, 'A', 90, 'Pass', 'Excellent performance on the Database Final Project.', 2001, 18);

1 row created.

SQL>
SQL> INSERT INTO RESULT (Result_ID, Result_Grade, Result_Marks, Result_Status, Result_Feedback, Student_ID, Assessment_ID)
  2  VALUES (56, 'B', 70, 'Pass', 'Good performance on the Database Final Project.', 2002, 18);

1 row created.

SQL>
SQL> INSERT INTO RESULT (Result_ID, Result_Grade, Result_Marks, Result_Status, Result_Feedback, Student_ID, Assessment_ID)
  2  VALUES (57, 'C', 50, 'Fail', 'Did not meet requirements for the Database Final Project.', 2003, 18);

1 row created.
```

*Figure 24 Inserting Data in Result Table*

### 5.3.11 Inserting Data in Assessment_Result Table

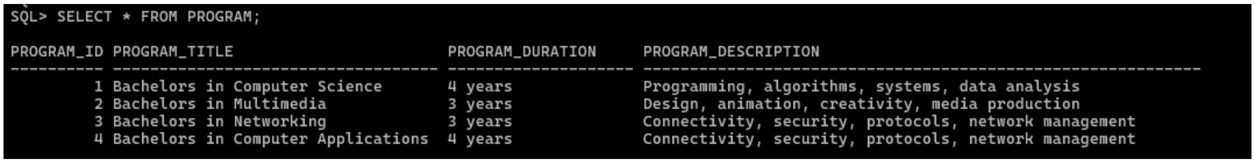


```
SQL> INSERT INTO ASSESSMENT_RESULT (Student_ID, Assessment_ID, Result_ID)
  2  VALUES (2001, 1, 27);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (Student_ID, Assessment_ID, Result_ID)
  2  VALUES (2002, 2, 28);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (Student_ID, Assessment_ID, Result_ID)
  2  VALUES (2003, 7, 29);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (Student_ID, Assessment_ID, Result_ID)
  2  VALUES (2004, 4, 30);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (Student_ID, Assessment_ID, Result_ID)
  2  VALUES (2005, 5, 31);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (Student_ID, Assessment_ID, Result_ID)
  2  VALUES (2006, 3, 32);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (Student_ID, Assessment_ID, Result_ID)
  2  VALUES (2007, 6, 33);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (Student_ID, Assessment_ID, Result_ID)
  2  VALUES (2008, 8, 34);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (Student_ID, Assessment_ID, Result_ID)
  2  VALUES (2009, 9, 35);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (Student_ID, Assessment_ID, Result_ID)
  2  VALUES (2010, 4, 36);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (Student_ID, Assessment_ID, Result_ID)
  2  VALUES (2011, 7, 37);

1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (Student_ID, Assessment_ID, Result_ID)
  2  VALUES (2012, 5, 38);

1 row created.
```

*Figure 25 Inserting Data in Assessment_Result Table*

# 5.4 Displaying Tables

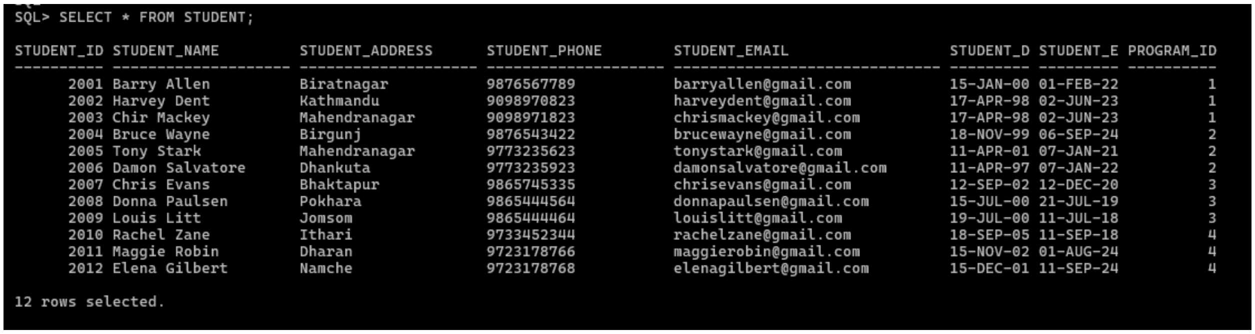
## 5.4.1 Displaying Program Tables



```
SQL> SELECT * FROM PROGRAM;

PROGRAM_ID PROGRAM_TITLE                       PROGRAM_DURATION     PROGRAM_DESCRIPTION
---------- ----------------------------------- -------------------- ------------------------------------------------------------
         1 Bachelors in Computer Science       4 years              Programming, algorithms, systems, data analysis
         2 Bachelors in Multimedia             3 years              Design, animation, creativity, media production
         3 Bachelors in Networking             3 years              Connectivity, security, protocols, network management
         4 Bachelors in Computer Applications  4 years              Connectivity, security, protocols, network management
```

*Figure 26 Displaying Program Tables*

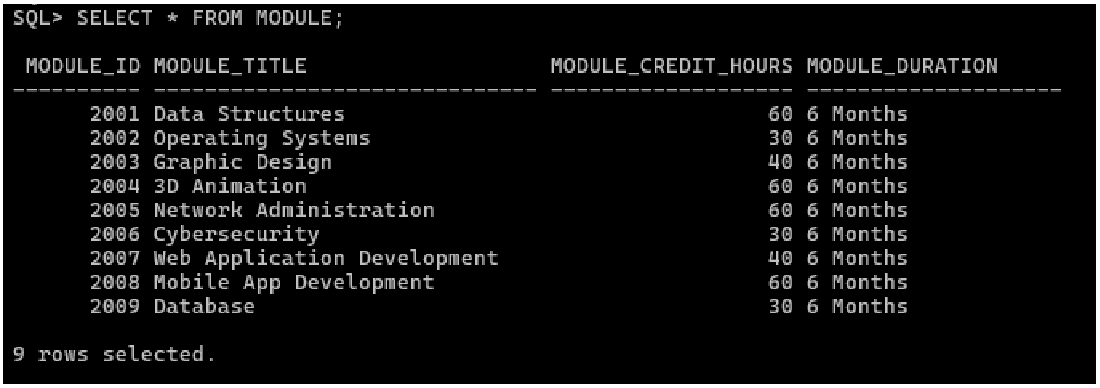
## 5.4.2 Displaying Student Tables



```
SQL> SELECT * FROM STUDENT;

STUDENT_ID STUDENT_NAME         STUDENT_ADDRESS      STUDENT_PHONE        STUDENT_EMAIL                  STUDENT_D STUDENT_E PROGRAM_ID
---------- -------------------- -------------------- -------------------- ------------------------------ --------- --------- ----------
      2001 Barry Allen          Biratnagar           9876567789           barryallen@gmail.com           15-JAN-00 01-FEB-22          1
      2002 Harvey Dent          Kathmandu            9098970823           harveydent@gmail.com           17-APR-98 02-JUN-23          1
      2003 Chir Mackey          Mahendranagar        9098971823           chrismackey@gmail.com          17-APR-98 02-JUN-23          1
      2004 Bruce Wayne          Birgunj              9876543422           brucewayne@gmail.com           18-NOV-99 06-SEP-24          2
      2005 Tony Stark           Mahendranagar        9773235623           tonystark@gmail.com            11-APR-01 07-JAN-21          2
      2006 Damon Salvatore      Dhankuta             9773235923           damonsalvatore@gmail.com       11-APR-97 07-JAN-22          2
      2007 Chris Evans          Bhaktapur            9865745335           chrisevans@gmail.com           12-SEP-02 12-DEC-20          3
      2008 Donna Paulsen        Pokhara              9865444564           donnapaulsen@gmail.com         15-JUL-00 21-JUL-19          3
      2009 Louis Litt           Jomsom               9865444464           louislitt@gmail.com            19-JUL-00 11-JUL-18          3
      2010 Rachel Zane          Ithari               9733452344           rachelzane@gmail.com           18-SEP-05 11-SEP-18          4
      2011 Maggie Robin         Dharan               9723178766           maggierobin@gmail.com          15-NOV-02 01-AUG-24          4
      2012 Elena Gilbert        Namche               9723178768           elenagilbert@gmail.com         15-DEC-01 11-SEP-24          4

12 rows selected.
```

*Figure 27 Displaying Student Tables*

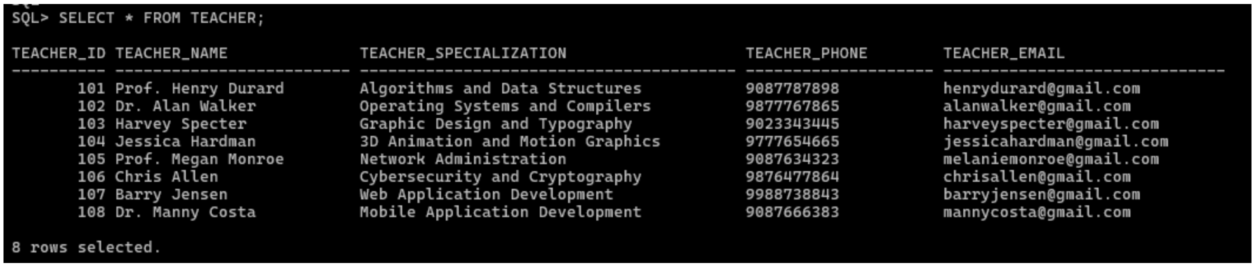
## 5.4.3 Displaying Module Tables



```
SQL> SELECT * FROM MODULE;

 MODULE_ID MODULE_TITLE                   MODULE_CREDIT_HOURS MODULE_DURATION
---------- ------------------------------ ------------------- --------------------
      2001 Data Structures                                 60 6 Months
      2002 Operating Systems                               30 6 Months
      2003 Graphic Design                                  40 6 Months
      2004 3D Animation                                    60 6 Months
      2005 Network Administration                          60 6 Months
      2006 Cybersecurity                                   30 6 Months
      2007 Web Application Development                     40 6 Months
      2008 Mobile App Development                          60 6 Months
      2009 Database                                        30 6 Months

9 rows selected.
```

*Figure 28 Displaying Module Tables*

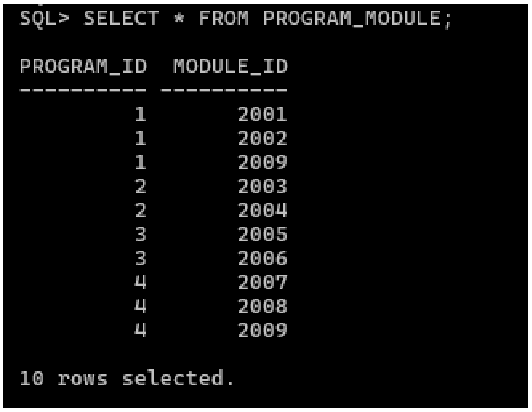
### 5.4.4 Displaying Teacher Table



```
SQL> SELECT * FROM TEACHER;

TEACHER_ID TEACHER_NAME              TEACHER_SPECIALIZATION                   TEACHER_PHONE        TEACHER_EMAIL
---------- ------------------------- ---------------------------------------- -------------------- ------------------------------
       101 Prof. Henry Durard        Algorithms and Data Structures           9087787898           henrydurard@gmail.com
       102 Dr. Alan Walker           Operating Systems and Compilers          9877767865           alanwalker@gmail.com
       103 Harvey Specter            Graphic Design and Typography            9023343445           harveyspecter@gmail.com
       104 Jessica Hardman           3D Animation and Motion Graphics         9777654665           jessicahardman@gmail.com
       105 Prof. Megan Monroe        Network Administration                   9087634323           melaniemonroe@gmail.com
       106 Chris Allen               Cybersecurity and Cryptography           9876477864           chrisallen@gmail.com
       107 Barry Jensen              Web Application Development              9988738843           barryjensen@gmail.com
       108 Dr. Manny Costa           Mobile Application Development           9087666383           mannycosta@gmail.com

8 rows selected.
```

*Figure 29 Displaying Teacher Table*

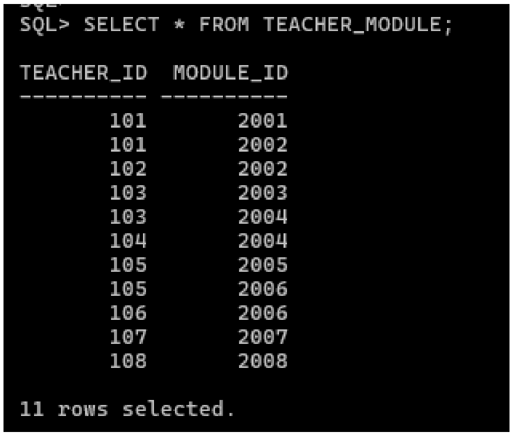
### 5.4.5 Displaying Program_Module Table



```
SQL> SELECT * FROM PROGRAM_MODULE;

PROGRAM_ID  MODULE_ID
---------- ----------
         1       2001
         1       2002
         1       2009
         2       2003
         2       2004
         3       2005
         3       2006
         4       2007
         4       2008
         4       2009

10 rows selected.
```

*Figure 30 Displaying Program_Module Table*

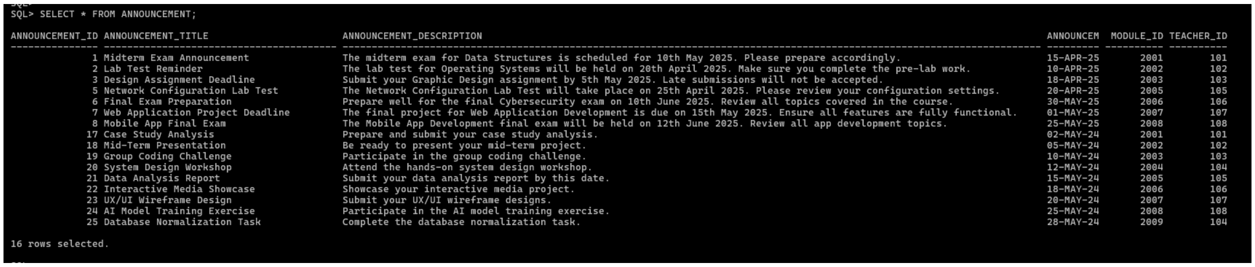
### 5.4.6 Displaying Teacher_Module Table



```
SQL> SELECT * FROM TEACHER_MODULE;

TEACHER_ID  MODULE_ID
---------- ----------
       101       2001
       101       2002
       102       2002
       103       2003
       103       2004
       104       2004
       105       2005
       105       2006
       106       2006
       107       2007
       108       2008

11 rows selected.
```

*Figure 31 Displaying Teacher_Module Table*

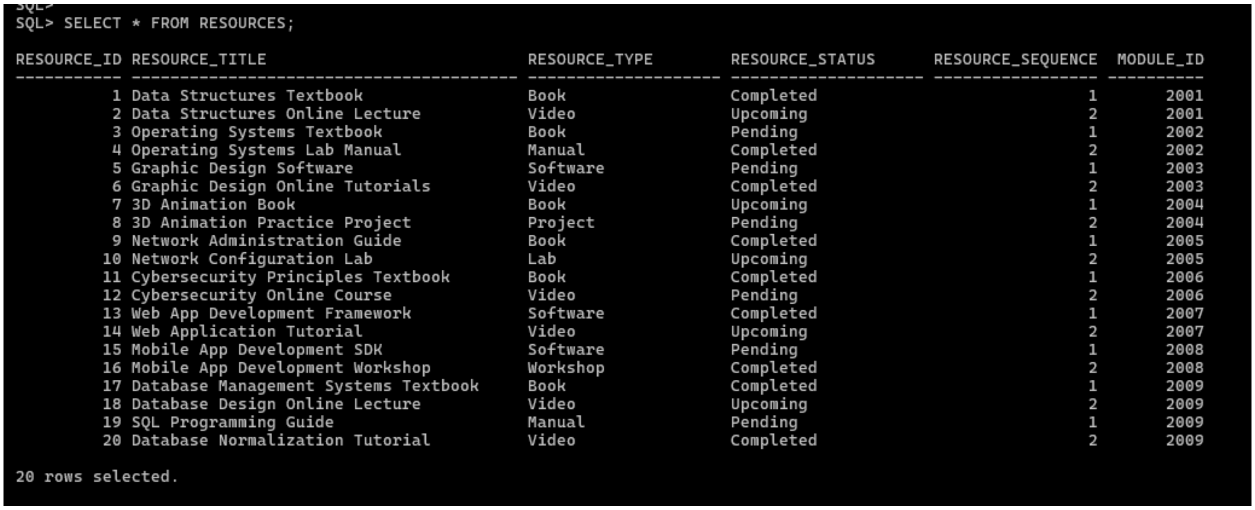
### 5.4.7 Displaying Announcement Table



```
SQL> SELECT * FROM ANNOUNCEMENT;

ANNOUNCEMENT_ID ANNOUNCEMENT_TITLE                       ANNOUNCEMENT_DESCRIPTION                                                                                                 ANNOUNCEM  MODULE_ID TEACHER_ID
--------------- ---------------------------------------- ------------------------------------------------------------------------------------------------------------------------ --------- ---------- ----------
              1 Midterm Exam Announcement                The midterm exam for Data Structures is scheduled for 10th May 2025. Please prepare accordingly.                         15-APR-25       2001        101
              2 Lab Test Reminder                        The lab test for Operating Systems will be held on 20th April 2025. Make sure you complete the pre-lab work.             10-APR-25       2002        102
              3 Design Assignment Deadline               Submit your Graphic Design assignment by 5th May 2025. Late submissions will not be accepted.                            18-APR-25       2003        103
              5 Network Configuration Lab Test           The Network Configuration Lab Test will take place on 25th April 2025. Please review your configuration settings.        20-APR-25       2005        105
              6 Final Exam Preparation                   Prepare well for the final Cybersecurity exam on 10th June 2025. Review all topics covered in the course.                30-MAY-25       2006        106
              7 Web Application Project Deadline         The final project for Web Application Development is due on 15th May 2025. Ensure all features are fully functional.     01-MAY-25       2007        107
              8 Mobile App Final Exam                    The Mobile App Development final exam will be held on 12th June 2025. Review all app development topics.                 25-MAY-25       2008        108
             17 Case Study Analysis                      Prepare and submit your case study analysis.                                                                             02-MAY-24       2001        101
             18 Mid-Term Presentation                    Be ready to present your mid-term project.                                                                               05-MAY-24       2002        102
             19 Group Coding Challenge                   Participate in the group coding challenge.                                                                               10-MAY-24       2003        103
             20 System Design Workshop                   Attend the hands-on system design workshop.                                                                              12-MAY-24       2004        104
             21 Data Analysis Report                     Submit your data analysis report by this date.                                                                           15-MAY-24       2005        105
             22 Interactive Media Showcase               Showcase your interactive media project.                                                                                 18-MAY-24       2006        106
             23 UX/UI Wireframe Design                   Submit your UX/UI wireframe designs.                                                                                     20-MAY-24       2007        107
             24 AI Model Training Exercise               Participate in the AI model training exercise.                                                                           25-MAY-24       2008        108
             25 Database Normalization Task              Complete the database normalization task.                                                                                28-MAY-24       2009        104

16 rows selected.
```

*Figure 32 Displaying Announcement Table*

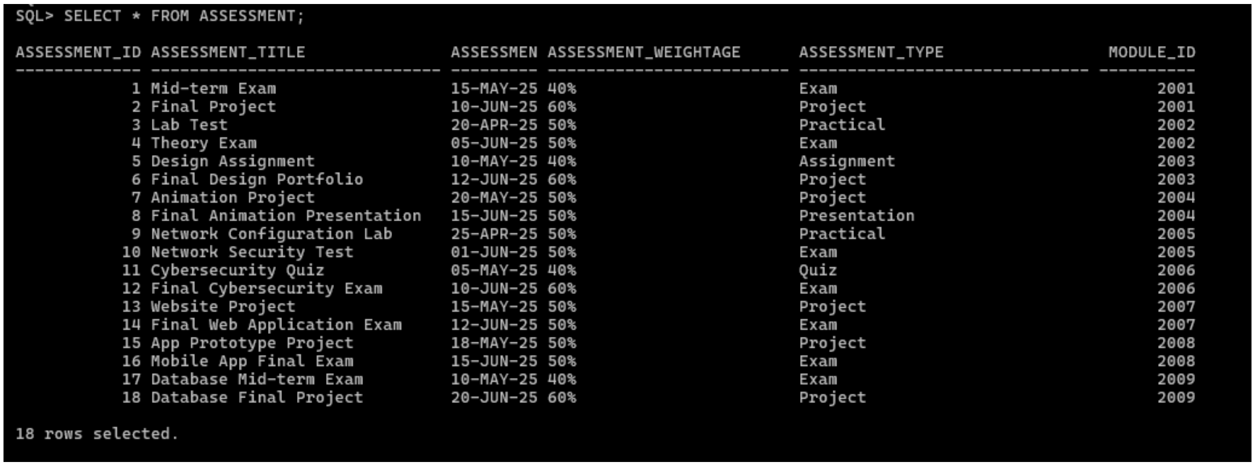
### 5.4.8 Displaying Resources Table



```
SQL>
SQL> SELECT * FROM RESOURCES;

RESOURCE_ID RESOURCE_TITLE                           RESOURCE_TYPE        RESOURCE_STATUS      RESOURCE_SEQUENCE  MODULE_ID
----------- ---------------------------------------- -------------------- -------------------- ----------------- ----------
          1 Data Structures Textbook                 Book                 Completed                            1       2001
          2 Data Structures Online Lecture           Video                Upcoming                             2       2001
          3 Operating Systems Textbook               Book                 Pending                              1       2002
          4 Operating Systems Lab Manual             Manual               Completed                            2       2002
          5 Graphic Design Software                  Software             Pending                              1       2003
          6 Graphic Design Online Tutorials          Video                Completed                            2       2003
          7 3D Animation Book                        Book                 Upcoming                             1       2004
          8 3D Animation Practice Project            Project              Pending                              2       2004
          9 Network Administration Guide             Book                 Completed                            1       2005
         10 Network Configuration Lab                Lab                  Upcoming                             2       2005
         11 Cybersecurity Principles Textbook        Book                 Completed                            1       2006
         12 Cybersecurity Online Course              Video                Pending                              2       2006
         13 Web App Development Framework            Software             Completed                            1       2007
         14 Web Application Tutorial                 Video                Upcoming                             2       2007
         15 Mobile App Development SDK               Software             Pending                              1       2008
         16 Mobile App Development Workshop          Workshop             Completed                            2       2008
         17 Database Management Systems Textbook     Book                 Completed                            1       2009
         18 Database Design Online Lecture           Video                Upcoming                             2       2009
         19 SQL Programming Guide                    Manual               Pending                              1       2009
         20 Database Normalization Tutorial          Video                Completed                            2       2009

20 rows selected.
```

*Figure 33 Displaying Resources Table*

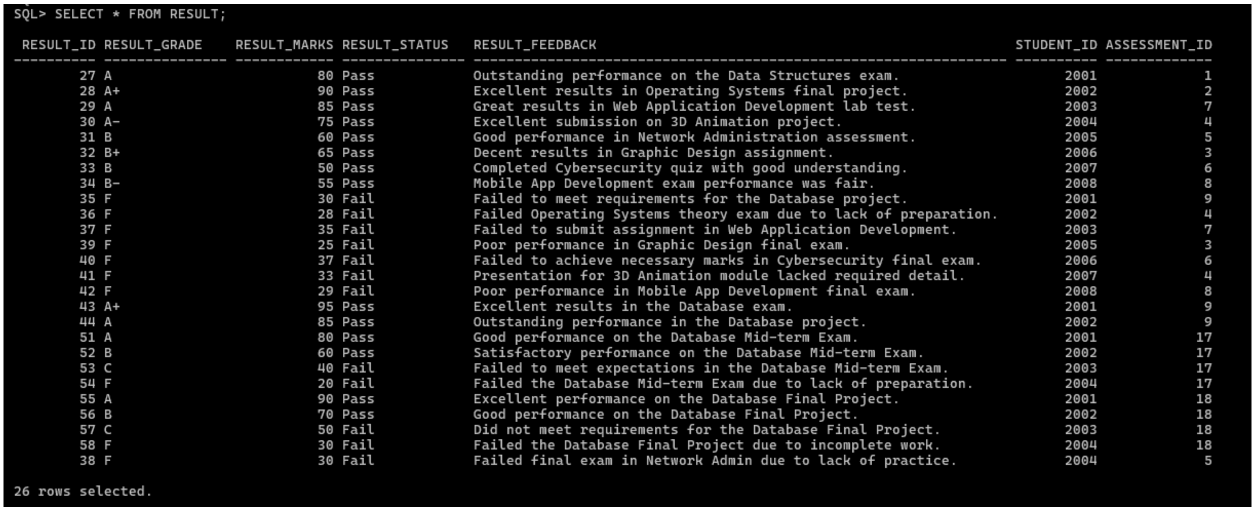
### 5.4.9 Displaying Assessment Table



```
SQL> SELECT * FROM ASSESSMENT;

ASSESSMENT_ID ASSESSMENT_TITLE               ASSESSMEN ASSESSMENT_WEIGHTAGE      ASSESSMENT_TYPE                 MODULE_ID
------------- ------------------------------ --------- ------------------------- ------------------------------ ----------
            1 Mid-term Exam                  15-MAY-25 40%                       Exam                                 2001
            2 Final Project                  10-JUN-25 60%                       Project                              2001
            3 Lab Test                       20-APR-25 50%                       Practical                            2002
            4 Theory Exam                    05-JUN-25 50%                       Exam                                 2002
            5 Design Assignment              10-MAY-25 40%                       Assignment                           2003
            6 Final Design Portfolio         12-JUN-25 60%                       Project                              2003
            7 Animation Project              20-MAY-25 50%                       Project                              2004
            8 Final Animation Presentation   15-JUN-25 50%                       Presentation                         2004
            9 Network Configuration Lab      25-APR-25 50%                       Practical                            2005
           10 Network Security Test          01-JUN-25 50%                       Exam                                 2005
           11 Cybersecurity Quiz             05-MAY-25 40%                       Quiz                                 2006
           12 Final Cybersecurity Exam       10-JUN-25 60%                       Exam                                 2006
           13 Website Project                15-MAY-25 50%                       Project                              2007
           14 Final Web Application Exam     12-JUN-25 50%                       Exam                                 2007
           15 App Prototype Project          18-MAY-25 50%                       Project                              2008
           16 Mobile App Final Exam          15-JUN-25 50%                       Exam                                 2008
           17 Database Mid-term Exam         10-MAY-25 40%                       Exam                                 2009
           18 Database Final Project         20-JUN-25 60%                       Project                              2009

18 rows selected.
```

*Figure 34 Displaying Assessment Table*

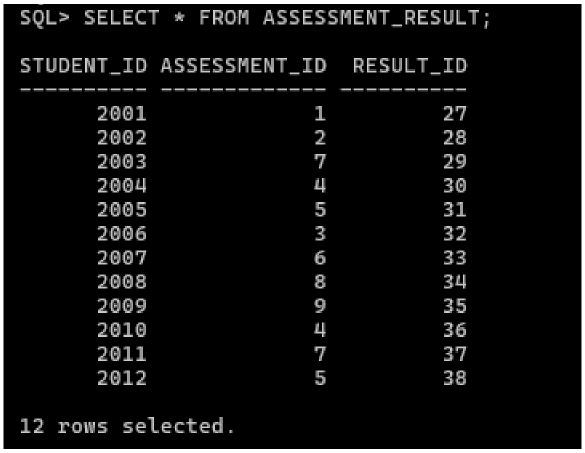
### 5.4.10 Displaying Result Table



```
SQL> SELECT * FROM RESULT;

 RESULT_ID RESULT_GRADE    RESULT_MARKS RESULT_STATUS   RESULT_FEEDBACK                                                   STUDENT_ID ASSESSMENT_ID
---------- --------------- ------------ --------------- ----------------------------------------------------------------- ---------- -------------
        27 A                         80 Pass            Outstanding performance on the Data Structures exam.                    2001             1
        28 A+                        90 Pass            Excellent results in Operating Systems final project.                   2002             2
        29 A                         85 Pass            Great results in Web Application Development lab test.                  2003             7
        30 A-                        75 Pass            Excellent submission on 3D Animation project.                           2004             4
        31 B                         60 Pass            Good performance in Network Administration assessment.                  2005             5
        32 B+                        65 Pass            Decent results in Graphic Design assignment.                            2006             3
        33 B                         50 Pass            Completed Cybersecurity quiz with good understanding.                   2007             6
        34 B-                        55 Pass            Mobile App Development exam performance was fair.                       2008             8
        35 F                         30 Fail            Failed to meet requirements for the Database project.                   2001             9
        36 F                         28 Fail            Failed Operating Systems theory exam due to lack of preparation.        2002             4
        37 F                         35 Fail            Failed to submit assignment in Web Application Development.             2003             7
        39 F                         25 Fail            Poor performance in Graphic Design final exam.                          2005             3
        40 F                         37 Fail            Failed to achieve necessary marks in Cybersecurity final exam.          2006             6
        41 F                         33 Fail            Presentation for 3D Animation module lacked required detail.            2007             4
        42 F                         29 Fail            Poor performance in Mobile App Development final exam.                  2008             8
        43 A+                        95 Pass            Excellent results in the Database exam.                                 2001             9
        44 A                         85 Pass            Outstanding performance in the Database project.                        2002             9
        51 A                         80 Pass            Good performance on the Database Mid-term Exam.                         2001            17
        52 B                         60 Pass            Satisfactory performance on the Database Mid-term Exam.                 2002            17
        53 C                         40 Fail            Failed to meet expectations in the Database Mid-term Exam.              2003            17
        54 F                         20 Fail            Failed the Database Mid-term Exam due to lack of preparation.           2004            17
        55 A                         90 Pass            Excellent performance on the Database Final Project.                    2001            18
        56 B                         70 Pass            Good performance on the Database Final Project.                         2002            18
        57 C                         50 Fail            Did not meet requirements for the Database Final Project.               2003            18
        58 F                         30 Fail            Failed the Database Final Project due to incomplete work.               2004            18
        38 F                         30 Fail            Failed final exam in Network Admin due to lack of practice.             2004             5

26 rows selected.
```

*Figure 35 Displaying Result Table*

### 5.4.11 Displaying Assessment_Result Table



```
SQL> SELECT * FROM ASSESSMENT_RESULT;

STUDENT_ID ASSESSMENT_ID  RESULT_ID
---------- ------------- ----------
      2001             1         27
      2002             2         28
      2003             7         29
      2004             4         30
      2005             5         31
      2006             3         32
      2007             6         33
      2008             8         34
      2009             9         35
      2010             4         36
      2011             7         37
      2012             5         38

12 rows selected.
```

*Figure 36 Displaying Assessment_Result Table*

# 6.0 Database Querying

## 6.1 Information Query

### 6.1.1 Listing available programs in college and total number of students enrolled

List the programs that are available in the college and the total number of students enrolled in each.

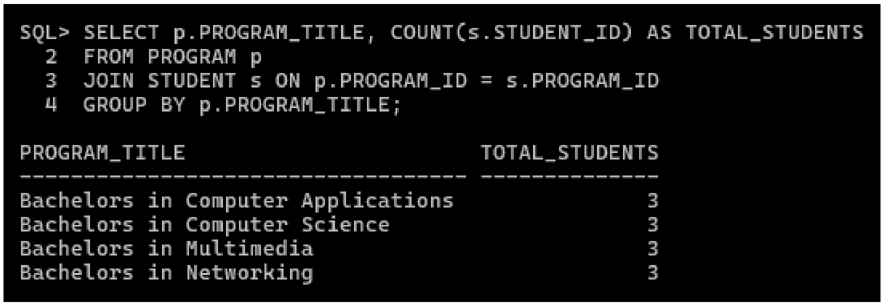
**Query**

SELECT p.PROGRAM_TITLE, COUNT(s.STUDENT_ID) AS TOTAL_STUDENTS

FROM PROGRAM p

JOIN STUDENT s ON p.PROGRAM_ID = s.PROGRAM_ID

GROUP BY p.PROGRAM_TITLE;



```
SQL> SELECT p.PROGRAM_TITLE, COUNT(s.STUDENT_ID) AS TOTAL_STUDENTS
  2  FROM PROGRAM p
  3  JOIN STUDENT s ON p.PROGRAM_ID = s.PROGRAM_ID
  4  GROUP BY p.PROGRAM_TITLE;

PROGRAM_TITLE                           TOTAL_STUDENTS
--------------------------------------- --------------
Bachelors in Computer Applications                   3
Bachelors in Computer Science                        3
Bachelors in Multimedia                              3
Bachelors in Networking                              3
```

*Figure 36 Displaying list of available programs and total number of students*

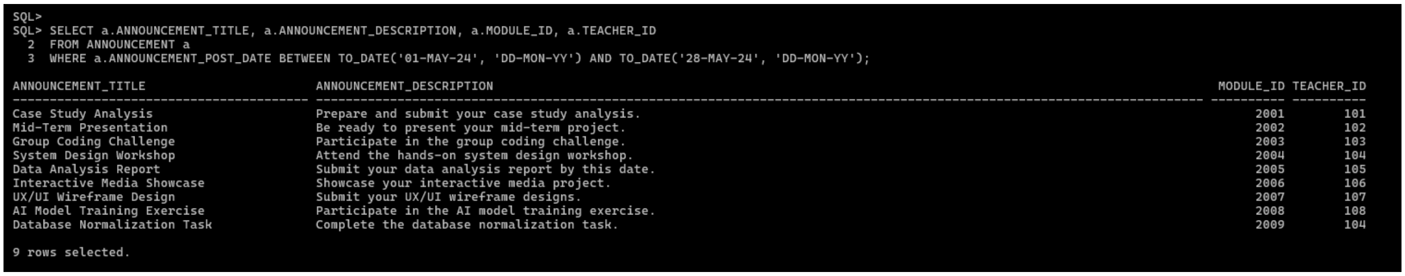
The query gives all the programs available in college along with the total number of students that are enrolled in each program. This data collates per program title for counting students enrolled therein. This is mainly because it gives knowledge about the popularity of each of these programs.

### 6.1.2 Listing all announcement of a particular module between 1st May 2024 to 28th May 2024

List all the announcements made for a particular module starting from 1st May 2024 to 28th May 2024.

**Query**

```
SELECT a.ANNOUNCEMENT_TITLE, a.ANNOUNCEMENT_DESCRIPTION, a.MODULE_ID, a.TEACHER_ID

FROM ANNOUNCEMENT a

WHERE a.ANNOUNCEMENT_POST_DATE BETWEEN TO_DATE('01-MAY-24', 'DD-MON-YY') AND TO_DATE('28-MAY-24', 'DD-MON-YY');
```



```
SQL>
SQL> SELECT a.ANNOUNCEMENT_TITLE, a.ANNOUNCEMENT_DESCRIPTION, a.MODULE_ID, a.TEACHER_ID
     FROM ANNOUNCEMENT a
     WHERE a.ANNOUNCEMENT_POST_DATE BETWEEN TO_DATE('01-MAY-24', 'DD-MON-YY') AND TO_DATE('28-MAY-24', 'DD-MON-YY');

ANNOUNCEMENT_TITLE                       ANNOUNCEMENT_DESCRIPTION                                                                                     MODULE_ID TEACHER_ID
---------------------------------------- ------------------------------------------------------------------------------------------------------------ ---------- ----------
Case Study Analysis                      Prepare and submit your case study analysis.                                                                      2001        101
Mid-Term Presentation                    Be ready to present your mid-term project.                                                                        2002        102
Group Coding Challenge                   Participate in the group coding challenge.                                                                        2003        103
System Design Workshop                   Attend the hands-on system design workshop.                                                                       2004        104
Data Analysis Report                     Submit your data analysis report by this date.                                                                    2005        105
Interactive Media Showcase               Showcase your interactive media project.                                                                          2006        106
UX/UI Wireframe Design                   Submit your UX/UI wireframe designs.                                                                              2007        107
AI Model Training Exercise               Participate in the AI model training exercise.                                                                    2008        108
Database Normalization Task              Complete the database normalization task.                                                                         2009        104

9 rows selected.
```

*Figure 37 Displaying list of announcement posted between 1st May 2024 to 28th May 2024*

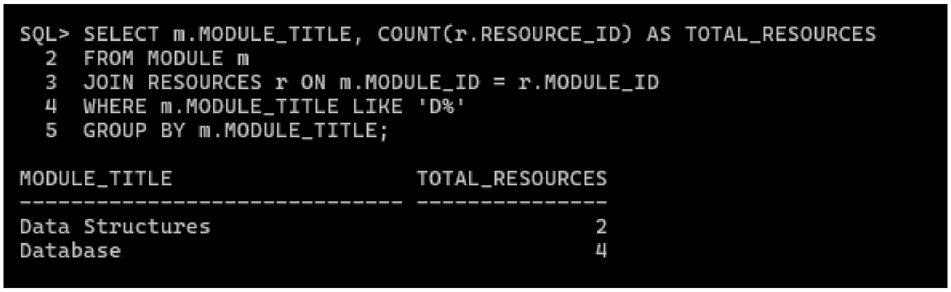
The query lists announcements made for a specific module-it brings up to date-all related announcements made between 1 May 2024 to 28 May 2024. It reflects one's dependence on date filtering because in essence no extraneous information will appear. For a follow-up of the concerned important updates for the specific period, this is true.

### 6.1.3 Listing module names with letter 'D' along with number of resources

List the names of all modules that begin with the letter 'D', along with the total number of resources uploaded for those modules.

**<u>Query</u>**

```
SELECT m.MODULE_TITLE, COUNT(r.RESOURCE_ID) AS TOTAL_RESOURCES

FROM MODULE m

JOIN RESOURCES r ON m.MODULE_ID = r.MODULE_ID

WHERE m.MODULE_TITLE LIKE 'D%'

GROUP BY m.MODULE_TITLE;
```



```
SQL> SELECT m.MODULE_TITLE, COUNT(r.RESOURCE_ID) AS TOTAL_RESOURCES
  2  FROM MODULE m
  3  JOIN RESOURCES r ON m.MODULE_ID = r.MODULE_ID
  4  WHERE m.MODULE_TITLE LIKE 'D%'
  5  GROUP BY m.MODULE_TITLE;

MODULE_TITLE                   TOTAL_RESOURCES
------------------------------ ---------------
Data Structures                              2
Database                                     4
```

*Figure 38 Displaying list of modules with initial letter 'D'*

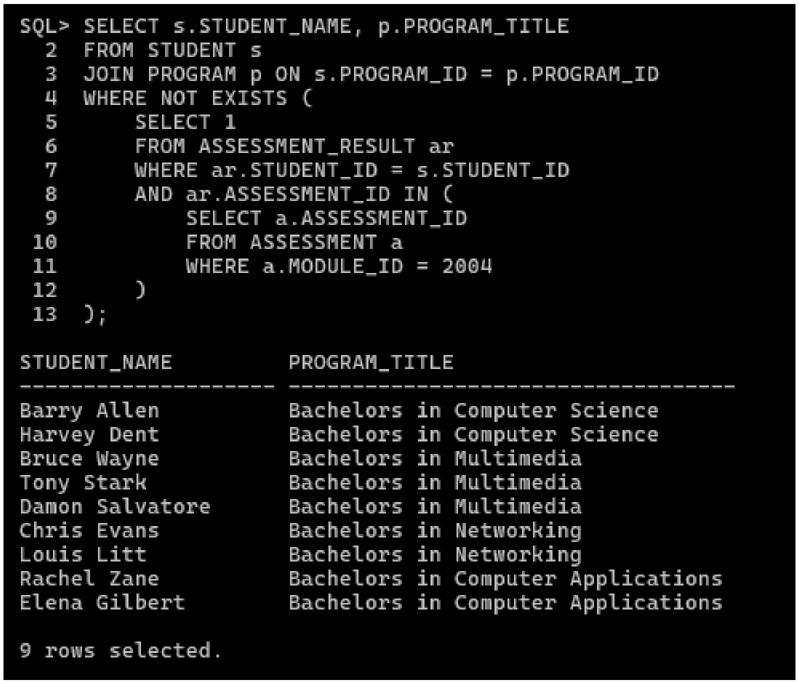
The query checks modules starting with letter 'D'. The title concerning all names beginning with the letter 'D' and the resources for each modules are counted. Grouping by module title makes sure that for all resources associated with that particular module, the computed figures are accurate. The picture shows what resources are available to a specific module.

### 6.1.4 Listing names of student along with enrolled programs who have not submitted assessment for particular module

List the names of all students along with their enrolled program who have not submitted any assessments for a particular module.

**Query**

```
SELECT s.STUDENT_NAME, p.PROGRAM_TITLE
FROM STUDENT s
JOIN PROGRAM p ON s.PROGRAM_ID = p.PROGRAM_ID
WHERE NOT EXISTS (
            SELECT 1
             FROM ASSESSMENT_RESULT ar
           WHERE ar.STUDENT_ID = s.STUDENT_ID
            AND ar.ASSESSMENT_ID IN (
                      SELECT a.ASSESSMENT_ID
                       FROM ASSESSMENT a
                      WHERE a.MODULE_ID = 2004
              )
);
```



```
SQL> SELECT s.STUDENT_NAME, p.PROGRAM_TITLE
     FROM STUDENT s
     JOIN PROGRAM p ON s.PROGRAM_ID = p.PROGRAM_ID
     WHERE NOT EXISTS (
         SELECT 1
         FROM ASSESSMENT_RESULT ar
         WHERE ar.STUDENT_ID = s.STUDENT_ID
         AND ar.ASSESSMENT_ID IN (
             SELECT a.ASSESSMENT_ID
             FROM ASSESSMENT a
             WHERE a.MODULE_ID = 2004
         )
     );

STUDENT_NAME         PROGRAM_TITLE
-------------------- ------------------------------------
Barry Allen          Bachelors in Computer Science
Harvey Dent          Bachelors in Computer Science
Bruce Wayne          Bachelors in Multimedia
Tony Stark           Bachelors in Multimedia
Damon Salvatore      Bachelors in Multimedia
Chris Evans          Bachelors in Networking
Louis Litt           Bachelors in Networking
Rachel Zane          Bachelors in Computer Applications
Elena Gilbert        Bachelors in Computer Applications

9 rows selected.
```

*Figure 39  Displaying list of students enrolled in program with no submitted assessments for particular module*

The query lists names of the students along with the programs they are enrolled in but have not submitted their assignments.

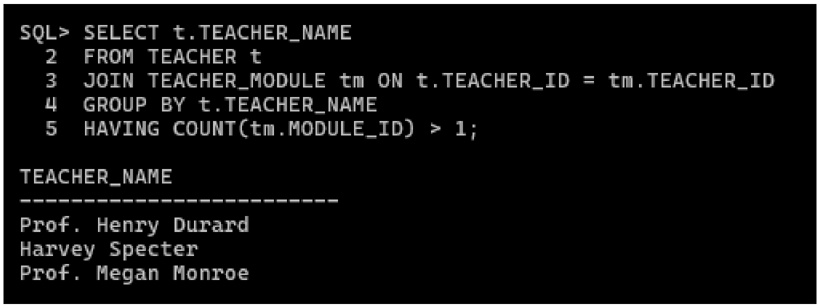
And the above subquery is for students with their corresponding programs who have not submitted even a single assessment for that course-specific module. It identifies students with no assessment submissions by using a subquery that checks for non-existence of submission. This is very useful in the case of monitoring the students' participation in a given module.

### 6.1.5 Listing teachers who teach more than one module

List all the teachers who teach more than one module.

**<u>Query</u>**

```
SELECT t.TEACHER_NAME

FROM TEACHER t

JOIN TEACHER_MODULE tm ON t.TEACHER_ID = tm.TEACHER_ID

GROUP BY t.TEACHER_NAME

HAVING COUNT(tm.MODULE_ID) > 1;
```



```
SQL> SELECT t.TEACHER_NAME
  2  FROM TEACHER t
  3  JOIN TEACHER_MODULE tm ON t.TEACHER_ID = tm.TEACHER_ID
  4  GROUP BY t.TEACHER_NAME
  5  HAVING COUNT(tm.MODULE_ID) > 1;

TEACHER_NAME
-------------------------
Prof. Henry Durard
Harvey Specter
Prof. Megan Monroe
```

*Figure 40 Displaying list of teacher teaching multiple modules*

This query returns the list of teachers that lectured in multiple modules by counting the assigned modules to each teacher. The HAVING clause allows only showing teachers that have more than one module on their list. This way, it will be easy to distinguish teachers' multi-teaching responsibilities.

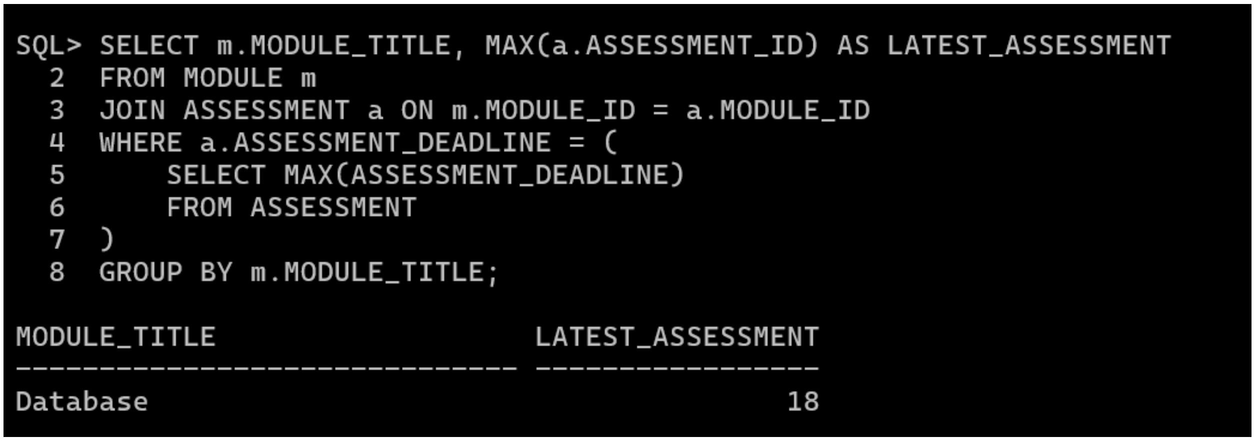
## 6.2 Transaction Query

### 6.2.1 Identification of mudule with latest deadline

Identify the module that has the latest assessment deadline.

**Query**

```
SELECT m.MODULE_TITLE, MAX(a.ASSESSMENT_ID) AS
LATEST_ASSESSMENT

FROM MODULE m

JOIN ASSESSMENT a ON m.MODULE_ID = a.MODULE_ID

WHERE a.ASSESSMENT_DEADLINE = (

    SELECT MAX(ASSESSMENT_DEADLINE)

    FROM ASSESSMENT

)

GROUP BY m.MODULE_TITLE;
```



```
SQL> SELECT m.MODULE_TITLE, MAX(a.ASSESSMENT_ID) AS LATEST_ASSESSMENT
  2  FROM MODULE m
  3  JOIN ASSESSMENT a ON m.MODULE_ID = a.MODULE_ID
  4  WHERE a.ASSESSMENT_DEADLINE = (
  5      SELECT MAX(ASSESSMENT_DEADLINE)
  6      FROM ASSESSMENT
  7  )
  8  GROUP BY m.MODULE_TITLE;

MODULE_TITLE                   LATEST_ASSESSMENT
------------------------------ -----------------
Database                                      18
```

*Figure 41 Displaying Module with latest deadline*

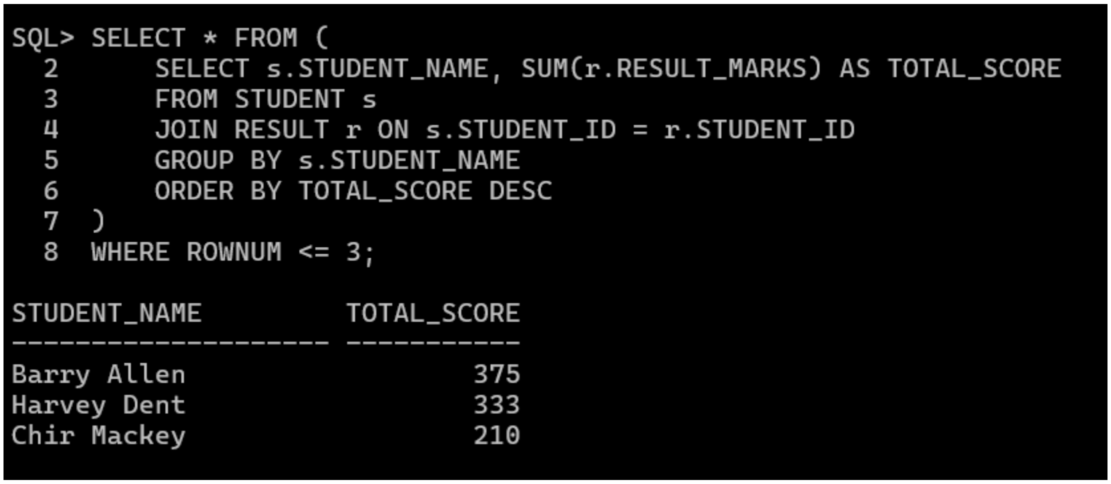
The query is for identification of the latest assessment deadline among modules. It does this by comparing the maximum deadline status date of all the assessments in the last. Next, it joins the ASSESSMENT and MODULE tables for extracting module titles and latest assessment details. It will identify the module needed to be focused on immediately.

### 6.2.2 Finding top three students with highest total scores in all modules

Find the top three students who have the highest total score across all modules.

**Query**

```
SELECT * FROM (
        SELECT s.STUDENT_NAME, SUM(r.RESULT_MARKS) AS TOTAL_SCORE
        FROM STUDENT s
        JOIN RESULT r ON s.STUDENT_ID = r.STUDENT_ID
        GROUP BY s.STUDENT_NAME
ORDER BY TOTAL_SCORE DESC
)
WHERE ROWNUM <= 3;
```



```
SQL> SELECT * FROM (
  2      SELECT s.STUDENT_NAME, SUM(r.RESULT_MARKS) AS TOTAL_SCORE
  3      FROM STUDENT s
  4      JOIN RESULT r ON s.STUDENT_ID = r.STUDENT_ID
  5      GROUP BY s.STUDENT_NAME
  6      ORDER BY TOTAL_SCORE DESC
  7  )
  8  WHERE ROWNUM <= 3;

STUDENT_NAME         TOTAL_SCORE
-------------------- -----------
Barry Allen                  375
Harvey Dent                  333
Chir Mackey                  210
```

*Figure 37 Displaying top three students with highest score in all modules*

The query ranks students on the aggregate score earned across different modules. With the top three students qualified by this query, it determines their final scores using `SUM`, sorts them in descending order, and narrows them into only three highest counts. The result has identified the students whose overall score is most outstanding.

### 6.2.3 Finding total number of assessments for each program and average score across all assessments for a particular module

Find the total number of assessments for each program and the average score across all assessments in those programs.

**Query**

SELECT p.PROGRAM_TITLE, COUNT(a.ASSESSMENT_ID) AS TOTAL_ASSESSMENTS, AVG(r.RESULT_MARKS) AS AVG_SCORE

FROM PROGRAM p

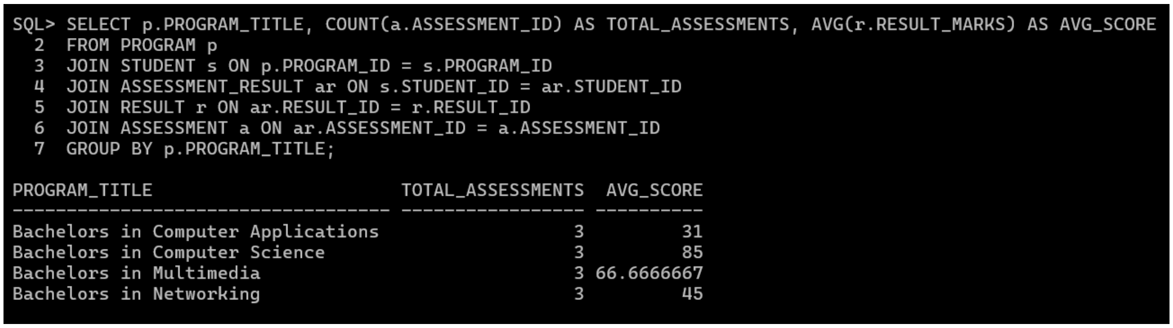
JOIN STUDENT s ON p.PROGRAM_ID = s.PROGRAM_ID

JOIN ASSESSMENT_RESULT ar ON s.STUDENT_ID = ar.STUDENT_ID

JOIN RESULT r ON ar.RESULT_ID = r.RESULT_ID

JOIN ASSESSMENT a ON ar.ASSESSMENT_ID = a.ASSESSMENT_ID

GROUP BY p.PROGRAM_TITLE;



```
SQL> SELECT p.PROGRAM_TITLE, COUNT(a.ASSESSMENT_ID) AS TOTAL_ASSESSMENTS, AVG(r.RESULT_MARKS) AS AVG_SCORE
  2  FROM PROGRAM p
  3  JOIN STUDENT s ON p.PROGRAM_ID = s.PROGRAM_ID
  4  JOIN ASSESSMENT_RESULT ar ON s.STUDENT_ID = ar.STUDENT_ID
  5  JOIN RESULT r ON ar.RESULT_ID = r.RESULT_ID
  6  JOIN ASSESSMENT a ON ar.ASSESSMENT_ID = a.ASSESSMENT_ID
  7  GROUP BY p.PROGRAM_TITLE;

PROGRAM_TITLE                        TOTAL_ASSESSMENTS  AVG_SCORE
------------------------------------ ----------------- ----------
Bachelors in Computer Applications                    3         31
Bachelors in Computer Science                        3         85
Bachelors in Multimedia                              3 66.6666667
Bachelors in Networking                              3         45
```

*Figure 38 Displaying total number of assessments in each program with average score across all assessment for a particular module*

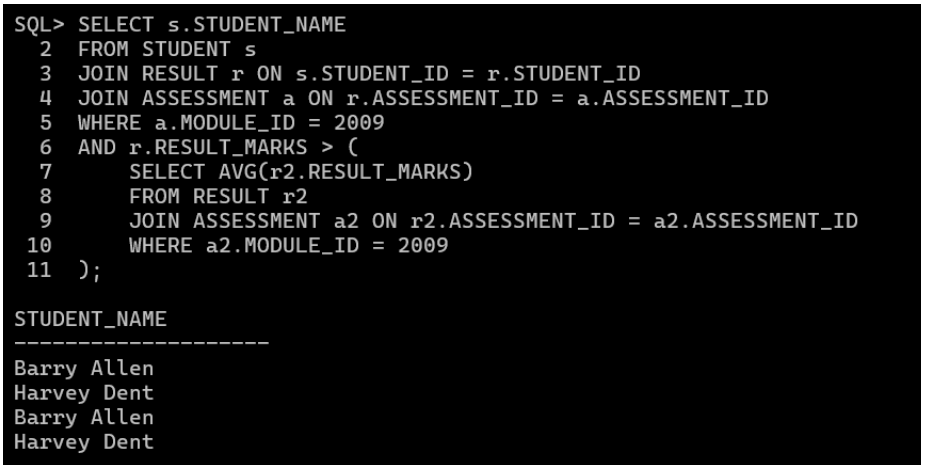
The query computes the total numbers of assessments for each program and the average scores derived for each program by students in the assessment. Grouping the data through program title gives this description where an overview can be obtained both from the count of assessments performed and the metrics of achievement. That would be a good way to assess the assessment activity and achievement of students as well.

### 6.2.4 Listing students with above average score in 'Database' Module

List the students who have scored above the average score in the 'Databases' module.

**Query**

```
SELECT s.STUDENT_NAME

FROM STUDENT s

JOIN RESULT r ON s.STUDENT_ID = r.STUDENT_ID

JOIN ASSESSMENT a ON r.ASSESSMENT_ID = a.ASSESSMENT_ID

WHERE a.MODULE_ID = 2009

AND r.RESULT_MARKS > (

        SELECT AVG(r2.RESULT_MARKS)

        FROM RESULT r2

        JOIN ASSESSMENT a2 ON r2.ASSESSMENT_ID = a2.ASSESSMENT_ID

        WHERE a2.MODULE_ID = 2009

);
```



```
SQL> SELECT s.STUDENT_NAME
     FROM STUDENT s
     JOIN RESULT r ON s.STUDENT_ID = r.STUDENT_ID
     JOIN ASSESSMENT a ON r.ASSESSMENT_ID = a.ASSESSMENT_ID
     WHERE a.MODULE_ID = 2009
     AND r.RESULT_MARKS > (
         SELECT AVG(r2.RESULT_MARKS)
         FROM RESULT r2
         JOIN ASSESSMENT a2 ON r2.ASSESSMENT_ID = a2.ASSESSMENT_ID
         WHERE a2.MODULE_ID = 2009
     );

STUDENT_NAME
--------------------
Barry Allen
Harvey Dent
Barry Allen
Harvey Dent
```

*Figure 39 Diplaying list of students who scored above average marksin 'Databases'  module*

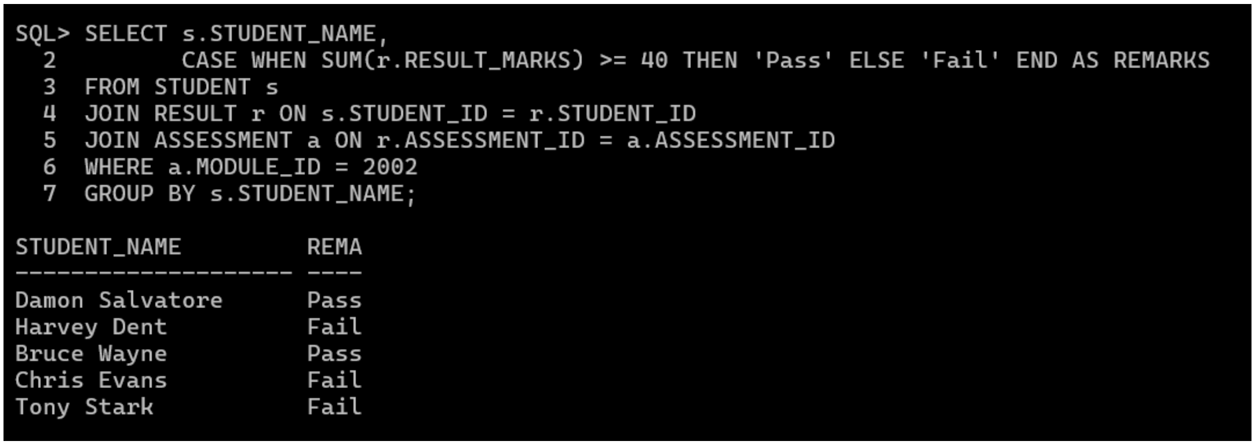
The query will give you the list of students who scored above average in the 'Databases' module. It obtains the average score for the module from a subquery and compares individual scores to this average. The result will show the high-flyers in that module.

### 6.2.5 Displaying whether student has passed or failed in a particular module

Display whether a student has passed or failed as remarks as per their total aggregate marks obtained in a particular module. (NOTE: Consider total aggregate marks equal to or above 40 is pass , below 40 is fail).

**Query**

```
SELECT s.STUDENT_NAME,
    CASE WHEN SUM(r.RESULT_MARKS) >= 40 THEN 'Pass' ELSE 'Fail' END AS REMARKS
FROM STUDENT s
JOIN RESULT r ON s.STUDENT_ID = r.STUDENT_ID
JOIN ASSESSMENT a ON r.ASSESSMENT_ID = a.ASSESSMENT_ID
WHERE a.MODULE_ID = 2002
GROUP BY s.STUDENT_NAME;
```



```
SQL> SELECT s.STUDENT_NAME,
            CASE WHEN SUM(r.RESULT_MARKS) >= 40 THEN 'Pass' ELSE 'Fail' END AS REMARKS
     FROM STUDENT s
     JOIN RESULT r ON s.STUDENT_ID = r.STUDENT_ID
     JOIN ASSESSMENT a ON r.ASSESSMENT_ID = a.ASSESSMENT_ID
     WHERE a.MODULE_ID = 2002
     GROUP BY s.STUDENT_NAME;

STUDENT_NAME         REMA
-------------------- ----
Damon Salvatore      Pass
Harvey Dent          Fail
Bruce Wayne          Pass
Chris Evans          Fail
Tony Stark           Fail
```

*Figure 40 Displaying whether student has passed or failed in a particular module*

The query checks whether or not students pass or fail a given module, on the basis of their total aggregate marks out of a pass mark of 40 points. It further uses a 'CASE' statement to show 'Pass'.

# 7.0 Critical Evaluation

## 7.1 Evaluation of module and its Relevance

The module related to database design implementation plays an important role in understanding the principles of creating a well-managed, and efficient database systems. The case study provided focused on designing a robust database for Ms. Mary's "E-Classroom Platform" which aims in providing a comprehensive digital learning environment for Stark College. Through this module, it was possible for us to learn and implement database concepts such as conceptual modeling, normalization, and queries in SQL to manage data related to students, teachers, programs and their interactions in effective manner.

This module also acted as a pivotal point in demonstrating theoretical knowledge in application to real-world scenarios. For instance, designing of database for E-Classroom involved defining of relationships such as one-to-many, many-to-one, many-to-many and one-to-one, connections between different entities and attributes of the database model. These concepts are directly applicable in maintaining and managing structure in education related data, ensuring seamless execution and efficient tracking of academic performance.

Moreover, the module is interconnected with various subjects and establish strong connections with it. Various subjects like business management, data structure and information technology related systems, etc, Proper understanding about database design concepts helped in building foundations fron technology integration for operational workflow, highlighting the importance in disciplines of data analytics, development of software and system administration. The interdisciplinary nature is relevant to point out the major beneficial impacts on developing skillsets necessary for learners about various roles of database management design concepts essential in tech and business related field and roles.

## 7.2 Assessment of the Coursework

The coursework have presented a case study which was reviewed and step-by-step approach was followed in creating a database for Ms. Mary's E-Classroom Platform. Initiation of database design was through identification of business requirements, which proceeded to conceptual data modeling and normalization processes. This ensured clear and efficient database structure. Database design was implemented accordingly to manage key attributes of entities like student, program, module, teacher, announcement, assessment and result while maintaining the relationship and integrity within those entities.

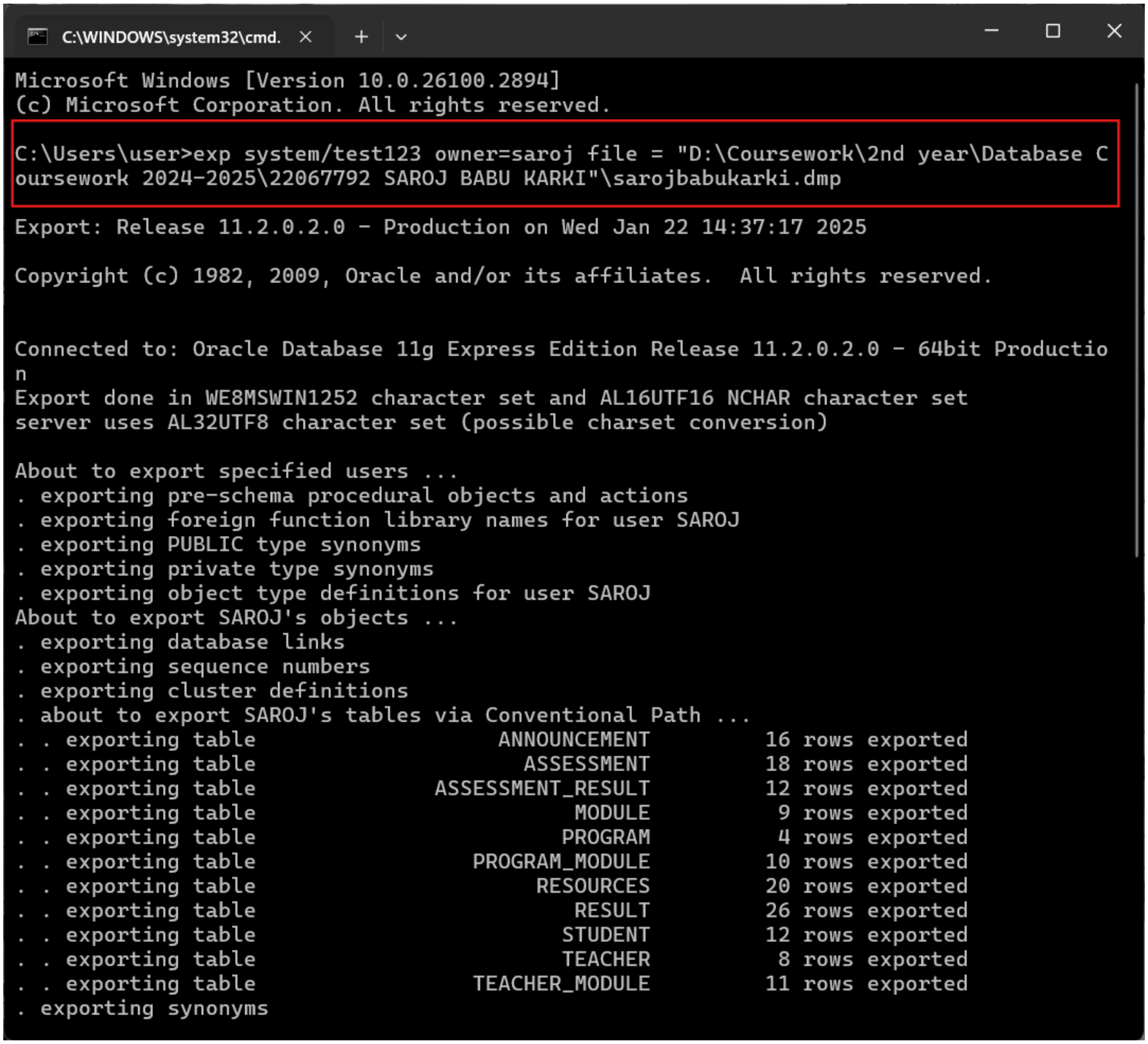
Integration of SQL queries was much effective in demonstration of practical database functionality in real-world. Queries were used to retrieve various data related to student performance, management of module related resources, and generating results for assessments for providing insights to the students. This ensured in supporting daily decision-making activities and manage daily operation in an educational environment.

A notable aspect of the coursework was that the structured approach towards managing module related resources. The resources were ensured if it was completed or not in a predefined sequence to emphasize progressive learning environment for each students. Additionally, the ability to link assessments and announcements to specific modules enhanced platform's usability for both students and teachers.

Overall the coursework successfully met its objectives by applying necessary database concepts and design principles to the real-world scenario with the help of case study included in database design process. It demonstrated the process of developing a dynamic and well structured database system to support operational needs of an E-Classroom Platform. The experience was very valuable and prepared for managing complex systems across various industries.

# 8.0 Dump file creation and Drop Queries

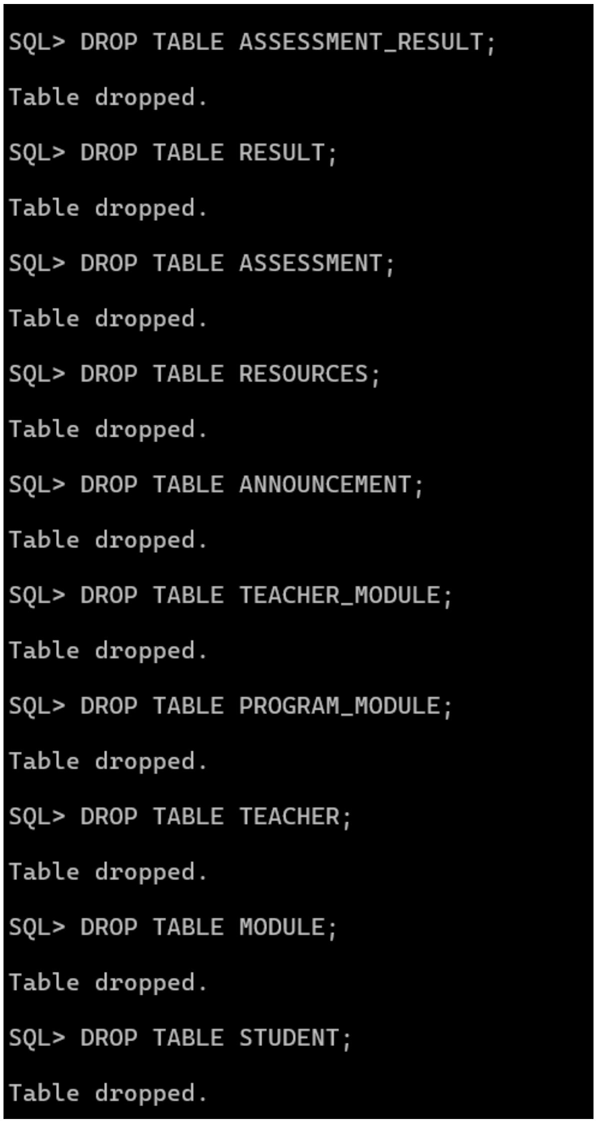
## 8.1 Dump file creation



```
Microsoft Windows [Version 10.0.26100.2894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>exp system/test123 owner=saroj file = "D:\Coursework\2nd year\Database C
oursework 2024-2025\22067792 SAROJ BABU KARKI"\sarojbabukarki.dmp

Export: Release 11.2.0.2.0 - Production on Wed Jan 22 14:37:17 2025

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 Productio
n
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 SAROJ
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user SAROJ
About to export SAROJ's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export SAROJ's tables via Conventional Path ...
. . exporting table                   ANNOUNCEMENT         16 rows exported
. . exporting table                     ASSESSMENT         18 rows exported
. . exporting table              ASSESSMENT_RESULT         12 rows exported
. . exporting table                         MODULE          9 rows exported
. . exporting table                        PROGRAM          4 rows exported
. . exporting table                 PROGRAM_MODULE         10 rows exported
. . exporting table                      RESOURCES         20 rows exported
. . exporting table                         RESULT         26 rows exported
. . exporting table                        STUDENT         12 rows exported
. . exporting table                        TEACHER          8 rows exported
. . exporting table                 TEACHER_MODULE         11 rows exported
. exporting synonyms
```

*Figure 41 Dump File Cration*

## 8.2 Drop Queries



```
SQL> DROP TABLE ASSESSMENT_RESULT;

Table dropped.

SQL> DROP TABLE RESULT;

Table dropped.

SQL> DROP TABLE ASSESSMENT;

Table dropped.

SQL> DROP TABLE RESOURCES;

Table dropped.

SQL> DROP TABLE ANNOUNCEMENT;

Table dropped.

SQL> DROP TABLE TEACHER_MODULE;

Table dropped.

SQL> DROP TABLE PROGRAM_MODULE;

Table dropped.

SQL> DROP TABLE TEACHER;

Table dropped.

SQL> DROP TABLE MODULE;

Table dropped.

SQL> DROP TABLE STUDENT;

Table dropped.
```

*Figure 42 List of Drop Queries*

To drop a table involved in a foreign key, the child table should be dropped first. This is because the foreign key in the child table depends on the primary key of the parent table to maintain its relationship. Dropping the child table along with the foreign key constraint breaks this relationship, putting the parent table in a situation in such a way that it could be dropped without violating referential integrity. Therefore, dropping the child table is important in this case.

## 9.0 References

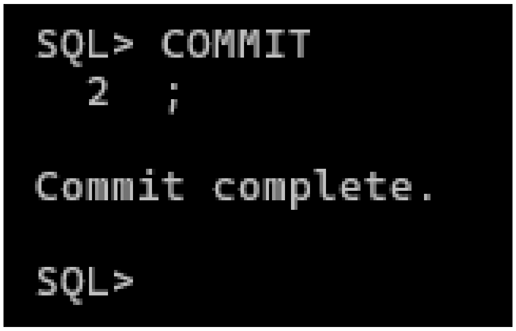
Chen, Q. L.-L. (2009). Entity-Relationship Diagram. *Modeling and Analysis of Enterprise and Information Systems* , 125-139.

Ram, S., & Khatri, V. (2005). A comprehensive framework for modeling set-based business rules during conceptual database design. *Information Systems*, 89-118. Retrieved from https://doi.org/10.1016/j.is.2003.10.008

V. P. (30-06-2020). “Database Normalization: A Review. *International Journal for Research Publication and Seminar*, 4–16. Retrieved 12 30, 2024, from https://jrps.shodhsagar.com/index.php/j/article/view/1104

*Visual Paradigm*. (2024). Retrieved from What is Entity Relationship Diagram (ERD)?: https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/#:~:text=Entity%20Attributes&text=An%20attribute%20has%20a%20name,supported%20by%20the%20target%20RDBMS.

# 10.0 Appendix



```
SQL> COMMIT
  2  ;

Commit complete.

SQL>
```

*Figure 43 Commit Command*