Module Code & Module Title

CC5051 – Database

Assessment Type

100% Individual Coursework

Semester

2024/25 Autumn

Student Name: Anuj Sapkota

London Met ID: 23049194

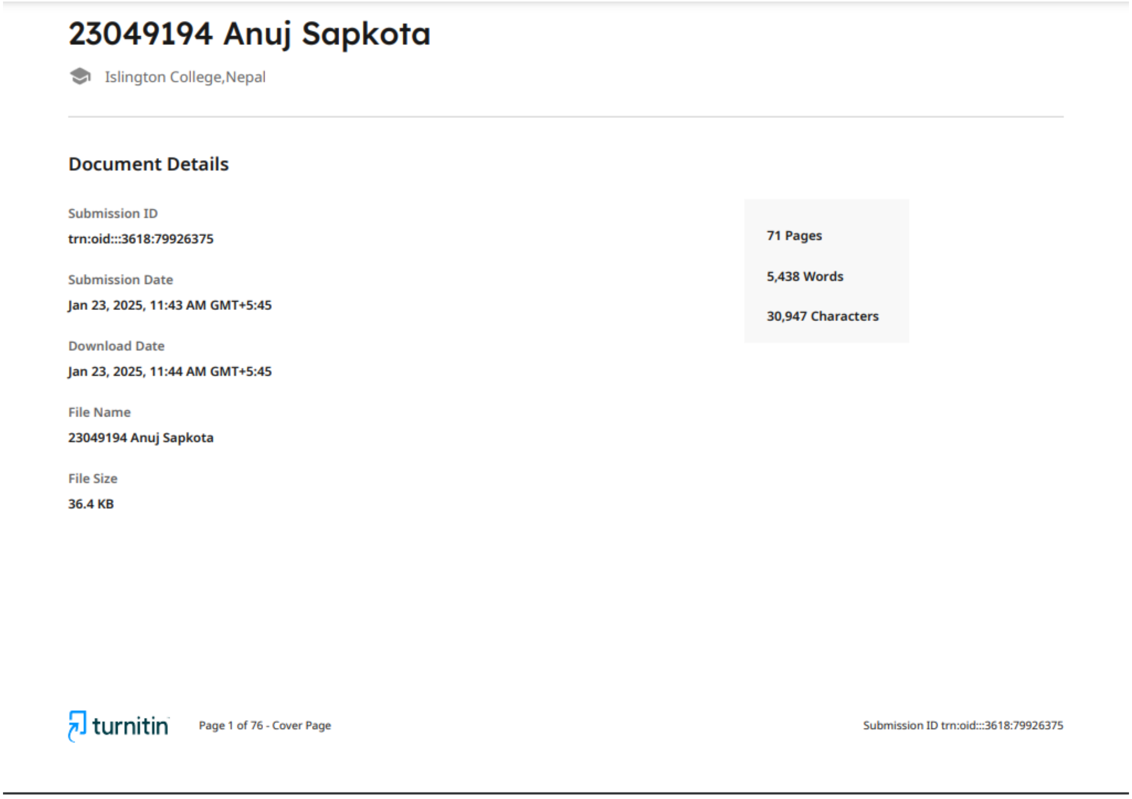
College ID: NP05CP4A230175

Assignment Due Date: Thursday, January 23, 2025

Assignment Submission Date: Thursday, January 23, 2025

Word Count: 8336 Words

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



23049194 Anuj Sapkota

Islington College,Nepal

Document Details

Submission ID
trn:oid:::3618:79926375

Submission Date
Jan 23, 2025, 11:43 AM GMT+5:45

Download Date
Jan 23, 2025, 11:44 AM GMT+5:45

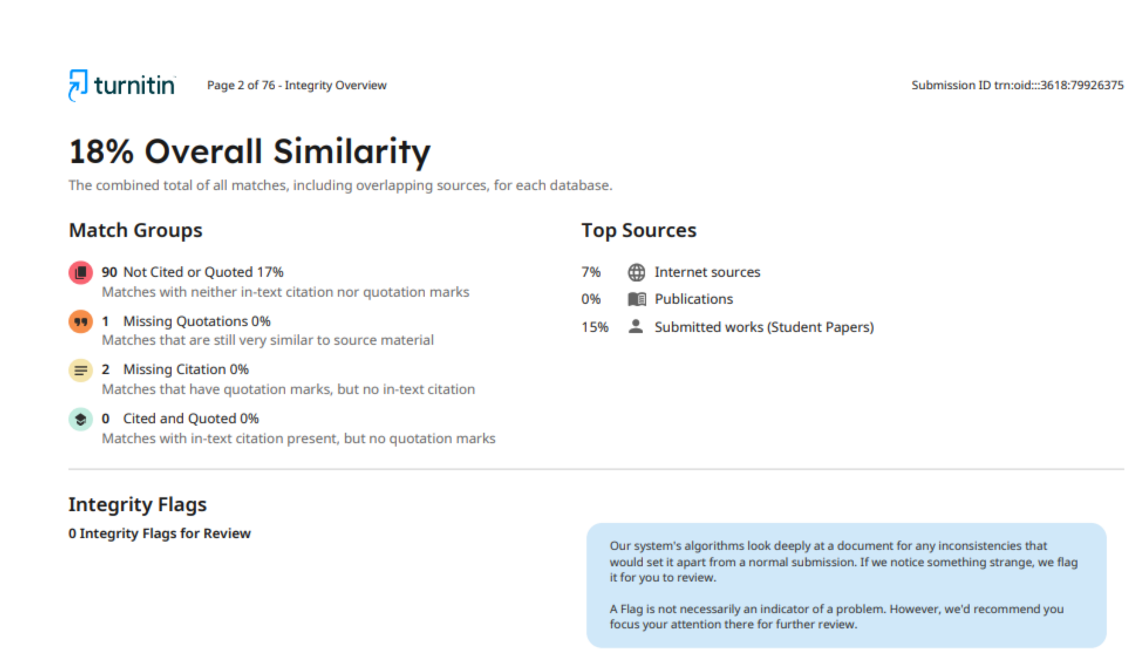
File Name
23049194 Anuj Sapkota

File Size
36.4 KB

71 Pages

5,438 Words

30,947 Characters



18% Overall Similarity

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

Match Groups

- **90** Not Cited or Quoted 17%
  Matches with neither in-text citation nor quotation marks
- **1** Missing Quotations 0%
  Matches that are still very similar to source material
- **2** 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

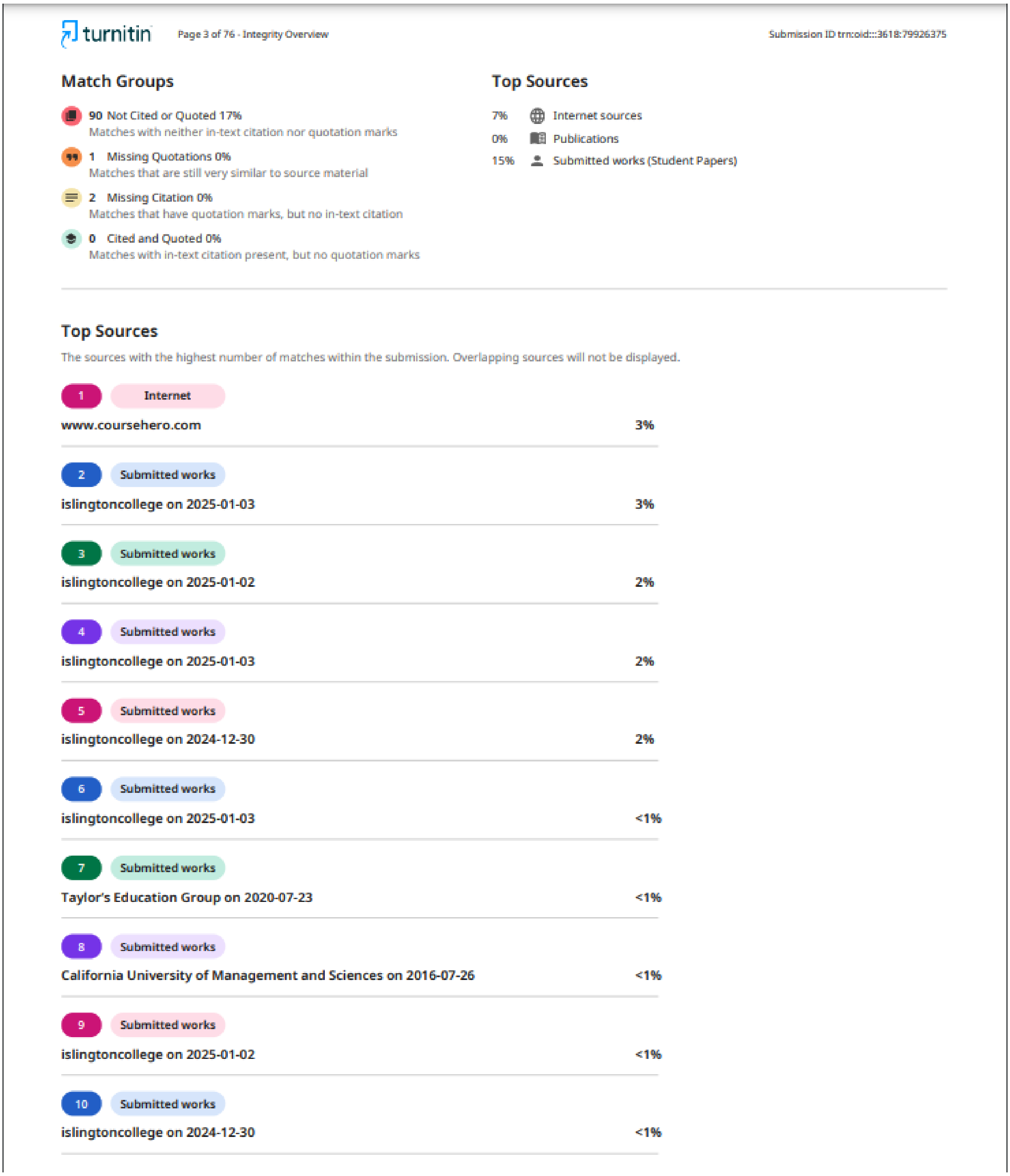
7%	Internet sources
0%	Publications
15%	Submitted works (Student Papers)

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.



Match Groups

- **90** Not Cited or Quoted 17%
  Matches with neither in-text citation nor quotation marks
- **1** Missing Quotations 0%
  Matches that are still very similar to source material
- **2** 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

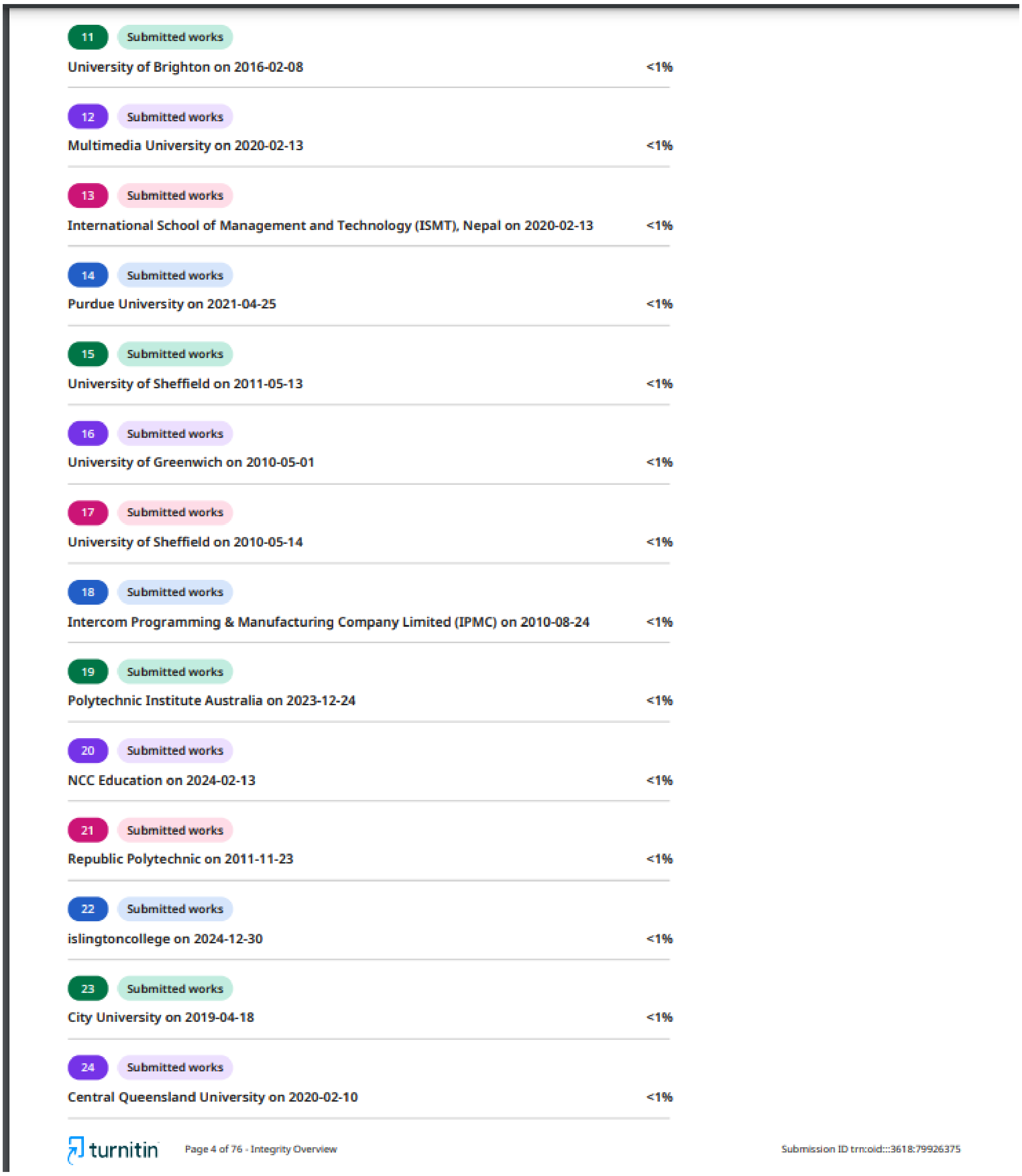
Top Sources

7%	Internet sources
0%	Publications
15%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	www.coursehero.com	3%
2	Submitted works	islingtoncollege on 2025-01-03	3%
3	Submitted works	islingtoncollege on 2025-01-02	2%
4	Submitted works	islingtoncollege on 2025-01-03	2%
5	Submitted works	islingtoncollege on 2024-12-30	2%
6	Submitted works	islingtoncollege on 2025-01-03	<1%
7	Submitted works	Taylor's Education Group on 2020-07-23	<1%
8	Submitted works	California University of Management and Sciences on 2016-07-26	<1%
9	Submitted works	islingtoncollege on 2025-01-02	<1%
10	Submitted works	islingtoncollege on 2024-12-30	<1%



11 Submitted works

University of Brighton on 2016-02-08 <1%

12 Submitted works

Multimedia University on 2020-02-13 <1%

13 Submitted works

International School of Management and Technology (ISMT), Nepal on 2020-02-13 <1%

14 Submitted works

Purdue University on 2021-04-25 <1%

15 Submitted works

University of Sheffield on 2011-05-13 <1%

16 Submitted works

University of Greenwich on 2010-05-01 <1%

17 Submitted works

University of Sheffield on 2010-05-14 <1%

18 Submitted works

Intercom Programming & Manufacturing Company Limited (IPMC) on 2010-08-24 <1%

19 Submitted works

Polytechnic Institute Australia on 2023-12-24 <1%

20 Submitted works

NCC Education on 2024-02-13 <1%

21 Submitted works

Republic Polytechnic on 2011-11-23 <1%

22 Submitted works

islingtoncollege on 2024-12-30 <1%

23 Submitted works

City University on 2019-04-18 <1%

24 Submitted works

Central Queensland University on 2020-02-10 <1%

25 Submitted works

The Robert Gordon University on 2024-12-16 <1%

Table of Contents

Table of Figures

Table of Tables

1. Introduction

1.1. Introduction of Business and its Forte and Current Business Activities and Operations

1.1.1. Business and its Forte

Established in 1979 AD, Shree Panchayat College is a well-renounced college situated in Pathari Municipality, Morang. This college has become one of the best colleges people prefer for gaining a bachelor's degree. It provides students with practical experience and a well-suited learning environment.

Missions and Values

The mission of Shree Panchayat college is to educate the learners and help them reach their full potential. This college is committed to transforming the learning experience of students and helping them achieve the growth mindset that is essential for reaching their desired dreams.

Figure 1: Shree Panchayat College

Some of the values of this college are:

- **Excellency:** Provide students with great learning environments for exponential growth.
- **Equity:** Welcoming diverse groups of students where every individual feels valued and respected.
- **Partnership:** Partnership with various organizations for carrying out various programs and activities.
- **Student Clubs:** Encourage students to build various clubs and organizations focusing on humanitarian work.
- **Health and Activities:** Conduct various activities that empower health and fitness among students.

As the Founder of this highly prestigious college, Ms. Mary is thrived to launch **SikshyaSudhar**, an school management app for **Shree Panchayat Campus** which will efficiently manage the overall college resources. It aims at solving problems faced by traditional classrooms like complexity in management of teachers, students, modules, programs and other resources. The **E-classroom platform** is aimed at transforming traditional classroom experience to make it simpler to manage it.

Currently, with an increasing number of technological advancements, the world demands a scalable, efficient, simple and user-friendly and highly available platforms for almost everything. The classroom system is not an exception. SikshyaSudhar is designed to be able to handle all the students, staff, programs, modules and other various college resources and still be able to provide users with great user experience and simplicity. The database system will store the overall data in a proper as well as in a structured form making the system storage more flexible and consistent.

The digital study environment will keep track of all ongoing activities ,College Resources and members. SikshyaSudhar will help both the teachers and the students to engage in course activities digitally and will be able to enroll the students in various programs and the teachers will also be able to provide assignments or assessments for the students. At the same time, it features students viewing their results.

Therefore, the platform acts as a great tool for modern technologically enhanced colleges and addresses the issues faced by traditional institutions like Shree Panchayat Higher Secondary School and ensures that the education system becomes flexible, simpler and structured.

Aims and Objectives

- The Aims of the college are:
    - To launch SikshyaSudhar that transforms the traditional way of classroom management into a digitalized form.
    - To make the platform highly scalable and efficient, which improves the experience of overall students and staff.
    - To guarantee that the education system is scalable and simple to manage.

- The Objectives of the College in launching SikshyaSudhar are:
  - To effectively manage teachers, students, Programs, modules and various college resources.
  - To digitalize the traditional education management system and make the system simpler.
  - To track student resource usage status for orderly learning.

1.1.2. Current Business Activities and Operations

The college engages in various activities which are aimed at providing better education and helping students with their professional as well as their personal development. This college provides physical as well as recorded lectures through cloud storage. This helps students by making the lectures accessible at any time, developing a flexible learning environment. Students are also encouraged to take part in various on-going sports and activities which in turn help strengthen overall physical health and reduce mental stress. Various Scholarship programs are also conducted as a vital support system for students to help them pursue their goals and achieve their dreams without any financial obstacles coming in their way.

SikshyaSudhar revolves around providing different activities that are necessary to build a proper studying platform. It handles several operations including managing teachers, students, modules, assessments and college resources. The college provides various programs including bachelor's in computing, Bachelors in Networking, Bachelors in Multimedia and so on. These various programs help students to gain essential knowledge and skills required in various IT sectors.

The platform enables various operations to be done like:

- Shree Panchayat Campus provides multiple programs like BSc in Computing, BSc in Networking, BSc in Multimedia. Students take one program and the programs that the students take consist of various modules which are compulsory for students to take to finish the program course.

- Subjects like bachelor’s in computing and Bachelors in Multimedia both share the same programming module which makes the curriculum more flexible.

- Teachers teach modules individually and give students one or many assessments for each module for which results are also calculated which will reflect the students’ work.

- Each module consists of several resources ordered in a predefined sequence which must be completed in an orderly fashion to get access to its next resource which helps students follow a standardized learning process.

- Teachers are able to post announcements for specific modules they teach to notify students about any messages.

1.2. Business Rules

Business rules are the set of rules and regulations that define the operations in an organization. They help to build a proper database system by deriving logic for every important decision made during the operations.

The business rules that will be used in the systems are given below:

- ❖ **Student**
  - Each Student is admitted in exactly one of the programs.
- ❖ **Program**
  - Each program contains various students.
  - Each program contains various modules.
- ❖ **Module**
  - A Module can be included in more than one programs.
  - A module can either have single or more assessments.
  - A module contains various resources.
  - Multiple modules can be taught by one teacher.
- ❖ **Teacher**
  - One teacher can teach various modules.
- ❖ **Announcement**
  - Each Module can have more than one announcement.
- ❖ **Assessment**
  - Each assessment is assigned to exactly one module.
  - Each assessment has either one or multiple results
- ❖ **Result**
  - Multiple Results can be related to one Assessment.
- ❖ **Resources**
  - Each module consists of multiple resources.
  - Resources must be completed in sequential order.

2. Initial ERD

2.1. Identification of Entities and Attributes

Entities are real or abstract objects having their own set of attributes /properties and relationships with other entities. Attributes are the properties that entities have. It is like a feature or character of an entity. From the given scenario, several entities along with their respective attributes can be found. They are given below:

Based on the given scenario, 8 entities were formed in total which are given below:

Entity	Description
Program	It carries data of different programs taught in the institution.
Modules	It carries data of all the Modules available in the institution.
Students	It carries data of all the students enrolled in the institution.
Resources	It carries data of all the resources available in specific modules.
Teacher	It carries data of all the teachers in the institution.
Announcement	It carries data of all the announcements made by the modules.
Assessment	It carries data of all the assessments for the modules
Result	It carries data of all the results students got in their assessments.

Table 1: Entities and their description

The following are the list of each entity with a description of their respective attributes:

- **Program**

S. N.	Attribute Name	Data Type	Size	Constraint
1	Program_ID	Number	38	Primary Key
2	Program_Name	Character	50	-
3	Program_Duration_Years	Number	Precision 2, Scale 1	Not Null
4	Program_Fees	Decimal	Precision 2, Scale 1	Not Null
5	Total_Credits	Number	38	Not Null

Table 2: Description of Program entity.

- **Modules**

S. N.	Attribute Name	Data Type	Size	Constraint
1	Module_ID	Number	38	Primary Key
2	Module_Name	Character	50	-
3	Duration_Months	Number	38	-
4	Credits	Number	38	-

Table 3: Description of Module entity.

- **Student**

S. N.	Attribute Name	Data Type	Size	Constraint
1	Student_ID	Number	38	Primary Key
2	First_Name	Character	50	Not Null
3	Last_Name	Character	20	Not Null
4	Email	Character	20	Unique
5	Phone	Character	15	-
6	Address	Character	25	-
7	Enroll_Date	Date		Not Null

Table 4: Description of Student entity.

- **Resources**

S. N.	Attribute Name	Data Type	Size	Constraint
1	Resource_ID	Number	38	Primary Key
2	Resource_Name	Character	50	-
3	Format	Character	50	-
4	Duration_Weeks	Number	38	-

Table 5: Description of Resources entity.

- **Teacher**

S. N.	Attribute Name	Data Type	Size	Constraint
1	Teacher_ID	Number	38	Primary Key
2	First_Name	Character	20	Not Null
3	Last_Name	Character	20	Not Null
4	Specialization	Character	50	Not Null
5	Contact_Number	Character	15	-
6	Address	Character	25	-
7	Join_Date	Date	-	Not Null

Table 6: Description of Teacher entity.

- **Announcement**

S. No.	Attribute Name	Data Type	Size	Constraint
1	Announcement_ID	Number	38	Primary Key
2	Announcement_Title	Character	100	Not Null
3	Body	Character	250	-
4	Date_Posted	Date	-	Not Null
5	Expiry_Date	Date	-	Not Null

Table 7: Description of Announcement entity.

- **Assessment**

S. N.	Attribute Name	Data Type	Size	Constraint
1	Assessment_ID	Number	38	Primary Key
2	Title	Character	50	Not Null
3	Deadline	Date	-	Not Null
4	Weightage_Percentage	Number	Precision 5, Scale 2	Not Null
5	Full_Marks	Number	38	Not Null

Table 8: Description of Assessment entity.

- **Result**

S. N.	Attribute Name	Data Type	Size	Constraint
1	Result_ID	Number	38	Primary Key
2	Marks_Obtained	Number	38	-
3	Grade	Character	1	-
4	Percentage	Number	Precision 5, Scale 2	-

Table 9: Description of Result entity.

Primary keys of Entities

There are about 8 entities in total for this scenario, each having their respective primary key. The program entity has Primary key Program_ID, Student has Student_ID, Teacher has Teacher_ID. Similarly, Module has Module_ID and Assessment_ID, Result has Result_ID, Resources has Resource_ID and Announcement has Announcement_ID.

2.2. ERD of the identified entities with attributes and relations.

2.2.1. Relations between entities

i. One student is enrolled in one program and many students are also enrolled in one program.

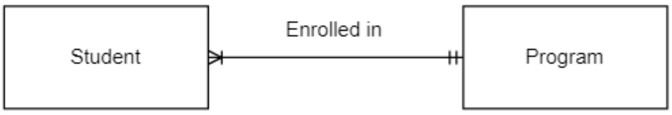


Figure 2: Relation of Student and Program.

ii. One program consists of one module at a minimum, but many programs can also consist of the same module and there are many modules in one program too.

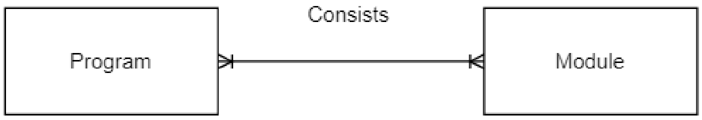


Figure 3: Relation of Program and Module.

iii. One Module has one resource at minimum and many resources at maximum.

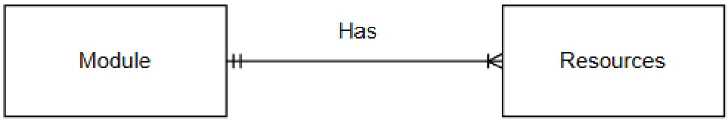


Figure 4: Relation of module and resources.

iv. One module has one or more than one assessment assigned.

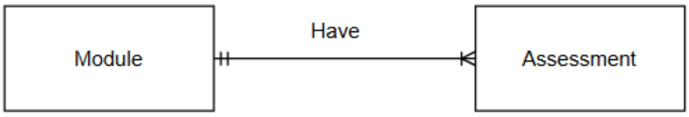


Figure 5: Relation of module and assessment.

v. One assessment has either one or many results.

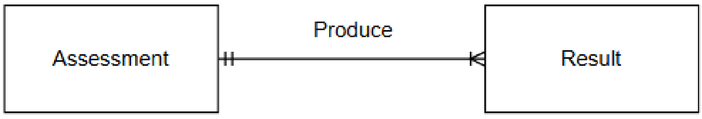


Figure 6: Relation of assessment and result.

vi. Many modules can be taught by one teacher.

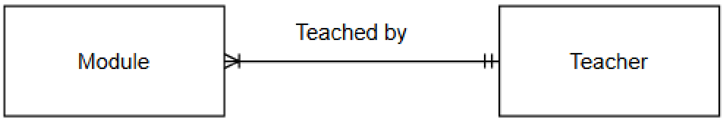


Figure 7: Relation of module and teacher.

vii. One teacher can announce one announcement at minimum but can also announce many announcements.

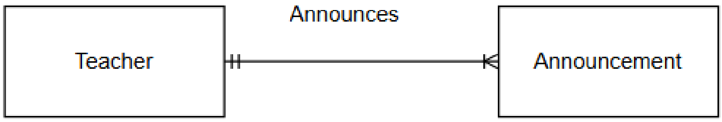


Figure 8: Relation of teacher and announcement.

2.2.2. Initial Entity-Relationship Diagram.

ERD is the abbreviation of Entity-Relationship Diagram which is a graphical presentation of Entity, attributes including the relationships between them. This diagram is very crucial for the development of the database because it defines the format of the database. ERDs are often normalized to remove any inconsistencies and redundancies and anomalies in the design. (Hanna, 2025)

Below is the Initial ERD before it is normalized:

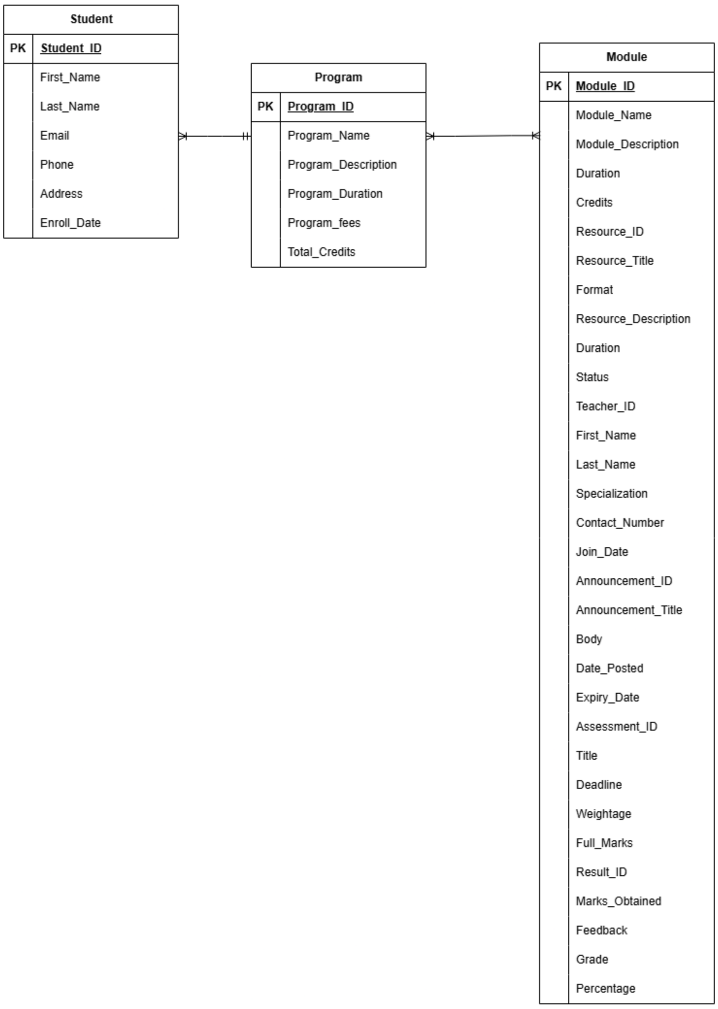


Figure 9: Initial ERD

3. Normalization

Normalization is the system of managing the data of tables and reducing data duplication and increasing data integrity and consistency. It helps reduce redundancy and properly manage the database and make it simpler. It also helps to remove any anomalies which are errors occurring while performing operations in database. After successful normalization the table becomes much simpler and smaller. The relationship between the tables is also handled well. (Ian, 2017)

3.1. UNF (Unnormalized Form)

UNF, also known as 0NF, is the unstructured form of normalization which does not meet any normalization rules. Here, the data is anatomized and there may be repeating groups and multivalued attributes. The data is initial in a loosely structured format onto which normalization is applied. It is a starting point for normalization. UNF can be represented in both tabular and nested format. In nested format, there is a root entity on which all other entities are tied to. The attributes of the tables are either separated by curly braces or not separated by anything. If two entities are in a relation and let's suppose the second entity has a many then they are separated by curly braces and if the second entity has a one cardinality, then they are not separated by curly braces. (Wikipedia, 2024)

Representing data into unnormalized form

Student (Student_ID, First_Name, Last_Name, Email, Phone, Address, Enroll_Date, Program_ID, Program_Name, Program_Duration_Years, Program_Fees, Total_Credits { Module_ID, Module_Name, Duration_Months, Credits , Teacher_ID, First_Name, Last_Name, Specialization, Contact_Number, Join_Date, { Announcement_ID, Announcement_Title, Body, Date_Posted, Expiry_Date }, { Resource_ID, Resource_Name, Format, Duration, Status}, { Assessment_ID, Title, Deadline, Weightage_Percentage, Full_Marks, { Result_ID, Marks_Obtained, Feedback, Percentage, Grade} })

3.2. 1NF (First Normalized Form)

In 1NF, repeating groups are removed and written separately. Here, the attributes have values that cannot be further simplified i.e. they have atomic values. All the multivalued attributes are split. So, 1NF of normalization follows atomicity. Furthermore, here a primary key is assigned to an attribute of each of the separated repeating groups. This removes the redundancy and further simplifies the data. 1NF acts as a great foundation for the upcoming higher forms of normalization. (Geeks For Geeks, 2025) (Thakur, 2016)

Here, the previous UNF is broken down into 6 separate groups each containing a primary key attribute

Representing the data into 1NF

Student_Module-1: [Module_ID, Module_Name, Duration_Months, Credits, Teacher_ID, First_Name, Last_Name, Specialization, Contact_Number, Join_Date, Student_ID*]

Module_Resource-1: [Resource_ID, Resource_Name, Format, Duration_Weeks, Status, Student_ID*, Module_ID*]

Module_Announcement-1: [Announcement_ID, Announcement_Title, Body, Date_Posted, Expiry_Date, Module_ID*]

Module_Assessment-1: [Assessment_ID, Title, Deadline, Weightage_Percentage, Full_Marks Module_ID*, Student_ID*]

Assessment_Result-1: [Result_ID, Marks_Obtained, Feedback, Grade, Percentage, Student_ID*, Module_ID*, Assessment_ID*]

3.3. 2NF (Second Normal Form)

In 2NF, the database is organized in such a way that all the non-key attributes of each entity are fully functional dependent on the primary key of the entity. For the database to be in 2NF, they must first be in 1NF. There partial dependency should not exist. In this way, data duplication is minimized as well as data consistency is maintained. Here in 2NF, if we face any partial dependencies, we remove them from the relation and place them in a new relation with the part of the composite key. The part of the composite key exists both on the original and the new relation. This redundancy here is necessary to ensure referential integrity. (GeeksForGeeks, 2025)

Checking partial dependency in Student_Program

- **Student_Program-1**: [Student_ID, First_Name, Last_Name, Email, Phone, Address, Enroll_Date, Program_ID, Program_Name, Program_Duration_Years, Program_Fees, Total_Credits]

    - **Student_ID** → First_Name, Last_Name, Email, Phone, Address, Enroll_Date, Program_ID, Program_Name, Program_Duration_Years, Program_Fees, Total_Credits

  Here, fully dependent exists between all the attributes and the key attribute Student_ID, the group structure remains the same.

Checking partial Dependency in Student_Module

- **Student_Module-1:** [Module_ID, Module_Name, Duration_Months, Credits, Teacher_ID, First_Name, Last_Name, Specialization, Contact_Number, Join_Date, Student_ID*]

    - **Module_ID** → Module_Name, Duration_Months, Credits, Teacher_ID, First_Name, Last_Name, Specialization, Contact_Number, Join_Date
    - **Module_ID, Student_ID*** → No attributes

  Here, as all the attributes except for Module_ID and Student_ID which are Primary key and composite key respectively are partially dependent upon Module_ID. So, all the partially depending attributes are kept on another table along with the key attribute Module_ID.

Checking partial dependency in Module_Resource

- **Module_Resource-1:** [Resource_ID, Resource_Name, Format, Duration_Weeks, Status, Student_ID*, Module_ID*]

    - **Resource_ID** → Resource_Name, Format, Duration_Weeks
    - **Resource_ID, Student_ID*, Module_ID*** → Status

Here, all the attributes except for Student_ID, Module_ID (Composite keys) and Status are partially dependent on Resource_ID. So, all the partially depending attributes are kept on another table along with the key attribute Resource_ID.

Checking partial dependency in Module_Announcement

- **Module_Announcement-1:** [Announcement_ID, Announcement_Title, Body, Date_Posted, Expiry_Date, Module_ID*]

    - **Announcement_ID** → Announcement_Title, Body, Date_Posted, Expiry_Date
    - **Announcement_ID, Module_ID*** → No attributes

Here, all the attributes except for the Module_ID (Composite key) are partially dependent on Announcement_ID. So, all the partially depending attributes are kept on another table along with the key attribute Announcement_ID.

Checking partial dependency in Module_Assessment

- **Module_Assessment-1:** [Assessment_ID, Title, Deadline, Weightage_Percentage, Full_Marks, Module_ID*, Student_ID*]
    - **Assessment_ID** → Title, Deadline, Weightage_Percentage, Full_Marks
    - **Assessment_ID, Module_ID*, Student_ID*** → No attributes

Here, all the attributes except for the Module_ID, Student_ID (Composite keys) are partially dependent on Assessment_ID. So, all the partially depending attributes are kept on another table along with the key attribute Assessment_ID.

Checking partial dependency in Assessment_Result

- **Assessment_Result-1:** [Result_ID, Marks_Obtained, Feedback, Grade, Percentage, Student_ID*, Module_ID*, Assessment_ID*]
    - **Result_ID, Student_ID*, Module_ID*, Assessment_ID*** → Result_ID, Marks_Obtained, Feedback, Grade, Percentage,

Here, as every attribute is fully dependent on the composite key. The table structure remains the same.

Representing the data into 2NF

Student-Program-2: [Student_ID, First_Name, Last_Name, Email, Phone, Address, Enroll_Date, Program_ID, Program_Name, Program_Duration_Years, Program_Fees, Total_Credits]

Student_Module-2: [Student_ID*, Module_ID*]

Module-2: [Module_ID, Module_Name, Duration_Months, Credits, Teacher_ID, First_Name, Last_Name, Specialization, Contact_Number, Join_Date]

Module_Resource-2: [Student_ID*, Module_ID*, Resource_ID*, Status]

Resource-2: [Resource_ID, Format, Resource_Name, Duration_Weeks]

Module_Announcement-2: [Module_ID*, Announcement_ID*, Student_ID*]

Announcement-2: [Announcement_ID, Announcement_Title, Body, Date_Posted, Expiry_Date]

Module_Assessment-2: [Module_ID*, Student_ID*,Assessment_ID*]

Assessment-2: [Assessment_ID, Title, Deadline, Weightage_Percentage, Full_Marks]

Assessment_Result-2: [Student_ID*, Module_ID*, Assessment_ID*, Result_ID*, Marks_Obtained, Feedback, Percentage, Grade]

3.4. 3NF (Third Normal Form)

To be in 3NF, the database is organized in such a way that transitive dependency for non-key attributes does not exist. Transitive Dependency exists when any three attributes are A, B and C, then if A depends on C through B, then A is said to be transitively dependent on C. This helps improve data integrity and creates an organized database. (GeeksForGeeks, 2025)

Checking transitive dependency on Student Program

- **Student-Program-2**: [Student_ID, First_Name, Last_Name, Email, Phone, Address, Enroll_Date, Program_ID, Program_Name, Program_Duration_Years, Program_Fees, Total_Credits]
  - Student_ID → Program_ID → (Program_Name, Program_Duration_Years, Program_Fees, Total_Credits)

Here, Student_ID is transitively dependent on Program_Name, Program_Duration_Years, Program_Fees and Total_Credits through Program_ID. So, they break with Student_ID and its attributes such as: First_Name, Last_Name, Email, Phone, Address, Enroll_Date with Program_ID as foreign key on one table and Program_ID, Program_Name, Program_Fees Program_Duration_Years, and Total_Credits on another table onto which Program_ID becomes the Primary key.

Checking transitive dependency on Module

- **Module-2**: [Module_ID, Module_Name, Duration_Months, Credits, Teacher_ID, First_Name, Last_Name, Specialization, Contact_Number, Join_Date]
  - Module_ID → Teacher_ID → (First_Name, Last_Name, Specialization, Contact_Number, Join_Date)

Here, Module_ID is transitively dependent on First_Name, Last_Name, Specialization, Contact_Number and Join_Date through Teacher_ID. So, they break with Module_ID and its attributes such as Module_Name, Duration_Months, Credits with Teacher_ID as foreign key on one table and Teacher_ID, First_Name, Last_Name, Specialization, Contact_Number and Join_Date on another table where Teacher_ID is the Primary key.

Representing the data into 3NF

Student-3: [Student_ID, First_Name, Last_Name, Email, Phone, Address, Enroll_Date, Program_ID*]

Program-3: [Program_ID, Program_Name, Program_Duration_Years, Program_Fees, Total_Credits]

Student-Module-3: [Student_ID*, Module_ID*]

Module-3: [Module_ID, Module_Name, Duration_Months, Credits, Teacher_ID*]

Module-Resource-3: [Student_ID*, Module_ID*, Resource_ID*, Status]

Resource-3: [Resource_ID, Format, Resource_Name, Duration_Weeks]

Teacher-3: [Teacher_ID, First_Name, Last_Name, Specialization, Contact_Number, Join_Date]

Module-Announcement-3: [Student_ID*, Module_ID*, Announcement_ID*]

Announcement-3: [Announcement_ID, Announcement_Title, Body, Date_Posted, Expiry_Date]

Module-Assessment-3: [Module_ID*, Student_ID*, Assessment_ID*]

Assessment-3: [Assessment_ID, Title, Deadline, Weightage_Percentage, Full_Marks]

Assessment- Result-3: [Student_ID*, Module_ID*, Assessment_ID*, Result_ID*, Marks_Obtained, Feedback, Percentage, Grade]

4. Data Dictionary and Final ERD

4.1. Data Dictionary

It is defined as the list of metadata of a table or relation. It provides a description of the entity or relation, columns, data type, length of data that is to be inserted into it, the keys used in the table and other data. It is an asset used to get into on the table on a deeper level. It helps in reducing the inconsistencies in data that can occur. They help with data modeling as it provides organized information about the table. (atlan, 2024)

Data dictionary of Program.

Entity name	Entity description	Column name	Column description	Data type	Length	PK	FK	Nullable	Unique	Notes
Program	Represents the programs available in the college.	Program_ID	Primary key for program	Int	38	Yes	No	No	Yes	Auto-Increment
		Program_Name	Program name	Varchar	50	No	No	Yes	No	
		Program_Duration_Years	Shows the duration of each program	Decimal	(2,1)	No	No	No	No	
		Program_Fees	Shows the fees of each program	Decimal	(9,1)	No	No	No	No	
		Total_Credits	Shows the total credits of each program	Int	38	No	No	No	No	

Table 10: Data Dictionary for Program entity.

Data Dictionary of Module.

Entity name	Entity description	Column name	Column description	Data type	Length	PK	FK	Nullable	Unique	Notes
Module	Represents the Module available in each Program.	Module_ID	Primary key for module	Int	38	Yes	No	No	Yes	Auto-Increment
		Module_Name	Module name	Varchar	100	No	No	Yes	No	
		Module_Duration_Months	Shows the duration of each Module	Number	38	No	No	Yes	No	
		Credits	Shows the credits of each Module	Int	38	No	No	Yes	No	

Table 11: Data Dictionary for Module entity.

Data Dictionary of Student.

Entity name	Entity description	Column name	Column description	Data type	Length	PK	FK	Nullable	Unique	Notes
Student	Represents the Student studying in each Program.	Student_ID	Primary key for student	Int	38	Yes	No	No	Yes	Auto-Increment
		First_Name	Shows the first name of student	Varchar	50	No	No	No	No	
		Last_Name	Shows the last name of student	Varchar	50	No	No	No	No	
		Email	Shows the email of the student	Varchar	250	No	No	Yes	No	
		Phone	Shows the Contact number of student	Varchar	15	No	No	Yes	Yes	
		Address	Shows the address of the student	Varchar	100	No	No	Yes	No	
		Enroll_Date	Shows the enroll date of the student	Date	---	No	No	No	No	
		Program_ID	Shows the enrolled program of the student.	Int	38	No	Yes	Yes	No	

Table 12: Data Dictionary for Student entity

Data Dictionary of Student_Module

Entity name	Entity Description	Column name	Column description	Data type	Length	PK	FK	Nullable	Unique	Notes
Student_Module	Is a junction table between Student and Module	Student_ID	Part of composite key	Int	10	Yes	Yes	No	Yes	
		Module_ID	Part of composite key	Int	10	Yes	Yes	No	Yes	

Table 13: Data Dictionary for Student_Module entity.

Data Dictionary of Module table:

Entity Name	Entity Description	Column name	Column description	Data type	Length	PK	FK	Nullable	Unique	Notes
Module	This table stores data of all the modules available in the institution.	Module_ ID	Primary key for the module	Int	10	Yes	No	No	Yes	Auto-Increment
		Module_ Name	Name of the module	Varchar	50	No	No	No	No	
		Duration _Months	Duration of the module in months	Decimal	(3,1)	No	No	No	No	
		Credits	Credits of the module	Int	10	No	No	No	No	
		Teacher_ ID	Identifier of the teacher teaching the module	Int	10	No	Yes	Yes	No	

Table 14: Data Dictionary for Module entity.

Data dictionary of Module_Resource table

Entity Name	Entity Description	Column name	Column description	Data type	Length	PK	FK	Nullable	Unique	Notes
Module_Resource	This is the junction table between Module and Resources	Student_ID	Part of composite key	Int	10	Yes	Yes	No	Yes	
		Module_ID	Part of composite key	Int	10	Yes	Yes	No	Yes	
		Resource_ID	Part of composite key	Int	10	Yes	No	No	Yes	
		Status	Status of resource completion.	Varchar	20	No	No	No	No	

Table 15: Data Dictionary for Module_Resource entity.

Data dictionary of Resources table

Entity Name	Entity Description	Column name	Column description	Data type	Length	PK	FK	Nullable	Unique	Notes
Resources	This entity contains data of all the resources available in the modules	Resource_ ID	Primary key for the resource	Int	10	Yes	No	No	Yes	Auto-Increment
		Format	Format of the resource	Varchar	20	No	No	No	No	
		Resource_ Name	Name of the resource	Varchar	50	No	No	No	No	
		Duration_ Weeks	Duration of the resource in weeks	Decimal	(3,1)	No	No	No	No	

Table 16: Data Dictionary for Resources entity.

Data dictionary of Teacher table

Entity Name	Entity Description	Column name	Column description	Data type	Length	PK	FK	Nullable	Unique	Notes
Teacher	This entity contains data of all the teachers in the institution	Teacher_ID	Primary key for the teacher	Int	10	Yes	No	No	Yes	Auto-Increment
		First_Name	First name of the teacher	Varchar	20	No	No	No	No	
		Last_Name	Last name of the teacher	Varchar	20	No	No	No	No	
		Specialization	Specialization of the teacher	Varchar	50	No	No	No	No	
		Contact_Number	Contact number of the teacher	Varchar	15	No	No	Yes	Yes	
		Join_Date	Date of teacher joining	Date	---	No	No	No	No	

Table 17: Data Dictionary for Teacher entity.

Data Dictionary of Module_Announcement table

Entity Name	Entity Description	Column name	Column description	Data type	Length	PK	FK	Nullable	Unique	Nots
Module_Announcement	This is the junction table between Module and Announcements	Student_ID	Part of composite key	Int	10	Yes	Yes	No	Yes	
		Module_ID	Part of composite key	Int	10	Yes	Yes	No	Yes	
		Announcement_ID	Part of composite key	Int	10	Yes	No	No	Yes	

Table 18: Data Dictionary for Module_Announcement entity.

Data Dictionary of Announcement table

Entity Name	Entity Description	Column name	Column description	Data type	Length	PK	FK	Nullable	Unique	Notes
Announcement	This entity contains info about all the announcements made	Announcement_ID	Primary key for the announcement	Int	10	Yes	No	No	Yes	Auto-Increment
		Announcement_Title	Title of the announcement	Varchar	50	No	No	No	No	
		Body	Body content of the announcement	Text	---	No	No	No	No	
		Date_Posted	Date the announcement was posted	Date	---	No	No	No	No	
		Expiry_Date	Expiry date of the announcement	Date	---	No	No	Yes	No	

Table 19: Data Dictionary for Announcement entity.

Data Dictionary of Module_Assessment table

Entity Name	Entity Description	Column name	Column description	Data type	Length	PK	FK	Nullable	Unique	Notes
Module_Assessment	This is the junction table for Module and Assessment	Module_ID	Part of composite key	Int	10	Yes	Yes	No	Yes	
		Student_ID	Part of composite key	Int	10	Yes	Yes	No	Yes	
		Assessment_ID	Part of composite key	Int	10	Yes	No	No	Yes	

Table 20: Data Dictionary for Module_Assessment entity.

Data Dictionary of Assessment table

Entity Name	Entity Description	Column name	Column description	Data type	Length	PK	FK	Nullable	Unique	Notes
Assessment	This entity contains info about all the given assessments	Assessment_ID	Primary key for the assessment	Int	10	Yes	No	No	Yes	Auto increase
		Title	Title of the assessment	Varchar	50	No	No	No	No	
		Deadline	Deadline for the assessment submission	Date	---	No	No	No	No	
		Weightage_Percentage	Weightage of the assessment in percentage	Decimal	(5,2)	No	No	No	No	
		Full_Marks	Full marks of the assessment	Int	10	No	No	No	No	

Table 21: Data Dictionary for Assessment entity.

Data Dictionary of Assessment_Result table

Entity Name	Entity Description	Column name	Column description	Data type	Length	PK	FK	Nullable	Unique	Notes
Assessment_Result	This table consists related into about Assessment and Result	Student_ID	Part of composite key	Int	10	Yes	Yes	No	Yes	
		Module_ID	Part of composite key	Int	10	Yes	Yes	No	Yes	
		Assessment_ID	Part of composite key	Int	10	Yes	Yes	No	Yes	
		Result_ID	Part of composite key	Int	10	Yes	Yes	No	Yes	
		Marks_Obtained	Marks obtained in the assessment	Int	10	No	No	No	No	
		Feedback	Feedback for the assessment	Text	---	No	No	Yes	No	
		Percentage	Percentage obtained in the assessment	Decimal	(5,2)	No	No	No	No	
		Grade	Grade awarded for the assessment	Varchar	5	No	No	No	No	

Table 22: Data Dictionary for Assessment_Result entity.

4.2. Final ERD

A final ERD is prepared after performing normalization. This Final ERD shows the correct relationships between the entities and precisely showcase the data integrity and removal of data redundancy. This ERD is then used to create a database which holds data consistently. A set of entities are added after performing normalization. They are given below:

Lists of Entities after Normalization

Entity	**Description**
Program	It carries data of different programs taught in the institution.
Modules	It carries data of all the Modules available in the institution.
Students	It carries data of all the students enrolled in the institution.
Resources	It carries data of all the resources available in specific modules.
Teacher	It carries data of all the teachers in the institution.
Announcement	It carries data of all the announcements made by the modules.
Assessment	It carries data of all the assessments for the modules.
Assessment_Result	It carries data of Result and assessments of students.
Module_Announcement	It is a junction table between Module and Announcement.
Module_Assessment	It is a junction table between Module and Assessment.
Module_Resource	It is a junction table between Module and Resources.
Student_Module	It is a junction table between Student and Module.

Table 23: List of entities after Normalization.

Below are Entities and Attributes after Normalization

- **Program**

S. No.	Attribute Name	Data Type	Size	Constraint
1	Program_ID	Number	38	Primary Key
2	Program_Name	Character	50	-
3	Program_Duration_Years	Number	Precision 2, Scale 1	Not Null
4	Program_Fees	Decimal	Precision 2, Scale 1	Not Null
5	Total_Credits	Number	38	Not Null

Table 24: Attributes of Program entity.

- **Module**

S. No.	Attribute Name	Data Type	Size	Constraint
1	Module_ID	Number	38	Primary Key
2	Module_Name	Character	50	-
3	Duration_Months	Number	38	-
4	Credits	Number	38	-
5	Teacher_ID	Number	38	Foreign Key

Table 25: Attributes of Module entity.

- **Student**

S. No.	Attribute Name	Data Type	Size	Constraint
1	Student_ID	Number	38	Primary Key
2	First_Name	Character	50	Not Null
3	Last_Name	Character	20	Not Null
4	Email	Character	20	Unique
5	Phone	Character	15	-
6	Address	Character	25	-
7	Enroll_Date	Date		Not Null
8	Program_ID	Number	38	Foreign Key

Table 26: Attributes of Student entity.

- **Resources**

S. No.	Attribute Name	Data Type	Size	Constraint
1	Resource_ID	Number	38	Primary Key
2	Resource_Name	Character	50	-
3	Format	Character	50	-
4	Duration_Weeks	Number	38	-

Table 27: Attributes of Resources entity.

- **Teacher**

S. No.	Attribute Name	Data Type	Size	Constraint
1	Teacher_ID	Number	38	Primary Key
2	First_Name	Character	20	Not Null
3	Last_Name	Character	20	Not Null
4	Specialization	Character	50	Not Null
5	Contact_Number	Character	15	-
6	Address	Character	25	-
7	Join_Date	Date		Not Null

Table 28: Attributes of Teacher entity.

- **Announcement**

S. No.	Attribute Name	Data Type	Size	Constraint
1	Announcement_ID	Number	38	Primary Key
2	Announcement_Title	Character	100	Not Null
3	Body	Character	250	-
4	Date_Posted	Date		Not Null
5	Expiry_Date	Date		Not Null

Table 29: Attributes of Announcement entity.

- **Assessment**

S. No.	Attribute Name	Data Type	Size	Constraint
1	Assessment_ID	Number	38	Primary Key
2	Title	Character	50	Not Null
3	Deadline	Date		Not Null
4	Weightage_Percentage	Number	Precision 5, Scale 2	Not Null
5	Full_Marks	Number	38	Not Null

Table 30: Attributes of Assessment entity.

- **Assessment_Result**

S. No.	Attribute Name	Data Type	Size	Constraint
1	Result_ID	Number	38	PK
2	Assessment_ID	Number	38	PK, FK
3	Module_ID	Number	38	PK, FK
4	Student_ID	Number	38	PK, FK
2	Marks_Obtained	Number	38	-
3	Grade	Character	1	-
4	Percentage	Number	Precision 5, Scale 2	-

Table 31: Attributes of Assessment_Result entity.

- **Module_Assessment**

S. No.	Attribute Name	Data Type	Size	Constraint
1	Module_ID	Number	38	PK, FK
2	Assessment_ID	Number	38	PK, FK
3	Student_ID	Number	38	PK, FK

Table 32: Attributes of Module_Assessment entity.

- **Module_Announcement**

S. No.	Attribute Name	Data Type	Size	Constraint
1	Student_ID	Number	38	PK, FK
2	Module_ID	Number	38	PK, FK
3	Announcement_ID	Number	38	PK, FK

Table 33: Attributes of Module_Announcement entity.

- **Student_Module**

S. No.	Attribute Name	Data Type	Size	Constraint
1	Student_ID	Number	38	PK, FK
2	Module_ID	Number	38	PK, FK

Table 34: Attributes of Student_Module entity.

- **Module_Resource**

S. No.	Attribute Name	Data Type	Size	Constraint
1	Module_ID	Number	38	PK, FK
2	Resource_ID	Number	38	PK, FK
3	Student_ID	Number	38	PK, FK
4	Status	Character	9	Not Null

Table 35: Attributes of Module_Resource entity.

Final ERD:

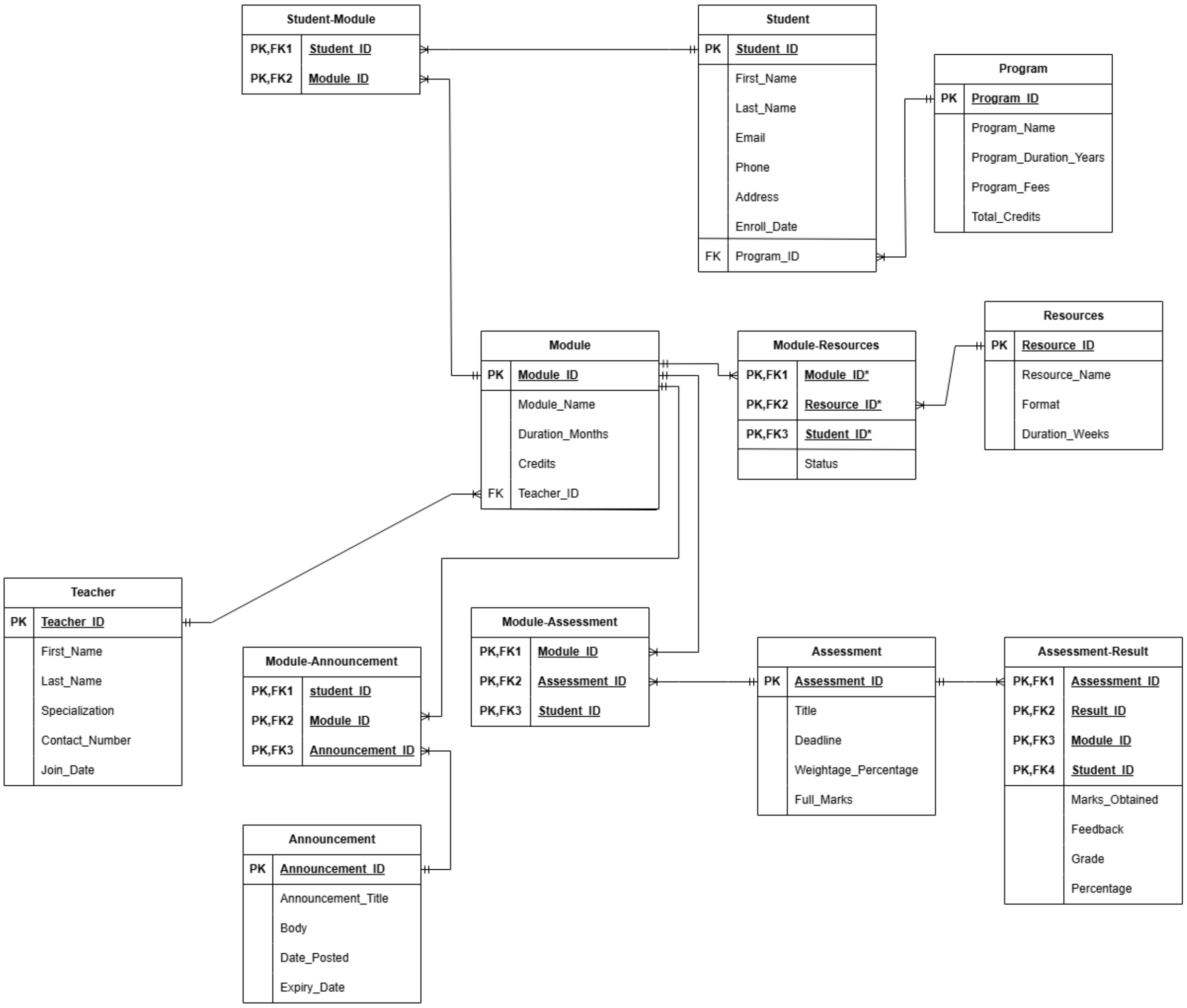


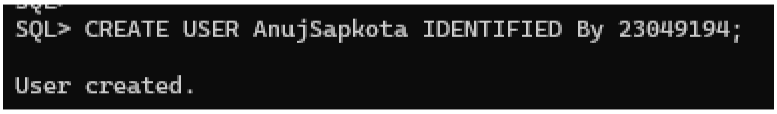
Figure 10: Final ERD.

5. Implementation

After normalizing the initial database and Developing a proper final ERD, different operations are performed to create the database. Database ae created by users so first we must create a user who will create the tables and insert data onto it. After that, tables are created having no foreign keys, then tables having foreign keys are created and then tables having composite keys are created.

5.1. User Creation

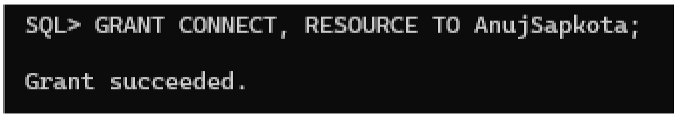
1. Developing User with Password



```
SQL> CREATE USER AnujSapkota IDENTIFIED By 23049194;

User created.
```

Figure 11: Developing User.

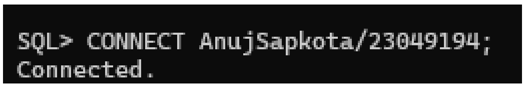
2. Granting Connect and Resources to the Created User



```
SQL> GRANT CONNECT, RESOURCE TO AnujSapkota;

Grant succeeded.
```

Figure 12: Granting Connect and Resources to the user.

3. Connecting the User



```
SQL> CONNECT AnujSapkota/23049194;
Connected.
```

Figure 13: Connecting to the user.

5.2. Tables Creation

In SQL, tables are created using CREATE TABLE statement with the name of the table after it. The column names and their Data type and Constraints are written inside the parenthesis. We can assign keys in table in mainly 2 ways: First, in the same line as the column creation and second is after creation of the column both helps us make the column any key we want. The Syntax for table creation is :

CREATE TABLE *table_Name (*

Row1 Data_type Constaint,

Row2 Data_type Constraint,

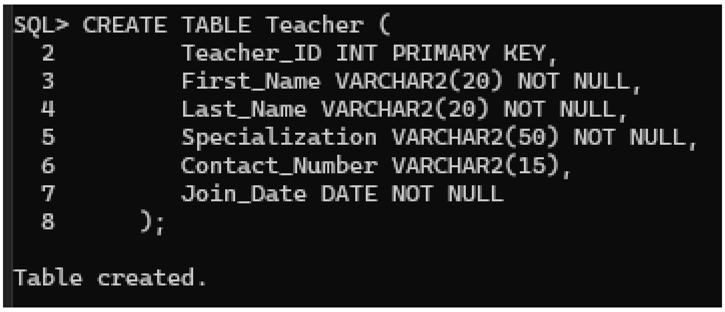
…………………………………

Primary key (row1, row2,……) [Optional]

Foreign key (row1) References another_table (row1) [Optional]

)

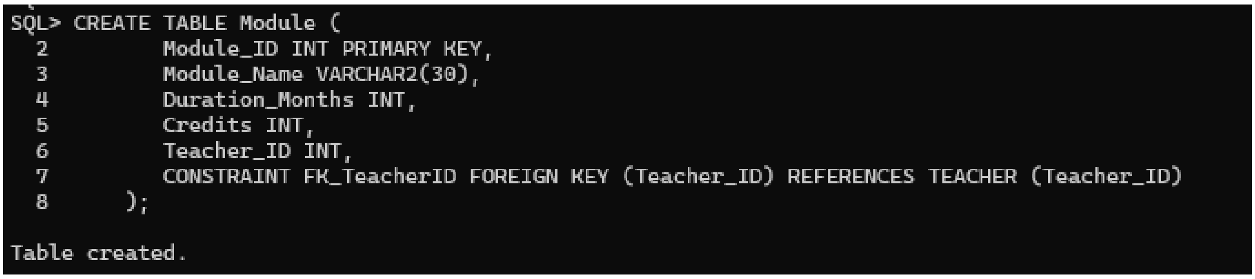
1. Developing Teacher table



```
SQL> CREATE TABLE Teacher (
        Teacher_ID INT PRIMARY KEY,
        First_Name VARCHAR2(20) NOT NULL,
        Last_Name VARCHAR2(20) NOT NULL,
        Specialization VARCHAR2(50) NOT NULL,
        Contact_Number VARCHAR2(15),
        Join_Date DATE NOT NULL
     );

Table created.
```

Figure 14: Developing Teacher table.

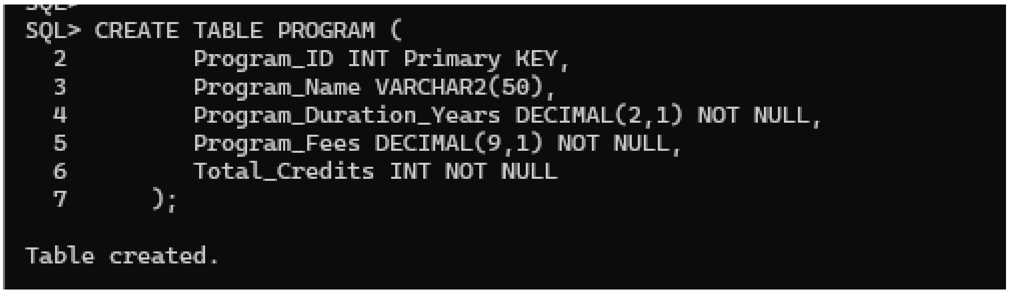
2. Developing Module table



```
SQL> CREATE TABLE Module (
        Module_ID INT PRIMARY KEY,
        Module_Name VARCHAR2(30),
        Duration_Months INT,
        Credits INT,
        Teacher_ID INT,
        CONSTRAINT FK_TeacherID FOREIGN KEY (Teacher_ID) REFERENCES TEACHER (Teacher_ID)
     );

Table created.
```

Figure 15: Developing Module table.

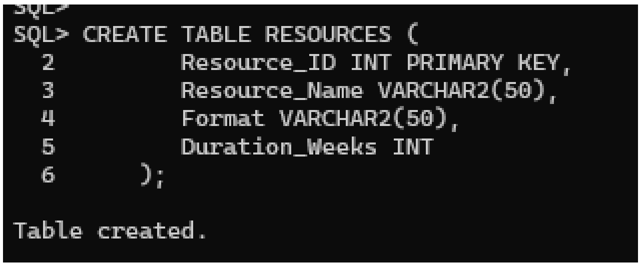
3. Developing Program table



```
SQL> CREATE TABLE PROGRAM (
             Program_ID INT Primary KEY,
             Program_Name VARCHAR2(50),
             Program_Duration_Years DECIMAL(2,1) NOT NULL,
             Program_Fees DECIMAL(9,1) NOT NULL,
             Total_Credits INT NOT NULL
          );

Table created.
```

Figure 16: Developing Program table.

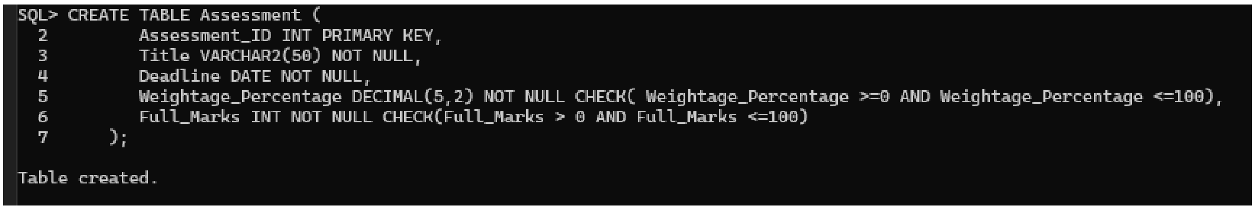
4. Developing Resources table



```
SQL> CREATE TABLE RESOURCES (
             Resource_ID INT PRIMARY KEY,
             Resource_Name VARCHAR2(50),
             Format VARCHAR2(50),
             Duration_Weeks INT
          );

Table created.
```

Figure 17: Developing Resources table.

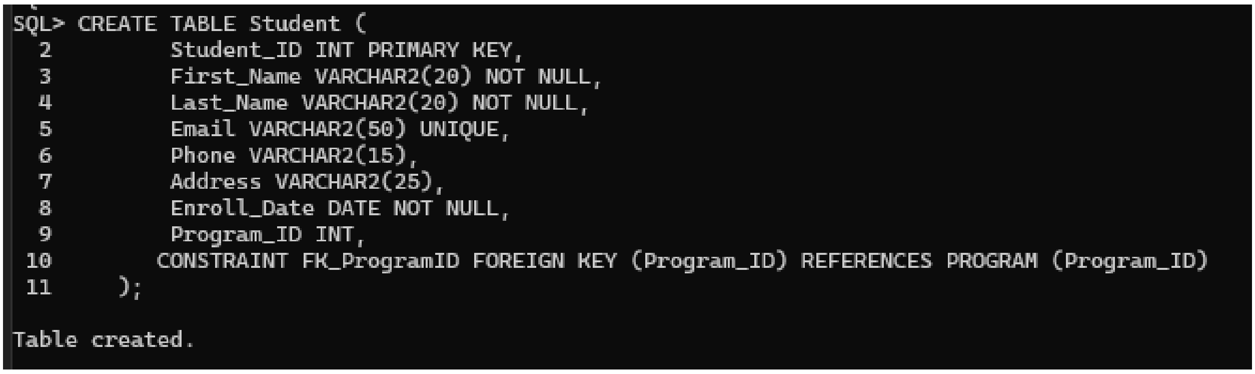
5. Developing Assessment table



```
SQL> CREATE TABLE Assessment (
             Assessment_ID INT PRIMARY KEY,
             Title VARCHAR2(50) NOT NULL,
             Deadline DATE NOT NULL,
             Weightage_Percentage DECIMAL(5,2) NOT NULL CHECK( Weightage_Percentage >=0 AND Weightage_Percentage <=100),
             Full_Marks INT NOT NULL CHECK(Full_Marks > 0 AND Full_Marks <=100)
         );

Table created.
```

Figure 18: Developing Assessment table.

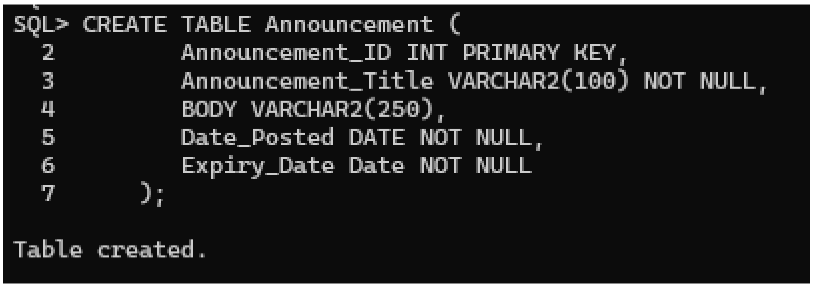
6. Developing Student table



```
SQL> CREATE TABLE Student (
             Student_ID INT PRIMARY KEY,
             First_Name VARCHAR2(20) NOT NULL,
             Last_Name VARCHAR2(20) NOT NULL,
             Email VARCHAR2(50) UNIQUE,
             Phone VARCHAR2(15),
             Address VARCHAR2(25),
             Enroll_Date DATE NOT NULL,
             Program_ID INT,
            CONSTRAINT FK_ProgramID FOREIGN KEY (Program_ID) REFERENCES PROGRAM (Program_ID)
        );

Table created.
```

Figure 19: Developing Student Table.

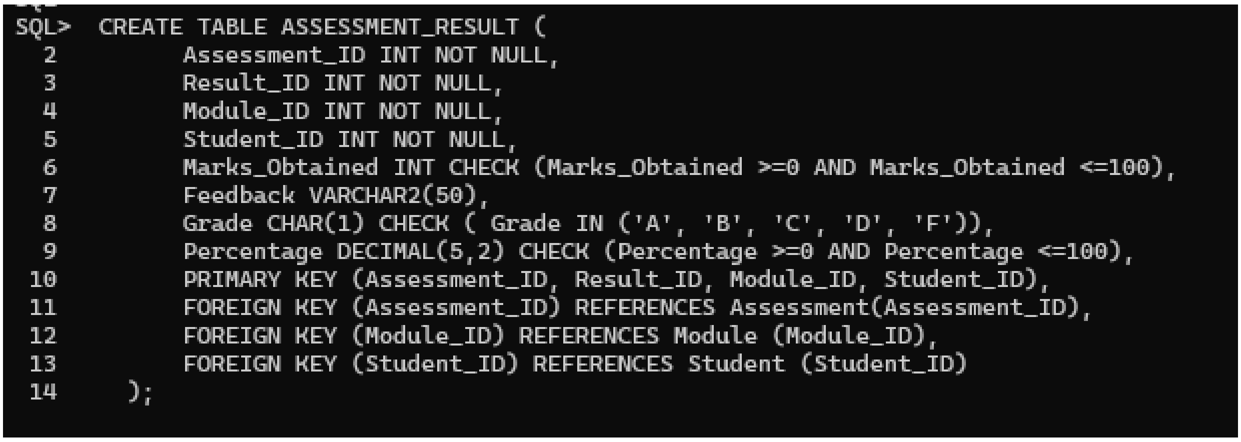
7. Developing Announcement table



```
SQL> CREATE TABLE Announcement (
        Announcement_ID INT PRIMARY KEY,
        Announcement_Title VARCHAR2(100) NOT NULL,
        BODY VARCHAR2(250),
        Date_Posted DATE NOT NULL,
        Expiry_Date Date NOT NULL
     );

Table created.
```

Figure 20: Developing Announcement table.

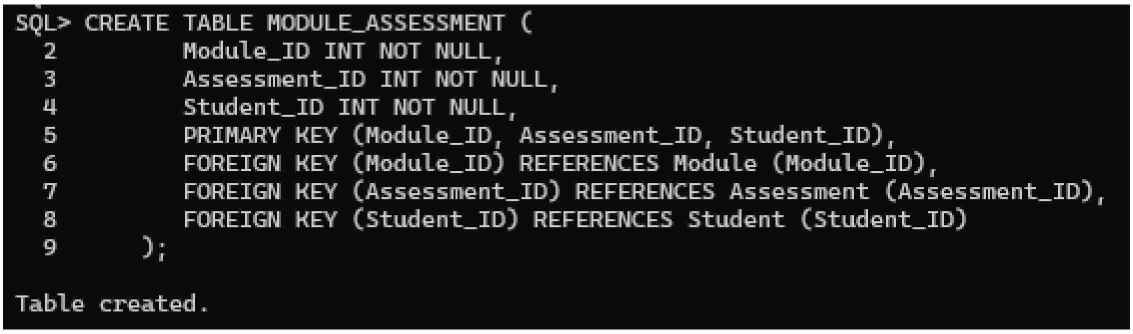
8. Developing Assessment_Result table



```
SQL>  CREATE TABLE ASSESSMENT_RESULT (
        Assessment_ID INT NOT NULL,
        Result_ID INT NOT NULL,
        Module_ID INT NOT NULL,
        Student_ID INT NOT NULL,
        Marks_Obtained INT CHECK (Marks_Obtained >=0 AND Marks_Obtained <=100),
        Feedback VARCHAR2(50),
        Grade CHAR(1) CHECK ( Grade IN ('A', 'B', 'C', 'D', 'F')),
        Percentage DECIMAL(5,2) CHECK (Percentage >=0 AND Percentage <=100),
        PRIMARY KEY (Assessment_ID, Result_ID, Module_ID, Student_ID),
        FOREIGN KEY (Assessment_ID) REFERENCES Assessment(Assessment_ID),
        FOREIGN KEY (Module_ID) REFERENCES Module (Module_ID),
        FOREIGN KEY (Student_ID) REFERENCES Student (Student_ID)
     );
```

Figure 21: Developing Assessment_Result table.

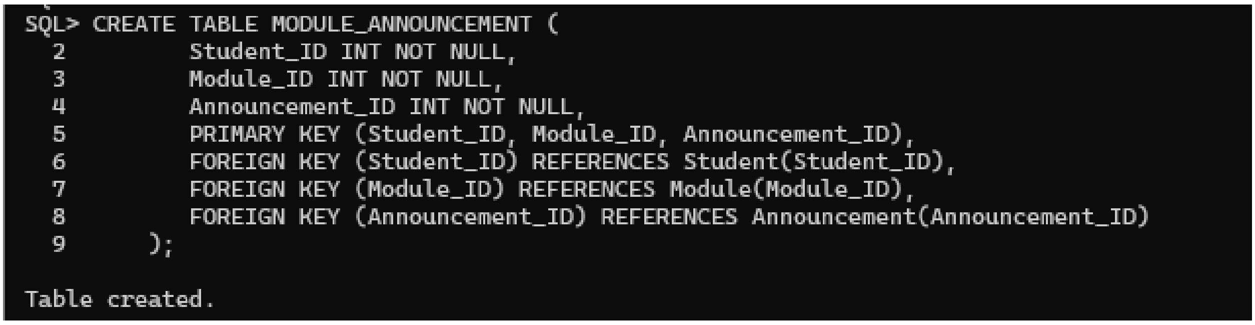
9. Developing Module_Assessment table



```
SQL> CREATE TABLE MODULE_ASSESSMENT (
        Module_ID INT NOT NULL,
        Assessment_ID INT NOT NULL,
        Student_ID INT NOT NULL,
        PRIMARY KEY (Module_ID, Assessment_ID, Student_ID),
        FOREIGN KEY (Module_ID) REFERENCES Module (Module_ID),
        FOREIGN KEY (Assessment_ID) REFERENCES Assessment (Assessment_ID),
        FOREIGN KEY (Student_ID) REFERENCES Student (Student_ID)
     );

Table created.
```

Figure 22: Developing Module_Assessment table.

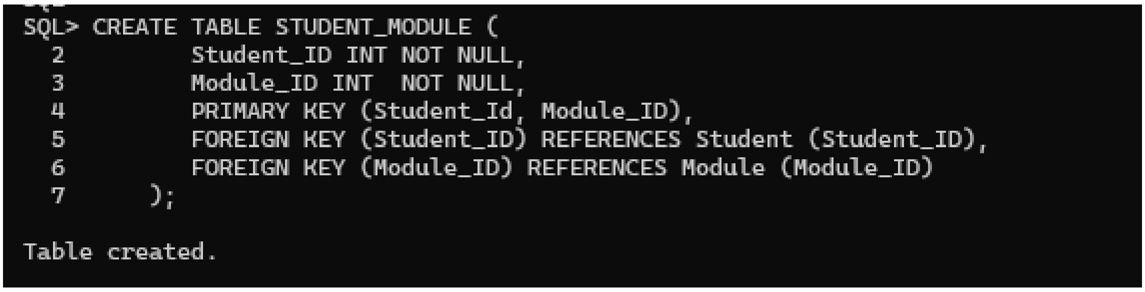
10. Developing Module_Announcement table



```
SQL> CREATE TABLE MODULE_ANNOUNCEMENT (
            Student_ID INT NOT NULL,
            Module_ID INT NOT NULL,
            Announcement_ID INT NOT NULL,
            PRIMARY KEY (Student_ID, Module_ID, Announcement_ID),
            FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),
            FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID),
            FOREIGN KEY (Announcement_ID) REFERENCES Announcement(Announcement_ID)
        );

Table created.
```

Figure 23: Developing Module_Announcement table.

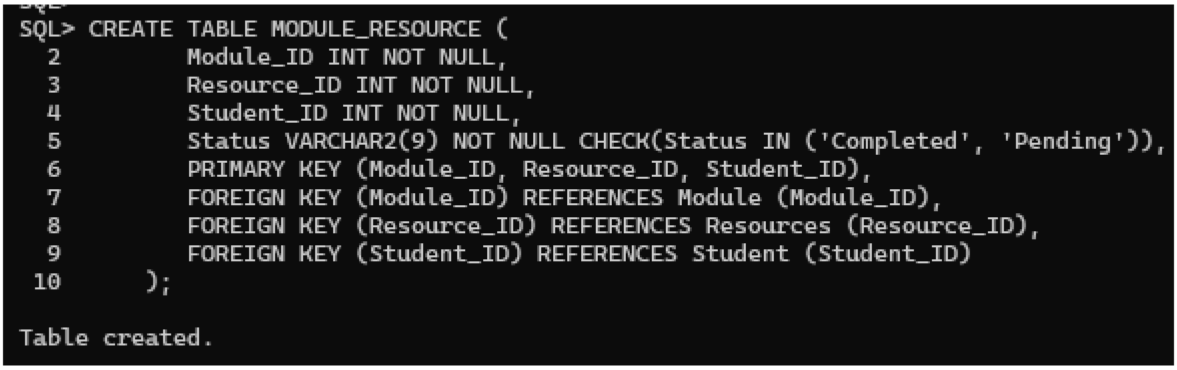
11. Developing Student_Module table



```
SQL> CREATE TABLE STUDENT_MODULE (
            Student_ID INT NOT NULL,
            Module_ID INT  NOT NULL,
            PRIMARY KEY (Student_Id, Module_ID),
            FOREIGN KEY (Student_ID) REFERENCES Student (Student_ID),
            FOREIGN KEY (Module_ID) REFERENCES Module (Module_ID)
        );

Table created.
```

Figure 24: Developing Student_Module table.

12. Developing Module_Resource table



```
SQL> CREATE TABLE MODULE_RESOURCE (
            Module_ID INT NOT NULL,
            Resource_ID INT NOT NULL,
            Student_ID INT NOT NULL,
            Status VARCHAR2(9) NOT NULL CHECK(Status IN ('Completed', 'Pending')),
            PRIMARY KEY (Module_ID, Resource_ID, Student_ID),
            FOREIGN KEY (Module_ID) REFERENCES Module (Module_ID),
            FOREIGN KEY (Resource_ID) REFERENCES Resources (Resource_ID),
            FOREIGN KEY (Student_ID) REFERENCES Student (Student_ID)
        );

Table created.
```

Figure 25: Developing Module_Resource table.

5.3. Adding Rows to the Tables.

After successfully Developing the necessary tables, data/ rows need to be inserted into the tables. Necessary data are inserted into the table filling the table with the data. INSERT statement is used to input data into a table. We can insert data into the table using two ways. The first way is inserting every row individually: For this way we insert data one by one this method is simple but a bit time taking. The syntax for this method is :

INSERT INTO *table_Name (row1, row2,……)* VALUES (*data1, data2,…..);*

Another method is inserting all data at once. This method is little complex but still very useful when inserting a chunk of data all at once. The syntax for this method is:

INSERT ALL

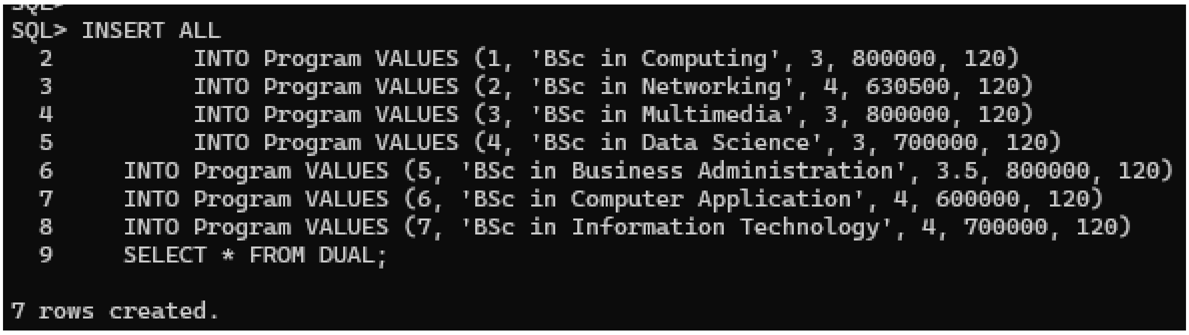
INTO *table_Name* (row1, row2,..) VALUES (data1, data2,..)

INTO *table_Name* (row1, row2,..) VALUES (data3, data4,..)

……………………………………………………………….

SELECT * FROM DUAL;

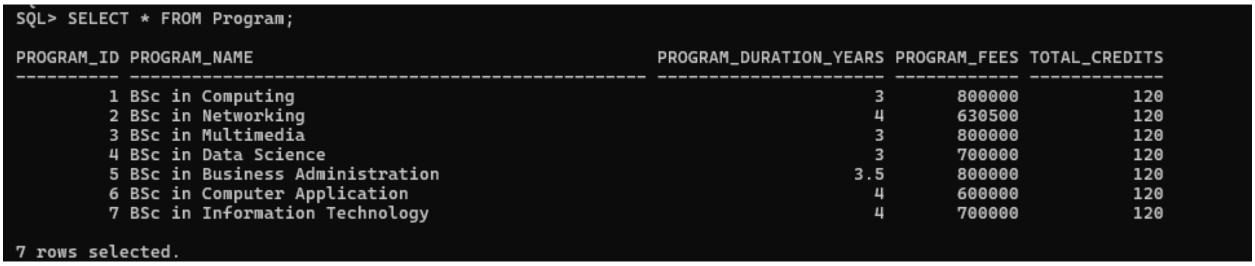
1. **Putting data inside the Program table**.



```
SQL> INSERT ALL
          INTO Program VALUES (1, 'BSc in Computing', 3, 800000, 120)
          INTO Program VALUES (2, 'BSc in Networking', 4, 630500, 120)
          INTO Program VALUES (3, 'BSc in Multimedia', 3, 800000, 120)
          INTO Program VALUES (4, 'BSc in Data Science', 3, 700000, 120)
     INTO Program VALUES (5, 'BSc in Business Administration', 3.5, 800000, 120)
     INTO Program VALUES (6, 'BSc in Computer Application', 4, 600000, 120)
     INTO Program VALUES (7, 'BSc in Information Technology', 4, 700000, 120)
     SELECT * FROM DUAL;

7 rows created.
```

Figure 26: Putting data inside Program Table.

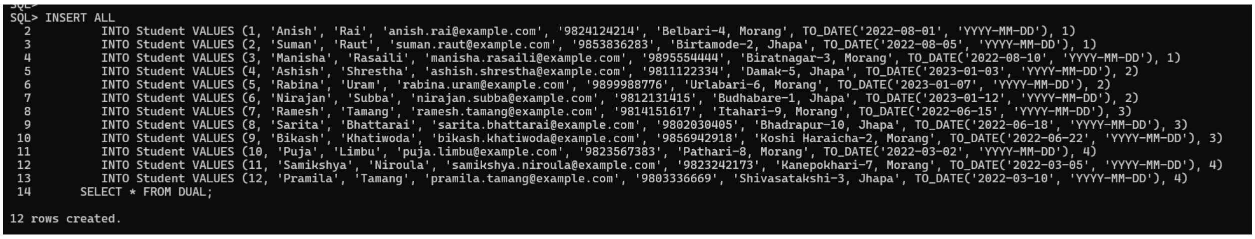
2. **Using SELECT statement to retrieve Program data**



```
SQL> SELECT * FROM Program;

PROGRAM_ID PROGRAM_NAME                                       PROGRAM_DURATION_YEARS PROGRAM_FEES TOTAL_CREDITS
---------- -------------------------------------------------- ---------------------- ------------ -------------
         1 BSc in Computing                                                        3       800000           120
         2 BSc in Networking                                                       4       630500           120
         3 BSc in Multimedia                                                       3       800000           120
         4 BSc in Data Science                                                     3       700000           120
         5 BSc in Business Administration                                        3.5       800000           120
         6 BSc in Computer Application                                             4       600000           120
         7 BSc in Information Technology                                           4       700000           120

7 rows selected.
```

Figure 27: Retrieving Program Data.

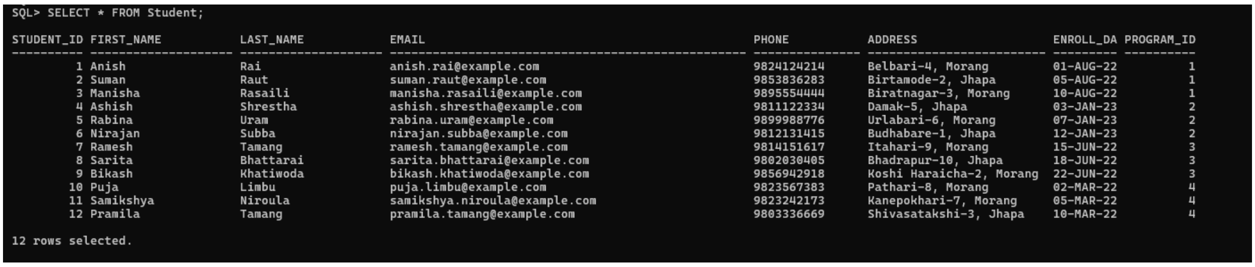
3. Putting data inside the Student table



```
SQL>
SQL> INSERT ALL
        INTO Student VALUES (1, 'Anish', 'Rai', 'anish.rai@example.com', '9824124214', 'Belbari-4, Morang', TO_DATE('2022-08-01', 'YYYY-MM-DD'), 1)
        INTO Student VALUES (2, 'Suman', 'Raut', 'suman.raut@example.com', '9853836283', 'Birtamode-2, Jhapa', TO_DATE('2022-08-05', 'YYYY-MM-DD'), 1)
        INTO Student VALUES (3, 'Manisha', 'Rasaili', 'manisha.rasaili@example.com', '9895554444', 'Biratnagar-3, Morang', TO_DATE('2022-08-10', 'YYYY-MM-DD'), 1)
        INTO Student VALUES (4, 'Ashish', 'Shrestha', 'ashish.shrestha@example.com', '9811122334', 'Damak-5, Jhapa', TO_DATE('2023-01-03', 'YYYY-MM-DD'), 2)
        INTO Student VALUES (5, 'Rabina', 'Uram', 'rabina.uram@example.com', '9899988776', 'Urlabari-6, Morang', TO_DATE('2023-01-07', 'YYYY-MM-DD'), 2)
        INTO Student VALUES (6, 'Nirajan', 'Subba', 'nirajan.subba@example.com', '9812131415', 'Budhabare-1, Jhapa', TO_DATE('2023-01-12', 'YYYY-MM-DD'), 2)
        INTO Student VALUES (7, 'Ramesh', 'Tamang', 'ramesh.tamang@example.com', '9814151617', 'Itahari-9, Morang', TO_DATE('2022-06-15', 'YYYY-MM-DD'), 3)
        INTO Student VALUES (8, 'Sarita', 'Bhattarai', 'sarita.bhattarai@example.com', '9802030405', 'Bhadrapur-10, Jhapa', TO_DATE('2022-06-18', 'YYYY-MM-DD'), 3)
        INTO Student VALUES (9, 'Bikash', 'Khatiwoda', 'bikash.khatiwoda@example.com', '9856942918', 'Koshi Haraicha-2, Morang', TO_DATE('2022-06-22', 'YYYY-MM-DD'), 3)
        INTO Student VALUES (10, 'Puja', 'Limbu', 'puja.limbu@example.com', '9823567383', 'Pathari-8, Morang', TO_DATE('2022-03-02', 'YYYY-MM-DD'), 4)
        INTO Student VALUES (11, 'Samikshya', 'Niroula', 'samikshya.niroula@example.com', '9823242173', 'Kanepokhari-7, Morang', TO_DATE('2022-03-05', 'YYYY-MM-DD'), 4)
        INTO Student VALUES (12, 'Pramila', 'Tamang', 'pramila.tamang@example.com', '9803336669', 'Shivasatakshi-3, Jhapa', TO_DATE('2022-03-10', 'YYYY-MM-DD'), 4)
     SELECT * FROM DUAL;

12 rows created.
```

Figure 28: putting Data inside Student table.

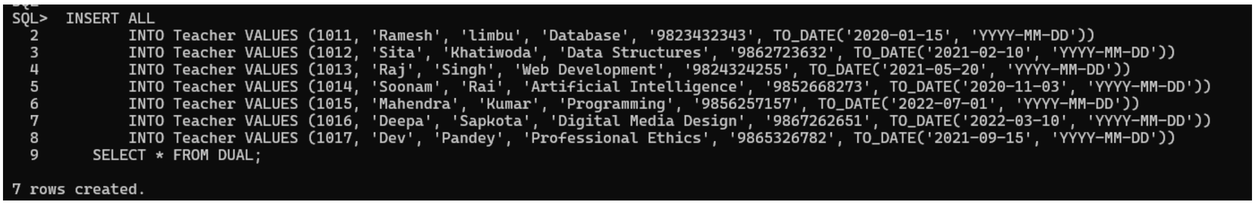
4. Using SELECT statement to retrieve data from Student table



```
SQL> SELECT * FROM Student;

STUDENT_ID FIRST_NAME           LAST_NAME            EMAIL                                              PHONE           ADDRESS                   ENROLL_DA PROGRAM_ID
---------- -------------------- -------------------- -------------------------------------------------- --------------- ------------------------- --------- ----------
         1 Anish                Rai                  anish.rai@example.com                              9824124214      Belbari-4, Morang         01-AUG-22          1
         2 Suman                Raut                 suman.raut@example.com                             9853836283      Birtamode-2, Jhapa        05-AUG-22          1
         3 Manisha              Rasaili              manisha.rasaili@example.com                        9895554444      Biratnagar-3, Morang      10-AUG-22          1
         4 Ashish               Shrestha             ashish.shrestha@example.com                        9811122334      Damak-5, Jhapa            03-JAN-23          2
         5 Rabina               Uram                 rabina.uram@example.com                            9899988776      Urlabari-6, Morang        07-JAN-23          2
         6 Nirajan              Subba                nirajan.subba@example.com                          9812131415      Budhabare-1, Jhapa        12-JAN-23          2
         7 Ramesh               Tamang               ramesh.tamang@example.com                          9814151617      Itahari-9, Morang         15-JUN-22          3
         8 Sarita               Bhattarai            sarita.bhattarai@example.com                       9802030405      Bhadrapur-10, Jhapa       18-JUN-22          3
         9 Bikash               Khatiwoda            bikash.khatiwoda@example.com                       9856942918      Koshi Haraicha-2, Morang  22-JUN-22          3
        10 Puja                 Limbu                puja.limbu@example.com                             9823567383      Pathari-8, Morang         02-MAR-22          4
        11 Samikshya            Niroula              samikshya.niroula@example.com                      9823242173      Kanepokhari-7, Morang     05-MAR-22          4
        12 Pramila              Tamang               pramila.tamang@example.com                         9803336669      Shivasatakshi-3, Jhapa    10-MAR-22          4

12 rows selected.
```

Figure 29: Retrieving data from Student table.

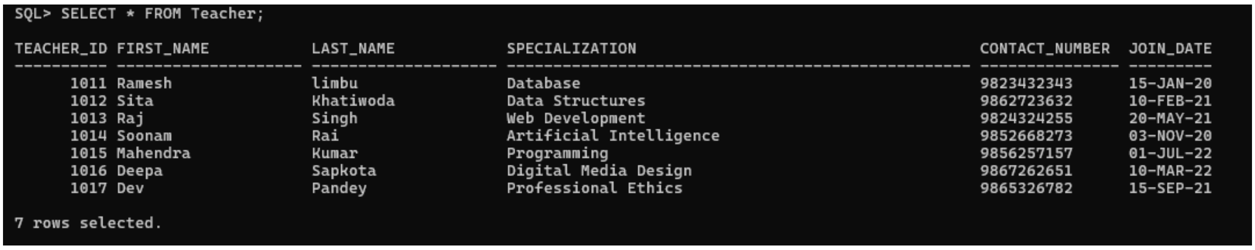
5. Putting data inside Teacher table



```
SQL>  INSERT ALL
        INTO Teacher VALUES (1011, 'Ramesh', 'limbu', 'Database', '9823432343', TO_DATE('2020-01-15', 'YYYY-MM-DD'))
        INTO Teacher VALUES (1012, 'Sita', 'Khatiwoda', 'Data Structures', '9862723632', TO_DATE('2021-02-10', 'YYYY-MM-DD'))
        INTO Teacher VALUES (1013, 'Raj', 'Singh', 'Web Development', '9824324255', TO_DATE('2021-05-20', 'YYYY-MM-DD'))
        INTO Teacher VALUES (1014, 'Soonam', 'Rai', 'Artificial Intelligence', '9852668273', TO_DATE('2020-11-03', 'YYYY-MM-DD'))
        INTO Teacher VALUES (1015, 'Mahendra', 'Kumar', 'Programming', '9856257157', TO_DATE('2022-07-01', 'YYYY-MM-DD'))
        INTO Teacher VALUES (1016, 'Deepa', 'Sapkota', 'Digital Media Design', '9867262651', TO_DATE('2022-03-10', 'YYYY-MM-DD'))
        INTO Teacher VALUES (1017, 'Dev', 'Pandey', 'Professional Ethics', '9865326782', TO_DATE('2021-09-15', 'YYYY-MM-DD'))
     SELECT * FROM DUAL;

7 rows created.
```

Figure 30: putting data inside Teacher table.

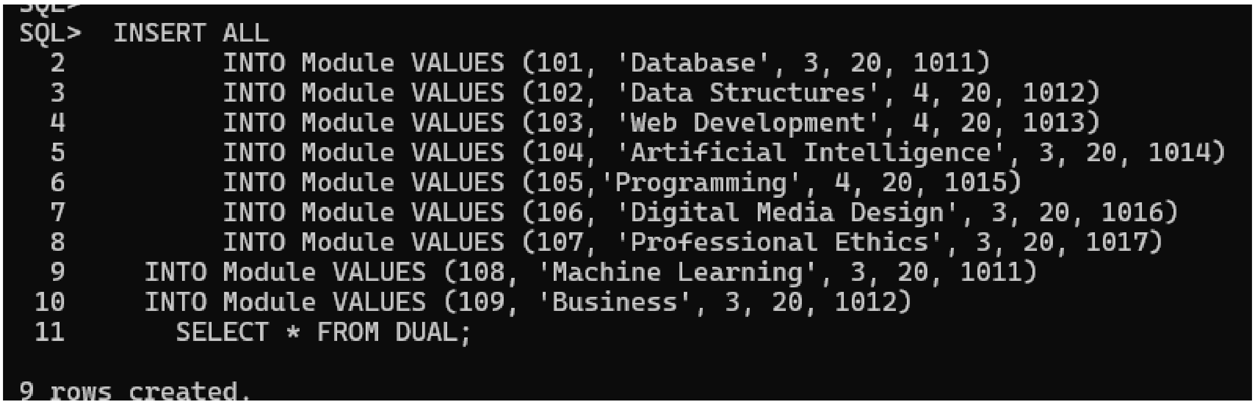
6. Using SELECT statement to extract data from Teacher table



```
SQL> SELECT * FROM Teacher;

TEACHER_ID FIRST_NAME           LAST_NAME            SPECIALIZATION                                     CONTACT_NUMBER  JOIN_DATE
---------- -------------------- -------------------- -------------------------------------------------- --------------- ---------
      1011 Ramesh               limbu                Database                                           9823432343      15-JAN-20
      1012 Sita                 Khatiwoda            Data Structures                                    9862723632      10-FEB-21
      1013 Raj                  Singh                Web Development                                    9824324255      20-MAY-21
      1014 Soonam               Rai                  Artificial Intelligence                            9852668273      03-NOV-20
      1015 Mahendra             Kumar                Programming                                        9856257157      01-JUL-22
      1016 Deepa                Sapkota              Digital Media Design                               9867262651      10-MAR-22
      1017 Dev                  Pandey               Professional Ethics                                9865326782      15-SEP-21

7 rows selected.
```

Figure 31: Retrieving rows from Teacher table.

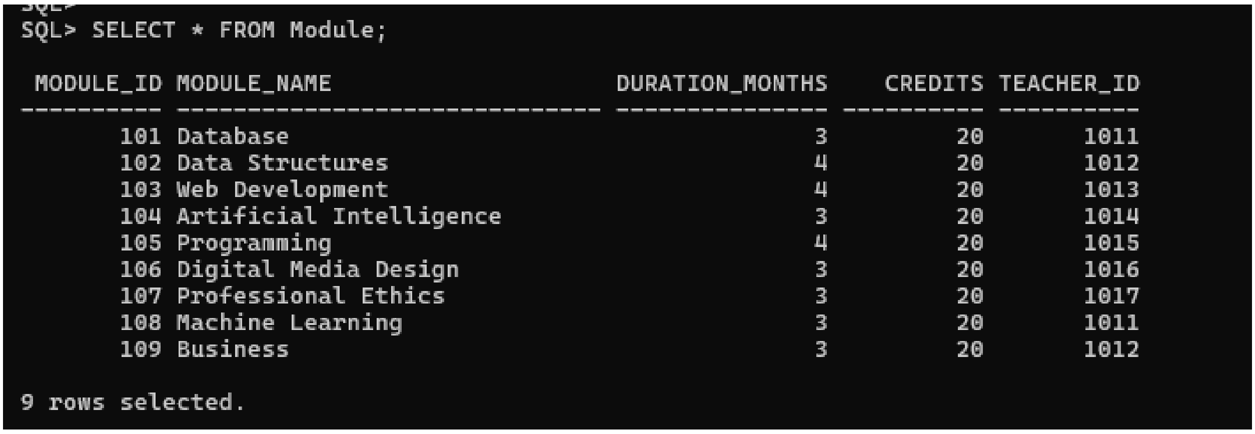
7. Putting data inside Module table



```
SQL>  INSERT ALL
  2         INTO Module VALUES (101, 'Database', 3, 20, 1011)
  3         INTO Module VALUES (102, 'Data Structures', 4, 20, 1012)
  4         INTO Module VALUES (103, 'Web Development', 4, 20, 1013)
  5         INTO Module VALUES (104, 'Artificial Intelligence', 3, 20, 1014)
  6         INTO Module VALUES (105,'Programming', 4, 20, 1015)
  7         INTO Module VALUES (106, 'Digital Media Design', 3, 20, 1016)
  8         INTO Module VALUES (107, 'Professional Ethics', 3, 20, 1017)
  9     INTO Module VALUES (108, 'Machine Learning', 3, 20, 1011)
 10     INTO Module VALUES (109, 'Business', 3, 20, 1012)
 11       SELECT * FROM DUAL;

9 rows created.
```

Figure 32: Putting data inside Module table.

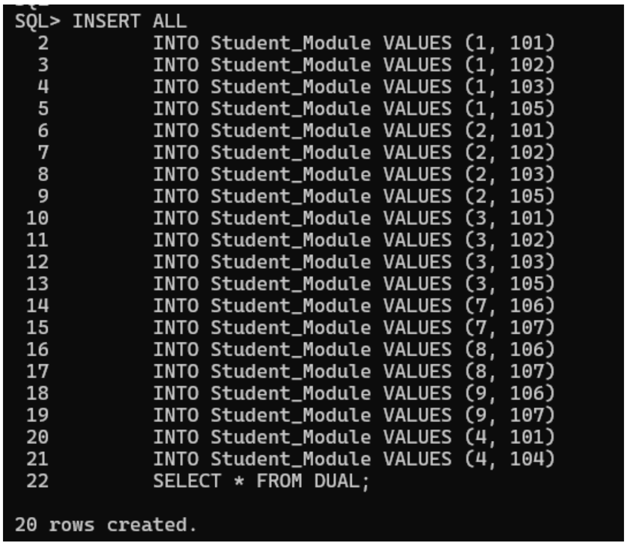
8. Using SELECT statement to retrieve data from Module table



```
SQL> SELECT * FROM Module;

 MODULE_ID MODULE_NAME                    DURATION_MONTHS    CREDITS TEACHER_ID
---------- ------------------------------ --------------- ---------- ----------
       101 Database                                     3         20       1011
       102 Data Structures                              4         20       1012
       103 Web Development                              4         20       1013
       104 Artificial Intelligence                      3         20       1014
       105 Programming                                  4         20       1015
       106 Digital Media Design                         3         20       1016
       107 Professional Ethics                          3         20       1017
       108 Machine Learning                             3         20       1011
       109 Business                                     3         20       1012

9 rows selected.
```

Figure 33: Retrieving data from Module table.

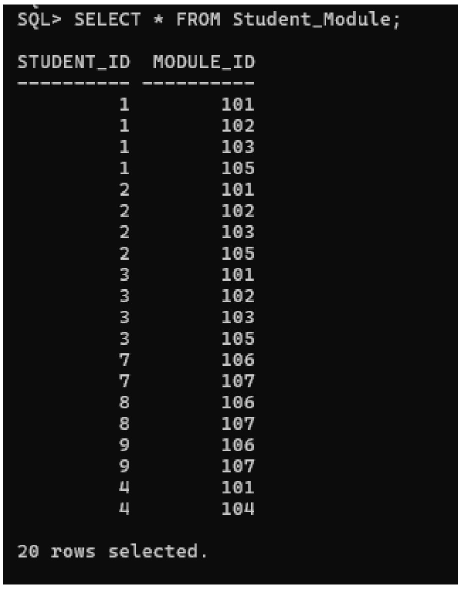
9. Putting data inside Student_Module table



```
SQL> INSERT ALL
  2         INTO Student_Module VALUES (1, 101)
  3         INTO Student_Module VALUES (1, 102)
  4         INTO Student_Module VALUES (1, 103)
  5         INTO Student_Module VALUES (1, 105)
  6         INTO Student_Module VALUES (2, 101)
  7         INTO Student_Module VALUES (2, 102)
  8         INTO Student_Module VALUES (2, 103)
  9         INTO Student_Module VALUES (2, 105)
 10         INTO Student_Module VALUES (3, 101)
 11         INTO Student_Module VALUES (3, 102)
 12         INTO Student_Module VALUES (3, 103)
 13         INTO Student_Module VALUES (3, 105)
 14         INTO Student_Module VALUES (7, 106)
 15         INTO Student_Module VALUES (7, 107)
 16         INTO Student_Module VALUES (8, 106)
 17         INTO Student_Module VALUES (8, 107)
 18         INTO Student_Module VALUES (9, 106)
 19         INTO Student_Module VALUES (9, 107)
 20         INTO Student_Module VALUES (4, 101)
 21         INTO Student_Module VALUES (4, 104)
 22         SELECT * FROM DUAL;

20 rows created.
```

Figure 34: Putting data inside Student_Module table.

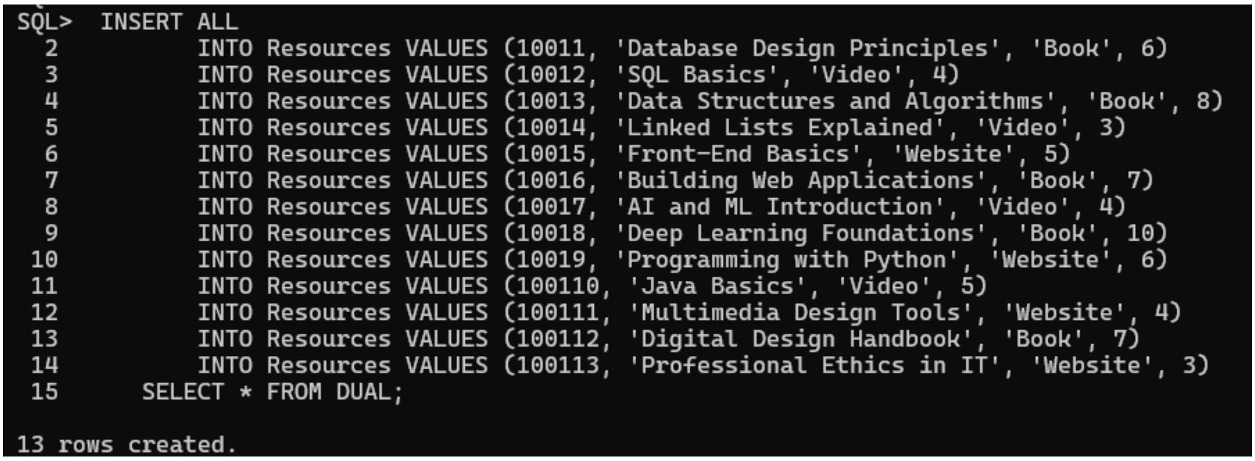
10. Using SELECT statement to retrieve data from Student_Module table



```
SQL> SELECT * FROM Student_Module;

STUDENT_ID  MODULE_ID
---------- ----------
         1        101
         1        102
         1        103
         1        105
         2        101
         2        102
         2        103
         2        105
         3        101
         3        102
         3        103
         3        105
         7        106
         7        107
         8        106
         8        107
         9        106
         9        107
         4        101
         4        104

20 rows selected.
```

Figure 35: Retrieving rows from Student_Module table.

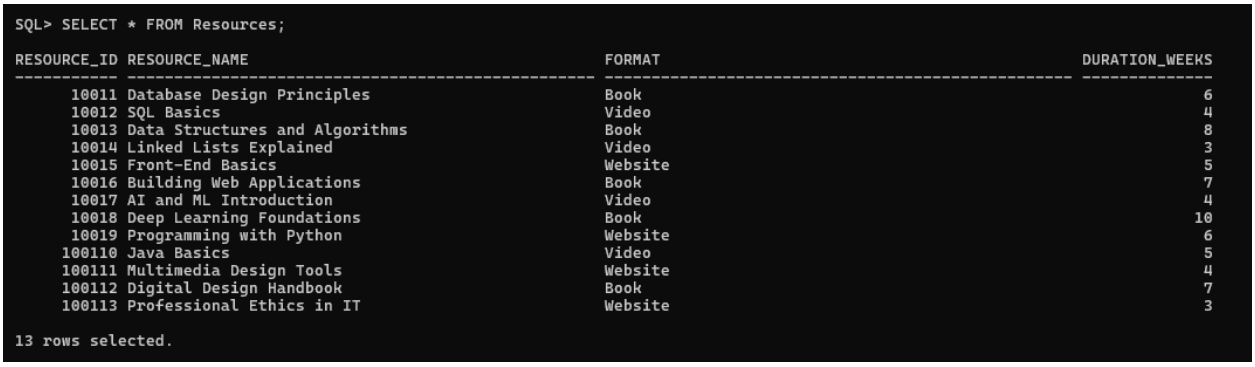
11. Putting data inside Resources table



```
SQL>  INSERT ALL
            INTO Resources VALUES (10011, 'Database Design Principles', 'Book', 6)
            INTO Resources VALUES (10012, 'SQL Basics', 'Video', 4)
            INTO Resources VALUES (10013, 'Data Structures and Algorithms', 'Book', 8)
            INTO Resources VALUES (10014, 'Linked Lists Explained', 'Video', 3)
            INTO Resources VALUES (10015, 'Front-End Basics', 'Website', 5)
            INTO Resources VALUES (10016, 'Building Web Applications', 'Book', 7)
            INTO Resources VALUES (10017, 'AI and ML Introduction', 'Video', 4)
            INTO Resources VALUES (10018, 'Deep Learning Foundations', 'Book', 10)
            INTO Resources VALUES (10019, 'Programming with Python', 'Website', 6)
            INTO Resources VALUES (100110, 'Java Basics', 'Video', 5)
            INTO Resources VALUES (100111, 'Multimedia Design Tools', 'Website', 4)
            INTO Resources VALUES (100112, 'Digital Design Handbook', 'Book', 7)
            INTO Resources VALUES (100113, 'Professional Ethics in IT', 'Website', 3)
        SELECT * FROM DUAL;

13 rows created.
```

Figure 36: Putting rows inside Resources table.

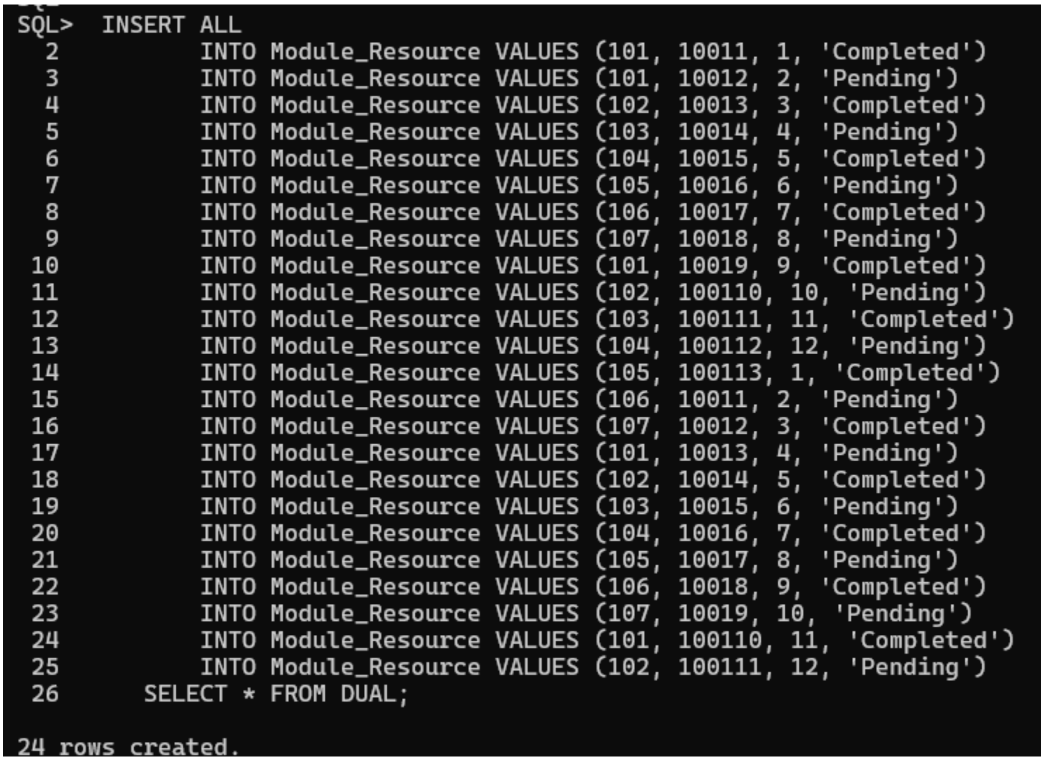
12. Using SELECT statement to retrieve data from Resources table



```
SQL> SELECT * FROM Resources;

RESOURCE_ID RESOURCE_NAME                                      FORMAT                                             DURATION_WEEKS
----------- -------------------------------------------------- -------------------------------------------------- --------------
      10011 Database Design Principles                         Book                                                            6
      10012 SQL Basics                                         Video                                                           4
      10013 Data Structures and Algorithms                     Book                                                            8
      10014 Linked Lists Explained                             Video                                                           3
      10015 Front-End Basics                                   Website                                                         5
      10016 Building Web Applications                          Book                                                            7
      10017 AI and ML Introduction                             Video                                                           4
      10018 Deep Learning Foundations                          Book                                                           10
      10019 Programming with Python                            Website                                                         6
     100110 Java Basics                                        Video                                                           5
     100111 Multimedia Design Tools                            Website                                                         4
     100112 Digital Design Handbook                            Book                                                            7
     100113 Professional Ethics in IT                          Website                                                         3

13 rows selected.
```

Figure 37: Retrieving rows from Resources table.

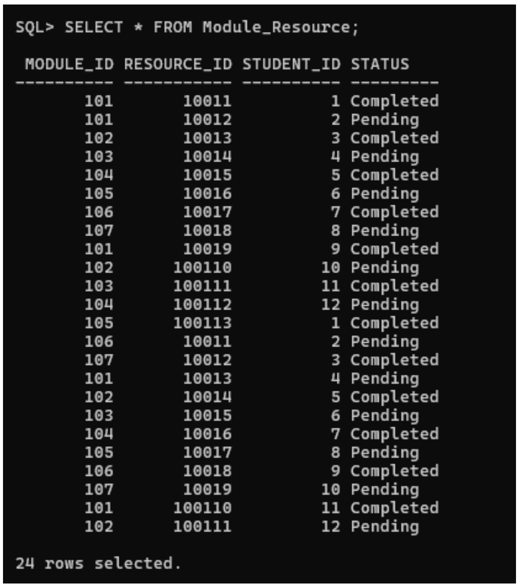
13. Putting data inside Module_Resource table



```
SQL>  INSERT ALL
            INTO Module_Resource VALUES (101, 10011, 1, 'Completed')
            INTO Module_Resource VALUES (101, 10012, 2, 'Pending')
            INTO Module_Resource VALUES (102, 10013, 3, 'Completed')
            INTO Module_Resource VALUES (103, 10014, 4, 'Pending')
            INTO Module_Resource VALUES (104, 10015, 5, 'Completed')
            INTO Module_Resource VALUES (105, 10016, 6, 'Pending')
            INTO Module_Resource VALUES (106, 10017, 7, 'Completed')
            INTO Module_Resource VALUES (107, 10018, 8, 'Pending')
            INTO Module_Resource VALUES (101, 10019, 9, 'Completed')
            INTO Module_Resource VALUES (102, 100110, 10, 'Pending')
            INTO Module_Resource VALUES (103, 100111, 11, 'Completed')
            INTO Module_Resource VALUES (104, 100112, 12, 'Pending')
            INTO Module_Resource VALUES (105, 100113, 1, 'Completed')
            INTO Module_Resource VALUES (106, 10011, 2, 'Pending')
            INTO Module_Resource VALUES (107, 10012, 3, 'Completed')
            INTO Module_Resource VALUES (101, 10013, 4, 'Pending')
            INTO Module_Resource VALUES (102, 10014, 5, 'Completed')
            INTO Module_Resource VALUES (103, 10015, 6, 'Pending')
            INTO Module_Resource VALUES (104, 10016, 7, 'Completed')
            INTO Module_Resource VALUES (105, 10017, 8, 'Pending')
            INTO Module_Resource VALUES (106, 10018, 9, 'Completed')
            INTO Module_Resource VALUES (107, 10019, 10, 'Pending')
            INTO Module_Resource VALUES (101, 100110, 11, 'Completed')
            INTO Module_Resource VALUES (102, 100111, 12, 'Pending')
        SELECT * FROM DUAL;

24 rows created.
```

Figure 38: Putting rows inside Module_Resource table.

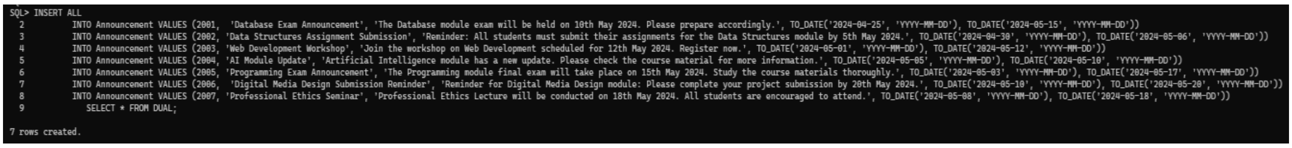
14. Using SELECT statement to retrieve data from Module_Resource



```
SQL> SELECT * FROM Module_Resource;

 MODULE_ID RESOURCE_ID STUDENT_ID STATUS
---------- ----------- ---------- ---------
       101       10011          1 Completed
       101       10012          2 Pending
       102       10013          3 Completed
       103       10014          4 Pending
       104       10015          5 Completed
       105       10016          6 Pending
       106       10017          7 Completed
       107       10018          8 Pending
       101       10019          9 Completed
       102      100110         10 Pending
       103      100111         11 Completed
       104      100112         12 Pending
       105      100113          1 Completed
       106       10011          2 Pending
       107       10012          3 Completed
       101       10013          4 Pending
       102       10014          5 Completed
       103       10015          6 Pending
       104       10016          7 Completed
       105       10017          8 Pending
       106       10018          9 Completed
       107       10019         10 Pending
       101      100110         11 Completed
       102      100111         12 Pending

24 rows selected.
```

Figure 39: Retrieving Rows from Module_Resource table.

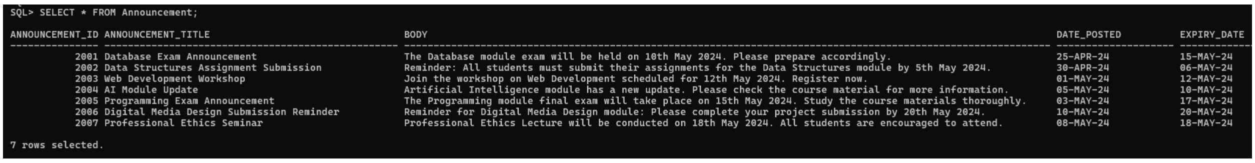
15. Putting data inside Announcement table



```
SQL> INSERT ALL
  2         INTO Announcement VALUES (2001,  'Database Exam Announcement', 'The Database module exam will be held on 10th May 2024. Please prepare accordingly.', TO_DATE('2024-04-25', 'YYYY-MM-DD'), TO_DATE('2024-05-15', 'YYYY-MM-DD'))
  3         INTO Announcement VALUES (2002, 'Data Structures Assignment Submission', 'Reminder: All students must submit their assignments for the Data Structures module by 5th May 2024.', TO_DATE('2024-04-30', 'YYYY-MM-DD'), TO_DATE('2024-05-06', 'YYYY-MM-DD'))
  4         INTO Announcement VALUES (2003, 'Web Development Workshop', 'Join the workshop on Web Development scheduled for 12th May 2024. Register now.', TO_DATE('2024-05-01', 'YYYY-MM-DD'), TO_DATE('2024-05-12', 'YYYY-MM-DD'))
  5         INTO Announcement VALUES (2004, 'AI Module Update', 'Artificial Intelligence module has a new update. Please check the course material for more information.', TO_DATE('2024-05-05', 'YYYY-MM-DD'), TO_DATE('2024-05-10', 'YYYY-MM-DD'))
  6         INTO Announcement VALUES (2005, 'Programming Exam Announcement', 'The Programming module final exam will take place on 15th May 2024. Study the course materials thoroughly.', TO_DATE('2024-05-03', 'YYYY-MM-DD'), TO_DATE('2024-05-17', 'YYYY-MM-DD'))
  7         INTO Announcement VALUES (2006,  'Digital Media Design Submission Reminder', 'Reminder for Digital Media Design module: Please complete your project submission by 20th May 2024.', TO_DATE('2024-05-10', 'YYYY-MM-DD'), TO_DATE('2024-05-20', 'YYYY-MM-DD'))
  8         INTO Announcement VALUES (2007, 'Professional Ethics Seminar', 'Professional Ethics Lecture will be conducted on 18th May 2024. All students are encouraged to attend.', TO_DATE('2024-05-08', 'YYYY-MM-DD'), TO_DATE('2024-05-18', 'YYYY-MM-DD'))
  9            SELECT * FROM DUAL;

7 rows created.
```

Figure 40: Putting Rows inside Announcement table.

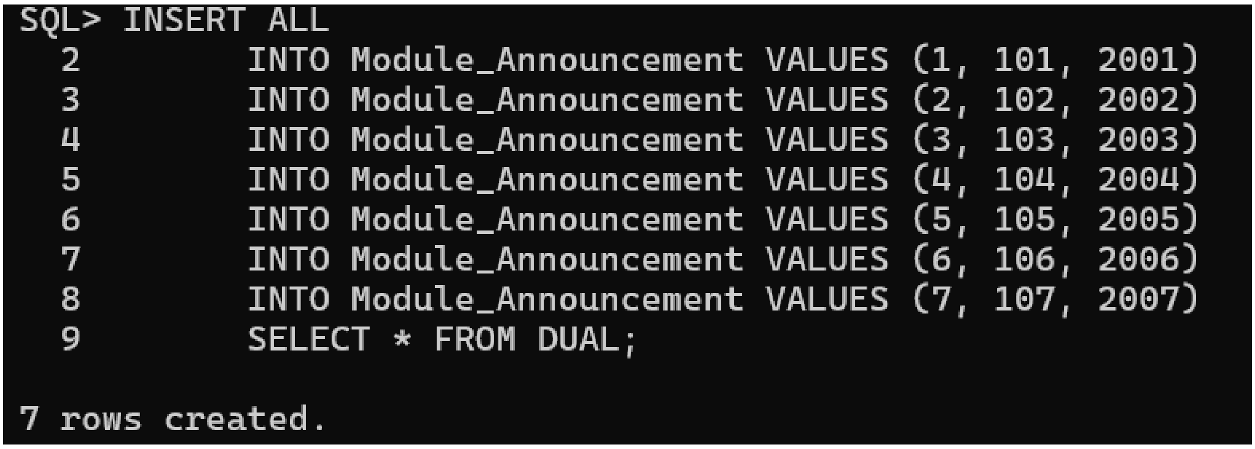
16. Using SELECT statement to retrieve data from Announcement table



```
SQL> SELECT * FROM Announcement;

ANNOUNCEMENT_ID ANNOUNCEMENT_TITLE                                  BODY                                                                                                DATE_POSTED         EXPIRY_DATE
--------------- --------------------------------------------------- --------------------------------------------------------------------------------------------------- ------------------- ------------
           2001 Database Exam Announcement                          The Database module exam will be held on 10th May 2024. Please prepare accordingly.                25-APR-24           15-MAY-24
           2002 Data Structures Assignment Submission               Reminder: All students must submit their assignments for the Data Structures module by 5th May 2024. 30-APR-24         06-MAY-24
           2003 Web Development Workshop                            Join the workshop on Web Development scheduled for 12th May 2024. Register now.                    01-MAY-24           12-MAY-24
           2004 AI Module Update                                    Artificial Intelligence module has a new update. Please check the course material for more information. 05-MAY-24     10-MAY-24
           2005 Programming Exam Announcement                       The Programming module final exam will take place on 15th May 2024. Study the course materials thoroughly. 03-MAY-24  17-MAY-24
           2006 Digital Media Design Submission Reminder            Reminder for Digital Media Design module: Please complete your project submission by 20th May 2024. 10-MAY-24          20-MAY-24
           2007 Professional Ethics Seminar                         Professional Ethics Lecture will be conducted on 18th May 2024. All students are encouraged to attend. 08-MAY-24      18-MAY-24

7 rows selected.
```

Figure 41: Retrieving rows from Announcement table.

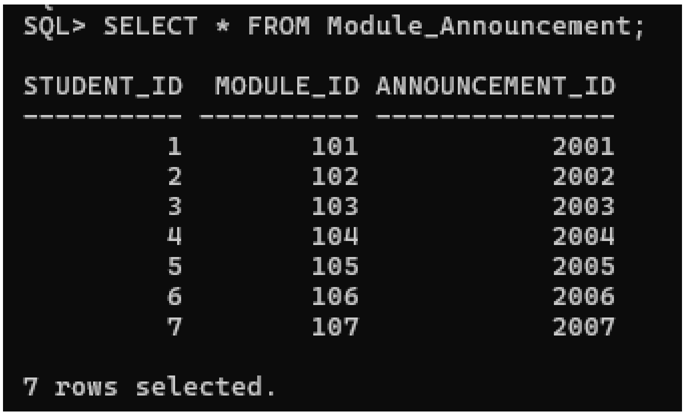
17. Putting data inside Module_Announcement table



```
SQL> INSERT ALL
          INTO Module_Announcement VALUES (1, 101, 2001)
          INTO Module_Announcement VALUES (2, 102, 2002)
          INTO Module_Announcement VALUES (3, 103, 2003)
          INTO Module_Announcement VALUES (4, 104, 2004)
          INTO Module_Announcement VALUES (5, 105, 2005)
          INTO Module_Announcement VALUES (6, 106, 2006)
          INTO Module_Announcement VALUES (7, 107, 2007)
          SELECT * FROM DUAL;

7 rows created.
```

Figure 42: Putting rows inside Module_Announcement table.

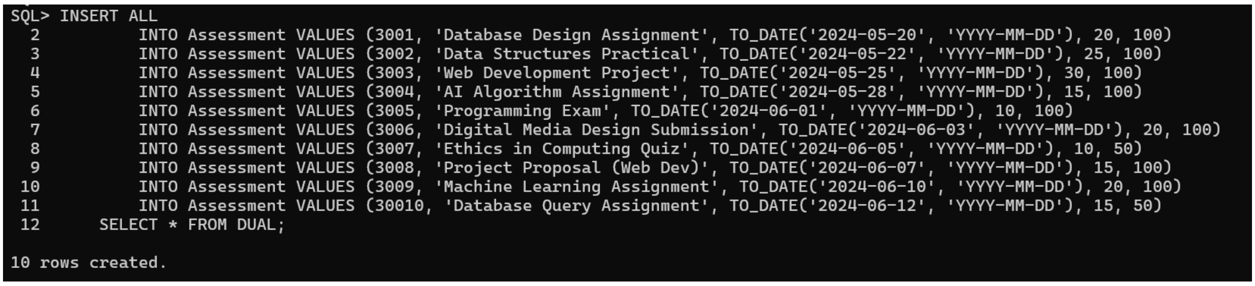
18. Using SELECT statement to retrieve data from Module_Announcement table



```
SQL> SELECT * FROM Module_Announcement;

STUDENT_ID  MODULE_ID ANNOUNCEMENT_ID
---------- ---------- ---------------
         1        101            2001
         2        102            2002
         3        103            2003
         4        104            2004
         5        105            2005
         6        106            2006
         7        107            2007

7 rows selected.
```

Figure 43: Retrieving rows from Module_Announcement.

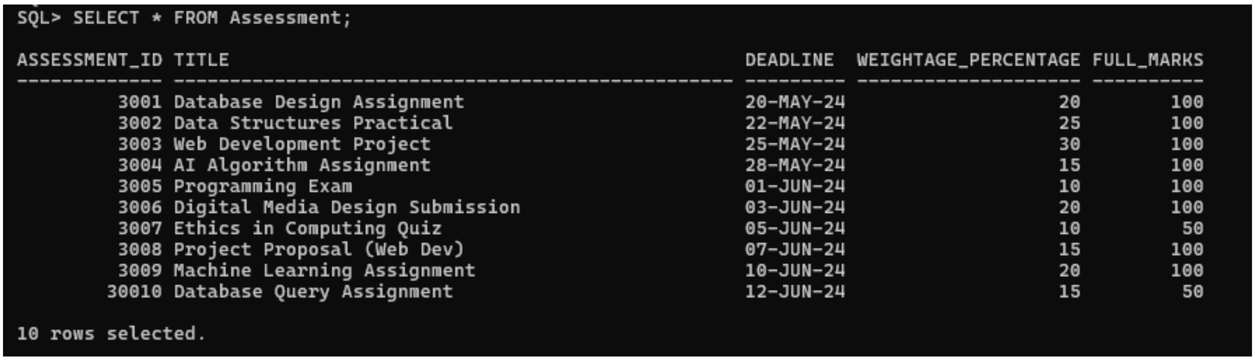
19. Putting data inside Assessment table



```
SQL> INSERT ALL
          INTO Assessment VALUES (3001, 'Database Design Assignment', TO_DATE('2024-05-20', 'YYYY-MM-DD'), 20, 100)
          INTO Assessment VALUES (3002, 'Data Structures Practical', TO_DATE('2024-05-22', 'YYYY-MM-DD'), 25, 100)
          INTO Assessment VALUES (3003, 'Web Development Project', TO_DATE('2024-05-25', 'YYYY-MM-DD'), 30, 100)
          INTO Assessment VALUES (3004, 'AI Algorithm Assignment', TO_DATE('2024-05-28', 'YYYY-MM-DD'), 15, 100)
          INTO Assessment VALUES (3005, 'Programming Exam', TO_DATE('2024-06-01', 'YYYY-MM-DD'), 10, 100)
          INTO Assessment VALUES (3006, 'Digital Media Design Submission', TO_DATE('2024-06-03', 'YYYY-MM-DD'), 20, 100)
          INTO Assessment VALUES (3007, 'Ethics in Computing Quiz', TO_DATE('2024-06-05', 'YYYY-MM-DD'), 10, 50)
          INTO Assessment VALUES (3008, 'Project Proposal (Web Dev)', TO_DATE('2024-06-07', 'YYYY-MM-DD'), 15, 100)
          INTO Assessment VALUES (3009, 'Machine Learning Assignment', TO_DATE('2024-06-10', 'YYYY-MM-DD'), 20, 100)
          INTO Assessment VALUES (30010, 'Database Query Assignment', TO_DATE('2024-06-12', 'YYYY-MM-DD'), 15, 50)
      SELECT * FROM DUAL;

10 rows created.
```

Figure 44: Putting rows inside Assessment table.

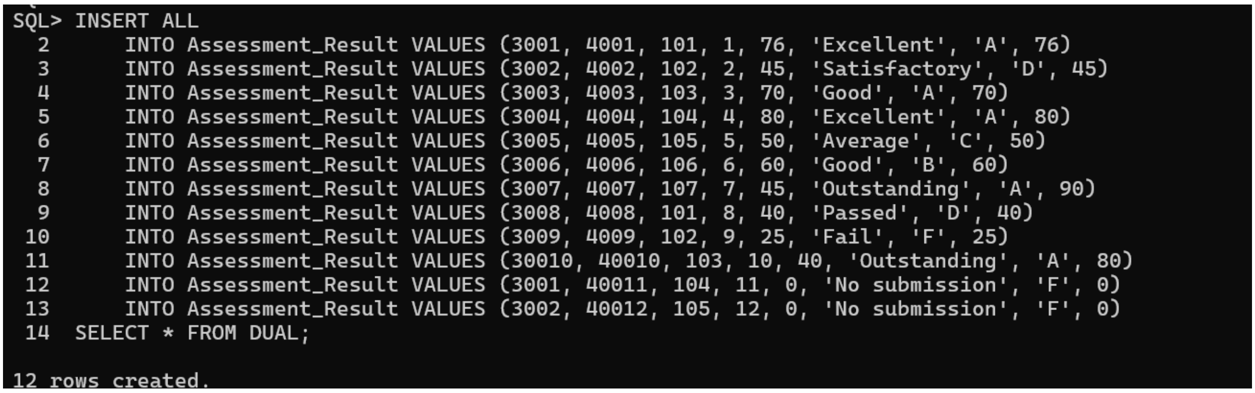
20. Using SELECT Statement to retrieve data from Assessment table



```
SQL> SELECT * FROM Assessment;

ASSESSMENT_ID TITLE                                              DEADLINE  WEIGHTAGE_PERCENTAGE FULL_MARKS
------------- -------------------------------------------------- --------- -------------------- ----------
         3001 Database Design Assignment                         20-MAY-24                   20        100
         3002 Data Structures Practical                          22-MAY-24                   25        100
         3003 Web Development Project                            25-MAY-24                   30        100
         3004 AI Algorithm Assignment                            28-MAY-24                   15        100
         3005 Programming Exam                                   01-JUN-24                   10        100
         3006 Digital Media Design Submission                    03-JUN-24                   20        100
         3007 Ethics in Computing Quiz                           05-JUN-24                   10         50
         3008 Project Proposal (Web Dev)                         07-JUN-24                   15        100
         3009 Machine Learning Assignment                        10-JUN-24                   20        100
        30010 Database Query Assignment                          12-JUN-24                   15         50

10 rows selected.
```

Figure 45: Retrieving rows from Assessment table.

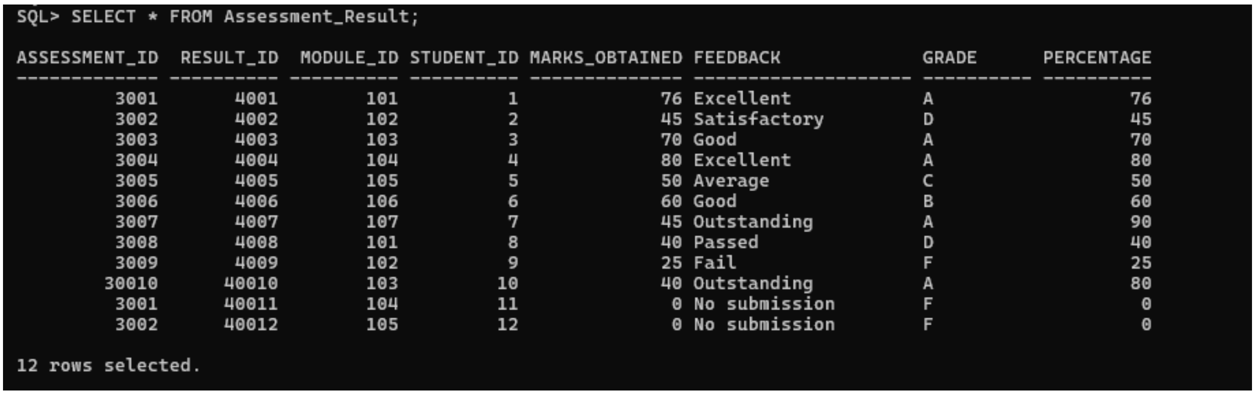
21. Putting data into Assessment_Result table



```
SQL> INSERT ALL
  2      INTO Assessment_Result VALUES (3001, 4001, 101, 1, 76, 'Excellent', 'A', 76)
  3      INTO Assessment_Result VALUES (3002, 4002, 102, 2, 45, 'Satisfactory', 'D', 45)
  4      INTO Assessment_Result VALUES (3003, 4003, 103, 3, 70, 'Good', 'A', 70)
  5      INTO Assessment_Result VALUES (3004, 4004, 104, 4, 80, 'Excellent', 'A', 80)
  6      INTO Assessment_Result VALUES (3005, 4005, 105, 5, 50, 'Average', 'C', 50)
  7      INTO Assessment_Result VALUES (3006, 4006, 106, 6, 60, 'Good', 'B', 60)
  8      INTO Assessment_Result VALUES (3007, 4007, 107, 7, 45, 'Outstanding', 'A', 90)
  9      INTO Assessment_Result VALUES (3008, 4008, 101, 8, 40, 'Passed', 'D', 40)
 10      INTO Assessment_Result VALUES (3009, 4009, 102, 9, 25, 'Fail', 'F', 25)
 11      INTO Assessment_Result VALUES (30010, 40010, 103, 10, 40, 'Outstanding', 'A', 80)
 12      INTO Assessment_Result VALUES (3001, 40011, 104, 11, 0, 'No submission', 'F', 0)
 13      INTO Assessment_Result VALUES (3002, 40012, 105, 12, 0, 'No submission', 'F', 0)
 14  SELECT * FROM DUAL;

12 rows created.
```

Figure 46: Putting rows inside Assessment_Result table.

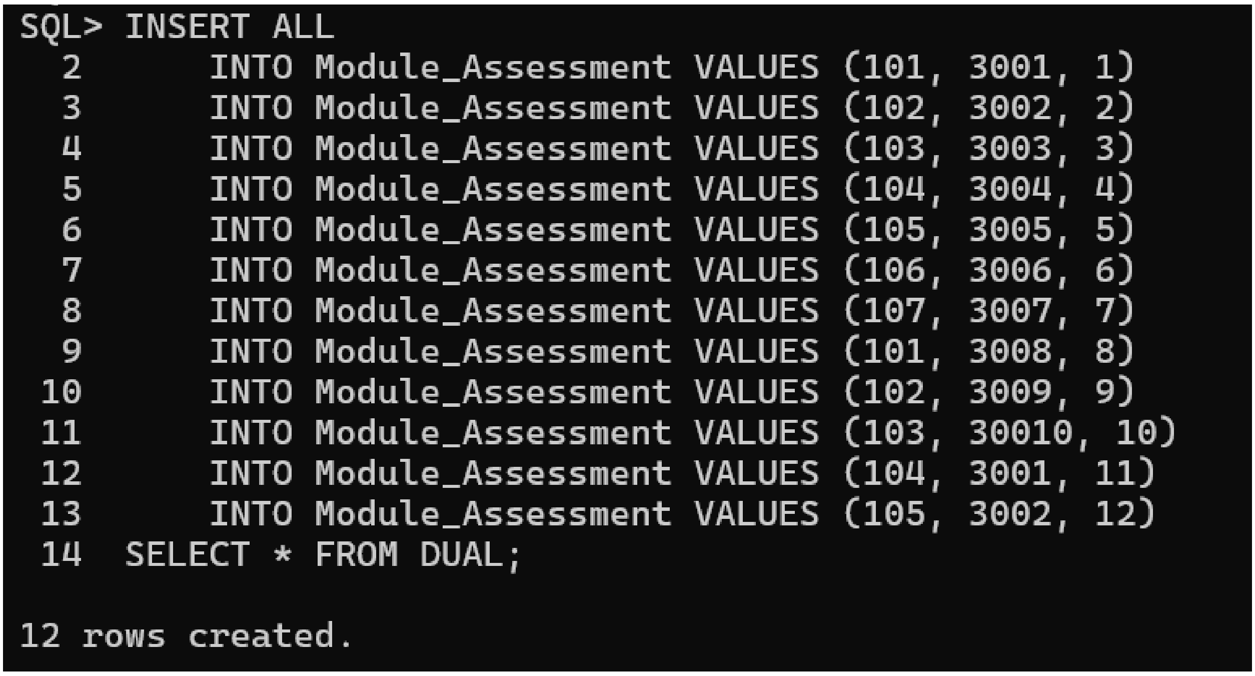
22. Using SELECT Statement to retrieve data from Assessment_Result table



```
SQL> SELECT * FROM Assessment_Result;

ASSESSMENT_ID  RESULT_ID  MODULE_ID STUDENT_ID MARKS_OBTAINED FEEDBACK             GRADE      PERCENTAGE
------------- ---------- ---------- ---------- -------------- -------------------- ---------- ----------
         3001       4001        101          1             76 Excellent            A                  76
         3002       4002        102          2             45 Satisfactory         D                  45
         3003       4003        103          3             70 Good                 A                  70
         3004       4004        104          4             80 Excellent            A                  80
         3005       4005        105          5             50 Average              C                  50
         3006       4006        106          6             60 Good                 B                  60
         3007       4007        107          7             45 Outstanding          A                  90
         3008       4008        101          8             40 Passed               D                  40
         3009       4009        102          9             25 Fail                 F                  25
        30010      40010        103         10             40 Outstanding          A                  80
         3001      40011        104         11              0 No submission        F                   0
         3002      40012        105         12              0 No submission        F                   0

12 rows selected.
```

Figure 47: Retrieving rows from Assessment_Result table.

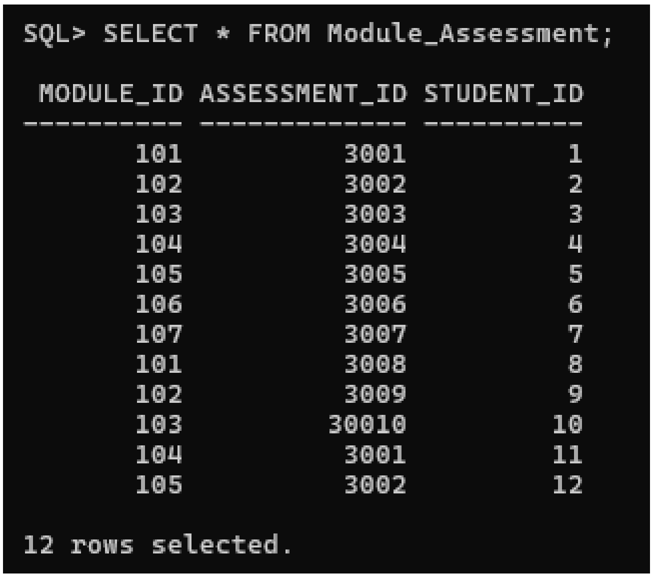
23. Putting data into Module_Assessment table



```
SQL> INSERT ALL
        INTO Module_Assessment VALUES (101, 3001, 1)
        INTO Module_Assessment VALUES (102, 3002, 2)
        INTO Module_Assessment VALUES (103, 3003, 3)
        INTO Module_Assessment VALUES (104, 3004, 4)
        INTO Module_Assessment VALUES (105, 3005, 5)
        INTO Module_Assessment VALUES (106, 3006, 6)
        INTO Module_Assessment VALUES (107, 3007, 7)
        INTO Module_Assessment VALUES (101, 3008, 8)
        INTO Module_Assessment VALUES (102, 3009, 9)
        INTO Module_Assessment VALUES (103, 30010, 10)
        INTO Module_Assessment VALUES (104, 3001, 11)
        INTO Module_Assessment VALUES (105, 3002, 12)
     SELECT * FROM DUAL;

12 rows created.
```

Figure 48: Putting rows inside Module_Assessment table.

24. Using SELECT Statement to retrieve data from Module_Assessment table



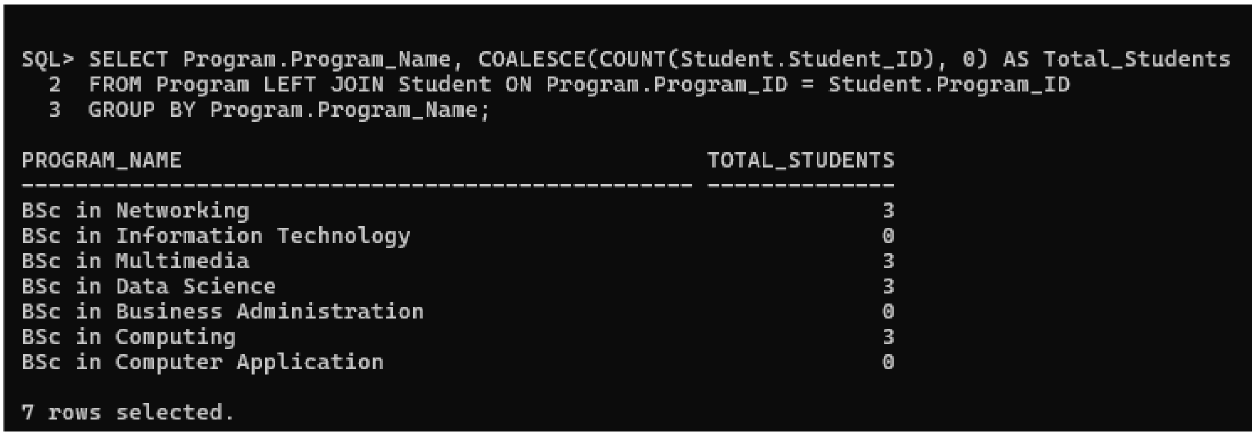
```
SQL> SELECT * FROM Module_Assessment;

 MODULE_ID ASSESSMENT_ID STUDENT_ID
---------- ------------- ----------
       101          3001          1
       102          3002          2
       103          3003          3
       104          3004          4
       105          3005          5
       106          3006          6
       107          3007          7
       101          3008          8
       102          3009          9
       103         30010         10
       104          3001         11
       105          3002         12

12 rows selected.
```

Figure 49: Retrieving rows from Module_Assessment table.

5.4. Solving operations.

5.4.1. Information Query

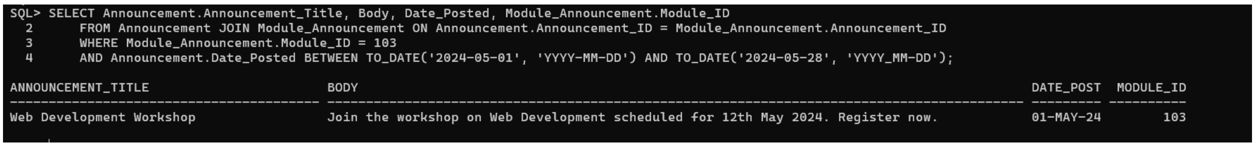
1. List the programs that are available in the college and the total number of students enrolled in each.



```
SQL> SELECT Program.Program_Name, COALESCE(COUNT(Student.Student_ID), 0) AS Total_Students
  2  FROM Program LEFT JOIN Student ON Program.Program_ID = Student.Program_ID
  3  GROUP BY Program.Program_Name;

PROGRAM_NAME                                       TOTAL_STUDENTS
-------------------------------------------------- --------------
BSc in Networking                                               3
BSc in Information Technology                                   0
BSc in Multimedia                                               3
BSc in Data Science                                             3
BSc in Business Administration                                  0
BSc in Computing                                                3
BSc in Computer Application                                     0

7 rows selected.
```

Figure 50: Programs provided by the college and total number of students admitted.

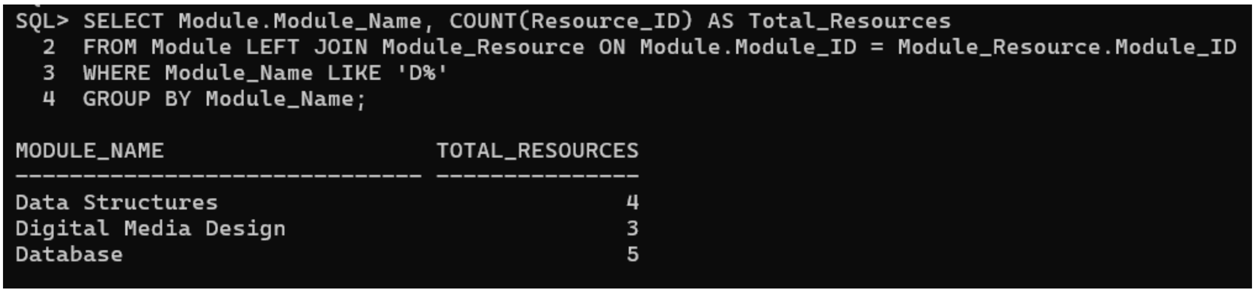
2. **List all the announcements made for a particular module starting from 1st May 2024 to 28th May 2024.**



```
SQL> SELECT Announcement.Announcement_Title, Body, Date_Posted, Module_Announcement.Module_ID
  2      FROM Announcement JOIN Module_Announcement ON Announcement.Announcement_ID = Module_Announcement.Announcement_ID
  3      WHERE Module_Announcement.Module_ID = 103
  4      AND Announcement.Date_Posted BETWEEN TO_DATE('2024-05-01', 'YYYY-MM-DD') AND TO_DATE('2024-05-28', 'YYYY_MM-DD');

ANNOUNCEMENT_TITLE                       BODY                                                                                  DATE_POST  MODULE_ID
---------------------------------------- ------------------------------------------------------------------------------------- --------- ----------
Web Development Workshop                 Join the workshop on Web Development scheduled for 12th May 2024. Register now.       01-MAY-24        103
```

Figure 51: Announcements made for Module_ID 103 between May 1st and May 28th.

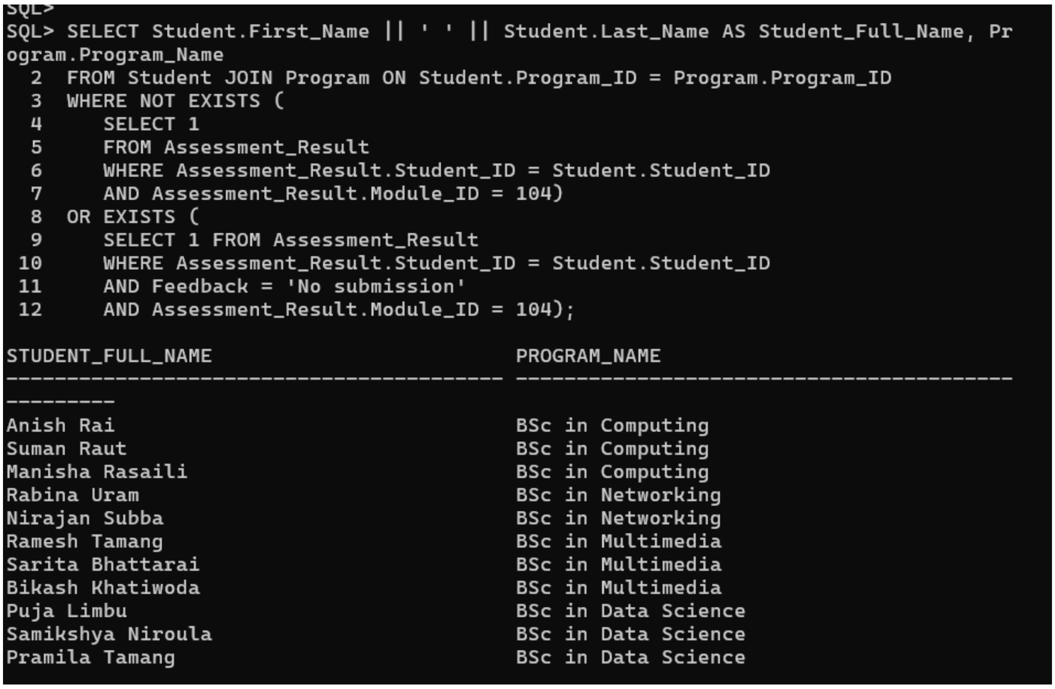
3. **List the names of all modules that begin with the letter 'D', along with the total number of resources uploaded for those modules.**



```
SQL> SELECT Module.Module_Name, COUNT(Resource_ID) AS Total_Resources
  2  FROM Module LEFT JOIN Module_Resource ON Module.Module_ID = Module_Resource.Module_ID
  3  WHERE Module_Name LIKE 'D%'
  4  GROUP BY Module_Name;

MODULE_NAME                    TOTAL_RESOURCES
------------------------------ ---------------
Data Structures                              4
Digital Media Design                         3
Database                                     5
```

Figure 52: List of modules starting with letter 'D' with their respective resource number..

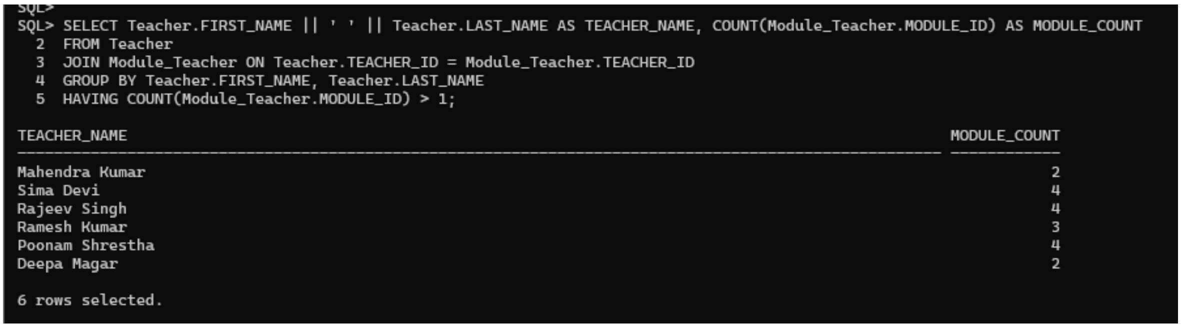
4. **List the names of all students along with their enrolled program who have not submitted any assessments for a particular module.**



```
SQL>
SQL> SELECT Student.First_Name || ' ' || Student.Last_Name AS Student_Full_Name, Pr
ogram.Program_Name
FROM Student JOIN Program ON Student.Program_ID = Program.Program_ID
WHERE NOT EXISTS (
   SELECT 1
   FROM Assessment_Result
   WHERE Assessment_Result.Student_ID = Student.Student_ID
   AND Assessment_Result.Module_ID = 104)
OR EXISTS (
   SELECT 1 FROM Assessment_Result
   WHERE Assessment_Result.Student_ID = Student.Student_ID
   AND Feedback = 'No submission'
   AND Assessment_Result.Module_ID = 104);

STUDENT_FULL_NAME                         PROGRAM_NAME
----------------------------------------- -----------------------------------------
---------
Anish Rai                                 BSc in Computing
Suman Raut                                BSc in Computing
Manisha Rasaili                           BSc in Computing
Rabina Uram                               BSc in Networking
Nirajan Subba                             BSc in Networking
Ramesh Tamang                             BSc in Multimedia
Sarita Bhattarai                          BSc in Multimedia
Bikash Khatiwoda                          BSc in Multimedia
Puja Limbu                                BSc in Data Science
Samikshya Niroula                         BSc in Data Science
Pramila Tamang                            BSc in Data Science
```

Figure 53: Names of all students who have not submitted assignments.

5. **List all the teachers who teach more than one module.**

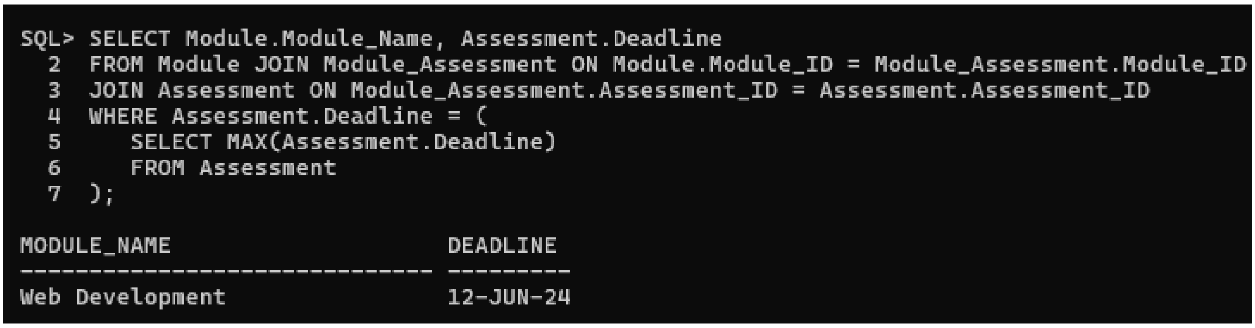


```
SQL>
SQL> SELECT Teacher.FIRST_NAME || ' ' || Teacher.LAST_NAME AS TEACHER_NAME, COUNT(Module_Teacher.MODULE_ID) AS MODULE_COUNT
FROM Teacher
JOIN Module_Teacher ON Teacher.TEACHER_ID = Module_Teacher.TEACHER_ID
GROUP BY Teacher.FIRST_NAME, Teacher.LAST_NAME
HAVING COUNT(Module_Teacher.MODULE_ID) > 1;

TEACHER_NAME                                                                                      MODULE_COUNT
------------------------------------------------------------------------------------------------- ------------
Mahendra Kumar                                                                                               2
Sima Devi                                                                                                    4
Rajeev Singh                                                                                                 4
Ramesh Kumar                                                                                                 3
Poonam Shrestha                                                                                              4
Deepa Magar                                                                                                  2

6 rows selected.
```

Figure 54: List of teachers teaching multiple modules.

5.4.2. Transaction Query.

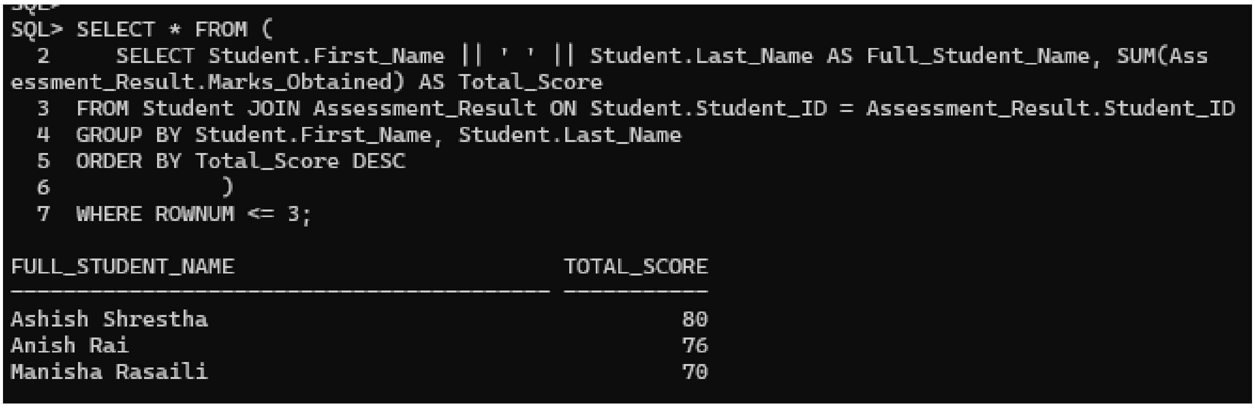
1. **Identify the module that has the latest assessment deadline.**



```
SQL> SELECT Module.Module_Name, Assessment.Deadline
     FROM Module JOIN Module_Assessment ON Module.Module_ID = Module_Assessment.Module_ID
     JOIN Assessment ON Module_Assessment.Assessment_ID = Assessment.Assessment_ID
     WHERE Assessment.Deadline = (
        SELECT MAX(Assessment.Deadline)
        FROM Assessment
     );

MODULE_NAME                    DEADLINE
------------------------------ ---------
Web Development                12-JUN-24
```

Figure 55: Module with the latest deadline.

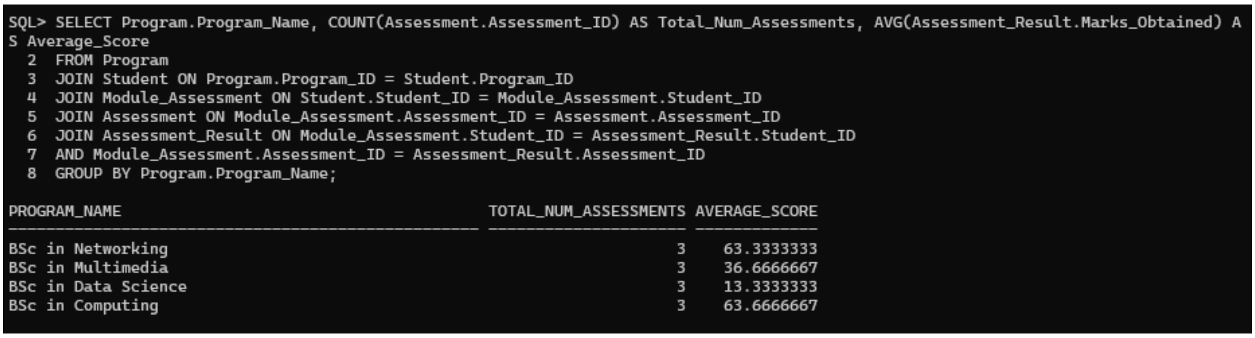
2. **Find the top three students who have the highest total score across all modules.**



```
SQL> SELECT * FROM (
        SELECT Student.First_Name || ' ' || Student.Last_Name AS Full_Student_Name, SUM(Ass
essment_Result.Marks_Obtained) AS Total_Score
     FROM Student JOIN Assessment_Result ON Student.Student_ID = Assessment_Result.Student_ID
     GROUP BY Student.First_Name, Student.Last_Name
     ORDER BY Total_Score DESC
               )
     WHERE ROWNUM <= 3;

FULL_STUDENT_NAME                         TOTAL_SCORE
----------------------------------------- -----------
Ashish Shrestha                                    80
Anish Rai                                          76
Manisha Rasaili                                    70
```

Figure 56: Top 3 students with the highest marks

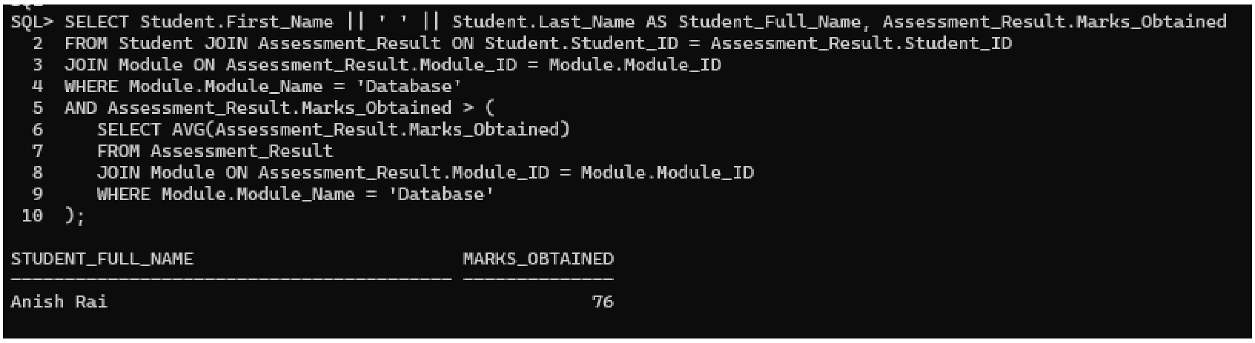
3. **Find the total number of assessments for each program and the average score across all assessments in those programs.**



```
SQL> SELECT Program.Program_Name, COUNT(Assessment.Assessment_ID) AS Total_Num_Assessments, AVG(Assessment_Result.Marks_Obtained) A
S Average_Score
     FROM Program
     JOIN Student ON Program.Program_ID = Student.Program_ID
     JOIN Module_Assessment ON Student.Student_ID = Module_Assessment.Student_ID
     JOIN Assessment ON Module_Assessment.Assessment_ID = Assessment.Assessment_ID
     JOIN Assessment_Result ON Module_Assessment.Student_ID = Assessment_Result.Student_ID
     AND Module_Assessment.Assessment_ID = Assessment_Result.Assessment_ID
     GROUP BY Program.Program_Name;

PROGRAM_NAME                                       TOTAL_NUM_ASSESSMENTS AVERAGE_SCORE
-------------------------------------------------- --------------------- -------------
BSc in Networking                                                      3    63.3333333
BSc in Multimedia                                                      3    36.6666667
BSc in Data Science                                                    3    13.3333333
BSc in Computing                                                       3    63.6666667
```

Figure 57: Total number of assessments and their average score for each program.

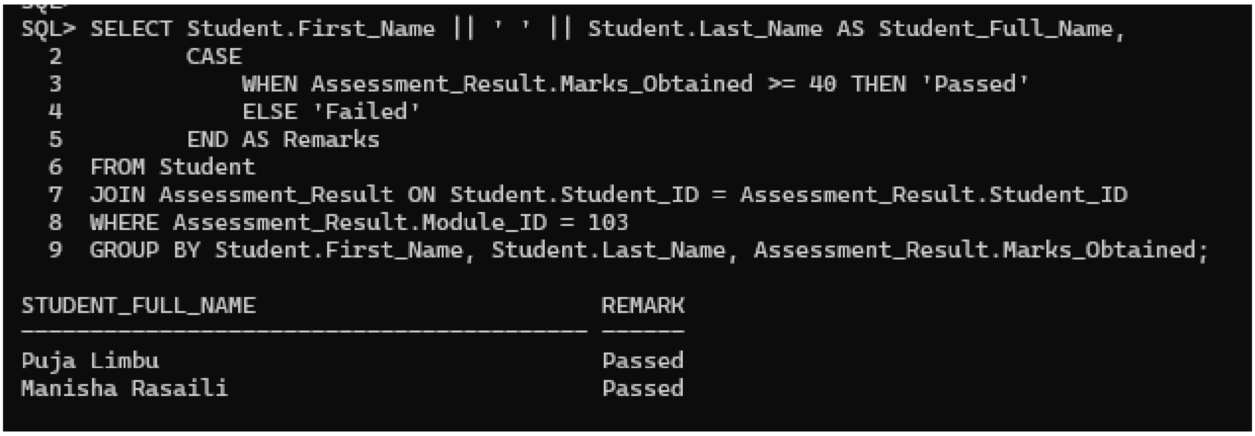
4. **List the students who have scored above the average score in the 'Databases' module.**



```
SQL> SELECT Student.First_Name || ' ' || Student.Last_Name AS Student_Full_Name, Assessment_Result.Marks_Obtained
     FROM Student JOIN Assessment_Result ON Student.Student_ID = Assessment_Result.Student_ID
     JOIN Module ON Assessment_Result.Module_ID = Module.Module_ID
     WHERE Module.Module_Name = 'Database'
     AND Assessment_Result.Marks_Obtained > (
        SELECT AVG(Assessment_Result.Marks_Obtained)
        FROM Assessment_Result
        JOIN Module ON Assessment_Result.Module_ID = Module.Module_ID
        WHERE Module.Module_Name = 'Database'
     );

STUDENT_FULL_NAME                         MARKS_OBTAINED
----------------------------------------- --------------
Anish Rai                                             76
```

Figure 58: list of students who scored higher than average marks in Database.

5. **Display whether a student has passed or failed as remarks as per their total aggregate marks obtained in a particular module.**



```
SQL> SELECT Student.First_Name || ' ' || Student.Last_Name AS Student_Full_Name,
            CASE
                WHEN Assessment_Result.Marks_Obtained >= 40 THEN 'Passed'
                ELSE 'Failed'
            END AS Remarks
     FROM Student
     JOIN Assessment_Result ON Student.Student_ID = Assessment_Result.Student_ID
     WHERE Assessment_Result.Module_ID = 103
     GROUP BY Student.First_Name, Student.Last_Name, Assessment_Result.Marks_Obtained;

STUDENT_FULL_NAME                         REMARK
----------------------------------------- ------
Puja Limbu                                Passed
Manisha Rasaili                           Passed
```

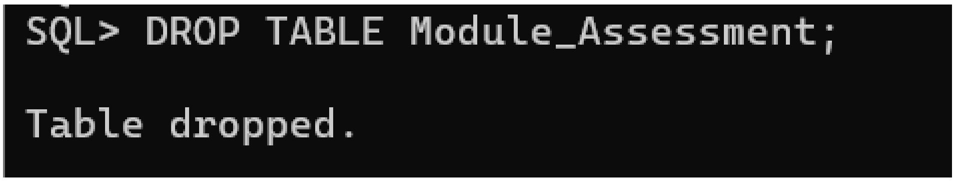
Figure 59: List of students who passed or failed in module 103.

6. Drop Query and Database Dump File Creation

6.1. Drop Query

Tables in the database need to be dropped in order to successfully delete them. As, various relationships were formed between several relations. They must be deleted in an orderly manner. First, the junction/bridge tables should be removed then after that tables carrying foreign keys should be removed and then only tables without foreign keys should be removed. If not, several issues can be faced due to the relationships between the tables.

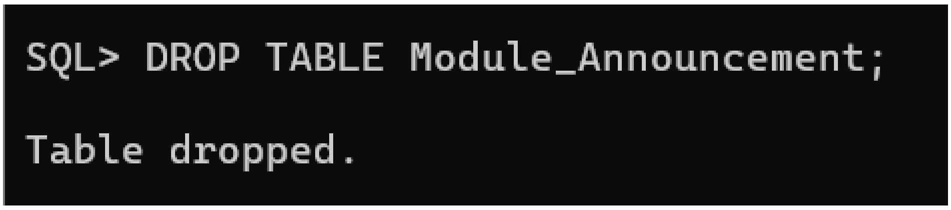
1. Dropping Module_Assessment table



```
SQL> DROP TABLE Module_Assessment;

Table dropped.
```

Figure 60: Dropping Module_Assessment;

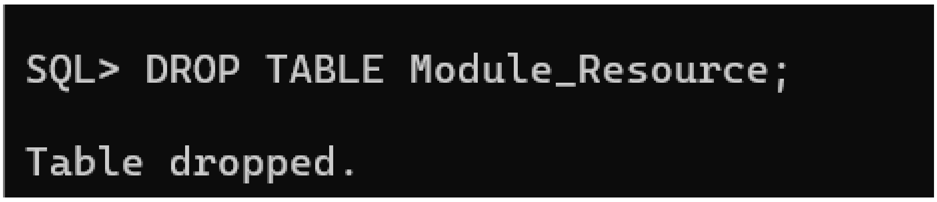
2. Dropping Module_Announcement table



```
SQL> DROP TABLE Module_Announcement;

Table dropped.
```

Figure 61: Dropping Module_Announcement

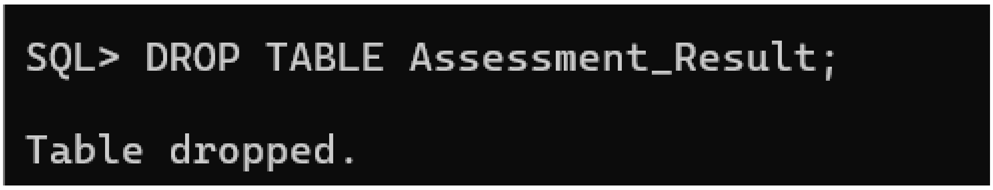
3. Dropping Module_Resource table



```
SQL> DROP TABLE Module_Resource;

Table dropped.
```

Figure 62: : Dropping Module_Resource.

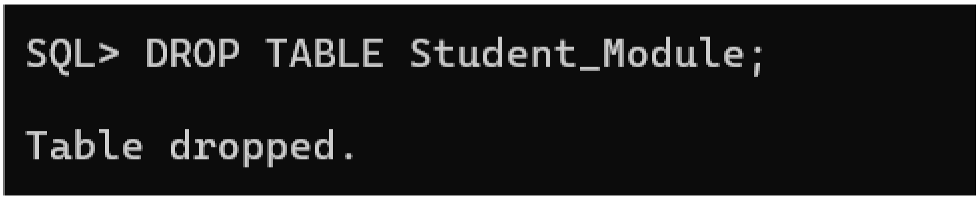
4. Dropping Assessment_Result table



```
SQL> DROP TABLE Assessment_Result;

Table dropped.
```

Figure 63: Dropping Assessment_Result.

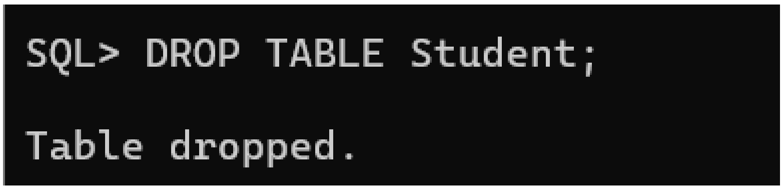
5. Dropping Student_Module table



```
SQL> DROP TABLE Student_Module;

Table dropped.
```

Figure 64: Dropping Student_Module.

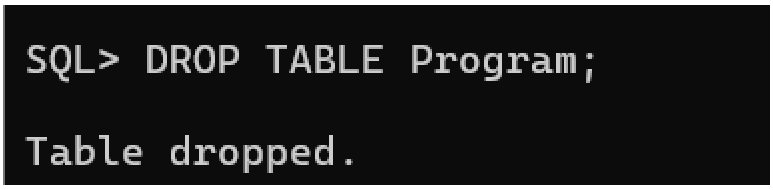
6. Dropping Student table



```
SQL> DROP TABLE Student;

Table dropped.
```

Figure 65: Dropping Student.

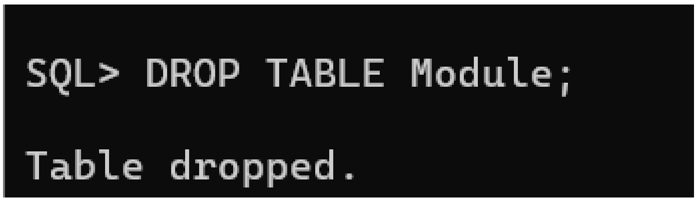
7. Dropping Program table



```
SQL> DROP TABLE Program;

Table dropped.
```

Figure 66: Dropping Program.

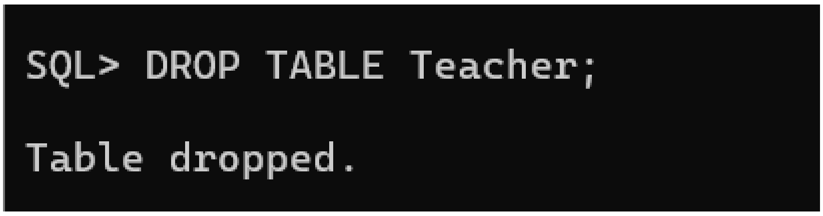
8. Dropping Module table



```
SQL> DROP TABLE Module;

Table dropped.
```

Figure 67: Dropping Module.

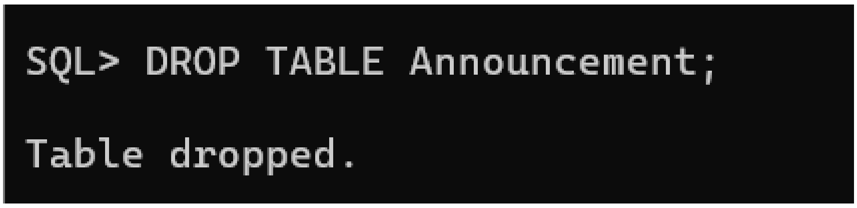
9. Dropping Teacher table



```
SQL> DROP TABLE Teacher;

Table dropped.
```

Figure 68: Dropping Teacher.

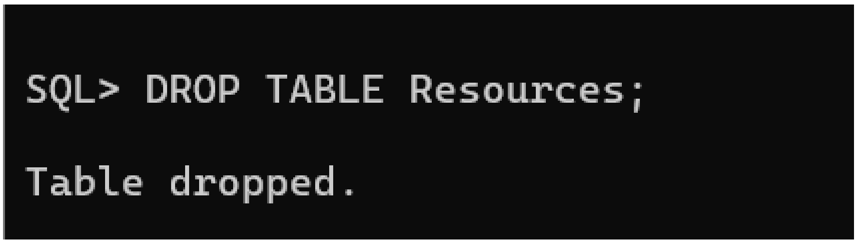
10. Dropping Announcement table



```
SQL> DROP TABLE Announcement;

Table dropped.
```

Figure 69: Dropping Announcement.

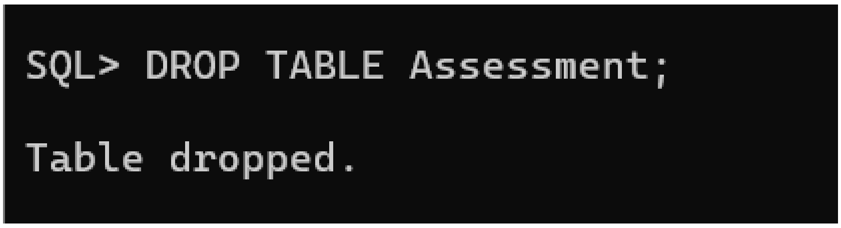
11. Dropping Resources table



```
SQL> DROP TABLE Resources;

Table dropped.
```

Figure 70: Dropping Resources.

12. Dropping Assessment table



```
SQL> DROP TABLE Assessment;

Table dropped.
```

Figure 71: Dropping Assessment.

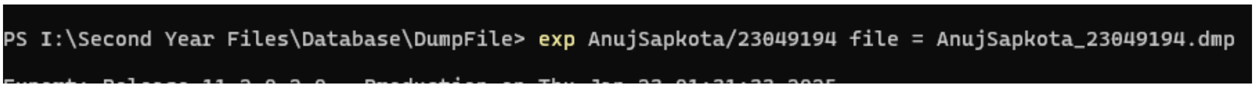
6.2. Dump File Creation

Dump File in a database is like a snapshot or record of the structure of the relations and data. It is the backup of the entire database on a logical level. It is used to restore the database and can be used to export or import data between systems. The extension for dump file is *.dmp.* (Handy Backup, 2025)

The syntax in oracle to create a dump file is:

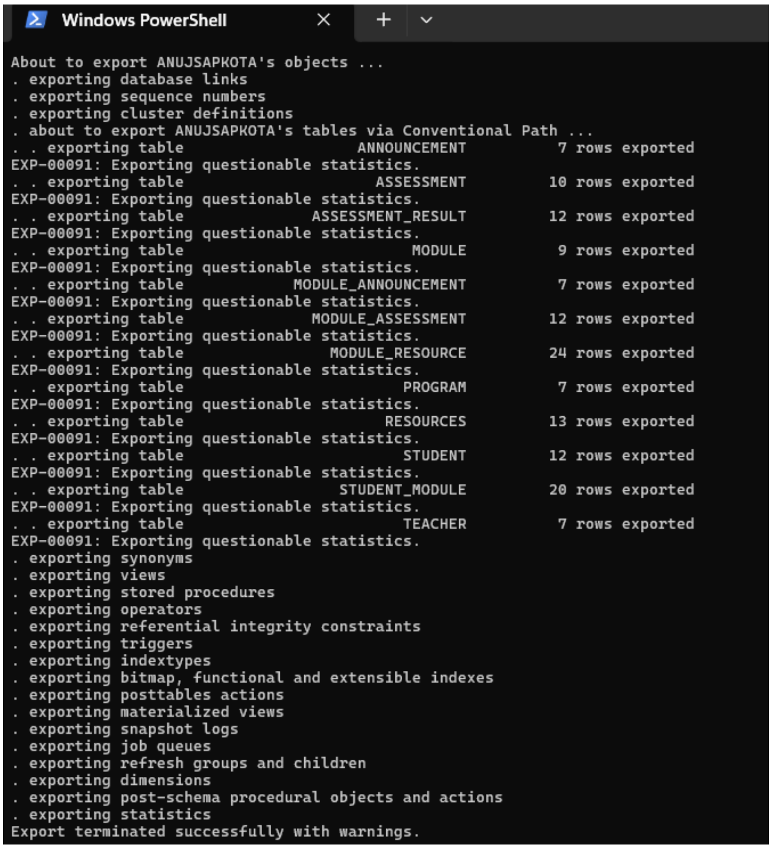
exp *databaseUsername/password* file = *dumpfilename.*dmp

To create the dump file, open the terminal/PowerShell and move to the location where the dump file is to be created and use the code. In the code, in place of *databaseUsername/password* , we write the username/password of database's User . Similarly, in place of *dumpfilename,* we write the name of the dump file.



```
PS I:\Second Year Files\Database\DumpFile> exp AnujSapkota/23049194 file = AnujSapkota_23049194.dmp
```

Table 36: using command to make the dump file



```
About to export ANUJSAPKOTA's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export ANUJSAPKOTA's tables via Conventional Path ...
. . exporting table                   ANNOUNCEMENT          7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                     ASSESSMENT         10 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table              ASSESSMENT_RESULT         12 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                         MODULE          9 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table            MODULE_ANNOUNCEMENT          7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table              MODULE_ASSESSMENT         12 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                MODULE_RESOURCE         24 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                        PROGRAM          7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                      RESOURCES         13 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                        STUDENT         12 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                 STUDENT_MODULE         20 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table                        TEACHER          7 rows exported
EXP-00091: Exporting questionable statistics.
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting operators
. exporting referential integrity constraints
. exporting triggers
. exporting indextypes
. exporting bitmap, functional and extensible indexes
. exporting posttables actions
. exporting materialized views
. exporting snapshot logs
. exporting job queues
. exporting refresh groups and children
. exporting dimensions
. exporting post-schema procedural objects and actions
. exporting statistics
Export terminated successfully with warnings.
```

Table 37: Dump file created successfully.

7. Critical Evaluation

7.1. Critical Evaluation of module, its usage and relation with other subjects

This database module greatly helped in gaining learning about database concepts from basic to intermediate level. The way it is structured has simplified and boosted the learning process. It provides practical learning and hands-on experience which is essential for performing operation on real world scenarios.

The implementation of databases as a module helps both in academic as well as in an occupational and professional scenario. It provides a foundational knowledge for organizing and managing data in a structured format like making tables, declaring columns and their datatypes and constraints. This module also covered core database concepts like normalization and querying. This module helps create scalable systems which is one of the most required in today's real-world scenarios.

The database module works well in relation to other subjects. Programming languages, web development, Machine Learning are few of the subject's databases have put its hands on. This module is also corelated with our Information Systems module which we studied in our first year of this course.

Therefore, this module has been a essential aspect of this course which helped develop various skills and knowledge which are highly required in almost every IT field in today's world.

7.2. Critical Assessment of coursework

In this coursework, a consistent database system was to be developed for an organization to store staff data as well as college resources. The database system has to be consistent; data integrity has to be maintained, and redundancy has to be removed as much as possible. To maintain this, normalization was performed up to 3NF. This helped remove both partial and transitive dependencies.

The normalization part was the most puzzling thought the whole coursework. The normalization was performed in a stepwise manner: UNF, 1NF, 2NF and 3NF. Initial and final ERD were developed for better visualizing the database

This overall coursework has been more than just a form of assessment. This whole journey was like a storm of emotion. . Various thoughtful decisions were carried out during this coursework. There were obstacles and various setbacks in the way but nonetheless they were essential in building not only the professional skills but the whole mindset required to solve problems and has a steady mindset throught the whole project development. This assessment has greatly inspired me to seek more information on database relating modules and I look forward to gaining more knowledge and solving any obstacles coming in my way.

8. References

atlan, 2024. *What is a data dictionary.* [Online]
Available at: https://atlan.com/what-is-a-data-dictionary/
[Accessed 21 01 2025].

Geeks For Geeks, 2025. *First Normal Form (1NF).* [Online]
Available at: https://www.geeksforgeeks.org/first-normal-form-1nf/
[Accessed 21 01 2025].

GeeksForGeeks, 2025. *Second Normal Form (2NF).* [Online]
Available at: https://www.geeksforgeeks.org/second-normal-form-2nf/
[Accessed 21 01 2025].

GeeksForGeeks, 2025. *Third Normal Form (3NF).* [Online]
Available at: https://www.geeksforgeeks.org/third-normal-form-3nf/
[Accessed 21 01 2025].

Handy Backup, 2025. *Database Dump. What is DB dump? How do you create a database dump.* [Online]
Available at: https://www.handybackup.net/backup_terms/database-dump.shtml
[Accessed 23 01 2025].

Hanna, K. T., 2025. *entity relationship diagram (ERD).* [Online]
Available at: https://www.techtarget.com/searchdatamanagement/definition/entity-

relationship-diagram-ERD
[Accessed 23 01 2025].

Ian, 2017. *Database Guide.* [Online]
Available at: https://database.guide/what-is-normalization/
[Accessed 21 01 2025].

Thakur, S., 2016. *Database Normalization: Explain 1NF, 2NF, 3NF, BCNF With Examples.* [Online]
Available at: https://whatisdbms.com/database-normalization-explain-1nf-2nf-3nf-bcnf-with-examples/
[Accessed 21 01 2025].

Wikipedia, 2024. *Unnormalized form.* [Online]
Available at: https://en.wikipedia.org/wiki/Unnormalized_form
[Accessed 23 01 2025].