



CC5051NI Databases

100% Individual Coursework

Autumn 2024

Credit: 15 Semester Long Module

Student Name: ROHIT JOSHI

London Met ID: 23049330

Assignment Submission Date: 01/21/2025

Word Count: 4159

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

23049330 ROHIT JOSHI2.docx

 Islington College,Nepal

Document Details

Submission ID

trn:oid:::3618:79770592

57 Pages

Submission Date

Jan 21, 2025, 10:28 PM GMT+5:45

4,128 Words

Download Date

Jan 21, 2025, 10:33 PM GMT+5:45

26,546 Characters

File Name

23049330 ROHIT JOSHI2.docx

File Size

30.9 KB



Page 1 of 62 - Cover Page

Submission ID trn:oid:::3618:79770592



Page 2 of 62 - Integrity Overview

Submission ID trn:oid:::3618:79770592

36% Overall Similarity

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

Match Groups

-  103 Not Cited or Quoted 29%
Matches with neither in-text citation nor quotation marks
-  31 Missing Quotations 7%
Matches that are still very similar to source material
-  0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation
-  0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 9%  Internet sources
- 0%  Publications
- 32%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

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

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

TABLE OF CONTENTS

1. INTRODUCTION:	1
1.1 Description of Current Business Activities and Operations	2
1.2 BUSINESS RULES	3
1.3 ASSUMPTIONS	4
2. IDENTIFICATION OF ENTITIES AND ATTRIBUTES:	5
2.2 For Student:	5
2.1 For Program:	5
2.3 For Module:	6
3. INITIAL ERD:	7
4. NORMALIZATION	8
4.1 Unnormalized Form (UNF)	8
4.2 First Normal Form (1NF)	9
4.3 Second Normal Form (2NF)	10
4.4 Third Normal Form (3NF)	15
4.5 Final Tables Formed:	17
5. FINAL ER-DIAGRAM:	18
6. DATA DICTIONARY	19
6.1 Data Dictionary table for Program	19
6.2 Data Dictionary table for Student	19

6.3 Data Dictionary table for Module_____	20
6.4 Data Dictionary table for Mod_Stud_____	20
6.5 Data Dictionary table for Teacher _____	20
6.6 Data Dictionary table for Teach_Mod_Stud_____	21
6.7 Data Dictionary table for Announcement_____	21
6.8 Data Dictionary table for Announce_Teach_Mod_Stud_____	21
6.9 Data Dictionary table for Assessment _____	22
6.10 Data Dictionary table for Assess_Mod_Stud_____	22
6.11 Data Dictionary table for Resources_____	23
6.12 Data Dictionary table for Reso_Mod_Stud_____	23
7. IMPLEMENTATION_____	24
7.1 Creation and Description of Program table_____	25
7.2 Creation and Description of Student table_____	26
7.3 Creation and Description of Module table_____	27
7.4 Creation and Description of Mod_Stud table_____	28
7.5 Creation and Description of Teacher table_____	29
7.6 Creation and Description of Teach_Mod_Stud table_____	30
7.7 Creation and Description of Announcement table_____	31
7.8 Creation and Description of Announce_Teach_Mod_Stud table_____	32
7.9 Creation and Description of Assessment table_____	33
7.10 Creation and Description of Assess_Mod_Stud table_____	34
7.11 Creation and Description of Resources table_____	35

7.12 Creation and Description of Reso_Mod_Stud table	36
7.13 Insertion of Test Data in Program Table	37
7.14 Insertion of Test Data in Student Table	38
7.15 Insertion of Test Data in Module Table	39
7.16 Insertion of Test Data in Mod_Stud Table	40
7.17 Insertion of Test Data in Teacher Table	42
7.18 Insertion of Test Data in Teach_Mod_Stud Table	43
7.19 Insertion of Test Data in Announcement Table	45
7.20 Insertion of Test Data in Announce_Teach_Mod_Stud Table	46
7.21 Insertion of Test Data in Assessment Table	47
7.22 Insertion of Test Data in Assess_Mod_Stud Table	49
7.23 Insertion of Test Data in Resources Table	50
7.24 Insertion of Test Data in Reso_Mod_Stud Table	51
8. DATABASE QUERYING	53
8.1 Information Query	53
8.2 Transaction Query	55
9. CRITICAL EVALUATION	58
10. DUMP FILE CREATION	59
11. DROPPING TABLES	60
12. BIBLIOGRAPHY	61

TABLE OF FIGURES

FIGURE 1: LITTLE ANGLES COLLEGE (LACM)	1
FIGURE 2: INITIAL ERD	7
FIGURE 3: FINAL ERD	18
FIGURE 4: CREATING USER AND GRANTING PERMISSIONS	24
FIGURE 5: CONNECTING TO USER (ROHITJOSHI)	24
FIGURE 6: CREATING PROGRAM TABLE	25
FIGURE 7: DESCRIBING OF PROGRAM TABLE	25
FIGURE 8: CREATING STUDENT TABLE	26
FIGURE 9: DESCRIBING STUDENT TABLE	26
FIGURE 10: CREATING MODULE TABLE	27
FIGURE 11: DESCRIBING MODULE TABLE	27
FIGURE 12: CREATING MOD_STUD TABLE	28
FIGURE 13: DESCRIBING MOD_STUD TABLE	28
FIGURE 14: CREATING TEACHER TABLE	29
FIGURE 15: DESCRIBING TEACHER TABLE	29
FIGURE 16: CREATING TEACH_MOD_STUD TABLE	30
FIGURE 17: DESCRIBING TEACH_MOD_STUD TABLE	30
FIGURE 18: CREATING ANNOUNCEMENT TABLE	31
FIGURE 19: DESCRIBING ANNOUNCEMENT TABLE	31
FIGURE 20: CREATING ANNOUNCE_TEACH_MOD_STUD TABLE	32
FIGURE 21: DESCRIBING ANNOUNCE_TEACH_MOD_STUD TABLE	32
FIGURE 22: CREATING ASSESSMENT TABLE	33
FIGURE 23: DESCRIBING ASSESSMENT TABLE	33
FIGURE 24: CREATING ASSESS_MOD_STUD TABLE	34
FIGURE 25: DESCRIBING ASSESS_MOD_STUD TABLE	34
FIGURE 26: CREATING RESOURCES TABLE	35
FIGURE 27: DESCRIBING RESOURCES TABLE	35
FIGURE 28: CREATING RESO_MOD_STUD TABLE	36
FIGURE 29: DESCRIBING RESO_MOD_STUD TABLE	36
FIGURE 30: INSERTING DATA OF PROGRAM TABLE	37
FIGURE 31: SELECTING DATA OF PROGRAM TABLE	37
FIGURE 32: INSERTING DATA OF STUDENT TABLE	38
FIGURE 33: SELECTING DATA OF STUDENT TABLE	38
FIGURE 34: INSERTING DATA OF MODULE TABLE	39
FIGURE 35: SELECTING DATA OF MODULE TABLE	39

FIGURE 36: INSERTING DATA OF MOD_STUD TABLE	40
FIGURE 37: SELECTING DATA OF MOD_STUD TABLE	41
FIGURE 38: INSERTING DATA OF TEACHER TABLE	42
FIGURE 39: SELECTING DATA OF TEACHER TABLE	42
FIGURE 40: INSERTING DATA OF TEACH_MOD_STUD TABLE	43
FIGURE 41: SELECTING DATA OF TEACH_MOD_STUD TABLE	44
FIGURE 42: INSERTING DATA OF ANNOUNCEMENT TABLE	45
FIGURE 43: SELECTING DATA OF ANNOUNCEMENT TABLE	45
FIGURE 44: INSERTING DATA OF ANNOUNCE_TEACH_MOD_STUD TABLE	46
FIGURE 45: SELECTING DATA OF ANNOUNCE_TEACH_MOD_STUD TABLE	46
FIGURE 46: INSERTING DATA OF ASSESSMENT TABLE	47
FIGURE 47: SELECTING DATA OF ASSESSMENT TABLE	48
FIGURE 48: INSERTING DATA OF ASSESS_MOD_STUD TABLE	49
FIGURE 49: SELECTING DATA OF ASSESS_MOD_STUD TABLE	49
FIGURE 50: INSERTING DATA OF RESOURCES TABLE	50
FIGURE 51: SELECTING DATA OF RESOURCES TABLE	50
FIGURE 52: INSERTING DATA OF RESO_MOD_STUD TABLE	51
FIGURE 53: SELECTING DATA OF RESO_MOD_STUD TABLE	52
FIGURE 54: LISTING AVAILABLE PROGRAMS AND STUDENTS	53
FIGURE 55: LISTING ANNOUNCEMENT FOR SELECTED DATE	53
FIGURE 56: LISTING ALL MODULES THAT BEGIN WITH THE LETTER ‘D’	54
FIGURE 57: LISTING STUDENTS NAME WHO HAVE NOT SUBMITTED ASSESSMENT	54
FIGURE 58: LISTING TEACHER WHO TEACH MORE THAN ONE MODULE	54
FIGURE 59: LISTING MODULE THAT HAS LATEST ASSESSMENT DEADLINE	55
FIGURE 60: LISTING TOP THREE STUDENT WITH HIGHEST MARKS	55
FIGURE 61: LISTING PROGRAM WITH NUMBER OF ASSESSMENT AND AVERAGE SCORE	56
FIGURE 62: LISTING STUDENT WITH SCORE OF ABOVE AVERAGE	56
FIGURE 63: LISTING REMARKS OF STUDENT IN A PARTICULAR MODULE	57
FIGURE 64: DUMP FILE CREATION	59
FIGURE 65: DUMP FILE CREATED	59
FIGURE 66: DROPPING ALL TABLES	60

TABLE OF TABLES

TABLE 1: IDENTIFICATION OF ATTRIBUTES FOR STUDENT	5
TABLE 2: IDENTIFICATION OF ATTRIBUTES FOR PROGRAM	5
TABLE 3: IDENTIFICATION OF ATTRIBUTES FOR MODULE	6
TABLE 4: DATA DICTIONARY FOR PROGRAM	19
TABLE 5: DATA DICTIONARY FOR STUDENT	19
TABLE 6: DATA DICTIONARY FOR MODULE	20
TABLE 7: DATA DICTIONARY FOR MOD_STUD	20
TABLE 8: DATA DICTIONARY FOR TEACHER	20
TABLE 9: DATA DICTIONARY FOR TEACH_MOD_STUD	21
TABLE 10: DATA DICTIONARY FOR ANNOUNCEMENT	21
TABLE 11: DATA DICTIONARY FOR ANNOUNCE_TEACH_MOD_STUD	21
TABLE 12: DATA DICTIONARY FOR ASSESSMENT	22
TABLE 13: DATA DICTIONARY FOR ASSESS_MOD_STUD	22
TABLE 14: DATA DICTIONARY FOR RESOURCES	23
TABLE 15: DATA DICTIONARY FOR RESO_MOD_STUD	23

1. INTRODUCTION:

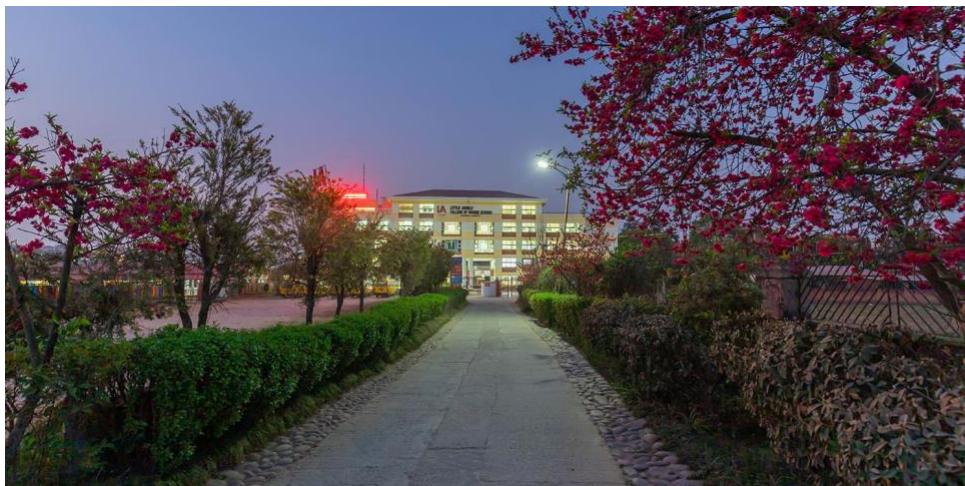


Figure 1: Little Angles College (LACM)

Little Angels College of Management (LACM), established in 2009, is a premier educational institution located in Hattiban, Lalitpur, Nepal. Renowned for its commitment to academic excellence and holistic student development, LACM offers diverse programs in Business Management, Computing, Networking, and Multimedia. The institution emphasizes a well-rounded learning experience by combining quality education with extracurricular activities such as sports, cultural events, and student-led clubs. With modern infrastructure, experienced faculty, and a strong focus on fostering innovation and leadership, LACM aims to prepare students for a competitive, technology-driven world. Upholding values of creativity, collaboration, and inclusivity, the college nurtures a global perspective among learners and empowers them to excel academically while growing into responsible, innovative leaders. (Limbu & Shrestha, 1996)

Entrepreneur Ms. Mary, recognizing the need for a structured digital learning environment, has proposed the development of the E-Classroom Platform. This platform is designed to modernize LACM's operations, enabling seamless management of students, programs, modules, and assessments while enhancing the teaching and learning experience. It will further the institution's mission by integrating advanced technological tools to support structured, progressive, and dynamic education.

1.1 Description of Current Business Activities and Operations

The college currently operates with traditional classroom methods, delivering lectures, conducting practical workshops, and organizing interactive sessions to facilitate student learning. Alongside its academic activities, the college fosters a dynamic extracurricular environment, offering sports, cultural events, and student-run clubs. These initiatives aim to promote both the physical and mental well-being of students, while encouraging leadership skills and creativity.

To further enhance its operations, Ms. Mary has proposed the introduction of the "E-Classroom Platform." This innovative digital tool aims to modernize key academic and administrative functions. The platform will facilitate efficient student management, allowing for seamless enrollment and tracking across various programs. It will also enable the organization of programs and modules, ensuring curriculum flexibility while maintaining a structured approach. The system will integrate assessment management, enabling the link between assessments and modules, recording student performance, and generating detailed reports. Furthermore, the platform will streamline resource access, organizing learning materials in a structured, progressive manner. To improve communication, the platform will allow real-time interaction between teachers and students, enabling the posting of announcements and notices specific to individual modules.

1.2 BUSINESS RULES

Mary's E-Classroom has established its own set of business rules for maintaining the Students and Teachers Activities and operations.

- Multiple Student can be enrolled in a program, and only one program can be enrolled in by a student at a time
- Every program can have multiple number of modules, and A module can also be included in multiple number of programs.
- A student can be enrolled in multiple modules, and multiple modules can have multiple number of students.
- There are one or more assessments in each module, and each assessment is specific to a single module.
- Although teachers might be in more than one module, at least one teacher teaches each module.
- At least one assessment and resource must be included in each module, and each assessment and resource must be connected to that module.
- A teacher can announce notice for their respective modules, and each notice must be linked to their own specific module.
- Multiple Resources can be included in a module, and students much finish resources in a predefined sequence (finish resource 1 before going on to the resource 2)

1.3 ASSUMPTIONS

- A student can only be enrolled for one program at a time, and students are unable to change until and unless their current program is withdrawn or finished.
- Every assessment must record the results (marks obtained) for all enrolled students in the module.
- A module can contain multiple number of teachers, but they are only responsible for their assigned roles within each module.

2. IDENTIFICATION OF ENTITIES AND ATTRIBUTES:

An Entity can be an object or a thing in the real-world scenario where it contains attributes which describes about that entity. (Jain, n.d.)

An Attribute can be defined as a property or characteristic of an entity which is known as data that provides more information about the entity. In this case there are 3 entities student, program and module where the Student_ID, Program_ID and Module_ID define the three entities. (Jain, n.d.)

Table below contains all the entities and their attributes with the Constraints.

2.2 For Student:

S. No.	Attribute Name	Data Type	Size	Constraint
1	Student_ID	Character	15	Primary Key
2	Name	Character	20	Not Null
3	Address	Character	20	Not Null
4	Contact	Character	15	Not Null
5	Email	Character	30	Unique
6	JoinedDate	Date		Not Null

Table 1: Identification of Attributes for Student

2.1 For Program:

S. No.	Attribute Name	Data Type	Size	Constraint
1	Program_ID	Character	15	Primary Key
2	Program_Name	Character	25	Not Null
3	Duration	Number	10	Not Null
4	Description	Character	60	Not Null

Table 2: Identification of Attributes for Program

2.3 For Module:

S. No.	Attribute Name	Data Type	Size	Constraint
1	Module_ID	Character	15	Primary Key
2	Module_Name	Character	40	Not Null
3	Credit_Hours	Number	5	Not Null
4	Assessment_ID	Character	15	Primary Key
5	Assessment_Title	Character	15	Not Null
6	Deadline	Date		Not Null
7	Weightage	Number	10	Not Null
8	Marks_Obtained	Number	3	Not Null
9	Resource_ID	Character	15	Primary Key
10	Resource_Name	Character	40	Not Null
11	Resource_Type	Character	15	Not Null
12	R_Duration	Number	10	Not Null
13	ResourceStatus	Character	20	Not Null
14	Announcement_ID	Character	20	Primary Key
15	Content	Character	50	Not Null
16	AnnounceDate	Date		Not Null
17	Announcement_Title	Character	30	Not Null
18	Teacher_ID	Character	15	Primary Key
19	Teacher_Name	Character	20	Not Null
20	Teacher_Email	Character	40	Unique
21	Specialization	Character	40	Not Null
22	Grade	Character	5	Not Null
23	Submission_Status	Character	20	Not Null

Table 3: Identification of Attributes for Module

3. INITIAL ERD:

An Entity-Relational Diagram (ERD) is a visual representation of database structure which shows how the data is being organized and being stored, where the entities represent the objects such as Teacher, Student, Module, Program, etc. This is the Initial Stage for the ER-diagram where the data is not arranged properly, we will move forward with the process to get it more refined and arranged. (Jain, 2024)

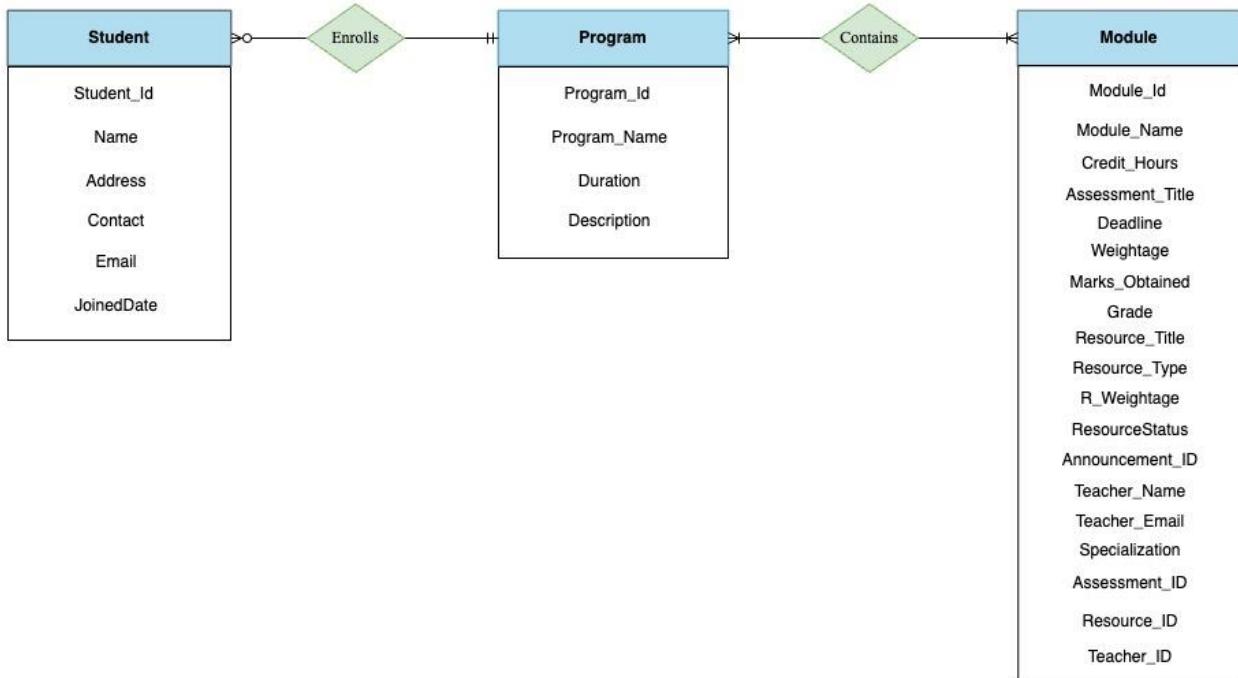


Figure 2: Initial ERD

4. Normalization

Normal forms are a collection of guidelines used in database management systems (DBMS) to ensure that a database's design is effective, well-structured, and devoid of data abnormalities. Each of the several normalization phases has a unique set of guiding principles known as normal forms. For this case study we are going through 1NF, 2NF, 3NF which are briefly described in this Case. (Jain, 2024)

4.1 Unnormalized Form (UNF)

Unnormalized form (UNF) is a very simple database where all the data organizes in a database. Unnormalized form can also be known as the beginning of a normalization process in the relational model. Unnormalized form can deal with complex data structure by also making then querying process lot simpler and restructuring the date is easy on this form. (Jain, 2024).

This is UNF:

```
E-Classroom - (Student_ID, Name, Email, Address, Contact, Enrollment_Date,  
Program_ID, Program_Name, Duration, Description), {Module_ID, Module_Name,  
Credit_Hours, {Teacher_ID, Teacher_Name, Teacher_Email, Specialization,  
{Announcement_ID, Announcement_Title, Content, AnnounceDate}}, {Assessment_ID,  
Assessment_Title, Deadline, Submission_Status, Weightage, Marks_Obtained, Grade},  
{Resource_ID, Resource_Name, Resource_Type, R_Duration, ResourceStatus}  
}
```

4.2 First Normal Form (1NF)

If a relation has no repeating groups or multivalued columns and only atomic (single-valued) properties, it is said to be in 1NF. This implies that there should be no sets, lists, or arrays in any column of a table and that each field must have a single value. By ensuring that the data is arranged in a straightforward tabular format, 1NF enhances consistency and minimizes redundancy. (Jaiswal, n.d.)

Removing Repeating Groups

Table 1

Stud_Prog-1 (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID, Program_Name, Duration, Description)

Table 2

Stud_Module-1 (Student_ID*, Module_ID, Module_Name, Credit_Hours)

Table 3

Teacher-1 (Teacher_ID, Student_ID*, Module_ID*, Teacher_Name, Teacher_Email, Specialization)

Table 4

Announcement-1 (Announcement_ID, Teacher_ID*, Student_ID*, Module_ID*, Announcement_Title, Content, AnnounceDate)

Table 5

Assessment-1 (Assessment_ID, Student_ID*, Module_ID*, Assessment_Title,
Submission_Status, Deadline, Weightage, Marks_Obtained, Grade)

Table 6

Resources-1 (Resource_ID, Student_ID*, Module_ID*, Resource_Name, Resource_Type,
R_Duration, ResourceStatus)

4.3 Second Normal Form (2NF)

If every non-primary-key property of a connection in First Normal Form depends fully upon the Primary key for any kind of function, the relation is in 2NF. The goal of 2NF is to remove partial dependencies and bring the relation into a higher normal form, such as Third Normal Form. (Jaiswal, n.d.)

Removing Partial Dependencies

- ❖ **For Table:** Stud_Prog-1 (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID, Program_Name, Duration, Description)

$\text{Student_ID} \rightarrow \text{Name, Address, Contact, Email, JoinedDate, Program_ID, Program_Name, Duration, Description}$

Formed Table:

Stud_Prog-2 (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID, Program_Name, Duration, Description)

- ❖ **For Table:** Stud_Mod-1 (Student_ID*, Module_ID, Module_Name, Credit_Hours)

Module_ID → Module_Name, Credit_Hours

Module_ID, Student_ID → None

Formed Tables:

Module-2 (Module_ID, Module_Name, Credit_Hours)

Mod_Stud-2 (Module_ID*, Student_ID*)

- ❖ **For Table:** Teacher-1 (Teacher_ID, Student_ID*, Module_ID*, Teacher_Name, Teacher_Email, Specialization)

Teacher_ID → Teacher_Name, Teacher_Email, Specialization

Student_ID, Module_ID, Teacher_ID → None

Formed Tables:

Teacher-2 (Teacher_ID, Teacher_Name, Teacher_Email, Specialization)

Teach_Mod_Stud-2 (Teacher_ID*, Student_ID*, Module_ID*)

- ❖ **For Table:** Announcement-1 (Announcement_ID, Teacher_ID*, Student_ID*,
Module_ID*, Announcement_Title, Content, AnnounceDate)

Announcement_ID → Announcement_Title, Content, AnnounceDate

Announcement_ID, Teacher_ID, Student_ID, Module_ID → None

Formed Tables:

Announcement-2 (Announcement_ID, Announcement_Title, Content, AnnounceDate)

Announce_Teach_Mod_Stud -2 (Announcement_ID*, Teacher_ID*, Student_ID*,
Module_ID*)

- ❖ **For Table: Assessment-2** (Assessment_ID*, Student_ID*, Module_ID*, Assessment_Title,
Submission_Status, Deadline, Weightage, Marks_Obtained, Grade)

Assessment_ID → Assessment_Title, Deadline, Weightage,

Assessment_ID, Student_ID, Module_ID → Submission_Status, Marks_Obtained, Grade

Formed Tables:

Assessment-2 (Assessment_ID, Assessment_Title, Deadline, Weightage)

Assess_Module_Student-2 (Assessment_ID*, Student_ID*, Module_ID*,
Submission_Status, Marks_Obtained, Grade)

- ❖ **For Table:** Resource-1 (Resource_ID, Student_ID*, Module_ID*, Resource_Name, Resource_Type, R_Duration, ResourceStatus)

Resource_ID → Resource_Name, Resource_Type, R_Duration

Resource_ID, Student_ID, Module_ID → ResourceStatus

Formed Tables:

Resources-2 (Resource_ID, Resource_Name, Resource_Type, R_Duration)

Reso_Mod_Stud -2 (Resource_ID*, Student_ID*, Module_ID*, ResourceStatus)

Final Tables for 2NF

1. **Stud_Prog-2** (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID, Program_Name, Duration, Description)
2. **Module-2** (Module_ID, Module_Name, Credit_Hours)
3. **Mod_Stud-2** (Module_ID*, Student_ID*)
4. **Teacher-2** (Teacher_ID, Teacher_Name, Teacher_Email, Specialization)
5. **Teach_Mod_Stud-2** (Teacher_ID*, Student_ID*, Module_ID*)
6. **Announcement-2** (Announcement_ID, Announcement_Title, Content, AnnounceDate)
7. **Announce_Teach_Mod_Stud -2** (Announcement_ID*, Teacher_ID*, Student_ID*, Module_ID*)
8. **Assessment-2** (Assessment_ID, Assessment_Title, Deadline, Weightage)
9. **Assess_Mod_Stud-2** (Assessment_ID*, Student_ID*, Module_ID*, Submission_Status, Marks_Obtained, Grade)
10. **Resources-2** (Resource_ID, Resource_Name, Resource_Type, R_Duration)
11. **Reso_Mod_Stud -2** (Resource_ID*, Student_ID*, Module_ID*, ResourceStatus)

4.4 Third Normal Form (3NF)

By avoiding transitive dependencies, the Third Normal Form (3NF) is a database normalization strategy that ensures non-prime attributes are independent of other non-prime properties. If a relation has a functional dependency ($X \rightarrow Y$) and is in Second Normal Form (2NF), with X being a super key or Y being a prime attribute, then the relation is in 3NF. 3NF's major goal is to remove duplicate data while preserving data integrity, which improves database efficiency. (Jaiswal, n.d.)

Checking and Removing Transitive Dependencies:

For Stud_Prog-2 (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID, Program_Name, Duration, Description)

Student_ID \rightarrow Program_ID \rightarrow (Program_Name, Duration, Description)

Student_ID \rightarrow None

Formed Tables:

Program-3: (Program_ID, Program_Name, Duration, Description)

Student-3: (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID)

In the Stud_Prog table, a transitive dependency exists where Student_ID determines Program_ID, and Program_ID determines Program_Name, Program_Duration, and Program_Description. This creates redundancy as program details repeat for every student in the same program. To solve this problem, we have separated the Stud_Prog table into two tables named Student and Program tables for their respective attributes which resolves the problem of transitive dependency.

Module-3 (Module_ID, Module_Name, Credit_Hours)

Mod_Stud-3 (Module_ID*, Student_ID*)

Teacher-3 (Teacher_ID, Teacher_Name, Teacher_Email, Specialization)

Teach_Mod_Stud-3 (Teacher_ID*, Student_ID*, Module_ID*)

Announcement-3 (Announcement_ID, Announcement_Title, Content, AnnounceDate)

Announce_Teach_Mod_Stud-3 (Announcement_ID*, Teacher_ID*, Student_ID*,
Module_ID*)

Assessment-3 (Assessment_ID, Assessment_Title, Deadline, Weightage)

Assess_Mod_Stud-3 (Assessment_ID*, Student_ID*, Module_ID*, Submission_Status
Marks_Obtained, Grade)

Resources-3 (Resource_ID, Resource_Name, Resource_Type, R_Duration)

Reso_Mod_Stud -3 (Resource_ID*, Student_ID*, Module_ID*, ResourceStatus)

The remaining tables (Module, Mod_Stud, Teacher, Teach_Mod_Stud, Announcement, Announce_Teach_Mod_Stud, Resource, Reso_Mod_Stud, Assessment, and Assess_Mod_Stud) does not have transitive dependencies. So, all the present attributes dependents upon their respective primary keys, Because of which they are already in the Third Normal Form (3NF).

4.5 Final Tables Formed:

1. **Program** (Program_ID, Program_Name, Duration, Description)
2. **Student** (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID*)
3. **Module** (Module_ID, Module_Name, Credit_Hours)
4. **Mod_Stud** (Module_ID*, Student_ID*)
5. **Teacher** (Teacher_ID, Teacher_Name, Teacher_Email, Specialization)
6. **Teach_Mod_Stud** (Teacher_ID*, Student_ID*, Module_ID*)
7. **Announcement** (Announcement_ID, Announcement_Title, Content, AnnounceDate)
8. **Announce_Teach_Mod_Stud** (Announcement_ID*, Teacher_ID*, Student_ID*, Module_ID*)
9. **Assessment** (Assessment_ID, Assessment_Title, Deadline, Weightage)
10. **Assess_Mod_Stud** (Assessment_ID*, Student_ID*, Module_ID*, Submission_Status
Marks_Obtained, Grade)
11. **Resources** (Resource_ID, Resource_Name, Resource_Type, R_Duration)
12. **Reso_Mod_Stud** (Resource_ID*, Student_ID*, Module_ID*, ResourceStatus)

5. FINAL ER-DIAGRAM:

An Entity-Relational Diagram (ERD) is a visual representation of database structure which shows how the data is being organized and being stored, where the entities represent the objects such as Teacher, Student, Module, Program, etc. This is the final ER-Diagram which is made after the Normalization process were we know which are actual tables formed and based on those final tables after the Normalization process we create this Final ER-Diagram which is represented in the figure 3 below: (Jain, 2024)

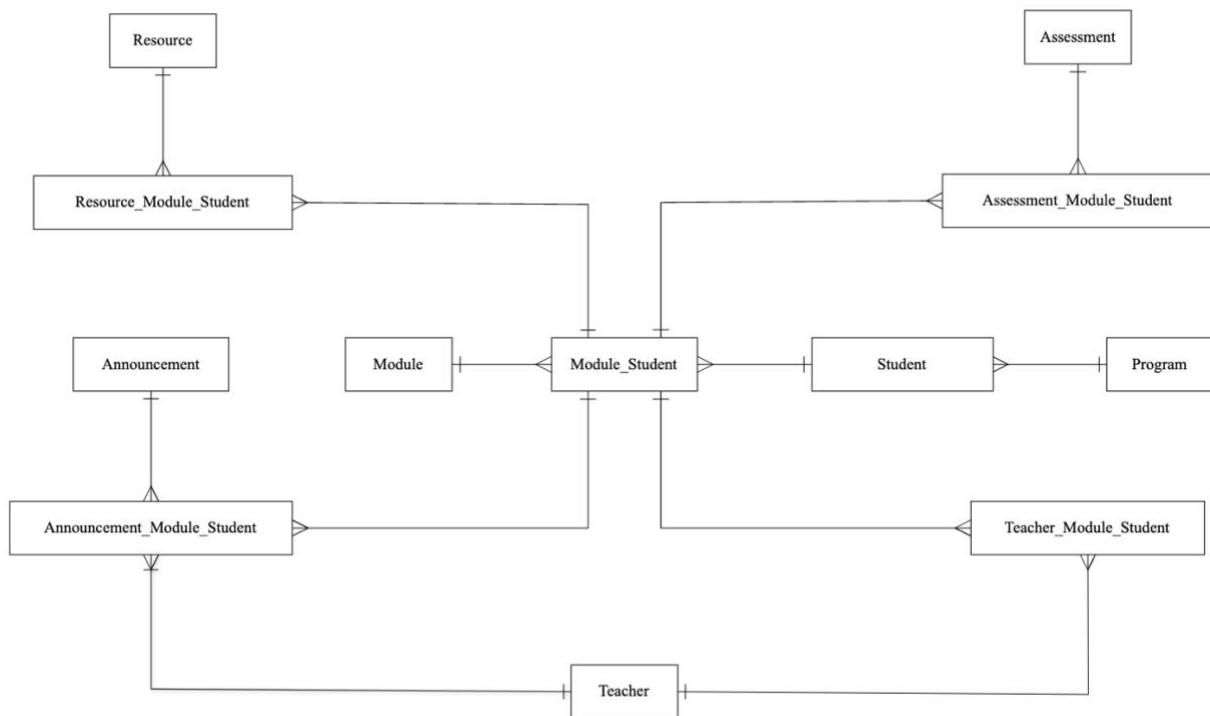


Figure 3: Final ERD

6. Data Dictionary

Collection of names, numbers and different types of attributes are used in database is known as Data Dictionary, which helps the user to handle the data more efficiently with in an ordered manner to prevent data redundancy. (Jain, n.d.)

6.1 Data Dictionary table for Program

S. No.	Attribute Name	Data Type	Size	Constraint
1	Program_ID	Character	15	Primary Key
2	Program_Name	Character	25	Not Null
3	Duration	Character	10	Not Null
4	Description	Character	60	Not Null

Table 4: Data Dictionary for Program

6.2 Data Dictionary table for Student

S. No.	Attribute Name	Data Type	Size	Constraint
1	Student_ID	Character	15	Primary Key
2	Name	Character	20	Not Null
3	Address	Character	20	Not Null
4	Contact	Character	15	Not Null
5	Email	Character	30	Unique
6	JoinedDate	Date		Not Null
7	Program_ID	Character	15	Foreign Key

Table 5: Data Dictionary for Student

6.3 Data Dictionary table for Module

S. No.	Attribute Name	Data Type	Size	Constraint
1	Module_ID	Character	15	Primary Key
2	Module_Name	Character	40	Not Null
3	Credit_Hours	Number	5	Not Null

Table 6: Data Dictionary for Module

6.4 Data Dictionary table for Mod_Stud

S. No.	Attribute Name	Data Type	Size	Constraint	Composite Constraint
1	Module_ID	Character	15	Primary Key, Foreign Key	Composite Key
2	Student_ID	Character	15	Primary Key, Foreign Key	

Table 7: Data Dictionary for Mod_Stud

6.5 Data Dictionary table for Teacher

S. No.	Attribute Name	Data Type	Size	Constraint
1	Teacher_ID	Character	15	Primary Key
2	Teacher_Name	Character	20	Not Null
3	Teacher_Email	Character	40	Unique
4	Specialization	Character	40	Not Null

Table 8: Data Dictionary for Teacher

6.6 Data Dictionary table for Teach_Mod_Stud

S. No.	Attribute Name	Data Type	Size	Constraint	Composite Constraint
1	Teacher_ID	Character	15	Primary Key, Foreign Key	Composite Key
2	Student_ID	Character	15	Primary Key, Foreign Key	
3	Module_ID	Character	15	Primary Key, Foreign Key	

Table 9: Data Dictionary for Teach_Mod_Stud

6.7 Data Dictionary table for Announcement

S. No.	Attribute Name	Data Type	Size	Constraint
1	Announcement_ID	Character	20	Primary Key
2	Announcement_Title	Character	30	Not Null
3	Announcement_Content	Character	50	Not Null
4	Announcement_Date	Date		Not Null

Table 10: Data Dictionary for Announcement

6.8 Data Dictionary table for Announce_Teach_Mod_Stud

S. No.	Attribute Name	Data Type	Size	Constraint	Composite Constraint
1	Announcement_ID	Character	20	Primary Key, Foreign Key	Composite Key
2	Teacher_ID	Character	15	Primary Key, Foreign Key	
3	Module_ID	Character	15	Primary Key, Foreign Key	
4	Student_ID	Character	15	Primary Key, Foreign Key	

Table 11: Data Dictionary for Announce_Teach_Mod_Stud

6.9 Data Dictionary table for Assessment

S. No.	Attribute Name	Data Type	Size	Constraint
1	Assessment_ID	Character	15	Primary Key
2	Assessment_Title	Character	15	Not Null
3	Deadline	Date		Not Null
4	Weightage	Number	10	

Table 12: Data Dictionary for Assessment

6.10 Data Dictionary table for Assess_Mod_Stud

S. No.	Attribute Name	Data Type	Size	Constraint	Composite Constraint
1	Assessment_ID	Character	15	Primary Key, Foreign Key	Composite key
2	Student_ID	Character	15	Primary Key, Foreign Key	
3	Module_ID	Character	15	Primary Key, Foreign Key	
4	Submission_Status	Character	20	Not Null	
5	Marks_Obtained	Number	20	Not Null	
6	Grade	Character	5	Not Null	

Table 13: Data Dictionary for Assess_Mod_Stud

6.11 Data Dictionary table for Resources

S. No.	Attribute Name	Data Type	Size	Constraint
1	Resource_ID	Number	15	Primary Key
2	Resource_Name	Character	40	Not Null
3	Resource_Type	Character	15	Not Null
4	Duration	Character	10	Not Null

Table 14: Data Dictionary for Resources

6.12 Data Dictionary table for Reso_Mod_Stud

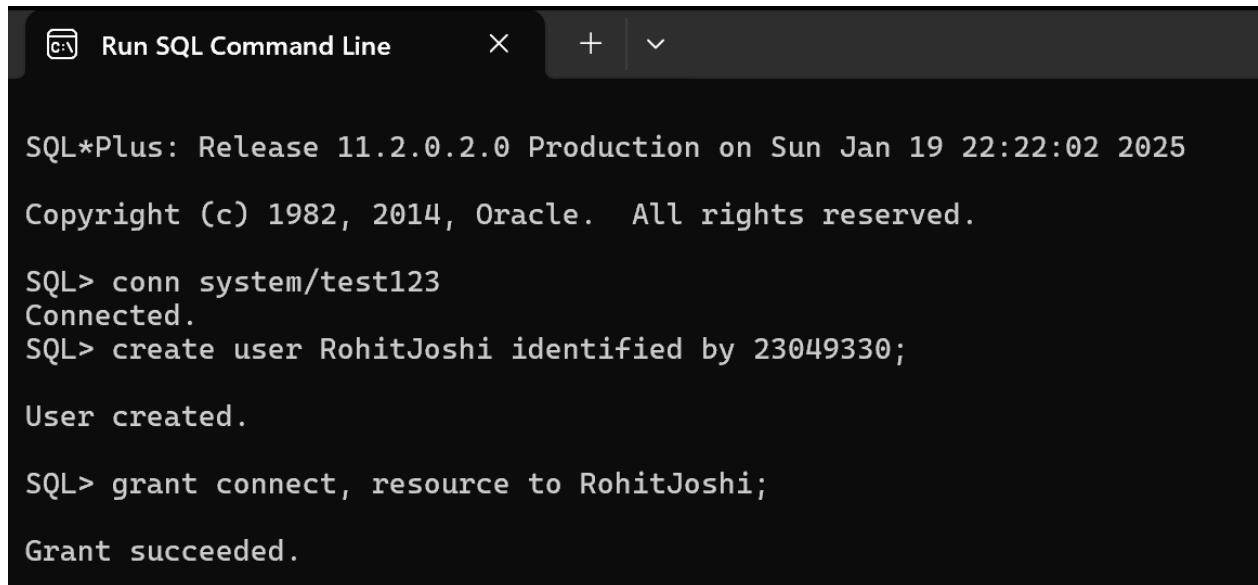
S. No.	Attribute Name	Data Type	Size	Constraint	Composite Constraint
1	Resource_ID	Number	15	Primary Key, Foreign Key	Composite Key
2	Student_ID	Number	15	Primary Key, Foreign Key	
3	Module_ID	Number	15	Primary Key, Foreign Key	
5	Resource_Status	Character	20	Not Null	

Table 15: Data Dictionary for Reso_Mod_Stud

7. Implementation

Here in implementation process, we must create the 12 tables in SQL*Plus which we get after the normalization process which are Student, Program, Module, Mod_Stud, Teacher, Teach_Mod_Stud, Announcement, Announce_Teach_Mod_Stud, Assessment, Assess_Mod_Stud, Resources, Reso_Mod_Stud.

Here we have firstly connected to the system and then we have Created the User, which is named RohitJoshi and Identified by password 23049330, After that we have granted the resources to the system, which is shown in the fig 4 and finally, we connect to the User as shown in the fig 5.



The screenshot shows a terminal window titled "Run SQL Command Line". The output of the session is as follows:

```
SQL*Plus: Release 11.2.0.2.0 Production on Sun Jan 19 22:22:02 2025
Copyright (c) 1982, 2014, Oracle. All rights reserved.

SQL> conn system/test123
Connected.
SQL> create user RohitJoshi identified by 23049330;
User created.

SQL> grant connect, resource to RohitJoshi;
Grant succeeded.
```

Figure 4: Creating User and granting permissions



```
SQL> conn RohitJoshi/23049330
Connected.
```

Figure 5: Connecting to User (RohitJoshi)

7.1 Creation and Description of Program table

Creation Program Table

```
SQL> conn RohitJoshi/23049330
Connected.
SQL> CREATE TABLE Program (
  2  Program_ID VARCHAR(15) PRIMARY KEY,
  3  Program_Name VARCHAR(25) NOT NULL,
  4  Duration VARCHAR(10) NOT NULL,
  5  Description VARCHAR(60) NOT NULL);
```

Table created.

Figure 6: Creating Program Table

Description of Program table

Name	Null?	Type
PROGRAM_ID	NOT NULL	VARCHAR2(15)
PROGRAM_NAME	NOT NULL	VARCHAR2(25)
DURATION	NOT NULL	VARCHAR2(10)
DESCRIPTION	NOT NULL	VARCHAR2(60)

Figure 7: Describing of Program table

7.2 Creation and Description of Student table

Creating Student table

```
SQL> CREATE TABLE Student (
  2  Student_ID VARCHAR(15) PRIMARY KEY,
  3  Name VARCHAR(20) NOT NULL,
  4  Address VARCHAR(20) NOT NULL,
  5  Contact VARCHAR(15) NOT NULL,
  6  Email VARCHAR(30) UNIQUE,
  7  JoinedDate DATE NOT NULL,
  8  Program_ID VARCHAR(15) NOT NULL,
  9  FOREIGN KEY (Program_ID) REFERENCES Program(Program_ID));
```

Table created.

Figure 8: Creating Student table

Describing Student table

```
SQL> DESC Student;
      Name          Null?    Type
-----  -----
STUDENT_ID           NOT NULL  VARCHAR2(15)
NAME                NOT NULL  VARCHAR2(20)
ADDRESS              NOT NULL  VARCHAR2(20)
CONTACT              NOT NULL  VARCHAR2(15)
EMAIL                VARCHAR2(30)
JOINEDDATE          NOT NULL  DATE
PROGRAM_ID           NOT NULL  VARCHAR2(15)
```

Figure 9: Describing Student table

7.3 Creation and Description of Module table

Creating Module table

```
SQL> CREATE TABLE Module (
  2  Module_ID VARCHAR(15) PRIMARY KEY,
  3  Module_Name VARCHAR(40) NOT NULL,
  4  Credit_Hours NUMBER(5) NOT NULL);
```

Table created.

Figure 10: Creating Module table

Describing Module table

```
SQL> DESC Module;
Name                           Null?    Type
-----                         -----
MODULE_ID                      NOT NULL VARCHAR2(15)
MODULE_NAME                     NOT NULL VARCHAR2(40)
CREDIT_HOURS                   NOT NULL NUMBER(5)
```

Figure 11: Describing Module table

7.4 Creation and Description of Mod_Stud table

Creating Mod_Stud table

```
SQL> CREATE TABLE Mod_Stud (
  2 Module_ID VARCHAR(15) NOT NULL,
  3 Student_ID VARCHAR(15) NOT NULL,
  4 PRIMARY KEY (Module_ID, Student_ID),
  5 FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID),
  6 FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID));
Table created.
```

Figure 12: Creating Mod_Stud table

Describing Mod_Stud table

```
SQL> DESC Mod_Stud;
      Name          Null?    Type
-----  
MODULE_ID           NOT NULL  VARCHAR2(15)  
STUDENT_ID          NOT NULL  VARCHAR2(15)
```

Figure 13: Describing Mod_Stud table

7.5 Creation and Description of Teacher table

Creating Teacher table

```
SQL> CREATE TABLE Teacher (
 2 Teacher_ID VARCHAR(15) PRIMARY KEY,
 3 Teacher_Name VARCHAR(20) NOT NULL,
 4 Teacher_Email VARCHAR(40) UNIQUE,
 5 Specialization VARCHAR(40) NOT NULL);
```

Table created.

Figure 14: Creating Teacher table

Describing Teacher table

```
SQL> DESC Teacher;
Name                           Null?    Type
-----                         -----
TEACHER_ID                      NOT NULL  VARCHAR2(15)
TEACHER_NAME                     NOT NULL  VARCHAR2(20)
TEACHER_EMAIL                    VARCHAR2(40)
SPECIALIZATION                  NOT NULL  VARCHAR2(40)
```

Figure 15: Describing Teacher table

7.6 Creation and Description of Teach_Mod_Stud table

Creating Teach_Mod_Stud table

```
SQL> CREATE TABLE Teach_Mod_Stud (
 2 Teacher_ID VARCHAR(15) NOT NULL,
 3 Module_ID VARCHAR(15) NOT NULL,
 4 Student_ID VARCHAR(15) NOT NULL,
 5 PRIMARY KEY (Teacher_ID, Student_ID, Module_ID),
 6 FOREIGN KEY (Teacher_ID) REFERENCES Teacher(Teacher_ID),
 7 FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),
 8 FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID);
```

Table created.

Figure 16: Creating Teach_Mod_Stud table

Describing Teach_Mod_Stud table

```
SQL> DESC Teach_Mod_Stud;
Name          Null?    Type
-----        -----   -----
TEACHER_ID    NOT NULL VARCHAR2(15)
MODULE_ID     NOT NULL VARCHAR2(15)
STUDENT_ID    NOT NULL VARCHAR2(15)
```

Figure 17: Describing Teach_Mod_Stud table

7.7 Creation and Description of Announcement table

Creating Announcement table

```
SQL> CREATE TABLE Announcement (
 2  Announcement_ID VARCHAR(20) PRIMARY KEY,
 3  Announcement_Title VARCHAR(30) NOT NULL,
 4  Content VARCHAR(50) NOT NULL,
 5  AnnounceDate DATE NOT NULL);
```

Table created.

Figure 18: Creating Announcement table

Describing Announcement table

```
SQL> DESC Announcement;
Name          Null?    Type
-----        -----
ANNOUNCEMENT_ID      NOT NULL  VARCHAR2(20)
ANNOUNCEMENT_TITLE   NOT NULL  VARCHAR2(30)
CONTENT            NOT NULL  VARCHAR2(50)
ANNOUNCEDATE       NOT NULL  DATE
```

Figure 19: Describing Announcement table

7.8 Creation and Description of Announce_Teach_Mod_Stud table

Creating Announce_Teach_Mod_Stud table

```
SQL> CREATE TABLE Announce_Teach_Mod_Stud (
 2 Announcement_ID VARCHAR(20),
 3 Teacher_ID VARCHAR(15) NOT NULL,
 4 Student_ID VARCHAR(15) NOT NULL,
 5 Module_ID VARCHAR(15) NOT NULL,
 6 PRIMARY KEY (Announcement_ID, Teacher_ID, Student_ID, Module_ID),
 7 FOREIGN KEY (Announcement_ID) REFERENCES Announcement(Announcement_ID),
 8 FOREIGN KEY (Teacher_ID) REFERENCES Teacher(Teacher_ID),
 9 FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),
10 FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID));
```

Table created.

Figure 20: Creating Announce_Teach_Mod_Stud table

Describing Announce_Teach_Mod_Stud table

```
SQL> DESC Announce_Teach_Mod_Stud;
Name          Null?    Type
-----        -----   -----
ANNOUNCEMENT_ID      NOT NULL  VARCHAR2(20)
TEACHER_ID          NOT NULL  VARCHAR2(15)
STUDENT_ID          NOT NULL  VARCHAR2(15)
MODULE_ID           NOT NULL  VARCHAR2(15)
```

Figure 21: Describing Announce_Teach_Mod_Stud table

7.9 Creation and Description of Assessment table

Creating Assessment table

```
SQL> CREATE TABLE Assessment (
  2  Assessment_ID VARCHAR(15) PRIMARY KEY,
  3  Assessment_Title VARCHAR(15) NOT NULL,
  4  Deadline DATE NOT NULL,
  5  Weightage NUMBER(10) NOT NULL);

Table created.
```

Figure 22: Creating Assessment table

Describing Assessment table

```
SQL> DESC Assessment;
Name                           Null?    Type
-----                         ----- 
ASSESSMENT_ID                  NOT NULL VARCHAR2(15)
ASSESSMENT_TITLE                NOT NULL VARCHAR2(15)
DEADLINE                        NOT NULL DATE
WEIGHTAGE                       NOT NULL NUMBER(10)
```

Figure 23: Describing Assessment table

7.10 Creation and Description of Assess_Mod_Stud table

Creating Assess_Mod_Stud table

```
SQL> CREATE TABLE Assess_Mod_Stud (
  2 Assessment_ID VARCHAR(15) NOT NULL,
  3 Module_ID VARCHAR(15) NOT NULL,
  4 Student_ID VARCHAR(15) NOT NULL,
  5 Submission_Status VARCHAR(20) NOT NULL,
  6 Marks_Obtained NUMBER(20) NOT NULL,
  7 Grade VARCHAR(5) NOT NULL,
  8 PRIMARY KEY (Assessment_ID, Student_ID, Module_ID),
  9 FOREIGN KEY (Assessment_ID) REFERENCES Assessment(Assessment_ID),
 10 FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),
 11 FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID);
```

Table created.

Figure 24: Creating Assess_Mod_Stud table

Describing Assess_Mod_Stud table

Name	Null?	Type
ASSESSMENT_ID	NOT NULL	VARCHAR2(15)
MODULE_ID	NOT NULL	VARCHAR2(15)
STUDENT_ID	NOT NULL	VARCHAR2(15)
SUBMISSION_STATUS	NOT NULL	VARCHAR2(20)
MARKS_OBTAINED	NOT NULL	NUMBER(20)
GRADE	NOT NULL	VARCHAR2(5)

Figure 25: Describing Assess_Mod_Stud table

7.11 Creation and Description of Resources table

Creating Resources table

```
SQL> CREATE TABLE Resources (
  2  Resource_ID VARCHAR(15) PRIMARY KEY,
  3  Resource_Name VARCHAR(40) NOT NULL,
  4  Resource_Type VARCHAR(15) NOT NULL,
  5  R_Duration VARCHAR(10));
```

Table created.

Figure 26: Creating Resources table

Describing Resources table

```
SQL> DESC Resources;
      Name          Null?    Type
-----+
RESOURCE_ID           NOT NULL  VARCHAR2(15)
RESOURCE_NAME         NOT NULL  VARCHAR2(40)
RESOURCE_TYPE         NOT NULL  VARCHAR2(15)
R_DURATION            VARCHAR2(10)
```

Figure 27: Describing Resources table

7.12 Creation and Description of Reso_Mod_Stud table

Creating Reso_Mod_Stud table

```
SQL> CREATE TABLE Reso_Mod_Stud (
  2 Resource_ID VARCHAR(15) NOT NULL,
  3 Module_ID VARCHAR(15) NOT NULL,
  4 Student_ID VARCHAR(15) NOT NULL,
  5 ResourceStatus VARCHAR(20) NOT NULL,
  6 PRIMARY KEY (Resource_ID, Student_ID, Module_ID),
  7 FOREIGN KEY (Resource_ID) REFERENCES Resources(Resource_ID),
  8 FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),
  9 FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID));
```

Table created.

Figure 28: Creating Reso_Mod_Stud table

Describing Reso_Mod_Stud table

Name	Null?	Type
RESOURCE_ID	NOT NULL	VARCHAR2(15)
MODULE_ID	NOT NULL	VARCHAR2(15)
STUDENT_ID	NOT NULL	VARCHAR2(15)
RESOURCESSATUS	NOT NULL	VARCHAR2(20)

Figure 29: Describing Reso_Mod_Stud table

7.13 Insertion of Test Data in Program Table

Inserting data in Program table

```
SQL> INSERT INTO Program (Program_ID, Program_Name, Duration, Description)
  2  VALUES ('P101', 'Multimedia', '3 Years', 'VFX, Video Editing');

1 row created.

SQL>
SQL> INSERT INTO Program (Program_ID, Program_Name, Duration, Description)
  2  VALUES ('P102', 'Computing', '4 Years', 'Software Development, IT systems');

1 row created.

SQL>
SQL> INSERT INTO Program (Program_ID, Program_Name, Duration, Description)
  2  VALUES ('P103', 'BBA', '3 Years', 'Management, Finance, Marketing');

1 row created.

SQL>
SQL> INSERT INTO Program (Program_ID, Program_Name, Duration, Description)
  2  VALUES ('P104', 'Computing With AI', '2 Years', 'Artificial Intelligence, Machine Learning');

1 row created.

SQL>
SQL> INSERT INTO Program (Program_ID, Program_Name, Duration, Description)
  2  VALUES ('P105', 'Computer Engineering', '4 Years', 'Hardware Design, Programming, Systems Integration');

1 row created.

SQL>
SQL> INSERT INTO Program (Program_ID, Program_Name, Duration, Description)
  2  VALUES ('P106', 'Networking', '3 Years', 'Network Design, Infrastructure Management');

1 row created.

SQL>
SQL> INSERT INTO Program (Program_ID, Program_Name, Duration, Description)
  2  VALUES ('P107', 'Information Technology', '5 Years', 'Software Development, Project Management');

1 row created.

SQL>
```

Figure 30: Inserting data of Program table

Adjusting the linesize and Pagesize to get proper formatting and Selecting data of Program table

```
SQL> set linesize 600;
SQL> set pagesize 200;
SQL> SELECT * FROM Program;

PROGRAM_ID      PROGRAM_NAME          DURATION  DESCRIPTION
-----          -----
P101            Multimedia           3 Years    VFX, Video Editing
P102            Computing            4 Years    Software Development, IT systems
P103            BBA                 3 Years    Management, Finance, Marketing
P104            Computing With AI   2 Years    Artificial Intelligence, Machine Learning
P105            Computer Engineering 4 Years    Hardware Design, Programming, Systems Integration
P106            Networking           3 Years    Network Design, Infrastructure Management
P107            Information Technology 5 Years   Software Development, Project Management

7 rows selected.
```

Figure 31: Selecting data of Program table

7.14 Insertion of Test Data in Student Table

Inserting data in Student table

```

SQL> INSERT INTO Student (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID)
  2  VALUES ('NP012001', 'Manan Chataut', 'Mahendranagar', 9812345678, 'manan_chatauti@gmail.com', TO_DATE('2024-01-15', 'YYYY-MM-DD'), 'P101');
1 row created.

SQL>
SQL> INSERT INTO Student (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID)
  2  VALUES ('NP012002', 'Anil Joshi', 'Nepalgunj', 9889012375, 'anil_joshi45@gmail.com', TO_DATE('2024-04-10', 'YYYY-MM-DD'), 'P106');
1 row created.

SQL>
SQL> INSERT INTO Student (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID)
  2  VALUES ('NP012003', 'Aditya Bhatta', 'Butwal', 9823456789, 'aditya_bhatta12@gmail.com', TO_DATE('2024-07-20', 'YYYY-MM-DD'), 'P102');
1 row created.

SQL>
SQL> INSERT INTO Student (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID)
  2  VALUES ('NP012004', 'Aryan Bhandari', 'Lalitpur', 9856789012, 'aryan_bhandari911@gmail.com', TO_DATE('2024-11-15', 'YYYY-MM-DD'), 'P102');
1 row created.

SQL>
SQL> INSERT INTO Student (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID)
  2  VALUES ('NP012005', 'Bhuwan Bhatta', 'Dhangadi', 9834567890, 'bhuwan_bhatta123@gmail.com', TO_DATE('2024-05-10', 'YYYY-MM-DD'), 'P103');
1 row created.

SQL>
SQL> INSERT INTO Student (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID)
  2  VALUES ('NP012006', 'Sakshyam Kafley', 'Butwal', 9889012385, 'sakshyam_kafley78@gmail.com', TO_DATE('2024-05-07', 'YYYY-MM-DD'), 'P103');
1 row created.

SQL>
SQL> INSERT INTO Student (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID)
  2  VALUES ('NP012007', 'Pranay Shakya', 'Chitwan', 9889012345, 'pranay_shakya15@gmail.com', TO_DATE('2024-06-10', 'YYYY-MM-DD'), 'P107');
1 row created.

SQL>
SQL> INSERT INTO Student (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID)
  2  VALUES ('NP012008', 'Aakar Singh', 'Kathmandu', 9845678901, 'aakar_singh911@gmail.com', TO_DATE('2024-03-25', 'YYYY-MM-DD'), 'P104');
1 row created.

SQL>
SQL> INSERT INTO Student (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID)
  2  VALUES ('NP012009', 'Shrine Ghimire', 'Chitwan', 9889012365, 'shrine_ghimire23@gmail.com', TO_DATE('2024-03-20', 'YYYY-MM-DD'), 'P105');
1 row created.

SQL>
SQL> INSERT INTO Student (Student_ID, Name, Address, Contact, Email, JoinedDate, Program_ID)
  2  VALUES ('NP012010', 'Najib Thapa', 'Bhaktapur', 9867890123, 'najib_thapa101@gmail.com', TO_DATE('2024-08-05', 'YYYY-MM-DD'), 'P106');
1 row created.

```

Figure 32: Inserting data of Student table

Selecting data of Student table

STUDENT_ID	NAME	ADDRESS	CONTACT	EMAIL	JOINEDDATE	PROGRAM_ID
NP012001	Manan Chataut	Mahendranagar	9812345678	manan_chatauti@gmail.com	15-JAN-24	P101
NP012002	Anil Joshi	Nepalgunj	9889012375	anil_joshi45@gmail.com	10-APR-24	P106
NP012003	Aditya Bhatta	Butwal	9823456789	aditya_bhatta12@gmail.com	20-JUL-24	P102
NP012004	Aryan Bhandari	Lalitpur	9856789012	aryan_bhandari911@gmail.com	15-NOV-24	P102
NP012005	Bhuwan Bhatta	Dhangadi	9834567890	bhuwan_bhatta123@gmail.com	10-MAY-24	P103
NP012006	Sakshyam Kafley	Butwal	9889012385	sakshyam_kafley78@gmail.com	07-MAY-24	P103
NP012007	Pranay Shakya	Chitwan	9889012345	pranay_shakya15@gmail.com	10-JUN-24	P107
NP012008	Aakar Singh	Kathmandu	9845678901	aakar_singh911@gmail.com	25-MAR-24	P104
NP012009	Shrine Ghimire	Chitwan	9889012365	shrine_ghimire23@gmail.com	20-MAR-24	P105
NP012010	Najib Thapa	Bhaktapur	9867890123	najib_thapa101@gmail.com	05-AUG-24	P106

Figure 33: Selecting data of Student table

7.15 Insertion of Test Data in Module Table

Inserting data in Module table

```

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Name, Credit_Hours)
  2  VALUES ('MOD301', 'VFX Design', 13);

1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Name, Credit_Hours)
  2  VALUES ('MOD302', 'Video Editing Techniques', 15);

1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Name, Credit_Hours)
  2  VALUES ('MOD303', 'DataBase', 20);

1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Name, Credit_Hours)
  2  VALUES ('MOD304', 'Marketing Essentials', 10);

1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Name, Credit_Hours)
  2  VALUES ('MOD305', 'Artificial Intelligence Fundamentals', 10);

1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Name, Credit_Hours)
  2  VALUES ('MOD306', 'Digital Electronics', 10);

1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Name, Credit_Hours)
  2  VALUES ('MOD307', 'Cyber Security Fundamentals', 18);

1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Name, Credit_Hours)
  2  VALUES ('MOD308', 'IT Systems and Architecture', 15);

1 row created.

```

Figure 34: Inserting data of Module table

Selecting data of Module table

```

SQL> SELECT * FROM Module;

MODULE_ID      MODULE_NAME          CREDIT_HOURS
-----  -----
MOD301        VFX Design            13
MOD302        Video Editing Techniques    15
MOD303        DataBase              20
MOD304        Marketing Essentials    10
MOD305        Artificial Intelligence Fundamentals 10
MOD306        Digital Electronics     10
MOD307        Cyber Security Fundamentals 18
MOD308        IT Systems and Architecture 15

8 rows selected.

```

Figure 35: Selecting data of Module table

7.16 Insertion of Test Data in Mod_Stud Table

Inserting data in Mod_Stud table

```
SQL>
SQL> INSERT INTO Mod_Stud (Module_ID, Student_ID)
  2  VALUES ('MOD301', 'NP012001');

1 row created.

SQL>
SQL> INSERT INTO Mod_Stud (Module_ID, Student_ID)
  2  VALUES ('MOD302', 'NP012001');

1 row created.

SQL>
SQL> INSERT INTO Mod_Stud (Module_ID, Student_ID)
  2  VALUES ('MOD307', 'NP012002');

1 row created.

SQL>
SQL> INSERT INTO Mod_Stud (Module_ID, Student_ID)
  2  VALUES ('MOD303', 'NP012003');

1 row created.

SQL>
SQL> INSERT INTO Mod_Stud (Module_ID, Student_ID)
  2  VALUES ('MOD303', 'NP012004');

1 row created.

SQL>
SQL> INSERT INTO Mod_Stud (Module_ID, Student_ID)
  2  VALUES ('MOD304', 'NP012005');

1 row created.

SQL>
SQL> INSERT INTO Mod_Stud (Module_ID, Student_ID)
  2  VALUES ('MOD304', 'NP012006');

1 row created.

SQL>
SQL> INSERT INTO Mod_Stud (Module_ID, Student_ID)
  2  VALUES ('MOD308', 'NP012007');

1 row created.

SQL>
SQL> INSERT INTO Mod_Stud (Module_ID, Student_ID)
  2  VALUES ('MOD305', 'NP012008');

1 row created.

SQL>
SQL> INSERT INTO Mod_Stud (Module_ID, Student_ID)
  2  VALUES ('MOD306', 'NP012009');

1 row created.

SQL>
SQL> INSERT INTO Mod_Stud (Module_ID, Student_ID)
  2  VALUES ('MOD307', 'NP012010');

1 row created.
```

Figure 36: Inserting data of Mod_Stud table

Selecting data of Mod_Stud table

```
SQL> SELECT * FROM Mod_Stud;

MODULE_ID          STUDENT_ID
-----  -----
MOD301              NP012001
MOD302              NP012001
MOD307              NP012002
MOD303              NP012003
MOD303              NP012004
MOD304              NP012005
MOD304              NP012006
MOD308              NP012007
MOD305              NP012008
MOD306              NP012009
MOD307              NP012010

11 rows selected.
```

Figure 37: Selecting data of Mod_Stud table

7.17 Insertion of Test Data in Teacher Table

Inserting data in Teacher table

```

SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Teacher_Email, Specialization)
  2  VALUES ('T401', 'Sujan Sakya', 'Sujan_sakya401@gmail.com', 'VFX Design');

1 row created.

SQL>
SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Teacher_Email, Specialization)
  2  VALUES ('T402', 'Jivan Pandey', 'jivan_pandey402@gmail.com', 'Video Editing Techniques');

1 row created.

SQL>
SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Teacher_Email, Specialization)
  2  VALUES ('T403', 'Bijay Gautam', 'bijay_gautam403@gmail.com', 'DataBase');

1 row created.

SQL>
SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Teacher_Email, Specialization)
  2  VALUES ('T404', 'Shresha Bhandari', 'shresha_bhandari404@gmail.com', 'Marketing Essentials');

1 row created.

SQL>
SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Teacher_Email, Specialization)
  2  VALUES ('T405', 'Prashun Ayer', 'prashun_ayer405@gmail.com', 'Artificial Intelligence Fundamentals');

1 row created.

SQL>
SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Teacher_Email, Specialization)
  2  VALUES ('T406', 'Bishnu Pandey', 'bishnu_pandey406@gmail.com', 'Digital Electronics');

1 row created.

SQL>
SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Teacher_Email, Specialization)
  2  VALUES ('T407', 'Ujal Bhatta', 'ujal_bhatta406@gmail.com', 'Cyber Security Fundamentals');

1 row created.

SQL>
SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Teacher_Email, Specialization)
  2  VALUES ('T408', 'Dipeshor Silwal', 'dipeshor_silwal407@gmail.com', 'IT Systems and Architecture');

1 row created.

SQL>
SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Teacher_Email, Specialization)
  2  VALUES ('T409', 'Tejaswi Joshi', 'Tejaswi_Joshi407@gmail.com', 'DataBase');

1 row created.

```

Figure 38: Inserting data of Teacher table

Selecting data of Teacher table

TEACHER_ID	TEACHER_NAME	TEACHER_EMAIL	SPECIALIZATION
T401	Sujan Sakya	Sujan_sakya401@gmail.com	VFX Design
T402	Jivan Pandey	jivan_pandey402@gmail.com	Video Editing Techniques
T403	Bijay Gautam	bijay_gautam403@gmail.com	DataBase
T404	Shresha Bhandari	shresha_bhandari404@gmail.com	Marketing Essentials
T405	Prashun Ayer	prashun_ayer405@gmail.com	Artificial Intelligence Fundamentals
T406	Bishnu Pandey	bishnu_pandey406@gmail.com	Digital Electronics
T407	Ujal Bhatta	ujal_bhatta406@gmail.com	Cyber Security Fundamentals
T408	Dipeshor Silwal	dipeshor_silwal407@gmail.com	IT Systems and Architecture
T409	Tejaswi Joshi	Tejaswi_Joshi407@gmail.com	DataBase

9 rows selected.

Figure 39: Selecting data of Teacher table

7.18 Insertion of Test Data in Teach_Mod_Stud Table

Inserting data in Teach_Mod_Stud table

```

SQL> INSERT INTO Teach_Mod_Stud (Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('T401', 'MOD301', 'NP012001');

1 row created.

SQL>
SQL> INSERT INTO Teach_Mod_Stud (Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('T401', 'MOD302', 'NP012001');

1 row created.

SQL>
SQL> INSERT INTO Teach_Mod_Stud (Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('T402', 'MOD302', 'NP012001');

1 row created.

SQL>
SQL> INSERT INTO Teach_Mod_Stud (Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('T403', 'MOD303', 'NP012003');

1 row created.

SQL>
SQL> INSERT INTO Teach_Mod_Stud (Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('T404', 'MOD304', 'NP012005');

1 row created.

SQL>
SQL> INSERT INTO Teach_Mod_Stud (Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('T405', 'MOD305', 'NP012008');

1 row created.

SQL>
SQL> INSERT INTO Teach_Mod_Stud (Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('T406', 'MOD306', 'NP012009');

1 row created.

SQL>
SQL> INSERT INTO Teach_Mod_Stud (Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('T407', 'MOD307', 'NP012010');

1 row created.

SQL>
SQL> INSERT INTO Teach_Mod_Stud (Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('T408', 'MOD308', 'NP012007');

1 row created.

SQL>
SQL> INSERT INTO Teach_Mod_Stud (Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('T409', 'MOD303', 'NP012003');

1 row created.

```

Figure 40: Inserting data of Teach_Mod_Stud table

Selecting data of Teach_Mod_Stud table

```
SQL> SELECT * FROM Teach_Mod_Stud;
```

TEACHER_ID	MODULE_ID	STUDENT_ID
T401	MOD301	NP012001
T401	MOD302	NP012001
T402	MOD302	NP012001
T403	MOD303	NP012003
T404	MOD304	NP012005
T405	MOD305	NP012008
T406	MOD306	NP012009
T407	MOD307	NP012010
T408	MOD308	NP012007
T409	MOD303	NP012003

10 rows selected.

Figure 41: Selecting data of Teach_Mod_Stud table

7.19 Insertion of Test Data in Announcement Table

Inserting data in Announcement table

```

SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Content, AnnounceDate)
  2  VALUES ('AN701', 'VFX Workshop', 'VFX Design techniques, applications', TO_DATE('2024-05-05', 'YYYY-MM-DD'));

1 row created.

SQL>
SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Content, AnnounceDate)
  2  VALUES ('AN702', 'Video Editing Session', 'Training for video editing', TO_DATE('2024-05-28', 'YYYY-MM-DD'));

1 row created.

SQL>
SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Content, AnnounceDate)
  2  VALUES ('AN703', 'Marketing Seminar', 'Essential Marketing strategies', TO_DATE('2024-10-10', 'YYYY-MM-DD'));

1 row created.

SQL>
SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Content, AnnounceDate)
  2  VALUES ('AN704', 'Database WorkShop', 'Learn Concepts Of Oracle', TO_DATE('2024-09-21', 'YYYY-MM-DD'));

1 row created.

SQL>
SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Content, AnnounceDate)
  2  VALUES ('AN705', 'AI Fundamentals Class', 'AI and its fundamentals', TO_DATE('2024-05-20', 'YYYY-MM-DD'));

1 row created.

SQL>
SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Content, AnnounceDate)
  2  VALUES ('AN706', 'Digital Electronics Workshop', 'Digital electronics, circuit design', TO_DATE('2024-06-30', 'YYYY-MM-DD'));

1 row created.

SQL>
SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Content, AnnounceDate)
  2  VALUES ('AN707', 'Cyber Security Discussion', 'Fundamentals of cyber security', TO_DATE('2024-07-25', 'YYYY-MM-DD'));

1 row created.

SQL>
SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Content, AnnounceDate)
  2  VALUES ('AN708', 'IT Systems and Architecture', 'Training for IT systems Week1', TO_DATE('2024-05-15', 'YYYY-MM-DD'));

1 row created.

SQL>
SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Content, AnnounceDate)
  2  VALUES ('AN709', 'IT Systems and Architecture', 'Training for IT Architecture Week2', TO_DATE('2024-01-25', 'YYYY-MM-DD'));

1 row created.

```

Figure 42: Inserting data of Announcement table

Selecting data of Announcement table

SQL> SELECT * FROM Announcement;			
ANNOUNCEMENT_ID	ANNOUNCEMENT_TITLE	CONTENT	ANNOUNCED
AN701	VFX Workshop	VFX Design techniques, applications	05-MAY-24
AN702	Video Editing Session	Training for video editing	28-MAY-24
AN703	Marketing Seminar	Essential Marketing strategies	10-OCT-24
AN704	Database WorkShop	Learn Concepts Of Oracle	21-SEP-24
AN705	AI Fundamentals Class	AI and its fundamentals	20-MAY-24
AN706	Digital Electronics Workshop	Digital electronics, circuit design	30-JUN-24
AN707	Cyber Security Discussion	Fundamentals of cyber security	25-JUL-24
AN708	IT Systems and Architecture	Training for IT systems Week1	15-MAY-24
AN709	IT Systems and Architecture	Training for IT Architecture Week2	25-JAN-24

9 rows selected.

Figure 43: Selecting data of Announcement table

7.20 Insertion of Test Data in Announce_Teach_Mod_Stud Table

Inserting data in Announce_Teach_Mod_Stud table

```

SQL> INSERT INTO Announce_Teach_Mod_Stud (Announcement_ID, Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('AN701', 'T401', 'MOD301', 'NP012001');
1 row created.

SQL>
SQL> INSERT INTO Announce_Teach_Mod_Stud (Announcement_ID, Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('AN702', 'T402', 'MOD302', 'NP012001');
1 row created.

SQL>
SQL> INSERT INTO Announce_Teach_Mod_Stud (Announcement_ID, Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('AN703', 'T403', 'MOD303', 'NP012003');
1 row created.

SQL>
SQL> INSERT INTO Announce_Teach_Mod_Stud (Announcement_ID, Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('AN704', 'T404', 'MOD304', 'NP012005');
1 row created.

SQL>
SQL> INSERT INTO Announce_Teach_Mod_Stud (Announcement_ID, Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('AN705', 'T405', 'MOD305', 'NP012008');
1 row created.

SQL>
SQL> INSERT INTO Announce_Teach_Mod_Stud (Announcement_ID, Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('AN706', 'T406', 'MOD306', 'NP012009');
1 row created.

SQL>
SQL> INSERT INTO Announce_Teach_Mod_Stud (Announcement_ID, Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('AN707', 'T407', 'MOD307', 'NP012010');
1 row created.

SQL>
SQL> INSERT INTO Announce_Teach_Mod_Stud (Announcement_ID, Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('AN708', 'T408', 'MOD308', 'NP012007');
1 row created.

SQL>
SQL> INSERT INTO Announce_Teach_Mod_Stud (Announcement_ID, Teacher_ID, Module_ID, Student_ID)
  2  VALUES ('AN709', 'T408', 'MOD308', 'NP012007');
1 row created.

```

Figure 44: Inserting data of Announce_Teach_Mod_Stud table

Selecting data of Announce_Teach_Mod_Stud table

```

SQL> SELECT * FROM Announce_Teach_Mod_Stud;
-----+
ANNOUNCEMENT_ID      TEACHER_ID      STUDENT_ID      MODULE_ID
-----+
AN701                T401           NP012001        MOD301
AN702                T402           NP012001        MOD302
AN703                T403           NP012003        MOD303
AN704                T404           NP012005        MOD304
AN705                T405           NP012008        MOD305
AN706                T406           NP012009        MOD306
AN707                T407           NP012010        MOD307
AN708                T408           NP012007        MOD308
AN709                T408           NP012007        MOD308
-----+
9 rows selected.

```

Figure 45: Selecting data of Announce_Teach_Mod_Stud table

7.21 Insertion of Test Data in Assessment Table

Inserting data in Assessment table

```
-->
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES ('ASS901', 'Class Test', TO_DATE('2024-02-10', 'YYYY-MM-DD'), 20);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES ('ASS902', 'Practical Exam', TO_DATE('2024-05-25', 'YYYY-MM-DD'), 60);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES ('ASS903', 'Project', TO_DATE('2024-04-15', 'YYYY-MM-DD'), 50);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES ('ASS904', 'Quiz', TO_DATE('2024-03-10', 'YYYY-MM-DD'), 10);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES ('ASS905', 'Assignment', TO_DATE('2024-03-05', 'YYYY-MM-DD'), 30);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES ('ASS906', 'Practical Exam', TO_DATE('2024-04-20', 'YYYY-MM-DD'), 40);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES ('ASS907', 'Mid-Term Exam', TO_DATE('2024-06-10', 'YYYY-MM-DD'), 70);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES ('ASS908', 'Final Exam', TO_DATE('2024-07-25', 'YYYY-MM-DD'), 80);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES ('ASS909', 'Written Test', TO_DATE('2024-05-20', 'YYYY-MM-DD'), 40);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES ('ASS910', 'Practical Test', TO_DATE('2024-02-20', 'YYYY-MM-DD'), 40);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES ('ASS911', 'Written Test', TO_DATE('2024-02-25', 'YYYY-MM-DD'), 40);
1 row created.
```

Figure 46: Inserting data of Assessment table

Selecting data of Assessment table

```
SQL> SELECT * FROM Assessment;
```

ASSESSMENT_ID	ASSESSMENT_TITL	DEADLINE	WEIGHTAGE
ASS901	Class Test	10-FEB-24	20
ASS902	Practical Exam	25-MAY-24	60
ASS903	Project	15-APR-24	50
ASS904	Quiz	10-MAR-24	10
ASS905	Assignment	05-MAR-24	30
ASS906	Practical Exam	20-APR-24	40
ASS907	Mid-Term Exam	10-JUN-24	70
ASS908	Final Exam	25-JUL-24	80
ASS909	Written Test	20-MAY-24	40
ASS910	Practical Test	20-FEB-24	40
ASS911	Written Test	25-FEB-24	40

```
11 rows selected.
```

Figure 47: Selecting data of Assessment table

7.22 Insertion of Test Data in Assess_Mod_Stud Table

Inserting data in Assess_Mod_Stud table

```

SQL> INSERT INTO Assess_Mod_Stud (Assessment_ID, Module_ID, Student_ID, Submission_Status, Marks_Obtained, Grade)
  2  VALUES ('ASS901', 'MOD301', 'NP012001', 'Not Submitted', 0, 'N/A');

1 row created.

SQL>
SQL> INSERT INTO Assess_Mod_Stud (Assessment_ID, Module_ID, Student_ID, Submission_Status, Marks_Obtained, Grade)
  2  VALUES ('ASS902', 'MOD307', 'NP012002', 'Submitted', 91, 'A');

1 row created.

SQL>
SQL> INSERT INTO Assess_Mod_Stud (Assessment_ID, Module_ID, Student_ID, Submission_Status, Marks_Obtained, Grade)
  2  VALUES ('ASS903', 'MOD303', 'NP012003', 'Not Submitted', 0, 'N/A');

1 row created.

SQL>
SQL> INSERT INTO Assess_Mod_Stud (Assessment_ID, Module_ID, Student_ID, Submission_Status, Marks_Obtained, Grade)
  2  VALUES ('ASS904', 'MOD303', 'NP012004', 'Submitted', 65, 'B');

1 row created.

SQL>
SQL> INSERT INTO Assess_Mod_Stud (Assessment_ID, Module_ID, Student_ID, Submission_Status, Marks_Obtained, Grade)
  2  VALUES ('ASS905', 'MOD304', 'NP012006', 'Submitted', 80, 'A');

1 row created.

SQL>
SQL> INSERT INTO Assess_Mod_Stud (Assessment_ID, Module_ID, Student_ID, Submission_Status, Marks_Obtained, Grade)
  2  VALUES ('ASS906', 'MOD308', 'NP012007', 'Submitted', 59, 'C');

1 row created.

SQL>
SQL> INSERT INTO Assess_Mod_Stud (Assessment_ID, Module_ID, Student_ID, Submission_Status, Marks_Obtained, Grade)
  2  VALUES ('ASS907', 'MOD305', 'NP012008', 'Submitted', 39, 'E');

1 row created.

SQL>
SQL> INSERT INTO Assess_Mod_Stud (Assessment_ID, Module_ID, Student_ID, Submission_Status, Marks_Obtained, Grade)
  2  VALUES ('ASS908', 'MOD306', 'NP012009', 'Submitted', 89, 'A');

1 row created.

SQL>
SQL> INSERT INTO Assess_Mod_Stud (Assessment_ID, Module_ID, Student_ID, Submission_Status, Marks_Obtained, Grade)
  2  VALUES ('ASS909', 'MOD302', 'NP012001', 'Submitted', 70, 'A');

1 row created.

SQL>
SQL> INSERT INTO Assess_Mod_Stud (Assessment_ID, Module_ID, Student_ID, Submission_Status, Marks_Obtained, Grade)
  2  VALUES ('ASS910', 'MOD301', 'NP012010', 'Submitted', 70, 'A');

1 row created.

SQL>
SQL> INSERT INTO Assess_Mod_Stud (Assessment_ID, Module_ID, Student_ID, Submission_Status, Marks_Obtained, Grade)
  2  VALUES ('ASS911', 'MOD301', 'NP012005', 'Not Submitted', 0, 'N/A');

1 row created.

SQL>
```

Figure 48: Inserting data of Assess_Mod_Stud table

Selecting data of Assess_Mod_Stud table

ASSESSMENT_ID	MODULE_ID	STUDENT_ID	SUBMISSION_STATUS	MARKS_OBTAINED	GRADE
ASS901	MOD301	NP012001	Not Submitted	0	N/A
ASS902	MOD307	NP012002	Submitted	91	A
ASS903	MOD303	NP012003	Not Submitted	0	N/A
ASS904	MOD303	NP012004	Submitted	65	B
ASS905	MOD304	NP012006	Submitted	80	A
ASS906	MOD308	NP012007	Submitted	59	C
ASS907	MOD305	NP012008	Submitted	39	E
ASS908	MOD306	NP012009	Submitted	89	A
ASS909	MOD302	NP012001	Submitted	70	A
ASS910	MOD301	NP012010	Submitted	70	A
ASS911	MOD301	NP012005	Not Submitted	0	N/A

11 rows selected.

Figure 49: Selecting data of Assess_Mod_Stud table

7.23 Insertion of Test Data in Resources Table

Inserting data in Resources table

```

SQL> INSERT INTO Resources (Resource_ID, Resource_Name, Resource_Type, R_Duration)
  2  VALUES ('R1001', 'VFX Design Guide', 'PDF', '3 Months');

1 row created.

SQL>
SQL> INSERT INTO Resources (Resource_ID, Resource_Name, Resource_Type, R_Duration)
  2  VALUES ('R1002', 'Video Editing Tools', 'Video', '1 Month');

1 row created.

SQL>
SQL> INSERT INTO Resources (Resource_ID, Resource_Name, Resource_Type, R_Duration)
  2  VALUES ('R1003', 'Programming Basics', 'E-Book', '2 Months');

1 row created.

SQL>
SQL> INSERT INTO Resources (Resource_ID, Resource_Name, Resource_Type, R_Duration)
  2  VALUES ('R1004', 'Marketing, Accounts Essentials', 'Excel Sheet', '1 Month');

1 row created.

SQL>
SQL> INSERT INTO Resources (Resource_ID, Resource_Name, Resource_Type, R_Duration)
  2  VALUES ('R1005', 'AI Fundamentals Resources', 'Video', '2 Months');

1 row created.

SQL>
SQL> INSERT INTO Resources (Resource_ID, Resource_Name, Resource_Type, R_Duration)
  2  VALUES ('R1006', 'Digital Electronics E-Book', 'E-Book', '4 Months');

1 row created.

SQL>
SQL> INSERT INTO Resources (Resource_ID, Resource_Name, Resource_Type, R_Duration)
  2  VALUES ('R1007', 'Cyber Security Guide', 'PDF', '6 Months');

1 row created.

SQL>
SQL> INSERT INTO Resources (Resource_ID, Resource_Name, Resource_Type, R_Duration)
  2  VALUES ('R1008', 'IT Systems, Architecture Book', 'Book', '1 Month');

1 row created.

```

Figure 50: Inserting data of Resources table

Selecting data of Resources table

```

SQL> select *from Resources;

RESOURCE_ID      RESOURCE_NAME          RESOURCE_TYPE    R_DURATION
-----          -----
R1001            VFX Design Guide        PDF             3 Months
R1002            Video Editing Tools     Video           1 Month
R1003            Programming Basics      E-Book          2 Months
R1004            Marketing, Accounts Essentials   Excel Sheet  1 Month
R1005            AI Fundamentals Resources  Video           2 Months
R1006            Digital Electronics E-Book  E-Book          4 Months
R1007            Cyber Security Guide     PDF             6 Months
R1008            IT Systems, Architecture Book Book            1 Month

8 rows selected.

SQL>

```

Figure 51: Selecting data of Resources table

7.24 Insertion of Test Data in Reso_Mod_Stud Table

Inserting data in Reso_Mod_Stud table

```

SQL> INSERT INTO Reso_Mod_Stud (Resource_ID, Student_ID, Module_ID, ResourceStatus)
  2  VALUES ('R1001', 'NP012001', 'MOD301', 'Completed');

1 row created.

SQL>
SQL> INSERT INTO Reso_Mod_Stud (Resource_ID, Student_ID, Module_ID, ResourceStatus)
  2  VALUES ('R1002', 'NP012001', 'MOD302', 'In Progress');

1 row created.

SQL>
SQL> INSERT INTO Reso_Mod_Stud (Resource_ID, Student_ID, Module_ID, ResourceStatus)
  2  VALUES ('R1007', 'NP012002', 'MOD307', 'Pending');

1 row created.

SQL>
SQL> INSERT INTO Reso_Mod_Stud (Resource_ID, Student_ID, Module_ID, ResourceStatus)
  2  VALUES ('R1003', 'NP012003', 'MOD303', 'Pending');

1 row created.

SQL>
SQL> INSERT INTO Reso_Mod_Stud (Resource_ID, Student_ID, Module_ID, ResourceStatus)
  2  VALUES ('R1003', 'NP012004', 'MOD303', 'Pending');

1 row created.

SQL>
SQL> INSERT INTO Reso_Mod_Stud (Resource_ID, Student_ID, Module_ID, ResourceStatus)
  2  VALUES ('R1004', 'NP012005', 'MOD304', 'Completed');

1 row created.

SQL>
SQL> INSERT INTO Reso_Mod_Stud (Resource_ID, Student_ID, Module_ID, ResourceStatus)
  2  VALUES ('R1004', 'NP012006', 'MOD304', 'Completed');

1 row created.

SQL>
SQL> INSERT INTO Reso_Mod_Stud (Resource_ID, Student_ID, Module_ID, ResourceStatus)
  2  VALUES ('R1008', 'NP012007', 'MOD308', 'In Progress');

1 row created.

SQL>
SQL> INSERT INTO Reso_Mod_Stud (Resource_ID, Student_ID, Module_ID, ResourceStatus)
  2  VALUES ('R1005', 'NP012008', 'MOD305', 'In Progress');

1 row created.

SQL>
SQL> INSERT INTO Reso_Mod_Stud (Resource_ID, Student_ID, Module_ID, ResourceStatus)
  2  VALUES ('R1006', 'NP012009', 'MOD306', 'Completed');

1 row created.

SQL>
SQL> INSERT INTO Reso_Mod_Stud (Resource_ID, Student_ID, Module_ID, ResourceStatus)
  2  VALUES ('R1007', 'NP012010', 'MOD307', 'Pending');

1 row created.
```

Figure 52: Inserting data of Reso_Mod_Stud table

Selecting data of Reso_Mod_Stud table

```
SQL> SELECT * FROM Reso_Mod_Stud;

+-----+-----+-----+-----+
| RESOURCE_ID | MODULE_ID | STUDENT_ID | RESOURCESTATUS |
+-----+-----+-----+-----+
| R1001 | MOD301 | NP012001 | Completed |
| R1002 | MOD302 | NP012001 | In Progress |
| R1007 | MOD307 | NP012002 | Pending |
| R1003 | MOD303 | NP012003 | Pending |
| R1003 | MOD303 | NP012004 | Pending |
| R1004 | MOD304 | NP012005 | Completed |
| R1004 | MOD304 | NP012006 | Completed |
| R1008 | MOD308 | NP012007 | In Progress |
| R1005 | MOD305 | NP012008 | In Progress |
| R1006 | MOD306 | NP012009 | Completed |
| R1007 | MOD307 | NP012010 | Pending |

11 rows selected.

SQL> |
```

Figure 53: Selecting data of Reso_Mod_Stud table

8. Database Querying

8.1 Information Query

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

```
SQL> SELECT p.Program_Name,
  2 count(s.Student_ID) as Students from Program p
  3 left join Student s ON p.Program_ID = s.Program_ID
  4 group by p.Program_Name;

PROGRAM_NAME          STUDENTS
-----              -----
Networking                  2
Information Technology      1
Computing With AI          1
Computer Engineering        1
Computing                   2
Multimedia                  1
BBA                         2

7 rows selected.
```

Figure 54: Listing Available Programs and Students

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

```
SQL> SELECT a.Announcement_Title, a.Content,
  2 a.AnnounceDate from Announcement a
  3 Join Announce_Teach_Mod_Stud ats ON a.Announcement_ID = ats.Announcement_ID
  4 Where a.AnnounceDate
  5 Between TO_DATE('2024-05-01', 'YYYY-MM-DD') and TO_DATE('2024-05-28', 'YYYY-MM-DD')
  6 order by a.AnnounceDate;

ANNOUNCEMENT_TITLE          CONTENT          ANNOUNCED
-----          -----          -----
VFX Workshop                VFX Design techniques, applications    05-MAY-24
IT Systems and Architecture Training for IT systems Week1    15-MAY-24
AI Fundamentals Class       AI and its fundamentals            20-MAY-24
Video Editing Session        Training for video editing           28-MAY-24
```

Figure 55: Listing Announcement for selected date

8.1.3 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 m.Module_Name,
  2 count(r.Resource_ID) as Resource_Count from Module m
  3 Left Join Reso_Mod_Stud r ON m.Module_ID = r.Module_ID
  4 Where m.Module_Name LIKE 'D%'
  5 group by m.Module_Name;
```

MODULE_NAME	RESOURCE_COUNT
DataBase	2
Digital Electronics	1

Figure 56: Listing all Modules that begin with the letter 'D'

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

```
SQL> SELECT s.Name AS StudentName, p.Program_Name From Student s
  2 Join Program p ON s.Program_ID = p.Program_ID Where Exists ( Select 1 FROM Assess_Mod_Stud ams
  3 Join Module m on ams.Module_ID = m.Module_ID Where ams.Student_ID = s.Student_ID
  4 And amsSubmission_Status = 'Not Submitted' And m.Module_Name = 'VFX Design');
```

STUDENTNAME	PROGRAM_NAME
Bhuwan Bhatta	BBA
Manan Chataut	Multimedia

Figure 57: Listing Students name who have not submitted Assessment

8.1.5 List all the teachers who teach more than one module.

```
SQL> SELECT t.Teacher_ID, t.Teacher_Name
  2 From Teacher t
  3 Join Teach_Mod_Stud tms ON t.Teacher_ID = tms.Teacher_ID
  4 Group by t.Teacher_ID, t.Teacher_Name having COUNT(DISTINCT tms.Module_ID) > 1;
```

TEACHER_ID	TEACHER_NAME
T401	Sujan Sakya

Figure 58: Listing Teacher who teach more than one module

8.2 Transaction Query

8.2.1 Identify the module that has the latest assessment deadline.

SQL> SELECT ams.Module_ID, m.Module_Name, a.Deadline		
MODULE_ID	MODULE_NAME	DEADLINE
MOD306	Digital Electronics	25-JUL-24

Figure 59: Listing module that has latest assessment deadline

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

SQL> SELECT Student_ID, Student_Name, Total_Score		
STUDENT_ID	STUDENT_NAME	TOTAL_SCORE
NP012002	Anil Joshi	91
NP012009	Shrine Ghimire	89
NP012006	Sakshyam Kafley	80

Figure 60: Listing top three student with highest marks

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

```
SQL> SELECT p.Program_Name,
  2 Count(a.Assessment_ID) AS TotalAssessments, AVG(ams.Marks_Obtained) AS AverageMarks
  3 From Program p
  4 Join Student s ON p.Program_ID = s.Program_ID
  5 Join Assess_Mod_Stud ams ON s.Student_ID = ams.Student_ID
  6 Join Assessment a ON ams.Assessment_ID = a.Assessment_ID GROUP BY p.Program_Name;

PROGRAM_NAME          TOTALASSESSMENTS  AVERAGEMARKS
-----  -----  -----
Networking                  2            80.5
Information Technology      1            59
Computing With AI           1            39
Computer Engineering        1            89
Computing                   2            32.5
Multimedia                  2            35
BBA                         2            40

7 rows selected.
```

Figure 61: Listing program with number of assessment and average score

8.2.4 List the students who have scored above the average score in the ‘Databases’ module.

```
SQL> SELECT s.Name AS StudentName, ams.Marks_Obtained
  2 From Student s
  3 Join Assess_Mod_Stud ams ON s.Student_ID = ams.Student_ID
  4 Join Module m ON ams.Module_ID = m.Module_ID
  5 Where m.Module_Name = 'DataBase' AND
  6 ams.Marks_Obtained > (SELECT AVG(ams1.Marks_Obtained)
  7 From Assess_Mod_Stud ams1
  8 Join Module m1 ON ams1.Module_ID = m1.Module_ID
  9 Where m1.Module_Name = 'DataBase');

STUDENTNAME          MARKS_OBTAINED
-----  -----
Aryan Bhandari                65
```

Figure 62: Listing student with score of above average

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

```
SQL> SELECT s.Name as Student_Name, m.Module_Name AS Module_Name,
  2 case when SUM(ams.Marks_Obtained) >= 40 then 'Pass'
  3 Else 'Fail' END as Remarks from Student s
  4 Join Assess_Mod_Stud ams on s.Student_ID = ams.Student_ID
  5 Join Module m ON ams.Module_ID = m.Module_ID
  6 Where m.Module_Name = 'DataBase' Group by s.Name, m.Module_Name;
```

STUDENT_NAME	MODULE_NAME	REMA
Aditya Bhatta	DataBase	Fail
Aryan Bhandari	DataBase	Pass

```
SQL>
```

Figure 63: Listing Remarks of student in a particular module

9. Critical Evaluation

Developing a database system for Ms. Mary's E-Classroom platform, including students, instructors, programs, modules, assessments, resources, and announcements, was the main goal of the case study. One of the major challenges was structuring the database to handle the complex relationships among these elements. For instance, I had to ensure that students were correctly linked to specific programs, modules were shared across different programs, and teachers were assigned to the right modules. In addition, managing assessments, tracking resources completed by students in sequence, and storing module-specific announcements added further complexity to the overall design, requiring a thoughtful approach.

Through this project, I was able to sharpen my skills in designing a relational database that effectively manages interconnected data. I spent time focusing on how to create meaningful relationships between entities like students, modules, assessments, and resources while tackling challenges related to data integrity and system performance. The hands-on experience of writing SQL queries, applying normalization techniques, and designing a system that incorporated various features helped me build practical skills for future projects. Overall, this case study offered valuable insights into creating dynamic, scalable database systems for educational platforms, which will serve me well in both academic and professional endeavors.

10.Dump File Creation

```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\rohitjoshi> cd "C:\Mac\Home\Documents\UNI\2ND YEAR\DATABASE\Database Coursework\Final Submission"
PS Microsoft.PowerShell.Core\FileSystem::C:\Mac\Home\Documents\UNI\2ND YEAR\DATABASE\Database Coursework\Final Submission> mkdir dumpfile

Directory: C:\Mac\Home\Documents\UNI\2ND YEAR\DATABASE\Database Coursework\Final Submission

Mode                LastWriteTime         Length Name
----                -----        ----  -
d-----       1/20/2025 7:43 PM          0  dumpfile

PS Microsoft.PowerShell.Core\FileSystem::C:\Mac\Home\Documents\UNI\2ND YEAR\DATABASE\Database Coursework\Final Submission> cd dumpfile
PS Microsoft.PowerShell.Core\FileSystem::C:\Mac\Home\Documents\UNI\2ND YEAR\DATABASE\Database Coursework\Final Submission\dumpfile> exp RohitJoshi/23049330 file=RohitJoshi.dmp

Export: Release 11.2.0.2.0 - Production on Mon Jan 20 19:45:38 2025
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

Connected to: Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
Export done in WEBSM1N252 character set and AL16UTF16 NCHAR character set
server uses AL32UTF8 character set (possible charset conversion)
exporting pre-schema procedural objects and actions
exporting synonyms
exporting temporary synonym names for user ROHITJOSHI
exporting PUBLIC type synonyms
exporting private type synonyms
exporting object type definitions for user ROHITJOSHI
About to export ROHITJOSHI's objects ...
exporting database links
exporting sequence numbers
exporting post-schema procedural actions
about to export ROHITJOSHI's tables via Conventional Path ...
exporting table ANNOUNCEMENT      9 rows exported
exporting table ANNOUNCE_TEACH_MOD_STUD   9 rows exported
exporting table ASSESSMENT      11 rows exported
exporting table ASSES_MOD_STUD    11 rows exported
exporting table MODULE           8 rows exported
exporting table MOD_STUD         11 rows exported
exporting table PROGRAM          7 rows exported
exporting table RESOURCES        8 rows exported
exporting table RESO_MOD_STUD    11 rows exported
exporting table STUDENT          10 rows exported
exporting table TEACHER          9 rows exported
exporting table TEACH_MOD_STUD   10 rows exported
exporting Synonyms
exporting grants
exporting stored procedures
exporting operators
exporting primary key integrity constraints
exporting triggers
exporting indextypes
exporting bitmap, functional and extensible indexes
exporting materialized views
exporting snapshot logs
exporting jobs
exporting refresh groups and children
exporting dimensions
exporting post-schema procedural objects and actions
exporting materialized views
Export terminated successfully without warnings.
PS Microsoft.PowerShell.Core\FileSystem::C:\Mac\Home\Documents\UNI\2ND YEAR\DATABASE\Database Coursework\Final Submission\dumpfile>

```

Figure 64: Dump File Creation

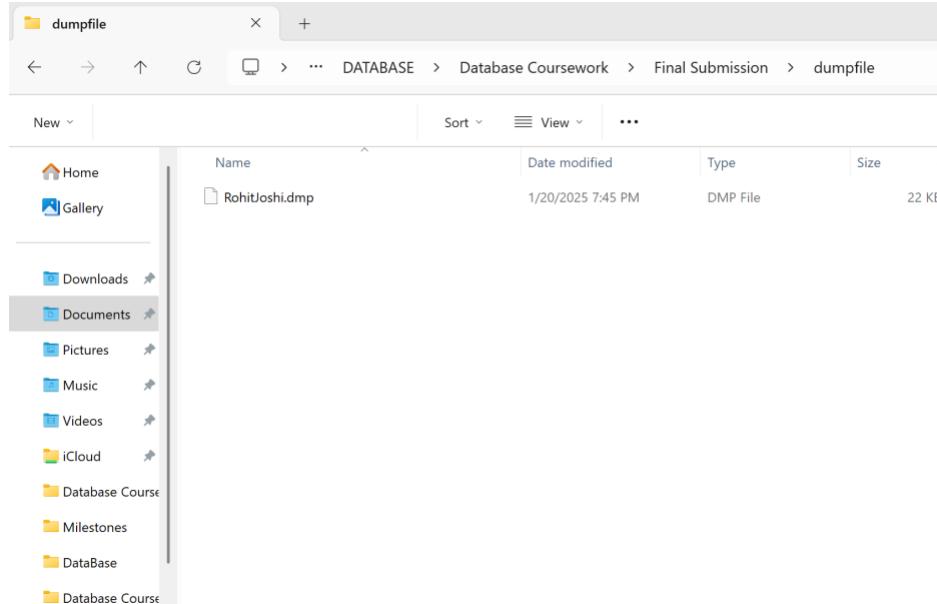


Figure 65: Dump File Created

11.Dropping Tables

```
SQL> conn rohitjoshi/23049330;
Connected.
SQL> Drop table Reso_Mod_Stud;
Table dropped.

SQL>
SQL> Drop table Resources;
Table dropped.

SQL>
SQL> Drop table Announce_Teach_Mod_Stud;
Table dropped.

SQL>
SQL> Drop table Announcement;
Table dropped.

SQL>
SQL> Drop table Assess_Mod_Stud;
Table dropped.

SQL>
SQL> Drop table Assessment;
Table dropped.

SQL>
SQL> Drop table Teach_Mod_Stud;
Table dropped.

SQL>
SQL> Drop table Teacher;
Table dropped.

SQL>
SQL> Drop table Mod_Stud;
Table dropped.

SQL>
SQL> Drop table Module;
Table dropped.

SQL>
SQL> Drop table Student;
Table dropped.

SQL>
SQL> Drop table Program;
Table dropped.

SQL> |
```

Figure 66: Dropping all tables

12.Bibliography

Jaiswal, S., n.d. *First Normal Form (1NF)*. [Online]

Available at: <https://www.javatpoint.com/dbms-first-normal-form>

[Accessed 27 12 2024].

Jaiswal, S., n.d. *Second Normal From (2NF)*. [Online]

Available at: <https://www.javatpoint.com/dbms-second-normal-form>

[Accessed 25 12 2024].

Jaiswal, S., n.d. *Third Normal Form (3NF)*. [Online]

Available at: <https://www.javatpoint.com/dbms-third-normal-form>

[Accessed 26 12 2024].

Jain, S., n.d. *Attributes in DBMS*. [Online]

Available at: <https://www.geeksforgeeks.org/attributes-in-dbms/>

[Accessed 23 12 2024].

Jain, S., n.d. *Entities in DBMS*. [Online]

Available at: <https://www.geeksforgeeks.org/entity-in-dbms/>

[Accessed 23 12 2024].

Jain, S., n.d. *What is Data Dictionary*. [Online]

Available at: <https://www.geeksforgeeks.org/what-is-data-dictionary/>

[Accessed 10 01 2025].

Limbu, B. & Shrestha, U., 1996. *Let us Introduce ourselves (LACM)*. [Online]

Available at: <https://las.edu.np/aboutus>

[Accessed 10 01 2025].

Jain, S., 2024. *Introduction to ER Model*. [Online]

Available at: <https://www.geeksforgeeks.org/introduction-of-er-model/>

[Accessed 2024].

Jain, S., 2024. *Normalization*. [Online]

Available at: <https://www.geeksforgeeks.org/types-of-normal-forms-in-dbms/>

[Accessed 20 12 2024].