

Table of Contents

1	Introduction.....	1
1.1	Introduction of the business	1
1.2	Current Business Activities and Operations.....	1
1.3	Business Rules Derived from Operational Procedures	2
1.4	Assumptions:.....	3
2	Initial ERD	4
2.1	Identification of Entities and Attributes	4
2.1.1	Program.....	4
2.1.2	Student	4
2.1.3	Module	4
2.2	Entity Relationship Diagram.....	5
3	Normalization	6
3.1	UNF	6
3.2	1NF.....	6
3.3	2NF.....	8
3.4	3NF.....	11
4	Data Dictionary and Final ERD.....	15
4.1	Data Dictionary	15
4.1.1	Program.....	15
4.1.2	Student	15
4.1.3	Module	15
4.1.4	Module_Student.....	16
4.1.5	Teacher	16
4.1.6	Teacher_Module_Student	16
4.1.7	Announcement	17
4.1.8	Announcement_Mod_Stud_Tea.....	17
4.1.9	Resources	17
4.1.10	Resource_Module_Student	18
4.1.11	Assessment.....	18
4.1.12	Result.....	18
4.2	Final ERD.....	19

5	Implementation	20
5.1	Creating Entities and Establishing Relationships.....	20
5.1.1	Creating Program Table	20
5.1.2	Creating Student Table.....	21
5.1.3	Creating Module Table.....	21
5.1.4	Creating Module_Student Table.....	22
5.1.5	Creating Teacher Table.....	22
5.1.6	Creating Teacher_Module_Student Table	23
5.1.7	Creating Announcement Table	23
5.1.8	Creating Announcement_Mod_Std_Tea Table	24
5.1.9	Creating Resources Table.....	24
5.1.10	Creating Resource_Module_Student Table.....	25
5.1.11	Creating Assessment	25
5.1.12	Creating Result Table	26
5.2	Populating the database with suitable test data	26
5.2.1	Inserting test data in the Program Table.....	26
5.2.2	Inserting test data in the Student Table	27
5.2.3	Inserting test data in the Module Table	28
5.2.4	Inserting test data in the Module_Student Table.....	29
5.2.5	Inserting test data in the Teacher Table	30
5.2.6	Inserting test data in the Teacher_Module_Student Table	31
5.2.7	Inserting test data in the Announcement Table	32
5.2.8	Inserting test data in the Announcement_Mod_Std_Tea Table.....	33
5.2.9	Inserting test data in the Resource Table.....	34
5.2.10	Inserting test data in the Resource_Module_Student Table	35
5.2.11	Inserting test data in the Assessment Table	36
5.2.12	Inserting test data in the Result Table	37
6	Database Query	39
6.1	Information Query	39
6.1.1	List the programs that are available in the college and the total number of students enrolled in each.	39
6.1.2	List all the announcements made for a particular module starting from 1st May 2024 to 28th May 2024.	39
6.1.3	List the names of all modules with the letter 'D', along with the total number of resources uploaded for those modules.	40
6.1.4	List the names of all students along with their enrolled program who have not submitted any assessments for a particular module.	40
6.1.5	List all the teachers who teach more than one module	40

6.2	Transaction Query	41
6.2.1	Identify the module that has the latest assessment deadline.	41
6.2.2	Find the top three students who have the highest total score across all modules	41
6.2.3	Find the total number of assessments for each program and the average score across all assessments in those programs.	42
6.2.4	List the students who have scored above the average score in the ‘Databases’ module.....	42
6.2.5	Display whether a student has passed or failed as remarks as per their total aggregate marks obtained in a particular module.	43
7	Critical Evaluation	44
7.1	Critical Evaluation of module, its usage and relation with other subject.....	44
7.2	Critical Assessment of Coursework	44
8	Creating Dump File.....	46
9	Drop Table.....	48
10	Reference	50

Table of Figures

Figure 1 Initial ERD	5
Figure 2 Final ERD	19
Figure 3 Creating User and Granting Permission	20
Figure 4 Connecting User.....	20
Figure 5 Creating Program Table	20
Figure 6 Creating Student Table.....	21
Figure 7 Creating Module Table.....	21
Figure 8 Creating Module_Student Table.....	22
Figure 9 Creating Teacher Table.....	22
Figure 10 Teacher_Module_Student	23
Figure 11 Creating Announcement Table	23
Figure 12 Creating Announcement_Mod_Std_Tea Table	24
Figure 13 Creating Resources Table	24
Figure 14 Creating Resource_Module_Student Table.....	25
Figure 15 Creating Assessment Table	25
Figure 16 Creating Result Table.....	26
Figure 17 Inserting test data in Program Table.....	26
Figure 18 Selecting data from Program Table.....	27
Figure 19 Inserting test data in Student Table	27
Figure 20 Selecting data from Student Table	28
Figure 21 Inserting test data in Module Table	28
Figure 22 Selecting data from Module Table	29
Figure 23 Inserting test data to Module_Student Table	29
Figure 24 Selecting data from Module _ Student Table	30
Figure 25 Inserting test data in Teacher Table	30
Figure 26 Selecting data from Teacher Table	31

Figure 27 Inserting test data in Teacher_Module_Student Table	31
Figure 28 Selecting data from Teacher_Module_Student Table	32
Figure 29 Inserting test data in Announcement Table	32
Figure 30 Selecting data from Announcement Table	33
Figure 31 Inserting test data in Announcement_Module_Student Table.....	33
Figure 32 Selecting data from Announcement_Mod_Std_Tea Table	34
Figure 33 Inserting test data in Resources Table	34
Figure 34 Selecting data from Resources Table	35
Figure 35 Inserting test data in Resource_Module_Student Table	35
Figure 36 Selecting data from Resource_Module_Student Table	36
Figure 37 Inserting test data in Assessment Table	36
Figure 38 Selecting data from Assessment Table	37
Figure 39 Inserting test data in Result Table	37
Figure 40 Selecting data from Result Table	38
Figure 41 Listing all available programs along with total number of students enrolled in each	39
Figure 42 Listing all the announcements made for module MR14 from 1st May 2024 to 28th May 2024	39
Figure 43 Listing the names of modules beginning with the letter 'D' along with the total number of resources uploaded	40
Figure 44 Listing the names of all students along with their enrolled program who have not submitted any assessments for module MD11	40
Figure 45 Listing all teachers who teach more than one module	40
Figure 46 Listing the module having latest assessment deadline	41
Figure 47 Listing the top three students who have the highest total score across all modules.	41
Figure 48 Listing the total number of assessments for each program and the average score across all assessments in those programs	42
Figure 49 Listing the students who have scored the average score in Database Module	42

Figure 50 Displaying whether a student has passed or failed as remarks as per their total aggregate marks obtained in module MR14.....	43
Figure 51 Creating Dump File 1.....	46
Figure 52 Creating Dump File 2.....	47
Figure 53 Dropping Tables	48
Figure 54 Tables after dropping	49

Table of Tables

Table 1 Program (initial)	4
Table 2 Student (initial).....	4
Table 3 Module (initial)	4
Table 4 Program Data Dictionary	15
Table 5 Student Data Dictionary	15
Table 6 Module Data Dictionary	15
Table 7 Module_Student Data Dictionary	16
Table 8 Teacher Data Dictionary	16
Table 9 Teacher_Module_Student Data Dictionary	16
Table 10 Announcement Data Dictionary	17
Table 11 Announcement _ Module _ Student Data Dictionary	17
Table 12 Resource Data Dictionary	17
Table 13 Resource_Module_Student Data Dictionary	18
Table 14 Assessment Data Dictionary	18
Table 15 Result Data Dictionary	18

1 Introduction

1.1 Introduction of the business

College is a prestigious educational institution established

. It is dedicated to provide high-quality education. Being an IT college, it offers undergraduate and postgraduate programs in Computing, Networking, Multimedia, and AI, along with business, marketing, etc. Its mission is to inspire students through innovative learning and a vision to achieve academic success.

Ms. Mary, the chairperson of Islington College, with a hope to uplift the education system, wants to launch an E-classroom platform to provide a digital study environment for both students and teachers. After attending a specialized teaching training program in London, Ms. Mary observed a vast difference between the teaching techniques used there and in Nepal. Inspired by her visit and observation, she wants to implement similar advanced teaching methods through the E-classroom platform, creating an effective learning environment.

1.2 Current Business Activities and Operations

The college has been running different activities, including sports activities. It hosts yearly sports events where students compete to secure the first position. Both physical and mental activities such as basketball, futsal, badminton, table tennis, and chess are conducted by the college. A scholarship program is also included here, where students are given scholarships on the basis of their class performance, attendance and attitude (AAA).

The college's current activities include student enrollments, managing programs and modules included in the programs, assigning modules to teachers, assessments, resources and result management, and many more. Student enrollment, program and module management, instructor module assignments, assessments, resource and result management, and many more tasks are among the college's current activities. Many programs have several required modules for the students to complete. Teachers are in charge of issuing announcements and are tasked with instructing particular modules. The module includes assessments as per which students

are reviewed, and result is generated. Additionally, the module includes several resources that are necessary to convey the contents and that students must finish in a specific order.

In order to efficiently handle every entity, student, program, module, teacher, assessment, announcement, resource, and outcome, Ms. Mary's suggested E-classroom platform also incorporates a database system. The platform will guarantee a dynamic and organized learning environment for both teachers and students, monitoring them in addition to

1.3 Business Rules Derived from Operational Procedures

Business rules are rules applied to the database that affect its structure. According to the defined business rules, the data flow is carried out throughout the system.

The business rules for the E-classroom platform are:

Student and Program

- A student must enroll in one of the various programs available in the college, and a program can have multiple students enrolled.

Program and Module

- Each program in the college consists of multiple modules which are mandatory for students and each module can be part of many programs.

Module and Assessments

- Students are given one or more assessments for each module, and each assessment is connected to a specific module.

Module and Resources

- Each module contains various Resources and a resource belongs to a specific module.

Module and Announcement

- Each announcement is linked to one specific module and each module can have multiple announcements.

Module and Teacher

- Each teacher is assigned to teach a specific module and each module contains multiple teachers.

1.4 Assumptions:

Some of the assumptions made for the database system are:

Module and Resource

- A resource belongs to a specific module.

Module and Announcement

- Each module can have multiple announcements.

Module and Teacher

- Each module contains multiple teachers.

Assessment and Result

- Each assessment generates results for multiple students and each result is linked to a specific assessment.

Student and Result

- Each student can have results from different assessments and each result is linked to a specific student.

Student and Resource

- Each student accesses multiple resources in a module and each resource can be accessed by multiple students.

Teacher and Announcement

- Each teacher can post multiple announcements for their respective module, and each announcement is created by one teacher.

2 Initial ERD

2.1 Identification of Entities and Attributes

2.1.1 Program

The attributes of the program entity are:

Table 1 Program (initial)

S. No.	Attributes	Data Type	Size	Constraint
1	Program_Code	CHARACTER	10	Primary Key
2	Program_Name	CHARACTER	30	Not Null
3	Program_Duration	NUMBER	5	Not Null
4	Program_Description	CHARACTER	100	Not Null

2.1.2 Student

The attributes of student entity are:

Table 2 Student (initial)

S.No.	Attributes	Data Type	Size	Constraint
1	Student_ID	CHARACTER	10	Primary Key
2	Student_Name	CHARACTER	40	Not Null
3	Address	CHARACTER	30	Not Null
4	Phone_No	CHARACTER	10	Unique
5	Email	CHARACTER	30	Unique

2.1.3 Module

The attributes of module entity are:

Table 3 Module (initial)

S.No.	Attributes	Data Type	Size	Constraint
1	Module_Code	CHARACTER	10	Primary Key
2	Module_Title	CHARACTER	30	Not Null
3	Credits	NUMBER	5	Not Null
4	Module_Description	CHARACTER	100	Not Null
5	Teacher_ID	CHARACTER	10	Unique
6	Teacher_Name	CHARACTER	40	Not Null

7	Adress	CHARACTER	30	Not Null
8	Phone	CHARACTER	10	Unique
9	Email	CHARACTER	30	Unique
10	Announcement_ID	CHARACTER	10	Unique
11	Announcement_Title	CHARACTER	30	Not Null
12	Date_Posted	DATE		Not Null
13	Announcement_Description	CHARACTER	100	Not Null
14	Resource_ID	CHARACTER	10	Unique
15	Resource_Title	CHARACTER	20	Not Null
16	Resource_Duration_Days_Days	NUMBER	10	Not Null
17	Type	CHARACTER	20	Not Null
18	Completion_Status	CHARACTER	10	Not Null
19	Completion_Date	DATE		Null
20	Assessment_ID	NUMBER	10	Unique
21	Assessment_Title	CHARACTER	20	Not Null
22	Deadline	DATE		Not Null
23	Weightage	NUMBER	10	Not Null
24	Marks_Obtained	NUMBER	10	Null
25	Grade	CHARACTER	2	Null

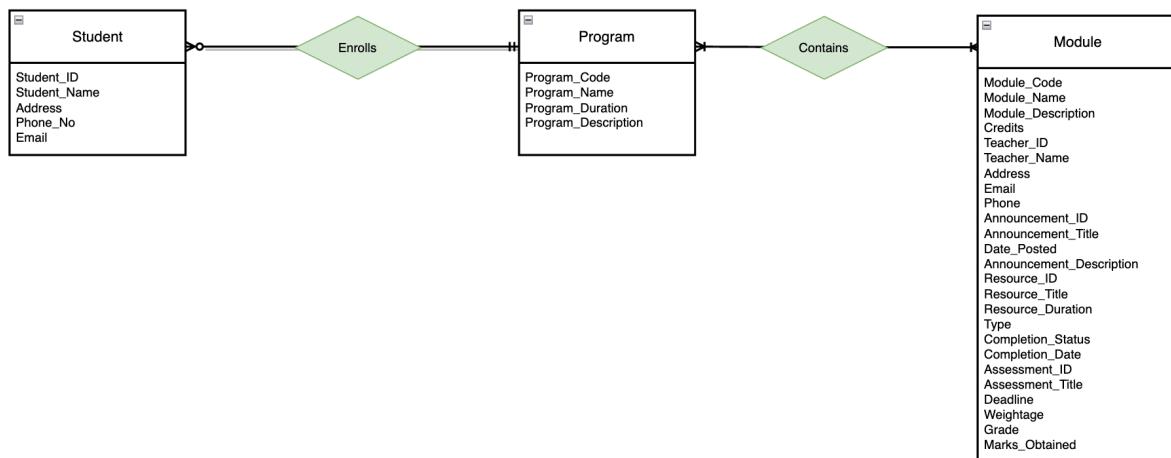


Figure 1 Initial ERD

3 Normalization

Normalization in a database is the process of structuring an RDMS in an organized way to avoid redundancy and complexities and maintain integrity. (Chris, 2022) In normalization, a complex table is divided into several tables, making it easier to understand.

3.1 UNF

Unnormalized Form (UNF) is the simplest database model, also referred to as non-first form, and contains only one table with all attributes. It is the first process of normalization.

The UNF for the system is:

E-Classroom – (Student_ID, Student_Name, Address, Phone_no, Email, Program_Code, Program_Name, Program_Duration, Program_Description, { Module_Code, Module_Name, Credits, Module_Description, { Teacher_ID, Teacher_Name, Address, Phone, Email }, {Announcement_ID, Announcement_Title, Date_Posted, Announcement_Description} }, { Resource_ID, Resource_Title, Resource_Duration_Days, Type, Completion_Status, Completion_Date }, { Assessment_ID, Assessment_Title, Deadline, Weightage, Marks_Obtained, Grade })

Here, Student_ID is the unique identifier for the table.

In this UNF, since teachers, resources, and assessment attributes repeat module attributes and module attributes also repeat, student attributes and program attributes, they fall under the repeating groups. Also, the announcement attribute gets repeated within the teacher as the teacher posts multiple announcements, it falls under the repeating group within the teacher.

3.2 1NF

A relation is said to be in its first normal form if the table contains only single-valued attributes. (GeeksforGeeks, 2020) In 1NF, all the repeating group from UNF is separated to another table.

The criteria for 1NF are:

- The table should contain atomic values.
- There should be a unique identifier, i.e. primary key in each table.
- The attribute values in each row should be unique.

The 1NF for the system:

The repeating group from UNF is separated into different tables.

Student 1 - (Student_ID, Student_Name, Address, Phone_no, Email, Program_Code, Program_Name, Program_Duration, Program_Description)

In Student 1 table, Student_ID is primary key.

Module 1 - (Module_Code, Student_ID*, Module_Name, Credits, Module_Description)

In Module 1, Module_Code and Student_ID are composite keys.

Teacher 1 - (Teacher_ID, Module_Code*, Student_ID*, Teacher_Name, Address, Phone, Email)

In Teacher 1, Teacher_ID, Module_Code and Student_ID are composite keys.

Announcement 1 - (Announcement_ID, Student_ID*, Module_ID*, Teacher_ID*, Announcement_Title, Date_Posted, Announcement_Description)

In Announcement 1, Announcement_ID, Student_ID, Module_ID and Teacher_ID are composite key.

Resources 1 - (Resource_ID, Module_Code*, Student_ID*, Resource_Title, Resource_Duration_Days, Type, Completion_Status, Completion_Date)

In Resource 1, Resource_ID, Module_Code and Student_ID are composite keys.

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

In Assessment 1, Assessment_ID, Module_Code and Student_ID are composite keys.

3.3 2NF

A table is said to be in 2NF if it meets the following criteria:

- It is in 1NF.
- Has no partial dependency (All the non-key attributes should be fully dependent on the primary key)

The process for 2NF for the system:

Eliminating Partial dependencies,

Since there are no composite keys in the Student1 table, it is already in 2NF.

Student 1 - (Student_ID, Student_Name, Address, Phone_no, Email, Program_Code, Program_Name, Program_Duration, Program_Description)

For Module 1 table,

Module_Code \rightarrow Module_Name, Credits, Module_Description (Partial Dependency)

Module_Code, Student_ID \rightarrow x

Student_ID \rightarrow x

Module_Code gives Module_Name, Credits, and Module_Description, creating a partial dependency, and Module_Code and Student_ID together do not give any non-key and value.

For Teacher 1 table,

Teacher_ID \rightarrow Teacher_Name, Address, Phone, Email (Partial Dependency)

Teacher_ID, Module_Code, Student_ID \rightarrow x

Module_Code \rightarrow x

Student_ID -> x

Teacher_ID gives Teacher_Name, Address, Phone and Email, creating a partial dependency, and Teacher_ID, Module_Code and Student_ID together do not give any non-key and value.

For Announcement 1 table,

Announcement_ID -> Announcement_Title, Date_Posted, Announcement_Description
(Partial Dependency)

Announcement_ID, Module_Code, Student_ID, Teacher_ID -> x

Module_Code -> x

Student_ID -> x

Teacher_ID -> x

Announcement_ID gives Announcement_Title, Date_Posted, and Announcement_Description, creating a partial dependency, and Announcement_ID, Module_Code, Student_ID and Teacher_ID together do not give any non-key and value.

For Resources 1 table,

Resource_ID -> Resource_Title, Resource_Duration_Days, Type (Partial Dependency)

Resorce_ID, Module_Code, Student_ID -> Completion_Status, Completion_Date (Fully Dependency)

Module_Code -> x

Student_ID -> x

Resource_ID gives Resource_Title, Resource_Duration_Days, and Type, creating a partial dependency, and Resrouce_ID, Module_Code and Student_ID together give Completion_Status and Completion_Date because, as per the assumption made, the student completes the resource of a particular module.

For Assessment 1 table,

Assessment_ID -> Assessment_Title, Deadline, Weightage (Partial Dependency)

Assessment_ID, Module_Code, Student_ID -> Marks_Obtained, Grade (Fully Dependency)

Module_Code -> x

Student_ID -> x

Assessment_ID gives Assessment_Title, Deadline, and Weightage, creating a partial dependency, and Assement_ID, Module_Code and Student_ID together give Makrs_Obtained and Grade because, as per the assumption made, the student receives the marks or result of an assessment related to a specific module

After eliminating the partial dependencies, the tables formed in 2NF are:

Student 1 - (Student_ID, Student_Name, Address Phone_no, Email, Program_Code, Program_Name, Program_Duration, Program_Description)

Module 2 – (Module_Code, Module_Name, Credits, Module_Description)

Module_Student 2 – (Module_Code*, Student_ID*)

Teacher 2 – (Teacher_ID, Teacher_Name, Adress, Phone, Email)

Teacher_Module_Student 2 – (Teacher_ID*, Module_Code*, Student_ID*)

Announcement 2 – (Announcement_ID, Announcement_Title, Date_Posted, Announcement_Description)

Announcement_Mod_Stud_Tea 2 – (Announcement_ID*, Module_Code*,
Student_ID*,Teacher_ID*)

Resources 2 – (Resource_ID, Resource_Title, Resource_Duration_Days, Type)

Resource_Module_Student 2 – (Resource_ID*, Module_Code*, Student_ID*,
Completion_Status, Completion_Date)

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

Result 2 – (Assessment_ID*, Module_Code*, Student_ID*, Marks_Obtained, Grade)

3.4 3NF

A table is said to be in 3NF if it meets the following criteria:

- It is in 2NF.
- Has no transitive dependencies.

Eliminating transitive dependency,

For Student 1 Table,

Student 1 - (Student_ID, Student_Name, Address Phone_no, Email, Program_Code,
Program_Name, Program_Duration, Program_Description)

Student_ID -> Program_Code -> Program_Name, Program_Duration, Program_Description

Here, a transitive dependency is seen as Program_Code is linked with Student_ID but also Program_name, Program_Duration, Program_Description are linked with Program_Code. So a new table **Program** is formed.

For Module 2 Table,

Module 2 – (Module_Code, Module_Name, Credits, Module_Description)

No transitive dependency exists as all the non-key attributes are linked with Module_Code only.

For Module_Student 2 Table,

Module_Student 2 – (Module_Code*, Student_ID*)

No transitive dependency exists as there are no non-key attributes.

For Teacher 2 Table,

Teacher 2 – (Teacher_ID, Teacher_Name, Adress, Phone, Email)

No transitive dependency exists as all the non-key attributes are linked with Teacher_ID only.

For Teacher_Module_Student 2 Table,

Teacher_Module_Student 2 – (Teacher_ID*, Module_Code*, Student_ID*)

No transitive dependency exists as there are no non-key attributes.

For Announcement 2 Table,

Announcement 2 – (Announcement_ID, Announcement_Title, Date_Posted,
Announcement_Description)

No transitive dependency exists as all the non-key attributes are linked with Announcement_ID only.

For Announceent_Mod_Stud_Tea 2 Table,

Announcement_Mod_Stud_Tea 2 – (Announcement_ID*, Module_Code*,
Student_ID*,Teacher_ID*)

No transitive dependency exists as there are no non-key attributes.

For Resource 2 Table,

Resources 2 – (Resource_ID, Resource_Title, Resource_Duration_Days, Type)

No transitive dependency exists as all the non-key attributes are linked with Resource_ID only.

For Resource_Module_Student 2 Table,

Resource_Module_Student 2 – (Resource_ID*, Module_Code*, Student_ID*,
Completion_Status, Completion_Date)

No transitive dependency exists as the non-key attributes are only linked with the composite primary keys.

For Assessment 2 Table,

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

No transitive dependency exists as all the non-key attributes are linked with Assessment_ID only.

For Result 2 Table,

Result 2 – (Assessment_ID*, Module_Code*, Student_ID*, Marks_Obtained, Grade)

No transitive dependency exists as the non-key attributes are only linked with the composite primary keys.

The final tables formed after 3NF are:

Student - (Student_ID, Student_Name, Address, Phone_no, Email, Program_Code*)

Program – (Program_Code, Program_Name, Program_Duration, Program_Description)

Module – (Module_Code, Module_Name, Credits, Module_Description)

Module_Student – (Module_Code*, Student_ID*)

Teacher – (Teacher_ID, Teacher_Name, Adress, Phone, Email)

Teacher_Module_Student – (Teacher_ID*, Module_Code*, Student_ID*)

Announcement – – (Announcement_ID, Announcement_Title, Date_Posted, Announcement_Description)

Announcement_Mod_Stud_Tea – (Announcement_ID*, Module_Code*, Student_ID*, Teacher_ID*)

Resources – (Resource_ID, Resource_Title, Resource_Duration_Days, Type)

Resource_Module_Student – – (Resource_ID*, Module_Code*, Student_ID*, Completion_Status, Completion_Date)

Assessment – (Assessment_ID, Assessment_Title, Deadline, Weightage)

Result – (Assessment_ID*, Module_Code*, Student_ID*, Marks_Obtained, Grade)

4 Data Dictionary and Final ERD

4.1 Data Dictionary

4.1.1 Program

The attributes of the program entity are:

Table 4 Program Data Dictionary

S. No.	Attributes	Data Type	Size	Constraint
1	Program_Code	CHARACTER	10	Primary Key
2	Program_Name	CHARACTER	30	Not Null
3	Program_Duration	NUMBER	5	Not Null
4	Program_Description	CHARACTER	100	Not Null

4.1.2 Student

The attributes of the student entity are:

Table 5 Student Data Dictionary

S.No.	Attributes	Data Type	Size	Constraint
1	Student_ID	CHARACTER	10	Primary Key
2	Student_Name	CHARACTER	30	Not Null
3	Address	CHARACTER	30	Not Null
4	Phone_No	CHARACTER	10	Unique, Not Null
5	Email	CHARACTER	30	Unique, Not Null
6	Program_Code	CHARACTER	10	Foreign Key

4.1.3 Module

The attributes of the module entity are:

Table 6 Module Data Dictionary

S.No.	Attributes	Data Type	Size	Constraint
1	Module_Code	CHARACTER	10	Primary Key
2	Module_Title	CHARACTER	30	Not Null
3	Credits	NUMBER	5	Not Null
4	Module_Description	CHARACTER	100	Not Null

4.1.4 Module_Student

The attributes of the module_student entity are:

Table 7 Module_Student Data Dictionary

S.No.	Attributes	Data Type	Size	Constraint	Composite Constraints
1	Module_Code	CHARACTER	10	Foreign Key	Primary Key
2	Student_ID	CHARACTER	10	Foreign Key	

4.1.5 Teacher

The attributes of the teacher entity are:

Table 8 Teacher Data Dictionary

S.No.	Attributes	Data Type	Size	Constraint
1	Teacher_ID	CHARACTER	10	Primary Key
2	Teacher_Name	CHARACTER	30	Not Null
3	Address	CHARACTER	30	Not Null
4	Phone	CHARACTER	10	Not Null
5	Email	CHARACTER	30	Not Null

4.1.6 Teacher_Module_Student

The attributes of the teacher_module_student entity are:

Table 9 Teacher_Module_Student Data Dictionary

S.No.	Attributes	Data Type	Size	Constraint	Composite Constraints
1	Teacher_ID	CHARACTER	10	Foreign Key	Primary Key
2	Module_Code	CHARACTER	10	Foreign Key	
3	Student_ID	CHARACTER	10	Foreign Key	

4.1.7 Announcement

The attributes of the announcement entity are:

Table 10 Announcement Data Dictionary

S.No.	Attributes	Data Type	Size	Constraint
1	Announcement_ID	NUMBER	10	Primary Key
2	Announcement_Title	CHARACTER	20	Not Null
3	Date_Posted	DATE		Not Null
4	Announcement_Description	CHARACTER	100	Not Null

4.1.8 Announcement_Mod_Stud_Tea

The attributes of the announcement_module_student entity are:

Table 11 Announcement_Module_Student Data Dictionary

S.No.	Attributes	Data Type	Size	Constraint	Composite Constraints
1	Announcement_ID	NUMBER	10	Foreign Key	Primary Key
2	Module_Code	CHARACTER	10	Foreign Key	
3	Student_ID	CHARACTER	10	Foreign Key	
4	Teacher_ID	CHARACTER	10	Foreign Key	

4.1.9 Resources

The attributes of the resource entity are:

Table 12 Resource Data Dictionary

S.No.	Attributes	Data Type	Size	Constraint
1	Resource_ID	NUMBER	10	Primary Key
2	Resource_Title	CHARACTER	20	Not Null
3	Resource_Duration_Days	NUMBER	10	Not Null
4	Type	CHARACTER	10	Not Null

4.1.10 Resource_Module_Student

The attributes of the resource_module_student entity are:

Table 13 Resource_Module_Student Data Dictionary

S.No.	Attributes	Data Type	Size	Constraint	Composite Constraints
1	Resource_ID	Number	10	Foreign Key	Primary Key
2	Module_Code	CHARACTER	10	Foreign Key	
3	Student_ID	CHARACTER	10	Foreign Key	
4	Completion_Status	CHARACTER	10	Not Null	-
5	Completion_Date	DATE		Null	-

4.1.11 Assessment

The attributes of the assessment entity are:

Table 14 Assessment Data Dictionary

S.No.	Attributes	Data Type	Size	Constraint
1	Assessment_ID	NUMBER	10	Primary Key
2	Assessment_Title	CHARACTER	30	Not Null
3	Deadline	DATE		Not Null
4	Weightage	NUMBER	10	Not Null

4.1.12 Result

The attributes of the result entity are:

Table 15 Result Data Dictionary

S.No.	Attributes	Data Type	Size	Constraint	Composite Constraints
1	Announcement_ID	NUMBER	10	Foreign Key	Primary Key
2	Module_Code	CHARACTER	10	Foreign Key	
3	Student_ID	CHARACTER	10	Foreign Key	
4	Marks_Obtained	NUMBER	10	Null	-
5	Grade	CHARACTER	1	Null	-

4.2 Final ERD

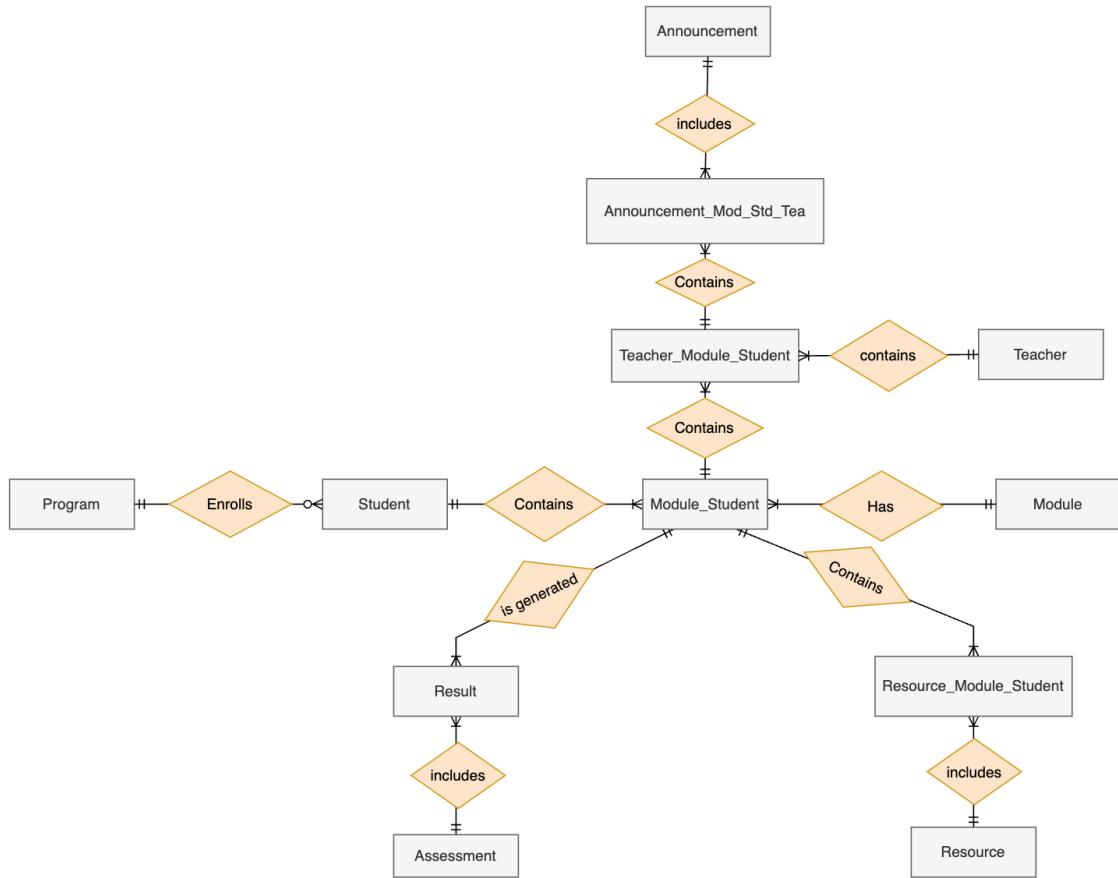


Figure 2 Final ERD

5 Implementation

After normalization, user is created and connected to the system.

```
Enter user-name: System
Enter password:

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> CREATE USER NajibThapa IDENTIFIED BY 23050403;

User created.

SQL> GRANT CONNECT, RESOURCE to NajibThapa;

Grant succeeded.
```

Figure 3 Creating User and Granting Permission

Figure 4 Connecting User

5.1 Creating Entities and Establishing Relationships

5.1.1 Creating Program Table

```
SQL> CREATE TABLE Program
  2    (Program_Code VARCHAR(10) NOT NULL,
  3     Program_Name VARCHAR(30) NOT NULL,
  4     Program_Duration NUMBER NOT NULL,
  5     Program_Description VARCHAR(100) NOT NULL,
  6     CONSTRAINT Program_PK PRIMARY KEY (Program_Code)
  7   );

Table created.
```

Figure 5 Creating Program Table

Name	Null?	Type
PROGRAM_CODE	NOT NULL	VARCHAR2(10)
PROGRAM_NAME	NOT NULL	VARCHAR2(30)
PROGRAM_DURATION	NOT NULL	NUMBER
PROGRAM_DESCRIPTION	NOT NULL	VARCHAR2(100)

5.1.2 Creating Student Table

```
SQL> CREATE TABLE Student
  2  (Student_ID VARCHAR(10) NOT NULL,
  3   Student_Name VARCHAR(30) NOT NULL,
  4   Address VARCHAR(30) NOT NULL,
  5   Phone_No VARCHAR(10) NOT NULL UNIQUE,
  6   Email VARCHAR(30) NOT NULL UNIQUE,
  7   Program_Code VARCHAR(10) NOT NULL,
  8   CONSTRAINT Student_PK PRIMARY KEY (Student_ID),
  9   CONSTRAINT Program_FK FOREIGN KEY (Program_Code) REFERENCES Program(Program_Code)
10  );
Table created.
```

Figure 6 Creating Student Table

```
SQL> DESC Student;
      Name          Null?    Type
-----+
STUDENT_ID           NOT NULL  VARCHAR2(10)
STUDENT_NAME         NOT NULL  VARCHAR2(30)
ADDRESS              NOT NULL  VARCHAR2(30)
PHONE_NO             NOT NULL  VARCHAR2(10)
EMAIL                NOT NULL  VARCHAR2(30)
PROGRAM_CODE         NOT NULL  VARCHAR2(10)
```

5.1.3 Creating Module Table

```
SQL> CREATE TABLE Module
  2  (Module_Code VARCHAR(10) Not Null,
  3   Module_Title VARCHAR(30) NOT NULL,
  4   Credits NUMBER NOT NULL,
  5   Module_Description VARCHAR(100) NOT NULL,
  6   CONSTRAINT Module_PK PRIMARY KEY (Module_Code)
  7  );
Table created.
```

Figure 7 Creating Module Table

```
SQL> DESC Module;
      Name          Null?    Type
-----+
MODULE_CODE           NOT NULL  VARCHAR2(10)
MODULE_TITLE          NOT NULL  VARCHAR2(30)
CREDITS               NOT NULL  NUMBER
MODULE_DESCRIPTION    NOT NULL  VARCHAR2(100)
```

5.1.4 Creating Module_Student Table

```
SQL> CREATE TABLE Module_Student
  2  (Module_Code VARCHAR(10) NOT NULL,
  3   Student_ID VARCHAR(10) NOT NULL,
  4   CONSTRAINT Mod_Std_PK PRIMARY KEY (Module_Code, Student_ID),
  5   CONSTRAINT Module_FK FOREIGN KEY (Module_Code) REFERENCES Module(Module_Code),
  6   CONSTRAINT Student_FK FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID)
  7 );
```

Table created.

Figure 8 Creating Module_Student Table

```
SQL> DESC Module_Student;
```

Name	Null?	Type
MODULE_CODE	NOT NULL	VARCHAR2(10)
STUDENT_ID	NOT NULL	VARCHAR2(10)

5.1.5 Creating Teacher Table

```
SQL> CREATE TABLE Teacher
  2  (Teacher_ID VARCHAR(10) NOT NULL,
  3   Teacher_Name VARCHAR(30) NOT NULL,
  4   Address VARCHAR(30) NOT NULL,
  5   Phone VARCHAR(10) NOT NULL UNIQUE,
  6   Email VARCHAR(30) NOT NULL UNIQUE,
  7   CONSTRAINT Teacher_PK PRIMARY KEY (Teacher_ID)
  8 );
```

Table created.

Figure 9 Creating Teacher Table

```
SQL> DESC Teacher;
```

Name	Null?	Type
TEACHER_ID	NOT NULL	VARCHAR2(10)
TEACHER_NAME	NOT NULL	VARCHAR2(30)
ADDRESS	NOT NULL	VARCHAR2(30)
PHONE	NOT NULL	VARCHAR2(10)
EMAIL	NOT NULL	VARCHAR2(30)

5.1.6 Creating Teacher_Module_Student Table

```
SQL> CREATE TABLE Teacher_Module_Student
  2  (Module_Code VARCHAR(10) NOT NULL,
  3   Student_ID VARCHAR(10) NOT NULL,
  4   Teacher_ID VARCHAR(10) NOT NULL,
  5   CONSTRAINT Tea_Mod_Std_PK PRIMARY KEY (Module_Code, Student_ID, Teacher_ID),
  6   CONSTRAINT Module_FK2 FOREIGN KEY (Module_Code) REFERENCES Module(Module_Code),
  7   CONSTRAINT Student_FK2 FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),
  8   CONSTRAINT Teacher_FK FOREIGN KEY (Teacher_ID) REFERENCES Teacher(Teacher_ID)
  9  );
```

Table created.

Figure 10 Teacher_Module_Student

Name	Null?	Type
MODULE_CODE	NOT NULL	VARCHAR2(10)
STUDENT_ID	NOT NULL	VARCHAR2(10)
TEACHER_ID	NOT NULL	VARCHAR2(10)

5.1.7 Creating Announcement Table

```
SQL> CREATE TABLE Announcement
  2  (Announcement_ID NUMBER NOT NULL,
  3   Announcement_Title VARCHAR(20) NOT NULL,
  4   Date_Posted DATE NOT NULL,
  5   Announcement_Description VARCHAR(50) NOT NULL,
  6   CONSTRAINT Ann_PK PRIMARY KEY (Announcement_ID)
  7  );
```

Table created.

Figure 11 Creating Announcement Table

Name	Null?	Type
ANNOUNCEMENT_ID	NOT NULL	NUMBER
ANNOUNCEMENT_TITLE	NOT NULL	VARCHAR2(20)
DATE_POSTED	NOT NULL	DATE
ANNOUNCEMENT_DESCRIPTION	NOT NULL	VARCHAR2(100)

5.1.8 Creating Announcement_Mod_Std_Tea Table

```
SQL> CREATE TABLE Announcement_Mod_Std_Tea
  2  (Module_Code VARCHAR(10) NOT NULL,
  3   Student_ID VARCHAR(10) NOT NULL,
  4   Teacher_ID VARCHAR(10) NOT NULL,
  5   Announcement_ID NUMBER NOT NULL,
  6   CONSTRAINT Ann_Mod_Std_Tea_PK PRIMARY KEY (Module_Code, Student_ID, Teacher_ID, Announcement_ID),
  7   CONSTRAINT Module_FK3 FOREIGN KEY (Module_Code) REFERENCES Module(Module_Code),
  8   CONSTRAINT Student_FK3 FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),
  9   CONSTRAINT Teacher_FK2 FOREIGN KEY (Teacher_ID) REFERENCES Teacher(Teacher_ID),
 10  CONSTRAINT Ann_FK FOREIGN KEY (Announcement_ID) REFERENCES Announcement(Announcement_ID)
 11  );
```

Table created.

Figure 12 Creating Announcement_Mod_Std_Tea Table

```
SQL> DESC Announcement_Mod_Std_Tea;
```

Name	Null?	Type
MODULE_CODE	NOT NULL	VARCHAR2(10)
STUDENT_ID	NOT NULL	VARCHAR2(10)
TEACHER_ID	NOT NULL	VARCHAR2(10)
ANNOUNCEMENT_ID	NOT NULL	NUMBER

5.1.9 Creating Resources Table

```
SQL> CREATE TABLE Resources
  2  (Resource_ID NUMBER NOT NULL,
  3   Resource_Title VARCHAR(20) NOT NULL,
  4   Resource_Duration_Days NUMBER NOT NULL,
  5   Type VARCHAR(10) NOT NULL,
  6   CONSTRAINT Res_PK PRIMARY KEY (Resource_ID)
  7  );
```

Table created.

Figure 13 Creating Resources Table

```
SQL> DESC Resources;
```

Name	Null?	Type
RESOURCE_ID	NOT NULL	NUMBER
RESOURCE_TITLE	NOT NULL	VARCHAR2(20)
RESOURCE_DURATION_DAYS	NOT NULL	NUMBER
TYPE	NOT NULL	VARCHAR2(10)

5.1.10 Creating Resource_Module_Student Table

```
SQL> CREATE TABLE Resource_Module_Student
  2  (Module_Code VARCHAR(10) NOT NULL,
  3   Student_ID VARCHAR(10) NOT NULL,
  4   Resource_ID NUMBER NOT NULL,
  5   Completion_Status VARCHAR(10) NOT NULL,
  6   Completion_Date DATE,
  7   CONSTRAINT Res_Mod_Std_PK PRIMARY KEY (Module_Code, Student_ID, Resource_ID),
  8   CONSTRAINT Module_FK4 FOREIGN KEY (Module_Code) REFERENCES Module(Module_Code),
  9   CONSTRAINT Student_FK4 FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),
 10  CONSTRAINT Resource_FK FOREIGN KEY (Resource_ID) REFERENCES Resources(Resource_ID)
 11 );
```

Table created.

Figure 14 Creating Resource_Module_Student Table

```
SQL> DESC Resource_Module_Student;
```

Name	Null?	Type
MODULE_CODE	NOT NULL	VARCHAR2(10)
STUDENT_ID	NOT NULL	VARCHAR2(10)
RESOURCE_ID	NOT NULL	NUMBER
COMPLETION_STATUS	NOT NULL	VARCHAR2(10)
COMPLETION_DATE		DATE

5.1.11 Creating Assessment

```
SQL> CREATE TABLE Assessment
  2  (Assessment_ID NUMBER NOT NULL,
  3   Assessment_Title VARCHAR(30) NOT NULL,
  4   Deadline DATE NOT NULL,
  5   Weightage NUMBER NOT NULL,
  6   CONSTRAINT Assessment_PK PRIMARY KEY (Assessment_ID)
  7 );
```

Table created.

Figure 15 Creating Assessment Table

```
SQL> DESC Assessment;
```

Name	Null?	Type
ASSESSMENT_ID	NOT NULL	NUMBER
ASSESSMENT_TITLE	NOT NULL	VARCHAR2(30)
DEADLINE	NOT NULL	DATE
WEIGHTAGE	NOT NULL	NUMBER

5.1.12 Creating Result Table

```
SQL> CREATE TABLE Result
  2  (Module_Code VARCHAR(10) NOT NULL,
  3   Student_ID VARCHAR(10) NOT NULL,
  4   Assessment_ID NUMBER NOT NULL,
  5   Marks_Obtained NUMBER,
  6   Grade CHAR(1),
  7   CONSTRAINT Assess_Mod_Std_PK PRIMARY KEY (Module_Code, Student_ID, Assessment_ID),
  8   CONSTRAINT Module_FK5 FOREIGN KEY (Module_Code) REFERENCES Module(Module_Code),
  9   CONSTRAINT Student_FK5 FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),
 10  CONSTRAINT Assessment_FK FOREIGN KEY (Assessment_ID) REFERENCES Assessment(Assessment_ID)
 11 );
```

Table created.

Figure 16 Creating Result Table

```
SQL> DESC Result;
      Name          Null?    Type
-----+-----+-----+
MODULE_CODE          NOT NULL  VARCHAR2(10)
STUDENT_ID          NOT NULL  VARCHAR2(10)
ASSESSMENT_ID       NOT NULL  NUMBER
MARKS_OBTAINED      NUMBER
GRADE               CHAR(1)
```

5.2 Populating the database with suitable test data

5.2.1 Inserting test data in the Program Table

```
SQL> INSERT INTO Program (Program_Code, Program_Name, Program_Duration, Program_Description)
  2  VALUES
  3  ('CC101', 'B.Sc.Computing', 3, 'This IT program combines practical and theoretical learning to develop computing skills.');
1 row created.

SQL> INSERT INTO Program (Program_Code, Program_Name, Program_Duration, Program_Description)
  2  VALUES
  3  ('CN102', 'B.Sc.Networking', 3, 'This IT program combines practical and theoretical learning to develop networking skills.');
1 row created.

SQL> INSERT INTO Program (Program_Code, Program_Name, Program_Duration, Program_Description)
  2  VALUES
  3  ('CM103', 'B.Sc.Multimedia', 3, 'This IT program focuses on multimedia design and development');
1 row created.

SQL> INSERT INTO Program (Program_Code, Program_Name, Program_Duration, Program_Description)
  2  VALUES
  3  ('CA104', 'B.Sc.AI', 3, 'This IT program focuses on robotics, artificial intelligence and machine learning');
1 row created.

SQL> INSERT INTO Program (Program_Code, Program_Name, Program_Duration, Program_Description)
  2  VALUES
  3  ('BB101', 'B.Business Administration', 4, 'This program focuses on business analytics, finance, accounting, economics, and marketing');
1 row created.

SQL> INSERT INTO Program (Program_Code, Program_Name, Program_Duration, Program_Description)
  2  VALUES
  3  ('BH102', 'B.Hotel Management', 4, 'This program focuses on hospitality and travel field');
1 row created.
```

Figure 17 Inserting test data in Program Table

SQL> SELECT * FROM Program;					
PROGRAM_CODE	PROGRAM_NAME	PROGRAM_DURATION	PROGRAM_DESCRIPTION		
CC101	B.Sc.Computing	3	This IT program combines practical and theoretical learning to develop computing skills.		
CN102	B.Sc.Networking	3	This IT program combines practical and theoretical learning to develop networking skills.		
CM103	B.Sc.Multimedia	3	This IT program focuses on multimedia design and development		
CA104	B.Sc.AI	3	This IT program focuses on robotics, artificial intelligence and machine learning		
BB101	B.Business Administration	4	This program focuses on business analytics, finance, accounting, economics, and marketing		
BH102	B.Hotel Management	4	This program focuses on focuses on hospitality and travel field		

6 rows selected.

Figure 18 Selecting data from Program Table

5.2.2 Inserting test data in the Student Table

```
SQL> INSERT INTO Student (Student_ID, Student_Name, Address, Phone_No, Email, Program_Code)
  2  VALUES
  3  ('S248132', 'Alex Chhetri' , 'Jhapa', '9888888880', 'chh_axl12@gmail.com', 'CN102');

1 row created.

SQL> INSERT INTO Student (Student_ID, Student_Name, Address, Phone_No, Email, Program_Code)
  2  VALUES
  3  ('S248154', 'Reeya Gurung' , 'Pokhara', '9888888811', 'reeya12@gmail.com', 'CN102');

1 row created.

SQL> INSERT INTO Student (Student_ID, Student_Name, Address, Phone_No, Email, Program_Code)
  2  VALUES
  3  ('S248022', 'Rohan Karki' , 'Kathmandu', '9888888233', 'karkirohan@gmail.com', 'CC101');

1 row created.

SQL> INSERT INTO Student (Student_ID, Student_Name, Address, Phone_No, Email, Program_Code)
  2  VALUES
  3  ('S248144', 'Shivangi Thapa' , 'Kathmandu', '9888888333', 'shivgi@gmail.com', 'BB101');

1 row created.

SQL> INSERT INTO Student (Student_ID, Student_Name, Address, Phone_No, Email, Program_Code)
  2  VALUES
  3  ('S248111', 'Rubin Shrestha' , 'Bhaktapur', '9888881333', 'shtrubin@gmail.com', 'CM103');

1 row created.

SQL> INSERT INTO Student (Student_ID, Student_Name, Address, Phone_No, Email, Program_Code)
  2  VALUES
  3  ('S248109', 'Rubina Shrestha' , 'Palpa', '9888881133', 'rubina@gmail.com', 'CA104');

1 row created.

SQL> INSERT INTO Student (Student_ID, Student_Name, Address, Phone_No, Email, Program_Code)
  2  VALUES
  3  ('S248021', 'Prabin Ghimire' , 'Palpa', '9888801133', 'ghiprabin@gmail.com', 'CC101');

1 row created.

SQL> INSERT INTO Student (Student_ID, Student_Name, Address, Phone_No, Email, Program_Code)
  2  VALUES
  3  ('S248088', 'Pranita Shakya' , 'Kathmandu', '9888801156', 'shakya12@gmail.com', 'CC101');

1 row created.
```

Figure 19 Inserting test data in Student Table

STUDENT_ID	STUDENT_NAME	ADDRESS	PHONE_NO	EMAIL	PROGRAM_CO
S248132	Alex Chhetri	Jhapa	9888888880	chh_axl12@gmail.com	CN102
S248154	Reeya Gurung	Pokhara	9888888811	reeya12@gmail.com	CN102
S248022	Rohan Karki	Kathmandu	98888888233	karkirohan@gmail.com	CC101
S248144	Shivangi Thapa	Kathmandu	9888888833	shivgi@gmail.com	BB101
S248111	Rubin Shrestha	Bhaktapur	9888881333	shtrubin@gmail.com	CM103
S248109	Rubina Shrestha	Palpa	9888881133	rubina@gmail.com	CA104
S248021	Prabin Ghimire	Palpa	9888801133	ghiprabin@gmail.com	CC101
S248088	Pranita Shakya	Kathmandu	9888801156	shakya12@gmail.com	CC101

8 rows selected.

Figure 20 Selecting data from Student Table

5.2.3 Inserting test data in the Module Table

```

SQL> INSERT INTO Module (Module_Code, Module_Title, Credits, Module_Description)
  2  VALUES
  3  ('MD11', 'Database', 15, 'Includes an depth knowledge of database using Oracle SQL plus');

1 row created.

SQL> INSERT INTO Module (Module_Code, Module_Title, Credits, Module_Description)
  2  VALUES
  3  ('MP12', 'Programming', 30, 'Covers fundamental and advanced programming concepts.');

1 row created.

SQL> INSERT INTO Module (Module_Code, Module_Title, Credits, Module_Description)
  2  VALUES
  3  ('MF13', 'Finance', 15, 'Focuses on financial management and analysis.');

1 row created.

SQL> INSERT INTO Module (Module_Code, Module_Title, Credits, Module_Description)
  2  VALUES
  3  ('MR14', 'Robotics', 30, 'Explores the design and implementation of robotic systems.');

1 row created.

SQL> INSERT INTO Module (Module_Code, Module_Title, Credits, Module_Description)
  2  VALUES
  3  ('MI16', 'Internet of Things', 20, 'Covers IoT architecture, protocols, and applications.');

1 row created.

SQL> INSERT INTO Module (Module_Code, Module_Title, Credits, Module_Description)
  2  VALUES
  3  ('MA15', 'Account', 10, 'Provides knowledge on accounting principles and practices.');

1 row created.

SQL> INSERT INTO Module (Module_Code, Module_Title, Credits, Module_Description)
  2  VALUES
  3  ('MC17', 'Cyber Security', 15, 'Focuses on securing networks and systems against cyber threats.');

1 row created.

SQL> INSERT INTO Module (Module_Code, Module_Title, Credits, Module_Description)
  2  VALUES
  3  ('MN18', 'Network OS', 20, 'Teaches the configuration and management of network operating systems.');

1 row created.

```

Figure 21 Inserting test data in Module Table

SQL> SELECT * FROM Module;			
MODULE_COD	MODULE_TITLE	CREDITS	MODULE_DESCRIPTION
MD11	Database	15	Includes a depth knowledge on database using oracle sql plus
MP12	Programming	30	Covers fundamental and advanced programming concepts.
MF13	Finance	15	Focuses on financial management and analysis.
MR14	Robotics	30	Explores the design and implementation of robotic systems.
MI16	Internet of Things	20	Covers IoT architecture, protocols, and applications.
MA15	Account	10	Provides knowledge on accounting principles and practices.
MC17	Cyber Security	15	Focuses on securing networks and systems against cyber threats.
MN18	Network OS	20	Teaches the configuration and management of network operating systems.

8 rows selected.

Figure 22 Selecting data from Module Table

5.2.4 Inserting test data in the Module_Student Table

```
SQL> INSERT INTO Module_Student (Module_Code, Student_ID)
  2  VALUES
  3  ('MD11','S248132');

1 row created.

SQL> INSERT INTO Module_Student (Module_Code, Student_ID)
  2  VALUES
  3  ('MP12','S248154');

1 row created.

SQL> INSERT INTO Module_Student (Module_Code, Student_ID)
  2  VALUES
  3  ('MF13','S248144');

1 row created.

SQL> INSERT INTO Module_Student (Module_Code, Student_ID)
  2  VALUES
  3  ('MR14','S248109');

1 row created.

SQL> INSERT INTO Module_Student (Module_Code, Student_ID)
  2  VALUES
  3  ('MI16','S248021');

1 row created.

SQL> INSERT INTO Module_Student (Module_Code, Student_ID)
  2  VALUES
  3  ('MN18','S248088');

1 row created.

SQL> INSERT INTO Module_Student (Module_Code, Student_ID)
  2  VALUES
  3  ('MA15','S248144');

1 row created.

SQL> INSERT INTO Module_Student (Module_Code, Student_ID)
  2  VALUES
  3  ('MC17','S248132');

1 row created.

SQL> INSERT INTO Module_Student (Module_Code, Student_ID)
  2  VALUES
  3  ('MC17','S248154');

1 row created.

SQL> INSERT INTO Module_Student (Module_Code, Student_ID)
  2  VALUES
  3  ('MD11','S248021');

1 row created.
```

Figure 23 Inserting test data to Module_Student Table

```

SQL> SELECT * FROM Module_Student;

MODULE_COD STUDENT_ID
----- -----
MA15      S248144
MC17      S248132
MC17      S248154
MD11      S248021
MD11      S248132
MF13      S248144
MI16      S248021
MN18      S248088
MP12      S248154
MR14      S248109

10 rows selected.

```

Figure 24 Selecting data from Module_Student Table

5.2.5 Inserting test data in the Teacher Table

```

SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Address, Phone, Email)
  2  VALUES
  3  ('T1102', 'Shyam Thapa', 'Lalitpur', '9800000022', 'shyam_islington@gmail.com');

1 row created.

SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Address, Phone, Email)
  2  VALUES
  3  ('T1103', 'Sita Rai', 'Bhaktapur', '9800000323', 'sita_islington@gmail.com');

1 row created.

SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Address, Phone, Email)
  2  VALUES
  3  ('T1104', 'Gita Shrestha', 'Pokhara', '9800000111', 'gita_islington@gmail.com');

1 row created.

SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Address, Phone, Email)
  2  VALUES
  3  ('T1105', 'Hari Basnet', 'Chitwan', '9800001122', 'hari_islington@gmail.com');

1 row created.

SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Address, Phone, Email)
  2  VALUES
  3  ('T1106', 'Nabin Bista', 'Dharan', '9800000556', 'nabin_islington@gmail.com');

1 row created.

SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Address, Phone, Email)
  2  VALUES
  3  ('T1107', 'Rita Gurung', 'Butwal', '9800000330', 'rita_islington@gmail.com');

1 row created.

SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Address, Phone, Email)
  2  VALUES
  3  ('T1108', 'Prakash Poudel', 'Biratnagar', '9800000099', 'prakash_islington@gmail.com');

1 row created.

SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Address, Phone, Email)
  2  VALUES
  3  ('T1101', 'Ram Khatrī', 'Kathmandu', '9800000001', 'ram_islington@gmail.com');

1 row created.

```

Figure 25 Inserting test data in Teacher Table

SQL> SELECT * FROM Teacher;				
TEACHER_ID	TEACHER_NAME	ADDRESS	PHONE	EMAIL
T1102	Shyam Thapa	Lalitpur	9800000022	shyam_islington@gmail.com
T1103	Sita Rai	Bhaktapur	98000000323	sita_islington@gmail.com
T1104	Gita Shrestha	Pokhara	98000000111	gita_islington@gmail.com
T1105	Hari Basnet	Chitwan	98000001122	hari_islington@gmail.com
T1106	Nabin Bista	Dharan	9800000556	nabin_islington@gmail.com
T1107	Rita Gurung	Butwal	98000000330	rita_islington@gmail.com
T1108	Prakash Poudel	Biratnagar	9800000099	prakash_islington@gmail.com
T1101	Ram Khatri	Kathmandu	9800000001	ram_islington@gmail.com

8 rows selected.

Figure 26 Selecting data from Teacher Table

5.2.6 Inserting test data in the Teacher_Module_Student Table

```
SQL> INSERT INTO Teacher_Module_Student (Module_Code, Student_ID, Teacher_ID)
  2 VALUES
  3 ('MD11', 'S248021', 'T1101');

1 row created.

SQL> INSERT INTO Teacher_Module_Student (Module_Code, Student_ID, Teacher_ID)
  2 VALUES
  3 ('MP12', 'S248154', 'T1102');

1 row created.

SQL> INSERT INTO Teacher_Module_Student (Module_Code, Student_ID, Teacher_ID)
  2 VALUES
  3 ('MF13', 'S248144', 'T1103');

1 row created.

SQL> INSERT INTO Teacher_Module_Student (Module_Code, Student_ID, Teacher_ID)
  2 VALUES
  3 ('MR14', 'S248109', 'T1104');

1 row created.

SQL> INSERT INTO Teacher_Module_Student (Module_Code, Student_ID, Teacher_ID)
  2 VALUES
  3 ('MA15', 'S248144', 'T1105');

1 row created.

SQL> INSERT INTO Teacher_Module_Student (Module_Code, Student_ID, Teacher_ID)
  2 VALUES
  3 ('MI16', 'S248021', 'T1106');

1 row created.

SQL> INSERT INTO Teacher_Module_Student (Module_Code, Student_ID, Teacher_ID)
  2 VALUES
  3 ('MC17', 'S248132', 'T1107');

1 row created.

SQL> INSERT INTO Teacher_Module_Student (Module_Code, Student_ID, Teacher_ID)
  2 VALUES
  3 ('MN18', 'S248088', 'T1108');

1 row created.
```

Figure 27 Inserting test data in Teacher_Module_Student Table

```

SQL> SELECT * FROM Teacher_Module_Student;

MODULE_COD STUDENT_ID TEACHER_ID
----- ----- -----
MA15      S248144    T1105
MC17      S248132    T1107
MD11      S248021    T1101
MF13      S248144    T1103
MI16      S248021    T1106
MN18      S248088    T1108
MP12      S248154    T1102
MR14      S248109    T1104

8 rows selected.

```

Figure 28 Selecting data from Teacher_Module_Student Table

5.2.7 Inserting test data in the Announcement Table

```

SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Date_Posted, Announcement_Description)
2 VALUES
3 (101, 'Class Update', TO_DATE('19-12-2024', 'DD-MM-YYYY'), 'There will not be lecture class tomorrow on 20 December.');
1 row created.

SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Date_Posted, Announcement_Description)
2 VALUES
3 (102, 'Teaching Material', TO_DATE('29-11-2024', 'DD-MM-YYYY'), 'The teaching material for database has been uploaded.');
1 row created.

SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Date_Posted, Announcement_Description)
2 VALUES
3 (103, 'Exam Time Table', TO_DATE('30-12-2024', 'DD-MM-YYYY'), 'The exam time table for this module has been uploaded.');
1 row created.

SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Date_Posted, Announcement_Description)
2 VALUES
3 (104, 'Class Update', TO_DATE('31-01-2024', 'DD-MM-YYYY'), 'Tomorrow there will not be any class.');
1 row created.

SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Date_Posted, Announcement_Description)
2 VALUES
3 (105, 'Teacher Absent', TO_DATE('23-12-2024', 'DD-MM-YYYY'), 'Due to the absence of Mr. Basnet sir, the class will be handled by Mrs. Karki.');
1 row created.

SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Date_Posted, Announcement_Description)
2 VALUES
3 (106, 'Class Update', TO_DATE('23-11-2024', 'DD-MM-YYYY'), 'The class time table has been changed for tomorrow only.');
1 row created.

SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Date_Posted, Announcement_Description)
2 VALUES
3 (107, 'Tomorrow requirement', TO_DATE('02-01-2025', 'DD-MM-YYYY'), 'All students are required to bring laptops for tomorrow tutorial class.')
4 ;

1 row created.

SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Date_Posted, Announcement_Description)
2 VALUES
3 (108, 'Admit Card', TO_DATE('04-05-2024', 'DD-MM-YYYY'), 'Students are requested to take the admit card from Student Service.');
1 row created.

SQL> INSERT INTO Announcement (Announcement_ID, Announcement_Title, Date_Posted, Announcement_Description)
2 VALUES
3 (109, 'Mandatory Class', TO_DATE('27-04-2024', 'DD-MM-YYYY'), 'Students are requested to attend the mandatory class.');
1 row created.

```

Figure 29 Inserting test data in Announcement Table

SQL> SELECT * FROM ANNOUNCEMENT;			
ANNOUNCEMENT_ID	ANNOUNCEMENT_TITLE	DATE_POST	ANNOUNCEMENT_DESCRIPTION
101	Class Update	19-DEC-24	There will not be lecture class tomorrow on 20 December.
102	Teaching Material	29-NOV-24	The teaching material for database has been uploaded.
103	Exam Time Table	30-DEC-24	The exam time table for this module has been uploaded.
104	Class Update	31-JAN-24	Tomorrow there will not be any class.
105	Teacher Absent	23-DEC-24	Due to the absence of Mr.Basnet sir, the class will be handled by Mrs. Karki.
106	Class Update	23-NOV-24	The class time table has been changed for tomorrow only.
107	Tomorrow requirement	02-JAN-25	All students are required to bring laptops for tomorrow tutorial class.
108	Admit Card	04-MAY-24	Students are requested to take the admit card from Student Service.
109	Mandatory Class	27-APR-24	Students are requested to attend the mandatory class.

9 rows selected.

Figure 30 Selecting data from Announcement Table

5.2.8 Inserting test data in the Announcement_Mod_Std_Tea Table

```
SQL> INSERT INTO Announcement_Mod_Std_Tea (Module_Code, Student_ID, Teacher_ID, Announcement_ID)
  2 VALUES
  3 ('MA15','S248144','T1105',101);

1 row created.

SQL> INSERT INTO Announcement_Mod_Std_Tea (Module_Code, Student_ID, Teacher_ID, Announcement_ID)
  2 VALUES
  3 ('MC17','S248132','T1107',102);

1 row created.

SQL> INSERT INTO Announcement_Mod_Std_Tea (Module_Code, Student_ID, Teacher_ID, Announcement_ID)
  2 VALUES
  3 ('MD11','S248021','T1101',103);

1 row created.

SQL> INSERT INTO Announcement_Mod_Std_Tea (Module_Code, Student_ID, Teacher_ID, Announcement_ID)
  2 VALUES
  3 ('MF13','S248144','T1103',104);

1 row created.

SQL> INSERT INTO Announcement_Mod_Std_Tea (Module_Code, Student_ID, Teacher_ID, Announcement_ID)
  2 VALUES
  3 ('MI16','S248021','T1106',105);

1 row created.

SQL> INSERT INTO Announcement_Mod_Std_Tea (Module_Code, Student_ID, Teacher_ID, Announcement_ID)
  2 VALUES
  3 ('MN18','S248088','T1108',106);

1 row created.

SQL> INSERT INTO Announcement_Mod_Std_Tea (Module_Code, Student_ID, Teacher_ID, Announcement_ID)
  2 VALUES
  3 ('MP12','S248154','T1102',107);

1 row created.

SQL> INSERT INTO Announcement_Mod_Std_Tea (Module_Code, Student_ID, Teacher_ID, Announcement_ID)
  2 VALUES
  3 ('MR14','S248109','T1104',108);

1 row created.

SQL> INSERT INTO Announcement_Mod_Std_Tea (Module_Code, Student_ID, Teacher_ID, Announcement_ID)
  2 VALUES
  3 ('MR14','S248109','T1104',109);

1 row created.
```

Figure 31 Inserting test data in Announcement_Module_Student Table

MODULE_COD	STUDENT_ID	TEACHER_ID	ANNOUNCEMENT_ID
MA15	S248144	T1105	101
MC17	S248132	T1107	102
MD11	S248021	T1101	103
MF13	S248144	T1103	104
MI16	S248021	T1106	105
MN18	S248088	T1108	106
MP12	S248154	T1102	107
MR14	S248109	T1104	108
MR14	S248109	T1104	109

9 rows selected.

Figure 32 Selecting data from Announcement_Mod_Std_Tea Table

5.2.9 Inserting test data in the Resource Table

```

SQL> INSERT INTO Resources (Resource_ID, Resource_Title, Resource_Duration_Days, Type)
  2  VALUES
  3  (101, 'Amazon Web Service', 30, 'Course');

1 row created.

SQL> INSERT INTO Resources (Resource_ID, Resource_Title, Resource_Duration_Days, Type)
  2  VALUES
  3  (102, 'Educational Video', 5, 'Video');

1 row created.

SQL> INSERT INTO Resources (Resource_ID, Resource_Title, Resource_Duration_Days, Type)
  2  VALUES
  3  (103, 'Robotics Basics', 15, 'Video');

1 row created.

SQL> INSERT INTO Resources (Resource_ID, Resource_Title, Resource_Duration_Days, Type)
  2  VALUES
  3  (104, 'SQL Quiz', 10, 'Quiz');

1 row created.

SQL> INSERT INTO Resources (Resource_ID, Resource_Title, Resource_Duration_Days, Type)
  2  VALUES
  3  (105, 'Ethical Hacking', 10, 'Document');

1 row created.

SQL> INSERT INTO Resources (Resource_ID, Resource_Title, Resource_Duration_Days, Type)
  2  VALUES
  3  (106, 'Network Security', 10, 'Document');

1 row created.

SQL> INSERT INTO Resources (Resource_ID, Resource_Title, Resource_Duration_Days, Type)
  2  VALUES
  3  (107, 'Financial Accounting', 10, 'Case Study');

1 row created.

```

Figure 33 Inserting test data in Resources Table

```

SQL> SELECT * FROM Resources;

RESOURCE_ID RESOURCE_TITLE      RESOURCE_DURATION_DAYS TYPE
-----  -----
    101 Amazon Web Service          30 Course
    102 Educational Video           5 Video
    103 Robotics Basics            15 Video
    104 SQL Quiz                   10 Quiz
    105 Ethical Hacking            10 Document
    106 Network Security           10 Document
    107 Financial Accounting       10 Case Study

7 rows selected.

```

Figure 34 Selecting data from Resources Table

5.2.10 Inserting test data in the Resource_Module_Student Table

```

SQL> INSERT INTO Resource_Module_Student (Module_Code, Student_ID, Resource_ID, Completion_Status, Completion_Date)
  2  VALUES
  3  ('MI16', 'S248021','101','Completed',TO_DATE('03-01-2025', 'DD-MM-YYYY'));

1 row created.

SQL> INSERT INTO Resource_Module_Student (Module_Code, Student_ID, Resource_ID, Completion_Status, Completion_Date)
  2  VALUES
  3  ('MA15', 'S248144','102','No',NULL);

1 row created.

SQL> INSERT INTO Resource_Module_Student (Module_Code, Student_ID, Resource_ID, Completion_Status, Completion_Date)
  2  VALUES
  3  ('MR14', 'S248109','103','Completed',TO_DATE('31-12-2024', 'DD-MM-YYYY'));

1 row created.

SQL> INSERT INTO Resource_Module_Student (Module_Code, Student_ID, Resource_ID, Completion_Status, Completion_Date)
  2  VALUES
  3  ('MD11', 'S248021','104','Completed',TO_DATE('29-12-2024', 'DD-MM-YYYY'));

1 row created.

SQL> INSERT INTO Resource_Module_Student (Module_Code, Student_ID, Resource_ID, Completion_Status, Completion_Date)
  2  VALUES
  3  ('MN18', 'S248088','105','Completed',TO_DATE('30-12-2024', 'DD-MM-YYYY'));

1 row created.

SQL> INSERT INTO Resource_Module_Student (Module_Code, Student_ID, Resource_ID, Completion_Status, Completion_Date)
  2  VALUES
  3  ('MC17', 'S248132','106','Completed',TO_DATE('25-12-2024', 'DD-MM-YYYY'));

1 row created.

SQL> INSERT INTO Resource_Module_Student (Module_Code, Student_ID, Resource_ID, Completion_Status, Completion_Date)
  2  VALUES
  3  ('MF13', 'S248144','107','Completed',TO_DATE('25-11-2024', 'DD-MM-YYYY'));

1 row created.

SQL> INSERT INTO Resource_Module_Student (Module_Code, Student_ID, Resource_ID, Completion_Status, Completion_Date)
  2  VALUES
  3  ('MP12', 'S248154','102','Completed',TO_DATE('28-11-2024', 'DD-MM-YYYY'));

1 row created.

```

Figure 35 Inserting test data in Resource_Module_Student Table

```

SQL> SELECT * FROM Resource_Module_Student;

MODULE_COD STUDENT_ID RESOURCE_ID COMPLETION COMPLETIO
----- ----- -----
MI16      S248021      101 Completed   03-JAN-25
MA15      S248144      102 No          0
MR14      S248109      103 Completed   31-DEC-24
MD11      S248021      104 Completed   29-DEC-24
MN18      S248088      105 Completed   30-DEC-24
MC17      S248132      106 Completed   25-DEC-24
MF13      S248144      107 Completed   25-NOV-24
MP12      S248154      102 Completed   28-NOV-24

8 rows selected.

```

Figure 36 Selecting data from Resource_Module_Student Table

5.2.11 Inserting test data in the Assessment Table

```

SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2 VALUES
  3 (101, 'Database Coursework', TO_DATE('29-12-2024', 'DD-MM-YYYY'), 100);

1 row created.

SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2 VALUES
  3 (102, 'Circuit Designing', TO_DATE('02-01-2025', 'DD-MM-YYYY'), 100);

1 row created.

SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2 VALUES
  3 (103, 'Case Study', TO_DATE('05-01-2025', 'DD-MM-YYYY'), 100);

1 row created.

SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2 VALUES
  3 (104, 'Logbook', TO_DATE('29-12-2025', 'DD-MM-YYYY'), 100);

1 row created.

SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2 VALUES
  3 (105, 'Logbook', TO_DATE('29-12-2025', 'DD-MM-YYYY'), 100);

1 row created.

SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2 VALUES
  3 (106, 'Final Report', TO_DATE('07-01-2025', 'DD-MM-YYYY'), 100);

1 row created.

SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2 VALUES
  3 (107, 'Presentation', TO_DATE('31-12-2024', 'DD-MM-YYYY'), 100);

1 row created.

SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2 VALUES
  3 (108, 'Research Paper', TO_DATE('02-01-2025', 'DD-MM-YYYY'), 100);

1 row created.

SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2 VALUES
  3 (109, 'SQL Queries', TO_DATE('11-01-2025', 'DD-MM-YYYY'), 100);

1 row created.

```

Figure 37 Inserting test data in Assessment Table

SQL> SELECT * FROM Assessment;			
ASSESSMENT_ID	ASSESSMENT_TITLE	DEADLINE	WEIGHTAGE
101	Database Coursework	29-DEC-24	100
102	Circuit Designing	02-JAN-25	100
103	Case Study	05-JAN-25	100
104	Logbook	29-DEC-25	100
105	Logbook	29-DEC-25	100
106	Final Report	07-JAN-25	100
107	Presentation	31-DEC-24	100
108	Research Paper	02-JAN-25	100
109	SQL Queries	11-JAN-25	100

9 rows selected.

Figure 38 Selecting data from Assessment Table

5.2.12 Inserting test data in the Result Table

```
SQL> INSERT INTO Result (Module_Code, Student_ID, Assessment_ID, Marks_Obtained, Grade)
  2  VALUES
  3  ('MD11','S248021','101',80,'A');

1 row created.

SQL> INSERT INTO Result (Module_Code, Student_ID, Assessment_ID, Marks_Obtained, Grade)
  2  VALUES
  3  ('MI16','S248021','102',80,'A');

1 row created.

SQL> INSERT INTO Result (Module_Code, Student_ID, Assessment_ID, Marks_Obtained, Grade)
  2  VALUES
  3  ('MR14','S248109','103',30,NULL);

1 row created.

SQL> INSERT INTO Result (Module_Code, Student_ID, Assessment_ID, Marks_Obtained, Grade)
  2  VALUES
  3  ('MN18','S248088','104',60,'B');

1 row created.

SQL> INSERT INTO Result (Module_Code, Student_ID, Assessment_ID, Marks_Obtained, Grade)
  2  VALUES
  3  ('MA15','S248144','105',60,'B');

1 row created.

SQL> INSERT INTO Result (Module_Code, Student_ID, Assessment_ID, Marks_Obtained, Grade)
  2  VALUES
  3  ('MC17','S248132','106',90,'A');

1 row created.

SQL> INSERT INTO Result (Module_Code, Student_ID, Assessment_ID, Marks_Obtained, Grade)
  2  VALUES
  3  ('MF13','S248144','107',90,'A');

1 row created.

SQL> INSERT INTO Result (Module_Code, Student_ID, Assessment_ID, Marks_Obtained, Grade)
  2  VALUES
  3  ('MP12','S248154','108',50,'C');

1 row created.

SQL> INSERT INTO Result (Module_Code, Student_ID, Assessment_ID, Marks_Obtained, Grade)
  2  VALUES
  3  ('MD11','S248088','109',80,'A');

1 row created.
```

Figure 39 Inserting test data in Result Table

```
SQL> UPDATE RESULT
  2  SET Grade = 'F'
  3  WHERE Student_ID = 'S248109'
  4  AND Assessment_ID = '103';

1 row updated.
```

```
SQL> UPDATE RESULT
  2  SET Grade = NULL, Marks_Obtained = 0
  3  WHERE STUDENT_ID='S248088'
  4  AND Assessment_ID =109;

1 row updated.
```

```
SQL> SELECT * FROM RESULT;

MODULE_COD STUDENT_ID ASSESSMENT_ID MARKS_OBTAINED G
----- ----- ----- -----
MD11      S248021        101          80  A
MI16      S248021        102          80  A
MR14      S248109        103          30  F
MN18      S248088        104          60  B
MA15      S248144        105          60  B
MC17      S248132        106          90  A
MF13      S248144        107          90  A
MP12      S248154        108          50  C
MD11      S248088        109          0

9 rows selected.
```

Figure 40 Selecting data from Result Table

6 Database Query

6.1 Information Query

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

```
SQL> SELECT P.Program_Code, P.Program_Name, COUNT(S.Student_ID) AS Total_Students
  2  FROM
  3 Program P
  4 LEFT JOIN
  5 Student S
  6 ON
  7 P.Program_Code = S.Program_Code
  8 GROUP BY P.Program_Code, P.Program_Name;

PROGRAM_CODE PROGRAM_NAME          TOTAL_STUDENTS
-----  -----
BB101      B.Business Administration           1
CM103      B.Sc.Multimedia                   1
BH102      B.Hotel Management                 0
CN102      B.Sc.Networking                  2
CC101      B.Sc.Computing                    3
CA104      B.Sc.AI                         1

6 rows selected.
```

Figure 41 Listing all available programs along with total number of students enrolled in each

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

```
SQL> SELECT AB.Module_Code, M.Module_Title, A.Announcement_ID, A.Announcement_Title, A.Date_Posted, AB.Teacher_ID, A.Announcement_Description
  2  FROM
  3 Announcement A
  4 JOIN
  5 Announcement_Mod_Std_Tea AB
  6 ON
  7 A.Announcement_ID = AB.Announcement_ID
  8 JOIN
  9 Module M
 10 ON
 11 AB.Module_Code = M.Module_Code
 12 WHERE
 13 M.Module_Code = 'MR14'
 14 AND
 15 A.Date_Posted BETWEEN TO_DATE ('01-05-2024', 'DD-MM-YYYY') AND TO_DATE ('28-05-2024', 'DD-MM-YYYY');

MODULE_CODE MODULE_TITLE          ANNOUNCEMENT_ID ANNOUNCEMENT_TITLE    DATE_POST TEACHER_ID ANNOUNCEMENT_DESCRIPTION
-----  -----
MR14        Robotics             108            Admit Card       04-MAY-24 T1104   Students are requested to take the admit card from Student Service.
```

Figure 42 Listing all the announcements made for module MR14 from 1st May 2024 to 28th May 2024

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

```
SQL> SELECT M.Module_Code,
  2   M.Module_Title,
  3   M.Module_Description,
  4   COUNT(R.Resource_ID) AS Total_Resource_Uploaded
  5   FROM Module M
  6   LEFT JOIN
  7   Resource_Module_Student R
  8   ON
  9   M.Module_Code = R.Module_Code
 10  WHERE
 11  M.Module_Title LIKE 'D%'
 12 GROUP BY M.Module_Code, M.Module_Title, M.Module_Description;
```

MODULE_CODE	MODULE_TITLE	MODULE_DESCRIPTION	TOTAL_RESOURCE_UPLOADED
MD11	Database	Includes an depth knowledge of database using Oracle SQL plus	1

Figure 43 Listing the names of modules beginning with the letter 'D' along with the total number of resources uploaded

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

```
SQL> SELECT S.Student_Name, P.Program_Name, M.Module_Title, A.Assessment_Title
  2   FROM
  3   Student S
  4   JOIN
  5   Program P
  6   ON
  7   S.Program_Code = P.Program_Code
  8   JOIN Result R ON S.Student_ID = R.Student_ID AND R.Module_Code = 'MD11'
  9   JOIN Assessment A ON R.Assessment_ID = A.Assessment_ID
 10  JOIN Module M ON R.Module_Code = M.Module_Code
 11 WHERE R.Grade IS NULL;
```

STUDENT_NAME	PROGRAM_NAME	MODULE_TITLE	ASSESSMENT_TITLE
Pranita Shakya	B.Sc.Computing	Database	SQL Queries

Figure 44 Listing the names of all students along with their enrolled program who have not submitted any assessments for module MD11

As per the assumption here, a student is said to have not submitted any assessments if the grade in his/her result is null.

6.1.5 List all the teachers who teach more than one module

```
SQL> SELECT T.Teacher_Name,
  2   COUNT(M.Module_Code) AS Total_Module
  3   FROM Teacher T
  4   JOIN Teacher_Module_Student TMS ON T.Teacher_ID = TMS.Teacher_ID
  5   JOIN Module M ON TMS.Module_Code = M.Module_Code
  6   GROUP BY T.Teacher_ID, T.Teacher_Name
  7   HAVING COUNT(M.Module_Code) > 1;

no rows selected
```

Figure 45 Listing all teachers who teach more than one module

The output is null because, as per the assumption, a teacher teaches only one module.

6.2 Transaction Query

6.2.1 Identify the module that has the latest assessment deadline.

MODULE_COD	MODULE_TITLE	ASSESSMENT_TITLE	DEADLINE
MN18	Network OS	Logbook	29-DEC-25
MA15	Account	Logbook	29-DEC-25

Figure 46 Listing the module having latest assessment deadline

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

STUDENT_ID	STUDENT_NAME	TOTAL_SCORE
S248132	Alex Chhetri	90
S248021	Prabin Ghimire	80
S248144	Shivangi Thapa	75

Figure 47 Listing the top three students who have the highest total score across all modules.

6.2.3 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(DISTINCT A.Assessment_ID) AS Total_Assessments,
 3 AVG(R.Marks_Obtained) AS Average_Score
 4 FROM Program P
 5 JOIN Student S ON P.Program_Code = S.Program_Code
 6 JOIN Result R ON S.Student_ID = R.Student_ID
 7 JOIN Assessment A ON R.Assessment_ID = A.Assessment_ID
 8 GROUP BY P.Program_Name
 9 ORDER BY P.Program_Name;
```

PROGRAM_NAME	TOTAL_ASSESSMENTS	AVERAGE_SCORE
B.Business Administration	2	75
B.Sc.AI	1	30
B.Sc.Computing	4	55
B.Sc.Networking	2	70

Figure 48 Listing the total number of assessments for each program and the average score across all assessments in those programs

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

```
SQL> SELECT S.Student_ID,
 2 S.Student_Name,
 3 R.Marks_Obtained
 4 FROM Result R
 5 JOIN Student S ON R.Student_ID = S.Student_ID
 6 JOIN Module M ON R.Module_Code = M.Module_Code
 7 WHERE M.Module_Title = 'Database' AND R.Marks_Obtained > (
 8 SELECT AVG(R2.Marks_Obtained)
 9 FROM Result R2
10 JOIN Module M2 ON R2.Module_Code = M2.Module_Code
11 WHERE M2.Module_Title = 'Database'
12 );
```

STUDENT_ID	STUDENT_NAME	MARKS_OBTAINED
S248021	Prabin Ghimire	80

Figure 49 Listing the students who have scored the average score in Database Module

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

```
SQL> SELECT S.Student_ID,
 2   S.Student_Name,
 3   M.Module_Title,
 4   SUM(R.Marks_Obtained) AS Total_Marks,
 5   CASE
 6     WHEN SUM(R.Marks_Obtained) >= 40 THEN 'Passed'
 7     ELSE 'Failed'
 8   END AS Remarks
 9   FROM Result R
10  JOIN Student S ON R.Student_ID = S.Student_ID
11  JOIN Module M ON R.Module_Code = M.Module_Code
12  WHERE M.Module_Code = 'MR14'
13  GROUP BY S.Student_ID, S.Student_Name, M.Module_Title
14  ORDER BY S.Student_Name;
```

STUDENT_ID	STUDENT_NAME	MODULE_TITLE	TOTAL_MARKS	REMARK
S248109	Rubina Shrestha	Robotics	30	Failed

Figure 50 Displaying whether a student has passed or failed as remarks as per their total aggregate marks obtained in module MR14.

7 Critical Evaluation

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

The Database module offers a comprehensive understanding of designing, managing and querying databases. It also acknowledges key concepts of entity, attributes, entity relationship diagram, relation, modality, structured query languages and its types.

The Database module helps to gain practical skills in database management systems (DBMS), data structuring, and SQL programming. These skills play critical roles in career fields like software development, database management, and data analytics.

The knowledge of databases goes along with programming languages like Python, Java and C#. The database can be integrated into software applications for dynamic data handling. Also, the SQL codes can be referred to in those programming languages. The ERD (Entity Relationship Diagram) construction can also go along with subjects such as software engineering for project management. The basics of data storing, processing and retrieving through this module can create a foundation for data analysis. The database concept also links up with networking modules for understanding the data flow in different types of database networked environments such as centralized or distributed databases.

7.2 Critical Assessment of Coursework

This coursework offers a diverse perspective, which is why a thorough analysis of the program's strengths, weaknesses and functioning is required. The particular coursework offers an important contribution to the database, though some lack of clarity in certain areas undermines the overall performance. The coursework, with a mixture of theoretical and practical tasks, has helped to design and manage a database effectively.

Upon deeper examination, this study presents practical knowledge on creating a database, establishing relations between the entities, normalizing them and later showcasing them in SQL. Tackling the problem raised while constructing an Entity Relationship Diagram and performing normalizing and SQL queries has helped in building critical analysis. The coursework being focused on Oracle SQL has limited the exposure to other advanced tools such as NoSQL, Microsoft SQL Server, etc. However, the coursework has helped in gain an in-depth understanding of what a database is, how an individual can create a system based on a given scenario, identify the entities, analyse relation and modality between them, normalize

the data from unnormalised form to third normal form, working with SQL, creating user, giving permissions, creating tables, inserting data and performing queries.

8 Creating Dump File

```
PS C:\Users\najibthapa1> cd \\Mac\Home\Documents\College\2\1\coursework\db\23050403_NajibThapa
PS Microsoft.PowerShell.Core\FileSystem:\\Mac\Home\Documents\College\2\1\coursework\db\23050403_NajibThapa> mkdir dmpfile

Directory: \\Mac\Home\Documents\College\2\1\coursework\db\23050403_NajibThapa

Mode                LastWriteTime         Length Name
----                -----           -----   -----
d----      1/21/2025 11:12 AM            0     dmpfile

PS Microsoft.PowerShell.Core\FileSystem:\\Mac\Home\Documents\College\2\1\coursework\db\23050403_NajibThapa> cd dmpfile
PS Microsoft.PowerShell.Core\FileSystem:\\Mac\Home\Documents\College\2\1\coursework\db\23050403_NajibThapa\dmpfile> exp NajibThapa/23050403 file=najibthapa.dmp

Export: Release 11.2.0.2.0 - Production on Tue Jan 21 11:13:44 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 WE8MSWIN1252 character set and AL16UTF16 NCHAR character set
server uses AL32UTF8 character set (possible charset conversion)
. exporting pre-schema procedural objects and actions
. exporting foreign function library names for user NAJIBTHAPA
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user NAJIBTHAPA
About to export NAJIBTHAPA's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export NAJIBTHAPA's tables via Conventional Path ...
. . . exporting table ANNOUNCEMENT          9 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . . exporting table ANNOUNCEMENT_MOD_STD_TEA    9 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . . exporting table ASSESSMENT               9 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . . exporting table MODULE                  8 rows exported
EXP-00091: Exporting questionable statistics.

Activate Windows
Go to Settings to activate Windows.
```

Figure 51 Creating Dump File 1

```

. . exporting table          MODULE           8 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table          MODULE_STUDENT    10 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table          PROGRAM          6 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table          RESOURCES        7 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table          RESOURCE_MODULE_STUDENT 8 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table          RESULT           9 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table          STUDENT          8 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table          TEACHER          8 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. . exporting table          TEACHER_MODULE_STUDENT 8 rows exported
EXP-00091: Exporting questionable statistics.
EXP-00091: Exporting questionable statistics.
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting operators
. exporting referential integrity constraints
. exporting triggers
. exporting indextypes
. exporting bitmap, functional and extensible indexes
. exporting posttables actions
. exporting materialized views
. exporting snapshot logs
. exporting job queues
. exporting refresh groups and children
. exporting dimensions
. exporting post-schema procedural objects and actions
. exporting statistics
Export terminated successfully with warnings.

```

Figure 52 Creating Dump File 2

9 Drop Table

```
SQL> DROP TABLE Result;  
Table dropped.  
  
SQL> DROP TABLE Assessment;  
Table dropped.  
  
SQL> DROP TABLE Resource_Module_Student;  
Table dropped.  
  
SQL> DROP TABLE Resources;  
Table dropped.  
  
SQL> DROP TABLE Announcement_Mod_Std_Tea;  
Table dropped.  
  
SQL> DROP TABLE Announcement;  
Table dropped.  
  
SQL> DROP TABLE Teacher_Module_Student;  
Table dropped.  
  
SQL> DROP TABLE Teacher;  
Table dropped.  
  
SQL> DROP TABLE Module_Student;  
Table dropped.  
  
SQL> DROP TABLE Module;  
Table dropped.  
  
SQL> DROP TABLE Student;  
Table dropped.  
  
SQL> DROP TABLE Program;  
Table dropped.
```

Figure 53 Dropping Tables