



CC5051NI Databases

100% Individual Coursework

Autumn 2024

Credit: 15 Semester Long Module

Student Name: Niran Bhatta

London Met ID: 23047617

Assignment Submission Date: December 31, 2024

Word Count: 3676

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.

Contents

1) Introduction.....	4
2) Current business activities and operations	4
3) Business Rule	4
4) Assumptions	5
5) Entities and attributes identification.....	5
6) Entity Relationship Diagram (ERD)	7
7) Normalization	7
8) Unnormalized Form (UNF)	8
9) First Normal Form (1NF).....	8
10) Second Normal Form (2NF).....	10
11) Third Normal Form(3NF).....	12
12) Conclusion	15
13) Data Dictionary	16
Final ERD	19
14) Implementation.....	20
15) Querying of Database.....	36
17) Critical Evaluation.....	38
18) Bibliography	Error! Bookmark not defined.

Table of Tables

Table 1 Student table after 3NF.....	16
Table 2 Program table after 3NF.....	16
Table 3 Module table after 3NF.....	16
Table 4 Teacher table after 3NF	17
Table 5 Assessment table after 3NF.....	17
Table 6 Announcement table after 3NF	17
Table 7 Resource table after 3NF.....	17
Table 8 Student_Module table after 3NF	18
Table 9 Module_Teacher table after 3NF.....	18
Table 10 Teacher_Announcement table after 3NF	18
Table 11 Module_Resource table after 3NF.....	18
Table 12 Module_Assessment table after 3NF	19

Table of Figures

Figure 1 Final ERD	19
Figure 2 Creating new user	20
Figure 3 Creating student table	21
Figure 4 Creating program table	21
Figure 5 Creating module table	21
Figure 6 Creating Student_Module table	22
Figure 7 Creating teacher table	22
Figure 8 Creating Module_Teacher table	23
Figure 9 Creating announcement	23
Figure 10 Creating Teacher_Announcement table	24
Figure 11 Creating Resources table	24
Figure 12 Creating Stud_Module_Resource	25
Figure 13 Creating assessment table	25
Figure 14 Creating Module_Asst table	26
Figure 15 Inserting values in program table	26
Figure 16 Inserting values in module tabe	27
Figure 17 17 Inserting values in student table:	28
Figure 18 Inserting values in teacher table	29
Figure 19 Inserting values in student_module	30
Figure 20 Inserting values in Module_Teacher	31
Figure 21 Inserting values in Announcement table	32
Figure 22 Inserting values in Techer_Announcement	33
Figure 23 Inserting values in Resource table	34
Figure 24 Inserting values in assessment	35
Figure 25 Inserting values in Module_Asst table	36

1) Introduction

E Classroom Platform is a tool for online education, using technological aids to promote a regular, organized learning environment. The strong side of it is that this will contribute to a continuous educational journey by making use of modern technologies, whereby the content learned will be readily available to learners and easy to monitor and follow. With this comes more flexibility in learning to facilitate diversities in learning and educational needs requirements crucial in a contemporary academic landscape.

Miss Mary wishes to develop an eLearning platform known as the E Classroom System, which can connect student and teacher in a more organized and interactive manner. It would provide several academic programs, such as Computing and Multimedia, with several required modules that the student needs to be studied step by step, so that it becomes easy to follow their studies step by step. Teachers will be able to assign assessments and give feedback on students' work while also sharing helpful resources like video lectures and quizzes. The platform will be easy to use and accessible on different devices, ensuring that students can learn from anywhere. All the information regarding students, programs, modules, and assessment results will be managed with the help of a strong database. Finally, Miss Mary wishes this platform would make learning fun and more effective for all parties.

2) Current business activities and operations

The business activities of the E Classroom Platform operational procedure is explained below in a short paragraph:

The student may get enrolled into different programs, and each program may have multiple modules that may fit the fields of study. Then teachers are allocated to different modules. Module allocations are assigned to educators within the module to provide content and support to students, so each module is well supported. Then the program module is managed. Every program consists of multiple modules, hence requiring proper organization, management, and allocation of education resources to present a structured curriculum. The resources are allocated in the platform. The platform is designed to allow access to a variety of learning resources such as video lectures, assignments, and quizzes that must be completed sequentially in order to facilitate progressive learning. After that, the students' work is marked, and results for each assessment are generated which students can then view. This activity is important in terms of reflecting student performance accurately.

3) Business Rule

- 1) Each student must enroll in just one program, which can have one or more students.
Students are not allowed to join multiple programs until they finish the first one.
- 2) A teacher can instruct several modules, and a module can be taught by different

teachers

- 3) Each module should include several assignments, and each assignment must relate to its specific module.
- 4) Programs are made up of various modules that students need to complete to graduate.
- 5) Teachers can access multiple resources related to the module
- 6) An assignment will only be graded if it is submitted by the deadline.
- 7) Teachers can only make announcements for the modules they are assigned to, and each announcement must relate to a specific module.
- 8) Results should be published according to the student's program and module.

4) Assumptions

- A) One module can have many teachers assigned to it
- B) Once submitted, students are not allowed to change or remove any assessments.
- C) A single teacher can make multiple announcements.

5) Entities and attributes identification

Entities and attributes

Entity:

An entity is anything that exists in the real world with independent existence. Each entity is identifiable and can be differentiated from other objects. It has a unique identifier. For example, some entities in a gadget store database are customer, order, and product.

Attribute:

The characteristics of an entity are called attributes. Attributes consist of three types of keys named primary, foreign and non_key. For instance, in students, some of the attributes are Student_ID, Student_Name, and Student_PhoneNumber.

Initial ERD

The entities, attributes and primary key are displayed in table below

STUDENT

S.N.	Attribute	Data type	Size	Constraint
1	Student_ID	Number	18	Primary Key
2	Student_Name	Character	35	Not Null
3	Enrolled_Date	Date	10	Not null
4	Email	Character	35	Unique
5	DOB	Date	10	Not null

PROGRAM

S.N.	Attribute	Datatype	Size	Constraint
1	Program_Name	Character	22	Primary Key
2	Program_ID	Number	18	Not null
3	Program_Description	Character	200	Not null
4	Program_Duration	Number	6	

MODULES

S.N.	Attribute	Datatype	Size	Constraint
1	Module_ID	Number	12	Not null
2	Module_Name	Character	22	Not null
3	Credit_Hours	Number	5	Not null
4	Teacher_Name	Character	30	Not null
5	Teacher_ID	Number	12	Not null
6	Contact	Number	15	Unique
7	Specialization	Character	20	Not null
8	Resource_ID	Number	12	Not null
9	Resource_Title	Character	30	Not null
10	Resource_Duration	Number	5	Not null
11	Resource_Type	Character	25	Not null
12	Resource_Status	Character	30	Not Null
13	Sequence	Character	10	Unique
14	Assessment_ID	Number	22	Not null
15	Assssment_Title	Character	25	Not null
16	Assessment_Description	Character	1200	Not null
17	Assessment_Status	Character	30	Not Null
18	Weightage	Number	5	Not null
19	Deadline	Date	10	Not null
20	Result_Marks	Number	5	Not null
21	Grade	Character	5	Not null
22	Result_Date	Date	10	Not null
23	Announcement_ID	Number	25	Not null
24	Date_Posted	Date	10	Not null
25	Content	Character	1200	Not null

6) Entity Relationship Diagram (ERD)

The entity relationship diagram of the business rule is shown below:

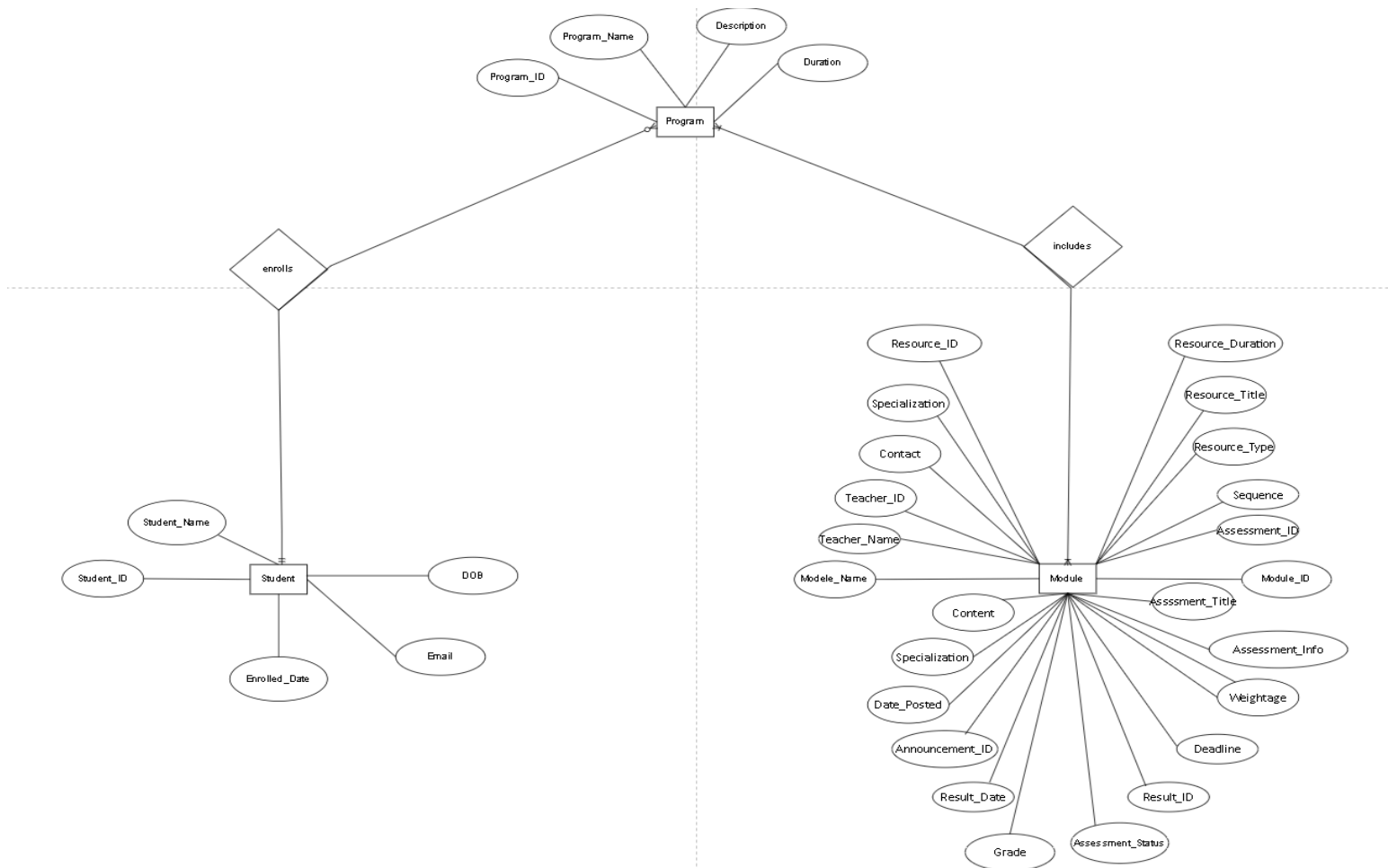


Figure 1 ERD (Entity Relationship) Diagram

7) Normalization

In databases, normalization is a method for arranging information in a smart way. The main purposes of normalization are to cut down on duplicate data and to make sure the data stays consistent. When we use normalization, it helps us manage tasks like adding, removing, or changing records easily, like how cleaning up your room helps you find your favorite book faster. Normalization means splitting your data into different tables and creating connections between them based on certain rules. You can think of each step in normalization to tidy up your room until everything is in its right place and easy to reach.

8) Unnormalized Form (UNF)

Description:

Unnormalized Form is the first state of data organization, often derived from a single source of requirements. In this form, data may include repeated groups and does not adhere to any structured format.

Characteristics:

- Typically, the structure straightforward list of attributes.
- Contains a collection of data with possible redundancy.
- Repeating groups may exist, indicated by curly braces.

Student: { Student_Name, Student_ID, Enrolled_Date, Email, DOB, Program_Name, Program_ID, Program_Description, Program_Duration, { Module_ID, Module_Name, Credit_Hours, { Teacher_Name, Teacher_ID, Contact, Specialization, { Announcement_ID, Date_Posted, Content } }, Resource_ID, Resource_Title, Resource_Duration, Resource_Type, Sequence } { Assessment_ID, Assessment_Title, Assessment_Info, Assessment_Status, Weightage, Deadline, Result_Marks, Grade, Result_Date } }

What it looks like: Imagine you have a list with lots of repeated information about students, their courses, and assigned tasks all mixed.

The Problem: A lot of information might be redundant, making it difficult to retrieve or update later.

9) First Normal Form (1NF)

First Normal Form (1NF) introduces basic structure to the data by ensuring that each cell contains only one value and all entries in a column have same type

Now, let's take the first step towards organizing that closet by ensuring that each item has its place. This is First Normal Form (1NF). If something tends to repeat, like a student taking multiple courses, we need to create separate shelves for those items.

For instance, instead of keeping every course listed under each student's name, we create a separate table that neatly lists each student alongside their assigned courses. This way, you can easily add or change courses without messing up other records.

Process to Achieve 1NF:

{ Module_ID, Module_Name, Credit_Hours, { Teacher_Name, Teacher_ID, Contact, Specialization, { Announcement_ID, Date_Posted, Content } }, Resource_ID, Resource_Title, Resource_Duration, Resource_Type, Sequence } { Assessment_ID, Assessment_Title, Assessment_Info, Assessment_Status, Weightage, Deadline, Result_Marks, Grade, Result_Date } }

Result After Normalization (1NF)

- STUDENT (1NF): Student_Name, Student_ID, Enrolled_Date, Email, DOB, Program_Name, Program_ID, Program_Description, Program_Duration}
- Student_Module (1NF): Student_ID, Module_ID, Module_Name, Credit_Hours}
- Module_Teacher (1NF): Student_ID, Module_ID, Teacher_Name, Teacher_ID, Contact, Specialization}
- Module_Teacher_Announcement (1NF): Student_ID, Module_ID, Teacher_ID, Announcement_ID, Date_Posted, Content}
- Student_Module_Resource (1NF): Student_ID, Module_ID, Resource_ID, Resource_Title, Resource_Duration, Resource_Type, Resource_Status, Sequence}
- Student_Module_Assessment (1NF): Student_ID, Module_ID, Assessment_ID, Assessment_Title, Assessment_Info, Assessment_Status, Weightage, Deadline, Result_Marks, Grade, Result_Date}

S.N.	Attribute	Datatype	Size	Constraint
1	Module_ID	Number	12	Not null
2	Module_Name	Character	22	Not null
3	Credit_Hours	Number	5	Not null
4	Teacher_Name	Character	30	Not null
5	Teacher_ID	Number	12	Not null
6	Contact	Number	15	Unique
7	Specialization	Character	20	Not null
8	Resource_ID	Number	12	Not null
9	Resource_Title	Character	30	Not null
10	Resource_Duration	Number	5	Not null
11	Resource_Type	Character	25	Not null
12	Sequence	Character	10	Unique
13	Assessment_ID	Number	22	Not null
14	Assssment_Title	Character	25	Not null
15	Assessment_Info	Character	1200	Not null
16	Assessment_Status	Character	30	Not Null
17	Weightage	Number	5	Not null
18	Deadline	Date	10	Not null
19	Result_Marks	Number	5	Not null
20	Grade	Character	5	Not null
21	Result_Date	Date	10	Not null
22	Announcement_ID	Number	25	Not null
23	Date_Posted	Date	10	Not null
24	Content	Character	1200	Not null

Rules to Follow:

- Each column must have unique values.
- Repeating groups should be kept in separate tables, thus removing redundancy.

10) Second Normal Form (2NF)

Second Normal Form (2NF) takes 1NF a step further by eliminating partial dependencies. In this form, all non_key attributes must depend on the entire primary key.

Process to Achieve 2NF:

1) Student:

Since students have only one primary key, the 2NF of student is given as:

Student(2NF) : (Student_Name, Student_ID, Enrolled_Date, Email, DOB, Program_Name, Program_ID, Program_Description, Program_Duration)

2) Student_Module:

Primary key: Student_ID, Module_ID

- Module_Name is partially dependent on Module_ID
- Credit_Hours is partially dependent on Module_ID

2NF Scheme:

Module(2NF):

(Module_ID, Module_Name, Credit_Hours)

Student_Module(2NF):

(Module_ID, Student_ID)

3) Module_Teacher

Primary key: Module_ID, Teacher_ID

- Teacher_Name is partially dependent on Teacher_ID
- Contact is partially dependent on Teacher_ID
- Specialization is partially dependent on Teacher_ID

2NF Scheme:

Teacher(2NF):

(Teacher_ID, Teacher_Name, Contact, Specialization)

Module_Teacher(2NF):

(Student_ID, Module_ID, Teacher_ID)

4) Module_Teacher_Announcement

Primary Key: Module_ID, Teacher_ID, Announcement_ID

- Date_Posted is partially dependent on Announcement_ID
- Content is partially dependent on Announcement_ID

2NF Scheme:

Announcement(2NF):

(Announcement_ID, Date_Posted, Content)

Teacher_Announcement(2NF):

(Student_ID, Module_ID, Teacher_ID, Announcement_ID)

5) Student_Module_Resource

Primary key: Student_ID, Module_ID, Resource_ID

- Resource_Title is partially dependent on Resource_ID
- Resource_Duration is partially dependent on Resource_ID
- Resource_Type is partially dependent on Resource_ID
- Sequence is partially dependent on Resource_ID

2NF Scheme:

Resource(2NF):

(Resource_ID, Resource_Title, Resource_Duration, Resource_Type, Sequence)

Student_Module_Resource(2NF) :

(Student_ID, Module_ID, Resource_ID, Resource_Status)

6) Student_Module_Assessment

Primary key: Student_ID, Module_ID, Assessment_ID

- Assessment_Title is partially dependent on Assessment_ID
- Assessment_Info is partially dependent on Assessment_ID
- Assessment_Status is partially dependent on Assessment_ID
- Weightage is partially dependent on Assessment_ID
- Deadline is partially dependent on Assessment_ID
- Result_Marks is partially dependent on Assessment_ID
- Grade is partially dependent on Assessment_ID
- Result_Date is partially dependent on Assessment_ID

2NF Scheme:

Assessment(2NF):

(Assessment_ID, Assessment_Title, Assessment_Info, Weightage, Deadline)

Module_Assessment(2NF):

(Student_ID, Module_ID, Assessment_ID, Assessment_Status, Result_Marks, Grade, Date_Posted)

Final Results after 2NF

- Student(2NF): (Student_Name, Student_ID, Enrolled_Date, Email, DOB, Program_Name, Program_ID, Program_Description, Program_Duration)
- Module(2NF): (Module_ID, Module_Name, Credit_Hours)
- Student_Module(2NF): (Module_ID, Student_ID)
- Teacher(2NF): (Teacher_ID, Teacher_Name, Contact, Specialization)
- Module_Teacher(2NF): (Student_ID, Module_ID, Teacher_ID)
- Announcement(2NF): (Announcement_ID, Date_Posted, Content)
- Teacher_Announcement(2NF): (Student_ID, Module_ID, Teacher_ID, Announcement_ID)
- Resource(2NF): (Resource_ID, Resource_Title, Resource_Duration, Resource_Type, Sequence)
- Student_Module_Resource(2NF): (Student_ID, Module_ID, Resource_ID)
- Assessment(2NF): (Assessment_ID, Assessment_Title, Assessment_Info, Weightage, Deadline)
- Module_Assessment(2NF): (Student_ID, Module_ID, Assessment_ID, Assessment_Status, Result_Marks, Grade, Date_Posted)

Rules to Follow:

- Ensure that no non_key attribute is partially dependent on the primary key.
- All attributes should directly relate to the primary key.

11) Third Normal Form(3NF)

Third Normal Form (3NF) aims to eliminate transitive dependencies, which occur when a non_key attribute depends on another non_key attribute. Let's refine the organization even further! Second Normal Form (2NF) helps eliminate redundancy that might still exist.

Rules to Achieve 3NF:

- 1) Ensure the database is in 2NF
- 2) Eliminate transitive dependencies by ensuring all non_key attributes are directly related to the primary key.

Process to Achieve 3NF

1) Student:

- Program_Name directly depends on Program_ID
- Program_Description directly depends on Program_ID
- Program_Duration directly depends on Program_ID

Thus, the required 3NF will be:

Student(3NF):

(Student_Name, Student_ID, Enrolled_Date, Email, DOB, Program_Name, Program_ID)

Program(3NF):

(Program_ID, Program_Name, Program_Description, Program_Duration)

2) Module

All the non key attributes directly depends on Module_ID resulting in absence of transitive dependency.

Thus, the required 3NF will be:

Module(3NF):

(Module_ID, Module_Name, Credit_Hours)

3) Student_Module

Non key attributes in Student_Module is absent. Thus, the required 3NF will be

Student_Module(3NF):

(Student_ID, Module_ID)

4) Teacher

All the non key attributes directly depend on Teacher_ID resulting in absence of transitive dependency

Thus, the required 3NF will be

Teacher(3NF):

(Teacher_ID, Teacher_Name, Contact, Specialization)

5) Module_Teacher

Non key attributes in Module_Teacher is absent. Thus, the required 3NF will be

Module_Teacher(3NF):

(Student_ID, Module_ID, Teacher_ID)

6) Announcement

All the non key attributes directly depend on Announcement_ID resulting in absence of transitive dependency

Thus, the required 3NF will be

Announcement(3NF):

(Announcement_ID, Date_Posted, Content)

7) Teacher_Announcement

Non key attributes in Teacher_Announcement is absent. Thus, the required 3NF will be

Teacher_Announcement(3NF):

(Student_ID, Module_ID, Teacher_ID, Announcement_ID)

8) Resource

All the non key attributes directly depend on Resource_ID resulting in absence of transitive dependency

Thus, the required 3NF will be:

Resource(3NF):

(Resource_ID, Resource_Title, Resource_Duration, Resource_Type, Sequence)

9) Stud_Module_Resource

All the non key attributes directly depend on Resource_Status resulting in absence of transitive dependency

Thus, the required 3NF will be:

Stud_Module_Resource(3NF):

(Student_ID, Module_ID, Resource_ID)

10) Assessment

All the non key attributes directly depend on Assessment_ID resulting in absence of transitive dependency

Thus, the required 3NF will be

Assessment(3NF):

(Assessment_ID, Assessment_Title, Assessment_Info, Weightage, Deadline)

11) Module_Assessment

Result_Marks, Grade, Date_Posted are directly dependent to Student_ID, Module_ID and Assessment_ID resulting in absence of transitive dependency

Thus, the required 3NF will be

Module_Asst(3NF):

(Student_ID, Module_ID, Assessment_ID, Assessment_Status, Result_Marks, Grade, Date_Posted)

Final Results after 3NF:

- Student(3NF) :(Student_Name, Student_ID, Enrolled_Date, Email, DOB, Program_Name, Program_ID)
- Program(3NF) :(Program_ID, Program_Name, Program_Description, Program_Duration)
- Module(3NF) :(Module_ID, Module_Name, Credit_Hours)
- Student_Module(3NF) :(Student_ID, Module_ID)
- Teacher(3NF) :(Teacher_ID, Teacher_Name, Contact, Specialization)
- Module_Teacher(3NF) :(Student_ID, Module_ID, Teacher_ID)
- Announcement(3NF) :(Announcement_ID, Date_Posted, Content)
- Teacher_Announcement(3NF) :(Student_ID, Module_ID, Teacher_ID, Announcement_ID)
- Resource(3NF) :(Resource_ID, Resource_Title, Resource_Duration, Resource_Type, Sequence)
- Stud_Module_Resource(3NF) :(Student_ID, Module_ID, Resource_ID,)
- Assessment(3NF) :(Assessment_ID, Assessment_Title, Assessment_Info, Weightage, Deadline)
- Module_Asst(3NF) :(Student_ID, Module_ID, Assessment_ID, Assessment_Status, Result_Marks, Grade, Date_Posted)

12) Conclusion

By following these normalization steps—starting from Unnormalized Form through First, Second, and Third Normal Forms—you ensure that your database remains organized, minimizes redundancy, and maintains data integrity. This systematic approach makes your data easier to manage, reduces errors, and enhances overall efficiency. By organizing data through these normalization steps, we can manage our databases like a tidy closet where everything is accessible and efficient.

13) Data Dictionary

Student Table:

S.N.	Attribute	Data type	Size	Constraint
1	Student_ID	Number	18	Primary Key
2	Student_Name	Character	35	Not Null
3	Enrolled_Date	Date	10	Not null
4	Email	Character	35	Unique
5	DOB	Date	10	Not null
6	Program_ID	Number	18	Foreign Key

Table 1 Student table after 3NF

Program Table:

S.N.	Attribute	Data type	Size	Constraint
1	Program_Name	Character	18	Primary Key
2	Program_ID	Number	22	Not Null
3	Program_Description	Character	200	Not null
4	Program_Duration	Number	6	Unique

Table 2 Program table after 3NF

Module Table:

S.N.	Attribute	Data type	Size	Constraint
1	Module_ID	Number	12	Primary Key
2	Module_Name	Character	22	Not Null
3	Credit_Hours	Number	5	Not null

Table 3 Module table after 3NF

Teacher Table:

S.N.	Attribute	Data type	Size	Constraint
1	Teacher_Name	Character	30	Not null
2	Teacher_ID	Number	12	Primary Key
3	Contact	Number	15	Unique
4	Specialization	Character	20	Not null

Table 4 Teacher table after 3NF

Assessment Table:

S.N.	Attribute	Data type	Size	Constraint
1	Assessment_ID	Number	22	Primary_Key
2	Assssment_Title	Character	25	Not null
3	Assessment_Info	Character	1200	Not null
4	Weightage	Number	5	Not null
5	Deadline	Date	10	Not Null
6	Assessment_Status	Character	30	Not Null

Table 5 Assessment table after 3NF

Announcement Table:

S.N.	Attribute	Data type	Size	Constraint
1	Announcement_ID	Number	25	Primary Key
2	Date_Posted	Date	10	Not null
3	Content	Character	1200	Not null

Table 6 Announcement table after 3NF

Resource Table:

S.N.	Attribute	Data type	Size	Constraint
1	Resource_ID	Number	12	PrimaryKey
2	Resource_Title	Character	30	Not null
3	Resource_Duration	Number	5	Not null
4	Resource_Type	Character	25	Not null
5	Sequence	Character	10	Unique

Table 7 Resource table after 3NF

Bridging Entities

Student_Module

S.N.	Attribute	Data type	Size	Constraint	Composite Constraint
1	Student_ID	Number	18	Foreign Key	Primary Key
2	Module_ID	Number	12	Foreign Key	Primary Key

Table 8 Student_Module table after 3NF

Module_Teacher

S.N.	Attribute	Data type	Size	Constraint	Composite Constraint
1	Student_ID	Number	18	Foreign Key	Primary Key
2	Module_ID	Number	12	Foreign Key	Primary Key
3	Teacher_ID	Number	12	Foreign Key	Primary Key

Table 9 Module_Teacher table after 3NF

Teacher_Announcement

S.N.	Attribute	Data type	Size	Constraint	Composite Constraint
1	Student_ID	Number	18	Foreign Key	Primary Key
2	Module_ID	Number	12	Foreign Key	Primary Key
3	Teacher_ID	Number	22	Foreign Key	Primary Key
4	Announcement_ID	Number	25	Foreign Key	Primary Key

Table 10 Teacher_Announcement table after 3NF

Module_Resource

S.N.	Attribute	Data type	Size	Constraint	Composite Constraint
1	Student_ID	Number	18	Foreign Key	Primary Key
2	Module_ID	Number	12	Foreign Key	Primary Key
3	Resource_ID	Number	12	Foreign Key	Primary Key

Table 11 Module_Resource table after 3NF

Module_Ass

S.N.	Attribute	Data type	Size	Constraint	Composite Constraint
1	Module_ID	Number	12	Foreign Key	Primary Key
2	Student_ID	Number	18	Foreign Key	Primary Key
3	Assessment_ID	Number	22	Foreign Key	Primary Key
4	Assessment_Status	Character	30	Not Null	
5	Result_Marks	Number	5	Not null	
6	Grade	Character	5	Not null	
7	Result_Date	Date	10	Not null	

Table 12 Module_Assessment table after 3NF

Final ERD

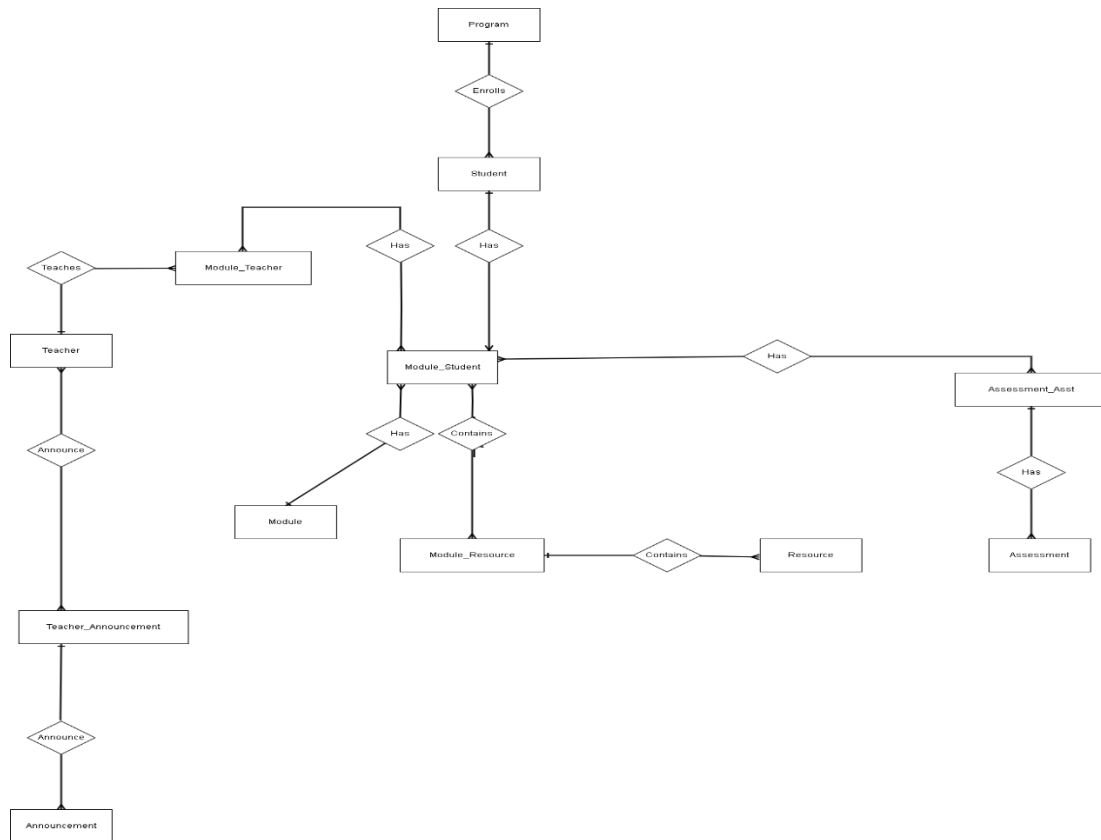


Figure 1 Final ERD

14) Implementation

14.1) Creating new user.

```
SQL> connect system/niran
Connected.
SQL> create user Nnirann identified by 23047617;

User created.

SQL> grant connect, resource to Nnirann;

Grant succeeded.

SQL> connect Nirann/23047617
ERROR:
ORA-01017: invalid username/password; logon denied

Warning: You are no longer connected to ORACLE.
SQL> connect Nnirann/23047617
Connected.
SQL> |
```

Figure 2 Creating new user

14.2) Creating student table and describing table

```
SQL> connect Niraan/23047617
Connected.
SQL> CREATE TABLE Student (
2     Student_ID NUMBER(18) PRIMARY KEY,
3     Student_Name VARCHAR(35) NOT NULL,
4     Enrolled_Date DATE NOT NULL,
5     Email VARCHAR(35) UNIQUE,
6     DOB DATE NOT NULL,
7     Program_ID NUMBER(18) NOT NULL
8 );
```

Table created.

```
SQL> desc student
```

Name	Null?	Type
STUDENT_ID	NOT NULL	NUMBER(18)
STUDENT_NAME	NOT NULL	VARCHAR2(35)
ENROLLED_DATE	NOT NULL	DATE
EMAIL		VARCHAR2(35)
DOB	NOT NULL	DATE
PROGRAM_ID	NOT NULL	NUMBER(18)

```
SQL> |
```

Figure 3 Creating student table.

14.3) Creating program table and describing table

```
2      Program_ID NUMBER(18) PRIMARY KEY,  
3      Program_Name VARCHAR2(22) NOT NULL,  
4      Program_Description VARCHAR2(200) NOT NULL,  
5      Program_Duration NUMBER(6)  
6 );  
SQL> CREATE TABLE Program (  
Table created.
```

```
SQL> desc program  
Name                               Null?    Type  
-----  
PROGRAM_ID                         NOT NULL NUMBER(18)  
PROGRAM_NAME                       NOT NULL VARCHAR2(22)  
PROGRAM_DESCRIPTION                 NOT NULL VARCHAR2(1200)  
PROGRAM_DURATION                    NUMBER(6)  
  
SQL> |
```

Figure 4 Creating program table

14.4) Creating module table and describing table

```
SQL> CREATE TABLE Module (  
2      Module_ID NUMBER(12) PRIMARY KEY,  
3      Module_Name VARCHAR(22) NOT NULL,  
4      Credit_Hours NUMBER(5) NOT NULL  
5 );  
  
Table created.  
  
SQL> desc module  
Name                               Null?    Type  
-----  
MODULE_ID                         NOT NULL NUMBER(12)  
MODULE_NAME                       NOT NULL VARCHAR2(22)  
CREDIT_HOURS                      NOT NULL NUMBER(5)  
  
SQL> |
```

Figure 5 Creating module table

14.5) Creating Student_Module table and describing table

```
SQL> CREATE TABLE Student_Module (  
2     Student_ID NUMBER(18) NOT NULL,  
3     Module_ID NUMBER(12) NOT NULL,  
4     PRIMARY KEY (Student_ID, Module_ID),  
5     FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),  
6     FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID)  
7 );  
  
Table created.  
  
SQL> desc student_module  
Name                               Null?    Type  
-----  
STUDENT_ID                         NOT NULL NUMBER(18)  
MODULE_ID                          NOT NULL NUMBER(12)  
  
SQL> |
```

Figure 6 Creating Student_Module table

14.6) Creating teacher table and describing table

```
SQL> CREATE TABLE Teacher (  
2     Teacher_ID NUMBER(12) PRIMARY KEY,  
3     Teacher_Name VARCHAR(30) NOT NULL,  
4     Contact NUMBER(15) UNIQUE,  
5     Specialization VARCHAR(20) NOT NULL  
6 );  
  
Table created.  
  
SQL> desc teacher  
Name                               Null?    Type  
-----  
TEACHER_ID                         NOT NULL NUMBER(12)  
TEACHER_NAME                       NOT NULL VARCHAR2(30)  
CONTACT                            NUMBER(15)  
SPECIALIZATION                     NOT NULL VARCHAR2(20)  
  
SQL> |
```

Figure 7 Creating teacher table

14.7) Creating Module_Teacher table and describing table

```
SQL> CREATE TABLE Module_Teacher (  
 2     Student_ID NUMBER(18) NOT NULL,  
 3     Module_ID NUMBER(12) NOT NULL,  
 4     Teacher_ID NUMBER(12) NOT NULL,  
 5     PRIMARY KEY (Student_ID, Module_ID, Teacher_ID),  
 6     FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),  
 7     FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID),  
 8     FOREIGN KEY (Teacher_ID) REFERENCES Teacher(Teacher_ID)  
 9 );
```

Table created.

```
SQL> desc module_teacher
```

Name	Null?	Type
STUDENT_ID	NOT NULL	NUMBER(18)
MODULE_ID	NOT NULL	NUMBER(12)
TEACHER_ID	NOT NULL	NUMBER(12)

```
SQL> |
```

Figure 8 Creating Module_Teacher table

14.8) Creating announcement table and describing table

```
SQL> CREATE TABLE Announcement (  
 2     Announcement_ID NUMBER(25) PRIMARY KEY,  
 3     Date_Posted DATE NOT NULL,  
 4     Content VARCHAR(1200) NOT NULL  
 5 );
```

Table created.

```
SQL> desc announcement
```

Name	Null?	Type
ANNOUNCEMENT_ID	NOT NULL	NUMBER(25)
DATE_POSTED	NOT NULL	DATE
CONTENT	NOT NULL	VARCHAR2(1200)

```
SQL> |
```

Figure 9 Creating announcement

14.9) Creating Teacher_Announcement table and describing table

```
SQL> CREATE TABLE Teacher_Announcement (  
 2     Student_ID NUMBER(18) NOT NULL,  
 3     Module_ID NUMBER(12) NOT NULL,  
 4     Teacher_ID NUMBER(12) NOT NULL,  
 5     Announcement_ID NUMBER(25) NOT NULL,  
 6     PRIMARY KEY (Student_ID, Module_ID, Teacher_ID, Announcement_ID),  
 7     FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),  
 8     FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID),  
 9     FOREIGN KEY (Teacher_ID) REFERENCES Teacher(Teacher_ID),  
10     FOREIGN KEY (Announcement_ID) REFERENCES Announcement(Announcement_ID)  
11 );
```

Table created.

```
SQL> desc teacher_announcement
```

Name	Null?	Type
STUDENT_ID	NOT NULL	NUMBER(18)
MODULE_ID	NOT NULL	NUMBER(12)
TEACHER_ID	NOT NULL	NUMBER(12)
ANNOUNCEMENT_ID	NOT NULL	NUMBER(25)

```
SQL> |
```

Figure 10 Creating Teacher_Announcement table

14.10) Creating Resources table and describing table

```
SQL> CREATE TABLE Resources (  
 2     Resource_ID NUMBER(12) PRIMARY KEY,  
 3     Resource_Title VARCHAR(30) NOT NULL,  
 4     Resource_Duration NUMBER(5) NOT NULL,  
 5     Resource_Type VARCHAR(25) NOT NULL,  
 6     Sequence VARCHAR(10) UNIQUE  
 7 );
```

Table created.

```
SQL> desc resources
```

Name	Null?	Type
RESOURCE_ID	NOT NULL	NUMBER(12)
RESOURCE_TITLE	NOT NULL	VARCHAR2(30)
RESOURCE_DURATION	NOT NULL	NUMBER(5)
RESOURCE_TYPE	NOT NULL	VARCHAR2(25)
SEQUENCE		VARCHAR2(10)

```
SQL> |
```

Figure 11 Creating Resources table

14.11) Creating Stud_Module_Resource and describing table

```
SQL> CREATE TABLE Stud_Module_Resource (  
 2     Student_ID NUMBER(18) NOT NULL,  
 3     Module_ID NUMBER(12) NOT NULL,  
 4     Resource_ID NUMBER(12) NOT NULL,  
 5     PRIMARY KEY (Student_ID, Module_ID, Resource_ID),  
 6     FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),  
 7     FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID),  
 8     FOREIGN KEY (Resource_ID) REFERENCES Resources(Resource_ID)  
 9 );
```

Table created.

```
SQL> desc stud_module_resource
```

Name	Null?	Type
STUDENT_ID	NOT NULL	NUMBER(18)
MODULE_ID	NOT NULL	NUMBER(12)
RESOURCE_ID	NOT NULL	NUMBER(12)

```
SQL> |
```

Figure 12 Creating Stud_Module_Resource

14.12) Creating assessment table and describing table

```
SQL> CREATE TABLE Assessment (  
 2     Assessment_ID NUMBER(22) PRIMARY KEY,  
 3     Assessment_Title VARCHAR(25) NOT NULL,  
 4     Assessment_Info VARCHAR(1200) NOT NULL,  
 5     Weightage NUMBER(5) NOT NULL,  
 6     Deadline DATE NOT NULL,  
 7     Assessment_Status VARCHAR(30) NOT NULL  
 8 );
```

Table created.

```
SQL> desc assessment
```

Name	Null?	Type
ASSESSMENT_ID	NOT NULL	NUMBER(22)
ASSESSMENT_TITLE	NOT NULL	VARCHAR2(25)
ASSESSMENT_INFO	NOT NULL	VARCHAR2(1200)
WEIGHTAGE	NOT NULL	NUMBER(5)
DEADLINE	NOT NULL	DATE
ASSESSMENT_STATUS	NOT NULL	VARCHAR2(30)

```
SQL> |
```

Figure 13 Creating assessment table

14.13) Creating Module_Asst table and describing table

```
SQL> CREATE TABLE Module_Asst (  
2     Module_ID NUMBER(12) NOT NULL,  
3     Student_ID NUMBER(18) NOT NULL,  
4     Assessment_ID NUMBER(22) NOT NULL,  
5     Assessment_Status VARCHAR(30) NOT NULL,  
6     Result_Marks NUMBER(5),  
7     Grade VARCHAR(5),  
8     Result_Date DATE,  
9     PRIMARY KEY (Module_ID, Student_ID, Assessment_ID),  
10    FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID),  
11    FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),  
12    FOREIGN KEY (Assessment_ID) REFERENCES Assessment(Assessment_ID)  
13 );
```

Table created.

```
SQL> desc Module_Asst
```

Name	Null?	Type
MODULE_ID	NOT NULL	NUMBER(12)
STUDENT_ID	NOT NULL	NUMBER(18)
ASSESSMENT_ID	NOT NULL	NUMBER(22)
ASSESSMENT_STATUS	NOT NULL	VARCHAR2(30)
RESULT_MARKS		NUMBER(5)
GRADE		VARCHAR2(5)
RESULT_DATE		DATE

```
SQL> |
```

Figure 14 Creating Module_Asst table

Inserting the values in table we created

14.14) Inserting the values in program table:

```
SQL> INSERT INTO Program VALUES (101, 'Computer Science', 'Focuses on computing principles and software development.', 4);
```

1 row created.

```
SQL> INSERT INTO Program VALUES (102, 'Multimedia', 'Covers digital media creation and editing.', 3);
```

1 row created.

```
SQL> INSERT INTO Program VALUES (103, 'Business Management', 'Covers business strategies and operations.', 3);
```

1 row created.

```
SQL> INSERT INTO Program VALUES (104, 'Information Technology', 'Emphasis on IT systems and networking.', 4);
```

1 row created.

```
SQL> INSERT INTO Program VALUES (105, 'Digital Marketing', 'Focus on online marketing strategies.', 2);
```

1 row created.

```
SQL> INSERT INTO Program VALUES (106, 'Cyber Security', 'Specialized in network and data security.', 3);
```

1 row created.

```
SQL> INSERT INTO Program VALUES (107, 'Data Analytics', 'Exploration of data processing techniques.', 2);
```

1 row created.

Figure 15 Inserting values in program table

```
SQL> select * from program;
```

	PROGRAM_ID	PROGRAM_NAME	PROGRAM_DURATION	PROGRAM_DESCRIPTION
4	101	Computer Science		Focuses on computing principles and software development.
3	102	Multimedia		Covers digital media creation and editing.
3	103	Business Management		Covers business strategies and operations.
4	104	Information Technology		Emphasis on IT systems and networking.
2	105	Digital Marketing		Focus on online marketing strategies.
3	106	Cyber Security		Specialized in network and data security.
2	107	Data Analytics		Exploration of data processing techniques.

7 rows selected.

```
SQL> |
```

14.15) Inserting values in module table

```
SQL>
SQL> INSERT INTO Module VALUES (201, 'Databases', 3);
1 row created.
SQL> INSERT INTO Module VALUES (202, 'Data Structure', 4);
1 row created.
SQL> INSERT INTO Module VALUES (203, 'Algorithms', 3);
1 row created.
SQL> INSERT INTO Module VALUES (204, 'Cyber Security', 4);
1 row created.
SQL> INSERT INTO Module VALUES (205, 'Digital Market', 2);
1 row created.
SQL> INSERT INTO Module VALUES (206, 'Advanced Java', 4);
1 row created.
SQL> INSERT INTO Module VALUES (207, 'Big Data ', 3);
1 row created.
SQL>
SQL> select * from module;
```

MODULE_ID	MODULE_NAME	CREDIT_HOURS
201	Databases	3
202	Data Structure	4
203	Algorithms	3
204	Cyber Security	4
205	Digital Market	2
206	Advanced Java	4
207	Big Data	3

7 rows selected.

```
SQL> |
```

Figure 16 Inserting values in module table

14.16) Inserting values in student table:

```
SQL> INSERT INTO Student VALUES (1, 'Niran Bhatta', TO_DATE('2024-01-15', 'YYYY-MM-DD'), 'niran.bhatta@example.com',
, TO_DATE('2000-05-20', 'YYYY-MM-DD'), 101);

1 row created.

SQL> INSERT INTO Student VALUES (2, 'Manish Thapa', TO_DATE('2023-09-05', 'YYYY-MM-DD'), 'manish.thapa@example.com',
, TO_DATE('1999-03-18', 'YYYY-MM-DD'), 102);

1 row created.

SQL> INSERT INTO Student VALUES (3, 'Sita Sharma', TO_DATE('2023-11-22', 'YYYY-MM-DD'), 'sita.sharma@example.com',
TO_DATE('2001-07-30', 'YYYY-MM-DD'), 103);

1 row created.

SQL> INSERT INTO Student VALUES (4, 'Ravi Karki', TO_DATE('2024-01-01', 'YYYY-MM-DD'), 'ravi.karki@example.com', TO
_DATE('2002-02-28', 'YYYY-MM-DD'), 101);

1 row created.

SQL> INSERT INTO Student VALUES (5, 'Anju Tamang', TO_DATE('2023-10-12', 'YYYY-MM-DD'), 'anju.tamang@example.com',
TO_DATE('2000-06-15', 'YYYY-MM-DD'), 102);

1 row created.

SQL> INSERT INTO Student VALUES (6, 'Gita Pradhan', TO_DATE('2024-03-01', 'YYYY-MM-DD'), 'gita.pradhan@example.com'
, TO_DATE('1998-12-09', 'YYYY-MM-DD'), 103);

1 row created.

SQL> INSERT INTO Student VALUES (7, 'Kamal Shrestha', TO_DATE('2023-12-10', 'YYYY-MM-DD'), 'kamal.shrestha@example.
com', TO_DATE('2001-01-20', 'YYYY-MM-DD'), 101);

1 row created.

SQL> select * from student;
```

STUDENT_ID	STUDENT_NAME	ENROLLED_	EMAIL	DOB	PROGRAM_ID
1	Niran Bhatta	15-JAN-24	niran.bhatta@example.com	20-MAY-00	101
2	Manish Thapa	05-SEP-23	manish.thapa@example.com	18-MAR-99	102
3	Sita Sharma	22-NOV-23	sita.sharma@example.com	30-JUL-01	103
4	Ravi Karki	01-JAN-24	ravi.karki@example.com	28-FEB-02	101
5	Anju Tamang	12-OCT-23	anju.tamang@example.com	15-JUN-00	102
6	Gita Pradhan	01-MAR-24	gita.pradhan@example.com	09-DEC-98	103
7	Kamal Shrestha	10-DEC-23	kamal.shrestha@example.com	20-JAN-01	101

```
7 rows selected.
SQL>
```

Figure 17 17 Inserting values in student table:

14.17) Inserting values in teacher table:

```
7 rows selected.
INSERT INTO Teacher VALUES (301, 'Rajesh Karki', 9841234567, 'Database Systems');
SQL>
1 row created.

SQL> INSERT INTO Teacher VALUES (302, 'Anita Basnet', 9847654321, 'Programming');
1 row created.

SQL> INSERT INTO Teacher VALUES (303, 'Keshav Adhikari', 9811122233, 'Cyber Security');
1 row created.

SQL> INSERT INTO Teacher VALUES (304, 'Binita Sharma', 9805678912, 'Digital Marketing');
1 row created.

SQL> INSERT INTO Teacher VALUES (305, 'Madan Rai', 9843334455, 'Networking');
1 row created.

SQL> INSERT INTO Teacher VALUES (306, 'Sarita Tamang', 9810987654, 'Data Analysis');
1 row created.

SQL> INSERT INTO Teacher VALUES (307, 'Suresh Thapa', 9802345678, 'Algorithms');
1 row created.

SQL>
SQL> select * from teacher;
```

TEACHER_ID	TEACHER_NAME	CONTACT	SPECIALIZATION
301	Rajesh Karki	9841234567	Database Systems
302	Anita Basnet	9847654321	Programming
303	Keshav Adhikari	9811122233	Cyber Security
304	Binita Sharma	9805678912	Digital Marketing
305	Madan Rai	9843334455	Networking
306	Sarita Tamang	9810987654	Data Analysis
307	Suresh Thapa	9802345678	Algorithms

```
7 rows selected.

SQL> |
```

Figure 18 Inserting values in teacher table

14.18) Inserting values in Student_Module table:

```
SQL> INSERT INTO Student_Module VALUES (1, 201);
1 row created.

SQL> INSERT INTO Student_Module VALUES (2, 202);
1 row created.

SQL> INSERT INTO Student_Module VALUES (3, 203);
1 row created.

SQL> INSERT INTO Student_Module VALUES (4, 204);
1 row created.

SQL> INSERT INTO Student_Module VALUES (5, 205);
1 row created.

SQL> INSERT INTO Student_Module VALUES (6, 206);
1 row created.

SQL> INSERT INTO Student_Module VALUES (7, 207);
1 row created.

SQL> select * from dual;

D
-
X

SQL> select * from student_module
2 ;

STUDENT_ID  MODULE_ID
-----
1           201
2           202
3           203
4           204
5           205
6           206
7           207

7 rows selected.

SQL> |
```

Figure 19 Inserting values in student_module

14.19) Inserting values in Module_Teacher

```
SQL> INSERT INTO Module_Teacher VALUES (1, 201, 301);
1 row created.

SQL> INSERT INTO Module_Teacher VALUES (2, 202, 302);
1 row created.

SQL> INSERT INTO Module_Teacher VALUES (3, 203, 307);
1 row created.

SQL> INSERT INTO Module_Teacher VALUES (4, 204, 303);
1 row created.

SQL> INSERT INTO Module_Teacher VALUES (5, 205, 304);
1 row created.

SQL> INSERT INTO Module_Teacher VALUES (6, 206, 305);
1 row created.

SQL> INSERT INTO Module_Teacher VALUES (7, 207, 306);
1 row created.

SQL> select * from dual;

D
-
X

SQL> select * from module_teacher;

STUDENT_ID  MODULE_ID  TEACHER_ID
-----
          1         201         301
          2         202         302
          3         203         307
          4         204         303
          5         205         304
          6         206         305
          7         207         306

7 rows selected.

SQL> |
```

Figure 20 Inserting values in Module_Teacher

14.20) Inserting values in Announcement table.

```
SQL> INSERT INTO Announcement VALUES (501, TO_DATE('2024-05-01', 'YYYY-MM-DD'), 'Upcoming database seminar on 15th May.');
```

1 row created.

```
SQL> INSERT INTO Announcement VALUES (502, TO_DATE('2024-05-10', 'YYYY-MM-DD'), 'Submission deadline reminder for database quiz.');
```

1 row created.

```
SQL> INSERT INTO Announcement VALUES (503, TO_DATE('2024-05-20', 'YYYY-MM-DD'), 'Guest lecture on advanced algorithms.');
```

1 row created.

```
SQL> INSERT INTO Announcement VALUES (504, TO_DATE('2024-05-12', 'YYYY-MM-DD'), 'Networking lab schedule updates.');
```

1 row created.

```
SQL> INSERT INTO Announcement VALUES (505, TO_DATE('2024-05-18', 'YYYY-MM-DD'), 'Reminder for marketing campaign submission.');
```

1 row created.

```
SQL> INSERT INTO Announcement VALUES (506, TO_DATE('2024-05-05', 'YYYY-MM-DD'), 'Cyber security workshop announcement.');
```

1 row created.

```
SQL> INSERT INTO Announcement VALUES (507, TO_DATE('2024-05-25', 'YYYY-MM-DD'), 'Final notes for Big Data module uploaded.');
```

1 row created.

```
SQL> select * from dual;
```

D
-
X

```
SQL> select * from announcement;
```

ANNOUNCEMENT_ID	DATE_POST	CONTENT
501	01-MAY-24	Upcoming database seminar on 15th May.
502	10-MAY-24	Submission deadline reminder for database quiz.
503	20-MAY-24	Guest lecture on advanced algorithms.
504	12-MAY-24	Networking lab schedule updates.
505	18-MAY-24	Reminder for marketing campaign submission.
506	05-MAY-24	Cyber security workshop announcement.
507	25-MAY-24	Final notes for Big Data module uploaded.

7 rows selected.

Figure 21 Inserting values in Announcement table.

14.21) Inserting values in Techer_Announcement

```
SQL> INSERT INTO Teacher_Announcement VALUES (1, 201, 301, 501);
1 row created.

SQL> INSERT INTO Teacher_Announcement VALUES (2, 202, 302, 502);
1 row created.

SQL> INSERT INTO Teacher_Announcement VALUES (3, 203, 307, 503);
1 row created.

SQL> INSERT INTO Teacher_Announcement VALUES (4, 204, 303, 506);
1 row created.

SQL> INSERT INTO Teacher_Announcement VALUES (5, 205, 304, 505);
1 row created.

SQL> INSERT INTO Teacher_Announcement VALUES (6, 206, 305, 504);
1 row created.

SQL> INSERT INTO Teacher_Announcement VALUES (7, 207, 306, 507);
1 row created.

SQL> select * from dual;
D
-
X

SQL> select * from Teacher_Announcement;

STUDENT_ID  MODULE_ID  TEACHER_ID  ANNOUNCEMENT_ID
-----
1           201        301         501
2           202        302         502
3           203        307         503
4           204        303         506
5           205        304         505
6           206        305         504
7           207        306         507

7 rows selected.

SQL> |
```

Figure 22 Inserting values in Techer_Announcement

14.22) Inserting values in Resource table

```
SQL> INSERT INTO Resources VALUES (601, 'Database Lecture Video', 60, 'Video', '1');
1 row created.

SQL> INSERT INTO Resources VALUES (602, 'SQL Cheat Sheet', 5, 'PDF', '2');
1 row created.

SQL> INSERT INTO Resources VALUES (603, 'Algorithm Visualizations', 15, 'Interactive', '3');
1 row created.

SQL> INSERT INTO Resources VALUES (604, 'Cyber Security Tools', 45, 'Video', '4');
1 row created.

SQL> INSERT INTO Resources VALUES (605, 'Networking Guide', 20, 'PDF', '5');
1 row created.

SQL> INSERT INTO Resources VALUES (606, 'Digital Marketing Trends', 30, 'Article', '6');
1 row created.

SQL> INSERT INTO Resources VALUES (607, 'Big Data Case Studies', 50, 'Ebook', '7');
1 row created.

SQL> select * from dual;
D
-
X

SQL> select * from resources;

RESOURCE_ID RESOURCE_TITLE                RESOURCE_DURATION RESOURCE_TYPE      SEQUENCE
-----
601 Database Lecture Video                60 Video            1
602 SQL Cheat Sheet                      5 PDF               2
603 Algorithm Visualizations             15 Interactive       3
604 Cyber Security Tools                 45 Video            4
605 Networking Guide                    20 PDF               5
606 Digital Marketing Trends             30 Article           6
607 Big Data Case Studies                50 Ebook            7

7 rows selected.

SQL> |
```

Figure 23 Inserting values in Resource table

14.23) Inserting values in assessment

```
SQL> INSERT INTO Assessment VALUES (401, 'Database Quiz', 'A quiz on SQL and database concepts.', 10, TO_DATE('2024-05-15', 'YYYY-MM-DD'), 'Pending');

1 row created.

SQL> INSERT INTO Assessment VALUES (402, 'Final Project', 'A comprehensive database project.', 50, TO_DATE('2024-06-30', 'YYYY-MM-DD'), 'Pending');

1 row created.

SQL> INSERT INTO Assessment VALUES (403, 'Algorithm Design', 'Assignment on algorithm efficiency.', 15, TO_DATE('2024-05-25', 'YYYY-MM-DD'), 'Pending');

1 row created.

SQL> INSERT INTO Assessment VALUES (404, 'Networking Lab', 'Hands-on networking tasks.', 20, TO_DATE('2024-06-10', 'YYYY-MM-DD'), 'Pending');

1 row created.

SQL> INSERT INTO Assessment VALUES (405, 'Digital Marketing Campaign', 'Develop a marketing strategy.', 25, TO_DATE('2024-06-20', 'YYYY-MM-DD'), 'Pending');
INSERT INTO Assessment VALUES (405, 'Digital Marketing Campaign', 'Develop a marketing strategy.', 25, TO_DATE('2024-06-20', 'YYYY-MM-DD'), 'Pending') *
ERROR at line 1:
ORA-12899: value too large for column "NNIRANN"."ASSESSMENT"."ASSESSMENT_TITLE" (actual: 26, maximum: 25)

SQL> INSERT INTO Assessment VALUES (406, 'Cyber Security Case Study', 'Analyze a security breach.', 30, TO_DATE('2024-06-15', 'YYYY-MM-DD'), 'Pending');

1 row created.

SQL> INSERT INTO Assessment VALUES (407, 'Big Data Analytics', 'Practical tasks on big data.', 20, TO_DATE('2024-06-25', 'YYYY-MM-DD'), 'Pending');
```

ASSESSMENT_ID	ASSESSMENT_TITLE

ASSESSMENT_INFO	

WEIGHTAGE	DEADLINE ASSESSMENT_STATUS

401 Database Quiz	
A quiz on SQL and database concepts.	
10	15-MAY-24 Pending
402 Final Project	
A comprehensive database project.	
50	30-JUN-24 Pending
403 Algorithm Design	
Assignment on algorithm efficiency.	
15	25-MAY-24 Pending
404 Networking Lab	
Hands-on networking tasks.	
20	10-JUN-24 Pending
406 Cyber Security Case Study	
Analyze a security breach.	
30	15-JUN-24 Pending
407 Big Data Analytics	
Practical tasks on big data.	
20	25-JUN-24 Pending
405 Digital Marketing Plan	
Develop a marketing strategy.	
25	20-JUN-24 Pending
7 rows selected.	
SQL>	

Figure 24 Inserting values in assessment

14.24) Inserting values in Module_Asst table

```
SQL> INSERT ALL
  2 INTO module_asst (STUDENT_ID, MODULE_ID, ASSESSMENT_ID, ASSESSMENT_STATUS, RESULT_MARKS, GRADE, RESULT_DATE)
  3 VALUES (1, 201, 401, 'Submitted', 85, 'A', TO_DATE('2024-05-20', 'YYYY-MM-DD'))
  4 INTO module_asst (STUDENT_ID, MODULE_ID, ASSESSMENT_ID, ASSESSMENT_STATUS, RESULT_MARKS, GRADE, RESULT_DATE)
  5 VALUES (2, 202, 402, 'Submitted', 90, 'A+', TO_DATE('2024-06-30', 'YYYY-MM-DD'))
  6 INTO module_asst (STUDENT_ID, MODULE_ID, ASSESSMENT_ID, ASSESSMENT_STATUS, RESULT_MARKS, GRADE, RESULT_DATE)
  7 VALUES (3, 203, 403, 'Pending', NULL, NULL, NULL)
  8 INTO module_asst (STUDENT_ID, MODULE_ID, ASSESSMENT_ID, ASSESSMENT_STATUS, RESULT_MARKS, GRADE, RESULT_DATE)
  9 VALUES (4, 204, 404, 'Pending', NULL, NULL, NULL)
 10 INTO module_asst (STUDENT_ID, MODULE_ID, ASSESSMENT_ID, ASSESSMENT_STATUS, RESULT_MARKS, GRADE, RESULT_DATE)
 11 VALUES (5, 205, 405, 'Submitted', 75, 'B', TO_DATE('2024-06-20', 'YYYY-MM-DD'))
 12 INTO module_asst (STUDENT_ID, MODULE_ID, ASSESSMENT_ID, ASSESSMENT_STATUS, RESULT_MARKS, GRADE, RESULT_DATE)
 13 VALUES (6, 206, 406, 'Pending', NULL, NULL, NULL)
 14 INTO module_asst (STUDENT_ID, MODULE_ID, ASSESSMENT_ID, ASSESSMENT_STATUS, RESULT_MARKS, GRADE, RESULT_DATE)
 15 VALUES (7, 207, 407, 'Submitted', 80, 'A', TO_DATE('2024-06-25', 'YYYY-MM-DD'))
 16 SELECT * FROM dual;

7 rows created.

SQL> select * from module_asst;

MODULE_ID STUDENT_ID ASSESSMENT_ID ASSESSMENT_STATUS RESULT_MARKS GRADE RESULT_DA
201        1          401 Submitted          85 A      20-MAY-24
202        2          402 Submitted          90 A+    30-JUN-24
203        3          403 Pending
204        4          404 Pending
205        5          405 Submitted          75 B      20-JUN-24
206        6          406 Pending
207        7          407 Submitted          80 A      25-JUN-24

7 rows selected.

SQL> |
```

Figure 25 Inserting values in Module_Asst table

15) Querying of Database

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

Query- SELECT

```
P.PROGRAM_ID AS Program_ID,
P.PROGRAM_NAME AS Program_Name,
COUNT(S.STUDENT_ID) AS Total_Students
FROM
PROGRAM P
LEFT JOIN
STUDENT S
ON
P.PROGRAM_ID = S.PROGRAM_ID
GROUP BY
P.PROGRAM_ID, P.PROGRAM_NAME
ORDER BY
P.PROGRAM_ID;
```

```

SQL> SELECT
  2     P.PROGRAM_ID AS Program_ID,
  3     P.PROGRAM_NAME AS Program_Name,
  4     COUNT(S.STUDENT_ID) AS Total_Students
  5 FROM
  6     PROGRAM P
  7 LEFT JOIN
  8     STUDENT S
  9 ON
10     P.PROGRAM_ID = S.PROGRAM_ID
11 GROUP BY
12     P.PROGRAM_ID, P.PROGRAM_NAME
13 ORDER BY
14     P.PROGRAM_ID;

```

PROGRAM_ID	PROGRAM_NAME	TOTAL_STUDENTS
101	Computer Science	3
102	Multimedia	2
103	Business Management	2
104	Information Technology	0
105	Digital Marketing	0
106	Cyber Security	0
107	Data Analytics	0

7 rows selected.

```

SQL> |

```

This query makes sure that all the programs are listed in a database. Even if the students are not enrolled, all the programs that are listed in the database are displayed.

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

17) Critical Evaluation

17.1) Evaluation of the Module, Its Purpose, and Connections to Other Subjects:

The modules in the system play a key role in structuring academic programs. It is designed to cover a wide range of modules like programming, data analysis, cybersecurity, and digital marketing. Each module has its own attributes such as its name, ID, and credit hours.

Modules are the glue that holds everything together. They connect teachers to their areas of expertise, students to their courses, and resources like lectures, quizzes, and assignments. This makes it easy to manage what each person needs to do within the system.

What makes modules even more important is how they relate to other parts of the curriculum. For example, they're linked to specific programs, which create a logical flow for students to follow step by step. Modules also integrate multiple disciplines, like combining digital marketing with data analysis to evaluate marketing strategies—showing how interconnected knowledge can be.

17.2) Assessment of the Coursework:

This coursework helps to tackle problems such as managing students, teachers, and assessments in a structured way.

- The database design follows proper normalization steps i.e the data is stored in a efficient way with no unnecessary repetition.
- The relationship between entities like students, modules, and teachers are defined, which makes it easy to add more entities in future.
- It mirrors real-life challenges of managing academic platforms, preparing you for practical applications.

Areas for Improvement:

- Add tools for analyzing data, such as tracking student progress or module success rates, to make the system more insightful.
- Include features that allow students to progress at their own pace, like marking modules by difficulty level or prerequisites.

In summary, the module system is well-structured and has a lot of potential for practical use. It connects the dots between different elements of education, creating a strong foundation for an interactive and scalable learning platform.

Bibliography

- Chris, K. (2022, 12 21). *Database*. Retrieved from FreeCodeCamp: <https://www.freecodecamp.org/news/database-normalization-1nf-2nf-3nf-table-examples/>
- GFG. (2024, 6 3). Retrieved from geeksforgeeks: <https://www.geeksforgeeks.org/how-to-draw-entity-relationship-diagrams/>
- Schools. (n.d.). *SqlTutorial*. Retrieved from W3Schools: https://www.w3schools.com/sql/sql_insert.asp
- team, O. (2023, 5 3). *Ovaledge*. Retrieved from Ovaledge: <https://www.ovaledge.com/blog/what-is-data-dictionary>

(23047617) Niran Bhatta database.docx

 Islington College, Nepal

Document Details

Submission ID

trn:oid:::3618:79924192

Submission Date

Jan 23, 2025, 11:23 AM GMT+5:45

Download Date

Jan 23, 2025, 11:26 AM GMT+5:45

File Name

(23047617) Niran Bhatta database.docx

File Size

29.4 KB

71 Pages





4,021 Words

25,373 Characters




34% Overall Similarity

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

Match Groups

-  **117** Not Cited or Quoted 33%
Matches with neither in-text citation nor quotation marks
-  **5** Missing Quotations 1%
Matches that are still very similar to source material
-  **1** 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

- 4%  Internet sources
- 1%  Publications
- 34%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

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

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

Match Groups

- 117** Not Cited or Quoted 33%
Matches with neither in-text citation nor quotation marks
- 5** Missing Quotations 1%
Matches that are still very similar to source material
- 1** 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

- 4% Internet sources
- 1% Publications
- 34% Submitted works (Student Papers)

Top Sources

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

1	Submitted works	
	islingtoncollege on 2025-01-03	7%
2	Submitted works	
	islingtoncollege on 2025-01-03	4%
3	Submitted works	
	islingtoncollege on 2025-01-02	3%
4	Submitted works	
	islingtoncollege on 2025-01-03	3%
5	Submitted works	
	islingtoncollege on 2025-01-02	3%
6	Submitted works	
	islingtoncollege on 2024-12-30	2%
7	Submitted works	
	islingtoncollege on 2025-01-03	1%
8	Submitted works	
	Softwarica College Of IT & E-Commerce on 2022-02-23	1%
9	Submitted works	
	Australian Catholic University on 2024-08-30	<1%
10	Submitted works	
	University of Wales Institute, Cardiff on 2023-09-18	<1%

11	Submitted works	University of Wollongong on 2023-03-26	<1%
12	Submitted works	Kingston University on 2024-01-09	<1%
13	Submitted works	Federation University on 2023-01-10	<1%
14	Submitted works	Middle East College of Information Technology on 2024-05-22	<1%
15	Submitted works	University College Birmingham on 2025-01-13	<1%
16	Submitted works	University of Warwick on 2024-06-24	<1%
17	Submitted works	islingtoncollege on 2024-12-30	<1%
18	Submitted works	islingtoncollege on 2024-12-30	<1%
19	Submitted works	Lincoln Institute of Higher Education Ltd on 2024-10-06	<1%
20	Submitted works	Middle East College on 2024-12-18	<1%
21	Submitted works	University of East London on 2024-09-06	<1%
22	Submitted works	York St John University on 2025-01-21	<1%
23	Submitted works	Aberystwyth University on 2017-11-09	<1%
24	Submitted works	Monash University on 2024-10-28	<1%

25	Submitted works	University of Bedfordshire on 2023-01-19	<1%
26	Submitted works	University of Lancaster on 2024-07-19	<1%
27	Submitted works	University of Sheffield on 2010-05-14	<1%
28	Publication	Sanjiv Purba. "High-Performance Web Databases - Design, Development, and Dep...	<1%
29	Submitted works	University of Bedfordshire on 2023-09-06	<1%
30	Submitted works	islingtoncollege on 2024-12-22	<1%
31	Submitted works	Icon College of Technology and Management on 2019-06-13	<1%
32	Submitted works	Informatics Education Limited on 2010-10-27	<1%
33	Submitted works	Middle East College of Information Technology on 2016-05-04	<1%
34	Submitted works	Monash University on 2024-10-27	<1%
35	Submitted works	United Colleges Group on 2025-01-21	<1%
36	Submitted works	Informatics Education Limited on 2013-11-18	<1%
37	Submitted works	Softwarica College of IT & E-Commerce on 2015-04-21	<1%
38	Submitted works	The University of Wolverhampton on 2022-08-15	<1%

39

Submitted works

University of Salford on 2024-04-16

<1%

40

Submitted works

University of Suffolk on 2023-03-17

<1%

CC5051NI Databases

100% Individual Coursework

Autumn 2024

Credit: 15 Semester Long Module

Student Name: Niran Bhatta

London Met ID: 23047617

Assignment Submission Date: December 31, 2024

Word Count: 3676

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.

Contents

1)	Introduction	3
2)	Current business activities and operations	3
3)	Business Rule	3
4)	Assumptions	4
5)	Entities and attributes identification	4
6)	Entity Relationship Diagram (ERD)	6
7)	Normalization	6
8)	Unnormalized Form (UNF)	7
9)	First Normal Form (1NF)	7
10)	Second Normal Form (2NF)	9
11)	Third Normal Form(3NF)	11
12)	Conclusion	14
13)	Data Dictionary	15
	Final ERD	18
14)	Implementation	19
15)	Querying of Database	35
17)	Critical Evaluation	37
18)	Bibliography	37

Table of Tables

Table 1 Student table after 3NF 16

Table 2 Program table after 3NF 16

Table 3 Module table after 3NF 16

Table 4 Teacher table after 3NF 17

Table 5 Assessment table after 3NF 17

Table 6 Announcement table after 3NF 17

Table 7 Resource table after 3NF 17

Table 8 Student_Module table after 3NF 18

Table 9 Module_Teacher table after 3NF 18

Table 10 Teacher_Announcement table after 3NF 18

Table 11 Module_Resource table after 3NF 18

Table 12 Module_Assessment table after 3NF 19

Table of Figures

Figure 1 Final ERD 19

Figure 2 Creating new user 20

Figure 3 Creating student table.	21
Figure 4 Creating program table	21
Figure 5 Creating module table	21
Figure 6 Creating Student_Module table	22
Figure 7 Creating teacher table	22
Figure 8 Creating Module_Teacher table	23
Figure 9 Creating announcement	23
Figure 10 Creating Teacher_Announcement table	24
Figure 11 Creating Resources table	24
Figure 12 Creating Stud_Module_Resource	25
Figure 13 Creating assessment table	25
Figure 14 Creating Module_Asst table	26
Figure 15 Inserting values in program table	26
Figure 16 Inserting values in module table	27
Figure 17 Inserting values in student table:	28
Figure 18 Inserting values in teacher table	29
Figure 19 Inserting values in student_module	30
Figure 20 Inserting values in Module_Teacher	31
Figure 21 Inserting values in Announcement table.	32
Figure 22 Inserting values in Teacher_Announcement	33
Figure 23 Inserting values in Resource table	34
Figure 24 Inserting values in assessment	35
Figure 25 Inserting values in Module_Asst table	36

Introduction

E Classroom Platform is a tool for online education, using technological aids to promote a regular, organized learning environment. The strong side of it is that this will contribute to a continuous educational journey by making use of modern technologies, whereby the content learned will be readily available to learners and easy to monitor and follow. With this comes more flexibility in learning to facilitate diversities in learning and educational needs requirements crucial in a contemporary academic landscape.

Miss Mary wishes to develop an eLearning platform known as the E Classroom System, which can connect student and teacher in a more organized and interactive manner. It would provide several academic programs, such as Computing and Multimedia, with several required modules that the student needs to be studied step by step, so that it becomes easy to follow their studies step by step. Teachers will be able to assign assessments and give feedback on students' work while also sharing helpful resources like video lectures and quizzes. The platform will be easy to use and accessible on different devices, ensuring that students can learn from anywhere. All the information regarding students, programs, modules, and assessment results will be

managed with the help of a strong database. Finally, Miss Mary wishes this platform would make learning fun and more effective for all parties.

Current business activities and operations

The business activities of the E Classroom Platform operational procedure is explained below in a short paragraph:

The student may get enrolled into different programs, and each program may have multiple modules that may fit the fields of study. Then teachers are allocated to different modules. Module allocations are assigned to educators within the module to provide content and support to students, so each module is well supported. Then the program module is managed. Every program consists of multiple modules, hence requiring proper organization, management, and allocation of education resources to present a structured curriculum. The resources are allocated in the platform. The platform is designed to allow access to a variety of learning resources such as video lectures, assignments, and quizzes that must be completed sequentially in order to facilitate progressive learning. After that, the students' work is marked, and results for each assessment are generated which students can then view. This activity is important in terms of reflecting student performance accurately.

Business Rule

Each student must enroll in just one program, which can have one or more students.

Students are not allowed to join multiple programs until they finish the first one.

A teacher can instruct several modules, and a module can be taught by different

teachers

Each module should include several assignments, and each assignment must relate to its specific module.

Programs are made up of various modules that students need to complete to graduate.

Teachers can access multiple resources related to the module

An assignment will only be graded if it is submitted by the deadline.

Teachers can only make announcements for the modules they are assigned to, and each announcement must relate to a specific module.

Results should be published according to the student's program and module.

Assumptions

One module can have many teachers assigned to it

Once submitted, students are not allowed to change or remove any assessments.

A single teacher can make multiple announcements.

Entities and attributes identification

Entities and attributes

Entity:

An entity is anything that exists in the real world with independent existence. Each entity is identifiable and can be differentiated from other objects. It has a unique identifier. For example, some entities in a gadget store database are customer, order, and product.

Attribute:

The characteristics of an entity are called attributes. Attributes consist of three types of keys named primary, foreign and non_key. For instance, in students, some of the attributes are Student_ID, Student_Name, and Student_PhoneNumber.

Initial ERD

The entities, attributes and primary key are displayed in table below

STUDENT

S.N.

Attribute



Data type

Size

Constraint

1

Student_ID

Number

18

Primary Key

2

Student_Name

Character

35

Not Null

3

Enrolled_Date

Date

10

Not null

4

Email

Character

35

Unique

5

DOB

Date

10

Not null

 PROGRAM

S.N.

Attribute

Datatype

Size


Constraint

1

Program_Name

Character

22

 Primary Key

2

Program_ID

Number

18

Not null

3

Program_Description

Character

200

Not null

4

Program_Duration

Number

6

MODULES

S.N.

Attribute

Datatype

Size

Constraint

1

Module_ID

Number

12

Not null

2

Module_Name

Character

22

Not null

3

Credit_Hours

Number

5

Not null

4

Teacher_Name

Character

30

Not null

5

Teacher_ID

Number

12

Not null

6

Contact

Number

15

Unique

7

Specialization

Character

20

Not null

8

Resource_ID

Number

12

Not null

9

Resource_Title

Character

30

Not null

10

Resource_Duration

Number

5

Not null

11

Resource_Type

Character

25

Not null

12

Resource_Status

Character

30

Not Null

13

Sequence

Character

10

Unique

14

Assessment_ID

Number

22

Not null

15

Assssment_Title

Character

25

Not null

16



Assessment_Description

Character

1200

Not null

17

Assessment_Status

Character

30

Not Null

18

Weightage

Number

5

Not null

19

Deadline

Date

10

Not null

20

Result_Marks

Number

5

Not null

21

Grade

Character

5

Not null

22

Result_Date

Date

10

Not null

23

Announcement_ID

Number

25

Not null

24

Date_Posted

Date

10

Not null

25

Content

Character

1200

Not null

Entity Relationship Diagram (ERD)

The entity relationship diagram of the business rule is shown below:

Figure 1 ERD (Entity Relationship) Diagram

Normalization

In databases, normalization is a method for arranging information in a smart way. The main purposes of normalization are to cut down on duplicate data and to make sure the data stays consistent. When we use normalization, it helps us manage tasks like adding, removing, or changing records easily, like how cleaning up your room helps you find your favorite book faster. Normalization means splitting your data into different tables and creating connections between them based on certain rules. You can think of each step in normalization to tidy up your room until everything is in its right place and easy to reach.

Unnormalized Form (UNF)

Description:

Unnormalized Form is the first state of data organization, often derived from a single source of requirements. In this form, data may include repeated groups and does not adhere to any structured format.

Characteristics:

Typically, the structure straightforward list of attributes.

Contains a collection of data with possible redundancy.

Repeating groups may exist, indicated by curly braces.

Student: { Student_Name, Student_ID, Enrolled_Date, Email, DOB, Program_Name,

Program_ID, Program_Description, Program_Duration, { Module_ID, Module_Name, Credit_Hours, { Teacher_Name, Teacher_ID, Contact, Specialization, {Announcement_ID, Date_Posted, Content }}, Resource_ID, Resource_Title, Resource_Duration, Resource_Type, Sequence} {Assessment_ID, Assessment_Title, Assessment_Info, Assessment_Status, Weightage, Deadline, Result_Marks, Grade, Result_Date}}

What it looks like: Imagine you have a list with lots of repeated information about students, their courses, and assigned tasks all mixed.

The Problem: A lot of information might be redundant, making it difficult to retrieve or update later.

First Normal Form (1NF)

First Normal Form (1NF) introduces basic structure to the data by ensuring that each cell contains only one value and all entries in a column have same type

Now, let's take the first step towards organizing that closet by ensuring that each item has its place. This is First Normal Form (1NF). If something tends to repeat, like a student taking multiple courses, we need to create separate shelves for those items. For instance, instead of keeping every course listed under each student's name, we create a separate table that neatly lists each student alongside their assigned courses. This way, you can easily add or change courses without messing up other records.

Process to Achieve 1NF:

Separating the repeating groups. After separation, we get:

{ Module_ID, Module_Name, Credit_Hours, { Teacher_Name, Teacher_ID, Contact, Specialization, {Announcement_ID, Date_Posted, Content }}, Resource_ID, Resource_Title, Resource_Duration, Resource_Type, Sequence} {Assessment_ID, Assessment_Title, Assessment_Info, Assessment_Status ,Weightage, Deadline, Result_Marks, Grade, Result_Date}}}

Result After Normalization (1NF)

STUDENT (1NF): Student_Name, Student_ID, Enrolled_Date, Email, DOB, Program_Name, Program_ID, Program_Description, Program_Duration}

Student_Module (1NF): Student_ID, Module_ID, Module_Name, Credit_Hours}

Module_Teacher (1NF): Student_ID, Module_ID, Teacher_Name, Teacher_ID, Contact, Specialization}

Module_Teacher_Announcement (1NF): Student_ID, Module_ID, Teacher_ID,

Announcement_ID, Date_Posted, Content}

Student_Module_Resource (1NF): Student_ID, Module_ID, Resource_ID,
Resource_Title, Resource_Duration, Resource_Type, Resource_Status, Sequence}
Student_Module_Assessment (1NF): Student_ID, Module_ID, Assessment_ID,
Assessment_Title, Assessment_Info, Assessment_Status, Weightage, Deadline,
Result_Marks, Grade, Result_Date}

S.N.

Attribute

Datatype

Size

Constraint

1

Module_ID

Number

12

Not null

2

Module_Name

Character

22

Not null

3

Credit_Hours

Number

5

Not null

4

Teacher_Name

Character

30

Not null

5

Teacher_ID

Number

12

Not null

6

Contact

Number

15

Unique

7

Specialization



Character

20

Not null

8

Resource_ID

Number

12

Not null

9

Resource_Title

Character

30

Not null

10

Resource_Duration



Number

5

Not null

11

Resource_Type

Character

25

Not null

12

Sequence

Character

10

Unique

13

Assessment_ID

Number

22

Not null

14

Assessment_Title

Character

25

Not null

15

Assessment_Info

Character

1200

Not null

16

Assessment_Status

Character

30

Not Null

17

Weightage

Number

5

Not null

18

Deadline

Date

10

Not null

19

Result_Marks

Number

5

Not null

20

Grade

Character

5

Not null

21



Result_Date

Date

10

Not null

22

Announcement_ID

Number

25

Not null

23

Date_Posted

Date

10

Not null

24

Content

Character

1200

Not null

Rules to Follow:

Each column must have unique values.

Repeating groups should be kept in separate tables, thus removing redundancy.

Second Normal Form (2NF)

Second Normal Form (2NF) takes 1NF a step further by eliminating partial dependencies. In this form, all non_key attributes must depend on the entire primary key.

Process to Achieve 2NF:

Student:

Since students have only one primary key, the 2NF of student is given as:

Student(2NF) : (Student_Name, Student_ID, Enrolled_Date, Email, DOB, Program_Name, Program_ID, Program_Description, Program_Duration)

Student_Module:

Primary key: Student_ID, Module_ID

Module_Name is partially dependent on Module_ID

Credit_Hours is partially dependent on Module_ID

2NF Scheme:

Module(2NF):

(Module_ID, Module_Name, Credit_Hours)

Student_Module(2NF):

(Module_ID, Student_ID)

Module_Teacher

Primary key: Module_ID, Teacher_ID

Teacher_Name is partially dependent on Teacher_ID

Contact is partially dependent on Teacher_ID

Specialization is partially dependent on Teacher_ID

2NF Scheme:

Teacher(2NF):

(Teacher_ID, Teacher_Name, Contact, Specialization)

Module_Teacher(2NF):

(Student_ID, Module_ID, Teacher_ID)

Module_Teacher_Announcement

Primary Key: Module_ID, Teacher_ID, Announcement_ID

Date_Posted is partially dependent on Announcement_ID

Content is partially dependent on Announcement_ID

2NF Scheme:

Announcement(2NF):

(Announcement_ID, Date_Posted, Content)

Teacher_Announcement(2NF):

(Student_ID, Module_ID, Teacher_ID, Announcement_ID)

Student_Module_Resource

Primary key: Student_ID, Module_ID, Resource_ID

Resource_Title is partially dependent on Resource_ID

Resource_Duration is partially dependent on Resource_ID

Resource_Type is partially dependent on Resource_ID

Sequence is partially dependent on Resource_ID

2NF Scheme:

Resource(2NF):

(Resource_ID, Resource_Title, Resource_Duration, Resource_Type, Sequence)

Student_Module_Resource(2NF) :

(Student_ID, Module_ID, Resource_ID, Resource_Status)

Student_Module_Assessment

Primary key: **Student_ID, Module_ID, Assessment_ID**

Assessment_Title is partially dependent on Assessment_ID

Assessment_Info is partially dependent on Assessment_ID

Assessment_Status is partially dependent on Assessment_ID

Weightage is partially dependent on Assessment_ID

Deadline is partially dependent on Assessment_ID

Result_Marks is partially dependent on Assessment_ID

Grade is partially dependent on Assessment_ID

Result_Date is partially dependent on Assessment_ID

2NF Scheme:

Assessment(2NF):

(**Assessment_ID, Assessment_Title, Assessment_Info, Weightage, Deadline**)

Module_Asst(2NF):

(**Student_ID, Module_ID, Assessment_ID, Assessment_Status** , Result_Marks, Grade,

Date_Posted)

Final Results after 2NF

Student(2NF): (Student_Name, Student_ID, Enrolled_Date, Email, DOB, Program_Name, Program_ID, Program_Description, Program_Duration)

Module(2NF): (Module_ID, Module_Name, Credit_Hours)

Student_Module(2NF): (Module_ID, Student_ID)

Teacher(2NF): (Teacher_ID, Teacher_Name, Contact, Specialization)

Module_Teacher(2NF): (Student_ID, Module_ID, Teacher_ID)

Announcement(2NF): (Announcement_ID, Date_Posted, Content)

Teacher_Announcement(2NF): (Student_ID, Module_ID, Teacher_ID, Announcement_ID)

Resource(2NF): (Resource_ID, Resource_Title, Resource_Duration, Resource_Type, Sequence)

Student_Module_Resource(2NF): (Student_ID, Module_ID, Resource_ID)

Assessment(2NF) : (Assessment_ID, Assessment_Title, Assessment_Info, Weightage, Deadline)

Module_Assst(2NF): (Student_ID, Module_ID, Assessment_ID, Assessment_Status, Result_Marks, Grade, Date_Posted)

Rules to Follow:

Ensure that no non_key attribute is partially dependent on the primary key.

All attributes should directly relate to the primary key.

Third Normal Form(3NF)

Third Normal Form (3NF) aims to eliminate transitive dependencies, which occur when a non_key attribute depends on another non_key attribute. Let's refine the organization even further! Second Normal Form (2NF) helps eliminate redundancy that might still exist.

Rules to Achieve 3NF:

Ensure the database is in 2NF

Eliminate transitive dependencies by ensuring all non_key attributes are directly related to the primary key.

Process to Achieve 3NF

Student:

Program_Name directly depends on Program_ID

Program_Description directly depends on Program_ID

Program_Duration directly depends on Program_ID

Thus, the required 3NF will be:

Student(3NF):

(Student_Name, Student_ID, Enrolled_Date, Email, DOB, Program_Name, Program_ID)

Program(3NF):

(Program_ID, Program_Name, Program_Description, Program_Duration)

Module

All the non key attributes directly depends on Module_ID resulting in absence of transitive dependency.

Thus, the required 3NF will be:

Module(3NF):

(Module_ID, Module_Name, Credit_Hours)

Student_Module

Non key attributes in Student_Module is absent. Thus, the required 3NF will be

Student_Module(3NF):

(Student_ID, Module_ID)

Teacher

All the non key attributes directly depend on Teacher_ID resulting in absence of transitive dependency

Thus, the required 3NF will be

Teacher(3NF):

(Teacher_ID, Teacher_Name, Contact, Specialization)

Module_Teacher

Non key attributes in Module_Teacher is absent. Thus, the required 3NF will be

Module_Teacher(3NF):

(Student_ID, Module_ID, Teacher_ID)

Announcement

All the non key attributes directly depend on Announcement_ID resulting in absence of transitive dependency

Thus, the required 3NF will be

Announcement(3NF):

(Announcement_ID, Date_Posted, Content)

Teacher_Announcement

Non key attributes in Teacher_Announcement is absent. Thus, the required 3NF will be

Teacher_Announcement(3NF):

(Student_ID, Module_ID, Teacher_ID, Announcement_ID)

Resource

All the non key attributes directly depend on Resource_ID resulting in absence of transitive dependency

Thus, the required 3NF will be:

Resource(3NF):

(Resource_ID, Resource_Title, Resource_Duration, Resource_Type, Sequence)

Stud_Module_Resource

All the non key attributes directly depend on Resource_Status resulting in absence of transitive dependency

Thus, the required 3NF will be:

Stud_Module_Resource(3NF):

(Student_ID, Module_ID, Resource_ID)

Assessment

All the non key attributes directly depend on Assessment_ID resulting in absence of transitive dependency

Thus, the required 3NF will be

Assessment(3NF):

(Assessment_ID, Assessment_Title, Assessment_Info, Weightage, Deadline)

Module_Assessment

Result_Marks, Grade, Date_Posted are directly dependent to Student_ID, Module_ID and Assessment_ID resulting in absence of transitive dependency

Thus, the required 3NF will be

Module_Asst(3NF):

(Student_ID, Module_ID, Assessment_ID, Assessment_Status, Result_Marks, Grade, Date_Posted)

Final Results after 3NF:

Student(3NF) :(Student_Name, Student_ID, Enrolled_Date, Email, DOB, Program_Name, Program_ID)

Program(3NF) :(Program_ID, Program_Name, Program_Description, Program_Duration)

Module(3NF) :(Module_ID, Module_Name, Credit_Hours)

Student_Module(3NF) :(Student_ID, Module_ID)

17 Teacher(3NF) : (Teacher_ID, Teacher_Name, Contact, Specialization)

1 Module_Teacher(3NF) : (Student_ID, Module_ID, Teacher_ID)

Announcement(3NF) : (Announcement_ID, Date_Posted, Content)

6 Teacher_Announcement(3NF) : (Student_ID, Module_ID, Teacher_ID, Announcement_ID)

1 Resource(3NF) : (Resource_ID, Resource_Title, Resource_Duration, Resource_Type, Sequence)

Stud_Module_Resource(3NF) : (Student_ID, Module_ID, Resource_ID,)

1 Assessment(3NF) : (Assessment_ID, Assessment_Title, Assessment_Info, Weightage, Deadline)

Module_Asst(3NF) : (Student_ID, Module_ID, Assessment_ID, Assessment_Status, Result_Marks, Grade, Date_Posted)

12) Conclusion


By following these normalization steps—starting from Unnormalized Form through First, Second, and Third Normal Forms—you ensure that your database remains organized, minimizes redundancy, and maintains data integrity. This systematic approach makes your data easier to manage, reduces errors, and enhances overall efficiency. By organizing data through these normalization steps, we can manage our databases like a tidy closet where everything is accessible and efficient.

13) Data Dictionary

Student Table:

S.N.

Attribute

 Data type

Size

Constraint

1

Student_ID

Number

18

Primary Key

2

Student_Name

Character

35

Not Null

3

Enrolled_Date

Date

10

Not null

4

Email

Character

35

Unique

5

DOB

Date

10

Not null

6

Program_ID

Number

18

Foreign Key

Table 1 Student table after 3NF



Program Table:

S.N.

Attribute

Data type

Size

Constraint

1

Program_Name

Character

18

Primary Key

2

Program_ID

Number

22

Not Null

3

Program_Description

Character

200

Not null

4

Program_Duration

Number

6

Unique

Table 2 Program table after 3NF



Module Table:

S.N.

Attribute

Data type

Size

Constraint

1

Module_ID

Number

12

Primary Key

2

Module_Name

Character

22

Not Null

3

Credit_Hours

Number

5

Not null

Table 3 Module table after 3NF

Teacher Table:

S.N.

Attribute

Data type

Size

Constraint

1

Teacher_Name

Character

30

Not null

2

Teacher_ID

Number

12

Primary Key

3

Contact

Number

15

Unique

4

Specialization

Character

20

Not null

Table 4 Teacher table after 3NF

Assessment Table:

S.N.

Attribute

Data type

Size

Constraint

1

Assessment_ID

Number

22

Primary_Key

2

Assessment_Title

Character

25

Not null

3

Assessment_Info

Character

1200

Not null

4

Weightage

Number

5

Not null

5

Deadline

Date

10

Not Null

6

Assessment_Status

Character

30

Not Null

Table 5 Assessment table after 3NF

Announcement Table:

S.N.

Attribute

Data type

Size

Constraint

1

Announcement_ID

Number

25

Primary Key

2

Date_Posted

Date

10

Not null

3

Content

Character

1200

Not null

Table 6 Announcement table after 3NF

Resource Table:

S.N.

Attribute

Data type

Size

Constraint

1

Resource_ID

Number

12

PrimaryKey

2

Resource_Title

Character

30

Not null

3

Resource_Duration

Number

5

Not null

4

Resource_Type

Character

25

Not null

5

Sequence

Character

10

Unique

Table 7 Resource table after 3NF

Bridging Entities

Student_**Module**

S.N.

Attribute

Data type

Size

Constraint

Composite Constraint

1

Student_**ID**

Number

18

Foreign Key

Primary **Key**

2

Module_ID

Number

12

Foreign Key

Primary Key

Table 8 Student_ Module table after 3NF

Module_Teacher

S.N.

Attribute

Data type

Size

Constraint

Composite Constraint

1

Student_ID

Number

18

Foreign Key

Primary Key

2

Module_ID

Number

12

Foreign Key

Primary Key

3

Teacher_ID

Number

12

Foreign Key

Primary Key

Table 9 Module_Teacher table after 3NF

Teacher_Announcement

S.N.

Attribute

Data type

Size

Constraint

Composite Constraint

1

Student_ID

Number

18

Foreign Key

Primary Key

2

Module_ID

Number

12

Foreign Key

Primary Key

3

Teacher_ID

Number

22

Foreign Key

Primary Key

4

Announcement_ID

Number

25

Foreign Key

Primary Key

Table 10 Teacher_Announcement table after 3NF

Module_Resource

S.N.

Attribute

Data type

Size

Constraint

Composite Constraint

1

Student_ID

Number

18

Foreign Key

Primary Key

2

Module_ID

Number

12

Foreign Key

Primary Key

3

Resource_ID

Number

12

Foreign Key

Primary Key

Table 11 Module_Resource table after 3NF

Module_Ass

S.N.

Attribute

Data type

Size

Constraint

Composite Constraint

1

Module_ID

Number

12

Foreign Key

Primary Key

2

Student_ID

Number

18

Foreign Key

Primary Key

3

Assessment_ID

Number

22

Foreign Key


Primary Key

4

Assessment_Status

Character

30

 24 **Not Null****5**

Result_Marks

Number**5****Not null****6**

Grade

Character

5**Not null****7**

Result_Date

Date

10

Not null

Table 12 Module_Assessment table after 3NF

Final ERD

Figure 1 Final ERD

14) Implementation

14.1) Creating new user.

Figure 2 Creating new user

14.2) Creating student table and describing table

Figure 3 Creating student table.

14.3) Creating program table and describing table

Figure 4 Creating program table

14.4) Creating module table and describing table

Figure 5 Creating module table

14.5) Creating Student_Module table and describing table

Figure 6 Creating Student_Module table

14.6) Creating teacher table and describing table

Figure 7 Creating teacher table

14.7) Creating Module_Teacher table and describing table

Figure 8 Creating Module_Teacher table

14.8) Creating announcement table and describing table

Figure 9 Creating announcement

14.9) Creating Teacher_Announcement table and describing table

Figure 10 Creating Teacher_Announcement table

14.10) Creating Resources table and describing table

Figure 11 Creating Resources table

14.11) Creating Stud_Module_Resource and describing table

Figure 12 Creating Stud_Module_Resource

14.12) Creating assessment table and describing table

Figure 13 Creating assessment table

14.13) Creating Module_Asst table and describing table

Figure 14 Creating Module_Asst table

Inserting the values in table we created

14.14) Inserting the values in program table:

Figure 15 Inserting values in program table

14.15) Inserting values in module table

Figure 16 Inserting values in module table

14.16) Inserting values in student table:

Figure 17 17 Inserting values in student table:

14.17) Inserting values in teacher table:

Figure 18 Inserting values in teacher table



14.18) Inserting values in Student_Module table:

Figure 19 Inserting values in student_module

14.19) Inserting values in Module_Teacher

Figure 20 Inserting values in Module_Teacher

14.20) Inserting values in Announcement table.

Figure 21 Inserting values in Announcement table.

 15

14.21) Inserting values in Techer_Announcement

Figure 22 Inserting values in Techer_Announcement

 12

14.22) Inserting values in Resource table

Figure 23 Inserting values in Resource table

14.23) Inserting values in assessment

Figure 24 Inserting values in assessment

14.24) Inserting values in Module_Asst table

Figure 25 Inserting values in Module_Asst table

15) Querying of Database

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

Query- SELECT

```
P.PROGRAM_ID AS Program_ID,
P.PROGRAM_NAME AS Program_Name,
COUNT(S.STUDENT_ID) AS Total_Students
FROM
PROGRAM P
LEFT JOIN
STUDENT S
ON
P.PROGRAM_ID = S.PROGRAM_ID
GROUP BY
P.PROGRAM_ID, P.PROGRAM_NAME
ORDER BY
```


P.PROGRAM_ID;

This query makes sure that all the programs are listed in a database. Even if the students are not enrolled, all the programs that are listed in the database are displayed.

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

17) Critical Evaluation

17.1) Evaluation of the Module, Its Purpose, and Connections to Other Subjects:

The modules in the system play a key role in structuring academic programs. It is designed to cover a wide range of modules like programming, data analysis, cybersecurity, and digital marketing. Each module has its own attributes such as its name, ID, and credit hours.

Modules are the glue that holds everything together. They connect teachers to their areas of expertise, students to their courses, and resources like lectures, quizzes, and assignments. This makes it easy to manage what each person needs to do within the system.

What makes modules even more important is how they relate to other parts of the curriculum. For example, they're linked to specific programs, which create a logical flow for students to follow step by step. Modules also integrate multiple disciplines, like combining digital marketing with data analysis to evaluate marketing strategies—showing how interconnected knowledge can be.

17.2) Assessment of the Coursework:

This coursework helps to tackle problems such as managing students, teachers, and assessments in a structured way.

The database design follows proper normalization steps i.e the data is stored in a

efficient way with no unnecessary repetition.

The relationship between entities like students, modules, and teachers are defined, which makes it easy to add more entities in future.

It mirrors real-life challenges of managing academic platforms, preparing you for practical applications.

Areas for Improvement:

Add tools for analyzing data, such as tracking student progress or module success rates, to make the system more insightful.

Include features that allow students to progress at their own pace, like marking modules by difficulty level or prerequisites.

In summary, the module system is well-structured and has a lot of potential for practical use. It connects the dots between different elements of education, creating a strong foundation for an interactive and scalable learning platform.

Bibliography

 39

Chris, K. (2022, 12 21). Database. Retrieved from FreeCodeCamp:

<https://www.freecodecamp.org/news/database-normalization-1nf-2nf-3nf-table-examples/>

 1

GFG. (2024, 6 3). Retrieved from geeksforgeeks: <https://www.geeksforgeeks.org/how-to-draw-entity-relationship-diagrams/>

 13

Schools. (n.d.). SqlTutorial. Retrieved from W3Schools:

https://www.w3schools.com/sql/sql_insert.asp

 30

team, O. (2023, 5 3). Ovaledge. Retrieved from Ovaledge:

<https://www.ovaledge.com/blog/what-is-data-dictionary>