



CC5051NI Databases

100% Individual Coursework

Autumn 2024

Credit: 15 Semester Long Module

Student Name: Sadiksha Karki

London Met ID: 23049347

Assignment Submission Date: Thursday, January 23, 2025

Word Count: 5690

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.

Similarity Index Report



Page 1 of 84 - Cover Page

Submission ID trn:oid::3618:79917382

23049347 SadikshaKarki.docx

Islington College,Nepal

Document Details

Submission ID
trn:oid::3618:79917382

77 Pages

Submission Date
Jan 23, 2025, 10:21 AM GMT+5:45

5,532 Words

Download Date
Jan 23, 2025, 10:24 AM GMT+5:45

33,275 Characters

File Name
23049347 SadikshaKarki.docx

File Size
38.7 KB



Page 1 of 84 - Cover Page

Submission ID trn:oid::3618:79917382

28% Overall Similarity

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

Match Groups

-  **119** Not Cited or Quoted 26%
Matches with neither in-text citation nor quotation marks
-  **11** Missing Quotations 2%
Matches that are still very similar to source material
-  **0** Missing Citation 0%
Matches that have quotation marks, but no in-text citation
-  **0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 11%  Internet sources
- 1%  Publications
- 27%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

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

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

Table of Contents

1	INTRODUCTION	1
1.1	BUSINESS DESCRIPTION AND ITS FORTE	1
1.2	CURRENT BUSINESS ACTIVITIES AND OPERATIONS.....	2
1.3	BUSINESS RULES	4
1.3.1	<i>Assumptions</i>	4
2	INITIAL ENTITY RELATIONSHIP DIAGRAM	5
2.1	IDENTIFICATION OF ENTITIES AND ATTRIBUTES.....	5
2.2	ENTITY RELATIONSHIP DIAGRAM (ERD)	7
3	NORMALIZATION.....	8
3.1	UNF.....	8
3.2	1NF	9
3.3	2NF.....	10
3.4	3NF	13
4	DATA DICTIONARY AND FINAL ERD	17
4.1	DATA DICTIONARY.....	17
4.2	FINAL ERD	23
5	IMPLEMENTATION.....	24
5.1	CREATE USER.....	24
5.2	CREATE TABLES	25
5.2.1	<i>Program Table</i>	25
5.2.2	<i>Student Table</i>	25
5.2.3	<i>Module Table</i>	26
5.2.4	<i>Module_Student Table</i>	26
5.2.5	<i>Teacher Table</i>	26
5.2.6	<i>Teacher_Module Table</i>	27
5.2.7	<i>Teacher_Announcement Table</i>	27
5.2.8	<i>Announcement Table</i>	27
5.2.9	<i>Assessment Table</i>	28
5.2.10	<i>Result table</i>	28
5.2.11	<i>Assessment_Result Table</i>	28
5.2.12	<i>Resource Table</i>	29
5.2.13	<i>Resource_Module_Student Table</i>	29
5.3	ADDING ROWS TO TABLES.....	30

5.3.1	<i>INSERT of Program Table</i>	30
5.3.2	<i>INSERT of Student Table</i>	31
5.3.3	<i>INSERT of Module Table</i>	32
5.3.4	<i>INSERT of Module_Student Table</i>	33
5.3.5	<i>INSERT of Teacher Table</i>	34
5.3.6	<i>INSERT of Teacher_Module Table</i>	35
5.3.7	<i>INSERT of Announcement Table</i>	36
5.3.8	<i>INSERT of Teacher_Announcement Table</i>	37
5.3.9	<i>INSERT of Assessment Table</i>	38
5.3.10	<i>INSERT of Result Table</i>	39
5.3.11	<i>INSERT of Assessment_Result Table</i>	40
5.3.12	<i>INSERT of Resource Table</i>	41
5.3.13	<i>INSERT of Resource_Module_Student Table</i>	42
5.4	CONTENT OF TABLES.....	43
5.4.1	<i>Program Table</i>	43
5.4.2	<i>Student Table</i>	43
5.4.3	<i>Module Table</i>	44
5.4.4	<i>Module_Student Table</i>	44
5.4.5	<i>Teacher Table</i>	45
5.4.6	<i>Teacher_Module</i>	45
5.4.7	<i>Teacher_Announcement Table</i>	45
5.4.8	<i>Announcement Table</i>	46
5.4.9	<i>Assessment Table</i>	46
5.4.10	<i>Result Table</i>	47
5.4.11	<i>Assessment_Result Table</i>	47
5.4.12	<i>Resource Table</i>	48
5.4.13	<i>Resource_Module_Student Table</i>	48
6	DATABASE QUERY	49
6.1	INFORMATION QUERY	49
6.2	TRANSACTION QUERY	53
7	CRITICAL EVALUATION	58
8	DUMP FILE CREATION AND DROP QUERIES	60
8.1	DUMP FILE COMMAND EXECUTION	60
8.2	DROP TABLE QUERY	62
8.2.1	<i>Drop Program Table</i>	62
8.2.2	<i>Drop Student Table</i>	62

8.2.3	<i>Drop Module Table</i>	63
8.2.4	<i>Drop Module_Student Table</i>	63
8.2.5	<i>Drop Teacher Table</i>	63
8.2.6	<i>Drop Teacher_Module Table</i>	63
8.2.7	<i>Drop Teacher_Announcement Table</i>	64
8.2.8	<i>Drop Announcement Table</i>	64
8.2.9	<i>Drop Assessment Table</i>	64
8.2.10	<i>Drop Result Table</i>	64
8.2.11	<i>Drop Assessment_Result Table</i>	65
8.2.12	<i>Drop Resource Table</i>	65
8.2.13	<i>Drop Resource_Module_Student Table</i>	65
	BIBLIOGRAPHY	66

Table of Figures

Figure 1 Initial ERD.....	7
Figure 2 Final ERD	23
Figure 3 Applying Create user command	24
Figure 4 Grant Connect command	24
Figure 5 Create Program Table.....	25
Figure 6 Create Student Table	25
Figure 7 Create Module Table	26
Figure 8 Create Module_Student Table.....	26
Figure 9 Create Teacher Table.....	26
Figure 10 Create Teacher_Module Table	27
Figure 11 Create Teacher_Announcement Table	27
Figure 12 Create Announcement Table	27
Figure 13 Create Assessment Table.....	28
Figure 14 Create Result Table	28
Figure 15 Create Assessment_Result Table	28
Figure 16 Create Resource Table.....	29
Figure 17 Create Resource_Module_Student Table	29
Figure 18 Insert values in Program Table	30
Figure 19 Insert values in Student Table	31
Figure 20 Insert values in Module Table	32
Figure 21 Insert values in Module_Student table	33
Figure 22 Insert values in Teacher Table.....	34
Figure 23 Insert values in Teacher_Module Table	35
Figure 24 Insert values in Announcement Table	36
Figure 25 Insert values in Teacher_Announcement Table	37
Figure 26 Insert values in Assessment Table.....	38
Figure 27 Insert values in Result Table	39
Figure 28 Insert values in Assessment_Result Table	40
Figure 29 Insert values in Resource Table.....	41
Figure 30 Insert values in Resource_Module_Student Table	42

Figure 31 Content of Program Table	43
Figure 32 Content of Student Table.....	43
Figure 33 Content of Module Table.....	44
Figure 34 Content of Module_Student Table	44
Figure 35 Content of Teacher Table	45
Figure 36 Content of Teacher_Module Table.....	45
Figure 37 Content of Teacher_Announcement Table.....	45
Figure 38 Content of Announcement Table.....	46
Figure 39 Content of Assessment Table	46
Figure 40 Content of Result Table.....	47
Figure 41 Content of Assessment_Result Table	47
Figure 42 Content of Resource Table	48
Figure 43 Content of Resource_Module_Student Table	48
Figure 44 Information Query Question no.1 output	49
Figure 45 Information Query Question no.2 output	50
Figure 46 Information Query Question no.3 output	51
Figure 47 Information Query Question no. 4 output	51
Figure 48 Information Query Question no.5 output	52
Figure 49 Transaction Query Question no.1 output.....	53
Figure 50 Transaction Query Question no.2 output.....	54
Figure 51 Transaction Query Question no.3 output.....	55
Figure 52 Transaction Query Question no.4 output.....	56
Figure 53 Transaction Query Question no.5 output.....	57
Figure 54 Dump file execution command	60
Figure 55 Dump file execution command	61
Figure 56 Drop Query of Program Table.....	62
Figure 57 Drop Query of Student Table	62
Figure 58 Drop Query of Module Table].....	63
Figure 59 Drop Query of Module_Student Table	63
Figure 60 Drop Table of Teacher Table	63
Figure 61 Drop Table of Teacher_Module Table	63

Figure 62 Drop Query of Teacher_Announcement Table	64
Figure 63 Drop Query of Announcement Table	64
Figure 64 Drop Query of Assessment Table.....	64
Figure 65 Drop Query of Result Table	64
Figure 66 Drop Query of Assessment_Result Table	65
Figure 67 Drop Query of Resource Table.....	65
Figure 68 Drop Query of Resource_Module_Student Table	65

Table of Tables

Table 1 Attributes of Student Entity	5
Table 2 Attributes of Program Entity.....	6
Table 3 Attributes of Module Entity	6
Table 4 Attributes of Student Entity	17
Table 5 Attributes of Program Entity.....	18
Table 6 Attributes of Module entity.....	18
Table 7 Attributes of Module_Student Entity.....	18
Table 8 Attributes of Teacher Entity	19
Table 9 Attributes of Teacher_Module Entity	19
Table 10 Attributes of Teacher_Announcement Entity	20
Table 11 Attributes of Announcement Entity	20
Table 12 Attributes of Assessment Entity	20
Table 13 Attributes of Result Entity	21
Table 14 Attributes of Assessment_Result Entity	21
Table 15 Attributes of Resource Entity	22
Table 16 Attributes of Resource_Module_Student Entity	22

1 Introduction

1.1 Business Description and its forte



(Islington College, 2024)

Ms. Mary, the founder and the principal of Islington College, which was established in 1996, is known for its excellence in Information Technology (IT) education. The college is located in the heart of Kathmandu; the capital of Nepal. Islington college has become well recognized for its best education and variety of extracurricular activities. The college specializes in Information Technology (IT) and business degrees programs that are combined with both theoretical knowledge and practical experience. The college offers various programs and opportunities to the young people, preparing them for future careers in the tech and business sectors.

Islington college follows an international education system, providing students an experience similar to studying abroad, offering a global standard of education in Nepal. The college aims to be a leader in transforming education in Nepal. The diverse subjects prepare students with the knowledge and skills to compete in this rapidly growing time. It has successfully made an impact on the country by transforming young students into responsible, highly skilled professionals, that

are now working on a respected-fields. The college emphasizes values like leadership, professionalism, excellence and responsiveness. Also, students at Islington college are encouraged to develop soft skills that are essential for success in any field they are interested in. It has always valued the talent and skills of the students and is dedicated to helping them become more innovative and creative. The institution also focuses on character building and personal growth ensuring the students are well-prepared for the real-life scenarios. This approach to education helps students develop a strong sense of responsibility, making sure they are ready to lead with a clear sense of purpose in both their life and career. Many of the college's graduates are playing important roles in leading global companies. Under Ms. Mary's leadership, the college is committed to providing the students with the best education in IT field with strong technical skills and opportunities.

As the founder, Ms. Mary recognized that in order to keep up with the modern education system, the college needed to take a step further and introduce the E-Classroom Platform for the academic operations. With this digital learning environment, the college aims to enhance the learning experience by providing an interactive platform for students, teachers and administrators. Students can also access the platform anytime, anywhere. This platform will allow the students to access course materials, submit their works and also get the announcements, track progress from the teachers. The E-Classroom platform will enhance the interaction between students and teachers and management itself. Introducing the E-Classroom platform will also reflect the college's mission to provide modern education while encouraging more creativity and innovation.

1.2 Current Business Activities and Operations

Islington college offers various Information Technology (IT) and business degree programs, such as BSc in Computing, BSc in Networking, BSc in multimedia, BA (Hons) Business Administration, BA (Hons) Accounting and finance and many more. These programs are combined with both theoretical knowledge and practical experience, preparing students for fields of technology and business. The college provides different types of classes such as lecture class,

Tutorial class and workshop classes. In the lecture class, professors deliver the concept but since there are many students in the lecture class, there is high chance of one-way communication where the professor or the lecturer provides the information of certain topic. In tutorial class, it is allocated in classroom with smaller groups of students where the students get personalized attention from their respective tutors and also can clear their misunderstanding or questions about the topic taught in lecture halls. And in workshop class, the students are provided with hands-on sessions to get the practical knowledge and chance to work with the real-world tools and projects. These three types of classes are provided every week for every program. This leads to the proper understanding of the concept that the program is trying to give to the students.

The college is well-known for its extracurricular activities and student engagement. On various occasions, Islington college offers events like international tour, Islington yatra, aspire, spring carnival, a day in Britain, sports-week Christmas, Holi and many more. Islington college provides scholarship opportunities for the students which is highly valued. It provides two type of scholarship that are ING postgraduate scholarship and the Attitude, Academics and Attendance (AAA) Scholarship. The ING postgraduate scholarship will lead you to fully funded master's degree and the AAA scholarship will lead the undergraduate scholarship to students from each intake. These scholarship opportunities encourage students to work hard and do their best.

To successfully implement the E-Classroom Platform, Ms. Mary has pointed out some important requirements for designing the database. The database must be secure and easy to use for all different type of users; administrators, teachers and students. The system should manage all the academic activities effectively. She requires a database that handles all information related to students, teachers, modules, assessments, resources and announcements. The database should effectively manage details of student enrollment details with their specific programs, personal information, academic progress. It should track each module and its resources, as well as manage assessment with its deadlines and weightages. The platform should handle the assignments and grading for each module. Teachers must be able to create and manage announcements and also manage resources for their specific modules. And at the end, the database must generate the results that reflect student's performance in each module.

1.3 Business Rules

- A student can enroll in only one program.
- A program can have many students.
- Students must complete all required modules to complete their program.
- Each program consists of multiple modules.
- A module can belong to multiple programs.
- Each module can have multiple teachers.
- Each teacher is assigned to teach only one specific module.
- A module can have one or more assessments.
- Each assessment is linked to only one module.
- Each assessment must have details such as ID, title, deadline and weightage.
- Each assessment will have results for each student.
- Assessments can have different formats, such as quiz, multiple-choice question, coursework, unseen examination and practical examination.
- Each student's result is linked to specific assessment in a module.
- Each module contains various resources.
- Each resource is linked to only one module.
- Each resource in a module must be completed in predefined sequence.
- Resources can have different formats such as slides and videos and multimedia content.
- Teachers can post multiple announcements for their respective modules.
- Each announcement is linked to a specific module.
- Announcement will be only visible to students who are currently enrolled in the module.

1.3.1 Assumptions

- Students must complete all required modules to complete their program.
- Each teacher is assigned to teach only one specific module.
- Announcement will be only visible to students who are currently enrolled in the module.
- Assessments can have different formats, such as quiz, multiple-choice question, coursework, unseen examination and practical examination.
- Resources can have different formats such as slides and videos and multimedia content.

2 Initial Entity Relationship Diagram

An Initial Entity Relationship Diagram (ERD) is a foundational visual map that presents the main entities such as students, program and module and their attributes. It helps to understand relationship between the main entities and database structure.

2.1 Identification of Entities and Attributes

The following table displays the entities and their attributes with its data type, size and constraints. The data type defines the kind of data that can be stored. Some of the data types are:

- Character Data Type = Character Data Types are used to store text data.
- Number Data Type = Number Data Types are used to store numeric values.
- Date Data Type = Date Data Types are used to store date or time-values.

Student

S.No.	Attribute Name	Data Type	Size	Constraint
1	Student_ID	Number	10	Primary key
2	Student_Name	Character	50	Not Null
3	Date_of_Birth	Date	-	Not Null
4	Email	Character	60	Unique
5	Enrollment_Date	Date	-	Not Null

Table 1 Attributes of Student Entity

Program

S.No.	Attribute Name	Data Type	Size	Constraint
1	Program_ID	Number	10	Primary key
2	Program_Name	Character	50	Not Null
3	Program_Duration	Number	2	Not Null
4	Program_Start_Date	Date	-	Not Null
5	Program_End_Date	Date	-	Not Null

*Table 2 Attributes of Program Entity***Module**

S.No.	Attribute Name	Data Type	Size	Constraint
1	Module_ID	Number	10	Primary key
2	Module_Name	Character	40	Not Null
3	Module_Duration	Number	2	Not Null
4	Module_Start_Date	Date	-	Not Null
5	Module_End_Date	Date	-	Not Null

Table 3 Attributes of Module Entity

2.2 Entity Relationship Diagram (ERD)

Entity Relationship diagram (ERD) describes the structure of the database. ERD represents the relationship and dependencies between entities. The main components of ERD are entities, attributes, relationships (one to many, many to one and one to one). Entity is divided into two types; they are strong entity type which contains key attributes and weak entity type which does not contain key attributes. Attributes are generally divided into four and they are Key attributes, Composite attribute, Multi-valued attribute and Derived attribute.

Here, the entities shown in the diagram are Students, Program and Module. The attributes of Students are Student_ID, Student_Name, Date_of_Birth, Email, Enrollment_Date. The attributes of Program are Program_ID, Program_Name, Program_Duration, Program_Start_Date, Program_End_Date. The attributes of Module are Module_ID, Module_Name, Module_Duration, Module_Start_Date, Module_End_Date.

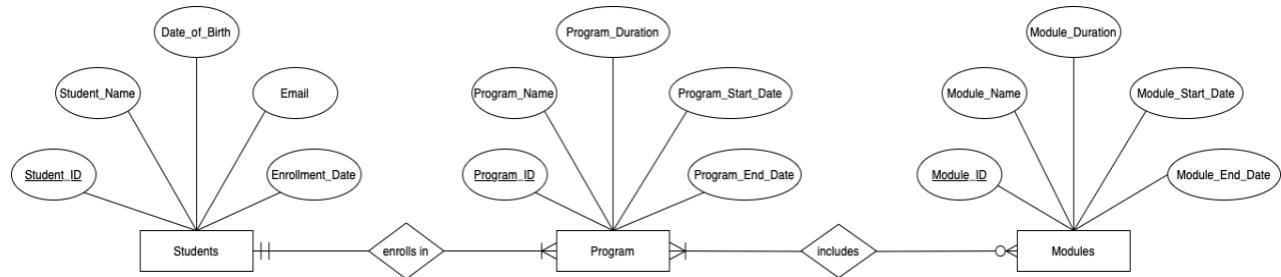


Figure 1 Initial ERD

3 Normalization

Normalization is a systematic approach to organize the data in database. It eliminates data redundancy and avoid anomalies. Normalization process involves breaking down large tables into smaller ones with proper defined relationship between them. The advantages of normalization are it helps to reduce data redundancy, improve query performance and it helps to simplify database. Normalization ensures that the data is systematically structured and preserves its accuracy and consistency. (Geeks for geeks, 2025)

3.1 UNF

UNF stands for Unnormalized form which represents the beginning phase of normalization process. UNF includes repeating group and complex structure which can result to data redundancy. This form contains all the attributes and displays repeating groups using curly brackets. Below is the unnormalized form, where repeating groups are separated by using curly brackets. The student table contains repeating groups like Module, Teacher, Assessment and Resource.

Student (Student_ID, Student_Name, Phone_Number, Email, Date_of_Birth, Address, Enrollment_date, Program_ID, Program_Name, Program_Duration, Program_Start_Date, Program_End_Date, {Module_ID, Module_Name, Module_Duration, Total_Resources, Total_Assessments, {Teacher_ID, Teacher_Name, Email, Phone_Number, Department, Announcement_ID, Title, Announcement_Duration, Issued_date}, {Assessment_ID, Assessment_Title, Deadline, Weightage, Result_ID, Total_Marks, Total_Marks_Obtained, Component_Details, Feedback}, {Resource_ID, Title, Type, Resource_Duration, Sequence_Number}})

3.2 1NF

In First Normal Form (1NF), repeating groups are removed by organizing the data into separate tables, ensuring all values in each table are atomic with assigning a primary key and foreign key to show relation between tables.

- Student (Student_ID, Student_Name, Phone_Number, Email, Date_of_Birth, Address, Enrollment_date, Program_ID, Program_Name, Program_Duration, Program_Start_Date, Program_End_Date)
- Module (Module_ID, Student_ID, Module_Name, Module_Duration, Total_Resources, Total_Assessments)
- Teacher (Teacher_ID, Module_ID, Student_ID, Teacher_Name, Email, Phone_Number, Department, Announcement_ID, Title, Announcement_Duration, Issued_date)
- Assessment (Assessment_ID, Module_ID, Student_ID, Assessment_Title, Deadline, Weightage, Result_ID, Total_Marks, Total_Marks_Obtained, Component_Details, Feedback)
- Resource (Resource_ID, Module_ID, Student_ID, Title, Type, Resource_Duration, Sequence_Number)

Therefore, the data from the UNF is now separated into groups assigning their primary and foreign key. The resulting tables are Student, Module, Teacher, Assessment and Resource. The primary key for student table is Student_ID. For module table, primary key is Module_ID and foreign key is Student_ID. For teacher table, primary key is Teacher_ID and foreign key is Module_ID and Student_ID. Primary key for assessment table is Assessment_ID and foreign key is Module_ID, and Student_ID. For Resource table, primary key is Resource_ID and foreign key is Module_ID and Student_ID. Student table contains details of student details such as their name, contact information, enrollment and module table contains information about module name, resources, assessments and duration, teacher table contains details of teacher details and announcement

related to that module. Assessment contains details such as deadline, weightage, marks and resource contain resources related to the module such as its types and titles.

3.3 2Nf

In Second Normal Form (2NF), the goal is to remove partial dependencies and ensure that the non-key attribute is fully dependent on the primary key. When a non-prime attribute depends on only part of the composite primary key, we separate them in different tables. Here in this 2NF we have created new tables because of partial dependencies. The new tables are Teacher_Announcement, Teacher_Module_Student, Assessment_Result and Resource_Module_Student. Explanation of the process is given below:

- Student (Student_ID, Student_Name, Phone_Number, Email, Date_of_Birth, Address, Enrollment_date, Program_ID, Program_Name, Program_Duration, Program_Start_Date, Program_End_Date)
- The student table remains the same as there is no partial dependencies as the primary key of student table is a single attribute and all non-key attributes depend on Student_ID.
- Module (Module_ID, Module_Name, Module_Duration, Total_Resources, Total_Assessments)
- In 1NF there was student_ID which was removed in 2nf and separate table was create for it because there was partial dependency.
- Module_Student (Module_ID, Student_ID)
- A new junction table was created from the Student_ID of the module table to eliminate the partial dependency.

- Teacher (Teacher_ID, Teacher_Name, Email, Phone_Number, Department)
 - In this teacher table, we have removed announcement related attributes because the non-key attribute of announcement only depends on the Teacher_ID but not on full composite key that are Teacher_ID, Module_ID, and Announcement_ID and now all the other attributes are depended on the primary key; Teacher_ID.
- Teacher_Announcement (Teacher_ID, Announcement_ID, Title, Announcement_Duration, Issued_date)
 - The new table Teacher_Announcement was created to eliminate the partial dependency in Teacher table.
- Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
 - Since there was partial dependency in Assessment table in 1NF, the modified table of assessment table only contains assessment attributes which fully depend on only one primary key.
- Assessment_Result (Assessment_ID, Module_ID, Student_ID, Result_ID, Total_Marks, Total_Marks_Obtained, Component_Details, Feedback)
 - The new table Assessment_Result was created so that the non-key attributes depend on all composite key.
- Resource (Resource_ID, Module_ID, Title, Type, Resource_Duration, Sequence_Number)
 - The resource table was modified as there was partial dependency and now all the attributes depend on the primary key. And of Module_ID it should be here because it

links the resource to a particular module but the resource attributes depend on the Resource_ID.

- Resource_Module_Student (Resource_ID, Module_ID, Student_ID)
- To reduce the redundancy as there is high chance of multiple students using the same resource within the same module, a junction table is created to manage many to many relationships.

The Final 2NF Tables are:

- Student (Student_ID, Student_Name, Phone_Number, Email, Date_of_Birth, Address, Enrollment_date, Program_ID, Program_Name, Program_Duration, Program_Start_Date, Program_End_Date)
- Module (Module_ID, Module_Name, Module_Duration, Total_Resources, Total_Assessments)
- Module_Student (Module_ID, Student_ID)
- Teacher (Teacher_ID, Teacher_Name, Email, Phone_Number, Department)
- Teacher_Announcement (Teacher_ID, Announcement_ID, Title, Announcement_Duration, Issued_date)
- Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
- Assessment_Result (Assessment_ID, Module_ID, Student_ID, Result_ID, Total_Marks, Total_Marks_Obtained, Component_Details, Feedback)

- Resource (Resource_ID, Module_ID, Title, Type, Resource_Duration, Sequence_Number)

3.4 3NF

In Third Normal Form (3NF), we separate those attributes which have transitive dependencies which means that the non-key attributes should depend only on the primary key not on other non-key attributes. Explanation of the process is given below:

- Student (Student_ID, Program_ID, Student_Name, Phone_Number, Email, Date_of_Birth, Address, Enrollment_date)
 - The student table from 2NF has transitive dependencies, so all program related attributes are separated in another table and now all the non-key attributes depend on primary key and the reason Program_ID is still in this table is to link the student to their particular program.
- Program (Program_ID, Program_Name, Program_Duration, Program_Start_Date, Program_End_Date)
 - The new program table is created from the student table. There are no transitive dependencies because all program related attributes depend on the primary key; Program_ID.
- Module (Module_ID, Module_Name, Module_Duration, Total_Resources, Total_Assessments)
 - Module table remains the same from the 2NF because all the non-key attributes depend on solely on primary key which means there is no partial dependency.
- Module_Student (Module_ID, Student_ID)

- Remains the same from the 2NF because it is already in 3NF.
- Teacher (Teacher_ID, Teacher_Name, Email, Phone_Number, Department)
 - Teacher table remains the same because there is no transitive dependencies.
- Teacher_Module (Teacher_ID, Module_ID)
 - This junction table is created because there is many to many relationships between the teacher and module.
- Teacher_Announcement (Teacher_ID, Announcement_ID)
 - This junction table is created because there is many to many relationships between the teacher and announcement.
- Announcement (Announcement_ID, Title, Announcement_Duration, Issued_date)
 - A new table announcement is created because in 2NF there was transitive dependency so in this new table all the non-key attributes depend only on the primary key.
- Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
 - Assessment table remains the same as there is no transitive dependency.
- Result (Result_ID, Total_Marks, Total_Marks_Obtained, Component_Details, Feedback)
 - The new table result is created in 3NF because there was transitive dependency on 2NF.
- Assessment_Result (Assessment_ID, Module_ID, Student_ID, Result_ID)

- This is a junction table which was created to manage the composite key.
- Resource (Resource_ID, Module_ID, Title, Type, Resource_Duration, Sequence_Number)
- Resource table remains the same as there is no transitive dependency.
- Resource_Module_Student (Resource_ID, Module_ID, Student_ID)
- Resource_Module_Student table remains the same because it created in 2NF to manage the many to many relationships between the students, module and resource.

The Final 3NF Tables are:

- Student (Student_ID, Program_ID, Student_Name, Phone_Number, Email, Date_of_Birth, Address, Enrollment_date)
- Program (Program_ID, Program_Name, Program_Duration, Program_Start_Date, Program_End_Date)
- Module (Module_ID, Module_Name, Module_Duration, Total_Resources, Total_Assessments)
- Module_Student (Module_ID, Student_ID)
- Teacher (Teacher_ID, Teacher_Name, Email, Phone_Number, Department)
- Teacher_Module (Teacher_ID, Module_ID)

- Teacher_Announcement (Teacher_ID, Announcement_ID)
- Announcement (Announcement_ID, Title, Announcement_Duration, Issued_date)
- Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
- Result (Result_ID, Total_Marks, Total_Marks_Obtained, Component_Details, Feedback)
- Assessment_Result (Assessment_ID, Module_ID, Student_ID, Result_ID)
- Resource (Resource_ID, Module_ID, Title, Type, Resource_Duration, Sequence_Number)
- Resource_Module_Student (Resource_ID, Module_ID, Student_ID)

4 Data Dictionary and Final ERD

4.1 Data Dictionary

Data Dictionaries were introduced in the 1960s as an early approach to database management. It provides information about the data and helps to define constraints and rules. Data dictionary ensures everyone understand what each piece of data represents. Data Dictionary serves as documentation for users and developers to understand database design and structure.

In the table below several key components are used and they are attribute name, data type which means it defines the type of data the attribute holds, size which specifies the maximum length and constraints defines rules or regulation for the attribute. Primary key uniquely identifies each record, foreign key links record in one table to another, UNIQUE means all values in column are distinct across all records in table. Not Null means the field must contain value and Null means absence of a value in column. The list of entities after Normalization with its attributes are given below:

Student

S.No.	Attribute Name	Data Type	Size	Constraint
1	Student_ID	Number	-	PRIMARY KEY
2	Program_ID	Number	-	FOREIGN KEY
3	Student_Name	Character	50	NOT NULL
4	Phone_Number	Character	15	UNIQUE
5	Email	Character	50	UNIQUE
6	Date_of_Birth	Date	-	NOT NULL
7	Address	Character	150	NOT NULL
8	Enrollment_Date	Date	-	NOT NULL

Table 4 Attributes of Student Entity

Program

S.No.	Attribute Name	Data Type	Size	Constraint
1	Program_ID	Number	-	PRMARY KEY
2	Program_Name	Character	100	NOT NULL
3	Program_Duration	Number	-	NOT NULL
4	Program_Start_Date	Date	-	NOT NULL
5	Program_End_Date	Date	-	NOT NULL

Table 5 Attributes of Program Entity

Module

S.No.	Attribute Name	Data Type	Size	Constraint
1	Module_ID	Number	-	PRMARY KEY
2	Module_Name	Character	100	NOT NULL
3	Module_Duration	Number	-	NOT NULL
4	Total_Resources	Number	-	NOT NULL
5	Total_Assessments	Number	-	NOT NULL

Table 6 Attributes of Module entity

Module_Student

S.No.	Attribute Name	Data Type	Size	Constraint	Composite Constraint
1	Module_ID	Number	-	FOREIGN KEY	Primary Key
2	Student_ID	Number	-	FOREIGN KEY	

Table 7 Attributes of Module_Student Entity

Teacher

S.No.	Attribute Name	Data Type	Size	Constraint
1	Teacher_ID	Number	-	PRIMARY KEY
2	Teacher_Name	Character	50	NOT NULL
3	Email	Character	50	UNIQUE
4	Phone_Number	Character	15	UNIQUE
5	Department	Character	50	-

*Table 8 Attributes of Teacher Entity***Teacher_Module**

S.No.	Attribute Name	Data Type	Size	Constraint	Composite Constraint
1	Teacher_ID	Number	-	FOREIGN KEY	Primary Key
2	Module_ID	Number	-	FOREIGN KEY	

Table 9 Attributes of Teacher_Module Entity

Teacher_Announcement

S.No.	Attribute Name	Data Type	Size	Constraint	Composite Constraint
1	Teacher_ID	Number	-	FOREIGN KEY	Primary Key
2	Announcement_ID	Number	-	FOREIGN KEY	

Table 10 Attributes of Teacher_Announcement Entity

Announcement

S.No.	Attribute Name	Data Type	Size	Constraint
1	Announcement_ID	Number	-	PRMARY KEY
2	Title	Character	255	NOT NULL
3	Announcement_Duration	Number	-	NOT NULL
4	Issued_Date	Date	-	NOT NULL

Table 11 Attributes of Announcement Entity

Assessment

S.No.	Attribute Name	Data Type	Size	Constraint
1	Assessment_ID	Number	-	PRMARY KEY
2	Assessment_Title	Character	100	NOT NULL
3	Deadline	Date	-	NOT NULL
4	Weightage	Number	5, 2	NOT NULL

Table 12 Attributes of Assessment Entity

Result

S.No.	Attribute Name	Data Type	Size	Constraint
1	Result_ID	Number	-	PRIMARY KEY
2	Total_Marks	Number	-	NOT NULL
3	Total_Marks_Obtained	Number	5, 2	NOT NULL
4	Component_Details	Character	100	NULL
5	Feedback	Character	255	NULL

Table 13 Attributes of Result Entity

Assessment_Result

S.No.	Attribute Name	Data Type	Size	Constraint	Composite Constraint
1	Assessment_ID	Number	-	FOREIGN KEY	Primary Key
2	Module_ID	Number	-	FOREIGN KEY	
3	Student_ID	Number	-	FOREIGN KEY	
4	Result_ID	Number	-	FOREIGN KEY	

Table 14 Attributes of Assessment_Result Entity

Resource

S.No.	Attribute Name	Data Type	Size	Constraint
1	Resource_ID	Number	-	PRIMARY KEY
2	Module_ID	Number	-	FOREIGN KEY
3	Title	Character	100	NOT NULL
4	Type	Character	50	NOT NULL
5	Resource_Duration	Number	5, 2	NOT NULL
6	Sequence_Number	Number	-	NOT NULL

*Table 15 Attributes of Resource Entity***Resource_Module_Student**

S.No.	Attribute Name	Data Type	Size	Constraint	Composite Constraint
1	Resource_ID	Number	-	FOREIGN KEY	Primary Key
2	Module_ID	Number	-	FOREIGN KEY	
3	Student_ID	Number	-	FOREIGN KEY	

Table 16 Attributes of Resource_Module_Student Entity

4.2 Final ERD

The final Entity Relationship Diagram (ERD) includes of 13 entities. They are given below:

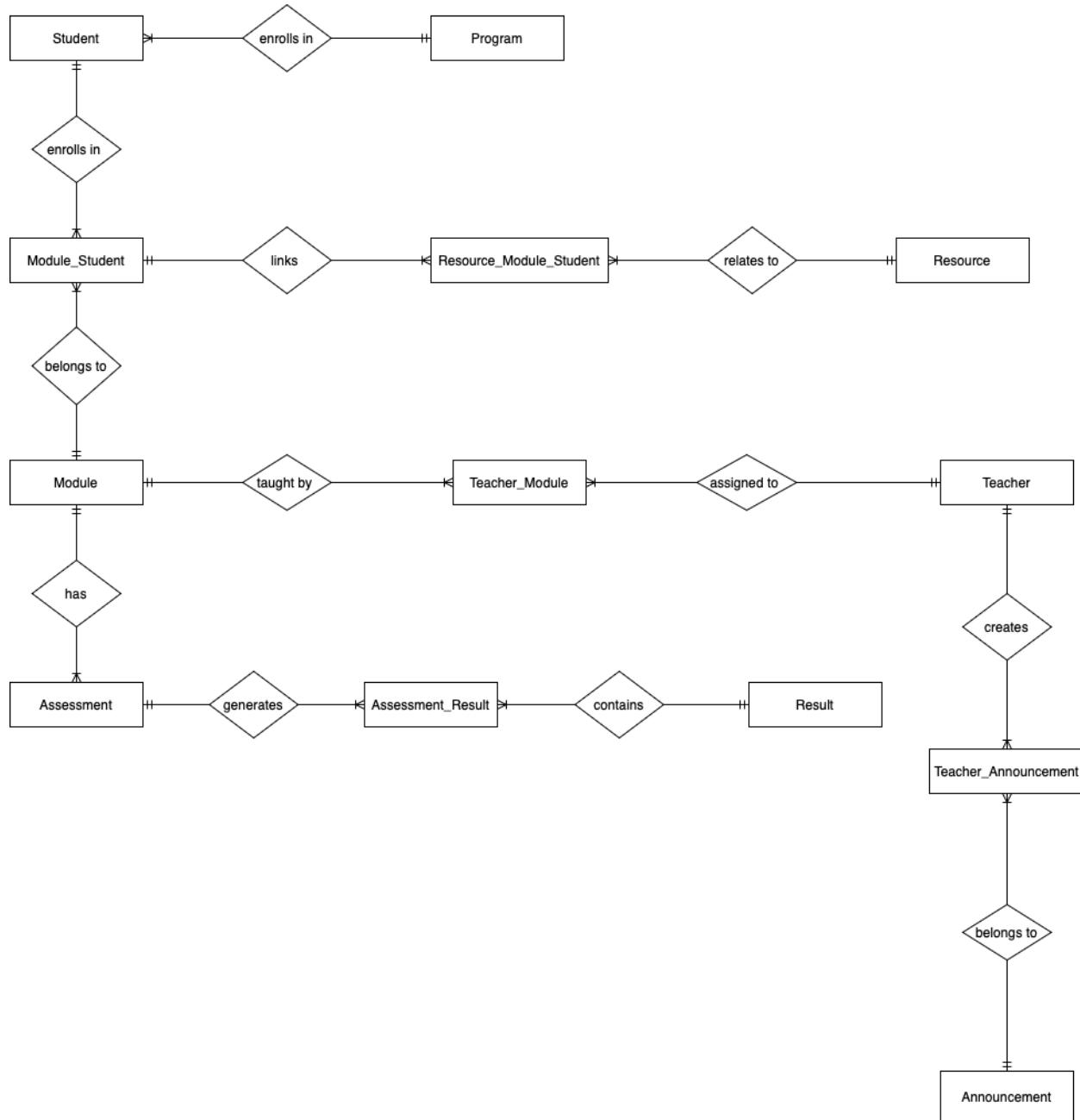


Figure 2 Final ERD

5 Implementation

5.1 Create User

The command to create user in SQL PLUS environment:

```
CREATE USER username IDENTIFIED BY londonmetid;  
GRANT CONNECT, RESOURCE TO username;  
CONNECT username/londonmetid
```

```
SQL*Plus: Release 11.2.0.2.0 Production on Sun Dec 29 12:59:02 2024  
Copyright (c) 1982, 2014, Oracle. All rights reserved.  
  
SQL> connect  
Enter user-name: system  
Enter password:  
Connected.  
SQL> CREATE USER Sadiksha IDENTIFIED BY 23049347;  
User created.
```

Figure 3 Applying Create user command

```
SQL> GRANT CONNECT,RESOURCE TO Sadiksha;  
Grant succeeded.  
  
SQL> connect  
Enter user-name: sadiksha  
Enter password:  
Connected.  
SQL>
```

Figure 4 Grant Connect command

5.2 Create Tables

5.2.1 Program Table

```
SQL> CREATE TABLE Program (
 2      Program_ID INT PRIMARY KEY,
 3      Program_Name VARCHAR(30) NOT NULL,
 4      Program_Duration INT NOT NULL,
 5      Program_Start_Date DATE NOT NULL,
 6      Program_End_Date DATE NOT NULL
 7  );
```

Figure 5 Create Program Table

5.2.2 Student Table

```
SQL> CREATE TABLE Student (
 2      Student_ID INT PRIMARY KEY,
 3      Program_ID INT,
 4      Student_Name VARCHAR(12) NOT NULL,
 5      Phone_Number VARCHAR(10) UNIQUE,
 6      Email VARCHAR(20) UNIQUE,
 7      Date_of_Birth DATE NOT NULL,
 8      Address VARCHAR(20) NOT NULL,
 9      Enrollment_Date DATE NOT NULL,
10      FOREIGN KEY (Program_ID) REFERENCES Program(Program_ID)
11  );
Table created.
```

Figure 6 Create Student Table

5.2.3 Module Table

```
SQL> CREATE TABLE Module (
 2   Module_ID INT PRIMARY KEY,
 3   Module_Name VARCHAR(30) NOT NULL,
 4   Module_Duration INT NOT NULL,
 5   Total_Resources INT NOT NULL,
 6   Total_Assessments INT NOT NULL
 7 );
```

Figure 7 Create Module Table

5.2.4 Module_Student Table

```
SQL> CREATE TABLE Module_Student (
 2 Module_ID INT,
 3 Student_ID INT,
 4 FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID),
 5 FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),
 6 PRIMARY KEY (Module_ID, Student_ID)
 7 );
```

Table created.

Figure 8 Create Module_Student Table

5.2.5 Teacher Table

```
SQL> CREATE TABLE Teacher (
 2   Teacher_ID INT PRIMARY KEY,
 3   Teacher_Name VARCHAR(20) NOT NULL,
 4   Email VARCHAR(25) UNIQUE,
 5   Phone_Number VARCHAR(15) UNIQUE,
 6   Department VARCHAR(25)
 7 );|
```

Figure 9 Create Teacher Table

5.2.6 Teacher_Module Table

```
SQL> CREATE TABLE Teacher_Module (
 2 Teacher_ID INT,
 3 Module_ID INT,
 4 FOREIGN KEY (Teacher_ID) REFERENCES Teacher(Teacher_ID),
 5 FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID),
 6 PRIMARY KEY (Teacher_ID, Module_ID)
 7 );
```

Table created.

Figure 10 Create Teacher_Module Table

5.2.7 Teacher_Announcement Table

```
SQL> CREATE TABLE Teacher_Announcement (
 2 Teacher_ID INT,
 3 Announcement_ID INT,
 4 PRIMARY KEY (Teacher_ID, Announcement_ID),
 5 FOREIGN KEY (Teacher_ID) REFERENCES Teacher(Teacher_ID),
 6 FOREIGN KEY (Announcement_ID) REFERENCES Announcement(Announcement_ID)
 7 );
```

Table created.

Figure 11 Create Teacher_Announcement Table

5.2.8 Announcement Table

```
SQL> CREATE TABLE Announcement (
 2 Announcement_ID INT PRIMARY KEY,
 3 Title VARCHAR(57) NOT NULL,
 4 Announcement_Duration INT NOT NULL,
 5 Issued_Date DATE NOT NULL
 6 );
```

Figure 12 Create Announcement Table

5.2.9 Assessment Table

```
SQL> CREATE TABLE Assessment (
 2      Assessment_ID INT PRIMARY KEY,
 3      Assessment_Title VARCHAR(30) NOT NULL,
 4      Deadline DATE NOT NULL,
 5      Weightage DECIMAL(5, 2) NOT NULL
 6  );|
```

Figure 13 Create Assessment Table

5.2.10 Result table

```
SQL> CREATE TABLE Result (
 2      Result_ID INT PRIMARY KEY,
 3      Total_Marks INT NOT NULL,
 4      Total_Marks_Obtained DECIMAL(5, 2) NOT NULL,
 5      Component_Details VARCHAR(60) NULL,
 6      Feedback VARCHAR(60) NULL
 7  );
Table created.
```

Figure 14 Create Result Table

5.2.11 Assessment_Result Table

```
SQL> CREATE TABLE Assessment_Result (
 2  Assessment_ID INT,
 3  Module_ID INT,
 4  Student_ID INT,
 5  Result_ID INT,
 6  FOREIGN KEY (Assessment_ID) REFERENCES Assessment(Assessment_ID),
 7  FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID),
 8  FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),
 9  FOREIGN KEY (Result_ID) REFERENCES Result(Result_ID),
10  PRIMARY KEY (Assessment_ID, Module_ID, Student_ID)
11  );
Table created.
```

Figure 15 Create Assessment_Result Table

5.2.12 Resource Table

```
SQL> CREATE TABLE "Resource" (
  2      Resource_ID INT PRIMARY KEY,
  3      Module_ID INT,
  4      Title VARCHAR(50) NOT NULL,
  5      Type VARCHAR(20) NOT NULL,
  6      Resource_Duration DECIMAL(5, 2) NOT NULL,
  7      Sequence_number INT NOT NULL,
  8      FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID)
  9  );
```

Table created.

Figure 16 Create Resource Table

5.2.13 Resource_Module_Student Table

```
SQL> CREATE TABLE Resource_Module_Student (
  2      Module_ID      INT,
  3      Student_ID     INT,
  4      Resource_ID    INT,
  5      PRIMARY KEY (Module_ID, Student_ID, Resource_ID),
  6      FOREIGN KEY (Module_ID) REFERENCES Module(Module_ID),
  7      FOREIGN KEY (Student_ID) REFERENCES Student(Student_ID),
  8      FOREIGN KEY (Resource_ID) REFERENCES "Resource"(Resource_ID)
  9  );
```

Table created.

Figure 17 Create Resource_Module_Student Table

5.3 Adding rows to Tables

5.3.1 INSERT of Program Table

```

SQL> INSERT INTO Program (Program_ID, Program_Name, Program_Duration, Program_Start_Date, Program_End_Date)
  2  VALUES (55052, 'Computing', 36, TO_DATE('2020-01-01', 'YYYY-MM-DD'), TO_DATE('2023-01-15', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Program (Program_ID, Program_Name, Program_Duration, Program_Start_Date, Program_End_Date)
  2  VALUES (55021, 'Networking', 36, TO_DATE('2020-03-01', 'YYYY-MM-DD'), TO_DATE('2023-03-15', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Program (Program_ID, Program_Name, Program_Duration, Program_Start_Date, Program_End_Date)
  2  VALUES (55050, 'Multimedia', 24, TO_DATE('2021-03-01', 'YYYY-MM-DD'), TO_DATE('2023-05-05', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Program (Program_ID, Program_Name, Program_Duration, Program_Start_Date, Program_End_Date)
  2  VALUES (55555, 'Artificial Intelligence', 36, TO_DATE('2021-10-10', 'YYYY-MM-DD'), TO_DATE('2024-12-01', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Program (Program_ID, Program_Name, Program_Duration, Program_Start_Date, Program_End_Date)
  2  VALUES (58900, 'Cybersecurity', 24, TO_DATE('2020-12-10', 'YYYY-MM-DD'), TO_DATE('2022-12-31', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Program (Program_ID, Program_Name, Program_Duration, Program_Start_Date, Program_End_Date)
  2  VALUES (55670, 'Software Engineering', 36, TO_DATE('2020-06-19', 'YYYY-MM-DD'), TO_DATE('2023-07-10', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Program (Program_ID, Program_Name, Program_Duration, Program_Start_Date, Program_End_Date)
  2  VALUES (55890, 'Data Science', 36, TO_DATE('2022-02-15', 'YYYY-MM-DD'), TO_DATE('2025-03-20', 'YYYY-MM-DD'));
1 row created.

```

Activate Windows
Go to Settings to activate Windows

Figure 18 Insert values in Program Table

5.3.2 INSERT of Student Table

```

SQL> INSERT INTO Student (Student_ID, Program_ID, Student_Name, Phone_Number, Email, Date_of_Birth, Address, Enrollment_Date
  2  VALUES (123, 55052, 'Saira Wills', '9815556777', 'sairawills@gmail.com', TO_DATE('2000-01-10', 'YYYY-MM-DD'), 'Kathmandu, Nepal', TO_DATE('2020
-01-01', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Student (Student_ID, Program_ID, Student_Name, Phone_Number, Email, Date_of_Birth, Address, Enrollment_Date
  2  VALUES (456, 55052, 'David Jones', '9856565038', 'davidjons@gmail.com', TO_DATE('2000-04-23', 'YYYY-MM-DD'), 'Kathmandu, Nepal', TO_DATE('2020-
01-01', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Student (Student_ID, Program_ID, Student_Name, Phone_Number, Email, Date_of_Birth, Address, Enrollment_Date
  2  VALUES (789, 55021, 'Saubab Ali', '9746462434', 'alisaubab@icloud.com', TO_DATE('2002-04-01', 'YYYY-MM-DD'), 'Imadol, Lalitpur', TO_DATE('2020-
03-01', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Student (Student_ID, Program_ID, Student_Name, Phone_Number, Email, Date_of_Birth, Address, Enrollment_Date
  2  VALUES (549, 55052, 'Nishh John', '9836363672', 'johnnish@gmail.com', TO_DATE('2001-12-01', 'YYYY-MM-DD'), 'Bhaisepati, Lalitpur', TO_DATE('202
-01-01', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Student (Student_ID, Program_ID, Student_Name, Phone_Number, Email, Date_of_Birth, Address, Enrollment_Date
  2  VALUES (813, 55050, 'Henry John', '9724243678', 'henry100@gmail.com', TO_DATE('2002-07-03', 'YYYY-MM-DD'), 'Thimi, Bhaktapur', TO_DATE('2021-03
-01', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Student (Student_ID, Program_ID, Student_Name, Phone_Number, Email, Date_of_Birth, Address, Enrollment_Date
  2  VALUES (629, 55052, 'Liam Rays', '9847476321', 'raysliam@gmail.com', TO_DATE('2003-02-12', 'YYYY-MM-DD'), 'Thamel, Kathmandu', TO_DATE('2020-01
-01', 'YYYY-MM-DD'));
1 row created.

```

Activate Windows
Go to Settings to activate Windows.

Figure 19 Insert values in Student Table

5.3.3 INSERT of Module Table

```
SQL> INSERT INTO Module (Module_ID, Module_Name, Module_Duration, Total_Resources, Total_Assessments)
  2  VALUES (111, 'Database', 12, 5, 3);
1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Name, Module_Duration, Total_Resources, Total_Assessments)
  2  VALUES (222, 'Software Engineering', 10, 5, 2);
1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Name, Module_Duration, Total_Resources, Total_Assessments)
  2  VALUES (333, 'Network Operating System', 10, 4, 3);
1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Name, Module_Duration, Total_Resources, Total_Assessments)
  2  VALUES (444, 'Programming', 12, 4, 3);
1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Name, Module_Duration, Total_Resources, Total_Assessments)
  2  VALUES (555, 'Cloud Computing', 9, 3, 1);
1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Name, Module_Duration, Total_Resources, Total_Assessments)
  2  VALUES (666, 'Data Analysis', 12, 5, 3);
1 row created.

SQL>
SQL> INSERT INTO Module (Module_ID, Module_Name, Module_Duration, Total_Resources, Total_Assessments)
  2  VALUES (777, 'Artificial Intelligence', 12, 4, 1);
1 row created.
```

Activate W
Go to Settings

Figure 20 Insert values in Module Table

5.3.4 INSERT of Module_Student Table

```
SQL> INSERT INTO Module_Student (Module_ID, Student_ID)
  2  VALUES (444, 123);
1 row created.

SQL>
SQL> INSERT INTO Module_Student (Module_ID, Student_ID)
  2  VALUES (555, 123);
1 row created.

SQL>
SQL> INSERT INTO Module_Student (Module_ID, Student_ID)
  2  VALUES (666, 123);
1 row created.

SQL>
SQL> INSERT INTO Module_Student (Module_ID, Student_ID)
  2  VALUES (777, 123);
1 row created.

SQL>
SQL> INSERT INTO Module_Student (Module_ID, Student_ID)
  2  VALUES (111, 456);
1 row created.

SQL>
SQL> INSERT INTO Module_Student (Module_ID, Student_ID)
  2  VALUES (222, 456);
1 row created.

SQL>
SQL> INSERT INTO Module_Student (Module_ID, Student_ID)
  2  VALUES (333, 456);
1 row created.
```

Figure 21 Insert values in Module_Student table

5.3.5 INSERT of Teacher Table

```
SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Email, Phone_Number, Department)
  2  VALUES (1, 'Ram Yadav', 'ramyadav@gmail.com', '9272738494', 'Computer Science');

1 row created.

SQL>
SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Email, Phone_Number, Department)
  2  VALUES (2, 'Riya Shrestha', 'riyastha@gmail.com', '9826263547', 'Networking');

1 row created.

SQL>
SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Email, Phone_Number, Department)
  2  VALUES (3, 'Sailesh Tuladhar', 'saileshldr@gmail.com', '9763534253', 'Multimedia');

1 row created.

SQL>
SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Email, Phone_Number, Department)
  2  VALUES (4, 'Arya Tripathi', 'tripathiary@gmail.com', '9872627849', 'Computer Science');

1 row created.

SQL>
SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Email, Phone_Number, Department)
  2  VALUES (5, 'Park Seon', 'seonpark@gmail.com', '9874838262', 'Artificial Intelligence');

1 row created.

SQL>
SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Email, Phone_Number, Department)
  2  VALUES (6, 'Ray Sharma', 'raysharma@icloud.com', '9837374838', 'Cybersecurity');

1 row created.

SQL>
SQL> INSERT INTO Teacher (Teacher_ID, Teacher_Name, Email, Phone_Number, Department)
  2  VALUES (7, 'Avyudaya Raj', 'avyuraj77@gmail.com', '9839309392', 'Cloud Computing');

1 row created.
```

Figure 22 Insert values in Teacher Table

5.3.6 INSERT of Teacher_Module Table

```
SQL> INSERT INTO Teacher_Module (Teacher_ID, Module_ID)
  2  VALUES (1, 111);
1 row created.

SQL>
SQL> INSERT INTO Teacher_Module (Teacher_ID, Module_ID)
  2  VALUES (2, 222);
1 row created.

SQL>
SQL> INSERT INTO Teacher_Module (Teacher_ID, Module_ID)
  2  VALUES (3, 333);
1 row created.

SQL>
SQL> INSERT INTO Teacher_Module (Teacher_ID, Module_ID)
  2  VALUES (4, 444);
1 row created.

SQL>
SQL> INSERT INTO Teacher_Module (Teacher_ID, Module_ID)
  2  VALUES (5, 555);
1 row created.

SQL>
SQL> INSERT INTO Teacher_Module (Teacher_ID, Module_ID)
  2  VALUES (6, 666);
1 row created.

SQL>
SQL> INSERT INTO Teacher_Module (Teacher_ID, Module_ID)
  2  VALUES (7, 777);
1 row created.
```

Figure 23 Insert values in Teacher_Module Table

5.3.7 INSERT of Announcement Table

```

SQL> INSERT INTO Announcement (Announcement_ID, Title, Announcement_Duration, Issued_Date)
  2  VALUES (1, 'Assignment Submission Deadline - Final Reminder', 5, TO_DATE('2024-05-03', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Announcement (Announcement_ID, Title, Announcement_Duration, Issued_Date)
  2  VALUES (2, 'Important Notice: No Class on May 10th', 1, TO_DATE('2024-05-05', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Announcement (Announcement_ID, Title, Announcement_Duration, Issued_Date)
  2  VALUES (3, 'Exam Date Postponed: New Schedule Announced', 7, TO_DATE('2024-05-10', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Announcement (Announcement_ID, Title, Announcement_Duration, Issued_Date)
  2  VALUES (4, 'Mid-Semester Break Announcement: Classes Resume June 1st', 10, TO_DATE('2024-05-25', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Announcement (Announcement_ID, Title, Announcement_Duration, Issued_Date)
  2  VALUES (5, 'Important: Final Exam Date Changed to July 15th', 14, TO_DATE('2024-06-01', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Announcement (Announcement_ID, Title, Announcement_Duration, Issued_Date)
  2  VALUES (6, 'Reminder: Last Day for Coursework Submission on June 20th', 3, TO_DATE('2024-06-15', 'YYYY-MM-DD'));
1 row created.

SQL>
SQL> INSERT INTO Announcement (Announcement_ID, Title, Announcement_Duration, Issued_Date)
  2  VALUES (7, 'Final Exam on August 1st: Review Sessions Available', 7, TO_DATE('2024-07-20', 'YYYY-MM-DD'));
1 row created.


```

Activate
Go to Sett

Figure 24 Insert values in Announcement Table

5.3.8 INSERT of Teacher_Announcement Table

```
SQL> INSERT INTO Teacher_Announcement (Teacher_ID, Announcement_ID)
  2  VALUES (1, 1);
1 row created.

SQL>
SQL> INSERT INTO Teacher_Announcement (Teacher_ID, Announcement_ID)
  2  VALUES (2, 2);
1 row created.

SQL>
SQL> INSERT INTO Teacher_Announcement (Teacher_ID, Announcement_ID)
  2  VALUES (3, 3);
1 row created.

SQL>
SQL> INSERT INTO Teacher_Announcement (Teacher_ID, Announcement_ID)
  2  VALUES (4, 4);
1 row created.

SQL>
SQL> INSERT INTO Teacher_Announcement (Teacher_ID, Announcement_ID)
  2  VALUES (5, 5);
1 row created.

SQL>
SQL> INSERT INTO Teacher_Announcement (Teacher_ID, Announcement_ID)
  2  VALUES (6, 6);
1 row created.

SQL>
SQL> INSERT INTO Teacher_Announcement (Teacher_ID, Announcement_ID)
  2  VALUES (7, 7);
1 row created.
```

Figure 25 Insert values in Teacher_Announcement Table

5.3.9 INSERT of Assessment Table

```
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES (1, 'MCQ', TO_DATE('2023-04-10', 'YYYY-MM-DD'), 10.00);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES (2, 'Practical Exam', TO_DATE('2023-05-20', 'YYYY-MM-DD'), 20.00);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES (3, 'Coursework', TO_DATE('2023-06-15', 'YYYY-MM-DD'), 40.00);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES (4, 'Unseen Exam', TO_DATE('2023-07-30', 'YYYY-MM-DD'), 30.00);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES (5, 'Quiz 1', TO_DATE('2023-04-15', 'YYYY-MM-DD'), 10.00);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES (6, 'Quiz 2', TO_DATE('2023-05-30', 'YYYY-MM-DD'), 10.00);
1 row created.

SQL>
SQL> INSERT INTO Assessment (Assessment_ID, Assessment_Title, Deadline, Weightage)
  2  VALUES (7, 'Quiz 3', TO_DATE('2023-06-20', 'YYYY-MM-DD'), 10.00);
1 row created.
```

Figure 26 Insert values in Assessment Table

5.3.10 INSERT of Result Table

```
SQL> INSERT INTO Result (Result_ID, Total_Marks, Total_Marks_Obtained, Component_Details, Feedback)
  2  VALUES (1, 100, 85.50, 'MCQ, Practical Exam, Coursework, Unseen Exam', 'Good job, keep it up!');
1 row created.

SQL>
SQL> INSERT INTO Result (Result_ID, Total_Marks, Total_Marks_Obtained, Component_Details, Feedback)
  2  VALUES (2, 100, 92.00, 'MCQ, Practical Exam, Coursework, Unseen Exam', 'Great work! Focus on quiz as well.');
1 row created.

SQL>
SQL> INSERT INTO Result (Result_ID, Total_Marks, Total_Marks_Obtained, Component_Details, Feedback)
  2  VALUES (3, 100, 76.80, 'MCQ, Practical Exam, Coursework, Unseen Exam', 'Focus more on practicals next time.');
1 row created.

SQL>
SQL> INSERT INTO Result (Result_ID, Total_Marks, Total_Marks_Obtained, Component_Details, Feedback)
  2  VALUES (4, 100, 88.30, 'MCQ, Practical Exam, Coursework, Unseen Exam', 'Nice performance.');
1 row created.

SQL>
SQL> INSERT INTO Result (Result_ID, Total_Marks, Total_Marks_Obtained, Component_Details, Feedback)
  2  VALUES (5, 100, 37, 'MCQ, Practical Exam, Coursework, Unseen Exam', 'Fail. Please focus on all areas for improvement');
1 row created.

SQL>
SQL> INSERT INTO Result (Result_ID, Total_Marks, Total_Marks_Obtained, Component_Details, Feedback)
  2  VALUES (6, 100, 85.00, 'MCQ, Practical Exam, Coursework, Unseen Exam', 'Good job overall!');
1 row created.

SQL>
SQL> INSERT INTO Result (Result_ID, Total_Marks, Total_Marks_Obtained, Component_Details, Feedback)
  2  VALUES (7, 100, 90.50, 'MCQ, Practical Exam, Coursework, Unseen Exam', 'Excellent performance.');
1 row created.
```

Activate W
Go to Settings

Figure 27 Insert values in Result Table

5.3.11 INSERT of Assessment_Result Table

```
SQL> INSERT INTO ASSESSMENT_RESULT (ASSESSMENT_ID, MODULE_ID, STUDENT_ID, RESULT_ID)
  2  VALUES (4, 111, 123, 1);
1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (ASSESSMENT_ID, MODULE_ID, STUDENT_ID, RESULT_ID)
  2  VALUES (5, 111, 123, 1);
1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (ASSESSMENT_ID, MODULE_ID, STUDENT_ID, RESULT_ID)
  2  VALUES (6, 111, 123, 1);
1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (ASSESSMENT_ID, MODULE_ID, STUDENT_ID, RESULT_ID)
  2  VALUES (7, 111, 123, 1);
1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (ASSESSMENT_ID, MODULE_ID, STUDENT_ID, RESULT_ID)
  2  VALUES (1, 222, 456, 2);
1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (ASSESSMENT_ID, MODULE_ID, STUDENT_ID, RESULT_ID)
  2  VALUES (2, 222, 456, 2);
1 row created.

SQL>
SQL> INSERT INTO ASSESSMENT_RESULT (ASSESSMENT_ID, MODULE_ID, STUDENT_ID, RESULT_ID)
  2  VALUES (3, 222, 456, 2);
1 row created.
```

Figure 28 Insert values in Assessment_Result Table

5.3.12 INSERT of Resource Table

```
SQL> INSERT INTO "Resource" (Resource_ID, Module_ID, Title, Type, Resource_Duration, Sequence_Number)
  2  VALUES (1, 111, 'Introduction to Databases - Slides', 'Slides', 1.5, 1);
1 row created.

SQL>
SQL> INSERT INTO "Resource" (Resource_ID, Module_ID, Title, Type, Resource_Duration, Sequence_Number)
  2  VALUES (2, 111, 'Basic SQL Queries - Video Tutorial', 'Video', 2.0, 2);
1 row created.

SQL>
SQL> INSERT INTO "Resource" (Resource_ID, Module_ID, Title, Type, Resource_Duration, Sequence_Number)
  2  VALUES (3, 222, 'Software Engineering Principles - Slides', 'Slides', 2.5, 1);
1 row created.

SQL>
SQL> INSERT INTO "Resource" (Resource_ID, Module_ID, Title, Type, Resource_Duration, Sequence_Number)
  2  VALUES (4, 222, 'Software Development Life Cycle - Video', 'Video', 1.5, 2);
1 row created.

SQL>
SQL> INSERT INTO "Resource" (Resource_ID, Module_ID, Title, Type, Resource_Duration, Sequence_Number)
  2  VALUES (5, 333, 'Network Operating System Basics - Tutorial Slides', 'Slides', 2.0, 1);
1 row created.

SQL>
SQL> INSERT INTO "Resource" (Resource_ID, Module_ID, Title, Type, Resource_Duration, Sequence_Number)
  2  VALUES (6, 444, 'Introduction to Programming - Tutorial Video', 'Video', 2.0, 1);
1 row created.

SQL>
SQL> INSERT INTO "Resource" (Resource_ID, Module_ID, Title, Type, Resource_Duration, Sequence_Number)
  2  VALUES (7, 444, 'Best Coding Practices - Code Examples', 'Slides', 2.0, 2);
1 row created.
```

Figure 29 Insert values in Resource Table

5.3.13 INSERT of Resource_Module_Student Table

```
SQL> INSERT INTO Resource_Module_Student (Module_ID, Student_ID, Resource_ID)
  2  VALUES (222, 549, 4);
1 row created.

SQL>
SQL> INSERT INTO Resource_Module_Student (Module_ID, Student_ID, Resource_ID)
  2  VALUES (333, 549, 5);
1 row created.

SQL>
SQL> INSERT INTO Resource_Module_Student (Module_ID, Student_ID, Resource_ID)
  2  VALUES (444, 549, 6);
1 row created.

SQL>
SQL> INSERT INTO Resource_Module_Student (Module_ID, Student_ID, Resource_ID)
  2  VALUES (444, 549, 7);
1 row created.

SQL>
SQL> INSERT INTO Resource_Module_Student (Module_ID, Student_ID, Resource_ID)
  2  VALUES (111, 813, 1);
1 row created.

SQL>
SQL> INSERT INTO Resource_Module_Student (Module_ID, Student_ID, Resource_ID)
  2  VALUES (111, 813, 2);
1 row created.

SQL>
SQL> INSERT INTO Resource_Module_Student (Module_ID, Student_ID, Resource_ID)
  2  VALUES (222, 813, 3);
1 row created.
```

Figure 30 Insert values in Resource_Module_Student Table

5.4 Content of Tables

5.4.1 Program Table

```
SQL> SELECT * FROM Program;

PROGRAM_ID PROGRAM_NAME          PROGRAM_DURATION PROGRAM_S PROGRAM_E
-----  -----
      55052 Computing                  36 01-JAN-20 15-JAN-23
      55021 Networking                 36 01-MAR-20 15-MAR-23
      55050 Multimedia                24 01-MAR-21 05-MAY-23
      55555 Artificial Intelligence   36 10-OCT-21 01-DEC-24
      58900 Cybersecurity              24 10-DEC-20 31-DEC-22
      55670 Software Engineering     36 19-JUN-20 10-JUL-23
      55890 Data Science               36 15-FEB-22 20-MAR-25

7 rows selected.
```

Figure 31 Content of Program Table

5.4.2 Student Table

```
SQL> SELECT * FROM Student;

STUDENT_ID PROGRAM_ID STUDENT_NAME PHONE_NUMBER      EMAIL           ADDRESS        Enrollment_Date
-----  -----
       123      55052 Saira Wills  9815556777  sairawills@gmail.com Kathmandu, Nepal 01-JAN-20
       456      55052 David Jones  9856565038  davidjons@gmail.com Kathmandu, Nepal 01-JAN-20
       789      55021 Saabab Ali   9746462434  alisaabab@icloud.com Imadol, Lalitpur 01-MAR-20
       549      55052 Nish John    9836363672  johnnish@gmail.com Bhaisepati, Lalitpur 01-JAN-20
       813      55050 Henry John   9724243678  henry100@gmail.com Thimi, Bhaktapur 01-MAR-21
       629      55052 Liam Rays   9847476321  raysliam@gmail.com Thamel, Kathmandu 01-JAN-20
       401      55050 Yash Raj    9726274938  yashraj@gmail.com Lazimpat, Kathmandu 01-MAR-21

7 rows selected.
```

Figure 32 Content of Student Table

5.4.3 Module Table

```
SQL> SELECT * FROM Module;
-----  

MODULE_ID MODULE_NAME          MODULE_DURATION TOTAL_RESOURCES TOTAL_ASSESSMENTS  

-----  

111     Database                12              5                  3  

222     Software Engineering    10              5                  2  

333     Network Operating System 10              4                  3  

444     Programming             12              4                  3  

555     Cloud Computing          9               3                  1  

666     Data Analysis            12              5                  3  

777     Artificial Intelligence   12              4                  1  

  
7 rows selected.
```

Figure 33 Content of Module Table

5.4.4 Module_Student Table

```
SQL> SELECT * FROM Module_Student;
-----  

MODULE_ID STUDENT_ID  

-----  

111      123  

222      123  

333      123  

444      123  

555      123  

666      123  

777      123  

111      456  

222      456  

333      456  

444      456  

555      456  

666      456  

777      456  

111      789  

222      789  

333      789  

444      789  

555      789  

666      789  

777      789  

111      549  

222      549  

333      549  

444      549  

555      549  

666      549  

777      549  

111      813  

222      813  

333      813  

444      813  

555      813  

666      813  

777      813  

111      629
```

Figure 34 Content of Module_Student Table

5.4.5 Teacher Table

SQL> SELECT * FROM Teacher;				
TEACHER_ID	TEACHER_NAME	EMAIL	PHONE_NUMBER	DEPARTMENT
1	Ram Yadav	ramyadav@gmail.com	9272738494	Computer Science
2	Riya Shrestha	riyastha@gmail.com	9826263547	Networking
3	Sailesh Tuladhar	saileshldr@gmail.com	9763534253	Multimedia
4	Arya Tripathi	tripathiarya@gmail.com	9872627849	Computer Science
5	Park Seon	seonpark@gmail.com	9874838262	Artificial Intelligence
6	Ray Sharma	raysharma@icloud.com	9837374838	Cybersecurity
7	Avyudaya Raj	avyuraj77@gmail.com	9839309392	Cloud Computing

7 rows selected.

Figure 35 Content of Teacher Table

5.4.6 Teacher_Module

SQL> SELECT * FROM Teacher_Module;	
TEACHER_ID	MODULE_ID
1	111
2	222
3	333
4	444
5	555
6	666
7	777

7 rows selected.

SQL> |

Figure 36 Content of Teacher_Module Table

5.4.7 Teacher_Announcement Table

SQL> SELECT * FROM Teacher_Announcement;	
TEACHER_ID	ANNOUNCEMENT_ID
1	1
2	2
3	3
4	4
5	5
6	6
7	7

7 rows selected.

Figure 37 Content of Teacher_Announcement Table

5.4.8 Announcement Table

ANNOUNCEMENT_ID	TITLE	ANNOUNCEMENT_DURATION	ISSUED_DATE
1	Assignment Submission Deadline - Final Reminder	5	03-MAY-24
2	Important Notice: No Class on May 10th	1	05-MAY-24
3	Exam Date Postponed: New Schedule Announced	7	10-MAY-24
4	Mid-Semester Break Announcement: Classes Resume June 1st	10	25-MAY-24
5	Important: Final Exam Date Changed to July 15th	14	01-JUN-24
6	Reminder: Last Day for Coursework Submission on June 20th	3	15-JUN-24
7	Final Exam on August 1st: Review Sessions Available	7	20-JUL-24

Figure 38 Content of Announcement Table

5.4.9 Assessment Table

Figure 39 Content of Assessment Table

5.4.10 Result Table

RESULT_ID	TOTAL_MARKS	TOTAL_MARKS_OBTAINED	COMPONENT_DETAILS	FEEDBACK
1	100	85.5	MCQ, Practical Exam, Coursework, Unseen Exam	Good job, keep it up!
2	100	92	MCQ, Practical Exam, Coursework, Unseen Exam	Great work! Focus on quiz as well.
3	100	76.8	MCQ, Practical Exam, Coursework, Unseen Exam	Focus more on practical exams next time.
4	100	88.3	MCQ, Practical Exam, Coursework, Unseen Exam	Nice performance.
5	100	37	MCQ, Practical Exam, Coursework, Unseen Exam	Fail. Please focus on all areas for improvement.
6	100	85	MCQ, Practical Exam, Coursework, Unseen Exam	Good job overall!
7	100	90.5	MCQ, Practical Exam, Coursework, Unseen Exam	Excellent performance.

7 rows selected.

Figure 40 Content of Result Table

5.4.11 Assessment_Result Table

ASSESSMENT_ID	MODULE_ID	STUDENT_ID	RESULT_ID
1	111	123	1
2	111	123	1
3	111	123	1
4	111	123	1
5	111	123	1
6	111	123	1
7	111	123	1
1	222	456	2
2	222	456	2
3	222	456	2
4	222	456	2
5	222	456	2
6	222	456	2
7	222	456	2
1	333	789	3
2	333	789	3
3	333	789	3
4	333	789	3
5	333	789	3
6	333	789	3
7	333	789	3
1	444	549	4
2	444	549	4
3	444	549	4
4	444	549	4
5	444	549	4
6	444	549	4
7	444	549	4
1	777	813	7
2	777	813	7
3	777	813	7
4	777	813	7
5	777	813	7
6	777	813	7
7	777	813	7
1	555	629	5
2	555	629	5

Figure 41 Content of Assessment_Result Table

5.4.12 Resource Table

```
SQL> SELECT * FROM "Resource";
RESOURCE_ID MODULE_ID TITLE                                     TYPE          RESOURCE_DURATION SEQUENCE_NUMBER
-----  -----  -----
 1       111  Introduction to Databases - Slides      Slides        1.5            1
 2       111  Basic SQL Queries - Video Tutorial    Video         2               2
 3       222  Software Engineering Principles - Slides Slides        2.5            1
 4       222  Software Development Life Cycle - Video Video         1.5            2
 5       333  Network Operating System Basics - Tutorial Slides      2               1
 6       444  Introduction to Programming - Tutorial Video   Video         2               1
 7       444  Best Coding Practices - Code Examples     Slides        2               2
7 rows selected.
```

Figure 42 Content of Resource Table

5.4.13 Resource_Module_Student Table

```
SQL> SELECT * FROM Resource_Module_Student;
MODULE_ID STUDENT_ID RESOURCE_ID
-----  -----  -----
 111      123      1
 111      123      2
 111      456      1
 111      456      2
 111      549      1
 111      549      2
 111      789      1
 111      789      2
 111      813      1
 111      813      2
 222      123      3
 222      123      4
 222      456      3
 222      456      4
 222      549      3
 222      549      4
 222      789      3
 222      789      4
 222      813      3
 222      813      4
 333      123      5
 333      456      5
 333      549      5
 333      789      5
 333      813      5
 444      123      6
 444      123      7
 444      456      6
 444      456      7
 444      549      6
 444      549      7
 444      789      6
 444      789      7
 444      813      6
 444      813      7
35 rows selected.
```

Figure 43 Content of Resource_Module_Student Table

6 Database Query

6.1 Information Query

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

= In this Query, SELECT statement is used which retrieves data from database and specifies the column that will be included in the output. COUNT is used to count the numbers of rows/number of Student_ID entries in student table for each program. LEFT JOIN is used to include all the records from the left table. And GROUP BY is used to group the result by program to get the number of students for each program.

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

PROGRAM_NAME          TOTAL_STUDENTS
-----  -----
Data Science                  0
Networking                   1
Cybersecurity                 0
Artificial Intelligence       0
Computing                     4
Multimedia                    2
Software Engineering           0

7 rows selected.
```

Figure 44 Information Query Question no.1 output

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

= FROM means that the announcement is the main table that is being queried. JOIN is used to combine rows from multiple tables based on related column. WHERE is used to filter the rows according to specified criteria. TO_DATE transforms string values into date format for accurate comparison and ORDER_BY arranges result set based on specific column.

```
SQL> SELECT a.Announcement_ID, a.Title, a.Announcement_Duration, a.Issued_Date
  2  FROM Announcement a
  3  JOIN Teacher_Announcement ta ON a.Announcement_ID = ta.Announcement_ID
  4  JOIN Teacher_Module tm ON ta.Teacher_ID = tm.Teacher_ID
  5  WHERE tm.Module_ID = 111
  6  AND a.Issued_Date BETWEEN TO_DATE('2024-05-01', 'YYYY-MM-DD') AND TO_DATE('2024-05-28', 'YYYY-MM-DD')
  7  ORDER BY a.Issued_Date;
```

ANNOUNCEMENT_ID	TITLE	ANNOUNCEMENT_DURATION	ISSUED_DA
1	Assignment Submission Deadline - Final Reminder		5 03-MAY-24

Figure 45 Information Query Question no.2 output

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

= SELECT, COUNT, FROM, LEFT JOIN, WHERE, GROUP BY has been used just like from the question 1 and 2. The output of this query is the module name that starts with the letter ‘D’.

MODULE_NAME	TOTAL_RESOURCES
Data Analysis	0
Database	2

Figure 46 Information Query Question no.3 output

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

= In this query, the output is of the list of students who have not submitted assessment of module 777.

STUDENT_ID	STUDENT_NAME	PROGRAM_NAME
456	David Jones	Computing
401	Yash Raj	Multimedia
123	Saira Wills	Computing
789	Saubab Ali	Networking
549	Nishh John	Computing
629	Liam Rays	Computing

6 rows selected.

Figure 47 Information Query Question no. 4 output

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

= SELECT specifies the column that will be included in the output, the JOIN query combines data from the table and HAVING clause ensures that only teachers who are assigned to more than one module are included in the result. The query returns no result because based on the business rule, a teacher can only be assigned to one module at a time whereas a module can have multiple teachers.

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

no rows selected
```

Figure 48 Information Query Question no.5 output

6.2 Transaction Query

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

= Join tables joins the table to gather necessary data, Subquery is used in this that finds the latest deadline from the assessment table and ROWNUM is used to limit the result set to only one row. As the result shows, the latest deadline is of Database module.

```
SQL> SELECT m MODULE_NAME, a DEADLINE
  2  FROM ASSESSMENT a
  3  JOIN ASSESSMENT_RESULT ar ON a.ASSESSMENT_ID = ar.ASSESSMENT_ID
  4  JOIN MODULE m ON ar MODULE_ID = m MODULE_ID
  5  WHERE a DEADLINE =
  6    SELECT MAX(DEADLINE)
  7    FROM ASSESSMENT
  8  )
  9  AND ROWNUM = 1;
```

MODULE_NAME	DEADLINE
Database	30-JUL-23

Figure 49 Transaction Query Question no.1 output

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

= Here, Subquery is used to sum the total marks for the student across all their assessments. Joins connect the tables and ORDER BY TOTAL_SCORE DESC means the order results in descending order by the total score.

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

STUDENT_NAME TOTAL_SCORE
----- -----
David Jones          644
Henry John           633.5
Nishh John           618.1
```

Figure 50 Transaction Query Question no.2 output

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

= In this case, Count is used to count the number of assessment id assigned for the program and AVG function is used to calculate the average score for all students in the program.

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

PROGRAM_NAME	TOTAL_ASSESSMENTS	AVERAGE_SCORE
Networking	7	76.8
Computing	35	77.56
Multimedia	7	90.5

Figure 51 Transaction Query Question no.3 output

4. List the students who have scored above the average score in database module.

= Join tables joins multiple tables for important information, AVG function computes the average score of a student for assessments in the “Database” module, WHERE to filter data of only the module and GROUP BY to calculate average score for each student and as the output student name and the average score of the module prints out.

```
SQL> SELECT s.Student_Name,
  2          AVG(r.TOTAL_MARKS_OBTAINED) AS average_score
  3    FROM STUDENT s
  4   JOIN ASSESSMENT_RESULT ar ON s.Student_ID = ar.Student_ID
  5   JOIN MODULE m ON ar.MODULE_ID = m.MODULE_ID
  6   JOIN RESULT r ON ar.RESULT_ID = r.RESULT_ID
  7 WHERE m.Module_Name = 'Database'
  8 GROUP BY s.Student_Name;

STUDENT_NAME AVERAGE_SCORE
----- -----
Saira Wills      85.5
```

Figure 52 Transaction Query Question no.4 output

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

= The SUM adds up the total marks obtained by each student and CASE is used to categorizes students as pass or fail based on the total marks and the condition given in question and GROUP BY is used to group the output by student, module, assessment. Lastly, ORDER BY sorts the result set by the name.

```
SQL> SELECT s.STUDENT_NAME,
  2      m.MODULE_NAME,
  3      a.ASSESSMENT_TITLE,
  4      SUM(r.TOTAL_MARKS_OBTAINED) AS TOTAL_MARKS,
  5      CASE
  6          WHEN SUM(r.TOTAL_MARKS_OBTAINED) < 40 THEN 'Fail'
  7          ELSE 'Pass'
  8      END AS REMARKS
 9  FROM STUDENT s
10 JOIN ASSESSMENT_RESULT ar ON s.STUDENT_ID = ar.STUDENT_ID
11 JOIN RESULT r ON ar.RESULT_ID = r.RESULT_ID
12 JOIN ASSESSMENT a ON ar.ASSESSMENT_ID = a.ASSESSMENT_ID
13 JOIN MODULE_STUDENT ms ON s.STUDENT_ID = ms.STUDENT_ID
14 JOIN MODULE m ON ms.MODULE_ID = m.MODULE_ID
15 WHERE m.MODULE_NAME = 'Database'
16     AND a.ASSESSMENT_TITLE = 'Coursework'
17 GROUP BY s.STUDENT_NAME, m.MODULE_NAME, a.ASSESSMENT_TITLE
18 ORDER BY s.STUDENT_NAME;
```

STUDENT_NAME	MODULE_NAME	ASSESSMENT_TITLE	TOTAL_MARKS	REMA
David Jones	Database	Coursework	92	Pass
Henry John	Database	Coursework	90.5	Pass
Liam Rays	Database	Coursework	37	Fail
Nishhh John	Database	Coursework	88.3	Pass
Saira Wills	Database	Coursework	85.5	Pass
Saubab Ali	Database	Coursework	76.8	Pass

6 rows selected.

Figure 53 Transaction Query Question no.5 output

7 Critical Evaluation

The evaluation reviews the Database module, focusing on its practical applications and its relationship to other subject areas. The module covers key topics such as Database and Database management system, Data analysis and Modeling, Database Models, Introduction to Database languages and SQL, Relational Database Theory, Relational Database Languages and Relational algebra. Each of these topics plays a crucial role in understanding databases and is highly applicable in today's tech industry.

The purpose of this module is to provide students practical applications of databases. It equips students with necessary skills to design, implement and manage databases effectively with problem solving skills. Through the hands-on practical application and theoretical learning, students can have experience related to it and use it in real-world work area. Talking about the practical application of the topics covered in the module, Database and Database Management System (DBMS) are extremely useful for handling large amounts of data in an organized and structured way. They are highly effective in industries like Finance, Educational centers, E-commerce, Healthcare and many more.

Structure language Query (SQL) is the primary language used to interact with the databases. With SQL, we can easily retrieve, store, update and delete data. It helps to generate the desired output and manage databases effectively. Additionally, Relational Database theory, languages and Relational algebra are key for preventing redundancy, designing effective database tables and provides understanding the relationships between them. These principles ensure that data is stored efficiently and avoid duplication.

The Database module is connected to various other subjects. Let's discuss the first one: Software Engineering. In software development, databases are crucial for storing and retrieving data. The knowledge gained from the Database module helps in designing efficient database management system that supports applications. For example: Relational Database theory supports software

engineering when building scalable applications. The skills learned in the database module such as using Structure language Query (SQL) to query data and understanding how data is structured, are essential for the Data science and Analytics, making the module highly relevant to these fields. For Data Analytics, it is very useful to use database queries. For example: He / She can generate sales report with the help of SQL. Additionally, it plays an important role in Business subject areas as it helps in decision-making and managing data. The Database module is also connected to the Computer Networking, as understanding how data stored and accessed in databases is foundational when studying Computer Networking. In this way, the Database module links to various other subjects.

The E-Classroom Platform coursework focuses on designing and implementing a database solution for education institution. Through this coursework, students gain hands-on experience in database design and deep understanding of how to structure and manipulate data effectively. Students are tasked with creating a detailed business rule, normalizing the data to 3NF and using SQL commands to create and manage tables. These tasks are highly relevant to real-world scenarios. This coursework helps to the development of critical thinking and problem-solving skills. They need to work according to the business rules and design database according to it. The implementation of the coursework where students create tables and manipulate data using SQL commands, it helps to solidify these concepts and enhance their technical skills. Throughout the coursework, students apply theoretical knowledge to practical tasks, preparing them for the real-world database management system.

In conclusion, The E-Classroom platform coursework serves a crucial learning experience for students. It strengthens and enhances both their theoretical knowledge and practical skills, offering a comprehensive understanding of Database design and management. The skills gained throughout this coursework are directly applicable to real-world industries, providing students with the necessary skills to handle database challenges in professional environment.

8 Dump File Creation and Drop Queries

8.1 Dump file command execution

Dump file is mainly used for backup and restoration. It is also referred as .dmp file. Steps to create a dump file are:

- Step 1: Open Windows command line and go to the path where the dump file should be created.
- Step 2: Use the command {exp username/password file = filename .dmp}

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22631.2861]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>c:
C:\Windows\System32>cd c:\dumpfile
c:\dumpfile>exp sadiksha/23049347 file = dumpfile.dmp

Export: Release 11.2.0.2.0 - Production on Wed Jan 22 17:42:49 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 SADIKSHA
. exporting PUBLIC type synonyms
. exporting private type synonyms
. exporting object type definitions for user SADIKSHA
About to export SADIKSHA's objects ...
. exporting database links
. exporting sequence numbers
. exporting cluster definitions
. about to export SADIKSHA's tables via Conventional Path ...
. . exporting table          ANNOUNCEMENT      7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table          ASSESSMENT       7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table          ASSESSMENT_RESULT 42 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table          MODULE           7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table          MODULE_STUDENT    49 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table          PROGRAM          7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table          RESOURCE_MODULE_STUDENT 35 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table          RESULT           7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table          Resource          7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table          STUDENT          7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table          TEACHER          7 rows exported
EXP-00091: Exporting questionable statistics.
```

Figure 54 Dump file execution command

```
c:\ Administrator: Command Prompt
EXP-00091: Exporting questionable statistics.
. . exporting table ASSESSMENT 7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table ASSESSMENT_RESULT 42 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table MODULE 7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table MODULE_STUDENT 49 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table PROGRAM 7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table RESOURCE_MODULE_STUDENT 35 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table RESULT 7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table Resource 7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table STUDENT 7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table TEACHER 7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table TEACHER_ANNOUNCEMENT 7 rows exported
EXP-00091: Exporting questionable statistics.
. . exporting table TEACHER_MODULE 7 rows exported
EXP-00091: Exporting questionable statistics.
. exporting synonyms
. exporting views
. exporting stored procedures
. exporting operators
. exporting referential integrity constraints
. exporting triggers
. exporting indextypes
. exporting bitmap, functional and extensible indexes
. exporting posttables actions
. exporting materialized views
. exporting snapshot logs
. exporting job queues
. exporting refresh groups and children
. exporting dimensions
. exporting post-schema procedural objects and actions
. exporting statistics
Export terminated successfully with warnings.

c:\dumpfile>
```

Figure 55 Dump file execution command

8.2 Drop Table Query

The command to delete a table is by using **DROP TABLE tablename CASCADE CONSTRAINTS;**. The DROP TABLE command deletes the specified table from the database. If there are no foreign key constraints, the table will be deleted successfully. However, if the table has foreign key referencing it from other tables, then an error will occur. Using the DROP TABLE command with the CASCADE CONSTRAINTS ensures that the table is deleted along with the foreign key constraints that depend on it. This clause automatically deletes all the associated constraints, which enables the table to be dropped without errors caused by the dependent relationships.

8.2.1 Drop Program Table

```
SQL> DROP TABLE Program CASCADE CONSTRAINTS;  
Table dropped.
```

Figure 56 Drop Query of Program Table

8.2.2 Drop Student Table

```
SQL> DROP TABLE Student CASCADE CONSTRAINTS;  
Table dropped.
```

Figure 57 Drop Query of Student Table

8.2.3 Drop Module Table

```
SQL> DROP TABLE Module CASCADE CONSTRAINTS;  
Table dropped.
```

Figure 58 Drop Query of Module Table]

8.2.4 Drop Module_Student Table

```
SQL> DROP TABLE Module_Student CASCADE CONSTRAINTS;  
Table dropped.
```

Figure 59 Drop Query of Module_Student Table

8.2.5 Drop Teacher Table

```
SQL> DROP TABLE Teacher CASCADE CONSTRAINTS;  
Table dropped.
```

Figure 60 Drop Table of Teacher Table

8.2.6 Drop Teacher_Module Table

```
SQL> DROP TABLE Teacher_Module CASCADE CONSTRAINTS;  
Table dropped.
```

Figure 61 Drop Table of Teacher_Module Table

8.2.7 Drop Teacher_Announcement Table

```
SQL> DROP TABLE Teacher_Announcement CASCADE CONSTRAINTS;  
Table dropped.
```

Figure 62 Drop Query of Teacher_Announcement Table

8.2.8 Drop Announcement Table

```
SQL> DROP TABLE Announcement CASCADE CONSTRAINTS;  
Table dropped.
```

Figure 63 Drop Query of Announcement Table

8.2.9 Drop Assessment Table

```
SQL> DROP TABLE Assessment CASCADE CONSTRAINTS;  
Table dropped.
```

Figure 64 Drop Query of Assessment Table

8.2.10 Drop Result Table

```
SQL> DROP TABLE Result CASCADE CONSTRAINTS;  
Table dropped.
```

Figure 65 Drop Query of Result Table

8.2.11 Drop Assessment_Result Table

```
SQL> DROP TABLE Assessment_Result CASCADE CONSTRAINTS;  
Table dropped.
```

Figure 66 Drop Query of Assessment_Result Table

8.2.12 Drop Resource Table

```
SQL> DROP TABLE "Resource" CASCADE CONSTRAINTS;  
Table dropped.
```

Figure 67 Drop Query of Resource Table

8.2.13 Drop Resource_Module_Student Table

```
SQL> DROP TABLE Resource_Module_Student CASCADE CONSTRAINTS;  
Table dropped.
```

Figure 68 Drop Query of Resource_Module_Student Table

Bibliography

Islington College, 2024. *Islington College*. [Online]

Available at: <https://islington.edu.np/facilities>

[Accessed 30 December 2024].

Geeks for geeks, 2025. *Normal Forms in DBMS*. [Online]

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

[Accessed 21 January 2025].