

23050262 Sujal Parajuli (Database) 1.docx



Islington College, Nepal

Document Details

Submission ID

trn:oid:::3618:79887232

Submission Date

Jan 23, 2025, 1:09 AM GMT+5:45

Download Date

Jan 23, 2025, 1:15 AM GMT+5:45

23050262 Sujal Parajuli (Database) 1.docx

File Size

32.3 KB

69 Pages

4,027 Words

26,447 Characters





34% Overall Similarity

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

Match Groups

124Not Cited or Quoted 32%

Matches with neither in-text citation nor quotation marks

9 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

0% 📕 Publications

30% L 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

124Not Cited or Quoted 32%

Matches with neither in-text citation nor quotation marks

9 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

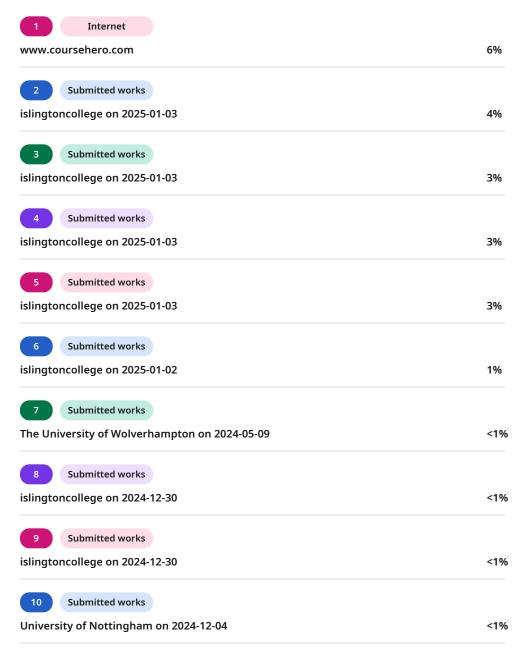
11% Internet sources

0% 🔳 Publications

30% L Submitted works (Student Papers)

Top Sources

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







11 Submitted works	
islingtoncollege on 2024-12-30	<1%
12 Submitted works	
islingtoncollege on 2025-01-03	<1%
13 Submitted works	
University of Northampton on 2024-09-01	<1%
14 Submitted works	
University of Wollongong on 2023-03-26	<1%
15 Submitted works	
islingtoncollege on 2025-01-02	<1%
16 Submitted works	
Suleyman Demirel University, Kazakhstan on 2023-12-01	<1%
17 Submitted works	
University of Sunderland on 2021-05-26	<1%
18 Submitted works	
Colorado Technical University Online on 2009-11-08	<1%
19 Submitted works Pathfinder Enterprises on 2023-04-27	<1%
ratifilities Efficiency 1365 of 2023-04-27	~170
20 Submitted works	
Kaplan International Colleges on 2023-04-07	<1%
21 Submitted works	
University of Central Lancashire on 2024-04-18	<1%
22 Submitted works Lincoln Institute of Higher Education Ltd on 2024-09-22	<1%
Lincom Institute of Higher Education Eta on 2024-05-22	~1%
23 Submitted works	
Majan College on 2012-11-30	<1%
24 Submitted works	
University Of Tasmania on 2017-08-14	<1%





25 Submitted works	
University of Greenwich on 2024-08-18	<1%
26 Submitted works	
Kaplan International Colleges on 2024-06-24	<1%
27 Submitted works	
Sim University on 2023-03-05	<1%
28 Submitted works	
University Of Tasmania on 2024-05-24	<1%
29 Submitted works	
University of East London on 2016-04-26	<1%
30 Submitted works	
University of Huddersfield on 2013-04-15	<1%
31 Submitted works	
University of Huddersfield on 2013-04-15	<1%
32 Submitted works	
University of Newcastle on 2011-05-22	<1%
33 Submitted works	
Kaplan College on 2023-06-01	<1%
34 Submitted works	
University of Northampton on 2024-09-01	<1%
35 Submitted works	
Lincoln Institute of Higher Education Ltd on 2024-10-04	<1%
36 Submitted works	
University of Ulster on 2017-04-25	<1%
37 Submitted works	
University of Wolverhampton on 2024-12-31	<1%
38 Submitted works	
MAALOT Baltimore on 2013-07-14	<1%





39 Submitted works	
RDI Distance Learning on 2012-05-16	<1%
40 Submitted works	
Sekolah Ciiputra High School on 2013-02-28	<1%
41 Submitted works	
University of Strathclyde on 2023-03-30	<1%
42 Submitted works	
islingtoncollege on 2024-12-30	<1%
43 Submitted works	
Queen Mary and Westfield College on 2023-12-01	<1%
44 Submitted works	
University of Warwick on 2024-05-16	<1%
45 Submitted works	
University Of Tasmania on 2017-08-14	<1%
46 Submitted works	
University of Greenwich on 2024-04-24	<1%
47 Submitted works	
University of Huddersfield on 2013-04-15	<1%
48 Submitted works	
University of Wolverhampton on 2025-01-03	<1%





Introduction

St. Mary's College of Technology was established in 2010 as one of the leading colleges in teaching technology to young students. Their college aims to bring in skilled tech professionals by practically teaching them with modern teaching methodology. They believe in making quality education available to all students. Their college has excellent computer labs, a big library, and modern classrooms that help students learn better.

The college is also located on a beautiful campus, with facilities for all types of sports, including a basketball court and a gym. Here, one can enjoy both studies and sports. The college has grown successfully over the years with the help of its founder and principal, Ms. Mary. She has very recently proposed starting an ambitious project for launching the "E-Classroom Platform" online. Such a project will revolutionize the area of education by connecting students, tutors, and academic programs in an integrated digital environment for effectively managing purposes. As a database designer, my responsibility is to develop a robust database system that will efficiently support the operational needs of this platform. This system will ensure smooth tracking and management of critical data, including student information, program detail, etc. The E-Classroom setup will likely cover several academic programs. In which we're talking Bachelor of Science tracks like Computing, Networking, and Multimedia, etc. The database system is designed to efficiently allocate teachers to classes and ensure that each course is well supported with qualified instructors. Each class will have mechanisms for assessing the performance of the students through various tests and assignments. Details regarding these assessments will be elaborately linked to their



respective classes, providing essential information such as assessment ID, title, due date, and weightage.

Current Business Activities and operations:

Ms. Mary is trying to start an online platform that is an E-Classroom that will provide a virtual learning environment for a college and its individuals. It will manage students, teachers, and programs with courses such as BSc in Computing, Networking, and Multimedia. Every program will have modules, assessments, resources, and results to keep track of the progress of the students. Teachers will create modules, make announcements, and lead students through structured and efficient learning. It ensures that students go through the resources in order and helps them organize their work. It will make learning easier and more accessible for all concerned individuals.

Business Rule:

Business rules determine the way a database will arrange the data for presentation and process that data for accuracy and reliability. For this platform, they provide well-defined structures among students, programs, modules, assessments, and resources to allow for greater efficiency in tracking progress and assignment management to achieve a planned and proper end goal.

The E-Classroom Platform has established its own set of business rules to control the relationships between students, teachers, programs, and modules:

One student must enroll in exactly one program whereas one program can have multiple enrolled students.





One program has several compulsory modules that a student needs to complete.

One module can be part of multiple programs, and one program can have multiple modules.

One module can have multiple assessments, and each assessment belongs to only one module.

To complete a module, a student needs to finish all the tasks and assessments of that module.

One teacher can be assigned to teach multiple modules, and a single module can be taught by multiple teachers.

One teacher post multiple announcement in his/her assigned modules.

Once submitted, students cannot edit or delete the submission of their assessments.

- One module can have multiple resources, and multiple resources can be in a single
- module.



Resources in a module must be completed in a linear fashion and are only unlocked

once previous resources have been completed.

One Teacher can give multiple announcements in any module they have been assigned to teach.

One Teacher is assigned to grade the assessment of multiple students.

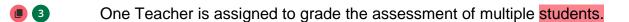
Assumption

For this case study, I have made some assumptions which have been integrated into the list of business rules which are given below:

One module can have multiple assessments, and each assessment belongs to only one module.

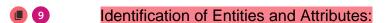


To complete a module, a student needs to finish all the tasks and assessments of that module.



One module can have multiple resources, and multiple resources can be in a single module.

Initial ERD



In the project, the entities are all the things which are an integral part of the system, and which must be stored and maintained in the database. The attributes are the characteristics or details of each entity.







Student Table:

S.No.

Attribute Name

Data Type

Size

Constraint

1

Student_ID

Number

10

Primary Key

2

Student_Name

Character

40

Not Null

3

DOB

Date

-

Not Null

4





Enrollment_Date

Date

_

Not Null

5

Std_Email

Character

30

Unique



Table 1 (Entity and Attribute table for Student)

Program Table:



S.No.

Attribute Name

Data Type

Size

Constraint

1

Program_ID

Number

10





		1.7
Dri	mar	, K 0
-11	IIIai \	/ Key
		, ,

2

Program_Name

Character

40

Not Null

3

P_Duration

Date

_

Not Null

4

P_Description

Character

100

Not Null



Table 2 (Entity and Attribute table for Program)

Module Table



S.No.

Attribute Name







Data Type

Size

Constraint

Module_ID

Number

10

Primary Key

Module_Name

Character

40

Not Null

M_Duration

Date

_

Not Null

M_Description

Character

100

Not Null





M_Credits

Number

10

No Null





Number

10

Not Null

Teacher_Name

Character

40

Not Null

Teach_Specialization

Character

30

Not Null

Teach_Email

Character





30

Unique



Number

10

Not Null

Asse_Description



40

Not Null

Asse_Weightage

Character

50

Not Null

Asse_Deadline

Date

-

Not Null





Asse_Status

Character

10



Not Null

Marks_Obtained

Number

Not Null

Grade_Obtained

Character

1

Not Null

Announcement_ID

Number

10

Not Null





Date_Posted

Date





Not Null

Ann_Title

Character

20

Not Null

Ann_Content

Character

100

Not Null





Number

10

Not Null

R_Duration

Date



Not Null

R_Title





Character

20

Not Null

R_Type



Character

30

Not Null

Sequence_Order

Character

30

Not Null

Table 3 (Entity and Attribute table for Module)



Initial Entity Relationship Diagram (E.R.D)

Figure 2 Initial ERD

Normalization:

It is a data organizing processing a database in such a manner that duplication





reduces and accuracy increases. The normalizing of databases makes them perform optimally by reducing lots of problems during adding, deleting, or updating data. In normalization, data is organized in tables, and the relationship between different tables is established with the help of certain rules called normal forms. These rules keep the database organized, efficient, and error-free (Geeksofgeeks, 2024).

7.1. Un-Normalized Form (UNF):

Here Data is stored in the form it is collected, and no effort is made to eliminate redundancy or guarantee atomicity.

It allows repeating groups and non-atomic attributes (attributes with multiple values or sets).

Organizing data for UNF:



Student: (Student_ID, Student_Name, DOB, Enrollment_Date, Std_Email, Program_ID, Program_Name, P_Duration, P_Description {Module_ID,

Module_Name, M_Duration, M_Description, M_Credits { Teacher_ID, Teacher_Name, Teach_Specialization, Teach_Email, { Announcement_ID, Date_Posted, Ann_Title, Ann_Content}},

Resource_ID, R_Description, R_Title, R_Type, Sequence_Order}, {Assessment_ID, Asse_Description, Asse_Weightage, Asse_Deadline, Asse_Status, Marks_Obtained, Grade_Obtained}})





7.2. First Normalization Form (1NF):

Rules for 1NF

No repeating values in a group

No repeating groups

UNF to 1NF:

Repeating groups (nested sets like {Student}, {Module}) were separated into different rows.

All attributes were atomic; for example, no {Resource_ID, Title} in one field.

• 18 First Normal Form Tables:

Student Table for 1NF:

Student-1: (Student_ID, Student_Name, DOB, Enrollment_Date, DOB, Std_Email,

Program_ID, Program_Name, P_Duration, P_Description)

Student and Module Table for 1NF:

Module_Student1: (Module_ID, Module_Name, M_Duration, M_Credits,

M_Description, Student_ID*)

Student, Module and Teacher Table for 1NF:

Student_Module_Teacher1: (Teacher_ID, Teacher_Name, Teach_Specialization,

Teach_Email, Student_ID*, Module_ID*)

Student, Teacher and Announcement Table for 1NF:

Student_Teacher_Announcement1: (Announcement_ID, Date_Posted, Ann_Title,





44

Ann_Content, Student_ID*, Module_ID*, Teacher_ID*)

Resource, Module and Student Table for 1NF:

Resource_Module_Student1: (Resource_ID, R_Duration, R_Title, R_Type,

Sequence_Order, Student_ID*, Module_ID*)

Student, Module and Assessment Table for 1NF:

Assessment_Module_Student1: (Assessment_ID, Asse_Description, Asse_Weightage,

Asse_Deadline, Asse_Status, Marks_Obtained, Grade_Obtained, Student_ID*,

Module_ID*)

7.3. Second Normalization Form (2NF):

Rules for 2NF:

If it is necessary, we need to split the composite keys into separate tables.

All non-keys should depend on all primary keys.

Transiting from 1NF to 2NF:

Identifying and removing many-to-many relationships among entities by creating linking entities or adding a bridging entity if needed.

Attributes depending on the composite key partially are transferred to other tables.



Process for Conversion of Second Normal Form:



18	Student Table for 2NF:				
	Student2: (Student	ID, Student_Name, D.O.B, Enrollment_Date, Std_Email,			
	Program_ID, Program_Name, P_Description, P_Duration)				
6	Note: 1NF table of	Student is already in 2NF because it has only one primary key i.e.			
	Student_ID and it d	oes not have any composite key.			
	Module_Student Table for 2NF:				
1 2	Module_ID	Module_Name (partially dependent on Module_ID)			
	Module_ID	M_Description (It has partial dependency with			
	Module_ID)				
1 3	Module_ID	M_Credit (It has partial dependency with Module_ID)			
	Module_ID, Studen	t_ID X			
	Student_ID	X			
	Module2: (Module_	ID, Module_Name, M_Duration, M_Credits, M_Description)			
1 3	Module_Student2:	(Module_ID, Student_ID)			
	Student_Module_T	eacher Table for 2NF:			
38	Teacher_ID	Teacher_Name (partially dependent on Teacher_ID)			
	Teacher_ID	Specialization (It has partial dependency with			
	Teacher_ID)				
1 3	Teacher_ID	Teach_Email (It has partial dependency with Teacher_ID)			
	Module_ID, Teach	er_ <mark>ID</mark> X			
	Module_ID	X			
	Teacher2: (Teache	r_ID, Teacher_Name, Teach_Specialization, Teach_Email)			
4 0	Student_Module_T	eacher2: (Teacher_ID, Module_ID, Student_ID)			



Student_Module_Announcement Table for 2NF: Announcement_ID Ann_Title (partially dependent on Announcement_ID) Date_Posted (Partial dependency with Announcement_ID Announcement_ID) Announcement_ID Ann_Content (Partial dependency with Announcement_ID) Student ID Χ Module ID Χ Teacher_ID, Module_ID Χ Teacher_ID, Module_ID, Announcement_ID Χ Announcement2: (Announcement_ID, Ann_Title, Date_Posted, Ann_Content) Student_Teacher_Announcement2: (Teacher_ID, Announcement _ID, Module_ID, Student_ID) Resource_Module_Student Table for 2NF: R_Duration (partially dependent on Resource _ID) Resource_ID Resource_ID R_Title (It has partial dependency with Resource ID) Resource_ID R_Type (It has partial dependency with Resource _ID) Resource_ID Sequence_Order (It has partial dependency with Resource



ID)



3	Student_ID	X	
	Module_ID	X	
	Resource_ID, Student	_ID, Module_ID X	
	Resource2: (Resource	e_ID, R_Duration, R_Title, R	_ <mark>Type,</mark> Sequence_Order)
3	Student_Module_Reso	ource2: (Resource _ <mark>ID, Mod</mark>	ule_ID, Student_ID)
	Assessment_Module_	Student Table for 2NF:	
	Assessment_ID, Mod	ule_ <mark>ID,</mark> Student_ID	Grade_Obtained (Fully dependent)
3	Assessment_ID, Mod	ule_ID, Student_ID	Marks_Obtained (Fully dependent)
3	Assessment_ID, Mod	ule_ID, Student_ID	Asse_Status (Fully
	dependent)		
	Assessment_ID	Asse_Weightage (It ha	as partial dependency with
	Assessment _ID)		
	Assessment_ID	Asse_Description (It ha	as partial dependency with
	Assessment _ID)		
	Assessment_ID	Asse_Deadline (It has	partial dependency with
	Assessment _ID)		
1 3	Assessment_ID	Date_Published (It ha	s partial dependency with Result_ID)
	Module_ID, Student_I	D X	
	Student_ID	X	

Student_ID X

Module_ID X

Assessment2: (Assessment_ID, Asse_Description, Asse_Weightage, Asse_Deadline)

Assessment_Module_Student2: (Assessment_ID, Module_ID, Student_ID,



Asse_Status, Marks_Obtained, Grade_Obtained)

Final Second Normal Form Tables:

Student2: (Student_ID, Student_Name, D.O.B, Enrollment_Date, Std_Email,

Program_ID, Program_Name, P_Description, P_Duration)

Module2: (Module_ID, Module_Name, M_Duration, M_Credits, M_Description)

Module_Student2: (Module_ID, Student_ID)

Teacher2: (Teacher_ID, Teacher_Name, Teach_Specialization, Teach_Email)

Student_Module_Teacher2: (Teacher_ID, Module_ID, Student_ID)

Announcement2: (Announcement_ID, Ann_Title, Date_Posted, Ann_Content)





Student_Teacher_Announcement2: (Teacher_ID, Announcement _ID, Module_ID, Student_ID)

Resources2: (Resource_ID, R_Duration, R_Title, R_Type, Sequence_Order)

Resource_Module_Student 2: (Resource _ID, Module_ID, Student_ID)

Assessment2: (Assessment_ID, Asse_Description, Asse_Weightage, Asse_Deadline)

Assessment _Module_Student2: (Assessment_ID, Module_ID, Student_ID,

Asse_Status, Marks_Obtained, Grade_Obtained)

7.4. Third Normalization Form (3NF):

Rules for 3NF:

Removal of transitive dependency and non-key attributes must not depend upon other non-key attributes.





Ensure that each non-key attribute depends upon the primary key only (Chris, 2022).

Transformation from 2NF to 3NF:

Getting rid of transitive dependencies, like when Program_Name depends on

Student_ID indirectly through Program_ID.

Make distinct relations for dependencies or derived attributes.

Process for conversion of Third Normal Form:

Student Table for 3NF:

In the 2NF table of Student, Program_Name, P_Duration and P_Description depends directly on the Program_ID, which depends upon Student_ID:

Student_ID Program_ID Program_Name, P_Duration and P_Description

To solve this issue, the Student Table and the Program Table are separated into different tables. Therefore, in 3NF the structure will look like:

Student3: (Student_ID, Student_Name, D.O.B, Enrollment_Date, Std_Email, Program_ID*)

Program3: (Program_ID, P_Description, P_Duration, Program_Name)

Module Table for 3NF:

Transitivity dependency is not here because all non-key attributes depend directly upon Module_ID.

Module3: (Module_ID, Module_Name, M_Duration, M_Credits, M_Description)





Module_Student:

It is automatically in 3NF because non key attributes do not exist.

Module_Student3: (Student_ID, Module_ID)

Teacher:

There is no transitivity dependency because all non-key attributes depend directly upon Teacher_ID.

Teacher3: (Teacher_ID, Teacher_Name, Teach_Specialization, Teach_Email)

Student_Module_Teacher:

It is automatically in 3NF because non key attributes do not exist.

Student_Module_Teacher3: (Teacher_ID, Module_ID, Student_ID)

Announcement:

Transitivity dependency is not here because all non-key attributes depend directly upon Announcement_ID.

Announcement3: (Announcement_ID, Ann_Title, Date_Posted, Ann_Content)

Student_Teacher_Announcement:

It is automatically in 3NF because non key attributes do not exist.





Student_Teacher_Announcement3: (Announcement _ID, Module_ID, Student_ID, Teacher_ID)

Resources:

Transitivity dependency is not here because all non-key attributes depend directly upon Resource_ID.

Resource3: (Resource_ID, R_Duration, R_Title, R_Type, Sequence_Order)

Student_Module_Resource:

It is automatically in 3NF because non key attributes do not exist.

Student_Module_Resource3: (Resource_ID, Module_ID, Student_ID)

Assessment:

Transitivity dependency is not here because all non-key attributes depend directly upon Assessment_ID.

Assessment3: (Assessment_ID, Asse_Description, Asse_Weightage, Asse_Deadline)

Assessment_Module_Student:

It is automatically in 3NF because non key attributes do not exist.





Assessment_Module_Student3: (Assessment_ID, Module_ID, Student_ID, Asse_Status, Marks_Obtained, Grade_Obtained))

Final Third Normal Form Tables:

Student3: (Student_ID, Student_Name, D.O.B, Enrollment_Date, Std_Email, Program_ID*)

Program3: (Program_ID, P_Description, P_Duration, Program_Name)
Module3: (Module_ID, Module_Name, M_Duration, M_Credits, M_Description)

Module_Student3: (Student_ID, Module_ID)

- Modulos. (Modulo_12, Modulo_Namo, M_Buration, M_Broate, M_Boothpalon)
- Teacher3: (Teacher_ID, Teacher_Name, Teach_Specialization, Teach_Email)
- Student_Module_Teacher3: (Teacher_ID, Module_ID, Student_ID)
 Announcement3: (Announcement_ID, Ann_Title, Date_Posted, Ann_Content)
- Student_Teacher_Announcement3: (Announcement _ID, Module_ID, Student_ID, Teacher_ID)

Resources3: (Resource_ID, R_Duration, R_Title, R_Type)



Student_Module_Resource3: (Resource_ID, Module_ID, Student_ID)

Assessment3: (Assessment_ID, Asse_Description, Asse_Weightage, Asse_Deadline)

Assessment_Module_Student3: (Assessment_ID, Module_ID, Student_ID,

Asse_Status, Marks_Obtained, Grade_Obtained))





Final ERD:

Data Dictionary:







Student Table:

S.No.

Attribute Name

Data Type

Size

Constraint

1

Student_ID

Number

10

Primary Key

2

Student_Name

Character

40

Not Null

3

DOB

Date

-

Not Null

4





Enrollment_Date

Date

-

Not Null

5

Std_Email



Character

30

Unique

6

Program_ID

Number

10

Foreign Key

Table 4 Data Dictionary of Student

Program Table:

S.No.

Attribute Name

Data Type

Size

Constraint





1

Program_ID

Number

10

Primary Key

2

Program_Name

Character

40

Not Null

3

P_Duration



Date

Not Null

4

P_Description

Character

100

Not Null

Table 5 Data Dictionary of Table





Module Table:



S.No.

Attribute Name

Data Type

Size

Constraint

1

Module_ID

Number

10

Primary Key

2

Module_Name

Character

40

Not Null

3





M_Duration

Date



Not Null

M_Credits

Number

5

Not Null

5

M_Description

Character

100

Not Null

Table 6 Data Dictionary of Module





S.No.

Attribute Name

Data Type

Size

Constraint

Composite Constraint





Student_ID Number 10 Foreign Key Primary Key 2 Module_ID Number 10 Foreign Key Module_Student: Table 7 Data Dictionary of Module_Student Teacher Table: S.No. Attribute Name Data Type Size



Constraint



Teacher_ID

Number

10

Primary Key

2

Teacher_Name

Character

40

Not Null

3

Teach_Specialization

Character

50





4

Teach_Email

Character

30

Unique

Table 8 Data Dictionary of Teacher





S.No.

Attribute Name

Data Type

Size

Constraint

Composite Constraint





Teacher_ID

Number

10

Foreign Key

Primary

Key

2

Module_ID

Number

10

Foreign Key

3

Student_ID







Number

10

Foreign Key

Student_Module_Teacher:

Table 9 Data Dictionary of Student_Module_Teacher

Announcement Table:

S.No.

Attribute Name

Data Type

Size

Constraint

1

Announcement_ID

Number

10

Primary Key

2

Date_Posted

Date

_







Not Null



Ann_Title

Character

20

Not Null

4

Ann_Content

Character

100

Not Null

Table 10 Data Dictionary of Announcement

Student_Teacher_Announcement:





Attribute Name

Data Type

Size

Constraint

Composite Constraint



Announcement_ID





Number

10

Foreign Key

Primary

Key

2

Module_ID

Number

10

Foreign Key

3

Student_ID



Number

10

Foreign Key

Table 11 Data Dictionary of Student_Teacher_Announcement





Resources Table:

S.No.

Attribute Name

Data Type

Size

Constraint

1

Resource_ID

Number

10

Primary Key

2

R_Duration

Date

_



Not Null

3

R_Title

Character

20

Not Null

1

R_Type





Character

30

Not Null

5

Sequence_Order

Character

Not Null

30



Table 12 Data Dictionary of Resource

Student_Module_Resource:





Attribute Name

Data Type

Size

Constraint

Composite Constraint

1

Resource_ID

Number





10

Foreign Key

Primary

Key



Module_ID

Number

10

Foreign Key

3

Student_ID

Number

10





Foreign Key

Table 13 Data Dictionary of Student_Module_Resource





Ass	ess	me	nt	Tah	de:
പാാ	ರಾವ	1110	וונ	ıaı	ИC.

S.No.

Attribute Name

Data Type

Size

Constraint

1

Assessment_ID

Number

10

Primary Key

2

Asse_Description

Character

40

Not Null

3

Asse_Weightage

Number

3





Not Null



Table 14 Data Dictionary of Assessment

Assessment_Module_Student:

S.No.

Attribute Name

Data Type

Size

Constraint

Composite Constraint

1

Assessment_ID

Number

10

Foreign Key

5

Primary

Key

2





Module_ID

Number

10

Foreign Key

3

Student_ID

Number

10

Foreign Key

4

Asse_Status

Character

10

Not Null

5

Marks_Obtained

Number

3

Not Null





6

Grade_Obtained



Character

1

Not Null

Table 15 Data Dictionary of Assessment_Module_Student

Implementation for creating tables:



10.1. Creating Student Table:

Figure 4 Creating Student Table

10.2. Creating Program Table:





Figure 5 Creating Program Table





10.3. Creating Module Table:

Figure 6 Creating Module table

10.4. Creating Module_Student Table:

Figure 7 Creating Module_Student Table

10.5. Creating Teacher Table:





Figure	8	Creating	Teacher	Table
90.0	_	CICALITIES	i oaciici	

10.6. Creating Student_Module_Teacher Table:

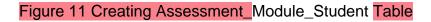
Figure 9 Creating Student_Teacher_Announcement Table

10.7. Creating Assessment Table:

Figure 10 Creating Assessment Table:

10.8. Creating Assessment_Module_Student Table:





10.9. Creating Announcement Table:

Figure 12 Creating Announcement Table:

10.10. Creating Student_Teacher_Announcement Table:





Figure 13 Creating Student_Teacher_Announcement Table

10.11. Creating Resources Table:

Figure 14 Creating Resources Table:

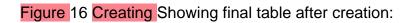
10.12. Creating Student_Module_Resource Table:

Figure 15 Creating Student_Module_Resource

10.13. Creating Showing final table after creation:







Implementation for inserting data in the tables:

11.1. Inserting Program data:

Figure 17Inserting Program data:



11.2. lr	nserting	Student	data:
----------	----------	---------	-------

Figure 18 Inserting Student data:

Updated Insertion:

11.3. Inserting Module data:

Figure 19 Inserting Module

11.4. Inserting Module_Student data:







11.5. Inserting Resources data:

Figure 21 Inserting Resources data:





Figure 22 Inserting Student_Module_Resource data:





11.7. Inserting Teacher data:

Figure 23 Inserting Teacher data:



11.8. Inserting Student_Module_Teacher data:

Figure 24 Inserting Student_Module_Teacher data:

11.9. Inserting Announcement data:







11.10. Inserting Student_Teacher_Announcement data:

Figure 26 Inserting Announcement_Module_Student data





11.11.	Inserting	Assessment	data:
--------	-----------	------------	-------

Figure 27 Inserting Assessment data:

11.12. Inserting Assessment_Module_Student data:

Figure 28 Inserting Assessment_Module_Student data:





Query regarding database:

Information Query:

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

Figure 29 Query One: listing students enrolled in a program and total program Entered Query:

Select P.Program_Name, Count(S.Student_ID) As Total

From Program P

Left Join Student S

On P.Program_ID = S.Program_ID

Group by P.Program_Name;





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

Figure 30 Announcement for a particular module i: e Software Engineering

Entered Query:

SELECT A.Announcement_ID, M.Module_Name, A.Ann_Title, A.Date_Posted as dates

FROM Announcement A

JOIN Student_Teacher_Announcement N ON A.Announcement_ID =

N.Announcement_ID

JOIN Module M ON N.Module_ID = M.Module_ID

WHERE A.Date_Posted >= TO_DATE('01-MAY-2024', 'DD-MON-YYYY')
AND A.Date_Posted <= TO_DATE('28-MAY-2024', 'DD-MON-YYYY')</p>





AND M.Module_Name = 'Software Engineering';

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

Figure 31 Module name that starts from D

Entered Query:

SELECT A.Module_Name, COUNT(B.Resource_ID) as SUM

From Module A

LEFT JOIN Student_Module_Resource B ON A.Module_ID = B.Module_ID

Where A. Module_Name LIKE 'D%'

Group by A.Module_Name;





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

Figure 32 All students along with their enrolled program who have not submitted any assessments

Entered Query:

SELECT s.Student_ID, s.Student_Name, p.Program_Name,m.Module_Name,

ms.Asse_Status AS status

FROM Student s

INNER JOIN Program p ON s.Program_ID = p.Program_ID

INNER JOIN Module_Student sm ON s.Student_ID = sm.Student_ID
INNER JOIN Module m ON sm.Module_ID = m.Module_ID





LEFT JOIN Assessment_Module_Student ms ON s.Student_ID = ms.Student_ID

AND m.Module_ID = ms.Module_ID

WHERE ms.Asse_Status = 'Not Submitted' OR ms.Asse_Status IS NULL;

List all the teachers who teach more than one module.

- Figure 33 teachers who teach more than one module

 Entered Query:
- Select T.Teacher_Name, Count(M.Module_ID) As SUM From Teacher T
- join Student_Module_Teacher M ON T.Teacher_ID = M.Teacher_ID

Group by T.Teacher_Name



HAVING COUNT(M.Module_ID) > 1;

Transaction Query:

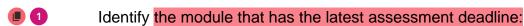


Figure 34 module that has the latest assessment deadline

Entered Query:

SELECT m.Module_Name, a.Asse_Deadline

FROM Module m

JOIN Assessment_Module_Student ma ON m.Module_ID = ma.Module_ID





JOIN Assessment a ON ma.Assessment_ID = a.Assessment_ID

WHERE a.Asse_Deadline = (SELECT MAX(Asse_Deadline)

FROM Assessment);

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

Figure 35 top three students who have the highest total score across all modules

Entered Query:

SELECT *

FROM (

SELECT s.Student_Name, SUM(ams.Marks_Obtained) AS Total_Score





FROM Student s JOIN Assessment_Module_Student ams ON s.Student_ID = ams.Student_ID WHERE ams.Marks_Obtained IS NOT NULL GROUP BY s.Student_Name ORDER BY Total_Score DESC WHERE ROWNUM <= 3;

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

Figure 36 total number of assessments for each program and the average score across all assessments in those programs





Entered Query:

SELECT p.Program_Name AS Program_Name,

COUNT(a.Assessment_ID) AS Total_Assessments_Count,

AVG(ams.Marks_Obtained) AS Average_Marks_Obtained

FROM Program p

JOIN Student s ON p.Program_ID = s.Program_ID

JOIN Assessment_Module_Student ams ON s.Student_ID = ams.Student_ID

JOIN Assessment a ON ams. Assessment_ID = a. Assessment_ID

WHERE ams.Marks_Obtained IS NOT NULL

GROUP BY p.Program_Name;

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





Figure 37 total number of assessments for each program and the average score across all assessments in those programs

Entered Query:

SELECT s.Student_Name AS Student_Name, ams.Marks_Obtained AS
Marks_Obtained

FROM Student s

JOIN Assessment_Module_Student ams ON s.Student_ID = ams.Student_ID

JOIN Module m ON ams.Module_ID = m.Module_ID

WHERE m.Module_Name = 'Database Systems'

AND ams.Marks_Obtained > (

SELECT AVG(ams2.Marks_Obtained)

FROM Assessment_Module_Student ams2

JOIN Module m2 ON ams2. Module_ID = m2. Module_ID

WHERE m2. Module_Name = 'Database Systems'

);

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



Figure 38 student has passed or failed as remarks as per their total aggregate marks obtained in a particular module

Entered Query:

SELECT s.Student_Name AS Student_Name, m.Module_Name AS Module_Name, SUM(ams.Marks_Obtained) AS Total_Marks_Obtained,
CASE

WHEN SUM(ams.Marks_Obtained) >= 40 THEN 'Pass'

ELSE 'Fail'

END AS Pass_Fail_Status

FROM Student s

JOIN Assessment_Module_Student ams ON s.Student_ID = ams.Student_ID

JOIN Module m ON ams. Module ID = m. Module ID

GROUP BY s.Student_Name, m.Module_Name

ORDER BY s. Student_Name;

Critical Evaluation:

14.1. Overview of Module:

The Database Design and Implementation module is vital for understanding how to





organize and manage data within computer systems. It covers fundamental database concepts, including:

Data organization and storage: How to structure and store data effectively.

Database creation: Basic principles of building databases.

Normalization: Organizing data to reduce redundancy and improve efficiency.

SQL queries: Using SQL to interact with and manipulate data (Ana L.C. Bazzan, n.d.).

This module will be practical experience for the students to build a real-world database, for example, an e-commerce platform. This will show the students how applicable database concepts are in other fields, such as:

E-learning: Building online learning platforms.

Healthcare: Patient records and medical data management.

Finance: Handling financial transactions and data analysis.

Business: Data-driven decisions and reports.

Interconnections with Other Subjects

The database module has very strong interconnections with other subjects within IT and business:

Software Engineering: Efficient code that interfaces with data should be written based on database design.

Data Structures and Algorithms: Advanced data structures such as trees and graphs are used to optimize data storage in databases.

Business Management: Database skills will be crucial in making informed business decisions based on the analysis of data.





14.2. Coursework Critical Assessment:

St. Mary's College E-Learning Platform Project

I built a database for an online learning website called St. Mary's College. This website would help students learn through the use of modern technology. Although the project was new and challenging for me, I completed it with the help of my teacher.

The major steps involved in the project were that first, I identified the basic elements of the database, which included student information, courses, and study programs. From these, I created an Entity Relationship Diagram, which is referred to as an ERD. I then decomposed the big data tables into smaller ones by a process called normalization, and information was thus handled easily. Some problems did arise at this stage; I just solved them through an inquiry from my teacher and looking online for a solution. Once the normalization to 3NF was done, the final ERD was produced.

The next major activity was to create the database commands in SQL. I have created various types of commands like CREATE statements to create database tables; INSERT statements for creating students and courses information. Each table needed at least 7 rows of data. COMMIT statements to save all the changes. For the last part of this project, I wrote an evaluation on two main topics: How this module of database relates to other subjects; Detailed review of the coursework.







I faced many challenges, but at the end of it all, I successfully built a working elearning platform database. This project taught me a lot about database design and management.

Screenshot of Dump File:

Figure 39 Screenshot of Dump File

Screenshot of Dropping all Tables:

Figure 40 Screenshot of Dropping all Tables