Document 3.docx

Islington College,Nepal

Document Details

Submission ID

trn:oid:::3618:79886597

Submission Date

Jan 23, 2025, 12:48 AM GMT+5:45

Download Date

Jan 23, 2025, 12:51 AM GMT+5:45

File Name

Document 3.docx

File Size

51.5 KB

79 Pages

6,899 Words

44,169 Characters

35% Overall Similarity

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

Match Groups

214 Not Cited or Quoted 34%
Matches with neither in-text citation nor quotation marks

2 Missing Quotations 0%
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%	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

214Not Cited or Quoted 34%
Matches with neither in-text citation nor quotation marks

2 Missing Quotations 0%
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%	Submitted works (Student Papers)

Top Sources

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

1	Internet	www.coursehero.com	6%
2	Submitted works	islingtoncollege on 2025-01-03	5%
3	Submitted works	islingtoncollege on 2024-12-30	4%
4	Submitted works	islingtoncollege on 2025-01-02	3%
5	Submitted works	islingtoncollege on 2025-01-03	2%
6	Submitted works	islingtoncollege on 2025-01-03	2%
7	Submitted works	islingtoncollege on 2025-01-02	2%
8	Submitted works	islingtoncollege on 2024-12-30	1%
9	Submitted works	University of Greenwich on 2012-05-04	1%
10	Submitted works	islingtoncollege on 2025-01-03	<1%

11	Submitted works	Crown Institute of Business and Technology on 2024-09-22	<1%
12	Submitted works	Macquarie University on 2024-09-22	<1%
13	Internet	library.fiveable.me	<1%
14	Submitted works	Higher Education Commission Pakistan on 2011-06-06	<1%
15	Submitted works	York St John University on 2025-01-21	<1%
16	Submitted works	islingtoncollege on 2025-01-02	<1%
17	Internet	prepinsta.com	<1%
18	Submitted works	AlHussein Technical University on 2024-04-20	<1%
19	Submitted works	Roehampton University on 2021-03-21	<1%
20	Submitted works	Monash University on 2024-10-28	<1%
21	Submitted works	University of Northumbria at Newcastle on 2018-06-11	<1%
22	Submitted works	islingtoncollege on 2024-12-30	<1%
23	Submitted works	Australian Catholic University on 2024-09-03	<1%
24	Submitted works	Colorado Technical University Online on 2024-08-22	<1%

25	Submitted works	Embry Riddle Aeronautical University on 2024-06-14	<1%
26	Submitted works	Middle East College on 2024-12-18	<1%
27	Submitted works	islingtoncollege on 2025-01-03	<1%
28	Submitted works	American Public University System on 2023-11-14	<1%
29	Submitted works	CTI Education Group on 2017-05-22	<1%
30	Submitted works	Central Queensland University on 2017-05-26	<1%
31	Submitted works	New College Swindon on 2024-12-08	<1%
32	Submitted works	University College Birmingham on 2025-01-13	<1%
33	Submitted works	islingtoncollege on 2024-12-30	<1%
34	Submitted works	Higher Education Commission Pakistan on 2012-04-11	<1%
35	Submitted works	University of Ulster on 2015-12-11	<1%

1.Introduction

Ms. Mary has come with a revolutionary E-Classroom Platform that bridges the gaps in virtual education, integrating students, teachers, programs, and modules seamlessly into an academic ecosystem for continuous growth for Southwestern college established in 2018 AD. This digital solution, driven by this new-age, state-of-the-art platform, will bring a facelift to the evolving educational landscape at colleges, focusing on operation efficiency improvement. The E-Classroom Platform ensures that all processes regarding student admissions, module administration, assessment tracking, and the delivery of resources are smoothly executed and meaningful interactions among its key stakeholders are fostered by digitizing such processes. The structured yet flexible approach of the platform aspires to create an intuitive, engaging learning experience that will afford both students and educators the power to thrive in a modern, interconnected academic environment.

The platform digitizes core academic processes, including student admissions, module administration, assessment tracking, and resource delivery. By streamlining these operations, it eliminates unnecessary administrative burdens, allowing educators to focus on teaching and students to concentrate on learning. Moreover, it offers a highly structured yet adaptable framework, catering to the unique needs of institutions while ensuring that all stakeholders have a seamless and intuitive experience. Its design encourages collaboration and engagement, bridging gaps that often exist in virtual education.

More than just a tool, the E-Classroom Platform is an innovative solution that enhances both teaching and learning. It empowers students by providing them with access to interactive resources and tools that make learning more engaging, while educators are equipped with systems to efficiently manage their classes and track student progress. This flexible and intuitive approach ensures that the platform is not just a response to the current needs of education but a visionary step toward a more connected and impactful academic future.

Figure 1: Southwestern College.

2. Current Business Activities and Operations

The college currently operates multiple degree programs in various disciplines, such as BSc in Computing, Networking, and Multimedia. Key activities include:

2.1 Program Management:

Students enroll in one of several programs, each comprising mandatory modules that define their academic path.

2.2 Module Delivery and Assessment:

Each teaching module is assigned to certain teachers. There are modules between programs (such as Programming in Computing and Multimedia) where duplication of modules occurs. Each module has one or more assessments to be graded for performance measurement for students.

2.3 Resource Management:

Every module is equipped with resource-analyzed structure (for instance, video lectures, notes) which is required to be completed by the students in a prescribed order so that he/she steps up improve learning accordingly.

2.4 Announcements:

Most instructors provide reminders for their students through announcements related to the module for deadlines, additional resources, or changes in the syllabus. The system or partially digitized process suffers from inefficiencies, lack of scalability, and limited data integration. The proposed databasing system, therefore, would make all three possible through fully automating the operations while optimizing it.

3. Business Rules Derived from Operational Procedures

To maintain consistency and efficiency, the following business rules are proposed:

One student can be enrolled in only one of the programs, and every program has many students.

A program has many modules, and modules can be part of many programs.

A teacher is assigned to teach specific modules, and a module consists of different teachers.

A teacher can post announcements for their respective module only, and announcements can be posted about different modules.

A module has single or multiple assessments, and each assessment is linked to only one module.

Student can see result of each module.

Every module can have multiple resources but resources will only belong to one module.

Each assessment can have multiple results since every student taking an assessment will generate a different result.

4.Entities and Attributes

1. Student

S.no

Attribute_Name

Datatype

Size

Constraint

1

Student_Id

number

10

Primary key

2

Student_Name

character

50

Not null

3

Student_Email

date

-

Not null

4

Student_Address

character

100

unique

Table 2- Student(Entities and Atrributes)

2. Program

S.no

Attribute_Name

Datetype

Size

Constraint

1

Program_id

Number

10

Primary key

2

Program_name

character

50

Not null

3

Program_Duration

character

255

Not null

4

Program_Title

Number

3

Not null

Table 3- Program Table.

3. Module

S.no

Attribute_Name

Datatype

Size

Constraint

1

Module_id

number

10

Primary key

2

Module Name

character

50

Not null

3

Credits

number

10

Not Null

4

Resource_id

number

10

Not Null

5

Resource title

character

100

Not null

6

Resource type

character

10

Not null

7

Resource status

character

50

Not null

8

Assessment_id

number

10

Not null

9

Assessment title

character

50

Not null

10

Assessment deadline

date

-

Not null

11

weightage

number

3

null

12

Result_id

number

10

Not Null

13

Result total mark

number

5,2

Not null

14

Result remark

character

10

Not null

15

Announcement_id

Number

10

Not null

16

Announcement_Title

character

50

Not null

17

Announcement date

date

-

Null allowed

18

Announcement description

character

100

Null allowed

19

Teacher_id

number

10

Not null

20

Teachers_name

character

50

Not null

21

Teachers_Email

character

100

Not Null

Table 4- Module Table.

4. Initial Entity Relationship Diagram.

Figure 2 Entity Relationship Diagram.

5. Normalization

Normalization in database design is the process of organizing data into related, smaller tables to reduce data redundancy and improve data integrity. This has to do with breaking down a big table that could have some anomalies into efficient forms; this would be 1NF-just the atomic values and elimination of the repeating groups, 2NF-eliminate the partial dependencies by ensuring non-key attributes depend on the entirety of the primary key, and 3NF-transitive dependency would need elimination, making sure all the non-key attributes depend on a primary key. This process enhances consistency, reduces redundancy, and makes maintenance easier while the database remains scalable and efficient.

UNF

Unnormalized Form (UNF) is the raw data representation where all information is stored in a single table with repeating groups or arrays, lacking structure or normalization. It often includes nested and duplicate data.

(student_id,student_Name,Student_Email,Student_Address,program_id,program_name,program_duration,program_Title{Module_id,Module Name,credits{Resource_id,Resource title,Resourse type,Resource Status},{Assessment_id,Assessment title,Assessment deadline,Weightage,Result id,Result total marks,Result remark},{Teacher_id,Teacher_name,Teacher_email{Announcement_id,Announcement_Title,Announcement_date, Announcement_Description }})

1NF

A relation violates the First Normal Form (1NF) if it contains composite attributes (attributes combining multiple pieces of information) or multi-valued attributes (attributes storing multiple values for a single entity). To comply with 1NF, each attribute must hold a single, atomic value, meaning each cell in the table contains only one value. This ensures the data is unambiguous and easy to query, update, and manage. For example, storing multiple phone numbers or subjects in a single attribute violates 1NF, but breaking them into separate rows for each value ensures the table adheres to 1NF, promoting clarity and eliminating redundancy (Geeksforgeeks, 2025).

Student-1

(Student_id,Student_Name,Student_Email,Student_Addreess,program_id,program_name,program_duration,program_Title)

Module-1

(Student_ id,Module_Id,Module Name,credits)

Resource-1

(Student_ id,Module_Id,Resource Id, Resource ID,Resource title,Resourse type,Resource Status)

Assessment-1

(Student_ id,Module_Id,Assessment Id,Assessment title,Assessment deadline,Weightage,Result id,Result total marks,Result remark)

Teacher-1

8 (Student_ id,Module_Id,Teacher_id,Teacher_name,Teacher_email)

Announcement-1

(Student_id,Module_Id,Teacher_id,Announcement_id,Announcement_Title,Announcement_date,Announcement_Description)

2NF

15 The First Normal Form (1NF) focuses solely on eliminating repeating groups and ensuring that all attributes contain atomic (single) values, but it does not address redundancy. 18 This is why the Second Normal Form (2NF) is introduced. A table is 12 considered to be in 2NF if it satisfies two conditions: it is already in 1NF, and there are no partial dependencies. This means that every non-key attribute must be fully 28 dependent on the entire primary key, rather than just a part of it. Partial dependency 23 typically occurs in tables with composite primary keys, where some attributes depend only on a subset of the key rather than the full key. By removing partial dependencies, 2NF reduces redundancy and enhances data consistency (Chris, 2022).

Checking Functional dependency:

Module:

Module id→ module Name,credit.

5 Student_id→×

Teacher:

Teacher id→Teacher_id,Teacher_name,Teacher_email

Student_id→×

Module_id→×

Announcement:

teacher_id→×

student_id→×

module_id→×

Announcement_id-Announcement_Title,Announcement_date

Resources:

Resource_id→ Resource_title,Resource_Type,

Student_id,Module_id→Resource_status

Assessment:

Assessment_id → Assessment title,Assessment deadline,Weightage

Student_id, Module_id→ Result_id,Result total marks,Result remark

2NF-

Student-2

(Student_id,Student_Name,Student_Email,Student_Addreess,program_id,program_name,program_duration,program_Title)

Module-2

(Module_id, Module Name,Module credits)

Student-module-2

(Student_ id*,Module id*)

Resource-2

(Resource_id, Resource title,Resource Type)

Student-module-Resource-2

(Student_id*,Module_id*,Resource Status)

Assessment-2

(Assessment_id, Assessment title,Assessment deadline,Weightage)

Student-module-Assessment-2

(Student_ id*,Module_ id*,Assessment_ id,Result_id,Result total marks,Result remark)

Teacher-2

(Teacher_ id,Teacher_name,Teacher_email)

Student-Module-teacher-2

(Student_ id*,Module_id*,Teacher_ id*)

Announcement-2

(Announcement_id,Announcement Title,Announcement Date,Announcement Description)

Student-annoucement-2

(Student_id,Module_id,Teacher_ id,Announcement_ id)

3NF

Third Normal Form (3NF): A relation is in Third Normal Form (3NF) if it satisfies the conditions of Second Normal Form (2NF) and eliminates transitive dependencies, meaning no non-key attribute depends on another non-key attribute. In 3NF, all non-key attributes must depend only on the primary key, ensuring that the relation is free from redundancy and anomalies caused by indirect dependencies. This normalization step improves data integrity and results in a well-organized and efficient database design, reducing the risk of inconsistencies during data updates or modifications (Geeksforgeeks, 2025).

3NF-

Student Table: This table holds information about students, like their ID, name, email, address, and the program they are enrolled in. The program_id links each student to a specific program.

Program Table: This table lists details about the programs available, such as the program's name, duration, and title. Each program is uniquely identified by program_id.

Module Table: Modules, which are parts of a program, are listed here. It includes the module's name, ID, and credits. Each module has a unique module_id.

Student-Module Table: This is a connection table that links students to the modules they are taking. It records which student is taking which module by storing their respective IDs.

Resource Table: This table stores information about various resources available to students, such as their title and type. Each resource is identified by a resource_id.

Student-Module-Resource Table: This table tracks the status of resources assigned to students within specific modules. It connects students, modules, and the status of each resource they have access to.

Student-Resource Table: This table links students to specific resources, showing which resources are assigned to them in which modules.

Assessment Table: This table includes details about assessments, such as the assessment's ID, title, deadline, and weightage (importance). Each assessment is identified by a assessment_id.

Student-Module-Assessment Table: This table maps students to the assessments in the modules they are taking, showing which assessment each student has for each module.

Student-Module-Assessment-Result Table: This table stores the results of students' assessments, including their marks and any feedback. It links students, modules, assessments, and results.

Teacher Table: This table holds information about teachers, like their ID, name, and email. Each teacher has a unique teacher_id.

Student-Module-Teacher Table: This table links students with the teachers for the modules they are enrolled in, showing which teacher is teaching which student in which module.

Announcement Table: This table includes announcements made by teachers, such as the title, date, and description. Each announcement has a unique announcement_id.

Student-Announcement Table: This table tracks which students have received which announcements from teachers for specific modules, linking students, modules, teachers, and announcements.

.

5

Student-3

(Student_id,Program id*,Student_Name,Student_Email,Student_Addreess)

Program - 3

(program_id,program_name,program_duration,program_Title)

Module-3

(Module_id, Module Name,Module credits)

Student-module-3

(Student_ id*,Module_id*)

Resource-3

(Resource id, Resource title,Resource Type)

Student-module-Resource-3

(Student_id*,Module_id*,Resource_id*,Resource Status)

Assessment-3

(Assessment_id, Assessment title,Assessment deadline,Weightage)

Student-module-Assessment-Result-3

(Student_id*,Module_id*, Assessment_id*,Result id)

Result -3

(Result_id,Result total marks,Result remark)

Teacher-3

(Teacher_ id,Teacher_name,Teacher_email)

Student-Module-teacher-3

(Student_ id*,Module_ id*,Teacher_ id*)

Announcement-3

(Announcement_id,Announcement Title,Announcement Date,Announcement Description)

Student-annoucement-3

(Student_id*,Module _id*,Teacher_ id*,Announcement_ id*)

6.Data Dictionary.

1. Student

Sno

Attribute

Datatype

Size

Constraints

Composite Constraint

1

student_id

Number

10

Primary Key, Not Null, Unique

-

2

student_name

Character

50

Not Null

-

3

enrollment_date

Date

-

Not Null

-

4

student_email

Varchar

50

Not Null, Unique

-

5

program_id

Number

10

Foreign Key (references Program-3)

-

Table 5- Student Table(Data Dictionary)

2. Program

Sno

Attribute

Datatype

Size

Constraints

Composite Constraint

1

program_id

Number

10

Primary Key, Not Null, Unique

-

2

program_name

Character

50

Not Null

-

3

program_duration

Number

3

Not Null

-

4

program_title

Character

100

Not Null

-

Table 6- Program Table.

3. Module

Sno

Attribute

Datatype

Size

Constraints

Composite Constraint

1

module_id

Number

10

Primary Key, Not Null, Unique

-

2

module_name

Character

50

Not Null

-

3

module_credits

Number

3

Not Null

-

Table 7- Module Table.

4. Student-Module.

Sno

Attribute

Datatype

Size

Constraints

Composite Constraint

1

student_id

Number

10

Foreign Key (references Student-3)

Part of Composite Primary Key

2

module_id

Number

10

Foreign Key (references Module-3)

Part of Composite Primary Key

Table 8- Student_Module Table.

5. Resource

Sno

Attribute

Datatype

Size

Constraints

Composite Constraint

1

resource_id

Number

10

Primary Key, Not Null, Unique

-

2

resource_title

Character

100

Not Null

-

3

resource_type

Character

50

Not Null

-

Table 9- Resource Table.

6. Student-Module-Resource.

Sno

Attribute

Datatype

Size

Constraints

Composite Constraint

1

student_id

Number

10

Foreign Key (references Student-3)

Part of Composite Primary Key

2

module_id

Number

10

Foreign Key (references Module-3)

Part of Composite Primary Key

3

resource_id

Number

10

Foreign Key (references Resource-3)

Part of Composite Primary Key

4

resource_status

Character

20

Not Null

-

Table 10- Student-Module-Resource Table.

7. Assessment.

Sno

Attribute

Datatype

Size

Constraints

Composite Constraint

1

assessment_id

Number

10

Primary Key, Not Null, Unique

-

2

assessment_title

Character

100

Not Null

-

3

assessment_deadline

Date

-

Not Null

-

4

weightage

Number

3

Not Null

-

Table 11- Assessment Table.

8. Student-Module-Assessment-Result.

Sno

Attribute

Datatype

Size

Constraints

Composite Constraint

1

student_id

Number

10

Foreign Key (references Student-3)

Part of Composite Primary Key

2

module_id

Number

10

Foreign Key (references Module-3)

Part of Composite Primary Key

3

assessment_id

Number

10

Foreign Key (references Assessment-3)

Part of Composite Primary Key

4

result_id

Number

10

Foreign Key (references Result-3)

-

Table 12- Student-Module-Assessment Table.

9. Result.

Sno

Attribute

Datatype

Size

Constraints

Composite Constraint

1

result_id

Number

10

Primary Key, Not Null, Unique

-

2

result_total_marks

Number

5

Not Null

-

3

result_remark

Character

100

-

-

Table 13- Result Table.

10. Teacher.

Sno

Attribute

Datatype

Size

Constraints

Composite Constraint

1

teacher_id

Number

10

Primary Key, Not Null, Unique

-

2

teacher_name

Character

50

Not Null

-

3

teacher_email

Varchar

50

Not Null, Unique

-

Table 14- Teacher Table.

11. Student-Module-Teacher.

Sno

Attribute

Datatype

Size

Constraints

Composite Constraint

1

student_id

Number

10

Foreign Key (references Student-3)

Part of Composite Primary Key

2

module_id

Number

10

Foreign Key (references Module-3)

Part of Composite Primary Key

3

teacher_id

Number

10

Foreign Key (references Teacher-3)

Part of Composite Primary Key

Table 15- Student-Module-Teacher Table.

12. Announcement.

Sno

Attribute

Datatype

Size

Constraints

Composite Constraint

1

announcement_id

Number

10

Primary Key, Not Null, Unique

-

2

announcement_title

Character

100

Not Null

-

3

announcement_date

Date

-

Not Null

-

4

announcement_description

Character

255

-

-

Table 16- Announcement Table.

13. Student-Announcement.

Sno

Attribute

Datatype

Size

Constraints

Composite Constraint

1

student_id

Number

10

Foreign Key (references Student-3)

Part of Composite Primary Key

2

module_id

Number

10

Foreign Key (references Module-3)

Part of Composite Primary Key

3

teacher_id

Number

10

Foreign Key (references Teacher-3)

Part of Composite Primary Key

4

announcement_id

Number

10

Foreign Key (references Announcement-3)

Part of Composite Primary Key

Table 17- Student-Announcement Table.

6. Final ERD

The final ERD represents a normalized and optimized database design, following all the requirements and business rules outlined for the "E-Classroom Platform." It reflects a structured relationship between entities and embodies all the constraints derived during the normalization process.

Figure 3: Final ERD

7. Implementation

1. Creating User.

Figure 4: Creating User.

29

2.Creating Tables.

2.1 Create Program Table.

Figure 5: Creating Tables.

10

2.2 Create Student Table.

Figure 6: Creating Student Table.

2.3 Create Module Table.

Figure 7: Creating Module Table.

4

2.4 Create Student_Module Table.

Figure 8: Creating Student_Module Table.

2.5 Create Resource_Table.

Figure 9: Creating Resource Table.

2.6 Create Student_Module_Resource Table.

Figure 10: Creating Student_Module_Resource Table.

2.7 Create Assessment Table.

Figure 11: Creating Assessment Table.

2.8 Create Result Table.

Figure 12: Creating Result Table.

2.9 Create Student_Module_Assessment Table.

Figure 13: Creating Student_Module_Assessment Table.

2.10 Create Teacher Table.

Figure 14: Creating Teacher Table.

2.11 Create Student_Module_Teacher Table.

Figure 15: Create Student_Module_Teacher Table.

2.12 Create Announcement Table.

Figure 16: Creating Announcement Table.

2.13 Create Student_Announcement Table.

Figure 17: Creating Student_Announcement Table.

3.Inserting the values

1)Inserting values in program table

Figure 18: Inserting in Program Table.

2) Student table

Figure 19: Inserting in Student Table.

Figure 20: Select from Student.

3)module table

Figure 21: Inserting in module Table.

Figure 22: Select from Module.

4) Student Module

Figure 23: Inserting into Student_Module Table.

Figure 24: Select Student_Module Table.

5) Resource_Table

Figure 25: Inserting into Resource Table.

Figure 26: Select from Resource Table.

6)student module resource

Figure 27: Inserting into Student_Module_Resource Table.

Figure 28: Selecting From Student_Module_Resource Table.

7)Assessment

19

Figure 29: Inserting into Assessment Table.

Figure 30: Select from Assessment Table.

8)Result

Figure 31: Inserting into Result Table.

Figure 32: Select from Results.

9)student_ module _assessment

Figure 33: Inserting into Student_Module_Assessment.

Figure 34: Selecting From Student_Module_Assessment.

10)Teacher

Figure 35: Insert into Teacher.

Figure 36: Select from Teacher.

11) student module teacher

Figure 37: Insert into Student_Module_Teacher Table.

Figure 38:Select from student module teacher Table.

12)Announcement New

Figure 39: Inserting into Announcement Table.

Figure 40: Selecting From Announcement table.

13)student announcement

Figure 41: Insert into Student_Announcement Table.

Figure 42: Select from student_Announcement Table.

4)Query

4.1 Information query

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

Figure 43: Information query"1".

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

Figure 44: Information query "2".

As I have no data entry starting from 1st May 2024 to 28th May 2024 so as a result there is no rows selected.

3) List the names of all modules that begin with the letter 'C', along with the total number of resources uploaded for those modules

Figure 45: Information query "3".

There in question it is asked to list all the modules that begin with “D” but I have not inserted any module starting with letter “D” so here I have performed a query using letter “C”.

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

Figure 46: Information query "4".

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

Figure 47: Information query "5".

Here in my database system one teacher is supposed to teach only one module so there is no rows selected to list all the teachers who teach more than one module.

4.2 Transaction Query

1) Identify the module that has the latest assessment deadline

Figure 48: Transaction query "1".

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

Figure 49: Transaction query "2".

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

Figure 50:Transaction query "3".

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

Figure 51: Transaction query "4".

In my database system Module named Databases doesn’t exists.

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

Figure 52:Transaction query"5".

Dump file

Figure 53: Dump File I

Figure 54: Dump File II.

Drop table

Figure 55: Dropping Table (I).

Figure 56: Dropping Table (II).

8)Critical Evaluation

Learning from the Coursework.

It was during this course that I learned most about database design and implementation in developing a very robust e-classroom platform system. Having understood entities, attributes, and relationships, I also learned normalizing data structures into Third Normal Form, 3NF, and implementing them in Oracle SQL. Preparing ERDs further enabled me to visualize the flow of data in order to obtain appropriately logical arrangement for entities. This assignment on normalization and querying indeed gave insight into how theoretical concepts could actually be applied to solving real world-type problems and helped me boost my critical and problem-solving aspect of thinking.

Challenges Faced:

There were ups and downs in the journey, with much of the integrity of data compromised while establishing relationships and constraints, especially in complex entities, such as assessments and resources, that may need a highly efficient writing of Oracle SQL queries for advanced functionalities and transactions where one needed syntax and logical accuracy to be maniacally attentive. It was also very time draining to adhere to the demand of course-required detailed documentation and screenshots for every step involved in creating the object. Besides, balancing theoretical understanding with technical accuracy became exasperating in itself. It was not easy going.

Overall Experience:

Overall, this course was an extremely enriching experience in many ways; indeed, it enhanced both my technical and analytical abilities. Here was the opportunity to put academic knowledge into practice in one large project, and it was a real exhaustive experience preparatory to the database management challenges awaiting me in the real world. The setbacks notwithstanding, there is gratification in designing something from scratch and in observing its progress.

9. Conclusion

The coursework designates the development of a robust database system that would serve the needs of the "E-Classroom Platform," conforming to proper normalisation principles while observing the set business rules. This involves analysing the operational needs by developing entity-relationship diagrams, further developing the structure for consistency and efficiency in the data. We reduced data redundancy, improved data integrity, and established relationships between key entities such as programs, modules, students, teachers, assessments, and resources with the development of a fully normalized database.

Also, the use of Oracle SQL in implementing this database demonstrated how to apply theoretical concepts to make such a system workable, scalable, and pertinent to practical needs. The incorporation of structured queries and comprehensive test data was the validation that this system was indeed capable of managing complex educational operations efficiently. This project has also laid a very solid foundation for enhancements that could be made in the future to this platform, aside from bringing forth the importance of proper database design in creating dynamic digital environments. Knowledge gained from this coursework strengthened our understanding of database principles and their critical role in modern information systems.

Figure 1: Southwestern College.

Current Business Activities and Operations

The college currently operates multiple degree programs in various disciplines, such as BSc in Computing, Networking, and Multimedia. Key activities include: 2.1 Program Management: Students enroll in one of several programs, each comprising mandatory modules that define their academic path. 2.2 Module Delivery and Assessment: Each teaching module is assigned to certain teachers. There are modules between programs (such as Programming in Computing and Multimedia) where duplication of modules occurs. Each module has one or more assessments to be graded for performance measurement for students. 2.3 Resource Management: Every module is equipped with resource-analyzed structure (for instance, video lectures, notes) which is required to be completed by the students in a prescribed order so that he/she steps up improve learning accordingly. 2.4 Announcements: Most instructors provide reminders for their students through announcements related to the module for deadlines, additional resources, or changes in the syllabus. The system or partially digitized process suffers from inefficiencies, lack of scalability, and limited data integration. The proposed databasing system, therefore, would make all three possible through fully automating the operations while optimizing it.

Business Rules Derived from Operational Procedures To maintain consistency and efficiency, the following business rules are proposed: • One student can be enrolled in only one of the programs, and every program has many students. • A program has many modules, and modules can be part of many programs. • A teacher is assigned to teach specific modules, and a module consists of different teachers. • A teacher can post announcements for their respective module only, and announcements can be posted about different modules. • A module has single or multiple assessments, and each assessment is linked to only one module. • Student can see result of each module. • Every module can have multiple resources but resources will only belong to one module. • Each assessment can have multiple results since every student taking an assessment will generate a different result.

4.Entities and Attributes

Student S.no Attribute_Name Datatype Size Constraint 1 Student_Id number 10 Primary key 2 Student_Name character 50 Not null 3 Student_Email date - Not null 4 Student_Address character 100 unique Table 2- Student(Entities and Atrributes)

Program S.no Attribute_Name Datetype Size Constraint 1 Program_id Number 10 Primary key 2 Program_name character 50 Not null 3 Program_Duration character 255

Not null 4 Program_Title Number 3 Not null Table 3- Program Table.

Module S.no Attribute_Name Datatype Size Constraint 1 Module_id number 10 Primary key 2 Module Name character 50 Not null 3 Credits number 10 Not Null 4 Resource_id number 10 Not Null 5 Resource title character 100 Not null 6 Resource type character 10 Not null 7 Resource status character 50 Not null 8 Assessment_id number 10 Not null 9 Assessment title character 50 Not null 10 Assessment deadline date - Not null 11 weightage number 3 null 12 Result_id number 10 Not Null 13 Result total mark number 5,2 Not null 14 Result remark character 10 Not null 15 Announcement_id Number 10 Not null 16 Announcement_Title character 50 Not null 17 Announcement date date - Null allowed 18 Announcement description character 100 Null allowed 19 Teacher_id number 10 Not null 20 Teachers_name character 50 Not null 21 Teachers_Email character 100 Not Null Table 4- Module Table.

Initial Entity Relationship Diagram.

Figure 2 Entity Relationship Diagram.

Normalization

Normalization in database design is the process of organizing data into related, smaller tables to reduce data redundancy and improve data integrity. This has to do with breaking down a big table that could have some anomalies into efficient forms; this would be 1NF-just the atomic values and elimination of the repeating groups, 2NF-eliminate the partial dependencies by ensuring non-key attributes depend on the entirety of the primary key, and 3NF-transitive dependency would need elimination, making sure all the non-key attributes depend on a primary key. This process enhances consistency, reduces redundancy, and makes maintenance easier while the database remains scalable and efficient.

UNF

Unnormalized Form (UNF) is the raw data representation where all information is stored in a single table with repeating groups or arrays, lacking structure or normalization. It often includes nested and duplicate data.

(student_id,student_Name,Student_Email,Student_Address,program_id,program_name,program_duration,program_Title{Module_id,Module Name,credits{Resource_id,Resource title,Resourse type,Resource Status},{Assessment_id,Assessment title,Assessment deadline,Weightage,Result id,Result total marks,Result remark},{Teacher_id,Teacher_name,Teacher_email{Announcement_id,Announcement_Title,Announcement_date, Announcement_Description }})

1NF A relation violates the First Normal Form (1NF) if it contains composite attributes (attributes combining multiple pieces of information) or multi-valued attributes (attributes storing multiple values for a single entity). To comply with 1NF, each attribute must hold a single, atomic value, meaning each cell in the table contains only one value. This ensures the data is unambiguous and easy to query, update, and manage. For example, storing multiple phone numbers or subjects in a single attribute violates 1NF, but breaking them into separate rows for each value ensures the table adheres to 1NF, promoting clarity and eliminating redundancy (Geeksforgeeks, 2025).

Student-1

(Student_id,Student_Name,Student_Email,Student_Addreess,program_id,program_name,program_duration,program_Title) Module-1 (Student_ id,Module_Id,Module

Name,credits) Resource-1 (Student_ id,Module_Id,Resource Id, Resource ID,Resource title,Resourse type,Resource Status) Assessment-1 (Student_id,Module_Id,Assessment Id,Assessment title,Assessment deadline,Weightage,Result id,Result total marks,Result remark) Teacher-1 (Student_id,Module_Id,Teacher_id,Teacher_name,Teacher_email) Announcement-1 (Student_id,Module_Id,Teacher_id,Announcement_id,Announcement_Title,Announcement_date,Announcement_Description)

2NF The First Normal Form (1NF) focuses solely on eliminating repeating groups and ensuring that all attributes contain atomic (single) values, but it does not address redundancy. This is why the Second Normal Form (2NF) is introduced. A table is considered to be in 2NF if it satisfies two conditions: it is already in 1NF, and there are no partial dependencies. This means that every non-key attribute must be fully dependent on the entire primary key, rather than just a part of it. Partial dependency typically occurs in tables with composite primary keys, where some attributes depend only on a subset of the key rather than the full key. By removing partial dependencies, 2NF reduces redundancy and enhances data consistency (Chris, 2022). Checking Functional dependency: Module: Module id→ module Name,credit. Student_id→× Teacher: Teacher id→Teacher_id,Teacher_name,Teacher_email Student_id→× Module_id→× Announcement: teacher_id→× student_id→× module_id→× Announcement_id-Announcement_Title,Announcement_date Resources: Resource_id→ Resource_title,Resource_Type, Student_id,Module_id→Resource_status Assessment: Assessment_id → Assessment title,Assessment deadline,Weightage Student_id, Module_id→ Result_id,Result total marks,Result remark

2NF- Student-2

2 (Student_id,Student_Name,Student_Email,Student_Addreess,program_id,program_na
me,program_duration,program_Title) Module-2 (Module_id, Module Name,Module
3 credits) Student-module-2 (Student_ id*,Module id*) Resource-2 (Resource_id,
Resource title,Resource Type) Student-module-Resource-2
(Student_id*,Module_id*,Resource Status) Assessment-2 (Assessment_id,
Assessment title,Assessment deadline,Weightage) Student-module-Assessment-2
8 (Student_ id*,Module_ id*,Assessment_ id,Result_id,Result total marks,Result remark)
1 Teacher-2 (Teacher_ id,Teacher_name,Teacher_email) Student-Module-teacher-2
3 (Student_ id*,Module_id*,Teacher_ id*) Announcement-2
(Announcement_id,Announcement Title,Announcement Date,Announcement
Description) Student-annoucement-2 (Student_id,Module_id,Teacher_
id,Announcement_ id)

11 3NF Third Normal Form (3NF): A relation is in Third Normal Form (3NF) if it satisfies
the conditions of Second Normal Form (2NF) and eliminates transitive dependencies,
13 meaning no non-key attribute depends on another non-key attribute. In 3NF, all non-
key attributes must depend only on the primary key, ensuring that the relation is free
from redundancy and anomalies caused by indirect dependencies. This normalization
step improves data integrity and results in a well-organized and efficient database
design, reducing the risk of inconsistencies during data updates or modifications
(Geeksforgeeks, 2025).

3NF-

Student Table: This table holds information about students, like their ID, name, email, address, and the program they are enrolled in. The program_id links each student to a specific program.

24 Program Table: This table lists details about the programs available, such as the program's name, duration, and title. Each program is uniquely identified by program_id.

Module Table: Modules, which are parts of a program, are listed here. It includes the module's name, ID, and credits. Each module has a unique module_id.

Student-Module Table: This is a connection table that links students to the modules they are taking. It records which student is taking which module by storing their respective IDs.

Resource Table: This table stores information about various resources available to students, such as their title and type. Each resource is identified by a resource_id.

Student-Module-Resource Table: This table tracks the status of resources assigned to students within specific modules. It connects students, modules, and the status of each resource they have access to.

Student-Resource Table: This table links students to specific resources, showing which resources are assigned to them in which modules.

Assessment Table: This table includes details about assessments, such as the assessment's ID, title, deadline, and weightage (importance). Each assessment is identified by a assessment_id.

Student-Module-Assessment Table: This table maps students to the assessments in the modules they are taking, showing which assessment each student has for each module.

Student-Module-Assessment-Result Table: This table stores the results of students' assessments, including their marks and any feedback. It links students, modules, assessments, and results.

Teacher Table: This table holds information about teachers, like their ID, name, and email. Each teacher has a unique teacher_id.

Student-Module-Teacher Table: This table links students with the teachers for the modules they are enrolled in, showing which teacher is teaching which student in which module.

Announcement Table: This table includes announcements made by teachers, such as the title, date, and description. Each announcement has a unique announcement_id.

Student-Announcement Table: This table tracks which students have received which announcements from teachers for specific modules, linking students, modules, teachers, and announcements. .

5 Student-3 (Student_id,Program id*,Student_Name,Student_Email,Student_Addreess) Program - 3 (program_id,program_name,program_duration,program_Title) Module-3 (Module_id, Module Name,Module credits) Student-module-3 (Student_ id*,Module_id*) Resource-3 (Resource id, Resource title,Resource Type) Student-module-Resource-3 (Student_id*,Module_id*,Resource_id*,Resource Status) Assessment-3 (Assessment_id, Assessment title,Assessment deadline,Weightage) Student-module-Assessment-Result-3 (Student_id*,Module_id*, Assessment_id*,Result id) Result -3 (Result_id,Result total marks,Result remark) Teacher-3 (Teacher_ id,Teacher_name,Teacher_email) Student-Module-teacher-3

(Student_ id*,Module_ id*,Teacher_ id*) Announcement-3 (Announcement_id,Announcement Title,Announcement Date,Announcement Description) Student-annoucement-3 (Student_id*,Module id*,Teacher id*,Announcement_ id*)

6.Data Dictionary.

Student Sno Attribute Datatype Size Constraints Composite Constraint 1 student_id Number 10 Primary Key, Not Null, Unique - 2 student_name Character 50 Not Null - 3 enrollment_date Date - Not Null - 4 student_email Varchar 50 Not Null, Unique - 5 program_id Number 10 Foreign Key (references Program-3) - Table 5- Student Table(Data Dictionary)

Program Sno Attribute Datatype Size Constraints Composite Constraint 1 program_id Number 10 Primary Key, Not Null, Unique - 2 program_name Character 50 Not Null - 3 program_duration Number 3 Not Null - 4 program_title Character 100 Not Null - Table 6- Program Table.

Module Sno Attribute Datatype Size Constraints Composite Constraint 1 module_id Number 10 Primary Key, Not Null, Unique - 2 module_name Character 50 Not Null - 3 module_credits Number 3 Not Null - Table 7- Module Table.

Student-Module. Sno Attribute Datatype Size Constraints Composite Constraint 1 student_id Number 10 Foreign Key (references Student-3) Part of Composite Primary Key 2 module_id Number 10 Foreign Key (references Module-3) Part of Composite Primary Key Table 8- Student_Module Table.

Resource Sno Attribute Datatype Size Constraints Composite Constraint 1 resource_id Number 10 Primary Key, Not Null, Unique - 2 resource_title Character 100 Not Null - 3 resource_type Character 50 Not Null - Table 9- Resource Table.

Student-Module-Resource. Sno Attribute Datatype Size Constraints Composite Constraint 1 student_id Number 10 Foreign Key (references Student-3) Part of Composite Primary Key 2 module_id Number 10 Foreign Key (references Module-3) Part of Composite Primary Key 3 resource_id Number 10 Foreign Key (references Resource-3) Part of Composite Primary Key 4 resource_status Character 20 Not Null - Table 10- Student-Module-Resource Table.

Assessment. Sno Attribute Datatype Size Constraints Composite Constraint 1 assessment_id Number 10 Primary Key, Not Null, Unique - 2 assessment_title Character 100 Not Null - 3 assessment_deadline Date - Not Null - 4 weightage Number 3 Not Null - Table 11- Assessment Table.

Student-Module-Assessment-Result. Sno Attribute Datatype Size Constraints Composite Constraint 1 student_id Number 10 Foreign Key (references Student-3) Part of Composite Primary Key 2 module_id Number 10 Foreign Key (references Module-3) Part of Composite Primary Key 3 assessment_id Number 10 Foreign Key (references Assessment-3) Part of Composite Primary Key 4 result_id Number 10 Foreign Key (references Result-3) - Table 12- Student-Module-Assessment Table.

Result. Sno Attribute Datatype Size Constraints Composite Constraint 1 result_id Number 10 Primary Key, Not Null, Unique - 2 result_total_marks Number 5 Not Null - 3 result_remark Character 100 - - Table 13- Result Table.

Teacher. Sno Attribute Datatype Size Constraints Composite Constraint 1 teacher_id Number 10 Primary Key, Not Null, Unique - 2 teacher_name Character 50 Not Null - 3 teacher_email Varchar 50 Not Null, Unique - Table 14- Teacher Table.

Student-Module-Teacher. Sno Attribute Datatype Size Constraints Composite Constraint 1 student_id Number 10 Foreign Key (references Student-3) Part of Composite Primary Key 2 module_id Number 10 Foreign Key (references Module-3) Part of Composite Primary Key 3 teacher_id Number 10 Foreign Key (references Teacher-3) Part of Composite Primary Key Table 15- Student-Module-Teacher Table. Announcement. Sno Attribute Datatype Size Constraints Composite Constraint 1 announcement_id Number 10 Primary Key, Not Null, Unique - 2 announcement_title Character 100 Not Null - 3 announcement_date Date - Not Null - 4 announcement_description Character 255 - - Table 16- Announcement Table. Student-Announcement. Sno Attribute Datatype Size Constraints Composite Constraint 1 student_id Number 10 Foreign Key (references Student-3) Part of Composite Primary Key 2 module_id Number 10 Foreign Key (references Module-3) Part of Composite Primary Key 3 teacher_id Number 10 Foreign Key (references Teacher-3) Part of Composite Primary Key 4 announcement_id Number 10 Foreign Key (references Announcement-3) Part of Composite Primary Key Table 17- Student-Announcement Table.

Final ERD

The final ERD represents a normalized and optimized database design, following all the requirements and business rules outlined for the "E-Classroom Platform." It reflects a structured relationship between entities and embodies all the constraints derived during the normalization process.

Figure 3: Final ERD

Implementation

Creating User.

Figure 4: Creating User. 2.Creating Tables. 2.1 Create Program Table.

Figure 5: Creating Tables. 2.2 Create Student Table.

Figure 6: Creating Student Table.

2.3 Create Module Table.

Figure 7: Creating Module Table.

2.4 Create Student_Module Table.

Figure 8: Creating Student_Module Table.

2.5 Create Resource_Table.

Figure 9: Creating Resource Table.

2.6 Create Student_Module_Resource Table.

Figure 10: Creating Student_Module_Resource Table.

2.7 Create Assessment Table.

Figure 11: Creating Assessment Table.

2.8 Create Result Table.

Figure 12: Creating Result Table.

2.9 Create Student_Module_Assessment Table.

Figure 13: Creating Student_Module_Assessment Table.

2.10 Create Teacher Table.

Figure 14: Creating Teacher Table.

2.11 Create Student_Module_Teacher Table.

Figure 15: Create Student_Module_Teacher Table.

2.12 Create Announcement Table.

Figure 16: Creating Announcement Table.

2.13 Create Student_Announcement Table.

Figure 17: Creating Student_Announcement Table.

3.Inserting the values

1)Inserting values in program table

Figure 18: Inserting in Program Table.

Student table

Figure 19: Inserting in Student Table.

Figure 20: Select from Student.

3)module table

Figure 21: Inserting in module Table.

Figure 22: Select from Module.

Student Module

Figure 23: Inserting into Student_Module Table.

Figure 24: Select Student_Module Table.

Resource_Table

Figure 25: Inserting into Resource Table.

Figure 26: Select from Resource Table.

6)student module resource

Figure 27: Inserting into Student_Module_Resource Table.

Figure 28: Selecting From Student_Module_Resource Table.

7)Assessment

Figure 29: Inserting into Assessment Table.

Figure 30: Select from Assessment Table.

8)Result

Figure 31: Inserting into Result Table.

Figure 32: Select from Results. 9)student_ module _assessment

Figure 33: Inserting into Student_Module_Assessment.

Figure 34: Selecting From Student_Module_Assessment.

10)Teacher

Figure 35: Insert into Teacher.

Figure 36: Select from Teacher.

student module teacher

Figure 37: Insert into Student_Module_Teacher Table.

Figure 38:Select from student module teacher Table. 12)Announcement New

Figure 39: Inserting into Announcement Table.

Figure 40: Selecting From Announcement table.

13)student announcement

Figure 41: Insert into Student_Announcement Table.

Figure 42: Select from student_Announcement Table.

4)Query 4.1 Information query

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

Figure 43: Information query"1".

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

Figure 44: Information query "2".

As I have no data entry starting from 1st May 2024 to 28th May 2024 so as a result there is no rows selected. 3) List the names of all modules that begin with the letter 'C', along with the total number of resources uploaded for those modules

Figure 45: Information query "3". There in question it is asked to list all the modules that begin with “D” but I have not inserted any module starting with letter “D” so here I have performed a query using letter “C”.

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

Figure 46: Information query "4". 5) List all the teachers who teach more than one module

Figure 47: Information query "5". Here in my database system one teacher is supposed to teach only one module so there is no rows selected to list all the teachers who teach more than one module.

4.2 Transaction Query

Identify the module that has the latest assessment deadline

Figure 48: Transaction query "1".

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

Figure 49: Transaction query "2".

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

Figure 50:Transaction query "3". 4) List the students who have scored above the average score in the ‘Databases’ module.

Figure 51: Transaction query "4". In my database system Module named Databases doesn't exists.

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

Figure 52:Transaction query"5".

Dump file

Figure 53: Dump File I

Figure 54: Dump File II.

Drop table

Figure 55: Dropping Table (I).

Figure 56: Dropping Table (II).

8)Critical Evaluation Learning from the Coursework. It was during this course that I learned most about database design and implementation in developing a very robust e-classroom platform system. Having understood entities, attributes, and relationships, I also learned normalizing data structures into Third Normal Form, 3NF, and implementing them in Oracle SQL. Preparing ERDs further enabled me to visualize the flow of data in order to obtain appropriately logical arrangement for entities. This assignment on normalization and querying indeed gave insight into how theoretical concepts could actually be applied to solving real world-type problems and helped me boost my critical and problem-solving aspect of thinking. Challenges Faced: There were ups and downs in the journey, with much of the integrity of data compromised while establishing relationships and constraints, especially in complex entities, such as assessments and resources, that may need a highly efficient writing of Oracle SQL queries for advanced functionalities and transactions where one needed syntax and logical accuracy to be maniacally attentive. It was also very time draining to adhere to the demand of course-required detailed documentation and screenshots for every step involved in creating the object. Besides, balancing theoretical understanding with technical accuracy became exasperating in itself. It was not easy going. Overall Experience: Overall, this course was an extremely enriching experience in many ways; indeed, it enhanced both my technical and analytical abilities. Here was the opportunity to put academic knowledge into practice in one large project, and it was a real exhaustive experience preparatory to the database management challenges awaiting me in the real world. The setbacks notwithstanding, there is gratification in designing something from scratch and in observing its progress.

Conclusion

The coursework designates the development of a robust database system that would serve the needs of the "E-Classroom Platform," conforming to proper normalisation principles while observing the set business rules. This involves analysing the operational needs by developing entity-relationship diagrams, further developing the structure for consistency and efficiency in the data. We reduced data redundancy, improved data integrity, and established relationships between key entities such as programs, modules, students, teachers, assessments, and resources with the development of a fully normalized database. Also, the use of Oracle SQL in implementing this database demonstrated how to apply theoretical concepts to make such a system workable, scalable, and pertinent to practical needs. The incorporation of structured queries and comprehensive test data was the validation that this system was indeed capable of managing complex educational operations efficiently. This project has also laid a very solid foundation for enhancements that could be made in the future to this platform, aside from bringing forth the importance of proper database design in creating dynamic digital environments. Knowledge gained from this coursework strengthened our understanding of database principles and their critical role in modern information systems.