23048598ERIKASHRESTHAFINAL.docx

Islington College,Nepal

Document Details

Submission ID
trn:oid:::3618:79930045

Submission Date
Jan 23, 2025, 12:16 PM GMT+5:45

Download Date
Jan 23, 2025, 12:18 PM GMT+5:45

File Name
23048598ERIKASHRESTHAFINAL.docx

File Size
51.5 KB

98 Pages

6,783 Words

44,714 Characters

20% Overall Similarity

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

Match Groups

- **133** Not Cited or Quoted 19%
  Matches with neither in-text citation nor quotation marks
- **11** Missing Quotations 2%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

6% Internet sources
0% Publications
19% 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

- 133 Not Cited or Quoted 19%
  Matches with neither in-text citation nor quotation marks
- 11 Missing Quotations 2%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

- 6% Internet sources
- 0% Publications
- 19% 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 3%
2	Submitted works	islingtoncollege on 2025-01-03 3%
3	Submitted works	Edge Hill College of Higher Education on 2010-01-29 2%
4	Submitted works	islingtoncollege on 2025-01-03 1%
5	Submitted works	islingtoncollege on 2024-12-30 1%
6	Submitted works	Westford School of Management on 2024-12-02 1%
7	Submitted works	islingtoncollege on 2025-01-02 <1%
8	Submitted works	Kaplan International Colleges on 2013-07-16 <1%
9	Submitted works	American Public University System on 2023-02-04 <1%
10	Submitted works	The University of Wolverhampton on 2022-08-15 <1%

11 Submitted works
islingtoncollege on 2025-01-02 <1%

12 Submitted works
University of Greenwich on 2020-05-06 <1%

13 Submitted works
Kaplan Professional on 2025-01-13 <1%

14 Submitted works
Asia Pacific University College of Technology and Innovation (UCTI) on 2021-02-08 <1%

15 Submitted works
Gusto University on 2024-11-14 <1%

16 Submitted works
Informatics Education Limited on 2010-10-28 <1%

17 Submitted works
Toronto Business College on 2023-09-02 <1%

18 Submitted works
Australian Catholic University on 2024-11-09 <1%

19 Submitted works
Monash University on 2024-10-28 <1%

20 Submitted works
Greenwich School of Management on 2014-12-05 <1%

21 Submitted works
Kaplan International Colleges on 2023-07-09 <1%

22 Submitted works
Monash University on 2024-10-28 <1%

23 Submitted works
University of Hertfordshire on 2024-01-02 <1%

24 Submitted works
University of Wales Institute, Cardiff on 2024-10-21 <1%

25 Submitted works
Illinois Institute of Technology on 2024-09-24 <1%

26 Submitted works
Monash University on 2023-08-02 <1%

27 Submitted works
University of Northampton on 2012-01-24 <1%

28 Submitted works
islingtoncollege on 2024-12-30 <1%

29 Submitted works
Monash University on 2024-10-28 <1%

30 Submitted works
University of Sheffield on 2017-04-14 <1%

31 Submitted works
essex on 2019-10-12 <1%

32 Submitted works
Arab Open University on 2024-11-07 <1%

33 Submitted works
Monash University on 2024-10-28 <1%

34 Submitted works
University of Bolton on 2022-08-02 <1%

35 Submitted works
Academy of Information Technology on 2024-03-17 <1%

36 Submitted works
American Intercontinental University Online on 2006-11-02 <1%

37 Submitted works
Intercom Programming & Manufacturing Company Limited (IPMC) on 2012-02-17 <1%

38 Submitted works
The University of Wolverhampton on 2022-08-16 <1%

39 Submitted works
University of Sheffield on 2010-05-03 <1%

40 Submitted works
islingtoncollege on 2024-12-22 <1%

41 Submitted works
IPMC Kumasi on 2024-10-27 <1%

42 Submitted works
Pathfinder Enterprises on 2017-11-29 <1%

43 Submitted works
ICL Education Group on 2025-01-13 <1%

44 Submitted works
Middle East College of Information Technology on 2017-05-04 <1%

45 Submitted works
Pathfinder Enterprises on 2018-01-10 <1%

46 Submitted works
University of Portsmouth on 2023-11-22 <1%

47 Submitted works
University of Wales Institute, Cardiff on 2024-10-20 <1%

48 Submitted works
islingtoncollege on 2024-12-30 <1%

INTRODUCTION

The “EVISION ROOM” database, created by Contech Design Company acts as a data manager or guide for Ms. Mary to keep track of all related objects of the college and achieve its core objectives and goals. It aims to introduce the concept of database management system which involves practice of normalization, use of constraints for an e-classroom platform where the users can experience a user-friendly service. Upon successful completion of this subject, users can get a better understanding of what the business is and how each of its functionalities works.

DATABASE

The increased use in the application of computers is seen due to the introduction of a database approach in areas like business, e-classrooms, libraries, hospitals and many more. A general description of database comes from storing, modifying and managing related data in tabular form i.e. rows and columns where rows are values and columns are attributes. (Ramez Elmasri, 2015) A database can cope with both simplicity and complexity of data in various platforms. To support this, A large commercial database is MyTeacher where millions of active users like students and teachers interact with feedback, study slides for each subject areas and assignments portals.

DATABASE MANAGEMENT SYSTEM

Database is basically an alternative solution to the file-handling system and Database Management system is a platform to interact with databases using SQL (Structured Query Language). (Ramez Elmasri, 2015)` DBMS like Oracle is used in this project where interaction between oracle databases is done by SQL + plus version 11.2.0.2.

Figure 2 Sql Plus Logo

OTHER TOOLS USED

Apart from the actual operational tools used in creating a database “EVISION ROOM”, a few tools were implemented to complete the documentation and create figures to support the statements involved in the project. For the preparation of documentation word (Microsoft 365) is used whereas draw.io is used for tables, diagrams and figure creation.

Figure 4 Word Logo

BUSINESS AND ITS FORTE

“CONTECH designs” is a well-established company which specializes in introducing strong databases suitable for both small and large-scale enterprises. In this case, the company is planning to create a database called “EVISION ROOM” at request to our client “Ms. Mary” who insisted on a system for her e-classroom endeavor and expressed support for the success of the new innings.

Before 20th January of the year 2025, our mission is to make a full-fledge database with integrated functionalities just as Ms. Mary suggested involving access and organization of students, teachers and programs details and their allocation, provision of assessment, and grading results. Additionally, with the help of our database, students and teachers can smoothly interact with various means like feedback, announcements, and assignment markings.

CURRENT SCENARIO AND OPERATIONS

Ms. Mary runs a college “OAK TREE ACADEMY” with total students of 10 from the bachelor’s slot and 5 from master’s slot. The college offers various programs like BSCs in Computing, BSCs in Networking, BSCs in Multimedia and BSCs in Computing with

Artificial Intelligence under IT bachelor's degree whereas IT master's degree offers two programs i.e. MSc in Data Analyst and MSc in Cyber Security with a total of 10 academic staff. The program contains different modules like java, fundamentals of computing, information system and many more. With that, new opportunities have been discussed and proceeded for scholarship awards categorized by academic performance and sport activities based where eligible students must hold dignified achievements in available sports like basketball, football, chess, long tennis etc. However, just after 5 months of college, difficulties in manual handling the records of students, teachers and programs containing various modules resulted in fluctuation of details, tracking problems, delays in updates and lack of proper course enrollments. The proper implementation of a structured and user-friendly environment must be carried out which will further help faculty staff in decision-making and improve the situation of the college's operations. Apart from functionality, data integrity, data redundancy and data security are practiced which Ms. Mary, the principal of the college stated as her top priority.

IDENTIFICATION OF ENTITY AND ATTRIBUTES

In this section, we identify the core objects (entities) which relate to things in the real world and likewise provide properties (attributes) for each of them and the rules the attributes need to follow. According to the "OAK TREE ACADEMY", involvement of both tangible and intangible entities is seen where tangible entities include Student, Teacher, Resource and Program, Module, Assessment, Announcement, Result fall under intangible entities.

CONSTRAINTS

A set of rules termed constraints are applied to necessary attributes of identified entities and limit the data format of a table which further improves the readability and accuracy of data.

Domain constraints → contact_no must be 10 digits strictly.

35 Entity Integrity constraints → Ensures every table has primary key like student_id, program_id and module_code and these values cannot be null.

Referential integrity constraints → ensures consistency between two tables using foreign key identified after normalization.

Key constraints → ensures uniqueness of records apart from primary key like contact_no, teacher_id, email_address.

KEY:

Primary key: it is a selected candidate key or a key attribute which single handily helps to acquire/retrieve all other related attributes(column). (Abraham Silberschatz, 2011)

43 An entity can have one or multiple primary keys called a composite primary key. A primary key must always be unique and not null which itself is a constraint. A primary key is represented by (PK) or underlining.

Foreign key: They act as a connector between entities because foreign key represents

primary key of another entity present in an entity. Usually, a foreign key is indicated as FK or *.

Not null: Nulling or not nulling a value depends on whether the entities are mandatory or optional relations. If an optional relation is seen, values can be null otherwise it cannot be null. NN or double N is used to represent not null.

Unique: Apart from primary keys, candidate keys can have unique data. Uniqueness of an attribute is denoted by U.

Primary key in the listed entities is mentioned below:

ENTITY

PRIMARY KEY

Student

student_id

Program

program_id

Module

module_code

Table 1 Core Entities with key attribute

ENITITY AND ATTRIBUTES

25 An entity is a thing or object in the real world that can be described as such which is

called attributes (GeeksforGeeks, 2023). Entities are recorded in database which can be easily distinguished from groups.

STUDENT

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINTS

1

student_id

Number

7

Primary key

2

first_name

Character

15

Not null

3

middle_name

Character

15

4

last_name

Character

15

Not null

5

home_address

Character

30

Not null

6

contact_no

Number

10

Unique

7

email_address

Character

15

Unique

8

gender

Character

6

33 Not null

9

age

Number

2

Not null

10

enrolled_date

Date

Not null

Table 2 Student entity: attributes

11 PROGRAM

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINTS

1

program_id

Number

7

Primary key

2

program_title

Character

15

Not null

3

program_length

Number

5

Not null

19 4

total_credit

Number

5

Not null

5

tuition_fee

Number

20

Not null

6

entry_requirement

Character

5

Not null

7

maximum_capacity

Number

20

Not null

8

Mode_of_delivery

Character

15

Not null

Table 3 Program entity: attributes

MODULE

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINTS

1

module_code

Number

7

Primary key

2

module_title

Character

15

Unique

3

credit_score

Number

5

Not null

4

duration_period

Number

5

Not null

26 5

academic_level

Number

5

Not null

6

teacher_id

Number

7

Unique

7

first_name

Character

15

Not null

8

middle_name

Character

15

9

2 last_name

2 Character

15

Not null

10

home_address

Character

30

Not null

11

contact_no

Number

10

Unique

12

educational_background

Character

25

Not null

13

years_of_experience

22 Number

5

Not null

22 14

salary

Number

20

Not null

15

joined_date

Date

Not null

16

assessment_id

Number

8

Unique

17

assessment_title

2 Character

15

Not null

18

assigned_date

Date

Not null

29

due_date

Date

Not null

20

category

Character

10

Not null

21

weightage

Number

5

Not null

22

resource_id

Number

8

Unique

23

4

resource_title

Character

15

Not null

24

format_type

Character

10

Not null

25

duration

Number

5

Not null

26

result_code

Number

8

Unique

27

obtained_mark

Number

5

Not null

28

grade

Character

29 5

Not null

29

attempt_no

Number

5

Not null

30

result_released_date

Date

Not null

31

announcement_id

Number

8

Unique

32

notice_headline

Character

50

Not null

33

34 uploaded_date

Date

Not null

34

expired_date

Date

Not null

Table 4 Module entity: attributes

NOTE: The category of attributes is represented in the first row highlighted in grey whereas the second row contains a set of the entity's attributes.

BUSINESS RULE

This section elaborates guidelines of the system's daily operation, assumption made and relation between entities and how they are dependent on each other. With that, it helps to understand the requirements of the system and its core objectives and goals.

BUSINESS RULE FOR THE SYSTEM

A student must enroll in one program only and each program should contain at least one student.

A single module can belong to various programs and a program must contain at least one module.

Every student is required to attend all modules of their respective program.

A teacher can be responsible for single or many modules.

A module is assigned to one or many teachers.

Each teacher must assign assessments for their specific modules only.

A module can include one or many assessments.

Each result is associated with only one module.

Students can view results of only completed assessments.

A resource is associated with a single module.

The programs must be related to information Technology subject only.

Each module can contain at least one resource.

Students must complete a resource to access another resource.

A teacher can make announcements for their specific modules only.

An announcement is associated with one module only.

Each assessment must include attributes like ID, title, deadline and weightage.

Resources must be marked as completed to proceed with the next resource.

The new module may not decide which program or teachers it belongs for the moment.

SCENARIO ASSUMPTIONS

Assumption1: A module is taught by many teachers instead of a single teacher.

Assumption 2: Different programs are associated with common modules.

Assumption 3: Some modules may or may not be assigned to any teacher which generates optional modality.

Assumption 4: Some modules may or may not be associated with a program which also indicates a possibility of optional modality.

Assumption 5: Teacher and student may not have direct relationship but can be indirectly associated with the help of module.

CARDINALITY AND MODALITY

Cardinality is the largest occurrence of entities that are connected to each other whereas modalities describe whether the requirement of relationship between the entities is optional or mandatory. (Sudarshan, n.d.) Usually cardinality involves:

One-to-one (1: 1): This relation shows that at most one instance of two connected entities are associated. It is the rarest relationship to occur.

Figure 6 One-to-one relation

One-to-many (1: N): This relation occurs when at most one instance of entity A is associated with any number of entity B but at most one instance of entity B is associated with at most one instance of entity A (Alalouf, 2018).

Figure 7 One-to-many relation

Many-to-many (M: N): This relation indicates that at most one instance of entity A is assigned with any number of instances of entity B and vice versa. A bridging entity is needed to solve this relationship.

Figure 8 Many-to-many relation

Likewise, a modality can be of two types:

Optional (O): it means that the relationship between two entities may or may not be required.

Mandatory (|): it means that the relationship between two entities is required.

Figure 9 Optional and Mandatory relation

The diagram below shows the actual position of cardinality and modality:

Figure 10 Cardinality and Modality position

NOTE: In crow foot notation one (1) can be represented by | a bar and many(N) as “”.

CONCEPTUAL DATABASE DESIGN

This section describes the importance of data modelling before initiating database development because it represents the requirement of the system like entities (Strong and weak entities), attributes, relations and constraints that were identified in section 2.

The design is based on requirement gathering and data analysis to build a basis for the physical database design. The last part of the section contains initial er-diagram before any modification to give an overview of the core entities.

ENTITY-RELATION DIAGRAM

In the year 1976, the Entity-relation data model was first introduced by Peter Chen to provide a simple and understandable working flow of the system. Later its variation was further used to sketch real-world objects like people, events, non-living things and many more.

An er-diagram is a visual representation of identified entities with their respective attributes and applies connectivity to those entities according to their relation (Brown, 2024). In other words, it displays an overall logical structure of our database “EVISION ROOM”. This data model design contributes to building a blueprint for databases, provides clear vision of dependencies, determines flow of data, and helps in making relational databases.

ENTITY TYPES: STRONG ENTITIES VS WEAK ENTITIES

An entity includes two different types of characteristics that are strong and weak where each type determines their dependencies with one another.

STRONG ENTITIES

Strong entities are those entities which have key attributes and do not need any composite key to uniquely identify them. For example, entities like Student, Teacher, Program, Module. Strong entities are represented by a single line rectangle.

Therefore, listed Strong entities are mentioned below:

Student

Teacher

Program

Module

WEAK ENTITIES

Weak entities are represented by a double line rectangle where they do need a composite key to uniquely identify them and are dependent on string entity. In this case, Assessment, Resource, Result, Announcement are identified as weak entities because they may have key attributes but need a key attribute of strong entities that is Module to identify them uniquely.

Therefore, listed weak entities are mentioned below:

Assessment

Resource

Result

Announcement

A brief description of strong and weak entities:

Figure 11 Strong and Weak Entity Relation

NOTE: The weak and string entities will be shown after normalization in final entity relation diagram

37 INITIAL ER-DIAGRAM WITH CORE ENTITIES

In the initial Er-diagram, our core entities are Student, Program and Module. Inclusion

of all identified entities with the specification of foreign key and bridging entities will be done in logical database design.

Figure 12 Initial Er-Diagram

NORMALIZATION

This section involves a process to avoid data redundancy and ensures each row with unique data. Normalization is done to also solve many-to-many relationships by introducing bridging entities. Additionally, completing and achieving a normalized form further helps in developing final er-diagrams.

TYPE OF DEPENDENCIES IN UNNORMALIZATION

The functional dependencies and transitive dependencies may prevail in unnormalized data where in full functional dependencies, attributes depend on two or more composite keys. On the other hand, the partial functional dependencies means that attributes do not need additional key attributes to be dependent on. Transitive dependencies, another dependent type which works in format of A→B→C where non-key attributes must be dependent on candidate keys which in turn depends on key attribute (Primary key) (geeksforgeeks, 2024). This shows an indirect relationship between non-key attributes and primary key.

47

According to the normalization process, the functional dependencies and transitive

dependencies occur in the second normalization form 2NF and third normalization form 3NF. The notation of functional dependencies is A□B, A,B□C and transitive dependencies is A□B B□C.

NORMALIZATION LEVEL

According to the entities and database involved in our database “EVISION ROOM”, there are four level breakdown of the process that is UNF, 1NF, 2NF, 3NF. To make the process easier and understandable, use of table structure for UNF is done with the help of Microsoft excel tool.

UNF (Unnormalized Form)

In this form, we determine the repeating value and repeating group where repeating values belong to those outside the curly brackets {} but inside small brackets () and repeating groups belong to value inside the curly brackets{}. To identify and differentiate each, use of a table and few row values is practiced.

UNF in brackets:

TABLE(student_id, student_name, stud_address, stud_contact, stud_email, stud_gender, stud_age, enrolled_date, prog_id, prog_title, prog_length, prog_credit, prog_fee, entry_req, max_capacity, mode_of_delivery{mod_code, mod_title, mod_credit, mod_duration, mod_level{teacher_id, tea_name, tea_address, tea_contact, tea_edu, tea_experience, salary, joined_date{ announcement_id, announcement_headline, uploaded_date, expired_date}}, {assessment_id, ass_title, assigned_date, due_date, category, weightage, result_code, obtained_mark, grade, attempt_no, released_date}, {resource_id, resource_title, resource_type, resource_duration}})

1NF (First Normalized Form)

This form operates to separate each repeating groups in different tables (old and new) where old belongs to repeating values and new represent nested repeating groups is present (repeating group inside repeating group) i.e. module, resource, teacher, assessment, resource, announcement, result. Here, assessment and result attributes are kept side by side as result is influenced by each assessment within a module. Also, announcement is a repeating group of teachers which in turn is a repeating group of modules, so it acts as a nested repeating group of modules.

Repeating values in student table

student (student_id, student_name, student_address, student_contact, student_email,

student_gender, student_age, enrolled_date, program_id, program_title, program_length, program_credit, program_fee, entry_requirement, max_capacity, mode_of_delivery)

each repeating and nested repeating groups in separate table

student_module(student_id , module_code, module_title, module_credit, module_duration, module_level)

module_teacher(student_id,module_code, teacher_id, teacher_name, teacher_address, teacher_contact, teacher_educational_background, teacher_experience, salary, joined_date)

teacher_announcement(student_id,module_code,teacher_id,announcement_id, announcement_headline, uploaded_date, expired_date)

module_assessment(student_id, module_code, assessment_id, assessment_title, assigned_date, due_date, category, weightage, submission_status, result_code, obtained_mark, grade, attempt_no, result_released_date)

module_resource(student_id, module_code, resource_id, resource_title, resource_type, resource_duration, completion_status)

2NF (Second Normalized Form)

Since the 1NF is in full functional dependency and does not contain partial dependency in 2NF, we analyze and divide each non-key attributes regarding their dependencies with composite keys.

NOTE: The dependencies can only occur where composite keys reside.

In the student table there are no composite keys and only one primary key called student_id, no occurrence of partial dependencies can be seen. This means student_id gives each non-key attribute uniquely.

1 student (student_id, student_name, student_address, student_contact, student_email,
2 student_gender, student_age, enrolled_date, program_id, program_title, program_length, program_credit, program_fee, entry_requirement, max_capacity, mode_of_delivery)

Here, the dependencies are checked as the student_module table meets the criteria that is (composite keys).

5 student_module(student_id , module_code, module_title, module_credit, module_duration, module_level)

Since module_level is the level year. It is determined by students also as a module can be in different programs which might involve masters and bachelors.

module_code → module_title, module_credit, module_duration

student_id → ×

module_code, student_id → module_level

The breakdown of student_module table is:

module_details(module_code,module_title,module_credit, module_duration)

student_module(student_id , module_code, module_level)

The attributes in module_teacher table have functional dependencies as three composite keys reside with many attributes. All the attributes like teacher_name, teacher_address, teacher_contact, teacher_educational_background, teacher_experience, salary, joined_date are partially dependent on teacher_id.

module_teacher(student_id,module_code,teacher_id,teacher_name, teacher_address, teacher_contact, teacher_educational_background, teacher_experience, salary, joined_date)

teacher_id → teacher_name, teacher_address, teacher_contact, teacher_educational_background, teacher_experience, salary, joined_date

teacher_id, module_code, student_id →×

student_id, teacher_id → ×

teacher_id,module_code → ×

student_id, module_code → ×

student_id → ×

module_code → ×

The table above table after dependence are:

teacher_details(teacher_id,teacher_name,teacher_address, teacher_contact, teacher_edu, teacher_experience, salary, joined_date)

teacher_module_student(teacher_id, module_code, student_id)

The next dependencies are identified in module_announcement table where non key attributes are announcement_headline, uploaded_date, expired_date where all attributes solely depend on only announcement_id.

teacher_announcement(student_id,module_code,teacher_id,announcement_id, announcement_headline, uploaded_date, expired_date)

announcement_id → announcement_headline, uploaded_date, expired_date

student_id, module_code → ×

module_code, announcement_id → ×

teacher_id, announcement_id → ×

teacher_id, module_code → ×

teacher_id, student_id → ×

module_code, student_id, teacher_id → ×

teacher_id, announcement_id, student_id → ×

module_code, teacher_id, announcement_id → ×

module_code, student_id, announcement_id → ×

announcement_id, student_id → ×

student_id → ×

module_code → ×

teacher_id → ×

student_id, module_code, teacher_id, announcement_id → ×

The table below are after checking the dependencies:

announcement_details(announcement_id,announcement_headline, uploaded_date, expired_date)

student_module_teacher_announcement(student_id,module_code, teacher_id, announcement_id)

The module_assessment is now checked which includes attributes of both assessment

and result where all attributes and result released date depends on assessment whereas the result_code, obtained_marks, grade, attempt_no depends equally on result_code and student_id, module_code and assessment_id.

module_assessment(student_id, module_code, assessment_id, assessment_title, assigned_date, due_date, category, weightage, submission_status, result_code, obtained_mark, grade, attempt_no, result_released_date)

assessment_id → assessment_title, assigned_date, due_date, category, weightage, result_released_date

student_id, module_code, assessment_id → result_code, obtained_mark, grade, attempt_no, submission_status

student_id, module_code → ×

module_code, assessment_id → ×

student_id, assessment_id → ×

student_id → ×

module_code → ×

The creation of two tables is seen where one has primary key assessment_id which includes only assessment details, another is composite key of assessment_id, student_id, module_code and result_code which showcase result details connected to student_id, module_code and assessment_id.

assessment_details(assessment_id,assessment_title,assigned_date, due_date, category, weightage, result_released_date)

student_module_assessment (student_id, module_code, assessment_id, result_code, obtained_mark, grade, attempt_no, submission_status)

The last check is done on module_resource table where non key attributes like resource_title, resource_type, resource_duration depend on only resource_id whereas completion_status depend on student_id, resource_id as the resource completion depends on each student's sequential process.

module_resource(student_id, module_code, resource_id, resource_title, resource_type, resource_duration, completion_status)

resource_id → resource_title, resource_type, resource_duration

student_id → ×

mod_code → ×

resource_id, student_id → ×

student_id, mod_code → ×

mod_code, resource_id → ×

resource_id, student_id, module_code → completion_status

The creation of the previous table is divided into two separate table where the first table contains only resource details, the next table is to track the status of student for a resource in a module.

resource_details(resource_id, resource_title, resource_type, resource_duration)

student_module_resource(resource_id, student_id, module_code, completion_status)

So, the identified tables from 1NF to 2NF are mentioned as follows:

student (student_id, student_name, student_address, student_contact, student_email, student_gender, student_age, enrolled_date, program_id, program_title, program_length, program_credit, program_fee, entry_requirement, max_capacity, mode_of_delivery)

module_details(module_code,module_title,module_credit, module_duration)

student_module(student_id , module_code, module_level)

teacher_details(teacher_id,teacher_name,teacher_address, teacher_contact, teacher_educational_background, teacher_experience, salary, joined_date)

teacher_module_student(teacher_id, module_code, student_id)

announcement_details(announcement_id,announcement_headline, uploaded_date, expired_date)

student_module_teacher_announcement(student_id,module_code, teacher_id,

announcement_id)

assessment_details(assessment_id,assessment_title,assigned_date, due_date, category, weightage, result_released_date)

3 student_module_assessment (student_id, module_code, assessment_id, result_code, obtained_mark, grade, attempt_no, submission_status)

5 resource_details(resource_id,resource_title,resource_type, resource_duration)

student_module_resource(resource_id, student_id, module_code, completion_status)

3NF (Third Normalized Form))

The last form but not the least implemented form is the 3NF where transitive dependencies are identified like A → B → C is checked and separated in form A→B B→C where A = primary key attribute, B = candidate key attribute and C is a non-key attribute.

NOTE: The occurrence can only be checked on tables where non key attributes are more than one.

student (student_id, student_name, student_address, student_contact, student_email, student_gender, student_age, enrolled_date, program_id, program_title, program_length, program_credit, program_fee, entry_requirement, max_capacity, mode_of_delivery)

The transitive dependencies in the student table occur as a candidate key like program_id uniquely gives other non-key attributes like program_title, program_length, program_credit, program_fee, entry_requirement, max_capacity, mode_of_delivery.

student_id → program_id and program_id → program_name, program_credit, program_fee, entry_req, max_capacity, mode_of_delivery so A→B B→C.

student_details(student_id,student_name,student_address, student_contact,student_email,student_gender,enrolled_date, program_id*)
program_details(program_id,program_title,program_length, program_credit,program_fee,entry_requirement,max_capacity, mode_of_delivery)
module_details(module_code,module_title,module_credit, module_duration)

All non-key attributes directly depend on module_code so there is no transitive dependency.

teacher_details(teacher_id,teacher_name,teacher_address, teacher_contact, teacher_educational_background, teacher_experience, salary, joined_date)

All non-key attributes depend upon teacher_id so there is no transitive dependency.

announcement_details(announcement_id,announcement_headline, uploaded_date, expired_date)

All the non-key attributes depend on announcement_id so it is not in transitive dependency.

assessment_details(assessment_id,assessment_title,assigned_date, due_date, category, weightage, result_released_date)

All non-key attributes depend on directly to assessment_id so there are no transitive keys.

student_module_assessment(student_id,module_code, assessment_id, result_code, obtained_mark, grade, attempt_no, submission_status)

The grade is determined by obtained_mark and obtained_mark is dependent on result_code which in turn is dependent on composite keys student_id, module_code and assessment_id so there is a transitive dependency of student_id, module_code,

assessment_id → result_code → obtained_mark and obtained_mark → grade so A→B B→C.

student_module_assessment(student_id,module_code, assessment_id, result_code*,submission_status, attempt_no)

result_details(result_code, obtained_mark*)

grade_details(obtained_mark, grade)

resource_details(resource_id,resource_title,resource_type, resource_duration)

All non-key attributes directly depend on resource_id so there is no presence of transitive dependency.

Some tables do not need to be checked for transitive dependencies as they do not fit in with the criteria of more than one non-key attribute. The below mentioned tables are student_module, teacher_module_student,

student_module(student_id , module_code, module_level)

teacher_module_student(teacher_id, mod_code, student_id)

student_module_teacher_announcement(student_id,module_code, teacher_id, announcement_id)

student_module_resource(resource_id, student_id, module_code, completion_status)

The final tables after 3NF are:

student_details(student_id,student_name,student_address,
student_contact,student_email, student_gender, enrolled_date, program_id*)

program_details(program_id,program_title,program_length,
program_credit,program_fee,entry_requirement,max_capacity, mode_of_delivery)

module_details(module_code,module_title,module_credit, module_duration)

student_module(student_id , module_code, module_level)

teacher_details(teacher_id,teacher_name,teacher_address,teacher_contact,
teacher_educational_background, teacher_experience, salary, joined_date)

teacher_module_student(teacher_id, module_code, student_id)

announcement_details(announcement_id,announcement_headline, uploaded_date,
expired_date)

student_module_teacher_announcement(student_id,module_code, teacher_id,
announcement_id)

assessment_details(assessment_id,assessment_title,assigned_date, due_date,
category, weightage, result_released_date)

student_module_assessment(student_id,module_code,assessment_id,
result_code*,submission_status, attempt_no)

result_details(result_code, obtained_mark*)

result_grade (obtained_mark, grade)

resource_details(resource_id,resource_title,resource_type, resource_duration)

student_module_resource(resource_id,student_id,module_code, completion_status)

DATA DICTIONARY

After normalization, each data element like title, type, attributes of the entities are stored in the tabular form as meta data of the database. The data dictionary also helps to locate relationships between entities (geeks, 2024).

META DATA

Table → Students

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINT

DESCRIPTION

1

student_id

Character

7

Primary key

Unique identifier for each student

2

first_name

Character

15

Not null

First name of student

3

middle_name

Character

15

17 Optional middle name of student

4

last_name

Character

15

Not null

Last name of student

5

home_address

Character

30

Not null

Residence of student

6

contact_no

Number

10

Unique

Numeric value representing phone number

7

email_address

Character

15

Unique

Mailing address of student

8

gender

Character

6

Not null

Gender of student

9

age

Number

2

Not null

Numeric value representing age of students

10

enrolled_date

Date

Not null

Date when student enrolled

11

program_id

Character

7

Foreign key

Unique identifier for each program

Table 5 Table: Students

46 Where previous student_name = first_name + middle_name + last_name.

Table → Programs

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINT

DESCRIPTION

1

program_id

Character

7

Primary key

Unique identifier for each program

2

program_title

Character

15

Not null

Name of the program

3

program_length

Number

5

Not null

duration to complete the program

4

total_credit

Number

5

Not null

Overall numeric value for credit of program

5

tuition_fee

Number

20

Not null

Total amount to study the program

6

entry_requirement

Character

15

Not null

criteria to take the program

7

maximum_capacity

Number

20

Not null

Numeric value for total program intake

8

Mode_of_delivery

Character

15

Not null

Medium to study the program

Table 6 Table: Programs

4 Table → Modules

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINT

DESCRIPTION

1

module_code

Character

7

Primary key

6

Unique identifier of module

2

module_title

Character

15

Unique

Name of module

3

credit_score

Number

5

Not null

Credit of module

4

duration_period

Number

5

Not null

Duration to complete each module

Table 7 Table: Modules

7

Table → student_module

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINT

DESCRIPTION

1

student_id

Character

7

Foreign key

Reference to unique identifier of student table’s student_id

2

module_code

Character

7

Foreign key

Reference to unique identifier of module table’s module_code

3

module_level

Number

5

Not null

Year level category of module

Table 8 Relation Table: student_module

Table → Teachers

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINT

DESCRIPTION

1

teacher_id

Character

7

Primary key

Unique identifier of teachers

2

first_name

Character

15

Not null

First name of teacher

3

middle_name

Character

15

Optional middle name of teacher

4

last_name

Character

15

Not null

last name of teacher

5

home_address

Character

30

Not null

Residence of teacher

6

contact_no

Number

10

Unique

Numeric value representing phone number

7

educational_background

Character

25

Not null

Mailing address of teacher

8

years_of_experience

Number

5

Not null

Numeric value representing working experience of teacher

9

salary

Number

20

Not null

Numeric value representing salary of teacher

10

joined_date

Date

Not null

Date when teacher joined

Table 9 Table: Teachers

Table → student_module_teacher

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINT

DESCRIPTION

COMPOSITE CONSTRAINT

1

student_id

Number

7

Foreign key

Reference to unique identifier of student table’s student_id

4 Primary key

2

module_code

Number

7

Foreign key

Reference to unique identifier of module table’s module_code

3

teacher_id

Number

7

Foreign key

Reference to unique identifier of teacher table’s teacher_id

Table 10 Relation Table: student_module_teacher

Table → Announcements

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINT

DESCRIPTION

1

announcement_id

Character

8

Primary key

Unique identifier of announcement

2

notice_headline

Character

100

Not null

subject of announcement

3

uploaded_date

Date

Not null

Date when uploaded

4

expired_date

Date

Not null

Date when expired

Table 11 Relation Table: Announcements

Table → student_module_teacher_announcement

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINT

DESCRIPTION

2 COMPOSITE CONSTRAINT

1

student_id

Character

7

Foreign key

Reference to unique identifier of student table’s student_id

Primary key

2

module_code

Character

7

Foreign key

Reference to unique identifier of module table’s module_code

3

teacher_id

Character

7

Foreign key

Reference to unique identifier of teacher table’s teacher_id

4

announcement_id

Character

8

Foreign key

Reference to unique identifier of announcement table’s announcement_id

Table 12 Relation Table: student_module_teacher_announcement

Table → Assessments

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINT

DESCRIPTION

1

assessment_id

Character

8

Primary key

Unique identifier of assessment

2

assessment_title

Character

20

Not null

Name of assessment

3

assigned_date

Date

Not null

Date when assigned

4

due_date

Date

Not null

Deadline of assessment

5

category

Character

10

Not null

Type of assessment

6

weightage

Number

5

Not null

Numeric value for percentage of assessment

7

released_date

Date

Not null

Released date of result for assessment

Table 13 Table: Assessments

4

Table → Results

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINT

DESCRIPTION

1

result_code

Character

8

Primary key

Unique identifier of result

2

obtained_mark

Number

5

Foreign key

References result_grade table obtained_mark

Table 14 Table: Results

Table → result_grade

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINT

DESCRIPTION

1

obtained_mark

Number

5

Primary key

Assessment mark of student

2

grade

Character

5

Not null

Grade marked by obtained mark

Table 15 Relation Table: result_grade

Table → student_module_assessment

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINT

DESCRIPTION

COMPOSITE CONSTRAINT

1

student_id

Character

7

Foreign key

Reference to unique identifier of student table’s student_id

Primary key

2

module_code

Character

7

Foreign key

Reference to unique identifier of module table’s module_code

3

assessment_id

Character

8

Foreign key

Reference to unique identifier of assessment table’s assessment_id

4

result_code

Character

8

Foreign key

Reference to unique identifier of result table’s result_code

5

Submission_status

Character

10

Not null

Assessment status by student

6

attempt_no

Number

5

Not null

Numeric value representing attempt made by student for an assessment

Table 16 Relation Table: student_module_assessmen_resultt

Table → Resources

S.NO

ATTRIBUTE NAME

DATA TYPE

6 SIZE

CONSTRAINT

DESCRIPTION

1

resource_id

Character

8

Primary key

Unique identifier of resource

4 2

resource_title

Character

30

Not null

Name of resource

3

format_type

Character

15

Not null

Type of resource

4

duration

Number

5

Not null

Duration of resource

Table 17 Table: Resources

Table → student_module_resource

S.NO

ATTRIBUTE NAME

DATA TYPE

SIZE

CONSTRAINT

DESCRIPTION

COMPOSITE CONSTRAINT

1

student_id

Character

7

Foreign key

Reference to unique identifier of student table’s student_id

Primary key

2

module_code

Character

7

Foreign key

Reference to unique identifier of module table’s module_code

3

resource_id

Character

8

Foreign key

Reference to unique identifier of resource table’s resource_id

4

completion_status

Character

15

Not null

Resource completion status by student

Table 19 Relation Table: student_module_resource

FINAL ER-DIAGRAM

Figure 13 Final er diagram

NOTE: Blue is an individual entity whereas Yellow is connecting entities.

IMPLEMENTION

In this section, physical database design is done with the help of provided tools. The main operations are done which involve SELECT, CREATE, UPDATE AND INSERT. The final tables created after normalization is implemented into the “EVISION ROOM” database and each valid data are inserted into rows to main referential integrity.

SQL KEYWORD

Usage of predefined keywords can be seen while developing the physical design database. The main basic keywords are CONNECT and RESOURCE in DCL apart from other SQL languages like DDL, DML, TCL which helps in REVOKE and GRANT permissions. DDL involves keywords to create a structural base of table, DML provides keyword which helps to manipulate data in the table whereas TCL is to record a transaction done in transaction query.

INITIAL STEP FOR CREATION OF EVISION ROOM DATABASE

The creation of individual plus bridging entities table is done to avoid redundancy, increase data integrity and accuracy.

CREATE, GRANT AND CONNECT TO USER

The process of creating a user who handles the further steps of table creation, data insertion and implementing transaction queries need to be done by the system which is the in-between users in term of access to the system. The system then grants permission with the required sql keyword (mentioned in 7.2 SQL KEYWORD) and proceeds to the next step that is table creation.

PRE-WRITTEN ROUGH NOTEPAD:

Figure 14 Screenshot: Notepad to grant permissions

ACTUAL IMPLEMENTATION:

Figure 15 Screenshot: connect to system

Figure 16 Screenshot: user and password creation

Figure 17 Screenshot: grant permission

Figure 18 Screenshot: Connect to created user

REQUIRED ENTITIES TABLE CREATION

After connecting to the required user interface, we create the structural base of the tables needed for required entities and their relations with one another. This section involves the tables identified after normalization 3NF which is implemented with help of creating syntax.

PRE-WRITTEN ROUGH NOTEPAD:

Figure 19 Screenshot: Notepad to create individual entity tables

Figure 20 Screenshot: Create table programs and students

Figure 21 Screenshot: Create table modules, teachers and announcements

Figure 22 Screenshot: create table assessments, resources and result_grades

Figure 23 Screenshot: Create table results

PRE-WRITTEN ROUGH NOTEPAD:

Figure 24 Screenshot: Notepad to create bridging entity tables

27 Figure 25 Screenshot: Create table student_module and student_module_teacher

27

Figure 26 Screenshot: Create table student_module_teacher_announcement and student_module_assessment

Figure 27 Screenshot: Create table: student_module_resource

Finalized all tables:

PRE-WRITTEN ROUGH NOTEPAD:

Figure 28 Screenshot: Finalized implemented entities

INSERT ROWS INTO TABLES

PRE-WRITTEN ROUGH NOTEPAD:

Figure 29 Screenshot: Notepad to individual entities insertion

TABLE students:

PRE-WRITTEN ROUGH NOTEPAD:

Figure 30 Screenshot: Notepad to insert into students

Figure 31 Screenshot: Actual implementation of students insertion

Figure 32 Screenshot: update student values

TABLE programs:

PRE-WRITTEN ROUGH NOTEPAD:

Figure 33 Screenshot: Notepad to insert program values

Figure 34 Screenshot: Actual implementation of programs insertion

Figure 35 Screenshot: Alter program columns

TABLE modules:

PRE-WRITTEN ROUGH NOTEPAD:

Figure 36 Screenshot: Notepad to insert modules values

Figure 37 Screenshot: Actual implementation of modules insertion

TABLE teachers:

PRE-WRITTEN ROUGH NOTEPAD:

Figure 38 Screenshot: Notepad to insert teachers values

Figure 39 Screenshot: Actual implementation of teachers insertion

TABLE announcements:

PRE-WRITTEN ROUGH NOTEPAD:

Figure 40 Screenshot: Notepad to insert announcement values

Figure 41 Screenshot: Actual implementation of announcements insertion

TABLE assessments:

PRE-WRITTEN ROUGH NOTEPAD:

Figure 42 Screenshot: Notepad to insert assessment values

Figure 43Screenshot: Actual implementation of assessments insertion

Figure 44 Alter assessment columns

TABLE result_grade:

PRE-WRITTEN ROUGH NOTEPAD:

Figure 45 Screenshot: Notepad to insert results_grade values

Figure 46 Screenshot: Actual implementation of results_grade insertion

TABLE: Results

PRE-WRITTEN ROUGH NOTEPAD:

Figure 47 Screenshot: Notepad to insert into results values

Figure 48 Screenshot: Actual implementation of results insertion

TABLE resources:

PRE-WRITTEN ROUGH NOTEPAD:

Figure 49 Screenshot: Notepad to insert into resource values

Figure 50 Screenshot: Actual Implementation of resource insertion

Figure 51Alter resources columns

TABLE student_module:

PRE-WRITTEN ROUGH NOTEPAD:

Figure 52 Screenshot: Notepad to insert into student_module values

Figure 53 Screenshot: Actual implementation of student_module insertion

TABLE student_module_teacher_announcement:

PRE-WRITTEN ROUGH NOTEPAD:

Figure 54 Screenshot: Notepad to insert into student_module_teacher values

Figure 55 Screenshot: Actual implementation of student_module_teacher insertion

TABLE student_module_teacher:

PRE-WRITTEN ROUGH NOTEPAD:

Figure 56 Screenshot: Notepad to insert into student_module_teacher values

Figure 57 Screenshot: Actual implementation of student_module_teacher insertion

TABLE student_module_assessment:

PRE-WRITTEN ROUGH NOTEPAD:

Figure 58 Screenshot: Notepad to insert into student_module_assessment values

Figure 59 Screenshot: Actual implementation of student_module_assessment insertion

TABLE student_module_resource:

PRE-WRITTEN ROUGH NOTEPAD:

Figure 60 Screenshot: Notepad to insert into student_module_resource values

Figure 61 Screenshot: Actual implementation of student_module_resource insertion

SELECT ALL DETAILS FROM ALL TABLES

After inserting the required data(rows) in the tables. To view the details, SELECT keyword is used to view the details of every column.

SELECT * FROM programs

Figure 62 Screenshot: Select programs table

SELECT * FROM students

Figure 63 Screenshot: Select students table

SELECT * FROM modules

Figure 64 Screenshot: Select modules table

SELECT * FROM teachers

Figure 65 Screenshot: Select teachers table

SELECT * FROM announcements

Figure 66 Screenshot: Select announcements table

SELECT * FROM assessments

Figure 67 Screenshot: Select assessments table

SELECT * FROM resources

Figure 68 Screenshot: Select resources table

SELECT * FROM results

Figure 69 Screenshot: Select results table

SELECT * FROM result_grades

\

Figure 70 Screenshot: Select result_grades table

SELECT * FROM student_module

Figure 71 Screenshot: Select student_module table

SELECT * FROM student_module_teacher

Figure 72 Screenshot: Select student_module_teacher table

SELECT * FROM student_module_teacher_announcement

Figure 73 Screenshot: Select student_module_teacher_announcement table

SELECT * FROM student_module_assessment

Figure 74 Screenshot: Select student_module_assessment table

SELECT * FROM student_module_resource

Figure 75 Screenshot: Select student_module_resource table

DROP TABLES FROM USER

PRE-WRITTEN ROUGH NOTEPAD:

Figure 76 Screenshot: Notepad to drop tables

Figure 77 Screenshot: Actual implementation of drop student_module_resource and student_module_assessment tables

Figure 78 Screenshot: Actual implementation of drop smt_announcement, student_module_teacher, student_module, resources and assessments tables

Figure 79 Screenshot: Actual implementation of drop results, result_grades, announcements, teachers, modules, students, programs tables

DUMP FILE CREATION AND OTHER COMMANDS

Figure 80 Screenshot: Dump file creation

Figure 81 Screenshot: Dump file in folder

Figure 82 Screenshot: set linesize and pagesize

SCREENSHOT OF QUERIES APPLIED IN DATABASE

The queries are divided into two sub-parts information and transaction queries where information queries involve retrieving data from the table like a QnA whereas the transaction queries are to follow TCL (Transaction control language) while retrieving the data.

INFORMATION QUERY

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

PRE-WRITTEN ROUGH NOTEPAD:

14 Figure 83 Screenshot: Notepad to information query 1

14 Result:

Figure 84 Screenshot: Result of information query 1

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

Figure 85 Screenshot: Notepad to information query 2

Figure 86 Screenshot: Result of information query 2

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

Figure 87 Screenshot: Notepad to information query 3

Figure 88 Screenshot: Result of information query 3

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

Figure 89 Screenshot: Notepad to information query 4

Figure 90 Screenshot: Result of information query 4 part 1

Figure 91 Screenshot: Result of information query 5 part 2

List all the teachers who teach more than one module.

Figure 92 Screenshot: Notepad to information query 5

Figure 93 Screenshot: Result of information query 5

TRANSACTION QUERY

Identify the module that has the latest assessment deadline.

Figure 94 Screenshot: Notepad to transaction query 1

Figure 95 Screenshot: Result of transaction query 1

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

Figure 96 Screenshot: Notepad to transaction query 2

Figure 97 Screenshot: Result of transaction query 2

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

Figure 98 Screenshot: Notepad to transaction query 3

Figure 99 Screenshot: Result of transaction query 3

1 List the students who have scared above the average score in the ‘Databases’ module.

Figure 100 Screenshot: Notepad to transaction query 4

Figure 101 Screenshot: Result of transaction query 4

1 Display whether a student has passed or failed as remarks as per their total aggregate marks obtained in a particular module. (total obtained marks equal or above 40 is marked as pass, below 40 or null values are marked as fail.)

14 Figure 102 Screenshot: Notepad to transaction query 5

14 Figure 103 Screenshot: Result of transaction query 5 part 1

Figure 104 Screenshot: Result of transaction query 5 part 2

CRITICAL EVALUATION

This section evaluates the usages and relation of the database project with other subjects. The aim is to critically analysis, evaluate and understand more about the database project.

USAGE

The database system developed for this coursework performs excellent yet simple and user-friendly interface ensuring ease of use even for non-programmer users. In

addition, the SQL language is a simple language and like English which is why the retrieval of data can be done with few practices.

With the help of referential constraints and keys, the implementation of data integrity and minimized redundancy is seen which leads to saving storage and reducing manual errors.

In addition, performing multiple testing resulted in less unexpected errors as the problems were solved.

RELATION WITH OTHER SUBJECTS

The database was a creation of mixed knowledge and logic applied from different subjects which directly related to it whereas some indirectly.

PROGRAMMING (JAVA, PYTHON)

The implementation of queries and constraint validation like NUMBER for age, CHARACTER for genders acts like a data type just as in java and python programming.

MATHEMATICS (SETS/JOINS)

The queries were mostly made with the help of joints which were derived from sets used in mathematic calculations. The join combines two or more tables into one

categorized by union, left, right, inner etc (GeeksforGeeks, 2024).

Figure 105 joins implemented

PROJECT MANAGEMENT

The management of the project started from breaking the task into smaller milestones to highlight important tasks and to complete within the time frame. It was practiced making the project successful, complete and functional.

Abraham Silberschatz, H. F. K. S. S., 2011. Database System Concepts. 6th ed. New York: McGraw-Hill Companies.

Alalouf, A., 2018. Cardinality in Data Modelling, Philadelphia: Temple University.

Brown, F., 2024. Guru99. [Online] Available at: https://www.guru99.com/er-diagram-tutorial-dbms.html[Accessed 21 January 2025].

GeeksforGeeks, 2023. Entity in DBMS. [Online] Available at: https://www.geeksforgeeks.org/entity-in-dbms/[Accessed 21 January 2025].

GeeksforGeeks, 2024. Joins in DBMS. [Online] Available at: https://www.geeksforgeeks.org/joins-in-dbms/[Accessed 23 January 2025].

geeksforgeeks, 2024. What is functional dependencies in dbms?. [Online] Available at:

https://www.geeksforgeeks.org/what-is-functional-dependency-in-dbms/[Accessed 29 dec 2024].

13 geeks, g. f., 2024. What is data dictionary in dbms?. [Online] Available at: https://www.geeksforgeeks.org/what-is-data-dictionary/[Accessed 30 dec 2024].

23 Ramez Elmasri, S. B. N., 2015. Fundamentals Of Database Systems. 7th ed. Texas: Pearson.

32 Sudarshan, K. a., n.d. Chapter-6: Database Design Using the E-R Model. In: K. a.

41 Sudarshan, ed. Database System Concepts. s.l.:Silberschatz, p. 80.