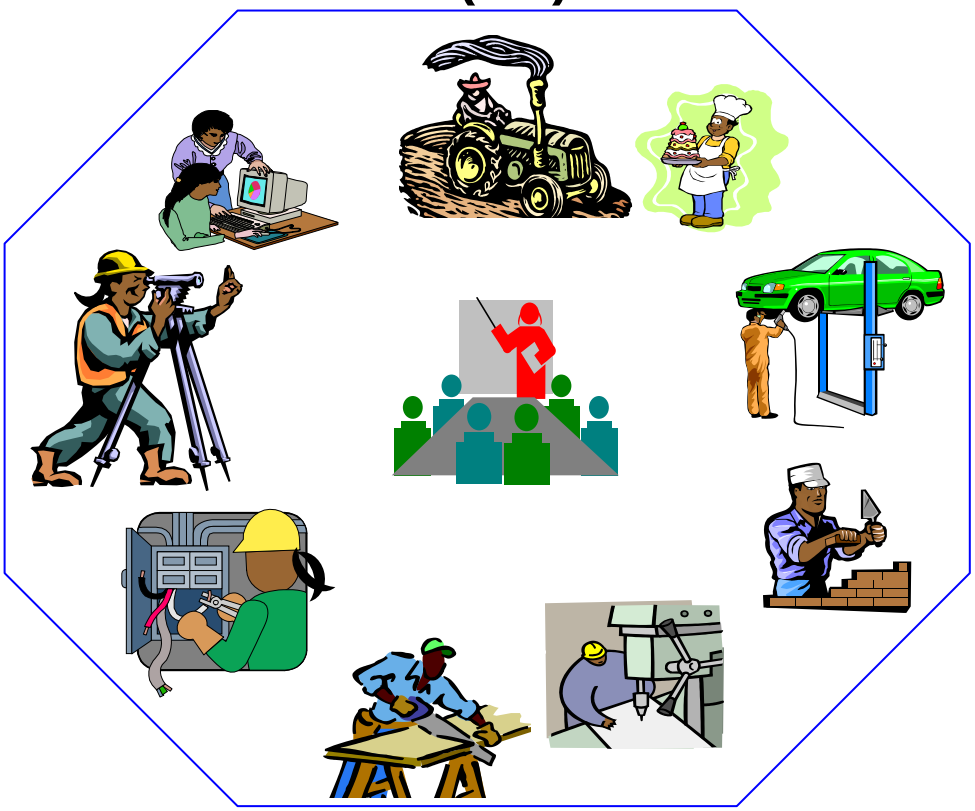
Database Administration Level III

Based on August, 2011, Version 3 Occupational Standards (OS) and Curriculum



Module Title: Designing Database

LG code: EIS DBA3 M07 1220 LO (1-6) (LG24-29)

TTLM Code: EIS DBA3 TTLM07 1220 V1

December, 2020

Bishoftu, Ethiopia

Contents

L #24	LO #1- Determine database requirements

Instruction sheet

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:

- Conducting client need analysis
- Determining database functionality
- Analyzing user needs to identify technical requirements
- Developing conceptual model.
- submitting conceptual model to client for review
- Evaluating client feedback and making changes.

This guide will also assist you to attain the learning outcomes stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Conduct client need analysis
- Determine database functionality
- Analyze user needs to identify technical requirements
- Develop conceptual model.
- submit conceptual model to client for review
- Evaluate client feedback and making changes.

Learning Instructions:

Read the specific objectives of this Learning Guide.

1. Follow the instructions described below.
2. Read the information written in the “Information Sheets”. Try to understand what are being discussed. Ask your trainer for assistance if you have hard time understanding them.
3. Accomplish the “Self-checks” which are placed following all information sheets.
4. Ask from your trainer the key to correction (key answers) or you can request your trainer to correct your work. (You are to get the key answer only after you finished answering the Self-checks).
5. If you earned a satisfactory evaluation proceed to the next learning guide.

Information sheet 1: Conducting client need analysis

1.1. Conducting client need analysis

Developing any system begins with the identification of business or client need since requirements are imperative to the successful design of any system.

Understanding user requirements is an integral part of systems design and is critical to the success of the system. It determines what the users need to do using the system.

Needs analysis is a formal process of identifying and evaluating how a product or service addresses the needs of a human. It involves discovering and documenting the goals a user has and the capabilities needed to assist the user in meeting those goals.

The activities involved in a user need analysis include:

- Identifying and understanding the business's goals
- Clarifying the business problem
- Identifying Stakeholders
- Capturing any key concerns pertaining to the business's successes
- Articulating strategic direction of the business
- Understanding the target audience and their typical tasks
- Understanding specific constraints
- Identifying the challenges, risks, etc.

1.1.1. Characteristics of Requirements

Requirements should have certain characteristics in order to be effective. Writing the requirements well is as important as determining the right ones. Requirements should be:

- **Solution independent-** This means that requirements should not specify a solution to the problem; they should specify what needs to be done, but not how it will be done.
- **Complete-** Requirements must cover all areas of concern, including all phases of the product life cycle.
- **Clear-** Requirements should not leave anyone guessing what is required.
- **Concise-** Unnecessary requirements should be omitted, and the wording of requirements should be concise. Do not bury the requirement in unneeded text.

- **Testable-** When possible, quantitative a (numerical) limit, tolerances, ranges and intended values should be indicated. Testable requirements can be measured in order to determine if the design goal is met.
- **Traceable-** It should be possible to trace a requirement back to the rationale from which it was derived and forward to its implementation in the design.

1.1.2. Benefits of Requirement

Majority of project failures can relate back to poor requirement definitions. Well-defined requirements can provide the numerous benefits. Some of them are;

- Increase productivity
- Enhanced quality of work
- Reductions in support and training costs
- Improve user satisfaction
- Facilitate better planning and less back-tracking down
- Promote accurate Cost and schedule estimation
- Smoother project integration
- Smooth the progress of test and Validation, etc.

1.1.3. Steps in Needs Analysis

Step1: Identify users and uses of the need analysis

Step2: Describe the target population and the service environment.

Step3: Identify needs.

Step4: Assess the importance of the needs

Step5: Communicate results

1.1.4. Techniques for Need Analysis

Several basic Needs Assessment techniques include:

- Direct observation.
- Questionnaires.
- Consultation with persons in key positions, and/or with specific knowledge.
- Review of relevant literature.
- Interviews.
- Assessments/surveys.
- Records & report studies.

Self-Check -1 : Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

I. Choose the best answer (2 pts each)

1. Good designed database _____
   A. May perform poorly
   B. May be hard to maintain
   C. May not be scalable
   D. All
   E. none
2. One is correct about poorly designed databases _____
   A. May not be scalable
   B. May not do what they are supposed to do
   C. May result in incorrect information
   D. All
   E. None
3. Requirements must cover all areas of concern, including all phases of the product life cycle. This can shows the_____ of requirement.
   A. Completeness B. Testability C. Traceability D. Conciseness

II. Short Answer

1. _____ determines what the users need to do using the system.(1pts)
2. List down at least three benefits of well defined requirements (3 pts)

i._____

ii. _____

iii. _____

Note: Satisfactory rating – 5 and 10 points Unsatisfactory - below 5 points

Answer Sheet

Name: _____ Date: _____ Score = _____

Information sheet 2: Database functionality

2.1. Database functionality

Database functionality implies the operations or activities that can be performed using the database. The functionality of database depends on the need of business as well as end user. So you must have to specify what they want to be able to do with the database.

However, a database program must be able to:

- Store very large numbers of records efficiently (they take up little space).
- Easily find information.
- add new data, to edit or delete unwanted information
- Arrange or sort the data as desired easily
- Import or export data into/from other applications
- access the same database at the same time(concurrently)
- strengthen security issues,
- facilitate the Backup and Recovery process, etc

There are several functions that a DBMS performs to ensure the integrity and consistency of data in the database. Some of them are**:**

2.1.1. Data Dictionary Management

Data Dictionary is where the DBMS stores definitions of the data elements and their relationships (metadata). The DBMS uses this function to look up the required data component structures and relationships. The Data Dictionary is often hidden from the user and is used by Database Administrators and Programmers.

2.1.2. Data Storage Management

This is used for the storage of data and any related data entry forms or screen definitions, report definitions, data validation rules, procedural code, and structures that can handle video and picture formats. Users do not need to know how data is stored or manipulated.

2.1.3. Data Transformation and Presentation

This function exists to transform any data entered into required data structures. By using the data transformation and presentation function the DBMS can determine the difference between logical and physical data formats.

2.1.4. Security Management

It is the most important functions in the DBMS. Security management sets rules that determine specific users that are allowed to access the database. Users are given a username and password or sometimes through biometric authentication (such as a fingerprint or retina scan).

2.1.5. Multiuser Access Control

Multiuser access control is a very useful tool in a DBMS, it enables multiple users to access the database simultaneously without affecting the integrity of the database.

2.1.6. Backup and Recovery Management

Backup management refers to the data safety and integrity.

2.1.7. Data Integrity Management

The DBMS enforces these rules to reduce things such as data redundancy, which is when data is stored in more than one place unnecessarily, and maximizing data consistency, making sure database is returning correct/same answer each time for same question asked.

2.1.8. Database Communication Interfaces

This refers to how a DBMS can accept different end user requests through different network environments. A DBMS can provide access to the database using the Internet through Web Browsers (Mozilla Firefox, Internet Explorer, Netscape).

2.1.9. Transaction Management

Transaction Management refers to how a DBMS must supply a method that will guarantee that all the updates in a given transaction are made or not made. All transactions must follow what is called the ACID properties.

- **A – Atomicity**: states a transaction is an indivisible unit that is either performed as a whole and not by its parts, or not performed at all. It is the responsibility of recovery management to make sure this takes place.
- **C – Consistency:** A transaction must alter the database from one constant state to another constant state.
- **I – Isolation:** Transactions must be executed independently of one another. Part of a transaction in progress should not be able to be seen by another transaction.
- **D – Durability:** A successfully completed transaction is recorded permanently in the database and must not be lost due to failures.

Self-Check -2	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not.

I. Choose the best answer (2 pts each)

1. Which one of the following states that a transaction is an indivisible unit
   A. Atomicity
   B. Consistency
   C. Isolation
   D. Durability
2. _____is where the DBMS stores definitions of the data elements and their relationships
   A. Data Dictionary
   B. Backup and Recovery
   C. Interface
   D. None
3. A successfully completed transaction is recorded permanently in the database. This shows _____ of the transaction.
   A. Consistency
   B. Atomicity
   C. Isolation
   D. Durability
4. A transaction alters the database from one constant state to another constant state.
   A. True
   B. False
   C. Unknown

Note: **Satisfactory rating – 4 and 8 points** **Unsatisfactory - below 4 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Name: _____ Date: _____

Information sheet 3: Analyzing user needs to identify technical requirements

3.1. Identifying technical requirements

Technical requirements are the factors required to deliver a desired function or services from a system to satisfy a user's standards and needs.

Once you have assessed the business problem and identified business and user requirements the next step is to identify the technical requirements for the database.

These are aspects such as performance, reliability, and availability that your project must meet. These types of requirements are often called quality of service (QoS) requirements, service-level requirements or non-functional requirements.

Identifying technical requirements enable the business or process to meet expectations.

Technical requirements involve:

3.1.1. System Requirement

System requirements are the required specifications a system must have in order to operate. Typical system requirements for a software program include:

- Operating system
- Minimum CPU or processor speed
- Minimum GPU or video memory
- Minimum system memory (RAM)
- Minimum free storage space
- Audio hardware (sound card, speakers, etc)

System requirements listed for a hardware device may include:

- Operating system
- Available ports (USB, Ethernet, etc)
- Minimum GPU (for displays and graphics hardware)

3.1.2. Platform

A platform is a group of technologies that are used as a base upon which other applications, processes or technologies are developed. A platform is the basic hardware (computer) and software (operating system) on which software applications

can be run. This environment constitutes the basic foundation upon which any application or software is supported and/or developed.

3.1.3. Software application

The best database design software will allow you to simply and easily construct the databases you need, regardless as to whether you are looking to design a relational or non-relational database. Additionally, it should allow you to properly test it by running queries to ensure what you have will work as required.

3.1.4. people (Skill requirement)

Designing and implementing a database requires careful thought and planning. DBA must be able to do or possess the following skills;

- Data modelling and database design ability.
- Metadata management and repository usage.
- Database schema creation and management.
- Backup and recovery.
- Ensuring data integrity.
- Performance management and tuning.
- SQL code
- Data security
- Capacity planning
- General database management, etc.

Self-Check -3	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided in the next page:

Choose the best Answer (2 pts each)

1. Identifying business needs involves:
   A. Clarifying the business problem
   B. Identifying the vision or strategic mission
   C. Identifying stakeholders
   D. All
   E. none
2. A system may achieve the desired output or behaviour without meeting technical requirements, even if it will likely be unusable.
   A. True
   B. False
   C. Unknown
3. _____ is the basic hardware and operating system on which software applications can be run.
   A. Computer
   B. Platform
   C. Form factor
   D. All

Note: **Satisfactory rating - 4 points** **Unsatisfactory - below 4points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet 4: Developing conceptual model

Introduction

Data modelling is the process of visually representing all the business requirements that were established during the requirement analysis. Generally data modelling facilitates the development of a sound database that does not allow anomalies and inconsistencies.

4.1. Conceptual Modelling

Conceptual modelling is used to show how data is stored, connected, accessed and updated in the database management system. It is a conceptual representation of Data objects, the associations between different data objects, and the rules that to be stored in a database. It defines all the required information and information is represented only once.

At a high-level, they describe the things that an organization wants to collect data from and the relationships between these objects. The objective of a conceptual database design is to build a conceptual data model.

To do that, it is important to follow some steps:

1. Draw an Entity-Relationship Diagram. First create entity sets to identify attributes and to establish the proper relationship sets.
2. Define integrity constraints. Identify and document integrity constraints such as required data, referential integrity, attribute domain constraints, enterprise constraints, and entity integrity.
3. Review the final model. This requires you remove M:N relationships, remove recursive relationships, remove super types, remove relationships with attributes, and re-examine 1:1 relationships, which are normally not necessary.

Conceptual Database Design step

Step1: Identify entities and entity types

Step2: Determine the relationship and relationship cardinality

Step3: Identify and associate attributes with entity or relationship.

Step4: Determine attribute domains.

Step5: Determine candidate and primary key attributes.

Step6: Specialize /generalize entity types (optional step).

Step7: Draw Entity-Relationship diagram

Step8: Review Local Conceptual Data Model with User

4.1.1. Entity Relationship Model

ER Model is a high-level data model diagram that was introduced by Peter Chen (1976). This approach views real world data as systems of entities and relationships. In this model, we represent the real-world problem in the pictorial form to make it easy for the stakeholders to understand.

Generally it provides useful concepts that allow us to move from an informal description of what users want from their database to a more detailed, and description that can be implemented in a DBMS.

ERM starts with the identification of all major entities, their attributes and relationships among them. It is created for several reasons few of them are:

- To give a high level or over all idea about the system
- To give an idea about the developer's understanding of the system
- to give the systems project a kick start
- It's a basis for upcoming activities
- To produce a conceptual schema of the database.
- To represent the objects and their roles in the database within their domain independent of the DBMS to be used, etc.

We use the ER diagram as a visual tool to represent an ER Model. An ERD provide several benefits. Some of them are;

- Allows the analyst/designer gains a better understanding of the information to be contained in the database
- Used as a documentation tool
- It constitutes a framework for creating and manipulating databases.
- Makes someone to easily understand business process.
- Readily translatable into relational tables.
- Used as the blueprint for implementation.
- Used as effective communication tool
- Facilitate easy Conversion to any Model.

The creation of an ERD requires the identification and definition of entities, entity type, attributes, attributes type, relationship between entities and the constraints (business rules).

ERD may be drawn with pen and paper or may be drawn with the assistance of computer diagramming tools such as MS Visio, SMART Draw and other Computer Aided Systems Engineering (CASE) tools.

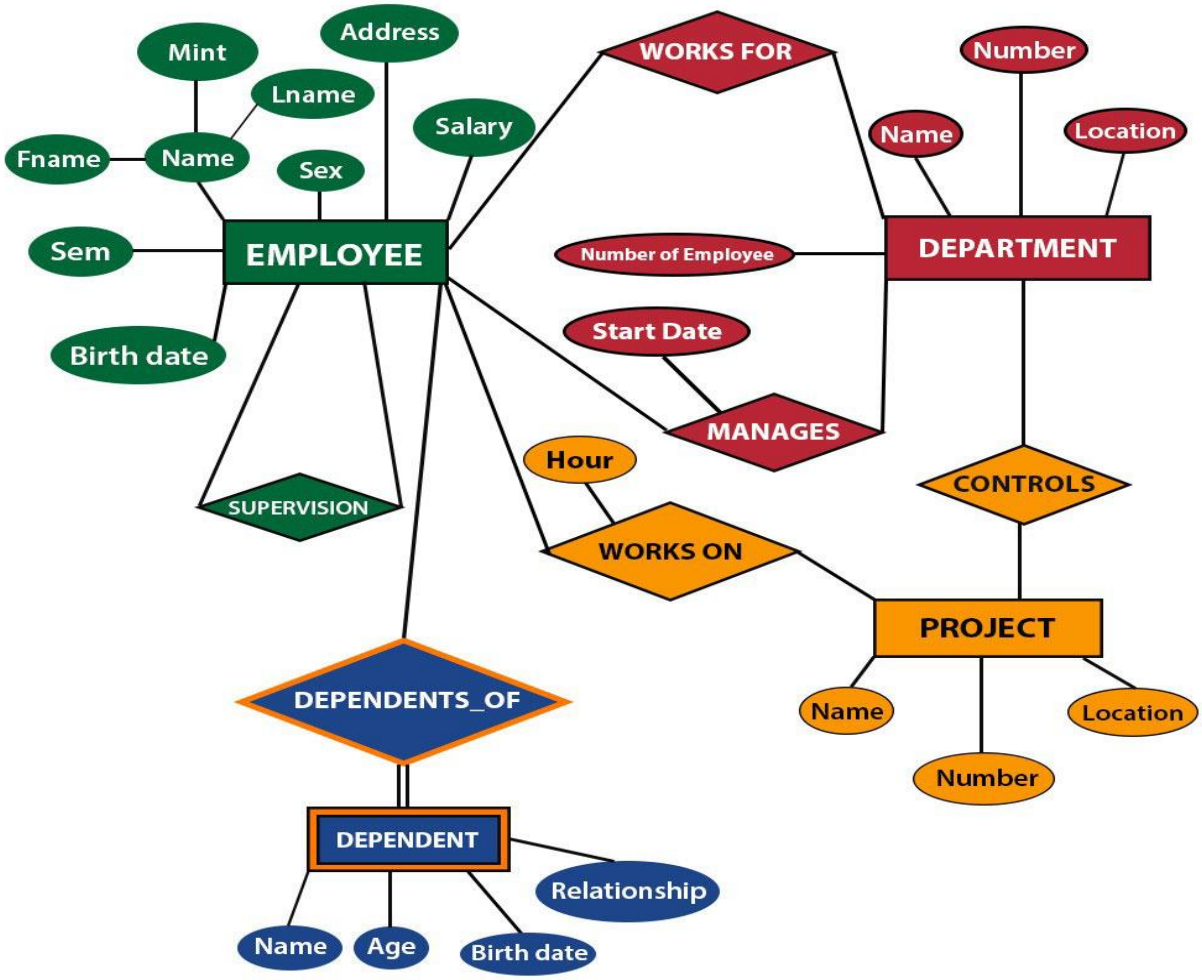


Figure 4:1 Entity Relationship Diagram

1.1.1. Components of ERD

ER diagram has the following three basic components/ elements:

- Entity
- Attribute
- Relationship

However, there are more elements which are based on the main or basic elements. These are weak entity, multi valued attribute, derived attribute, weak relationship, etc.

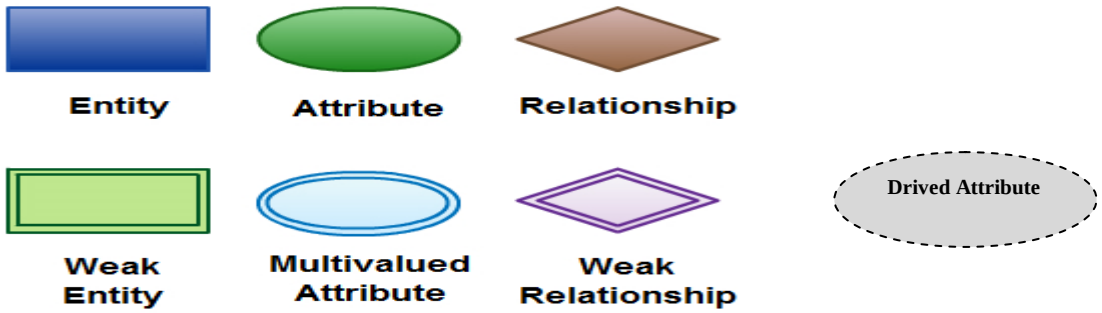


Figure: 4.2**.** Components of ERD

1.1.1.1. Entity

An entity is a self-determining and distinguishable item within an entity set. It is a thing that can be distinctly identified and described using a set of attributes. Entities are the "things" about which the database stores information. It may be;

- Tangible items, such as equipment,
- Concepts, such as accounts,
- People, such as employees
- Events, such as sales, or
- Places, such as business locations.

In ERD Entities are represented by a rectangle shape and named using singular and descriptive word (noun).

E**ntity instance** refers to a single occurrence of an entity type.

1.1.1.2. Weak Entity

A weak entity is an entity that depends on the existence of another entity. It cannot be identified by its own attributes. It uses a foreign key combined with its attributed to form

the primary key. The discriminator (or partial key) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set.

The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existence dependent, plus the weak entity set's discriminator. In ERD weak Entities are represented by a double lined rectangle shape and named using singular and descriptive word (noun).

1.1.1.3. Attribute

An attribute is a property, trait, or characteristic of an entity, relationship, or another attribute. It is a data item that describes a property of an entity.

Attributes can determine, explain, or categorize an entity set.

Attributes have values, which can be of different data types such as numbers, character strings, dates, images, sounds, and so on. Each attribute can have only one value for each instance of the entity set.

In a physical model, an attribute is a named column in a table, and has a domain. Each table has a list of attributes (or columns).

They are represented by oval shapes.

There are different types of attributes;

A. Composite Attribute

Composite attribute is an attribute that have its own attributes. Composite attributes can be divided into smaller sub parts, which represent more basic attributes with independent meanings. For example in the following picture; Address can be decomposed into Number, street and city

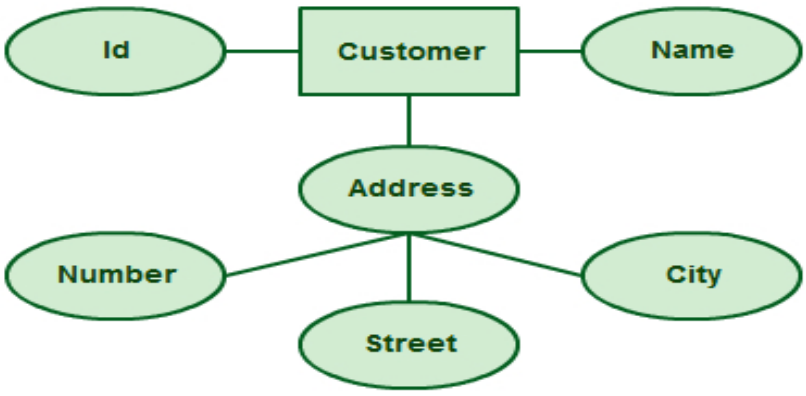


Figure 4.3: Composite Attribute

B. Simple or atomic attributes:

Atomic attribute is an attribute that are not divisible E.g. Age, Gender, city, Street, etc.

C. Multi-valued Attribute

If an attribute can have more than one value for a single column it is called multi-valued attribute. For example a teacher entity can have multiple subject values as well as field of specialization or qualification. Multi-valued attribute can be represented by double lined oval shape.

Figure 4.4: Multi-valued Attribute

D. Derived Attribute

It is an attribute for which the value is derived or computed from another attribute. This is found rarely in ER diagrams. For example: for a circle the Area can be derived from the radius whereas, the radius is the stored attribute.

Age may be derived from Date of birth. In ERD it is represented by dotted line oval shape.

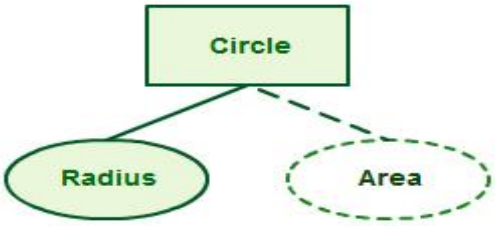


Figure 4.5: Derived Attribute

E. Unstable attributes

This type of attribute has values that always change. For example, the study year of a student in campus.

1.1.1.4. Relationship

A relationship describes how entities interact or associate with other entities. Relationships among entities are a critical part of the Entity Relationship Diagram.

The function that an entity plays in a relationship is called its role. Roles are normally explicit and not specified. They are useful when the meaning of a relationship set needs clarification.

There are a number of notations used to present design ER diagrams. Some of the popular notations are:

- Chen Notation
- Crow's foot
- UML
- Bachman

In Chen Notation Relationships are represented by diamond shapes and are labelled using verbs. For example:

Figure 4.6: relationship between Entities

In the above example the entity "Carpenter" may be related to the entity "table" by the relationship "builds" or "makes". Relationships are represented by diamond shapes and are labelled using verbs.

1.1.1.5. Constraints

Every business has restrictions on which attribute values and which relationships are allowed. In the conceptual data model constraints are used to handle these restrictions. A constraint is a requirement that entity sets must satisfy in a relationship. Constraints may refer to a single attribute of an entity set, or to relationship sets between entities. For example, "every TEACHER must work in one and only one DEPARTMENT".

The types of constraints are:

- **Cardinalities** – refers to the relationship of data in one database table with respect to another table. They are based on the number of possible relationship sets for every entity set. There are three types of cardinality constraints;

  Given a binary relation R between A (left entity in the figures below) and B (right entity in the figures below), R is said to be:

- ✓ **One-to-one (1:1)** - If both A and B have single-valued participation as depicted in the figure below.

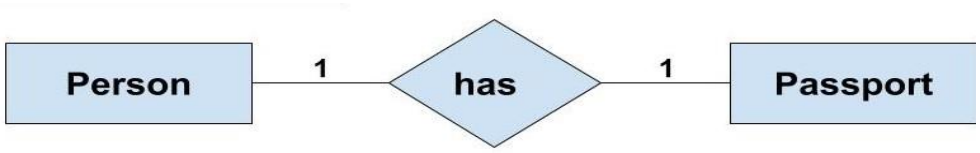


Figure 4.7: One to One relationships

- ✓ **One-to-many (1: M)** - if A has single and B has multi-valued participation as depicted in the figure below.

Figure 4.8: One - to -many relationships

- ✓ **many-to-many (M:M**) - if both A and B have multi-valued participation

Figure 4.9: Many to many relationships

Many-to-many relationships are not supported by the relational model and must be resolved by splitting the original M: N relationship set into two 1: M relationship sets.
Usually, the unique identifiers of the two entity sets participate in building the unique identifier of the third entity set.

- **Participation:** This type of constraint specifies whether the existence of an entity set depends on being related to another entity set via the relationship set. Participation cardinalities can be: -

- ✓ **Total or mandatory**: Each entity set must participate in a relationship and it cannot exist without that participation; the participation is compulsory.

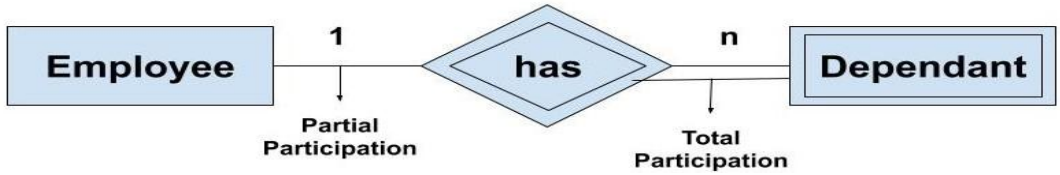


Figure 4.10: Relationship degree

Example*:* We have two entity type 'Employee' and 'Dependant'. Then the 'Dependent' entity is related to one or the other 'Employee' entity. This is called total participation of the entity in the relationship. But, it may be possible that some 'Employee' is not related to any of the 'Dependant' entity. So, 'Employee' is showing partial participation whereas the 'Dependant' is showing total participation in the relationship.

- ✓ **Partial or optional:** Each entity set may participate in a relationship; the participation is non-compulsory.

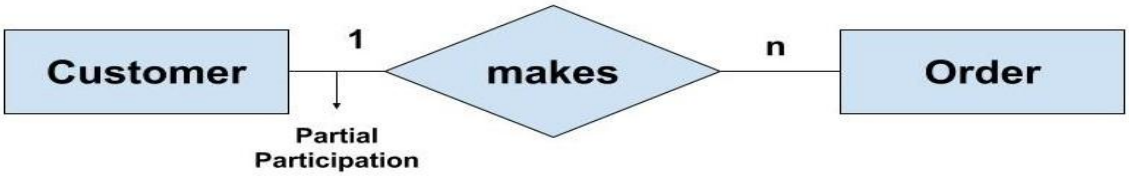


Figure4.11: Partial Participation

1.1.1.6. Degree of Relationship

It indicates the number of entities that participate in a relationship (number of associated entities)

1. **Unary relationship**

It is also called a recursive relationship. It is a relationship involving a single entity. A recursive relationship is a circular relationship that exists between two attributes in the same entity. Recursive relationships are rare, but useful. The most common example used to illustrate a recursive relationship is employee and manager names. Every employee has a manager, who is also an employee.

2. **Binary relationship**: relationship between two entities.
3. **Ternary relationship:** When three entities are involved in a relationship.

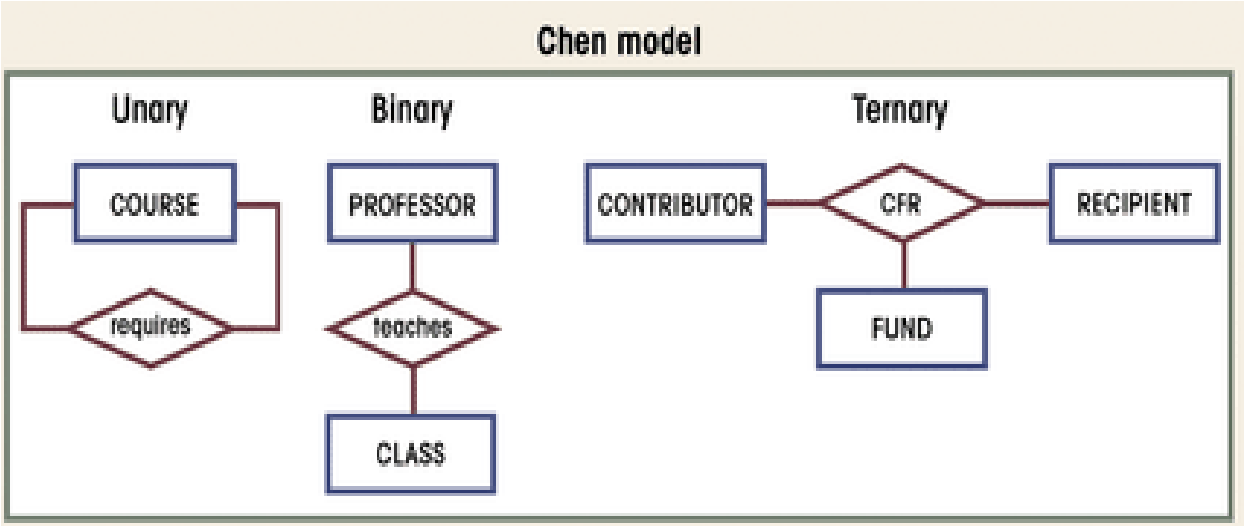


Figure 4.12: Relationship degree

Relationships that involve more than three entities are referred to as n-array relationships, where n is the number of entities involved in the relationship.

Self-Check -4	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Chose the best Answer(2 pts each)

1. The attributes that can be arranged into hierarchy are called
   A. Composite attributes
   B. Atomic attributes
   C. Derived attributes
   D. Simple attributes
2. In database management system, the term which is used to represent the real world concept or object is classified as
   A. Attribute B. Entity C. Relationship D. Abstraction
3. If an entity instance appears in only one relationship and vice versa then it is
   A. N:M relationship
   B. 1:N relationship
   C. 1:1 relationship
   D. All
4. Which of the following indicates the maximum number of entities that can be involved in a relationship?
   A. Minimum cardinality
   B. Maximum cardinality
   C. Optional participation
   D. None
5. Pick the relationship from the following:
   A. classroom B. teacher C. attends D. All

Note: **Satisfactory rating - 5 and 10 points** **Unsatisfactory - below 5points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet5: Submitting conceptual model to client for review

5.1. Submitting Conceptual model for feedback

Design reviews are critical for ensuring that an application is properly designed to achieve its purpose. Design reviews are an important facet of the system development lifecycle for database applications. It is during the design review that all aspects of the database and application code are reviewed for efficiency, effectiveness, and accuracy.

The best way to ensure a successful design is to have the end users' and customer's stamp of approval on each deliverable of the design process that pertains to the completeness of the database.

So as soon as you complete conceptual data modelling you should have to submit it to the client or organization for review. Because you should have to comply with the organization before you go further or turn to other design phase.

The purpose of this review is to validate the application concept. This involves a presentation of the statement of purpose as well as a general overview of the desired functionality that will be provided by the application. A CDR should be conducted as early as possible to determine the overall feasibility of a project. Failure to conduct a CDR can result in:

- Projects which provide duplicate or inadequate functionality.
- projects which are cancelled due to lack of funds, staffing, planning, user participation, and/or management interest
- over-budget projects

5.2. Importance of Database Design Reviews

It is during the design review that all aspects of the database and application code are reviewed for efficiency, effectiveness, and accuracy. It is imperative that all database applications, regardless of their size, are reviewed to assure that the application was design properly.

The design review is an important process for checking the validity of design decisions and correcting errors before applications and databases are promoted to production status.

Multiple design reviews should be conducted over the course of an application's life.

Generally conceptual design review is used to validate the concept of the data and proposed application.

5.3. Personnel's to be involved

Reviewing the scope and content of a database implementation before it moves to production is an important aspect of database systems and application development. By conducting database design reviews you can validate your assumptions and ensure that requirements are being met. And furthermore, you can identify and remediate problems before the system/app becomes operational.

The personnel who should engage in the design review process follow:

- Application Development personnel assigned to this development effort [AD]
- Representatives from other applications that are affected by this new application that is being reviewed(due to the need to interface with the new application, shared data requirements, scheduling needs, etc) [AA]
- Data Administration representative [DA]
- Database Administration representative [DBA]
- End users representatives [EU]
- End User Management [EUM]
- Information Center representative [IC]
- IT Management for the new application & all affected applications [ITM]
- On-Line Support representative (transaction unit, e.g. web/CICS) [OLS]
- Operational Support Management [OS]
- Technical Support and Systems Programming representatives [TS]

Self-Check -5	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Write True if the statement is correct and False if the statement is incorrect(2pts each)

1. The best way to ensure a successful design is to have the end users' and customer's stamp of approval
2. Conceptual design review is used to validate the concept of the data and proposed application.
3. Conceptual design review should be conducted at the end of project deployment.
4. Write at least two problems that may occur due to the failure of conducting a Conceptual design review.(2pts)

3.1.1. _____

3.1.2. _____

Note: **Satisfactory rating - 4 points** **Unsatisfactory - below 4 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet 6: Evaluating client feedback and making changes.

6.1. Customer feedback

Customer feedback is information provided by clients about whether they are satisfied or dissatisfied with a product or service. Their opinion is a resource for improving customer experience and adjusting your actions to their needs.

Never stop listening to customer feedback whether it is positive or negative, prompted or unprompted.

Gather users a feedback regarding the model. If any change are required by the organization evaluate the requirement weather it is acceptable thought or not and perform the requirement as needed.

Generally after a user feedback session, prepare yourself to make adjustments to work that have been accomplished. After required adjustments are performed, the documents will be prepared for final approval, in order to shift to the next phase of database design.

6.2. Importance of feedback

Feedback is important in any job role as it enables you to better yourself and grow your skill sets. As human beings, it is natural that we all have opinions, some of which are likely to be different to others. When it comes to design in general, it is totally subjective. It is very much opinion-based, which is why feedback from clients is extremely important to designers. They need to ensure that you are happy and therefore, it is imperative that you like the way in which your website is designed.

- Customer feedback helps improve products and services
- Customer feedback helps you measure customer satisfaction
- Collecting customer feedback shows you value their opinions

Self-Check -6	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Write True if the statement is correct and False if the statement is incorrect(2pts each)

1. Feedback is information that provided by development team to client about the usage of the system.

2. Customer feedback helps improve products and services

3. You must have to listen only the negative customer feedback.

Note: **Satisfactory rating - 2 points** **Unsatisfactory - below 2 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

L #25	LO #2- Develop Logical Data Model

Instruction sheet

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:

- Identifying attributes and Determining data types
- Undertaking normalization of attributes
- Developing entity relationship diagram to clarify cardinality of relationships
- Documenting normalized attributes and entity relationship diagram
- Forwarding documentation to client for confirmation

This guide will also assist you to attain the learning outcomes stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Identify attributes and Determining data types
- Undertake normalization of attributes
- Develop entity relationship diagram to clarify cardinality of relationships
- Document normalized attributes and entity relationship diagram
- Forward documentation to client for confirmation

Learning Instructions:

Read the specific objectives of this Learning Guide.

1. Follow the instructions described below.
2. Read the information written in the “Information Sheets”. Try to understand what are being discussed. Ask your trainer for assistance if you have hard time understanding them.
3. Accomplish the “Self-checks” which are placed following all information sheets.
4. Ask from your trainer the key to correction (key answers) or you can request your trainer to correct your work. (You are to get the key answer only after you finished answering the Self-checks).
5. If you earned a satisfactory evaluation proceed to the next learning guide,
6. If your performance is unsatisfactory, see your trainer for further instructions.

Information sheet 1: Identifying attributes and determining data types

Introduction

Logical data modelling describes how the system will be implemented, regardless of the DBMS. The purpose of logical modelling is to correctly model the data structure requirements for the business processes the database must support and to do away with data redundancy and other data structure errors.

Logical modelling affects database design, and also affects the performance and administration of an implemented database.

The task in the logical design step is to convert an ER schema into a relational database schema.

A table's logical structure (column names, column data types, and column sizes) is derived from the logical entity structures defined during conceptual database design.

- Entities are representing data and these entities are converted into tables.
- Entity attributes will become table columns.
- Entity unique identifiers will become table primary keys.
- Entity domains will become table data constraints.
- Entity relationships will become table primary and foreign key referential integrity constraints.

A logical data model describes the data in as much detail as possible, without regard to how they will be physical implemented in the database. Features of a logical data model include:

- Includes all entities and relationships among them.
- All attributes for each entity are specified.
- The primary key for each entity is specified.
- Foreign keys (keys identifying the relationship between different entities) are specified.
- Normalization occurs at this level.

The steps for designing the logical data model are as follows:

1. Specify primary keys for all entities.
2. Find the relationships between different entities.
3. Find all attributes for each entity.
4. Resolve many-to-many relationships.
5. Normalization.

1.1. Determining data types

A data type is a set of data with values having predefined characteristics. **It is** a constraint that specifies the possible value for each field in the table. The standard data types are;

- **CHAR (length)-** It includes text (characters, numbers, punctuations...). CHAR has as characteristic that it always saves a fixed amount of positions. If you define a CHAR (10) you can save up to ten positions maximum, but if you only use two positions the database will still save 10 positions. The remaining eight positions will be filled by spaces.
- **VARCHAR (length)-** VARCHAR is the same as CHAR; the difference is that VARCHAR only takes as much space as necessary. If you define a VARCHAR (10) you can save up to ten positions maximum, but if you only use two positions the database will save 2 positions. The remaining eight positions will be considered as free spaces.
- **TEXT:** It can contain large amounts of text. Depending on the type of database this can add up to gigabytes.
- **Bit:** –Integer data with either a 1 or 0 value
- **Int:–**Integer (whole number) data from -2^31 (-2,147,483,648) through 2^31 – 1 (2,147,483,647)
- **Smallint: –**Integer data from 2^15 (-32,768) through 2^15 – 1 (32,767)
- **Tinyint: –**Integer data from 0 through 255
- **Decimal: –**Fixed precision and scale numeric data from -10^38 -1 through 10^38
- **Numeric: –**A synonym for **decimal**
- **Timestamp: –**A database-wide unique number

- **Unique identifier: –**A globally unique identifier (GUID)
- **Money:** - Monetary data values from -2^63 (-922,337,203,685,477.5808) through 2^63 – 1 (+922,337,203,685,477.5807), with accuracy to one-ten-thousandth of a monetary unit.
- **Small money: –**Monetary data values from -214,748.3648 through +214,748.3647, with accuracy to one-ten-thousandth of a monetary unit
- **Float:–**Floating precision number data from -1.79E + 308 through 1.79E + 308.
- **Real:–**Floating precision number data from -3.40E + 38 through 3.40E + 38.
- **Date time –**Date and time data from January 1, 1753, to December 31, 9999, with an accuracy of one-three-hundredths of a second, or 3.33 milliseconds
- **Small date time: –**Date and time data from January 1, 1900, through June 6, 2079, with an accuracy of one minute.
- **Binary: –**Fixed-length binary data with a maximum length of 8,000 bytes
- **Varbinary: –**Variable-length binary data with a maximum length of 8,000 bytes
- **Image: –** Variable-length binary data with a maximum length of 2^31 – 1 (2,147,483,647) bytes
- **BLOB (Binary Large Object):** for binary data such as files. NET, IP addresses and also useable for net masks, etc.

Self-Check -1	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Choose the best answer (2pts each)

1. It is a constraint that specifies the possible value for each field in the table.
   A. Char
   B. Varchar
   C. Constraints
   D. Data type
   E. E. None
2. Integer data with either a 1 or 0 value
   A. Binary
   B. Integer
   C. Text
   D. Image
   E. All
3. Which one of the following will be affected due to poor logical design?
   A. performance
   B. administration
   C. implementation
   D. All
   E. None

Note: **Satisfactory rating - 2 points** **Unsatisfactory - below 2 points**
You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet2	Undertaking normalization of attributes

2.1. Normalization

Normalization was developed by Dr. E.F.Codd in 1972 as part of the Relational Database.

It is a procedure that aims at converting relational schemas into a more desirable form. The goal is to remove redundancy in relations and anomalies.

Depending on the level of normalization, redundant data might be completely eliminated. During the normalization process, large tables with many columns are divided into smaller tables with a smaller number of columns and multi-valued attribute must be normalized. Some user-related design considerations during the normalization include:

- What data should be stored in the database?
- How will the user access the database?
- What privileges does the user require?
- How should the data be grouped in the database?
- What data is the most commonly accessed?
- How is all data related in the database?
- What measures should be taken to ensure accurate data?

The process of converting one entity into multiple entities in order to normalize the original is called decomposition.

2.1.1. Advantages of Normalization

The main benefit of normalization is to promote overall data consistency between tables and data accuracy through the reduction of redundant information that is stored. Some of the major benefits include:

- Good database organization will be gained.
- The amount of unnecessary redundant data is reduced.
- Data integrity is easily maintained within the database.
- The database and application design processes are much more flexible.
- Security is easier to manage.
- Save disk space, etc.
- Prevent Insert, Update and Delete anomalies, etc.

2.1.2. Disadvantages of Normalization

It is necessary to normalize the database to a certain degree. Unless it;

- will be affect performance
- Result in complicated table joins: Since it produces a lot of tables with a relatively small number of columns; these columns then have to be joined in order to put the information back together. For example, a query might require retrieval of data from multiple normalized tables.
- Challenges developers to code queries that return desired information.
- Will complicate the database maintenance. The higher the level of normalization, the more stupendous the amount of tables in the database, etc.

2.2. Un-Normalized Form (UNF)

Un-normalized form is a preparatory stage of the normalization process allowing us to create a structured frame, representative of a piece of organizational data such as a form or document (e.g invoice, report, purchase order etc.). This is our initial Normalization 'relation' that contains both real data, taken from the form or document, and modelled data, based upon and extended from the original from or document.

Table 2.1: Un normalized table

EmpID	Name	Manager	Dept	Sector	Spouse/Children
285	Chala Dabale	Sifen	Engineering	6G	
365	Yosen	Sifen	Marketing	8G	
458	Simbo Girma	Derertu	Safety	7G	Singitan, walif, kenna,sena

2.3. Normal forms

Normal form is a way of measuring the levels or degree, or depth, to which a database has been normalized. There are different levels of normal forms. The three most common normal forms implemented in production databases are the First, Second, and Third normal forms.

In normal forms, each subsequent normal form depends on normalization steps taken in the previous normal form.

For instance, in order to normalize a database using the Second normal form the database must initially be in the First normal form.

2.3.1. First Normal Form (1NF)

A relation is considered to be in first normal form if all of its attributes have domains that are indivisible or atomic. The idea of atomic values for attribute ensures that there are no 'repeating groups' . This is because a relational database management system is capable of storing a single value only at the intersection of a row and a column. Repeating Groups are when we attempt to store multiple values at the intersection of a row and a column and a table that will contain such a value is not strictly relational

A relation is in **1NF** if;

- All values stored in the relation are single-valued or atomic.
- Repeating groups in individual tables are eliminated.
- Each row of data must have a unique identifier (Primary key.)

Table:2.2. Fields contain smallest meaningful values

EmpID	FName	LName	Manager	Dept	Sector	Spouse	Child1	Child2	Child3
285	Chala	Dabale	Sifen	Eng.	6G				
365	Yosen		Sifen	Marketing	8G				
458	Simbo	Girma	Derartu	Safety	7G	Singitan	Walif	kenna	sena

Table 2.3: No more repeated fields

EmpID	**FName**	**LName**	**Manager**	**Department**	**Sector**	**Dependent**
285	Chala	Dabale	Sifen	Eng.	6G	
365	Yosen		Sifen	Marketing	8G	
458	Simbo	Girma	Derartu	Safety	7G	Singitan
458	Simbo	Girma	Derartu	Safety	7G	Walif
458	Simbo	Girma	Derartu	Safety	7G	Sena
458	Simbo	Girma	Derartu	Safety	7G	Kenna

2.3.2. Second Normal Form (2NF)

A relation is in 2NF if;

- Meet all the requirements of the first normal form.
- Remove subsets of data that apply to multiple rows of a table and place them in separate tables.
- Create relationships between these new tables and their predecessors through the use of foreign keys.

Table 2.4: Remove Repeated Data from Table

EmpID	Dependent
458	Singitan
458	Walif
458	Sena
458	Kenna

EmpID	FName	LName	Manager	Department	Sector
285	Chala	Dabale	Sifen	Eng.	6G
365	Yosen		Sifen	Marketing	8G
458	Simbo	Girma	Derartu	Safety	7G

2.3.3. Third Normal form

A relation is in third normal form if it is in 2NF and there is no non-key attribute that depends transitively on the candidate key.

Every attribute depends directly on the primary key and not through a transitive relation.

A **transitive dependency** is a functional dependency which holds by virtue of transitivity.

A transitive dependency can occur only in a relation that has three or more attributes.

For example, if A, B, and C designates three distinct attributes (or distinct collections of attributes) in the relation and if we have a primary key A and a non-key domain B and C where C is more dependent on B than A and B is directly dependent on A, then C can be considered transitively dependant on A. By removing transitive dependency we can;

- Reduce the amount of data duplication.

- Achieved data integrity

Functional dependency is a relationship that exists when one attribute uniquely determines another attribute. If R is a relation with attributes X and Y, a functional dependency between the attributes is represented as X->Y, which specifies Y is functionally dependent on X.

Functional dependency helps ensure the validity of data.

For example, consider a table Employees that lists characteristics including ID, name, date of birth, address and so on.

The attribute **ID** will determine the value of name, date of birth, address and perhaps other values, because a ID is unique, while a name, date of birth or address may not be. We can write it like this:

ID -> name, date of birth, address

Therefore, name, date of birth and address are functionally dependent on ID. However, the reverse statement (name -> ID) is not true because more than one employee can have the same name but will never have the same ID. In other word, if we know the value of the ID attribute, we can find the value of name, date of birth and address. But if we instead know the value of only the name attribute, we cannot identify the ID.

Therefore every non-prime attribute of table must be dependent on primary key. The transitive functional dependency should be removed from the table.

A relation is in 3NF if;

- Meet all the requirements of the second normal form.
- Remove columns that are not dependent upon the primary key.
- Nothing but the Key (attributes are depend only on the primary key not depends on another non key attributes).

Table 2.5 Tables in Third Normal

EmpID	FName	LName	Manager	Dept_code
285	Chala	Dabale	Sifen	Tech01
365	Yosen		Sifen	Com01
458	Simbo	Girma	Derartu	Com02

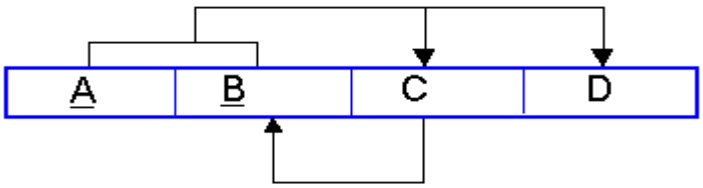
Dept_code	Department	Sector
Tech01	Eng.	6G
Com01	Marketing	8G
Com02	Safety	7G

EmpID	Dependent
458	Singitan
458	Walif
458	Sena
458	Kenna

2.3.4. Boyce-Codd Normal form

BCNF is considered a special condition of third Normal form and sometimes referred to as 3.5NF. A database table is in BCNF if and only if there are no non-trivial functional dependencies of attributes on anything other than a superset of a candidate key.

A table is in BCNF if every determinant is a candidate key. A table can be in 3NF but not in BCNF when a non key attribute is a determinant of a key attribute. The dependency diagram may look like the one below



A and B are the keys and C and D depend on both A and B. There are no transitive dependencies existing between the non key attributes, C and D.

The table is not in BCNF because a dependency exists between C and B. In other words if we know the value of C we can determine the value of B.

2.3.5. Fourth Normal Form

Fourth normal form is in effect when an entity is in Boyce-Codd normal form and it has no multi-valued dependencies.

, and neither of the following conditions exists:

A multi-valued dependency is a situation in which two or more attributes are dependent on a determinant and each dependent attribute has a specific set of values. The values in these dependent attributes are independent of each other.

For example a SALES_REGION table that includes attributes for sales persons and customers. CUSTOMER is functionally dependent on SALES_REGION_ID, and SALES_PERSON is dependent on SALES_REGION_ID. Neither SALES_PERSON in union with SALES_ REGION_ID nor CUSTOMER in union with SALES_REGION_ID is the whole entity. Analyzing and Modeling Business Requirements neither CUSTOMER nor SALES_PERSON are subsets of SALES_REGION_ID. This entity is violating Fourth normal form.

The problem is that we would have to create a row for every possible combination of sales person and customer to ensure consistency for the resulting table. Although this might be what the developer intended, it would produce redundancy with other table's data and could be accomplished as well with relations.

2.3.6. Fifth Normal Form

Fifth normal form (also called project-join normal form) is in effect when Fourth normal form is in effect and the entity has no join dependencies that are not related to that entity's candidate keys.

When we decompose (break up) entities as we apply normalization rules, we should be able to reconstruct the original entity by doing joins between the two resulting entities without losing any data and without generating extra and generally incorrect rows. This is called a lossless join.

An entity is in Fifth normal form when it cannot be decomposed into several smaller entities, which have different keys from the original without data losses. If data was lost after decomposition, a lossless-join would not be possible. If all the decomposed entities use one of the original entity's candidate keys as keys, joins of those entities would result in a lossless join. With respect to the original entity, it would still be considered to be in Fifth normal form without decomposition.

2.4. Lossless and Lossy Decompositions

Decomposition of a relation R into relations X and Y is lossless if no information from the original relation is lost after the decomposition. In other words, the original relation can be constructed back from the decomposed relations and no spurious rows of information are added as data to the resulting rows. It follows then, that if a decomposition results in loss of information from the original relation R then the decomposition is lossy and is obviously not a desirable property. Let R be a relation with functional dependency F that

is decomposed into R1 and R2. Then the decomposition is lossless if one of the following FDs is true for the decomposed relations R1 and R2: R1 ∩ R2 → R1 or R1 ∩ R2 → R2 More simply if the common attributes in R1 and R2 (i.e. R1 ∩ R2) form a super key for either R1 or R2, then the original relation R has undergone a lossless decomposition into R1 and R2. An essential property of decomposition in relational database design is that they should be lossless. A lossless join of decomposed relation X and Y into R results in no information loss and no addition of extra misguiding information when joining X and Y to get back the original relation R. Example Consider the relation employee as follows:

EMP_ID	EMP_NAME	WORK_EXP	DEPT_ID	DEPT_NAME

A desired ***lossless decomposition*** for the above relation employee is as follows:

DEPT_ID	DEPT_NAME

Department relation

EMP_ID	EMP_NAME	WORK_EXP	DEPT_ID

Employee relation

In the above, (Department) ∩ (Employee) = DEPT_ID and DEPT_ID → { DEPT_ID , DEPT_NAME}, that is, DEPT_ID is a super key for the ***Department*** relation and the decomposition is lossless.

A ***lossy decomposition*** for the above would be:

DEPT_ID	DEPT_NAME	EMP_NAME

Department relation

EMP_ID	EMP_NAME	WORK_EXP

Employee relation

In the above, (Department) ∩ (Employee) = EMP_NAME which is not a super key for department or employee table hence the decomposition is lossy.

Figure 2.2 Lossy and lossless decomposition

2.5. Denormalization

It is the process of taking a normalized database and modifying table structures to allow controlled redundancy to increase database performance. In other word, it is the process of taking the level of normalization within the database down by recombining separate tables, or creating duplicate data within tables. This will reduce the number of tables that need to be joined in order to retrieve the requested data, which results in less I/O and CPU time.

The goal of denormalization is improving performance.

A denormalized database is not the same as a database that has not been normalized. Before you denormalize a database, ensure that the desired performance can't be obtained by some other method, such as:

- indexing the tables,
- using views, or
- Optimizing the use of hardware devices.
- Parallel processing (multiple host processors)
- Decreasing server workload
- Tuning application
- Network related issue i.e. too many router hops between the user and the host, traffic, etc.

Denormalization must have to be undertaken only to the degree required unless it comes with consequences such as:

- ✓ The data model becomes much less flexible and more complex.
- ✓ Maintenance becomes more difficult.
- ✓ Changes to the database and its applications are more difficult.
- ✓ Combining tables puts back the data redundancy & may create problems they were normalized to overcome.

Self-Check -2	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Write Choose the best answer(2pts each)

1. Modifying table structures to allow controlled redundancy to increase database performance.
   A. Normalization
   B. Normal form
   C. Denormalization
   D. All
   E. E. None
2. The Degree to which the table is normalized
   A. Normalization B. Normal form C. Denormalization D. All E. None
3. In which normal form all attributes are depend only on primary key.
   A. 1NF B. 2NF C. 3NF D. BCNF E. None
4. Which normal form should ensure the atomicity of value?
   A. 5NF B. 1NF C. 3NF D. BCNF E. 2NF
5. The goal of normalization is_____.
   A. Improving performance
   B. Eliminating Redundancy
   C. Ensuring Security
   D. All
   E. None

Note: **Satisfactory rating - 5 points** **Unsatisfactory - below 5 points**
You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet 3: Developing entity relationship diagram to clarify cardinality of relationships

3.1. Developing entity relationship diagram

Entity Relationship Diagrams are a major data modelling tool that used to organize the data into entities and define the relationships between the entities. This process enables the analyst to produce a good database structure so that the data can be stored and retrieved in a most efficient manner.

Example 1; a company has several departments. Each department has a supervisor and at least one employee. Employees must be assigned to at least one, but possibly more departments. At least one employee is assigned to a project, but an employee may be on vacation and not assigned to any projects. The important data fields are the names of the departments, projects, supervisors and employees, as well as the supervisor and employee number and a unique project number.

Steps to develop ERD;

Step 1. Identify Entities

A data entity is anything real or abstract about which we want to store data. Entity types fall into five classes: roles, events, locations, tangible things, or concepts. The best way to identify entities is to ask the system owners and users to identify things about which they would like to capture, store and produce information. Another source for identifying entities is to study the forms, files, and reports generated by the current system.

The entities in **Example 1** are Department, Employee, Supervisor and Project. One is tempted to make Company an entity, but it is a false entity because it has only one instance in this problem. True entities must have more than one instance.

Step 2. Find Relationships

There are natural associations between pairs of entities.

For example; a student (Entity) is **enrolled (role) in** one or more courses (Entity).

We construct the following Entity Relationship Matrix from Example 1:

Table1:Find

	Department	**Employee**	**Supervisor**	**Project**
Department		is assigned	run by	
Employee	belongs to			works on
Supervisor	Runs			
Project		uses		

Step 3. Draw Rough ERD

Using rectangles for entities and lines for relationships, we can draw an ERD.

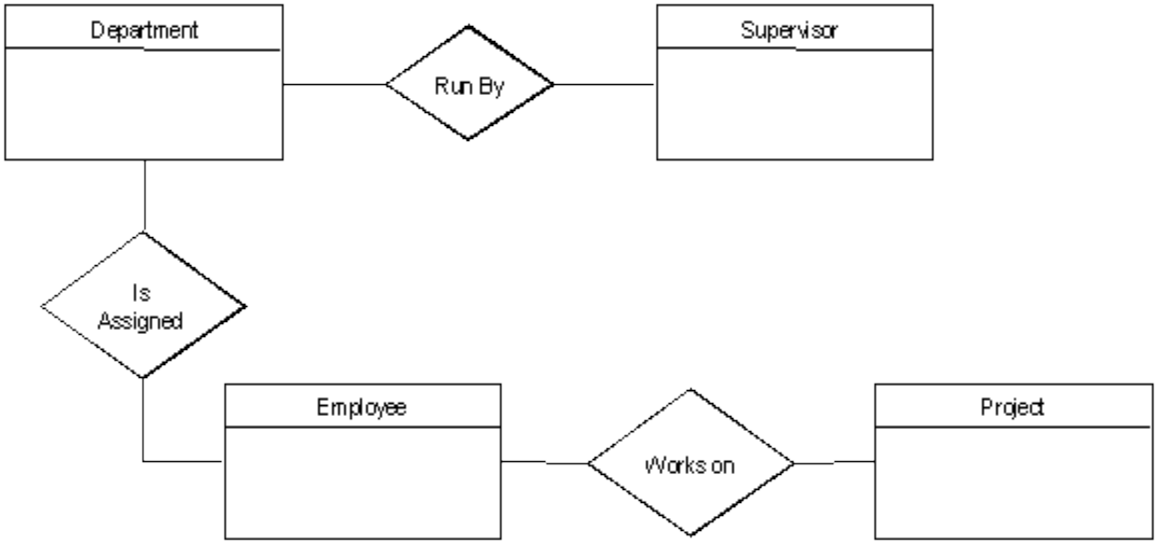


Figure 3.1. Rough ERD

Step 4. Fill in Cardinality

At each end of each connector joining rectangles, we need to place a symbol indicating the minimum and maximum number of instances of the adjacent rectangle there are for one instance of the rectangle at the other end of the relationship line.

The first symbol is either 0 to indicate that it is possible for no instances of the entity joining the connector to be related to a given instance of the entity on the other side of the relationship, 1 if at least one instance is necessary or it is omitted if more than one instance is required. For example, more than one student must be enrolled in a course

for it to run, but it is possible for no students to have a particular instructor (if they are on leave).

The second symbol gives the maximum number of instances of the entity joining the connector for each instance of the entity on the other side of the relationship. If there is only one such instance, this symbol is 1. If more than 1, the symbol is a crow's foot opening towards the rectangle.

If you read it like a sentence, the first entity is the subject, the relationship is the verb, the cardinality after the relationship tells how many direct objects (second entity) there are.

From the description of the problem (example 1) we see that:

- Each department has exactly one supervisor.
- A supervisor is in charge of one and only one department.
- Each department is assigned at least one employee.
- Each employee works for at least one department.
- Each project has at least one employee working on it.
- An employee is assigned to 0 or more projects.

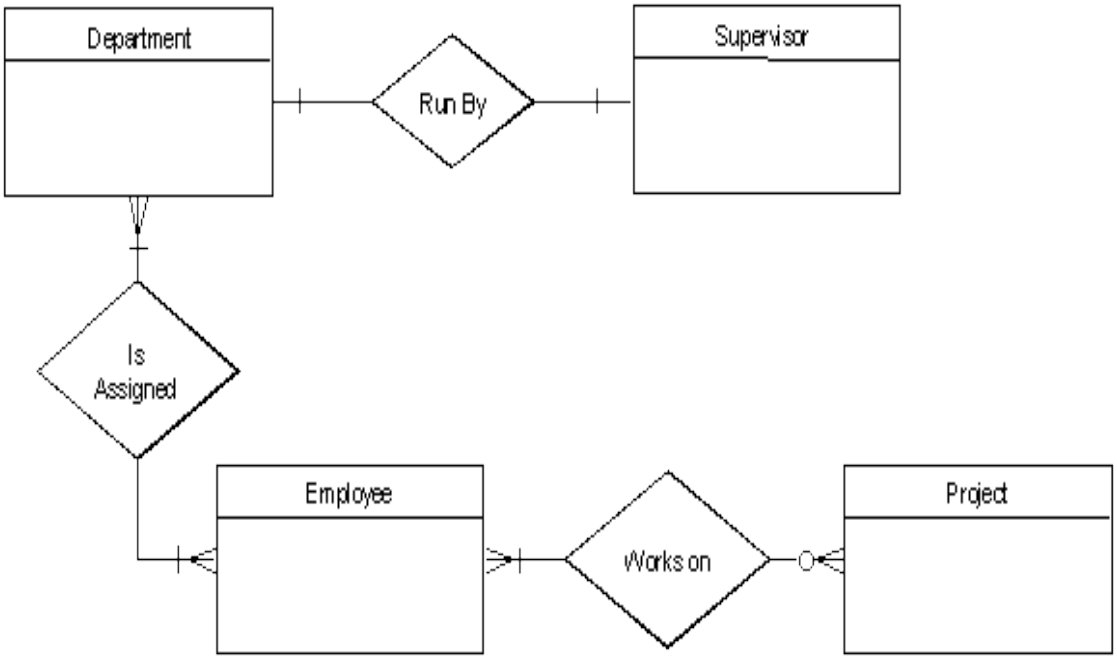


Figure:3.2. ERD with cardinality

Step 5. Define Primary Keys

For each entity we must find a unique primary key so that instances of that entity can be distinguished from one another. Often a single field or property is a primary key.

The primary keys (Example1) are Department Name, Supervisor Number, Employee Number, and Project Number.

Step 6. Draw Key-Based ERD

Looking at the Rough Draft ERD, we may see some relationships which are non-specific or many-to-many. I.e., there are crows feet on both ends of the relationship line. Such relationships spell trouble later when we try to implement the related entities as data stores or data files, since each record will need an indefinite number of fields to maintain the many-to-many relationship. This case, you can solve the problem by introducing an extra entity, called an associative entity for each many-to-many relationship.

The new associative entity's name will be the hyphenation of the names of the two originating entities. It will have a concatenated key consisting of the keys of these two entities.

There are two many-to-many relationships in the rough ERD above(Figure 3.2), between Department and Employee and between Employee and Project. Thus we need the associative entities Department-Employee and Employee-Project. The primary key for Department-Employee is the concatenated key Department Name and Employee Number. The primary key for Employee-Project is the concatenated key Employee Number and Project Number.

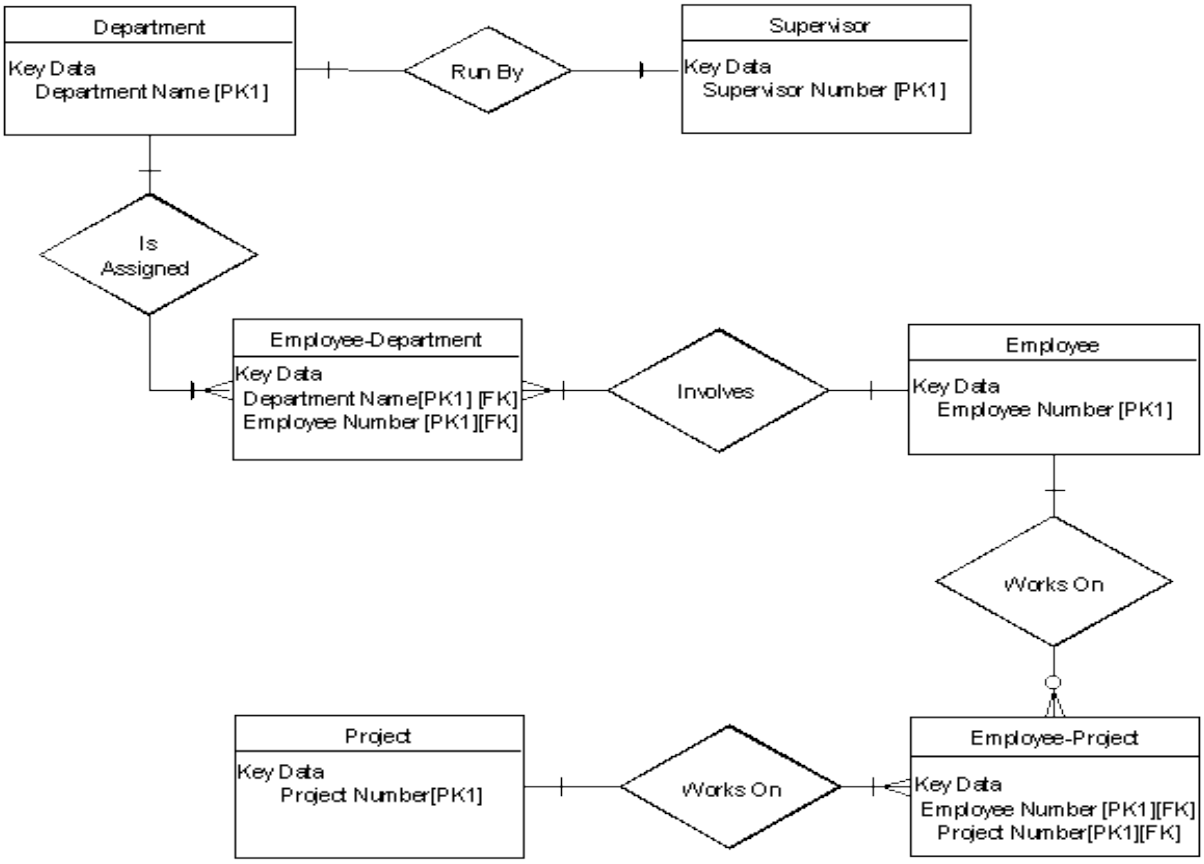


Figure 3.3. Key based ERD

Step 7. Identify Attributes

A data attribute is a characteristic common to all or most instances of a particular entity. In this step we try to identify and name all the attributes essential to the system we are studying without trying to match them to particular entities. The best way to do this is to study the forms, files and reports currently kept by the users of the system and circle each data item on the paper copy. Cross out those which will not be transferred to the new system, extraneous items such as signatures, and constant information which is the same for all instances of the form (e.g. your company name and address). The remaining circled items should represent the attributes you need. You should always verify these with your system users. (Sometimes forms or reports are out of date.)

The only attributes indicated in **Example 1** are the names of the departments, projects, supervisors and employees, as well as the supervisor and employee NUMBER and a unique project number.

Step 8. Map Attributes

For each attribute we need to match it with exactly one entity. Often it seems like an attribute should go with more than one entity (e.g. Name). In this case you need to add a modifier to the attribute name to make it unique (e.g. Customer Name, Employee Name, etc.) or determine which entity an attribute "best' describes. If you have attributes left over without corresponding entities.

Table 2. Mapping attribute

Attribute	Entity	Attribute	Entity
Department Name	Department	Supervisor Number	Supervisor
Employee Number	Employee	Supervisor Name	Supervisor
Employee Name	Employee	Project Name	Project
		Project Number	Project

Step 9. Draw Fully-Attributed ERD

If you introduced new entities and attributes in step 8, you need to redraw the entity relationship diagram. When you do so, try to rearrange it so no lines cross by putting the entities with the most relationships in the middle. If you use a tool like Systems Architect, redrawing the diagram is relatively easy.

Even if you have no new entities to add to the Key-Based ERD, you still need to add the attributes to the Non-Key Data section of each rectangle. Adding these attributes automatically puts them in the repository, so when we use the entity to design the new system, all its attributes will be available.

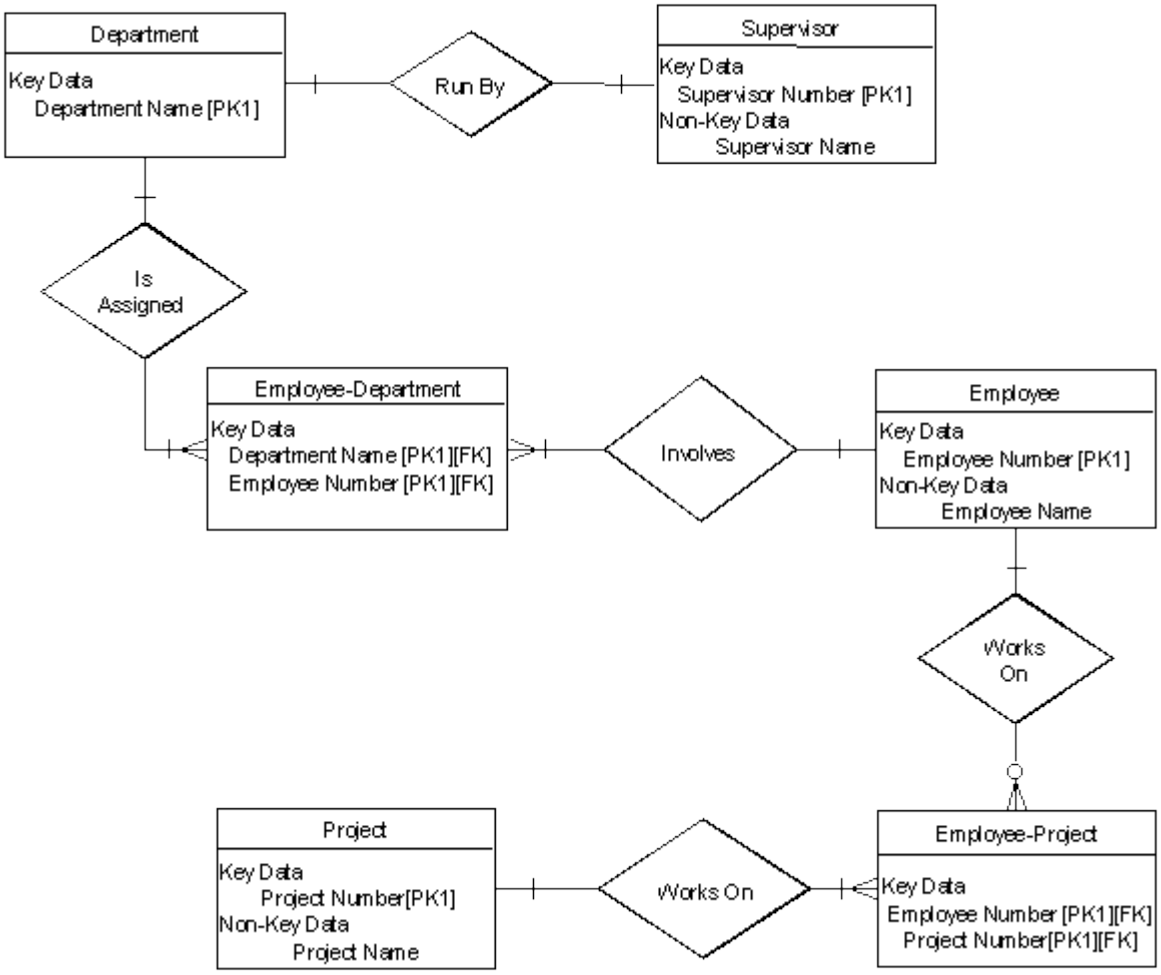


Figure 3.4 fully attributed ERD

Step 10. Check Results

Look at your diagram from the point of view of a system owner or user. Is everything clear? Check through the Cardinality pairs. Also, look over the list of attributes associated with each entity to see if anything has been omitted.

Self-Check -3	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Choose the best answer(2pts each)

1. It determines the number of occurrences of one entity for a single occurrence of the related entity.
   A. Cardinality
   B. Relationship
   C. Attribute
   D. None
2. A things or concepts about which the end-users want to store data.
   A. Cardinality
   B. Relationship
   C. Attribute
   D. Entity
3. An attribute(s) that uniquely identify one and only one occurrence of each entity.
   A. Cardinality
   B. Primary key
   C. Foreign key
   D. Attribute

Note: **Satisfactory rating - 2 points** **Unsatisfactory - below 2 points**
You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____
Rating: _____

Information sheet 4: Documenting normalized attributes and entity relationship diagram

4.1. Documenting attributes and ERD

Documentation is the organized collection of records that describe the structure, purpose, operation, maintenance, and data requirements for a computer program, operating system, or hardware device.

Documentation is important because it helps to provide important information and visual representations of the code and other parts of the application. This will make it easier to do handoffs with other developers and engineers as well as perform checks for troubleshooting, updates or maintenance.

Documentation may ranges from documenting the code and its functions internally to providing external documentation that provides added context about what was built. There is also user documentation that tells the user how to access and use the application.

So you have to document the attributes of all table either in tabular form or simply by listing table with their respective column.

For example;

Course-code: Method descriptive name

Primary/Foreign Key

Column data type: varchar

Field length:20

Column Description: Short title

Field constraint: Cannot be blank.

Course-name: Method descriptive name

Column Description: Short title

Column data type: varchar

Field length:50

Field constraint: unique and not null

Field length:20

4.2. Documentation Standard

IEEE defines software design documentation as ‘a description of software created to facilitate analysis, planning, implementation, and decision-making’. This design description is used as a medium for communicating software design information and can be considered as a blueprint or model of the system.

The information that the software design document should describe depends on various factors including the type of software being developed and the approach used in its development.

A number of standards have been suggested to develop a software design document. However, the most widely used standard is by IEEE, which acts as a general framework. This general framework can be customized and adapted to meet the needs of a particular organization.

The documentation both explains how the software operates or how to use it, and may mean different things to people in different roles.

Documentation is an important part of software engineering.

Types of documentation include:

- **Requirements** – Statements that identify attributes, capabilities, characteristics, or qualities of a system. This is the foundation for what will be or has been implemented.
- **Architecture/Design** – Overview of software which includes relations to an environment and construction principles to be used in design of software components.
- **Technica**l – Documentation of code, algorithms, interfaces, and APIs.
- **End user** – Manuals for the end-user, system administrators and support staff.
- **Marketing** – How to market the product and analysis of the market demand, etc.

Self-Check -4	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Choose the best answer (2 pts each)

1. _____ is the organized collection of records that describe the structure, purpose, operation, maintenance, and data requirements for a computer program, operating system, or hardware device.

   A. Requirement B. Documentation C. Design D. None

2. There is no standard for documentation.

   A. True B. False C. Unknown

3. Documentation of code, algorithms, interfaces, and APIs

   A. User manual

   B. Technical requirement

   C. System requirement

   D. All

4. Statements that identify attributes, capabilities, characteristics, or qualities of a system.

   A. Technical requirement

   B. User manual

   C. System requirement

   D. All

Note: **Satisfactory rating - 4 points** **Unsatisfactory - below 4 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet 5: Forwarding documentation to client for confirmation

5.1. Forwarding documentation to client for confirmation

The approval by the client describes the acceptance by the client. It denotes the successfulness of design which meets the requirements established at the beginning of the project.

The approval documentation is essential for any project in order to keep a record of what has been agreed. Documentation serves as a simple reminder of the expectations of both the parties, clients and project team.

Anyone with a question about the different phases of the project can refer to the approval documentation to see what actions need to be taken based on the approved requirements.

The 3 advantages of the acceptance documentation are:

- Managing the expectations between the client and the project team.
- Leaving room for additions or modifications while maintaining clear communication.
- Maintaining maximum responsibility for each phase of the project.

5.2. Verification and acceptance criteria

The verification and acceptance criteria are conditions that are used to determine whether the work has been completed according to the requirements initially established.

In particular, the acceptance criteria are designed so as not to be ambiguous.

Some examples of possible **verification and acceptance criteria** used by the client:

- Behavior of a system.
- Specifications of the phases of a process including automated steps and human activities.
- Specification for a calculation that can include business rules, algorithms, and formulas.
- Aspects of a product that make it pleasant to use.
- Implementation expectations that allow architects, designers, engineers and experts to be flexible in their work.
- Performance requirements.

- Definitions of internal controls.
- Operational requirements such as “the platform integration”.
- Quality expectations.
- Constraints on materials, etc.

Self-Check -5	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Short Answer

1. List down the advantages of the acceptance documentation(3 pts)
2. _____ serves as a simple reminder of the expectations of both the parties.(2 pts)
3. _____denotes the successfulness of design which meets the requirements established at the beginning of the project.(2pts)

Note: **Satisfactory rating - 4points** **Unsatisfactory - below 4 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Operation sheet 1: Steps to Draw ERD diagram

Steps to Draw ER Diagrams

1. Identifying Entities
2. Find Relationships
3. Draw Rough ERD
4. Fill in Cardinality
5. Define Primary Keys
6. Draw Key-Based ERD
7. Identify Attributes
8. Map Attributes
9. Draw Fully Attributed ERD
10. Check Results

Operation sheet 2: Steps to Normalize attribute

Table 2.1: Un-normalized table

EmpID	Name	Manager	Dept	Sector	Spouse/Children
285	Chala Dabale	Sifen	Engineering	6G	
365	Yosen	Sifen	Marketing	8G	
458	Simbo Girma	Derertu	Safety	7G	Singitan, walif, kenna,sena

First Normal Form (1NF)

i. All values stored in the relation must be single-valued or atomic.

EmpID	FName	LName	Manager	Dept	Sector	Spouse	Child1	Child2	Child3
285	Chala	Dabale	Sifen	Eng.	6G				
365	Yosen		Sifen	Marketing	8G				
458	Simbo	Girma	Derartu	Safety	7G	Singitan	Walif	kenna	sena

ii. Eliminate repeating groups in individual tables.

EmpID	FName	LName	Manager	Dept	Sector	Dependent name
285	Chala	Dabale	Sifen	Eng.	6G	
365	Yosen		Sifen	Marketing	8G	
458	Simbo	Girma	Derartu	Safety	7G	Singitan
458	Simbo	Girma	Derartu	Safety	7G	Walif
458	Simbo	Girma	Derartu	Safety	7G	Sena
458	Simbo	Girma	Derartu	Safety	7G	Kenna

iii. Identify the primary key for each row.

EmpID (pk)	FName	LName	Manager	Dept	Sector	Dependent name
285	Chala	Dabale	Sifen	Eng.	6G	
365	Yosen		Sifen	Marketing	8G	
458	Simbo	Girma	Derartu	Safety	7G	Singitan
458	Simbo	Girma	Derartu	Safety	7G	Walif
458	Simbo	Girma	Derartu	Safety	7G	Sena
458	Simbo	Girma	Derartu	Safety	7G	Kenna

Second Normal Form (2NF)

iv. Remove subsets of data that apply to multiple rows of a table and place them in separate tables.

EmpID (PK)	FName	LName	Manager	Dept	Sector
285	Chala	Dabale	Sifen	Eng.	6G
365	Yosen		Sifen	Marketing	8G
458	Simbo	Girma	Derartu	Safety	7G

Dependent Name
Singitan
Walif
Sena
Kenna

EmpID(FK)	Dependent Name
458	Singitan
458	Walif
458	Sena
458	Kenna

v. Create relationships between these new tables and their predecessors through the use of foreign keys.

EmpID (PK)	FName	LName	Manager	Dept	Sector
285	Chala	Dabale	Sifen	Eng.	6G
365	Yosen		Sifen	Marketing	8G
458	Simbo	Girma	Derartu	Safety	7G

Third Normal form

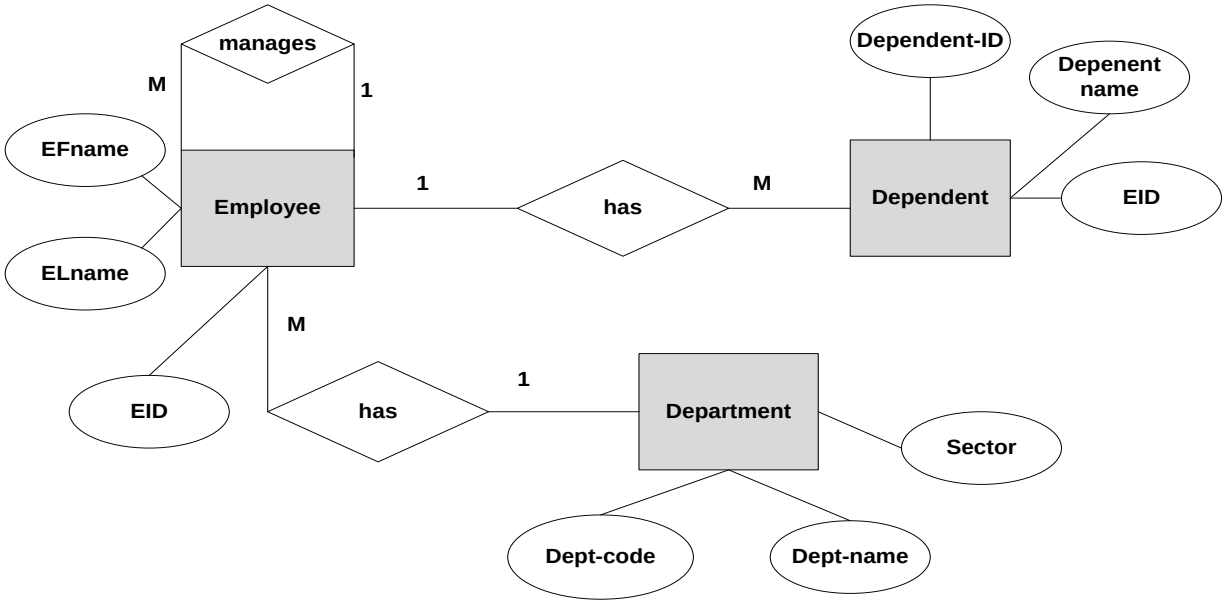
vi. Remove transitive dependency (columns that are not dependent upon the primary key because attributes must depend only on the primary key not depends on another non key attributes).

EmpID (pk)	FName	LName	Manager	Dept_cod(Fk)
285	Chala	Dabale	Sifen	Tech01
365	Yosen		Sifen	Com01
458	Simbo	Girma	Derartu	Com02

Dept_code (PK)	Dept-name	Sector
Tech01	Eng.	6G
Com01	Marketing	8G
Com02	Safety	7G

EmpID (FK)	Dependent Name
458	Singitan
458	Walif
458	Sena
458	Kenna

vii. Draw ERD Diagram



LAP Test 1: Draw ERD diagram

Requirement

Assume that, ABC College needs to develop a database in order to store information about enrollment and you are invited to design & develop the database based on the following information. Student has Unique-ID, Name, DOB, Gender & Mobile. Each student takes several courses and one course should be taken by many students. Course has unique Course-Code, unique Course-name and Credit hour. The database also expected to store Instructors' profile and Instructor has Unique- ID, Name, Gender, Age, Date of Hire, Year of service which is computed from Date of Hire, Qualification, and Phone. Each instructor may teach more than one courses and one course should be delivered (thought) by different instructors. The database will be designed to store mark, Grade & registration date of each student of those who attend & complete the courses. In addition the database also keeps the total number of hours thought by each instructor.

Based on the given (requirement);

Task #1: identify;

i. Entity and entity type
ii. Entity's attribute and attribute type

Task2: Determine the relationship among the entity and relationship cardinality

Task #3: Draw ERD using Chen notation or Crow's feet Notation

LAP Test 2: Draw ERD diagram

Given: un-normalized attribute

EmpID	Name	Dept Code	Dept Name	Project1	Time	Project2	Time	Project3	Time

Using the above un-normalized table;

<u>Task 1:</u> Normalize the attribute;

Task 2:

1. Using the result of 3NF draw or sketch ERD either by using Chen notation or Crow's foot notation.

L #26	LO #3- Design Data Structures

Instruction sheet

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:

- Confirming primary and foreign keys for tables
- Reviewing client business rules
- Identifying referential integrity constraints
- Establishing database Management System constraints and incorporating database design
- Developing validation rules
- Designing indexes and developing data dictionary
- Documenting database design

This guide will also assist you to attain the learning outcomes stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Confirm primary and foreign keys for tables
- Review client business rules
- Identify referential integrity constraints
- Establish database Management System constraints and incorporating database design
- Develop validation rules
- Design indexes and developing data dictionary
- Document database design

Learning Instructions:

Read the specific objectives of this Learning Guide.

1. Follow the instructions described below.
2. Read the information written in the “Information Sheets”. Try to understand what are being discussed. Ask your trainer for assistance if you have hard time understanding them.
3. Accomplish the “Self-checks” which are placed following all information sheets.
4. Ask from your trainer the key to correction (key answers) or you can request your trainer to correct your work. (You are to get the key answer only after you finished answering the Self-checks).
5. If you earned a satisfactory evaluation proceed to the next learning guide

Information sheet 1	Confirming primary and foreign keys for tables

1.1. Confirming primary and foreign keys for tables

Primary keys (PKs) and foreign keys (FKs) are important elements of relational schemata in various applications, such as query optimization and data integration. However, in many cases, these constraints are unknown or not documented. Detecting them manually is time-consuming and even infeasible in large-scale datasets.

1.1.1. Primary Key Constraints

A table typically has a column or combination of columns that contain values that uniquely identify each row in the table. This column, or columns, is called the primary key (PK) of the table and enforces the entity integrity of the table. Because primary key constraints guarantee unique data, they are frequently defined on an identity column.

When you specify a primary key constraint for a table, the Database Engine enforces data uniqueness by automatically creating a unique index for the primary key columns. This index also permits fast access to data when the primary key is used in queries. If a primary key constraint is defined on more than one column, values may be duplicated within one column, but each combination of values from all the columns in the primary key constraint definition must be unique.

- A table can contain only one primary key constraint.
- A primary key cannot exceed 16 columns and a total key length of 900 bytes.
- The index generated by a primary key constraint cannot cause the number of indexes on the table to exceed 999 non-clustered indexes and 1 clustered index.
- If clustered or non-clustered is not specified for a primary key constraint, clustered is used if there no clustered index on the table.
- All columns defined within a primary key constraint must be defined as not null. If null ability is not specified, all columns participating in a primary key constraint have their null ability set to not null.

1.1.2. Foreign Key Constraints

A foreign key (FK) is a column or combination of columns that is used to establish and enforce a link between the data in two tables to control the data that can be stored in the

foreign key table. In a foreign key reference, a link is created between two tables when the column or columns that hold the primary key value for one table are referenced by the column or columns in another table. This column becomes a foreign key in the second table.

A table can reference a maximum of 253 other tables and columns as foreign keys (outgoing references). SQL Server 2016 (13.x) increases the limit for the number of other table and columns that can reference columns in a single table (incoming references), from 253 to 10,000.

The increase has the following restrictions:

- Greater than 253 foreign key references are only supported for DELETE DML operations. UPDATE and MERGE operations are not supported.
- A table with a foreign key reference to itself is still limited to 253 foreign key references.
- Greater than 253 foreign key references are not currently available for column store indexes, memory-optimized tables, Stretch Database, or partitioned foreign key tables.

Unlike primary key constraints, creating a foreign key constraint does not automatically create a corresponding index.

Foreign key columns are frequently used in join criteria when the data from related tables is combined in queries by matching the column or columns in the foreign key constraint of one table with the primary or unique key column or columns in the other table. An index enables the Database Engine to quickly find related data in the foreign key table. However, creating this index is not required.

Data from two related tables can be combined even if no primary key or foreign key constraints are defined between the tables, but a foreign key relationship between two tables indicates that the two tables have been optimized to be combined in a query that uses the keys as its criteria. Changes to primary key constraints are checked with foreign key constraints in related tables.

Self-Check -1	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Choose the best answer(2pts each)

1. By default a column with primary key constraint should be;
   A. Not null
   B. Unique
   C. All
   D. None
2. A table must contain _____ to ensure entity integrity.
   A. Alternate key
   B. Primary key
   C. Foreign key
   D. Supper key
3. One is correct about foreign key.
   A. It cannot be null
   B. By default it is unique
   C. Its value must match the values of primary key or it may be null
   D. All

Note: **Satisfactory rating - 4 points** **Unsatisfactory - below 4 points**
You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____
Rating: _____

Information sheet 2	Reviewing Client business rules

2.1. Business Rules

Each organization has its own data and information requirements, and its own unique way of conducting its business; therefore, every organization needs its own specific set of business rules.

A business rule is a specific directive that constrains or defines a business activity. Business rules determine how an organization operates and utilizes its data. It also affects how data flows through an organization, how it's propagated to database tables, and how it may be accessed and manipulated. It defines specific instructions or constraints on how certain day-to-day actions should be performed.

Without the existence of business rules, there would be a total lack of control.

Business rules can apply to people, processes, corporate behaviour and computing systems in an organization, and are put in place to help the organization achieve its goals.

It also determines the relationships that exist between entities and tables, and how data is handled by the end-user through processes.

Business rules may be integrated at the back-end database level, the front-end application level, or at the middle tier level in n-tier applications.

In database design, for example, you must choose which data to store in the database; you would not necessarily want or need to store every last piece of data the organization might possibly use. The data you finally choose to store and how you decide to store it will be determined by the way the organization uses its data. Business rules will influence a wide variety of database issues, such as the data you collect and store, the manner in which you define and establish relationships, the types of information that the database can provide, and the very security and confidentiality of the data itself.

Example of business rules:

- An instructor must teach at least one class.
- Many students can register for a class that has been scheduled.
- No more than 15 students can register for a given class, etc.

During database design, business rules are primarily integrated through the use of database constraints. At the database level, business rules are enforced using the following items:

- Database constraints
  - ✓ Primary keys
  - ✓ Foreign keys
  - ✓ Unique constraints
  - ✓ Check constraints
  - ✓ Data types
  - ✓ Data precision and scale
  - ✓ NULL/NOT NULL
- Audit scripts to monitor data
- Database triggers

2.2. Benefits of Business Rules

Using business rules provides a number of advantages. Some of them are;

- Increase efficiency
- Increase productivity
- Ensure consistency
- Promote compliance
- Reduced Complexity, etc.

2.3. Reviewing the Business Rule Specification Sheets

Once you've established the Business Rules you believe to be appropriate, review their specification sheets. Carefully examine the specification sheet and make certain that the rule has been properly established and that all the appropriate areas on the sheet are clearly marked. If you find an error, make the necessary modifications and review it once more. Repeat this process until you've reviewed every Business Rule.

Even if rule may be informal or unwritten, documenting the rules clearly and making sure that they don't conflict is a valuable activity.

Because a carefully managed, rules can be used to help the organization to better achieve goals, remove obstacles, reduce costly mistakes, improve communication, comply with legal requirements, and increase customer loyalty.

Self-Check -2	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Choose the best answer(2pts each)

1. One is incorrect about business rule
   A. Increase productivity
   B. Increase inconsistency
   C. Promote compliance
   D. Reduced Complexity
2. Business rule determines the relationships that exist between entities and tables.
   A. True
   B. False

Short answer

3. List down Database constraints(6)

viii. _____

ix. _____

x. _____

xi. _____

xii. _____

Note: **Satisfactory rating - 5 points** **Unsatisfactory - below 5 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet 3	Identifying referential integrity constraints

3.1. Referential integrity constraint

The referential integrity constraint says that if a relation R2 includes a foreign key FK matching the primary key PK of other relation R1, then every value of FK in R2 must either be equal to the value of PK in some tuple of R1 or be wholly null (each attributes value participating in that FK value must be null). R1 and R2 are not necessarily distinct.
The justification for referential integrity constraint is:
If some tuple t2 from relation R2 references some tuple t1 from relation R1, then tuple t1 must exist, otherwise it does not make sense.
Therefore, a given foreign key value must have a matching primary key value somewhere in the referenced relation if that foreign key value is different from null
Sometimes, for practical reasons, it is necessary to permit the foreign key to accept null values. For each foreign key in the database, the database designer has to answer the following important questions:

- Can the foreign key accept null values? The answer to this question depends, not on the whim of the database designer, but on the policies in effect in the portion of the real-world that is to be represented in the database.
- What should happen on an attempt to delete the primary key value tuple of a foreign key reference? In general, there are three possibilities:
  - ✓ Cascade
  - ✓ Restrict
  - ✓ Nullifies

3.2. Cascading Referential Integrity

By using cascading referential integrity constraints, you can define the actions that the Database Engine takes when a user tries to delete or update a key to which existing foreign keys point. The following cascading actions can be defined.

- NO ACTION: The Database Engine raises an error and the delete or update action on the row in the parent table is rolled back.

- CASCADE: Corresponding rows are updated or deleted in the referencing table when that row is updated or deleted in the parent table. CASCADE cannot be specified if a **timestamp** column is part of either the foreign key or the referenced key. ON DELETE CASCADE cannot be specified for a table that has an INSTEAD OF DELETE trigger. ON UPDATE CASCADE cannot be specified for tables that have INSTEAD OF UPDATE triggers.
- SET NUL: All the values that make up the foreign key are set to NULL when the corresponding row in the parent table is updated or deleted. For this constraint to execute, the foreign key columns must be null able.
- SET DEFAULT: All the values that make up the foreign key are set to their default values if the corresponding row in the parent table is updated or deleted. For this constraint to execute, all foreign key columns must have default definitions. If a column is null able, and there is no explicit default value set, NULL becomes the implicit default value of the column.
- RESTRICT - the update operation is “restricted” to the case where there are no such matching tuples (it is rejected otherwise).

Self-Check -3	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Choose the best answer(2pts each)

1. Database Engine raises an error and the delete or update action on the row in the parent table is rolled back.
   A. SET NUL
   B. NO ACTION
   C. CASCADE
   D. SET DEFAULT
2. The corresponding rows are updated or deleted in the referencing table when that row is updated or deleted in the parent table.
   A. SET NUL
   B. CASCADE
   C. NO ACTION
   D. SET DEFAULT
3. If some tuple t2 from relation R2 references some tuple t1 from relation R1, then;
   A. Tuple t1 must exist
   B. Tuple t2 must exist
   C. Tuple t2 may exist
   D. None

Note: **Satisfactory rating - 4 points** **Unsatisfactory - below 4 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet 4	Establishing database Management System constraints and incorporating database design

4.1. Establishing DBMS constraints

Constraints within a database are rules which control values allowed in columns and also enforce the integrity between columns and tables. They are user-defined structures that let you restrict the behaviours of columns.Constraints enforce limits to the data or type of data that can be inserted/updated/deleted from a table. The whole purpose of constraints is to maintain the data integrity during an update/delete/insert into a table.

4.1.1. Types of constraints

- Not null
- Unique
- Default
- Check
- Key Constraints (PRIMARY KEY, FOREIGN KEY)
- Domain constraints

- **NOT NULL:** NOT NULL constraint makes sure that a column does not hold NULL value. When we don't provide value for a particular column while inserting a record into a table, it takes NULL value by default. By specifying NULL constraint, we can be sure that a particular column(s) cannot have NULL values.
- **UNIQUE:** UNIQUE Constraint enforces a column or set of columns to have unique values. If a column has a unique constraint, it means that particular column cannot have duplicate values in a table.
- **DEFAULT:** The DEFAULT constraint provides a default value to a column when there is no value provided while inserting a record into a table.
- **CHECK:** This constraint is used for specifying range of values for a particular column of a table. When this constraint is being set on a column, it ensures that the specified column must have the value falling in the specified range.

- **Key constraints**
  - ✓ **PRIMARY KEY: Primary key** uniquely identifies each record in a table. It must have unique values and cannot contain nulls. In the below example the ROLL_NO field is marked as primary key, that means the ROLL_NO field cannot have duplicate and null values.
  - ✓ **FOREIGN KEY:** Foreign keys are the columns of a table that points to the primary key of another table. They act as a cross-reference between tables. Read more about it **here**.
- **Domain constraints** Each table has certain set of columns and each column allows a same type of data, based on its data type. The column does not accept values of any other data type.Domain constraints are user defined data type and we can define them like this:

Domain Constraint = data type + Constraints (NOT NULL / UNIQUE / PRIMARY KEY / FOREIGN KEY / CHECK / DEFAULT)

Self-Check -4	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Choose the best answer(2pts each)

1. A rule which control the values allowed in columns.
   A. Constraint
   B. Validation
   C. Index
   D. None
2. It is used to specify range of values for a particular column.
   A. Check
   B. Primary key
   C. Default
   D. Unique
3. It uniquely identifies each record in a table.
   A. Primary key
   B. Foreign key
   C. Candidate Key
   D. All

Note: **Satisfactory rating - 4 points** **Unsatisfactory - below 4 points**
You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____
Rating: _____

Information sheet 5	Developing validation rules

5.1. Developing validation rules

Without validating your data, you run the risk of basing decisions on data with imperfections that are not accurately representative of the situation at hand. It is imperative that data is entered correctly in a database. If your data isn't accurate from the start, your results definitely won't be accurate either. That's why it's necessary to verify and validate your data before it is used.

Validation is a process whereby the data entered in the database is checked to make sure that it is sensible. Validation rules verify that the data a user enters in a record meets the standards you specify before the user can save the record. A validation rule can contain a formula or expression that evaluates the data in one or more fields and returns a value of "True" or "False". Validation rules also include an error message to display to the user when the rule returns a value of "True" due to an invalid value.

For example, validation can be utilized to check that only "Male" or "Female" is entered in a sex field. It cannot check whether or not the data entered is correct. It can only check whether or not the data makes sense.

Validation is a way of trying to lessen the number of errors during the process of data input.

Validation is carried out by the computer when you input data. It is a way of checking the input data against the set of validation rules.

The purpose of validation is to make sure that data is

- Logical,
- Rational
- Complete
- Within acceptable limits.

Data is validated or checked to see if it meets certain rules when entered into a field. It cannot be checked for accuracy.

5.2. Database Validation Methods

The most common data validation rules that help maintain integrity and clarity include:

- **Type** – If you make a specific field numeric then it won't allow you to input any letters or other non-numeric characters. Be wary when using the numeric data type. If you use it for fields like phone numbers, it won't allow you to enter spaces, or other human-friendly forms of formatting. For example Data type (ex. integer, float, string)
- **Presence** – This is sometimes called Allow Blank or Mandatory. This type of validation compels the user to enter data in the required field and the field cannot be left empty (No null values).
- **Unique Identifier** – It is essential that one record can be plainly recognized from another record. Generally speaking, each record has one field that functions as a unique identifier for a record. An easy validation check can be done to make sure that a value occurs only once in this field it doesn't matter if there are thousands of records in the database, the check can be carried out just the same.
- **Range Check** – Range check is a validation check which can be applied to numeric fields. This is done to ensure that only numbers within a certain domain can be entered into a field. eg between 1-100.

  Remember that this does not necessarily mean that the data entered will be correct.
- **Format (Input mask)** – data must be entered in a specific way. This is used for a field that requires an entry in a specific format. Examples include date format, postal codes, and driver's license numbers.
- **Restricted Choice(List check) –** There are times that fields in a database have a definite amount of data that can be entered into them. only specific data can be entered, eg male or female or the amount of days in a week are limited to Sunday, Monday, Tuesday, etc.

Programming a database to accept only one of a series of valid choices can prevent errors, and can also serve to lessen the time it takes to input data.

This has different forms like a list box, combo box, or radio button.

Benefits of Restricted Choice:

- ✓ Faster data entry, because it is typically much quicker to select from a list than to type each individual entry.
- ✓ Enhanced accuracy, because it lessens the risk of spelling mistakes.
- ✓ Limits the options to choose from by only displaying the essential choices.

- **Referential Integrity** – If you're using a relational database, then you can impose referential integrity to validate inputs. You can check data inputs in certain fields against values in other tables. For example, in the job database, when a new hire is entered, you could check the supervisor name against the employee table, just like you could check the department name against the department table.
- **Consistent expressions** (ex. Using one of St., Str, Street)
- **Field length check** – only allows a specific number of characters to be entered.

Self-Check -5	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Choose the best answer (2 pts each)

1. Which one of the following ensures that only numbers within a certain domain can be entered into a field?
   A. Range
   B. Format
   C. Type
   D. Presence
2. _____is a process whereby the data entered in the database is checked to make sure that it is sensible.
3. It only allows a specific number of characters to be entered.
   A. Range
   B. Format
   C. Type
   D. Field length

Note: **Satisfactory rating - 4points** **Unsatisfactory - below 4 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet 6	Designing indexes and developing data dictionary

6. Designing indexes and developing data dictionary

6.1. Designing indexes

The index is a type of data structure that is used to optimize the performance of a database by minimizing the number of disk accesses required when a query is processed.

It is used to locate and access the data in a database table quickly. A database table can have one or more indexes associated with it.

An Index is a key built from one or more columns in the database that speeds up fetching rows from the table or view. This key helps a Database like Oracle, SQL Server, MySQL, etc. to find the row associated with key values quickly.
Two types of Indexes are:

- Clustered Index
- Non-Clustered Index

6.1.1. Clustered Index

A clustered index can be defined as an ordered data file. Sometimes the index is created on non-primary key columns which may not be unique for each record.

In this case, to identify the record faster, we will group two or more columns to get the unique value and create index out of them. This method is called a clustering index.

The records which have similar characteristics are grouped, and indexes are created for these group. Clustered indexes are the unique index per table that uses the primary key to organize the data that is within the table. The clustered index ensures that the primary key is stored in increasing order, which is also the order the table holds in memory.

- Clustered indexes do not have to be explicitly declared.
- Created when the table is created.
- Use the primary key sorted in ascending order.

The clustered index will be automatically created when the primary key is defined

Both clustered and non-clustered indexes are stored and searched as B-trees, a data structure similar to a binary tree.

Example: suppose a company contains several employees in each department. Suppose we use a clustering index, where all employees which belong to the same Dept_ID are considered within a single cluster, and index pointers point to the cluster as a whole. Here Dept_Id is a non-unique key.

The previous schema is little confusing because one disk block is shared by records which belong to the different cluster. If we use separate disk block for separate clusters, then it is called better technique.

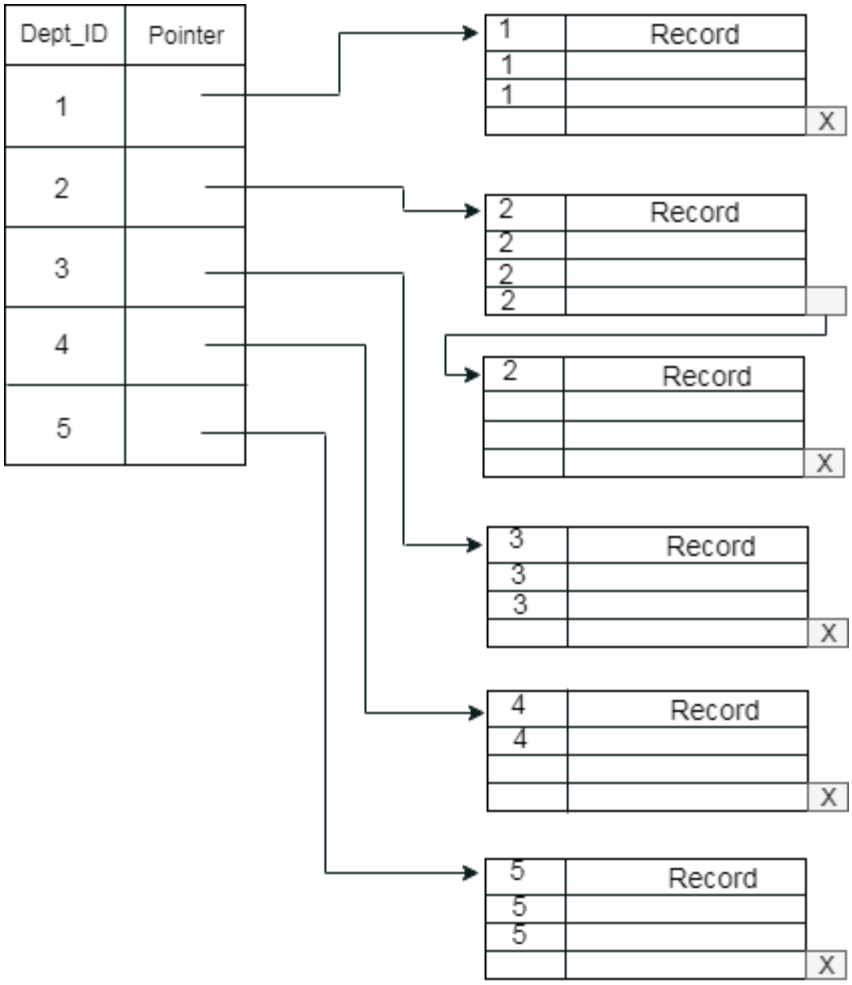


Figure 6.1 **clustered index**

6.1.2. Non-Clustered Indexes

A Non-clustered index stores the data at one location and indices at another location. The index contains pointers to the location of that data. A single table can have many non-clustered indexes as an index in the non-clustered index is stored in different places.

For example, a book can have more than one index, one at the beginning which displays the contents of a book unit wise while the second index shows the index of terms in alphabetical order.

A non-clustering index is defined in the non-ordering field of the table. This type of indexing method helps you to improve the performance of queries that use keys which are not assigned as a primary key. A non-clustered index allows you to add a unique key for a table. Non-clustered indexes point to memory addresses instead of storing data themselves. This makes them slower to query than clustered indexes but typically much faster than a non-indexed column.

You can create many non-clustered indexes. As of 2008, you can have up to 999 non-clustered indexes in SQL Server and there is no limit in PostgreSQL.

You can think of these just like indexes in a book. The index points to the location in the book where you can find the data you are looking for.

6.2. B+ Tree

The B+ tree is a balanced binary search tree. It follows a multi-level index format. In the B+ tree, leaf nodes denote actual data pointers. It ensures that all leaf nodes remain at the same height In the B+ tree, the leaf nodes are linked using a link list. Therefore, a B+ tree can support random access as well as sequential access.

In the B+ tree, every leaf node is at equal distance from the root node. The B+ tree is of the order n where n is fixed for every B+ tree.

It contains an internal node and leaf node.

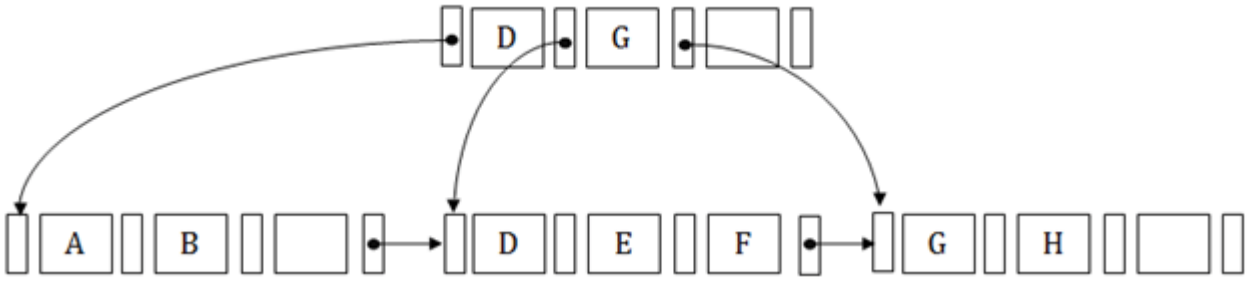


Figure 6.2 Structure of B+ Tree

An internal node of the B+ tree can contain at least n/2 record pointers except the root node.

At most, an internal node of the tree contains n pointers.

The leaf node of the B+ tree can contain at least n/2 record pointers and n/2 key values.

At most, a leaf node contains n record pointer and n key values.

Every leaf node of the B+ tree contains one block pointer P to point to next leaf node.

6.3. Searching Indexes

Indexes use an optimal search method known as binary search. Binary searches work by constantly cutting the data in half and checking if the entry you are searching for comes before or after the entry in the middle of the current portion of data. This works well with B-trees because they are designed to start at the middle entry; to search for the entries within the tree you know the entries down the left path will be smaller or before the current entry and the entries to the right will be larger or after the current entry.

Table 6.1 Searching index

SELECT * FROM friends WHERE name = 'Zack';

friends_name_asc	
Name	**Index**
Andrew	3
Blake	5
Dave	2
Evan	6
Matt	1
Nick	7
Todd	4
Zack	8

Comparing this method to the query of the non-indexed table at the beginning of the article, we are able to reduce the total number of searches from eight to three. Using this method, a search of 1,000,000 entries can be reduced down to just 20 jumps in a binary search.

6.4. Developing Data Dictionary

A data dictionary is also called a metadata repository. A data dictionary is a centralized repository of metadata. Metadata is data about data. The data dictionary contains records about other objects in the database, such as data ownership, data relationships to other objects, and other data.

It provides additional information about relationships between different database tables, helps to organize data in a neat and easily searchable way, and prevents data redundancy issues.

Because of its importance, it is invisible to most database users. Typically, only database administrators interact with the data dictionary.

The data dictionary in general contains information about;

- Names of all the database tables and their schemas.
- Details about all the tables in the database, such as their owners, their security constraints, when they were created etc.
- Physical information about the tables such as where they are stored and how.
- Table constraints such as primary key attributes, foreign key information etc.

Information about the database views that is visible.

Table 6.2: data dictionary

Entity Name	*Entity Description*	*Column Name*	*Column Description*	*Data Type*	*Length*	*Primary Key*	*Nullable*	*Unique*
Employee	An employee is someone who work in a company	CompanyID		integer	10	false	false	false
		ID	For the unique identification of employee records.	integer	10	true	false	false
		name	Name of the employee.	varchar	255	false	true	false
		jobtitle	The position of the employee in a company.	varchar	50	false	true	false

6.5. Types of data dictionary

6.5.1. Active Data Dictionary

If the structures of the database or its specifications change at any point of time, it should be reflected in the data dictionary. This is the responsibility of the database management system in which the data dictionary resides.

So, the data dictionary is automatically updated by the database management system when any changes are made in the database. This is known as an active data dictionary as it is self updating.

6.5.2. Passive Data Dictionary

This is not as useful or easy to handle as an active data dictionary. A passive data dictionary is maintained separately to the database whose contents are stored in the dictionary. That means that if the database is modified the database dictionary is not automatically updated as in the case of Active Data Dictionary.

So, the passive data dictionary has to be manually updated to match the database. This needs careful handling or else the database and data dictionary are out of sync.

6.6. Benefits of a Data Dictionary Service

Number of possible benefits may include:

- Improve knowledge about the data resource.
- Facilitate the design and use of the database.
- Accurate data definitions can be provided.
- Aid the recording, processing, storage and destruction of data.
- Improve documentation and control.
- Promote consistency in data use.
- Easier data analysis.
- Simplify programming.
- Better means of estimating the effect of change, etc.

6.7. Disadvantages

A Data Dictionary is a useful management tool, but at a price.

- It may consume time to create for large DB.
- It needs careful planning, defining the exact requirements designing its contents, testing, implementation and evaluation.

Self-Check -6	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Choose the best answer(2pts each)

1. How many indexes are there in SQL server?

   A. 3 B. 2 C. 4 D.5

2. What is the purpose of index in database?

   A. To enhance the query performance
   B. To provide an index to a record
   C. To perform fast searches
   D. All

3. _____is centralized repository of metadata.

   A. Index
   B. Data dictionary
   C. Table
   D. View

4. A data about data

   A. Meta data
   B. Table
   C. View
   D. Database

Note: **Satisfactory rating - 5 points** **Unsatisfactory - below 5 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet 7	Documenting Database Design

7.1. Documenting Database Design

Documenting database design is the process of recording all database technical and logical aspects for the implementation of database.

Database documentation is important because it meets several needs. Some of them are;

- Provides a common language between business decision makers and IT personnel.
- Provides a shortcut to finding 'hot-spots' by looking at a global functionality chart, one can easily outline the most troublesome parts of the system.
- Facilitates a 'no-panic' rule by having proper documentation and using it, the chances of making a wrong decision are diminished because the risk is easier to assess.
- Makes maintenance easier, and reduces risk when extending or upgrading a system.
- Reduces training costs, by acting as a mediator between newcomers and existing staff. For example, when a consultant or a new hire comes to the company, having up-to-date documentation reduces the time required from the existing staff to transfer knowledge.
- Improves productivity of both newcomers and seasoned employees, reducing the likelihood of costly misunderstandings by providing a glossary of commonly used terms, naming conventions, and even commonly-used strategy patterns.

7.1.1. Documentation types

Different levels of documentation exist at different stages of a project. The process of defining the documentation model is critical, as parts of it will reflect the unique requirements, goals, and resources of the particular organization. The most common classification of the different aspects of database documentation is by tier: **For example**,

- Application logic documentation
- Database logic documentation
- Hardware environment documentation

- Installation
- Maintenance
- DR (disaster recovery) planning and testing, etc.

Self-Check -7	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Short answer

1. _____is the process of recording all database technical and logical aspects. (2pts)
2. List down the importance of database documentation (5pts)
   i. _____
   ii. _____
   iii. _____
   iv. _____
   v. _____

Note: **Satisfactory rating - 4 points** **Unsatisfactory - below 4 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

L #27	LO #4- Design Queries, Screens and Reports

Instruction sheet

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:

- Designing user interface for database
- Designing queries based on requirements
- Designing output reports
- Comparing Physical design with conceptual model with user needs analysis
- Incorporating Changes

This guide will also assist you to attain the learning outcomes stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Design user interface for database
- Design queries based on requirements
- Design output reports
- Compare Physical design with conceptual model with user needs analysis
- Incorporate Changes

Learning Instructions:

Read the specific objectives of this Learning Guide.

1. Follow the instructions described below.
2. Read the information written in the “Information Sheets”. Try to understand what are being discussed. Ask your trainer for assistance if you have hard time understanding them.
3. Accomplish the “Self-checks” which are placed following all information sheets.
4. Ask from your trainer the key to correction (key answers) or you can request your trainer to correct your work. (You are to get the key answer only after you finished answering the Self-checks).
5. If you earned a satisfactory evaluation proceed to the next learning guide

Information sheet 1	Designing user interface for database

1.1. Designing user Interface for Database

User interface design is a part of application design through which the end user can interact with the database. It allows the user to input queries to a database without using the query language. It consists a set of forms that allow the end user to create new records in the database, update or delete existing records, or perform queries.

User-friendly interfaces provide by DBMS may include the following:

- **Menu-Based Interfaces for Web Clients or Browsing-**These interfaces present the user with lists of options (called menus) that lead the user through the formation of a request. Basic advantage of using menus is that they removes the tension of remembering specific commands and syntax of any query language, rather than query is basically composed step by step by collecting or picking options from a menu that is basically shown by the system. Pull-down menus are a very popular technique in Web based interfaces. They are also often used in browsing interface which allow a user to look through the contents of a database in an exploratory and unstructured manner.
- **Forms-Based Interfaces-** A forms-based interface displays a form to each user. Users can fill out all of the form entries to insert a new data, or they can fill out only certain entries, in which case the DBMS will redeem same type of data for other remaining entries. This type of forms are usually designed or created and programmed for the users that have no expertise in operating system. Many DBMSs have forms specification languages which are special languages that help specify such forms.
- **Graphical User Interface -** A GUI typically displays a schema to the user in diagrammatic form. The user then can specify a query by manipulating the diagram. In many cases, GUI's utilize both menus and forms. Most GUIs use a pointing device such as mouse, to pick certain part of the displayed schema diagram.

1.2. Principle for User Interface Design

- **Lay Out:-**Organize areas of the screen or document for different purpose. Standard windows approach screen lay out:
  - ✓ Navigation Button- top
  - ✓ Status bar- bottom
  - ✓ Working area- middle
- **Content Awareness:-** users should be aware of where they are in the system and what is being displayed.
  - ✓ All interface should have titles
  - ✓ Menus should show
    - ❖ Where they are
    - ❖ Where you came from to get there
  - ✓ It should be clear what info is within each area
- **Aesthetics:-** designing interface that are pleasing to the eye'
  - ✓ Interface need to be functional and inviting to use
  - ✓ Avoiding squeezing in too much
  - ✓ Design text carefully- font and size, avoid all capitals..
  - ✓ Colours and patterns should be used carefully
- **User Experience:-** designing the user interface with the users' level of computer experience in mind
  - ✓ Easy to learn the program for novice user
  - ✓ Easy to use the program for expert user
- **Minimize User Effort**:- the interface should be simple and minimize the number of clicks or keystrokes to move from one part of the system to another.

Self-Check -1	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Short answer

1. _____is an application design through which the end user can interact with the database.

2. List down at least 4 Principles to design graphical User Interface.

xiii. _____

xiv. _____

xv. _____

xvi. _____

Note: **Satisfactory rating - 5 points** **Unsatisfactory - below 5 points**

You can ask you teacher for the copy of the correct answers.

2Answer Sheet

Score = _____

Rating: _____

Information sheet 2	Designing queries based on requirements

1.1. Designing queries based on requirements

A query is a way of requesting information from the database. A database query can be either a select query or an action query. A select query is a query for retrieving data, while an action query requests additional actions to be performed on the data, like deletion, insertion, and updating.

Queries are:

- Written in a structured language that is understood by the DBMS.
- Submitted by client applications to the DBMS.
- Executed by the DBMS and returns the result to the client.

To Create a Query you must have to determine:

- What output do you want to see?
- What do you already know (or what constraints are given)?
- What tables are involved?
- How are the tables joined together?

1.1.1. steps to Design query

The query design process or steps are;

1. Identify the fields required in the query output
2. Identify the tables which contain these fields
3. Predict the output of the query
4. Write the query
5. Run and debug the query

Self-Check -2	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Short answer

1. _____is a way of requesting information from the database.(2pts)
2. _____is a query that used for retrieving data.(2pts)
3. _____is a query that perform the deletion, insertion, and updating.(2pts)

Note: **Satisfactory rating - 3 points** **Unsatisfactory - below 3 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet 3	Designing output reports based on requirements

1.1. Designing output reports

Databases are built to store data that may be useful to somebody who owns or uses that data. Data becomes information when it is presented in a format that is useful to a viewer. There are many ways to convert data into information, and the most common is by creating a report.

A report is a design document that displays formatted data from one or more database tables. A report organizes data into a format that it is useful for a specific purpose.

For example, sales data can be organized to discover trends in sales, rank salespeople, or find out which company promotions are generating the most income.

Reports are designed for printing and distribution to an audience. It used to summarize large quantities of data into meaningful information using calculations, record lists, totals, and other reports objects

A database report is a report created from a culmination of queried data visualized for the purposes of analysis, data discovery, and decision-making.

1.1.1. Steps to Designing Reports

Steps for designing simple and efficient report

Step1. **Know what information you need.**

The first step in creating a report is knowing what information you need and ask yourself the following questions:

- What is the purpose of this report?
- What would i like to know?
- Who will need to see this report?
- How will the information on this report be used by each person or group that will see it?

Step2. Determine the data that will be listed on the report.

Use the answers to the questions in step 1 to design the layout of your report. Ask yourself:

- What data needs to be listed on your report in order for it to achieve its purpose?

- Does the list of data need to be aggregated grouped and or sorted in a particular way? If so, what groups, sorts, and aggregated calculations are needed?

Step3. Sketch a sample report

Validate what you've determined in steps 1 and 2 by sketching your report on a piece of paper, word processor, spreadsheet or your favorite graphic layout tool.

Step4. Ensure the appropriate data is being collected.

Collecting the appropriate data means making sure your database is modeled and designed properly. Questions to ask your database designer are:

- Is the database storing the data I need for the report ?
- Is my database structured properly in a way that I can use the stored data to create the report?

This step requires knowledge of good database modeling and design.

Step5. Create your report.

Use your sample report to create a layout with the technology you are using to present data from your database to the people who need your report.

Some examples of such technology are:

- A web page built with php and mysql
- A content management system like WordPress or Drupal
- A database software platform like FileMaker Pro
- A reporting tool like Tableau.

Reports are an important and useful tool to help you understand your business and inform your decision-making processes. Be sure that your reports stay simple, effective and purposeful using the guidelines above.

1.1.2. Reporting Design Values

A reporting database should meet these non-functional requirements:

- **Flexibility**: quickly deliver on requirements changes as users demand new reporting data, different metrics, additional reference data, and changes in level of report detail
- **Simplicity**: make report SQL as simple as possible
- **Reliability**: data for reporting is available in a timely manner and is consistent with operational sources
- **Security**: users have access only to data they need

Self-Check -3	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Short answer

1. List down the steps required to creating Report.(5 pts)

xvii. _____

xviii. _____

xix. _____

xx. _____

xxi. _____

2. _____is a design document that displays formatted data from one or more database tables.(2 pts)

Note: **Satisfactory rating - 4 points** **Unsatisfactory - below 4 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet 4	Comparing Physical design with conceptual model with user needs analysis

4.1. Comparing Physical design with conceptual

As a database designer and implementer you should have to verify and validate the compatibility of requirements at Conceptual, logical and physical model based on the business, user and system requirements.

- **Conceptual Data Model**

A conceptual data model is a summary-level data model that is most often used on strategic data projects. It typically describes an entire enterprise. Due to its highly abstract nature, it may be referred to as a conceptual model.

Common characteristics of a conceptual data model:

- ✓ Enterprise-wide coverage of the business concepts.
- ✓ Designed and developed primarily for a business audience
- ✓ Contains around 20-50 entities (or concepts) with no or extremely limited number of attributes described.
- ✓ Contains relationships between entities, but may or may not include cardinality and nullability.
- ✓ Entities will have definitions.
- ✓ Designed and developed to be independent of DBMS, data storage locations or technologies.

- Logical Data Model

A logical data model is a fully-attributed data model that is independent of DBMS, technology, data storage or organizational constraints. It typically describes data requirements from the business point of view. While common data modeling techniques use a relational model notation, there is no requirement that resulting data implementations must be created using relational technologies.

Common characteristics of a logical data model:

- ✓ Typically describes data requirements for a single project or major subject area.
- ✓ May be integrated with other logical data models via a repository of shared entities
- ✓ Typically contains 100-1000 entities, although these numbers are highly variable depending on the scope of the data model.
- ✓ Contains relationships between entities that address cardinality and nullability (optionality) of the relationships.
- ✓ Designed and developed to be independent of DBMS, data storage locations or technologies.
- ✓ Data attributes will typically have datatypes with precisions and lengths assigned.
- ✓ Data attributes will have nullability (optionality) assigned.
- ✓ Entities and attributes will have definitions.
- ✓ All kinds of other meta data may be included (retention rules, privacy indicators, volumetrics, data lineage, etc.)

A logical data model will normally be derived from and or linked back to objects in a conceptual data model.

- **Physical Data Model**

A physical data model is a fully-attributed data model that is dependent upon a specific version of a data persistence technology. The target implementation technology may be a relational DBMS, an XML document, a NoSQL data storage component, a spreadsheet or any other data implementation option.

Common characteristics of a physical data model:

- ✓ Typically describes data requirements for a single project or application. Sometimes even a portion of an application.
- ✓ May be integrated with other physical data models via a repository of shared entities
- ✓ Typically contains 10-1000 tables, although these numbers are highly variable depending on the scope of the data model.

- ✓ Contains relationships between tables that address cardinality and nullability (optionality) of the relationships.
- ✓ Designed and developed to be dependent on a specific version of a DBMS, data storage location or technology.
- ✓ Columns will have datatypes with precisions and lengths assigned.
- ✓ Columns will have nullability (optionality) assigned.
- ✓ Tables and columns will have definitions.
- ✓ Will also include other physical objects such as views, primary key constraints, foreign key constraints, indexes, security roles, store procedures, XML extensions, file stores, etc.
- ✓ The diagram of a physical data model may show only a tiny percentage of the meta data contained within the model.

Self-Check -4	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Short answer

1. List down at least 3 characteristics of conceptual model (3 pts).
   - **i.** _____
   - **ii.** _____
   - **iii.** _____
2. List down at least 3 characteristics of logical model (3 pts).
   - **i.** _____
   - **ii.** _____
   - **iii.** _____
3. List down at least 3 characteristics of physical model (3 pts).
   - **i.** _____
   - **ii.** _____
   - **iii.** _____

Note: **Satisfactory rating - 5 points** **Unsatisfactory - below 5 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet 5	Incorporating change

Introduction

Change is inevitable and also necessary for business survival and success. So we better have solutions that enable us to better manage these inevitable changes.

Businesses must be able to meet customer expectations that are constantly changing, while at the same time growing revenue and increasing profitability. To attain these goals, businesses need to be able to change and enhance products and services to meet and exceed the offerings of competitors.

Applications and systems change to do different things with existing data to include more or different types of data. The types and structures of data changes requiring modifications to the underlying database schema to accommodate the new data.

5.1. Incorporating change

Most of today's business information is managed within the context of a complex, computerized business application. Almost every type of information you can think of from customer information to vendor information to product specifications to payroll information is managed by computer. And, this information is dynamic, it is always changing. Businesses must prepare to handle the challenge of managing and controlling their constantly changing information.

A DBMS is used to store the majority of today's computerized data. So when that data changes, the databases used to store the data must also change. If the data is not reliable and available, the system does not serve the business, but instead, threatens the health of the business. Today's major RDBMS products do not support fast and efficient database structure changes. Each RDBMS provides differing levels of support for making changes to its databases. But none easily supports every type of change that might be required. One quick example: most RDBMSs today do not enable a column to be added to the middle of an existing row. To do so, the table must be dropped and recreated with the new column in the middle. When the table is dropped the data is deleted, unless the DBA was wise enough to first unload the data. The indexes were dropped when the table was dropped, so unless the DBA knows this and recreated the indexes too, performance will suffer. The same is true for database security; when the table was dropped all security for the table was also dropped.

Other types of database change that are required from time to time include:

- Changing the name of a database, table, view, or column
- Modifying a stored procedure, trigger, or user-defined function
- Changing or adding relationships using referential integrity features
- Changing or adding database partitioning.
- Moving a table from one database, dbspace, or tablespace to another.
- Rearranging the order of column in a table.
- Changing a column's data type or length.
- Adding or removing columns from a table.
- Changing the primary key without dropping and adding the primary key.
- Adding or removing columns from a view.
- Changing the select statement on which a view is based.
- Changing the columns used for indexing.
- Changing the uniqueness specification of an index.
- Clustering the table data by a different index.
- Changing the order of an index (ascending or descending) ,etc.

5.1.1. Change Management Requirements

To successfully implement effective change management it is imperative to understand a set of basic requirements. The following factors needs to be incorporated into your change management discipline in order to ensure success:

- **Pro activity-** Proactive change, which can eliminate future problems, is an organization's most valuable type of change. The earlier in the development cycle that required changes can be identified and implemented, the lower the overall cost of the change will be.
- **Intelligence-** When implementing a change, every aspect of the change needs to be examined because it could result in an unanticipated cost to the company. The impact of each change must be examined and incorporated into the change process. Because a simple change in one area may cause a complex change in another area. Intelligence in the change management process often requires a thorough analysis including an efficient and low-risk implementation plan.

- **Planning analysis-** Planning maximizes the effectiveness of change. A well-planned change saves time. It is always easier to do it right the first time than to do it again after the first introduced change proved less than effective. An effective organization will have a thorough understanding of the impact of each change before allocating resources to implementing the change.
- **Impact analysis-** Comprehensive impact and risk analysis allows the organization to look at the entire problem and examine the involved risk to determine the best course of action. A single change usually can be accomplished in many different ways. But the impact of each type of change may be considerably different. Some carry more risks: the risk of failure, the risk associated with a more difficult change, the risk of additional change being required, the risk of causing downtime, etc. All considerations are important when determining the best approach to implementing change.
- **Automation-** With limited resources and a growing workload, automating a process serves to reduce the human-error factor and to eliminate more menial tasks from over burdened staff.
- **Standardization of procedure-** Attrition, job promotions and job changes require organizations to standardize processes to meet continued productivity levels. An organized and thoroughly documented approach to completing a task reduces the learning curve, as well as training time.
- **Reliable and predictable process** - When creating any deliverable, a business needs to know that all of the invested effort is not wasted. Because time is valuable, a high level of predictability will help to ensure continued success and profitability. Reliability and predictability are key factors in repeatedly producing a quality product.
- **Availability-** Most changes require down time to implement the change. To change an application the application must come down. The same is true of databases. But high availability is required of most applications these days, especially for e-businesses. Reducing the amount of downtime required to make a change will increase application availability.
- **Quick and efficient delivery**- With most products and services there is a consumer demand for quick turnaround. Profitability is at its best when a product is first-to-market. Conversely, the cost of slow or inefficient delivery of products can be enormous. In the case of implementing change, faster ***is*** better.

Self-Check -5	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Short answer

1. What is Change?(2 pts)

   _____

   _____

   _____

2. List down at list 6 types of database change that are required from time to time.(6pts)

   i. _____
   ii. _____
   iii. _____
   iv. _____
   v. _____
   vi. _____

3.

4. In which normal form all attributes are depend only on primary key.

   A. 1NF B. 2NF C. 3NF D. BCNF E. None

5. Which normal form should ensure the atomicity of value?

   A. 5NF B. 1NF C. 3NF D. BCNF E. 2NF

6. The goal of normalization is_____.

   A. Improving performance
   B. Eliminating Redundancy
   C. Ensuring Security
   D. All
   E. None

Note: **Satisfactory rating - 5 points** **Unsatisfactory - below 5 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

L #28	LO #5- Design access and security systems

Instruction sheet

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:

- Reviewing business security plan
- Designing password and access system
- Identifying multiple-user requirements
- Developing client access profiles

This guide will also assist you to attain the learning outcomes stated in the cover page.

Specifically, upon completion of this learning guide, you will be able to:

- Review business security plan
- Design password and access system
- Identify multiple-user requirements
- Develop client access profiles

Learning Instructions:

Read the specific objectives of this Learning Guide.

1. Follow the instructions described below.
2. Read the information written in the “Information Sheets”. Try to understand what are being discussed. Ask your trainer for assistance if you have hard time understanding them.
3. Accomplish the “Self-checks” which are placed following all information sheets.
4. Ask from your trainer the key to correction (key answers) or you can request your trainer to correct your work. (You are to get the key answer only after you finished answering the Self-checks).
5. If you earned a satisfactory evaluation proceed to the next learning guide

Information sheet 1	Reviewing business security plan

Introduction

Database security relates to protecting the database against intentional or unintentional threats, using elements of control that may or may not be based on the computing equipment. It should be established to protect the data from unauthorized users. It is important to restrict access to the database from individuals not requiring access. As a general rule, you should be able to limit the access to the data at a very low level. A lack of security can also cause problems that leave decision makers with the impression that they have a poor database.

Most information required for security plan will be gathered from interviews during analysis. In order to plan for database security during database design and enforce the security after implementation, all users of the database must first be identified. There are potentially different categories of users for any information system, ranging from end users to administrators of the information system.

Often the problems related to security are complex and may involve legal, social or ethical aspects, or issues related to policies implemented or related to control of the physical equipment.

Security considerations not only apply to data contained in the database, because the gaps of security can affect other parts of the system, which in turn may affect the database. In consequence, by focusing on database security alone will not ensure a secure database. All parts of the system must be secure: the database, the network, the operating system, the building in which the database resides physically and the persons who have an opportunity to access the system.

Designing and implementing a secure database involves achieving the following objectives:

- **Privacy**, which means that data should not be known by unauthorized users;
- **Integrity**, which means that only authorized users can change data;
- **Availability**, which means that authorized users should not be denied access;

To achieve these objectives, a clear security policy that describes which users can access the database, and to which data they can access must be developed.

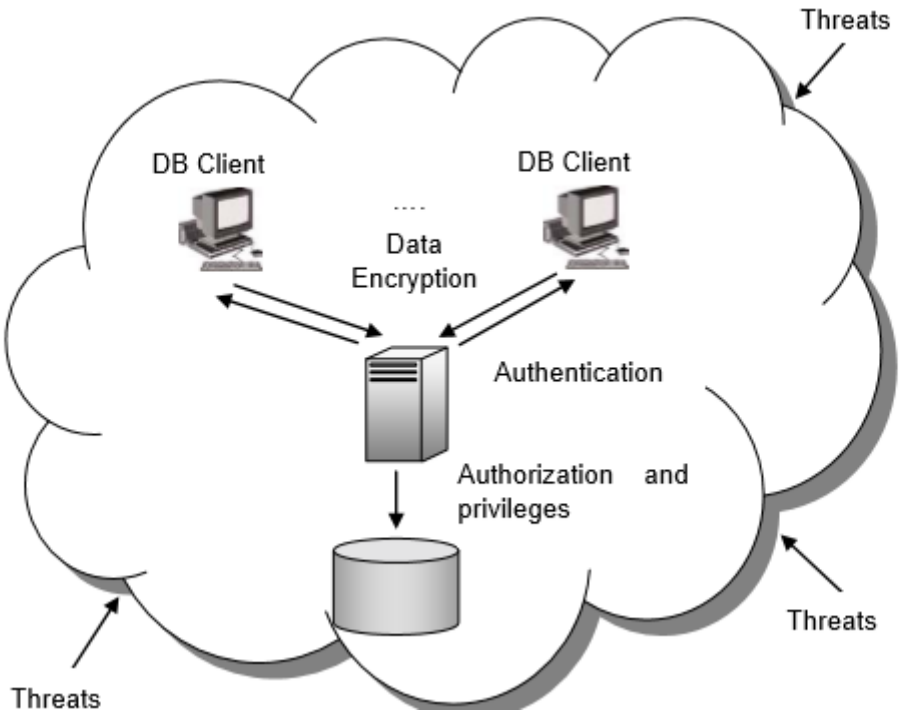


Figure 1.1 – Database security big picture

The person responsible for the security of a database is the database administrator (DBA), who needs to take into account the various threats to the system. The DBA defines the authorization rules that determine who can access the database, what parts of it can be accessed by each user, and which operations are allowed.

1.1. Business security plan

It is necessary for organizations to identify the types of threats that they may be exposed to and initiate plans and appropriate measures, taking into account the cost of their implementation.

A system security plan is a formal plan that defines the plan of action to secure a computer or information system. It provides a systematic approach and techniques for protecting a computer from being used by unauthorized users, guards against worms and viruses as well as any other incident/event/process that can jeopardize the underlying system's security.

A system security plan is primarily implemented in organizational IT environments. It can be a proposed plan to protect and control an information system, or a plan that is already

in implementation. It is usually created using the organization/IT environment security policy as the benchmark.

Typically a system security plan includes:

- List of authorized personnel/users that can access the system
- Level of access/tiered access, or what each user is allowed and not allowed to do on the system
- Access control methods, or how users will access the system (user ID/password, digital card, biometrics)
- Strengths and weaknesses of the system and how weaknesses are handled
- May also include system backup/restoration procedures

The purpose of a Security Plan is to enhance and maintain the security of a licensee's operation by assessing a site for security risks, developing measures to address security issues by incorporating current security programs and developing new ones if necessary, and formalizing responses to and reporting procedures. (System-security-plan)

1.1.1. Contents of Security plan

- **Describe security scope**: What is it intended to cover. For a small company the security plan scope might be the entire organization; for a larger organization, it might be limited to just one location or one department. The scope may also be limited by the type of threats it covers. Often a separate security plan is written just for IT related threats since these require specialized knowledge to understand and address.
- **Security assessment**: This is the part of the plan which answers the question: "where are we now?" The assessment needs to identify what we need to defend (people, locations, equipment, confidential information, service availability). Unless we know what we are defending, it's not possible to determine which threats we need to be concerned with. We need to determine the threats we need to defend against. These may include:
  - ✓ Physical threats (e.g. Theft, arson, sabotage),
  - ✓ Computer-related threats (e.g. Viruses, spam, malware, network intrusion)
  - ✓ Insider threats (e.g. Fraud, workplace violence, information theft or disclosure)
  - ✓ Natural threats (e.g. Earthquake, landslide, hurricane, tsunami, snowstorm, etc.)
  - ✓ Information threats (e.g. theft of trade secrets, customer lists)

For each threat we need to determine the risk; the combination of both how likely it is to occur and its impact on the organization.

We also need to determine what precautions are already in place to either reduce the likelihood of the threat or to reduce its impact. This may include physical measures (burglar alarms, fences, and firewalls), procedural controls, staff policies, and staff training.

- P**rioritize the risks:** Which are we going to take action on first, which can we safely ignore for now, and which can we safely ignore for the foreseeable future?

Note that rarer threats which may significantly impact the organization (fire, flood, earthquake, etc.) will often be excluded from the plan's scope and addressed in a separate business continuity plan or a disaster recovery plan, since these threats and the actions which must be undertaken to address them are further from the normal day-to-day running of the organization

Finally the plan needs to identify the actions we are going to take and when we are going to do them. Without this step, we just have a security assessment, not a security plan.

The actions may be of a one-off or of a continuing nature. They might involve:

- ✓ Purchase and installation of equipment (e.g. Security cameras, firewalls)
- ✓ Changes to procedures (e.g. Ensure all visitors have a visitor badge)
- ✓ Additional staff training (e.g. Handling of confidential material)
- ✓ Exercises (e.g. Fire drills, earthquake drills, lockdown drills)
- ✓ Curtailing of risky activities (e.g. No more on-site storage of flammable liquids)
- ✓ The creation of contingency plans for specific threats

Whatever the actions are, it is important that specific individuals need to be assigned the responsibility to carry out the required actions. The individual chosen must have the skills, time, budget, and resources to carry out the action.

- **Updated the plan regularly:** As the organization's assets change and the organization learns more about the threats to its operations there should typically be a formal security plan review once a year or whenever a significant change in the organization's operations occurs.

1.1.2. The need for database security plan

Everyone recognizes that any loss of availability or loss of data could be potentially disastrous. Threats on data security may be any situation or event, intentional or

unintentional, which will negatively affect a system and finally an organization. The damage caused can be tangible (loss of data) or intangible (loss of credibility or trust of a client). Any threat should be viewed as a potential breach in the security system, which if successful, will have an impact on the organization.

Some examples of potential threats that may occur include:

- Data use by a user having disk media access
- Collection or unauthorized copying of data
- Alteration of programs
- Illegal access by a hacker
- Theft of data, programs or equipment
- Inadequate training of staff
- Unauthorized disclosures of data
- Calamities (fires, floods, bombings)
- Breaking or disconnecting cables
- Viruses

1.1.3. Method of protection

Database security aims to minimize losses caused by the events listed above in an efficient way in terms of cost, and without undue constraints on users. Since computer-based crime is booming and this type of crime can threaten all the components of a system, appropriate security measures are vital. The most used measures to ensure protection and integrity of data include:

1.1.3.1. **Views**

In conjunction with the authorization process, views are an important component of the security mechanism provided by a relational DBMS. Views allow a user to see information while hiding any information that the user should not be given access to. A view is the dynamic result of one or more relational operations that apply to one or more base tables to produce another table. A view is always based on the current data in the base tables from which it is built. The advantage of a view is that it can be built to present only the data to which the user requires access and prevent the viewing of other data that may be private or confidential. A user may be granted the right to access the view but not to access the base tables upon which the view is based.

1.1.3.2. Integrity Control

The aim of integrity control is to protect data from unauthorized use and update, by restricting the values that may be held and the operations that can be performed on data. Integrity controls may also trigger the execution of some procedure, such as placing an entry in a log that records what users have done what with which data. There are more forms of integrity controls. The first form that we discuss is the integrity of the domain. A domain may be viewed like a way to create a user-defined data type. Once a domain is created it may be assigned to any field as its data type. Consequently any value inserted in the field must belong to the domain assigned. When a domain is created, it may use constraints (for example a CHECK constraint) to restrict the values to those which satisfy the imposed condition. An important advantage of a domain is that if it must change then it can be modified in a single place.

Assertions are also powerful constraints that enforce some desirable database conditions. They are checked automatically by the DBMS when transactions are run involving tables or fields on which assertion exists. If the assertion fails, the DBMS will generate an error message.

For security purposes one can use triggers as well. Triggers consist of blocks of procedural code that are stored in a database and which run only in response to an INSERT, UPDATE or DELETE command. A trigger, which includes an event, condition, and action, may be more complex than an assertion. It may prohibit inappropriate actions, it may cause special handling procedures to be executed, or it may cause a row to be written to a log file in order to store important information about the user and transactions made to sensitive data.

1.1.3.3. Data encryption

Sensitive and personal data stored within the database tables and critical data transmitted across the network, such as user credentials (user ID and password), are vulnerable and should be protected against intruders.

Encryption is the process of encoding data by a particular algorithm, which makes it impossible for a program to read data without the decryption key. Usually encryption protects data transmitted through communication lines. There are more techniques for encoding data, some of which are reversible, while others are irreversible. Irreversible techniques do not allow knowledge of the original data, but can be used to obtain valid

statistical information. Any system that provides encryption facilities must also provide adequate routines for decoding, routines which must be protected by proper security.

1.2. Privacy authentication and authorization

1.2.1. Authentication

Authentication is the process by which users are identified by the DBMS and prove their identity to access the database. User and group identity validation is achieved through security facilities located outside of the DBMS that is, they are performed as part of the operating system or using a third-party security facility, such as Kerberos or Lightweight Directory Access Protocol (LDAP).

Authentication of a user requires two elements: a user ID and an authentication token.

The user ID allows the security component to identify the user and by supplying the correct authentication token (a password known only by the user and the security component), the user identity is verified.

After successful authentication of a user, the authenticated user ID is mapped to an authorization ID.

1.2.2. Authorization

After a user is authenticated, it is necessary to determine whether that user is authorized to access certain data or resources.

Authorization is the process of granting privileges, which allows a subject to have legitimate access to a system or an object in a system. The definition of authorization contains the terms subject and object. The subject refers to a user or program and the term object addresses a table, a view, an application, procedure or any other object that can be created in the system.

Authorization control can be implemented by software elements and it can regulate both systems and objects to which a user has access and what a user can do with them. For this reason, the authorization is also called access control. For example, a user may be authorized to read records in a database, but cannot modify or insert a record. Authorization rules are controls incorporated in the DBMS that restrict the action that user may take when they access data.

1.3. Security policies and procedures

It is mandatory to establish a set of administrative policies and procedures in order to create a context for effectively implementing security measures. There are two types of security policies and procedures:

- Personnel controls and
- Physical access controls.

1.3.1. Personnel control

Often the greatest threat to data security is internal rather than external, so adequate controls of personnel must be developed and followed. The security authorization and authentication procedures must be enforced by procedures which ensure a selection hiring process that validate potential employees regarding their backgrounds and capabilities. Employees should be trained in those aspects of security that are relevant to their jobs and encouraged to be aware of and follow standard security measures. If an employee should need to be let go then there should be a set of procedures to remove authorizations and authentications and to notify other employees about the status change.

1.3.2. Physical access control

An important physical access control is related to limiting access to particular areas within a building. A proximity access card can be used to gain access to a secured area. In this case, each access can be recorded in a database. When it is necessary a guest should be issued badges and escorted in these areas. Sensitive equipment can be controlled by placement in a secure area. Other equipment can be locked in a desk or have an alarm attached.

Backup data media should be kept in fireproof data safe or kept outside in a safe location. Procedures must explicitly state a schedule for moving media or disposing of media, and establish labelling and indexing of such materials stored.

Lately, new companies are trending towards an increasingly mobile nature of work. Laptops are very susceptible to theft, which puts data on the laptop at risk. Encryption and multiple factor authentication can protect data. Antitheft devices, like security cables or geographical tracking chips, can help determine theft and quickly recover laptops on which critical data are stored.

1.4. How to improve database Security

To improve database security:

- Change administrator passwords regularly.
- Force users to change passwords frequently.
- Discourage the sharing of passwords.
- Remove inactive user accounts.
- Remove non-employee user accounts.
- Perform random monitoring of all activities.
- Perform database auditing.
- Educate the end user.
- Conduct brief security training sessions

Self-Check -1	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Choose the best answer(2pts each)

1. Protecting computers and information they contain against unwanted access, damage, modification or destruction is called
   A. Computer monitoring
   B. Electronic policing
   C. Audit control
   D. Computer security
2. Restricted access to the server computer room is present in a form of
   A. Logical security
   B. Enterprise security
   C. Physical security
   D. User security
3. Measurement of things such as finger prints and retinal scan used for security access is called

A. Bio metrics
B. Bio measurement
C. Computer security
D. Smart weapon machinery

4. Data encryption techniques are particularly useful for
   A. Reducing storage space requirements
   B. Improving data integrity
   C. Protecting data in data communication systems
   D. All of the above

Note: **Satisfactory rating - 4 points** **Unsatisfactory - below 4 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet 2	Designing Password and Access system

3.1. Designing Password and Access system

A password is a string of characters used to verify the identity of a user during the authentication process. Passwords are typically used in conjuncture with a username; they are designed to be known only to the user and allow that user to gain access to a device, application or website.

Passwords can vary in length and can contain letters, numbers and special characters. Other terms that can be used interchangeably are passphrase for when the password uses more than one word, and pass code and passkey for when the password uses only numbers instead of a mix of characters, such as a personal identification number.

3.1.1. Creating a secure password

Many organizations set password policies so employees create strong passwords and use best practices for their login credentials. Some of the best practices for password requirements include:

- A minimum length of eight characters with a limit of anywhere from 16 to 64 characters or possibly even higher;
- The inclusion of both uppercase and lowercase letters with case sensitivity;
- The use of at least one number; and
- The use of at least one special character.

Policies should prohibit certain characteristics in weak passwords.

For instance, any recognizable personal information -- such as birthdates, names of children, or favorite sports teams -- should not be part of a password, as well as any words or phrases that are on a password blacklist.

Password blacklists are lists of passwords that are too easily cracked and thus are not secure enough to use. Common offenders that wind up on blacklists include "123456," "password," "football," "qwerty" and so on.

Strong password policies also include a time limit for user passwords. This means that passwords will expire after a set period of time -- such as 90 or 180 days -- and users will be forced to change their password to prevent the reuse of the same couple of

passwords. The policy may also require the user to create a password that is different from any other they have used in the last six to 12 months.

While strong passwords are ideal, users often forget them. As a result, password recovery methods might vary depending upon access to an application, website or device. Methods might include answering security questions, confirming emails asking if users want to reset their passwords, or entering numerical security codes sent via text to a mobile phone to authenticate users who need to reset passwords or recover the original one.

The use of passwords in computing dates back to 1961 when the Massachusetts Institute of Technology introduced the Compatible Time-Sharing System or CTSS. The CTSS was one of the first time-sharing operating systems and had a LOGIN command that required a user password.

Today, many enterprises are looking to reduce their reliance on passwords and/or completely eliminate by developing technology standards that replace conventional passwords with alternative authentication techniques.

There are many authentication options available today so that users do not have to rely on passwords that can be easily cracked or compromised.

These options include:

- **Two-factor authentication (2FA)** -- 2FA requires users to provide two authentication factors that include a combination of something the user knows -- like a password or PIN; something the user has -- like an ID card, security token or smartphone; or something the user is -- biometrics.
- **Biometrics** -- Biometric technology is mainly used for identification and access control. Biometrics includes physiological characteristics such as fingerprints or retinal scans, and behavioral characteristics such as typing patterns and voice recognition.
- **Multifactor authentication (MFA)** -- MFA is similar to 2FA except that it is not limited to only two authentication factors. It also uses something the user knows, something the user has and something the user is.
- **Tokens** -- A security token is a physical hardware device like a smart card or key fob that a user carries to authorize access to a network.

- **One-time passwords (OTP)** -- An OTP is an automatically generated password that only authenticates a user for a single transaction or session. These passwords change for every use and are typically stored on security tokens.
- **Social logins** -- A social login in when users can authenticate themselves on applications or websites by connecting to their social media account such as Facebook or Google instead of using a separate login for each and every site.

3.1.1.1. Password strength

Password strength is a measure of the effectiveness of a password in resisting guessing and brute-force attacks. In its usual form, it estimates how many trials an attacker who does not have direct access to the password would need, on average, to guess it correctly. The strength of a password is a function of length, complexity, and unpredictability.

Using strong passwords lowers overall risk of a security breach, but strong passwords do not replace the need for other effective security controls.

3.1.1.2. Characteristics of Weak Passwords

- Repeating previously used passwords.
- Names of close family members or friends.
- Your name.
- Words in the dictionary.
- Common names.
- Short password (under 6 characters)
- Repeating your login code.
- Keyboard patterns and swipes (i.e., 123456 or QWERTY)

Characteristics of strong passwords

- At least 8 characters—the more characters, the better
- Never recycle
- use a password that contain at least *one of each* of the following:
  - ✓ digit (0..9)
  - ✓ letter (a..Z)

- ✓ punctuation symbol (e.g., !)
- ✓ control character (e.g., ^s, Ctrl-s)

- never record a password anywhere
- use a different password for each system/context
- be aware Trojan horse programs can masquerade as login prompts so always reset the system as appropriate to obtain a trusted login prompt
- check for keyboard buffer devices/software that intercept keystrokes (including password capture)
- change password occasionally
- change your password immediately if you suspect it has been “stolen”
- “Passwords should be protected in a manner that is consistent with the damage that could be caused by their compromise.
- monitor for possible eavesdroppers during entry of password
- Do not use the "Remember Password" feature of applications (e.g., Microsoft® Internet Explorer®), etc.

Self-Check -2	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Choose the best answer (2pts each)

1. Which one of the following is considered as weak password?
   A. Recycling passwords
   B. Using a password that contain 2 characters
   C. Recording passwords
   D. All
   E. None
2. An authentication method that include physiological characteristics such as fingerprints or retinal scans, and behavioral characteristics such as typing patterns and voice recognition.
   A. Tokens B. Social logins C. Biometrics D. All E. None
3. _____ a string of characters used to verify the identity of a user during the authentication process.
   A. Authentication B. Authorization C. Biometrics D. Password E. None

Answer Sheet

Note: **Satisfactory rating - 3 points** **Unsatisfactory - below 3 points**

You can ask you teacher for the copy of the correct answers.

Score = _____

Rating: _____

Information sheet 3	Identifying multiple-user requirements

3.1. Identifying multiple-user requirements

Databases are shared resources. You need to expect and plan for the likelihood that several users will attempt to access the same data at the same time. A multiuser database system must allow multiple users access to the database at the same time. As a result, the multiuser DBMS must have concurrency control strategies to ensure several users access to the same data item at the same time, and to do so in a manner that the data will always be correct data integrity.

Concurrency control is a method of locking data so that updates and changes to data are managed.

There are different levels of locking, these include:

- **Database lock** - entire database is locked to one user & unavailable to other users
- **Table lock** - entire table containing requested record is locked to one user
- **Block or page lock**- physical storage block containing requested record is locked
- **Record level lock**- only requested record is locked
- **Field level lock** - only the field within requested record is locked

Depending on the sensitivity and accuracy of the data within the database you may choose one of two different lock types; these are:

- **Exclusive locks (write locks)** - other users cannot read (or update) the record until the lock is released; this prevents other users from placing locks of any kind.
- **Shared locks (read locks)** - other users can read a record but not perform any updating; this prevents another user from placing an exclusive lock on the record
- **Deadlocks** can occur when two users lock common resources.
  - ✓ User 1 locks table "A" and needs to lock table 'B" to complete the transaction.
  - ✓ User 2 locks Table "B" and needs table "A" to complete their transaction.

Self-Check -3	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Choose the best answer(2pts each)

1. Other users cannot read or update the record until the lock is released; this lock type is;

   A. Exclusive locks B. Shared locks C. Deadlock D. All

2. A lock type that occur when two users lock common resources

   A. Shared locks B. Exclusive locks C. Deadlock D. Update lock

3. A method that DBMS employ to manage simultaneous user access to the same resource or data.

   A. Deadlock B. Concurrency C. Concurrency control D. None

Note: **Satisfactory rating - 3 points** **Unsatisfactory - below 3 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet 4	Developing client access profiles

4.1. Developing client access profiles

It is information kept on a computer that gives details about a user, for example their name, password, and the parts of the system they are allowed to use.

Access profiles provide the necessary IDs and passwords for the database logon operations that occur in the background. Every user profile must be assigned to an access profile, by way of a Symbolic ID. The Access ID consists of a relational database management system (RDBMS) ID and a password.

Access IDs are used:

- When an application server initializes and connects to a database.
- When a developer or power user signs in to the database directly (two-tier).
- When batch programs connect to the database.

Access profiles enable you to minimize the number of users who need to know system administrator passwords. In fact, only one person needs to know these passwords. That person can create the required access profiles by providing the necessary passwords when prompted and all other security administrators can assign users to the predefined access profiles.

Before you begin creating your user profiles, roles, and permission lists, you need to set up your access profiles in the database. Ultimately, the access profile is the profile that your users use to connect to your database. Without being associated with an access profile, users cannot sign in, even with a test ID.

The ID that you use must be defined at the RDBMS level as a valid RDBMS ID. You can create an RDBMS ID by using the utilities and procedures defined by your RDBMS platform. After you create the RDBMS ID, use the People Tools access profiles utility to link your RDBMS ID to the access profile. This profile is created when you first install your database.

Self-Check -4	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Short answer

1. _____is the profile that the users use to connect to the database.(2 pts)
2. List down the three uses of access IDs (3pts)

vii. _____

viii. _____

ix. _____

Note: **Satisfactory rating - 5 points** **Unsatisfactory - below 5 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

L #29	LO #6- confirm database design

Instruction sheet

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:

- Identifying database back-up and recovery requirements
- developing and documenting Database back-up and recovery procedures
- submitting database and documentation to client

This guide will also assist you to attain the learning outcomes stated in the cover page.

Specifically, upon completion of this learning guide, you will be able to:

- Identify database back-up and recovery requirements
- develop and documenting Database back-up and recovery procedures
- submit database and documentation to client

Learning Instructions:

Read the specific objectives of this Learning Guide.

1. Follow the instructions described below.
2. Read the information written in the “Information Sheets”. Try to understand what are being discussed. Ask your trainer for assistance if you have hard time understanding them.
3. Accomplish the “Self-checks” which are placed following all information sheets.
4. Ask from your trainer the key to correction (key answers) or you can request your trainer to correct your work. (You are to get the key answer only after you finished answering the Self-checks).
5. If you earned a satisfactory evaluation proceed to the next learning guide

Information sheet 1	Identifying database back-up and recovery requirements

Introduction

Backup and recovery describes the process of creating and storing copies of data that can be used to protect organizations against data loss. This is sometimes referred to as operational recovery. Backup and recovery refers to the various strategies and procedures involved in protecting your database against data loss and reconstructing the database after any kind of data loss.

- **Database backup** is the process of backing up the operational state, architecture and stored data of database software. It enables the creation of a duplicate instance or copy of a database in case the primary database crashes, is corrupted or is lost. Backup copies are made on a consistent, regular basis to minimize the amount data lost between backups. The more time passes between backup copies, the more potential for data loss when recovering from a backup. Retaining multiple copies of data provides the insurance and flexibility to restore to a point in time not affected by data corruption or malicious attacks.

The purpose of the backup is to create a copy of data that can be recovered in the event of a primary data failure. The reasons for data loss can be divided into five main groups:

- ✓ **Program errors**: During execution of a program, conditions may arise that abnormally terminate the program. Such program errors concern only the database application and usually have no impact on the entire database system. As these errors are based on faulty program logic, the database system cannot recover in such situations.
- ✓ **Administrator (human) errors:** Another source of data loss is human error. Users with sufficient permissions, or the database administrator, may accidentally lose or corrupt data (people have been known to drop the wrong table, update or delete data incorrectly, and so on). So due to people mistakes data can be affected. The best we can do is tried to avoid it, and be prepared to recover when it happens.

- ✓ **Computer failures (system crash):** A computer failure specifies different hardware or software errors. A hardware crash is an example of a system failure. In this case, the contents of the computer's main memory may be lost.
- ✓ **Disk failures:** A disk failure occurs either when a read/write head of the disk crashes or when the I/O system discovers corrupted disk blocks during I/O operations.
- ✓ **Catastrophes (fire, earthquake) or theft:** In the case of catastrophes or theft, the system must keep enough information to recover from the failure. This is normally done by means of media that offer the needed recovery information on a piece of hardware that has not been damaged by the failure

As a result it is essential that there is a plan for the database backup and recovery.

- **Recovery** is the process of restoring the database to a correct (consistent) state in the event of a failure. In other words, it is the process of restoring the database to the most recent consistent state that existed shortly before the time of system failure. The failure may be the result of a system crash due to hardware or software errors, a media failure such as head crash, or a software error in the application such as a logical error in the program that is accessing the database.

Recovery restores a database form a given state, usually inconsistent, to a previously consistent state. It ensures that the database is reliable and remains in consistent state in case of a failure.

1.1. Identifying database back-up and recovery requirements

Backup and Recovery processes commensurate with legislative and business requirements must be developed, maintained and regularly tested, to ensure continued business operation and access to data and information within the required timeframe, should a risk event occur.

Backup requirements will be determined by a business risk assessment completed by the owner, and is dependent on the:

- Importance of the data and information to the function.
- Acceptable transaction loss (business areas must determine what level of potential transaction loss would not be acceptable or would be too difficult to

recover. This can be determined in terms of a timeframe, the number of transactions, or the amount of effort and period of time required re-entering data.

- The maximum acceptable outage of the system while performing backups
- The maximum acceptable outage of system while recovering data

In addition to regular backup processes, backups will be performed before and after major technical or business related changes to a system or application.

An audit trail of all backup activities must be maintained.

1.1.1. Backup media and Off-site storage

Backups must be regularly tested as determined by a risk assessment or at a minimum on an annual basis to ensure data can be restored in case of a catastrophic event.

Protection mechanisms and access controls for backup media must be commensurate with the security requirements and criticality of the information stored in the backup.

Backup media must be stored and transported in an appropriate, safe and secure manner and access to backup media must be restricted to only authorized personnel.

Based on backup requirements and backup cycles, at least one instance of a backup within a cycle must be stored off-site (physically separate from the data or system being backed up) or geographically separate, as determined by a risk assessment.

Backup media stored off site must be stored in a secure location with environmental controls (if available) and appropriate access controls commensurate with the security requirements and criticality of the information stored in the backup.

- licenses

Self-Check -1	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Choose the best answer(2pts each)

1. _____ is process of creating and storing copies of data t to protect data loss.
   A. Security B. Recovery C. Backup D. All E. None
2. Which one of the following is not the cause for data failures?
   A. Program errors B. Human errors C. System crash D. All E. None
3. The accidental deletion of data or removal of table can be considered as;
   A. Program errors B. Human errors C. System crash D. All E. None

Note: **Satisfactory rating - 5 points** **Unsatisfactory - below 5 points**
You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____
Rating: _____

Information sheet 2	Developing and documenting Database back-up and recovery procedures

2.1. Backup and Recovery Procedures

Backup and recovery methods are essential to data protection and security. Any loss of data due to file corruption, virus, security or human error is a loss of time and money. Furthermore, loss of data can severely impact the success of a project. An effective server backup and recovery plan is crucial to the organization.

The purpose of this plan is to provide a successful procedure for backup and recovery of critical data. The backup and recovery plan includes, backup and recovery of file and print servers, mail servers, database servers, web servers, video streaming servers and domain controllers.

Example of Backup plan

- **Backup Plan**
  - ✓ Server backups will be performed every business night excluding holidays.
  - ✓ Backups performed on Friday will be kept for a month before recycling.
  - ✓ The last backup of every month will be considered the monthly backup and kept for a year before recycling.
  - ✓ Monthly backup tapes will be stored in a fireproof safe.
  - ✓ The last two monthly tapes will be stored off-site in a fireproof safe.
  - ✓ Backups will be performed and monitored by a fulltime IT staff member.
  - ✓ Backups will be automated
  - ✓ Tapes will be inserted routinely every night before leaving work.
  - ✓ Backup failures will be reported to the Director of Information Technology and action will be taken quickly to fix the problem.
  - ✓ Backups will always be performed before upgrading or modifying a server.
- **Loss of data**
  - ✓ If loss of data is discovered, evaluation and investigation by IT staff is immediately dispatched.
  - ✓ In most cases, loss of data is related to file corruption, virus, security or human error.

- ✓ If loss of data is related to data corruption, IT Staff must troubleshoot and determine if the problem is hardware or software related to prevent addition file corruption.
- ✓ If the loss of data is related to a virus, IT Staff must determine the extent of the virus and remove it to prevent further loss of data.
- ✓ If the loss of data is related to security or a compromised system, IT Staff must determine the extent of the compromise and fix the vulnerability quickly to prevent further loss of data.
- ✓ If the loss of data is related to human error, IT Staff must immediately inform and train the appropriate personnel to avoid further loss of data.
- ✓ Once the problem has been determined and loss of data minimized, IT Staff should proceed to restoration of data from backup media.

- **Restoration of data**
  - ✓ Once loss of data is discovered, evaluated and minimized, IT Staff will proceed to restoration of data from backup media.
  - ✓ IT Staff will determine the time and date of the lost data.
  - ✓ IT Staff will determine the appropriate backup media to restore the data.
  - ✓ IT Staff will insert the backup media into the appropriate server.
  - ✓ IT Staff will invoke the Backup/Restore software, such as Veritas.
  - ✓ IT Staff schedule the restore of the appropriate data within the Backup/Restore software.
  - ✓ IT Staff monitor the restore of data.
  - ✓ Upon restore, IT Staff evaluate the integrity of the restored data.
  - ✓ IT Staff will contact the end-user of the data to finalize restore.
  - ✓ Upon approval from the end-user, the restore is considered finished.
- **Disaster Recovery**
  - ✓ If a disaster is discovered, IT Staff will determine the extent of the problem and proceed accordingly.
  - ✓ If the disaster is hardware related, IT Staff will replace the failed hardware and restore according to the steps outlined above.

- ✓ If there is a natural disaster, such as water, fire, tornado, earthquake, or other, the hardware will be replaced and the server will be restored using the offsite backup media according to the steps outlined above.
- ✓ Upon restoration of data, IT Staff will check the data for integrity and validity.
- ✓ IT Staff will contact the end-user of the data to finalize restore.
- ✓ Upon approval from the end-user, the restore is considered finished.

1.2. Documentation

Documented procedures must exist for the backup and recovery processes and these documents must be readily accessible. Backup and recovery operations and the specified period of maximum acceptable outage must be documented for all systems.

Documentation help ensure consent and expectations. It helps to tell the narrative for decisions made, and how the client responded to different situations. In this same manor, it is important to record information that can help support the proper treatment plan and the reasoning for such services.

At a minimum documentation must contain:

- A description of the system to be backed up
- The individual or group responsible for ensuring that the backup and recovery occurs
- Backup and recovery requirements
- Backup media storage locations, including off-site storage
- Required backup frequency e.g. daily, weekly
- Backup cycles required
- Backup retention period (as prescribed by the University Data Retention Policy)
- Testing process
- Recovery schedule and plan

Self-Check -2	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Choose the best answer

1. In computer, step for maintaining the security of important data is known as____(2pts)

   A. Data backup
   B. Data protection
   C. Data locking
   D. Data securing

Short answer

2. What is the purpose of backup and recovery plan?(2pts)

   _____

   _____

   _____

3. List down at least two importance of documentation.(4 pts)

   i._____

   _

   ii.

   _____

Note: **Satisfactory rating - 4points** **Unsatisfactory - below 4 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

Information sheet 3	Submitting database and documentation to client

3.1. Submitting database and documentation for approval

An approval process is the method an organization uses to approve the validity or correctness of something. It is a type of workflow, which is any sequence of work from initiation to completion that you can create to ensure work is approved the same way every time. Creating an approval process can simply mean defining the procedures that you will follow to approve work.

So you must have to submit the design documentation to client for approval. The Client has the option to either approve or reject the content.

- **Approved:** The content is formally approved and sign-off by the client.
- **Rejected**: The designer must make more edits. This cycle of edits can continue until the client is satisfied and the content is officially approved.

There are a number of questions that can guide your structural design:

- **How should requests be submitted?** Decide how submitters should submit their work, including when, how/where (for example, an online form or portal), to whom, and what to include in the initial proposal.
- **What documentation should you include?** Decide if any supporting documentation must be part of the original submission, or if editors and approvers need to send documentation along with their approval or rejection. If so, make sure the software platform can support document storage.
- **How many steps are there, and what are they?** The decisions made at this point are the meat of your design process. Here you create the pathways from initial submission to final approval or rejection. For each stage, define the criteria that must be met for a submission to move to the next stage.
- **Who approves each step?** Identify approvers for each stage of the approval process.

- **Who should edit submissions, and at which steps?** If your process requires edits at certain stages, who will edit the submission (for example, it could be the original submitter, a designated editor, or the approver)? Deciding who is responsible for editing may change your sequencing.
- **Under what criteria should requests be automatically approved or rejected?** There might be instances where an automatic approval or rejection is permissible. Be sure to define these parameters in your logic so you can apply it to the program when you build your process.
- **What actions happen when a request is approved or rejected?** Your approval process should not exist in a silo, so consider how it is connected to the execution processes, once a submission is approved or rejected. Consider linking notifications to kick off a project or pull the next submission.

Self-Check -3	Written Test

Directions: Answer all the questions listed below. Use the Answer sheet provided by your instructor to check whether your answer is correct or not

Choose the best answer(2pts each)

1. What is Document approval? (2pts)

   _____

   _____

   ___

2. Why it is needed? (2pts)

   _____

   _____

   _____

3. What are the two expected results during Approval? (2pts)

   _____

   _____

   _

Note: **Satisfactory rating - 5 points** **Unsatisfactory - below 5 points**

You can ask you teacher for the copy of the correct answers.

Answer Sheet

Score = _____

Rating: _____

References

1. Sharma Neeraj, Perniu Liviu et al. Database Fundamentals
2. Raul F. Chong, Abhishek Iyer, Chaitali Nandan, Adi-Cristina Mitea, Mallarswami Nonvinkere, Mirela Danubianu 1st edition
3. Architecture of a Database System
4. Database Design - 2nd Edition by Adrienne Watt and Nelson
5. Joseph M. Hellerstein1, Michael Stonebraker2 and James Hamilton3
6. DBTechNet Draft V18 2013-11-19 Martti Laiho, Matti Kurki
7. A Silberschatz, H Korth, S Sudarshan, “Database System and Concepts”, fifth Edition McGraw-Hill
8. SYLLABUS DATABASE MANAGEMENT SYSTEMS
9. https://public.dhe.ibm.com/software/dw/db2/express-c/wiki/Database_fundamentals.pdf
10. https://www.javatpoint.com/dbms-tutorial

Acknowledgement

We wish to extend thanks and appreciation to the MoSHE, FTA, Oromia TVET bureau, and different college instructors, academic and government agencies that took vital role and donated their expertise and resource for the developments of this TTLM.

We would like also to express our appreciation to the Experts from different organizations with in the line MoSHE and FTA that made the developments of this TTLM possible in terms of financing and supporting technically.

The development of this TTLM held on December 2020 at Bishoftu, Ethiopia.

The trainers and Experts who are participated technically on the development of this TTLM

No	Name	Organization	Educational background & Level	Profession /Job title	Email address	Mobile
1.	Ayansa Ergiba	Ambo TVETC	Msc. ICT mgt	Instructor	aergiba@gmail.com	0917851343
2.	Ejigu Birhanu	Nekemte TVETC	Msc. ICT mgt	Instructor	ejigubiranu2011@gmail.com	0906566892
3.	Endale Lema	Adama PTC	Msc. ICT mgt	Instructor	endhiywet@gmai.com	0913292212
4.	Keresa Gadisa	Nekemte TVETC	Msc. ICT mgt	Instructor	keresag2010@gmail.com	0920420664
5.	Meseret Tezera	Atilet Kenanisa PTC	Msc. ICT mgt	Instructor	YafetMes@gmail.com	0911751285

This TTLM was developed under TVET program for Database Administration level III on the **December 2020** at **Bishoftu**, Ethiopia

Answer sheet

Answer Key Module Title: Designing Database

LO#1: Determine database requirements	
Self-Check1	**Written Test**

Directions: Answer all the questions listed below. Use the Answer sheet provided in the next page:

I. Choose the best answer (each 2 point)

1. E
2. D
3. A
4. User requirement
5.
   i. Facilitate better planning and less back-tracking
   ii. Cost and schedule can be better maintained
   iii. Smoother final project integration
   iv. Smooth the progress of testing and Validation, etc.

Self-check 2

1. **A**
2. **A**
3. **D**
4. **A**

Self-check 3

1. **D**
2. **A**
3. **B**

Self-check 4

1. **A**
2. **B**
3. **C**
4. **B**
5. **C**

Self-check 5

1. True
2. True
3. False
4.
   i. Projects which provide duplicate or inadequate functionality.

ii. over-budget projects
iii. projects which are cancelled due to lack of funds, staffing, planning, user participation, and/or management interest

Self-check 6

1. False
2. True
3. False

LO #2- Develop Logical Data Model

Self-check 1

1. D
2. A
3. D

Self-check 2

1. C
2. B
3. C
4. B
5. B

Self-check 3

1. C
2. D
3. C

Self-check 4

1. B
2. B
3. B
4. C

Self-check 5

1.

i. Managing the expectations between the client and the project team.
ii. Leaving room for additions or modifications while maintaining clear communication.
iii. Maintaining maximum responsibility for each phase of the project.

2. Documentation

3. Approval

LAP Test 1;

i. **Entity**: Student, Course, Instructor

ii. **Attributes:**

Student(Unique-ID, Name(Composite), DOB, Gender ,Mobile)

Course (Course-Code, unique Course-name, Credit hour)

Instructor (Unique- ID, Name (Composite), Gender, Age, Date of Hire, Year of service (derived), Qualification, Phone)

2. Relationship

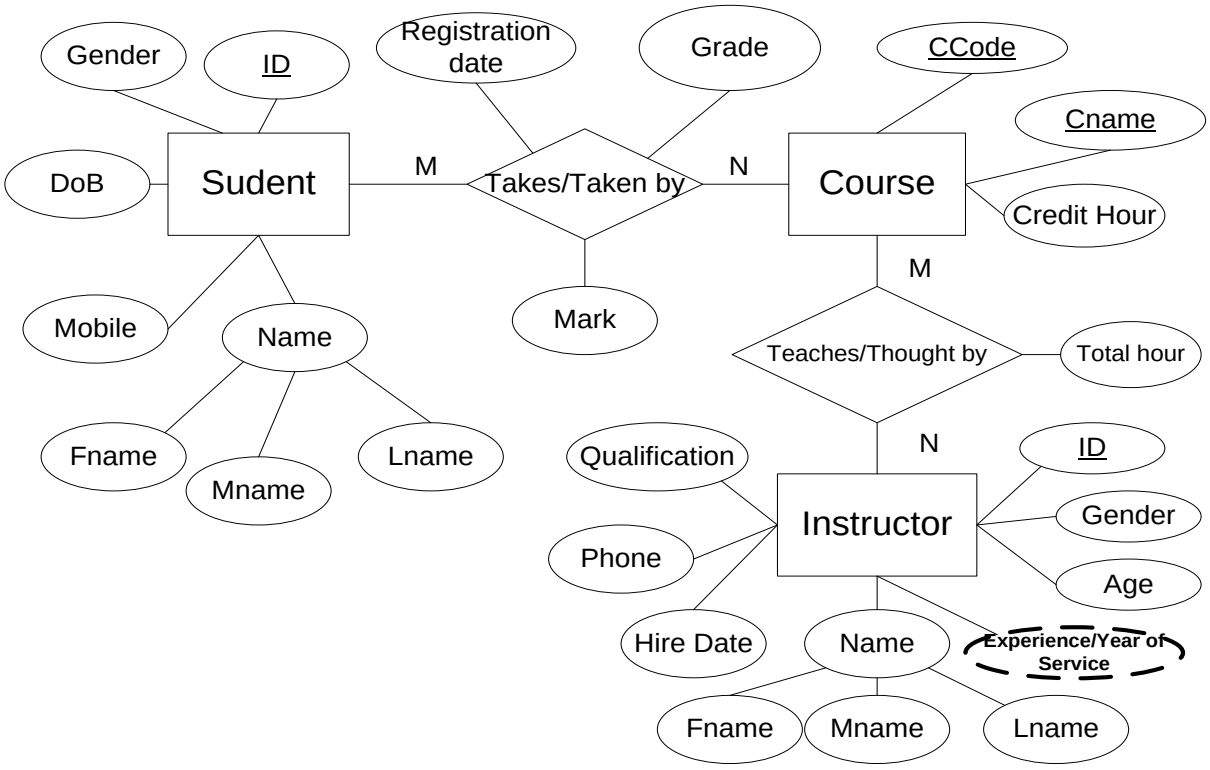
Course: Instructor (M: N)

Relationship attributes (Total Hour)

Course : Student (M:N)

Relationship attributes (Registration date, Mark, Grade)

3. ERD



LO #3- Develop Logical Data Model

Self-check 1

1. **C**
2. **B**
3. **C**

Self-check 2

1. B
2. **A**
3.

i. Primary keys
ii. Foreign keys
iii. Unique constraints
iv. Check constraints
v. Data types
vi. Data precision and scale
vii. NULL/NOT NULL

Self-check 3

1. B
2. **A**
3. **A**

Self-check 4

1. **A**
2. **A**
3. **A**

Self-check 5

1. **A**
2. **Validation**
3. **D**

Self-check 6

1. **B**
2. **D**
3. **B**
4. **A**

Self-check 7

1. **Documentation**
2.

i. Provides a common language between business decision makers and IT personnel.
ii. Provides a shortcut to finding 'hot-spots'.
iii. Minimize the chances of making a wrong decision.

iv. Makes maintenance easier.
v. Reduces risk when extending or upgrading a system.
vi. Reduces training costs, etc.

LAP Test1:

Task1:

Employee: Table

EID(PK)	EName	Gender	Salary	Dept_code(FK)

Project: Table

Project_code(PK)	Project_Title	Project_duratin	Dept_code(FK)

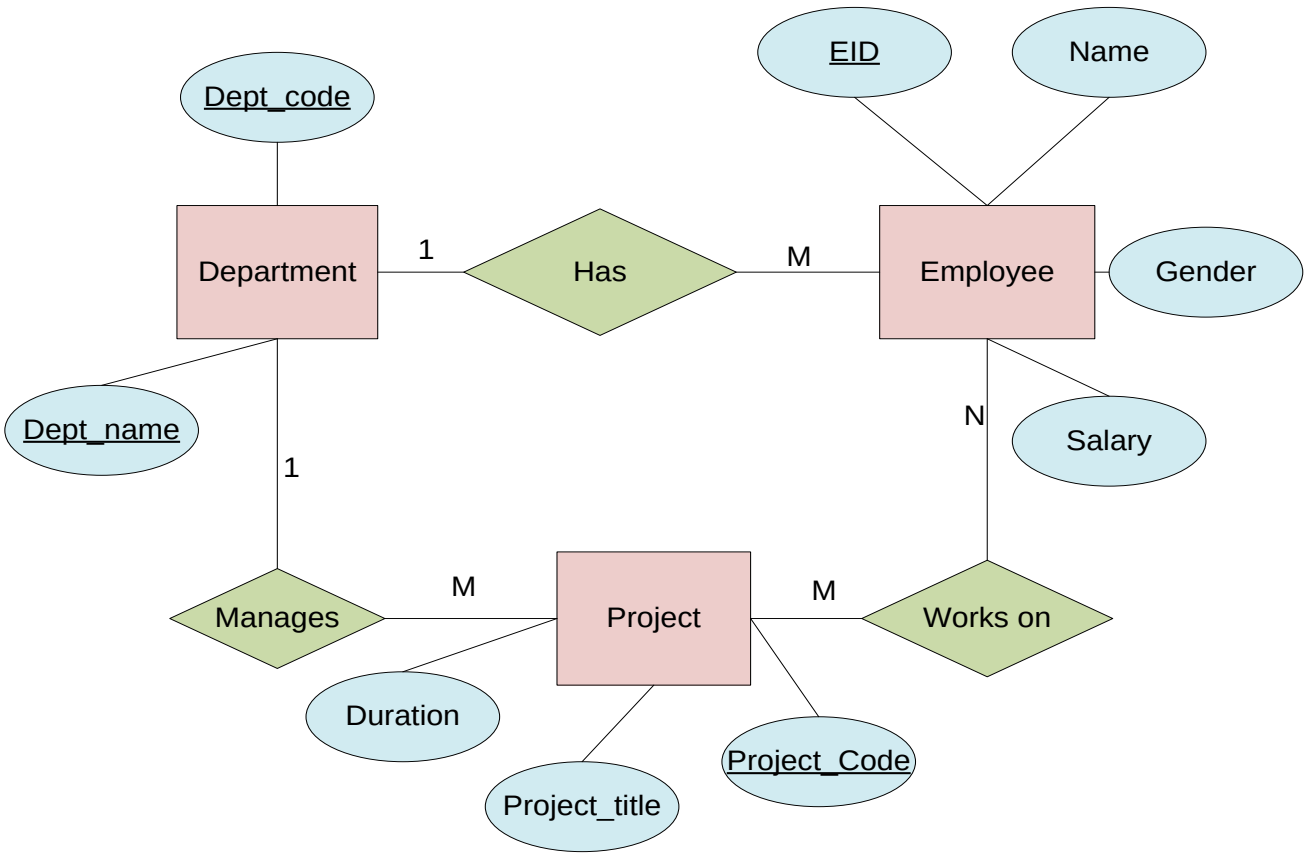
EMP_Project: Join table

EID(FK)	Project_code(FK)

Project: Table

Dept_code(PK)	Dept_name

Task2:


EID
Name
Dept_code
Department
1
Has
M
Employee
Gender
Dept_name
1
N
Salary
Manages
M
Project
M
Works on
Duration
Project_Code
Project_title

Lo4: Design queries, screens and reports

Self-check 1

1. GUI
2.

i. Lay Out
ii. Content Awareness
iii. Aesthetics
iv. User Experience
v. Minimize User Effort

Self-check 2

1. Query
2. Select query
3. Action query
4.

Self-check 3

1. **Report**
2.

i. Know what information you need.
ii. Determine the data that will be listed on the report.
iii. Sketch a sample report
iv. Ensure the appropriate data is being collected.
v. Create your report.

Self-check 4

1.

i. Entities will have definitions.
ii. Designed and developed to be independent of DBMS, data storage locations or technologies.
iii. Enterprise-wide coverage of the business concepts.
iv. Designed and developed primarily for a business audience

2.

i. Contains relationships between entities that address cardinality and nullability (optionality) of the relationships.
ii. Designed and developed to be independent of DBMS, data storage locations or technologies.
iii. Data attributes will typically have data types with precisions and lengths assigned.
iv. Data attributes will have nullability (optionality) assigned.

3.

i. Designed and developed to be dependent on a specific version of a DBMS
ii. Columns will have nullability (optionality) assigned.
iii. Columns will have datatypes with precisions and lengths assigned.
iv. Tables and columns will have definitions.

Self-check 5

1. An alteration or modification of existing system

2.

i. Changing the name of a database, table, view, or column
ii. Modifying a stored procedure, trigger, or user-defined function
iii. Changing or adding relationships using referential integrity features
iv. Changing or adding database partitioning.
v. Moving a table from one database, dbspace, or tablespace to another.
vi. Rearranging the order of column in a table.
vii. Changing a column's data type or length.
viii. Adding or removing columns from a table.
ix. Changing the primary key without dropping and adding the primary key.
x. Adding or removing columns from a view, etc

Lo5: Design access and security systems

Self-check 1

1. **D**
2. **C**
3. **A**
4. **C**

Self-check 2

1. D
2. **C**
3. **D**

Self-check 3

1. **A**
2. **C**
3. **A**

Self-check 4

- **Access profile**
-

i. When an application server initializes and connects to a database.

ii. When a developer or power user signs in to the database directly (two-tier).

iii. When batch programs connect to the database.

Lo6: Confirm database design

Self-check 1

1. **C**
2. **E**
3. **B**

Self-check 2

1. A
2. To provide a successful procedure for backup and recovery of critical data.
3.
   i. Documentation help ensure consent and expectations
   ii. Facilitate decision making
   iii. It is also used as communication tool

Self-check 3

1. **D**
2. **C**
3. **B**