Debere Birhan Polytechnic College

DATABASE ADMINISTRATION SERVICES Level III

MODULE TITLE: **Identifying Physical Database Requirements**

LO1. Identify database scope

- **Scope** is the range of one's perceptions, thoughts, or actions
- The purpose of defining the scope is to define **the boundary of the system and the project**.
- The scope shows what aspects of the system will and will not be included in the project.

A good database is one that is simple to understand and well planned.

One can use ERDs (Entity-Relationship Diagrams) or EERDs (Enhanced-Entity Relationship Diagrams) in order to make a good database.

The Scope of Good Database Design:

- Easy to locate the data.
- No redundant data (No repeated).
- More security.
- Table references (keys like: Primary and foreign keys) are easy to maintain.

1.1 Reviewing system architecture and user requirements

1.1.1 Operating system

- **Operating System** is a set of programs that manages computer hardware and various resources installed on it.
- Analyze various operating systems, noting their strengths and limitations with regards to I/O, processing, and memory
- Define the establishment of the network within the operating system and the management of I/O
- Explain the operating system support of applications
- Explain the configuration parameters of an operating system that affect a database.
- Plan and manage physical resource requirements
- Analyze limitations of specific operating systems

You have to check the operating system loaded on your machine whether it is compatible with the DBMS software you want to install or not.

1.1.2 DBMS software

A *database management system* is the software that:

- Enables users to define, create, and maintain the database
- Provides controlled access to the database.
- Interacts with the users, application programs, and the database itself.
- Allows users to insert values, retrieve it when needed, update/modify it when necessary, and delete it if not required from the database.

➢ Each database application has its own specific strength and weakness.

- Some of the common database applications include:
  - Microsoft Access
  - Microsoft SQL
  - Oracle and Informix.
  - MySQL

Database Support Software

- *List and reference the documentation of any DBMS utility software available to support the use or maintenance of the database.*
- *Describe all support software, including the operating system, directly related to the database, including name, version, function, and major operating characteristics.*

1.1.3 Configuration : small and large memory model

A memory of large size and processor with high speed of data processing might be required to run the DBMS software.

The Database Configuration utility creates the connection settings that will be used by either the management server or action server installed on the computer. Only Local Administrators or the user who ran the Database Configuration utility on the computer can access it. If this file is missing or the permissions have been changed, the management server or the action server may report errors that a connection to the data store cannot be made. If you have changed your database settings, such as the port, user account access, or computer where it is installed, you will need to re-run the Database Configuration utility on the management server and all action servers.

1.2 Determining Database size and technical specifications

Database Size means the amount of data to be stored in the designed system**.**

- You can specify sizes for data and log file in the definition of a database.
- **File Growth:** You can specify whether a file will grow in size in the future if necessary.

Example: - Specifying memory size during database Creation

When you design a database, you may have to estimate how large the database will be when filled with data. Estimating the size of the database can help you determine the hardware configuration you will require to do the following:

- Achieve the performance required by your applications.
- Guarantee the appropriate physical amount of disk space required to store the data and indexes.

Estimating the size of a database can also help you determine whether the database design needs refining.

1.3 Documenting Database and scope of project

Documenting a database means collecting and analyzing information about the organization to be supported by the database system, and use this information to identify the requirements for the new database system.

- Before attempting to design a database system, you must first identify the scope and boundary of the project.
- When defining the system boundary for a database system, we include not only the current user views but also any known future user views.

1.4 Evaluating Several database management systems and making appropriate selection

Evaluating DBMS software means comparing DBMSs software one from the other to select the appropriate one.

The selection must be based on its feature to create a database, disk space it needs, its strength, its flexibility and cost.

LO2. Identify database requirements

2.1. Reviewing Technical specifications

Technical specifications are typically written by computer department staff that answers specification questions such as: - What are the hardware and software requirements to run this system?

(Be sure to specify a minimum of disk space to store the data)

- "What is the compatible operating system for the selected DBMS?"
- "How many tables are included in the database and how are they related?"
- describe the file structure for each table.

2.2. Identifying database tables and relationships

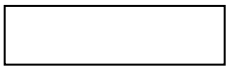
What is **entity** in a database?

- **An Entity** is a thing in real-world (an object with a physical existence or an object with a conceptual existence).
  - person/employee, car, book, house, employee, etc are example of objects with physical existence
  - A project, course, payment, department, etc are examples of objects with object with a conceptual/ intangible existence.
  - **Entity** can be roles, events, tangible things or concepts.

* Each entity must have its own identity that distinguishes it from every other entity, called **unique identifier**.

Example: Each Student has a unique ID that distinguishes one from others.

Generally in database terms, an **entity** is a ***table*** *which* is responsible for storing records in the database.

In E-R Diagram, an entity is represented by a rectangle. Example: 

Other terms related to Entity include:

- Entity type
- Entity set
- Entity instance

Entity type is a collection of entities that share common properties or characteristics.

Example**: -** Student is an entity type to all students.

In E-R Diagram, an entity type is represented by a rectangle, and the name is indicated in capital letters.

Example: 

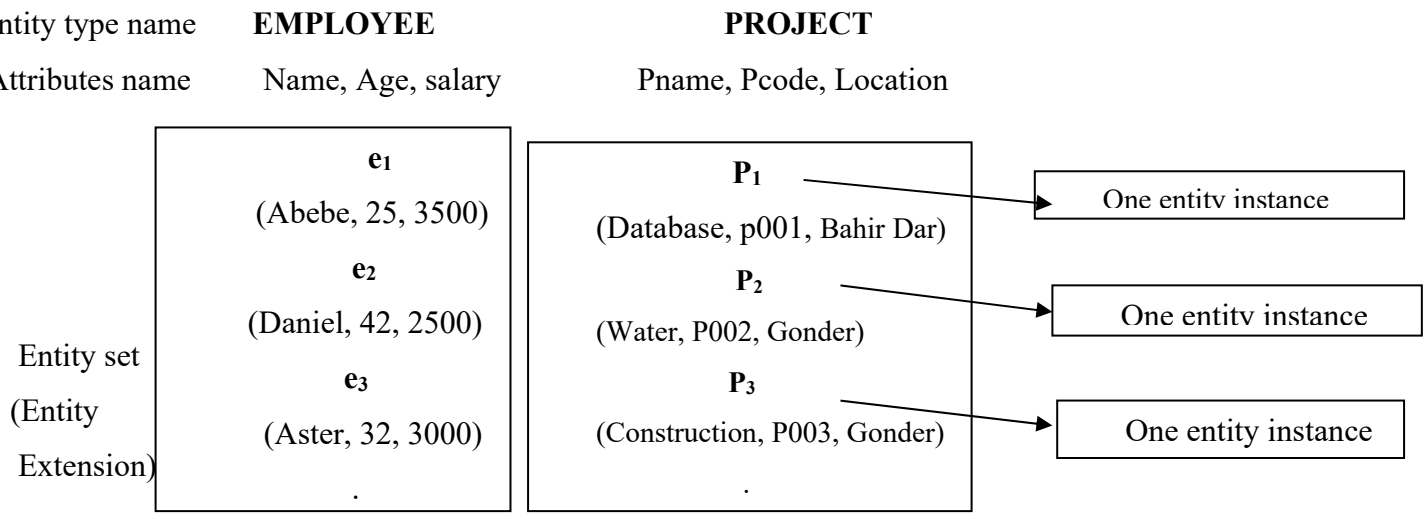
E**ntity set** is a set of entities (collection of Entities/instances) of the same type.

Example: - a company have many employees, and these employees are defined as entities (e1,e2,e3....) and all these entities having same attributes are defined under ENTITY TYPE employee.

I.e. E**ntity set** is also called Entity E**xtension**.

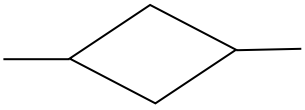
- **Entity Instance** is a single occurrence of a particular entity type. An entity type is described just one time in the data model, but many instances of that entity type may be represented by data stored in the database.

  Example: - There may be hundreds or thousands of employees in an organization – each one is an instance of the Employee entity type.



Relationships

A **relationship** is an association that exist b/n two or more participating entities.



- The function of relationship is to share data between entities.
- In E-R Diagram, a relationship is represented by a diamond shape.
- Some example of relationships are: - EMPLOYEES **works_on** PROJECTS
  - EMPLOYEES **process** PAYMENTS
  - EMPLOYEES **works_for** DEPARTEMENT

Types of relationship

There are three types of relationships between entities

* **One-to-one**: - one instance of an entity (A) is associated with one instance of another entity (B).

  Example: - the **manages** relationship between department and employee

  - President to country and husband to wife are examples of one-to-one relationship.

* **One-to-many**: - one instance of an entity (A) is associated with zero, one or many instances of another entity (B), but for one instance of entity (B) there is only one instance of entity (A).

  Example: - the **has** relationship between department and employee

  - Country to people, mother to children, department to students, and people to religion are of one-to-many relationships.

* **Many-to-many**: - one instance of an entity (A) is associated with zero, one or many instances of another entity (B), and one instance of entity (B) is associated with zero, one or many instances of entity (A).

  Example: - the **works_on** relationship between employee and project

  - Students to course, employee to projects, and people to language are of many-to- many relationship.

2.3. Identifying data dictionary, table attributes and keys

What is data dictionary?

A **data dictionary** is a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them.

* In DBMS, a data dictionary is a **file** that defines the basic organization of a database.
* Most DBMS keep the data dictionary hidden from users to prevent them from accidentally destroying its contents.
* A data dictionary may contains:
  - The definitions of all schema objects in the database (tables, views, indexes, procedures, functions, triggers, and so on). Example: Name of the tables, field names of each table, Data type of each field, Length of each field, constraints of each table and field, etc.
  - How much space has been allocated for, and is currently used by the schema objects
  - Default values for columns (Constraints on data i.e. range of values permitted)
  - Integrity constraint information (Constraints that apply to each field, if any)
  - Auditing information, such as who has accessed or updated various schema objects
  - Privileges and roles each user has been granted (Access Authorization)
  - Description of database users, their responsibilities and their access rights.

Simple Example of data dictionary

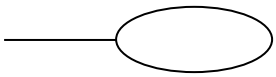
Table Name	Column Name	Data type	Field size	constraint	Description
EMPLOYEE	**EmpID**	Varchar	15	Primary key	Employee identity
	Lname	Char	15		Last name of employee
	Fname	Char	5		First name of employee
	Gender	Char	6	Male or female	
	DeptNumber	Varchar	10	Foreign key	
PROJECT	Pname	Char	17		
	Pcode	Char	10	Primary key	Project identity
	DeptNumber	varchar	10	Foreign key	
DEPARTMENT	Dname	Char	20		
	Dnumber	Varchar	10		Department identity

Data dictionaries do not contain any **actual data** from the database, only bookkeeping information for managing it.

What is an advantage of a Data Dictionary?

When a new user is introduced to the system or a new administrator takes over the system, identifying table structures and types becomes simpler.

- **Table attributes**
  - An attribute is a property that describes an entity.
  - Each attribute has a particular value based on the defined data type.
  - The set of all possible (allowable) values of an attribute is called **attribute domain**.
  - As a convention, attributes are named with an initial capital letter followed by lowercase letters.
  - In E-R Diagram, an attribute can be represented by an ellipse (oval) shape with a line connected it to the associated entity. **Example**:



  - It can also be represented by listing them within the entity rectangle, under the entity name.

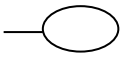
Example:

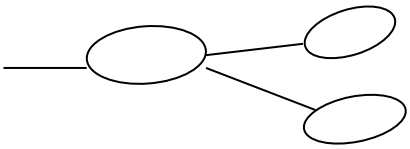
EMPLO
YEE
Emp_Id
Fname

Types of attributes

An attribute can be: - simple or composite, single-value or multi-value, stored or derived or null able.

1) Simple (Atomic) vs Composite attribute

- Simple attributes cannot be further divisible; whereas Composite attributes can be divided into smaller subparts.
    - › ID, Salary, Gender, etc are examples of simple attributes.
    - › Name, Address, etc are examples of composite attributes
- In E-R Diagram, simple attributes can be represented by an ellipse shape. Example: 
- In E-R Diagram, composite attributes can be represented by branched ellipse shape:

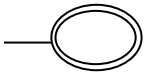
  Example: 

- The value of composite attribute is the concatenation of the values of its constituent simple attributes.

2) Single-valued vs Multi-valued attributes

- Most attributes have a single value for a particular entity.
- In some cases, an attribute can have a set of value for the same entity, called **multi-valued**.

Example: colour attribute for a car, college degree for a person.

- In E-R Diagram, **Multi**-valued attributes can be represented by double ellipse (oval) shape. 

3) Stored vs derived attributes

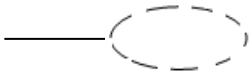
- In some cases, the value of two or more attributes can be related.

  Example: the **Age** and **Birth-Date** attributes of a person

  -- The **Age** attribute is derived from the **Birth-Date** attribute.

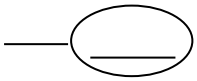
  -- **Age** attribute is **derived** attribute, and the **Birth-Date** attribute is **stored** attributed

  In E-R Diagram, **Derived** attributes can be represented by dotted ellipse.



Key attributes

- Attributes can be classified as identifiers or descriptors.
- **Identifiers** (more commonly called keys or key attributes) which uniquely identify each instance of an entity, called **candidate key**. If such an attribute doesn't exist naturally, a new attribute is defined for that purpose, for example an ID number or code.
- In some cases, more than one attribute is required to identify a unique entity, called **composite candidate key.**
- A descriptor describes a non-unique characteristic of an entity instance.
- When identifying attributes of entities, identifying key attribute is very important.
- In E-R Diagram, key attributes (identifiers) can be represented by ellipse shape with underline.



When selecting identifiers, you should follow these guidelines:

i) Choose a candidate key that has values that do not change over the life of each instance of the entity type.

ii) Choose a candidate key that is guaranteed to always have valid values and not be null for each instance.

iii) If a candidate key is a composite of two or more attributes, consider creating a new key with a single value.

✓ **<u>Types of Keys in Relational Database Model</u>**

- Super key
- Candidate key
- Primary key
- Alternate key
- Foreign key

* **Super Key** is an attribute or a composite attribute which functionally determines all of the entity's attributes.

- **Since, a super key may contain additional columns that are not necessary for unique identification; we're interested in identifying super keys that contain only the *minimum number* of columns necessary for unique identification.**

* **Candidate Key** is a super key whose values are not repeated in the table records.

- **It is a super key that contains only the minimum number of columns necessary for unique identification.**
- When the values in a super key are not repeated in the table's records, then such a key is called a candidate key.

- It is possible to have more than one choice of candidate key in a particular table. In that case, the selection of the primary key would be driven by the designer's choice or by end user requirements.

* **Primary Key** is a candidate key which doesn't have repeated values nor does it comes with a NULL value in the table.
  - A primary key in any table is both a super key as well as a candidate key.
  - **The candidate keys that are not selected to be the primary key are called alternate keys.**
  - Every entity in the data model must have a primary key whose values uniquely identify instances of the entity.
* **Alternate key** is th**e candidate keys that are not selected as the primary key of the entity.**
* **A foreign key is a column or group of columns in a relational database table that is used to establish and enforce a link between data in two tables. You can create a foreign key by defining a FOREIGN KEY constraint when you create or modify a table.**

2.4. **Developing database reports using acceptance criteria**

What does **Database Reports** mean?

Database reports are the formatted result of database queries and contain useful data for decision-making and analysis. Most good business applications contain a built-in reporting tool; this is simply a front-end interface that calls or runs back-end database queries that are formatted for easy application usage.

For example, a banking software application may contain specifically defined reports on all customers with large deposits or reports on monthly loan summaries for all customers.

- **Database reports** are developed based on acceptance criteria and requirements.

The reports are normally used to provide hard copy printouts for reviews at meetings or to mark up the information in your database that needs updating.

LO3. Identify security requirements

3.1 Reviewing system security

Reviewing system security includes:

- Application System security
- Computer System security
- Network system security
- DBMS security
- Financial and business system security

System security means Control of access to a computer system's resources, specially its data and operating system files.

- It covers access and use of the database at the system level, such as a username and password.

- **Application System security**

Application system security is the use of software, hardware, and procedural (technical) methods to protect applications from external threats.

- The most basic software to ensure application security is application firewall that limits the execution of files or the handling of data by specific installed programs.

- **Computer System security**

The term computer system security refers to the collective processes and mechanisms by which sensitive and valuable information and services are protected from publication (unauthorized activities), tampering (corrupting) or collapse.

- **Network system security**

Network security is a specialized field in computer networking that involves securing a computer network infrastructure.

Network security is typically handled by a network administrator or system administrator who implements the security policy, network software and hardware needed to protect a network and the resources accessed through the network from unauthorized access and also ensure that employees have adequate access to the network and resources to work.

- **DBMS**

Database Security means protection of the: - data from malicious attempts to steal (view) or modify data.
- Database from unauthorized users.

3.2 Clarifying and confirming DBMS and user security

Security within the DBMS protects the integrity of the data, records and databases. It can provide encryption protection at the data level.

Database Security means protection of the: - data from malicious attempts to steal (view) or modify data.

- Database from unauthorized users.

- **Data security covers access and use of database objects (such as tables and views) and the actions that users can have on the objects.**
- User security lets your application use security rules to determine what it shows.

3.3 Identifying, evaluating and recording DB performance recovery and audit trails.

What is **auditing**

Auditing is the monitoring and recording of selected user database actions.

Database performance can be defined as the optimization of resource use to increase throughput and minimize contention, enabling the largest possible workload to be processed.

Database Recovery is responsible for preserving the database consistency after a failure of any kind (transaction, system or media). Relevant information solely for recovery is saved in a log during normal transaction processing.

Recovery performance focuses primarily on crash recovery rather than on recovery after restoring a backup.

However, optimizations are possible for recovery after restoring from a backup.

An **audit trail** (or **audit log**) is a record showing who has accessed a computer system and what operations he or she has performed during a given period of time. Audit trails are useful both for maintaining security and for recovering lost transactions.

Most accounting systems and database management systems include an audit trail component. In addition, there are separate audit trail software products that enable network administrators to monitor use of network resources.

Database auditing involves observing a database so as to be aware of the actions of database users. Database administrators and consultants often set up auditing for security purposes.

LO4. Seek Client feedback and approval

- Presenting DB scope and technical requirements to user for feedback

Present ***database*** scope, technical requirements and security documentation to ***user*** for feedback is very important to develop a successful database.

- Review ***user*** feedback and adjust ***database*** as required
- Present ***database*** and documentation to ***user*** for final approval

What is feedback?

Feedback is a Process in which the effect or output of an action is 'returned' to modify the next action.

- Reviewing user feedback to adjust DB.
- Presenting DB and documentation to user for final approval