Aemilianum College Inc.
COLLEGE OF ENGINEERING, COMPUTER STUDIES,
TECHNOLOGY, and LIBERAL ARTS
Rizal St., Piot West District, Sorsogon City
Sorsogon, Philippines 4700
Email Address: *aemilianumcollege@gmail.com*
www.aemilianum.edu.ph

SUBJECT : INFORMATION TECHNOLOGY UPDATES

INSTRUCTOR : Ms. Irene Joy Sarmiento, LPT. MIT.

NAME : Pauline Anne E. Patosa

A. Database System

Define:

a. Data Models (relational, ER)

- Data Model
  - A data model is like a blueprint that shows how information is organized and connected in a system. It helps people and computers understand what kind of data is being stored like patient names, appointments, or lab results and how those pieces relate to each other.
- Entity-relationship model
  - It shows entities (things like patients, doctors, or visits) and the relationships between them using diagrams. It's mainly used during the planning phase to design how data will be structured before it's turned into a relational database.
- Relational model
  - It stores data in tables (also called relations) made up of rows and columns. Each table represents an entity (like patients or appointments), and tables are linked using keys. It's the foundation of most modern databases like MySQL or PostgreSQL.

b. Indexing (purpose, B-tree vs. hash)

- Indexing
    - Indexing in a database is a technique used to improve the speed and efficiency of data retrieval. It works much like an index in a book, allowing the system to quickly locate specific pieces of information without scanning every row in a table. The main purpose of indexing is to enhance query performance, especially for searches, filters, and sorting operations.
- B-tree vs. Hash
    - This are the two common methods used in databases to improved data retrieval. **B-tree indexing** stores data in a sorted, balanced tree structure, making it ideal for range queries, such as finding all patients between certain ages or retrieving records in a specific order. **Hash indexing** uses a hash function to quickly locate data based on exact values, like finding a patient by their exact ID. While **B-tree** is more flexible and widely used, especially in systems that need sorting and range filtering, **hash** is better suited for fast, exact-match lookups. The choice depends on the type of queries the system needs to handle most often.

c. ACID properties

ACID is a four set key properties that ensure reliable processing of database transactions. It stands for:

Atomicity - Each transaction is all or nothing, it either completes fully or doesn’t happen at all. If something goes wrong, the database rolls back to its previous state.

Consistency - The database must always remain in a valid state. A transaction takes the database from one consistent state to another, following all rules and constraints.

Isolation - Transactions happen independently, even if they run at the same time. This prevents data from being corrupted by overlapping operations.

Durability - Once a transaction is committed, the changes are permanent even if the system crashes right after.

Together, these properties make sure that data stays accurate, secure, and reliable, even in the face of errors or failures. They're especially important in systems like banking or healthcare, where data integrity is critical.