Eulogio “Amang” Rodriguez Institute of Science and Technology
College of Computing Students
Bachelor of Science and Information Technology Department

DATABASE MANAGEMENT SYSTEM 2
2nd Semester, SY. 2024-2025

LN:	BAUSAS	**FN:**	BEA ALEXA	**MI:**	D.	**DATE:**	07/04/2025

STUD.NO.	210-00924M	**SECTION:**	BS. INFOTECH – 2A	**WWNO:**	1

Answer the following. Ensure that your answer is studied well, as it is used for our discussion at the next meeting. Also, please provide your reference for each answer.

1. Define what database normalization is

Database normalization is the process of organizing data in a relational database to reduce data redundancy and improve data integrity. It involves dividing large tables into smaller, related tables and defining relationships between them using keys. The goal is to ensure that data is logically stored, efficient, and easy to maintain.

2. Discuss the processes of Database Normalization: 1NF, 2NF, and 3NF. Describe each process with examples.

The process of normalization is typically carried out through a series of stages known as **normal forms**. Each form builds on the previous one, addressing more advanced issues of data redundancy and dependency.

The **First Normal Form (1NF)** focuses on ensuring that the values in each column of a table are atomic, meaning indivisible. This means that each cell in a table must contain only a single value, and repeating groups or arrays are not allowed. This step is crucial for creating a solid foundation for relational structure.

Example of 1NF:

StudentID	Name	Courses
101	Bea Alexa	Math, Science

The table above is not in 1NF because the "Courses" column contains multiple values. To comply with 1NF, we must separate these values into individual records:

StudentID	Name	Courses
101	Bea Alexa	Math
101	Bea Alexa	Science

Once a relation is in 1NF, the next step is to ensure it meets the criteria for the **Second Normal Form (2NF).** A table is in 2NF if it is already in 1NF and all non-prime attributes are fully functionally dependent on the entire primary key. This means eliminating partial dependencies, where a non-prime attribute depends on part of a composite primary key rather than the whole.

Example of 2NF:

StudentID	Course	StudentName
101	Math	Bea Alexa

In this case, "StudentName" depends only on "StudentID" and not on the combination of "StudentID" and "Course", which is a violation of 2NF. By separating the student information into a new table, we eliminate the partial dependency:

Students Table:

StudentID	StudentName
101	Bea Alexa

Enrollments Table:

StudentID	Course
101	Math

Finally, the **Third Normal Form (3NF)** is achieved when the table is in 2NF and all its attributes are only dependent on the primary key. This form removes **transitive dependencies**, where a non-prime attribute depends on another non-prime attribute, which in turn depends on the primary key.

Example of 3NF:

StudentID	StudentName	AdvisorID	AdvisorName

The table above is not in 3NF because "AdvisorName" depends on "AdvisorID", which is not the primary key. To normalize this into 3NF, we should create a separate table for advisors:

Students Table:

StudentID	StudentName	AdvisorID
101	Bea Alexa	A01

Advisors Table:

AdvisorID	AdvisorName
A01	Prof. Carlos

Through these steps, data redundancy is significantly reduced, and the integrity and efficiency of the database are enhanced.

3. Differentiate Partial dependency, composite key, and primary key

To fully understand normalization, it's important to clarify the differences between three related concepts: **partial dependency**, **composite key**, and **primary key**.

A **primary key** is a unique identifier for a record in a table. It can consist of a single column (simple key) or multiple columns (composite key). Every table in a relational database should have a primary key to ensure each record is uniquely identifiable.

A **composite key** is a type of primary key that uses two or more attributes combined to form a unique identifier for each record. This is often used in junction tables that represent many-to-many relationships.
Partial dependency occurs when a non-prime attribute (i.e., an attribute that is not part of the primary key) is dependent on only a part of a composite primary key, rather than the whole key. This type of dependency leads to redundancy and must be eliminated in the process of achieving 2NF.

Concept	Definition	Example
Primary Key	Uniquely identifies each record in a table.	StudentID in a Student table.
Composite Key	A primary key that is made up of two or more attributes.	(StudentID, CourseID) in an Enrollment table.
Partial Dependency	A dependency where a non-prime attribute depends on only part of a composite key.	StudentName depending only on StudentID in a composite key table.

References:

1. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). *Database System Concepts* (6th ed.). McGraw-Hill Education.
2. Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of Database Systems* (7th ed.). Pearson Education.
3. Coronel, C., & Morris, S. (2015). *Database Systems: Design, Implementation, & Management* (11th ed.). Cengage Learning.