WEEK 1 ASSIGNMENT: DATABASE

1. A **database** is an organized collection of structured data, typically stored and accessed electronically from a computer system. It is designed to efficiently manage, retrieve, and update large volumes of data.
2. In a relational database, a **table** is a collection of data organized into rows and columns. Each row represents a record, and each column represents a specific attribute or field of the data.
3. **Record** also known as a row or tuple, a record is a single instance of data stored in a table. It contains values for each of the table's fields, representing a complete set of information about an entity.
4. **Field** also known as a column or attribute, a field is a single piece of data stored in a table. Each field corresponds to a specific characteristic or property of the data being stored.
5. A **primary** key is a unique identifier for each record in a table. It ensures that each record can be uniquely identified and provides a way to establish relationships between tables in a relational database.
6. **SQL** (Structured Query Language) is a standardized programming language used to manage and manipulate relational databases. It provides commands for querying data, defining and modifying database structures, and performing various data manipulation operations.
7. A **query** is a request for information from a database. It typically consists of SQL statements that specify the criteria for selecting and retrieving data from one or more tables.
8. An **index** is a data structure used to improve the speed of data retrieval operations in a database. It provides a way to quickly locate rows in a table based on the values of one or more columns.
9. **Normalization** is the process of organizing data in a database to reduce redundancy and improve data integrity. It involves breaking down a database into smaller, more manageable tables and establishing relationships between them.
10. A **Database Management System** (DBMS) is a software application or system that provides tools for storing, managing, and retrieving data in a database. It includes features for creating, modifying, and querying databases, as well as ensuring data security and integrity. Examples of DBMS include MySQL, PostgreSQL, Oracle Database, and Microsoft SQL Server.

2.1.1. Purpose of a Primary Key:

The primary key in a database table serves as a unique identifier for each record within that table. Its main purpose is to ensure data integrity and facilitate data retrieval and manipulation. By uniquely identifying each record, the primary key enables efficient querying, indexing, and referencing of data across different tables in a relational database.

Example: In a "Students" table, the primary key could be a column named "StudentID" assigned to each student. Each student's record in the table would have a unique "StudentID" value, allowing easy identification and retrieval of specific student information.

2.1.2. Difference between a DBMS and a Database:

A Database Management System (DBMS) is a software application or system that provides tools and services for managing databases. It includes functionalities for creating, modifying, querying, and administering databases. Examples of DBMS include MySQL, PostgreSQL, Oracle Database, and Microsoft SQL Server.

On the other hand, a database refers to an organized collection of structured data stored in electronic format. It consists of tables, each containing records with fields or attributes. A database can be created and managed using a DBMS.

In essence, a DBMS is the software that facilitates the creation and management of databases, while a database is the actual repository of data managed by the DBMS.

2.1.3. Importance of Normalization in Database Design:

Normalization is crucial in database design as it helps reduce data redundancy, improve data integrity, and optimize database performance. By organizing data into smaller, logically related tables and establishing relationships between them, normalization minimizes the risk of anomalies such as insertion, update, and deletion anomalies.

For example, consider a "Customer Orders" database where customer information and order details are stored in the same table. Without normalization, if a customer updates their address, it would require updating multiple records for each order they placed. However, by normalizing the database and separating customer information into a separate table, updates to customer data only need to be done in one place, improving data integrity and reducing redundancy.