**1. Define the following key terms related to databases:**

**Database -** Is an organized collection of structured information or data typically stored electronically in a computer.

**Table -** Is a set of data elements using a model of vertical columns and horizontal rows held in a table format within a database.

**Record -** Is a set of data stored in a table.

**Field -** Is a set of values arranged in a table and has same data type.

**Primary Key** **-** Is the column or columns that contains values that uniquely identify each row in a table.

**SQL -** Means Structured Query Language – Is a programming language for storing and processing information in a relational database.

**Query -** Is simply a request for data.

**Index -** Is an index offers an efficient way to quickly access the records from the database file stored on the desk drive.

**Normalization -** Is the process of organization of data database.

**Database Management System (DBMS)** - Is a computerized data system.

**2.1.1. Describe the purpose of a primary key in a database and provide an example.**

**A primary key** is a field or a combination of fields in a database table that uniquely identifies each record in that table. It serves as a means of uniquely identifying and accessing individual records in a database. Common examples of primary keys include social security numbers, email addresses, or unique identification numbers assigned to each record in the table.

It's important to carefully choose or design a primary key when designing a database, as it plays a crucial role in data integrity, relationships, and the overall functionality of the database.

**A primary key in a database serves several essential purposes:**

**Uniqueness:** A primary key uniquely identifies each record (row) in a table. No two records can have the same primary key value. This uniqueness ensures that each piece of data in the database can be identified and distinguished from others.

**Identification:** The primary key provides a way to uniquely identify and locate a specific record within a table. This is crucial for data retrieval, modification, and deletion operations. Without a primary key, it would be challenging to pinpoint and manipulate individual records.

**Indexing:** Primary keys are automatically indexed in most database systems. Indexing enhances the speed of data retrieval operations by providing a quick and efficient way to locate specific records based on the primary key.

**Relationships:** In relational databases, primary keys are used to establish relationships between tables. Foreign keys in other tables typically reference the primary key of another table. This linkage is fundamental for creating associations and connections between different sets of data.

**Data Integrity:** Primary keys play a vital role in maintaining data integrity. They help prevent duplicate records and ensure that each record is distinct. This is essential for accuracy and reliability of the data stored in the database.

**Enforcement of Constraints:** Primary keys often serve as a basis for enforcing constraints in the database, such as uniqueness and referential integrity. These constraints help ensure that the data remains consistent and accurate over time.

**Here’s is an example of a Primary key in a database table;**

**Customer ID;** In a customer table, a "Customer ID" field could serve as the primary key, guaranteeing each customer has a distinct identifier.

|  |  |  |  |
| --- | --- | --- | --- |
| **Customer ID** | **First name** | **Last name** | **Email** |
| **1001** | **Alice** | **Johnson** | **alice.j@example.com** |
| **1002** | **Bob** | **Smith** | **bob.smith@example.com** |
| **1003** | **Carol** | **White** | **carol.w@example.com** |

**In summary;** the primary key is a critical concept in database design, providing a unique identifier for records, facilitating relationships between tables, ensuring data integrity, and supporting efficient data retrieval operations.

**2.1.2 Explain the differences between Database Management System (DBMS) and Database.**

Due to the learning of the Structured query language in **Database course,** which means a **Structured query language (SQL)** is a programming language for storing and processing information in a relational database. A relational database stores information in tabular form, with rows and columns representing different data attributes and the various relationships between the data values. Here by one can learn various conceptual similarities and also the difference between a **Database** and the **Database Management System.**

**The following are the differences between Database Management System (DBMS) and Database:**

**Database Management System (DBMS):** is software that provides an interface for interacting with databases. It acts as an intermediary between the user and the database. While for **Database;** is an organized collection of data. It can be as simple as a single file or as complex as a set of interconnected files and tables.

**The primary function of a DBMS** is to manage and control access to the database. It handles tasks such as data storage, retrieval, modification, and deletion. While for **Databases** are designed to store, organize, and manage large volumes of data in a structured format.

**DBMS** enables users to define the structure of the database, including tables, relationships, and constraints. It also provides tools for querying the data and ensuring data integrity. While for **the data in a Database** is typically organized into tables, each consisting of rows and columns. Tables are used to represent entities, and relationships between tables can be established.

**Database Management System** Popular options include; MySQL, Postgre SQL, Microsoft SQL Server, Oracle Database, and SQLite.DBMS supports the execution of queries and transactions, ensuring data consistency and integrity. While; Data in a **database** is organized into tables, rows, and columns, and it can include various types of information. While for **Examples of Database** include a spreadsheet, a collection of text files, or a sophisticated relational database system.

**In summary, a** **Database** is the structured collection of data, while **a Database** **Management System** is the software that manages and provides an interface to interact with that data. The DBMS facilitates the creation, maintenance, and manipulation of databases, ensuring efficient and secure handling of information. It acts as a bridge between the user and the raw data stored in the database.

**2.1.3. Discuss in short, Importance of normalization in database design and provide an example and how it can improve data integrity.**

**A** **Database** is information that is set up for easy access, management and updating. Computer databases typically store aggregations of data records or files that contain information, such as sales transactions, customer data, financials and product information. While for **Database Management System (DBMS);** enables users to create and manage a database. It also helps users create, read, update and delete data in a database, and it assists with logging and auditing functions.

**The DBMS** provides physical and logical independence from data. Users and applications do not need to know either the physical or logical locations of data. A DBMS can also limit and control access to the database and provide different views of the same database schema to multiple users.

**Normalization is important in database design due to the following reasons;**

**Minimizing Data Redundancy;** By organizing data into related tables, redundancy is minimized. Redundant data can lead to inconsistencies and increases the likelihood of update anomalies.

**Preventing Update Anomalies;** Normalization helps to avoid anomalies such as insertion, update, and deletion anomalies. An update anomaly occurs when updating a piece of data in one place leads to inconsistencies or errors.

**Improving Data Integrity;** Normalization enhances data integrity by reducing the risk of conflicting information. Data is stored in a more organized and consistent manner.

**Simplifying Schema Maintenance;** Maintenance becomes more straightforward because changes or updates to the database structure can be localized to specific tables. This contrasts with denormalized structures where changes may require updates in multiple places.

**Facilitating Querying and Reporting;** Well-normalized databases make it easier to query and report on data. The relationships between tables allow for efficient retrieval of information without the need for complex joins or subqueries.

**Facilitated Data Validation;** with normalization, data validation becomes more straightforward. Validation rules can be applied at the table level, ensuring that only accurate and relevant information is stored. This contributes to maintaining the integrity of the data**.**

**Heres are examples to illustrate normalization:**

**Consider an initial table for a library;**

Table: Books

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Book ID | Title | Author | Genre | ISBN | Price |
| 1 | "The Great Gatsby" | F. Scott Fitzgerald | Fiction | 978-0743273565 | $10 |
| 2 | "To Kill a Mockingbird" | Harper Lee | Fiction | 978-0061120084 | $12 |
| 3 | "The Catcher in the Rye" | J.D. Salinger | Fiction | 978-0241950425 | $8 |

This table is not normalized and contains redundancy. To normalize it, you might create separate tables for Authors, Genres, and Books:

Table: Authors

|  |  |
| --- | --- |
| Author ID | Author |
| 101 | F. Scott Fitzgerald |
| 102 | Harper Lee |
| 103 | J.D. Salinger |
| Table: | Genres |

Table: Genres

|  |  |
| --- | --- |
| Genre ID | Genre |
| 201 | Fiction |

Table: Books

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Book ID | Title | Author ID | Genre ID | ISBN | Price |
| 1 | "The Great Gatsby" | 101 | 201 | 978-0743273565 | $10 |
| 2 | "To Kill a Mockingbird" | 102 | 201 | 978-0061120084 | $12 |
| 3 | "The Catcher in the Rye" | 103 | 201 | 978-0241950425 | $8 |

This is a simplified example, but it demonstrates how normalization can help eliminate redundancy and establish relationships between entities in a more efficient way.

**In summary, normalization is essential** for maintaining high levels of data integrity by preventing anomalies, ensuring consistent relationships, reducing redundancy, and facilitating effective data validation. It provides a systematic approach to organizing data that contributes to the accuracy and reliability of the information stored in a database.