CCINFOM Fundamentals of Databases

Introduction to Databases

Components of a Database System

- **Database**: A structured collection of data stored in a logical and accessible format.
    - Used to enable efficient access, management, and storage of data for various applications.
    - Databases use tables, rows, and columns in relational systems or other structures such as key-value pairs, documents, or graphs in non-relational systems.

BookID	Title	Author	Genre	CopiesAvailable
101	Database Systems	C. Coronel	Technology	5
102	SQL Queries for Mere Mortals	J. Viescas	Technology	3

- **Database Management System (DBMS)**: A software that acts as an intermediary between users or applications and the database itself.
    - Provides functionality for data storage, retrieval, updating, and deletion, as well as security, concurrency control, and recovery.
    - Key functions of a DBMS include:
        - **Data Definition**: Define the structure of the data using Data Definition Language (DDL).
        - **Data Manipulation**: Perform modifications using Data Manipulation Language (DML).
        - **Data Querying**: Perform queries using Data Query Language (DQL).
        - **Data Control**: Manage access permissions using Data Control Language (DCL).
- **Database Application**: A database application is a software program developed to interact with the DBMS.

```
SELECT * FROM Books WHERE CopiesAvailable > 0;
```

-
    - Enables users to input, query, or manipulate data in the database.
    - Built using programming languages (e.g., Python, Java) and include embedded SQL commands.
    - **Database Management Tools**: Used by database professionals (e.g., administrators, architects) to monitor, manage, and optimize databases.

        - Examples include tools like phpMyAdmin for MySQL or SQL Server Management Studio (SSMS) for SQL Server.

Subsets of SQL

- **Data Definition Language (DDL)**: Used to define, modify, and manage database schema and structure. It includes commands to create, alter, or drop database objects.
    - **CREATE**: Creates new database objects (e.g., tables, views).

```
CREATE TABLE Books (
    BookID INT PRIMARY KEY,
    Title VARCHAR(100) NOT NULL,
    Author VARCHAR(100),
    Genre VARCHAR(50),
    CopiesAvailable INT DEFAULT 0
);
```

    - **ALTER**: Modifies existing database objects.

```
ALTER TABLE Books ADD COLUMN Publisher VARCHAR(100);
```

    - **DROP**: Deletes database objects permanently.

```
DROP TABLE Books;
```

- **Data Manipulation Language (DML)**: Used to manipulate data within existing database objects. It includes operations to insert, update, and delete data.
    - **INSERT**: Adds new data to tables.

```
INSERT INTO Books (BookID, Title, Author, Genre, CopiesAvailable)
VALUES (101, 'Database Systems', 'C. J. Date', 'Technology', 5);
```

    - **UPDATE**: Modifies existing records.

```
UPDATE Books SET CopiesAvailable = CopiesAvailable - 1 WHERE BookID = 101;
```

    - **DELETE**: Removes data from tables.

```
DELETE FROM Books WHERE BookID = 101;
```

- **Data Query Language (DQL)**: Used for performing queries on the data.
    - **SELECT**: Retrieves data from tables.

```
SELECT Title, Author FROM Books WHERE CopiesAvailable > 0;
```

- **Data Control Language (DCL)**: Deals with the rights and permissions of the database system. Used to control access to data by granting or revoking permissions.
  - **GRANT**: Assigns new privileges to a user.
  - **REVOKE**: Removes previously granted privileges from a user.

Data Management Lifecycle

- **Input (I)**: Collecting raw data for storage in the database.
  - Example: A user fills out a form to borrow a book.
- **Create (C)**: Adding data to the database using DML commands.

```
INSERT INTO Loans (LoanID, BookID, MemberID, LoanDate, DueDate)
VALUES (1, 101, 1001, '2024-11-29', '2024-12-06');
```

- **Read (R)**: Accessing and viewing data without modifying it.

```
SELECT * FROM Loans WHERE DueDate > '2024-11-29';
```

- **Update (U)**: Modifying existing records in the database.

```
UPDATE Books
SET CopiesAvailable = CopiesAvailable - 1
WHERE BookID = 101;
```

- **Delete (D)**: Removing obsolete or unnecessary data.

```
DELETE FROM Members WHERE MemberID = 1002;
```

- **Organize (O)**: Structuring or sorting data for easier analysis.

```
SELECT * FROM Books ORDER BY Title ASC;
```

- **Processing (P)**: Performing calculations or transformations.

```
SELECT AVG(CopiesAvailable) AS AvgCopies FROM Books;
```

- **Share (S)**: Distributing data to authorized users or applications.

Types of Data Processing

- **T1: Form Records**: The process of adding new records to the database, typically through forms or structured input interfaces.
    - Ensures data is entered systematically and adheres to predefined constraints.
    - Example: Adding new customer information in a CRM system.

```sql
INSERT INTO Customers (CustomerID, Name, Email)
VALUES (1, 'John Doe', 'john.doe@example.com');
```

- **T2: Mathematical Processing and Update/Deletion of Records**: This type involves performing calculations on stored data and updating or removing existing records in the database.
    - **Mathematical Processing**: Used for computations like totals, averages, or other derived values.

```sql
SELECT SUM(Amount) AS TotalSales FROM Sales WHERE SaleDate >= '2024-01-01';
```

    - **Update of Records**: Modify existing data to reflect changes.

```sql
UPDATE Customers
SET Email = 'new.email@example.com'
WHERE CustomerID = 1;
```

    - **Deletion of Records**: Remove records that are no longer relevant or required.

```sql
DELETE FROM Customers WHERE LastActive < '2023-01-01';
```

- **T3: Relating Records**: This involves establishing and utilizing relationships between records in different tables.
    - Allows for meaningful queries across related data.

```sql
SELECT Customers.Name, Orders.OrderID, Orders.TotalAmount
FROM Customers
INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID;
```

- **T4: Generation of Lists, Reports, and Filtering Results**: This type of processing involves extracting and presenting data in meaningful ways, often for decision-making or analysis.
    - **Generation of Lists**: Produce simple lists of data matching specific criteria.

```sql
SELECT EmployeeID, Name
FROM Employees
WHERE Status = 'Active';
```

- **Generation of Reports**: Create detailed summaries or grouped data.

```sql
SELECT ProductCategory, SUM(Amount) AS TotalSales
FROM Sales
GROUP BY ProductCategory;
```

- **Grouping, Filtering, and Sorting**: Organize and refine data for easier interpretation.

```sql
SELECT ProductName, SUM(Amount) AS TotalSales
FROM Sales
GROUP BY ProductName
HAVING SUM(Amount) > 10000
ORDER BY TotalSales DESC;
```