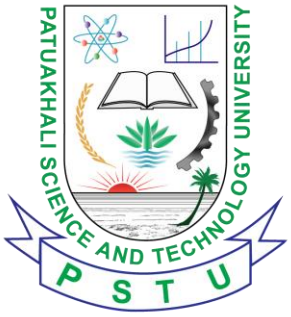


**Course Title : Database System**
**Course Code : CCE - 223**

# Assignment : 02

## Submitted To

**Dr. Md. Samsuzzaman**
Professor
Department of Computer and Communication Engineering.
Faculty Of Computer Science & Engineering.

## Submitted By

**Name** : Md Mohidul Alam
**Id** : 1902016
**Reg. No** : 08722
**Session** : 2019 - 2020

Patuakhali Science & Technology University
Dumki, Patuakhali - 8602

### 1.1 This chapter has described several major advantages of a database System. What are two disadvantages?

**Two disadvantages of a database system are:**

**Complexity:** Database systems can be complex to design, set up, and maintain. This can lead to a higher cost in terms of time, money, and resources.
**Overhead:** Database systems can have more overhead than a file system, as additional processing is required to maintain data integrity and enforce data constraints.

### 1.2 List five ways in which the type declaration system of a language such as Java or C++ differs from the data definition language used in a database.

Five ways in which the type declaration system of a language such as Java or C++ differs from the data definition language used in a database are: Syntax: The syntax used to declare types in a programming language is different from that used in a database.
Scope: Types declared in a programming language have a scope limited to the program or module, while database schemas are typically shared across applications.
Relationships: Programming languages typically do not have built-in support for specifying relationships between data entities, while databases do.
Constraints: Databases provide support for declarative constraints, such as unique keys and foreign keys, while programming languages provide support for imperative constraints.
Abstraction: Programming languages provide more abstraction than database schemas, allowing for easier manipulation of complex data structures.

### 1.3 Six major steps that one would take in setting up a database for a Particular enterprise are:

**Requirement analysis:** Identifying the data requirements of the enterprise and defining the scope of the database.
**Conceptual design:** Creating a conceptual data model that captures the entities and relationships in the enterprise.
**Logical design:** Transforming the conceptual model into a logical data model that can be implemented in a specific database management system.
**Physical design:** Deciding on the hardware and software components of the database system, including storage devices, network connectivity, and server software.
**Implementation:** Creating the database schema, loading initial data, and defining security and access controls.
**Testing and maintenance:** Testing the database to ensure that it meets the enterprise's requirements, and maintaining it over time to ensure its continued usefulness.

**1.4 When considering the storage of actual video data and metadata in a file-processing system, the following points from section 1.2 are relevant: Data redundancy:** Video files can be very large, and storing multiple copies of the same file can waste space and make it difficult to manage consistency.

**Data inconsistency:** Without a central system to manage metadata, inconsistencies can arise between different copies of the same data, leading to errors and confusion.
**Data isolation:** File-processing systems do not provide any mechanisms to manage concurrent access to data, which can lead to race conditions and inconsistent updates.
**Security:** Without built-in security features, file-processing systems can be vulnerable to unauthorized access and data breaches.
**Integrity:** Without transaction management, it can be difficult to maintain data integrity in a file-processing system.

**1.5 Key differences between keyword queries used in web search and database queries are:**
**Syntax:** Web search queries are typically entered as keywords or natural language phrases, while database queries are written in a specific query language.
**Result set:** Web search queries typically return a ranked list of results, while database queries return a structured set of rows and columns. **Scope:** Web search queries are typically broad in scope and return results from across the web, while database queries are typically targeted at a specific database and return results from a specific set of data.
**Consistency:** Web search results can be influenced by factors such as user location and search history, while database queries are designed to return consistent and repeatable results.
**Performance:** Web search engines are optimized for low latency and high throughput, while database systems are optimized for fast access to structured data.

**1.6 List four applications you have used that most likely employed a Database system to store persistent data.**
**Facebook -** Facebook uses a database system to store user data such as profile information, posts, comments, likes, and so on. The data is stored in a structured manner in a relational database system.
**Amazon -** Amazon uses a database system to store product information, order history, customer information, and so on. The data is stored in a structured manner in a relational database system.
**Netflix -** Netflix uses a database system to store user profiles, viewing history, movie and TV show information, and so on. The data is stored in a structured manner in a relational database system.
**Google -** Google uses a database system to store various types of data, such as search queries, user information, web pages, and so on. The data is stored in a structured manner in a distributed database system.

### 1.7 List four significant differences between a file-processing system and a DBMS.

i. **Data Independence:** In a file-processing system, data is usually stored in files that are specific to each application. This makes it difficult to change the structure of the data without impacting the application. In contrast, a DBMS provides data independence, which means that the data structure can be changed without affecting the application programs that use the data.

ii. **Data Integrity:** In a file-processing system, it is the responsibility of the application to maintain integrity. This can lead to problems such as duplicate data, inconsistent data, and data corruption. A DBMS, on the other hand, provides mechanisms to ensure data integrity, such as constraints, triggers, and referential integrity.

iii. **Data Sharing:** A file-processing system is usually designed to serve a specific application and is datanot designed to share data with other applications. A DBMS, on the other hand, is designed to allow multiple applications to access the same data concurrently.

iv. **Data Security:** In a file-processing system, security measures such as access controls and encryption are usually implemented within the application. In a DBMS, security measures are built into the system and can be applied at the user or object level, providing a more secure environment for data.

### 1.8 Explain the concept of physical data independence and its importance in database systems.

Physical data independence is the property of a database system that allows changes to be made to the physical storage structures of the database without affecting the logical schema or application programs that access the data. In other words, it enables changes to be made to the way data is stored without affecting the way it is accessed or used.

The importance of physical data independence in database systems lies in the fact that it provides flexibility and scalability. It allows database administrators to make changes to the storage structures of the database, such as adding or removing indexes, changing file organizations, or modifying storage media, without affecting the applications that use the database. This can be especially important in large-scale database systems, where performance and efficiency are critical.

Physical data independence also allows for easy migration of a database from one platform to another. For example, if a database needs to be moved from one type of hardware to another, or from one operating system to another, physical data independence ensures that the database can be easily transferred without requiring changes to the application programs that use it.

Overall, physical data independence is an essential feature of database systems that enables greater flexibility and scalability, as well as simplifying database maintenance and management.

### 1.9 List five responsibilities of a database-management system. For each responsibility, explain the problems that would arise if the responsibility were not discharged.

Data storage management: The database management system (DBMS) is responsible for efficiently and effectively storing and retrieving data from the database. Without proper data storage management, the data can become corrupted, lost, or difficult to access, leading to data inconsistencies, poor performance, and increased maintenance costs.

Data security management: The DBMS must ensure that the database is secure and protected from unauthorized access, modification, or theft. Without proper data security management, the database can be vulnerable to security breaches, data theft, and other security threats, resulting in loss of valuable data, reputational damage, and legal liabilities.

Data integrity management: The DBMS must ensure that the data stored in the database is accurate, consistent, and complete. Without proper data integrity management, data inconsistencies, inaccuracies, and errors can occur, leading to incorrect decision-making, regulatory non-compliance, and reputational damage.

Data concurrency management: The DBMS must ensure that multiple users can access and update the same data simultaneously without conflicts. Without proper data concurrency management, data conflicts, lost updates, and other issues can occur, leading to data inconsistencies, incorrect results, and user frustration.

Data backup and recovery management: The DBMS must ensure that the database is backed up regularly and can be recovered in case of a system failure, disaster, or other unforeseen events. Without proper data backup and recovery management, the database can be lost, resulting in significant downtime, data loss, and financial losses.

Overall, the responsibilities of a database-management system are critical to ensuring the reliability, security, and performance of a database. Failing to discharge any of these responsibilities can lead to a range of problems, including data inconsistencies, security breaches, user frustration, reputational damage, and financial losses.

### 1.10 List at least two reasons why database systems support data manipulation using a declarative query language such as SQL, instead of just providing a library of C or C++ functions to carry out data manipulation.

Increased productivity and ease of use: A declarative query language, such as SQL, allows users to express data manipulation operations in a way that is more natural and intuitive than using a programming language. This enables users to be more productive and efficient, as they do not need to write complex and error-prone code to access and manipulate the data.

Data independence: A declarative query language provides a higher level of abstraction between the user and the underlying database, which allows for greater data independence. Users can write queries that specify what data they want to retrieve or manipulate, without needing to know how the data is stored or how the query will be executed. This means that

changes to the database schema or the physical storage structure can be made without affecting the applications that use the database, as long as the same query semantics are maintained.
In summary, the use of a declarative query language such as SQL provides a more natural and intuitive way for users to manipulate data, while also providing greater data independence and allowing for changes to be made to the database schema or physical storage structure without affecting the applications that use the database.

### 1.11 Assume that two students are trying to register for a course in which there is only one open seat. What component of a database system prevents both students from being given that last seat?

The component of a database system that prevents both students from being given the last seat in a course is the concurrency control mechanism.

Concurrency control ensures that multiple users can access and update the same data simultaneously without conflicts. In this case, both students are attempting to access the same piece of data, namely the number of open seats in the course.

The concurrency control mechanism will ensure that only one student is able to successfully register for the course and that the other student is prevented from registering, as there is only one open seat. This is typically achieved through the use of locking, which prevents other users from accessing or modifying the same data while it is being used by another user.
Therefore, the concurrency control mechanism is an essential component of a database system that ensures data consistency and prevents conflicts when multiple users are accessing and modifying the same data.

### 1.12 Explain the difference between two-tier and three-tier application architectures. Which is better suited for web applications? Why?

Two-tier and three-tier application architectures are different approaches to designing software applications.

In a two-tier architecture, also known as a client-server architecture, the application is divided into two layers: the client layer and the server layer. The client layer, which runs on the user's computer, is responsible for the presentation logic and user interface. The server layer, which runs on a remote server, is responsible for the application logic and data management.

In a three-tier architecture, the application is divided into three layers: the presentation layer, the application logic layer, and the data storage layer. The presentation layer, also known as the user interface layer, is responsible for displaying information to the user and receiving input. The application logic layer, also known as the business logic layer, is responsible for processing the data and implementing the business rules of the application. The data storage layer, also known as the data access layer, is responsible for managing the data and interacting with the database.

Three-tier architecture is better suited for web applications because it provides better scalability, maintainability, and security. The three-tier architecture separates the concerns of the application, making it easier to maintain and modify the different layers independently. It also allows for better scalability by enabling the application to be deployed on multiple servers, each responsible for a different layer, which can handle higher traffic loads. Additionally, the three-tier architecture provides better security by isolating the data storage layer from the application logic and presentation layers, which can help prevent unauthorized access and attacks.
In summary, while two-tier architecture may be suitable for smaller applications, three-tier architecture is better suited for web applications due to its scalability, maintainability, and security benefits.

### 1.13 List two features developed in the 2000s and that help database systems handle data-analytics workloads.

Two features developed in the 2000s that help database systems handle data- analytics workloads are:
Columnar storage: Columnar storage is a storage format where data is stored in columns rather than rows. This format is particularly useful for data analytics workloads, as it allows for efficient processing of large volumes of data by enabling selective loading and scanning of only the columns that are required for a particular query. Columnar storage can also facilitate better compression and better performance on modern CPU architectures.
In-memory processing: In-memory processing refers to the use of main memory, rather than disk-based storage, to hold data in a database system. This can result in significant performance improvements for data analytics workloads, as data can be processed much faster when it is stored in memory rather than on disk. In-memory processing can also reduce the need for indexing and other optimizations, as data can be accessed and manipulated more quickly.
Both columnar storage and in-memory processing have become increasingly important features for database systems in the 2000s, as the volume and complexity of data being generated by organizations has grown exponentially, and the need for fast and efficient data analytics has become critical for businesses to remain competitive.

### 1.14 Explain why NoSQL systems emerged in the 2000s, and briefly contrast their features with traditional database systems.

NoSQL (Not Only SQL) systems emerged in the 2000s to address the limitations of traditional relational database management systems (RDBMS) in handling large-scale and complex data.

Traditional database systems were designed for structured data and were not well- suited to handle unstructured or semi-structured data, such as social media posts, sensor data, or multimedia content. RDBMS also have limitations in scalability, flexibility, and availability when dealing with distributed data and big data workloads.

NoSQL systems, on the other hand, were designed to handle these limitations and provide a more flexible, scalable, and available data management solution. NoSQL systems are

schema-agnostic, meaning they do not require a pre-defined schema and can handle unstructured and semi-structured data. They are also highly scalable, using distributed architectures that allow for horizontal scaling across multiple nodes. Additionally, NoSQL systems are optimized for read and write operations, making them suitable for high-volume and low-latency applications.
However, NoSQL systems sacrifice some of the consistency, atomicity, isolation, and durability (ACID) properties that are typically provided by traditional RDBMS. Instead, NoSQL systems provide a different set of guarantees, such as eventual consistency and high availability. NoSQL systems also often lack advanced query capabilities, such as joins and complex transactions, and may require more complex application logic to handle data operations.

In summary, NoSQL systems emerged in the 2000s to address the limitations of traditional database systems in handling large-scale and complex data. While NoSQL systems offer greater flexibility and scalability, they sacrifice some of the ACID guarantees and advanced query capabilities provided by traditional RDBMS.

### 1.15 Describe at least three tables that might be used to store information in a social networking system such as Facebook.

A social networking system such as Facebook would likely use a large number of tables to store its information, but here are three example tables that might be used:

User Table: This table would store information about each individual user, such as their name, email address, date of birth, gender, and location. It might also include information about their privacy settings, such as who can see their posts and personal information.

Post Table: This table would store information about each individual post that a user creates, such as the text of the post, any images or videos that are attached, and the date and time it was posted. It might also include information about the post's privacy settings, such as who can see it and whether it's public or private.

Friends Table: This table would store information about the connections between users, such as who each user is friends with or follows. It might include fields such as the user ID of each friend, the date they became friends, and any groups or pages they both belong to. This table could also be used to enforce privacy settings and ensure that only authorized users can see each other's posts and personal information.

These are just a few examples of the types of tables that a social networking system like Facebook might use. Other tables might include tables for storing messages, notifications, comments, likes, and more.