Vitaliy Prymak Assignment 1 02/07/24

Problem 1.6 List four applications you have used that most likely employed a database system to store persistent data.

- **Facebook**: Facebook utilizes a complex database system to store user profiles, posts, comments, likes, messages, and other relevant data. This data is structured and stored in databases to ensure efficient retrieval and management.
- **Amazon**: Amazon relies heavily on databases to manage product listings, user accounts, orders, and transaction histories. The data stored in these databases enables personalized recommendations, efficient order processing, and inventory management.
- **Netflix**: Netflix utilizes databases to store user preferences, viewing history, content metadata, and streaming information. This data is crucial for recommending personalized content to users and optimizing their streaming experience.
- **Airbnb**: Airbnb employs databases to manage property listings, user profiles, bookings, reviews, and payment information. These databases enable efficient matching of guests with accommodations, managing reservations, and ensuring a smooth booking process for users.

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

- **Data Redundancy and Consistency**:
  - **File-Processing System**: In a file-processing system, data redundancy is common since each file typically maintains its own set of data. This redundancy can lead to inconsistencies if the same data is stored in multiple files but is updated inconsistently.
  - **DBMS**: A DBMS minimizes data redundancy by centralizing data storage in a database. Data is organized in a structured manner, and relationships between data entities are defined, reducing redundancy and ensuring consistency through features like

normalization and ACID (Atomicity, Consistency, Isolation, Durability) properties.

- **Data Integrity and Security**:
  - **File-Processing System**: File-based systems often lack robust mechanisms for ensuring data integrity and security. Access control and data validation are typically implemented at the application level, making it prone to errors and vulnerabilities.
  - **DBMS**: A DBMS provides built-in mechanisms for enforcing data integrity constraints, such as primary key constraints, foreign key constraints, and referential integrity. Additionally, DBMSs offer sophisticated security features, including authentication, authorization, encryption, and auditing, to protect sensitive data from unauthorized access and manipulation.
- **Data Independence and Flexibility**:
  - **File-Processing System**: In a file-based system, applications are tightly coupled with the file structures and formats, making it difficult to modify or extend the system without impacting existing applications.
  - **DBMS**: A DBMS provides data independence, allowing applications to be independent of the underlying database schema. Changes to the database structure can be made without affecting the application programs, enhancing flexibility and facilitating easier system evolution and maintenance.
- **Concurrency Control and Transaction Management**:
  - **File-Processing System**: File-based systems often lack built-in mechanisms for managing concurrent access to data by multiple users or processes. This can lead to data inconsistency issues, such as lost updates or data corruption.

- **DBMS**: A DBMS offers robust concurrency control mechanisms to manage simultaneous access to data by multiple users or transactions. Techniques such as locking, timestamping, and multi-version concurrency control ensure data consistency and isolation, while transaction management features provide atomicity and durability guarantees for database operations.

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

- **Separation of Concerns**: Physical data independence separates the concerns of how data is stored physically from how it is logically represented and accessed by users or applications. This abstraction allows database administrators (DBAs) to focus on optimizing storage and performance without impacting the logical structure of the database.
- **Flexibility and Adaptability**: Physical data independence provides flexibility and adaptability to changing hardware or storage technologies. For example, if a DBMS needs to migrate from one storage technology to another (e.g., moving from traditional hard disk drives to solid-state drives), it can do so transparently to users and applications as long as the logical schema remains unchanged.
- **Performance Optimization**: Database administrators can optimize the physical storage structures based on performance requirements without affecting the logical schema. This may involve decisions such as index creation, partitioning, data compression, or clustering strategies to improve query performance, storage efficiency, and resource utilization.
- **Reduced Dependency on Hardware**: Physical data independence reduces dependency on specific hardware configurations or storage architectures. As hardware evolves or upgrades are required, DBAs can make adjustments to

the physical storage layer without disrupting the logical representation of data or requiring modifications to application code.

- **Database Evolution and Scalability**: Physical data independence facilitates database evolution and scalability by allowing changes to the underlying storage mechanisms as the database grows or requirements change. It enables seamless upgrades, migrations, or expansions of storage infrastructure to accommodate increasing data volumes or performance demands.

Problem 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 and Organization**:
  - Responsibility: The DBMS is responsible for efficiently storing and organizing data in a structured manner, typically using tables, rows, and columns.
  - Problem if Not Discharged: Without proper data storage and organization, data may become fragmented, scattered, or duplicated across multiple sources. This fragmentation can lead to data inconsistency, increased storage space requirements, and difficulty in retrieving and managing data efficiently.
- **Data Retrieval and Manipulation**:
  - Responsibility: The DBMS provides mechanisms for users and applications to retrieve, update, insert, and delete data from the database using queries and commands.
  - Problem if Not Discharged: Without effective data retrieval and manipulation capabilities, users and applications would struggle to access and modify data stored in the database. This could result in inefficient data processing, data integrity issues, and hindered functionality of the systems relying on the database.

- **Data Security and Access Control**:
  - Responsibility: The DBMS enforces security measures to protect the confidentiality, integrity, and availability of data stored in the database. This includes authentication, authorization, encryption, and auditing mechanisms.
  - Problem if Not Discharged: Inadequate data security and access control can lead to unauthorized access, data breaches, data manipulation, and data loss. This compromises the confidentiality of sensitive information, undermines trust in the system, and exposes organizations to legal and regulatory liabilities.
- **Concurrency Control and Transaction Management**:
  - Responsibility: The DBMS manages concurrent access to data by multiple users or transactions to ensure consistency and isolation. It implements techniques such as locking, timestamping, and transaction management to maintain data integrity.
  - Problem if Not Discharged: Without proper concurrency control and transaction management, concurrent transactions may interfere with each other, leading to data inconsistencies, lost updates, and transaction aborts. This can result in incorrect query results, data corruption, and degraded system performance.
- **Data Integrity and Consistency**:
  - Responsibility: The DBMS enforces data integrity constraints, such as primary key constraints, foreign key constraints, and referential integrity rules, to maintain the accuracy and consistency of data.
  - Problem if Not Discharged: Without data integrity enforcement, the database may contain duplicate, invalid, or inconsistent data, compromising its reliability and usability. This can lead to incorrect decision-

making, errors in business processes, and erosion of trust in the data stored in the database.