Course Info: Database Development and Administration FALL-2023-71478-INFT-6203-F01
Instructor: Dr. Abdus Salam Siddique
Student Name: Seyed Mohammad Sanjari Pirmahalleh
T#: T01294563
Assignment Number: Assignment 11, Week 12
Date: Nov 21, 2023

---

1)

1. **Concurrency Control**: Managing concurrent access to the database by multiple users is challenging. It's essential to ensure that one user's work does not inappropriately influence another user's work, which requires a careful balance between the level of protection and system throughput.

2. **Deadlock**: This occurs when two transactions are each waiting on a resource that the other transaction holds. Preventing and resolving deadlocks, where transactions are stuck waiting indefinitely, is a complex problem.

3. **Locking Mechanisms**: Implementing efficient locking mechanisms (optimistic and pessimistic locking) to manage access to data is challenging. Locks can be at different levels of granularity (row, page, table, database), and choosing the right granularity and type of lock (exclusive or shared) is crucial to prevent conflicts and maintain data integrity.

4. **Security**: Ensuring database security to allow only authorized users to perform authorized activities is difficult. This includes determining users' processing rights, enforcing security requirements, and protecting against SQL injection attacks and other security threats.

5. **Recovery Management**: Managing database recovery processes such as rollback, rollforward, and setting checkpoints is complex. This involves keeping a transaction log, synchronizing the database with the log, and ensuring the integrity and durability of data.

---

2)

The difference between a database administrator (DBA) and a data administrator is as follows:

- **Data Administration**: This function applies to the entire organization. It involves overseeing the organization's data policies, data quality, and overall data management strategy. This role is more concerned with the governance of data across the organization and ensuring that data is used effectively and responsibly.

- **Database Administration (DBA)**: This is a more technical role specific to a particular database and includes the applications that process that database. The DBA is responsible for managing the database structure, controlling concurrent processing, managing processing rights and responsibilities, developing database security, providing for database recovery, managing the Database Management System (DBMS), and maintaining the data repository. This role is more focused on the technical aspects of database management and maintenance, including database design, optimization, and ensuring the database is secure and operates efficiently.

3)

Seven important tasks of a Database Administrator (DBA) are:

- **Manage Database Structure**: Overseeing the structure of the database to ensure it meets the needs of the applications and users.
- **Control Concurrent Processing**: Managing the simultaneous processing of multiple operations, typically to ensure data consistency and integrity.
- **Manage Processing Rights and Responsibilities**: Assigning and controlling the rights and responsibilities of users accessing the database.
- **Develop Database Security**: Establishing and maintaining security measures to protect the database from unauthorized access and breaches.
- **Provide for Database Recovery**: Ensuring that there are mechanisms in place for data recovery in case of system failures or other issues that might compromise data integrity.
- **Manage the DBMS**: Overseeing the operation of the Database Management System, including its configuration, performance, and maintenance.
- **Maintain the Data Repository**: Keeping the repository of metadata about users, databases, and applications up-to-date and accurate.

4)

The DBA's responsibilities for managing database structure include:

- **Participating in Database and Application Development**: The DBA is involved in the development phase of both the database and its associated applications. This includes collaborating with other teams to ensure that the database aligns well with the application requirements.
- **Assisting in Requirements Analysis and Data Model Creation**: The DBA plays a crucial role in analyzing the requirements for the database and helps in creating the data model. This involves understanding the data needs of the organization and how they translate into a structured database design.
- **Playing an Active Role in Database Design and Creation**: The DBA is not just a passive participant but actively involved in the design and creation of the database. This includes decisions regarding the database schema, data types, and the relationships between different data elements.
- **Facilitating Changes to Database Structure**: As the needs of the organization evolve, the DBA is responsible for making necessary adjustments to the database structure. This could involve adding new tables, modifying existing ones, or reorganizing the database to better suit new requirements.
- **Seeking Community-Wide Solutions**: The DBA looks for solutions that are beneficial for the entire user community of the database. This involves considering the impact of any changes or enhancements on all users and stakeholders.

- **Assessing Impact on All Users**: Before implementing any changes, the DBA assesses how these changes will affect all users of the database. This ensures that any modifications are made with a comprehensive understanding of their implications.

5)

The difference between an explicit and an implicit lock, is as follows:

- **Implicit Locks**: These are locks placed by the Database Management System (DBMS) itself. Implicit locks are automatically managed by the DBMS without the need for explicit instruction from the application program.
- **Explicit Locks**: These locks are issued by the application program. In this case, the program specifies when to lock a resource, such as a database record, and when to release it. This requires direct commands within the application code to manage the locks.

In essence, implicit locks are managed internally by the DBMS, whereas explicit locks require direct control and instructions from the application utilizing the database.