



**MARMARA UNIVERSITY**

**FACULTY OF ENGINEERING  
COMPUTER SCIENCE & ENGINEERING  
DEPARTMENT**

**CSE3055  
DATABASE SYSTEMS  
Homework #9**

*A. Tunahan Cinsoy  
150117062*

1)

- a) Backward Recovery
- b) Forward Recovery (Using Backup Copy of Database)
- c) Forward Recovery (From Last Checkpoint)
- d) Reprocessing Transaction
- e) Forward Recovery (From Latest Checkpoint)

2) In pessimistic concurrency, we assume that concurrency conflicts will happen and we believe that they will happen often. So, from the perspective of efficiency, optimistic concurrency is much more efficient than pessimistic concurrency.

3) An exclusive lock gives a process exclusive access for writing to the specified part of the file. Whereas, shared lock prohibits any other process from requesting a write lock on the specified part of the file.

4) Deadlock Prevention ensures that at least one of the necessary conditions to cause a deadlock will never occur. However, Deadlock Resolution allows deadlocks to occur, in addition, it has some solutions for solving deadlocks.

5)

- Backup Facilities: Facility that produces a backup copy of the entire database.
- Journalizing Facilities: Facility that produces an audit trail of transactions and database changes.
- Checkpoint Facility: Facility which the DBMS periodically refuses to accept any new transaction.
- Recovery Manager: Module that restores the database to a correct condition when a failure occurs and resumes processing user requests.

6)

When you attempt to commit a transaction, transaction integrity ensures that if part of the transaction fails, none of the transaction will be written to the data source. With the help of that, integrity is satisfied among transactions.

7)

- System Crash: Usually refers to any kind of bugs or hardware malfunction in the OS or the DB Software.
- Statement Failure: Refers to the failures at SQL Queries.
- Media Failure: Some disk head crash, disk controller failure might cause all databases residing on that disk to be lost.
- Application Software Errors: It includes logical errors in the program that is accessing the database, which causes one or more transaction to fail.

8)

- Infrastructure Failure: The load placed on infrastructure to become so high so that it can no longer handle it. Best way to prevent it to build scalability into infrastructure.
- Data Inconsistency: Data that is not available in the format that you need to work with it, and cannot be transformed quickly enough to that format, poses a problem for high availability. Best solution is to ensure that having a flexible, automated data transformation tools at disposal.
- Malicious Activity: Bad people who want to disrupt infrastructure's high availability. Using anti DDoS routers might be a good way to mitigate the risk of attacks.
- Poor Data Quality: When the sets are filled with inconsistencies, redundancies, inaccuracies or other issues, it prevents using data effectively, and undercut high availability. It can be controlled by leveraging data quality tools to clean up data sets and building them into data management processes.

9)

- Atomicity: All the instructions within a transaction must be successfully executed or none of them must be executed.
- Consistency: Database should remain consistent after every transaction.
- Isolation: If the multiple transactions are running concurrently, they shouldn't be affected by each other.
- Durability: Changes that have been committed to the database should remain even in the case of software and hardware failure.