**Jennifer Faer**

**IS 5313 Structured Data and Querying**

**Dr. Bill Rosener**

**Fall 2024**

**https://billrosener.com/teaching/is4293/assignments/assignment6/assignment6-description.html**

**Chapter 6 Review Questions**

1. What is the purpose of database administration?

The function of database administration is to ensure the efficient, secure, and reliable management of data within an organization. This includes concurrency control, backup / recovery, security, and performance. (Pages 415-416)

1. Explain how database administration tasks vary with the size and complexity of the database.

For personal databases, the tasks tend to be simpler (such as backups). For multi-user databases, one or more database administrators work on a single database to handle all the tasks needed for maintenance and updating. (page 416)

1. What are two interpretations of the abbreviation DBA?

Database Administration or Database Administrator, which can be the office (of database administration) or a manager (of the database). (page 416)

1. What is the purpose of concurrency control? What is throughput, and how is concurrency control related to throughput?

Concurrency control helps to ensure one user’s work does not influence another user’s work. (At past jobs, we called this “clashing”!) Throughput is the number of transactions the DBMS can process at the same time, and the methods of concurrency control need to be adequate for the throughput. (page 429)

1. What is the goal of a database security system?

Database security systems help to ensure that “only authorized users can perform authorized activities at authorized times.” (page 436)

1. Explain the meaning of the wordinappropriately in the phrase “one user’s work does not inappropriately influence another user’s work.”

“Inappropriately” could be described as “unintentionally” or “erroneously.” If one user is updating the same record as another user, the DMBS will not be able to simultaneously update the record. (Page 480)

1. Explain the major trade-off that exists in concurrency control.

“Tradeoffs need to be made between the level of protection and data throughput.” (page 480)

1. Describe what an atomic transaction is, and explain why atomicity is important.

“An ACID transaction is one that is atomic, consistent, isolated, and durable. Durable means that database changes are permanent. Consistency can refer to either statement-level or transaction-level consistency. With transaction-level consistency, a transaction may not see its own changes.” (page 481)

1. Explain the difference between concurrent transactions and simultaneous transactions.

Concurrent transactions are two database changes that happen very close together; two users may update the same record at the same moment in time, but they are processed individually. Simultaneous transactions do not exist because computers can only conduct one task at a time; however, if the concurrent transactions are processed quickly enough, they appear to be simultaneous to the user. (page 419)

1. Give an example, other than the one in this text, of the lost update problem.

An example from my current role as operations manager for an RV repair company: Appointments can be scheduled by me as the manager, and by clients themselves. If I schedule an appointment in the same time a client is scheduling one, one of the appointments (probably the client-scheduled appt) will appear to be scheduled on the client’s end, but not show up on the calendar.

1. Define the terms dirty read, nonrepeatable read, and phantom read.

Dirty read: when a transaction reads a changed record in which the change has not been committed to the database.

Nonrepeatable read: when a transaction rereads data and finds changes from another transaction.

Phanton read: when a transaction rereads data and finds new rows inserted by a different transaction.

(page 480)

1. Explain the difference between an explicit lock and an implicit lock.

Locks allocate resources to specific transactions. Explicit locks are implemented by the application interacting with the DBMS and implicit locks are implemented by the DBMS itself. (page 482)

1. What is lock granularity?

Lock granularity is the size of the object accessed with the lock. (page 482)

1. Explain the difference between an exclusive lock and a shared lock.

Exclusive locks prevent all other transactions from reading or updating. Shared locks prevent the updating of the data but many transactions can concurrently read the data. (page 482)

1. Explain two-phase locking.

Two-phase locking is a two-phase process for ensuring consistency of db updates in a concurrent-processing environment. The growing phase is when locks are obtained, and the shrinking phase are when locks are released. (page 482)

1. How does releasing all locks at the end of a transaction relate to two-phase locking?

Releasing locks preserves the integrity of the two-phase lock, so that each phase is unique and separate from the other.

1. What is deadlock? How can it be avoided? How can it be resolved when it occurs?

Deadlock happens when two users are waiting for a resource the other has locked. It can be avoided by only allowing users to issue one lock request a time or to require all transactions to lock resources in the same order. Breaking deadlock can involve rolling back one of the transactions to remove the changes it made and starting that transaction over. (page 424)

1. Explain the difference between optimistic and pessimistic locking.

Optimistic locking assumes no conflicts will occur. Conflict checks happen after the transaction is processed, and if a conflict is detected, the transaction must be repeated.

Pessimistic locking assumes conflicts will occur, and locks are issued before the transaction is processed, then released after. (page 424)

1. Explain the benefits of marking transaction boundaries, declaring lock characteristics, and letting a DBMS place locks.

Maintaining concurrency control, consistency and integrity, minimizing conflicts, reducing complexity for developers.

1. Explain the use of the SQL transaction control language (TCL) statements BEGIN TRANSACTION, COMMIT TRANSACTION, and ROLLBACK TRANSACTION.

BEGIN TRANSACTION: marking the start of a new transaction, and the statements should be treated as a single block of work.

COMMIT TRANSACTION: permanently saves the transaction to the db.

ROLLBACK TRANSACTION: reverts the changes back to the state before the transactions began.

(page 425)

1. Explain the meaning of the expression ACID transaction.

ACID is an acronym for atomic, consistent, isolated, and durable.

Atomic: the transaction is “all or nothing” (either all of the transactions occur, or none do)

Consistent: additional transactions are not allowed while other transactions are active.

Isolated: a transaction is protected from other transactions to avoid dirty reads, nonrepeatable reads, and phantom reads.

Durable: the changes are permanent.

(page 428)

1. Describe statement-level consistency.

Where all rows affected by a SQL statement are protected from changes made by other users during the statement’s execution. (page 427)

1. Describe transaction-level consistency. What disadvantage can exist with it?

Where all rows affected by the SQL statement are protected from changes during the entire transaction. Transaction-level consistency will probably reduce throughput, and may prevent a transaction from seeing its own changes. (page 428)

1. Explain what read uncommitted isolation level is. Give an example of its use.

Read uncommitted isolation level is the isolation level that allows all three data read problems: dirty reads, nonrepeatable reads, and phantom reads. (page 428)

Example: Alice updates a product price, and Bob reads this updated price before Alice commits. If Alice rolls back, Bob has acted on incorrect data.

1. Define the terms rollback and rollforward.

Rollback: Essentially restoring from backup, recovering the db from an earlier image to bring it to a point where the db is logically consistent.

Rollforward: process of recovering a database by applying after-images to a saved copy of the db to bring it to a checkpoint or other point where the db is logically consistent.

(page 482)