CSC 355 Database Systems
Lecture 9

Eric J. Schwabe

School of Computing, DePaul University

Spring 2020

Today:

- Transactions

Problem: Interruptions

- Two SQL statements are written to transfer money from one bank account to another…
- …one executes, then the server crashes
  - What happens to the money?

Problem: Concurrency

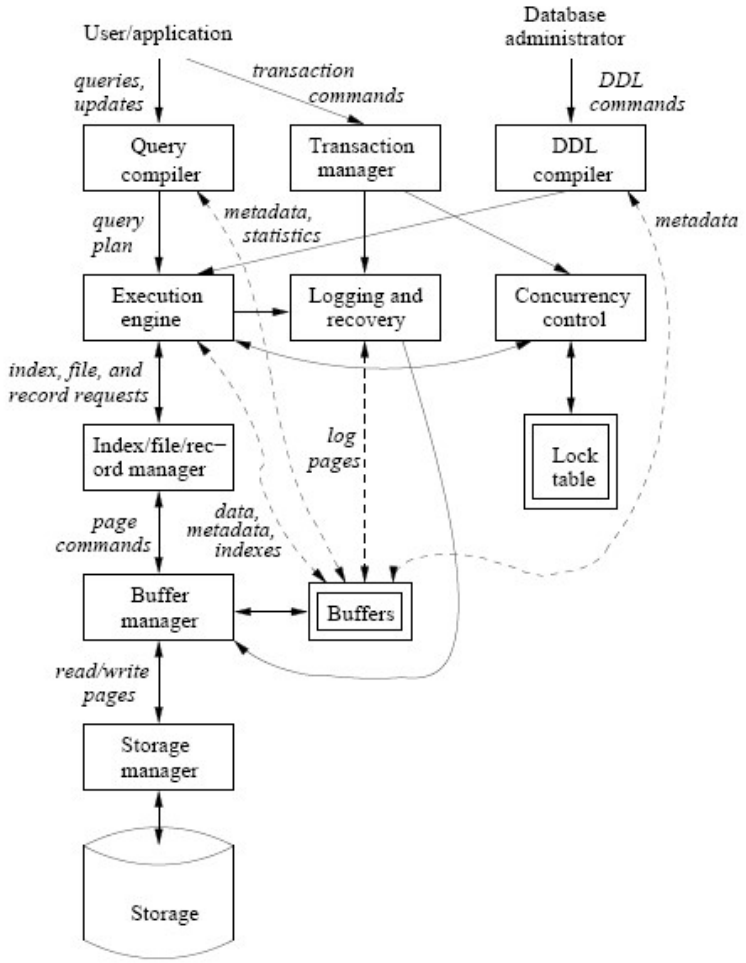
- Two users check the balance of the same bank account…
- …then both try to transfer money out of it
  - Who gets it?

Solution: Transactions

- A *transaction* is a collection of SQL statements that must be executed as a unit
- The Transaction manager in the DBMS must handle
  - Interruptions: Logging and recovery, Buffers
  - Concurrency: Concurrency control, Lock tables

Components of a DBMS

(From Ullman/Widom)



Transactions in Oracle

- Any operation that changes the database state starts an implicit transaction in Oracle
- Can also start a transaction explicitly with a SET TRANSACTION statement
- End a transaction with a COMMIT statement
  - Transaction can also be ended with ROLLBACK, usually done by system rather than user…

ACID Properties

- A transaction should satisfy the following properties:
  - Atomicity: Executes completely or not at all
  - Consistency: Satisfies all database constraints
  - Isolation: Executes “separately” from others
  - Durability: Once executed, results are permanent

Atomicity

- Transaction operations are kept in a local store, not applied to the database immediately
  - Transaction can see its own changes, others can't
- When a transaction is completed, COMMIT applies changes to the shared database in their entirety ("executes completely…")
- ROLLBACK during a transaction undoes any partial results ("…or not at all")

Durability

- After COMMIT, the changes applied to the shared database are permanent, and cannot be rolled back later

Consistency

- Constraints can be “deferred”, so that they are only checked when a transaction commits, not for each individual statement
  - Add DEFERRABLE INITIALLY DEFERRED to the constraint definition
  - If a constraint is violated when you COMMIT, a ROLLBACK of the entire transaction is done!

Isolation

- DBMS maintains “separation” among transactions that access data concurrently
  - Various different levels of isolation
    - Transactions might modify, or just read, data
  - Tradeoff between performance and data integrity

Serializable Isolation

- Transactions must behave as though they were run serially (first one, then the other)
- Usually implemented by “locking” the tables (or parts of tables) used by a transaction
  - Other transactions using the same tables will have to wait until they are released
  - Other transactions using other tables could run at the same time

Read Committed Isolation

- Transaction operations can be interleaved
- If one transaction tries to read data that were written by another, it can only see the changes that have been committed
  - Can’t be rolled back, but could be changed later
  - Multiple queries of the same table might not yield the same results... “non-repeatable reads”, including “phantoms”…

Read Uncommitted Isolation

- Transaction operations can be interleaved
- If one transaction tries to read data that were written by another, it can see all changes, even if they have not been committed
  - Could be rolled back by the other transaction!
  - Transaction might make decisions based on values that are later rolled back and so were never really there … “dirty reads”…

Isolation Levels

- **SERIALIZABLE:** Transactions must appear to run serially – cannot read any changes from others
- **[REPEATABLE READ:** Can read committed changes that only add data (allows only phantoms)]
- **READ COMMITTED:** Can read all committed changes (allows all non-repeatable reads)
- **READ UNCOMMITTED:** Can read all changes (allows all non-repeatable reads and dirty reads)

Transactions in Oracle

- SET TRANSACTION statement can specifiy ISOLATION LEVEL
  - READ COMMITTED (default)
  - SERIALIZABLE
  - Oracle does not support REPEATABLE READ and READ UNCOMMITTED
- COMMIT or ROLLBACK ends transaction

Next:

- Finish Transactions
- Review for Midterm Exam
- Introduction to Relational Database Design