Transaction

Wednesday, May 29, 2019 1:30 PM

- Data Management Pipeline

  conceptual schema → physical schema

  schema designer — database administrator

  application programmer

- Applications that talk to DB must use transaction to make data consistent.

  want multiple operations to be executed atomically over the same DBMS.

- Write - Read Conflict (dirty/inconsistent read)

  Read - Write Conflict (unreapeatable read)

  Write - Write Conflict (lost update)

  - Changes to the database should be ALL or NOTHING

- Transaction: Collection of statements that are executed automatically (logically)

  BEGIN TRANSACTION

  [SQL statement]

  COMMIT or

ROLLBACK (= ABORT)

[SQL statement] if BEGIN... missing

- ACID:
  - Atomic handle the failure. - recovery manager
    - State shows either all the effects of txn or non of them.
    - A transaction is atomic if all its updates must happen or not at all.
  - Consistent
    - TXN moves from a DBMS state where integrity holds, to another where integrity holds.
  - Isolated
    - Effect of txns is same as txn running one after another (orderly)
    - A execution ensures that transactions that are isolated, if the effect of each transaction is as if it were the only transaction running on the system.
  - Durable (More in CSE 444)
    → by writing to disk
    - A transaction is durable if its effect continue to exists after the transaction or after the program has terminated.
- If the app gets to a state where it cannot complete the transaction, execute ROLLBACK.
  - DB returns to the state prior to the transaction.
- By default every one query is its own transaction.
- To implement transaction, need to address two problems:
  - "I" Isolation

- "I" Isolation - Concurrency control.
- "A" Atomicity - Recover from trash.