CSC 350

TRANSACTIONS

SCENARIO

1. Sue goes to Best Buy to purchase a VR headset.
2. She swipes her debit card.
    A. The banking db checks her balance.
    B. The banking db deducts the amount from her account.
    C. The banking db credits the money to Best Buy's account.
    D. The banking db sends the OK back to Best Buy.
3. Best Buy says "Card Approved".
4. Sue goes home happy.

SCENARIO

1. Sue goes to Best Buy to purchase a VR headset.

2. She swipes her debit card.

System Crash! A. The banking db checks her balance.

B. The banking db deducts the amount from her account.

C. The banking db credits the money to Best Buy's account.

D. The banking db sends the OK back to Best Buy.

3. Best Buy says "Card Approved".

4. Sue goes home happy.

What changes?

SCENARIO

1. Sue goes to Best Buy to purchase a VR headset.

2. She swipes her debit card.

   A. The banking db checks her balance.

System Crash! B. The banking db deducts the amount from her account.

   C. The banking db credits the money to Best Buy's account.

   D. The banking db sends the OK back to Best Buy.

3. Best Buy says "Card Approved".

4. Sue goes home happy.

What changes?

SCENARIO

1. Sue goes to Best Buy to purchase a VR headset.
2. She swipes her debit card.
    A. The banking db checks her balance.
    B. The banking db deducts the amount from her account.

System Crash!

    C. The banking db credits the money to Best Buy's account.
    D. The banking db sends the OK back to Best Buy.

3. Best Buy says "Card Approved".
4. Sue goes home happy.

What changes?

SCENARIO

1. Sue goes to Best Buy to purchase a VR headset.
2. She swipes her debit card.
    A. The banking db checks her balance.
    B. The banking db deducts the amount from her account.
    C. The banking db credits the money to Best Buy's account.

System Crash!

    D. The banking db sends the OK back to Best Buy.

3. Best Buy says "Card Approved".
4. Sue goes home happy.

What changes?

SCENARIO

1. Sue goes to Best Buy to purchase a VR headset.

2. She swipes her debit card.

Transaction

A. The banking db checks her balance.

B. The banking db deducts the amount from her account.

C. The banking db credits the money to Best Buy's account.

D. The banking db sends the OK back to Best Buy.

3. Best Buy says "Card Approved".

4. Sue goes home happy.

TRANSACTION REQUIREMENTS

- Atomicity: Either all operations of the transaction are reflected properly in the database, or none are
- Consistency: Execution of a transaction in isolation (that is, with no other transaction executing concurrently) preserves the consistency of the data- base.
- Isolation: Even though multiple transactions may execute concurrently, the system guarantees that, for every pair of transactions X and Y , it appears to X that either Y finished execution before X started or Y started execution after X finished. Thus, each transaction is unaware of other transactions executing concurrently in the system.
- Durability: After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

ACID Properties

Consistency

This must hold true before and after the transaction completes.

Atomicity

BEST BUY ACCOUNT (BA)

SUE ACCOUNT (SA)

Recovery Log

SA - $200 | BA + $200

Stored on disk and used to undo (AKA “roll back”) the transaction if something goes wrong.

Durability

BEST BUY ACCOUNT (BA)

SUE ACCOUNT (SA)

Recovery Log

SA - $200 | BA + $200

Stored on disk and used to make sure the transaction completes (AKA “recover the transaction”) if something goes wrong.

Isolation

What if the system credits BA first, then deducts from SA?

What if, in between those, Sue's husband uses her card to buy something online?

How can we make sure that the "check balance" step of the second purchase knows we have spent $200, without allowing the transactions to overlap?

Concurrency Control! (Tune in next week…)

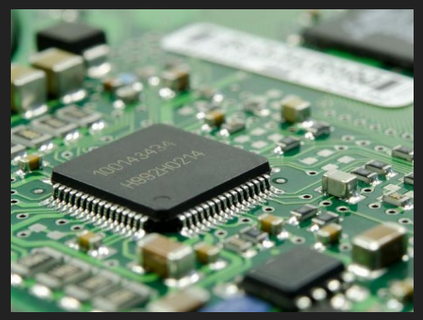
TRANSACTION MANAGEMENT

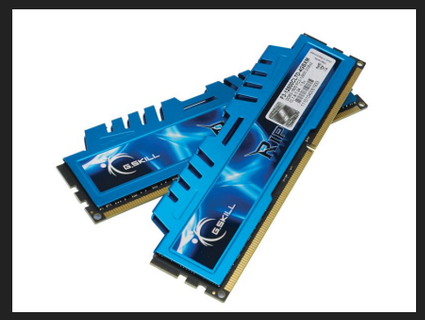
1. Transaction Begins
2. Write all steps of transactions to recovery log
3. Carry out each step of transaction
4. Signal transaction is complete

Where to store this?

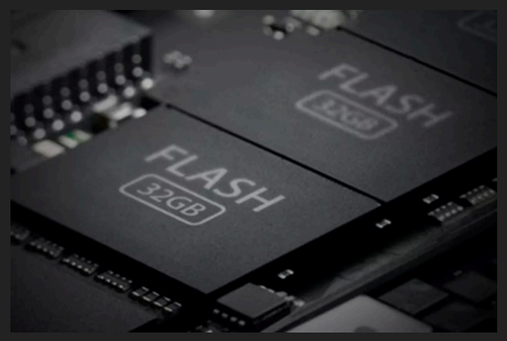
STORAGE LEVELS

Primary Storage: Fast, but volatile (cache and RAM)

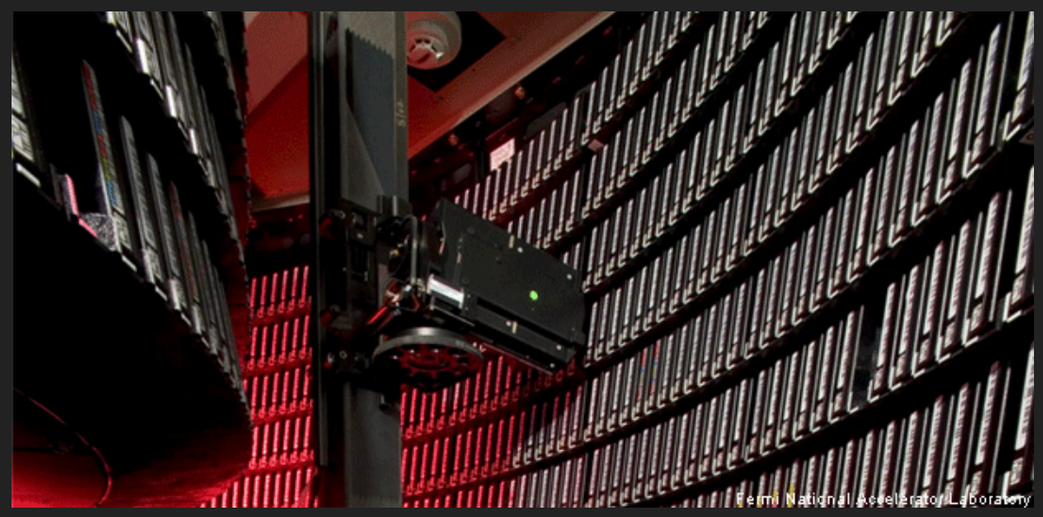




Secondary Storage: Moderately-Fast, non-volatile
(AKA “online storage”)



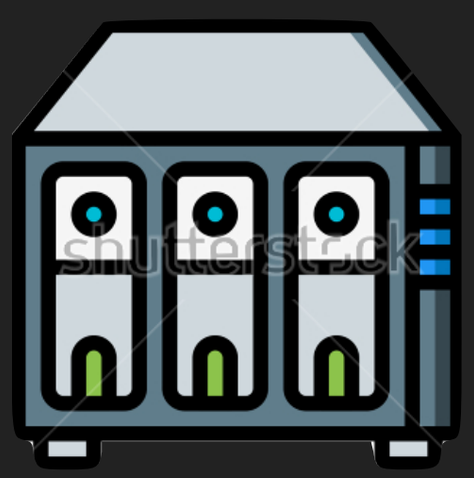
Tertiary Storage: Slow, non-volatile
(AKA “offline storage”)



STABLE STORAGE

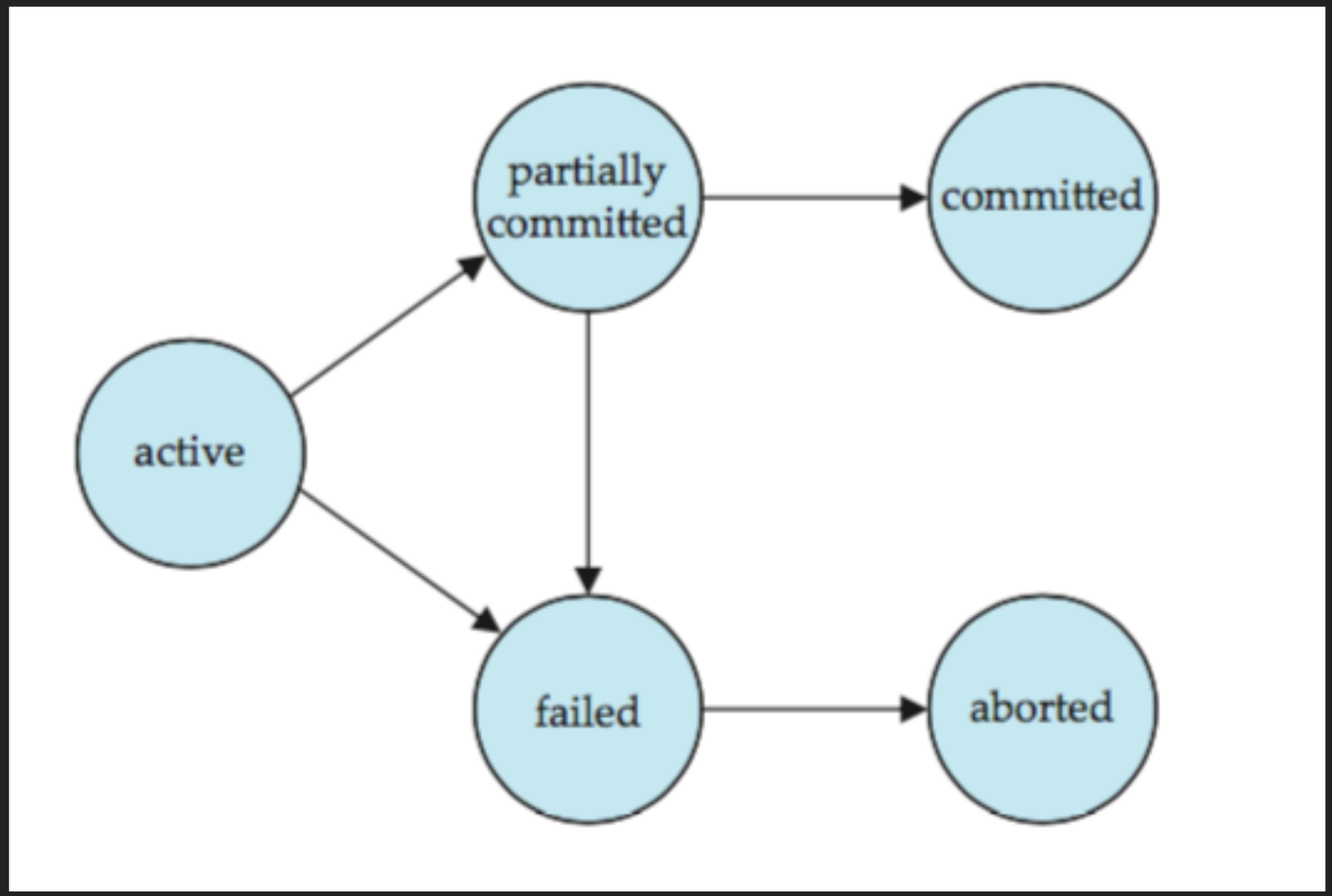
Stable storage is “never” lost

Normal Cases



Sensitive Data Cases

TRANSACTION STATES



OBSERVABLE EXTERNAL WRITES

- Some transactions result in persistent data being created *outside* of the database (sending a signal to a credit card terminal, sending an email receipt, generating a shipping manifest, etc…)
- These are called Observable External Writes, because the data is created (written) outside of the database (external) and can be seen by users (observable).
- The best safeguard is to only perform such actions after the transaction has completed successfully.