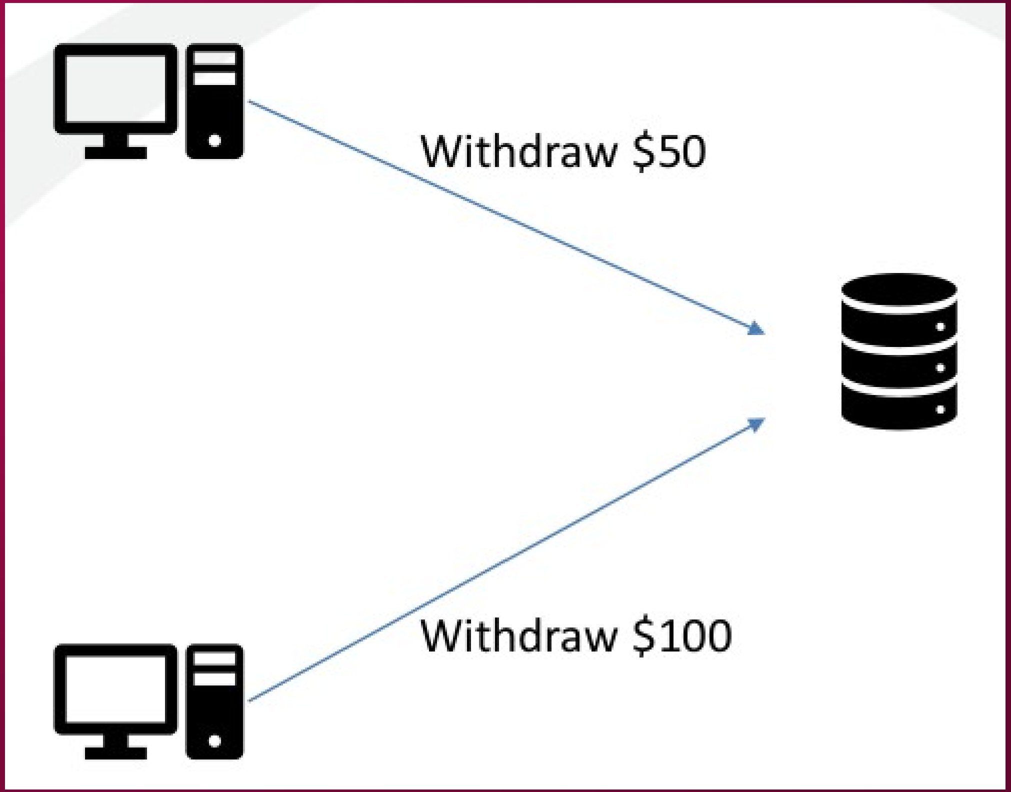
Transactions

UNIVERSITY OF MONTANA

Concurrency

- Two users manipulate a database simultaneously
- What happens?

YouTube: ACID Compliant

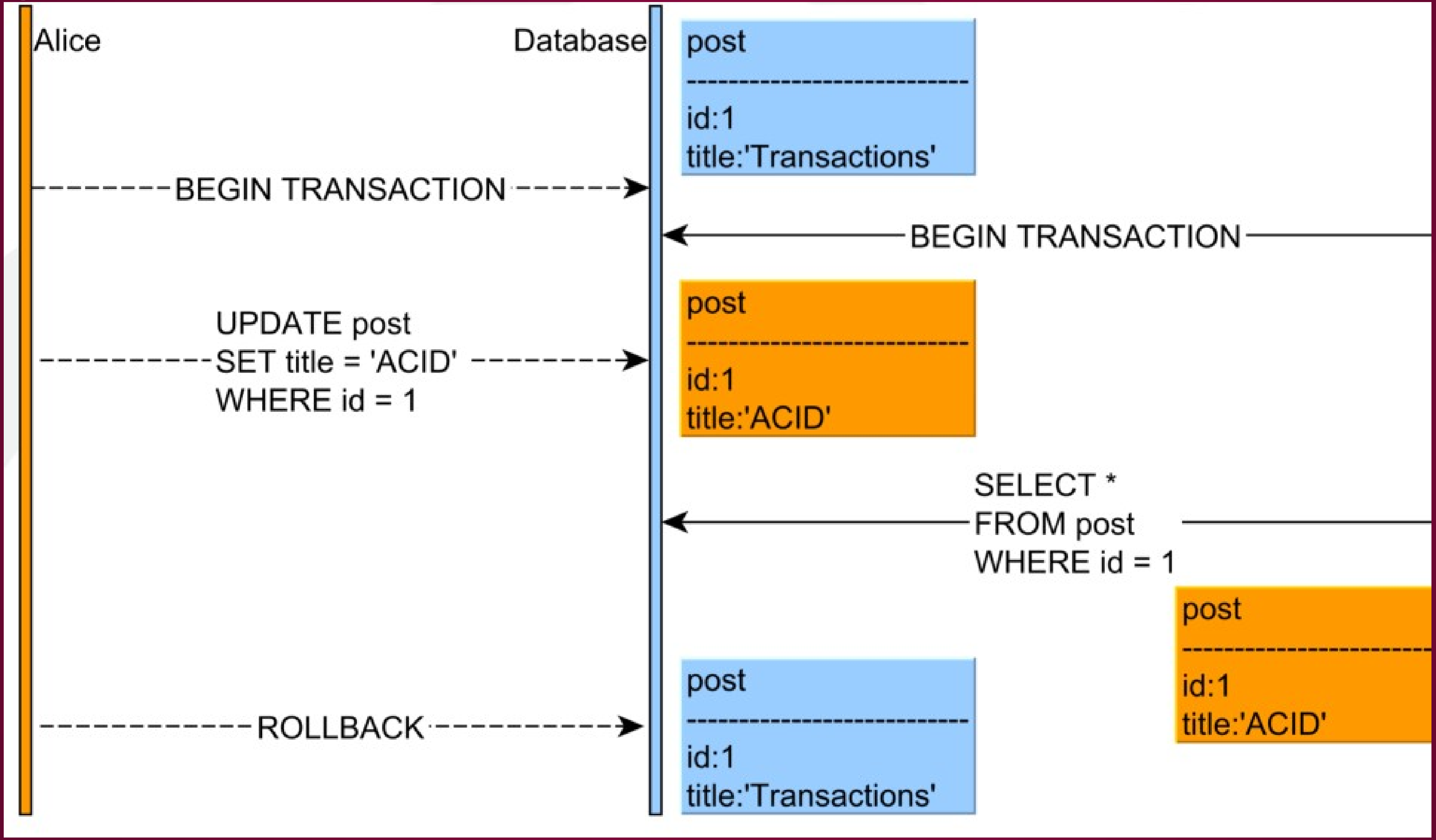


ACID

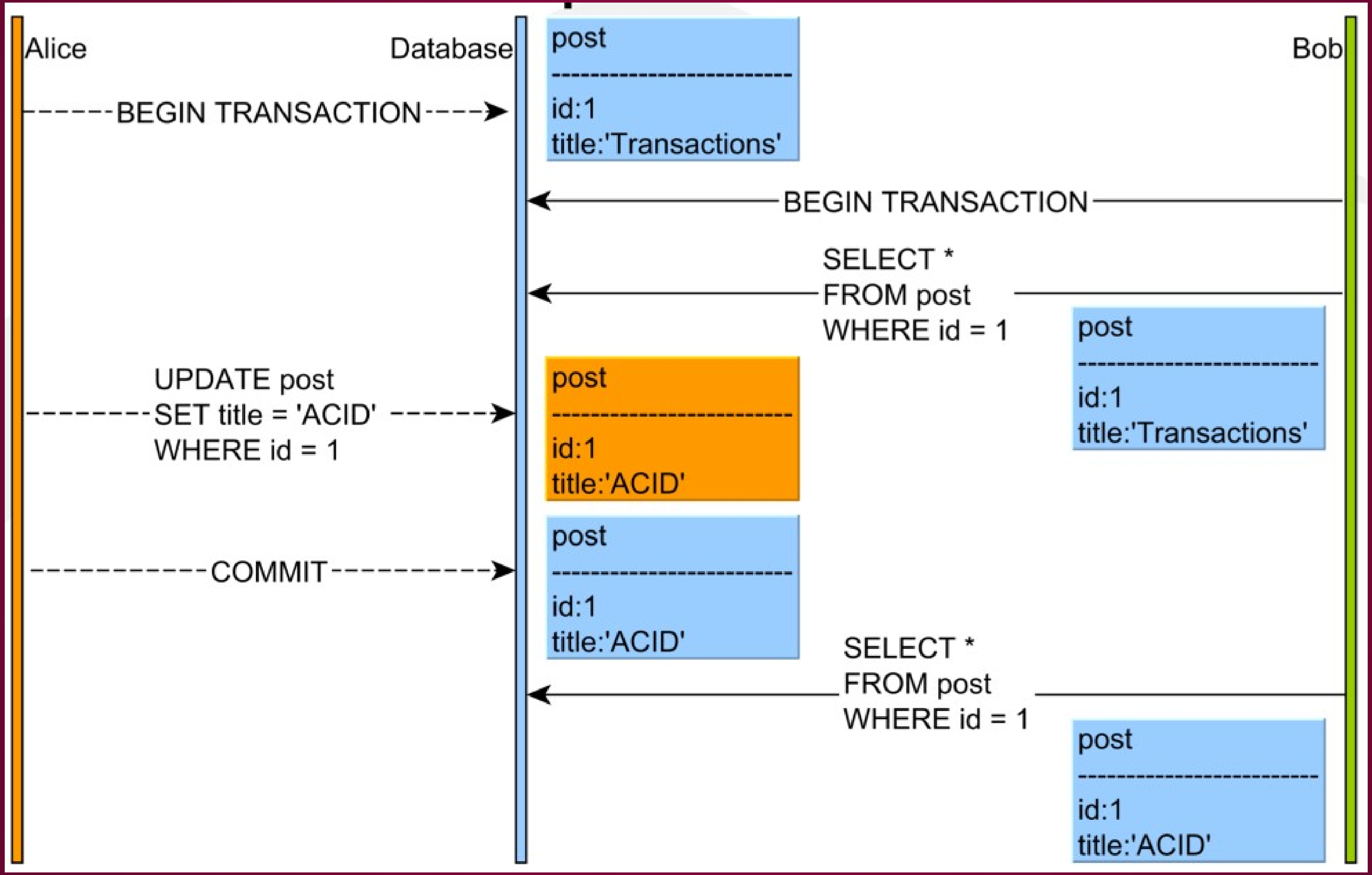
- Atomicity
  - All or Nothing Transactions
- Consistent
  - Rules are not violated (Business rules, etc)
- Isolated
  - Transactions are Independent
- Durable
  - Committed Data is Never Lost

UNIVERSITY OF
MONTANA

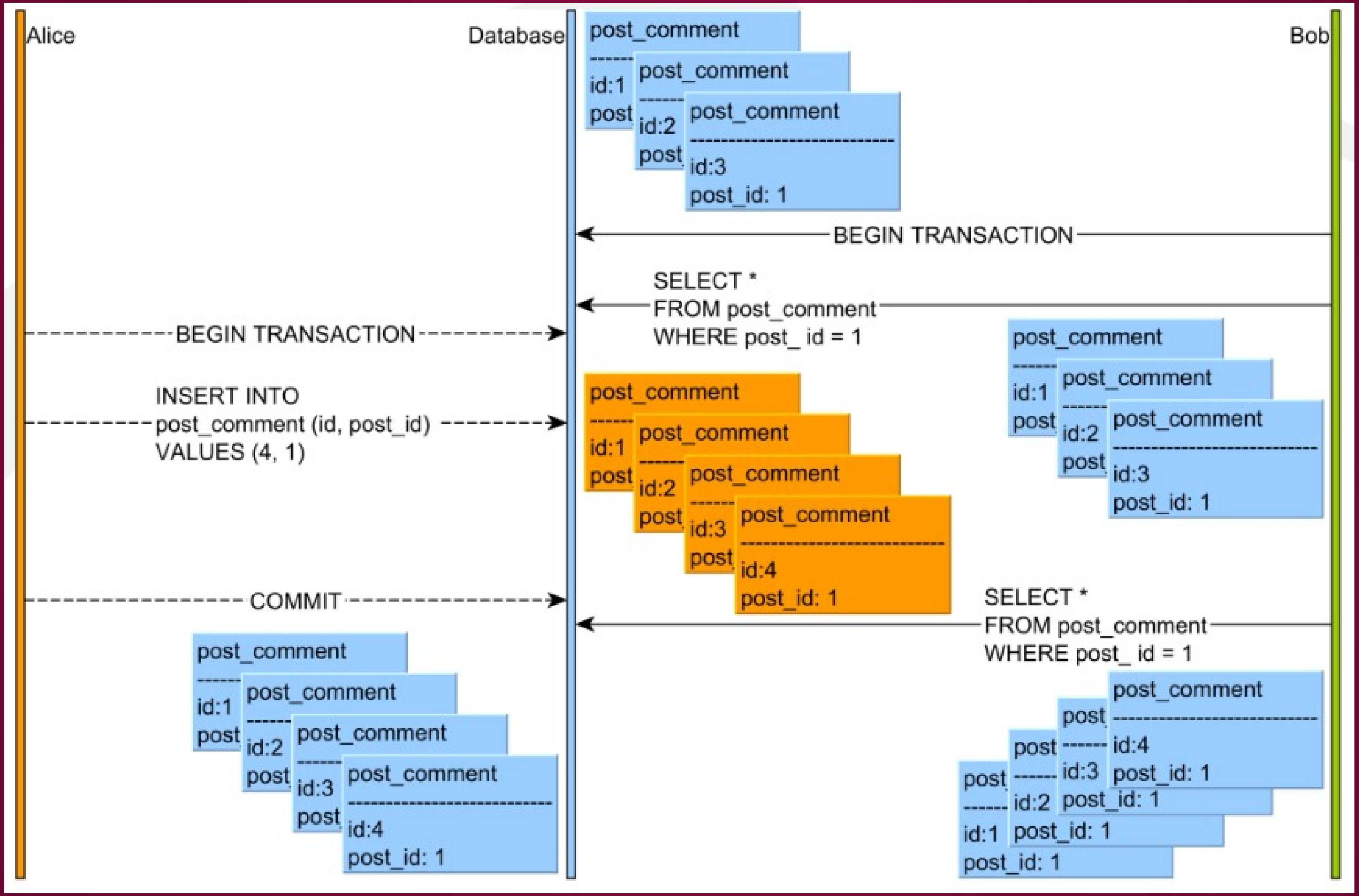
Dirty Read


Alice
Database
post
id:1
title:'Transactions'
BEGIN TRANSACTION
BEGIN TRANSACTION
UPDATE post
SET title = 'ACID'
WHERE id = 1
post
id:1
title:'ACID'
SELECT *
FROM post
WHERE id = 1
post
id:1
title:'ACID'
ROLLBACK
post
id:1
title:'Transactions'

Non-Repeatable Read



Phantom Read



Schedules

- Sequential order of operations for multiple transactions
- Operations can be interleaved in some instances
  - What instances cause no conflict?
- Equivalent schedules
  - Will give same result
- Conflicting schedules
  - Can cause differing results

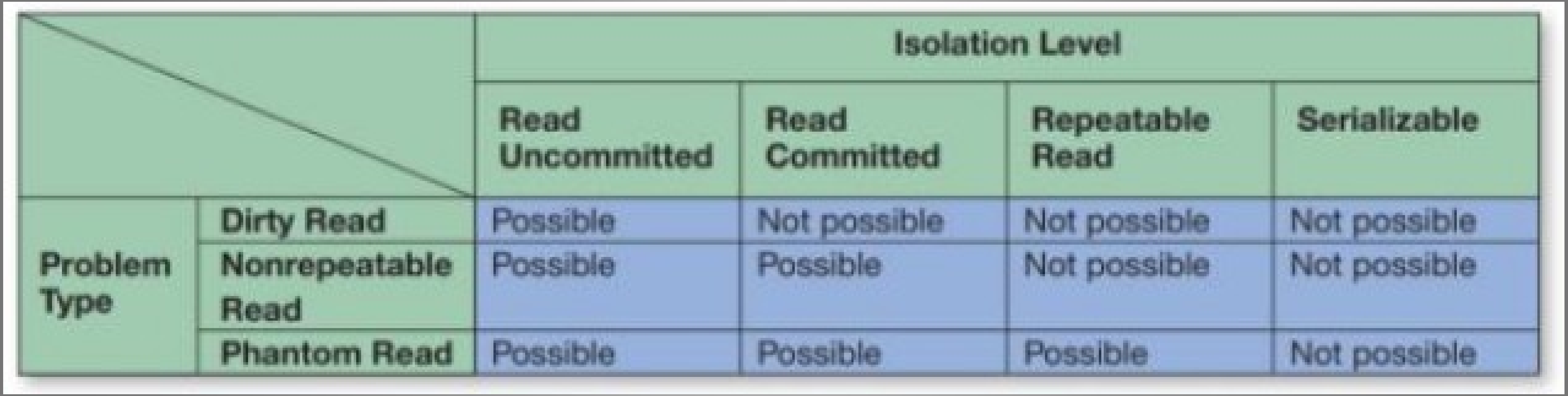
The gold standard

- Serial Schedule
  - Transactions are executed one at a time.
- Serializable Schedule
  - Equivalent to a serial schedule

Isolation Levels

- SERIALIZABLE
  - Transactions are run in a serializable schedule with concurrent transactions
  - Isolation is guaranteed.
- REPEATABLE READ
  - Transactions read only committed data
  - After the transaction reads data, other transactions cannot update the data
  - Prevents most types of isolation violations but allows phantom reads.
- READ COMMITTED
  - Transactions read only committed data
  - After the transaction reads data, other transactions can update the data
  - Allows nonrepeatable and phantom reads.
- READ UNCOMMITTED
  - Transactions read uncommitted data
  - Processes concurrent transactions efficiently but allows a broad range of isolation violations, including dirty, nonrepeatable, and phantom reads.

Isolation Levels



		Isolation Level			
		Read Uncommitted	Read Committed	Repeatable Read	Serializable
Problem Type	Dirty Read	Possible	Not possible	Not possible	Not possible
	Nonrepeatable Read	Possible	Possible	Not possible	Not possible
	Phantom Read	Possible	Possible	Possible	Not possible

Schedules and recovery

- Nonrecoverable schedule
  - One or more transactions cannot be rolled back
- Cascading schedule
  - Rollback of one transaction forces rollback of other transactions
- Strict schedule
  - Rollback of one transaction never forces the rollback of other transactions

UNIVERSITY OF MONTANA

Locking

- Shared Lock
  - Allows for a transaction to read, but not write data
  - There can be multiple shared locks
- Exclusive Lock
  - Allows for the transaction to read and write
  - If there is an exclusive lock, there can be no other locks

UNIVERSITY OF MONTANA

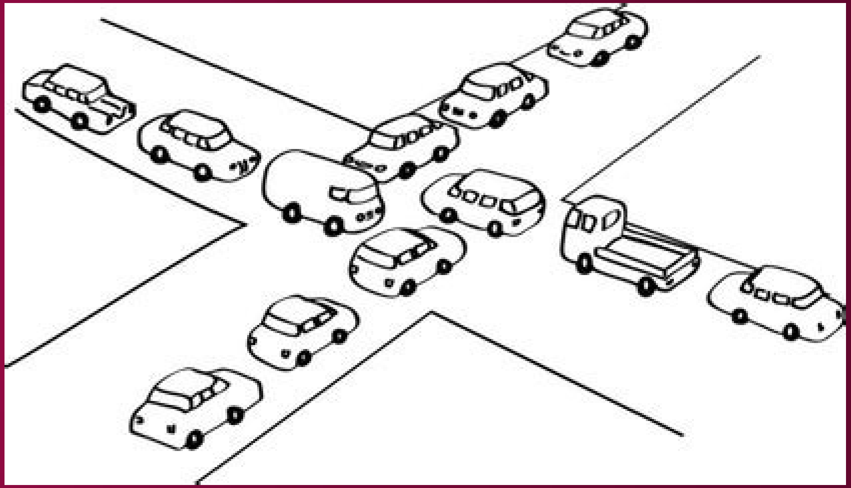
Two-phase locking

- Expand phase and contract phases
- Basic two-phase locking
  - The transaction can grow or shrink its scope as needed
- Strict two-phase locking
  - Exclusive locks are held until commit/rollback
  - Scope can be expanded as necessary
- Rigorous two-phase locking
  - Both shared and exclusive locks are held until commit/rollback
  - In effect, it has no contract phase

UNIVERSITY OF MONTANA

Problems with locking

- Dependent transaction
  - Waiting for data locked by another transaction
- A cycle of dependent transactions indicates a deadlock
- A deadlock is a state in which a group of transactions cannot be resolved



UNIVERSITY OF MONTANA

Avoiding Deadlock

- Aggressive Locking
  - Transactions request locks when the transaction starts
- Data ordering
  - For example, rearrange locks so that locks on X occur before locks on Y
- Timeout
  - If a transaction takes too long, it rolls back
- Cycle detection
  - When a cycle is detected, the ‘cheapest’ transaction is rolled back

UNIVERSITY OF MONTANA

Other options for concurrency

- Snapshot isolation
  - A snapshot is taken when the transaction starts
  - Updates are applied to the snapshot
  - Prior to commit, check for conflicts
  - If no conflict is detected, write the snapshot
  - If a conflict is detected, roll back

UNIVERSITY OF MONTANA

Transactions in SQL

SET [GLOBAL | SESSION] TRANSACTION
ISOLATION LEVEL [SERIALIZABLE | REPEATABLE READ | READ
COMMITTED | READ UNCOMMITTED];

- SESSION
  - Until the user or program disconnects
- GLOBAL
  - All transactions during this and later sessions. Can only be used by the administrator
- If neither is specified, defaults to only the next transaction in the session

UNIVERSITY OF MONTANA

Defining a transaction

- Start
  - START TRANSACTION
- End
  - COMMIT
  - ROLLBACK
- Autocommit can be set with: SET autocommmit = [OFF | ON];
- Optional keywords for COMMIT and ROLLBACK
  - AND CHAIN – override autocommit setting
  - RELEASE – end current session

UNIVERSITY OF MONTANA

Savepoints

- Savepoint – partial transaction results are saved temporarily

  SAVEPOINT identifier;

  ROLLBACK TO identifier;

  RELEASE SAVEPOINT identifier;

UNIVERSITY OF
MONTANA