Anomalies

Two operations are said to be in conflict if they are on the same object and one of them is a write

Can have WR
RW - conflicts
WW

Ex] WR-conflicts

Ex

T1	T2
R(A)	
W(A)	
	R(A)
	W(A)
	R(B)
	W(B)
	commit
R(B)	
W(B)	
commit	

→ value of A read is affected by T1 but not read of B (dirty read)

Reading uncommitted data

Ex) RW-conflicted

Suppose T1 reads A then T2 modified A and T1 read A again. Gets different value Called unrepeatable read.

Ex WW-conflict (overwrite uncommitted data)

blind write - write w/o reading data.

T1	T2
1000 W(A)	
	W(B) 1000
2000 W(B)	
	W(A) 2000

not equivalent to serial schedule (lost an update)

Schedules w/ aborted transaction

Remember:

A serializable schedule is a schedule whose effect on any consistent DB is identical to some complete serial schedule over the set of committed transactions.

i.e., the effects of aborted transactions must be undone and must not have an effect on DB

Ex

T1	T2
R(A)	
W(A)	R(A)
	W(A)
	commit
Abort	

this read depends on write

schedule called unrecoverable

Defn A recoverable schedule is one in w/ transactions commit only after all transactions whose changes they read commit.

Defn An abort is said to be cascading if it forces some other transaction to abort in order to return DB to a consistent state.

Locks

A lock is a bookkeeping device we associate w/ a DB object.

In order to access the object in some way (Ex read, write), a transaction is required to first obtain the lock on the object.

Ex] Usually have read & write locks. To get a write lock on O no one else is allowed to have a lock on O. To get a read lock noone is allowed to have a write lock

A locking protocol is a set of rules each transaction promises to follow concerning obtaining locks.

Ex] Strict two phase locking (strict 2PL)

(1) If T want to read O it must get a read lock (shared lock) on O first
To write O a write lock (exclusive lock) must be obtained 1st.

(2) Locks are held till the transaction completes when they are all released.

Strict 2PL guarantee serializability and avoids anomalies have discussed before

Deadlocks

Problem

T1 gets write lock on A
T2 get write lock on B
both use strict 2PL
T1 then request write lock on B
T2 on A.
Neither transaction can proceed.
(Deadlock)

Can use timeout mechanism to identify if in deadlock.
Will discuss latter what to do in deadlock situations. Usually one of transactions must abort.

SQL & Transactions

Transaction said to begin when first SQL statement Select, Insert, update, create applied to DB.

Ends w/ COMMIT command
or ROLLBACK command.

What to do about really long transactions?
Can define Savepoints

EX) SAVEPOINT MY_SAVE-PT;
ROLLBACK TO SAVEPOINT MY_SAVE_PT;