TRANSACCIONS I CONCURRÈNCIA

Conjunt operacions (Lectura/Actualització) que s'interpreten com una sola.
Han de complir propietat ACID.

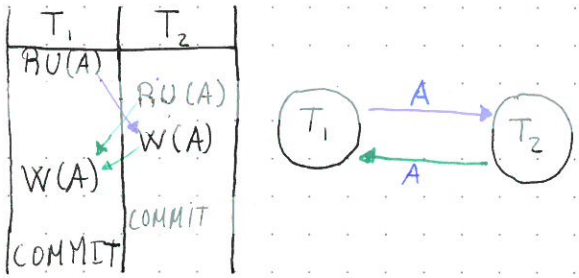
ACID

- **A**tomicitat: Tot o Res. Fem ús de logs per tirar endarrere.
- **C**onsistència: Coses estiguin ben fetes.
- **A**illament: Bloquejar BD pq no hi hagi interferències.
- **D**efinitivitat: Si és realitza, que perduri o sinó que es pugui recuperar.

Interferència entre Transaccions

Actualització Perduda (Lost Update)

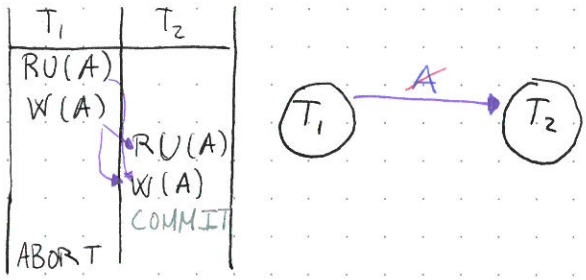
Es perd la escriptura (modificació) que ha realitzat una altra transacció prèviament.



Lectura No Confirmada (Dirty Read)

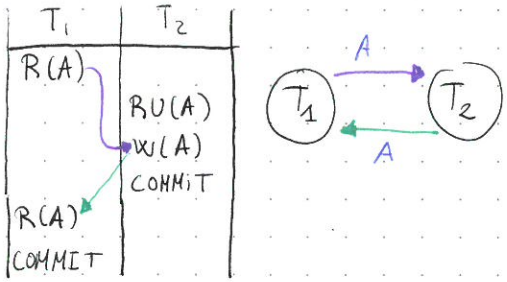
Una transacció llegeix un gràmul que ha sigut modificat per una altre transacció prev. i aquella altra → ABORT → la torna a modificar. ☹

// Això implica que la dada llegida per la segona Transacció sigui "bruta".



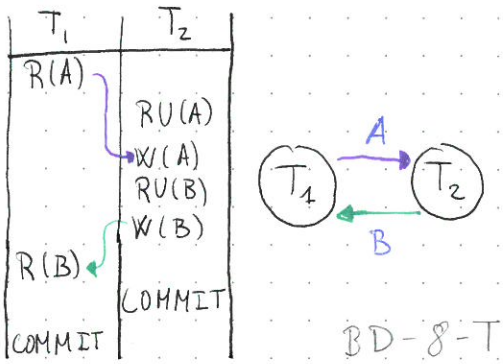
Lectura No Repetible (Unrepeatable Read)

Llegir dues vegades una dada (En mateixa T_i) i que aquesta tingui valors diferents.



Anàlisis Inconsistent (Inconsistent Analysis)

Quan una T_i llegeix 1 o + gràmuls i una segona T_j modifica algún d'aquells durant T_i i fa COMMIT abans que T_i.



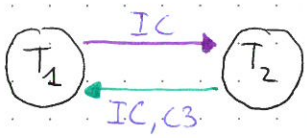
Fantasmes (Phantom Problem)

#És un cas d'Anàlisis Inconsistent.

T_i llegeix dades i abans de processar-les una altre T_j ↳ Afegeix / Modifica el conjunt de dades que ha llegit T_i i fa que no tingui mateixa quant que a l'inici. ▽ NO serà el cas si s'eliminen.

// Dades que han aparegut/desaparegut són fantasmes.

T_1	T_2
R(IC)	
R(C1)	
R(C2)	
	RU(C3)
	W(C3)
	RU(IC)
	W(IC)
R(IC)	
R(C1)	
R(C2)	
R(C3)	
COMMIT	
	COMMIT



#IC: Information Control.

Teoria de la Serialitzabilitat

Defineix les cond. que hem de complir les transaccions per considerar-les aillades.

- Gràmul: Unitat de dades que controla SGBD. Normalmt pàgina (bloc) de disc.
  - R(G): Lectura del gràmul G. SELECT.
  - RU(G): Lectura del gràmul G amb intenció de modificar. } UPDATE / INSERT DELETE.
  - W(G): Modificació del gràmul G.
- Horari: On es guarden, en ordre, les accions de cada T_i.
- Horari serial: No hi ha interferències entre transaccions.
- Accions Conflictives: Quan diferents transaccions operin sobre mateix gràmul i alguna acció sigui W(G).
- Horari Serialitzable: Horari que podem fer graf de precedència sense cicles.

Recuperabilitat

Horari compleix criteri de la recuperabilitat si T_2, que treballa sobre gràmul que ha modificat T_1 (previ mt), no confirma els seus canvis abans de que ho faci T_1.

No sig que modifiqui R(G) ja és suficient

// Si no complissim això: T_1 fes ABORT posteriorment a que T_2 hagi fet COMMIT sobre gràmul de T_1, no podríem ja saber quines dades eren les inicials.

Control Concurrència amb Reserves

Reservar Gràmul. Garanteix Aïllament.

- LOCK(G, S) = Permet fer R sobre G. Modalitat Compartit.
- LOCK(G, X) = Permet fer RU, W sobre G. Modalitat Exclusiva.
  Cap altra T_i pot fer LOCK de G (cap tipus)
- UNLOCK(G) = Desbloqueje gràmul G.

PR2F += Reserves fins acabament transaccions.

Nivells Aïllament

- READ UNCOMMITED: No permet actualitzar dades fins que acabi.
  NO es fan reserves de lectura.

Això fa que no hi hagi "Actualització perduda", però per exemple, no evita una "lectura no confirmada" donat que per fer Read no han de fer LOCK i pot ser que faci Read quan una altra està LOCK i ↳ Abort Mod. posteriorment.

- READ COMMITED: Evita que altra transacció llegeixi si no hem acabat.
  X fins al final; S fins després llegir.

Ara sí que evitem "Lectura No Confirmada" Pq fins que no acabi, no permetem fer Read. En canvi, "Lectura No Repetible" ens continua afectant.

- REPEATBLE READ: Impedeix que una altra T_j actualitzi dada que s'hagi llegit. X, S fins al final.

Podem obs. que evitem directament "Lectura No Repetible" i les anteriors. Això implica que tmb evita "Anàlisis Inconsistent" pq no permet que una altra T_j actualitzi gràmul que hem fet LOCK.

- SERIALITZABLE: Reserves X, S fins al final. ∀ o IC també.
  Evita fantasmes (Tot bàsicament).

Impedeix fantasmes pq a diferència de "REPETEABLE READ" que no LOCK IC, aquesta sí ho fa. Fent que si T_i llegeix IC, queda LOCK i una altre T_j no pugui fer ↳ INSERT, UPDATE, DELETE fins que acabi T_i.

Abraçades Mortals (Deadlock)

Si hi ha cicle en graf d'espera.

SGBD ho evita:

- Limitant temps per T_i
- Detectar i Resoldre.
- Previndre abans que passin.