

Proiect la baze de date

-Cerința 18-

Exemplificarea isolation levels prin exemple de tranzacții care se execută în paralel în condiții de concurență, evidențiind efectele diferitelor niveluri de izolare asupra concurenței și integrității datelor.

Isolation level-urile se referă la modul în care operațiile dintr-o tranzacție sunt vizibile pentru alte tranzacții concurente. În alte cuvinte, se ocupă de suprapunerile între operații de update-uri și determină ce anomalii de concurență sunt permise sau prevenite.

Cele trei probleme clasice sunt dirty read, non-repeatable read și phantom read.

- **Dirty read** apare în momentul în care o tranzacție citește date modificate de o altă tranzacție care nu a fost încă comisă.
- **Non-repeatable read** este o tranzacție ce citește de două ori aceleași date și obține rezultate diferite din cauza altei tranzacții ce a fost efectuată între timp.
- **Phantom read** reprezintă o tranzacție ce găsește rânduri noi/lipsa ce au fost adăugate, respectiv șterse de o altă tranzacție trecută.

Cele patru niveluri de izolare sunt “read uncommitted”, “read committed”, “repeatable read” și “serializable” ordonate de la cele mai puțin sigure la cele mai sigure.

Read uncommitted este cel mai slab nivel de izolare și este foarte rapid, însă riscă furnizarea de rezultate greșite din cauza “dirty read”-ului (există posibilitatea de roll-back, datele nu sunt confirmate). Rapiditatea acestuia vine la pachet cu un risc mare de inconsistență.

Read committed este nivelul standard folosit de oracle. După cum precizează și numele, putem citi doar tranzacțiile deja confirmate de utilizator, fără posibilitatea de roll-back. Astfel, oracle nu permite dirty reads. Dezavantajul său este că nu previne non-repeatable reads sau “phantom reads”. Asta înseamnă că ar putea citi multiple versiuni ce au fost committed înainte pe aceeași tranzacție. Un alt dezavantaj este reprezentat de faptul că reexecutarea aceleiași interogări poate returna rezultate diferite.

Repeatable read vede doar datele committed înainte ca tranzacția să înceapă, iar ca dezavantaj nici acest tip de izolare nu previne în totalitate aceste “phantom reads”.

Serializable este cel mai “sigur” nivel de izolare, prevenind orice fel de anomalie în timpul tranzacțiilor. Este folosit în principal în cazurile în care este extrem de important să avem aceste restricții puse. Acest tip este acceptat de oracle. Drept dezavantaje acesta reduce paralelismul și performanța, însă maximizează corectitudinea.

Pentru a înțelege mai bine aceste tipuri de izolare, următorul tabel le are grupate sugestiv:

	Dirty Read	Non-repeatable read	Phantom Read
READ UNCOMMITTED	DA	DA	DA
READ COMMITTED	NU	DA	DA
REPEATABLE READ	NU	NU	DA
SERIALIZABLE	NU	NU	NU

```

SET TRANSACTION ISOLATION LEVEL serializable ;

UPDATE FACTURA SET pret = 100 WHERE id_comanda = 10;

SELECT pret FROM FACTURA WHERE id_comanda = 10;

commit;

```

Output: Este implicit în READ COMMITTED

PRET
330.00

```

--pentru nivelurile de izolare
SET TRANSACTION ISOLATION LEVEL serializable ;

UPDATE FACTURA SET pret = 100 WHERE id_comanda = 10;

SELECT pret FROM FACTURA WHERE id_comanda = 10;

commit;
rollback;

```

Am experimentat cu aceste comenzi simple în DataGrip pentru a înțelege și apoi explica cele două tipuri de isolation levels. Ambele tipuri sunt dependente de commit și afișează datele fără anomalii. Este nevoie de rollback pentru a afișa datele anterioare.