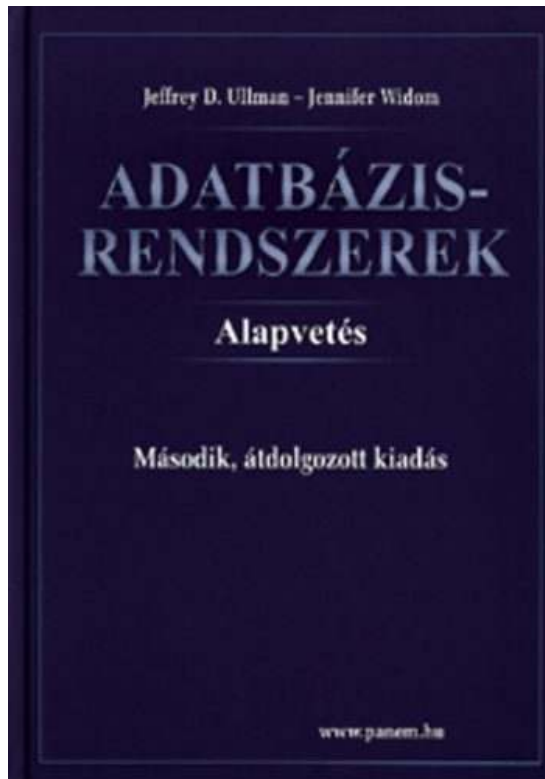


Tranzakciók az SQL-ben



Ullman-Widom: Adatbázisrendszerek
Alapvetés

Második, átdolgozott kiadás, Panem,
2009

6.6. Tranzakciók az SQL-ben
(Gyakorlaton csak SAVEPOINT,
COMMIT és ROLLBACK lesz.
Ez nem törzsanyag - nincs vizsgán,
később lesz az Adatbázisok-2-n)

Miért van szükség tranzakciókra?

- Az adatbázis rendszereket általában több felhasználó és folyamat használja egyidőben.
 - Lekérdezések és módosítások egyaránt történhetnek.
- Az operációs rendszerektől eltérően, amelyek *támogatják* folyamatok interakcióját, az adatbázis rendszereknek el kell különíteniük a folyamatokat.

Példa: rossz interakció

- Egy időben ketten töltenek fel 100 dollárt ugyanarra a számlára ATM-en keresztül.
 - Az adatbázis rendszernek biztosítania kell, hogy egyik művelet se vesszen el.
- **Ezzel szemben** az operációs rendszerek megengedik, hogy egy dokumentumot ketten szerkesszenek egyidőben. Ha mind a ketten írnak, akkor az egyik változtatás elvész (elveszhet).

Tranzakciók

- *Tranzakció* = olyan folyamat, ami adatbázis lekérdezéseket, módosításokat tartalmaz.
- Az utasítások egy „értelmes egészt” alkotnak.
- Egyetlen utasítást tartalmaznak, vagy az SQL-ben explicit módon megadhatóak.

ACID tranzakciók

■ *Az ACID tranzakciók:*

- ❑ *Atomiság (atomicity):* vagy az összes vagy egy utasítás sem hajtódik végre.
 - ❑ *Konzisztencia (consistency):* az adatbázis megszorítások megőrződnek.
 - ❑ *Elkülönítés (isolation):* a felhasználók számára úgy tűnik, mintha folyamatok, elkülönítve, egymás után futnának le.
 - ❑ *Tartósság (durability):* egy befejeződött tranzakció módosításai nem vesznek el.
- ## ■ *Opcionálisan:* gyengébb feltételek is megadhatóak.
-

COMMIT

- A COMMIT SQL utasítás végrehajtása után a tranzakció véglegesnek tekinthető.
 - A tranzakció módosításai véglegesítődnek.

ROLLBACK

- A ROLLBACK SQL utasítás esetén a tranzakció *abortál*.
 - Azaz az összes utasítás visszagörgetésre kerül.
- A 0-val való osztás vagy egyéb hibák, szintén visszagörgetést okozhatnak, akkor is, ha a programozó erre nem adott explicit utasítást.

Példa: egymásra ható folyamatok

- A **Felszolgál(bár, sör, ár)** táblánál tegyük fel, hogy Joe bárjában csak Bud és Miller sörök kaphatók 2.50 és 3.00 dollárért.
- Sally a **Felszolgál** táblából Joe legolcsóbb és legdrágább sörét kérdezi le.
- Joe viszont úgy dönt, hogy a Bud és Miller sörök helyett ezentúl Heinekent árul 3.50 dollárért.

Sally utasításai

(max) SELECT MAX(ár) FROM Felszolgál
WHERE bár = 'Joe bárja';

(min) SELECT MIN(ár) FROM Felszolgál
WHERE bár = 'Joe bárja';

Joe utasításai

- Ugyanabban a pillanatban Joe a következő utasításokat adja ki:

(del) DELETE FROM Felszolgál
WHERE bár = 'Joe bárja';

(ins) INSERT INTO Felszolgál
VALUES('Joe bárja', 'Heineken', 3.50);

Átfedésben álló utasítások

- A **(max)** utasításnak a **(min)** kell végrehajtódnia, hasonlóan **(del)** utasításnak az **(ins)** előtt, ettől eltekintve viszont nincsenek megszorítások a sorrendre vonatkozóan, ha Sally és Joe utasításait nem gyűjtjük egy-egy tranzakcióba.

Példa: egy furcsa átfedés

- Tételezzük fel a következő végrehajtási sorrendet: **(max)(del)(ins)(min)**.

Joe árai:

Utasítás:	{2.50,3.00}	{2.50,3.00}		{3.50}
Eredmény:	(max)	(del)	(ins)	(min)
	3.00			3.50

- Mit lát Sally? $MAX < MIN$!

A probléma megoldása tranzakciókkal

- Ha Sally utasításait, **(max)(min)**, egy tranzakcióba gyűjtjük, akkor az előbbi inkonzisztencia nem történhet meg.
- Joe árait ekkor egy adott időpontban látja.
 - Vagy a változtatások előtt vagy utánuk, vagy közben, de a MAX és a MIN ugyanazokból az árakból számolódik.

Egy másik hibaforrás: a visszagörgetés

- Tegyük fel, hogy Joe a **(del)(ins)** és utasításokat nem, mint tranzakció hajtja végre, utána viszont úgy dönt, jobb ha visszagörgeti a módosításokat.
- Ha Sally az **(ins)** után, de visszagörgetés előtt hajtja végre a tranzakciót, olyan értéket kap, 3.50, ami nincs is benne az adatbázisban végül.

Megoldás

- A **(del)(ins)** és utasításokat Joe-nak is, mint tranzakciót kell végrehajtatnia, így a változtatások akkor válnak láthatóvá, ha tranzakció egy COMMIT utasítást hajt végre.
 - Ha a tranzakció ehelyett visszagörgetődik, akkor a hatásai sohasem válnak láthatóvá.

Elkülönítési szintek

- Az SQL négy *elkülönítési szintet* definiál, amelyek megmondják, hogy milyen interakciók engedélyezettek az egy időben végrehajtódó tranzakciók közt.
- Ezek közül egy szint (“sorbarendevezhető”) = ACID tranzakciók.
- Minden db rendszer a saját tetszése szerint implementálhatja a tranzakciókat.

Az elkülönítési szint megválasztása

- Az utasítás:

SET TRANSACTION ISOLATION LEVEL X

ahol X =

1. SERIALIZABLE
2. REPEATABLE READ
3. READ COMMITTED
4. READ UNCOMMITTED

Sorbarendevezhető (serializable) tranzakciók

- Ha Sally a (max)(min), Joe a (del)(ins) tranzakciót hajtja végre, és Sally tranzakciója SERIALIZABLE elkülönítési szinten fut, akkor az adatbázist vagy Joe módosításai előtt vagy után látja, a (del) és (ins) közötti állapotban sohasem.

Az elkülönítési szint személyes választás

- Ez a döntés csak azt mondja meg, hogy az illető hogyan látja az adatbázist, és nem azt, hogy mások hogy látják azt.
- **Példa:** Ha Joe sorbarendevezhető elkülönítési szintet használ, de Sally nem, akkor lehet, hogy Sally nem talál árakat Joe bárja mellett.
 - azaz, mintha Sally Joe tranzakciójának közepén futtatná a sajátját.

Read-Committed tranzakciók

- Ha Sally READ COMMITTED elkülönítési szintet választ, akkor csak kommitálás utáni adatot láthat, de nem feltétlenül mindig ugyanazt az adatot.
- **Példa:** READ COMMITTED mellett megengedett a **(max)(del)(ins)(min)** átfedés amennyiben Joe kommitál.
 - Sally legnagyobb megdöbbenésére: $MAX < MIN$.

Repeatable-Read tranzakciók

- Hasonló a read-commited megszorításhoz. Itt, ha az adatot újra beolvassuk, akkor amit először láttunk, másodszor is látni fogjuk.
- De második és az azt követő beolvasások után akár *több* sort is láthatunk.

Példa: ismételhető olvasás

- Tegyük fel, hogy Sally REPEATABLE READ elkülönítési szintet választ, a végrehajtás sorrendje: **(max)(del)(ins)(min)**.
 - **(max)** a 2.50 és 3.00 dollár árakat látja.
 - **(min)** látja a 3.50 dollárt, de 2.50 és 3.00 árakat is látja, mert egy korábbi olvasáskor **(max)** már látta azokat.