

1. Ismertesse az adatbázis konkurencia problémákat és a tranzakciók alaptulajdonságait! Mire szolgálnak a tranzakció izolációs szintek? Milyen izolációs szinteket támogat az Microsoft SQL Server?

DIA 2 21 oldal

Probléma

- Konkurens adathozzáférés
- Egyazon adategységhez (itt: tábla egy rekordja) egy időben **többen férnek hozzá**, és legalább **egyikük módosítja**.



Konkurens adathozzáférés, azaz egyazon adategységhez (tábla egy rekordja) egy időben többen férnek hozzá, és legalább az egyikük módosítja. Ezáltal létrejöhet két alternatív rekord. Mármost, ha feltételezzük, hogy mindketten vagy többen lekérnek egy adott időpillanatban, ezután egy illető frissíti azt a rekordot. Így neki már a frissített adat lesz megjelenítve míg a többieknek még az első pillanatban megadott érték az. Ezen kívül, ha

egyszerre több ember próbálja meg szerkeszteni egy adott értéket.

DIA 2 22 oldal

A tranzakciók alaptulajdonságai a következők: **(ACID)**

- Atomicity - Atomitás
- Consistency – Konzisztencia
- Isolation – Izoláció
- Durability - Tartóság

DIA 2 27 oldal-tól 33-ig

A tranzakciós műveletek alapjáraton párhuzamos műveletek, viszont végrehajtásukat lineáris egymást követően kell véghez vinni. Ezekből több probléma is lehet ilyen például a „**piszkos olvasás**”, „**elveszett módosítás**”, „**nem megismételhető olvasás**”, „**fantom rekordok**”. Ahhoz, hogy ezeket megelőzzük izolációs szinteket vezetünk be. Mindegyik szint más problémákat képes megoldani azzal, hogy mennyire „strict” szabályokat vezetünk be.

SQL szabvány szerint a következő szinteket különböztetjük meg:

- **Read uncommitted**, itt mind a négy probléma fennáll
- **Read committed**, ez már a piszkos olvasásra add megoldást
- **Repeatable read**, itt nincs „piszkos olvasás” és „nem megismételhető olvasás”
- **Serializable**, itt már egyik probléma sem fordulhat elő

Izolációs szintek MSSQL szerint

- **Read uncommitted**
- **Read committed**, ez az alapértelmezett
- **Repeatable read**
- **Serializable**
- **Snapshot**, tranzakció kezdetekor pillanatképet készít, sor szintű verziók tranzakciókként a pillanatkép által