

# Háttéralkalmazások

Adatbázis-kezelő rendszerek architekturális felépítése

Tranzakciókezelés relációs adatbázisokban

Microsoft SQL Server sajátosságai



Automatizálási és  
Alkalmazott  
Informatikai Tanszék

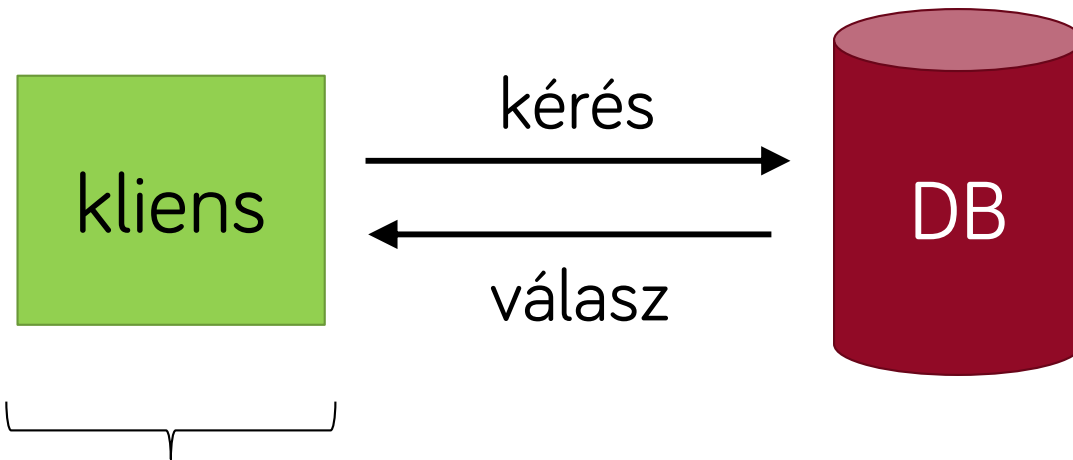
# Tartalom

- Adatbázis-kezelő rendszerek architektúrális felépítése
- Adatbázis, relációs adatmodell
  - > Microsoft SQL Server platform
- Tranzakciók

# Adatbázis-kezelő rendszerek architekturális felépítése

# Rendszer architektúra

- Kliens-szerver architektúra



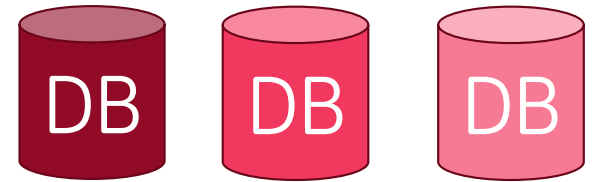
*Többrétegű alkalmazás*

# Pontosabb felépítés

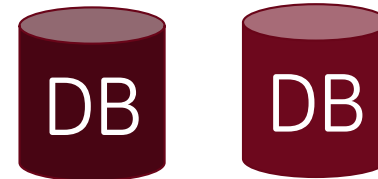
kliens

szerver gép

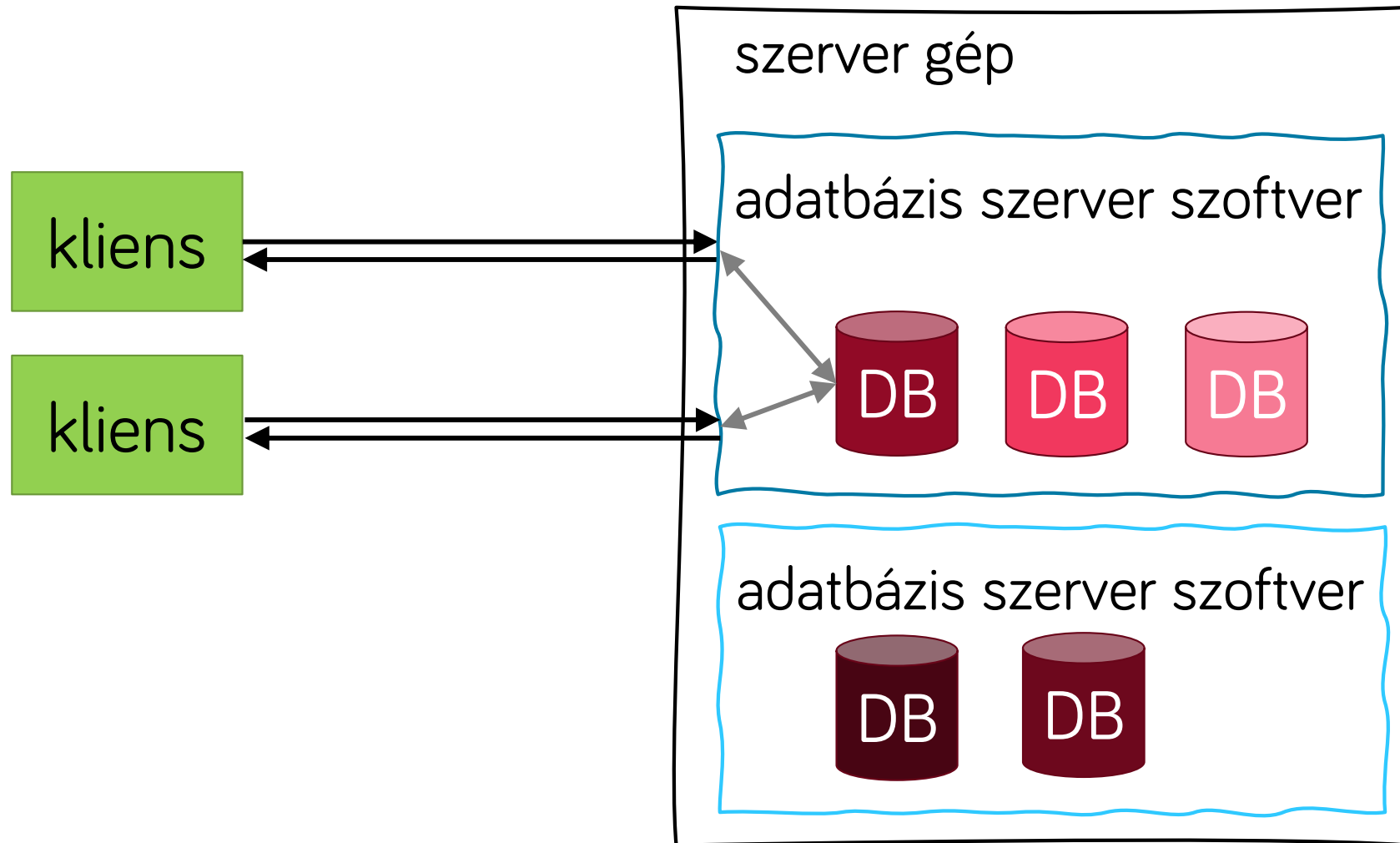
adatbázis server szoftver



adatbázis server szoftver



# Pontosabb felépítés



# Adatbázis, relációs adatbázis

# Adatbázis

- Logikailag összefüggő adatok rendezett gyűjteménye
- *Adatok*
  - > Ismert tények, amelyek számítógépes adattárolón rögzíthetők
- *Rendezett gyűjtemény*
  - > Könnyű tárolás
  - > Könnyű módosítás
  - > Könnyű lekérdezés
- *Összefüggő adatok*
  - > Lefedik egy felhasználó csoport érdeklődési területét
  - > Szükséges adatok



# Relációk (táblák) tulajdonságai

rekord / sor {

OszlopID	OszlopA	OszlopB	OszlopC
Kulcs1	Skalár érték		
Kulcs2			
Kulcs3			
Kulcs4			

oszlop / attribútum

# Integritási kritériumok

OszlopID	OszlopA	OszlopB	OszlopC
Kulcs1	123		
Kulcs2	456		
Kulcs3	789	Rek999	

XXX	YYY
Rek998	
Rek999	

1. Entitás integritás: kulcs  $\neq$  NULL
2. Tartományi integritás: oszlop értéktípus & tartomány
3. Referenciális integritás: külső kulcs (foreign key)

# Felhasználói séma

- Felhasználói séma = objektumok az adatbázisban
- Relációs adatmodellből adódó
  - > Tábla
  - > Megszorítások
- + platformfüggő elemek

# Microsoft SQL Server platform

# Szerver komponensek

- SQL Server Service
- SQL Server Browser
- SQL Server Agent
- SQL Server Analysis Services
- SQL Server Reporting Services
- SQL Server Integration Services
- SQL Writer

# Felhasználói séma elemei

- Tábla
  - > Oszlop
  - > Computed Column
    - Virtuális
    - Tárolt
- Nézet
  - > Indexelhető → Tárolódik
- Index
- Szekvencia
  - > SQL Server 2012
- Programmodul
  - > Eljárás
  - > Függvény
  - > Trigger
  - > Assembly

# Adattípusok

- Szöveges típusok

- > Char(n)      }
- > Varchar(n)    }      1 bájt
- }      Max 8000
- > Nchar(n)      }
- > Nvarchar(n)   }      2 bájt
- }      Max 4000

- Nagyméretű objektumok (max 2 GB)

- > Text → Varchar(max),
- > Ntext → Nvarchar(max)
- > Image → Varbinary (max)

- Numerikus

- > Int
- > Float
- > Numeric(p,s) (=decimal, dec)
  - p: 1-38, default 18

- Dátum

- > Datetime
  - 1753. január 1-től (!)
- > Datetime2 (ez a javasolt)

- Félig strukturált adat

- > XML

# Elsődleges kulcsok generálása

- Identity kulcsszó
  - > create table Statusz(  
    ID int identity(1,1) primary key,  
    Nev nvarchar(20))
  - > Insert into Statusz values ('Kész')
- Lekérdezése
  - > ident\_current('Statusz')
    - Konkrét táblára, bármilyen session és scope
  - > @@IDENTITY
    - Bármelyik tábla, aktuális session, minden scope
  - > SCOPE\_IDENTITY ()
    - Bármelyik tábla, aktuális session, aktuális scope



# Adatbázis fizikai tárolása

- Adatbázis
  - > Adatfájl (.mdf)
    - Filegroup is lehet
  - > Tranzakciós napló (.ldf)
  - > Több sémát tartalmazhat
    - Alapértelmezett séma: dbo
    - Logikai csoportosítás, hozzáférés szabályozás
- Rendszer adatbázisok
  - > master, model, distribution, msdb, tempdb

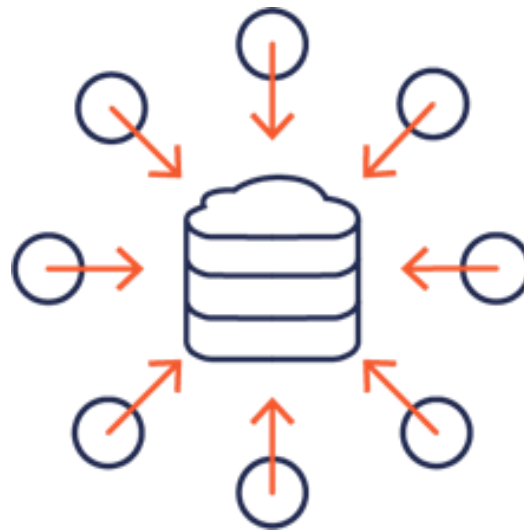
# Hozzáférés szabályozás

- Rendszer szintű
- Adatbázis szintű
- Séma szintű
- Objektum szintű
  - > Konkrét objektumhoz
  - > Táblák és nézetek esetén oszlop szint is megadható
- Nem csak engedélyezés, hanem tiltás is megadható
- Sor szintű hozzáférés nem szabályozható
  - MSSQL 2014-től részlegesen

# Tranzakciók

# Probléma

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

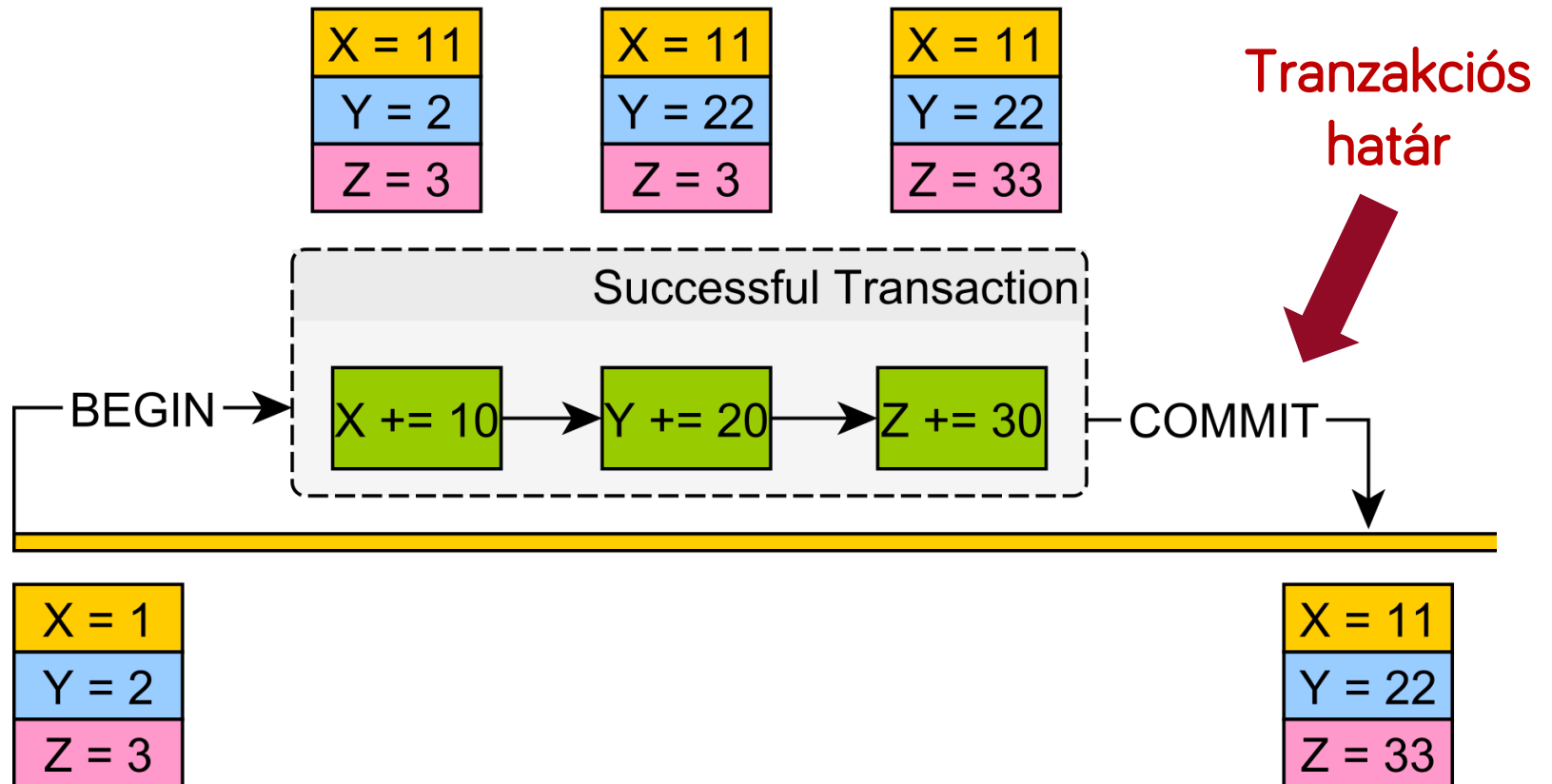


Kép forrása: <https://blog.yugabyte.com/a-primer-on-acid-transactions/>

# Tranzakció fogalma

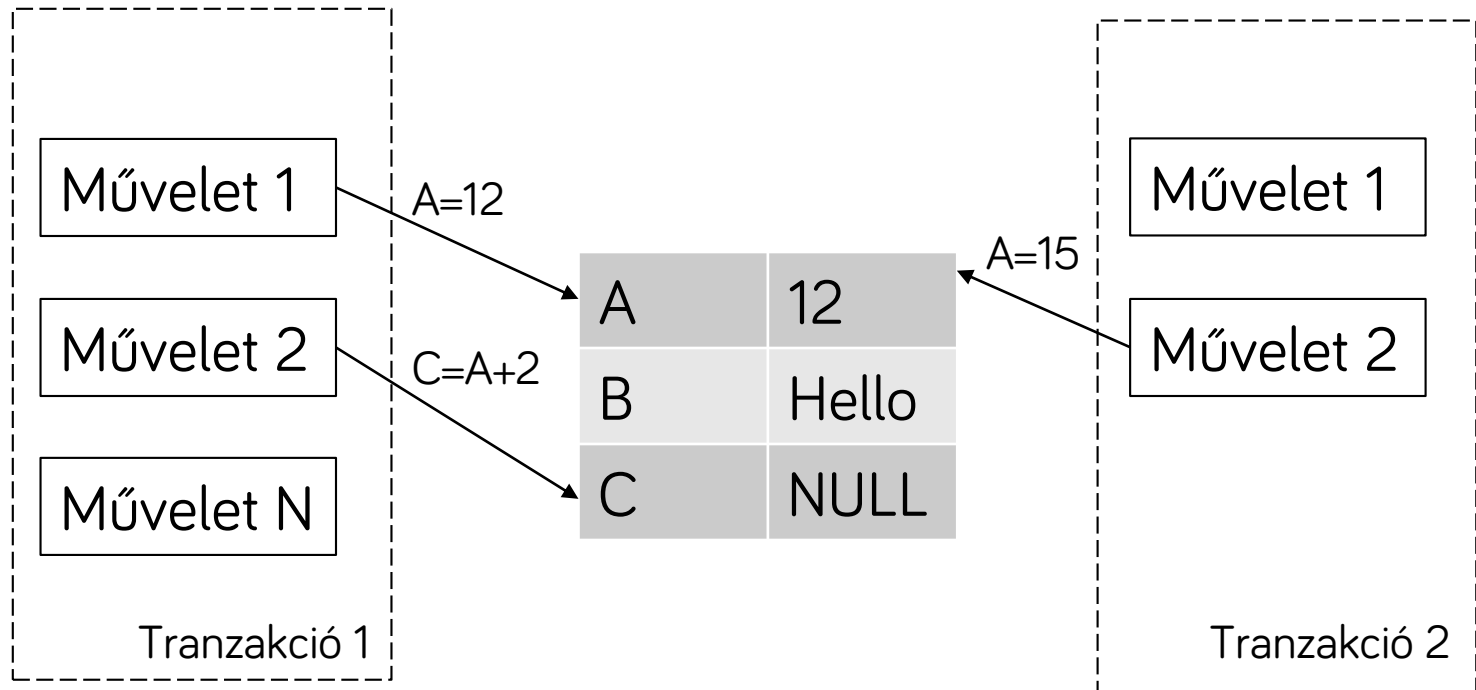
- A feldolgozás logikai egysége, olyan műveletek sorozata, melyek csak együttesen értelmesek.
- Alaptulajdonságok:
  - > Atomicity
  - > Consistency
  - > Isolation
  - > Durability

# Atomitás és konzisztencia

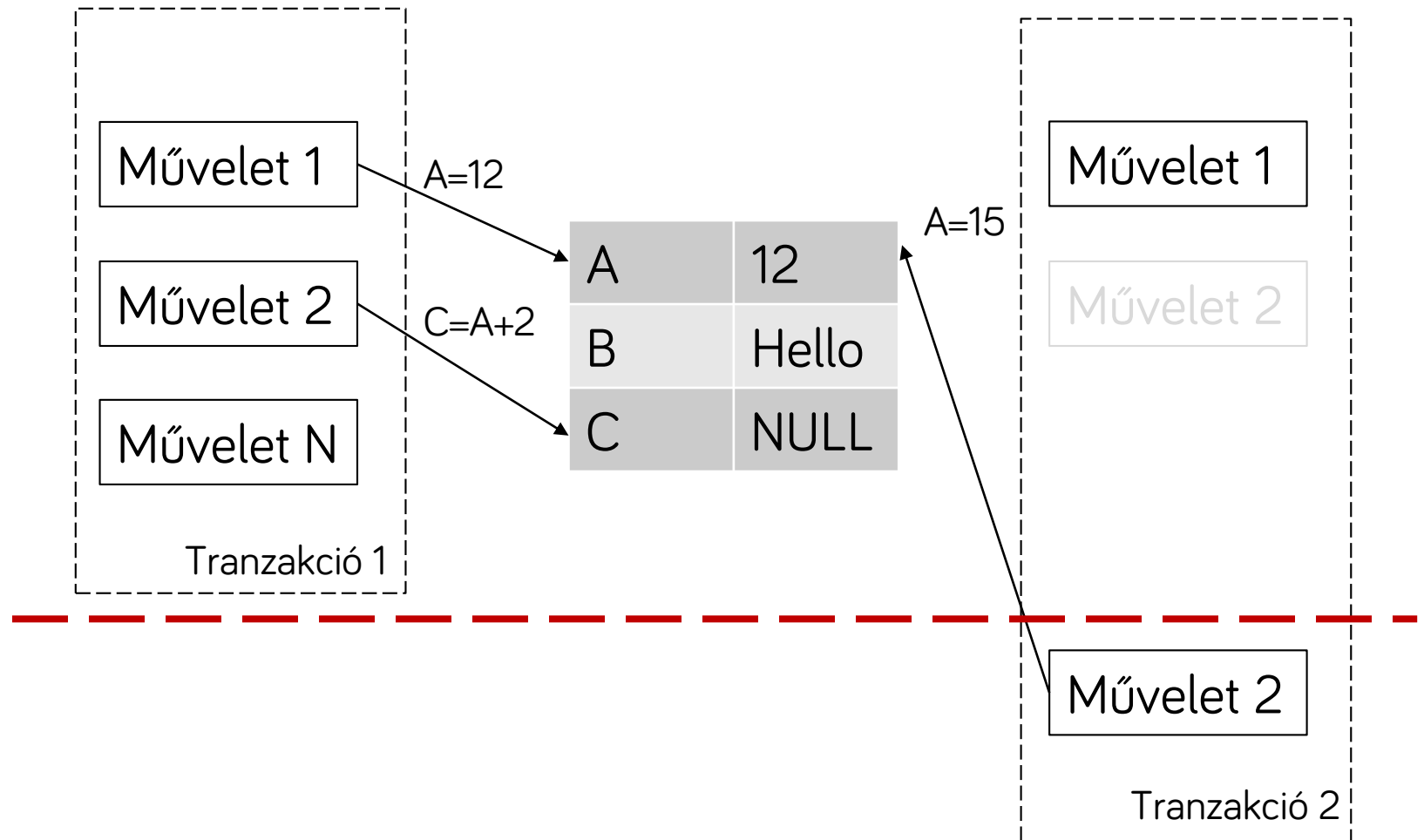


Forrás: <https://vladimihalcea.com/current-database-transaction-id/>

# Izoláció

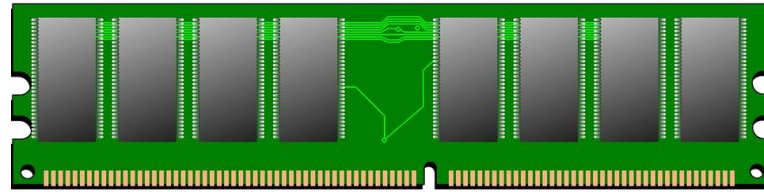
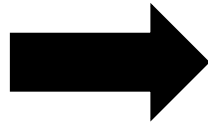


# Izoláció





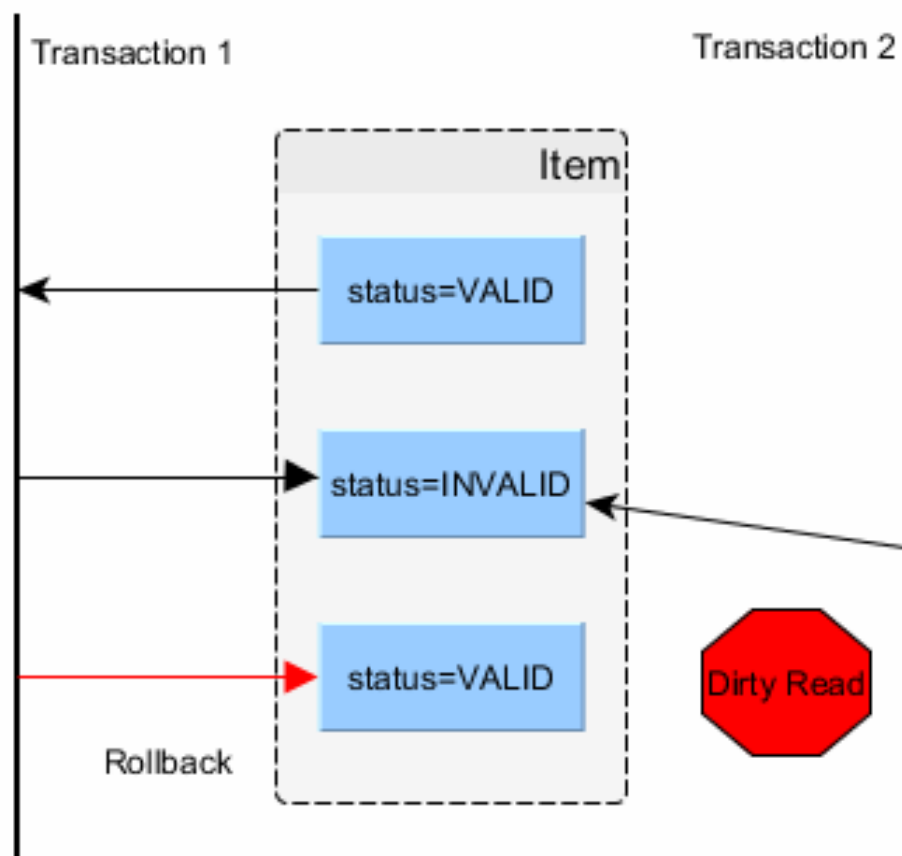
# Tartós



# Izolációs alapproblémák

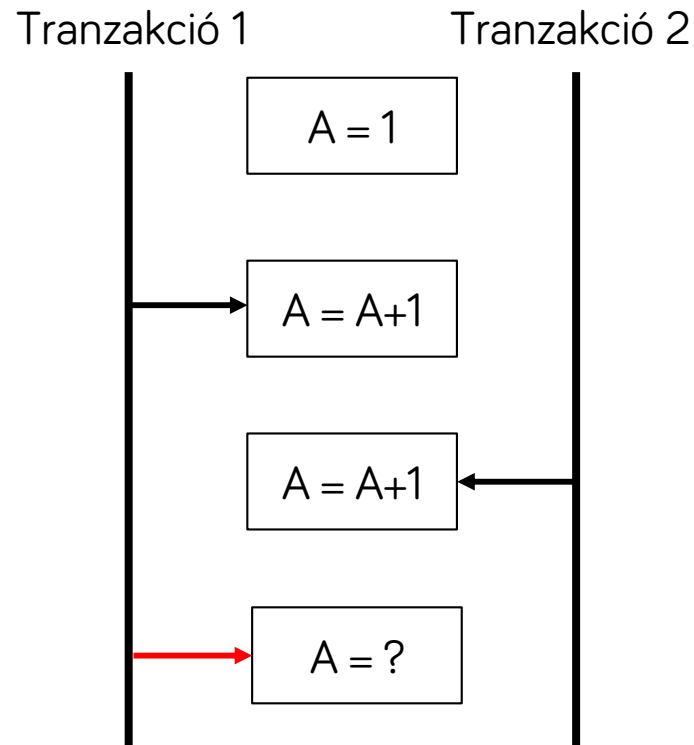
- Sok párhuzamos tranzakció
- Úgy kell végrehajtani, *mintha* egymás után történnének és nem párhuzamosan
- Problémák
  - > Piszkos olvasás
  - > Elveszett módosítás
  - > Nem megismételhető olvasás
  - > Fantom rekordok
- **Nem csak relációs adatbázis probléma!**
  - > Minden többfelhasználós, elosztott rendszerben valamilyen módon jelentkezik.

# Piszkos olvasás

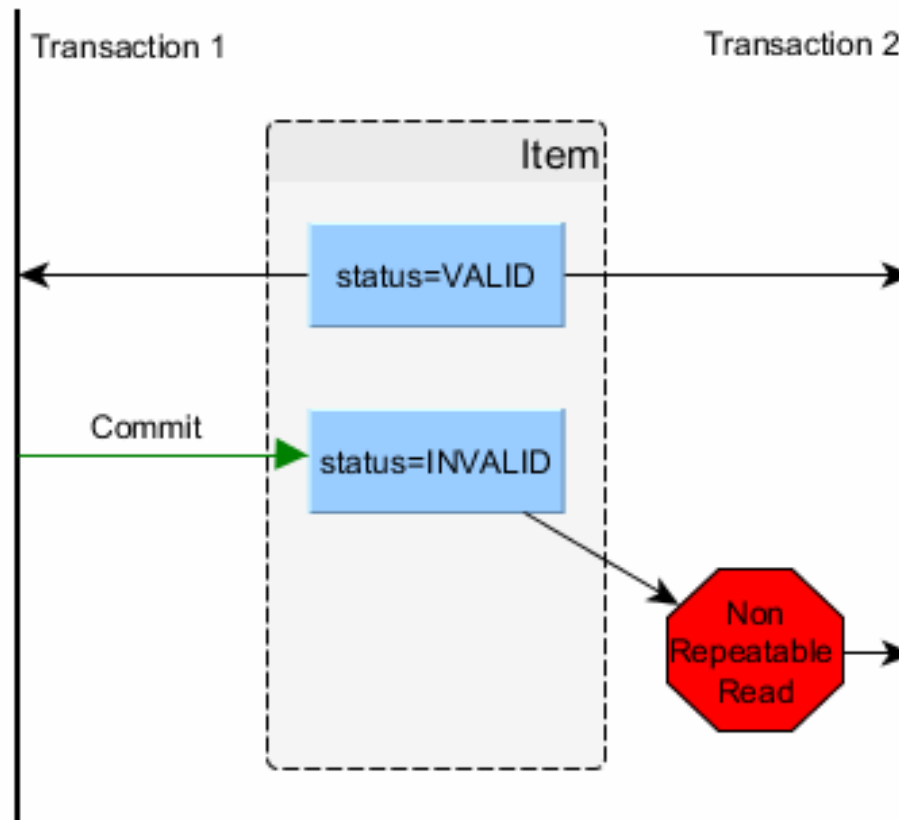


<https://vladmihalcea.com/2014/01/05/a-beginners-guide-to-acid-and-database-transactions/>

# Elvesztett módosítás

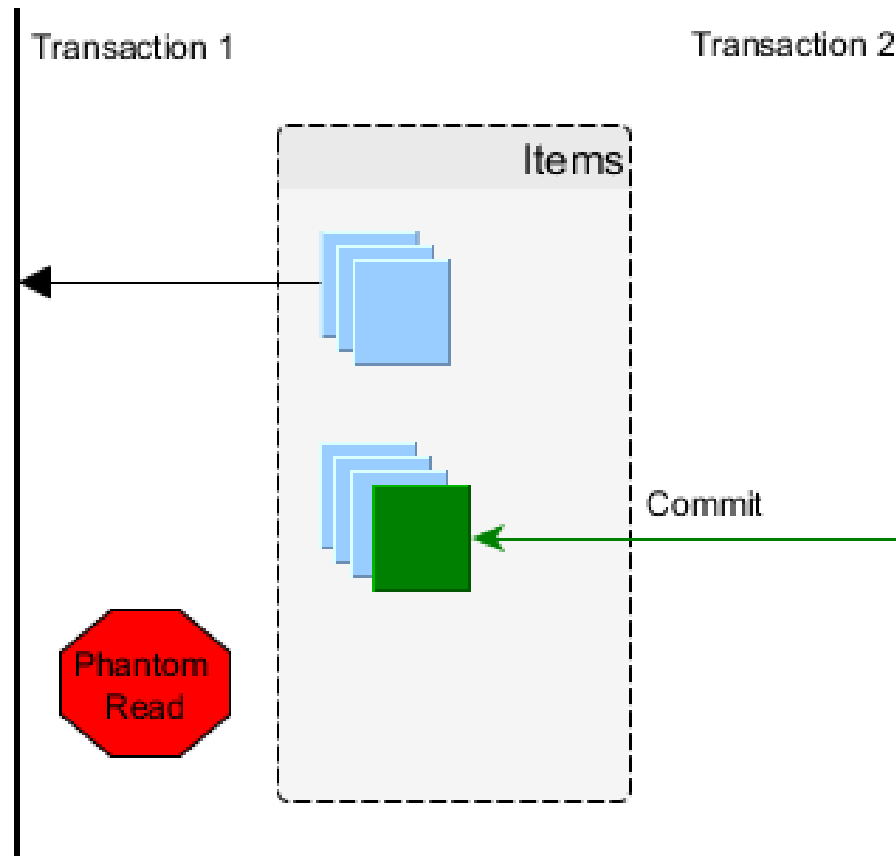


# Nem megismételhető olvasás



<https://vladmihalcea.com/2014/01/05/a-beginners-guide-to-acid-and-database-transactions/>

# Fantom rekordok



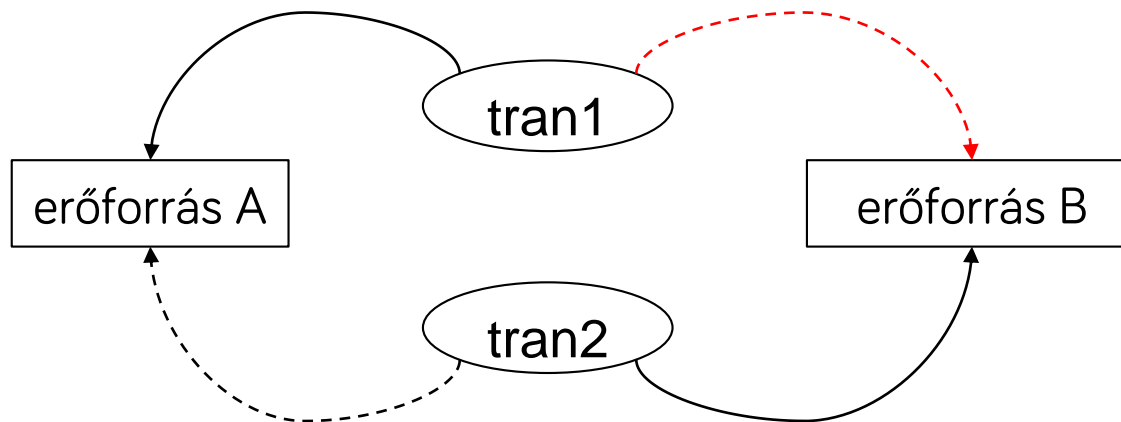
<https://vladimihalcea.com/2014/01/05/a-beginners-guide-to-acid-and-database-transactions/>

# Megoldás

- Tranzakciók ütemezése
- Csak olyan műveletek engedhetők meg, melyek nem sértik a helyes ütemezést
- Ha sérülne a helyes ütemezés, akkor a tranzakció vár
- Olyan ütemezés a megengedett, mely konfliktus-ekvivalens egy soros ütemezéssel
  - > Konfliktusmentes cserékkal soros ütemezéssé alakítható

# Ütemezés biztosítása

- Kétfázisú zárolás





# Izolációs szintek

- SQL szabvány szerinti izolációs szintek
  - > Read uncommitted → mind a 4 probléma
  - > Read committed → nincs piszkos olvasás
  - > Repeatable read → nincs piszkos olvasás, se nem megismételhető olvasás
  - > Serializable → egyik probléma sem fordulhat elő

# MSSQL: Izolációs szintek

- SQL szabvány szerintiek

- > Read uncommitted

- > Read committed

- > Repeatable read

- > Serializable

Alapértelmezett

Olvasás: megosztott zárat használ

Írás: más nem olvashatja

- Szabványtól eltérő

- > Snapshot

- Tranzakció kezdetekor pillanatkép.

- Sor szintű verziók tranzakciónként.

# MSSQL: Tranzakciók

- Tranzakció kezdés
  - > Auto commit: Minden utasítás önálló tranzakció
    - Ez az alapértelmezett mód
  - > Explicit tranzakciók: több utasítás esetén
    - begin tran[saction] + commit / rollback
    - (egymásba ágyazhatóak a tranzakciók)
- DML és DDL utasítások is tranzakció része
  - > Néhány kivétel van (create database, backup, restore)

# Jegyzet

Témához tartozó írásos jegyzet (más tárgy vonatkozó része)

<https://bmeviauac01.github.io/datadriven/hu/lecture-notes/transactions/>