

# Tartalom

- Legacy rendszerek integrációja
  - Legacy rendszerek jellemzői
  - Integrációs lehetőségek
- Integráció az adatok szintjén
  - Mit jelent adat szintű integráció?
  - Hogyan valósítható meg
- Master Data Management
  - Miért fontos
  - Hogyan

# Legacy rendszerek

## Definíció

- Angol eredetű terminológia
- Régi, ismeretlen vagy elavult belső technológiával működő hasznos rendszerek

## Miért működnek

- Lecserélésük költsége jelentős
- Szolgáltatásaik megfelelnek az elvárásoknak
- Kompromisszummal és/vagy olcsóbban lehet foltozni őket, mint lecserélni (ha van szakértelem)

# Legacy rendszerek

## Problémák

- Ismeretlen vagy elavult belső technológia
- Korlátozott kompatibilitás
- Szabványos interfészek hiánya
- Biztonsági kockázatok
- Adatok tárolása egyedi adatfájlokban

## Kommunikáció

- A kommunikáció sok esetben aszinkron, ritkábban szinkron
- Az programok közötti kommunikáció jellemzően fájlok írásával/olvasásával történik
- Kétféle fájlstruktúrát használnak:
  - Szöveges, fix hosszú vagy tagolt rekordszerkezettel
  - Bináris rekord szerkezet (az adatok, adatszerkezetek belső reprezentációját írják ki az fájlokba)

# Legacy rendszerek tulajdonságai

## Programkód

- Régen használt programozási nyelven íródott
- Ezek a programozási nyelvek ma már kevésbé ismertek
- Példák: RPG, COBOL, FORTRAN, CLIPPER, dBase, FoxPro, Modula-2, PL/1, Simula, Smalltalk, stb.

## Adattárolás

- Nincs relációs adatbázis, az adatokat jellemzően fájlokban tárolják
- A fájlok tagolt szöveges vagy rekord adatszerkezettel rendelkeznek
- Az adatok kezelése nem különül el az adatok feldolgozásától)

# Legacy rendszerek tulajdonságai

## Kommunikáció

- Egyedileg fejlesztett interfészeken
- Saját technológia alkalmazásával
- Jellemzően nem szabványos protokollok alkalmazásával

## Biztonság

- Kevés kommunikáció, interfész
- Zárt rendszerkörnyezetben működnek
- Kevesebb veszélyforrás (régén, a rendszer elkészítésének idején)

## Dokumentáció

- Nem áll rendelkezésre (nem készült, elavult, nem frissített)
- Kevés dokumentáció (nagyvonalú, elnagyolt)

# Miért szükséges integrálni?

## Munkafolyamatok támogatása

- Ugyanaz az adat több munkafolyamatot is támogat
- A különböző munkafolyamatokat különböző rendszerek támogatják
- Legacy rendszerek is kezelnek közösen használt adatokat

## Adatkonzisztencia biztosítása

- Ha az adatokat több rendszerben rögzítik
- Az adatok többszörös rögzítése során eltérések jelentkezhetnek
- Az eltérések megkeresése és kiküszöbölése idő- és munka-igényes
- Megoldás az egyszeri adatrögzítés

## Erőforrások optimalizálása

- Elvárás, hogy egy adatot csak egyszer és egy helyen kelljen rögzíteni
- Ne legyen szükség az adatok tisztítására (időigényes feladat)

# Integrációs lehetőségek

## Nézetek és direkt adatkapcsolat

- Megvalósíthatósága függ a legacy rendszer technológiájától
- Leggyorsabban kivitelezhető
- Az adatok elérése direkt módon történik (fizikailag ugyanaz az adat)
- Az integráció adat lekérdezési lehetőséget biztosít, adott esetben írási hozzáférés is megvalósítható

## Fájlok segítségével

- A rendszerek fájlokat írnak és fájlokat olvasnak
- A fájlok kötött formátumú rekordokat tartalmaznak
- A fájlok kódolása lehet szöveges vagy bináris

# Integrációs lehetőségek

## Egyedi Interfészek

- Minden adatcsatornához egyedi interfész elkészítése
- Egyedi kódolás, egyedi protokoll
- Egyedi adatátvitel
- Csak az adott két rendszer használja

## Szabványos interfészek

- Szabványos protokoll használata (jellemzően http/https)
- Adatcsatornák beteretelése a felépített szabványos kommunikációs protokollba
- Szabványos adatszerkezetek használata
- Csatlakozás ESB-hez



# Adat szintű integráció

## Definíció

- Két rendszer ugyanazt az adathalmazt használja
- Lehet fizikailag ugyanaz vagy logikailag ugyanaz

## Fizikailag ugyanaz

- Mindkét rendszer ugyanahhoz az adatbázishoz kapcsolódik, ugyanazokat a táblákat írja olvassa
- Kérdés a párhuzamos hozzáférés menedzselése
- Ki módosítja az adatokat? írás/olvasási jogok
- Kidolgozott tranzakciókezelésre van szükség

# Adat szintű integráció

## Logikailag ugyanaz

- Valamilyen másolat
- Kérdés az adatok hitelessége
- Ugyanazokat az adatokat látja mindkét rendszer?
- Szinkronizálásra van szükség

## Adatok szinkronizálása

- Mikor történik a szinkronizálás?
- Teljes adat áttöltés vagy csak a változások?
- Azonnali szinkronizálás – rendszer vagy rekord szinten
- Késleltetett szinkronizálás – rendszer vagy rekord szinten

# Megvalósítási lehetőségek

## Tranzakció-kezelés

- Az adatokat módosító rendszernek biztosítania kell az adatok integritását és konzisztenciáját
- Egy tranzakció akkor zárulhat le, ha minden az adatok konzisztenciáját biztosító adatmódosítás megtörtént

## Rendszer szintű szinkronizálás

- Az alkalmazásnak nem kell foglalkoznia az adatok szinkronizálásával, azt az adatbáziskezelő végzi
- Az adatok konzisztenciáját a rendszerkörnyezet biztosítja
- Egyirányú, mindig van egy mester adatbázis
- Teljes szinkronizáció és delta (változások átemelése)

# Megvalósítási lehetőségek

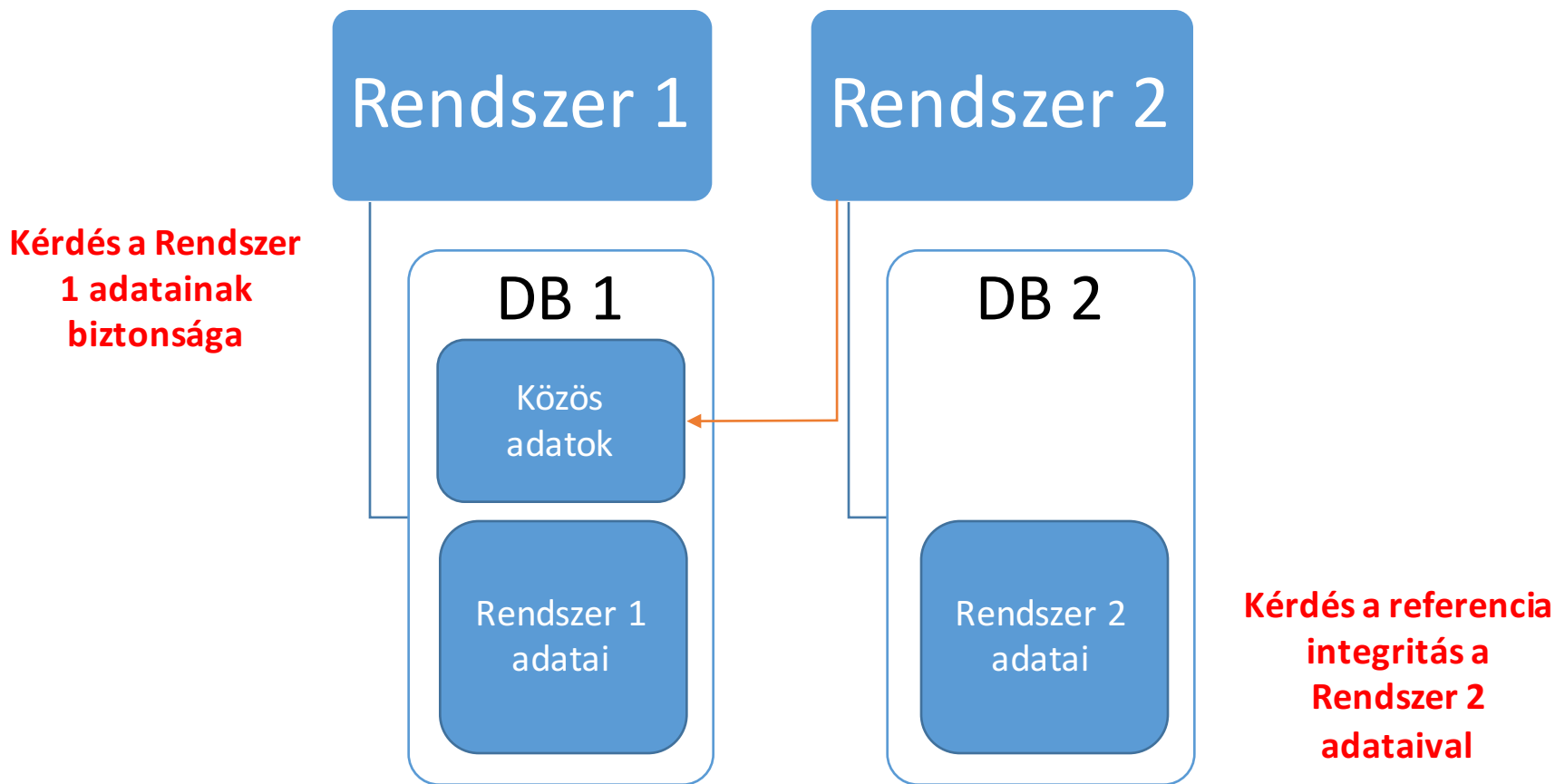
## Rekord szintű replikáció

- Egy tranzakció befejezésével az adatrekordok átemelése
- Rekordok kötegelt átemelése rendszeres időközönként
- Azonnali replikáció esetén van fontos jelentősége
- Egyirányú és kétirányú is lehet
- Az adatkonzisztencia megőrzése tervezést igényel

## Késleltetett szinkronizálás

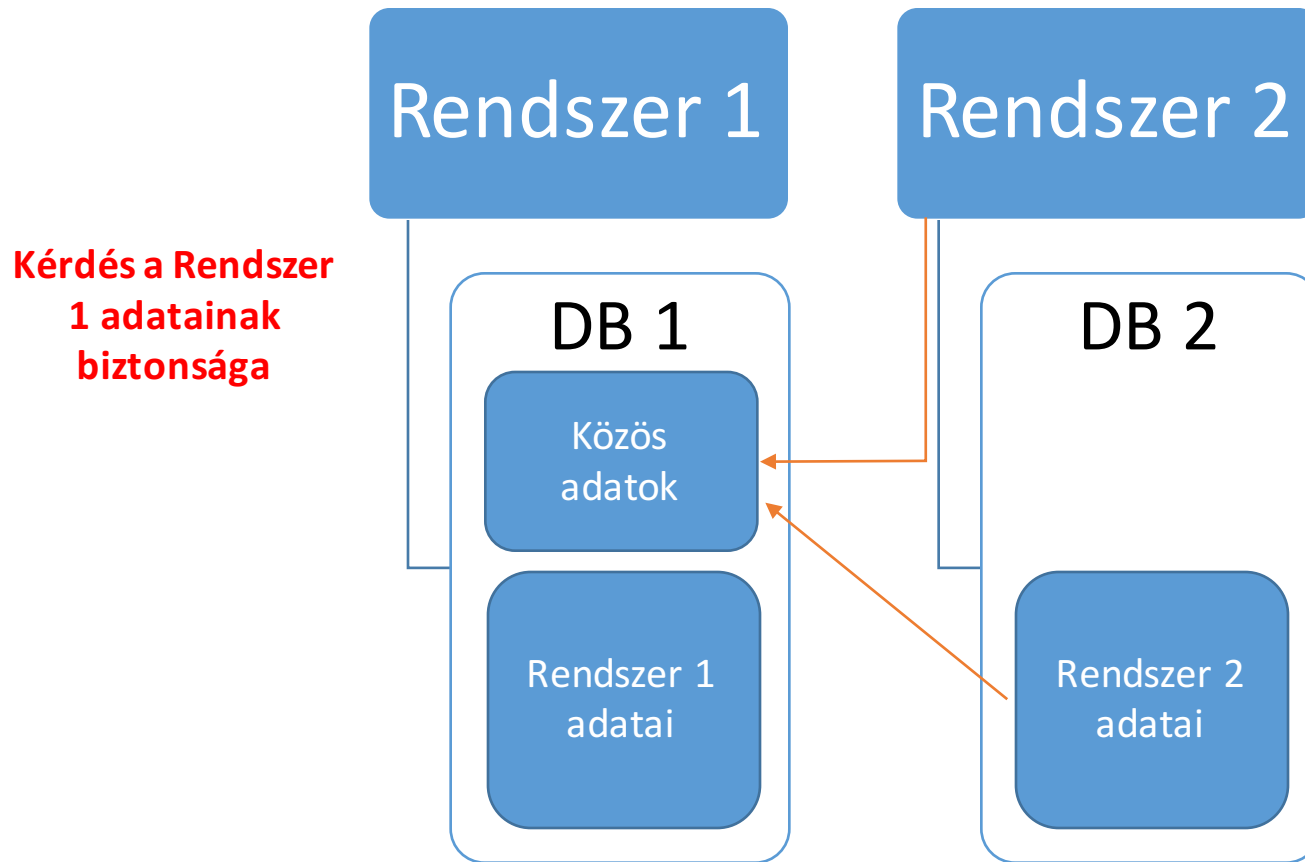
- Előnye: kisebb terhelés (hálózat rendszerek)
- Hátránya: a két rendszer adatai a szinkronizálás közötti időintervallumban eltérnek egymástól

# Közös adatok az egyik rendszeren belül



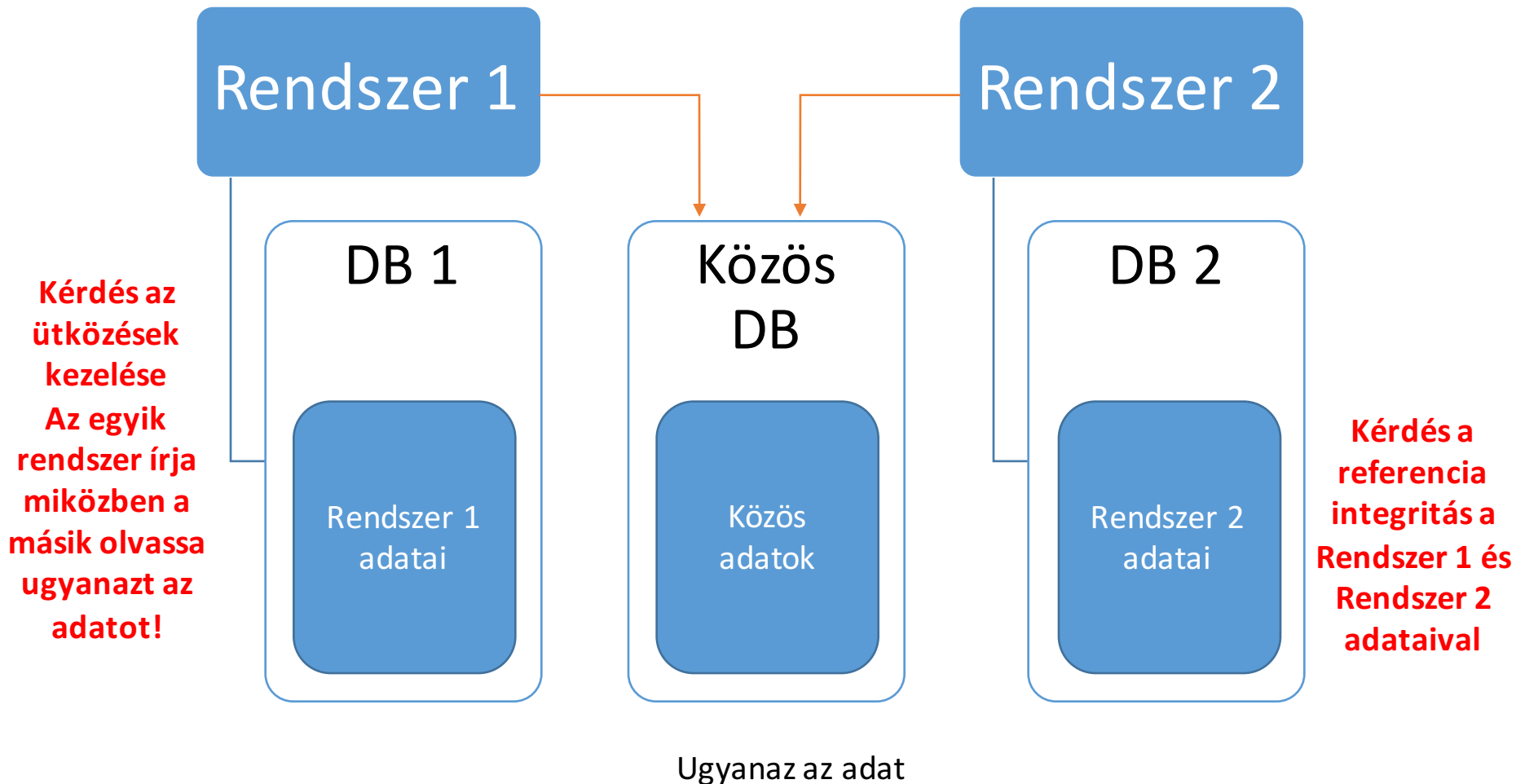
Ugyanaz az adat

# Közös adatok az egyik rendszeren belül

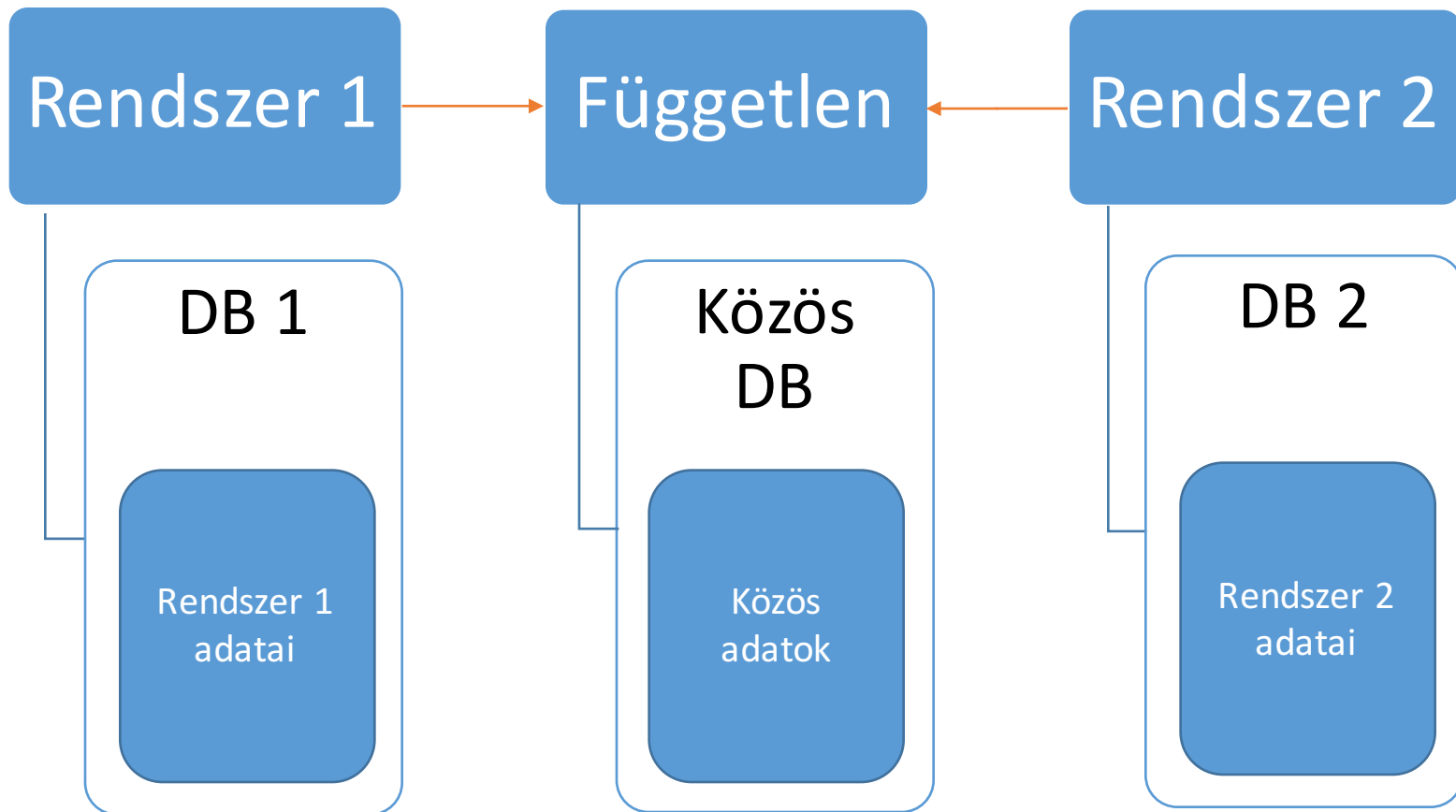


Ugyanaz az adat

# Közös adatok független helyen



# Közös adatok független rendszerben



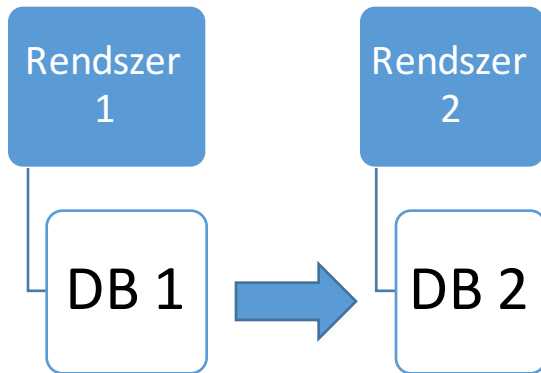
Ugyanaz az adat

**Kérdés a referencia integritás  
a Rendszer 1 és Rendszer 2  
adataival**

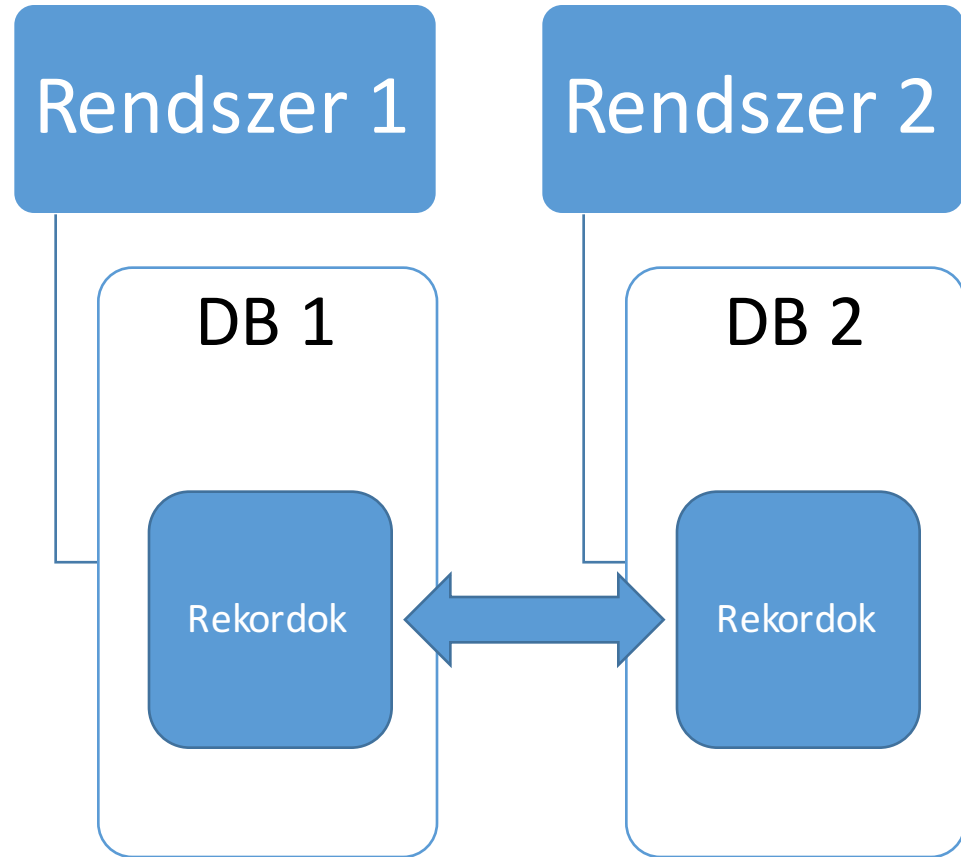


# Szinkronizálás

**Kérdés a szinkronizálás átfutási ideje**



Rendszer szintű szinkronizálás



Rekord szintű szinkronizálás

**Kérdés az ütközések kezelése**

# Késleltetett szinkronizálás

Előnyök

Hátrányok

Az adatok sérülése az elsődleges rendszerben nem terjed tovább azonnal

Hálózati kapcsolat megszakadása nem okoz problémát

Lehetőséget nyújt az adatrekordok kötegelt átemelésére

Kisebb hálózati terhelés

Az adatok mennyiségétől függően pillanatnyi terhelést okozhat a rendszerekben

A rendszerek adatai a szinkronizálások közötti időintervallumban eltérnek egymástól

# Master Data Management (MDM)

## Törzsadatkezelés

### Törzsadat

- Minden, a vállalat különböző alkalmazásaiban megtalálható, munkafolyamatot támogató adat, amely ritkán változik.
- Pl. dolgozók, szállítók, partnerek, termékek, alkatrészek, telephelyek stb. nyilvántartása

### Törzsadat kezelés

- A törzsadatkezelés (MDM) alkalmazásokat, információkezelési módszertanokat és adatkarbantartó eszközöket használ, hogy szabályokat, eljárásokat és infrastruktúrát biztosítson konzisztens, naprakész, hiteles és teljes körű törzsadatok biztosításához (rögzítés, gyűjtés, integráció)

# Master Data Management (MDM)

## Törzsadatkezelés

### Célja

- A törzsadatkezelés legfőbb célja, lehetővé tenni a munkatársak számára az egyes törzsadatok egyértelmű azonosítását a vállalaton belül

### Egységes rögzítés, felhasználás, megjelenítés feltételei

- Törzsadatnyilvántartás létrehozása:
  - amely elégséges információt tartalmaz az egyes törzsadat előfordulások egyedi azonosításához
  - kapcsolódási pontot képez az egyes törzsadat előfordulásokhoz

# Kapcsolat az üzleti folyamatokkal

## Fókuszban az információ-menedzsment

- Az MDM megoldáskínálat az információmenedzsmentre teszi a hangsúlyt
- A teljesebb MDM megoldások sokkal inkább fókuszálnak az üzleti folyamatokra és a megoldás alkalmazására

## Folyamat centrikusság

- Egy-egy törzsadat típushoz több üzleti folyamat kapcsolódik, adott esetben ezek száma jelentős
- Az egyes üzleti folyamatok üzletáganként változhatnak és adott esetben jelentősen eltérnek egymástól
- Minél inkább üzletág specifikus egy törzsadathoz kapcsolódó üzleti folyamat, annál rugalmasabb megoldásra van szükség

# Törzsadatok tárolása

## Tárolási lehetőségek

- Egy közös törzsadatkezelő rendszerben?
- Több tranzakciókezelő rendszerben?
- Egy minden rendszer által támogatott/töltött adattárházban

## Miért fontos ez a kérdés?

- A kiépítendő interfészek mennyisége és technológiája befolyásolja az integrációt
- Különösen fontos ez a kérdés, azon törzsadatok esetében, amelyek több különböző rendszerbe leképezett folyamatot támogatnak
- Sokszor történelmi okokra vezethető vissza az adatok tárolásának helye

# Adatok tárolása egy helyen

## Közösen használt cél adattár/alkalmazás

- Naprakész
- Konzisztens
- Szabványos interfész

## Kijelölt alkalmazás, csatlakozó alkalmazások

- Adattárház?
- Egyedi interfészek?
- Osztott rendszerben?

# Adatok tárolása egy helyen

## Megválaszolandó kérdések

- Rendelkezésre állás?
- Elérhetőség?
- Redundancia?
- Egy pontos sérülékenység?
- Helyi gyorsítótárazás?
- Szinkron vagy aszinkron adatelérés?



# Adatok tárolása több helyen

## Több tranzakciós adatbázisban

- Redundancia
- Rendelkezésre állás
- Szükséges meghatározni azokat a rendszereket amelyek az adatok kezelését végzik

## Megválaszolandó kérdések

- Naprakész biztositása rendszerenként?
- Hitelesség?
- Konzisztencia biztositása?

# Adatok birtoklása, hozzáférési jogok

## Ki rendelkezik az adatok felett?

- Az adatot birtokló szervezet vezetője rendszerint az adatgazda
- Az adatgazda a szervezet belső szabályzatai alapján határozza meg az adatokhoz hozzáféréssel rendelkező felhasználók jogosultságait és hozzáférési módjait

## Jogosultságok kiosztása

- A jogosultságok karbantartását az adatgazda által kijelölt személyek (rendszergazdák, alkalmazásgazdák) végzik az adatgazda írásos hozzájárulása alapján

# Törzsadatkezelés

## Interfészek

- Mely alkalmazások kérhetik le az adatokat?
- Az alkalmazások milyen hozzáférési modellt használnak?
- Szinkron vagy aszinkron hozzáférés?

## Alkalmazás jogosultságok

- Mely alkalmazások módosíthatják az adatokat a mester adatbázisban?
- Több alkalmazás esetén hogyan történik az adatok frissítése?

# Törzsadatkezelés

## Adatok karbantartása

- Az adatok karbantartását az adatgazda által kijelölt személyek végzik
- Ha az adatok több rendszerben is megtalálhatók szükséges meghatározni azt a rendszert amely az adatok kezelését végzi az adatkonzisztencia biztosítása érdekében
- Nyomon kell követni, hogy az egyes rendszerek szinkronban tartják-e a törzsadatokat a mester adatbázissal

# Bevezetés (1)

## Felderítés és dokumentálás

- A lényeges üzleti adatok felderítése, dokumentálása és modellezése
- Az adatforrások felderítése
- A metaadatok definiálása

## Végrehajtás

- Kezdjük a legfontosabb célterülettel és definiáljuk
- Egy IT architektúra tervezőnek össze kell állítania egy MDM architektúrát, amely:
  - Illeszkedik a szervezet MDM elképzeléséhez és céljához
  - Illeszkedik a szervezet meglévő architektúrához

# Bevezetés (2)

## Elemzés

- A célterület hiteles adatforrásainak felderítése
- Adatfolyam kiértékelés
- Módosítási szabályok meghatározása
- Metaadat pontosítása
- Törzsadat minőségi követelmények meghatározása.

## Végrehajtás

- Fontos szempont, a létrehozott és működő adatmenedzsment program képviselőjének részvétele
- A legkritikusabb feladat a bevezetésben
- Iteratív feladat, sok munkaterület képviselőjének a részvételére van szükség

# Bevezetés (3)

## Kialakítás - felépítés

- Az MDM adatbázis felépítése a megtervezett architektúrának megfelelően

## Bevezetés

- Az adatbázis feltöltése az első terület törzsadataival és metaadataival (ETL – Extract, Transform, Load)
- Hozzáférési jogok kialakítása és implementálása
- Változáskezelési folyamat kialakítása
- Adatminőségi szintek meghatározása

# Bevezetés (4)

## Üzemeltetés, soron következő adatok

- Változáskezelés kialakítása és bevezetése az első interfészre
- A soron következő adatkör létrehozásának tervezése
- Az első adatkörök megvalósításához hasonlóan soron következő adatkörök felépítése mindaddig, amíg a teljes MDM bevezetési program be nem fejeződik



Köszönöm a figyelmet!