MYSQL HALADÓ SZINT

Táblák Összekapcsolása: A JOIN Műveletek

Relációs adatok egyesítése, halmazelméleti logikák és az INNER, LEFT, RIGHT kulcsszavak gyakorlati alkalmazása.

A RELÁCIÓS MODELL

Miért Daraboljuk az Adatokat?

- **Normalizálás:** Felesleges ismétlődések (redundancia) megszüntetése.
- **Szeparáció:** Egy tábla kizárólag egyetlen logikai egységet ír le.
- **Dinamikus illesztés:** Lekérdezéskor a darabok újraegyesítése.



AZ ÖSSZEKÖTŐ KAPOCS

A Kapcsolat Anatómiája



Vásárlók Tábla

`id (Primary Key)`

Egyedi azonosító.

ILLESZTÉS



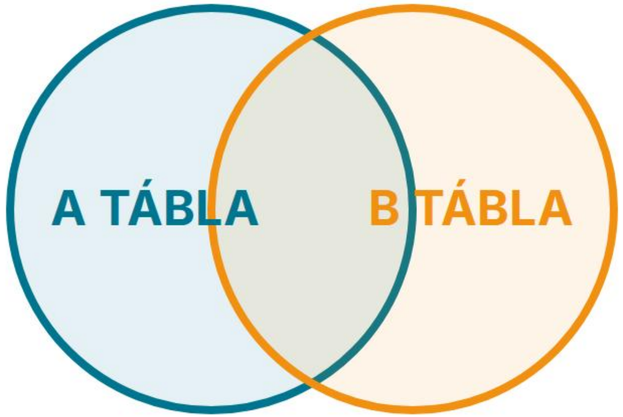
Rendelések Tábla

`vasarlo_id (Foreign Key)`

Hivatkozás a szülőre.

VIZUÁLIS LOGIKA

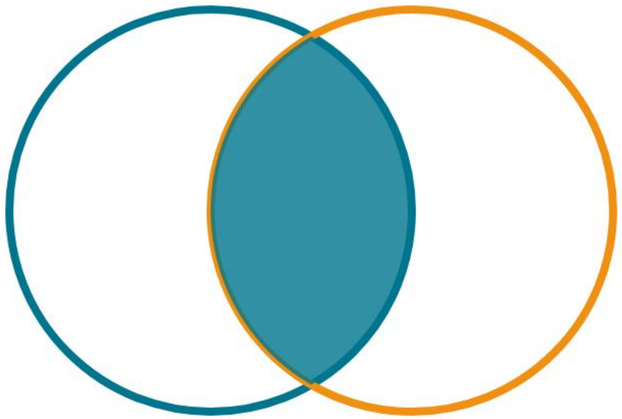
SQL és a Halmazelmélet



A JOIN műveletek matematikai halmazokként kezelik a táblákat. A metszetek és uniók határozzák meg a kimenetet.

A LEGGYAKORIBB TÍPUS

INNER JOIN: A Közös Metszet



$$A \cap B$$

Kizárólag a sikeresen párosított rekordok.

Mindkét táblában léteznie kell az egyezésnek.

SZINTAXIS ÉS GYAKORLAT

INNER JOIN a Lekérdezésekben

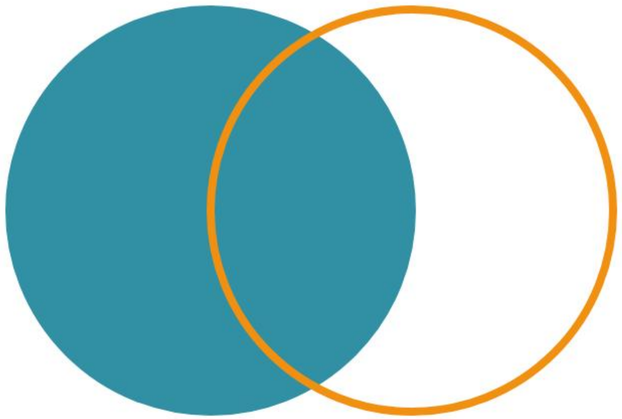
```
-- Vásárlók és leadott rendeléseik összekötése
SELECT vasarlok.nev, rendelesek.datum
FROM vasarlok
INNER JOIN rendelesek
ON vasarlok.id = rendelesek.vasarlo_id;
```

Főbb Szabályok:

- ✓ Az `ON` záradék definiálja a logikai hidat.
- ✓ A `JOIN` szó önmagában is `INNER JOIN`-t jelent.
- ✗ Rendelés nélküli vásárló nem jelenik meg.

A BAL OLDAL DOMINANCIÁJA

LEFT (OUTER) JOIN



A

Minden bal oldali rekord.

Párosítatlan jobb oldalon NULL érték.

HIÁNYZÓ ADATOK KEZELÉSE

LEFT JOIN a Gyakorlatban

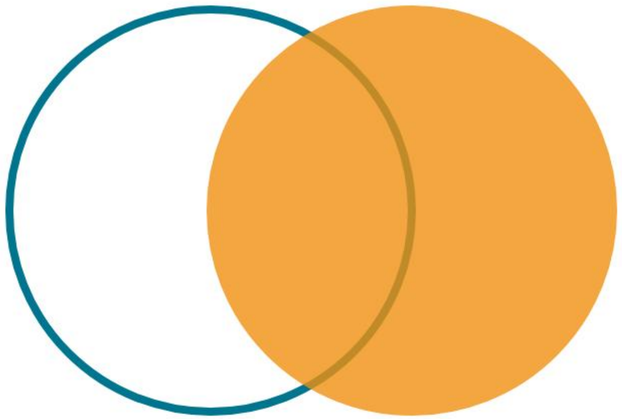
```sql
-- Minden vásárló listázása, rendeléstől függetlenül
SELECT v.nev, r.datum
FROM vasarlok v
LEFT JOIN rendelesek r
ON v.id = r.vasarlo_id;
```

Mikor Használjuk?

- → Inaktív ügyfelek megkeresése.
- → Készleten lévő, sosem eladott termékek.
- ⚠ Eredményben rengeteg NULL mező keletkezhet.

A JOBB OLDAL DOMINANCIÁJA

RIGHT (OUTER) JOIN



B

Minden jobb oldali rekord.

Párosítatlan bal oldalon NULL érték. Ritkán használt.

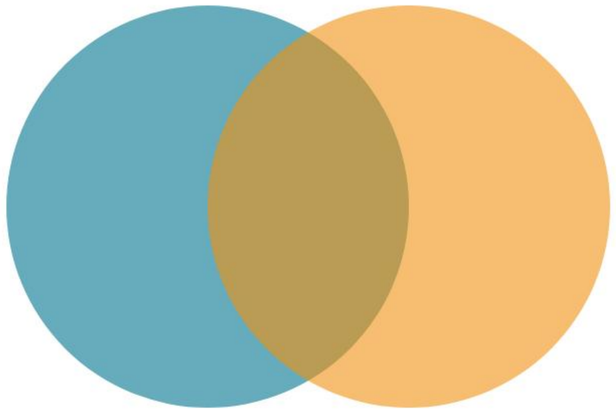
ÖSSZEGZÉS

A Három Alaptípus Összehasonlítása

Művelet	Bal Tábla (FROM)	Jobb Tábla (JOIN)
`INNER JOIN`	Csak párosított.	Csak párosított.
`LEFT JOIN`	**ÖSSZES REKORD.**	Párosított vagy NULL.
`RIGHT JOIN`	Párosított vagy NULL.	**ÖSSZES REKORD.**

A TELJES UNIÓ

FULL OUTER JOIN (MySQL Trükk)



$A \cup B$

Minden rekord mindkét táblából.

A MySQL nem támogatja natívan. LEFT JOIN és RIGHT JOIN egyesítése UNION operátorral.

A KÁOSZ RECEPTJE

CROSS JOIN: Descartes-szorzat

$$A \times B$$

Minden sor kombinálása minden sorral.

Nincs logikai híd (ON záradék hiányzik). 100 vásárló és 100 rendelés esetén 10 000 rekordot generál. Kockázatos.

TISZTA KÓD

Alias (AS): Táblanevek Rövidítése

Alias Nélkül (Hosszú)

```
SELECT vasarlok.nev, rendelesek.id
FROM vasarlok
JOIN rendelesek
ON vasarlok.id = rendelesek.vasarlo_id;
```

Alias Használatával (Tiszta)

```
SELECT v.nev, r.id
FROM vasarlok v
JOIN rendelesek r
ON v.id = r.vasarlo_id;
```

LÁNCOLT KAPCSOLATOK

Több Mint Két Tábla

```sql
SELECT v.nev, r.datum, t.megnevezes
FROM vasarlok v
JOIN rendelesek r ON v.id = r.vasarlo_id
JOIN termekek t ON r.termek_id = t.id;
```



ZÁRÓ ÁTTEKINTÉS

Melyiket Mikor Használjuk?

INNER JOIN

Alapértelmezett választás. Csak a létező, szigorú kapcsolatokhoz.

LEFT JOIN

Fő tábla megőrzése. Hiányzó adatok (NULL) felkutatásához.

Image Sources



https://i.etsystatic.com/44053442/r/il/d2a4c5/6172739006/il_fullxfull.6172739006_1ae5.jpg

Source: www.etsy.com