

Relációs algebra

- T_1 és T_2 táblák direkt szorzata: $T_1 \times T_2$
 - `SELECT ... FROM T_1, T_2`
- T táblára alkalmazott szelekció (f kifejezésnek eleget tevő sorok leválogatása): $\sigma_f(T)$
 - `SELECT ... WHERE f`
- T táblára alkalmazott projekció (J -ben levő oszlopok leválogatása): $\Pi_J(T)$
 - `SELECT m_1, m_2`

Példa:

V: vasarlok(vkod, vnev, vnem(F/N))

A: arucikk(akod, anev, ear)

F: forgalom(vkod, akod, menny, datum, sszam)

1. Mit jelent a $\Pi_J(\sigma_f(V \times A \times F))$ relációs algebrai kifejezés, ha
 $f: A.akod = F.akod \text{ .AND. } V.vkod = F.vkod$
 $J: vnev, anev, menny*ear$
megoldás: vnev, anev, menny*ear oszlopokat tartalmazó táblát, melyben csak a valóban összetartozó rekordok jelennek meg
2. Írjuk fel azt a relációs algebrai kifejezést, mely a következő táblát adja: Azon árucikkek nevei és egységárai, melyeket 1996. decemberében női vásárló vett
megoldás: $\Pi_J(\sigma_f(V \times A \times F))$, ahol
 $f: A.akod = F.akod \text{ .AND. } V.vkod = F.vkod \text{ .AND. } vnem = 'N' \text{ .AND. } datum \geq 1996.12.01$
 $\text{ .AND. } datum \leq 1996.12.31$
 $J: anev, ear$

SQL lekérdezések bevezetés

SELECT ... FROM ... WHERE ...
GROUP BY ... HAVING ORDER BY ...

aruk (akod, nev)

forgalom (akod, menny)

1. **SELECT * FROM** aruk, forgalom
2. **SELECT** nev, menny **FROM** aruk, forgalom
3. **SELECT** nev, menny **FROM** aruk, forgalom **WHERE** aruk.akod=forgalom.akod
4. **SELECT** nev, menny **FROM** aruk, forgalom **WHERE** aruk.akod=forgalom.akod
ORDER BY nev
5. **SELECT** nev, **SUM(menny)** **FROM** aruk, forgalom **WHERE** aruk.akod=forgalom.akod
HIBÁS!!! Meg kell modnani, hogy hogyan csoportosítsunk! Helyesen:
6. **SELECT** nev, **SUM(menny)** **FROM** aruk, forgalom **WHERE** aruk.akod=forgalom.akod
GROUP BY aruk.akod, nev
Mindenkit be kell venni a csoportosításba, aki nem aggregátból származik (itt: nev)!
7. **SELECT** nev, **SUM(menny)** **FROM** aruk, forgalom **WHERE** aruk.akod=forgalom.akod
GROUP BY aruk.akod, nev **HAVING** **SUM(menny)**>20