

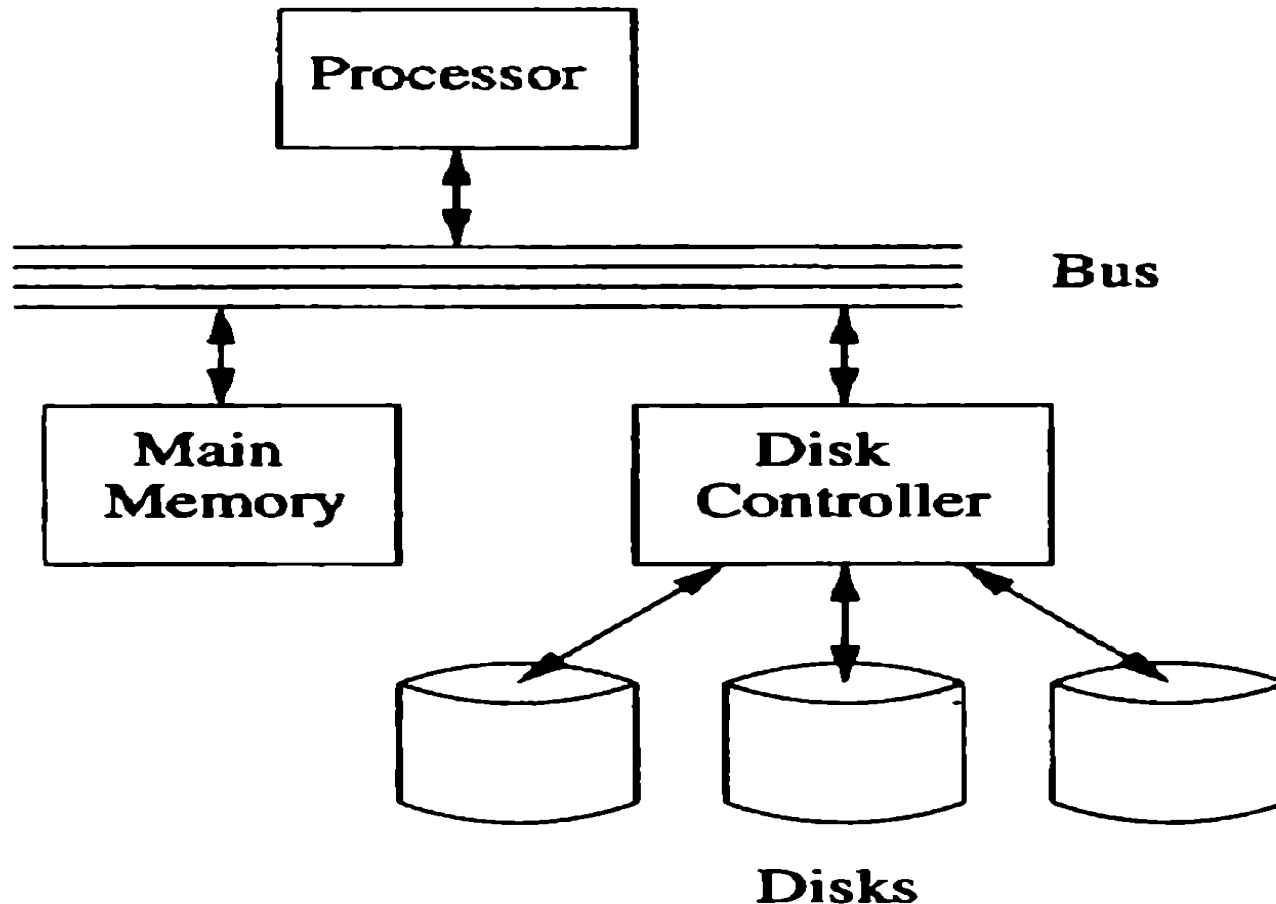
# Relációs algebra lekérdezések optimalizációja

Adatbázisok használata

# Mi a cél?

- **Moore-törvénye:** (Gordon Moore) szerint az integrált áramkörök sok jellemzőjének fejlődése exponenciális, ezek az értékek 18 havonta duplázódnak. Ilyenek például:
  - (i) processzorok sebességének és árának aránya,
  - (ii) lemez egy bitre eső ára és a lemezen tárolható bájtok száma.
- Más paraméterek azonban sokkal lassabban fejlődnek. Ilyenek például:
  - (i) központi memóriában milyen gyorsan lehet az adatokat elérni,
  - (ii) az a sebesség, amellyel a lemez mozog.
- Emiatt egy-egy nagy adathalmazmal dolgozó algoritmus optimalizációjánál az a lényeges szempont, hogy a feladatot minél kevesebb adatmozgatással tudjunk megoldani a háttértároló és a központi memória között.

# Számítógép rendszer sematikus ábrája



# Egy lehetséges megközelítés

- Az adatbázisoknál az előbbiek nyilván úgy értendők, hogy szeretnénk minél kevesebb **lemez olvasási és írási** (I/O) műveletet végrehajtani egy-egy lekérdezés végrehajtása során.
- Az legegyszerűbb megközelítés, ha **igyekszünk minél kisebb méretű relációkkal dolgozni**.
- Az optimalizáció során **relációs algebrai azonosságokat** fogunk alkalmazni. Ezek segítségével egy lekérdezésből az eredetivel ekvivalens lekérdezést készítünk, amelynek kiszámítása az esetek többségében kevesebb I/O műveletet igényel majd.
- A  $q$ ,  $q'$  relációs algebrai lekérdezések (vagy tetszőleges lekérdezések) **ekvivalensek**, ha tetszőleges  $I$  előfordulás esetén  $q(I) = q'(I)$  fennáll. Jelben:  **$q \equiv q'$** .

## Egy példa...

- A táblák legyenek:

Film (cím, év, hossz)

Szerepel (filmcím, év, színésznév)

- Ekkor a következő lekérdezés:

$\Pi_{\text{cím}}(\sigma_{\text{cím=filmcím} \wedge \text{F.év}=\text{Sz.év} \wedge \text{színésznév}=\text{'Edus'}} (F \times Sz))$  ekvivalens a

$\Pi_{\text{cím}}(\sigma_{\text{cím=filmcím} \wedge \text{F.év}=\text{Sz.év}} (F \times (\sigma_{\text{színésznév}=\text{'Edus'}} (Sz))))$  lekérdezéssel.

- Emellett az utóbbi valószínűleg gyorsabban végrehajtható.

# Descartes-szorzat és összekapcsolások

- Asszociativitás:

$(E_1 \Theta E_2) \Theta E_3 \equiv E_1 \Theta (E_2 \Theta E_3)$ , ahol  $\Theta \in \{\times, |X|\}$  és

$(E_1 |X|_{F1} E_2) |X|_{F2} E_3 \equiv E_1 |X|_{F1} (E_2 |X|_{F2} E_3)$ , ha

$\text{attr}(F1) \subseteq \text{attr}(E1) \cup \text{attr}(E2)$  és  $\text{attr}(F2) \subseteq \text{attr}(E2) \cup \text{attr}(E3)$

- Kommutativitás:

$E_1 \Theta E_2 \equiv E_2 \Theta E_1$ , ahol  $\Theta \in \{\times, |X|, |X|_F\}$ .

# Projekció és szelekció

- Projekció sorozat:

$$\Pi_X(\Pi_Y(E)) \equiv \Pi_X(E), \text{ ha } X \subseteq Y.$$

- Kiválasztás és a feltételek konjunkciója:

$$\sigma_{F_1 \wedge F_2}(E) \equiv \sigma_{F_1}(\sigma_{F_2}(E)).$$

- Kiválasztás és a feltételek diszjunkciója:

$$\sigma_{F_1 \vee F_2}(E) \equiv \sigma_{F_1}(E) \cup \sigma_{F_2}(E).$$

- Kiválasztás elé projekció beillesztése:

$$\Pi_X(\sigma_F(E)) \equiv \Pi_X(\sigma_F(\Pi_Y(E))), \text{ ahol } Y = \text{attr}(F) \cup X.$$

# Kiválasztás és Descartes-szorzat/összekapcsolás

- Kiválasztás és Descartes-szorzat, összekapcsolás felcserélése:  
 $\sigma_F (E_1 \Theta E_2) \equiv \sigma_F (E_1) \Theta E_2$ , ahol  $\text{attr}(F) \subseteq \text{attr}(E_1)$  és  $\Theta \in \{\times, |X|\}$ .
- Általánosabban:  
 $\sigma_F (E_1 \Theta E_2) \equiv \sigma_{F_1} (E_1) \Theta \sigma_{F_2} (E_2)$ , ahol  $\text{attr}(F_i) \subseteq \text{attr}(E_i)$  ( $i = (1, 2)$ )  
 $F = F_1 \wedge F_2$  és  $\Theta \in \{\times, |X|\}$ .
- Ezekből levezethető:  
 $\sigma_F (E_1 \Theta E_2) \equiv \sigma_{F_2} (\sigma_{F_1} (E_1) \Theta E_2)$ , ahol  $\text{attr}(F_1) \subseteq \text{attr}(E_1)$ ,  $F = F_1 \wedge F_2$ ,  
de  $\text{attr}(F_2) \subseteq \text{attr}(E_i)$  nem teljesül ( $i = (1, 2)$ ),  $\Theta \in \{\times, |X|\}$ .



# Projekció és Descartes-szorzat/összekapcsolás

- Projekció és Descartes-szorzat, összekapcsolás felcserélése:

$$\Pi_X(E_1 \Theta E_2) \equiv \Pi_Y(E_1) \Theta \Pi_Z(E_2),$$

ahol  $X = Y \cup Z$ ,  $Y \subseteq \text{attr}(E_1)$ ,  $Z \subseteq \text{attr}(E_2)$  és  $\Theta \in \{\times, \bowtie\}$ .

# Projekció/kiválasztás és halmazműveletek

- Kiválasztás és unió (különbség) felcserélése:

$$\sigma_F (E_1 \Theta E_2) \equiv \sigma_F (E_1) \Theta \sigma_F (E_2), \text{ ahol } \Theta \in \{\cup, -\}.$$

- Projekció unióval való felcserélése:

$$\Pi_X(E_1 \cup E_2) \equiv \Pi_X(E_1) \cup \Pi_X(E_2).$$

- **Megjegyzés:** nincs általános szabály a projekció különbséggel való felcserélésére.
- **Kérdés:** a metszettel mi a helyzet?

# Példa optimalizálásra

- A következő két feladathoz használt táblák:

Személy (név, kor, város, ISBN)

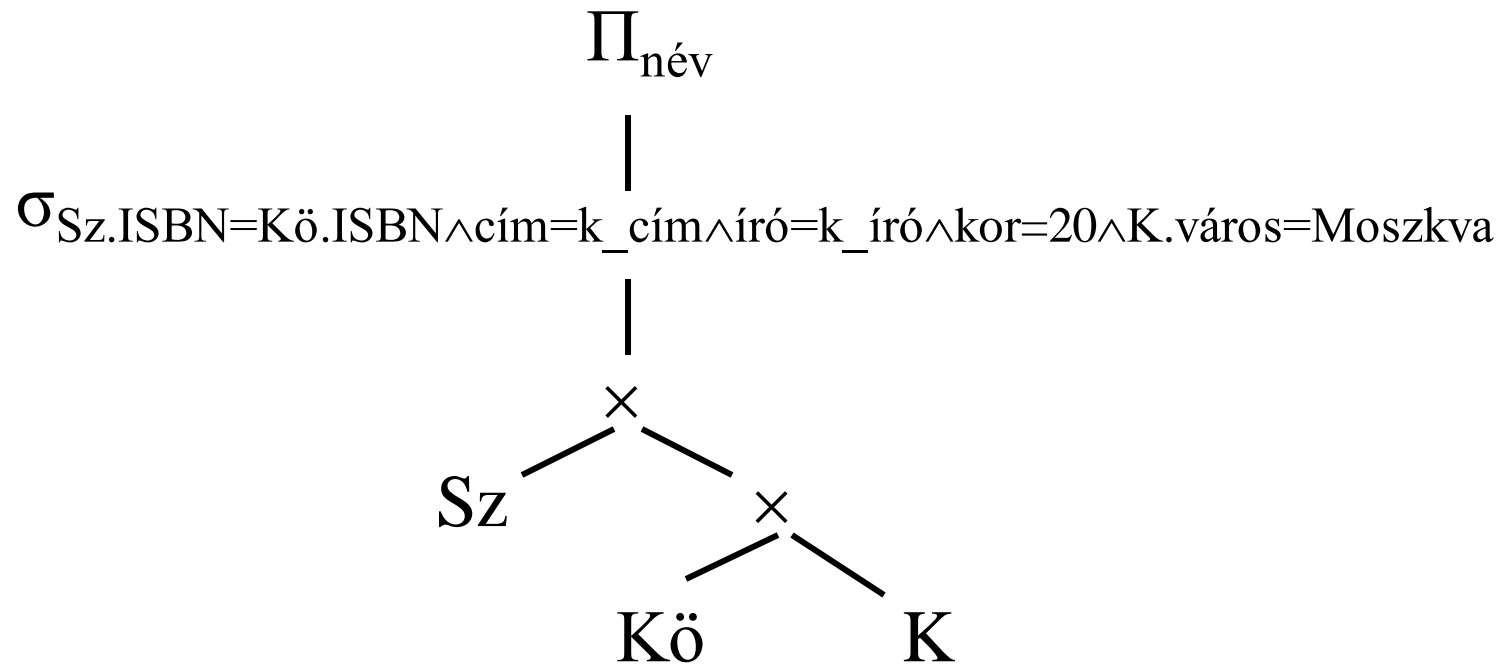
Könyv (cím, író, ISBN, ár)

Kiad (k\_cím, k\_író, város, ország)

- Kik azok, akik 20 évesek, és moszkvai kiadású könyvet kölcsönöztek ki?

$$\Pi_N(\sigma_{S_z.ISBN=Kö.ISBN \wedge cím=k\_cím \wedge író=k\_író \wedge kor=20 \wedge K.város=Moszkva} (S_z \times K_ö \times K))$$

# Lekérdezésfa

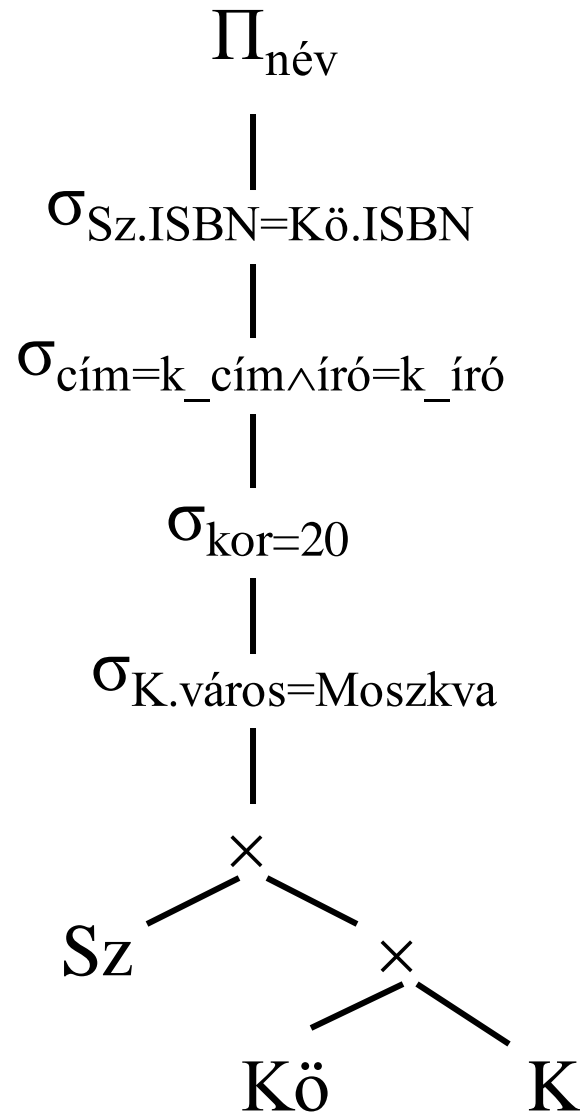


# Kiválasztások "lejjebb csúsztatása"

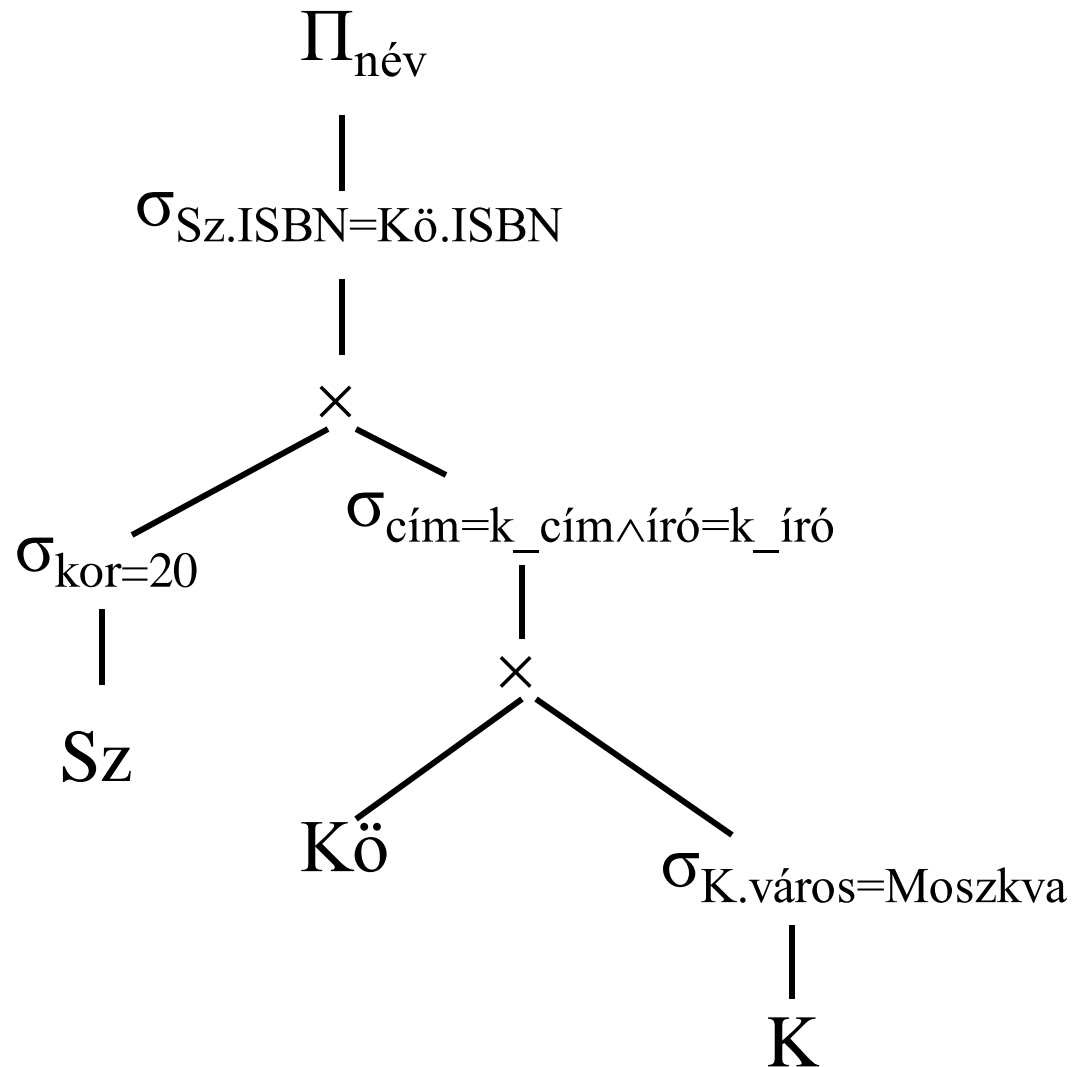
- Első lépésben a kiválasztások **konjunkciós feltételeit daraboljuk szét elemi feltételekké** a  $\sigma_{F_1 \wedge F_2}(E) \equiv \sigma_{F_1}(\sigma_{F_2}(E))$  szabály segítségével.
- Ezek után alkalmazzuk a **kiválasztás halmazműveletekkel illetve Descartes-szorzáttal és a természetes összekapcsolással való felcserélésének szabályait**.
- Azaz: igyekszünk a kiválasztásokat minél hamarabb végrehajtani, hiszen azok jelentősen csökkenthetik a feldolgozandó köztes relációk méretét.
- A Théta-összekapcsolást itt jobb, ha egy Descartes-szorzatra és egy azt követő kiválasztásra bontjuk.

$$R \bowtie_X S \equiv \sigma_F(R \times S).$$

# Darabolás



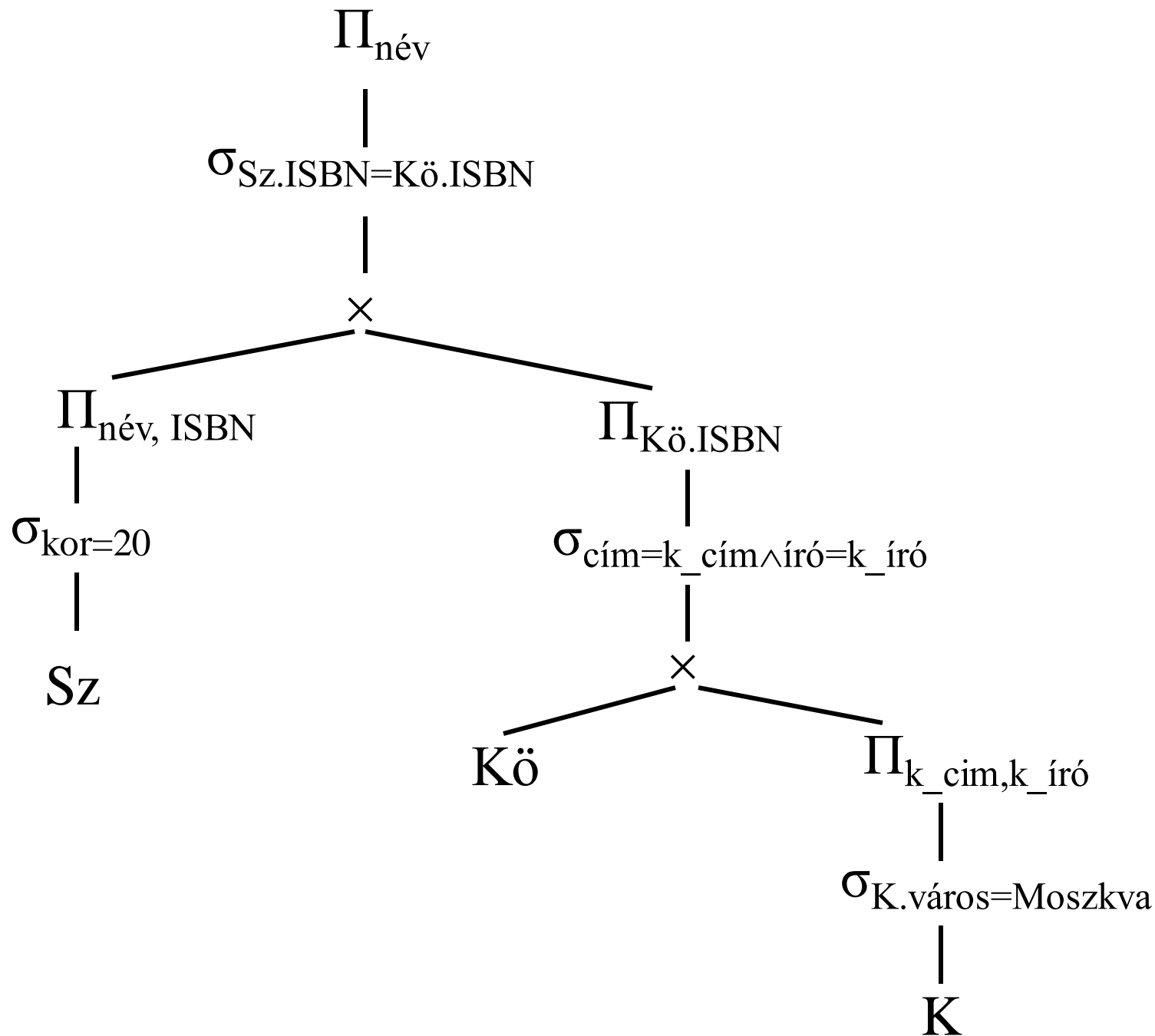
# Letolás



# Projekciók "beírása"

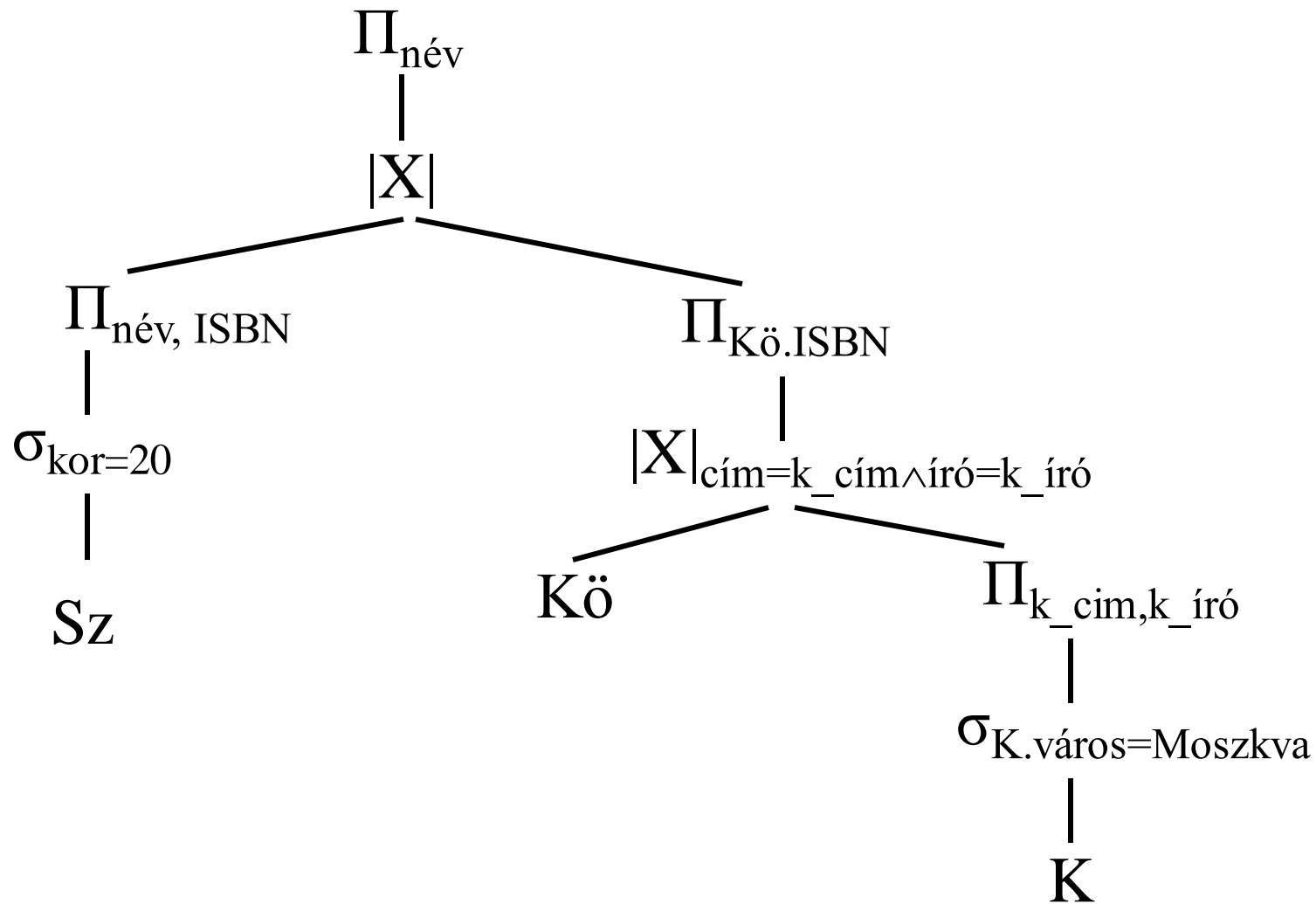
- Ennél a lépésnél igyekszünk csak azokat az oszlopokat megtartani a (köztes) relációkban, amelyekre később szükség lesz.
- **Általában itt nem olyan nagy a nyereség.** A projekciók végrehajtása viszont időt igényel, ezért meg kell gondolni, hogy tényleg végre akarjuk-e hajtani a vetítést.
- Az átalakításoknál értelemszerűen **a projekciókra vonatkozó szabályokat használjuk.**





# Összekapcsolások

- Az utolsó lépésben  $\Pi_L(\sigma_C(R \times S))$ ,  $\sigma_C(R \times S)$  kifejezéseket helyettesítjük természetes összekapcsolással, Théta-összekapcsolással úgy, hogy az eddigivel ekvivalens lekérdezést kapjunk.



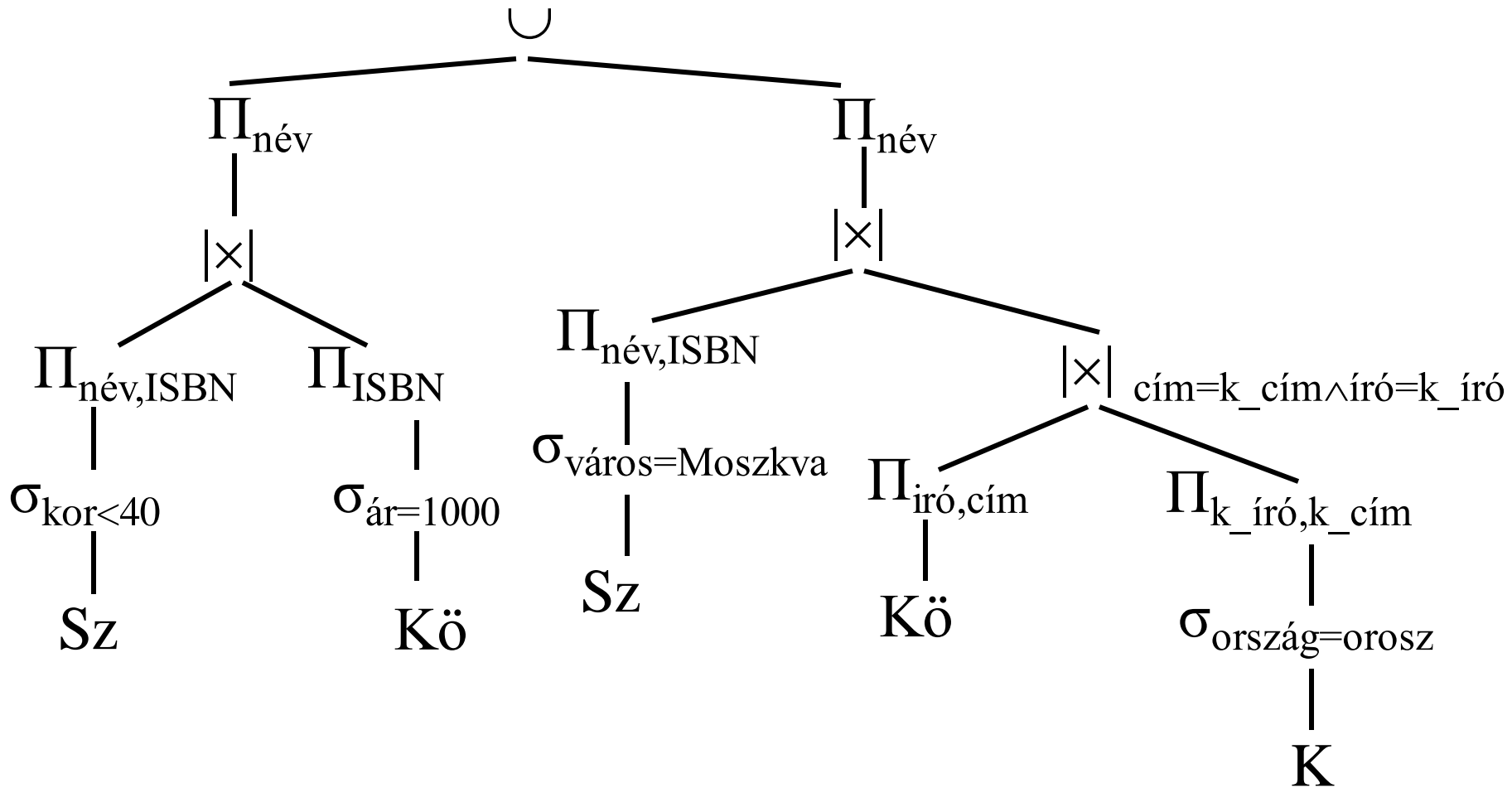
# Mi történik, ha a diszjunkció is megjelenik?

- Kik azok, akik 1000 forintos könyvet vásároltak, és még nincsenek 40 évesek, vagy moszkvaiak, és orosz kiadású könyvet vettek?

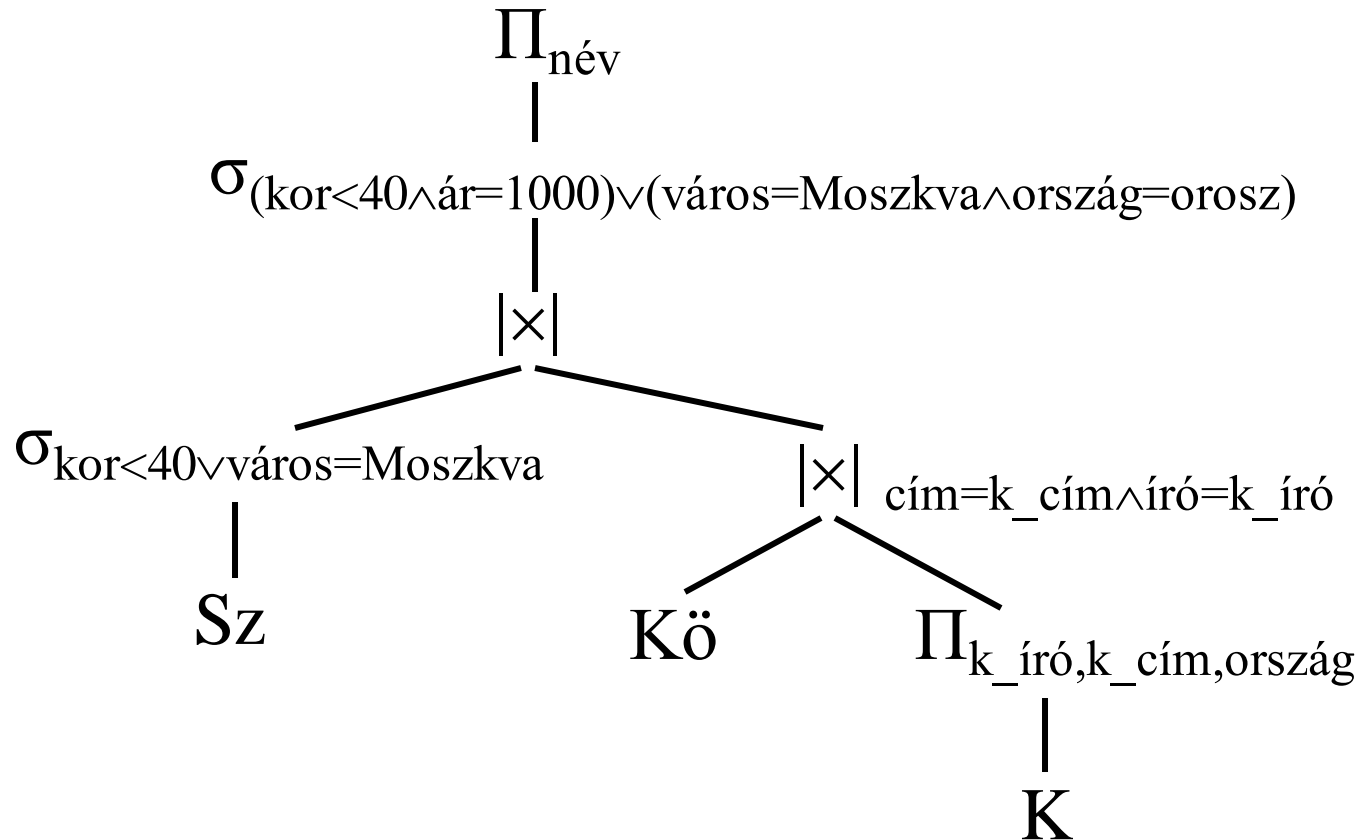
$$\Pi_N(\sigma_{C \wedge ((\text{ár}=1000 \wedge \text{kor} < 40) \vee (\text{Sz.város}=\text{Moszkva} \wedge \text{ország}=\text{orosz}))} (\text{Sz} \times \text{Kö} \times \text{K})).$$

- Itt **C** az  $\text{Sz.ISBN} = \text{Kö.ISBN} \wedge \text{Kö.cím} = \text{K.k\_cím} \wedge \text{Kö.író} = \text{K.k\_író}$  feltételt jelöli.

# Megoldás I.



## Megoldás II.



# Összegzés

- Ha tehát a kiválasztások feltételei diszjunkciót is tartalmaznak, **a helyzet bonyolultabbá válik**, és nem adható olyan egyértelmű optimalizációs algoritmus, mint konjunkciók esetén.

# Kiválasztások feljebb csúsztatása

- A következő példa azt szemlélteti, amikor egy **kiválasztást először felfelé kell csúsztatnunk**, hogy aztán letolhassuk.

- A **táblák:**

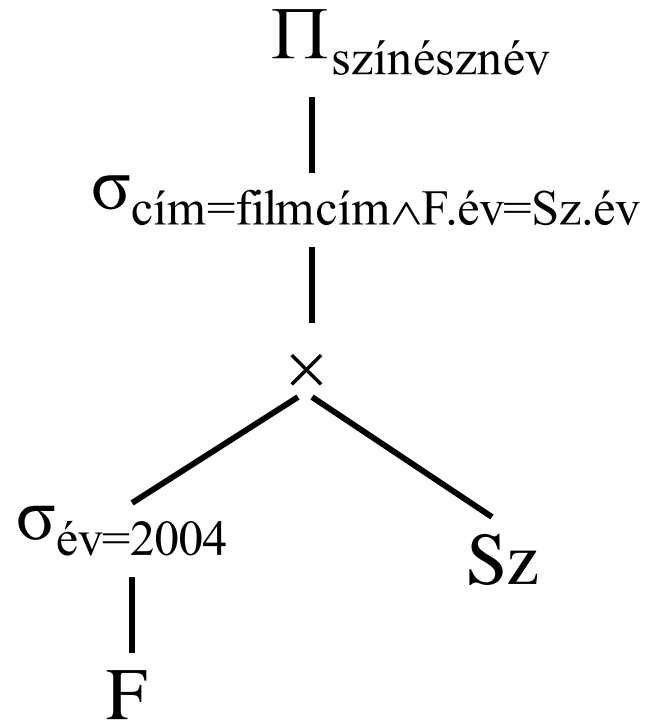
Film (cím, év, hossz)

Szerepel (filmcím, év, színésznév)

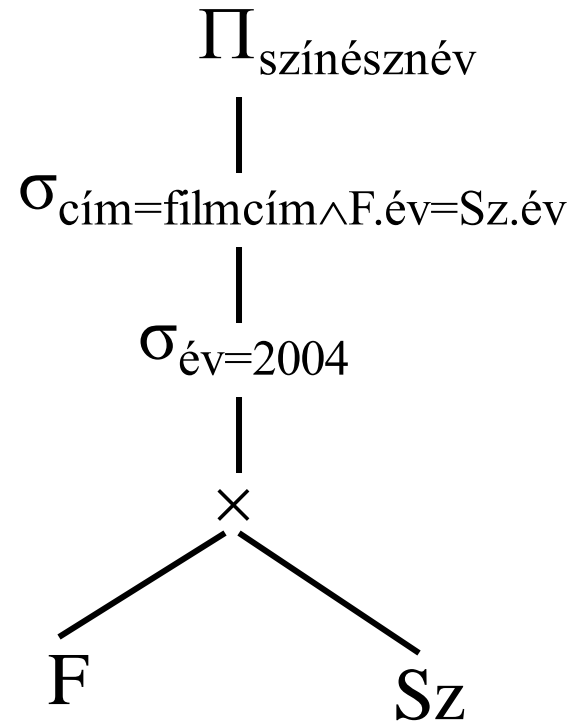
```
CREATE VIEW film04 AS  
(SELECT *  
FROM film  
WHERE év = 2004);  
SELECT színésznév  
FROM film04 f, Szerepel sz  
WHERE cím = filmcím AND  
f.év = sz.év;
```



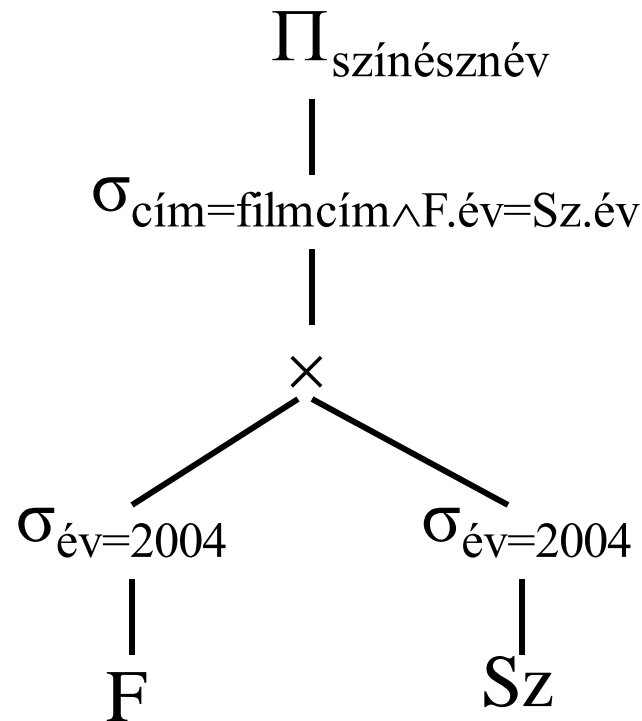
# Kezdeti lekérdezésfa



## Második lépés



# És az eredmény...



# Feladat

- A táblák legyenek:  
Film (cím, év, hossz)  
Szerepel (filmcím, év, színésznev)  
Színész (név, kor, város)
- Adjuk meg, hogy a nem budapesti, negyven évesnél idősebb színészek milyen filmekben játszottak 1998-ban. A lekérdezést optimalizáljuk.