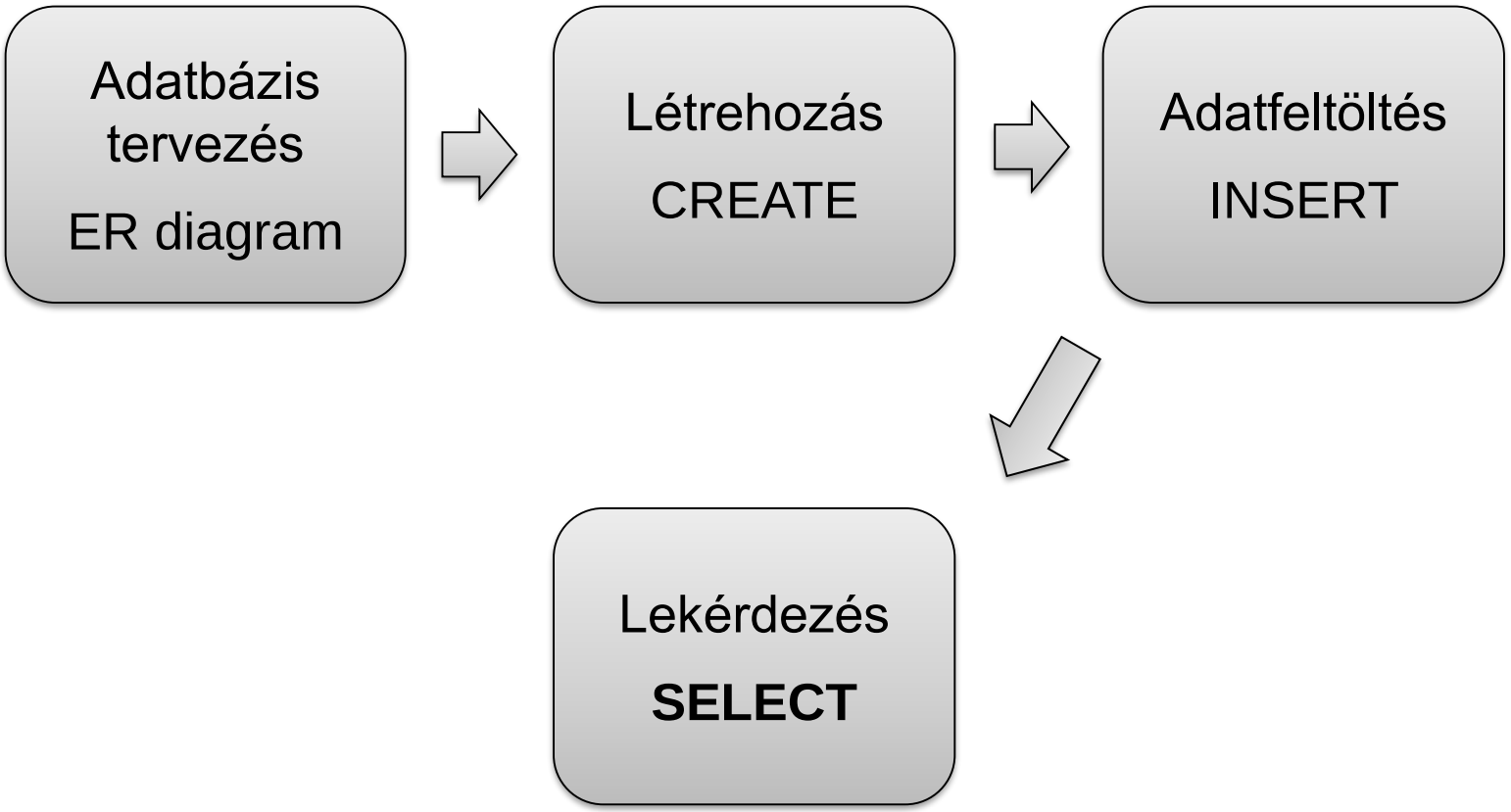
Lekérdezések

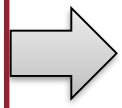
- Relációs algebra
- SQL lekérdezések

Cél


Adatbázis tervezés
ER diagram
Létrehozás
CREATE
Adatfeltöltés
INSERT
Lekérdezés
SELECT

Vetítés (projekció)

A1	A2	A3	A4
1	A	X	5
2	B	X	5
3	C	Y	5



A3	A4
X	5
X	5
Y	5

zsák

A3	A4
X	5
Y	5

halmaz

Azonos sorokkal mit kezdjünk?

Vetítés (projekció)

- Cél: a számunkra érdekes **oszlopok** meghagyása, a többi mellőzése
- **Az eredeti relációt nem akarjuk módosítani, hanem egy újat akarunk létrehozni**
- Ennek értéke egy olyan reláció, amely R-nek csak az A_1, A_2...A_n oszlopait tartalmazza
  - Még mindig reláció, tehát az egyes rekordok egyediek

Vetítés (projekció) – SQL

- Szintaxis:
  - **SELECT** A_1, A_2, ..., A_n **FROM** <táblanév>
  - Használhatók aritmetikai műveletek is

Vetítés (projekció) – SQL

- **SELECT** Cim, Mufaj **FROM** Film

Id	Studióld	Cím	Év	Hossz	Műfaj
1	1	Psycho	1960	1:49	Horror
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	2007	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

Cím	Műfaj
Psycho	Horror
Tremors	Comedy
The Simpsons Movie	Animation
Rambo 2	Action
Taken	Action

Vetítés (projekció) – SQL

- **SELECT** Mufaj **FROM** Film

Id	Studióld	Cím	Év	Hossz	Műfaj
1	1	Psycho	1960	1:49	Horror
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	2007	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

Műfaj
Horror
Comedy
Animation
Action
Action

Vetítés (projekció) – SQL

- **SELECT** * **FROM** Film

Id	Stúdióld	Cím	Év	Hossz	Műfaj
1	1	Psycho	1960	1:49	Horror
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	2007	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

Id	Stúdióld	Cím	Év	Hossz	Műfaj
1	1	Psycho	1960	1:49	Horror
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	2007	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

Vetítés (projekció) – SQL

- **SELECT** Ev-1000 **FROM** Film

Id	Studióld	Cím	Év	Hossz	Műfaj
1	1	Psycho	1960	1:49	Horror
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	2007	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

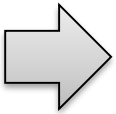
Év
960
990
1007
985
1009

Kiválasztás (szelekció)

		Év	
		1910	
		2007	
		1956	
		1998	

+ szűrési feltétel

(pl. Év < 1990)



		Év	
		1910	
		1956	

Kiválasztás (szelekció)

- Cél: a számunkra érdekes **sorok** meghagyása, a többi mellőzése
- **Az eredeti relációt nem akarjuk módosítani, hanem egy újat akarunk létrehozni**
- **Kényszer** (constraint): a reláció attribútumaira fogalmaz meg feltételt, a szokásos operátorokat és kifejezéseket tartalmazhatja (illetve függvényeket, ld később)
  - Aritmetikai (> = < + - * /)
  - Logikai (^ v !)
- az R reláció minden elemére kiértékelődik, és csak amelyikre teljesül, az kerül bele az eredményhalmazba

Kiválasztás (szelekció) – Példa

- FILM (Id, StudióId, Cím, Év, Hossz, Műfaj)

Id	StudióId	Cím	Év	Hossz	Műfaj
1	1	Psycho	1960	1:49	Horror
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	2007	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

- 2000 utáni filmek

Id	StudióId	Cím	Év	Hossz	Műfaj
3	3	The Simpsons Movie	2007	1:27	Animation
5	3	Taken	2009	1:33	Action

Kiválasztás (szelekció) – Példa – 2

- Mely akciófilmek készültek 2000 után?

Id	Studióld	Cím	Év	Hossz	Műfaj
5	3	Taken	2009	1:33	Action

- Az összes akció és horror film

Id	Studióld	Cím	Év	Hossz	Műfaj
1	1	Psycho	1960	1:49	Horror
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

Kiválasztás (szelekció) – SQL

- Szintaxis:
  - **SELECT** <oszlopok> **FROM** <táblanév> **WHERE** C
  - Operátorok:
    - AND
    - OR
    - <>
    - NOT
    - (...)
  - Operandusok:
    - Oszlopok
    - Konstansok

Kiválasztás (szelekció) – SQL

- **SELECT * FROM** Film **WHERE** Év > 2000

Id	Studióld	Cím	Év	Hossz	Műfaj
1	1	Psycho	1960	1:49	Horror
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	2007	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

Id	Studióld	Cím	Év	Hossz	Műfaj
3	3	The Simpsons Movie	2007	1:27	Animation
5	3	Taken	2009	1:33	Action

Kiválasztás (szelekció) – SQL

- **SELECT** * **FROM** Film

  WHERE Ev > 2000 **AND** Mufaj=’Action’

Id	Studióld	Cím	Év	Hossz	Műfaj
1	1	Psycho	1960	1:49	Horror
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	2007	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

Id	Studióld	Cím	Év	Hossz	Műfaj
5	3	Taken	2009	1:33	Action

Kiválasztás (szelekció) – SQL

- **SELECT * FROM** Film
  WHERE Mufaj=’Horror’ **OR** Mufaj=’Action’

Id	StudióId	Cím	Év	Hossz	Műfaj
1	1	Psycho	1960	1:49	Horror
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	2007	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

Id	StudióId	Cím	Év	Hossz	Műfaj
1	1	Psycho	1960	1:49	Horror
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

Kiválasztás (szelekció) – SQL

- **SELECT** * **FROM** Film **WHERE** Mufaj <> ’Horror’
- **SELECT** * **FROM** Filem **WHERE NOT** (Mufaj = ’Horror’)

Id	Studióld	Cím	Év	Hossz	Műfaj
1	1	Psycho	1960	1:49	Horror
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	2007	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

Id	Studióld	Cím	Év	Hossz	Műfaj
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	2007	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

Kiválasztás és vetítés

- FILM (Id, StudioId, Cim, Év, Hossz, Mufaj)

Id	StudioId	Cim	Ev	Hossz	Mufaj
1	1	Psycho	1960	1:49	Horror
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	2007	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

- Mi a 2000 után gyártott filmek címe?
  - Nem felcserélhető műveletek, először a kiválasztást hajtjuk végre

Cim
The Simpsons Movie
Taken

```
SELECT Cim FROM Film WHERE Ev > 2000
```

Háromértékű logika – NULL

- NULL: egy érték hiányzik (nem ismerjük, nem értelmezett)
- Háromértékű logika
  - Aritmetikai művelet egyik operandusa NULL => eredmény is
    - (3+ x) = NULL, ha x=NULL
  - Összehasonlítás egyik oldala NULL => eredmény **UNKNOWN**
    - (3 > x) = UNKNOWN, (3 = x) = UNKNOWN, ha x=NULL
    - NULL vizsgálatra „**IS NULL**”, „**IS NOT NULL**” (=NULL ill. <>NULL helyett)

Háromértékű logika – NULL

- Logikai operátorok

AND	False	UNK	True
False	False	False	False
UNK	False	UNK	UNK
True	False	UNK	True

OR	False	UNK	True
False	False	UNK	True
UNK	UNK	UNK	True
True	True	True	True

NOT	
False	True
UNK	UNK
True	False

Pl. UNKOWN AND TRUE = UNKNOWN

Háromértékű logika – NULL – Példa

- FILM (Id, StudioId, Cim, Ev, Hossz, Mufaj)

Id	StudioId	Cim	Ev	Hossz	Mufaj
1	1	Psycho	NULL	1:49	Horror
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	NULL	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

- `SELECT * FROM Film WHERE Ev < 1990 OR Ev >= 1990`
  - `SELECT * FROM Film WHERE Ev < 1990 OR NOT (Ev < 1990)`
  - Csak azok, ahol **True-ra** értékelődik ki

Id	StudioId	Cim	Ev	Hossz	Mufaj
2	1	Tremors	1990	1:36	Comedy
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

Háromértékű logika – NULL – Példa – 2

- FILM (Id, StudioId, Cim, Ev, Hossz, Mufaj)

Id	StudioId	Cim	Ev	Hossz	Mufaj
1	1	Psycho	NULL	1:49	Horror
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	NULL	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

- Filmek, ahol van évszám?
  - SELECT * FROM Film WHERE Ev **IS NOT NULL**
- Filmek, ahol nincs évszám, vagy ’90 utáni ?
  - SELECT * FROM Film WHERE Ev **IS NULL OR Ev > 1990**

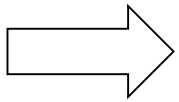
NULL kezelés

- Probléma: mi van, ha null helyett mást akarunk látni egy lekérdezésben?
  - Pl az adat hiánya azt jelzi, hogy 0?
  - Pl a vevőnél ha a cím nincs kitöltve, akkor azt szeretnénk látni, hogy ’<ismeretlen>’
- Megoldás:
  - **IFNULL (<érték>, <érték NULL helyett>) (MySQL)**

NULL kezelés – 2

- Vevő adatai, üres cím helyett ’<ismeretlen>’
  - **SELECT** Nev, **IFNULL**(Cim, ’<ismeretlen>’) **FROM** Vevo

Id	Nev	Cím
1	Kovács Béla	1111 Budapest...
2	Kiss István	NULL
3	Nagy Anna	9000 Győr ...



Nev	Cím
Kovács Béla	1111 Budapest...
Kiss István	**<ismeretlen>**
Nagy Anna	9000 Győr ...

Halmazműveletek

Halmazműveletek

- Operátorok
  - Unió
  - Metszet
  - Különbség
- Feltétel:
  - A két reláció attribútumai azonosak (azonos értelmezési tartomány)
  - A két reláció attribútumai azonos sorrendben vannak

Unió – SQL

- Szintaxis:

```
(SELECT ...) UNION (SELECT ...)
```

Unió – SQL

- Studio(Id, Nev, Cim), Szinesz(Id, Nev, Cim, Szuletesnap)
- Valamennyi filmstúdió és színész címe?
  - **SELECT** Cim **FROM** Studio **UNION** **SELECT** Cim **FROM** Szinesz

Unió – SQL – 2

- Ha ugyanaz az adatforrás, akkor nem célszerű használni
  - Pl azok a filmek, amelyek műfaja vagy horror vagy vígjáték
    - **SELECT** * **FROM** Film **WHERE** Mufaj = 'Horror' **UNION**
      SELECT * **FROM** Film **WHERE** Mufaj = 'Comedy'
  - Előállítja külön mindkét relációt, majd eltávolítja az egyezőket. => pazarlás
  - Elég egy relációt elkészíteni, rögtön az eredményt:
    - **SELECT** * **FROM** Film **WHERE** Mufaj='Horror' **OR** Mufaj='Comedy'
    - => egyszer kell a rekordokat átnézni, nem kétszer + összehasonlítani a végén, hogy a duplikátumokat kiszűrjük
  - => általában kiküszöbölhető **OR**-ral

Halmaz vs zsák – Unió

- Műveletek zsákokon
  - Unió: A U B => B összes rekordját A-hoz fűzzük
    - Ha r rekord n-szer szerepel A-ban, és m-szer B-ben, akkor (n+m)-szer fog szerepelni A U B-ben
    - SQL : külön operátor a kétfajta unióra:
      - **UNION**: a duplikátumokat eliminálja
      - **UNION ALL** : csak utána fűzi (pl ha tudom, hogy nincs duplikátum)

A
1
2

B
1
3

A UNION B
1
2
3

A UNION ALL B
1
2
1
3

Metszet

- Cél: azok az entitások, amelyek mindkét relációban szerepelnek, mindegyik csak 1x

Metszet – SQL

- Studio (Id, Nev, Cim), Film(Id, StudioId, Cim, Mufaj)
- Melyek azok a stúdiók, amelyek gyártottak horrot és vígjátékot is?
  - MSSQL, ORACLE:

```
SELECT StudioId FROM Film WHERE Mufaj = 'Horror'
INTERSECT
SELECT StudioId FROM Film WHERE Mufaj = 'Comedy'
```

  - MySQL: Nincs megfelelő operátor, máshogy kell megoldani

Különbség

- Cél: azok az entitások, amelyek az egyik relációban szerepelnek, de a másikban nem
- Jelölés: A – B
  - Nem ugyanaz, mint B – A !

Különbség – SQL

- Studio (Id, Nev, Cim), Film(Id, StudioId, Cim, Mufaj)
- Melyek azok a stúdiók, akikhez egy film sem tartozik?
  - MSSQL : **EXCEPT**
    - **SELECT** Id **FROM** Studio **EXCEPT** **SELECT** StudioId **FROM** Film
  - ORACLE: **MINUS**
    - **SELECT** Id **FROM** Studio **MINUS** **SELECT** StudioId **FROM** Film
  - MySQL: nincs implementáció, máshogy kell megoldani (pl join vagy beágyazott lekérdezéssel, lsd később)

Különbség – SQL – 2

- Ha ugyanaz az adatforrás, akkor nem célszerű használni
  - Pl azok a filmek, amelyek 1980 után készültek, de nem 2000 után
    - **SELECT** * **FROM** Film **WHERE** Ev > 1980 **EXCEPT**
      SELECT * **FROM** Film **WHERE** Ev > 2000
  - Előállítja külön mindkét relációt, majd eltávolítja az egyezőket. => pazarlás
  - Elég egy relációt elkészíteni, rögtön az eredményt:
    - **SELECT** * **FROM** Film **WHERE** Ev > 1980 **AND** Ev <= 2000
    - => egyszer kell a rekordokat átnézni, nem kétszer + összehasonlítani a végén
  - => általában kiküszöbölhető

Halmaz vs Zsák – halmazműveletek

- Metszet: A ⋂ B => a közös rekordok közül a kisebb darabszámú
  - Ha r rekord n-szer szerepel A-ban, és m-szer B-ben (lehet 0 is), akkor min (n,m)-szer fog szerepelni A ⋂ B-ben
- Különbség: A – B => A rekordjai közül eltávolítjuk azokat (és annyit), ahány B-ben szerepel
  - Ha r rekord n-szer szerepel A-ban, és m-szer B-ben, akkor max(0, (n-m))-szer fog szerepelni A – B -ben
- **!DE: SQL: halmaz alapú specifikáció alapján működik (duplikáció eliminálással), nincs külön operátor, mint az UNIÓhoz**

Átnevezés

- Gyakran szükséges, hogy a relációnknak kontrollálhassuk a nevét illetve az attribútumainak a nevét

Átnevezés – Példa

- FILM (Id, StudioId, Cim, Ev, Hossz, Mufaj)

Id	StudioId	Cim	Ev	Hossz	Mufaj
1	1	Psycho	1960	1:49	Horror
2	1	Tremors	1990	1:36	Comedy
3	3	The Simpsons Movie	2007	1:27	Animation
4	2	Rambo 2	1985	1:36	Action
5	3	Taken	2009	1:33	Action

- Film() → Horror()

Id	StudioId	Cim	Ev	Hossz	Mufaj
1	1	Psycho	1960	1:49	Horror

- Film(Cím, Hossz)
  - → Movie(Title, Length)

Title	Length
Psycho	1:49
Tremors	1:36
The Simpsons Movie	1:27
Rambo 2	1:36
Taken	1:33

Átnevezés – SQL

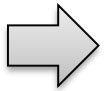
- Kulcsszó: [**AS**]
  - SELECT A1 **AS** B1, A2 **AS** B2, … An **AS** Bn FROM <table>
- Pl. **SELECT** Cim **AS** Title, Hossz **AS** Length **FROM** Film
  - **AS** nélkül: **SELECT** Cim Title, Hossz Length **FROM** Film
- Reláció átnevezés: ld később
  - Join műveletek
  - Beágyazott lekérdezés

Keresztszorzat (Descartes-szorzat)

A	B	C
A1	B1	C1
A2	B2	C3

X

X	Y
X1	Y1
X2	Y2
X3	Y3



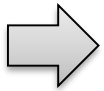
A	B	C	X	Y
A1	B1	C1	X1	Y1
A1	B1	C1	X2	Y2
A1	B1	C1	X3	Y3
A2	B1	C2	X1	Y1
A2	B2	C2	X2	Y2
A2	B2	C2	X3	Y3

Keresztszorzat (Descartes-szorzat)

A	B	C
A1	B1	C1
A2	B2	C3

X

X	Y
X1	Y1
X2	Y2
X3	Y3



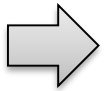
A	B	C	X	Y
A1	B1	C1	X1	Y1
A1	B1	C1	X2	Y2
A1	B1	C1	X3	Y3
A2	B1	C2	X1	Y1
A2	B2	C2	X2	Y2
A2	B2	C2	X3	Y3

Keresztszorzat (Descartes-szorzat)

A	B	C
A1	B1	C1
A2	B2	C3

X

X	Y
X1	Y1
X2	Y2
X3	Y3



A	B	C	X	Y
A1	B1	C1	X1	Y1
A1	B1	C1	X2	Y2
A1	B1	C1	X3	Y3
A2	B1	C2	X1	Y1
A2	B2	C2	X2	Y2
A2	B2	C2	X3	Y3

Keresztszorzat (Descartes-szorzat)

- Két reláció keresztszorzata: az egyik reláció minden elemét összepárosítjuk a másik reláció minden elemével
- Jelölés : A x B

Keresztszorzat (Descartes-szorzat)

- PL:
  - FILM (FilmId, StudioId, Cim, Ev)

FilmId	StudioId	Cim	Ev
1	1	Psycho	1960
2	1	Tremors	1990
3	3	The Simpsons Movie	2007
4	2	Rambo 2	1985
5	3	Taken	2009

  - Studio(StudioId, Nev)

StudioId	Nev
1	Universal
2	Columbia Pictures
3	20th Century Fox

Keresztszorzat – 2

- Film x Studio

FilmId	StudioId	Cim	Ev	StudioId	Nev
1	1	Psycho	1960	1	Universal
1	1	Psycho	1960	2	Columbia Pictures
1	1	Psycho	1960	3	20th Century Fox
2	1	Tremors	1990	1	Universal
2	1	Tremors	1990	2	Columbia Pictures
2	1	Tremors	1990	3	20th Century Fox
3	3	The Simpsons Movie	2007	1	Universal
3	3	The Simpsons Movie	2007	2	Columbia Pictures
3	3	The Simpsons Movie	2007	3	20th Century Fox
4	2	Rambo 2	1985	1	Universal
4	2	Rambo 2	1985	2	Columbia Pictures
4	2	Rambo 2	1985	3	20th Century Fox
5	3	Taken	2009	1	Universal
5	3	Taken	2009	2	Columbia Pictures
5	3	Taken	2009	3	20th Century Fox

Keresztszorzat – 3 – Névütközés

- Névütközés esetén átnevezünk

FilmId	StudioId	Cim	Ev	SID	Nev
1	1	Psycho	1960	1	Universal
1	1	Psycho	1960	2	Columbia Pictures
1	1	Psycho	1960	3	20th Century Fox
2	1	Tremors	1990	1	Universal
2	1	Tremors	1990	2	Columbia Pictures
2	1	Tremors	1990	3	20th Century Fox
3	3	The Simpsons Movie	2007	1	Universal
3	3	The Simpsons Movie	2007	2	Columbia Pictures
3	3	The Simpsons Movie	2007	3	20th Century Fox
4	2	Rambo 2	1985	1	Universal
4	2	Rambo 2	1985	2	Columbia Pictures
4	2	Rambo 2	1985	3	20th Century Fox
5	3	Taken	2009	1	Universal

Keresztszorzat – SQL

- Kulcsszó: **CROSS JOIN**
  - SELECT A1, A2, ..., An, B1, B2, ..., Bn FROM A **CROSS JOIN** B
  - Alternatív: SELECT A1, A2, ..., An, B1, B2, ..., Bn FROM A**,** B

Keresztszorzat - SQL

- **SELECT * FROM** Film, Studio

FilmId	StudioId	Cim	Ev

StudioId	Nev
1	Universal

FilmId	StudioId	Cim	Ev	StudioId	Nev
1	1	Psycho	1960	1	Universal
1	1	Psycho	1960	2	Columbia Pictures
1	1	Psycho	1960	3	20th Century Fox
2	1	Tremors	1990	1	Universal
2	1	Tremors	1990	2	Columbia Pictures
2	1	Tremors	1990	3	20th Century Fox
3	3	The Simpsons Movie	2007	1	Universal
3	3	The Simpsons Movie	2007	2	Columbia Pictures
3	3	The Simpsons Movie	2007	3	20th Century Fox
4	2	Rambo 2	1985	1	Universal
4	2	Rambo 2	1985	2	Columbia Pictures
4	2	Rambo 2	1985	3	20th Century Fox
5	3	Taken	2009	1	Universal
5	3	Taken	2009	2	Columbia Pictures
5	3	Taken	2009	3	20th Century Fox

Keresztszorzat - SQL

- **SELECT** Cim, Nev **FROM** Film, Studio

FilmId	StudioId	Cim	Ev
1	1	Psycho	1960
2	1	Tremors	1990
3	3	The Simpsons Movie	2007
4	2	Rambo 2	1985
5	3	Taken	2009

StudioId	Nev
1	Universal
2	Columbia Pictures
3	20th Century Fox

Cim	Nev
Psycho	Universal
Psycho	Columbia Pictures
Psycho	20th Century Fox
Tremors	Universal
Tremors	Columbia Pictures
Tremors	20th Century Fox
The Simpsons Movie	Universal
The Simpsons Movie	Columbia Pictures
The Simpsons Movie	20th Century Fox
Rambo 2	Universal
Rambo 2	Columbia Pictures
Rambo 2	20th Century Fox
Taken	Universal
Taken	Columbia Pictures
Taken	20th Century Fox

Keresztszorzat - SQL

- **SELECT** Film.*, Studio.StudioId **AS** SID, Nev **FROM** Film, Studio
  - Relációnévvel együtt egyértelműen azonosíthatók az oszlopok

FilmId	StudioId
1	1
2	1
3	3
4	2
5	3

StudioId	
1	
2	Colum
3	20th

FilmId	StudioId	Cim	Ev	SID
1	1	Psycho	1960	1
1	1	Psycho	1960	2
1	1	Psycho	1960	3
2	1	Tremors	1990	1
2	1	Tremors	1990	2
2	1	Tremors	1990	3
3	3	The Simpsons Movie	2007	1
3	3	The Simpsons Movie	2007	2
3	3	The Simpsons Movie	2007	3
4	2	Rambo 2	1985	1
4	2	Rambo 2	1985	2
4	2	Rambo 2	1985	3
5	3	Taken	2009	1
5	3	Taken	2009	2
5	3	Taken	2009	3

Természetes illesztés

- **SELECT * FROM** Film, Studio

FilmId	StudioId			StudioId	Nev
1	1		60	1	Universal
1	1		60	2	Columbia Pictures
1	1	Psycho	1960	3	20th Century Fox
2	1	Tremors	1990	1	Universal
2	1	Tremors	1990	2	Columbia Pictures
2	1	Tremors	1990	3	20th Century Fox
3	3	The Simpsons Movie	2007	1	Universal
3	3	Th	07	2	Columbia Pictures
3	3	Th	07	3	20th Century Fox
4	2		85	1	Universal
4	2	Rambo 2	1985	2	Columbia Pictures
4	2	Rambo 2	1985	3	20th Century Fox
5	3	Taken	2009	1	Universal
5	3	Taken	2009	2	Columbia Pictures
5	3	Taken	2009	3	20th Century Fox

Illeszkedik, mert a két StudioId azonos

Nem illeszkedik, mert a két StudioId különböző

Természetes illesztés

- Cél: a keresztszorzat csak azon elemei, amelyek *„összeillenek”*
  - Pl egyező StudioId
  - Általában: *az azonos nevű oszlopok tartalma legyen azonos*
  - Ezek az oszlopok (mivel úgyis azonos a tartalmuk), csak egyszer jelenjenek meg
- => Természetes illesztés (natural join)

Természetes illesztés – Példa

- Studio(StudioId, Nev)

StudioId	Nev
1	Universal
2	Columbia Pictures
3	20th Century Fox

- FILM (FilmId, StudioId, Cim, Ev)

FilmId	StudioId	Cim	Ev
1	1	Psycho	1960
2	1	Tremors	1990
3	3	The Simpsons Movie	2007
4	2	Rambo 2	1985
5	3	Taken	2009

- Természetes illesztés

FilmId	StudioId	Cim	Ev	Nev
1	1	Psycho	1960	Universal
2	1	Tremors	1990	Universal
3	3	The Simpsons Movie	2007	20th Century Fox
4	2	Rambo 2	1985	Columbia Pictures
5	3	Taken	2009	20th Century Fox

Természetes illesztés – SQL

- Kulcsszó: **NATURAL JOIN** (ORACLE, MySQL)
  - **SELECT** A1, A2, ..., An, B1, B2, ..., Bn **FROM** TableA **NATURAL JOIN** TableB
  - MySQL: nem támogatott (hibalehetőség)

Természetes illesztés - SQL

- **SELECT** Cim, Nev **FROM** Film **NATURAL JOIN** Studio

StudioId	Nev
1	Universal
2	Columbia Pictures
3	20th Century Fox

FilmId	StudioId	Cim	Ev
1	1	Psycho	1960
2	1	Tremors	1990
3	3	The Simpsons Movie	2007
4	2	Rambo 2	1985
5	3	Taken	2009

Cim	Nev
Psycho	Universal
Tremors	Universal
The Simpsons Movie	20th Century Fox
Rambo 2	Columbia Pictures
Taken	20th Century Fox

Természetes illesztés - SQL

- **SELECT** Cim, Nev **FROM** Film **NATURAL JOIN** Studio

  WHERE Ev > 2000

StudioId	Nev
1	Universal
2	Columbia Pictures
3	20th Century Fox

FilmId	StudioId	Cim	Ev
1	1	Psycho	1960
2	1	Tremors	1990
3	3	The Simpsons Movie	2007
4	2	Rambo 2	1985
5	3	Taken	2009

Cim	Nev
The Simpsons Movie	20th Century Fox
Taken	20th Century Fox

Természetes illesztés - SQL

- **SELECT** Film.*, Nev **FROM** Film **NATURAL JOIN** Studio

StudioId	Nev
1	Universal
2	Columbia Pictures
3	20th Century Fox

FilmId	StudioId	Cim	Ev
1	1	Psycho	1960
2	1	Tremors	1990
3	3	The Simpsons Movie	2007
4	2	Rambo 2	1985
5	3	Taken	2009

FilmId	StudioId	Cim	Ev	Nev
1	1	Psycho	1960	Universal
2	1	Tremors	1990	Universal
3	3	The Simpsons Movie	2007	20th Century Fox
4	2	Rambo 2	1985	Columbia Pictures
5	3	Taken	2009	20th Century Fox

Θ – illesztés

- Theta-illesztés: tetszőleges feltétel teljesülése
  - Vesszük a relációk keresztszorzatát
  - Ezek közül kiválogatjuk azokat, amelyek eleget tesznek egy általunk megadott feltételnek

Θ – illesztés – Példa

- Studio(**Id**, Nev)

Id	Nev
1	Universal
2	Columbia Pictures
3	20th Century Fox

- FILM (**Id**, StudioId, Cim, Ev)

Id	StudioId	Cim	Ev
1	1	Psycho	1960
2	1	Tremors	1990
3	3	The Simpsons Movie	2007
4	2	Rambo 2	1985
5	3	Taken	2009

- StudioId legyen azonos

Id	StudioId	Cim	Ev	Id	Nev
1	1	Psycho	1960	1	Universal
2	1	Tremors	1990	1	Universal
3	3	The Simpsons Movie	2007	3	20th Century Fox
4	2	Rambo 2	1985	2	Columbia Pictures
5	3	Taken	2009	3	20th Century Fox

Θ – illesztés – SQL

- Kulcsszó: [INNER] JOIN ... ON
  - **SELECT** A1..An,B1...Bm **FROM** A **INNER JOIN** B **ON** <Condition>
- Több tábla összekapcsolása:
  - **SELECT** A1..An,B1...Bm,C1...Cl **FROM** A
    INNER JOIN B **ON** <Condition1>
    INNER JOIN C **ON** <Condition2> ...

Θ – illesztés – SQL

- **SELECT** * **FROM** Film **INNER JOIN** Studio **ON** StudioId = Studio.Id

StudioId	Nev
1	Universal
2	Columbia Pictures
3	20th Century Fox

FilmId	StudioId	Cim	Ev
1	1	Psycho	1960
2	1	Tremors	1990
3	3	The Simpsons Movie	2007
4	2	Rambo 2	1985
5	3	Taken	2009

FilmId	StudioId	Cim	Ev	StudioId	Nev
1	1	Psycho	1960	1	Universal
2	1	Tremors	1990	1	Universal
3	3	The Simpsons Movie	2007	3	20th Century Fox
4	2	Rambo 2	1985	2	Columbia Pictures
5	3	Taken	2009	3	20th Century Fox

Θ – illesztés – SQL

- **SELECT** * **FROM** Vasarlo **INNER JOIN** Ajandek **ON** Minimum < Pontok
- Milyen ajándékok közül választhat egy vásárló?

Id	Név	Pontok
1	Kovács József	222
2	Kiss Anna	115
3	Nagy Béla	346

Id	Megnevezés	Minimum
1	Csoki	100
2	Üdítő	200
3	Pizza	300

Id	Név	Pontok	Id	Megnevezés	Minimum
1	Kovács József	222	1	Csoki	100
1	Kovács József	222	2	Üdítő	200
2	Kiss Anna	115	1	Csoki	100
3	Nagy Béla	346	1	Csoki	100
3	Nagy Béla	346	2	Üdítő	200
3	Nagy Béla	346	3	Pizza	300

Θ – illesztés – SQL

- **SELECT** Vasarlo.Id, Nev, Megnevezes **AS** Ajandek **FROM** Vasarlo **INNER JOIN** Ajandek **ON** Minimum < Pontok

Id	Név	Pontok
1	Kovács József	222
2	Kiss Anna	115
3	Nagy Béla	346

Id	Megnevezés	Minimum
1	Csoki	100
2	Üdítő	200
3	Pizza	300

Id	Név	Ajándék
1	Kovács József	Csoki
1	Kovács József	Üdítő
2	Kiss Anna	Csoki
3	Nagy Béla	Csoki
3	Nagy Béla	Üdítő
3	Nagy Béla	Pizza

Reláció átnevezés – SQL

- Önhivatkozó join esetén
  - Hogy meg tudjuk különböztetni a relációkat a join két oldalán
  - **SELECT** * **FROM** Alkalmazott **INNER JOIN** Alkalmazott **AS** Felettes **ON** Alkalmazott.FelettesId=Felettes.Id
- Rövidíteni akarunk
  - **SELECT** f.Id, f.Cim, s.Nev **FROM** Film **AS** f **INNER JOIN** Studio **AS** s **ON** f.StudioId=s.Id **WHERE** f.Ev > 2000
- AS szintén elhagyható

Θ – illesztés – Önhivatkozás – 2

- Családfa
  - Ember(Id, Nev, ApaId, AnyaId)
  - Soroljuk fel az összes testvérpárt
    - **SELECT** e1.Id, e1.Nev, e2.Id Id2, e2.Nev Nev2 **FROM** Ember e1 **INNER JOIN** Ember e2 **ON** e1.ApaId=e2.ApaId **OR** e1.AnyaId=e2.AnyaId (!!!NULL)
      - **Join saját magával, és mással mindkét irányban => felesleges**
    - Megoldás: kulcsok összehasonlítása

Id	Nev	ApaId	AnyaId
1	Kovács József	NULL	NULL
2	Kiss Anna	NULL	NULL
3	Kovács Béla	1	2
4	Kovács Anna	1	2

Id	Nev	Id2	Nev2
3	**Kovács Béla**	**3**	**Kovács Béla**
3	Kovács Béla	4	Kovács Anna
4	**Kovács Anna**	**3**	**Kovács Béla**
4	**Kovács Anna**	**4**	**Kovács Anna**

- **SELECT** e1.Id, e1.Nev, e2.Id Id2, e2.Nev Nev2 **FROM** Ember e1 **INNER JOIN** Ember e2 **ON** e1.ApaId=e2.ApaId **OR** e1.AnyaId=e2.AnyaId **WHERE e1.Id < e2.Id**

Külső illesztések

Külső illesztések

- Probléma: a join műveletnél (természetes, theta (inner)) azok a rekordok elvesznek, amelyhez nem tartozik pár a másik relációban.
  - Pl Alkalmazott(Id, Név, FelettesId), kinek ki a felettese => csak azok maradnak meg, akiknek VAN felettese
- Megoldás: külső illesztés
  - Vesszük a Theta-illesztést
  - Hozzávesszük A és B párosítatlan rekordjait, amelyeket NULL-okkal egészítünk ki

Külső illesztés – példa

- Pl. Alkalmazott(Id, Név, FelettesId), kinek ki a felettese? **Az is látszódjon, akinek nincs felettese, beosztottja.**

Id	Név	FelettesId
1	Kovács József	2
2	Kiss Anna	NULL
3	Nagy Béla	2

Id	Név	FelettesId	Id	Név	FelettesId
1	Kovács József	2	2	Kiss Anna	NULL
2	Kiss Anna	NULL	NULL	NULL	NULL
3	Nagy Béla	2	2	Kiss Anna	NULL
NULL	NULL	NULL	1	Kovács József	2
NULL	NULL	NULL	3	Nagy Béla	2

Bal külső illesztés

- Pl. Alkalmazott(Id, Név, FelettesId), kinek ki a felettese?
  Minden alkalmazott egyszer, illetve a hozzá tartozó felettes (ha van).

Id	Név	FelettesId
1	Kovács József	2
2	Kiss Anna	NULL
3	Nagy Béla	2

- Megoldás: bal külső illesztés (left outer join)
  - A természetes illesztéshez hozzávesszük még a **bal**oldali reláció párosítatlan rekordjait is (NULL-lal kiegészíve)

Id	Név	FelettesId	Id	Név	FelettesId
1	Kovács József	2	2	Kiss Anna	NULL
2	Kiss Anna	NULL	NULL	NULL	NULL
3	Nagy Béla	2	2	Kiss Anna	NULL

Jobb külső illesztés

- Pl. Alkalmazott(Id, Név, FelettesId), ki kinek a felettese?
  Minden alkalmazottat soroljunk fel, mellette a hozzá tartozó beosztott

Id	Név	FelettesId
1	Kovács József	2
2	Kiss Anna	NULL
3	Nagy Béla	2

- Megoldás: jobb külső illesztés (right outer join)
  - A természetes illesztéshez hozzávesszük még a **jobb**oldali reláció párosítatlan rekordjait is (NULL-lal feltöltve)

Id	Név	FelettesId	Id	Név	FelettesId
NULL	NULL	NULL	1	Kovács József	2
1	Kovács József	2	2	Kiss Anna	NULL
3	Nagy Béla	2	2	Kiss Anna	NULL
NULL	NULL	NULL	3	Nagy Béla	2

Külső illesztés – SQL

- (Teljes) külső illesztés
  - **SELECT** A1..An,B1...Bm **FROM** A **FULL OUTER JOIN** B **ON** <Condition>
  - MySQL: nem támogatott
- Bal külső illesztés
  - **SELECT** A1..An,B1...Bm **FROM** A **LEFT OUTER JOIN** B **ON** <Condition>
- Jobb külső illesztés
  - **SELECT** A1..An,B1...Bm **FROM** A **RIGHT OUTER JOIN** B **ON** <Condition>

Külső illesztés – SQL – Példa

- (Teljes) külső illesztés
  - **SELECT** * **FROM** Alkalmazott **AS** a **FULL OUTER JOIN** Alkalmazott **AS** f **ON** a.FelettesId = f.Id
  - MySQL: nem támogatott
- Bal külső illesztés
  - Alkalmazottak, és feletteseik
    - **SELECT** * **FROM** Alkalmazott **AS** a **LEFT OUTER JOIN** Alkalmazott **AS** f **ON** a.FelettesId = f.Id
  - Egyedüli beosztottak
    - **SELECT** a1.* **FROM** Alkalmazott **AS** a1 **LEFT OUTER JOIN** Alkalmazott **AS** a2 **ON** a1.FelettesId=a2.FelettesId **WHERE** a2.Id **IS NULL**
- Jobb külső illesztés
  - **SELECT** * **FROM** Alkalmazott **AS** a **RIGHT OUTER JOIN** Alkalmazott **AS** f **ON** a.FelettesId = f.Id

Teljes külső illesztés

- MySQL-ben megpoldható unióval:
  - **SELECT** A1..An, B1..Bm **FROM** A **LEFT OUTER JOIN** B **ON** <condition>
    UNION
    SELECT A1..An, B1..Bm **FROM** A **RIGHT OUTER JOIN** B **ON** <condition>

Külső illesztés – További példák

- Online bolt, ahol nyilvántartjuk a megrendeléseket
  - Vevo(Id, Nev, Cim)
  - Megrendeles(Id, VevoId, Datum, Osszertek)
  - MegrendelesTetel(Id, MegrendelesId, TermekId, Mennyiseg)
  - Termek(Id, Nev, Raktarkeszlet)
- Listázzuk ki az összes megrendelést, és a megrendelő nevét
  - **SELECT** Nev, Megrendeles.* **FROM** Megrendeles **INNER JOIN** Vevo **ON** VevoId=Vevo.Id
    - Minden megrendeléshez egyértelműen tartozik 1 vevő => inner join

Külső illesztés – További példák – 2

- Listázzuk ki az összes vásárló nevét, és a hozzájuk tartozó megrendeléseket
  - **SELECT** Nev, Megrendeles.* **FROM** Vevo **LEFT OUTER JOIN** Megrendeles **ON** VevoId=Vevo.Id
  - **SELECT** Nev, Megrendeles.* **FROM** Megrendeles **RIGHT OUTER JOIN** Vevo **ON** VevoId=Vevo.Id
  - Egy vásárlóhoz nem feltétlen tartozik megrendelés => outer join

Id	Nev	Cím
1	Kovács Béla	1111 Budapest...
2	Kiss István	5000 Szolnok...
3	Nagy Anna	9000 Győr ...

Id	VevoId	Datum	Osszertek
1	1	2012.01.01	1870
2	1	2012.02.02	1680
3	3	2012.03.03	930

Nev	Id	VevoId	Datum	Osszertek
Kovács Béla	1	1	2012.01.01	1870
Kovács Béla	2	1	2012.02.02	1680
Kiss István	NULL	NULL	NULL	NULL
Nagy Anna	3	3	2012.03.03	930

Külső illesztés – További példák – 3

- Listázzuk ki az összes vásárló nevét, és az általuk valaha megrendelt termékek nevét és mennyiségét
  - **SELECT** Vevo.Nev, Termek.Nev **AS** Termek, MegrendelesTetel.Mennyiseg
    FROM Vevo
    LEFT OUTER JOIN Megrendeles **ON** Vevo.Id=Megrendeles.VevoId
    LEFT OUTER JOIN MegrendelesTetel **ON**
    MegrendelesTetel.MegrendelesId = Megrendeles.Id
    LEFT OUTER JOIN Termek **ON** Termek.Id = MegrendelesTetel.TermekId
  - Egy vásárlóhoz nem feltétlen tartozik megrendelés => left outer join

Külső illesztés – További példák – 4

- Vevo

Id	Nev	Cím
1	Kovács Béla	1111 Budapest...
2	Kiss István	5000 Szolnok...
3	Nagy Anna	9000 Győr ...

Termek

Id	Nev	Raktarkeszlet	Ar
1	Alma	50	150
2	Körte	48	280
3	Szilva	63	120

- Megrendeles

Id	VevoId	Datum	Osszertek
1	1	2012.01.01	1870
2	1	2012.02.02	1680
3	3	2012.03.03	930

MegrendelesTetel

MegrendelesId	TermekId	Mennyiseg
1	1	5
1	2	4
2	2	6
3	1	3
3	3	4

- Eredmény:

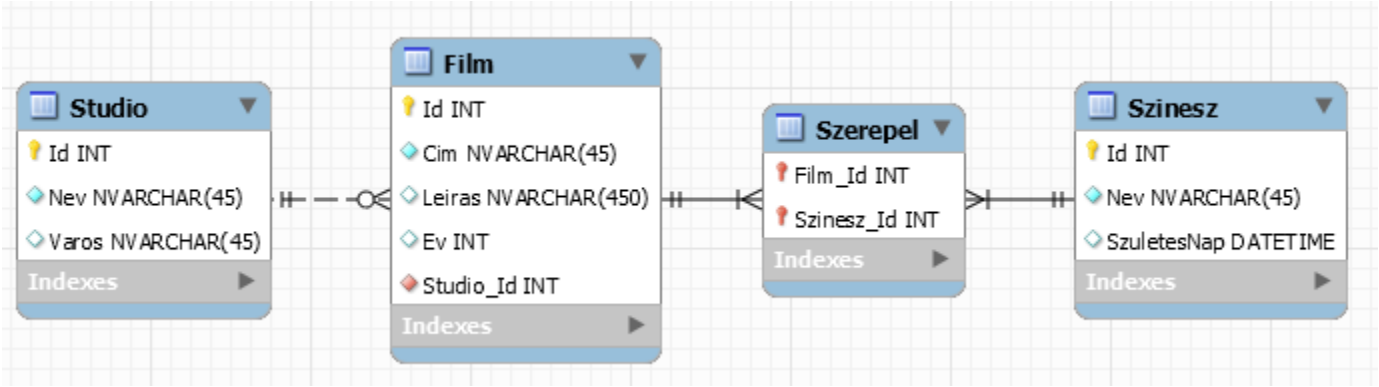
Nev	Termek	Mennyiseg
Kovács Béla	Alma	5
Kovács Béla	Körte	4
Kovács Béla	Körte	6
Nagy Anna	Alma	3
Nagy Anna	Szilva	4
Kiss István	NULL	NULL

Azonos termékek ugyanazon vevőnél többször is.

Jobb lenne összegezve....

Illesztés – JOIN - összefoglaló

Kapcsolótáblák bejárása

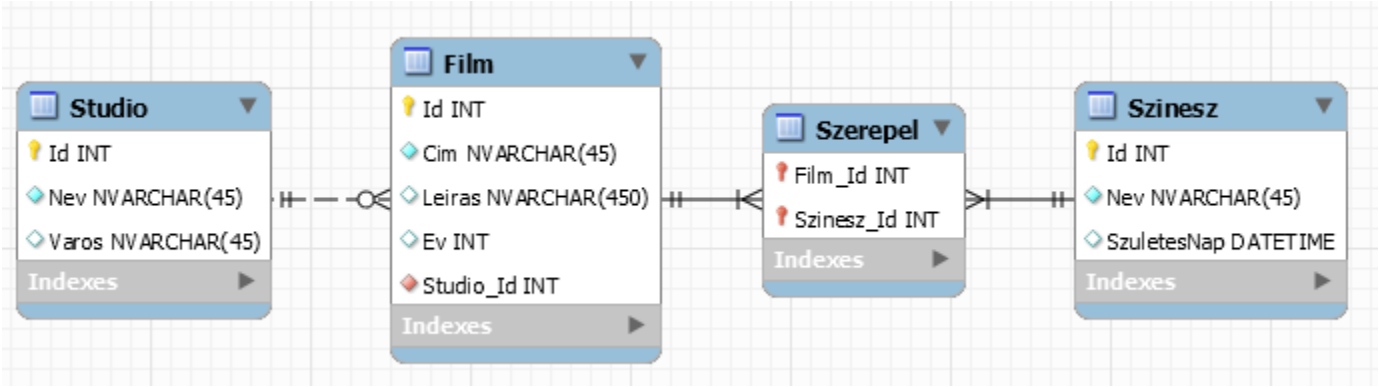


- Filmcímenként listázzuk a színészek neveit
  - SELECT f.Cim, szi.Nev FROM Film f
    INNER JOIN Szerepel sze ON f.Id = sze.Film_Id
    INNER JOIN Szinesz szi ON sze.Szinesz_Id = szi.Id

    =

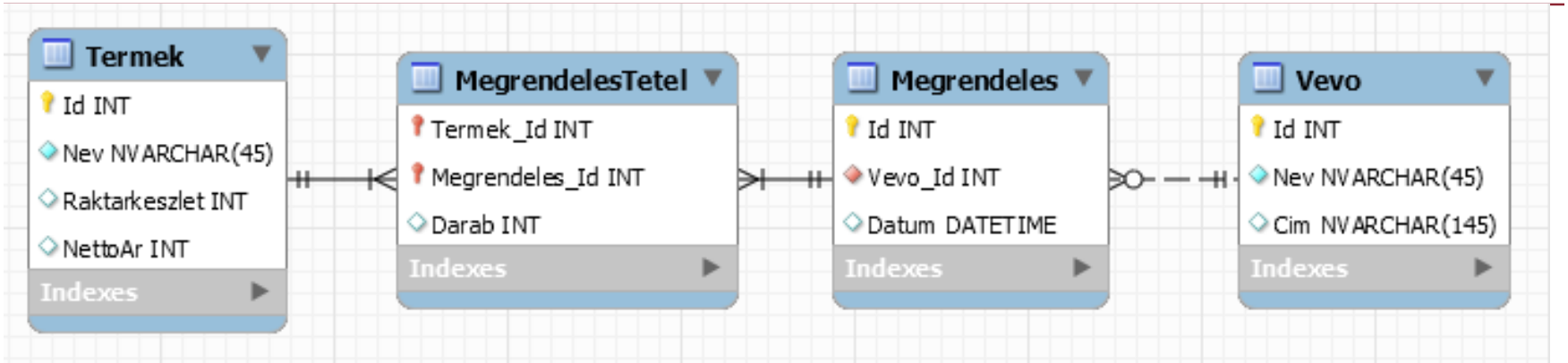
    SELECT f.Cim, szi.Nev FROM Film f, Szerepel sze, Szinesz szi
    WHERE f.Id = sze.Film_Id AND sze.Szinesz_Id = szi.Id

Kapcsolótáblák bejárása



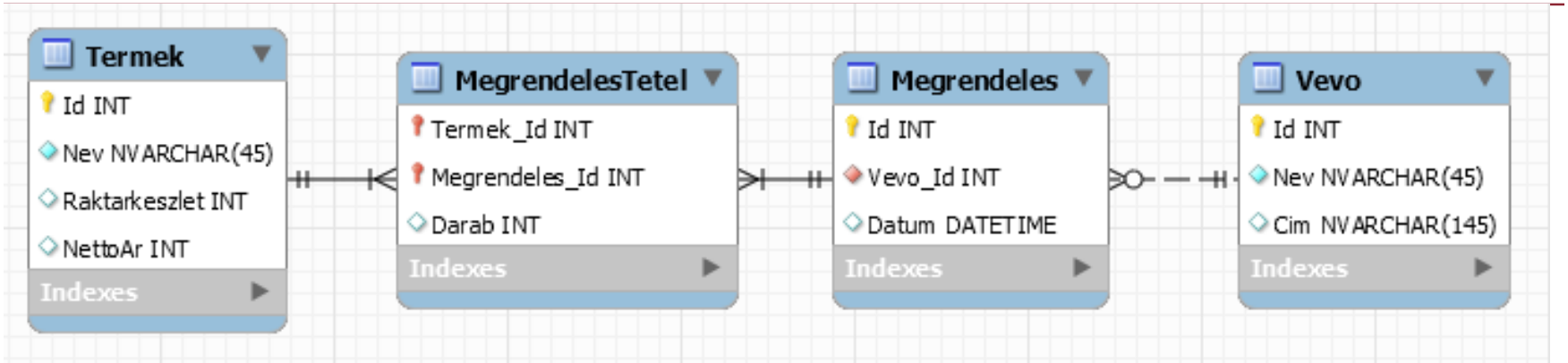
- Kik azok a színészek, akik az egyes stúdiók filmjeiben játszottak? (Írjuk ki a film nevét is)
  - SELECT s.Nev, f.Cim, szi.Nev FROM Studio s
    INNER JOIN Film f ON s.Id = f.Studio_Id
    INNER JOIN Szerepel sze ON f.Id = sze.Film_Id
    INNER JOIN Szinesz szi ON sze.Szinesz_Id = szi.Id
    =
    SELECT s.Nev, f.Cim, szi.Nev
    FROM Studio s, Film f, Szerepel sze, Szinesz szi
    WHERE s.Id = f.Studio_Id AND f.Id = sze.Film_Id AND sze.Szinesz_Id = szi.Id

Kapcsolótáblák bejárása



- Mely termékekből mikor és mennyit rendeltek?
  - SELECT t.Nev, TermekNev, mt.Darab Darabszam, m.Datum Mikor
    FROM Termek t
      INNER JOIN MegrendelesTetel mt ON mt.Termek_Id = t.Id
      INNER JOIN Megrendeles m ON mt.Id = mt.Megrendeles_Id
    =
    SELECT t.Nev, TermekNev, mt.Darab Darabszam, m.Datum Mikor
    FROM Termek t, MegrendelesTetel mt, Megrendeles m
    WHERE mt.Termek_Id = t.Id AND mt.Megrendeles_Id

Kapcsolótáblák bejárása

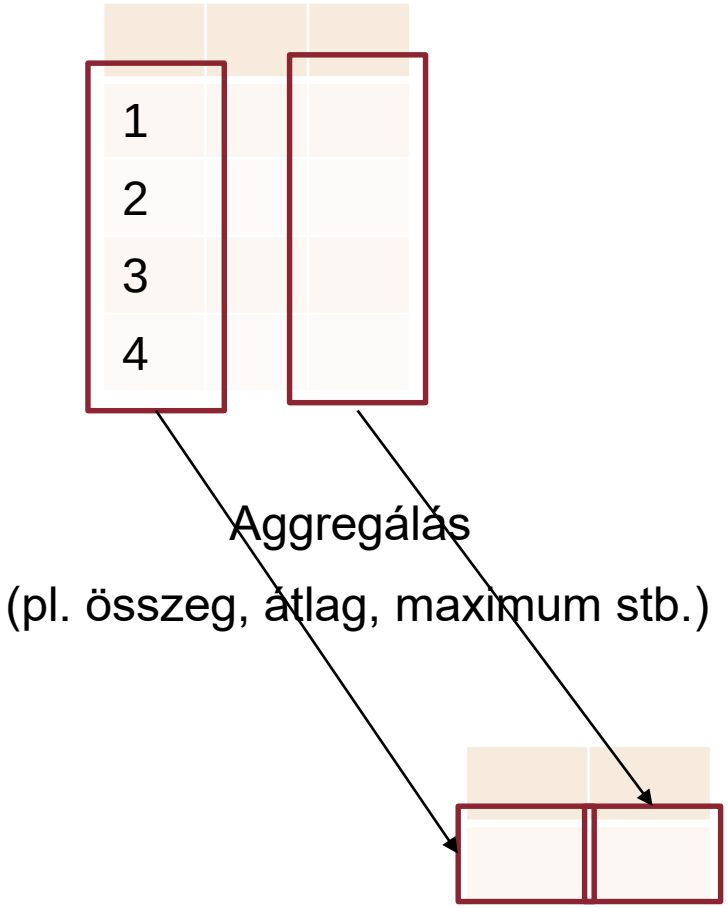


- Az egyes vevők milyen termékeket rendeltek?
  - SELECT v.Nev Vevo, t.Nev Termek Mikor FROM Termek t
    INNER JOIN MegrendelesTetel mt ON mt.Termek_Id = t.Id
    INNER JOIN Megrendeles m ON m.Id = mt.Megrendeles_Id
    INNER JOIN Vevo v ON v.Id = m.Vevo_Id

    =

    SELECT v.Nev Vevo, t.Nev Termek Mikor
    FROM Termek t, MegrendelesTetel mt, Megrendeles m, Vevo v
    WHERE mt.Termek_Id = t.Id AND mt.Megrendeles_Id AND v.Id = m.Vevo_Id

Aggregálás



A második oszlopot nem képezzük le, nem aggregáljuk (nem érdekel, vagy nincs értelme)

Aggregálás

- Probléma: nem az egyes rekordok egyes attribútumaira van szükség, hanem valamilyen aggregált értékre: összeg, minimum, maximum, darabszám...
  - Az összes megrendelés összege, a legnagyobb megrendelés értéke
- Megoldás: aggrágációs függvének:
  - SUM(A): az A attribútum értékeinek összege (numerikus)
  - MIN(A): az A attribútum értékei közül a legkisebb (num/alfa)
  - MAX(A): az A attribútum értékei közül a legnagyobb (num/alfa)
  - AVG(A): az A attribútum értékeinek az átlaga (numerikus)
  - COUNT(A): az A attribútum értékei közül a nem NULL elemek száma
  - => A teljes oszlopon alkalmazhatók

Aggregálás

- Termek(<u>Id</u>, Nev, Raktarkeszlet, Ar), MegrendelesTetel(<u>Id</u>, MegrendelesId, TermekId, Mennyiseg)

Id	Nev	Raktarkeszlet	Ar
1	Alma	50	150
2	Körte	48	280
3	Szilva	63	120

  - Maximális raktárkészlet
    - **SELECT MAX(Raktarkeszlet) FROM** Termek

63

  - Maximális raktárkészlet és termékszám
    - **SELECT MAX(**Raktarkeszlet**), COUNT(**Id**) FROM** Termek
      - **SELECT MAX(**Raktarkeszlet**), COUNT(*) FROM** Termek

63	3

  - Összes rendelés értéke Almából
    - **SELECT SUM (Mennyiseg*Ar) FROM** Termek **INNER JOIN** MegrendelesTetel **ON** Termek.Id=TermekId **WHERE** Nev='Alma'

450

- Aggregálás => 1 rekord => **minden attribútum csak aggregált lehet (!)**

Csoportosítás

- Termék(Id, Nev, Raktarkeszlet, ar),
  MegrendelesTetel (Id, TermekId, Darabszám)

Id	Nev	Raktarkeszlet	Ar
1	Lego	5	10000
2	Focilabda	18	5000

Id	TermekId	Darabszám
1	1	2
2	1	3
3	2	8

- A termékekhez tartozó rendelések és azok árai:
  - SELECT t.Id, t.Nev, t.Ar * mt.Darabszam as Osszeg
    FROM Termek t
      INNER JOIN MegrendelesTetel mt ON t.Id = mt.TermekId

Id	Nev	Osszeg
1	Lego	20000
1	Lego	30000
2	Focilabda	40000

Csoportosítás

Id	Nev	Osszeg
1	Lego	20000
1	Lego	30000
2	Focilabda	40000

Aggregálás

(maximum)

A többi oszlopnak nem lenne értelme

Osszeg
40000

Csoportosítás

Id	Nev	Osszeg
1	Lego	20000
1	Lego	30000
2	Focilabda	40000

Csoportosítás

(Id szerint)

Aggregálás

(összeg)

csoportonként

Id	Nev	Osszeg
1	Lego	50000
2	Focilabda	40000

Az aggregáló (csoportosító) oszlopok értéke egyértelmű

Csoportosítás

- Probléma: Az aggregálást nem az összes rekordra akarjuk elvégezni, hanem azok egyes csoportjaira külön-külön
- Pl Összes rendelés értéke *termékenként*
  - MegrendelesTetel(MegrendelesId, TermekId, Mennyiseg)
  - Megoldás: csoportosítás
  - R-t felpartícionáljuk az attribútumok szerint (TermekId), majd az egyes partíciókra elvégezzük az aggregálást
  - Az eredmény reláció minden egyes partícióhoz egy rekordot tartalmaz:
    - a csoportosító attribútumokat és
    - az aggregálás(ok) eredményét

Csoportosítás – 3

- MegrendelesTetel(<u>MegrendelesId</u>,<u>TermekId</u>,Mennyiseg)

- Pl Összes rendelés értéke *termékenként*

<u>MegrendelesId</u>	<u>TermekId</u>	Mennyiseg
1	1	5
1	2	4
2	2	6
3	1	3
3	3	4

TermekId	Osszeg
1	8
2	10
3	4

Csoportosítás – SQL

- **SELECT** A_i,..., A_j, $f_k(A_k)$,..., $f_l(A_l)$ **FROM** <tábla> **WHERE** <feltételek> **GROUP BY** A_i, ..., A_j
  - **SELECT** TermekId, **SUM(**Mennyiseg**) AS** Osszeg **FROM** MegrendelesTetel **GROUP BY** TermekId
- Összes megrendelés értéke termékenként:
  - **SELECT** Nev, **SUM**(Mennyiseg*Ar) **AS** Osszeg **FROM** Termek **INNER JOIN** MegrendelesTetel **ON** Termek.Id=TermekId **GROUP BY** Nev

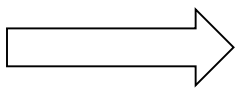
Csoportosítás – NULL kezelés

- Összegezzük termékenként a megrendelt mennyiséget
  - **SELECT** Nev, **IFNULL**(**SUM**(Mennyiseg),0) Mennyiseg **FROM** Termek **LEFT OUTER JOIN** MegrendelesTetel **ON** TermekId=Termek.Id **GROUP BY** Nev

Id	Nev	Raktarkeszlet
1	Alma	50
2	Körte	48
3	Szilva	63

MegrendelesId	TermekId	Mennyiseg
1	1	5
1	2	4
2	2	6
3	1	3

Nev	Mennyiseg
Alma	8
Körte	10
Szilva	NULL



Nev	Mennyiseg
Alma	8
Körte	10
Szilva	0

Csoportosítás – utószűrés

- Probléma: hogyan lehet a csoportosítás után előálló reláció elemeit szűrni?
  - Pl: azok a termékek, amiből 5 egységnél többet adtak el

- Megoldás 1: beágyazott szelekció
  - **SELECT** * **FROM** (
    SELECT TermekId, **SUM**(Mennyiseg) **AS** Osszeg **FROM** MegrendelesTetel **GROUP BY** TermekId) **AS** TMP
    WHERE Osszeg > 5

TermekId	Osszeg
1	8
2	10

Csoportosítás – utószűrés – 2

- Megoldás 2: **HAVING**
  - **SELECT** TermekId, **SUM**(Mennyiseg) **AS** Osszeg **FROM** MegrendelesTetel **GROUP BY** TermekId
    HAVING Osszeg > 5
  - Az egyes csoportokra fogalmaz meg feltételt
    - vs WHERE: a csoportosítás/aggregálás előtti rekordokra
  - Tartalmazhat
    - A kapcsolódó relációk bármely attribútuma AGGREGÁLVA
    - Csoportosító attribútumok (aggregálás nélkül is)
  - Pl: azok a termékek, amelyekből legalább 3000 Ft-ért rendeltek

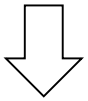
SELECT Nev, **SUM**(Mennyiseg*Ar) **AS** Osszeg **FROM** Termek
INNER JOIN MegrendelesTetel **ON** Termek.Id=TermekId
GROUP BY Nev
HAVING Osszeg >= 3000

TermekId	Osszeg
1	8
2	10

Duplikáció eliminálás

- Probléma: mivel halmaz helyett zsákok vannak, előfordulhat duplikáció, szükséges lehet, hogy ezeket eltávolítsuk.
  - Pl. Hány különböző nevű vevő található az adatbázisban?
- Megoldás:
  - SQL: DISTINCT
    - **SELECT DISTINCT** A1...An **FROM** ...
      - **SELECT DISTINCT** Nev **FROM** Vevo

Id	Nev	Cím
1	Kovács Béla	1111 Budapest...
2	Kiss István	5000 Szolnok...
3	Nagy Anna	9000 Győr ...
4	Kiss István	1221 Budapest..

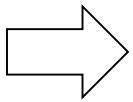


Nev
Kovács Béla
Kiss István
Nagy Anna

Rendezés

- Probléma: az eredményeket rendezetten szeretnénk megtekinteni
  - Pl az összes termék, eladott mennyiség szerint csökkenő sorrendben
- Megoldás: **ORDER BY**
  - **SELECT** ... **FROM** ... **WHERE**... **GROUP BY** ... **HAVING** ... **ORDER BY** A1, A2, ...
    - A projekció előtt történik a rendezés
    - A felsorolás szerint, növekvő sorrendben
  - **SELECT** * **FROM** Termek **ORDER BY** Raktarkeszlet

Id	Nev	Raktarkeszlet
1	Alma	50
2	Körte	48
3	Szilva	63

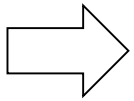


Id	Nev	Raktarkeszlet
2	Körte	48
1	Alma	50
3	Szilva	63

Rendezés – 2

- **SELECT** * **FROM** MegrendelesTetel **ORDER BY** TermekId, Mennyiseg

MegrendelesId	TermekId	Mennyiseg
1	1	5
1	2	4
2	2	6
3	1	3
3	3	4



MegrendelesId	TermekId	Mennyiseg
3	1	3
1	1	5
1	2	4
2	2	6
3	3	4

- Sorrend: ASC (alapértelmezett), DESC
  - **SELECT** * **FROM** MegrendelesTetel **ORDER BY** TermekId **ASC**, Mennyiseg **DESC**
    - Először ID szerint növekvő, majd azon belül mennyiség szerint csökkenő

MegrendelesId	TermekId	Mennyiseg
1	1	5
3	1	3
2	2	6
1	2	4
3	3	4

Rendezés – 3

- Az összes termék, eladott mennyiség szerint csökkenő sorrendben?
  - **SELECT** Nev, **SUM**(Mennyiseg) Osszeg **FROM** Termek
    LEFT OUTER JOIN MegrendelesTetel **ON** TermekId=Termek.Id
    GROUP BY Nev
    ORDER BY Osszeg **DESC**
- Tartalmazhat kifejezéseket is
  - ORDER BY Osszeg*Ar

Rendezés – 4

- Probléma: nincs szükségem az összes rekordra, amely kielégíti a feltételt, csak az első n darabra
  - Pl. Az első két legnépszerűbb termék (amelyből a legtöbbet adták el)
- Megoldás:
  - **SELECT** ... **LIMIT** n (MySQL)
    - **SELECT** Nev, **SUM**(Mennyiseg) Total **FROM** Termek **INNER JOIN** MegrendelesTetel **ON** TermekId = Termek.Id **GROUP BY** Nev **ORDER BY** Total **DESC** **LIMIT 2**

Beágyazott szelekció

- Ahogy korábban is volt rá példa, lekérdezések eredménye (mint reláció) felhasználható további lekérdezésekben
- 1) **FROM** listában
  - **SELECT** A1...An **FROM** T1...Tk...(**SELECT** ... **FROM** ...) **AS** Tx ...
    - Kötelező elnevezni
    - Lehet hozzá joinolni is akár
  - **SELECT** TermekId **FROM**
    (**SELECT** * **FROM** Termek **WHERE** Raktarkeszlet > 3) **AS** TMP
    - **SELECT** TermekId **FROM** Termek **WHERE** Raktarkeszlet > 3
  - Soroljuk fel az összes termék nevét, és a belőlük eladott teljes mennyiséget
    - **SELECT** Nev, Mennyiseg **FROM** Termek **INNER JOIN**
      (**SELECT** TermekId, SUM(Mennyiseg) **AS** Mennyiseg **FROM** MegrendelesTetel **GROUP BY** TermekId) **AS** TMP **ON** Termek.Id=TMP.TermekId
      - **SELECT** Nev, **SUM**(Mennyiseg) **FROM** MegrendelesTetel **INNER JOIN** Termek **ON** Termek.Id=TermekId **GROUP BY** Nev (v. Id,Nev))

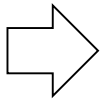
Beágyazott szelekció – 2

- 2) WHERE feltételben
  - Ha a sub-select csak egy egyattribútumos rekordot ad vissza, akkor használhatjuk mint egy skalár értéket
    - **SELECT** * **FROM** TERMEK **WHERE** Id = (**SELECT MAX**(Id) **FROM** Termek)
      - SELECT * FROM Termek ORDER BY Id Desc LIMIT 1
        - kevésbé hatékony, rendezni kell a rekordsetet, maximumkeresés helyett
  - Ha több rekordot is visszaadhat, akkor feltételeket fogalmazhatunk meg rá
    - EXISTS: nem üres a rekordset
      - **SELECT** * **FROM** Termek **WHERE EXISTS** (**SELECT** * **FROM** MegrendelesTetel **WHERE** TermekId=Termek.Id)
        - **SELECT DISTINCT** Termek.* **FROM** MegrendelesTetel **INNER JOIN** Termek **ON** TermekId = Termek.Id
    - IN: érték a rekordsetben található (egy-attribútumos)
      - **SELECT** * **FROM** Termek **WHERE** Id **IN** (**SELECT** TermekId **FROM** MegrendelesTetel)
    - NOT: előző kettő negálása (NOT EXISTS, NOT IN)
    - ANY, ALL : a rekordset bármely/összes elemére teljesül a feltétel
      - **SELECT** * **FROM** Termek **WHERE** Id > **ALL** (**SELECT** TermekId **FROM** MegrendelesTetel)

Beágyazott szelekció – 3

- 3) Attribútum listában
  - Szükséges lehet, hogy az eredményhalmaz valamely attribútuma egy teljesen másik lekérdezés eredménye legyen
  - **SELECT** Id, Nev, (**SELECT MAX**(Id) **FROM** Termek) MaxId **FROM** Termek
  - Csak skalár lehet az eredménye

Id	Nev	Raktarkeszlet
1	Alma	50
2	Körte	48
3	Szilva	63



Id	MaxId
1	3
2	3
3	3

Nézetek

- A gyakran használt lekérdezésekre készíthetünk shortcutokat
  - CREATE VIEW <nézet neve> AS <lekérdezés>
    - **CREATE VIEW** Osszesites **AS**
      SELECT TermekId, **SUM**(Mennyiseg) **AS** Mennyiseg **FROM** MegrendelesTetel Group BY TermekId
  - Használhatjuk, mint bármely más relációt
    - **SELECT** * **FROM** Osszesites
    - **SELECT** * **FROM** Osszesites **WHERE** Mennyiseg > 10
  - Modosítás, Törlés
    - **DROP VIEW** <nézet neve>
    - **ALTER VIEW** <nézet neve> **AS** <új lekérdezés>
  - Jogosultságkezelés lehetséges

Nézetek

- Virtuális nézet
  - A legtöbb RDBMS támogatja, ez az „alapértelmezett”
  - Tényleg csak egy shortcut a queryre, amikor a nézethez fordulunk, a query mindig kiértékelődik
- Materializált nézet
  - Csak a nagyok támogatják (Oracle, DB2, MSSQL), MySQL nem
  - A View rekordjai fizikailag is eltárolódnak
    - A hivatkozott táblák módosításakor vagy kérésre frissítődnek
      - Overhead, de optimalizált, nem az egész számítódik ki újra
    - Lekérdezéskor performance boost...

String kezelés

- Probléma: hogyan lehet feltételeket megfogalmazni stringekre?
  - Melyik vevők laknak Budapesten? (Cim tartalmazza a Budapest stringet)
  - Kik a Kovács vezetéknevű vevők?
- Megoldás: mintaillesztés **LIKE** operátorral
  - % : tetszőleges karaktersorozat
  - _: tetszőleges karakter (pontosan 1)

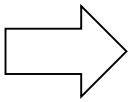
String kezelés – 2

- Kovács vezetéknév
  - **SELECT** * **FROM** Vevo **WHERE** Nev **LIKE** ’Kovács %’
- Budapesti lakcím
  - **SELECT** * **FROM** Vevo **WHERE** Cim **LIKE** ’%Budapest%’
- 4 karakteres vezetéknevű vásárlók
  - **SELECT** * **FROM** Vevo **WHERE** Nev **LIKE** ’____ %’
- Olyan vásárlók, akiknek a nevében szerepel a ’%’ jel
  - **SELECT** * **FROM** Vevo **WHERE** Nev **LIKE** ’%!%%’ **ESCAPE** ’!’
    - Escape után megadott karakter: a mintában escapeli a következő karaktert

String kezelés – 3

- Függvények
  - **char_length**(A) (MySQL), **LENGTH**(A) (ORACLE), **LEN**(A) (MSSQL) : hossz
    - LENGTH(A): MySQL: byteban
  - **UPPER**(A), **LOWER**(A) : nagy/kisbetűssé alakítás
  - **SELECT char_length**(Nev), **UPPER**(Nev), **LOWER**(Nev) **FROM** Vevo

Id	Nev	Cím
1	Kovács Béla	1111 Budapest...
2	Kiss István	5000 Szolnok...
3	Nagy Anna	9000 Győr ...



11	KOVÁCS BÉLA	kovács béla
11	KISS ISTVÁN	kiss istván
8	NAGY ANNA	nagy anna

Dátum kezelés – MySQL

- Automatikus konverzió ,'yyyy-MM-dd hh:mm:ss' formátumról
  - **INSERT INTO** Megrendeles (Vevold, Datum) **VALUES** (1, '2012-12-12')
- curdate() : aktuális dátum
  - **SELECT curdate**()
    - 2012-12-12
- now() : aktuális dátum és idő
  - **SELECT now**()
    - 2012-12-12 12:34:56
- str_to_date(str, format): str parszolása dátumként
  - **SELECT str_to_date**('12,12,2012','%d,%m,%Y');
    - 2012-12-12

Dátum kezelés – MySQL – 2

- date_format(date, format): dátum formázása
  - SELECT **date_format**(curdate(), '%d/%m/%y')
    - '12/12/2012'
- datediff(date1, date2): a két dátum közti **napok** száma
  - SELECT **datediff**('2012-12-12', '2012-12-11')
    - 1
- timediff(date1, date2) a két dátum közti **idő**
  - SELECT **timediff**('2012-12-12 12:12:12', '2012-12-11 13:13:13')
    - 22:58:59