# ADATBÁZIS RENDSZEREK

*SQL elmélet*

Krausz Nikol, Molnár Bence

**2022.04.14.**

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

# Mai témáink

- SQL alapok
- SQL lekérdezések műveletei
- SQL Adatleírás
- SQL Adatmódosítás

# SQL

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

# Történet

- 1970-es évek eleje IBM SEQUEL (Structured English QUery Language)
- Structured/Standard Query Language
- 1986-tól ANSI, 1987-től ISO szabvány
- SQL2 ('92), SQL3 ('99), ...
- folyamatos fejlesztés (SQL2011)
- szabvány, melyet többé-kevésbé a legtöbb relációs adatbázis-kezelő támogat
- Relational Software, Inc. → Oracle Corp.

# Alapok

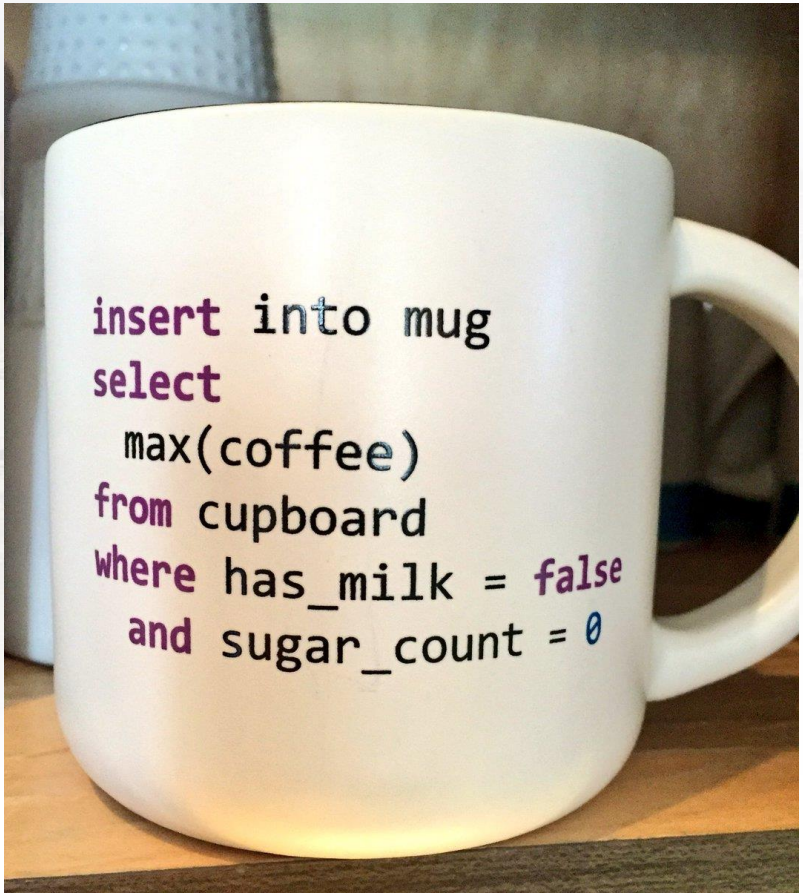
- Magja ekvivalens a relációs algebrával
- Nem procedurális nyelv  (deklaratív), tehát az alapvető kérdés az, hogy mit kell csinálni, és nem az, hogy hogyan.
- Adatleíró (DDL) -, Adatmódosító (DML) -, Adatelérést vezérlő (DCL) -, Lekérdező (QL) nyelv

# SQL Alapok

A feladatot szövegesen fogalmazzuk meg

A megfogalmazás szigorú szabályok szerint történik

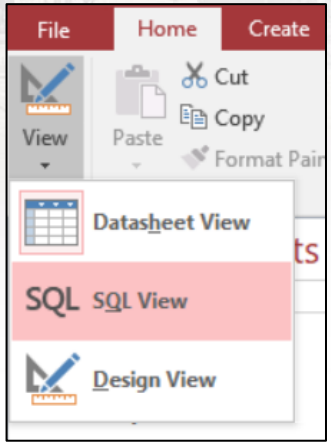
- Előre megadott parancsok
- Mondat végén: “;”
- Parancsok sorrendje nem tetszőleges


insert into mug
select
max(coffee)
from cupboard
where has_milk = false
and sugar_count = 0

# Access-ben

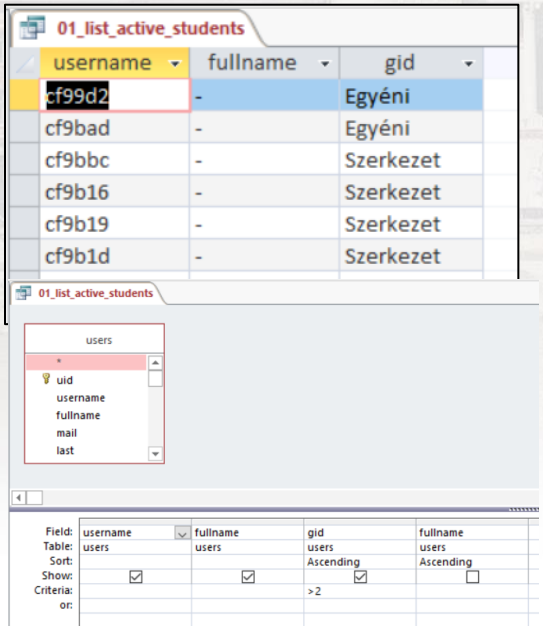
Ha Tervező nézetben összekattintjuk a lekérdezést, azt utána át tudjuk fordítani SQL-re is.

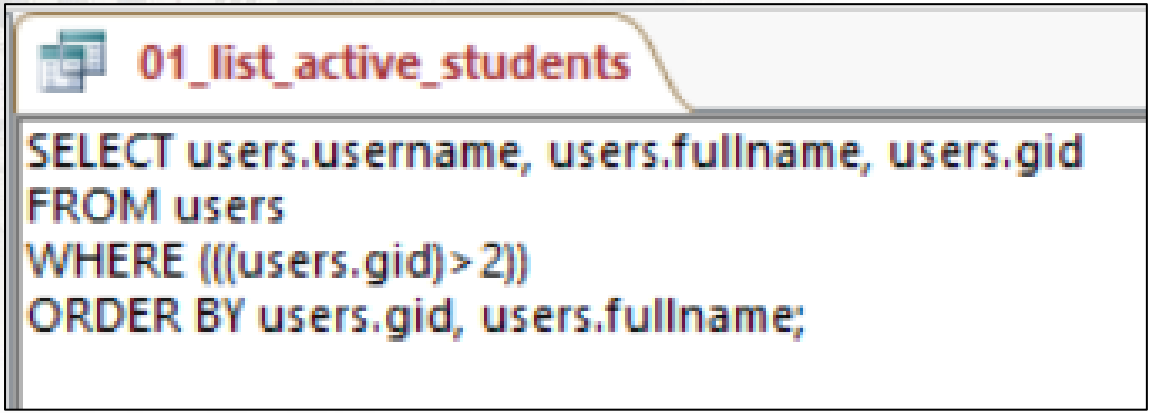
Viszont SQL-ben írt lekérdezést nem biztos, hogy át tud fordítani a grafikus felületre!



# Példák – I.

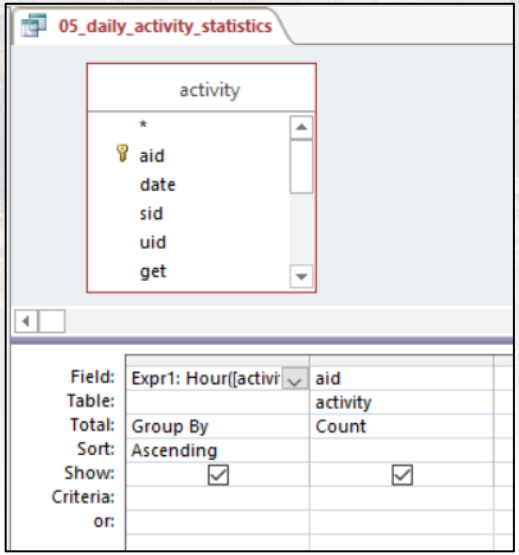
Listázzuk az aktív hallgatókat, rendezzük sorba szakirány alapján!



# Példák – II.

Mutassuk meg a napon belüli aktivitás statisztikát!




05_daily_activity_statistics

```
SELECT Hour([activity].[date]) AS Hours, Count(activity.aid) AS CountOfaid
FROM activity
GROUP BY Hour([activity].[date])
ORDER BY Hour([activity].[date]);
```

# Elemzés műveletei

*Lekérdezések*

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

# ÁTTÉRÉS RELÁCIÓS ALGEBRÁRÓL

| Relációs algebra | SQL |
|---|---|
| $R$ | `SELECT * FROM R;` |
| $\pi_{t \to tt}(R)$ | `SELECT t AS tt FROM r;` |
| $\sigma_{felt}(R)$ | `SELECT * FROM r WHERE felt;` |
| $\pi_t(\sigma_{felt}(R))$ | `SELECT t FROM r WHERE felt;` |
| $R \cup S$ | `SELECT * FROM r UNION SELECT * FROM s;` |
| $R \cap S$ | `SELECT * FROM r INTERSECT SELECT * FROM s;` |
| $R \backslash S$ | `SELECT * FROM r EXCEPT SELECT * FROM s;` |
| $R \times S$ | `SELECT * FROM r, s; (vagy CROSS JOIN)` |
| $R \bowtie S$ | `SELECT * FROM r NATURAL JOIN s;` |
| $R \bowtie_{felt} S$ | `SELECT * FROM r JOIN s ON felt;` |
| $\delta(R)$ | `SELECT DISTINCT * FROM r;` |
| $\gamma_t(R)$ | `SELECT t FROM r GROUP BY t;` |
| $\tau_t(R)$ | `SELECT * FROM r ORDER BY t;` |

# SELECT lekérdezések

Egy SELECT lekérdezés az adatokon sose módosít, nem változtatja meg az oszlopok neveit, csak az adatokat elemzi és megjeleníti ideiglenes formában.

Egy lekérdezés megtekintése mindig új futtatást jelent az aktuális adattartalom alapján.

A SELECT lekérdezések eredménye is egy reláció, amit további lekérdezésekkel elemezhetünk.

# Vetítés (projekció)

Vetítés (projekció) [π]: a reláció bizonyos attribútumait(oszlopait) megtartjuk és a többit törölve új relációt hozunk létre.

SQL: `SELECT attr1, attr2, … FROM r;`

- `SELECT nev, szuletesi_ev FROM hallgato;`

| neptun | nev | szuletesi_ev |
|---|---|---|

- `SELECT * FROM hallgato;`

| neptun | nev | szuletesi_ev |
|---|---|---|

# VETÍTÉS

- `SELECT nev FROM hallgato;`

$\pi_{nev}$(

| nev | jegy |
|---|---|
| Kiss Pista | 2 |
| Nagy Péter | 3 |
| Vál Péter | 5 |
| Nagy Ákos | 3 |

)=

| nev |
|---|
| Kiss Pista |
| Nagy Péter |
| Vál Péter |
| Nagy Ákos |

- `SELECT nev, jegy FROM hallgato;`

$\pi_{nev,jegy}$(

| nev | jegy | jelenlet |
|---|---|---|
| Kiss Pista | 2 | 14 |
| Nagy Péter | 3 | 14 |
| Vál Péter | 5 | 13 |
| Nagy Ákos | 3 | 10 |

)=

| nev | jegy |
|---|---|
| Kiss Pista | 2 |
| Nagy Péter | 3 |
| Vál Péter | 5 |
| Nagy Ákos | 3 |

# Szűrés (szelekció, kiválasztás)

Lekérdezés eredmény sorainak szűrése

SQL: `SELECT * FROM r WHERE attr1 comp1 value1 op1 attr2 comp2 value2…;`

compX ∈ ('=', '<', '>', '≠', '≤', '≥')

opX ∈ ('AND', 'OR', 'XOR', 'NOT')

- `SELECT * FROM hallgato WHERE nev = 'Kiss Péter';`

Több feltétel is lehet, logikai kapcsolatokkal összefűzve

**Módosítási parancsoknál különös jelentőségű**

# Szűrés

- `SELECT * FROM hallgato WHERE jegy=3;`

$\sigma_{jegy=3}($

| nev | jegy |
|---|---|
| Kiss Pista | 2 |
| Nagy Péter | 3 |
| Vál Péter | 5 |
| Kiss Pista | 3 |

$)=$

| nev | jegy |
|---|---|
| Nagy Péter | 3 |
| Kiss Pista | 3 |

- `SELECT * FROM hallgato WHERE jegy>1 AND jelenlet>10;`

$\sigma_{jegy>1\ AND\ jelenlet>10}($

| nev | jegy | jelenlet |
|---|---|---|
| Kiss Pista | 1 | 14 |
| Nagy Péter | 3 | 14 |
| Vál Péter | 5 | 13 |
| Nagy Ákos | 3 | 10 |

$)=?$

# Függvények használata

- `SELECT AVG(jegy) FROM hallgato;`
- `SELECT MIN(jegy) FROM hallgato;`
- `SELECT MAX(jegy) FROM hallgato;`
- `SELECT COUNT(*) FROM hallgato;`
- `SELECT SUM(jegy)/COUNT(*) FROM hallgato WHERE jegy IS NOT NULL;`

# Szabad szavas keresés

…mezo LIKE 'szoveg'…

- `SELECT * FROM hallgato WHERE nev LIKE 'Kiss*';`

Joker karakterek:

- *'*' tetszőleges karakter; 0,1,sok karakter ( MySQL-ben: %)*
- *'?' egy tetszőleges szöveges karakter (MySQL-ben: _)*
- *'#' egy tetszőleges szám karakter*

# Szöveg függvények

- `SELECT vnev & ' ' & knev FROM hallgato;`
- `SELECT * FROM hallgato WHERE vnev & ' ' & knev = 'Nagy Béla';`

Access: &

MySQL (MariaDB): CONCAT()

PgSQL: ||

# Bonyolultabb lekérdezések

- `SELECT nev FROM hallgato WHERE (nev LIKE '*Péter' OR nev LIKE '*Sándor') AND nev LIKE 'Nagy*';`
- `SELECT nev FROM hallgato WHERE nev LIKE 'Kov?cs*';`
- `SELECT eloadas_hossza*60 AS perc FROM targyak;`
- `SELECT targynev FROM targyak WHERE eloadasok_hossza*eloadasok_szama>kredit;`
- `SELECT nev FROM hallgato WHERE targy = 'Adatbázisok' AND NOT bukott;`

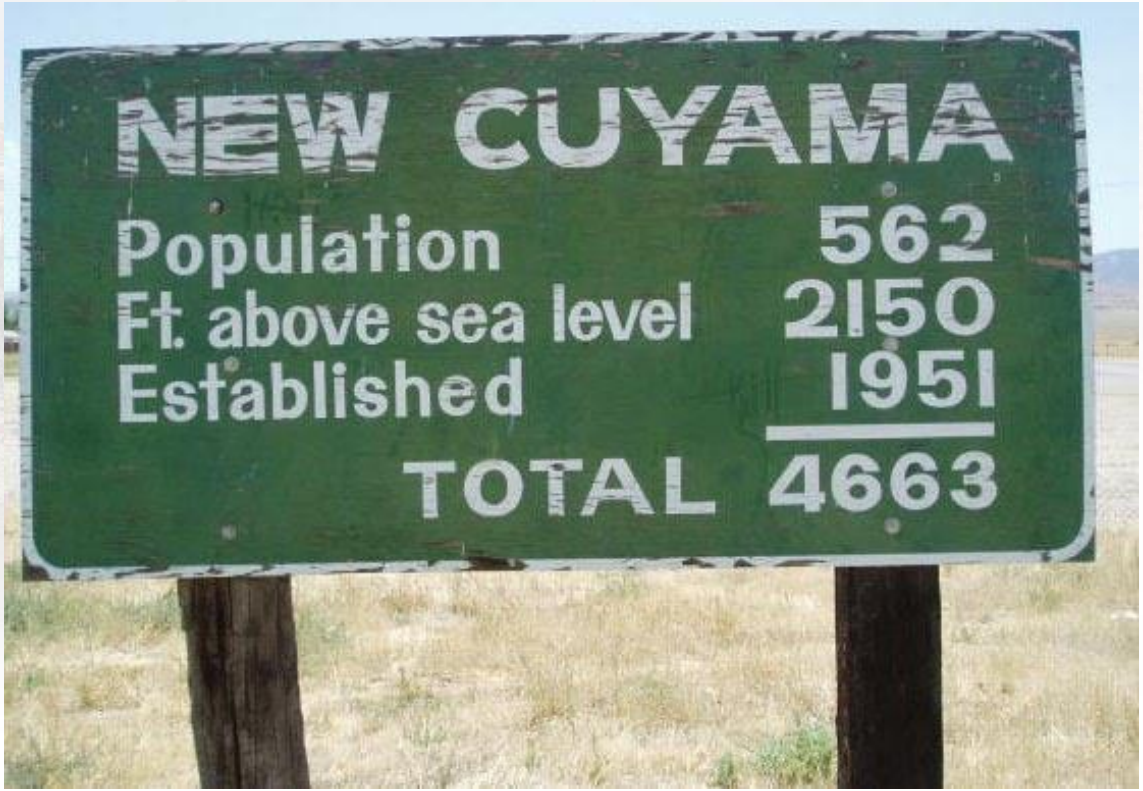
# Több táblát összekapcsolása

*Lekérdezések*

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

# Gondolkodva cselekedjünk!

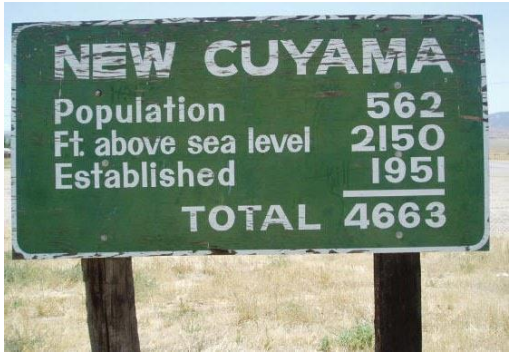


# Táblák kombinálása

- Halmazműveletek
- Al-lekérdezés
- Descartes szorzat
- Természetes összekapcsolás
- Théta összekapcsolás
  - *Inner*
  - *Left*
  - *Right*
  - *Full*

# Halmazműveletek kritériumai

- A **két változós halmaz műveletekhez** a következőknek kell teljesülni mindkét (R, és S) relációra
  - *Az R és S relációknak ugyanazt az attribútumhalmazt kell tárolnia*
  - *Az attribútumokat rendezni kell úgy, hogy az R i-ik oszlopa megegyezzen S i-ik oszlopával*



# Halmazműveletek 1 - Unió

Halmazok elemeit egymás alá rakja, oszlopok száma nem bővül.

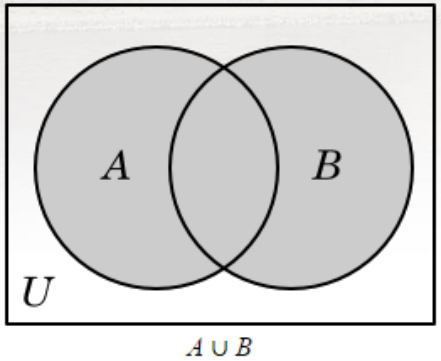
SQL: *SELECT * FROM A UNION SELECT * FROM B;*

| nev | jegy |
|---|---|
| Kiss Pista | 2 |
| Nagy Péter | 3 |
| Vál Péter | 5 |
| Nagy Ákos | 3 |

U

| nev | jegy |
|---|---|
| Kiss Lajos | 2 |
| Nagy Lajos | 3 |

=



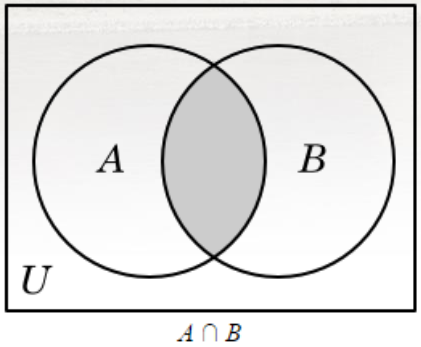

$A \cup B$

| nev | jegy |
|---|---|
| Kiss Pista | 2 |
| Nagy Péter | 3 |
| Vál Péter | 5 |
| Nagy Ákos | 3 |
| Kiss Lajos | 2 |
| Nagy Lajos | 3 |

# Halmazművelet 2 - Metszet

Oszlopok száma nem változik.

SQL: SELECT * FROM a INTERSECT SELECT * FROM b;




$A \cap B$

| nev | jegy |
|---|---|
| Kiss Pista | 2 |
| Nagy Péter | 3 |
| Vál Péter | 5 |
| Nagy Ákos | 3 |

$\cap$

| nev | jegy |
|---|---|
| Kiss Pista | 2 |
| Nagy Lajos | 3 |

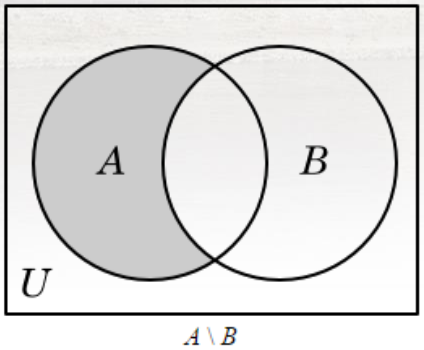
=

| nev | jegy |
|---|---|
| Kiss Pista | 2 |

# Halmazművelet 3 - Különbség

Oszlopok száma nem változik.

SQL: SELECT * FROM a EXCEPT SELECT * FROM b;



$A \setminus B$

| nev | jegy |
|---|---|
| Kiss Pista | 2 |
| Nagy Péter | 3 |
| Vál Péter | 5 |
| Nagy Ákos | 3 |

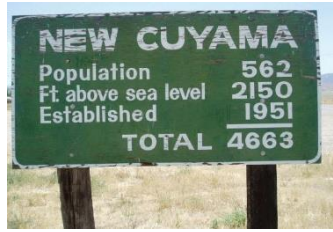
\

| nev | jegy |
|---|---|
| Kiss Pista | 2 |
| Nagy Lajos | 3 |

=

| nev | jegy |
|---|---|
| Nagy Péter | 3 |
| Vál Péter | 5 |
| Nagy Ákos | 3 |

# Al-lekérdezés (Subselect, subquery)




Al-lekérdezésekkel két relációt tudunk összekombinálni. Oszlopok száma növekszik, mert az adatokat egymás mellé akarjuk tenni.
Általában kerüljük az al-lekérdezéseket, mert lassú.

movie(title, year, length, studioname, directorid)
director(id, name, address, income)

- `SELECT name FROM director WHERE id = (SELECT directorid FROM movie WHERE title = 'Lord of The Rings');`
- `SELECT * FROM movie WHERE directorid IN (SELECT id FROM director WHERE name LIKE 'P?ter*');`

# Descartes-szorzat

"Mindent mindennel" összekombinál, gondolkodás nélkül, oszlopok száma növekszik.

SQL: `SELECT * FROM a,b;`

| nev | jegy |
|---|---|
| Kiss Pista | 2 |
| Nagy Péter | 3 |

×

| nev | jelenlet |
|---|---|
| Kiss Pista | 10 |
| Nagy Péter | 14 |
| Nagy Lajos | 5 |

=

| a.nev | jegy | b.nev | jelenlet |
|---|---|---|---|
| Kiss Pista | 2 | Kiss Pista | 10 |
| Kiss Pista | 2 | Nagy Péter | 14 |
| Kiss Pista | 2 | Nagy Lajos | 5 |
| Nagy Péter | 3 | Kiss Pista | 10 |
| Nagy Péter | 3 | Nagy Péter | 14 |
| Nagy Péter | 3 | Nagy Lajos | 5 |

Az eredmény nagyon sok sort tartalmazhat!

Általában WHERE szűréssel tesszük értelmessé a lekérdezést.

# Természetes Összekapcsolás

Két eltérő reláció kombinálása

- *a két táblában egy mező neve megegyezik, ez lesz azösszekapcsolás alapja*
- *az összekapcsolt mezők adattípusának meg kell egyeznie*
- *a lekérdezésben nincs más feltétel az összekapcsolásra*
- *a megfeleltetett attribútumok közül csak egyik oszlopot tartalmazza az eredmény*
- *Access nem ismeri*

# Természetes Összekapcsolás

SQL: `SELECT * FROM r NATURAL JOIN s;`

| nev | jegy |
|---|---|
| Kiss Pista | 2 |
| Nagy Péter | 3 |

⋈

| nev | jelenlet |
|---|---|
| Kiss Pista | 5 |
| Nagy Péter | 14 |
| Nagy Lajos | 11 |
| Nagy Péter | 12 |

=

| nev | jegy | jelenlet |
|---|---|---|
| Kiss Pista | 2 | 5 |
| Nagy Péter | 3 | 14 |
| Nagy Péter | 3 | 12 |

- Mindkét relációban található név oszlop.
- Név oszlop csak egyszer szerepel az eredményben!

# Lógó sorok

Olyan sor, amelyek egy másik reláció egyetlen sorával sem kapcsolódnak össze.

LEFT/RIGHT/FULL JOIN-okkal lehet kontrolálni melyik jelenjen meg és melyik nem.

# Théta Összekapcsolás

- Nem szükséges, hogy a kapcsolat a sémában definiált legyen.
- Legalább egy olyan feltételt meg kell adni, ahol a két adatforrás egyes oszlopai egyenlőek.
- Allekérdezések esetén is használható.

Általánosan minden théta eredmény halmazra igaz:

- A párosításra használt oszlopok mindkét relációból megjelennek az eredményben.
- Ha valamiből több párt talál, mindegyik bekerül az eredménybe.

# Théta Összekapcsolás - INNER

Descartes szorzatból indul ki, de kiegészítjük egy szűrési feltétellel.

SQL: `SELECT * FROM r INNER JOIN s ON r.attr1=s.attr2;`

| nev | jegy |
|---|---|
| Kiss Pista | 2 |
| Nagy Péter | 3 |

$\bowtie_{a.nev = b.teljesnev\ AND\ jelenlet > 10}$

| teljesnev | jelenlet |
|---|---|
| Kiss Pista | 5 |
| Nagy Péter | 14 |
| Nagy Lajos | 11 |
| Nagy Péter | 12 |

=

| nev | jegy | teljesnev | jelenlet |
|---|---|---|---|
| Nagy Péter | 3 | Nagy Péter | 14 |
| Nagy Péter | 3 | Nagy Péter | 12 |

SELECT * FROM a INNER JOIN b ON a.nev=b.teljesnev AND jelenlet>10;

- Csak azok a sorok jelennek meg, amelyek mindkét relációban szerepelnek.

# Théta Összekapcsolás - LEFT

Descartes szorzatból indul ki, de kiegészítjük egy szűrési feltétellel.

SQL: `SELECT * FROM r LEFT JOIN s ON r.attr1=s.attr2;`

| nev | jegy |
|---|---|
| Kiss Pista | 2 |
| Nagy Péter | 3 |

$\bowtie_{a.nev = b.teljesnev\ AND\ jelenlet > 10}$

| teljesnev | jelenlet |
|---|---|
| Kiss Pista | 5 |
| Nagy Péter | 14 |
| Nagy Lajos | 11 |
| Nagy Péter | 12 |

=

| nev | jegy | teljesnev | jelenlet |
|---|---|---|---|
| Kiss Pista | 2 | NULL | NULL |
| Nagy Péter | 3 | Nagy Péter | 14 |
| Nagy Péter | 3 | Nagy Péter | 12 |

- A bal oldali táblából minden sor megjelenik. Ha nincs párja, NULL értékekkel feltöltve.

# Théta Összekapcsolás - RIGHT

Descartes szorzatból indul ki, de kiegészítjük egy szűrési feltétellel.

SQL: `SELECT * FROM r RIGHT JOIN s ON r.attr1=s.attr2;`

| nev | jegy |
|---|---|
| Kiss Pista | 2 |
| Nagy Péter | 3 |

$\bowtie_{a.nev = b.teljesnev\ AND\ jelenlet > 10}$

| nev | jegy | teljesnev | jelenlet |
|---|---|---|---|
| Nagy Péter | 3 | Nagy Péter | 14 |
| Nagy Péter | 3 | Nagy Péter | 12 |
| NULL | NULL | Nagy Lajos | 11 |

=

| teljesnev | jelenlet |
|---|---|
| Kiss Pista | 5 |
| Nagy Péter | 14 |
| Nagy Lajos | 11 |
| Nagy Péter | 12 |

- A jobb oldali táblából minden sor megjelenik. Ha nincs párja, NULL értékekkel feltöltve.
- Kiss Pista azért nem kerül bele mindkét oldalról, mert van a jelenlétre is egy követelmény, aminek nem felel meg.

# Théta Összekapcsolás - FULL

Descartes szorzatból indul ki, de kiegészítjük egy szűrési feltétellel.

SQL: `SELECT * FROM r FULL OUTER JOIN s ON r.attr1=s.attr2…;`

| nev | jegy |
|---|---|
| Kiss Pista | 2 |
| Nagy Péter | 3 |

$\bowtie_{a.nev = b.teljesnev\ AND\ jelenlet > 10}$

| teljesnev | jelenlet |
|---|---|
| Kiss Pista | 5 |
| Nagy Péter | 14 |
| Nagy Lajos | 11 |
| Nagy Péter | 12 |

=

| nev | jegy | teljesnev | jelenlet |
|---|---|---|---|
| Kiss Pista | 2 | NULL | NULL |
| Nagy Péter | 3 | Nagy Péter | 14 |
| Nagy Péter | 3 | Nagy Péter | 12 |
| NULL | NULL | Nagy Lajos | 11 |

# Théta Összekapcsolás - FULL

```
SELECT * FROM a FULL OUTER JOIN b ON
a.nev=b.teljesnev AND jelenlet>10;
```

- Access nem támogatja ☹
- A mindkét oldali táblából minden sor megjelenik. Ha nincs párja, NULL értékekkel feltöltve.
- Kiss Pista azért nem kerül bele mindkét oldalról, mert van a jelenlétre is egy követelmény, aminek nem felel meg.

# További műveletek

*Lekérdezések*

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

# Sorba rendezés

SQL: `SELECT * FROM r ORDER BY attr1 ASC/DESC, attr2 ASC/DESC…;`

- `SELECT * FROM hallgato ORDER BY szuletesi_hely DESC;`

| neptun | nev | szuletesi_hely |
|---|---|---|
| | | Abádszalók |
| | | Abádszalók |
| | | Almásfüzitő |
| | | Budapest |
| | | … |

| neptun | nev | szuletesi_hely |
|---|---|---|
| | | Zalaegerszeg |
| | | Zalaegerszeg |
| | | Záhony |
| | | Záhony |
| | | … |

# Sorrendezés, több attribútumon

- Először az első attribútumon rendezünk, majd ha van egyező tétel, akkor a következőn, és így tovább.
- `SELECT * FROM hallgato ORDER BY nev, jegy;`

$\tau_{nev,jegy}($

| nev | jegy | jelenlet |
|---|---|---|
| Kiss Pista | 1 | 14 |
| Nagy Péter | 4 | 14 |
| Kiss Aladár | 2 | 14 |
| Nagy Péter | 3 | 10 |

$)=$

| nev | jegy | jelenlet |
|---|---|---|
| Kiss Aladár | 2 | 14 |
| Kiss Pista | 1 | 14 |
| **Nagy Péter** | **3** | **10** |
| **Nagy Péter** | **4** | **14** |

# Sorok számának korlátozása

SQL: `SELECT TOP N * FROM r;`

- `SELECT TOP 1 nev, jegy FROM hallgato ORDER BY jegy;`

MySQL:
```
SELECT * FROM hallgato ORDER BY jegy
LIMIT 1;
```

Ha egy sort keresünk, érdemes ezt beleírni, mert gyorsítja a lekérdezést, hiszen nem keres tovább.

# Csak az egyedi sorok megjelenítése

Ha vannak ismétlődő teljes sorok, akkor azokat kiszűrhetjük.

SQL: `SELECT DISTINCT * FROM r;`

- `SELECT DISTINCT * FROM hallgato;`

$\delta($

| nev | jegy | jelenlet |
|---|---|---|
| Kiss Pista | 1 | 14 |
| Nagy Péter | 3 | 14 |
| Kiss Pista | 1 | 14 |
| Nagy Ákos | 3 | 10 |

$)=$

| nev | jegy | jelenlet |
|---|---|---|
| Kiss Pista | 1 | 14 |
| Nagy Péter | 3 | 14 |
| Nagy Ákos | 3 | 10 |

# Csoportosítás

Csoportosítás - a reláció sorainak „csoportokba” történő beosztása a reláció egy vagy több attribútumának értékétől függően.

SQL: `SELECT attr1, attr2,… FROM r GROUP BY attr1, attr2,…;`

- `SELECT termek FROM eladasok GROUP BY termek;`

$$\gamma_{termek}(\pi_{termek}( \quad ))$$

| termek | darab |
|---|---|
| Kenyér | 1 |
| Kifli | 2 |
| Tej | 1 |
| Kifli | 3 |

# LÉPÉSRŐL LÉPÉSRE

$\pi_{termék}($

| termek | darab |
|---|---|
| Kenyér | 1 |
| Kifli | 2 |
| Tej | 1 |
| Kifli | 3 |

$)=$

| termek |
|---|
| Kenyér |
| Kifli |
| Tej |
| Kifli |

$\gamma_{termek}($

| termek |
|---|
| Kenyér |
| Kifli |
| Tej |
| Kifli |

$)=$

| termek |
|---|
| Kenyér |
| Kifli |
| Tej |

# Csoportosítás + Összesítés

- Ahogy láttuk a csoportosítás ugyanolyan elemeket von össze az adott attribútumon.
- Azonban a többi attribútum is tartalmaz adatot, ezeket csoportonként összesíthetjük valamilyen módon.
- Az összesítéshez különböző függvényeket alkalmazhatunk a gamma operátoron belül.
- Ezek  a következőek: SUM, AVG, MIN, MAX, COUNT, FIRST, LAST

# Az eredmény reláció felépítése

- Osszuk a reláció sorait csoportokba. Egy csoport azokat a sorokat tartalmazza, amelyeknek az {attr1, attr2, …} listán szereplő csoportosítási attribútumokhoz tartozó értékei megegyeznek. Ha nincs csoportosítási attribútum, akkor az egész R reláció egy csoportot képez.
- Minden csoporthoz hozzunk létre olyan sort, amelyik tartalmazza:
  - *Szóban forgó csoport csoportosítási attribútumait.*
  - *Az {attr1, attr2, …} lista összesítési attribútumaira vonatkozó összesítéseket.*

# Csoportosítás + Összesítés (Példa - SUM)

- `SELECT termek, SUM(darab) FROM eladas GROUP BY termek;`

$$\gamma_{termek,SUM(darab)}($$

| termek | darab |
|---|---|
| Kenyér | 1 |
| **Kifli** | **3** |
| Tej | 2 |
| **Kifli** | **5** |

$$)=$$

| termek | sumdarab |
|---|---|
| Kenyér | 1 |
| **Kifli** | **8** |
| Tej | 2 |

# Feltétel megadása a csoport tulajdonságaira

SQL: `SELECT attr1,AGGR(attr2)… FROM r GROUP BY attr1 HAVING AGGR(attr2) comp1 value1,…;`

- `SELECT AVG(jegy) FROM hallgato WHERE targy = 'Adatbázisok' AND jegy > 3 GROUP BY ev HAVING AVG(jegy)>3.5;`

| nev | ev | jegy |
|---|---|---|
| Kiss Pista | 1990 | 4 |
| Nagy Jenő | 1990 | 5 |
| Kertész Ágnes | 1990 | 1 |
| Bíró Gabriella | 1991 | 5 |
| Szántó Jóska | 1991 | 4 |
| Tóth Barna | 1991 | 2 |
| Sándor Lívia | 1992 | 4 |

SELECT ev, AVG(jegy) FROM hallgato GROUP BY ev;

| ev | jegy |
|---|---|
| 1990 | 3.33 |
| 1991 | 3.67 |
| 1992 | 4 |

| nev | ev | jegy |
|---|---|---|
| Kiss Pista | 1990 | 4 |
| Nagy Jenő | 1990 | 5 |
| Kertész Ágnes | 1990 | 1 |
| Bíró Gabriella | 1991 | 5 |
| Szántó Jóska | 1991 | 4 |
| Tóth Barna | 1991 | 2 |
| Sándor Lívia | 1992 | 4 |

SELECT ev, AVG(jegy) FROM hallgato GROUP BY ev;

| ev | jegy |
|---|---|
| 1990 | 3.33 |
| 1991 | 3.67 |
| 1992 | 4 |

SELECT ev, AVG(jegy) FROM hallgato WHERE jegy>3 GROUP BY ev;

Eredeti sorok közötti szűrés

| ev | jegy |
|---|---|
| 1990 | 4.5 |
| 1991 | 4.5 |
| 1992 | 4 |

| nev | ev | jegy |
|---|---|---|
| Kiss Pista | | |
| Nagy Jenő | | |
| Kertész Ágnes | | |
| Bíró Gabriella | 1991 | 5 |
| Szántó Jóska | 1991 | 4 |
| Tóth Barna | 1991 | 2 |
| Sándor Lívia | 1992 | 4 |

SELECT ev, AVG(jegy) FROM hallgato GROUP BY ev;

| ev | jegy |
|---|---|
| 1990 | 3.33 |
| 1991 | 3.67 |
| 1992 | 4 |

SELECT ev, AVG(jegy) FROM hallgato WHERE jegy>3 GROUP BY ev;

Eredeti sorok közötti szűrés

| ev | jegy |
|---|---|
| 1990 | 4.5 |
| 1991 | 4.5 |
| 1992 | 4 |

SELECT ev, AVG(jegy) FROM hallgato GROUP BY ev HAVING AVG(jegy)>=3.5;

Összesített sorok közötti szűrés

| ev | jegy |
|---|---|
| 1991 | 3.67 |
| 1992 | 4 |

# Csoportosítás + Összesítés (Példa – Több attribútumra)

- `SELECT termek, SUM(darab), SUM(darab*ar) FROM eladas GROUP BY termek;`

$$\gamma_{termek,SUM(darab),SUM(\text{darab}*ar)}($$

| termek | darab | ar |
|---|---|---|
| Kenyér | 1 | 100 |
| Kifli | 2 | 100 |
| Kifli | 3 | 150 |
| Kenyér | 2 | 150 |
| Tej | 2 | 100 |

$$)=$$

| termek | sumdarab | sumdarabar |
|---|---|---|
| Kenyér | 3 | 400 |
| Kifli | 5 | 650 |
| Tej | 2 | 200 |

*Fontos, hogy az eredmény relációban egy oszlop csak akkor jelenhet meg, ha az vagy*

- *csoportosítás alapja, vagy*
- *valamely összesítő művelet alkalmazva van rá.*

# Csoportosítás + Összesítés (Példa - COUNT)

- `SELECT termek, COUNT(darab) FROM eladas GROUP BY termek;`

$$\gamma_{termek,COUNT(darab)}($$

| termek | darab |
|---|---|
| Kenyér | 1 |
| **Kifli** | **3** |
| Tej | 2 |
| **Kifli** | **5** |

$$)=$$

| termek | countdarab |
|---|---|
| Kenyér | 1 |
| **Kifli** | **2** |
| Tej | 1 |

# Csoportosítás + Összesítés (Példa - FIRST)

- `SELECT termek, FIRST(darab) FROM eladas GROUP BY termek;`

$$\gamma_{termek,FIRST(darab)}($$

| termek | darab |
|---|---|
| Kenyér | 1 |
| **Kifli** | **3** |
| Tej | 2 |
| Kifli | 5 |

$$)=$$

| termek | firstdarab |
|---|---|
| Kenyér | 1 |
| **Kifli** | **3** |
| Tej | 2 |

# Átnevezés (Alias)

Bizonyos esetekben az oszlopokat át szeretnénk nevezni a lekérdezésben:

- Függvények vagy képletek által előállított oszlopokat
- Összekapcsolások esetén elkerülni a mezőnév ütközéseket

SQL: … AS…

- ```
  SELECT AVG(jegy) AS atlag, jelenlet/14*100 AS
  jelenletszazalek FROM hallgato;
  ```

Sőt, az AS parancs el is hagyható:

- ```
  SELECT AVG(jegy) atlag, jelenlet/14*100 jelenletszazalek
  FROM hallgato;
  ```

Sőt, al-lekérdezéseknél akár a lekérdezés egységeket is el tudjuk nevezni, hogy lehessen rá hivatkozni:

- ```
  SELECT c.currency, c.ratetohuf, c.date FROM currencyrate AS
  c LEFT JOIN (SELECT currency, MAX(ratetohuf) AS max FROM
  currencyrate GROUP BY currency) AS m ON
  c.currency=m.currency AND c.ratetohuf=m.max;
  ```
  ```

# Attribútum átnevezés

- `SELECT nev AS diak FROM hallgato;`

$$\pi_{nev \to diak}($$

| nev | jegy | jelenlet |
|---|---|---|
| Kiss Pista | 1 | 14 |
| Nagy Péter | 3 | 14 |
| Kiss Pista | 1 | 14 |
| Nagy Ákos | 3 | 10 |

$$) =$$

| diak |
|---|
| Kiss Pista |
| Nagy Péter |
| Kiss Pista |
| Nagy Ákos |

- `SELECT nev AS diak FROM hallgato WHERE diak='Kiss Pista';`

$$\sigma_{\text{diak}='Kiss\ Pista'}(\pi_{nev \to diak}($$

| nev | jegy | jelenlet |
|---|---|---|
| Kiss Pista | 1 | 14 |
| Nagy Péter | 3 | 14 |
| Kiss Péter | 1 | 14 |
| Nagy Ákos | 3 | 10 |

$$))$$

$$=$$

| diak |
|---|
| Kiss Pista |

# Parancsok sorrendje

A lekérdezésben a parancsok sorrendje fontos!!!

```
SELECT
DISTINCT
TOP
FROM
JOIN      ON
WHERE
GROUP BY
HAVING
ORDER BY
```

# Szintaktika

- A szelekciónál, Théta összekapcsolásnál, amennyiben több attribútumra végzünk lekérdezést, azokat AND/OR jellel válasszuk el egymástól! A projekció által megjelenítendő oszlopneveket viszont elegendő vesszővel elválasztani!
- Amennyiben valamilyen szöveges értékre végzünk szűrést, vagy Théta összekapcsolást, a szöveges érték kerüljön idézőjelbe.
- Amennyiben a lekérdezésben több reláció is szerepel, a mezőnevek előtt jelenjen meg a tartalmazó reláció neve (relacio.mezonev).
- Ha összekapcsolásnál különböző nevűek a kapcsolat alapjául szolgáló oszlopok nevei, akkor Théta összekapcsolást kell használni
- Théta összekapcsolás esetén a feltételek közt szerepelnie kell az összekapcsolás alapját jelentő két mező egyenlőségének.
- A dátumokat szövegként írjuk fel, ezért idézőjelek közé kerül, de ha szabványos formában adjuk meg, akkor az Adatbáziskezelő rendszerek numerikusan is tudják értelmezni.

# Cheat Sheet

SELECT attribútum(ok) FROM táblá(k) WHERE tulajdonság(ok);

- `SELECT nev FROM hallgato WHERE nev = 'Kiss Péter';`

Attribútumok, táblák elválasztása vesszővel
Tulajdonságok összefűzése logikai operátorral: AND, OR, XOR, NOT
Összehasonlító operátorok: =, <>, >, <, =>, <=, LIKE, BETWEEN, IN, IS
DISTINCT
aliasok (AS)
Különleges karakterek: Access-ben: *, ?, # (Egyes DBMS-ekben: *, _, %)
Escape karakterek (\)
Idézőjelek: ', "
Kis- és nagybetűk

# Példák lekérdezésekre

*Gyakorlás*

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

# Példa 1

Adjuk meg SQL segítségével, azon hallgatók neveit, akik átmentek a tárgyból.

| S | | |
|---|---|---|
| **Név** | **Jegy** | **Jelenlét** |
| Kiss Pista | 3 | 8 |
| Kiss István | 2 | 14 |
| Nagy Irén | 5 | 10 |
| Nagy Péter | 1 | 14 |

$$SELECT\ név\ FROM\ S\ WHERE jegy > 1\ AND\ jelenlét > 10;$$

# Példa 2

| oktatok | |
|---|---|
| oktato | tanszek_kod |
| Koppányi Zoltán | FMT |
| Lovas Tamás | FMT |
| Tuchband Tamás | AGT |
| Gipsz Jakab | NULL |

| tanszekek | |
|---|---|
| tanszek_kod | tanszek_nev |
| FMT | Fotogrammetria |
| AGT | Geodézia |
| OCT | Szerves Kémia |

# Példa 2 – Természetes Összekapcsolás

```
SELECT * FROM oktatok NATURAL JOIN
tanszekek;
```

| oktato | tanszek_kod | tanszek_nev |
|---|---|---|
| Koppányi Zoltán | FMT | Fotogrammetria |
| Lovas Tamás | FMT | Fotogrammetria |
| Tuchband Tamás | AGT | Geodézia |

# Példa 2 - LEFT JOIN

```
SELECT * FROM oktatok LEFT JOIN tanszekek ON
oktatok.tanszek_kod = tanszekek.tanszek_kod;
```

| oktatok.oktato | oktatok.tanszek_kod | tanszekek.tanszek_kod | tanszekek.tanszek_nev |
|---|---|---|---|
| Koppányi Zoltán | FMT | FMT | Fotogrammetria |
| Lovas Tamás | FMT | FMT | Fotogrammetria |
| Tuchband Tamás | AGT | AGT | Geodézia |
| Gipsz Jakab | NULL | NULL | NULL |

# Példa 2 - RIGHT JOIN

```
SELECT * FROM oktatok RIGHT JOIN tanszekek
ON oktatok.tanszek_kod =
tanszekek.tanszek_kod;
```

| oktatok.oktato | oktatok.tanszek_kod | tanszekek.tanszek_kod | tanszekek.tanszek_nev |
|---|---|---|---|
| Koppányi Zoltán | FMT | FMT | Fotogrammetria |
| Lovas Tamás | FMT | FMT | Fotogrammetria |
| Tuchband Tamás | AGT | AGT | Geodézia |
| NULL | NULL | OCT | Szerves Kémia |

# Gyakorlás - Séma

| A | | | |
|---|---|---|---|
| **nev** | **tantargy** | **pontszam** | **jelenlet** |
| Kiss Pista | Matek | 50 | 8 |
| Kiss Pista | Rajz | 60 | 14 |
| Nagy Iván | Statika | 45 | 10 |
| Nagy Péter | Matek | 15 | 14 |

| B | |
|---|---|
| **nev** | **evfolyam** |
| Kiss Pista | 1 |
| Kiss István | 2 |
| Nagy Iván | 1 |
| Nagy Péter | 1 |

| C | |
|---|---|
| **tantárgy** | **minpont** |
| Matek | 40 |
| Rajz | 60 |
| Statika | 50 |

# Gyakorlás - Kérdések

1) Adjuk meg, azon hallgatókat, akik Rajzra járnak!

2) Adjuk meg az elsős hallgatók neveit!

3) Adjuk meg azon tárgyakat, amelyek teljesítéséhez több mint 45 pont kell!

4) Adjuk meg azon hallgatókat, és évfolyamukat, akik Matekra járnak!

5) Adjuk meg, hogy mely hallgatóknak milyen tantárgyuk sikerült (jelenlét ellenőrzése nélkül)!

6) Adjuk meg, hogy az elsős hallgatóknak milyen tantárgyak sikerültek (jelenlét ellenőrzése nélkül)!

# Gyakorlás – Megoldás

1) `SELECT nev FROM A WHERE tantargy='Rajz';`
2) `SELECT nev FROM B WHERE evfolyam=1;`
3) `SELECT tantargy FROM C WHERE minpont>45;`
4) `SELECT nev, evfolyam FROM A NATURAL JOIN B WHERE A.tantargy='Matek';`
5) `SELECT nev, tantárgy FROM A INNER JOIN C ON A.tantargy=C.tantárgy AND A.pontszam>C.minpont;`
6) `SELECT nev, tantargy FROM A NATURAL JOIN B INNER JOIN C ON A.tantargy=C.tantárgy AND A.pontszam>C.minpont WHERE evfolyam=1;`

# Adatleírás

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

# ADATBÁZIS LÉTREHOZÁSA

- SQL: `CREATE DATABASE adatbazis;`
- jogosultságok
  - *felhasználó*
  - *elérés „helye”*
- kódolás (UTF8, LATIN2 [ISO-8859-2])
- sablonok (pl. téradatok tárolása esetén)
- használat
  - *explicit definíció kapcsolódáskor*
  - `USE adatbazis;`
- Használt karakterek: [_a-zA-Z0-9] (nem kezdődhet számmal)

# Emlékeztető
# Relációs adatbázis séma

Relációs séma: hallgato(neptun: Szöveg, nev: Szöveg, szuletesi_ev: Szám)

Relációs adatbázis séma:

| hallgato |
| --- |
| neptun: Szöveg<br>nev: Szöveg<br>szuletesi_ev: Szám<br>... |

# Tábla létrehozása

```
CREATE TABLE hallgato
(
      neptun VARCHAR(6),
      nev VARCHAR(50),
      szuletesi_ev INTEGER,
      atlag DOUBLE PRECISION,
      PRIMARY KEY (neptun)
);
```

# Tábla létrehozása

```
CREATE TABLE táblanév
```

…:

- *DEFAULT alapértelmezett érték*
- *NULL /NOT NULL*
- *UNIQUE (!= PRIMARY KEY!)*
- *AutoNumber/AUTO INCREMENT/Sequence*

# Tábla módosítása

SQL: `ALTER TABLE táblanév ADD oszlopnév típus;`
- *`ALTER TABLE hallgato ADD szuletesi_hely VARCHAR(50);`*

SQL: `ALTER TABLE táblanév DROP oszlopnév;`
- *`ALTER TABLE hallgato DROP nev;`*

SQL: `ALTER TABLE táblanév RENAME TO új_táblanév;`
- *`ALTER TABLE hallgato RENAME TO bme_hallgato;`*

SQL: `DROP táblanév;`
- *`DROP hallgato;`*

SQL: `TRUNCATE táblanév;`
- *`TRUNCATE hallgato;`*

Változtatások esetén figyelni kell a többi táblával való kapcsolatok frissítésére is!

# Adatmódosítás

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék

# Adatfeltöltés

## Beszúrás

- SQL: `INSERT INTO táblanév [(attribútum1, attribútum2, …)] VALUES (érték1, érték2, …);`
- `INSERT INTO hallgato (neptun, nev, szuletesi_ev, atlag) VALUES ('ABCDEF', 'Kiss Péter', 1993, 4.5);`
- SQL: `LOAD DATA`
- SQL: `COPY`

# Adatmódosítás

Szelekció – különben minden sort módosítunk

Módosítás

- *SQL:* `UPDATE táblanév SET attribútum = új_érték WHERE feltétel;`
- `UPDATE hallgato SET atlag = 4.6 WHERE nev = 'Nagy Sándor';`

Törlés

- *SQL:* `DELETE FROM táblanév WHERE attribútum = érték;`
- `DELETE FROM hallgato WHERE nev = 'Kovács József';`

# Összefoglalás

- Visszatekintés
- Attribútum típusok
- SQL és műveletei
- SQL Adatleírás
- SQL Adatmódosítás

# Köszönöm a figyelmet!

*Kérdések?*

BUDAPESTI MŰSZAKI
ÉS GAZDASÁGTUDOMÁNYI EGYETEM
Építőmérnöki Kar - építőmérnöki képzés 1782 óta

Fotogrammetria és Térinformatika Tanszék