

3.előadás: Adatbázisok-I.

dr. Hajas Csilla (ELTE IK)
<http://sila.hajas.elte.hu/>

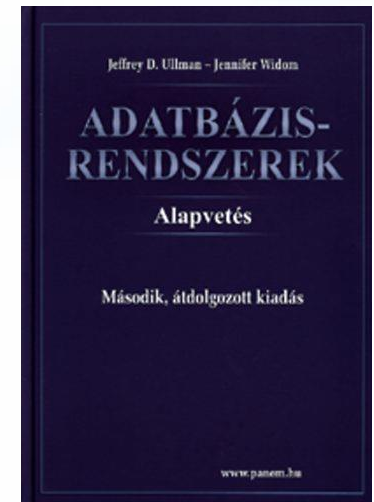
Relációs algebra egy táblára vonatkozó műveletei és az SQL SELECT utasítás

Tankönyv:

5.2. Kiterjesztett relációs algebra
összesítések és csoportosítás

6.4. Csoportosítás az SQL-ben
group by, having záradékok

Kiegészítés: Leckék Oracle gyakorlatra



Egy táblás lekérdezések az SQL-ben és az SELECT utasítás záradékai

- **Mit tanultunk eddig?**
 - E/K (egyed-tulajdonság-kapcsolat) modell alapjai,
 - egy egyed-és-tulajdonságainak megadása relációval,
 - egy relációra (egy táblára) vonatkozó lekérdezések,
 - alap relációs algebra vetítés a kiválasztás műveletei,
 - SELECT utasítás SELECT és WHERE záradékai.
- **Mi a mai előadás célja?** Mielőtt tovább mennénk a
 - magas szintű modellekkel, relációs sématervezéssel,
 - megszorításokkal, és a több táblás lekérdezésekkel,előtte legyen meg a gyakorlatunk az egy táblára vonatkozó SQL lekérdezésekben, a gyakorlati anyaghoz kapcsolódva a gépes I.ZH-ra feladatok (Példatár 1-2.fej. feladatsora).

Egy táblás lekérdezések az SQL-ben és az SELECT utasítás záradékai

- A mai 3. előadáson megbeszéljük az Oracle gyakorlatokhoz kapcsolódó leckéket, sorban:
 - [./sql/lecke01_select_alap.pdf](#)
 - [./sql/lecke02_where_feltetel.pdf](#)
 - [./sql/lecke03_fuggvenyek.pdf](#)
 - [./sql/lecke04_csoportok.pdf](#)
- A fenti leckék és az SQL Language Reference a gyakorlatokon és a gépes ZH-án is használható, de a vizsgán nem használható segédanyag, ezért a továbbiakban összefoglaljuk ami a vizsgára kell:

Egy táblára vonatkozó lekérdezések alap relációs algebra

- Emlékeztető: **Egy reláció (vagy tábla) két részből áll:**
 - (1) **relációs sémából** (sortípus és megszorítások)
 - (2) **reláció előfordulásából** (véges sok sor halmaza)
- **Egy táblára vonatkozó műveletek** relációs algebraiban:
Mi lesz az eredmény tábla sémája és előfordulása?
(Itt alap relációs algebraiban halmazként, SQL-ben és kiterjesztett relációs algebraiban majd multihalmazként)

Vetítés

1. táblán

Kiválasztás

1. táblán

Egy táblára vonatkozó lekérdezések kiterjesztett relációs algebra

- Az eddig tanult egy táblára vonatkozó műveleteket: **vetítés** (Π_{lista}) és **kiválasztás** (σ_{felt}) műveletét kiterjesztjük **multihalmazokra**, ahogyan az **SQL-ben**, egy reláció nem sorok halmazából, hanem sorok **multihalmazából** áll, vagyis megengedett a sorok ismétlődése.
- Ezeken kívül a **SELECT kiegészítéseinek és záradékainak** megfelelően **új műveletekkel** is kibővítjük a rel. algebrát:
 - **Ismétlődések megszüntetése** (δ) - **select distinct**
 - **Vetítési művelet kiterjesztése** (Π_{lista}) - **select kif [as onev]**
 - **Rendezési művelet** (τ_{lista}) - **order by**
 - **Összesítő művelet, csoportosítás** **GROUP BY** és **HAVING záradékok** ($\sigma_{csop.felt. \gamma_{csop.attr.lista}}$)

A relációs algebrai unér műveletek értelmezése multihalmazok fölött

- A projekció és szelekció végrehajtása során nem küszöböljük ki az ismétlődéseket.

R			$\Pi_A(R)$
A	B		A
1	2	➔	1
1	5		1
2	3		2

Új műveletek a kiterjesztett algebrában:

1.) Ismétlődések megszüntetése: **DISTINCT**

- **Ismétlődések megszüntetése**: $R1 := \delta(R2)$
- A művelet jelentése: $R2$ multihalmazból $R1$ halmazt állít elő, vagyis az $R2$ -ben egyszer vagy többször előforduló sorok csak egyszer szerepelnek az $R1$ -ben.
- A **DISTINCT** reprezentálására szolgál (jele: δ kis-delta)
- A δ speciális esete lesz az általánosabb γ műveletnek

$$R = \left(\begin{array}{|c|c|} \hline A & B \\ \hline 1 & 2 \\ 3 & 4 \\ 1 & 2 \\ \hline \end{array} \right)$$
$$\delta(R) = \begin{array}{|c|c|} \hline A & B \\ \hline 1 & 2 \\ 3 & 4 \\ \hline \end{array}$$

2.) A vetítési művelet kiterjesztése

- $\Pi_L(R)$ kiterjesztett vetítés L listájában szerepelhetnek:
- Az R reláció attribútuma
- $E \rightarrow z$ kifejezés, ahol E az R reláció attribútumaira vonatkozó (konstansokat, aritmetikai műveleteket, függvényeket tartalmazó kifejezés), z pedig az

R

A	B
1	2
1	5
2	3

$\Pi_{A+B \rightarrow z}(R)$



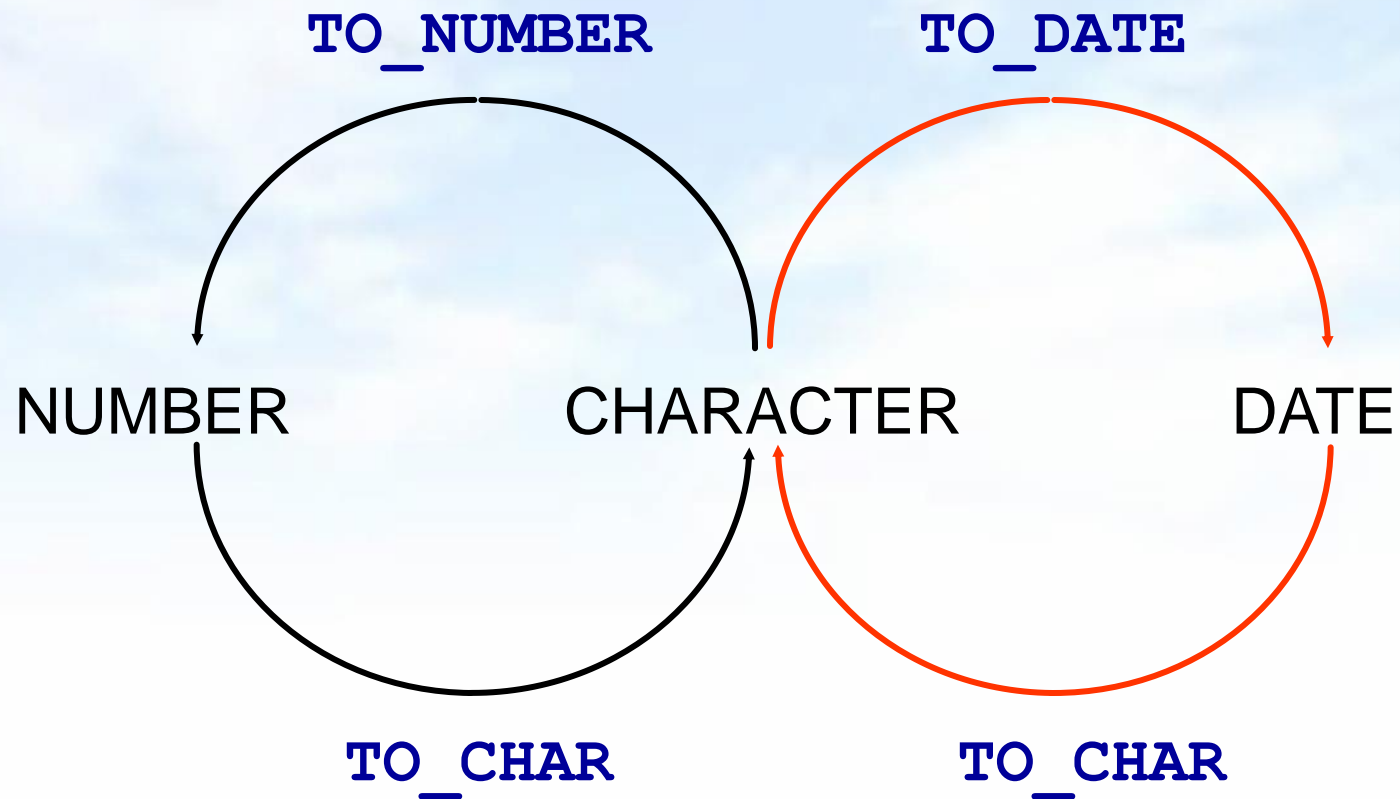
Z
3
6
5

E kifejezés által számolt, az eredményekhez tartozó új attribútum nevét jelöli

Folyt.2.) Kifejezések, sorfüggvények

- 1.) Az SQL-ben halmazok helyett **multihalmazokat** használunk (vagyis egy sor többször is előfordulhat)
- 2.) **SELECT ... FROM ... WHERE ...** lekérdezésekben vagyis $\Pi_{\text{select-lista}} \sigma_{\text{where-feltétel}}$ (from-lista táblák szorzata) a select-listán és where-feltételben az attribútumnevek helyén olyan **kifejezések** állhatnak az SQL-ben, amely függvényeket és műveleti jeleket is tartalmazhat
- Az attribútumnevek helyén álló kifejezésekben használt **legfontosabb sorfüggvényeket, lásd az SQL gyakorlaton:**
 - Numerikus, karakteres, dátum, konverziós függvények
 - NULL hiányzó értéket megadott értékkel helyettesítő függvények, például NVL, COALESCE használata, stb.
 - ... részletesen, lásd az SQL gyakorlatok példáit ...

Adattípusok és adattípus-konverzió



A nullértékek helyettesítése

- **NVL függvény**
- A nullértéket a megadott értékkel helyettesíti:
 - Az adattípus lehet dátum, karakteres, numerikus.
 - Az argumentumok adattípusának egyezőnek kell lenniük:
 - Ha jutalék numerikus: `NVL(jutalék,100)`
 - Ha belépés dátum: `NVL(belépés_dátuma, TO_DATE('01-JAN-1997', 'DD-MON-YYYY', 'NLS_DATE_LANGUAGE = Hungarian'))`
 - Ha név karakteres: `NVL(név, 'nincs még')`

3.) Összesítő (aggregáló) függvények

- az összesítő függvény csoportosított sorok halmazain működik, és egyetlen eredményt ad vissza csoportonként.

EMPLOYEES

DEPARTMENT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
50	5800
50	3500
50	3100
50	2600
50	2500
80	10500
80	11000
80	8600
	7000
10	4400

A legmagasabb
fizetés az
EMPLOYEES
táblában

MAX(SALARY)
24000

20 rows selected.

Összesítő (aggregáló) függvények

- Miért hívják **aggregáló** függvényeknek?
- Ha kiszámoltuk az összeget a tábla bizonyos soraira, akkor újabb sorok figyelembe vételével (aggregálva) felhasználhatjuk a korábban kapott eddigi összeget
- Kivéve például az AVG esetén a fenti nem igaz, viszont az AVG érték hányadosa a SUM és COUNT értékeknek, amelyeket aggregálva tudunk megkapni.

R =

A	B
1	3
3	4
3	2

$$\text{SUM}(A) = 7$$

$$\text{COUNT}(A) = 3$$

$$\text{MIN}(B) = 2$$

$$\text{MAX}(B) = 4$$

$$\text{AVG}(B) = 3$$

Adatcsoportok létrehozása

EMPLOYEES

DEPARTMENT_ID	SALARY
10	4400
20	13000
20	6000
50	5800
50	3500
50	3100
50	2500
50	2600
60	9000
60	6000
60	4200
80	10500
80	8600
80	11000
90	24000
90	17000

...

20 rows selected.

4400

9500

3500

6400

10033

Az
EMPLOYEES
tábla
osztályai
és azokon az
átlagfizetések

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333
90	19333.3333
110	10150
	7000

Csoportosítás több oszlopnév alapján

EMPLOYEES

DEPARTMENT_ID	JOB_ID	SALARY
90	AD_PRES	24000
90	AD_VP	17000
90	AD_VP	17000
60	IT_PROG	9000
60	IT_PROG	6000
60	IT_PROG	4200
50	ST_MAN	5800
50	ST_CLERK	3500
50	ST_CLERK	3100
50	ST_CLERK	2600
50	ST_CLERK	2500
80	SA_MAN	10500
80	SA_REP	11000
80	SA_REP	8600
...		
20	MK_REP	6000
110	AC_MGR	12000
110	AC_ACCOUNT	8300

20 rows selected.

Az
EMPLOYEES
tábla
osztályain
beosztások
szerint
a fizetések
összege

DEPARTMENT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
50	ST_CLERK	11700
50	ST_MAN	5800
60	IT_PROG	19200
80	SA_MAN	10500
80	SA_REP	19600
90	AD_PRES	24000
90	AD_VP	34000
110	AC_ACCOUNT	8300
110	AC_MGR	12000
	SA_REP	7000

13 rows selected.

Összesítések és csoportosítás --- 2

- A csoportosítást (**GROUP BY**), a csoportokon végezhető **összesítő függvényeket** (AVG, SUM, COUNT, MIN, MAX, stb...) reprezentálja a művelet, jele: **γ_L** (gamma)
- Itt az **L** lista valamennyi eleme a következők egyike:
 - R olyan attribútuma, amely szerepel a GROUP BY záradékban, egyike a **csoportosító attribútumoknak**.
 - R egyik attribútumára (ez az **összesítő attribútum**) alkalmazott **összesítő operátor**.
 - Ha az összesítés eredményére névvel szeretnénk hivatkozni, akkor nyilat és új nevet használunk.

Összesítések és csoportosítás --- 3

- **Értelmezése, kiértékelése:** Osszuk az R tábla sorait csoportokba. Egy csoport azokat a sorokat tartalmazza, amelyek az L listán szereplő **csoportosítási attribútumokhoz** tartozó értékei megegyeznek
 - Vagyis ezen attribútumok minden egyes különböző értéke egy csoportot alkot.
- Minden egyes csoporthoz számoljuk ki az L lista **összesítési attribútumaira** vonatkozó összesítéseket
- **Az eredmény** minden egyes csoportra egy sor:
 - Eredmény: a csoportosítási attribútumok és
 - az összesítési attribútumra vonatkozó összesítések (az adott csoport összes sorára)

Példa: Összesítés és csoportosításra

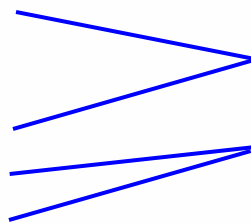
R =

A	B	C
1	2	3
4	5	6
1	2	5

$\gamma_{A,B,AVG(C)} \rightarrow X (R) = ??$

Először csoportosítunk

A	B	C
1	2	3
1	2	5
4	5	6



majd csoportonként
összesítünk:

A	B	X
1	2	4
4	5	6

4.) Kiválasztott sorok rendezése

- **Rendezés:** $\tau_{A_1, \dots, A_n}(R)$
- Először A_1 attribútum szerint rendezzük R sorait. Majd azokat a sorokat, amelyek értéke megegyezik az A_1 attribútumon, A_2 szerint, és így tovább.
- Az **ORDER BY** reprezentálására szolgál. A jele: τ (tau)
- Ez az egyetlen olyan művelet, amelynek az eredménye nem halmaz és nem multihalmaz, hanem rendezett lista.

$R =$

A	B
1	2
3	4
5	2

$$\tau_B(R) = [(1,2), (5,2), (3,4)]$$

Nézzük meg a kiterjesztett algebra műveleteit az SQL SELECT-ben

- **Emlékeztető:** Az előadások SQL lekérdezései az alábbi **Sörivók adatbázissémán** alapulnak

Sörök(név, gyártó)

Sörözők(név, város, tulaj, engedély)

Sörivók(név, város, tel)

Szeret(név, sör)

Felszolgál(söröző, sör, ár)

Látogat(név, söröző)

SQL: Ismétlődések megszüntetése

- SELECT **DISTINCT** ... FROM ...
- A δ művelet SQL-beli megfelelője, amellyel az eredményben kiszűrjük a duplikátumokat, vagyis multihalmazból halmazt állítunk elő.

SQL: Összesítések (aggregálás)

- **Összesítések** (aggregáló műveletek) a **SELECT** listán alkalmazhatjuk egy oszlopra (kifejezésre).
<Aggregáló művelet>(kifejezés) [[AS] onév], ...
- **Az 5 legfontosabb összesítő függvény:**
SUM, **COUNT**, **MIN**, **MAX** (aggregálással számolható),
AVG (bevezették ezt is, mivel gyakran kell AVG és aggregálással számolható függvényekből számolható)
- **Példa:** A **Felszolgál(söröző, sör, ár)** tábla segítségével adjuk meg a Bud átlagos árát:

```
SELECT AVG(ár)  
FROM Felszolgál  
WHERE sör = 'Bud' ;
```

Összesítések (aggregálás)

- Itt is fontos a **halmaz, multihalmaz** megkülönböztetés.
- Aggregáló_művelet(ALL|DISTINCT R.A)
ALL (ez az alapértelmezés, ha nincs jelezve, akkor ALL),
ha DISTINCT szerepel, akkor csak a különböző
értékűeket veszi figyelembe az összesítéseknél.
- NULL nem számít a SUM, AVG, COUNT, MIN, MAX
függvények kiértékelésekor. (implementáció függő,
ellenőrizzük le)
- De ha nincs NULL értéktől különböző érték az
oszlopban, akkor az összesítés eredménye NULL.
- Kivétel: COUNT az üres halmazon 0-t ad vissza.
- COUNT(*) az eredmény sorainak számát adja meg.

NULL értékek nem számítanak az összesítésben, kivéve COUNT(*)

```
SELECT count(*)  
FROM Felszolgal  
WHERE sör = 'Bud' ;
```

A Bud sört árusító kocsmák száma, üres halmazon 0-t ad

```
SELECT count(ár)  
FROM Felszolgal  
WHERE sör = 'Bud' ;
```

A Bud sört ismert áron árusító kocsmák száma, üres halmazon NULL-t ad.

Ismétlődések kiküszöbölése összesítésben (DISTINCT)

- Az összesítő függvényen belül DISTINCT.
- **Példa:** hány *különféle* áron árulják a Bud sört?

```
SELECT COUNT(DISTINCT ár)  
FROM Felszolgál  
WHERE sör = 'Bud' ;
```

SQL: Csoportosítás

- **SELECT ...**
FROM ...
[WHERE ...]
[GROUP BY kif_1 , ... kif_k]
- Egy SELECT-FROM-WHERE kifejezést **GROUP BY záradékkal** folytathatunk, melyet attribútumok (kifejezések) listája követ.
- A SELECT-FROM-WHERE eredménye a megadott attribútumok értékei szerint csoportosítódik, az összesítéseket ekkor minden csoportra külön alkalmazzuk.

Példa: Csoportosítás

- A **Felszolgál(bár, sör, ár)** tábla segítségével adjuk meg a sörök átlagos árát.

```
SELECT sör, AVG(ár)
FROM Felszolgál
GROUP BY sör;
```

sör	AVG(ár)
Bud	2.33
Miller	2.45

A SELECT lista és az összesítések

- Ha **összesítés** is szerepel a lekérdezésben, a SELECT-ben felsorolt attribútumok
 - vagy egy összesítő függvény paramétereiként szerepelnek,
 - vagy a GROUP BY attribútumlistájában is megjelennek.

Az összesítő függvények csak két mélységig ágyazhatóak egymásba

- A **Felszolgál(bár, sör, ár)** tábla segítségével fejezzük ki szavakkal mit jelent az alábbi két lekérdezés? Melyik adhat nagyobb eredményt?

a.) **SELECT MAX (AVG (ár))**
FROM Felszolgál
GROUP BY sör ;

b.) **SELECT AVG (MAX (ár))**
FROM Felszolgál
GROUP BY sör ;

Csoportok szűrése: HAVING záradék

- A GROUP BY záradékot egy **HAVING <feltétel>** záradék követheti.
- HAVING feltétel az egyes csoportokra vonatkozik, ha egy csoport nem teljesíti a feltételt, nem lesz benne az eredményben.
- csak olyan attribútumok szerepelhetnek, amelyek:
 - vagy csoportosító attribútumok,
 - vagy összesített attribútumok.(vagyis ugyanazok a szabályok érvényesek, mint a SELECT záradéknál).

SQL: Az eredmény rendezése

- SQL SELECT utasítás utolsó záradéka: **ORDER BY**
- Az SQL lehetővé teszi, hogy a lekérdezés eredménye bizonyos sorrendben legyen rendezve. Az első attribútum egyenlősége esetén a 2.attribútum szerint rendezve, stb, minden attribútumra lehet növekvő vagy csökkenő sorrend.
- Select-From-Where utasításhoz a következő záradékot adjuk, a WHERE záradék és minden más záradék (mint például GROUP BY és HAVING) után következik:

SELECT ... FROM ... [WHERE ...] [...]

ORDER BY {attribútum [DESC], ...}

- **Példa: SELECT * FROM Felszolgál
ORDER BY ár DESC, sör**

Összefoglalás: SELECT utasítás záradékai

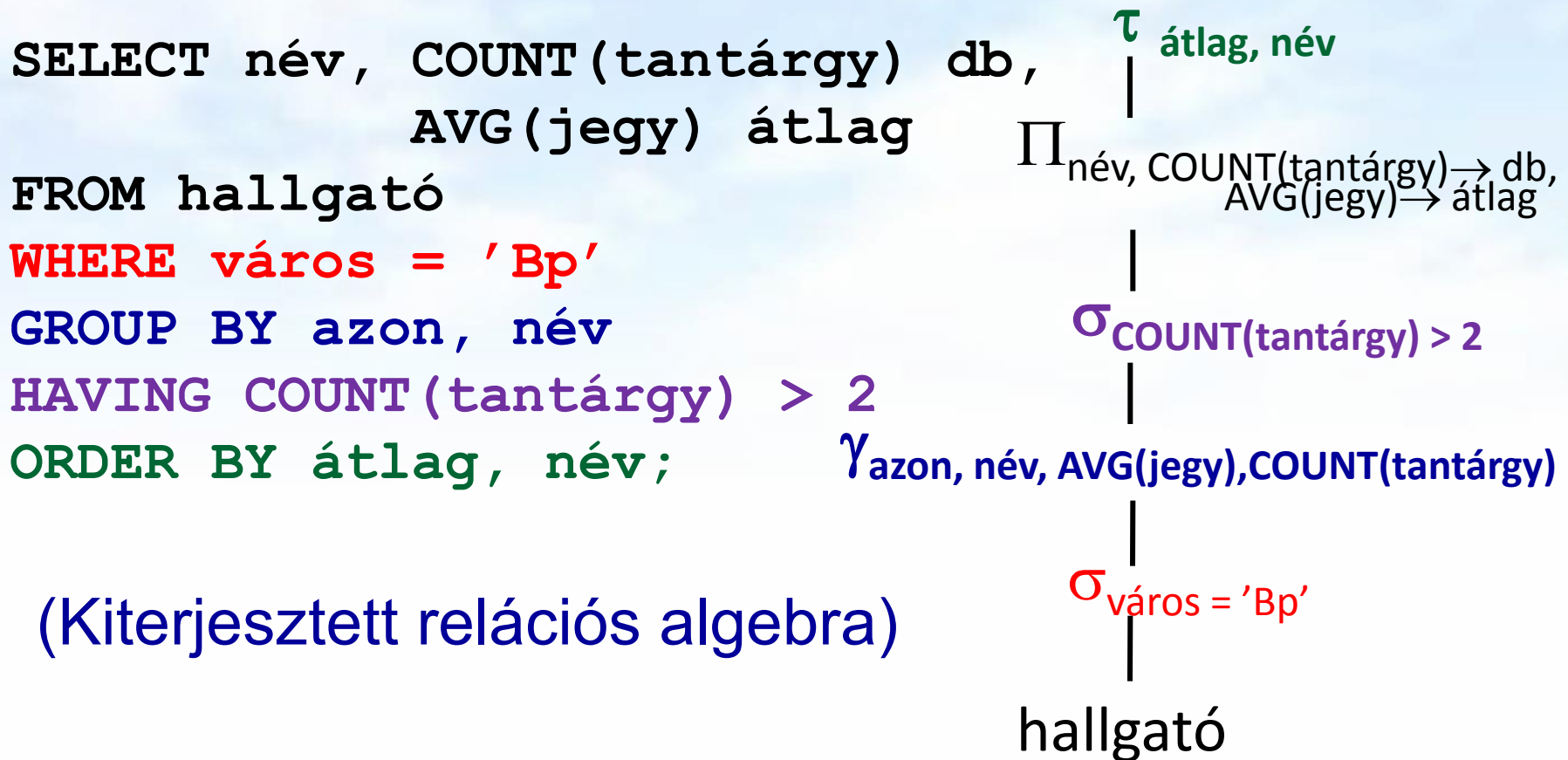
- Teljes SELECT utasítás(a záradékok sorrendje adott)

SELECT [DISTINCT] Lista1	-- 5 és 6
FROM R t	-- 1
[WHERE Felt1]	-- 2
[GROUP BY csopkif	-- 3
[HAVING Felt2]]	-- 4
[ORDER BY Lista2]	-- 7

$$\tau_{\text{Lista2}} \left(\delta \left(\Pi_{\text{Lista1}} \sigma_{\text{Felt2}} \left(\gamma_{\text{csopkif, AGGR(kif)}} \sigma_{\text{Felt1}} (R) \right) \right) \right)$$

Példa: group by, having és order by

Példa: hallgató (azon, név, város, tantárgy, jegy)



Kérdés / Válasz

- Köszönöm a figyelmet! Kérdés/Válasz?

Feladatok

- **Házi feladat: Gyakorlás az Oracle Példatár feladatai:**
- Példatár 1.-2. fejezetek feladatai SQL-lekérdezésekben kifejezések, függvények, összesítések és csoportosítás, sorok rendezése feladatok egy táblás lekérdezések
- **Keressünk új megoldásokat!** „Amikor azt gondolod, hogy már minden lehetőséget kimerítettél, még mindig van legalább egy.” (Thomas Alva Edison)