

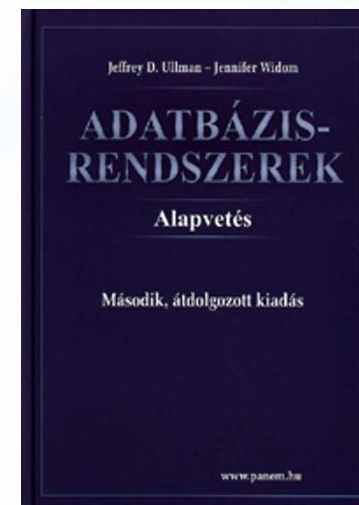
**5.előadás: Adatbázisok**  
**dr. Hajas Csilla (ELTE IK)**  
<http://sila.hajas.elte.hu/>

# **SQL SELECT - FROM – WHERE**

## **Alkérdeések - Összekapcsolások**

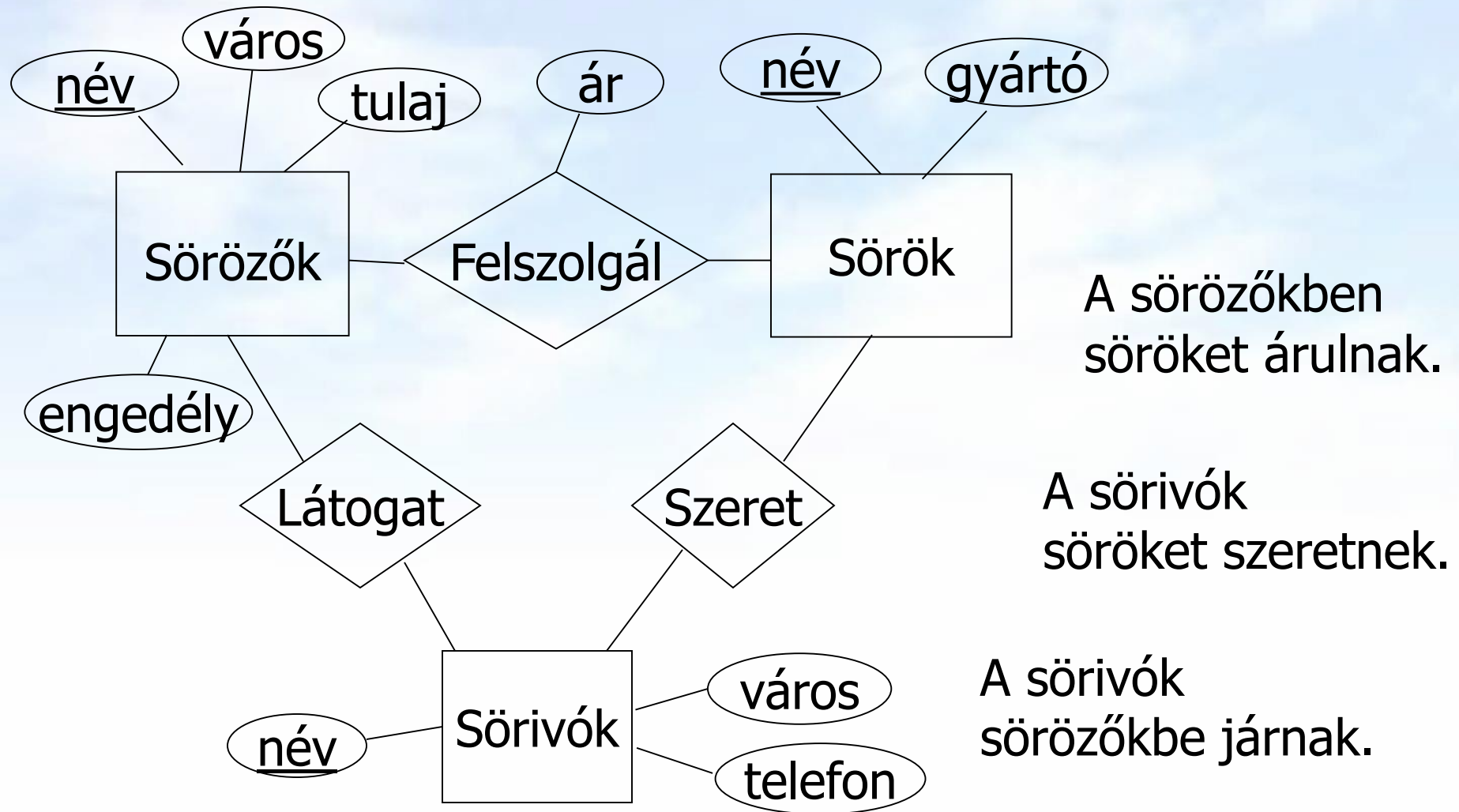
Tankönyv:

- 6.3.1-6.3.4. Alkérdeések WHERE-ben
- 6.3.5. Alkérdeések FROM záradékban
- 6.3.6. Összekapcsolások az SQL-ben
- 6.3.7. Természetes összekapcsolás
- 6.3.8. Külső összekapcsolások



# [Emlékeztető]

## Az előadás példája (E/K diagram)



# [Emlékeztető]

## Az előadás példája (relációs sémák)

- Az egyedhalmazok átírása  
(aláhúzás jelöli a kulcs attribútumokat)

**Sörözők(név, város, tulaj, engedély)**

**Sörök(név, gyártó)**

**Sörivők(név, város, telefon)**

- A sok-sok kapcsolatok átírása (és átnevezéssel:  
sör hivatkozási épség megszorítás a Sörök.név-re,  
söröző hivatk.épség pedig a Sörözők.név-re szól)

**Felhasználó(söröző, sör, ár)**

**Látogat(név, söröző)**

**Szeret(név, sör)**

# (2EA) ismétlés: Select-From-Where (SFW)

- Gyakran előforduló relációs algebrai kifejezés  
 $\Pi_{\text{Lista}} ( \sigma_{\text{Felt}} (R_1 \times \dots \times R_n) )$  típusú kifejezések
- Szorzat és összekapcsolás az SQL-ben
- SELECT s-lista -- milyen típusú sort szeretnénk az eredményben látni?  
FROM f-lista -- relációk (táblák) összekapcsolása, illetve szorzata  
WHERE felt -- milyen feltételeknek eleget tevő sorokat kell kiválasztani?
- FROM f-lista elemei (ezek ismétlődhetnek)  
táblanév [[AS] sorváltozó, ...]  
Itt: a from lista elemei a táblák direkt szorzatát jelenti, az összekapcsolási feltételt where-ben adjuk meg, később bevezetünk majd tovább lehetőségeket a különböző összekapcsolásokra az SQL from záradékában.

## (2EA) ismétlés: SFW alapértelmezése

- Tk.6.2.fej.: **2.EA: Több táblára vonatkozó lekérdezések:**

SELECT [DISTINCT] kif<sub>1</sub> [[AS] onév<sub>1</sub>], ..., kif<sub>n</sub> [[AS] onév<sub>n</sub>]  
FROM R<sub>1</sub> [t<sub>1</sub>], ..., R<sub>n</sub> [t<sub>n</sub>]  
WHERE feltétel (vagyis logikai kifejezés)

Alapértelmezés (a műveletek szemantikája -- általában)

- A FROM záradékban levő relációkhoz tekintünk egy-egy **sorváltozót**, amelyek a megfelelő reláció minden sorát bejárják (beágyazott ciklusban)
- Minden egyes „aktuális” sorhoz kiértékeljük a WHERE záradékot (csak az igaz sorok kerülnek az eredménybe)
- A SELECT záradékban szereplő kifejezéseknek megfelelően képezzük a sorokat. Ha van DISTINCT, akkor az ismétlődő sorokat elhagyjuk.

# (2EA) ismétlés: Halmazműveletek az SQL-ben

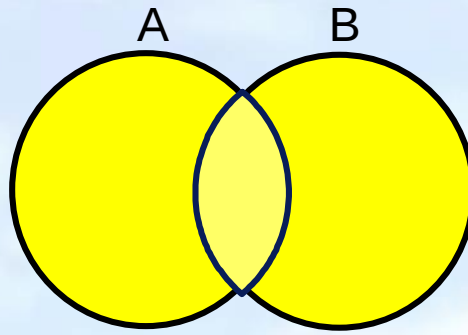
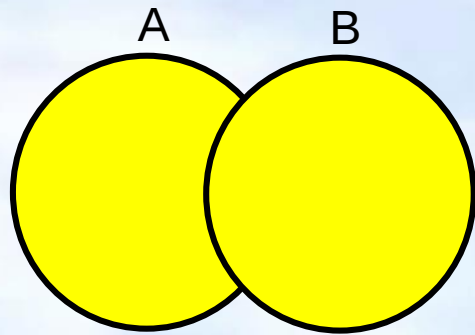
- A relációs algebrai halmazműveletek: **unió**, **különbség** és **metszet**, ebből csak az unió és különbség alpművelet, az SQL-ben mindhárom használható, implementálva van
- A **SELECT-FROM-WHERE** utasítások általában **multihalmaz** szemantikát használnak, külön kell kérni **DISTINCT**-tel ha halmazt szeretnénk kapni, viszont a **halmazműveleteknél** a **halmaz szemantika** az érvényes, (később lesz a **multihalmaz szemantika**, pl. **UNION ALL** )
- Az SQL-ben a halmazműveleteket úgy vezették be, hogy azt mindig két lekérdezés között lehet értelmezni:

(SFW-lekérdezés1)

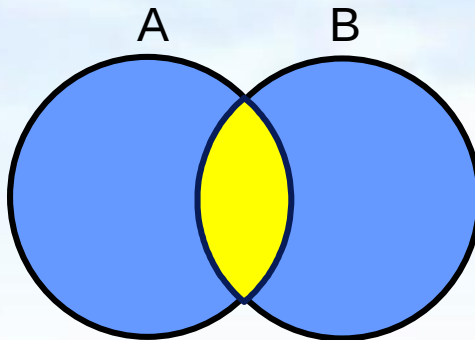
[ **UNION** |  
INTERSECT |  
{**EXCEPT** | **MINUS**} ]

(SFW-lekérdezés2);

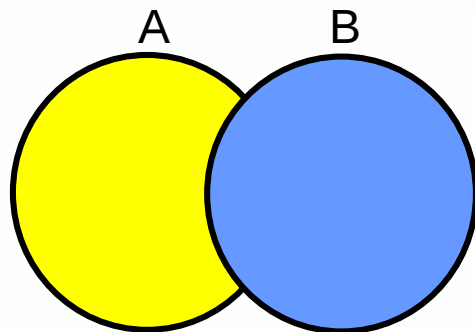
# (2EA) ismétlés: Halmazműveletek (Oracle)



UNION/UNION ALL



INTERSECT



MINUS

ORACLE-ben a halmazműveleti különbség műveletre az EXCEPT helyett a MINUS szót használjuk.

(Megj.: multialmazként csak UNION ALL van.)

# 6.3.1.-6.3.5. Alkérdeések

# Alkérdeések (áttekintés)

- Zárójelezett SFW SELECT-FROM-WHERE utasításokat **(alkérdeéseket)** is használhatunk a **FROM listán** a **WHERE záradékban**, és a **HAVING záradékban**.
- Szintaktikus alakja: zárójelbe kell tenni a lekérdezést
- Hol használható?

- (1) **WHERE záradékban**: sorokra vonatkozó kiválasztási feltételekben
- (2) **HAVING záradékban** (később lesz a többi záradék)
- (3) **FROM listában**: új listaelem (rel.név változó SQL-ben)  
**(lekérdezés) [AS] sorváltozó**

Ez felel meg annak, ahogyan a relációs algebrában tetsz.helyen használhattuk a lekérdezés eredményét.

# Alkérdeések a WHERE záradékban

## WHERE záradékban:

- (i) Az alkérdés eredménye egyetlen **skalárérték**, vagyis az alkérdés olyan, mint a konstans, ami egy új elemi kifejezésként tetszőleges kifejezésben használható.
- (ii) **Skalár értékekből álló multihalmaz** logikai kifejezésekben használható: [NOT] EXISTS (lekérdezés)  
kifejezés [NOT] IN (lekérdezés)  
kifejezés  $\Theta$  [ANY | ALL] (lekérdezés)
- (iii) **Teljes, többdimenziós tábla** a visszatérő érték:  
[NOT] EXISTS (lekérdezés)  
(kif<sub>1</sub>, ... kif<sub>n</sub>) [NOT] IN (lekérdezés)

# Alkérdések a WHERE záradékban

- Milyen változók szerepelhetnek egy alkérdésben?
  - Lokális saját változói a saját FROM listáról
  - Külső kérdés változói: ekkor az alkérdés korrelált.
- **Definíció:** Ha az alkérdés a külső kérdés FROM listában szereplő tábláiból származó attribútumot is tartalmaz, akkor azt **korrelált alkérdés**nek nevezzük.

## Szemantikája

- Ha az alkérdés **nem korrelált**, önállóan kiértékelhető és ez az eredmény a külső kérdés közben nem változik, a külső kérdés szempontjából ez egy konstanstábla, akkor a kiértékelés mindig a legbelsőből halad kifelé.
- **Korrelált alkérdés**, lásd később az EXISTS példánál

# Skalár értéket visszaadó alkérdések

- Ha egy alkérdés biztosan egy attribútumon egy sort ad vissza eredményként (egyelemű), akkor úgy használható, mint egy konstans érték.
  - az eredmény sornak egyetlen oszlopa van.
  - Futásidejű hiba keletkezik, ha az eredmény nem tartalmaz sort, vagy több sort tartalmaz.
- **Példa:** Felszolgál(söröző, sör, ár) táblában keressük meg azokat a sörözőket, ahol a Miller ugyanannyiba kerül, mint Joe-bárjában a Bud.
- Két lekérdezésre biztos szükségünk lesz:
  1. Mennyit kér Joe a Budért?
  2. Melyik kocsmákban adják ugyanennyiért a Millert?

# Skalár értéket visszaadó alkérdések

**Példa:** Felszolgál(söröző, sör, ár) táblában keressük meg azokat a sörözőket, ahol a Miller ugyanannyiba kerül, mint Joe bárjában a Bud.

```
SELECT söröző
```

```
FROM Felszolgál
```

```
WHERE sör = 'Miller' AND
```

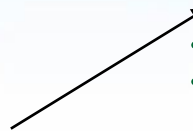
```
    ár = (SELECT ár
```

```
        FROM Felszolgál
```

```
        WHERE söröző = 'Joe' 's bar'
```

```
        AND sör = 'Bud') ;
```

Ennyit kér  
Joe a Budért.



**H.F:** Adjunk meg ugyanezt alkérdés használata nélkül is!

# Példa: Skalár értékű alkérdés

```
SELECT sör  
FROM Felszolgal  
WHERE ár = (  
    SELECT MAX(ár)  
    FROM Felszolgal);
```

(Itt az alkérdés egy  
skalár értéket ad vissza  
a legdrágább sör árát)



# Skalár értékekből álló multihalmazt visszaadó alkérdések: ANY művelet

- $x = \text{ANY}(\text{alkérdés})$  akkor és csak akkor igaz, ha  $x$  egyenlő az alkérdés legalább egy sorával.  
= helyett bármilyen aritmetikai összehasonlítás szerepelhet.
- **Példa:**  $x > \text{ANY}(\text{alkérdés})$  akkor igaz, ha  $x$  nem az alkérdés legkisebb elemével azonos.
  - Itt az alkérdés sorai egy mezőből állnak.

# Skalár értékekből álló multihalmazt visszaadó alkérdések: ALL művelet

- $x \neq$  ALL(alkérdés) akkor és csak akkor igaz, ha  $x$  az alkérdés egyetlen sorával sem egyezik meg.
- $\neq$  helyett tetszőleges összehasonlítás szerepelhet.
- **Példa:**  $x \geq$  ALL(alkérdés)  $x$  az alkérdés eredményének maximum értékével azonos.

# Példa: ALL

```
SELECT sör  
FROM Felszolgál  
WHERE ár >= ALL(  
    SELECT ár  
    FROM Felszolgál);
```

(A külső lekérdezés Felszolgáljának sör ára minden alkérdésbeli sör ára közül a legnagyobb, azaz egyetlen alkérdésbeli sörnél sem lehet olcsóbb.)

# Az IN művelet a WHERE záradékban

- sor IN (alkérdés) akkor és csak akkor **igaz**, ha a sor eleme az alkérdés eredményének (itt a sor egy sor/tuple, nem sör)
- Tagadás: sor NOT IN (alkérdés).
- Az IN-kifejezések a WHERE záradékban jelenhetnek meg

➤ Példa:

**SELECT \***

**FROM Sörök**

**WHERE név IN (SELECT sör**

**FROM Szeret**

**WHERE név = 'Fred' ) ;**

A sörök,  
melyeket  
Fred szeret.

# Több mélységig beágyazhatóak

- Tankönyv Filmek példájában sorokat tartalmazó feltételek:  
Keressük Harrison Ford filmjeinek gyártásirányítóját

```
SELECT név
FROM GyártásIrányító
WHERE azonosító IN
    (SELECT producerAzon
     FROM Filmek
     WHERE (cím, év) IN
         (SELECT filmCím, filmév
          FROM SzerepelBenne
          WHERE színész = 'Harrison Ford'))
    );
```

# Van-e különbség a kiértékelés között?

```
SELECT a
FROM R, S
WHERE R.b = S.b;
```

| a | b |
|---|---|
| 1 | 2 |
| 3 | 4 |

R

| b | c |
|---|---|
| 2 | 5 |
| 2 | 6 |

S

```
SELECT a
FROM R
WHERE b IN (SELECT b FROM S);
```

# IN az R soraira vonatkozó predikátum

SELECT a

FROM R

WHERE b IN (SELECT b FROM S);

Egy ciklus R sorai  
fölött.

| a | b |
|---|---|
| 1 | 2 |
| 3 | 4 |

R

| b | c |
|---|---|
| 2 | 5 |
| 2 | 6 |

S

(1,2) kielégíti a  
feltételt;  
1 egyszer jelenik  
meg az  
eredményben.

# Itt R és S sorait párosítjuk

```
SELECT a  
FROM R, S  
WHERE R.b = S.b;
```

Dupla ciklus R és S  
sorai fölött

| a | b |
|---|---|
| 1 | 2 |
| 3 | 4 |

R

| b | c |
|---|---|
| 2 | 5 |
| 2 | 6 |

S

(1,2) és (2,5)  
(1,2) és (2,6)  
is kielégíti a  
feltételt;  
1 kétszer kerül  
be az eredménybe.

# Korrelált alkérdés

- **Definíció:** Ha egy alkérdés nem csak lokális változókat használ a saját FROM listájáról, hanem a külső kérdés FROM listában szereplő tábláiból származó attribútumot is tartalmaz, akkor azt **korrelált alkérdés**nek nevezzük.

## Szemantikája

- Ha az alkérdés **nem korrelált**, önállóan kiértékelhető és ez az eredmény a külső kérdés közben nem változik, a külső kérdés szempontjából ez egy konstanstábla, akkor a kiértékelés mindig a legbelsőből halad kifelé.
- **Korrelált alkérdés**, többször kerül kiértékelésre, minden egyes kiértékelés megfelel egy olyan értékadásnak, amely az alkérdésen kívüli sorváltozóból származik.

# Az EXISTS művelet a WHERE-ben

- EXISTS (alkérdés) akkor és csak akkor igaz, ha az alkérdés eredménye nem üres.
  - Tagadása: NOT EXISTS (alkérdés)
- **Példa: A Sörök(név, gyártó)** táblában keressük meg azokat a söröket, amelyeken kívül a gyártójuk nem gyárt másikat.
- Ez korrelált alkérdés, többször kerül kiértékelésre, a külső tábla minden sorára kiértékeljük az alkérdést.
- A korrelált lekérdezések használata közben figyelembe kell vennünk a nevek érvényességi körére vonatkozó szabályokat.

# Példa: EXISTS

```
SELECT név  
FROM Sörök s1  
WHERE NOT EXISTS
```

```
(SELECT *
```

```
FROM Sörök
```

```
WHERE gyártó = s1.gyártó
```

```
AND név <> s1.név) ;
```

Azon s1  
sörtől  
különböző  
sörök,  
melyeknek  
ugyanaz  
a gyártója.

Változók láthatósága: itt  
a gyártó a legközelebbi  
beágyazott FROM-beli  
Táblából való, aminek  
van ilyen attribútuma.

A „nem egyenlő”  
művelet SQL-ben.

# Tk.példa: Korrelált alkérdés

- A több, mint egyszer előforduló filmcímek megkeresése:

```
SELECT DISTINCT cím
FROM Filmek Régi
WHERE év < ANY
    (SELECT év
     FROM Filmek
     WHERE cím = Régi.cím
    ) ;
```

# Alkérdeések használata FROM listán

- **FROM záradékban** a tényleges relációk helyett alkérdeéssel létrehozott ideiglenes táblát is megadhatunk. Minthogy az alkérdeés eredményrelációjának nincs neve, meg kell adnunk egy másodnevet (a sorváltozó nevét), ezzel tudunk az alkérdeés által létrehozott reláció soraira hivatkozni.
- **Szintaktikus alakja:**  
(lekérdezés) [AS] sorváltozó
- Ez felel meg annak, ahogyan a relációs algebrában tetsz.helyen használhattuk a lekérdezés eredményét.
- **Szemantikája:** A FROM záradékban kiértékelődik az alkérdeés, utána a sorváltozót ugyanúgy használjuk, mint a FROM lista többi közösleges relációk soraira.

# Alkérdeések használata FROM listán

- **Példa:** Keressük meg a Joe's bár vendégei által kedvelt söröket (a feladatnak sok megoldása van)

```
SELECT sör
FROM Szeret SZ,
      (SELECT név FROM Látogat
       WHERE söröző = 'Joe' 's bar') JD
WHERE SZ.név = JD.név;
```

Sörivók, akik látogatják  
Joe's bárja sörözőt.



- **H.F.:** Írjuk át egyszerű SELECT FROM utasításra, ahol csak táblák vannak, nincs alkérdés a FROM-ban

**6.3.6.-6.3.8.**

**Összekapcsolások az SQL-ben,  
természetes összekapcsolás,  
külső összekapcsolások  
és  
az Oracle gyakorlatok alapján**

# Összekapcsolások a FROM listán

- Az SQL-ben összekapcsolások számos változata megtalálható: Természetes összekapcsolás
- USING utasítással történő összekapcsolás
- Teljes (vagy két oldali) külső összekapcsolás
- Tetszőleges feltételen alapuló külső összekapcsolás
- Direktszorzat (kereszt-összekapcsolás).

```
SELECT tábla1.oszlop, tábla2.oszlop FROM tábla1  
[NATURAL JOIN tábla2] |  
[JOIN tábla2 USING (oszlopnév)] |  
[JOIN tábla2 ON (tábla1.oszlopnév = tábla2.oszlopnév)]  
[{LEFT | RIGHT | FULL} OUTER JOIN tábla2  
  ON (tábla1.oszlopnév = tábla2.oszlopnév)]  
[CROSS JOIN tábla2]
```

# Oracle gyakorlatok példája: több táblára vonatkozó lekérdezések

**EMPLOYEES**

| EMPLOYEE_ID | DEPARTMENT_ID |
|-------------|---------------|
| 200         | 10            |
| 201         | 20            |
| 202         | 20            |
| 124         | 50            |
| 141         | 50            |
| 142         | 50            |
| 143         | 50            |
| 144         | 50            |
| 103         | 60            |
| 104         | 60            |
| 107         | 60            |
| 149         | 80            |
| 174         | 80            |
| 176         | 80            |

...

**DEPARTMENTS**

| DEPARTMENT_ID | DEPARTMENT_NAME |
|---------------|-----------------|
| 10            | Administration  |
| 20            | Marketing       |
| 20            | Marketing       |
| 50            | Shipping        |
| 50            | Shipping        |
| 50            | Shipping        |
| 50            | Shipping        |
| 50            | Shipping        |
| 50            | Shipping        |
| 60            | IT              |
| 60            | IT              |
| 60            | IT              |
| 80            | Sales           |
| 80            | Sales           |
| 80            | Sales           |

...

**Foreign key**

**Primary key**

- Az osztályok dolgozóinak meghatározásához az EMPLOYEES tábla és a DEPARTMENTS tábla DEPARTMENT\_ID oszlopban szereplő értékeinek összehasonlítása kell. Ez egy egyenlőségen alapuló összekapcsolás lesz. Az ilyen típusú összekapcsolásban általában az elsődleges- és az idegen kulcs komponensei szerepelnek.

# Természetes összekapcsolás

- A NATURAL JOIN utasítás a benne szereplő két tábla azonos nevű oszlopain alapul.
- A két tábla azon sorait eredményezi, ahol az azonos nevű oszlopokban szereplő értékek megegyeznek.
- Ha az azonos nevű oszlopok adattípusa eltérő, akkor hibával tér vissza az utasítás.

# Példa a természetes összekapcsolásra

```
SELECT department_id, department_name, location_id, city
FROM departments
NATURAL JOIN locations;
```

| DEPARTMENT_ID | DEPARTMENT_NAME | LOCATION_ID | CITY                |
|---------------|-----------------|-------------|---------------------|
| 60            | IT              | 1400        | Southlake           |
| 50            | Shipping        | 1500        | South San Francisco |
| 10            | Administration  | 1700        | Seattle             |
| 90            | Executive       | 1700        | Seattle             |
| 110           | Accounting      | 1700        | Seattle             |
| 190           | Contracting     | 1700        | Seattle             |
| 20            | Marketing       | 1800        | Toronto             |
| 80            | Sales           | 2500        | Oxford              |

- A WHERE használható további megszorítások megfogalmazására. Például, ha csak a 20-as illetve 50-es department\_id-kra vagyunk kíváncsiak, akkor:
- ```
SELECT department_id, department_name, location_id, city
FROM departments NATURAL JOIN locations
WHERE department_id IN (20, 50);
```

# Összekapcsolás USING kulcsszóval

- Ha több oszlopnak azonos a neve, de az adattípusa eltérő, akkor a USING segítségével megadható, hogy mely oszlopokat lehet használni az egyenlőségen alapuló összekapcsoláshoz.
- Használjunk USING-ot, ha csak egy oszlop egyezik meg.
- Ne használjuk a tábla eredeti vagy alias nevét a kiválasztott oszlopok megadásánál.
- A NATURAL JOIN és a USING kulcsszavak együttes használata nem megengedett.

# A USING kulcsszó használata lekérdezésben

```
SELECT employees.employee_id, employees.last_name,  
       departments.location_id, department_id  
FROM   employees JOIN departments  
USING (department_id) ;
```

| EMPLOYEE_ID | LAST_NAME | LOCATION_ID | DEPARTMENT_ID |
|-------------|-----------|-------------|---------------|
| 200         | Whalen    | 1700        | 10            |
| 201         | Hartstein | 1800        | 20            |
| 202         | Fay       | 1800        | 20            |
| 124         | Mourgos   | 1500        | 50            |
| 141         | Rajs      | 1500        | 50            |
| 142         | Davies    | 1500        | 50            |
| 144         | Vargas    | 1500        | 50            |
| 143         | Matos     | 1500        | 50            |

■ ■ ■

19 rows selected.

# Azonos nevű oszlopok megkülönböztetése

- Használjuk a táblaneveket előtagként az azonos nevű oszlopok megkülönböztetésére
- A előtagok használata javítja a hatékonyságot is.
- Használhatunk alias neveket az olyan oszlopokra, amelyeket megkülönböztetünk a többi táblában lévő azonos nevű társaiktól.
- Ne használjunk alias nevet azon oszlopokra, amelyeket a USING kulcsszó után adtunk meg és az SQL utasításban még más helyen is szerepelnek.

# Sorváltók használata tábláknál

- A lekérdezések átláthatósága miatt használhatunk sorváltót (tábla alias neveket).
- A sorváltók használata javítja a lekérdezés teljesítményét.
- A sorváltók maximum 30 karakter hosszúak lehetnek (minél rövidebb annál jobb)
- A sorváltók csak az aktuális SELECT utasítás során lesznek használhatóak!

# Összekapcsolások az ON kulcsszó segítségével

- A természetes összekapcsolás alapvetően az azonos nevű oszlopok egyenlőségvizsgálatán alapuló összekapcsolása volt.
- Az ON kulcsszót használhatjuk az összekapcsolás tetszőleges feltételének vagy oszlopainak megadására.
- Az összekapcsolási feltétel független a többi keresési feltételtől.
- Az ON használata áttekinthetőbbé teszi a kódot

# Lekérdezés az ON kulcsszó használatával

```
SELECT e.employee_id, e.last_name, e.department_id,  
       d.department_id, d.location_id  
FROM   employees e JOIN departments d  
ON     (e.department_id = d.department_id);
```

| EMPLOYEE_ID | LAST_NAME | DEPARTMENT_ID | DEPARTMENT_ID | LOCATION_ID |
|-------------|-----------|---------------|---------------|-------------|
| 200         | Whalen    | 10            | 10            | 1700        |
| 201         | Hartstein | 20            | 20            | 1800        |
| 202         | Fay       | 20            | 20            | 1800        |
| 124         | Mourgos   | 50            | 50            | 1500        |
| 141         | Rajs      | 50            | 50            | 1500        |
| 142         | Davies    | 50            | 50            | 1500        |
| 143         | Matos     | 50            | 50            | 1500        |

...

19 rows selected.

- Az ON segítségével különböző nevű oszlopok is összekapcsolhatóak

# Önmagával való összekapcsolás (self-join) az ON kulcsszóval 1.

**EMPLOYEES (WORKER)**

| EMPLOYEE_ID | LAST_NAME | MANAGER_ID |
|-------------|-----------|------------|
| 100         | King      |            |
| 101         | Kochhar   | 100        |
| 102         | De Haan   | 100        |
| 103         | Hunold    | 102        |
| 104         | Ernst     | 103        |
| 107         | Lorentz   | 103        |
| 124         | Mourgos   | 100        |

...

**EMPLOYEES (MANAGER)**

| EMPLOYEE_ID | LAST_NAME |
|-------------|-----------|
| 100         | King      |
| 101         | Kochhar   |
| 102         | De Haan   |
| 103         | Hunold    |
| 104         | Ernst     |
| 107         | Lorentz   |
| 124         | Mourgos   |

...



**A WORKER tábla Manager\_ID mezője megfelel  
a MANAGER tábla EMPLOYEE\_ID mezőjével**

# Önmagával való összekapcsolás (self-join) az ON kulcsszóval 2

```
SELECT e.last_name emp, m.last_name mgr
FROM   employees e JOIN employees m
ON     (e.manager_id = m.employee_id);
```

| EMP       | MGR  |
|-----------|------|
| Hartstein | King |
| Zlotkey   | King |
| Mourgos   | King |
| De Haan   | King |
| Kochhar   | King |

---

19 rows selected.

# Három-utas összekapcsolás ON segítségével

```
SELECT employee_id, city, department_name
FROM   employees e
JOIN   departments d
ON     d.department_id = e.department_id
JOIN   locations l
ON     d.location_id = l.location_id;
```

| EMPLOYEE_ID | CITY                | DEPARTMENT_NAME |
|-------------|---------------------|-----------------|
| 103         | Southlake           | IT              |
| 104         | Southlake           | IT              |
| 107         | Southlake           | IT              |
| 124         | South San Francisco | Shipping        |
| 141         | South San Francisco | Shipping        |
| 142         | South San Francisco | Shipping        |
| 143         | South San Francisco | Shipping        |
| 144         | South San Francisco | Shipping        |

- Három tábla összekapcsolását nevezzük három-utas összekapcsolásnak
- Az SQL 1999-es szintaxis szerint az ilyen összekapcsolások balról jobbra
- haladva hajtódnak végre (DEPARTMENTS – EMPLOYEES) – LOCATION

# Nem egyenlőségvizsgálaton alapuló összekapcsolás

**EMPLOYEES**

| LAST_NAME | SALARY |
|-----------|--------|
| King      | 24000  |
| Kochhar   | 17000  |
| De Haan   | 17000  |
| Hunold    | 9000   |
| Ernst     | 6000   |
| Lorentz   | 4200   |
| Mourgos   | 5800   |
| Rajs      | 3500   |
| Davies    | 3100   |
| Matos     | 2600   |
| Vargas    | 2500   |
| Zlotkey   | 10500  |
| Abel      | 11000  |
| Taylor    | 8600   |

...  
20 rows selected.

**JOB\_GRADES**

| GRA | LOWEST_SAL | HIGHEST_SAL |
|-----|------------|-------------|
| A   | 1000       | 2999        |
| B   | 3000       | 5999        |
| C   | 6000       | 9999        |
| D   | 10000      | 14999       |
| E   | 15000      | 24999       |
| F   | 25000      | 40000       |

Az EMPLOYEES tábla  
fizetés mezőjének értéke  
a JOBS\_GRADE tábla  
legmagasabb illetve  
legalacsonyabb fizetés  
közötti kell legyen.

# Példa a nem egyenlőségvizsgálaton alapuló összekapcsolás

```
SELECT e.last_name, e.salary, j.grade_level
FROM   employees e JOIN job_grades j
ON     e.salary
      BETWEEN j.lowest_sal AND j.highest_sal;
```

| LAST_NAME | SALARY | GRA |
|-----------|--------|-----|
| Matos     | 2600   | A   |
| Vargas    | 2500   | A   |
| Lorentz   | 4200   | B   |
| Mourgos   | 5800   | B   |
| Rajs      | 3500   | B   |
| Davies    | 3100   | B   |
| Whalen    | 4400   | B   |
| Hunold    | 9000   | C   |
| Ernst     | 6000   | C   |

...

# Külső összekapcsolás

## DEPARTMENTS

| DEPARTMENT_NAME | DEPARTMENT_ID |
|-----------------|---------------|
| Administration  | 10            |
| Marketing       | 20            |
| Shipping        | 50            |
| IT              | 60            |
| Sales           | 80            |
| Executive       | 90            |
| Accounting      | 110           |
| Contracting     | 190           |

8 rows selected.

## EMPLOYEES

| DEPARTMENT_ID | LAST_NAME |
|---------------|-----------|
| 90            | King      |
| 90            | Kochhar   |
| 90            | De Haan   |
| 60            | Hunold    |
| 60            | Ernst     |
| 60            | Lorentz   |
| 50            | Mourgos   |
| 50            | Rajs      |
| 50            | Davies    |
| 50            | Matos     |
| 50            | Vargas    |
| 80            | Zlotkey   |

...  
20 rows selected.

**A 190-es számú osztályon  
nincs alkalmazott**

# Belső vagy külső összekapcsolás?

- SQL-1999: Belső összekapcsolásnak nevezzük azokat az összekapcsolásokat, amelyek két tábla megegyező soraival térnek vissza.
- Két tábla olyan összekapcsolását, amely a belső összekapcsolás eredményéhez hozzáveszi a bal (vagy jobboldali) tábla összes sorát, baloldali (vagy jobboldali) külső összekapcsolásnak nevezzük.
- Teljes külső összekapcsolásnak hívjuk azt az esetet, amikor a külső összekapcsolás egyszerre bal- és jobboldali.

# Baloldali külső összekapcsolás

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e LEFT OUTER JOIN departments d
ON (e.department_id = d.department_id) ;
```

| LAST_NAME | DEPARTMENT_ID | DEPARTMENT_NAME |
|-----------|---------------|-----------------|
| Whalen    | 10            | Administration  |
| Fay       | 20            | Marketing       |
| Hartstein | 20            | Marketing       |
| ...       |               |                 |
| De Haan   | 90            | Executive       |
| Kochhar   | 90            | Executive       |
| King      | 90            | Executive       |
| Gietz     | 110           | Accounting      |
| Higgins   | 110           | Accounting      |
| Grant     |               |                 |

20 rows selected.

# Jobboldali külső összekapcsolás

```
SELECT e.last_name, e.department_id, d.department_name
FROM employees e RIGHT OUTER JOIN departments d
ON (e.department_id = d.department_id) ;
```

| LAST_NAME | DEPARTMENT_ID | DEPARTMENT_NAME |
|-----------|---------------|-----------------|
| Whalen    | 10            | Administration  |
| Fay       | 20            | Marketing       |
| Hartstein | 20            | Marketing       |
| Davies    | 50            | Shipping        |
| ...       |               |                 |
| Kochhar   | 90            | Executive       |
| Gietz     | 110           | Accounting      |
| Higgins   | 110           | Accounting      |
|           | 190           | Contracting     |

20 rows selected.

# Teljes külső összekapcsolás

```
SELECT e.last_name, d.department_id, d.department_name
FROM employees e FULL OUTER JOIN departments d
ON (e.department_id = d.department_id) ;
```

| LAST_NAME | DEPARTMENT_ID | DEPARTMENT_NAME |
|-----------|---------------|-----------------|
| Whalen    | 10            | Administration  |
| Fay       | 20            | Marketing       |
| Hartstein | 20            | Marketing       |
| ...       |               |                 |
| King      | 90            | Executive       |
| Gietz     | 110           | Accounting      |
| Higgins   | 110           | Accounting      |
| Grant     |               |                 |
|           | 190           | Contracting     |

21 rows selected.

# Descartes-szorzat, direkt-szorzat

- A direkt-szorzat a következőként kapható:
  - az összekapcsolási feltétel elhagyásával,
  - nem megengedett összekapcsolási feltétellel,
  - az első tábla összes sorának összekapcsolása a másik tábla összes sorával.
- A direkt szorzatok elkerülése érdekében, mindig kell legalább egy megengedett összekapcsolási feltétel legyen.

# Descartes-szorzat, direkt-szorzat

**EMPLOYEES (20 rows)**

| EMPLOYEE_ID | LAST_NAME | DEPARTMENT_ID |
|-------------|-----------|---------------|
| 100         | King      | 90            |
| 101         | Kochhar   | 90            |
| ...         |           |               |
| 202         | Fay       | 20            |
| 205         | Higgins   | 110           |
| 206         | Gietz     | 110           |

20 rows selected.

**DEPARTMENTS (8 rows)**

| DEPARTMENT_ID | DEPARTMENT_NAME | LOCATION_ID |
|---------------|-----------------|-------------|
| 10            | Administration  | 1700        |
| 20            | Marketing       | 1800        |
| 50            | Shipping        | 1500        |
| 60            | IT              | 1400        |
| 80            | Sales           | 2500        |
| 90            | Executive       | 1700        |
| 110           | Accounting      | 1700        |
| 190           | Contracting     | 1700        |

8 rows selected.

**Direkt-szorzat :**  
**20 x 8 = 160 sor**

| EMPLOYEE_ID | DEPARTMENT_ID | LOCATION_ID |
|-------------|---------------|-------------|
| 100         | 90            | 1700        |
| 101         | 90            | 1700        |
| 102         | 90            | 1700        |
| 103         | 60            | 1700        |
| 104         | 60            | 1700        |
| 107         | 60            | 1700        |

...

# Descartes-szorzat, direkt-szorzat

- A CROSS JOIN kulcsszó előállítja két tábla keresztszorzatát (vagyis a direkt szorzatát)

```
SELECT last_name, department_name  
FROM employees CROSS JOIN departments ;
```

| LAST_NAME | DEPARTMENT_NAME |
|-----------|-----------------|
| King      | Administration  |
| Kochhar   | Administration  |
| De Haan   | Administration  |
| Hunold    | Administration  |

160 rows selected.

# Összekapcsolások típusai

- Descartes-szorzat, direkt-szorzat (CROSS JOIN)
- Belső összekapcsolások (INNER JOIN...)
  - Természetes összekapcsolás (NATURAL JOIN)
- Külső összekapcsolások

# Relációs algebra és az SQL lekérdezések kapcsolata

**R**

| A  | B |
|----|---|
| a1 | 1 |
| a2 | 2 |
| a3 | 3 |
| a3 | 2 |

$\sigma_{A='a3' \wedge B>2}(R)$

| A  | B |
|----|---|
| a3 | 3 |

```
SELECT * FROM R
WHERE A='a3' AND B>2;
```

$\Pi_A(\sigma_{A='a1'}(R))$

| A  |
|----|
| a1 |

```
SELECT A FROM R
WHERE A='a1';
```

**RUS**

| A  | B |
|----|---|
| a1 | 1 |
| a2 | 2 |
| a3 | 3 |
| a3 | 2 |
| a2 | 3 |
| a3 | 1 |

**S**

| A  | B |
|----|---|
| a1 | 1 |
| a2 | 3 |
| a3 | 1 |

**R ∩ S**

| A  | B |
|----|---|
| a1 | 1 |

```
(SELECT * FROM R)
INTERSECT
(SELECT * FROM S);
```

**R - S**

| A  | B |
|----|---|
| a2 | 2 |
| a3 | 3 |
| a3 | 2 |

```
(SELECT * FROM R)
EXCEPT
(SELECT * FROM S);
```

```
(SELECT * FROM R)
UNION
(SELECT * FROM S);
```

# Relációs algebra és az SQL

## Descartes-szorzat

**R**

| A  | B |
|----|---|
| a1 | 1 |
| a2 | 2 |
| a3 | 3 |
| a3 | 2 |

`SELECT * FROM R, S;`

vagy

`SELECT * FROM R CROSS JOIN S;`

**S**

| A  | B |
|----|---|
| a1 | 1 |
| a2 | 3 |
| a3 | 1 |

**R × S**

| R.A | R.B | S.A | S.B |
|-----|-----|-----|-----|
| a1  | 1   | a1  | 1   |
| a1  | 1   | a2  | 3   |
| a1  | 1   | a3  | 1   |
| a2  | 2   | a1  | 1   |
| a2  | 2   | a2  | 3   |
| a2  | 2   | a3  | 1   |
| a3  | 3   | a1  | 1   |
| a3  | 3   | a2  | 3   |
| a3  | 3   | a3  | 1   |
| a3  | 2   | a1  | 1   |
| a3  | 2   | a2  | 3   |
| a3  | 2   | a3  | 1   |

# Relációs algebra és az SQL

## Belső összekapcsolások

R

| A  | B |
|----|---|
| a1 | 1 |
| a2 | 2 |
| a3 | 3 |
| a3 | 2 |

R|X|S

| A  | B |
|----|---|
| a1 | 1 |

```
SELECT * FROM R
NATURAL JOIN S;
```

R|X|<sub>R.B=S.B</sub>S

| R.A | R.B | S.A | S.B |
|-----|-----|-----|-----|
| a1  | 1   | a1  | 1   |
| a1  | 1   | a3  | 1   |
| a3  | 3   | a2  | 3   |

```
SELECT * FROM R JOIN S
ON R.B = S.B;
```

vagy

```
SELECT * FROM R INNER JOIN S
ON R.B = S.B;
```

vagy

```
SELECT * FROM R, S
WHERE R.B = S.B;
```

→  $\sigma_{R.B=S.B}(R \times S)$

S

| A  | B |
|----|---|
| a1 | 1 |
| a2 | 3 |
| a3 | 1 |

# Relációs algebra és az SQL

## Félig-összekapcsolás

R

| A  | B |
|----|---|
| a1 | 1 |
| a2 | 2 |

$R \bowtie S$

| R.A | B |
|-----|---|
| a1  | 1 |

```
SELECT R.* FROM R
WHERE EXISTS(
  SELECT * FROM S
  WHERE R.A = S.A);
```

$R \triangleright S$

| R.A | B |
|-----|---|
| a2  | 2 |

```
SELECT R.* FROM R
WHERE NOT EXISTS(
  SELECT * FROM S
  WHERE R.A = S.A);
```

S

| A  | C |
|----|---|
| a1 | 2 |
| a3 | 3 |

- Félig összekapcsolás (*semi join*, jelölés:  $\bowtie$ ):
  - $R(A_1, \dots, A_n), S(B_1, \dots, B_m)$  sémájú táblák esetén:  $R \bowtie S \equiv \Pi_{A_1, A_2, \dots, A_n}(R \mid X \mid S)$
- Anti összekapcsolás (*anti join*, jelölés:  $\triangleright$ ):
  - $R \triangleright S \equiv R - R \bowtie S$

# Összefoglalás: Alkérdeések és a félig-összekapcsolás

- Az R és S relációk **félig-összekapcsolása**  $R \bowtie S$  (semijoin) az R azon sorainak halmaza, amely sorok megegyeznek az S legalább egy sorával az R és S összes közös attribútumán.
- **H.F:** Adjunk meg olyan három különböző relációs algebrai kifejezést, amelyek ekvivalensek az  $R \bowtie S$  kifejezéssel!

# Félig-összekapcsolás (semijoin)

- Legyen  $R(X, Y)$  és  $S(Y, Z)$ , ahol  $X, Y, Z$  attr.halmazok  
Az  $R$  és  $S$  relációk **félig-összekapcsolása**  $R \bowtie S$  (szemijoin)

(1a) `SELECT X, Y FROM R NATURAL JOIN S;`

(1b) `SELECT X, Y FROM R JOIN S USING (Y);`

(2a) `SELECT X, R.Y FROM R, S WHERE R.Y=S.Y;`

(2b) `SELECT X, R.Y FROM R JOIN S ON R.Y=S.Y;`

(3a) `SELECT X,Y FROM R --- spec. ha Y egy attribútum:  
WHERE Y = ANY (SELECT Y FROM S);`

(3b) `SELECT X,Y FROM R --- általában is Y attr.halmaz:  
WHERE Y IN (SELECT Y FROM S);`

(4) `SELECT X,Y FROM R  
WHERE EXISTS (SELECT * FROM S WHERE Y=R.Y);`

# Kérdés/Válasz

- Köszönöm a figyelmet! Kérdés/Válasz?

## Összefoglalás:

- Alkérdeések használata az SQL SELECT utasítás FROM és WHERE záradékaiban
- Összekapcsolások az SQL lekérdezésekben, lásd még az Oracle gyakorlatok példáit és feladatait is!