

# **Többléptáblás lekérdezések megjelenítése**

# Célkitűzés

- **Egynél több táblának egyenlőségen vagy nem-egyenlőségen alapuló összekapcsolást végző SELECT utasítások írása.**
- **Egy táblának önmagával történő összekapcsolása.**
- **Olyan adatok megjelenítése külső összekapcsolás segítségével, amelyek nem felelnek meg egy összekapcsolási feltételnek.**
- **Legalább két táblának a felhasználásával a sorok direktszorzatának előállítás.**

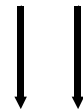
# Többléptáblás lekérdezés

## EMPLOYEES

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

## DEPARTMENTS

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



EMPLOYEE_ID	DEPARTMENT_ID	DEPARTMENT_NAME
200	10	Administration
201	20	Marketing
202	20	Marketing
...		
102	90	Executive
205	110	Accounting
206	110	Accounting

# Az összekapcsolások fajtái (SQL:1999)

- **Direktszorzat (kereszt-összekapcsolás).**
- **Természetes összekapcsolás.**
- **USING utasítással történő összekapcsolás**
- **Teljes ( vagy két oldali) külső összekapcsolás**
- **Tetszőleges feltételen alapuló külső összekapcsolás**

# Táblák összekapcsolása (SQL:1999 szintaxis)

- Több tábla lekérdezése esetén a szintaxis a következő:

```
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];
```

- **NATURAL JOIN** – azonos nevű oszloppal rendelkező táblák természetes összekapcsolása
- **JOIN tábla USING oszlopnév** – az oszlopnév értékeinek egyenlőségi vizsgálatán alapuló összekapcsolás
- **JOIN tábla ON tábla1.oszlopnév** – az ON után szereplő egyenlőségi feltétel vizsgálatán alapuló összekapcsolás
- **LEFT/RIGHT/FULL OUTER** – a külső összekapcsolás paraméterei
- **CROSS JOIN** – két tábla direkt szorzata

# Természetes összekapcsolás megfogalmazása

- **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

8 rows selected.

- **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 írása USING kulcsszóval

- Ha több oszlopnak azonos ugyan 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.**



# Oszlopnevek összekapcsolása

**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 a Departments tábla és az Employees tábla DEPARTMENT\_ID oszlopaikban szereplő értékeinek összehasonlítása kell. Így ez egy egyenlőségen alapuló összekapcsolás lesz.
- Az ilyen típusú összekapcsolásban általában az elsődleges- és az idegen kulcs komponensei szerepelnek.

# 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.**

## Alias nevek használata tábláknál

- A lekérdezések átláthatósága miatt használhatunk tábla alias neveket.
- A táblák alias neveinek használata javítja a lekérdezés teljesítményét.
- Az alias nevek maximum 30 karakter hosszúak lehetnek (minél rövidebb annál jobb)
- Az alias nevek 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**

# Önösszekapcsolás ON kulcsszóval 1.

**EMPLOYEES (WORKER)**

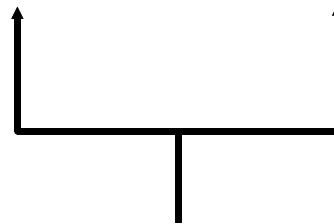
EMPLOYEE_ID	LAST_NAME	MANAGER_ID
100	King	
101	Kochhar	100
102	De Haan	100
103	Hunold	102
104	Ernst	103
107	Lorentz	103
124	Mourgos	100

...

**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

## Önösszekapcsolás 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.



## További feltételek megadása egy összekapcsoláshoz

```
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)  
AND   e.manager_id = 149 ;
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	LOCATION_ID
174	Abel	80	80	2500
176	Taylor	80	80	2500

- Ugyanezt érhetjük el a WHERE feltétellel is, azaz:

```
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)  
WHERE  e.manager_id = 149;
```

## 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, azaz : (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 azon 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 esete, 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.

# A 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.**

# A 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 sorok**

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

...

# A kereszt-összekapcsolás

- **A CROSS JOIN kulcsszó előállítja két tábla kereszt-szorzatát**
- **Ezt hívják direkt szorzatnak is.**

```
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.

# Összefoglalás

- **Ebben a részben megismertük:**
  - a többtáblás lekérdezéseket
  - **általános összekapcsolásokat**
  - **a természetes összekapcsolásokat**
  - **külső összekapcsolásokat**
  - **az összekapcsolások használatát**

## Feladatok

1. Írjunk olyan lekérdezést, amely előállítja az összes munkaügyi osztályt a címével együtt. Használjuk fel ehhez a LOCATIONS és a COUNTRIES táblákat. Az eredményben jelenjen meg az irányító szám, az utca név, a város neve, az állam vagy tartomány neve és az ország neve is. Használjuk NATURAL JOIN kulcsszót a lekérdező utasításunkban.
2. Kell egy jelentés az összes dolgozóról. Írjunk lekérdezést amely kiírja a dolgozók család nevét, (munka) osztályának számát és az osztály nevét.
3. Kell egy jelentés a Toronto-ban dolgozó alkalmazottokról. Irassuk ki a családi nevét, munkakörének-és osztályának a számát és az osztály nevét azon alkalmazottaknak, akik Toronto-ban dolgoznak.
4. Készítsünk egy jelentést, amely kiírja az alkalmazottak családi nevét és alkalmazotti azonosítóját a felettesének családi nevével és alkalmazotti azonosítójával egyetemben. A kiíratásban szereplő oszlopok neve legyenek: Employee, EMP#, Manager, Mgr#
5. Módosítsuk az előző lekérdezést úgy, hogy kiírja az össze dolgozót, még King-et is, akinek nincsen főnöke. Rendezzük a kapott eredményt az alkalmazott azonosítók alapján.

# Feladatok

6. **Készítsünk olyan lekérdezést, amely kiírja az olyan alkalmazottak családi nevét, osztály azonosítóját, akik egy osztályon dolgoznak egy adott dolgozóval.**
7. **Az beosztásokról és a fizetésekről szóló jelentés kell. Adott a JOB\_GRADES tábla, melynek a struktúrája az alábbi... Írjunk olyan lekérdezést, amely kiírja az összes dolgozó nevét, munkáját, osztály nevét, fizetését és beosztását.**
8. **Határozzuk meg azon alkalmazottaknak a nevét, akiket Davies után vettek fel. A lekérdezés eredményében jelenjen meg a dolgozó neve illetve a felvételének időpontja.**
9. **Azon dolgozók nevét és felvételüknek idejét akarjuk meghatározni, akik a főnökök előtt lettek felvéve. Emellett írjuk ki minden dolgozó mellé a főnökének nevét és felvételének idejét.**