



Adatbázisok

8. labor: Allekérdezések

IN, ANY/SOME, ALL

- IN
Igazat ad vissza, ha a bal oldalán levő érték benne van a jobb oldalán levő halmazban
- ANY (vagy SOME)
Igazat ad vissza, ha a jobb oldalán levő halmaz **legalább egyik** elemére igaz a reláció
- ALL
Igazat ad vissza, ha a jobb oldalán levő halmaz **minden** elemére igaz a reláció

IN, ANY/SOME, ALL

Mindháromnak közös tulajdonsága:

- egy értéket (bal oldal)
hasonlíttanak egy értékhalmozhoz
(jobb oldal)
- a kapcsolódó allekérdezés csak
egyetlen oszlopot adhat vissza

IN példa

- Listázzuk azon részlegek nevét és városát, ahol Programmer-ek dolgoznak.

```
SELECT      department_name Részlegnév,  
            city Város  
FROM departments NATURAL JOIN  
     locations  
WHERE department_id IN  
      (SELECT department_id  
       FROM employees NATURAL JOIN jobs  
       WHERE job_title = 'Programmer')  
ORDER BY Részlegnév;
```

Gyakorlás

- Listázzuk azon dolgozók vezetéknévét és munkakörét, akiknek a beosztottai között van Programmer munkakörű személy.
 - Az előző feladathoz nagyon hasonló!
 - Most nem a Programmer-ek department_id-ja, hanem a manager_id-ja érdekel minket...

Antijoin

```
SELECT * FROM employees
WHERE department_id NOT IN
(SELECT department_id
FROM departments
WHERE location_id = 1700)
ORDER BY last_name;
```

- Visszaadja azokat, amelyeknek **nincs** megfelelője az allekérdezésben.

ANY példa

- Listázzuk azon dolgozókat, akiknek a fizetése megegyezik a 30-as részleg valamely dolgozójának fizetésével

```
SELECT * FROM employees  
WHERE salary = ANY  
(SELECT salary FROM employees  
WHERE department_id = 30)
```

ANY példa

- Vegyük észre, hogy ugyanez megoldható IN segítségével!

```
SELECT * FROM employees  
WHERE salary IN  
(SELECT salary FROM employees  
WHERE department_id = 30)
```


ANY példa

- Listázzuk azon dolgozókat, akiknek a fizetése nagyobb a 30-as részleg valamely dolgozójának fizetésénél

```
SELECT * FROM employees  
WHERE salary > ANY  
(SELECT salary FROM employees  
WHERE department_id = 30)
```

- Ez már nem helyettesíthető IN-nel!

Feladat

- Listázzuk azon dolgozókat, akikre igaz, hogy van olyan részlegfőnök, akinél többet keresnek.

Feladat

- Listázzuk azon dolgozókat, akikre igaz, hogy van olyan részlegfőnök, akinél többet keresnek.

```
SELECT salary FROM departments d inner join  
employees e on  
d.manager_id = e.employee_id;
```

```
SELECT * FROM employees WHERE salary >  
ANY
```

```
(SELECT salary FROM departments d inner join  
employees e on  
d.manager_id = e.employee_id)
```

ALL példa

- Listázzuk azon dolgozókat, akiknek a fizetése nagyobb a 30-as részleg minden dolgozójának fizetésénél

```
SELECT * FROM employees  
WHERE salary > ALL  
(SELECT salary FROM employees  
WHERE department_id = 30)
```

ALL példa

- Listázzuk azon dolgozókat, akiknek a fizetése minden részleg átlagfizetésénél nagyobb.

```
SELECT * FROM employees  
WHERE salary > ALL  
(SELECT AVG(salary) FROM employees  
GROUP BY department_id)
```

- Vigyázat, az allekérdezés itt is csak egy oszlopot adhat vissza, tehát a department_id-t nem listázhatjuk!

Feladat

- Listázzuk azokat a dolgozókat, akik kevesebbet keresnek, mint bármelyik Manager munkakörű dolgozó.
 - Azaz kevesebbet keresnek minden olyan kollégájuknál, akiknek a munkakör megnevezésében szerepel a Manager szó.

EXISTS (Semijoin)

- Listázzuk azokat a részlegeket, ahol dolgozik 15.000 dollárnál magasabb fizetésű dolgozó.

```
SELECT * FROM departments d
WHERE EXISTS
  (SELECT * FROM employees e
   WHERE
    d.department_id=e.department_id
   AND e.salary > 15000)
ORDER BY department_name;
```

EXISTS (Semijoin)

- Visszaadja azokat, amelyeknek van megfelelője az allekérdezésben, de nem duplikál, mint a sima join:

```
SELECT d.*  
FROM departments d, employees e  
WHERE d.department_id = e.department_id  
AND  
e.salary > 15000 ORDER BY  
department_name;
```

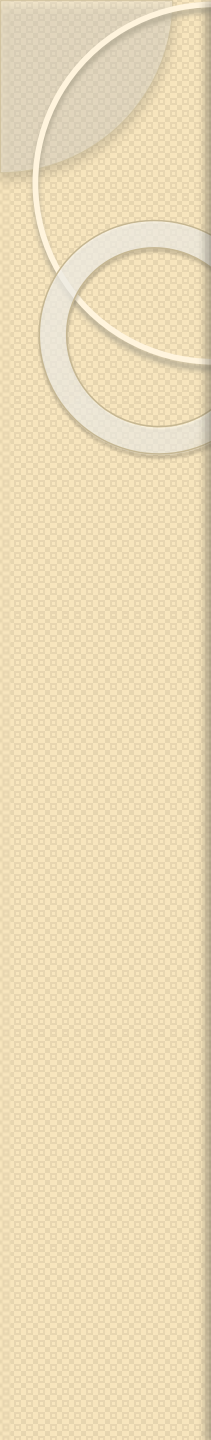

EXISTS példa

- Mely országokban vannak részlegek?

```
SELECT country_name FROM  
countries c  
WHERE EXISTS  
(SELECT * FROM  
departments NATURAL JOIN  
locations  
WHERE country_id = c.country_id);
```

Feladat

- Mely részlegekben dolgoznak olyanok, akiknek a munkakör megnevezésében szerepel a Manager szó?



```
SELECT department_name FROM  
departments d WHERE EXISTS  
(SELECT * FROM employees natural join  
jobs WHERE d.department_id =  
department_id and job_title like '%Manager  
%') )
```

```
SELECT department_name FROM  
departments WHERE  
department_id IN (SELECT department_id  
FROM employees natural join jobs WHERE  
job_title like '%Manager%') )
```

Halmazműveletek

- UNION

Unió: mindazon elemek, amelyek legalább az egyik halmazban megtalálhatók (ismétlődés nélkül)
UNION ALL: ismétlődésekkel

- INTERSECT

Metszet: mindazon elemek, amelyek mindkét halmazban megtalálhatók

- MINUS

Különbség: mindazon elemek, amelyek megtalálhatók az első halmazban, de a másodikban nem.

Halmazműveletek

- A halmazművelet mindkét oldalán egy-egy lekérdezés áll.
- A lekérdezések által visszaadott oszlopoknak egyezniük kell **darabszámban és típusban** (névben nem).

Union példa

- Mely régió azonosítók találhatók meg a regions vagy a countries táblában?

```
SELECT region_id FROM regions  
UNION  
SELECT region_id FROM countries;
```

```
SELECT region_id FROM regions  
UNION ALL  
SELECT region_id FROM countries;
```

Intersect példa

- Kik azok, akik már töltöttek be másik munkakört (azaz szerepelnek a job_history táblában)?

```
SELECT employee_id FROM employees  
INTERSECT  
SELECT employee_id FROM  
job_history;
```

Intersect példa

- Ha a nevüket szeretnénk látni:

```
SELECT first_name||' '||last_name  
FROM employees WHERE employee_id  
IN (SELECT employee_id FROM  
employees  
INTERSECT  
SELECT employee_id FROM  
job_history);
```


Minus példa

- Kik azok, akik részlegfőnökök, de nincs nekik közvetlen beosztottjuk?

```
SELECT manager_id FROM  
departments
```

```
MINUS
```

```
SELECT manager_id FROM employees;
```

Egy összetettebb példa

- Mi azoknak a dolgozóknak a neve és részlegük neve, akik részlegfőnökök, VAGY van beosztottjuk?
 - Vigyázat, a
SELECT manager_id FROM employees
itt már nem elég, mivel a részlegüket is tudnunk kell!

Megoldás 1

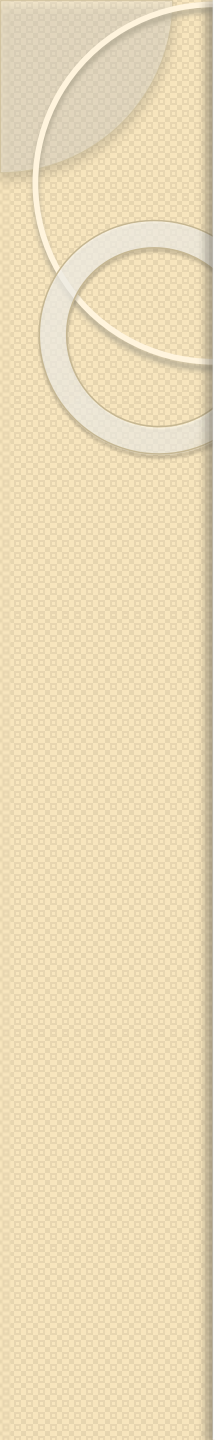
```
SELECT first_name||' '||last_name Name,  
department_name Department  
FROM employees e INNER JOIN departments d  
ON e.department_id=d.department_id  
INNER JOIN  
(SELECT manager_id, department_id FROM  
departments WHERE manager_id IS NOT NULL  
UNION  
SELECT employee_id, department_id FROM  
employees WHERE employee_id IN  
(SELECT manager_id FROM employees)) f  
ON f.manager_id=e.employee_id;
```

Megoldás 2

```
select last_name, department_name from  
employees inner join departments using  
(department_id)  
where employee_id IN  
(SELECT manager_id FROM departments  
UNION  
SELECT manager_id FROM employees);
```

Feladat

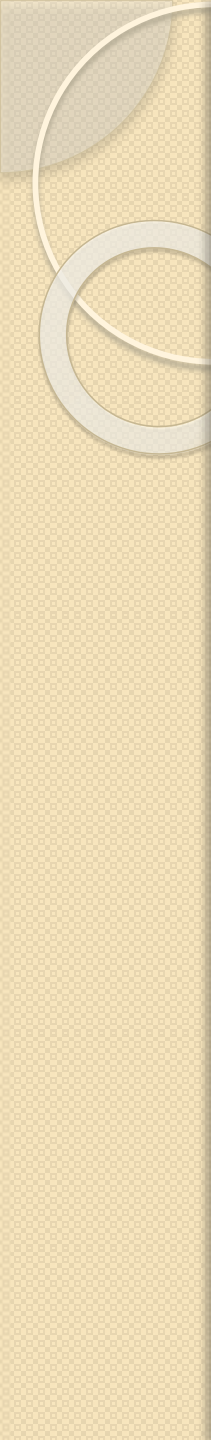
- Van-e olyan dolgozó a job_historyban, aki már nem dolgozik a cégnél?



```
select employee_id from  
job_history  
minus  
select employee_id from  
employees;
```

Feladat

- Hány db olyan ország van a Countries táblában, amelyben nincs egyetlen részleg sem?
 - Oldjuk meg ugyanezt antijoinnal is!



```
select count(*) from  
(select country_id from countries  
minus
```

```
select country_id from locations  
natural join departments)
```

```
select count(*) from  
(select country_id from countries  
where country_id not in  
(select country_id from locations  
natural join departments))
```


Felső-N analízis

- Más népszerű SQL rendszerekben az output rekordok darabszámának korlátozása megoldható a LIMIT segítségével.
 - Pl. MySQL-ben:
`SELECT * FROM tabla LIMIT 5;`
- Az Oracle-ben nincs LIMIT, helyette két másik lehetőség van.

ROWNUM

- A lekérdezésekben elérhető pseudo-oszlop, amely a rekord "sorszáma" a lekérdezésben (NEM a táblában!)
- A WHERE feltétel kiértékelésekor kerül kiosztásra, **csoportosítás és rendezés előtt**.
- MIUTÁN kiosztásra került egy ROWNUM, azaz találtunk egy rekordot, amire igaz a WHERE, utána fog inkrementálódni, hogy a következő eggyel nagyobb sorszámot kapjon.

ROWNUM

- Tehát ez **rossz ötlet**:

```
SELECT * FROM employees  
WHERE ROWNUM <= 5  
ORDER BY salary DESC;
```

- Ez pedig nem ad vissza semmit:

```
SELECT * FROM employees  
WHERE ROWNUM > 1;
```

ROWNUM

- Helyes használat:

```
SELECT * FROM  
(SELECT * FROM employees  
ORDER BY salary DESC)  
WHERE ROWNUM <= 5;
```

ROWNUM

- Figyeljünk rá, hova írjuk!

```
SELECT * FROM  
(SELECT e.*, ROWNUM FROM employees  
e  
ORDER BY salary);
```

```
SELECT al.*, ROWNUM FROM  
(SELECT * FROM employees  
ORDER BY salary) al;
```

Feladat

- Listázzuk ki a munkakörök nevét és a hozzájuk tartozó maximális fizetéseket, utóbbi szerinti csökkenő sorrendben, az **első 3 kivételével**.
 - Beugratós feladat, jól gondoljuk át!



```
SELECT job_title, max_salary FROM jobs  
ORDER BY max_salary;
```

```
SELECT al.*, ROWNUM Sorszám FROM  
(SELECT job_title, max_salary FROM jobs  
ORDER BY max_salary)
```

```
SELECT job_title, max_salary FROM  
(SELECT al.*, ROWNUM Sorszám FROM  
(SELECT job_title, max_salary FROM jobs  
ORDER BY max_salary) al  
)  
WHERE Sorszám>3;
```

FETCH, OFFSET

- Az Oracle 12c verziójától kezdve van egyszerűbb megoldás, mint a ROWNUM.

... FETCH FIRST n ROWS ONLY

vagy

... OFFSET n ROWS

[FETCH NEXT m ROWS ONLY]

Példák

```
SELECT * FROM employees  
ORDER BY salary DESC  
FETCH FIRST 5 ROWS ONLY;
```

```
SELECT job_title, max_salary FROM  
jobs  
ORDER BY max_salary  
OFFSET 3 ROWS;
```