

Adatok szűrése és rendezése

# Célkitűzések

- Szűrést kifejező lekérdezések végrehajtása
- A lekérdezés eredményének rendezése

# A lista korlátozása kiválasztással


## EMPLOYEES

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90
103	Hunold	IT_PROG	60
104	Ernst	IT_PROG	60
107	Lorentz	IT_PROG	60
124	Mourgos	ST_MAN	50

...

20 rows selected.

**Csak a 90-es  
osztály dolgozóit  
listázzuk ki!**



EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

# Az eredmény korlátozása WHERE feltétellel

- A WHERE feltétel használata:

```
SELECT * | { [DISTINCT] column | expression [alias], ... }  
FROM table  
[WHERE condition(s)];
```

- A WHERE feltétel a FROM után következik.
- A feltétel oszlopértékeket, konstansokat, aritmetikai kifejezéseket, függvényértékeket hasonlíthat össze.
- Három részből áll:
  - Oszlopnév
  - Összehasonlítási feltétel
  - Oszlopnév, konstans vagy értékhalmoz

# A WHERE feltétel használata

```
SELECT employee_id, last_name, job_id, department_id  
FROM employees  
WHERE department_id = 90 ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

# Karakterláncok és dátumok

- A karakterláncokat és dátumokat egyszeres idézőjelek közé tesszük.
- A karakteres értékek kisbetű-nagybetű érzékenyek.
- A dátumértékek formátum érzékenyek.
- Az alap dátumformátum YYYY-MM-DD

```
SELECT last_name, job_id, department_id  
FROM employees  
WHERE last_name = 'Whalen' ;
```

# Összehasonlító feltételek

... WHERE kifejezés összehasonlítás érték

Összehasonlítási művelet	Jelentés
=	Egyenlő
>	Nagyobb mint
>=	Nagyobb vagy egyenlő mint
<	Kisebb mint
<=	Kisebb egyenlő mint
<> vagy != vagy ^=	Nem egyenlő
BETWEEN ...AND...	A két érték közé esik (zárt intervallum)
IN (halmaz)	Megegyezik a halmaz valamelyik elemével
LIKE	Illeszkedik egy karakteres mintára
IS NULL	Az értéke nullérték

# Összehasonlító feltétel használata

```
SELECT last_name, salary
FROM employees
WHERE salary <= 3000 ;
```

LAST_NAME	SALARY
Matos	2600
Vargas	2500



# A BETWEEN feltétel használata

- A BETWEEN feltétellel megadhatók azok a dolgozók, akiknek a fizetése 2500 és 3500 dollár közé esik:

```
SELECT last_name, salary
FROM employees
WHERE salary BETWEEN 2500 AND 3500 ;
```

↑  
Alsó korlát

↑  
Felső korlát

LAST_NAME	SALARY
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500

# Az IN feltétel használata

- Az IN tartalmazási feltétellel megadhatók azok a dolgozók, akiknek a főnöke 100, 101 vagy 201 azonosítóval rendelkezik:

```
SELECT employee_id, last_name, salary, manager_id
FROM employees
WHERE manager_id IN (100, 101, 201) ;
```

EMPLOYEE_ID	LAST_NAME	SALARY	MANAGER_ID
202	Fay	6000	201
200	Whalen	4400	101
205	Higgins	12000	101
101	Kochhar	17000	100
102	De Haan	17000	100
124	Mourgos	5800	100
149	Zlotkey	10500	100
201	Hartstein	13000	100

8 rows selected.

# A LIKE feltétel használata

- A LIKE feltételben adott mintában a karaktereken kívül dzsókereket is lehet használni:
  - % nulla vagy több karaktert jelöl,
  - \_ pontosan egy karaktert jelöl.

Milyen keresztnévű dolgozóknak kezdődik a keresztneve S-sel (nagy S-sel)?

```
SELECT first_name
FROM employees
WHERE first_name LIKE 'S%';
```

## A LIKE feltétel használata

- Több dzsókert is lehet a mintában használni.
- Milyen vezetéknevű dolgozók vezetéknevének 2. betűje „o”?

```
SELECT last_name  
FROM employees  
WHERE last_name LIKE '_o%';
```

LAST_NAME
Kochhar
Lorentz
Mourgos

- A % vagy \_ dzsókerek ESCAPE karakterrel megjelölve közös karakterekké válnak.
- Kiknek az azonosítója kezdődik az SA\_ három karakterrel?

```
SELECT employee_id, last_name, job_id  
FROM employees WHERE job_id LIKE 'SA\_%' ESCAPE '\\';
```

# A nullértékek ellenőrzése

- A nullértékeket **IS NULL** vagy **IS NOT NULL** feltétellel tesztelhetjük.

```
SELECT last_name, manager_id
FROM employees
WHERE manager_id IS NULL ;
```

LAST_NAME	MANAGER_ID
King	

**A nullérték ismeretlen, meghatározatlan értéket jelöl, ezért nem egyenlő semmilyen értékkel, így a nullértéket nem lehet = vagy <> segítségével tesztelni!**

# Logikai feltételek

Művelet	Jelentés
AND	Igaz (TRUE), ha a feltétel mindkét tagja igaz
OR	Igaz (TRUE), ha a feltételnek legalább az egyik tagja igaz
NOT	Igaz (TRUE), ha a feltétel hamis

# Az AND művelet használata

Mely dolgozók beosztásában szerepel a **MAN** karakterlánc és legalább 10000 a fizetésük?

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary >=10000
AND job_id LIKE '%MAN%' ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
149	Zlotkey	SA_MAN	10500
201	Hartstein	MK_MAN	13000

Az **AND** igazságtáblája ismeretlen értékkel kiegészítve:

<b>AND</b>	<b>TRUE</b>	<b>FALSE</b>	<b>UNKNOWN (NULL)</b>
<b>TRUE</b>	TRUE	FALSE	UNKNOWN (NULL)
<b>FALSE</b>	FALSE	FALSE	FALSE
<b>UNKNOWN (NULL)</b>	UNKNOWN (NULL)	FALSE	UNKNOWN (NULL)

# Az OR művelet

Mely dolgozók beosztásában szerepel a **MAN** karakterlánc vagy legalább 10000 a fizetésük?

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary >= 10000
OR job_id LIKE '%MAN%' ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
100	King	AD_PRES	24000
101	Kochhar	AD_VP	17000
102	De Haan	AD_VP	17000
124	Mourgos	ST_MAN	5800
149	Zlotkey	SA_MAN	10500
174	Abel	SA_REP	11000
201	Hartstein	MK_MAN	13000
205	Higgins	AC_MGR	12000

8 rows selected.

**Az OR igazságtáblája ismeretlen értékkel kiegészítve:**

OR	TRUE	FALSE	UNKNOWN (NULL)
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN (NULL)
UNKNOWN (NULL)	TRUE	UNKNOWN (NULL)	UNKNOWN (NULL)



# A NOT művelet használata

Tetszőleges feltétel tagadható: **NOT IN**, **NOT LIKE**, **NOT BETWEEN**, **IS NOT NULL**.

Mely dolgozók beosztása se nem **IT\_PROG**, se nem **ST\_CLERK**, se nem **SA\_REP**?

```
SELECT last_name, job_id
FROM employees
WHERE job_id NOT IN ('IT_PROG', 'ST_CLERK', 'SA_REP');
```

LAST_NAME	JOB_ID
King	AD_PRES
Kochhar	AD_VP
De Haan	AD_VP
Mourgos	ST_MAN
Zlotkey	SA_MAN
Whalen	AD_ASST
Hartstein	MK_MAN
Fay	MK_REP
Higgins	AC_MGR
Gietz	AC_ACCOUNT

10 rows selected.

A **NOT** igazságtáblája ismeretlen értékkel kiegészítve:

<b>NOT</b>	<b>TRUE</b>	<b>FALSE</b>	<b>UNKNOWN (NULL)</b>
	FALSE	TRUE	UNKNOWN (NULL)

# A műveletek sorrendjének alapértelmezése

A kiértékelés sorrendje	A kifejezésben szereplő műveletek
1.	Aritmetikai műveletek
2.	Karakterláncok összefűzése (konkatenáció)
3.	Összehasonlító feltételek
4.	IS [NOT] NULL, LIKE, [NOT] IN
5.	[NOT] BETWEEN
6.	Nem egyenlő
7.	NOT logikai feltétel
8.	AND logikai feltétel
9.	OR logikai feltétel

**A sorrendet zárójelezéssel felülírhatjuk!**

# ① Precedencia szabályok alkalmazása

Kik azok, akik **vagy** 15000-nél többet kereső elnökök (AD\_PRES), **vagy** mindegy mennyit keresnek, de a beosztásuk képviselő (SA\_REP)?

```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id = 'SA_REP'
OR job_id = 'AD_PRES'
AND salary > 15000;
```

①

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000
Abel	SA_REP	11000
Taylor	SA_REP	8600
Grant	SA_REP	7000

## ② Precedencia szabályok alkalmazása

Kik azok, akik 15000-nél többet keresnek és vagy elnökök (AD\_PRES) vagy képviselők (SA\_REP)?

```
SELECT last_name, job_id, salary
FROM employees
WHERE (job_id = 'SA_REP'
OR job_id = 'AD_PRES')
AND salary > 15000;
```

②

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000

# Az ORDER BY használata

**Szintaxis:** `SELECT kifejezés`  
`FROM tábla`  
`[WHERE feltétel(ek)]`  
`[ORDER BY {oszlop, kif, oszlopsorszám} [ASC|DESC]];`

- Az eredmény sorai az ORDER BY szerint rendezettek:
  - **ASC:** növekvő sorrend, (alapértelmezés)
  - **DESC:** csökkenő sorrend
- Az ORDER BY rész a SELECT utasítás végén szerepel.
- Az ORDER BY nélkül a sorrend nem determinisztikus.

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date ;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
King	AD_PRES	90	17-JUN-87
Whalen	AD_ASST	10	17-SEP-87
Kochhar	AD_VP	90	21-SEP-89
Hunold	IT_PROG	60	03-JAN-90
Ernst	IT_PROG	60	21-MAY-91

...

20 rows selected.

**A belépési dátum szerint rendezett lista.**

# A rendezés használata

A nullértékek növekvő rendezés esetén a lista végén szerepelnek, csökkenő esetben az elején!

- Csökkenő sorrend a belépés dátuma szerint:

```
SELECT last_name, job_id, department_id, hire_date
FROM employees
ORDER BY hire_date DESC ;
```

1

- Számított oszlop másodneve szerinti rendezés:

```
SELECT employee_id, last_name, salary*12 annsal
FROM employees
ORDER BY annsal ;
```

2

- Az osztály szerint növekvő és azon belül fizetés szerint csökkenő rendezés:

```
SELECT last_name, department_id, salary
FROM employees
ORDER BY department_id, salary DESC ;
```

3