

# Adatbázis kezelés I.

## SQL bevezető

Rostagni Csaba

2024. szeptember 16.

# Ezen az órán... I

- 1 Műveleti jelek (emlékeztető)
- 2 Feltételes lekérdezések (WHERE)
- 3 A NULL érték

# Tartalom I

- 1 Műveleti jelek (emlékeztető)
  - Összehasonlító operátorok
  - Logikai operátorok

# Tartalom

- 1 Műveleti jelek (emlékeztető)
  - Összehasonlító operátorok
  - Logikai operátorok

# Összehasonlító operátorok

SQL	Mat.	Megnevezés
<	<	kisebb
>	>	nagyobb
<=	≤	kisebb egyenlő
>=	≥	nagyobb egyenlő
=	=	egyenlő
<>	≠	nem egyenlő

táblázat: összehasonlító operátorok

Megjegyzés:

- A != is a nem egyenlőt jelenti (MySQL, MS SQL, Oracle, ...)
- A ^= is a nem egyenlőt jelenti (Oracle)
- A <> formátum felel meg az ANSI szabványnak, ez az elvárt!

# Tartalom

- 1 Műveleti jelek (emlékeztető)
  - Összehasonlító operátorok
  - Logikai operátorok

# Logikai operátorok

SQL	Mat.	Megnevezés
NOT	$\neg$	nem
AND	$\wedge$	és
OR	$\vee$	vagy

táblázat: logikai operátorok

Az operátorok precedencia szerint csökkenő sorrendben vannak feltüntetve.

Linkek:

- MySQL dokumentáció: Operátorok precedenciája

# Tartalom I

## 2 Feltételes lekérdezések (WHERE)

- ÉS/VAGY kombinálása
- Azonos mezőre több lehetséges érték (IN)
- Azonos mezőre több lehetséges érték tagadással (NOT IN)
- Két érték között (BETWEEN)



# Tartalom

## 2 Feltételes lekérdezések (WHERE)

- ÉS/VAGY kombinálása
- Azonos mezőre több lehetséges érték (IN)
- Azonos mezőre több lehetséges érték tagadással (NOT IN)
- Két érték között (BETWEEN)

# Logikai operátorok kombinálása (hibás próbálkozás)

Listázzuk ki azoknak a **benzines** autóknak minden adatát, melyek **Ford** vagy **Honda** gyártmányúak.

```
SELECT * FROM `autok`  
WHERE `uzemanyag` = 'benzin'  
    AND `gyarto` = 'Ford'  
    OR `gyarto` = 'Honda';
```

logikai hiba

rendszám	gyarto	tipus	kategoria	uzemanyag	szin
AAA-555	Honda	Jazz	M1	hibrid	kék
AA-AX-1234	Ford	Fiesta	M1	benzin	sárga

- **Probléma:** a Honda Jazz is megjelent, ami nem benzines!

# Logikai operátorok kombinálása hiba magyarázata

Az ÉS logikai operátor precedenciája magasabb.  
Elsőnek ez a feltétel teljesül.

SELECT

\*

FROM

`autok`

WHERE

`uzemanyag`

=

'benzin'

AND

`gyarto`

=

'Ford'

OR

`gyarto`

=

'Honda'

;

A Honda esetén az üzemanyagra nem szűr.

# Logikai operátorok kombinálása

Listázzuk ki azoknak a **benzines** autóknak minden adatát, melyek **Ford** vagy **Honda** gyártmányúak.

MySQL

```
SELECT *  
FROM `autok`  
WHERE `uzemanyag` = 'benzin' AND `gyarto` = 'Ford'  
      OR `uzemanyag` = 'benzin' AND `gyarto` = 'Honda';
```

rendszám	gyarto	tipus	kategoria	uzemanyag
AA-AX-1234	Ford	Fiesta	M1	benzin

- Az üzemanyag a feltételben kétszer is szerepel!
- Bár működik, módosításkor oda kell figyelni, hogy ne okozzon hibát!

# Logikai operátorok kombinálása

Listázzuk ki azoknak a **benzines** autóknak minden adatát, melyek **Ford** vagy **Honda** gyártmányúak.

MySQL

```
SELECT *  
FROM `autok`  
WHERE (`uzemanyag` = 'benzin')  
      AND (`gyarto` = 'Ford' OR `gyarto` = 'Honda');
```

rendszám	gyarto	tipus	kategoria	uzemanyag
AA-AX-1234	Ford	Fiesta	M1	benzin

- Megfelelően zárójelezve rövid, tömör kódot kapunk
- Későbbi módosításkor is csak egy helyen kell módosítani

# Logikai operátorok kombinálása

Listázzuk ki a **Honda** és a **Ford** autói közül azokat, melyek üzemanyaga **diesel** vagy **benzin** .

MySQL

```
SELECT * FROM `autok`  
WHERE (`gyarto`='Honda' OR `gyarto`='Ford')  
      AND (`uzemanyag`='benzin' OR `uzemanyag`='diesel');
```

rendszam	gyarto	tipus	kategoria	uzemanyag	szin
ABC-123	Ford	Focus	M1	diesel	kék
AA-AX-1234	Ford	Fiesta	M1	benzin	sárga

- A Honda üzemanyaga hibrid, így az helyesen nem jelenik meg
- A fenti zárójelek nem elhagyhatóak
- Zárójel nélkül a lekérdezés bonyolult lenne

# Tartalom

## 2 Feltételes lekérdezések (WHERE)

- ÉS/VAGY kombinálása
- Azonos mezőre több lehetséges érték (IN)
- Azonos mezőre több lehetséges érték tagadással (NOT IN)
- Két érték között (BETWEEN)

# Választás több elem közül

Listázzuk ki a **benzines**, **diesel** és **elektromos** autók minden adatát

```
SELECT *  
FROM `autok`  
WHERE `uzemanyag` = 'benzin'  
           OR 'diesel'  
           OR 'elektromos';
```

logikai hiba

- Az ``uzemanyag` = 'benzin'` logikai értéke lehet igaz vagy hamis.
- A `'diesel'` és az `'elektromos'` logikai értéke MINDIG HAMIS!
- Mivel a **(bármilyen) vagy hamis** értéke **(bármilyen)**, így a benzines autók összes adatát kapjuk meg.!



# Választás több elem közül

Listázzuk ki a **benzines**, **diesel** és **elektromos** autók minden adatát

MySQL

```
SELECT *  
FROM `autok`  
WHERE `uzemanyag` = 'benzin'  
      OR `uzemanyag` = 'diesel'  
      OR `uzemanyag` = 'elektromos';
```

- Az `uzemanyag` mezőt fölöslegesen sokszor kellett felsorolni
- ÉS/VAGY együttes alkalmazása okozhat problémát, ha nincs jól zárójelezve

# Az IN operátor

Listázzuk ki a **benzines**, **diesel** és **elektromos** autók minden adatát

MySQL

```
SELECT *  
FROM `autok`  
WHERE `uzemanyag`  
      IN ('benzin', 'diesel', 'elektromos');
```

- Csak egyenlőség vizsgálat esetén alkalmazható
- Az `uzemanyag` mezőt csak egyszer kellett felsorolni
- ÉS alkalmazásakor nem zavar be a precedencia
- Számok esetén az aposztrófok elhagyhatóak. pl.: **IN**(1,5,8)

Linkek:

- MySQL dokumentáció: Az IN operátor

# Tartalom

## 2 Feltételes lekérdezések (WHERE)

- ÉS/VAGY kombinálása
- Azonos mezőre több lehetséges érték (IN)
- Azonos mezőre több lehetséges érték tagadással (NOT IN)
- Két érték között (BETWEEN)

# Azonos mezőre több lehetséges érték tagadással

Listázzuk ki azokat autókat, melyekre nem igaz, hogy **diesel** vagy **benzin** az üzemanyaga.

```
SELECT * FROM `autok`  
WHERE `uzemanyag` <> 'benzin'  
      OR `uzemanyag` <> 'diesel';
```

logikai hiba

rendszám	gyarto	tipus	kategoria	uzemanyag	szin
XXX-111	Opel	Adam	M1	benzin	piros
AAA-555	Honda	Jazz	M1	hibrid	kék
ABC-123	Ford	Focus	M1	diesel	kék
AA-AX-1234	Ford	Fiesta	M1	benzin	sárga

- Az `uzemanyag` <> 'benzin' megjeleníti az összes NEM benzinest, így az összes dieselt is!
- Az `uzemanyag` <> 'diesel' megjeleníti az összes NEM dieselt!

# Azonos mezőre több lehetséges érték tagadással

Listázzuk ki azokat autókat, melyekre nem igaz, hogy **diesel** vagy **benzin** az üzemanyaga.

MySQL

```
SELECT * FROM `autok`  
WHERE NOT (`uzemanyag` = 'benzin'  
          OR `uzemanyag` = 'diesel');
```

rendszám	gyarto	tipus	kategoria	uzemanyag	szin
AAA-555	Honda	Jazz	M1	hibrid	kék

- Megnézi minden egyes sorra, hogy diesel vagy benzin üzemű az autó,
- Amennyiben a válasz nem, akkor jeleníti csak meg.

# Azonos mezőre több lehetséges érték tagadással

Listázzuk ki azokat autókat, melyekre nem igaz, hogy **diesel** vagy **benzin** az üzemanyaga.

MySQL

```
SELECT * FROM `autok`  
WHERE `uzemanyag` <> 'benzin'  
AND `uzemanyag` <> 'diesel';
```

rendszám	gyarto	tipus	kategoria	uzemanyag	szin
AAA-555	Honda	Jazz	M1	hibrid	kék

- A feladatot átfogalmazva adódik egy másik megoldás:
  - "Listázzuk ki a se nem **benzines**, se nem **diesel** autók összes adatát."
- A zárójel felbontásakor a De Morgan-azonosságokat figyelembe kell venni

Linkek:

- De Morgan-azonosságok - Wikipedia

# De Morgan-azonosság példa

- Hogyan határozható meg, hogy egy jegy helyes?

$$\text{HelyesJegy} = ((1 \leq \text{jegy}) \wedge (\text{jegy} \leq 5))$$

- Mi ennek a tagadása?

$$\text{NemHelyesJegy} = \neg ((1 \leq \text{jegy}) \wedge (\text{jegy} \leq 5))$$

- A külső zárójel felbontható

$$\text{HelytelenJegy} = (1 > \text{jegy}) \wedge (\text{jegy} > 5)$$

- Melyik az a szám, amire egyszerre igaz, hogy kisebb, mint 1 és nagyobb is, mint 5?
- Egyik sem! **A felbontás hibás volt!**

# De Morgan-azonosság példa

- Hogyan határozható meg, hogy egy jegy helyes?

$$HelyesJegy = ((1 \leq jegy) \wedge (jegy \leq 5))$$

- Mi ennek a tagadása?

$$NemHelyesJegy = \neg ((1 \leq jegy) \wedge (jegy \leq 5))$$

- A külső zárójel felbontható

$$HelytelenJegy = (1 > jegy) \vee (jegy > 5)$$

- $A \leq$  tagadása  $a >$
- $A \geq$  tagadása  $a <$
- Az és ( $\wedge$ ) helyett vagy ( $\vee$ ) szerepel
- Mikor helytelen egy jegy?
  - Ha kisebb, mint 1, illetve
  - nagyobb, mint 5



# A NOT IN operátor

Listázzuk ki a se nem **benzines**, se nem **diesel** autók összes adatát.

MySQL

```
SELECT *  
FROM `autok`  
WHERE `uzemanyag`  
      NOT IN ('benzin', 'diesel');
```

- Egy zárójelben felsorolhatjuk a kizárandó értékeket.

Linkek:

- [MySQL dokumentáció: A NOT IN operátor](#)

# Tartalom

## 2 Feltételes lekérdezések (WHERE)

- ÉS/VAGY kombinálása
- Azonos mezőre több lehetséges érték (IN)
- Azonos mezőre több lehetséges érték tagadással (NOT IN)
- Két érték között (BETWEEN)

# Két érték közötti vizsgálat

Jelenítsük meg a 10 és 20 év közötti diákokat.

MySQL

```
SELECT * FROM `diakok`  
WHERE   `kor` >= 10  
        AND `kor` <= 20;
```

- A `kor` mezőt kétszer is szerepeltetni kell!
- Oda kell figyelni, hogy egyik oldalt se maradjon le az egyenlőség!
- Oda kell figyelni, hogy **és** kapcsolat legyen a két feltétel között.
- Összetett feltételben ez okozhat gondot!

# A BETWEEN ... AND ... operátor

Jelenítsük meg a 10 és 20 év közötti diákokat.

MySQL

```
SELECT * FROM `diakok`  
WHERE `kor` BETWEEN 10 AND 20;
```

- A 10 és 20, azaz a minimum és a maximum is benne lesz a szűrésben, nem lehet lefelejtetni az egyenlőséget
- Egy egységet alkot, így összetett feltételben zárójelek nélkül is használható

Linkek:

- [MySQL dokumentáció: A BETWEEN operátor](#)

# A NOT BETWEEN ... AND ... operátor

Jelenítsük meg azokat a diákokat, akik élettkora nem esik 10 és 20 közé.

MySQL

```
SELECT * FROM `diakok`  
WHERE `kor` NOT BETWEEN 10 AND 20;
```

Ugyanígy működne, ha a sima BETWEEN eredményét letagadnánk.

MySQL

```
SELECT * FROM `diakok`  
WHERE NOT (`kor` BETWEEN 10 AND 20);
```

- A zárójel itt elhagyható, az átláthatóság miatt került be.

Linkek:

- [MySQL dokumentáció: A NOT BETWEEN operátor](#)

# Tartalom I

## 3 A NULL érték

# A NULL érték

- A null érték speciális érték
- A NULL nem egyenlő 0-val!
- A NULL nem egyenlő az üres szöveggel!
- Adatbázisan NULL jelentése: Az adott mező értéke nincs megadva, nincs kitöltve, ismeretlen.

# A NULL vizualizálása



ábra: A NULL érték vizualizálása

**Forrás:** <https://www.reddit.com/r/ProgrammerHumor>



# IS NULL

Az IS NULL segítségével ellenőrizhetjük, hogy a mező értéke NULL-e.

```
SELECT * FROM `autok`  
WHERE `tipus` IS NULL;
```

MySQL

# IS NOT NULL

Az IS NOT NULL segítségével ellenőrizhetjük, hogy a mező értéke **nem** NULL.

```
FROM `autok`  
WHERE `tipus` IS NOT NULL;
```

MySQL

# COALESCE()

```
COALESCE(value,...)
```

- Visszaadja az első **nem NULL** értéket

```
SELECT COALESCE(NULL, NULL, NULL, 'A', 'B') AS `e`  
FROM DUAL;
```

MySQL

e

A

# COALESCE példa

Legyen a **users** tábla tartalma az alábbi:

id	full_name	nickname	username	email
1	Tóth Péter	Peti	Ptr99	ptr.99@gmail.com
2	-	Rebeka	7reb13	rebeka@gmail.com
3	Nagy Andrea	-	nandrea	n.andrea@gmail.com
4	-	-	joey	joey@gmail.com
5	-	-	-	kazmer@gmail.com

Mit eredményez az alábbi lekérdezés?

```
SELECT `id`,  
       COALESCE(`full_name`, `nickname`, `username`, 'Ismeretlen')  
FROM `users`
```

MySQL

1 – Tóth Péter, 2 – Rebeka, 3 – Nagy Andrea, 4 – Joey, 5 – Ismeretlen