

# Adatbázis kezelés I.

## SQL bevezető

Rostagni Csaba

2024. szeptember 9.

# Ezen az órán... I

- 1 Egyedi értékek
- 2 Feltételes lekérdezések (WHERE)
- 3 Sorok számának limitálása (LIMIT)
- 4 Dátum és idő függvények

# Tartalom I

## 1 Egyedi értékek

# Egyedi sorok

Listázzuk ki a **kategóriákat**!

MySQL

```
SELECT `kategoria`  
FROM `autok`;
```

kategoria
M1
M1
M1
M1

Látszik, hogy az M1 többször is megjelenik.

# Egyedi sorok (DISTINCT) 1.

Listázzuk ki a **kategóriákat**! Ismétlődések ne legyenek!

MySQL

```
SELECT DISTINCT `kategoria`  
FROM `autok`;
```

kategoria
M1

## Egyedi sorok (DISTINCT) 2.

Listázzuk ki az egyedi **gyártókat** és **kategóriákat**!

MySQL

```
SELECT DISTINCT `gyarto`, `kategoria`  
FROM `autok`;
```

gyarto	kategoria
Opel	M1
Honda	M1
Ford	M1

Az **M1** kategória ugyan többször is megjelenik, de minden sorban más más gyártó található mellette!

# Tartalom I

- 2 Feltételes lekérdezések (WHERE)
  - Tagadás

# Tartalom

- 2 Feltételes lekérdezések (WHERE)
  - Tagadás



# Egyszerű tagadás

Listázzuk ki a **nem benzines** autókat.

MySQL

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

rendszám	gyarto	tipus	kategoria	uzemanyag	szin
AAA-555	Honda	Jazz	M1	hibrid	kék
ABC-123	Ford	Focus	M1	diesel	kék

- Használhattuk volna a `!=` operátort, csak az SQL szabvány nem azt tartalmazza.
- Egyszerű feltételek esetén ilyen egyszerű tagadni.

# Egyszerű tagadás (NOT)

Listázzuk ki a **nem benzines** autókat.

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

MySQL

rendszám	gyarto	tipus	kategoria	uzemanyag	szin
AAA-555	Honda	Jazz	M1	hibrid	kék
ABC-123	Ford	Focus	M1	diesel	kék

Megjegyzés:

- Ennél az egyszerű példánál a zárójelezés elhagyható, összetettebbeknél célszerű kitenni

# Tartalom I

## 3 Sorok számának limitálása (LIMIT)

# Sorok számának limitálása (LIMIT)

A LIMIT segítségével megadhatjuk, hogy az eredmény hány sorát szeretnénk megkapni.

```
SELECT * FROM `tanulok`  
LIMIT 2;
```

MySQL

[https://www.w3schools.com/php/php\\_mysql\\_select\\_limit.asp](https://www.w3schools.com/php/php_mysql_select_limit.asp)

# Lapozó

Első N rekord kihagyása:

MySQL

```
SELECT id, vnev, knev  
FROM `tanulok`  
LIMIT 10, 5;
```

- Kihagyja az első 10 tanulót és onnantól kezdve jelenít meg legfeljebb 5 tanulót.
- Felhasználás a gyakorlatban:
  - Nagy mennyiségű adat esetén egy weboldalon nem jó ötlet az összes adatot egyszerre megjeleníteni, célszerű több oldalra bontani.
  - Amennyiben 1 oldalon 5 adatot jelenítünk meg, úgy a fenti példa a 3. oldal adatait kéri le
  - Az első két oldalon megjelenő 5-5, azaz összesen 10 rekord kerül kihagyásra, majd jön az aktuális oldal tartalma

# Legjobb 3 tanuló

Listázzuk ki a három legjobb tanulót.

MySQL

```
SELECT id, vnev, knev  
FROM `tanulok`  
ORDER BY `atlag` DESC  
LIMIT 3;
```

- A tanulókat átlag alapján rendezi **csökkenő** sorrendbe
- Az átlag nem kerül megjelenítésre
- Az **ORDER BY** és **LIMIT** kombinálásával létrehozható toplista

# A legmagasabb tanuló

Hogy hívják a legmagasabb tanulót?

MySQL

```
SELECT vnev, knev  
FROM `tanulok`  
ORDER BY `magassag` DESC  
LIMIT 1;
```

- A tanulókat a magasságuk alapján rendezi csökkenő sorrendbe
- A magasság nem kerül megjelenítésre
- Az **ORDER BY** és **LIMIT** kombinálásával meghatározható hogyan hívják a legmagasabb tanulót

# A legidősebb tanuló

Hogy hívják a legidősebb tanulót?

MySQL

```
SELECT vnev, knev  
FROM `tanulok`  
ORDER BY `szuletesi_datum` ASC  
LIMIT 1;
```

- A tanulókat a születési dátumok alapján rendezi **növekvő** sorrendbe
  - Minél korábban született valaki, annál idősebb lesz
- Az **ORDER BY** és **LIMIT** kombinálásával meghatározható hogyan hívják legidősebb tanulót



# Tartalom I

- 4 Dátum és idő függvények
  - Dátum/idő részének kinyerése
  - Aktuális dátum/idő

# Tartalom

- 4 Dátum és idő függvények
  - Dátum/idő részének kinyerése
  - Aktuális dátum/idő

# Dátum/Idő részének kinyerése

**DATE()** Megadja a dátum részt egy dátum/időből

**TIME()** Megadja az idő rész egy dátum/időből

**YEAR()** Megadja az évet

**MONTH()** Megadja a hónapot

**DAY()** Megadja a napot

**HOURL()** Megadja az órát

**MINUTE()** Megadja a percet

**SECOND()** Megadja a másodpercet

**WEEKDAY()** Megadja, hogy az adott dátum a hét hányadik napja

- Paramétere lehet egy mező **YEAR(`született`)**,
- vagy konkrét érték **YEAR('2022-01-12')**

Linkek:

- MySQL dokumentáció: Dátum és idő függvények

# A YEAR() függvény használata

Melyik évben született az 1-es azonosítójú tanuló?

MySQL

```
SELECT YEAR(`szul_ido`)  
FROM `tanulok`  
WHERE `id` = 1;
```

- A `szul\_ido` a születési dátumokat tartalmazza  
(pl.: '2003-03-16')
- A `YEAR()` függvény az évet nyeri ki belőle. (pl.: 2003)

# A YEAR() függvény használata

A hét melyik napján született az 1-es azonosítójú tanuló?

MySQL

```
SELECT WEEKDAY(`szul_ido`)  
FROM `tanulok`  
WHERE `id` = 1;
```

- A `szul\_ido` a születési dátumokat tartalmazza (pl.: '2003-03-16')
- A `WEEKDAY()` függvény a nap sorszámát határozza meg. (pl.: 6)
  - 0 - Hétfő
  - 1 - Kedd
  - ...
  - 6 - vasárnap

# Tartalom

- 4 Dátum és idő függvények
  - Dátum/idő részének kinyerése
  - Aktuális dátum/idő

# Aktuális dátum és idő

- Aktuális dátum
  - `CURDATE()`
  - `CURRENT_DATE()`
- Aktuális idő
  - `CURTIME()`
  - `CURRENT_TIME()`
- Aktuális dátum és idő
  - `NOW()`

Linkek:

- [MySQL dokumentáció: Dátum és idő függvények](#)