

# Adatbázis kezelés I.

## Aggregált (összesítő) függvények

Rostagni Csaba

2024. október 7.

# Ezen az órán... I

## 1 Aggregált (összesítő) függvények

# Tartalom I

## 1 Aggregált (összesítő) függvények

- COUNT
- SUM
- AVG
- MIN/MAX

# Összesítő függvények

Az összesítő (aggregált) függvények a meghatározott kifejezésen hajtanak végre különböző műveleteket.

- Alapértelmezetten a NULL értékeket nem veszik számításba
- Gyakran használt összesítő függvények:
  - COUNT()
  - SUM()
  - AVG()
  - MIN()
  - MAX()

Linkek:

- MySQL dokumentáció: Összesítő függvények

# Tartalom

## 1 Aggregált (összesítő) függvények

- COUNT
- SUM
- AVG
- MIN/MAX

# A COUNT() függvény

Megszámolja a lekérdezés által visszaadott sorokban a nem NULL értékeket.

- Van lehetőség NULL beleszámítására is
- Amennyiben nincs a feltételeknek megfelelő találat, úgy 0 lesz a függvény kimenete
- A MySQL nem csak a számokat tartalmazó mezőkön értelmezi
- Az eredmény BIGINT típusú lesz

Linkek:

- MySQL dokumentáció: [COUNT\(\)](#)

# A termékek tábla

id	nev	kategoria	netto	penznem	afa
1	4K TV	tv	499	EUR	0.19
2	Mobil 32GB	mobil	299	EUR	0.19
3	Mobil 128GB	mobil	679	EUR	0.19
4	Olcsó laptop	laptop	269	EUR	0.19
5	Drága laptop	laptop	1729	EUR	0.19
6	Könyv	könyv	NULL	NULL	NULL

## Példa: COUNT(`netto`) példa

MySQL

```
SELECT COUNT(`netto`) AS `darab`  
FROM `termekek`;
```

darab

5

- Ahol a netto értéke **NULL**, azt a sort kihagyja a számításból.



## Példa: COUNT() - szöveget tartalmazó oszlopon

MySQL

```
SELECT COUNT(`penznem`) AS `db`  
FROM `termekek`;
```

db

5

- Ahol a penznem értéke **NULL**, azt a sort kihagyja a számításból.

# Példa: COUNT(\*)

MySQL

```
SELECT COUNT(*) AS `db_csillag`  
FROM `termekek`;
```

db_csillag
6

- A **COUNT(\*)** a visszaadott sorok számát számolja meg, így a NULL értékeket is beleveszi a számításába!

# Példa: COUNT() - a tábla elsődleges kulcsára alkalmazva

MySQL

```
SELECT COUNT(`id`) AS `darab_id_szerint`  
FROM `termekek`;
```

darab_id_szerint
6

- Az elsődleges kulcs sosem lehet **NULL**
- Érdekes az elsődleges kulcsot megadni paraméterként
- Az elsődleges kulcs gyakran id vagy Azon néven szerepel

# Egyedi értékek az összesítő függvényekben

Hány **különböző** kategória található a táblában?

MySQL

```
SELECT COUNT(DISTINCT `kategoria`) AS `db_kategoria`  
FROM `termekek`;
```

db_kategoria
4

- Amennyiben a DISTINCT kulcsszót a COUNT() függvényen belül helyezzük el, úgy az azonos értékeket egyszer veszi csak számításba.

# Tartalom

## 1 Aggregált (összesítő) függvények

- COUNT
- SUM
- AVG
- MIN/MAX

# A SUM() függvény

Összeadja a meghatározott kifejezés értékeit.

- A **DISTINCT** megadásával csak az egyedi értékeket összegzi
- Amennyiben a lekérdezés egyetlen sorral sem tér vissza, úgy **NULL** lesz az eredmény
- A **NULL** értékek összege is **NULL** lesz

Linkek:

- MySQL dokumentáció: SUM()

# A termékek tábla

id	nev	kategoria	netto	penznem	afa
1	4K TV	tv	499	EUR	0.19
2	Mobil 32GB	mobil	299	EUR	0.19
3	Mobil 128GB	mobil	679	EUR	0.19
4	Olcsó laptop	laptop	269	EUR	0.19
5	Drága laptop	laptop	1729	EUR	0.19
6	Könyv	könyv	NULL	NULL	NULL

# Példa: SUM()

Mennyi a termékek *nettó értéke* **összesen**?

MySQL

```
SELECT SUM(`netto`) AS `ossz`  
FROM `termekek`;
```

OSSZ

3475

- A könyv netto értéke **NULL**
  - Nem adta hozzá az eredményhez
  - Nem lett a végeredmény **NULL**



## Példa: SUM() - NULL értékek kihagyásával

MySQL

```
SELECT SUM(`netto`) AS `ossz`  
FROM `termekek`  
WHERE `netto` IS NOT NULL;
```

OSSZ

3475

- A könyv sora kimarad a számításból

# Példa: SUM() - NULL értékekkel

MySQL

```
SELECT SUM(`netto`) AS `ossz`  
FROM `termekek`  
WHERE `netto` IS NULL;
```

OSSZ

NULL

- A könyv sorában lesz egyedül **NULL** érték, a végeredmény is **NULL** lett

## Példa: SUM() - feltétellel

Mennyibe kerülnek a mobilok?

MySQL

```
SELECT SUM(`netto`) AS `ossz_mobil`  
FROM `termekek`  
WHERE `kategoria` = 'mobil';
```

ossz_mobil
------------

978
-----

- Az összegzés előtt szűr a **WHERE** feltétel alapján
- Csak azokat a sorokat veszi, ahol a **kategoria** értéke "mobil"

# Példa: SUM() - számított mező összegzése

Mennyi a termékek **bruttó** értéke?

MySQL

```
SELECT  
    ROUND( SUM(`netto` * (1 + `afa`)) , 2) AS `ossz`  
FROM  
    `termekek`;
```

OSSZ

4135.25

- Kiszámítja a bruttó értéket: ``netto` * (1 + `afa`)`
- A számított értékeket összegzi: `SUM()`
- A végeredményt kerekíti 2 tizedesre: `ROUND()`

## italok

id	megnevezes	ar	db	drs
1	Fanta narancs 1,75 l	639	4	50
2	Fanta szőlő 500 ml	419	1	50
3	Xixo Ice Tea epres 0,25l	189	6	50
4	Magyar Tej ESL tej 2,8% 1 l	659	2	0
5	Piroska narancs ízű szörp 0,7 l	819	1	178

- DSR: Betétdíj (Deposit and Return System)

# COUNT vs SUM

Hány ital szerepel a a táblában?

- 5 ital van a táblában
- Valójában a sorok száma a fontás számunkra

```
SELECT COUNT(*) AS `darab` FROM `italok`;
```

MySQL

Hány italt vásároltunk?

```
SELECT SUM(`db`) AS `darab` FROM `italok`;
```

MySQL

- 14 darabot vásároltunk a tábla szerint
- A darabszámokat kellett összeadni

# Tartalom

## 1 Aggregált (összesítő) függvények

- COUNT
- SUM
- **AVG**
- MIN/MAX

# Az AVG() függvény

Meghatározza a megadott kifejezés átlagát.

- A **DISTINCT** megadásával csak az egyedi értékeket átlagolja
- Amennyiben a lekérdezés egyetlen sorral sem tér vissza, úgy **NULL** lesz az eredmény
- A **NULL** értékek átlaga is **NULL** lesz

Linkek:

- MySQL dokumentáció: [AVG\(\)](#)



# A termékek tábla

id	nev	kategoria	netto	penznem	afa
1	4K TV	tv	499	EUR	0.19
2	Mobil 32GB	mobil	299	EUR	0.19
3	Mobil 128GB	mobil	679	EUR	0.19
4	Olcsó laptop	laptop	269	EUR	0.19
5	Drága laptop	laptop	1729	EUR	0.19
6	Könyv	könyv	<i>NULL</i>	<i>NULL</i>	<i>NULL</i>

## Példa: AVG()

Mennyi a termékek *nettó árának* az **átlaga**?

MySQL

```
SELECT AVG(`netto`) AS `netto_atlag`  
FROM `termekek`;
```

netto_atlag
695

# Példa: AVG()

Mennyi a termékek *bruttó árának* az **átlaga**?

MySQL

```
SELECT  
    AVG(`netto` * ( 1 + `afa` ) ) AS `brutto_atlag`  
FROM  
    `termekek`;
```

brutto\_atlag

827.0499983429909

# Tartalom

## 1 Aggregált (összesítő) függvények

- COUNT
- SUM
- AVG
- MIN/MAX

# A MIN() függvény

A megadott kifejezés legkisebb értékével tér vissza.

- Szöveggel is működik, az eredmény a karakterkódolástól függhet
- A **DISTINCT** megadásával csak az egyedi értékeket veszi figyelembe, de itt nem számít, mivel csak egy értéket ad úgyis vissza
- Amennyiben a lekérdezés egyetlen sorral sem tér vissza, úgy **NULL** lesz az eredmény
- A **NULL** értékek minimuma is **NULL** lesz

Linkek:

- MySQL dokumentáció: MIN()

# A MAX() függvény

A megadott kifejezés legnagyobb értékével tér vissza.

- Szöveggel is működik, az eredmény a karakterkódolástól függhet
- A **DISTINCT** megadásával csak az egyedi értékeket veszi figyelembe, de itt nem számít, mivel csak egy értéket ad úgyis vissza
- Amennyiben a lekérdezés egyetlen sorral sem tér vissza, úgy **NULL** lesz az eredmény
- A **NULL** értékek maximum is **NULL** lesz

Linkek:

- MySQL dokumentáció: MAX()

# A termékek tábla

id	nev	kategoria	netto	penznem	afa
1	4K TV	tv	499	EUR	0.19
2	Mobil 32GB	mobil	299	EUR	0.19
3	Mobil 128GB	mobil	679	EUR	0.19
4	Olcsó laptop	laptop	269	EUR	0.19
5	Drága laptop	laptop	1729	EUR	0.19
6	Könyv	könyv	<i>NULL</i>	<i>NULL</i>	<i>NULL</i>

## Példa: MIN()

Mennyi a **legolcsóbb** termék *nettó ára*?

MySQL

```
SELECT MIN(`netto`) AS `min_netto`  
FROM `termekek`;
```

min_netto
269



# Példa: MIN()

Mi a *neve* annak a terméknek, ami **ABC**-ben az **első**?

MySQL

```
SELECT MIN(`nev`) AS `min_nev`  
FROM `termekek`;
```

min\_nev

4K TV

## Példa: a MIN() hibás használata

Mi a **neve** a *nettó ár* szerint **legolcsóbb** terméknek?

```
SELECT nev AS `min_nev`, MIN(`netto`)  
FROM `termekek`;
```

logikai hiba

#1140 - In aggregated query without GROUP BY, expression #1 of SELECT list contains nonaggregated column 'pelda.termek.nev'; this is incompatible with sql\_mode=only\_full\_group\_by

Hiba!

- Senki sem kérte az árat, hanem csak a nevet

## Példa: a MIN() hibás használata

Mi a **neve** a *nettó ár* szerint **legolcsóbb** terméknek?

```
SELECT nev  
FROM `termekek`  
WHERE MIN(`netto`);
```

logikai hiba

```
#1111 - Invalid use of group function
```

Hiba!

- Az összesítő függvények nem használhatóak a **WHERE** záradékban

## Példa: a MIN() helyett ORDER BY és LIMIT

Mi a **neve** a *nettó ár* szerint **legolcsóbb** terméknek?

```
SELECT nev AS `min_nev`  
FROM `termekek`  
ORDER BY `netto`  
LIMIT 1;
```

logikai hiba

min\_nev

Könyv

- A "legolcsóbb" nettó érték a **NULL** lesz a rendezés szerint
- Előre ki kell szűrni a **NULL** értékeket

# Példa: a MIN() helyett ORDER BY és LIMIT

Mi a **neve** a *nettó ár* szerint **legolcsóbb** terméknek?

MySQL

```
SELECT nev AS `min_nev`  
FROM `termekek`  
WHERE `netto` IS NOT NULL  
ORDER BY `netto`  
LIMIT 1;
```

min\_nev

Olcsó laptop

# Példa: MIN() és MAX() egy lekérdezésben

MySQL

```
SELECT
    MIN(`netto`) AS `min_ar`,
    MAX(`netto`) AS `max_ar`
FROM
    `termekek`;
```

min_ar	max_ar
269	1729

- Egy lekérdezésben több összesítő függvény is szerepelhet
- Nem csak a **MIN()** és a **MAX()**