

Adatbázis kezelés I.

Számított mezők

Rostagni Csaba

2022. október 6.

Ezen az órán... I

- 1 ALIAS
- 2 Számított értékek
- 3 Matematikai függvények
- 4 Szöveg függvények
- 5 Egyéb hasznos függvények

Tartalom I

1 ALIAS

ALIAS - Álnevek

- Ideiglenes neveket lehet adni
 - adatbázisnak
 - táblának
 - mezőknek
 - számított mezőknek
- Az alábbi záradékokban használható mezőkre hivatkozáskor
 - **GROUP BY**
 - **ORDER BY**
 - **HAVING**
- Szabvány szerint a **WHERE** záradékban nem használható

Linkek:

- MySQL dokumentáció: ALIAS problémák

ALIAS definiálása

MySQL

```
SELECT `gyarto` AS `marka`  
FROM autok;
```

- Az **AS** kulcsszóval lehet mezőt ideiglenesen átnevezni.

MySQL

```
SELECT `gyarto` `marka`  
FROM autok;
```

- Az **AS** kulcsszó elhagyható

ALIAS alkalmazása

MySQL

```
SELECT `gyarto` AS `marka`  
FROM autok;
```

- A név megadásánál **célszerű a backticket alkalmazni**

MySQL

```
SELECT `gyarto` AS 'marka'  
FROM autok;
```

- A név megadható aposztróffal is, de **nem ajánlott**

ALIAS táblanevekre

MySQL

```
SELECT `auto`.`gyarto`, `auto`.`tipus`  
FROM autok;
```

- A SELECT részben a tábla neve is megadható

MySQL

```
SELECT `a`.`gyarto`, `a`.`tipus`  
FROM auto AS a;
```

- A tábla neveket is át lehet nevezni
- Ez használható például a SELECT részben

Álnév problémák

```
SELECT `gyarto` AS 'marka'
FROM autok
ORDER BY 'marka' ASC;
```

logikai hiba

SELECT	ORDER BY
Opel	marka
Honda	marka
Ford	marka
Ford	marka

- Az eredmény **nincs növekvő sorrendben** az aposztróf használatakor
- A rendezést egy láthatatlan oszlop alapján végezte, de minden egyes sorában ugyanaz az adat található: marka

```
SELECT `gyarto` AS `marka`
FROM autok
ORDER BY `marka` ASC;
```

MySQL

SELECT	ORDER BY
Ford	Ford
Ford	Ford
Honda	Honda
Opel	Opel

- A backtick használata a kívánt sorrendet eredményezi

Tartalom I

2 Számított értékek

Számított mezők

- SQL lekérdezésben lehetőség van számítások elvégzésére
- Szerepelhet benne:
 - Konstans érték: 1, 'hello', '2000-01-01', true
 - Egy mező az adatbázisból: `ar`
 - Valamilyen függvény: sqrt(9), round(1.975,2)

A DUAL "tábla"

MySQL

```
SELECT 10 + 5 AS `eredmeny`  
FROM dual;
```

- Előfordulhatnak olyan lekérdezések, amit nem táblától szeretnénk lekérdezni.
- A **dual** egy speciális „tábla”, ahonnan bármit lekérdezhetünk.
 - **Itt a backtick nem használható!**

MySQL

```
SELECT 10 + 5 AS `eredmeny`;
```

- Más adatbázisoknál kötelező
- A MySQL-ben elhagyható

Linkek:

- [SELECT - MySQL dokumentáció](#)

Aritmetikai operátorok

Operátor	Művelet
-	Negatív előjel
*	Szorzás
/	(Valós) Osztás
MOD vagy %	Modulo operátor / Maradék képzés
DIV	Egész osztás
+	Összeadás
-	Kivonás

Linkek:

- Aritmetikai műveletek - MySQL dokumentáció

A termék 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>

Számított mezők

Jelenítsük meg a termékek bruttó árait.

MySQL

```
SELECT
    `nev` AS `termek_nev`,
    `netto` * 1.27 AS `brutto`
FROM
    `termekek`;
```

- A lekérdezések során a tábla mezői felhasználhatóak különböző számításokhoz.

Szűrés számított mező alapján

Jelenítsük meg azokat a termékeket, melyek **bruttó ára** több, mint 400 euro

```
SELECT
    `nev` AS `termek_nev`,
    `netto` * 1.27 AS `brutto`
FROM `termekek`
WHERE `brutto` > 400;
```

Szintaktikai hiba

#1054 - A(z) 'brutto' oszlop ervenytelen 'where clause'-ben

Hiba!

- Az ANSI SQL szabvány szerint a **WHERE** záradékban nem használható ALIAS

Számított mezők feltételként

Jelenítsük meg azokat a termékeket, melyek **bruttó ára** több, mint 400 euro

MySQL

```
SELECT
    `nev` AS `termek_nev`,
    `netto` * 1.27 AS `brutto`
FROM `termekek`
WHERE `netto` * 1.27 > 400;
```

`nev`	`brutto`
4K TV	633.73
Mobil 128GB	862.33
Drága laptop	2195.83

- A **WHERE** záradékban alkalmazhatóak számított értékek

Rendezés számított mező alapján

Jelenítsük meg a termékek nevét és a bruttó árat a bruttó szerinti növekvő sorrendben.

```
SELECT `nev`, `netto` * (1 + `afa`) AS 'brutto'  
FROM `termekek`  
ORDER BY 'brutto' ASC;
```

logikai hiba

- A kód le fog futni, de nem a várt eredménnyel.

Rendezés számított mező alapján

`nev`	`brutto`	'brutto'
4K TV	94.80999881029129	brutto
Mobil 32GB	56.80999928712845	brutto
Mobil 128GB	129.00999838113785	brutto
Olcsó laptop	51.10999935865402	brutto
Drága laptop	328.50999587774277	brutto
Könyv	-	brutto

- Az eredmény rendezett, de egy harmadik, mesterségesen generált mező alapján.
- Fontos, hogy az álnév backtick legyen, itt ennek hiányában nem működött a rendezés.

Rendezés számított mező alapján

Jelenítsük meg a termékek nevét és a bruttó árat a bruttó szerinti növekvő sorrendben.

MySQL

```
SELECT `nev`, `netto` * (1 + `afa`) AS `brutto`  
FROM `termekek`  
ORDER BY `netto` * (1 + `afa`) ASC;
```

- A rendezési feltétel kiszámítása elvégezhető az **ORDER BY** záradékban

MySQL

```
SELECT `nev`, `netto` * `afa` AS `brutto`  
FROM `termekek`  
ORDER BY `brutto` ASC;
```

- Az **ORDER BY** záradékban használható a **SELECT**-ben meghatározott álnév

Rendezés számított mező alapján

`nev`	`brutto`
Könyv	<i>NULL</i>
Olcsó laptop	51.10999935865402
Mobil 32GB	56.80999928712845
4K TV	94.80999881029129
Mobil 128GB	129.00999838113785
Drága laptop	328.50999587774277

Tartalom I

- 3 Matematikai függvények
 - Egyszerű matematikai függvények
 - Kerekítés

Tartalom

- 3 Matematikai függvények
 - Egyszerű matematikai függvények
 - Kerekítés

Matematikai függvények

`ABS(x)` $|x|$ abszolút érték

`MOD(x,y)` maradékos osztás

`POW(x,y)` x^y hatványozás

`POWER(x,y)` x^y hatványozás

`SQRT(x)` \sqrt{x} gyök

<https://dev.mysql.com/doc/refman/8.0/en/mathematical-functions.html>

Függvényhasználat: SQRT()

A lekérdezésben használhatunk függvényeket, például a gyök függvényt!

MySQL

```
SELECT SQRT(9) AS `negyzetgyok`  
FROM DUAL;
```

negyzetgyok
3

Linkek:

- [MySQL dokumentáció: Matematikai függvények](#)

PI()

PI()

- Megadja a π (pi) értékét.
- Alapértelmezetten 7 számjegyet jelenít meg
 - ebből 1 számjegy az egész résznek,
 - és 6 számjegy a tört résznek.
- Ennél nagyobb pontossággal tárolja és számol vele.

```
SELECT PI() as `e` FROM DUAL;
```

MySQL

e

3.141593

Tartalom

- 3 Matematikai függvények
 - Egyszerű matematikai függvények
 - Kerekítés

Kerekítő függvények

CEILING(x) $\lceil x \rceil$ felső egész rész

CEIL(x) alias a CEILING() függvényre

FLOOR(x) $\lfloor x \rfloor$ alsó egész rész

ROUND(x,n) matematikai kerekítés

TRUNCATE(x,n) nem kerekít, levágja a tizedes jegyeket

Linkek:

- MySQL dokumentáció: Matematikai függvények

Kerekítés $\text{ROUND}(x,d)$

x Kerekítendő érték

d tizedesek száma

A bruttó árat két tizedesre kerekítve jelenítse meg!

MySQL

```
SELECT ROUND(`netto` * 1.27 ,2) AS `brutto`  
FROM `termek`;
```

- Alapvetően a matematikai kerekítést alkalmazza, 5-től felfelé kerekít
- Lebegőpontos számábrázolás esetén bizonyos rendszereken előfordul, hogy a "Round to Even", más néven "Banker's Rounding" módszert alkalmazhatja

Linkek:

- MySQL dokumentáció: Az $\text{ROUND}()$ függvény
- Wikipedia: Szimmetrikus kerekítés (Banker's Rounding)

ROUND() példák

MySQL

```
SELECT ROUND(123.4567) as `eredmeny`  
FROM DUAL;
```

eredmeny

123

- Ha a második paraméter 0, vagy nincs, akkor egészre kerekít

MySQL

```
SELECT ROUND(123.4567,1) as `eredmeny`  
FROM DUAL;
```

eredmeny

123.5

- A második paraméter 1, így egy tizedesre kerekít

MySQL

```
SELECT ROUND(123.4567,-1) as `eredmeny`  
FROM DUAL;
```

eredmeny

120

- Mivel a második paraméter -1, így a tizes helyiértékű számra kerekíti

Linkek:

- [MySQL dokumentáció: Az ROUND\(\) függvény](#)

CEIL(), FLOOR(), és ROUND() összehasonlítása

MySQL

```
SELECT CEIL(222.111) as `eredmeny`  
FROM DUAL;
```

eredmeny

223

- A CEIL() függvény visszaadja a **tőle nem kisebb** legkisebb egész számot

MySQL

```
SELECT FLOOR(111.888) as `eredmeny`  
FROM DUAL;
```

eredmeny

111

- A FLOOR() függvény visszaadja a **tőle nem nagyobb** legnagyobb egész számot

MySQL

```
SELECT ROUND(111.888) as `eredmeny`  
FROM DUAL;
```

eredmeny

112

- A ROUND() függvény kerekítést alkalmaz

Linkek:

CEIL(), FLOOR(), és ROUND() negatív számokkal

MySQL

```
SELECT CEIL(-222.111) as `eredmeny`  
FROM DUAL;
```

eredmeny

-222

- A CEIL() függvény visszaadja a **tőle nem kisebb** legkisebb egész számot

MySQL

```
SELECT FLOOR(-111.888) as `eredmeny`  
FROM DUAL;
```

eredmeny

-112

- A FLOOR() függvény visszaadja a **tőle nem nagyobb** legnagyobb egész számot

MySQL

```
SELECT ROUND(-111.888) as `eredmeny`  
FROM DUAL;
```

eredmeny

-112

- A ROUND() függvény kerekítést alkalmaz

Linkek:

Tartalom I

- 4 Szöveg függvények
 - Összefűzés
 - Kis- és nagybetűk
 - Hossz

Tartalom

- 4 Szöveg függvények
 - Összefűzés
 - Kis- és nagybetűk
 - Hossz

CONCAT()

```
CONCAT(str1,str2,...)
```

- Összefűzi az argumentumként kapott értékeket
- A számokat átalakítja szöveggé
- Amennyiben tartalmaz **NULL** értéket, úgy a végeredmény is **NULL** lesz

Linkek:

- MySQL dokumentáció: [CONCAT\(\)](#)

CONCAT() példák

MySQL

```
SELECT CONCAT('A', 'B', 'C', 'D') as `e`  
FROM DUAL;
```

e

ABCD

- Több, mint két argumentum is megadható

MySQL

```
SELECT CONCAT('Hello', ' ', 'World') as `e`  
FROM DUAL;
```

e

Hello World

- A szóköz külön argumentumként lett megadva

MySQL

```
SELECT CONCAT(15, ' cm') as `e`  
FROM DUAL;
```

e

15 cm

- A szóköz a ' cm' értékben található meg

CONCAT() példák

MySQL

```
SELECT  
    CONCAT(`magassag` / 100, ' m') as `magassag_meterben`  
FROM `tanulok`;
```

magassag_meterben
1,72 m
1,83 m
1,85 m
...

- A `magassag` a `tanulok` tábla egyik oszlopa
- A magasság cm-ből m-re át lett számítva
- A szóköz a ' m' értékben található meg

CONCAT_WS()

```
CONCAT_WS(separator, str1, str2, ...)
```

- "Concatenate With Separator"
- Összefűzi az argumentumként kapott értékeket
- Az első argumentum az elválasztó karakter
- Az elválasztó karaktert a legvégére nem teszi ki

```
SELECT CONCAT_WS('*', 'alma', 'barack', 'eper') AS `e`  
FROM DUAL;
```

MySQL

e

alma*barack*eper

Linkek:

- [MySQL dokumentáció: CONCAT_WS\(\)](#)

Tartalom

- 4 Szöveg függvények
 - Összefűzés
 - Kis- és nagybetűk
 - Hossz

UPPER()

```
UPPER(str)
```

- Nagybetűssé alakítja a szöveget (str)
- Alapértelmezetten a latin1 kódolást (cp1252 West European) használja
- Amennyiben a tábla karakterkódolás jól van megadva multibyte karaktereket is jól kezel

MySQL

```
SELECT UPPER('heLLo') FROM dual;
```

e

HELLO

LOWER()

LOWER(str)

- Kisbetűssé alakítja a szöveget (str)
- Alapértelmezetten a latin1 kódolást (cp1252 West European) használja
- Amennyiben a tábla karakterkódolás jól van megadva multibyte karaktereket is jól kezel

MySQL

```
SELECT LOWER('heLLo') FROM dual;
```

e

hello

Tartalom

- 4 Szöveg függvények
 - Összefűzés
 - Kis- és nagybetűk
 - Hossz

LENGTH()

LENGTH(str)

- Megadja a szöveg (str), hosszát **byteokban**
- A multibyte karaktereket **többször** számolja

LENGTH() példák

MySQL

```
SELECT LENGTH('car') as `e`  
FROM DUAL;
```

e

3

- Az egy byteos karakterek (ASCII első 128 karaktere) hossza megegyezik a karaktereinek számával

logikai hiba

```
SELECT LENGTH('autó') as `e`  
FROM DUAL;
```

e

5

- Az "autó" 4 betűs szó, de a hosszú "ó" multibytos karakter, így lesz a végeredmény 5

CHAR_LENGTH()

```
CHAR_LENGTH(str)
```

- Megadja a szöveg (str), hosszát **karakterekben**
- A multibyte karaktereket **egyszer** számolja
- Szinonímák erre a függvényre:
 - **CHARACTER_LENGTH**(str)

CHAR_LENGTH() példák

```
SELECT CHAR_LENGTH('car') as `e`  
FROM DUAL;
```

MySQL

e

3

- Az egy byteos karakterek (ASCII első 128 karaktere) hossza megegyezik a karaktereinek számával

```
SELECT CHAR_LENGTH('autó') as `e`  
FROM DUAL;
```

MySQL

e

4

- Az "autó" 4 betűs szó, amit helyesen megállapított a függvény

LENGTH() és CHAR_LENGTH() összehasonlítása

```
SELECT LENGTH('árvíztűrőtükörfúrógép') as `e`  
FROM DUAL;
```

logikai hiba

e

30

```
SELECT CHAR_LENGTH('árvíztűrőtükörfúrógép') as `e`  
FROM DUAL;
```

MySQL

e

21

- A **LENGTH()** a byteok számát, míg a **CHAR_LENGTH()** a karakterek számát adja meg, így utóbbi a multibyteos karakterek esetén is helyesen állapítja meg a szöveg hosszát.

Tartalom I

5 Egyéb hasznos függvények

COALESCE()

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

- Visszaadja az első **nem NULL** értéket

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

MySQL

e

A