

- Beépített függvények

Már a középiskolai tanulmányaink során sok függvénnyel találkoztunk akár a táblázatkezelés akár az adatbáziskezelés témakörében. A DBMS rendszerek is sok beépített függvényekkel rendelkeznek. A függvényeket lehet csoportosítani a szövegkezelő, numerikus, dátumkezelő és egyéb függvények szerint. Ezek a függvények rendszerenként, gyártónként különbözhetnek, szintaktikájuk eltérhet.

Tekintsük át a MySQL főbb függvényei csoportosítva.

A függvényeket tesztelhetjük adatok nélkül is. Teszteléshez, tanuláshoz ez tökéletesen megfelelő és utána már magabiztosan használhatjuk egyéb helyzetekben.

Szövegkezelő, sztringkezelő függvények

Szöveg hosszának meghatározása: CHAR_LENGTH(), CHARACTER_LENGTH(). A függvény a hossz karakterszámában adja meg. A LENGTH() függvény a hosszt byte-ban adja meg.

Szintaktika

```
CHAR_LENGTH(string)
CHARACTER_LENGTH(string)
LENGTH(string)
```

	123 Szlogen
1	21

Példa

```
1. SELECT CHAR_LENGTH("Szeretek programozni!") AS Szlogen;
```

Írassuk ki azon országok neveit, amelyek nevei 20 karakternél hosszabbak.

```
1. SELECT
2.   orszag
3. FROM
4.   orszagok
5. WHERE
6.   CHARACTER_LENGTH(ORSZAG)>20;
```

	ABC ország
1	DÉL-AFRIKAI KÖZTÁRSASÁG
2	AMERIKAI EGYESÜLT ÁLLAMOK
3	SAINT VINCENT ÉS GRENADINE
4	EGYESÜLT ARAB EMIRSÉGEK
5	ZÖLD-FOKI KÖZTÁRSASÁG
6	DOMINIKAI KÖZTÁRSASÁG
7	KÖZÉP-AFRIKAI KÖZTÁRSASÁG

Nagy-illetve kisbetűssé alakítás

A LCASE(), LOWER() függvény egy karakterláncot kisbetűvé alakítja át. A UCASE(), UPPER() függvény egy karakterláncot nagybetűvé alakítja át.

Szintaktika

1. LCASE(text)
2. UCASE(text)
3. LOWER(text)
4. UPPER(text)

Példa

1. **SELECT**
2. LCASE("Szeretek programozni!");
- 3.
4. **SELECT**
5. **DISTINCT** LOWER(penzjel)
6. **FROM**
7. országok;

	ABC LCASE("Szeretek programozni!")
1	szeretek programozni!

	ABC LOWER(penzjel)
1	eur
2	gbp
3	nok
4	sek
5	chf
6	huf
7	csd
8	bgl

1. **SELECT**
2. UCASE("Szeretek programozni!");
- 3.
4. **SELECT**
5. **DISTINCT** UPPER(penznem)
6. **FROM**
7. országok;

	ABC UCASE("Szeretek programozni!")
1	SZERETEK PROGRAMOZNI!

	ABC UPPER(penznem)
1	EURÓ
2	FONT
3	NORVÉG KORONA
4	SVÉD KORONA
5	SVÁJCI FRANK
6	FORINT
7	DINÁR
8	LEVA
9	LEI

Sztringek összefűzése

Két vagy több sztring összefűzése: CONCAT().

Szintaktika

CONCAT(expression1, expression2, expression3,...)

Példa

```
1. SELECT CONCAT("Szeretek ", " ", "programozni ", "!" )
```

Írassuk ki az országok és a fővárosok neveit a következő módon: *Magyarország (Budapest)*

```
1. SELECT
2.     CONCAT(ország, " (", fovaros, ")")
3. FROM
4.     foldrajz.orszagok ;
```

	ABC CONCAT_WS("-", ország, fovaros, penzjel, autojel) ▼
1	SPANYOLORSZÁG-MADRID-EUR-E
2	PORTUGÁLIA-LISSZABON-EUR-P
3	FRANCIAORSZÁG-PÁRIZS-EUR-F
4	NAGY-BRITANNIA-LONDON-GBP-GB
5	NORVÉGIA-OSLO-NOK-N
6	SVÉDORSZÁG-STOCKHOLM-SEK-S
7	FINNORSZÁG-HELSINKI-EUR-SF
8	NÉMETORSZÁG-BERLIN-EUR-D
9	SVÁJC-BERN-CHF-CH

A sztringek összefűzésének speciális esete, amikor egy szeparátorjellel választhatjuk el a kiírandókat. Erre a CONCAT_WS() függvényt használhatjuk.

Szintaktika

```
CONCAT_WS(separator, expression1, expression2, expression3,...)
```

Példa

```
1. SELECT CONCAT_WS("-", "2019", "04", "15") AS Dátum;
```

	ABC Dátum ▼
1	2019-04-15

Írassuk ki az országok adatait a következő formátumba: SPANYOLORSZÁG-MADRID-EUR-E !

```
1. SELECT
2.     CONCAT_WS("-", ország, fovaros, penzjel, autojel)
3. FROM
4.     foldrajz.orszagok ;
```

	ABC CONCAT_WS("-", ország, fovaros, penzjel, autojel) ▼
1	SPANYOLORSZÁG-MADRID-EUR-E
2	PORTUGÁLIA-LISSZABON-EUR-P
3	FRANCIAORSZÁG-PÁRIZS-EUR-F
4	NAGY-BRITANNIA-LONDON-GBP-GB
5	NORVÉGIA-OSLO-NOK-N
6	SVÉDORSZÁG-STOCKHOLM-SEK-S

Részképzés

LEFT(), RIGHT(), MID(), SUBSTR(), SUBSTRING() függvények az segítségével a karakterlánc egy-egy részletét kaphatjuk meg. A LEFT() baloldali rész határoz meg, a RIGHT() pedig jobboldali részt. A MID(), SUBSTR(), SUBSTRING() pedig adott kezdősorszámtól adat számú karaktersorozatot határoz meg.

Szintaktika

```
1. LEFT(string, number_of_chars)
2. RIGHT(string, number_of_chars)
```

3. MID(string, start, length)
4. SUBSTR(string, start, length)
5. **SUBSTRING**(string, start, length)

Példa

1. **SELECT LEFT**("Szeretek programozni!",8);

	ABC LEFT("Szeretek programozni!",8)
1	Szeretek

Írassuk ki azokat az országokat, ahol az ország nevének az első karaktere megegyezik az ország autójelének első karakterével!

2. **SELECT**
3. ország
4. **FROM**
5. országok
6. **WHERE**
7. **LEFT**(ország,1)= **LEFT**(autojel,1);

	ABC ország
1	PORTUGÁLIA
2	FRANCIAORSZÁG
3	NORVÉGIA
4	SVÉDORSZÁG
5	AUSZTRIA
6	SZERBIA
7	BULGÁRIA
8	ROMÁNIA

1. **SELECT SUBSTRING**("Szeretek programozni!", 14, 3);

	ABC SUBSTRING("Szeretek programozni!", 14, 3)
1	ram

Írassuk azon országokat, amelyek nevében a 6. helytől szerepel a „IA” karaktersorozat!

1. **SELECT**
2. ország
3. **FROM**
4. országok
5. **WHERE**
6. MID(ország, 6, 2)= 'IA';

	ABC ország
1	FRANCIAORSZÁG
2	SZERBIA
3	ROMÁNIA
4	ALBÁNIA
5	ALGÉRIA
6	TUNÉZIA
7	ETIÓPIA
8	BOLÍVIA
9	NAMÍBIA

Sztring kiegészítése, feltöltése adott karakterekkel

Az LPAD() függvény egy karakterláncot egy másik karaktersorozattal balról feltöltött adott hosszúságra kiegészítve. A RPAD() függvény ugyanezt csinálja csak jobb oldalról.

Szintaktika

1. LPAD(string, length, lpad_string)
2. RPAD(string, length, rpad_string)

Példa

- ```
1. SELECT LPAD("Szeretek programozni!",50,"123");
```

[illegible]

- ```
1. SELECT RPAD("Szeretek programozni!",50,"123");
```

[illegible]

- ```
1. SELECT
2. LPAD(orszag, 50, " ")
3. FROM
4. orszagok;
```

```
lpad(ország,50," ")

SPANYOLORSZ G
PORTUG LIA
FRANCIAORSZ G
NAGY-BRITANNIA
NORV GIA
SV DORSZ G
FINNORSZ G
N METORSZ G
SV JC
AUSZTRIA
OLASZORSZ G
MAGYARORSZ G
SZERBIA
BULG RIA
```

- ```
5. SELECT
6.     RPAD(ország, 50, " ")
7. FROM
8.     orszagok;
```

```
RPAD(orszag, 50, " ")
-----
SPANYOLORSZ G
PORTUG LIA
FRANCIAORSZ G
NAGY-BRITANNIA
NORV GIA
SV DORSZ G
FINNORSZ G
N METORSZ G
```

Formázott kiiratás

A FORMAT () függvény egy számot formáz egy adott formátumra, mint például "#, ###, ###.##". A formátumrahozáskor kerekítés történik.

Szintaktika

- FORMAT(number, decimal places)**

Példa

```
1. SELECT FORMAT(12345.6789,2);
```

	ABC FORMAT(12345.6789,2) ↑↓
1	12,345.68

Irassuk ki az országok népsűrűségét 2 tizedesjegy pontossággal.

```
1. SELECT
2.   orszag, FORMAT(nepesseg*1000 / terület, 2) AS NÉPSŰRŰSSÉG
3. FROM
4.   orszagok;
```

	ABC orszag ↑↓	ABC NÉPSŰRŰSSÉG ↑↓
1	SPANYOLORSZÁG	84.59
2	PORTUGÁLIA	108.60
3	FRANCIAORSZÁG	122.22
4	NAGY-BRITANNIA	267.16
5	NORVÉGIA	14.19
6	SVÉDORSZÁG	19.71
7	FINNORSZÁG	15.38
8	NÉMETORSZÁG	230.79

Szóközök „levágása”

TRIM(), LTRIM(), RTRIM() segítségével a szóköz karaktereket lehet el levágni a sztring elejéről, végéről, vagy mindkettő helyről.

Szintaktika

```
TRIM(string)
LTRIM(string)
RTRIM(string)
```

Példa

```
1. SELECT
2.   TRIM("      Szeretek programozni!      "),
3.   LENGTH(TRIM("      Szeretek programozni!      "));
```

	ABC TRIM(" Szeretek programozni! ") ↑↓	123 LENGTH(TRIM(" Szeretek programozni! ")) ↑↓
1	Szeretek programozni!	21

```
1. SELECT
2.   LTRIM("      Szeretek programozni!      "),
3.   LENGTH(LTRIM("      Szeretek programozni!      "));
```

	ABC LTRIM(" Szeretek programozni! ") ↑↓	123 LENGTH(LTRIM(" Szeretek programozni! ")) ↑↓
1	Szeretek programozni!	25

```
1. SELECT
2.   RTRIM("      Szeretek programozni!      "),
3.   LENGTH(RTRIM("      Szeretek programozni!      "));
```

	ABC RTRIM(" Szeretek programozni! ") ↑↓	123 LENGTH(RTRIM(" Szeretek programozni! ")) ↑↓
	Szeretek programozni!	28

Részsstring beszúrása sztringbe

Az INSERT () függvény egy karaktersorozatot szúr be a megadott pozícióba és egy bizonyos számú karakter helyett.

Szintaktika

```
INSERT(string, position, number, string2)
```

Példa

```
1. SELECT INSERT("SZERETEK PROGRAMOZNI!",10,7,"KERÉKPÁR");
```

	abc INSERT("SZERETEK PROGRAMOZNI!",10,7,"KERÉKPÁR")
1	SZERETEK KERÉKPÁROZNI!

Részsstring keresése sztringben

Az INSTR () függvény visszaadja a karakterlánc első előfordulásának helyét egy másik karakterláncban.

Szintaktika

```
1. INSTR(string1, string2)
```

Példa

```
1. SELECT INSTR("SZERETEK PROGRAMOZNI!", "RETEK");
```

Írassuk ki azokat az országokat, amelyek nevében szerepel az „ORSZÁG”!

```
1. SELECT
2.   orszag
3. FROM
4.   orszagok
5. WHERE
6.   INSTR(orszag, "ORSZÁG")>0;
```

	abc orszag
1	SPANYOLORSZÁG
2	FRANCIAORSZÁG
3	SVÉDORSZÁG
4	FINNORSZÁG
5	NÉMETORSZÁG
6	OLASZORSZÁG
7	MAGYARORSZÁG
8	CSEHORSZÁG
9	LENGYELORSZÁG
10	OROSZORSZÁG
11	GÖRÖGORSZÁG

LOCATE(), POSITION()

Szintaktika

```
POSITION(substring IN string)
```

A POSITION() függvény visszaadja az első sztring előfordulási pozícióját a második sztringben. Amennyiben nem szerepel a keresett sztring, akkor a függvény visszatérési értéke 0.

LOCATE(substring, string, start)

A LOCATE() függvény visszaadja az első sztring (substring) előfordulási pozícióját a második sztringben (string). Opcionálisan megadható a keresés kezdőpozíciója (start). Amennyiben nem szerepel a keresett sztring, akkor a függvény visszatérési értéke itt is 0.

Példa

```
1. SELECT
2.     POSITION("reték" IN "Szeretek programozni!");
3.
4. SELECT
5.     LOCATE("reték", "Szeretek programozni!", 1);
6.
7. SELECT
8.     LOCATE("r", "Szeretek programozni!", 5);
```

	123 POSITION("reték" IN "Szeretek programozni!")	
1		4

	123 LOCATE("reték", "Szeretek programozni!")	
1		4

	123 LOCATE("r", "Szeretek programozni!", 5)	
1		11

Sztringek összehasonlítása

A STRCMP() függvény két karakterláncot hasonlít össze.

Szintaktika

STRCMP(string1, string2)

A függvény visszatérési értéke

- 0, ha string1=string2
- -1, ha a string1<string2
- 1, ha a string1>string2.

Példa

```
1. SELECT STRCMP("Nagy Elemér", "Nagy Éva");
```

	123 strcmp("Nagy Elemér", "Nagy Éva")	
1		-1

Matematikai függvények

SIN(), COS(), TAN(), COT(), ASIN(), ACOS(), ATAN() függvények a matematika tanulmányok során megismert matematikai függvényeket valósítják meg. A szöveget radiánban kell

használnunk. A fok és radián közötti átváltást a DEGREES(), RADIANS() segítségével valósíthatjuk meg.

FLOOR(), ROUND(), CEIL(), CEILING(), TRUNCATE() a kerekítés különböző eseteit valósítják meg.

AVG(), MIN(), MAX(), COUNT(), SUM() függvényeket megismertük már az eddigi feladatokban.

DIV(), MOD() az egész és maradékos osztást valósítják meg.




LN(), LOG(), LOG2(), LOG10(), POW(), POWER(), EXP() függvényekkel a hatványozás és a logaritmus különböző eseteit használhatjuk.

Dátumkezelő függvények

Aktuális dátum, idő

Aktuális dátumot, időpontot a CURDATE(), CURRENT_DATE(), CURTIME(), CURRENT_TIME(), NOW(), CURRENT_TIMESTAMP() függvényekkel kérdezhetjük le.

1. **SELECT** CURDATE(),CURTIME(), **CURRENT_TIMESTAMP**();

	 CURDATE()	 CURTIME()	 CURRENT_TIMESTAMP()
1	2019-03-24	23:29:57	2019-03-24 23:29:57

Dátum, idő kiírása adott formátumban.

DATE_FORMAT(), TIME_FORMAT()

A DATE_FORMAT () függvény a megadott dátumot formázza. A formázáskor formátummintát kell megadnunk, ahol speciális karakterekkel adjuk meg a kívánt alakot. Ezek részletes leírását megtalálhatjuk az online dokumentációban.

Szintaktika

DATE_FORMAT(**date**, format)

Példa

1. **SELECT** DATE_FORMAT(NOW(), "%Y. %M %D.");

A TIME_FORMAT () függvény egy megadott időadatot formáz. A formázáskor itt is speciális karakterekkel megadott mintát kell megadni.

Szintaktika

TIME_FORMAT(**time**, format)

Példa

1. **SELECT** TIME_FORMAT(CURRENT_TIME(), "%H óra %i perc %s mp");

	<code>ABC time_format(current_time(), "%H óra %i perc %s mp")</code>
1	22 óra 17 perc 21 mp

Két dátum különbsége

A DATEDIFF() függvény megadja a két dátum között eltelt napok számát.

Szintaktika

```
DATEDIFF(date1, date2)
```

Példa

Írassuk ki hány naposak vagyunk ma!

```
1. SELECT DATEDIFF(NOW(), "2000-02-04");
```

Adott egy aru(id, aruNev, gyartasiIdo, szavatossagiIdo). Írassuk azon áruknak a nevét, amelynek lejárt a szavatossága!

```
1. SELECT
2.     aruNev
3. FROM
4.     aru
5. WHERE
6.     DATEDIFF(NOW(), gyartasiIdo) >= szavatossagiIdo;
```

Dátum, idő összeadás, kivonás

DATE_ADD(), DATE_SUB(), SUBDATE(), SUBTIME()

A DATE_ADD () függvény idő/dátum intervallumot ad hozzá egy dátumhoz, majd visszaadja a keletkezett értéket. Az értékek az EXTRACT() függvényénél részletezett esetek lehetnek.

Szintaktika

```
DATE_ADD(date, INTERVAL value addunit)
```

Példa

```
1. SELECT DATE_ADD("2019-03-25", INTERVAL 14 DAY);
```

	<code>DATE_ADD("2019-03-25", INTERVAL 14 DAY)</code>
1	2019-04-08

A DATE_SUB() függvény egy dátumból kivonja az idő/dátum intervallumot, majd visszaadja a keletkezett értéket. SUBDATE() hasonlóan működik.

Szintaktika

```
1. DATE_SUB(date, INTERVAL value interval)
```

Példa

```
1. SELECT DATE_SUB("2019-01-05", INTERVAL 2 WEEK);
```

	DATE_SUB("2019-01-05", INTERVAL 2 WEEK)
1	2018-12-22

A SUBTIME() függvény egy adott idő/dátumból kivon egy időintervallumot, majd visszaadja az keletkezett időt/dátumot.

Szintaktika

```
1. SUBTIME(datetime, time_interval)
```

Példa

```
1. SELECT SUBTIME("2019-03-25 3:3:33.000003", "5:5:5.000001");
```

	SUBTIME("2019-03-25 3:3:33.000003", "5:5:5.000001")
1	2019-03-24 21:58:28.000002

Dátumhoz tartozó tulajdonságok meghatározása

A DAYNAME () függvény egy adott dátumra viszi vissza a hétköznapi angol nevét.

A DAYOFMONTH () függvény egy adott napon adja meg a hónap napját, megegyezik a DAY() függvénnyel.

A DAYOFWEEK () függvény egy adott dátumra visszaadja a héten belüli sorszámát: 1-hétfő, 2-kedd...

A DAYOFYEAR () függvény egy adott dátumból meghatározza, hogy az év hányadik napja.

A MONTHNAME () függvény egy adott dátumra visszaadja a hónap angol nevét.

Szintaktika

```
1. DAYNAME(date)
2. DAYOFMONTH(date)
3. DAYOFWEEK(date)
4. DAYOFYEAR(date)
5. MONTHNAME(date)
```

Példa

```
1. SELECT DAYNAME("2019-12-25");
2. SELECT DAYOFMONTH("2019-12-25");
3. SELECT DAYOFWEEK("2019-12-25");
4. SELECT DAYOFYEAR("2019-12-25");
5. SELECT MONTHNAME("2019-12-25");
```

	DAYNAME("2019-12-25")	DAYOFMONTH("2019-12-25")	DAYOFWEEK("2019-12-25")	DAYOFYEAR("2019-12-25")
1	Wednesday	25	4	359

Dátum, idő felbontása részinformációra

DAY(), MONTH(), YEAR(), HOUR(), MINUTE(), SECOND() függvények egy adott időpillanattól add vissza megfelelő részt.

Szintaktika

1. **DAY**(date)
2. **MONTH**(date)
3. **YEAR**(date)
4. **SECOND**(date)
5. **MINUTE**(date)
6. **HOURL**(date)

Példa

1. **SELECT DAY**(NOW()), **MONTH**(NOW()), **YEAR**(NOW());
2. **SELECT SECOND**(NOW()), **MINUTE**(NOW()), **HOURL**(NOW());

	123 DAY(NOW())	123 MONTH(NOW())	123 YEAR(NOW())		123 SECOND(NOW())	123 MINUTE(NOW())	123 HOUR(NOW())
1	25	3	2 019	1	45	39	23

Az **EXTRACT ()** függvény egy adott dátumból add vissza egy kívánt részt.

Szintaktika

EXTRACT(part **FROM** date)

A rész lehetséges értékei: MICROSECOND, SECOND, MINUTE, HOUR, DAY, WEEK, MONTH, QUARTER, YEAR, SECOND_MICROSECOND, MINUTE_MICROSECOND, MINUTE_SECOND, HOUR_MICROSECOND, HOUR_SECOND, HOUR_MINUTE, DAY_MICROSECOND, DAY_SECOND, DAY_MINUTE, DAY_HOUR, YEAR_MONTH. Ezeket értelemszerűen alkalmazhatjuk.

Példa

1. **SELECT EXTRACT**(WEEK **FROM** NOW());

	123 EXTRACT(WEEK FROM NOW())
1	12

Egyéb függvények

Konvertálás

A **CAST()** függvény egy tetszőleges típusú értéket átalakít a megadott adattípusra. A megadható adattípusok: DATE, DATETIME, TIME, CHAR, SIGNED ,UNSIGNED, BINARY. Hasonlóan működik a **CONVERT()** függvény is.

Szintaktika

CAST(value **AS** datatype)
CONVERT(value, type)

Példa

1. **SELECT CONVERT**("123", SIGNED)*2;

	123 CONVERT("123", SIGNED)*2
1	246

Feltételek

Az IF() függvény egy adott értéket ad vissza, ha egy feltétel igaz (true), vagy másik érték, ha egy feltétel hamis (false).

Szintaktika

```
IF(condition, value_if_true, value_if_false)
```

Példa

Írassuk ki az országok neveit és mellette az hogy „SOK”, ha a népsűrűsége 100 fő/km² felett van, illetve „KEVÉS”-t, ha nem.

```
1. SELECT
2.     ország,
3.     IF(nepesseg*1000 / terület > 100, "SOK", "KEVÉS")
4. FROM
5.     orszagok;
```

	ABC ország	ABC IF(nepesseg*1000/terulet>100,"SOK","KEVÉS")
1	SPANYOLORSZÁG	KEVÉS
2	PORTUGÁLIA	SOK
3	FRANCIAORSZÁG	SOK
4	NAGY-BRITANNIA	SOK
5	NORVÉGIA	KEVÉS
6	SVÉDORSZÁG	KEVÉS
7	FINNORSZÁG	KEVÉS
8	NÉMETORSZÁG	SOK

A CASE() függvénnyel a többágú kimenet valósítható meg. Amikor egy feltétel teljesül a WHEN utasítások közül, akkor a kiértékelés befejeződik. Ha egyik WHEN sem lesz igaz, akkor az ELSE ág értéke adódik vissza.

Szintaktika

```
1. CASE
2.     WHEN feltetel1 THEN eredmény1
3.     WHEN feltetel2 THEN eredmény2
4.     WHEN feltetelN THEN eredményN
5.     ELSE eredmény
6. END;
```

Példa

Írassuk ki több kategóriával az országok népsűrűségét!

```
1. SELECT
2.     ország,
3.     CASE
4.         WHEN nepesseg*1000 / terület>100 THEN "SOK"
5.         WHEN nepesseg*1000 / terület>50 THEN "ÁTLAGOS"
6.         WHEN nepesseg*1000 / terület>10 THEN "KEVÉS"
7.         ELSE "NAGYON RITKÁN LAKOTT"
8.     END
```

9. FROM

10. országok;

	asc ország	asc CASE WHEN nepesseg*100
6	SVÉDORSZÁG	KEVÉS
7	FINNORSZÁG	KEVÉS
8	NÉMETORSZÁG	SOK
9	SVÁJC	SOK
10	AUSZTRIA	ÁTLAGOS
11	OLASZORSZÁG	SOK
12	MAGYARORSZÁG	SOK
13	SZERBIA	SOK
14	BULGÁRIA	ÁTLAGOS
15	ROMÁNIA	ÁTLAGOS
16	SZLOVÁKIA	SOK
17	CSEHORSZÁG	SOK
18	LENGVELORSZÁG	SOK
19	OROSZORSZÁG	NAGYON RITKÁN LAKOTT
20	GÖRÖGORSZÁG	ÁTLAGOS
21	TÖRÖKORSZÁG	ÁTLAGOS
22	DÁNIA	SOK
23	IZLAND	NAGYON RITKÁN LAKOTT