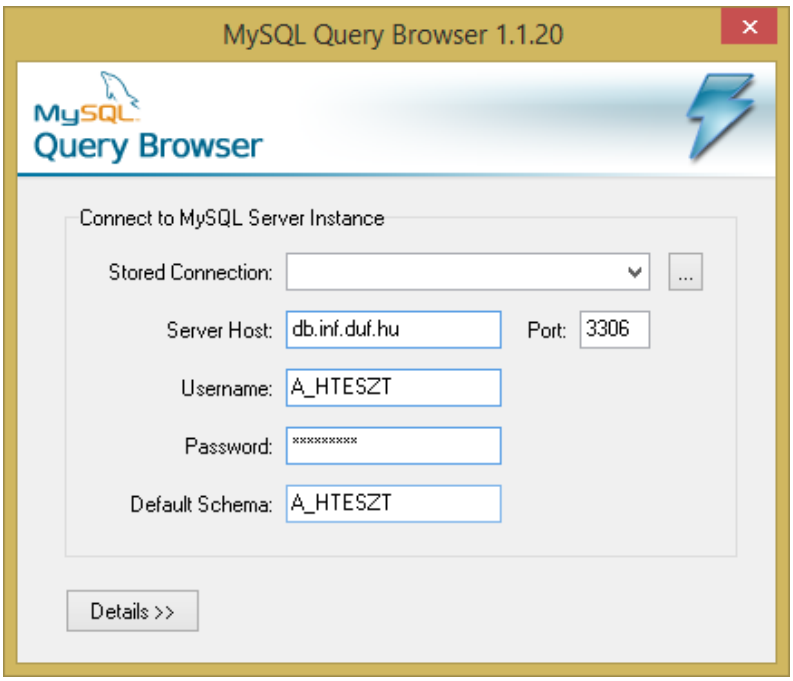
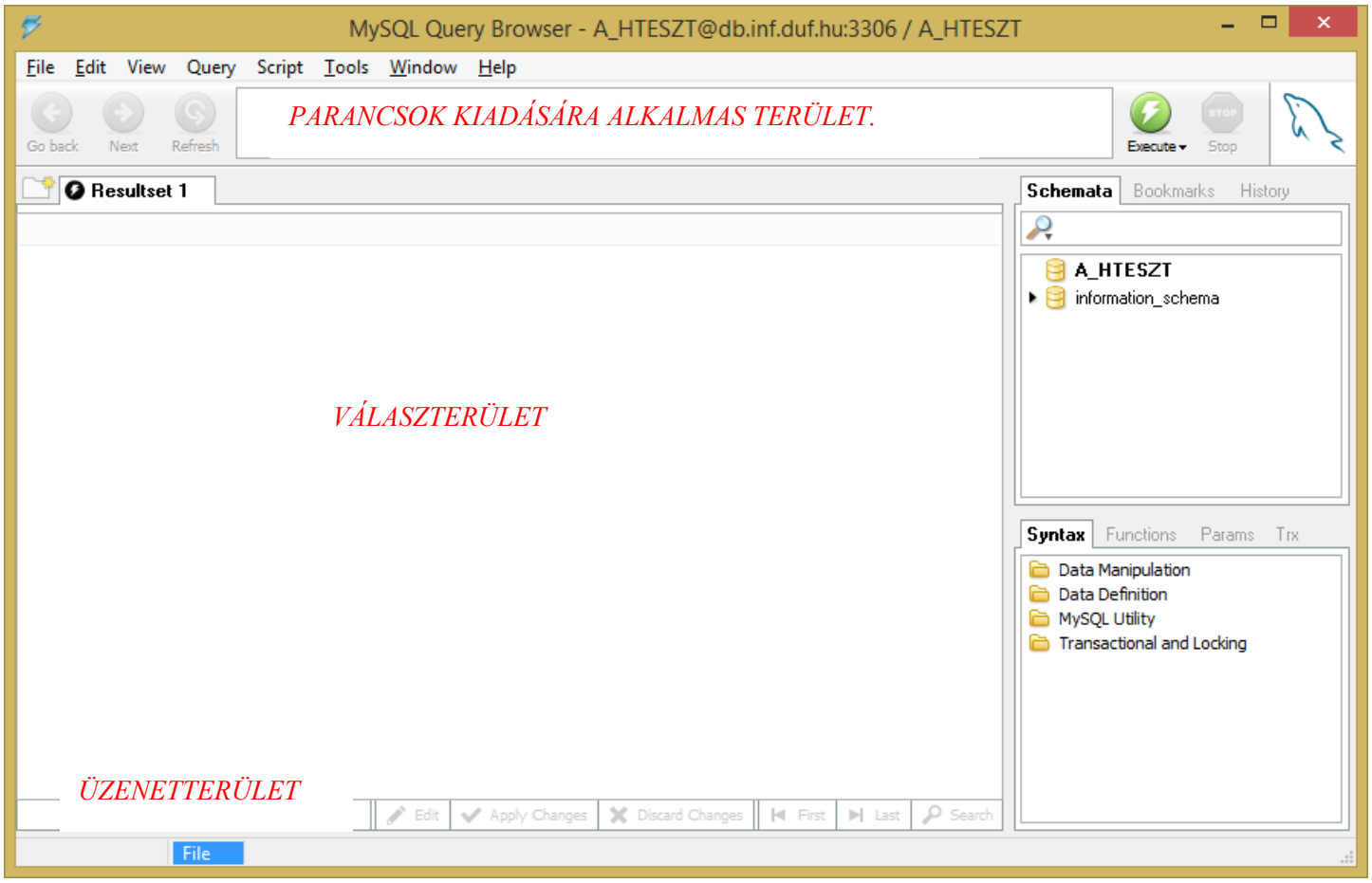
1. Ismerkedés a MySQL Query Browser-rel.

Belépés





Általános SQL felhasználói területet biztosító program (MySQL Query Browser)

2. SQL bevezetés.

MySQL mezőtípusok:

Egész számok:

Típus	Tárolási méret (byte)	Minimum érték	Maximum érték
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32768	32767
		0	65535
MEDIUMINT	3	-8388608	8388607
		0	16777215
INT vagy INTEGER	4	-2147483648	2147483647
		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615

Előjeles és előjel nélküli változataik is vannak. Alapértelmezett mindig az előjeles!

(Előjel nélkül pl.: TINYINT UNSIGNED)

Lebegőpontos valós számértékű mezőtípusok:

Típus	Tárolási méret (byte)	Megjegyzés
FLOAT(M,D) vagy REAL(M,D)	4	Egyszeres pontosságú lebegőpontos szám.
		M: összes számjegy pl.: FLOAT(7,4) D: tizedesjegyek száma 12.3456
DOUBLE (M,D) vagy DOUBLE PRECISION(M,D)	8	Dupla pontosságú lebegőpontos szám. M: összes számjegy D: tizedesjegyek száma

Fixpontos valós számértékű mezőtípusok:

Típus	Tárolási méret (byte)	Megjegyzés
DECIMAL(M,D) vagy DEC(M,D)	***Minden helyiérték 1 byte***	M: összes számjegy D: tizedesjegyek száma A pontosság is fontos! Előjeles, de elláthatjuk UNSIGNED módosítóval! Pl.: Pénzösszegek esetében DECIMAL – nem fog kerekíteni!
NUMERIC(M,D)		

Dátum idő típusok:

Típus	Tárolási méret (byte)	Megjegyzés
DATE	3	ÉÉÉÉ_HH_NN '1000-01-01' *-tól* '9999-12-31' *-ig*
TIME	3	ÓÓ:PP:MM
DATETIME	8	ÉÉÉÉ-HH-NN ÓÓ:PP:MM '1000-01-01 00:00:00' *-tól* '9999-12-31 23:59:59' *-ig*
TIMESTAMP	4	ÉÉÉÉ-HH-NN-ÓÓ:PP:MM (14 jegyű) ÉÉ-HH-NN-ÓÓ:PP:MM (12 jegyű) ÉÉÉÉ-HH-NN (8 jegyű) ÉÉ-HH-NN (6 jegyű) '1970-01-01 00:00:01' *-tól* '2038-01-19 03:14:07' *-ig*
YEAR	1	1901-2155 (4 jegyű) vagy (19)70-(20)69 (2 jegyű)

Szöveges és bináris típusok:

Típus	Tárolási méret (byte)	Megjegyzés
CHAR	0..255	Meghatározott hosszúságú. A fennmaradó terület jobbról szóközökkel feltöltve.
VARCHAR	0..255	Változó hosszúságú.

Pl. latin1 vagy latin2 kódolás esetében:

Érték	CHAR(4)	Tárolási méret (byte)	VARCHAR(4)	Tárolási méret (byte)
' '	' '	4	' '	1
'ab'	'ab '	4	'ab'	3
'abc'	'abc '	4	'abc'	4
'abcd'	'abcd'	4	'abcd'	5
'abcdf'	'abcd'	4	'abcd'	5

Típus	Megjegyzés
BINARY	Bináris tartalom tárolása fix. hosszúságon.
VARBINARY	Bináris tartalom tárolása változó hosszúságon.
BLOB	Max. 65535 byte - case sensitive, Pl.: Képek tárolása.
TEXT	Változó hosszúságú. Max. 65535 karakter - nem case sensitive
ENUM	Felsorolás típus.
SET	Felsorolás típus.

3. Relációk (táblák) létrehozása, módosítása, megszüntetése SQL-ben

Tábla létrehozása:

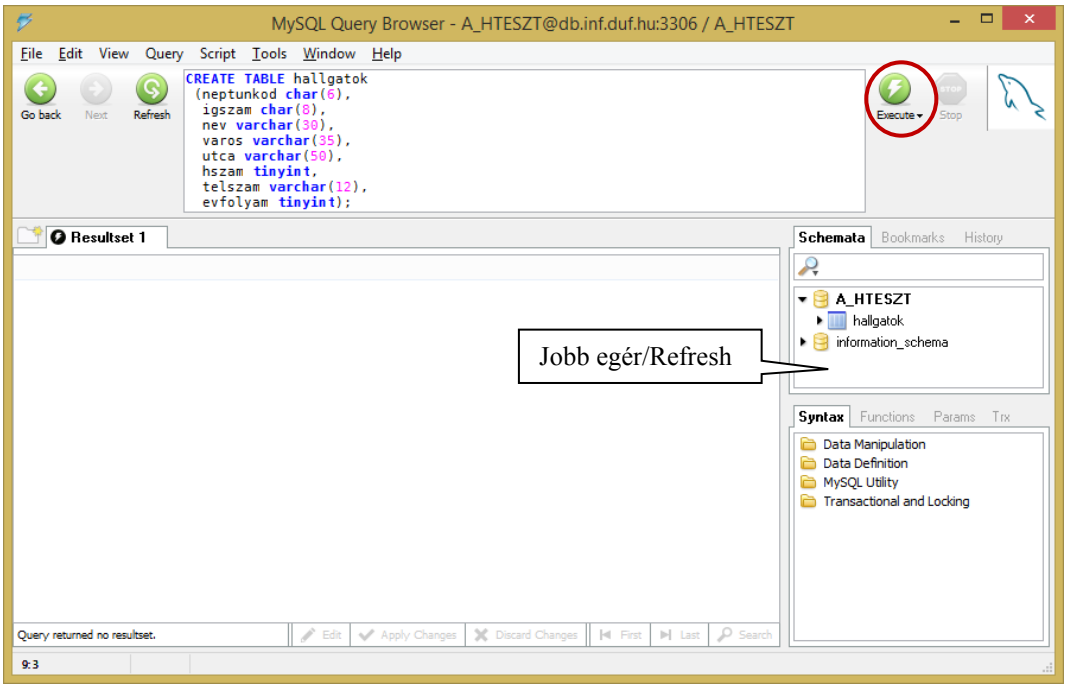
Az SQL táblalétrehozó utasítás általános alakja:

CREATE TABLE táblanév (*attribútumnév attribútumtípus*, …);

Pl.: ***hallgatok (neptunkod, igszam, nev, varos, utca, hszam, telszam, evfolyam)***

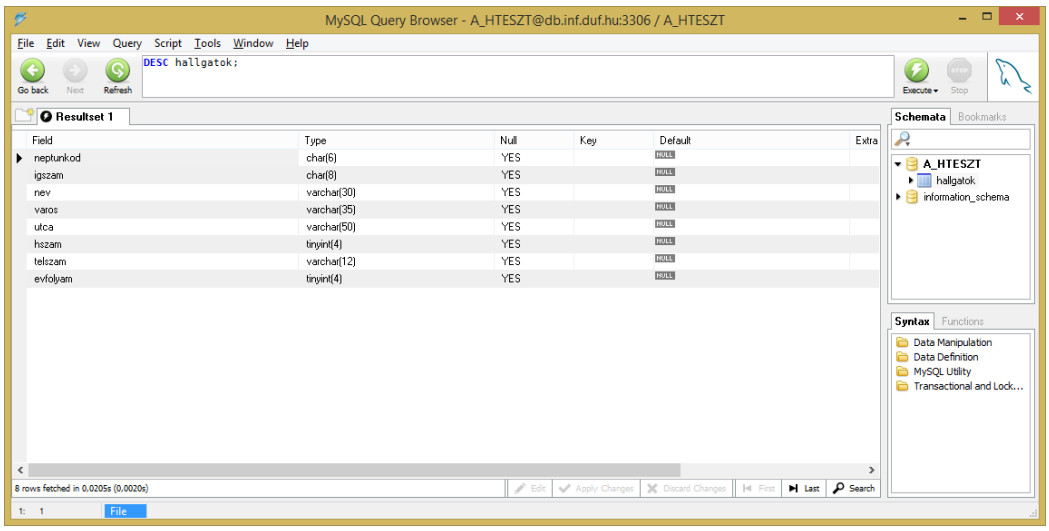
```
CREATE TABLE hallgatok
  (neptunkod char(6),
   igszam char(8),
   nev varchar(30),
   varos varchar(35),
   utca varchar(50),
   hszam tinyint,
   telszam varchar(12),
   evfolyam tinyint);
```

Tábladefiníció



Tábla szerkezetének megtekintése:

DESC hallgatok; vagy DESCRIBE hallgatok;



Alapértelmezés beállítása:

Default – NULL jelentése: Speciális érték hiányát jelképező érték. Nem a nulla szám! Az adatbázisrendszer számára ismeretlen az attribútum értéke.

Pl.: Hozzunk létre egy **h** nevű táblát, amelynek mezői ***nkod***, ***nev*** és ***evfoly***. Az evfoly mezőre állítsunk be alapértelmezést. Ha a felhasználó nem tölti ki ezt a mezőt, vagy rossz értéket ad meg (mondjuk nem 1-4 közötti számot), akkor az 1 évfolyamot írja be a rendszer.

```
CREATE TABLE h (nkod char(6), nev varchar(20), evfoly tinyint DEFAULT 1);
```

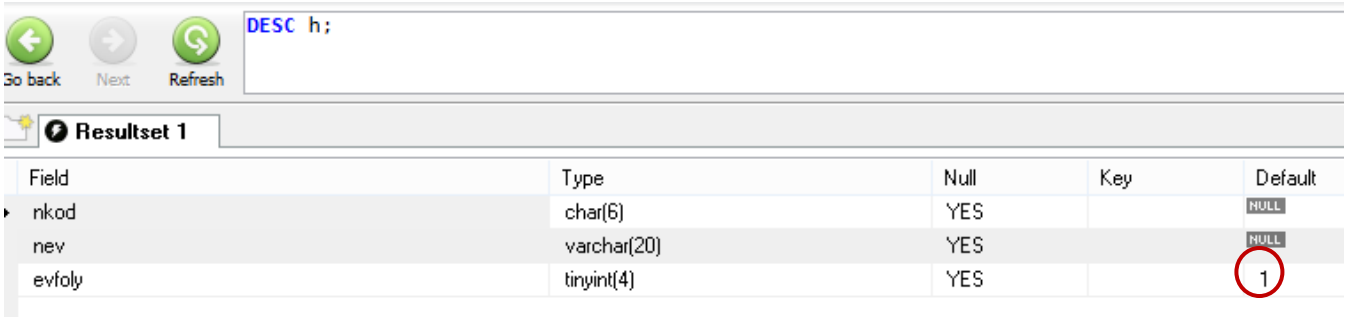
vagy egyéb megszorításokkal:

```
CREATE TABLE h (nkod char(6), nev varchar(20), evfoly tinyint DEFAULT 1 CHECK(evfoly>0 AND evfoly<=4));
```

DEFAULT *érték* : alapértelmezett érték megadása.

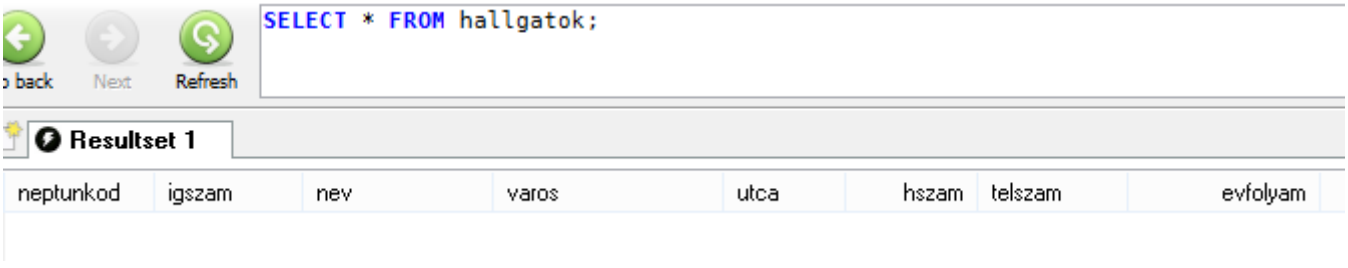
CHECK *feltétel* : a mező értékének a feltételben megadott megszorításnak kell megfelelnie.

Ha a CHECK után megadott feltétel ellenőrzése nem működik, akkor a megszorítás megadására triggert kell írni! (Lásd. később!)



Field	Type	Null	Key	Default
nkod	char(6)	YES		NULL
nev	varchar(20)	YES		NULL
evfoly	tinyint(4)	YES		1

Táblában szereplő sorok megjelenítése: **SELECT** * **FROM** hallgatok;



Táblában szereplő bizonyos oszlopok megjelenítése:

SELECT neptunkod, nev **FROM** hallgatok;

SELECT igszam, nev, evfolyam **FROM** hallgatok;

Tábla szerkezetének módosítása:

Új mező felvétele: **ALTER TABLE** táblanév **ADD** *attribútumnév típus*;

Utolsó mezőként:

```
ALTER TABLE hallgatok ADD anyja_neve varchar(35);
```

vagy

```
ALTER TABLE hallgatok ADD COLUMN anyja_neve varchar(35);
```

Meghatározott helyre:

- *igszam* mező után kerüljön:

```
ALTER TABLE hallgatok ADD  anyja_neve varchar(35) AFTER igszam;
```

- Első mezőként:

```
ALTER TABLE hallgatok ADD anyja_neve varchar(35) FIRST;
```

Mező törlése: **ALTER TABLE** táblanév **DROP** *attribútumnév*;

```
ALTER TABLE hallgatok DROP telszam;
```

vagy

```
ALTER TABLE hallgatok DROP COLUMN telszam;
```

Mező adattípusának módosítása: **ALTER TABLE** táblanév **MODIFY** *attribútumnév típus*;

Pl.: A **nev** mezőt *varchar(30)*-ról *varchar(50)*-re változtassuk:

```
ALTER TABLE hallgatok MODIFY nev varchar(50);
```

vagy

```
ALTER TABLE hallgatok MODIFY COLUMN nev varchar(50);
```

Mező átnevezése: **ALTER TABLE** táblanév **CHANGE** régi_*attr_név* *új_attr_név* *típus*;

```
ALTER TABLE hallgatok CHANGE neptunkod Neptunkod char(6);
```

vagy

```
ALTER TABLE hallgatok CHANGE COLUMN neptunkod  Neptunkod char(6);
```

Tábla átnevezése: **ALTER TABLE** régi_táblanév **RENAME TO** új_táblanév**;**

```
ALTER TABLE hallgatok RENAME TO hallgato;
```

Majd nevezzük vissza!

Tábla törlése: **DROP TABLE** táblanév;

```
DROP TABLE h;
```

Tábla feltöltése adatokkal (sorok/rekordok felvitele):

INSERT INTO táblanév (A_1, …,A_n) **VALUES** (v_1,..,v_n);

```
INSERT INTO hallgatok VALUES(
  'AAA111','123456AA','Kis Ede','Dunaújváros','Kossuth utca',10,1,'Nagy Anna');
```

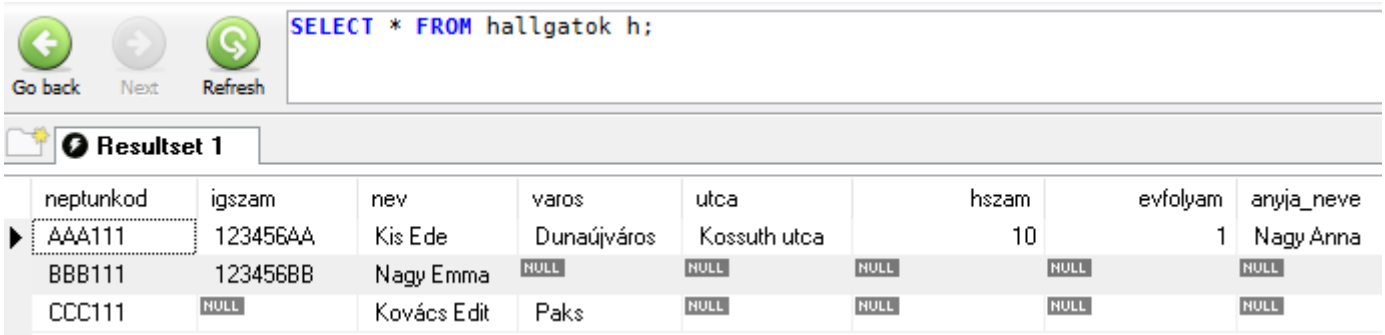
A mezők megadásának a sorrendje fontos!

```
INSERT INTO hallgatok VALUES(
  'AAA111','123456AA','Kis Ede','Dunaújváros','Kossuth utca',10,'Nagy Anna',1);
```

Részben kitöltött sorok:

```
INSERT INTO hallgatok (Neptunkod,igszam,nev) VALUES(
  'BBB111','123456BB','Nagy Emma');

INSERT INTO hallgatok (neptunkod,nev,varos) VALUES(
  'CCC111','Kovács Edit','Paks');
```



neptunkod	igszam	nev	varos	utca	hszam	evfolyam	anyja_neve
AAA111	123456AA	Kis Ede	Dunaújváros	Kossuth utca	10	1	Nagy Anna
BBB111	123456BB	Nagy Emma	NULL	NULL	NULL	NULL	NULL
CCC111	NULL	Kovács Edit	Paks	NULL	NULL	NULL	NULL

Részben kitöltött sorok későbbi kiegészítése:

Vigyázz!

```
UPDATE hallgatok SET varos='Dunaújváros', utca='Kossuth utca',hszam=12;
```

Helyette:

```
UPDATE hallgatok SET varos='Dunaújváros', utca='Kossuth utca', hszam=12
WHERE neptunkod='BBB111';
```

Azonos sorok felvitele – ÜTKÖZÉS!?

Töröljük a hallgatok táblát!

Hallgatok tábla ismételt létrehozása (ütközések kezelése):

```
CREATE TABLE hallgatok
  (neptunkod char(6) PRIMARY KEY,
   igszam char(8),
   nev varchar(30),
   varos varchar(35),
   utca varchar(50),
   hszam tinyint,
   telszam varchar(12),
   evfolyam tinyint DEFAULT 1 CHECK (evfolyam>0 AND evfolyam<=4));
```

Más szintaktikával:

```
CREATE TABLE hallgatok(
   neptunkod char(6),
   igszam  char(8),
   nev varchar(30),
   varos varchar(35),
   utca varchar(50),
   hszam tinyint,
   telszam varchar(12),
   evfolyam tinyint   DEFAULT 1,
   PRIMARY KEY(neptunkod),
   CHECK (evfolyam>0 AND evfolyam<=4)
 );
```

Logikai Operátorok

NOT, AND, OR (Ez a precedencia sorrend is, NOT a legerősebb)

Ha a kulcs több attribútumból áll (összetett kulcs), akkor azt csak a tábladefiníció végén lehet megadni!

Pl.:

```
CREATE TABLE autok(
  rendszam char(15) UNIQUE,
  tipus varchar(255),
  motorszam char(56),
  alvazszam char(31),
  evjarat year,
  szin char(30),
  PRIMARY KEY(tipus,motorszam,alvazszam));
```

Mezőkre vonatkozó megszorítások:

- NOT NULL : A mező nem tartalmazhat nullértéket, azaz kötelező kitölteni.
  Pl.: • nev varchar(30) NOT NULL
- UNIQUE : Nem lehet azonos értéket megadni egy adott mezőnél az egyes rekordok esetében, de lehet nullértéke.
  Pl.: • igszam char(8) UNIQUE
- CHECK : A mező értékének egy bizonyos feltételnek meg kell felelnie. CHECK *feltétel*
- PRIMARY KEY : Elsődleges kulcs megadása. Mindig kitöltöttnek kell lennie!
- FOREIGN KEY : Idegen kulcs megadása. (Idegen kulcs, amely egy másik táblára mutat.)
- DEFAULT : Alapértelmezett érték megadása. DEFAULT **érték**
  Pl.: evfolyam tinyint DEFAULT 1

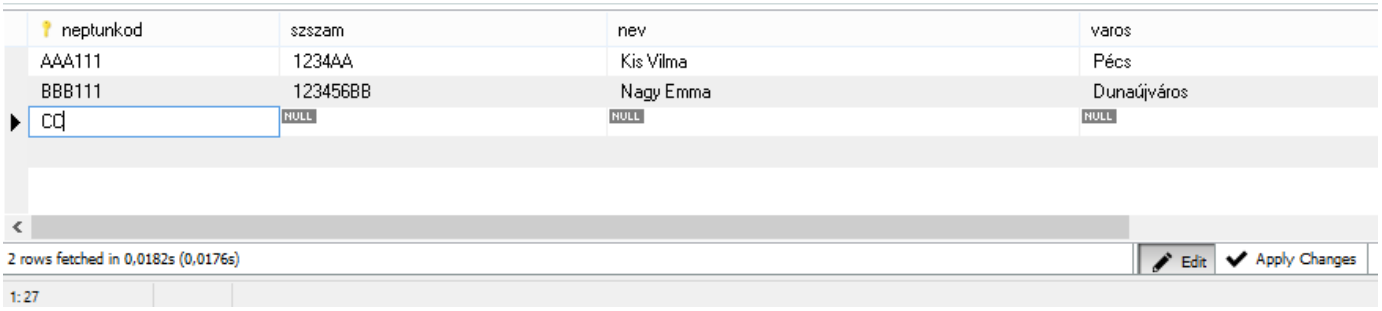
Ha a példában szereplő évfolyam mezőbe nem a feltételnek eleget tevő számot írunk, akkor az alapértelmezett értéket fogja értékül kapni!

Töltsük fel adatokkal a hallgatok táblát (minimum 10 sorral)!

Pl.:

```
INSERT INTO hallgatok VALUES('AAA111','1234AA','Kis Vilma','Pécs','Barbakán',1,'72111222',3);
```

vagy egy tábla listázás után:



neptunkod	szszam	nev	varos
AAA111	1234AA	Kis Vilma	Pécs
BBB111	123456BB	Nagy Emma	Dunaújváros
CC	NULL	NULL	NULL

2 rows fetched in 0,0182s (0,0176s) — Edit — Apply Changes

1: 27

Adatfelvitel után ne feledjük az ***Apply Changes*** *gombot!*

Adatok módosítása:

UPDATE táblanév **SET** *értékadás* **WHERE** *feltétel*;

Pl.:

```
UPDATE hallgatok SET igszam='123456XX' WHERE igszam='123456AA';

UPDATE hallgatok SET nev='Kis Vilmos' WHERE nev='Kis Vilma';

UPDATE hallgatok SET evfolyam=1 WHERE nev='Kis Vilmos';
```

WHERE záradék nélkül az összes sorra érvényes a módosítás!

Egyszerre több mező is módosítható:

```
UPDATE hallgatok SET evfolyam=2, varos='Dunaújváros' WHERE nev='Kis Vilmos';
```

A *hallgatok* táblában tárolt *telszam* mező értékét változtassuk meg úgy, hogy 36-tal kezdődjön, de csak olyan telefonszámokra, amelyek még nem így kezdődtek:

```
UPDATE hallgatok SET telszam=CONCAT('36',telszam) WHERE telszam NOT LIKE '36%';
```

Összehasonlító operátorok

=
<>
<, >, <=, >= (szövegre is működik ABC sorrend alapján)
IS NULL Nullérték esetén teljesül.
IS NOT NULL Kitöltött érték esetén teljesül.
LIKE Egy adott maszkhoz való illeszkedést vizsgál.
NOT LIKE Maszktól való eltérést vizsgál.

A LIKE maszkban használható helyettesítő karakterek:
_ 1 db tetszőleges karaktert helyettesít.
% Tetszőleges számú tetszőleges karaktert helyettesít.

Sorok törlése:

DELETE FROM táblanév **WHERE** *feltétel*;

```
DELETE FROM hallgatok WHERE nev='Kis Vilmos';
```

Vigyázz!

WHERE záradék nélkül az összes sorra érvényes a törlés!

```
DELETE FROM hallgatok;
```

Gyakorló feladatok

1. Feladat

Egy vállalat dolgozóit szeretnénk tárolni egy adattáblában. Szükség van a dolgozó személyi igazolvány számára, nevére, címére, fizetésére, születési dátumára, és a nemére.

DOLGOZOK (igszam, nev, varos, utca_hsz, fizetes, szulido, neme)

a) *A mezőkre a következők érvényesek:*

- **igszam**: karakteres, fix 8 hosszú, elsődleges kulcs
- **nev**: karakteres, változó hossz, max 30 karakter, kötelező kitölteni
- **varos**: karakteres, változó hossz, max 20 karakter
- **utca_hsz**: karakteres, változó hossz, max 30 karakter
- **fizetes**: egész
- **szulido**: dátum típus
- **neme**: karakteres fix 1 hosszú, értéke ‘F’ vagy ‘N’ lehet, alapértelmezett értéke F

b) *Bővítsük az adattáblát a születési hely, az anyja neve és a telefonszám mezőkkel:*

- **szhely**: karakteres, változó hossz, max 20 karakter
- **anyja_neve**: karakteres, változó hossz, max 25 karakter
- **telszam**: karakteres, változó hossz, max 12 karakter

c) *Módosítsuk a varos mező típusát karakteres, változó hossz, max 30 karakter-re.*

d) *Módosítsuk a fizetés mező alapértelmezett értékét 100 000 –re.*

e) *Módosítsuk a tábla nevét Dolgozok-ra.*

f) *Töltsük fel az adattáblát minimum 10 rekorddal úgy, hogy a további feladatok elvégezhetőek legyenek!*

g) *Módosítsuk a nők fizetését egységesen 200 000 Ft-ra.*

h) *A férfi dolgozók kapjanak egységesen 50 000 Ft-os béremelést.*

i) *Azok a férfi dolgozók, akiknek a fizetése nem éri el a 180 000 Ft-t, kapjanak 10 000 Ft-os béremelést.*

j) *Akik D betűvel kezdődő városban élnek, azok fizetése legyen 150 000 Ft.*

k) *Növelje 15%-kal azoknak a fizetését, akiké nem éri el 160 000 Ft-ot!*

2. Feladat

Egy háziorvos tárolni szeretné betegeinek legfontosabb adatait egy táblában.

BETEG (tajszam, nev, anyja_neve, varos, utca_hsz, szhely, szulido)

a) *A mezőkre a következők érvényesek:*

- **tajszam**: egész, elsődleges kulcs
- **nev**: karakteres, változó hossz, max 30 karakter, kötelező kitölteni
- **anyja_neve**: karakteres, változó hossz, max 30 karakter, kötelező kitölteni
- **varos**: karakteres, változó hossz, max 25 karakter
- **utca_hsz**: karakteres, változó hossz, max 30 karakter
- **szhely**: karakteres, változó hossz, max 35 karakter
- **szido**: dátum típus

b) *Bővítsük az adattáblát a telefonszám és a gyógyszerérzékenység mezőkkel:*

- **telszam**: karakteres, változó hossz, max 12 karakter
- **gye**: karakteres fix 1 hosszú, értéke 'I' vagy 'N' lehet, alapértelmezett értéke N

c) *Módosítsuk a varos mező típusát karakteres, változó hossz, max 30 karakter-re.*

d) *Módosítsuk a tábla nevét Beteg-re.*

e) *Töltsük fel az adattáblát minimum 10 rekorddal úgy, hogy a további feladatok elvégezhetőek legyenek!*

f) *Módosítsuk a varos mezők értékét egységesen Dunaújvárosra.*

g) *Töröljük az utca_hsz mezőt!*