

SQL DDL, duomenų bazės sukūrimas ir apribojimai

SQL DDL

- **Duomenų struktūrų aprašymo kalba** (*Data Definition Language*)

SQL DDL

- SQL sakinį sudaro *rezervuoti žodžiai* ir *vartotojo apibrėžti žodžiai*.
- Rezervuoti žodžiai: pastovi SQL dalis. Jie turi būti parašyti tiksliai. Perkėlimai į kitą eilutę negalimi.
- Vartotojo sukurti žodžiai: tai įvairių DB objektų vardai, pvz., lentelių, stulpelių, virtualių lentelių [views].

SQL DDL

- Daugelis SQL sakinio komponentų nepriklauso nuo raidžių registro (*case insensitive*)
- Naudojant išplėtotą BNF žymenų formą:
 - - REZERVINIAI ŽODŽIAI - didžiosios raidės
 - - vartotojo sukurti žodžiai - mažosios raidės
 - - | ženklų atskiriamos alternatyvos
 - - {} skliaustai skirti būtiniams elementams
 - - [] skliaustai skirti nebūtiniams elementams
 - - ... nurodo nebūtinus pasikartojimus

SQL DDL

```
CREATE TABLE TableName  
{(columnName dataType [NOT NULL] [UNIQUE]  
[DEFAULT defaultOption][,...]}  
[PRIMARY KEY (listOfColumns),]  
{[UNIQUE (listOfColumns),] [...,]}  
{[FOREIGN KEY (listOfFKColumns)  
REFERENCES ParentTableName [(listOfCKColumns)],  
[ON UPDATE referentialAction]  
[ON DELETE referentialAction ]] [...]}  
[,...]}
```

SQL Literalai

- Literalai - tai konstantos SQL sakiniuose
- Ne skaičių literalai pateikiami kabutėse (pvz., 'Londonas').
- Skaičių literalai pateikiami be kabučių (pvz., 650.00).

DB sukūrimas

Reliacinėse DBVS, duomenų bazę sudaro šie baziniai objektai:

- DB lentelės (angl. tables);
- Virtualios lentelės (angl. views);
- Trigeriai (angl. triggers);
- Serverio procedūros (angl. stored procedures);
- Serverio funkcijos (angl. functions);
- Indeksai (angl. index).

DB sukūrimas

```
CREATE DATABASE [IF NOT EXISTS] db_vardas;
```

db_vardas - jūsų kuriamos duomenų bazės vardas, rekomenduojam, kad tai būtų prasmingas pavadinimas. DBVS negali kartotis duomenų bazių vardai.

```
CREATE DATABASE autonuoma
```

✓ Your SQL query has been executed successfully (Query took 0.0374 sec)

```
CREATE DATABASE autonuoma
```

[Show query box](#)

DB sukūrimas

CREATE DATABASE [IF NOT EXISTS] db_vardas;

CREATE DATABASE **autonuoma**

❗ #1007 - Can't create database 'autonuoma'; database exists

Run SQL query/queries on database **autonuoma**: ?

```
1 CREATE DATABASE autonuoma;
```

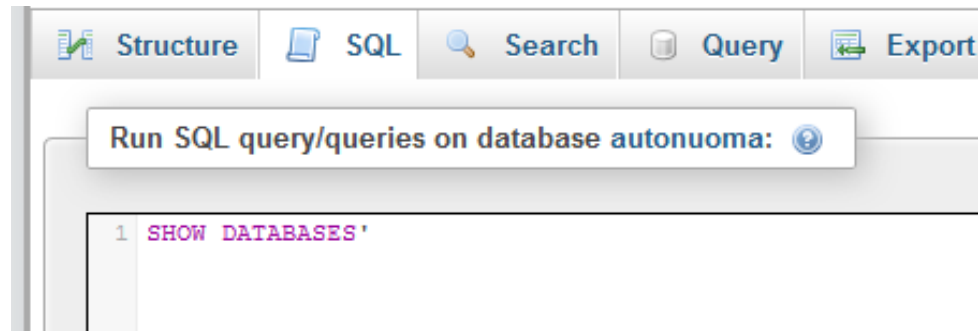
✓ Your SQL query has been executed successfully (Query took 0.0002 sec)

CREATE DATABASE IF NOT EXISTS autonuoma

DB sukūrimas

Norint peržiūrėti jums matomų duomenų bazių sąrašą reikia naudoti komanda:

SHOW DATABASES;



Information_schema

Meta duomenų bazė, kurioje saugoma kitų, serveryje esančių, duomenų bazių struktūrinė informacija.

DB šalinimas

Duomenų bazės šalinimas – fizinis duomenų bazės pašalinimas iš DBVS. Pašalinami visi duomenys ir visos DB struktūros:

- DB lentelės (angl. tables);
- Virtualios lentelės (angl. view);
- Trigeriai (angl. triggers);
- Serverio procedūros (angl. stored procedūros);
- Serverio funkcijos (angl. functions);
- Indeksai (angl. index).

SQL duomenų tipai

My sql duomenų tipai

Skaitiniai:

TINYINT	-128 iki 127
SMALLINT	-32768 iki 32767
MEDIUMINT	-8388608 iki 8388607
INT	-2147483648 iki 2147483647
BIGINT	-9223372036854775808 iki 9223372036854775807
FLOAT	4 baitų
DOUBLE	8 baitai
DECIMAL	16 baitai
BIT	1 iki 64 bitų seka

My sql duomenų tipai

Skaitiniai:

Pavadinimas	Galimos reikšmės su ženklu	Galimo reikšmės be ženklo	Užimama vieta
TINYINT	-128 iki 127	0 iki 255	1 byte
SMALLINT	-32768 iki 32767	0 iki 65535	2 bytes
MEDIUMINT	-8388608 iki 8388607	0 iki 16777215	3 bytes
INT	-2147483648 iki 2147483647	0 iki 4294967295	4 bytes
BIGINT	-9223372036854775808 iki 9223372036854775807	0 iki 18446744073709551615	8 bytes
BOOL, BOOLEAN			1 byte

My sql duomenų tipai



My sql duomenų tipai

AmazonAWS treileris gali
pervežti 100 PB duomenų mažiau
nei per savaitę.

Tą patį duomenų kiekį perkelti per 1 Gbps
tinklą
užtruks apie 20 metų.

My sql duomenų tipai

Eilutės:

Duomenų tipas	Ilgis
CHAR	0-255 simboliai
VARCHAR	0-65535 simboliai
BINARY	0-255 baitai
VARBINARY	0-65535 baitai
TINYBLOB	255 baitai
BLOB	65535 baitai
MEDIUMBLOB	16777215 baitai
LONGBLOB	4294967295 baitai
TINYTEXT	255 simboliai
TEXT	65535 baitai
MEDIUMTEXT	16777215 simboliai
LONGTEXT	4294967295 simboliai
ENUM	1 reikšmė iš max 65535 reikšmių sąrašo
SET	Iki 64 reikšmė iš max 65535 reikšmių sąrašo

My sql duomenų tipai

Reikšmė	CHAR(4)	Duomenų kiekis	VARCHAR(4)	Duomenų kiekis
"	' '	4 bytes	"	1 byte
'ab'	'ab '	4 bytes	'ab'	3 bytes
'abcd'	'abcd'	4 bytes	'abcd'	5 bytes
'abcdefgh'	'abcd'	4 bytes	'abcd'	5 bytes

- Paskutinės eilutės įterpimas galimas, tik tada, kai MySQL sisteminis kintamasis *SQL_MODE* nėra nustatytas į **STRICT_TRANS_TABLES**;
- CHAR galiniai tarpai pašalinami išgaunant duomenis, nebent įjungtas SQL režimas **PAD_CHAR_TO_FULL_LENGTH**.

My sql duomenų tipai

Laiko Datos:

DATE	CCYY-MM-DD
TIME	hh:mm:ss
DATETIME	CCYY-MM-DD hh:mm:ss
TIMESTAMP	CCYY-MM-DD hh:mm:ss
YEAR	CCYY or YY

My sql duomenų tipai

Erdvinių duomenų:

GEOMETRY

POINT

LINESTRING

POLYGON

GEOMETRYCOLLECTION

MULTILINESTRING

MULTIPOINT

MULTIPOLYGON

SQL DDL ir DB struktūrų kūrimas

DB lentelių sukūrimas

```
CREATE TABLE [IF NOT EXISTS] lentelės vardas(  
    stulpelių_sąrašas  
    ) engine=lentelės_tipas
```

lentelės_vardas - turi būti unikalus duomenų bazėje, rekomenduojam vardui naudoti daugiskaitą;

stulpelių_sąrašas - stulpeliai atskiriami kableliais;

DB variklis - gali pasirinkti bet kurį iš ankščiau išvadintų, jei nenurodote pagal nutylėjimą (nuo 5.5 versijos) parenkamas InnoDB.

DB lentelių stulpelio aprašymas

Stulpelio_vardas duomenų_tipas [size] [NOT
NULL|NULL] [DEFAULT value]

[AUTO_INCREMENT]

Stulpelio_vardas – unikalus lentelėje

Duomenų tipas ir lauko dydis – nurodomas vienas iš anksčiau paminėtų duomenų tipų ir priklausomai nuo tipo nurodomas lauko dydis;

NOT NULL|NULL – nurodo ar stulpelyje gali būti null reikšmės.

DEFAULT value – reikšmė kuri priskiriama stulpeliui pagal nutylėjimą.

AUTO_INCREMENT – nurodoma, kad stulpelio reikšmė didinama automatiškai, lentelėje gali būti tik vienas toks laukas.

DB lentelių stulpelio aprašymas

```
CREATE TABLE markes
(
  id int (11) NOT NULL AUTO_INCREMENT
  pavadinimas varchar (20) NOT NULL,
);

CREATE TABLE IF NOT EXISTS klientai
(
  asmens_kodas char (11) NOT NULL,
  vardas varchar (20) NOT NULL,
  pavarde varchar (20) NOT NULL,
  gimimo_data date NOT NULL,
  telefonas varchar (20) NOT NULL,
  epastas varchar (40) DEFAULT NULL,
  uzsakymu_kiekis int default 0
)ENGINE = InnoDB;
```


DB lentelių pervadinimas

RENAME TABLE senas_lentelės_vardas **TO**
naujas_lentelės_vardas;

RENAME TABLE miestai TO apskritys;

RENAME galima taikyti lentelėms ir virtualioms lentelėms (angl.view), bet negalima taikyti laikinoms lentelėms.

DB lentelių modifikavimas

*DB objektų modifikavimui naudojama komanda
ALTER*

DB lentelių atveju, tai

```
CREATE TABLE miestai  
(  
    id int (11) NOT NULL,  
    pavadinimas varchar (20) NOT NULL,);
```

```
ALTER TABLE miestai  
ADD PRIMARY KEY (ID);
```

DB lentelių modifikavimas

ALTER TABLE table_name

ADD column_name datatype

ALTER TABLE table_name

DROP COLUMN column_name

DB lentelių šalinimas

DB lentelių šalinimui naudojama DROP TABLE komanda.

```
DROP [TEMPORARY] TABLE [IF EXISTS]  
lenteles_vardas [, lentelės_vardas] ...
```

Galima šalinti ir keletą lentelių vienu metu, jei atskiriate kableliu.

DB lentelių šalinimas

DB lentelių šalinimui naudojama DROP TABLE komanda.

DROP [TEMPORARY] TABLE [IF EXISTS] table_name [, table_name] ...

DROP TABLE miestai

DB lentelių šalinimas

DB lentelių šalinimui galima naudoti ir LIKE sąlygą.

```
DROP TABLE LIKE '%pattern%'
```

```
DROP TABLE Miestai
```

```
DROP TABLE Miestai,Klientai
```

```
DROP TABLE LIKE ,Miest%'
```

DB laikinos lentelės

DB laikinos lentelės naudotinis, kai jums reikia laikinai pasisaugoti gautą rezultatą, kuri norite panaudoti keletą kartų esamoje sesijoje.

CREATE TEMPORARY TABLE

DROP TEMPORARY TABLE

MySQL variklis laikinasias lenteles pašalina automatiškai, kai užbaigiama sesija. Laikinos lentelės yra pasiekiamos tik sesijos kūrėjui. Laikinos lentelės vardas turi būti unikalus sesijos ribose.

DB laikinos lentelės

CREATE TEMPORARY TABLE

```
CREATE TEMPORARY TABLE top10customers  
SELECT p.customerNumber,  
        c.customerName,  
        FORMAT(SUM(p.amount),2) total  
FROM payments p  
INNER JOIN customers c ON  
c.customerNumber = p.customerNumber  
GROUP BY p.customerNumber  
ORDER BY total DESC  
LIMIT 10
```


DB laikinos lentelės

Laikinos lentelės vardas gali sutapti ir su esamos lentelės vardu. Sukūrus laikiną lentelę, esanti db lentelė bus nepasiekiam iki tol, kol bus sunaikinta laikinojo lentelė.

Tai galima daryti, bet nerekomenduotina.

PVZ.: Sistemoje, kurioje realizuotas automatinis prisijungimas dingus ryšiui, naudojas dirbes su laikina lentele bus perjungtas prie esamos stacionarios db lentelės. DB variklis šioje vietoje nekontroliuoja, koks lentelės statusas.

DB lentelių apribojimai

Apribojimai

SQL apribojimai naudojami aprašyti lentelių duomenų kontrolės taisykles.

Jei įvyksta prieštara tarp duomenų DML užklausų ir apribojimų, duomenų modifikavimo užklausa atšaukiamos.

Apribojimai gali būti sukurti:

- kuriant duomenų bazės lentelę (CREATE TABLE);
- arba modifikuojant (ALTER table).

CREATE TABLE <Lentelės vardas>

(

<1_stulelio_vardas> <duomenų tipas>[(dydis)] [apribojimas]

<2_stulelio_vardas> <duomenų tipas>[(dydis)] [apribojimas]

<3_stulelio_vardas> <duomenų tipas>[(dydis)] [apribojimas]

....

);

Apribojimai

SQL apribojimų tipai:

- **NOT NULL** – tuščios reikšmės saugojimo apribojimas - nurodo, kad lentelės stulpeliuose negali būti saugomos NULL reikšmės;
- **UNIQUE** – reikšmių unikalumo apribojimas - užtikrina, kad kiekviena stulpelio ar jų rinkinio reikšmė turėtų unikalią reikšmę;
- **PRIMARY KEY** – pirminio rakto apribojimas - t.y. NOT NULL ir UNIQUE apribojimo derinys, šis apribojimas užtikrina, kad stulpelis arba stulpelių derinys užtikrintų unikalią reikšmę, pagal kurią galima identifikuoti konkretų duomenų bazės lentelės įrašą;
- **FOREIGN KEY** – išorinio raktos apribojimas - užtikrina duomenų bazės lentelių tarpusavio reikšmių integralumą, t.y. tikrina, kad susijusių lentelių stulpelių reikšmės tarpusavyje derėtų;
- **CHECK** – reikšmių kontrolės apribojimas - užtikrina, kad stulpelio reikšmė tenkintų konkrečias sąlygas;
- **DEFAULT** – reikšmės pagal nutylėjimą apribojimas - nusako stulpelio reikšmę pagal nutylėjimą.

Apribojimai NOT NULL

SQL apribojimų tipai:

- **NOT NULL** – nurodo, kad lentelės stulpeliuose negali būti saugomos NULL reikšmės;

```
CREATE TABLE aiksteles
```

```
(
```

```
    id int NOT NULL AUTO_INCREMENT,
```

```
    pavadinimas varchar (20) NOT NULL,
```

```
    adresas text NOT NULL,
```

```
    fk_miestas int (11) NOT NULL,
```

```
    el_pastas varchar (50),
```

```
    PRIMARY KEY(id),
```

```
    CONSTRAINT fkc_vieta FOREIGN KEY(fk_miestas)
```

```
REFERENCES miestai (id)
```

```
)ENGINE = InnoDB;
```

Apribojimai UNIQUE

MYSQL apribojimų tipai:

- **UNIQUE** – užtikrina, kad kiekviena stulpelio ar jų rinkinio reikšmė turėtų unikalią reikšmę;

CREATE TABLE aiksteles

(

id int NOT NULL,

pavadinimas varchar (20) NOT NULL,

adresas text NOT NULL,

fk_miestas int (11) NOT NULL,

UNIQUE(id)

)ENGINE = InnoDB;

Apribojimai

MS SQL, ORACLE apribojimų tipai:

- **UNIQUE** – užtikrina, kad kiekviena stulpelio ar jų rinkinio reikšmė turėtų unikalią reikšmę;

CREATE TABLE aiksteles

(

id int NOT NULL **UNIQUE**,

pavadinimas varchar (20) NOT NULL,

adresas text NOT NULL,

fk_miestas int (11) NOT NULL,

);

Apribojimai

SQL apribojimų tipai:

- **UNIQUE** – užtikrina, kad kiekviena stulpelio ar jų rinkinio reikšmė turėtų unikalią reikšmę;

CREATE TABLE aiksteles

(

id int NOT NULL,

pavadinimas varchar (20) NOT NULL,

adresas text NOT NULL,

fk_miestas int (11) NOT NULL,

CONSTRAINT UC_aikstele UNIQUE(pavadinimas, adresas)

)ENGINE = InnoDB;

Apribojimai

SQL apribojimų tipai:

- **UNIQUE** – užtikrina, kad kiekviena stulpelio ar jų rinkinio reikšmė turėtų unikalią reikšmę;

ALTER TABLE aiksteles

ADD UNIQUE(id);

ALTER TABLE aiksteles

ADD CONSTRAINT UC_aikstele UNIQUE(pavadinimas, adresas);

ALTER TABLE aiksteles

DROP INDEX UC_aisktele

PASTABA: kitose DBVS šalinimui naudojama DROP CONSTRAINT <vardas>

DB lentelių pirminis raktas

3 taisyklės:

- Reikšmės lauke arba laukų kombinacija turi būti unikalūs;
- Pirminis raktas negali turėti NULL reikšmių.
- Lentelė turi tik vieną pirminį raktą.

Kadangi MySQL greičiau dirba su sveikais skaičiais, rekomenduotina pirminį raktą daryti sveiką skaičių: int, bigint. Galima naudoti ir tinyint, smallint, tik reikia įvertinti ar pakaks duomenų tipo leidžiamų reikšmių režio.

Apribojimai PRIMARY KEY

MY SQL apribojimų tipai:

- **PRIMARY KEY** – t.y. NOT NULL ir UNIQUE apribojimo derinys, šis apribojimas užtikrina, kad stulpelis arba stulpelių derinys užtikrintų unikalią reikšmę, pagal kurią galima identifikuoti konkretų duomenų bazės lentelės įrašą;

CREATE TABLE aiksteles

(

id int NOT NULL,

pavadinimas varchar (20) NOT NULL,

adresas text NOT NULL,

fk_miestas int (11) NOT NULL,

PRIMARY KEY(id),

CONSTRAINT fkc_vieta FOREIGN KEY(fk_miestas) REFERENCES miestai (id)

)ENGINE = InnoDB;

Apribojimai

MS SQL, ORACLE apribojimų tipai:

- **PRIMARY KEY** – t.y. NOT NULL ir UNIQUE apribojimo derinys, šis apribojimas užtikrina, kad stulpelis arba stulpelių derinys užtikrintų unikalią reikšmę, pagal kurią galima identifikuoti konkretų duomenų bazės lentelės įrašą;

```
CREATE TABLE aiksteles
```

```
(
```

```
    id int NOT NULL PRIMARY KEY,
```

```
    pavadinimas varchar (20) NOT NULL,
```

```
    adresas text NOT NULL,
```

```
    fk_miestas int (11) NOT NULL,
```

```
    CONSTRAINT fkc_vieta FOREIGN KEY(fk_miestas) REFERENCES miestai (id)
```

```
);
```

Apribojimai

MY SQL, MS SQL, ORACLE apribojimų tipai:

- **PRIMARY KEY** – t.y. NOT NULL ir UNIQUE apribojimo derinys, šis apribojimas užtikrina, kad stulpelis arba stulpelių derinys užtikrintų unikalią reikšmę, pagal kurią galima identifikuoti konkretų duomenų bazės lentelės įrašą;

```
CREATE TABLE aiksteles
```

```
(
```

```
    id int NOT NULL,
```

```
    pavadinimas varchar (20) NOT NULL,
```

```
    adresas text NOT NULL,
```

```
    fk_miestas int (11) NOT NULL,
```

```
    PRIMARY KEY(pavadinimas,adresas),
```

```
    CONSTRAINT fkc_vieta FOREIGN KEY(fk_miestas) REFERENCES miestai (id)
```

```
);
```

DB lentelių pirminis raktas

Sukuriant pirminį raktą MySQL variklis sukuria lentelei ir pirminio rakto indeksą, kurio vardas PRIMARY.

Rekomenduojama pirminį raktą, jei tai sveikas skaičius daryti auto increment.

```
REATE TABLE miestai
```

```
(
```

```
    id int (11) AUTO_INCREMENT PRIMARY KEY,
```

```
    pavadinimas varchar (20) NOT NULL
```

```
);
```

Apribojimai

ALTER TABLE aiksteles

ADD PRIMARY KEY (id);

ALTER TABLE aiksteles

ADD CONSTRAINT PK_aikstele PRIMARY KEY (pavadinimas, adresas);

Apribojimai

ŠALINIMAS pirminio rakto MYSQL

ALTER TABLE aiksteles

DROP PRIMARY KEY

ŠALINIMAS pirminio raktos MS SQL/ORACLE

ALTER TABLE aiksteles

DROP CONSTRAINT PK_aisktele

Apribojimai FOREIGN KEY

MYSQL apribojimų tipai:

- **FOREIGN KEY** – užtikrina duomenų bazės lentelių tarpusavio reikšmių integralumą, t.y. tikrina, kad susijusių lentelių stulpelių reikšmės tarpusavyje derėtų;

CREATE TABLE aiksteles

(

id int NOT NULL AUTO_INCREMENT,

pavadinimas varchar (20) NOT NULL,

adresas text NOT NULL,

fk_miestas int (11) NOT NULL,

PRIMARY KEY(id),

FOREIGN KEY(fk_miestas) REFERENCES miestai (id)

)ENGINE = InnoDB;

Apribojimai

MS SQL, ORACLE apribojimų tipai:

CREATE TABLE aiksteles

(

id int NOT NULL AUTO_INCREMENT,

pavadinimas varchar (20) NOT NULL,

adresas text NOT NULL,

fk_miestas int (11) NOT NULL,

PRIMARY KEY(id),

CONSTRAINT fkc_vieta FOREIGN KEY(fk_miestas) REFERENCES miestai (id)

);

Apribojimai

MYSQL, MS SQL, ORACLE apribojimų tipai:

CREATE TABLE aiksteles

(

id int NOT NULL AUTO_INCREMENT,

pavadinimas varchar (20) NOT NULL,

adresas text NOT NULL,

fk_miestas varchar (20) NOT NULL,

fk_rajonas varchar (20) NOT NULL,

PRIMARY KEY(id),

CONSTRAINT fkc_vieta FOREIGN KEY(fk_miestas,fk_rajonas) REFERENCES miestai
(miestas, rajonas)

)ENGINE = InnoDB;

DB lentelių Išorinio rakto apribojimas

CONSTRAINT constraint_name

FOREIGN KEY foreign_key_name (columns)

REFERENCES parent_table(columns)

ON DELETE action

ON UPDATE action

ON DELETE:

- ON DELETE SET NULL;
- ON DELETE NO ACTION;
- ON DELETE RESTRICT;
- ON DELETE SET DEFAULT; (palaiko MySQL InnoDB ir NDB)
- ON DELETE CASCADE;

DB lentelių Išorinio rakto apribojimas

CONSTRAINT constraint_name

FOREIGN KEY foreign_key_name (columns)

REFERENCES parent_table(columns)

ON DELETE action

ON UPDATE action

ON UPDATE:

- ON UPDATE SET NULL;
- ON UPDATE NO ACTION;
- ON UPDATE RESTRICT;
- ON UPDATE CASCADE;

Apribojimai

Keitimas išorinio rakto MYSQL , MS SQL, ORACLE

ALTER TABLE aiksteles

FOREIGN KEY(fk_miestas) REFERENCES miestai (id);

ALTER TABLE aiksteles

CONSTRAINT fkc_vieta FOREIGN KEY(fk_miestas,fk_rajonas) REFERENCES miestai
(miestas, rajonas);

Apribojimai

ŠALINIMAS išorinio rakto MySQL

ALTER TABLE aiksteles

DROP FOREIGN KEY fkc_vieta

ŠALINIMAS išorinio rakto MS SQL/ORACLE

ALTER TABLE aiksteles

DROP CONSTRAINT fkc_vieta

Apribojimai CHECK

SQL **CHECK** – užtikrina, kad stulpelio reikšmė tenkintų konkrečias sąlygas;

PASTABA: nuo 8.0.16 versijos **MYSQL** veikia šis apribojimas, iki šios versijos sintaksė numatyta, skriptas parsinamas, bet visi **MYSQL** varikliai **IGNORUOJA**.

```
CREATE TABLE automobiliai
```

```
(
```

```
    id int (11) NOT NULL AUTO_INCREMENT,
```

```
    valstybinis_nr char (6) NOT NULL,
```

```
    pagaminimo_data date NOT NULL,
```

```
    rida int (11) NOT NULL,
```

```
    vietu_skaicius smallint NOT NULL,
```

```
    verte decimal (8,2) NOT NULL,
```

```
    degalu_tipas int (11) NOT NULL,
```

```
    kebulas int (11) NOT NULL,
```

```
    CHECK (vietu_skaicius>2)
```

```
);
```


Apribojimai CHECK

CREATE TRIGGER check_vietu_skaicius BEFORE UPDATE ON
automobiliai

FOR EACH ROW

BEGIN

DECLARE msg varchar(255);

IF (new.vietu_skaicius <=2)

THEN

SET msg = concat('Vietų skaičius turi būti didesnis už 2, o jūsų
reikšmė:', cast(new.vietu_skaicius as char));

SIGNAL sqlstate '45000' SET message_text = msg;

END IF;

END

Apribojimai

MS SQL, ORACLE **CHECK** – užtikrina, kad stulpelio reikšmė tenkintų konkrečias sąlygas;

CREATE TABLE automobiliai

```
(  
    id int (11) NOT NULL AUTO_INCREMENT,  
    valstybinis_nr char (6) NOT NULL,  
    pagaminimo_data date NOT NULL,  
    rida int (11) NOT NULL,  
    vietu_skaicius smallint NOT NULL CHECK (vietu_skaicius>2),  
    verte decimal (8,2) NOT NULL,  
    degalu_tipas int (11) NOT NULL,  
    kebulas int (11) NOT NULL  
);
```

Apribojimai

MY SQL, MS SQL, ORACLE **CHECK** – užtikrina, kad stulpelio reikšmė tenkintų konkrečias sąlygas;

CREATE TABLE automobiliai

(

id int (11) NOT NULL AUTO_INCREMENT,

valstybinis_nr char (6) NOT NULL,

pagaminimo_data date NOT NULL,

rida int (11) NOT NULL,

vietu_skaicius smallint NOT NULL,

verte decimal (8,2) NOT NULL,

degalu_tipas int (11) NOT NULL,

kebulas int (11) NOT NULL

CONSTRAINT chk_masina CHECK (vietu_skaicius > 2 and rida < 90000)

)ENGINE = InnoDB;

Apribojimai

MY SQL,MS SQL, ORACLE **CHECK** – užtikrina, kad stulpelio reikšmė tenkintų konkrečias sąlygas;

```
CREATE TABLE aiksteles
```

```
(
```

```
    id int NOT NULL,
```

```
    pavadinimas varchar (20) NOT NULL CHECK (pavadinimas IN (select pavadinimas from aiksteles)),
```

```
    adresas text NOT NULL,
```

```
    fk_miestas int (11) NOT NULL,
```

```
    PRIMARY KEY(pavadinimas,adresas),
```

```
    CONSTRAINT fkc_vieta FOREIGN KEY(fk_miestas) REFERENCES miestai (id)
```

```
);
```

Apribojimai

MS SQL, ORACLE **CHECK** – užtikrina, kad stulpelio reikšmė tenkintų konkrečias sąlygas;

```
CREATE TABLE saskaitos
```

```
(
```

```
    nr char (5) NOT NULL,
```

```
    data date NOT NULL,
```

```
    suma decimal (8,2) NOT NULL,
```

```
    fk_sutartis int (11) NOT NULL,
```

```
    CONSTRAINT chk_data CHECK (data LIKE '--/--/----')
```

```
    PRIMARY KEY(nr),
```

```
    CONSTRAINT fkc_saskaitos_sutartis FOREIGN KEY(fk_sutartis) REFERENCES sutartys (nr)
```

```
);
```

Apribojimai

Keitimas check apribojimo MYSQL , MS SQL, ORACLE

ALTER TABLE aiksteles

CHECK (vietu_skaisius>2) ;

ALTER TABLE aiksteles

CONSTRAINT chk_masina CHECK (vietu_skaisius>2 and rida < 100000)

Apribojimai

ŠALINIMAS CHECK apribojimo MYSQL

ALTER TABLE aiksteles

DROP CHECK chk_masina

ŠALINIMAS CHECK apribojimo MS SQL/ORACLE

ALTER TABLE aiksteles

DROP CONSTRAINT chk_masina

Apribojimai DEFAULT

SQL apribojimų tipai:

- **DEFAULT** – nusako stulpelio reikšmę pagal nutylėjimą.

CREATE TABLE automobiliai

(

```
    id int (11) NOT NULL AUTO_INCREMENT,  
    valstybinis_nr char (6) NOT NULL DEFAULT `AAA 111`,  
    pagaminimo_data date NOT NULL,  
    rida int (11) NOT NULL,  
    vietu_skaicius smallint NOT NULL DEFAULT 2,  
    verte decimal (8,2) NOT NULL,  
    degalu_tipas int (11) NOT NULL,  
    kebulas int (11) NOT NULL  
    )ENGINE = InnoDB;
```


Apribojimai

SQL apribojimų tipai:

- **DEFAULT** – nusako stulpelio reikšmę pagal nutylėjimą.

CREATE TABLE saskaitos

(

nr char (5) NOT NULL,

data date NOT NULL **DEFAULT GETDATE()**,

suma decimal (8,2) NOT NULL,

fk_sutartis int (11) NOT NULL,

PRIMARY KEY(nr),

CONSTRAINT fkc_saskaitos_sutartis FOREIGN KEY(fk_sutartis) REFERENCES
sutartys (nr)

)ENGINE = InnoDB;

Apribojimai

Pridėjimas DEFAULT apribojimo MYSQL

ALTER TABLE aiksteles

ALTER valstybinis_nr char SET DEFAULT `AAA 111`;

Pridėjimas DEFAULT apribojimo MS SQL

ALTER TABLE aiksteles

ALTER COLUMN valstybinis_nr char SET DEFAULT `AAA 111`;

Pridėjimas DEFAULT apribojimo ORACLE

ALTER TABLE aiksteles

MODIFY valstybinis_nr char DEFAULT `AAA 111`;

Apribojimai

ŠALINIMAS DEFAULT apribojimo MYSQL

ALTER TABLE aiksteles

ALTER valstybinis_nr char DROP DEFAULT;

ŠALINIMAS DEFAULT apribojimo MS SQL/ORACLE

ALTER TABLE aiksteles

ALTER COLUMN DROP valstybinis_nr char DROP DEFAULT;