SQL DDL

Limbajul de definitie a datelor – DDL

Creare baza de date

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name
[create_specification] ...

create_specification:
[DEFAULT] CHARACTER SET [=] charset_name | [DEFAULT]
COLLATE [=] collation_name
```

Exemple:

```
CREATE DATABASE test
```

```
CREATE DATABASE IF NOT EXISTS test
```

```
CREATE DATABASE IF NOT EXISTS test
CHARACTER SET = 'ascii';
```

```
CREATE DATABASE IF NOT EXISTS test
CHARACTER SET = 'utf8';
COLLATE = 'utf8_bin';
```

```
SHOW COLLATION;
```

```
SHOW CHARACTER SET;
```

Modificare baza de date

```
ALTER {DATABASE | SCHEMA} [db_name]
alter_specification ...
ALTER {DATABASE | SCHEMA} db_name
UPGRADE DATA DIRECTORY NAME

alter_specification:
[DEFAULT] CHARACTER SET [=] charset_name |
[DEFAULT] COLLATE [=] collation_name
```

Exemple:

```
ALTER DATABASE
CHARACTER SET = 'ascii'
COLLATE = 'ascii_bin';
```

Stergere baza de date

```
DROP {DATABASE | SCHEMA} [IF EXISTS] db_name
```

Exemple:

```
DROP DATABASE test
```

```
DROP DATABASE IF EXISTS test
```

SQL DDL

Creare tabela

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
(create_definition,...)
[table_option] ...
[partition_options]
```

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
 [(create_definition,...)]
[table_option] ...
[partition_options]
select_statement
```

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name
 { LIKE old_tbl_name | (LIKE old_tbl_name) }
```

unde:

```
create_definition:

col_name column_definition
 | [CONSTRAINT [symbol]] PRIMARY KEY [index_type] (index_col_name,...)
      [index_option] ...
 | {INDEX|KEY} [index_name] [index_type] (index_col_name,...)
      [index_option] ...
| [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY] [index_name] [index_type] (index_col_name,...)
      [index_option] ...
 | {FULLTEXT|SPATIAL} [INDEX|KEY] [index_name] (index_col_name,...)
      [index_option] ...
| [CONSTRAINT [symbol]] FOREIGN KEY [index_name] (index_col_name,...) reference_definition
| CHECK (expr)
```

SQL DDL

Exemple:

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql>
mysql> CREATE TABLE t1
    -> (c1 INT UNIQUE AUTO_INCREMENT,
    ->  c2 INT DEFAULT 1,
    ->  c3 DATE,
    ->  c4 TIME,
    ->  c5 FLOAT,
    ->  c6 DOUBLE,
    ->  c7 CHAR(20),
    ->   PRIMARY KEY (c1));
Query OK, 0 rows affected (0.01 sec)

mysql> describe t1;
+-------+----------+------+-----+---------+----------------+
| Field | Type     | Null | Key | Default | Extra          |
+-------+----------+------+-----+---------+----------------+
| c1    | int(11)  | NO   | PRI | NULL    | auto_increment |
| c2    | int(11)  | YES  |     | 1       |                |
| c3    | date     | YES  |     | NULL    |                |
| c4    | time     | YES  |     | NULL    |                |
| c5    | float    | YES  |     | NULL    |                |
| c6    | double   | YES  |     | NULL    |                |
| c7    | char(20) | YES  |     | NULL    |                |
+-------+----------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

INSERT INTO t1 SET c2 = 10, c3='2008-02-20', c4='12:30:20', c5=1.2, c6=0.02, c7='cuvant';
INSERT INTO t1 SET c3='2008/02/12', c4='8.66.20', c7='curs';

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> select * from t1;
+----+------+------------+----------+------+------+--------+
| c1 | c2   | c3         | c4       | c5   | c6   | c7     |
+----+------+------------+----------+------+------+--------+
|  1 |   10 | 2008-02-20 | 12:30:20 |  1.2 | 0.02 | cuvant |
|  2 |    1 | 2008-02-12 | 00:00:08 | NULL | NULL | curs   |
+----+------+------------+----------+------+------+--------+
2 rows in set (0.00 sec)
```

SQL DDL

si:

```
column_definition:

data_type [NOT NULL | NULL] [DEFAULT default_value]
[AUTO_INCREMENT] [UNIQUE [KEY] | [PRIMARY] KEY]
[COMMENT 'string'] [reference_definition]
[COLUMN_FORMAT {FIXED|DYNAMIC|DEFAULT}]
[STORAGE {DISK|MEMORY|DEFAULT}]
```

si:

```
data_type:

BIT[(length)]
| TINYINT[(length)] [UNSIGNED] [ZEROFILL]
| SMALLINT[(length)] [UNSIGNED] [ZEROFILL]
| MEDIUMINT[(length)] [UNSIGNED] [ZEROFILL]
| INT[(length)] [UNSIGNED] [ZEROFILL]
| INTEGER[(length)] [UNSIGNED] [ZEROFILL]
| BIGINT[(length)] [UNSIGNED] [ZEROFILL]
| REAL[(length,decimals)] [UNSIGNED] [ZEROFILL]
| DOUBLE[(length,decimals)] [UNSIGNED] [ZEROFILL]
| FLOAT[(length,decimals)] [UNSIGNED] [ZEROFILL]
| DECIMAL[(length[,decimals])] [UNSIGNED] [ZEROFILL]
| NUMERIC[(length[,decimals])] [UNSIGNED] [ZEROFILL]
| DATE | TIME | TIMESTAMP | DATETIME | YEAR
| CHAR[(length)] [CHARACTER SET charset_name] [COLLATE collation_name]
| VARCHAR(length) [CHARACTER SET charset_name] [COLLATE collation_name]
| BINARY[(length)] | VARBINARY(length)
| TINYBLOB | BLOB | MEDIUMBLOB
| LONGBLOB
| TINYTEXT [BINARY] [CHARACTER SET charset_name] [COLLATE collation_name]
| TEXT [BINARY] [CHARACTER SET charset_name] [COLLATE collation_name]
| MEDIUMTEXT [BINARY] [CHARACTER SET charset_name] [COLLATE collation_name]
| LONGTEXT [BINARY] [CHARACTER SET charset_name] [COLLATE collation_name]
| ENUM(value1,value2,value3,...) [CHARACTER SET charset_name] [COLLATE collation_name]
| SET(value1,value2,value3,...) [CHARACTER SET charset_name] [COLLATE collation_name]
| spatial_type
```

SQL DDL

Observatii:

Folosind cuvantul cheie TEMPORARY, la crearea unei tabele, tabela va exista numai in timpul sesiunii/conexiunii curente la baza de date;

Cuvantul cheie IF NOT EXISTS impiedica aparitia unei erori, in cazul in care tabela declarata exista deja; pe de alta parte, nu se verifica daca tabela existenta are aceeasi structura cu cea indicata de CREATE TABLE;

Daca nu se specifica atributul NULL sau NOT NULL, coloana e tratata ca si cum s-ar fi specificat atributul NULL;

Atributul AUTO_INCREMENT nu se poate atribui decat unei singure coloane intr-o tabela; acest atribut nu se aplica decat tipurilor intregi sau reale (float, double).

Tipurile de tip caracter (char, varchar, text) pot avea atribuite CHARACTER SET – setul de caractere atribuit acelei coloane;

Clauza DEFAULT permite setarea unei valori default pentru o coloana; de exemplu, pentru un tip data, se poate folosi o functie de tip NOW() sau CURRENT_TIME;

KEY e in mod normal un sinonim pentru INDEX; PRIMARY KEY poate fi simplu KEY atunci cand este folosit in definirea unei coloane;

PRIMARY KEY este un index pentru care toate coloanele care intra in definirea lui trebuie sa fie NOT NULL: daca nu sunt astfel, sunt definite implicit (si tacit).

Se poate creea o tabela din alta utilizand clauza SELECT la sfarsitul comenzii CREATE TABLE;

Folosind clauza LIKE, se poate creea o tabela goala folosind structura tabelei originale invocate dupa clauza LIKE;

SQL DDL

Storage engines (motoare de stocare):

Exemplu:

CREATE TABLE t (i INT) ENGINE = ‘engine_name’;

Motor	Limita stocare	Tranzactii	B-tree index	Hash-index	Granularitate blocare
MyISAM	256TB	NU	DA	NU	Tabela
InnoDB	64TB	DA	DA	DA	Inregistrare
MEMORY	RAM	NU	DA	DA	Tabela

De ce sa utilizam totusi engine-uri netranzactionale ?

-mult mai rapide;

-mai putina memorie necesara (RAM si HD);

SQL DDL

Exemple:

c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe

```
mysql>
mysql> CREATE TABLE d2.persoana
    -> (nume char(10),
    ->  prenume char(10),
    ->  adresa char(30),
    ->  telefon int,
    ->  PRIMARY KEY (nume, prenume));
Query OK, 0 rows affected (0.01 sec)

mysql> describe d2.persoana;
+---------+---------+------+-----+---------+-------+
| Field   | Type    | Null | Key | Default | Extra |
+---------+---------+------+-----+---------+-------+
| nume    | char(10)| NO   | PRI |         |       |
| prenume | char(10)| NO   | PRI |         |       |
| adresa  | char(30)| YES  |     | NULL    |       |
| telefon | int(11) | YES  |     | NULL    |       |
+---------+---------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

```
INSERT INTO d2.persoana SET nume='Preda', prenume='Gabriel', adresa='Bucuresti S6';
INSERT INTO d2.persoana SET nume='Preda', prenume='Cristian', adresa='Bucuresti S4';
INSERT INTO d2.persoana SET nume='Preda', prenume='Caterina', adresa='Bucuresti S1';
INSERT INTO d2.persoana SET nume='Popescu', prenume='Gabriel', adresa='Bucuresti S4';
INSERT INTO d2.persoana SET nume='Cristian', prenume='Preda', adresa='Brasov';
```

c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe

```
mysql> select * from persoana;
+----------+----------+--------------+---------+
| nume     | prenume  | adresa       | telefon |
+----------+----------+--------------+---------+
| Cristian | Preda    | Brasov       |    NULL |
| Popescu  | Gabriel  | Bucuresti S4 |    NULL |
| Preda    | Caterina | Bucuresti S1 |    NULL |
| Preda    | Cristian | Bucuresti S4 |    NULL |
| Preda    | Gabriel  | Bucuresti S6 |    NULL |
+----------+----------+--------------+---------+
```

SQL DDL

Exemple:

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> CREATE TABLE traducere
    -> (token_id INT UNIQUE AUTO_INCREMENT PRIMARY KEY,
    ->  token_nume char(20),
    ->  engleza char(50) CHARACTER SET'ascii',
    ->  romana char(50) CHARACTER SET 'utf8',
    ->  chineza char(50) CHARACTER SET 'big5');
Query OK, 0 rows affected (0.02 sec)
```

```
INSERT INTO traducere set token_nume='OPEN_FILE', engleza='Open', romana='Deschide', chineza='開';
INSERT INTO traducere set token_nume='SAVE_FILE', engleza='Save', romana='Salvează', chineza='除';
```

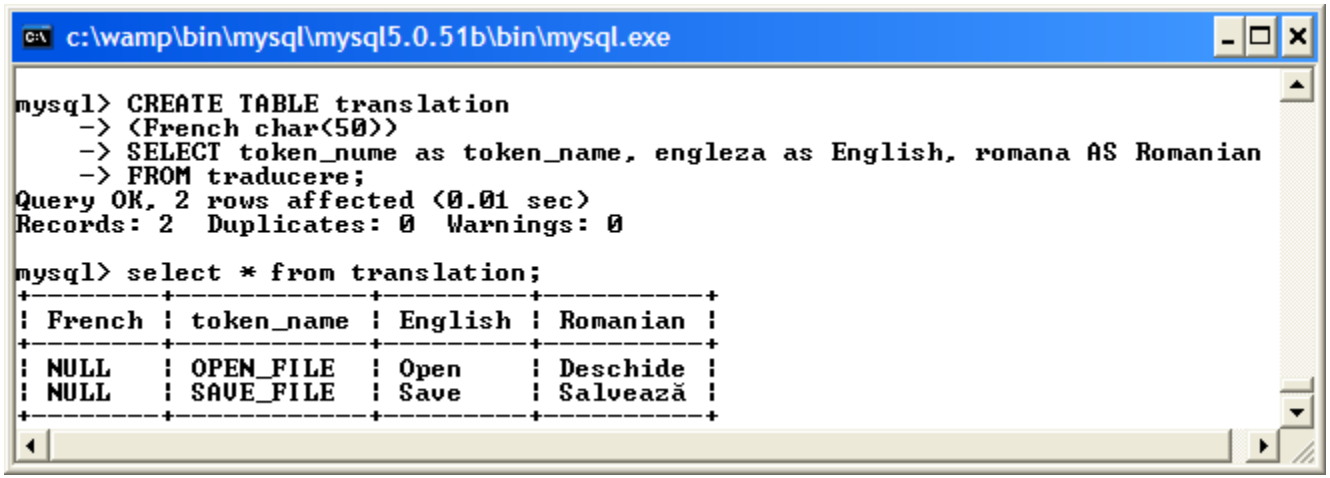
```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> select * from traducere;
+----------+------------+---------+----------+---------+
| token_id | token_nume | engleza | romana   | chineza |
+----------+------------+---------+----------+---------+
|        1 | OPEN_FILE  | Open    | Deschide | ?       |
|        2 | SAVE_FILE  | Save    | Salvează | ?       |
+----------+------------+---------+----------+---------+
2 rows in set (0.00 sec)
```

Creare tabela folosind SELECT (se copiaza si datele):

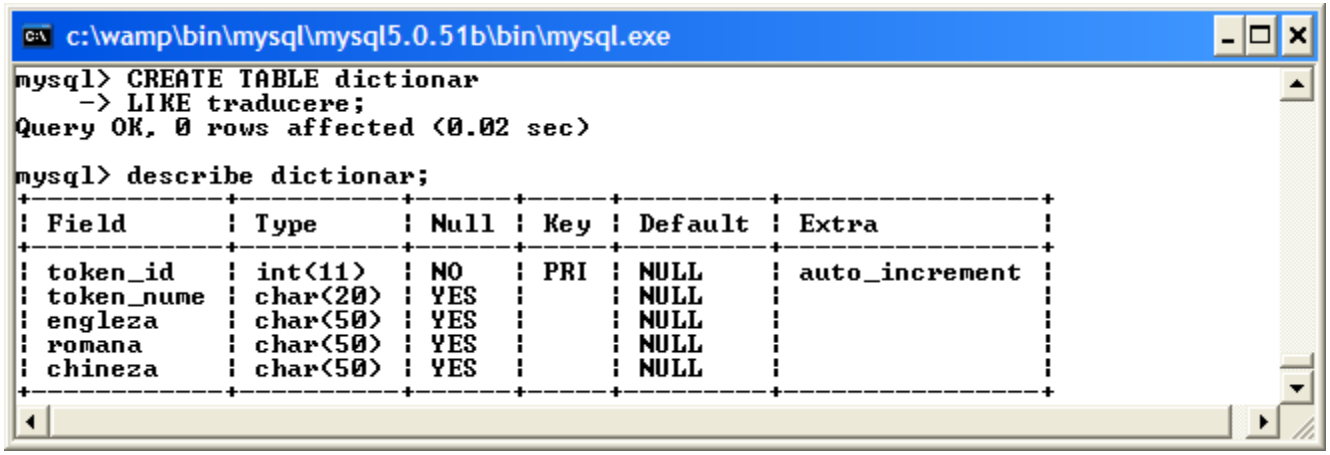
```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> CREATE TABLE translation
    -> SELECT token_nume as token_name, engleza as English, romana AS Romanian
    -> FROM traducere;
Query OK, 2 rows affected (0.02 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from translation;
+------------+---------+----------+
| token_name | English | Romanian |
+------------+---------+----------+
| OPEN_FILE  | Open    | Deschide |
| SAVE_FILE  | Save    | Salvează |
+------------+---------+----------+
2 rows in set (0.00 sec)
```

SQL DDL

Creare tabela folosind SELECT (se copiaza si datele):



Creare tabela folosind LIKE (Se copiaza numai structura tabelei, nu se pastreaza datele):



SQL DDL

Modificare tabela

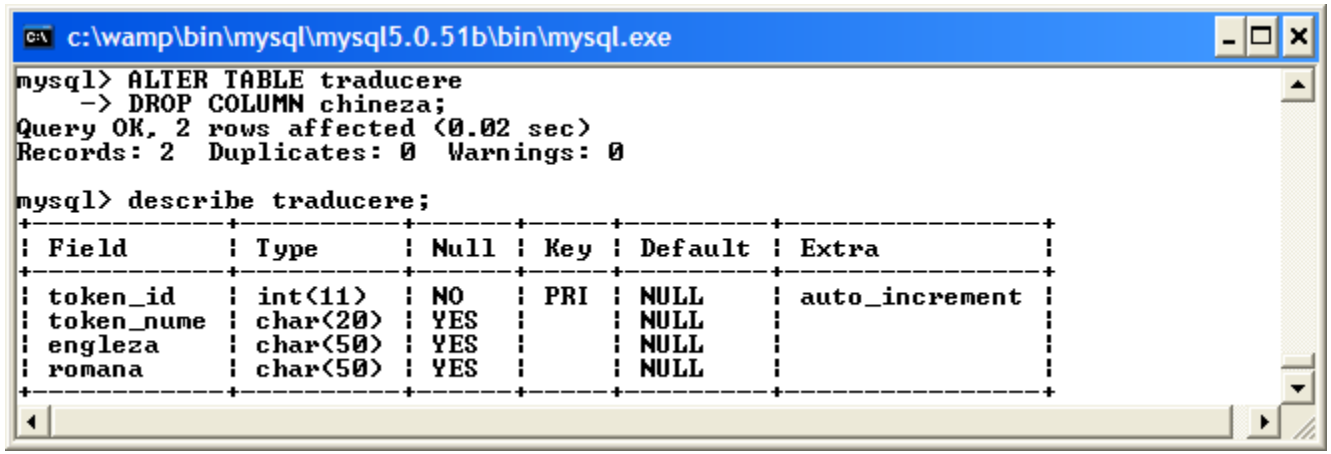
```
ALTER [ONLINE | OFFLINE] [IGNORE] TABLE tbl_name
alter_specification [, alter_specification] ...
```

```
alter_specification:

table_option ...
| ADD [COLUMN] col_name column_definition [FIRST | AFTER col_name ]
| ADD [COLUMN] (col_name column_definition,...)
| ADD {INDEX|KEY} [index_name] [index_type] (index_col_name,...) [index_option] ...
| ADD [CONSTRAINT [symbol]] PRIMARY KEY [index_type] (index_col_name,...) [index_option] ...
| ADD [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY] [index_name] [index_type] (index_col_name,...)
[index_option] ...
| ADD FULLTEXT [INDEX|KEY] [index_name] (index_col_name,...) [index_option] ...
| ADD SPATIAL [INDEX|KEY] [index_name] (index_col_name,...) [index_option] ...
| ADD [CONSTRAINT [symbol]] FOREIGN KEY [index_name] (index_col_name,...) reference_definition
| ALTER [COLUMN] col_name {SET DEFAULT literal
| DROP DEFAULT}
| CHANGE [COLUMN] old_col_name new_col_name column_definition [FIRST|AFTER col_name]
| MODIFY [COLUMN] col_name column_definition [FIRST | AFTER col_name]
| DROP [COLUMN] col_name
| DROP PRIMARY KEY
| DROP {INDEX|KEY} index_name
| DROP FOREIGN KEY fk_symbol
| DISABLE KEYS
| ENABLE KEYS
| RENAME [TO] new_tbl_name
| ORDER BY col_name [, col_name] ...
| CONVERT TO CHARACTER SET charset_name [COLLATE collation_name] | [DEFAULT] CHARACTER SET [=]
charset_name [COLLATE [=] collation_name]
[...]
```

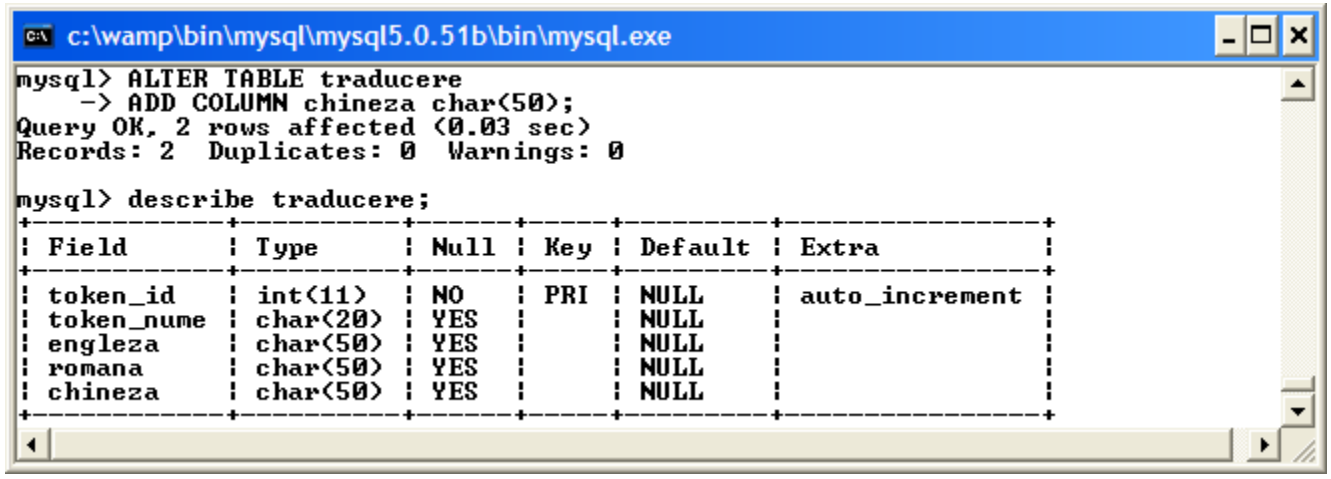
SQL DDL

Exemple:

Stergerea unei coloane



Adaugarea unei coloane



SQL DDL

Exemple:

Modificarea unei coloane

c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe

```
mysql> ALTER TABLE traducere
    -> ALTER COLUMN chineza SET DEFAULT 'Necompletat';
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from traducere;
+----------+------------+---------+----------+---------+
| token_id | token_nume | engleza | romana   | chineza |
+----------+------------+---------+----------+---------+
|        1 | OPEN_FILE  | Open    | Deschide | NULL    |
|        2 | SAVE_FILE  | Save    | Salvează | NULL    |
+----------+------------+---------+----------+---------+
2 rows in set (0.00 sec)

mysql> INSERT INTO traducere set token_nume='SAVE_AS_FILE', engleza='Save As', r
omana='Salvează ca';
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM traducere;
+----------+--------------+---------+-------------+-------------+
| token_id | token_nume   | engleza | romana      | chineza     |
+----------+--------------+---------+-------------+-------------+
|        1 | OPEN_FILE    | Open    | Deschide    | NULL        |
|        2 | SAVE_FILE    | Save    | Salvează    | NULL        |
|        3 | SAVE_AS_FILE | Save As | Salvează ca | Necompletat |
+----------+--------------+---------+-------------+-------------+
```

c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe

```
mysql> ALTER TABLE traducere
    -> ALTER COLUMN chineza DROP DEFAULT;
Query OK, 0 rows affected (0.01 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> INSERT INTO traducere set token_nume='PRINT_FILE', engleza='Print',
a='Tipăreşte';
Query OK, 1 row affected, 1 warning (0.02 sec)

mysql> SELECT * FROM traducere;
+----------+--------------+---------+-------------+-------------+
| token_id | token_nume   | engleza | romana      | chineza     |
+----------+--------------+---------+-------------+-------------+
|        1 | OPEN_FILE    | Open    | Deschide    | NULL        |
|        2 | SAVE_FILE    | Save    | Salvează    | NULL        |
|        3 | SAVE_AS_FILE | Save As | Salvează ca | Necompletat |
|        4 | PRINT_FILE   | Print   | Tipăreşte   | NULL        |
+----------+--------------+---------+-------------+-------------+
```

SQL DDL

Exemple:

Modificarea unei coloane

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> ALTER TABLE traducere
    -> CHANGE COLUMN chineza japoneza char(50);
Query OK, 4 rows affected (0.03 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM traducere;
+----------+--------------+---------+-------------+-------------+
| token_id | token_nume   | engleza | romana      | japoneza    |
+----------+--------------+---------+-------------+-------------+
|        1 | OPEN_FILE    | Open    | Deschide    | NULL        |
|        2 | SAVE_FILE    | Save    | Salvează    | NULL        |
|        3 | SAVE_AS_FILE | Save As | Salvează ca | Necompletat |
|        4 | PRINT_FILE   | Print   | Tipăreşte   | NULL        |
+----------+--------------+---------+-------------+-------------+
4 rows in set (0.00 sec)
```

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> ALTER TABLE translation
    -> MODIFY COLUMN French char(60) AFTER Romanian;
Query OK, 4 rows affected (0.03 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM translation;
+--------------+---------+-------------+--------+
| token_name   | English | Romanian    | French |
+--------------+---------+-------------+--------+
| OPEN_FILE    | Open    | Deschide    | NULL   |
| SAVE_FILE    | Save    | Salvează    | NULL   |
| SAVE_AS_FILE | Save As | Salvează ca | NULL   |
| PRINT_FILE   | Print   | Tipăreşte   | NULL   |
+--------------+---------+-------------+--------+
```

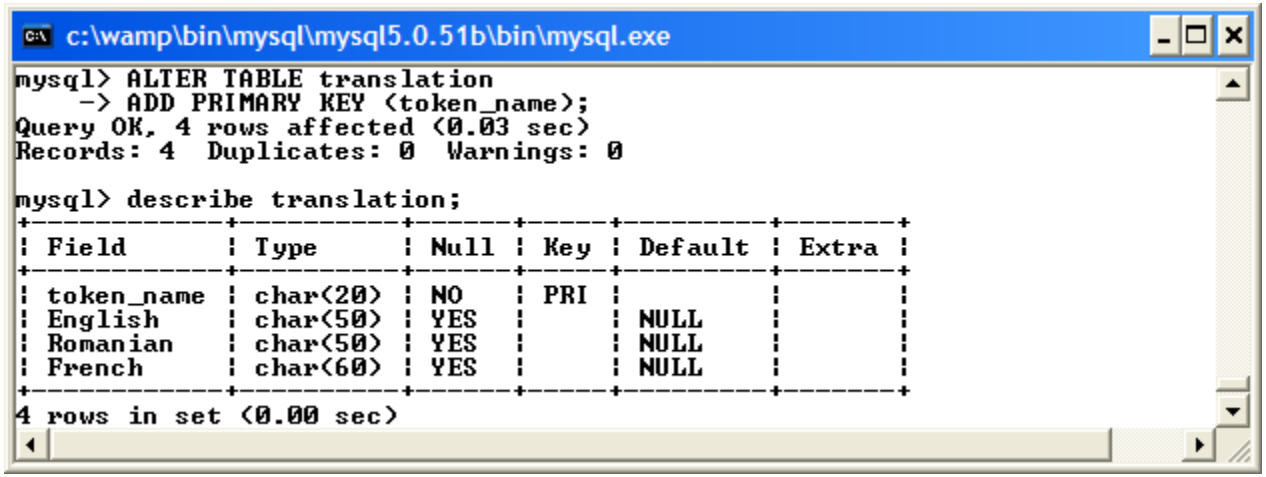
Redenumirea tabelei

```
c:\wamp\bin\mysql\mysql5.0.51b\bin\mysql.exe
mysql> ALTER TABLE translation
    -> RENAME TO translate;
Query OK, 0 rows affected (0.00 sec)

mysql> _
```

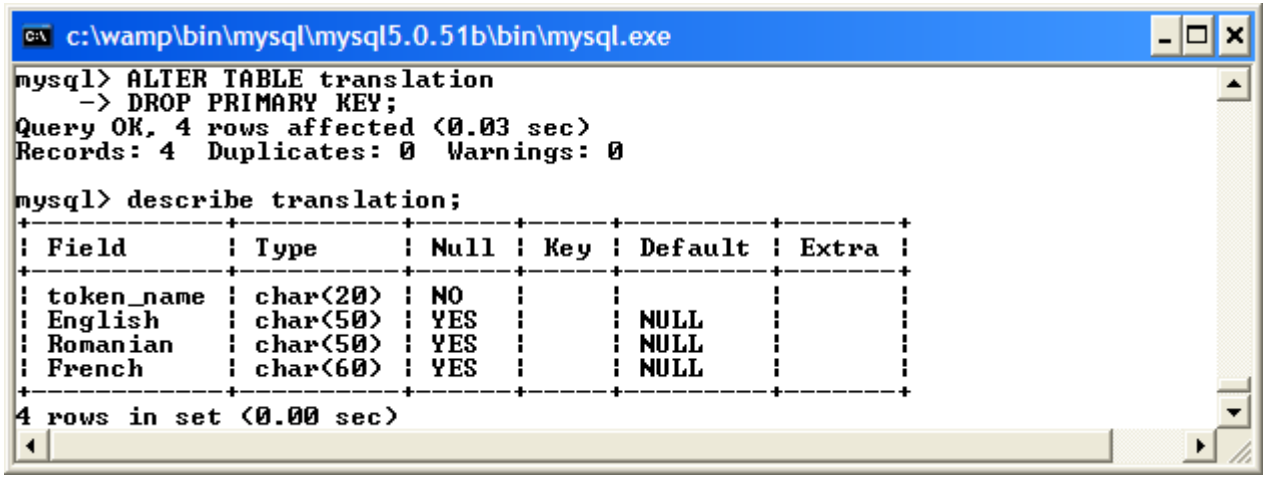
SQL DDL

Exemple:

Adaugarea unei chei



Stergerea unei chei



SQL DDL

Stergerea unei tabele

```
DROP [TEMPORARY] TABLE [IF EXISTS] tbl_name [, tbl_name]
... [RESTRICT | CASCADE]
```

Exemple:

```
DROP TABLE IF EXISTS traducere;
DROP TABLE IF EXISTS translation;
```

SQL DDL

Crearea unui view

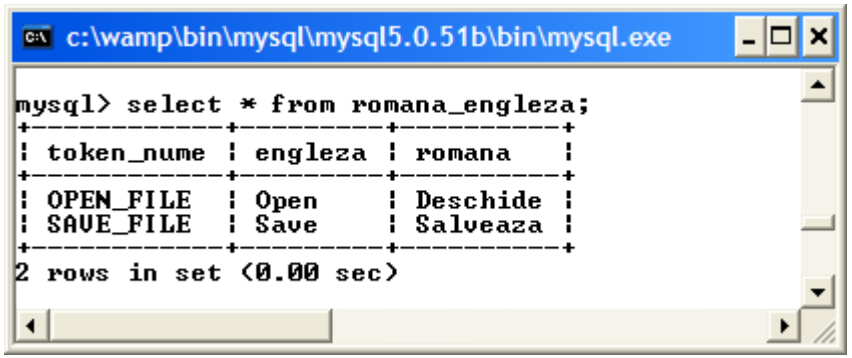
```
CREATE [OR REPLACE]
    [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
    [DEFINER = { user | CURRENT_USER }]  [SQL SECURITY { DEFINER | INVOKER }]
    VIEW view_name [(column_list)]
    AS select_statement [WITH [CASCADED | LOCAL] CHECK OPTION]
```

Mai simplu:

```
CREATE VIEW view_name AS select_statement
```

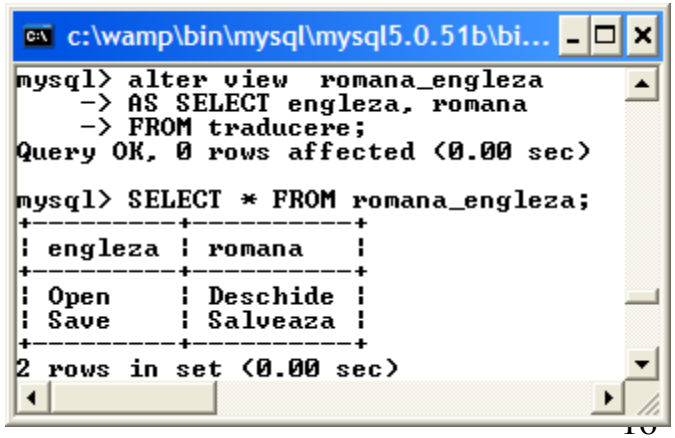
Exemplu:

```
CREATE VIEW romana_engleza
    AS SELECT token_nume, engleza, romana from traducere;
```



Modificarea unui view

```
ALTER
    [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
    [DEFINER = { user | CURRENT_USER }]
    [SQL SECURITY { DEFINER | INVOKER }]
    VIEW view_name [(column_list)]
    AS select_statement
    [WITH [CASCADED | LOCAL] CHECK OPTION]
```



Stergerea unui view

```
DROP VIEW [IF EXISTS]
    view_name [, view_name] ...
    [RESTRICT | CASCADE]
```