1. **CE ESTE CONSTRAINT / CONSTRÂNGEREA /LIMITAREA (SQL/MYSQL)?**

***O constrângere este*** pur și simplu ***o restricție*** plasată pe una sau mai multe coloane ale unui tabel ***pentru a limita tipul de valori care pot fi stocate în acea coloană***. Constrângerile oferă un mecanism standard pentru menținerea exactității și integrității datelor din cadrul unei tabele din baze de date.

Există mai multe tipuri de constrângeri în SQL, inclusiv:

***Ограничение*** — это просто ограничение, накладываемое ***на один или несколько столбцов таблицы***, ***чтобы ограничить тип значений, которые могут храниться в этом столбце***. ***Ограничения обеспечивают стандартный механизм поддержания точности и целостности данных в таблице базы данных***. В SQL существует несколько типов ограничений, в том числе:

* NOT NULL
* PRIMARY KEY
* UNIQUE
* DEFAULT
* FOREIGN KEY
* CHECK

**CONSTRÂNGEREA NOT NULL**

Constrângerea NOT NULL specifică faptul ***că coloana/столбец nu acceptă/принимает*** valorile NULL.

Aceasta înseamnă că dacă se aplică constrîngerea NOT NULL pe o coloană, atunci ***nu puteți introduce un nou rând*** în tabel fără a adăuga o valoare non-NULL pentru acea coloană.

***Это означает, что если к столбцу применяется ограничение NOT NULL, мы не можем вставить новую строку в таблицу без добавления значения, отличного от NULL, для этого столбца.***

Următoarea instrucțiune SQL creează un tabel numit **persoane** cu patru coloane, dintre care trei coloane, **id, nume și telefon** nu acceptă valori NULL.

**EXEMPLU:**

CREATE TABLE persons (

***Prezentati propriul exemplu!!!***

***Представить собственный пример***

id INT NOT NULL,

name VARCHAR(30) NOT NULL,

birth\_date DATE,

phone VARCHAR(15) NOT NULL

);

**Notă: O valoare nulă sau NULL este diferită de zero (0), nedeterminată sau un șir de caractere cu lungimea zero, cum ar fi “”. NULL înseamnă că nu a fost făcută nici o înregistrare.**

***Значение NULL не равно нулю (0), не определено или представляет собой строку символов нулевой длины, например "". NULL означает, что запись не производилась.***

## CONSTRÂNGEREA PRIMARY KEY

Constrângerea PRIMARY KEY ***identifică coloana sau setul de coloane*** care ***au valori care identifică un rând unic într-un tabel***. Nu pot două rânduri dintr-un tabel avea aceeași valoare de cheie primară. De asemenea, nu putem introduce valoarea NULL într-o coloană de chei primare.

Următoarea instrucțiune SQL creează un tabel numit ***persoane*** și specifică coloana ***ID ca cheie primară***. Aceasta înseamnă că acest câmp nu permite valori NULL sau duplicate.

**EXEMPLU:**

CREATE TABLE persons (

id INT NOT NULL auto\_increment **PRIMARY KEY**,

name VARCHAR(30) NOT NULL,

birth\_date DATE,

phone VARCHAR(15) NOT NULL

);

INSERT INTO Persons (name,birth\_date,phone)  
VALUES ('Lars', '22/22/2022','+888 22 22 22 22');

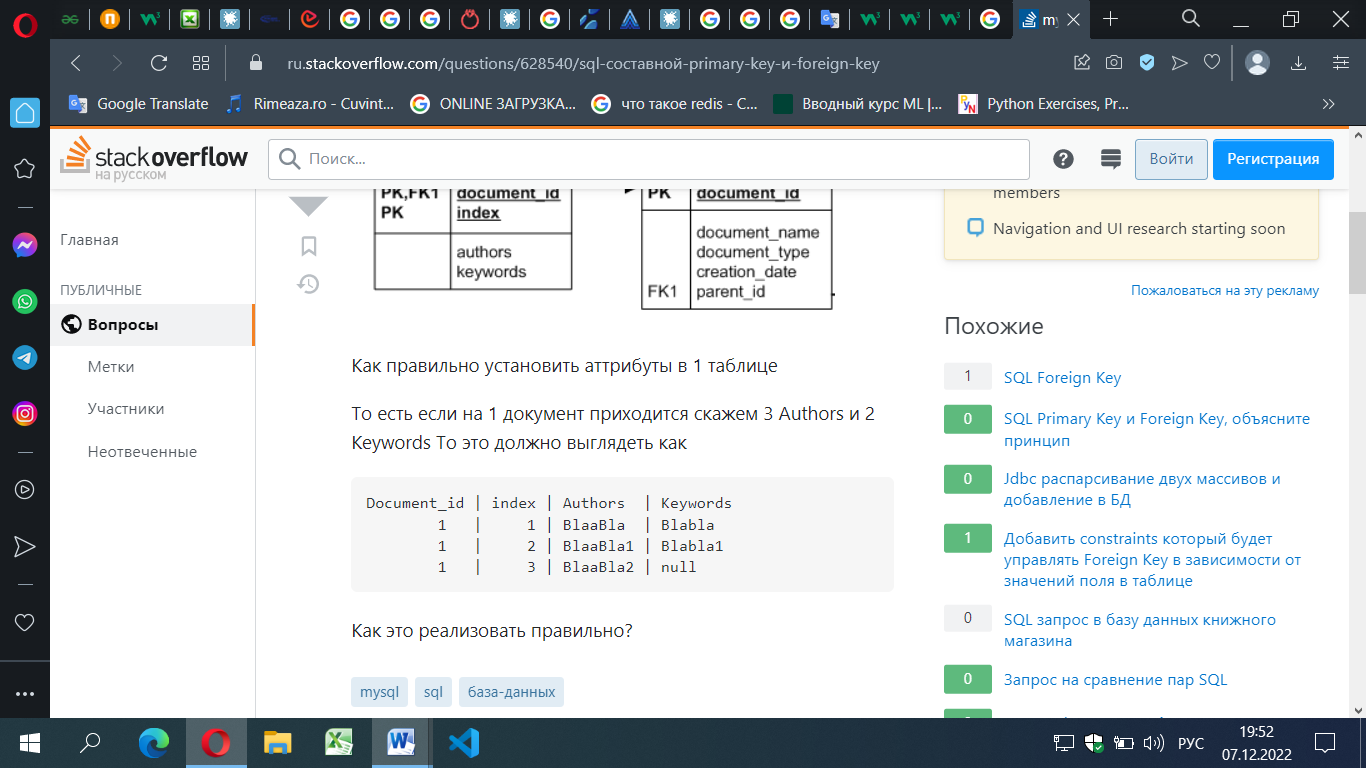
ALTER TABLE Persons AUTO\_INCREMENT=2;

Установка должного формата средствами SQL:

.... VALUES (STR\_TO\_DATE('24-09-2014', '%d-%m-%Y'))

Получение его обратно

SELECT DATE\_FORMAT(datefield, '%d-%m-%Y') FROM ......

**CREATE TABLE** document\_r

(

document\_id int NOT NULL,

index int NOT NULL,

authors varchar(255),

keywords varchar(255),

**PRIMARY KEY** (document\_id,index),

FOREIGN KEY (document\_id) REFERENCES document\_s(document\_id)

)

**CONSTRÂNGEREA UNIQUE**

Constrângerea UNIQUE ***restricționează una sau mai multe coloane*** pentru a conține valori unice într-un tabel.

Deși atât o restricție UNIQUE, cât și o restricție PRIMARY KEY impun unicitatea, utilizarea constringerii UNIQUE în loc de limitarea PRIMARY KEY ***este valabila*** ***când dorim să impunem unicitatea unei coloane sau unei combinații de coloane***. **Aceasta NU este cheie primară.**

**EXEMPLU:**

CREATE TABLE persons (

id INT NOT NULL PRIMARY KEY,

name VARCHAR(30) NOT NULL,

birth\_date DATE,

phone VARCHAR(15) NOT NULL **UNIQUE**

);

**Notă:** Intr-o tabelă pot fi mai multe constrângeri UNIQUE, în timp ce PRIMAR KEY este doar o singură constrângere. De asemenea, spre deosebire de restricția PRIMARY KEY, **restricțiile UNIQUE permit valori NULL.**

**CREATE TABLE STOCK**

**(MODEL SMALLINT NOT NULL UNIQUE,**

**MODELNAME CHAR(10) NOT NULL,**

**ITEMID INTEGER NOT NULL,**

**CONSTRAINT MOD\_UNIQUE UNIQUE (MODELNAME, ITEMID));**

**CONSTRÂNGERE DEFAULT**

Constrângerea DEFAULT ***specifică valoarea implicită/ значение по умолчанию pentru coloane/столбцов***.

O coloană implicită este o valoare care va fi introdusă în coloană de către motorul bazei de date, atunci când o instrucțiune INSERT nu atribuie explicit o anumită valoare.

Următoarea instrucțiune SQL creează o valoare implicită pentru coloana ***country***.

**EXEMPLU:**

CREATE TABLE persons (

id INT NOT NULL PRIMARY KEY,

name VARCHAR(30) NOT NULL UNIQUE,

birth\_date DATE,

phone VARCHAR(15) NOT NULL UNIQUE,

country VARCHAR(30) NOT NULL DEFAULT 'Australia'

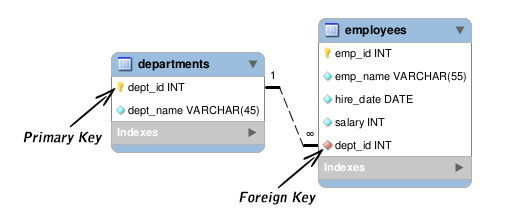
);

**Notă:** Dacă definiți o coloană de tabel drept NOT NULL, dar atribuiți coloanei o valoare implicită, ***atunci în instrucțiunea INSERT nu este necesar*** să atribuiți în mod explicit o valoare pentru acea coloană pentru a insera un nou rând în tabel.

**CONSTRÂNGERE FOREIGN KEY**

O cheie străină (FK) este o coloană sau o combinație de coloane care este utilizată pentru a stabili și impune o relație între datele dintre două tabele diferite.

Iată o diagramă care vine sa arate relația dintre tabelele angajați/**employees**/***сотрудники*** și *departamente/****departments/отделы***. Dacă o veti privi cu atenție, veți observa că coloana **dept\_id**  din tabelul angajați/**employees**/***сотрудники*** se potrivește cu coloana cheie primară **dept\_id** a tabelului *departamente/****departments/отделы***. Prin urmare, coloana **dept\_id** din tabelul angajaților **employees**  este cheia străină a tabelului de departamente/ **departments**.



În MySQL puteți crea o cheie străină definind o restricție FOREIGN KEY atunci când creați un tabel. Următoarea interogare stabilește o cheie străină pentru coloana ***dept\_id*** din tabelul angajaților/***employees*** care face referire la coloana ***dept\_id*** din tabelul departamente/*departments*.

**EXEMPLU:**

CREATE TABLE employees (

emp\_id INT NOT NULL PRIMARY KEY,

emp\_name VARCHAR(55) NOT NULL,

hire\_date DATE NOT NULL,

salary INT,

dept\_id INT,

FOREIGN KEY (dept\_id) REFERENCES departments(dept\_id)

);

Sau vedeti aici <https://www.softwaretestinghelp.com/mysql-foreign-key-constraint/>

**Altele vezi Anexa 1**

## CONSTRÂNGEREA CHECK

Constringerea CHECK este utilizată ***pentru a restricționa valorile/ ограничить значения*** ***care pot fi atribuite/plasate într-o coloană.***

De exemplu, gama de valori pentru o coloană de salariu poate fi limitată prin crearea unei constrângeri CHECK  care permite valori doar de la 3.000 la 10.000. Acest lucru previne/stopează/interzice introducerea salariilor dincolo de intervalul de salarii indicat de limita stabilită. Iată un exemplu:

**EXEMPLU:**

CREATE TABLE employees (

emp\_id INT NOT NULL PRIMARY KEY,

emp\_name VARCHAR(55) NOT NULL,

hire\_date DATE NOT NULL,

salary INT NOT NULL **CHECK** (salary >= 3000 AND salary <= 10000),

dept\_id INT,

FOREIGN KEY (dept\_id) REFERENCES departments(dept\_id)

);

**Notă:** MySQL nu suportă Constringerea CHECK a SQL. Cu toate acestea, clauza CHECK este analizată, dar ignorată de toate motoarele de stocare ale MySQL.

***MySQL не поддерживает ограничение SQL CHECK. Однако предложение CHECK анализируется, но игнорируется всеми механизмами хранения MySQL.***

1. **CONSTRINGERI UNICALE IN MariaDB**

**Care este o constringere unica în MariaDB?**

**O constrângere unică este un singur câmp sau combinație de câmpuri care**

**identifică în mod unic o înregistrare. Unele câmpuri pot conține valori NULL *dacă combinația de valori este unică.***

**Care este diferența dintre o constrângere unică și o cheie primară?**

| **Cheia primară** | **Constringerea unicală** |
| --- | --- |
| **Niciunul** din câmpurile care fac parte din cheia primară nu poate conține o valoare Null. | **Unele câmpuri** care fac parte din constrângerea UNIQUE pot conține valori NULL, dacă combinația de valori este unică. |

**CREAREA CONSTRINGERILOR UNICALE UTILIZÎND CREATE TABLE**

Sintaxa pentru crearea **Constringerilor unicale utilizînd CREATE TABLE** în MariaDB:

|  |  |
| --- | --- |
|  | CREATE TABLE table\_name  (    column1 datatype [ NULL | NOT NULL ],    column2 datatype [ NULL | NOT NULL ],    ..  **CONSTRAINT** constraint\_name UNIQUE (uc\_col1, uc\_col2, .. uc\_col\_n)  ); |

**table\_name**

**column1, column2**

**constraint\_name –** numele unical

**uc\_col1, uc\_col2, .. uc\_col\_n – coloanele component a constringerii unicale**

**EXEMPLU 1:**

|  |  |  |
| --- | --- | --- |
|  | CREATE TABLE websites  ( website\_id INT(11) PRIMARY KEY AUTO\_INCREMENT,  **website\_name** VARCHAR(25) NOT NULL,    server\_name VARCHAR(20),    creation\_date DATE,  **CONSTRAINT websites\_unique** UNIQUE (**website\_name**)  );  **EXEMPLU 2:** | |
|  | CREATE TABLE websites  ( website\_id INT(11) PRIMARY KEY AUTO\_INCREMENT,  **website\_name** VARCHAR(25) NOT NULL,  **server\_name** VARCHAR(20),    creation\_date DATE,    CONSTRAINT **websites\_server\_unique** UNIQUE (**website\_name, server\_name**)  ); |

**CREAREA CONSTRINGERILOR UNICALE UTILIZÎND ALTER TABLE**

Sintaxa pentru crearea **Constringerilor unicale utilizînd ALTER TABLE** în MariaDB:

|  |  |
| --- | --- |
|  | ALTER TABLE table\_name  ADD CONSTRAINT constraint\_name UNIQUE (column1, column2, .. column\_n); |

**table\_name –** nume table pentru modificare in care se introduce o constringere unicală

**constraint\_name –** numele constringerii unicale

**column1, column2, .. column\_n –** coloanele/atributele ce alcătuiescconstringerea unicală

**EXEMPLU 3:**

In acest exemplu adăugăm o constringere/restricție unică la un tabel existent în MariaDB folosind instrucțiunea ALTER TABLE.

|  |  |
| --- | --- |
|  | ALTER TABLE websites  ADD CONSTRAINT websites\_unique UNIQUE (website\_name); |

**EXEMPLU 4:**

Putem crea și o constrângere unică cu mai multe câmpuri,

|  |  |
| --- | --- |
|  | ALTER TABLE websites  ADD CONSTRAINT website\_server\_unique UNIQUE (website\_name, server\_name); |

**EXEMPLU 5:**

**Чтобы разрешить именование ограничения FOREIGN KEY и определить ограничение FOREIGN KEY для нескольких столбцов, используется следующий синтаксис SQL:**

**MySQL / SQL Server / Oracle / MS Access:**

CREATE TABLE Orders (  
    OrderID int NOT NULL,  
    OrderNumber int NOT NULL,  
    PersonID int,  
    PRIMARY KEY (OrderID),  
    CONSTRAINT FK\_PersonOrder FOREIGN KEY (PersonID)  
    REFERENCES Persons(PersonID)  
);

ALTER TABLE Orders  
ADD CONSTRAINT FK\_PersonOrder  
FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);

**SQL Server / Oracle / MS Access:**

ALTER TABLE Orders  
DROP CONSTRAINT FK\_PersonOrder;

**Anexa 1**

[**https://www-mysqltutorial-org.translate.goog/mysql-foreign-key/?\_x\_tr\_sl=en&\_x\_tr\_tl=ro&\_x\_tr\_hl=ro&\_x\_tr\_pto=wapp**](https://www-mysqltutorial-org.translate.goog/mysql-foreign-key/?_x_tr_sl=en&_x_tr_tl=ro&_x_tr_hl=ro&_x_tr_pto=wapp)

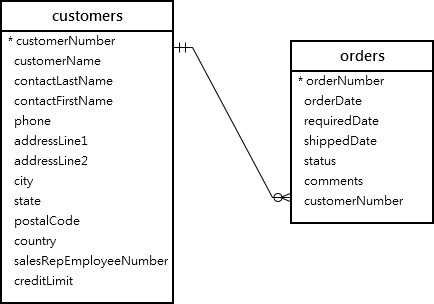
# FOREIGN KEY MySQL

**Rezumat** : în acest tutorial, veți afla despre **cheia străină MySQL** și cum să creați, să renunțați și să dezactivați o constrângere de cheie străină.

## Introducere în FOREIGN KEY MySQL

O cheie externă este o coloană sau un grup de coloane dintr-un tabel care face legătura cu o coloană sau un grup de coloane dintr-un alt tabel. Cheia externă impune constrângeri asupra datelor din tabelele aferente, ceea ce permite MySQL să mențină integritatea referențială.

Să aruncăm o privire la următoarele customers și orders tabelele din [baza de date exemplu](https://www-mysqltutorial-org.translate.goog/mysql-sample-database.aspx?_x_tr_sl=en&_x_tr_tl=ro&_x_tr_hl=ro&_x_tr_pto=wapp) .



În această diagramă, fiecare client poate avea zero sau mai multe comenzi și fiecare comandă aparține unui singur client.

Relația dintre customers masă și orders masă este unul la mai mulți. Și această relație este stabilită de cheia externă din orders tabelul specificat de customerNumber coloană.

Coloana customerNumber din orderstabel face legătura cu customerNumber coloana cheii primare din customers tabel.

Tabelul customers se numește  tabel părinte sau tabel de referință , iar orders tabelul este cunoscut ca tabel copil sau tabel de referință .

De obicei, coloanele cu cheie străină ale tabelului copil se referă adesea la coloanele   **primary key**  ale tabelului părinte.

Un tabel poate avea mai mult de o cheie străină în care fiecare cheie străină se referă la o cheie primară a diferitelor tabele părinte.

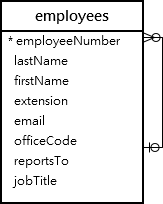
Odată ce o constrângere de cheie străină este aplicată, coloanele de cheie străină din tabelul copil trebuie să aibă rândul corespunzător în coloanele cheie părinte ale tabelului părinte sau valorile din aceste coloane cheie străină trebuie să fie NULL(vezi SET NULLexemplul de acțiune de mai jos).

De exemplu, fiecare rând din orderstabel are un customerNumber care există în customerNumber coloana customers tabelului. Mai multe rânduri din orderstabel pot avea același customerNumber.

### Cheie străină cu auto-referință

Uneori, tabelele pentru copii și părinți se pot referi la același tabel. În acest caz, cheia străină se referă la cheia primară din același tabel.

Consultați următorul employees tabel din [baza de date exemplu](https://www-mysqltutorial-org.translate.goog/mysql-sample-database.aspx?_x_tr_sl=en&_x_tr_tl=ro&_x_tr_hl=ro&_x_tr_pto=wapp) .



Coloana reportTo este o cheie străină care se referă la employeeNumber coloana care este cheia primară a employees tabelului.

Această relație permite employees tabelului să stocheze structura de raportare între angajați și manageri. Fiecare angajat raportează la zero sau un angajat și un angajat poate avea zero sau mai mulți subordonați.

Cheia străină de pe coloană reportToeste cunoscută ca o cheie străină recursivă sau auto-referință .

## FOREIGN KEY Sintaxa MySQL

Iată sintaxa de bază pentru definirea unei constrângeri de cheie străină în instrucțiunea [CREATE TABLE](https://www-mysqltutorial-org.translate.goog/mysql-create-table/?_x_tr_sl=en&_x_tr_tl=ro&_x_tr_hl=ro&_x_tr_pto=wapp)sau [ALTER TABLE](https://www-mysqltutorial-org.translate.goog/mysql-alter-table.aspx?_x_tr_sl=en&_x_tr_tl=ro&_x_tr_hl=ro&_x_tr_pto=wapp):

[CONSTRAINT constraint\_name]

FOREIGN KEY [foreign\_key\_name] (column\_name, ...)

REFERENCES parent\_table(colunm\_name,...)

[ON DELETE reference\_option]

[ON UPDATE reference\_option]

Limbajul codului: SQL (Structured Query Language) ( sql )

În această sintaxă:

Mai întâi, specificați numele constrângerii cheii străine pe care doriți să o creați după CONSTRAINTcuvântul cheie. Dacă omiteți numele constrângerii, MySQL generează automat un nume pentru constrângerea cheii externe.

În al doilea rând, specificați o listă de coloane de chei externe separate prin virgulă după FOREIGN KEYcuvintele cheie. Numele cheii externe este, de asemenea, opțional și este generat automat dacă îl omiteți.

În al treilea rând, specificați tabelul părinte urmat de o listă de coloane separate prin virgulă la care se referă coloanele cheii externe.

În cele din urmă, specificați modul în care cheia străină menține integritatea referențială dintre tabelele copil și tabelele părinte folosind clauzele ON DELETEși . ON UPDATEDetermină reference\_optionacțiunea pe care MySQL o va întreprinde atunci când valorile din coloanele cheii părinte sunt șterse ( ON DELETE) sau actualizate ( ON UPDATE).

MySQL are cinci opțiuni de referință: CASCADE, SET NULL, NO ACTION, RESTRICTși SET DEFAULT.

* CASCADE: dacă un rând din tabelul părinte este șters sau actualizat, valorile rândurilor care se potrivesc din tabelul copil sunt șterse sau actualizate automat.
* SET NULL: dacă un rând din tabelul părinte este șters sau actualizat, valorile coloanei (sau coloanelor) cheii externe din tabelul copil sunt setate la NULL.
* RESTRICT: dacă un rând din tabelul părinte are un rând potrivit în tabelul copil, MySQL respinge ștergerea sau actualizarea rândurilor din tabelul părinte.
* NO ACTION: este la fel ca RESTRICT.
* SET DEFAULT: este recunoscut de parserul MySQL. Cu toate acestea, această acțiune este respinsă atât de tabelele InnoDB, cât și de tabelele NDB.

De fapt, MySQL acceptă pe deplin trei acțiuni: RESTRICT, CASCADEși SET NULL.

Dacă nu specificați clauza ON DELETEși ON UPDATE, acțiunea implicită este RESTRICT.

## FOREIGN KEYExemple MySQL

Să [creăm o nouă bază de date](https://www-mysqltutorial-org.translate.goog/mysql-create-database/?_x_tr_sl=en&_x_tr_tl=ro&_x_tr_hl=ro&_x_tr_pto=wapp) numită fkdemopentru demonstrație.

CREATE DATABASE fkdemo;

USE fkdemo;

Limbajul codului: SQL (Structured Query Language) ( sql )

### RESTRICT& NO ACTIONacțiuni

În interiorul bazei de fkdemodate, creați două tabele categoriesși products:

CREATE TABLE categories(

categoryId INT AUTO\_INCREMENT PRIMARY KEY,

categoryName VARCHAR(100) NOT NULL

) ENGINE=INNODB;

CREATE TABLE products(

productId INT AUTO\_INCREMENT PRIMARY KEY,

productName varchar(100) not null,

categoryId INT,

CONSTRAINT fk\_category

FOREIGN KEY (categoryId)

REFERENCES categories(categoryId)

) ENGINE=INNODB;

Limbajul codului: SQL (Structured Query Language) ( sql )

În tabel este categoryIdcoloana productscheii externe care se referă la categoryIdcoloana din  categoriestabel.

Deoarece nu specificăm nicio clauză ON UPDATEși ON DELETE, acțiunea implicită este RESTRICTatât pentru operația de actualizare, cât și pentru operația de ștergere.

Următorii pași ilustrează RESTRICTacțiunea.

1) [Introduceți două rânduri](https://www-mysqltutorial-org.translate.goog/mysql-insert-multiple-rows/?_x_tr_sl=en&_x_tr_tl=ro&_x_tr_hl=ro&_x_tr_pto=wapp) în categoriestabel:

INSERT INTO categories(categoryName)

VALUES

('Smartphone'),

('Smartwatch');

Limbajul codului: SQL (Structured Query Language) ( sql )

2) [Selectați](https://www-mysqltutorial-org.translate.goog/mysql-select-statement-query-data.aspx?_x_tr_sl=en&_x_tr_tl=ro&_x_tr_hl=ro&_x_tr_pto=wapp) datele din categoriestabel:

SELECT \* FROM categories;

Limbajul codului: SQL (Structured Query Language) ( sql )

https://www.mysqltutorial.org/wp-content/uploads/2019/08/MySQL-Foreign-Key-categories-table.png

3) [Introduceți un nou rând](https://www-mysqltutorial-org.translate.goog/mysql-insert-statement.aspx?_x_tr_sl=en&_x_tr_tl=ro&_x_tr_hl=ro&_x_tr_pto=wapp) în productstabel:

INSERT INTO products(productName, categoryId)

VALUES('iPhone',1);

Limbajul codului: SQL (Structured Query Language) ( sql )

Funcționează pentru că categoryId1 există în categoriestabel.

4) Încercarea de a introduce un nou rând în productstabel cu o categoryId  valoare nu există în categoriestabel:

INSERT INTO products(productName, categoryId)

VALUES('iPad',3);

Limbajul codului: SQL (Structured Query Language) ( sql )

MySQL a emis următoarea eroare:

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`fkdemo`.`products`, CONSTRAINT `fk\_category` FOREIGN KEY (`categoryId`) REFERENCES `categories` (`categoryId`) ON DELETE RESTRICT ON UPDATE RESTRICT)

Limbajul codului: JavaScript ( javascript )

5) Actualizați valoarea din categoryIdcoloana din categoriestabel la 100:

UPDATE categories

SET categoryId = 100

WHERE categoryId = 1;

Limbajul codului: SQL (Structured Query Language) ( sql )

MySQL a emis această eroare:

Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails (`fkdemo`.`products`, CONSTRAINT `fk\_category` FOREIGN KEY (`categoryId`) REFERENCES `categories` (`categoryId`) ON DELETE RESTRICT ON UPDATE RESTRICT)

Limbajul codului: JavaScript ( javascript )

Din cauza RESTRICTopțiunii, nu puteți șterge sau actualiza categoryId 1, deoarece este referită de productId 1în productstabel.

### CASCADEacțiune

Acești pași ilustrează cum funcționează ON UPDATE CASCADEși [ON DELETE CASCADE](https://www-mysqltutorial-org.translate.goog/mysql-on-delete-cascade/?_x_tr_sl=en&_x_tr_tl=ro&_x_tr_hl=ro&_x_tr_pto=wapp)acțiunile.

1) [Aruncă](https://www-mysqltutorial-org.translate.goog/mysql-drop-table?_x_tr_sl=en&_x_tr_tl=ro&_x_tr_hl=ro&_x_tr_pto=wapp) tabelul products:

DROP TABLE products;

Limbajul codului: SQL (Structured Query Language) ( sql )

2) Creați productstabelul cu opțiunile ON UPDATE CASCADEși ON DELETE CASCADEpentru cheia externă:

CREATE TABLE products(

productId INT AUTO\_INCREMENT PRIMARY KEY,

productName varchar(100) not null,

categoryId INT NOT NULL,

CONSTRAINT fk\_category

FOREIGN KEY (categoryId)

REFERENCES categories(categoryId)

ON UPDATE CASCADE

ON DELETE CASCADE

) ENGINE=INNODB;

Limbajul codului: SQL (Structured Query Language) ( sql )

3) Introduceți patru rânduri în productstabel:

INSERT INTO products(productName, categoryId)

VALUES

('iPhone', 1),

('Galaxy Note',1),

('Apple Watch',2),

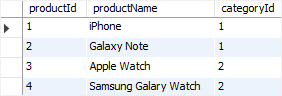
('Samsung Galary Watch',2);

Limbajul codului: SQL (Structured Query Language) ( sql )

4) Selectați datele din productstabel:

SELECT \* FROM products;

Limbajul codului: SQL (Structured Query Language) ( sql )



5) Actualizați categoryIdde la 1 la 100 în categoriestabel:

UPDATE categories

SET categoryId = 100

WHERE categoryId = 1;

Limbajul codului: SQL (Structured Query Language) ( sql )

6) Verificați actualizarea:

SELECT \* FROM categories;

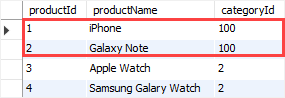
Limbajul codului: SQL (Structured Query Language) ( sql )

https://www.mysqltutorial.org/wp-content/uploads/2019/08/MySQL-Foreign-Key-categories-table-cascade.png

7) Obțineți date din productstabel:

SELECT \* FROM products;

Limbajul codului: SQL (Structured Query Language) ( sql )



După cum puteți vedea, două rânduri cu valoare 1în categoryIdcoloana productstabelului au fost actualizate automat 100din cauza ON UPDATE CASCADEacțiunii.

8) Ștergeți categoryId2 din categoriestabel:

DELETE FROM categories

WHERE categoryId = 2;

Limbajul codului: SQL (Structured Query Language) ( sql )

9) Verificați ștergerea:

SELECT \* FROM categories;

Limbajul codului: SQL (Structured Query Language) ( sql )

https://www.mysqltutorial.org/wp-content/uploads/2019/08/MySQL-Foreign-Key-categories-table-delete-cascade.png

10) Verificați productstabelul:

SELECT \* FROM products;

Limbajul codului: SQL (Structured Query Language) ( sql )

https://www.mysqltutorial.org/wp-content/uploads/2019/08/MySQL-Foreign-Key-products-table-delete-cascade.png

Toate produsele cu categoryId2 din productstabel au fost șterse automat din cauza ON DELETE CASCADEacțiunii.

### SET NULL acțiune

Acești pași ilustrează modul în care funcționează ON UPDATE SET NULLși ON DELETE SET NULLacțiunile.

1) Aruncă ambele categoriesși productstabele:

DROP TABLE IF EXISTS categories;

DROP TABLE IF EXISTS products;

Limbajul codului: SQL (Structured Query Language) ( sql )

2) Creați tabelele categoriesși :products

CREATE TABLE categories(

categoryId INT AUTO\_INCREMENT PRIMARY KEY,

categoryName VARCHAR(100) NOT NULL

)ENGINE=INNODB;

CREATE TABLE products(

productId INT AUTO\_INCREMENT PRIMARY KEY,

productName varchar(100) not null,

categoryId INT,

CONSTRAINT fk\_category

FOREIGN KEY (categoryId)

REFERENCES categories(categoryId)

ON UPDATE SET NULL

ON DELETE SET NULL

)ENGINE=INNODB;

Limbajul codului: SQL (Structured Query Language) ( sql )

Cheia externă din productstabel s-a schimbat în ON UPDATE SET NULLși ON DELETE SET NULLopțiuni.

3) Introduceți rânduri în categoriestabel:

INSERT INTO categories(categoryName)

VALUES

('Smartphone'),

('Smartwatch');

Limbajul codului: SQL (Structured Query Language) ( sql )

4) Introduceți rânduri în productstabel:

INSERT INTO products(productName, categoryId)

VALUES

('iPhone', 1),

('Galaxy Note',1),

('Apple Watch',2),

('Samsung Galary Watch',2);

Limbajul codului: SQL (Structured Query Language) ( sql )

5) Actualizare categoryIdde la 1 la 100 în categoriestabel:

UPDATE categories

SET categoryId = 100

WHERE categoryId = 1;

Limbajul codului: SQL (Structured Query Language) ( sql )

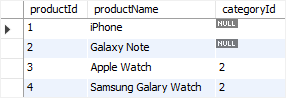
6) Verificați actualizarea:

SELECT \* FROM categories;

Limbajul codului: SQL (Structured Query Language) ( sql )

https://www.mysqltutorial.org/wp-content/uploads/2019/08/MySQL-Foreign-Key-categories-table-set-null.png

7) Selectați datele din productstabel:



Rândurile cu categoryId1 din productstabel au fost setate automat la NULLdatorită ON UPDATE SET NULLacțiunii.

8) Ștergeți cele categoryId2 din categoriestabel:

DELETE FROM categories

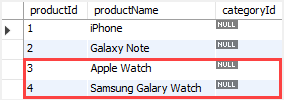
WHERE categoryId = 2;

Limbajul codului: SQL (Structured Query Language) ( sql )

9) Verificați productstabelul:

SELECT \* FROM products;

Limbajul codului: SQL (Structured Query Language) ( sql )



Valorile din categoryIdcoloana rândurilor cu categoryId2 din productstabel au fost setate automat la NULLdatorită ON DELETE SET NULLacțiunii.

## Eliminați constrângerile cheii externe MySQL

Pentru a elimina o constrângere de cheie străină, utilizați ALTER TABLEinstrucțiunea:

ALTER TABLE table\_name

DROP FOREIGN KEY constraint\_name;

Limbajul codului: SQL (Structured Query Language) ( sql )

În această sintaxă:

* Mai întâi, specificați numele tabelului din care doriți să aruncați cheia externă după ALTER TABLEcuvintele cheie.
* În al doilea rând, specificați numele constrângerii după DROP FOREIGN KEYcuvintele cheie.

Observați că acesta constraint\_nameeste numele constrângerii cheie străină specificată atunci când ați creat sau adăugat constrângerea cheii străine la tabel.

Pentru a obține numele de constrângere generat al unui tabel, utilizați SHOW CREATE TABLEinstrucțiunea:

SHOW CREATE TABLE table\_name;

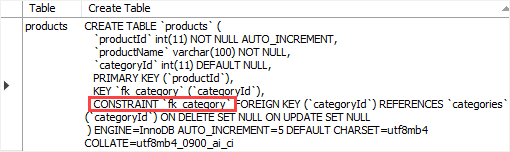
Limbajul codului: SQL (Structured Query Language) ( sql )

De exemplu, pentru a vedea cheile externe ale productstabelului, utilizați următoarea instrucțiune:

SHOW CREATE TABLE products;

Limbajul codului: SQL (Structured Query Language) ( sql )

Următorul este rezultatul declarației:



După cum puteți vedea clar din rezultat, productstabelul are o constrângere de cheie străină:fk\_category

Și această declarație elimină constrângerea cheii externe a productstabelului:

ALTER TABLE products

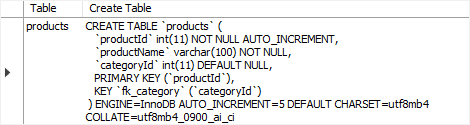
DROP FOREIGN KEY fk\_category;

Limbajul codului: SQL (Structured Query Language) ( sql )

Pentru a vă asigura că constrângerea cheii externe a fost eliminată, puteți vizualiza structura tabelului de produse:

SHOW CREATE TABLE products;

Limbajul codului: SQL (Structured Query Language) ( sql )



## Dezactivarea verificărilor cheii străine

Uneori, este foarte util să dezactivați verificările cheilor străine, de exemplu, atunci când [importați date dintr-un fișier CSV într-un tabel](https://www-mysqltutorial-org.translate.goog/import-csv-file-mysql-table/?_x_tr_sl=en&_x_tr_tl=ro&_x_tr_hl=ro&_x_tr_pto=wapp) . Dacă nu dezactivați verificările cheilor străine, trebuie să încărcați datele într-o ordine adecvată, adică mai întâi trebuie să încărcați datele în tabelele părinte și apoi în tabelele secundare, ceea ce poate fi plictisitor. Cu toate acestea, dacă dezactivați verificările cheii străine, puteți încărca datele în tabele în orice ordine.

Pentru a dezactiva verificările cheii străine, utilizați următoarea instrucțiune: