



CREATE TRIGGER

OLD va NEW

Triggerlar ro'yxati

Triggerni o'chirish

Trigger hosil qilishni o'rganamiz.

Umumiy qolip: `CREATE TRIGGER trigger_name {trigger_time} {trigger_event} ON table_name FOR EACH ROW trigger_body;` - `trigger_name` - triggerning nomi - `trigger_time` - triggerning qaysi paytda ishga tushishi. `trigger_time` - `BEFORE` yoki `AFTER` qiymatlardan bittasi bo'lishi mumkin. - `trigger_event` - triggerning qaysi hodisadan keyin yoki oldin ishga tushishi. `trigger_event` - `INSERT`, `UPDATE`, `DELETE` qiymatlardan bittasi bo'lishi mumkin. - `table_name` - jadval nomi - `trigger_body` - triggerning tana qismi

INSERT qilishdan oldin ishga tushadigan trigger (**BEFORE INSERT**):

```
CREATE TRIGGER trigger_name BEFORE INSERT ON table_name FOR EACH ROW trigger_body;
```

INSERT qilganidan keyin ishga tushadigan trigger (**AFTER INSERT**):

```
CREATE TRIGGER trigger_name AFTER INSERT ON table_name FOR EACH ROW trigger_body
```

OLD va NEW

MySQLda **OLD** va **NEW** deb nomlangan kalit so'zlar mavjud. Nomidan ham ko'rinib turibdi ESKI va YANGI degan ma'nolarni bildiradi. Xo'sh MySQL da bu kalit so'zlar nima vazifani bajaradi?

Misol bilan ko'raylik. Bazada `customerNumber=123` bo'lgan qator mavjud, uning `firstName` ustunidagi qiymati `'Harry'` ga teng. Biz shuni `firstName` degan ustunidagi qiymatni yangilamoqchimiz. **UPDATE** dan foydalanib quyidagicha so'rov yozamiz:

```
UPDATE customers SET firstName = 'Sardor' where customerNumber=123
```

Bu maydonning eski qiymati **'Harry'** edi. Eski qatorni ma'lumotlariga **OLD** kalit so'zi yordamida murojaat qilishimiz mumkin.

NEW bilan bo'lsa **UPDATE** bo'lganidan keyingi yangi qiymatli qatordagi ma'lumotlarga murojaat qilishimiz mumkin.

Show All

OLD || NEW

INSERT	No	Yes	
UPDATE	Yes	Yes	
DELETE	Yes	No	
Aa Trigger Event	☰ OLD	☰ NEW	

```
Misol:Yangi jadval hosil qilamiz. "orderdetails" jadvalidan biror qator o'ch
irilsa shuni "log_od" jadvaliga yozib boradigan trigger hosil qilamiz CREATE
TABLE log_od ( id int AUTO_INCREMENT PRIMARY KEY , orderNumber INT(11) NOT N
ULL, productCode VARCHAR(15), quantityOrdered INT(11), priceEach DECIMAL(10,
2), deleted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ); Trigger hosil qilamiz:
CREATE TRIGGER orderdetails_trigger AFTER DELETE ON orderdetails FOR EACH RO
W INSERT INTO log_od (orderNumber, productCode, quantityOrdered, priceEach)
VALUES (old.orderNumber, old.productCode, old.quantityOrdered, old.priceEac
h);
```

Trigger hosil qilganimizdan keyin, "**orderdetails**" jadvalidan biror qatorni **DELETE** qilamiz. **DELETE** qilganimizdan keyin "**log_od**" jadvalni **SELECT** qiladigan bo'lsak, u yerda "**orderdetails**" jadvalidan o'chirilgan qatordagi ma'lumotlarni ko'rish mumkin.

Triggerlar ro'yxati

SQL

Copy

```
SHOW TRIGGERS;
```

Triggerni o'chirish

```
DROP TRIGGER [IF EXISTS] trigger_name;
```