



# CURSOR

ERRORlarni ushlash

CURSOR

DECLARE CURSOR - e'lon qilish

OPEN - CURSORni ochish

FETCH - CURSORdan olish

CLOSE - CURSORni yopish

## ERRORlarni ushlash

Protsedurada qandaydir xatolik yuzaga kelganda, uni tegishli ravishda hal qilish juda muhim.

- Joriy bajarilayotgan blokni bajarilishini davom ettirish
- Joriy bajarilayotgan blokdan chiqib ketish
- Xatolik haqida xabar qilish

Protseduramizdan qandaydir xatolik sodir bo'lsa, nima qilish kerakligini protsedura tanasida ko'rsatib ketishimiz mumkin

```
DECLARE [action] HANDLER FOR {condition_value} {statement};
```

**action** - 2 xil qiymat qabul qilishi mumkin:

- **CONTINUE** - xatolik sodir bo'lganda davom ettirsin
- **EXIT** - xatolik sodir bo'lganda blokdan chiqib ketsin

CURSOR bilan ishlaganda, agarda CURSORda hechqanday qator topilmasa, qator topilmaganligini bilish uchun **NOT FOUND** - dan foydalanishimiz zarur

**condition\_value** - quyidagi qiymatni qabul qiladi

- `MySQL error code` - ya'ni mysqldagi errorlarning kodi
- `SQLWARNING` , `NOT FOUND` , `SQLException` - 3 tasidan birini qabul qiladi.

`statement` - xatolik sodir bo'lganda bajarilishi kerak bo'lgan ish.

```
DECLARE CONTINUE HANDLER FOR SQLException SET hasError = 1;
```

Agarda protseduramizda xatolik sodir bo'lib EXCEPTION qaytarsa, blokni bajarishda davom ettirish buyrug'i. `hasError` degan o'zgaruvchi e'lon qilib, agar xatolik sodir bo'lsa uni qiymatini o'zgartirib qo'yishimiz mumkin. Keyinchalik bu o'zgaruvchi orqali xatolik sodir bo'lgan yoki bo'lmaganligini bilib olishimiz mumkin

```
DECLARE CONTINUE HANDLER FOR NOT FOUND SET RowNotFound = 1;
```

Agarda protseduramizda `SELECT` orqali yoki `CURSOR` orqali ishlayotganimizda, jadvaldan birorta ham ma'lumot topilmasa, `NOT FOUND` dan foydalanib qator topilmaganini aniqlashimiz mumkin. Yuqorida agarda qator topilmasa `RowNotFound` o'zgaruvchi e'lon qilib, unga qiymat berildi. Shu qiymat orqali qator topilmaganini bilishimiz mumkin.

```
DECLARE CONTINUE HANDLER FOR 1062 SELECT 'Error, duplicate key occurred';
```

Protsedurada `Mysql error code` lar orqali xatolik sodir bo'lganda nima qilish kerak ekanligini ko'rsatishimiz mumkin.

## CURSOR

Saqlangan protsedura ichida natijalar to'plamini boshqarish uchun `CURSOR` dan foydalaniladi. `CURSOR` - so'rov orqali qaytarilgan natijalar ustida alohida-alohida amallar bajarishga imkon beradi

# DECLARE CURSOR - e'lon qilish

```
DECLARE cursor_name CURSOR FOR SELECT_statement;
```

Select qilib kerakli bo'lgan ma'lumotlarni olib biror nom ostida vaqtincha saqlab turish uchun **CURSOR** dan foydalanamiz.

## OPEN - CURSORni ochish

```
OPEN cursor_name;
```

Natijalardan foydalanish uchun **CURSOR** ni **OPEN** buyrug'i yordamida ochish zarur.

```
DECLARE finished INTEGER DEFAULT 0; DECLARE emailAddress var  
char(100) DEFAULT ""; -- declare cursor for employee email D  
ECLARE curEmail CURSOR FOR SELECT email FROM employees; -- d  
eclare NOT FOUND handler DECLARE CONTINUE HANDLER FOR NOT FO  
UND SET finished = 1; OPEN curEmail;
```

## FETCH - CURSORdan olish

**OPEN** orqali **CURSOR** ni ochganimizdan keyin, CURSORdagi natijalardan **FETCH** orqali alohida alohida qator qilib olib undan foydalanish mumkin.

```
FETCH cursor_name INTO variables list;
```

```
getEmail: LOOP FETCH curEmail INTO emailAddress; IF finished  
= 1 THEN LEAVE getEmail; END IF; -- build email list SET ema  
illist = CONCAT(emailAddress,";",emaillist); END LOOP getEma  
il;
```

# CLOSE - CURSORni yopish

**CLOSE** - buyrug'i orqali CURSORdan foydalanib bo'lganimizdan keyin uni yopishimiz mumkin. Agarda CURSOR boshqa ishlatilmaydigan bo'lsa CLOSE qilib yopish zarur.

```
CLOSE cursor_name;
```

## Misol

```
DELIMITER $$ CREATE PROCEDURE createEmailList ( INOUT emailList varchar(4000) ) BEGIN DECLARE finished INTEGER DEFAULT 0; DECLARE emailAddress varchar(100) DEFAULT ""; -- declare cursor for employee email DECLARE curEmail CURSOR FOR SELECT email FROM employees; -- declare NOT FOUND handler DECLARE CONTINUE HANDLER FOR NOT FOUND SET finished = 1; OPEN curEmail; getEmail: LOOP FETCH curEmail INTO emailAddress; IF finished = 1 THEN LEAVE getEmail; END IF; -- build email list SET emailList = CONCAT(emailAddress,";",emailList); END LOOP getEmail; CLOSE curEmail; END$$ DELIMITER ;
```

SQL

 Copy

```
SET @emailList = ""; CALL createEmailList(@emailList); SELECT @emailList; Natijani ko`rish uchun o'zingiz ushbu protsedurani ishga tushiring
```

```
mgerard@classicmodelcars.com;ykato@classicmodelcars.com;mnishi@classicmodelcars.com;tking@classicmodelcars.com;pmarsh@classicmodelcars.com;afixter@classicmodelcars.com;bjones@classicmodelcars.com;lbott@classicmodelcars.com;pcastillo@classicmodelcars.com;ghernande@classicmodelcars.com;lbondur@classicmodelcars.com;gvanauf@classicmodelcars.com;ftseng@classicmodelcars.com;spatterson@classicmodelcars.com;jfirrelli@classicmodelcars.com;lthompson@classicmodelcars.com;ljenning@classicmodelcars.com;abow@classicmodelcars.com;gbondur@classicmodelcars.com;wpatterson@classicmodelcars.com;jfirrelli@classicmodelcars.com;mpatterso@classicmodelcars.com;dmurphy@classicmodelcars.com;
```