

Saqlangan Protsedura parametrlari

Protsedura yana ham samarali va moslashuvchan ishlashligi uchun parametrlardan foydalaniladi.

```
[IN | OUT | INOUT] parameter_nomi tipi[(uzunligi)]
```

Mysql da parametrlar quyidagi 3 xil modeldan biri bo'lishi mumkin. Bular: **IN** , **OUT** , **INOUT**

IN paramterlar.

IN parametrlar - bular kiruvchi turidagi paramterlar desak yanglishmagan bo'lamiz. Qachonki biz protseduramizda IN dan paramterni ko'rsatgan bo'lsak, ushbu protsedurani ishga tushirayotganda, kiruvchi qiymatlar bilan chaqiriladi.

```
DELIMITER $$ CREATE PROCEDURE GetOfficeByCountry( IN countryName VARCHAR(255) ) BEG  
IN SELECT officeCode, city, state, country FROM offices WHERE country = countryNam  
e; END $$ DELIMITER ;
```

Yuqoridagi misolda **countryName** o'zgaruvchi, **IN** paramter sifatida ko'rsatilgan. Demak bu protsedurani chaqirib ishga tushirganimizda unga **countryName** uchun qiymat berishimiz shart.

```
CALL GetOfficeByCountry('USA'); Natija: +-----+-----+-----+-----  
----+ | officeCode | city | state | country | +-----+-----+-----+-----  
+-----+ | 1 | San Francisco | CA | USA | | 2 | Boston | MA | USA | | 3 | NYC |  
NY | USA | +-----+-----+-----+-----+ Agar protseduraga count  
ryName IN parametrini ko'rsatmasak xatolik yuz beradi: CALL GetOfficeByCountry(); E  
RROR 1318 (42000): Incorrect number of arguments for PROCEDURE dars2.GetOfficeByCou  
ntry; expected 1, got 0
```

OUT parametrlar

Quyidagi protsedura " **orders** " jadvalidan, buyurtmalarning **status** iga qarab, jami sonini qaytaradi

```
DELIMITER $$ CREATE PROCEDURE GetOrderCountByStatus ( IN orderStatus VARCHAR(25), O  
UT total INT ) BEGIN SELECT COUNT(orderNumber) INTO total FROM orders WHERE status  
= orderStatus; END$$ DELIMITER ;
```

Ushbu **GetOrderCountByStatus()** protsedura 2 ta paramterga ega:

- orderStatus** : **IN** parametr, ya'ni kiruvchi parametr
- total** : **OUT** parametr, ya'ni jami nechta buyurtma ekanligini o'zida saqlaydi

Endi ushbu protsedurani chaqirib ishga tushiramiz

```
CALL GetOrderCountByStatus('Shipped',@jami);
```

Ko'rib turganingizdek, protsedurani chaqirganimizda, 2 ta paramter berdik. " **orders** " jadvalidan **status** i "Shipped" bo'lgan buyrutmalarni jami nechtaligini, **@jami** degan o'zgaruvchiga qiymat qilib beradi. Yuqoridagi so'rovni amalga oshirsak bizga hechqanday natija chiqmaydi. Sababi natija **@jami** o'zgaruvchisiga qiymat qilib berilgan. **OUT** paramterlar - protseduraga paramter sifatida berilganda, protsedurani ichida shu parametrga qiymat berish imkoni mavjud. Protседura ishini tugatgandan keyin, **OUT** parametrning yangi qiymati bilan protsedura tashqarisida ham foydalanish mumkin. **IN** paramterlarda bunday qilishni imkoni yo'q.

Endi natijani ko'rish uchun:

```
SELECT @jami; Natija: +-----+ | @jami | +-----+ | 303 | +-----+
```

INOUT paramterlar

INOUT paramterlarni protseduraga kirish paramterlar sifatida ishlatish mumkin va protsedurani ichida ham INOUT paramterni qiymatini yangilash, uni ustida amallar bajarish va yangi qiymat berish mumkin. Protседura ishini tugatganda INOUT paramterning yangi qiymati bilan, protsedura tashqarisida ham foydalanish mumkin.

```
DELIMITER $$ CREATE PROCEDURE SetCounter( INOUT counter INT, IN inc INT ) BEGIN SET counter = counter + inc; END$$ DELIMITER ;
```

Yuqorida hosil qilgan protsedura, **INOUT** paramter va **IN** paramter qabul qiladi.

Endi shu protsedurani quyidagi hollarda ishga tushirib tekshiramiz:

```
-- @counter nomli o'zgaruvchiga 1 qiymati berildi SET @counter = 1; -- boshlang'ich qiymati -- SetCounter() protsedurasini ishga tushiramiz, unga ikkita paramter beramiz CALL SetCounter(@counter,1); -- @counter ni qiymati 1 ga teng edi. -- SET counter = counter + inc; bu amal bajarilgandan keyin, -- @counter ning yangi qiymati @counter = 1 + 1, ya'ni 2 ga teng bo'ladi -- Natijani chiqaramiz: SELECT @counter; --Natija: +-----+ | @counter | +-----+ | 2 | +-----+ -----  
----- --Davom etamiz va protsedurani yana ishga tushiramiz CALL SetCounter(@counter,1); -- @counter ni yangi qiymati 2 ga teng bo'lgan edi yuqoridagi so'rovda. -- SET counter = counter + inc; bu amal bajarilgandan keyin, -- @counter ning yangi qiymati @counter = 2 + 1, ya'ni 3 ga teng bo'ladi -- Natijani chiqaramiz: SELECT @counter; --Natija: +-----+ | @counter | +-----+ | 3 | +-----+ -----  
----- --Davom etamiz va protsedurani IN paramterga boshqa qiymat berib ishga tushiramiz CALL SetCounter(@counter,5); -- @counter ni yangi qiymati 3 ga teng bo'lgan edi yuqoridagi so'rovda. -- SET counter = counter + inc; bu amal bajarilgandan keyin, -- @counter ning yangi qiymati @counter = 3 + 5, ya'ni 8 ga teng bo'ladi -- Natijani chiqaramiz: SELECT @counter; --Natija: +-----+ | @counter | +-----+ | 8 | +-----+ Ko'rib turganingizdek natija 8 ga teng.
```