

Урок 12.2. Параметры в хранимых процедурах

| | |
|----------------------------|---|
| IN-параметры | 1 |
| OUT-параметры | 4 |
| INOUT-параметры | 6 |
| Сравнение типов параметров | 8 |
| Практическая работа | 9 |

IN-параметры



Важно!

База данных с доступом на запись:

hostname: ich-edit.edu.itcareerhub.de

MYSQL_USER: ich1

MYSQL_PASSWORD: ich1_password_lovedbs

В MySQL хранимые процедуры могут принимать параметры для обмена данными между вызывающей программой и самой процедурой. Параметры позволяют процедурам быть более гибкими и полезными, так как они могут работать с различными входными значениями и возвращать результаты. В MySQL поддерживаются три типа параметров: **IN**, **OUT** и **INOUT**.



IN — это входные параметры, которые передаются в процедуру при ее вызове, и их значения используются в процессе выполнения процедуры.

Внутри процедуры значение IN-параметра можно использовать, но его нельзя изменить — он только для чтения.

Характеристики IN-параметров:

- Значения передаются в процедуру.
- Значения параметров **нельзя изменять** внутри процедуры (они только для чтения).
- Если значение IN-параметра изменено, то это изменение не передается обратно вызывающей программе.



Пример IN-параметра

Unset

```
DELIMITER $$
```

```
CREATE PROCEDURE get_employee_name(IN emp_id INT)
BEGIN
    SELECT name FROM employees WHERE id = emp_id;
END $$
DELIMITER ;
```

В этом примере процедура `get_employee_name` принимает входной параметр `emp_id` и использует его для поиска имени сотрудника.

Вызов процедуры

Unset

```
CALL get_employee_name(1);
```

OUT-параметры



OUT — это выходные параметры, которые используются для того, чтобы **возвращать значение из процедуры**.

Значение OUT-параметра присваивается внутри процедуры и возвращается вызывающей программе после завершения работы процедуры. По умолчанию параметр не имеет начального значения, и его нужно явно присвоить внутри процедуры.

Характеристики OUT-параметров:

- Значение назначается в процедуре и **возвращается** после ее выполнения.
- Параметр не используется для входных данных.
- Вызывающая программа может получить значение OUT-параметра после завершения процедуры.



Пример OUT-параметра

```
Unset
DELIMITER $$

CREATE PROCEDURE get_employee_salary(IN emp_id INT, OUT emp_salary INT)
BEGIN
    SELECT salary INTO emp_salary FROM employees WHERE id = emp_id;
END $$

DELIMITER ;
```

В этом примере процедура `get_employee_salary` возвращает зарплату сотрудника через параметр `emp_salary`.

Вызов процедуры:

Unset

```
SET @salary = 0; -- Инициализируем переменную  
CALL get_employee_salary(1, @salary); -- Вызываем процедуру и передаем  
OUT-параметр  
SELECT @salary; -- Просматриваем возвращенное значение
```

INOUT-параметры



INOUT — это двунаправленные параметры, которые могут быть использованы как для передачи входных данных в процедуру, так и для возврата данных из нее.

Сначала процедура получает значение параметра, затем может изменить его, и измененное значение возвращается вызывающей программе.

Характеристики INOUT-параметров:

- Параметр используется как для входных, так и для выходных данных.
- Начальное значение передается в процедуру.
- Внутри процедуры параметр может быть изменен, и его новое значение возвращается вызывающей программе.



Пример INOUT-параметра

Unset

```
DELIMITER $$
```

```
CREATE PROCEDURE update_employee_salary(INOUT emp_salary INT)
BEGIN
    SET emp_salary = emp_salary * 1.2; -- Увеличиваем зарплату на 20%
END $$

DELIMITER ;
```

В этом примере процедура `update_employee_salary` принимает параметр `emp_salary`, увеличивает его значение на 10%, и возвращает измененное значение.

Вызов процедуры:

Unset

```
SET @salary = 5000; -- Инициализируем переменную  
CALL update_employee_salary(@salary); -- Передаем значение и получаем  
обновленное  
SELECT @salary; -- Просматриваем измененное значение
```

Сравнение типов параметров

| Тип параметра | Передаёт в процедуру | Возвращает из процедуры | Может изменяться внутри процедуры |
|---------------|----------------------|-------------------------|-----------------------------------|
| IN | Да | Нет | Нет |
| OUT | Нет | Да | Да |
| INOUT | Да | Да | Да |

Практическая работа

1. IN параметр

1. Создайте таблицу `products` с колонками `id` (INT) и `product_name` (VARCHAR), `price`.
2. Вставьте несколько записей в таблицу.
3. Создайте хранимую процедуру с IN-параметром для поиска имени товара по его идентификатору.
4. Вызовите эту процедуру и проверьте результат.

Решение

Unset

```
CREATE PROCEDURE get_product_name(IN product_id INT)
BEGIN
    SELECT product_name FROM products WHERE id = product_id;
END;
```

2. OUT параметр

1. Создайте таблицу `employees` с колонками `id` (INT), `name` (VARCHAR), `monthly_salary` (INT).
2. Вставьте несколько записей в таблицу.
3. Создайте хранимую процедуру, которая возвращает годовую зарплату (ежемесячная зарплата * 12) через OUT-параметр.
4. Вызовите процедуру и проверьте результат, используя переменную для OUT-параметра.

Решение

Unset

```
CREATE PROCEDURE get_annual_salary(IN emp_id INT, OUT annual_salary INT)
BEGIN
    SELECT monthly_salary * 12 INTO annual_salary FROM employees WHERE id =
emp_id;
END;
```

3. INOUT параметр

Создайте хранимую процедуру, которая принимает значение бонуса для сотрудника, увеличивает его на 15% и возвращает новое значение через INOUT-параметр.

1. Создайте переменную для хранения значения бонуса.
2. Создайте хранимую процедуру с INOUT-параметром, которая увеличивает бонус на 15%.
3. Вызовите процедуру с начальными значениями и проверьте измененные значения.

Решение

Unset

```
CREATE PROCEDURE increase_bonus(INOUT bonus_amount DECIMAL(10, 2))
```

```
BEGIN
```

```
    SET bonus_amount = bonus_amount * 1.15;
```

```
END;
```

```
SET @bonus = 1000; CALL increase_bonus(@bonus); SELECT @bonus; Ожидаемое  
значение: 1150
```