

Урок 11.2. Функции

Определение пользовательской функции	2
Примеры пользовательских функций	3
Использование пользовательских функций	5
Практическая работа	6

Определение пользовательской функции



Важно!

База данных с доступом на запись:

hostname: ich-edit.edu.itcareerhub.de

MYSQL_USER: ich1

MYSQL_PASSWORD: ich1_password_ilovedbs



Пользовательская функция — это функция, которая создается с помощью команды **CREATE FUNCTION**.

Они могут использоваться в SQL-запросах так же, как и встроенные функции.



Пример синтаксиса для создания функции

Unset

```
CREATE FUNCTION function_name (parameter1 datatype, parameter2 datatype, ...)  
RETURNS return_datatype  
DETERMINISTIC  
BEGIN  
    -- Function body  
    RETURN return_value;  
END;
```

Примеры пользовательских функций

1. Функция для вычисления квадратного корня

```
Unset
DELIMITER //

CREATE FUNCTION square_root(x DOUBLE)
RETURNS DOUBLE
DETERMINISTIC
BEGIN
    RETURN SQRT(x);
END //

DELIMITER ;
```

Примечание: В этом примере функция `square_root` возвращает квадратный корень из переданного значения.

2. Функция для вычисления возраста на основе даты рождения

```
Unset
DELIMITER //

CREATE FUNCTION calculate_age(birthdate DATE)
RETURNS INT
DETERMINISTIC
BEGIN
    RETURN TIMESTAMPDIFF(YEAR, birthdate, CURDATE());
END //

DELIMITER ;
```

Примечание: Эта функция `calculate_age` принимает дату рождения и возвращает возраст в годах.

3. Функция для получения приветственного сообщения

```
Unset
DELIMITER //

CREATE FUNCTION greet_user(name VARCHAR(100))
RETURNS VARCHAR(255)
DETERMINISTIC
BEGIN
    RETURN CONCAT('Hello, ', name, '!');
END //

DELIMITER ;
```

Примечание: Функция `greet_user` возвращает приветственное сообщение, используя имя, переданное в параметре.

Использование пользовательских функций

После создания пользовательской функции её можно использовать в SQL-запросах:

Функция	Использование
square_root	SELECT square_root(25);
calculate_age	SELECT calculate_age('1990-01-01');
greet_user	SELECT greet_user('Alice');

Важное свойство функций

- **DETERMINISTIC.** Указывает, что функция всегда возвращает один и тот же результат для одинаковых входных данных. Это улучшает производительность, так как MySQL может кэшировать результаты функции.
- **NON-DETERMINISTIC.** Если функция может возвращать разные результаты для одинаковых входных данных (например, использование функции NOW()), то её нужно указать как NON-DETERMINISTIC.

Практическая работа

1. Изменение регистра

Создайте функцию для перевода текста в верхний регистр. Функция принимает строку и возвращает её в верхнем регистре.

Решение

Unset

```
DELIMITER //
```

```
CREATE FUNCTION to_uppercase(input_text VARCHAR(255))
```

```
RETURNS VARCHAR(255)
```

```
DETERMINISTIC
```

```
BEGIN
```

```
    RETURN UPPER(input_text);
```

```
END //
```

```
DELIMITER ;
```

2. Проверка четности

Функция для проверки, является ли число четным. Функция принимает целое число и возвращает 1, если оно четное, и 0, если нечетное.

Решение

Unset

```
DELIMITER //
```

```
CREATE FUNCTION is_even(number INT)
```

```
RETURNS BOOLEAN
```

```
DETERMINISTIC
```

```
BEGIN
```

```
    RETURN number % 2 = 0;
```

```
END //
```

```
DELIMITER;
```