

BAB 13

FUNCTION

13.1. Tujuan Pembelajaran

1. Membuat function
2. Mengetahui cara kerja function
3. Mengetahui kegunaan function

13.2. Dasar Teori

Pengenalan Function

Fungsi tersimpan adalah program tersimpan jenis khusus yang mengembalikan nilai tunggal. Biasanya, kalian menggunakan fungsi tersimpan untuk mengenkapsulasi rumus umum atau aturan bisnis yang dapat digunakan kembali di antara pernyataan SQL atau program tersimpan.

Berbeda dari stored procedure, kalian bisa menggunakan stored function dalam pernyataan SQL di mana pun ekspresi digunakan. Ini membantu meningkatkan keterbacaan dan pemeliharaan kode prosedural.

Untuk membuat fungsi tersimpan, kalian menggunakan pernyataan CREATE FUNCTION.

13.3. Software

- XAMPP
- MySQL

13.4. Function

13.4.1 Latihan membuat function

Sintaks MySQL CREATE FUNCTION

Berikut ini mengilustrasikan sintaks dasar untuk membuat fungsi tersimpan baru:

```

DELIMITER $$

CREATE FUNCTION function_name(
    param1,
    param2,...
)
RETURNS datatype
[NOT] DETERMINISTIC
BEGIN
    -- statements
END $$

DELIMITER ;

```

Gambar 13.1 Sintak Dasar Membuat Stored Function

Dalam sintaks ini:

1. Pertama, tentukan nama fungsi tersimpan yang ingin kalian buat setelah kata kunci CREATE FUNCTION.
2. Kedua, daftarkan semua parameter fungsi yang disimpan di dalam tanda kurung diikuti dengan nama fungsi. Secara default, semua parameter adalah parameter IN. kalian tidak dapat menentukan pengubah IN , OUT atau INOUT ke parameter
3. Ketiga, tentukan tipe data dari nilai kembalian dalam pernyataan RETURNS, yang bisa berupa tipe data MySQL yang valid.
4. Keempat, tentukan apakah suatu fungsi bersifat deterministik atau tidak menggunakan kata kunci DETERMINISTIK.

Fungsi **deterministik** selalu mengembalikan hasil yang sama untuk parameter input yang sama sedangkan fungsi **non-deterministik** mengembalikan hasil yang berbeda untuk parameter input yang sama. Jika kalian tidak menggunakan DETERMINISTIC atau NOT DETERMINISTIC, MySQL menggunakan opsi NOT DETERMINISTIC secara default.

5. Kelima, tulis kode di badan fungsi yang disimpan di blok BEGIN END. Di dalam bagian body, kalian perlu menentukan setidaknya satu pernyataan RETURN. Pernyataan RETURN mengembalikan nilai ke program pemanggilan. Setiap kali pernyataan RETURN tercapai, eksekusi fungsi yang disimpan segera diakhiri.

Contoh MySQL CREATE FUNCTION

Mari kita ambil contoh membuat fungsi tersimpan. Kita akan menggunakan tabel customers di database yang sudah pernah kita buat sebelumnya.

```
DELIMITER $$

CREATE FUNCTION CustomerLevel(
    credit INT(5)
)
RETURNS VARCHAR(20)
DETERMINISTIC
BEGIN
    DECLARE customerLevel VARCHAR(20);

    IF credit > 50000 THEN
        SET customerLevel = 'PLATINUM';
    ELSEIF (credit >= 50000 AND
        credit <= 10000) THEN
        SET customerLevel = 'GOLD';
    ELSEIF credit < 10000 THEN
        SET customerLevel = 'SILVER';
    END IF;
    -- return the customer level
    RETURN (customerLevel);
END$$
DELIMITER ;
```

Gambar 13.2 Membuat Stored Function CustomerLevel Menggunakan Conditional

Setelah fungsi dibuat, kalian dapat melihatnya di bagian kiri MySQL di bawah bagian



Gambar 13.3 List Fungsi Yang Sudah Dibuat

fungsi:

Atau kalian dapat melihat semua fungsi yang disimpan dalam database saat ini dengan menggunakan SHOW FUNCTION STATUS sebagai berikut:

```
SHOW FUNCTION STATUS
WHERE db = 'subquery';
```

Gambar 13.4 Melihat Status Fungsi dari Database Subquery

Outputnya akan menampilkan:

Db	Name	Type	Definer	Modified	Created	Security_type
subquery	CustomerLevel	FUNCTION	root@localhost	2022-12-23 05:35:45	2022-12-23 05:35:45	DEFINER

Gambar 13.5 Tampilan Status Fungsi Database SubQuery

Memanggil fungsi tersimpan dalam pernyataan SQL

Pernyataan berikut menggunakan fungsi tersimpan CustomerLevel:

```
SELECT
    customerName,
    CustomerLevel(creditLimit)
FROM
    customers
ORDER BY
    customerName;
```

Gambar 13.6 Menampilkan Fungsi CustomerLevel

Outputnya akan menampilkan:

customerName ▲ 1	CustomerLevel(creditLimit)
Alex	NULL
Dani	NULL
Dwi	NULL
Edward	PLATINUM
Pratama	NULL
Raihan	NULL
Rio	PLATINUM
Sandra	PLATINUM
Sandrina	NULL
Xein	PLATINUM
Zayn	NULL
Zeith	PLATINUM

Gambar 13.7 Tampilan Fungsi CustomerLevel

Memanggil fungsi tersimpan dalam prosedur tersimpan

Pernyataan berikut membuat stored procedure baru yang memanggil fungsi tersimpan CustomerLevel() :

```

DELIMITER $$

CREATE PROCEDURE GetCustomerLevel(
    IN customerNo INT,
    OUT customerLevel VARCHAR(20)
)
BEGIN

    DECLARE credit INT(5) DEFAULT 0;

    -- get credit limit of a customer
    SELECT
        creditLimit
    INTO credit
    FROM customers
    WHERE
        customerNumber = customerNo;

    -- call the function
    SET customerLevel = CustomerLevel(credit);
END$$

DELIMITER ;

```

Gambar 13.8 Membuat Procedure GetCustomerLevel

Berikut ini mengilustrasikan cara memanggil prosedur tersimpan GetCustomerLevel() :

```

CALL GetCustomerLevel(42225,@customerLevel);
SELECT @customerLevel;

```

Gambar 13.9 Memanggil Procedure GetCustomerLevel dengan Parameter

Outputnya akan menampilkan:

```

@customerLevel
PLATINUM

```

Gambar 13.10 Tampilan Procedure Yang Sesuai dengan Parameter

Penting untuk diperhatikan bahwa jika fungsi tersimpan berisi pernyataan SQL yang meminta data dari tabel, maka Anda tidak boleh menggunakannya dalam pernyataan SQL lainnya; jika tidak, fungsi yang disimpan akan memperlambat kecepatan queri.

13.5. Latihan

1. Buatlah sebuah stored function dengan nama GetCustomerPrice!
2. Apakah efisien menggunakan stored function? Jelaskan menurut pendapat kalian!