

Advanced MySQL Topics

Maulana Hafidz Ismail

10 Oktober 2025

Contents

1	Functions and Triggers	1
1.1	Pengantar Functions dan Stored Procedures	1
1.2	Manfaat dan Organisasi Kode	1
1.3	Perbedaan Antara Functions dan Stored Procedures	1
1.4	Variabel dalam MySQL	1
1.5	Deklarasi Variabel	2
1.6	Mengatur Nilai Variabel	2
1.7	Contoh Penggunaan	2
2	Database Optimization	2

1 Functions and Triggers

1.1 Pengantar Functions dan Stored Procedures

- Functions dan Stored Procedures memungkinkan Anda untuk menggunakan kembali kode dalam proyek database, sehingga tidak perlu menetik kode yang sama berulang kali.
- Contoh penggunaan adalah pada perusahaan Lucky Shrub yang menggunakan metode ini untuk memeriksa data stok di tabel produk mereka.
- A function in MySQL is also called a stored function.

1.2 Manfaat dan Organisasi Kode

- Tujuan utama dari functions dan stored procedures adalah untuk mengenkapsulasi kode, sehingga anda dapat memanggil blok kode untuk melakukan operasi tertentu dengan menggunakan nama pengidentifikasi.
- Keduanya membuat kode lebih konsisten, terorganisir, dan memperkenalkan reusability, yang memudahkan penggunaan dan pemeliharaan kode.

1.3 Perbedaan Antara Functions dan Stored Procedures

- Functions selalu mengembalikan nilai, sedangkan stored procedures tidak selalu mengembalikan nilai.
- Function hanya dapat memiliki input parameters, sedangkan stored procedures dapat memiliki input dan output parameters.
- Contoh sintaks untuk membuat stored procedure:

```
CREATE PROCEDURE procedure_name AS
BEGIN
    -- logic here
END;
```

- Contoh sintaks untuk menggunakan function:

```
SELECT MOD(x, y)
```

1.4 Variabel dalam MySQL

- Variabel adalah placeholder yang menyimpan nilai yang dapat berubah sesuai kebutuhan query. Variabel digunakan untuk mengoper nilai antara pernyataan SQL atau antara prosedur dan pernyataan SQL.

- Variabel dapat dibuat di dalam atau di luar stored procedure dan di dalam atau di luar pernyataan SELECT.
- Contoh sintaks untuk membuat variabel:

```
SET @variabel_name = value;
```

1.5 Deklarasi Variabel

- DECLARE digunakan untuk mendeklarasikan variabel lokal di dalam stored procedure. Variabel ini hanya dapat digunakan dalam konteks prosedur tersebut. Contoh:

```
DECLARE variable_name datatype;
```

- SET @ digunakan untuk mendapatkan variabel sesi (session variable) yang dapat diakses di seluruh sesi MySQL. Variabel ini diawali dengan simbol @. Contoh:

```
SET @variable_name = value;
```

1.6 Mengatur Nilai Variabel

1. SET digunakan untuk memberikann nilai kepada variabel yang telah dideklarasikan. Ini adalah cara yang umum untuk menginisialisasi variabel. Contoh:

```
SET variable_name = value;
```

2. Operator (:=) digunakan untuk menetapkan nilai ke variabel, baik untuk variabel lokal yang dideklarasikan dengan DECLARE maupun untuk variabel sesi yang diawali dengan @. Contoh:

```
variable_name := value;
```

1.7 Contoh Penggunaan

```
CREATE PROCEDURE example_procedure()
BEGIN
    DECLARE total_cost DECIMAL(10,2);
    SET total_cost = 100.00;
    SET @discount = 10;
    total_cost := total_cost - @discount;
END;
```

2 Database Optimization