

## View, Function dan Trigger

### Definisi View

Sebuah view dalam basis data adalah **tabel virtual** yang **tidak memiliki nilai fisik**, tetapi merupakan tabel yang dihasilkan **berdasarkan hasil dari suatu query**. View sering kali digunakan untuk **mengakses** sekelompok bagian data yang berulang dan kompleks sehingga **tidak perlu menulis ulang query yang sama berulang-ulang**. Selain itu, view juga dapat digunakan untuk **meningkatkan keamanan** dengan **membatasi akses** pengguna hanya pada data tertentu.

### Query Untuk Membuat View

#### 1. Query Dasar View

```
CREATE VIEW nama_view AS

SELECT kolom1, kolom2, ...

FROM nama_tabel

WHERE kondisi;
```

#### 2. Contoh

Misalnya, kita ingin membuat view yang berisi daftar peminjaman buku dari perpustakaan yang belum dikembalikan, berikut adalah contoh query yang bisa digunakan:

```
CREATE VIEW peminjaman_belum_kembali AS

SELECT a.nama, b.judul, p.tanggal_pinjam

FROM anggota a

JOIN peminjaman p ON a.id_anggota = p.id_anggota

JOIN buku b ON p.id_buku = b.id_buku

WHERE p.tanggal_kembali IS NULL;
```

Dengan view di atas, kita bisa dengan mudah mendapatkan daftar peminjaman yang belum dikembalikan hanya dengan melakukan **SELECT \* FROM** `peminjaman_belum_kembali;`

### 3. Keterbatasan View

Meski view sangat membantu, ada beberapa keterbatasan, tergantung pada sistem manajemen basis data yang digunakan. Beberapa sistem manajemen basis data tidak memungkinkan perubahan data melalui view. Selain itu, tidak semua query bisa diubah menjadi view. Misalnya, dalam MySQL, query yang mengandung klausa ORDER BY, GROUP BY, atau variabel tidak bisa diubah menjadi view. Selalu periksa dokumentasi sistem manajemen basis data Anda untuk mengetahui keterbatasan spesifiknya.

## Definisi Trigger

Trigger adalah prosedur yang disimpan yang secara otomatis **dijalankan atau diaktifkan** ketika suatu **peristiwa** tertentu **terjadi dalam database**. Peristiwa ini bisa berupa operasi **Insert, Update, atau Delete**.

Misalnya, Anda dapat memiliki trigger yang dijalankan sebelum data ditambahkan ke tabel, yang memeriksa apakah data baru tersebut memenuhi kriteria tertentu. Jika tidak, trigger dapat membatalkan penambahan data tersebut.

## Query Untuk Membuat Trigger

### 1. Query dasar Trigger

```
CREATE TRIGGER trigger_name  
  
trigger_time trigger_event  
  
ON table_name FOR EACH ROW  
  
trigger_body;
```

#### Penjelasan Query :

- **trigger\_name** : Nama dari trigger.
- **trigger\_time** : Ini bisa BEFORE atau AFTER, yaitu trigger dijalankan sebelum atau setelah peristiwa.

- **trigger\_event** : Peristiwa yang menyebabkan trigger diaktifkan. Ini bisa INSERT, UPDATE, atau DELETE.
- **table\_name** : Nama tabel di mana trigger akan diaktifkan.
- **trigger\_body** : Blok kode yang dijalankan ketika trigger diaktifkan.

## 2. Contoh Trigger

Contoh kali ini akan menggunakan database perpustakaan yang sudah pernah dibuat sebelumnya.

```
DELIMITER //
```

```
CREATE TRIGGER trg_insert_peminjaman
```

```
AFTER INSERT
```

```
ON peminjaman
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    -- contoh kode trigger, misalnya mengurangi stok buku saat
```

```
    peminjaman
```

```
    UPDATE buku SET stok = stok - 1 WHERE id_buku = NEW.id_buku;
```

```
END;
```

```
// DELIMITER;
```

## Definisi Function

Fungsi dalam MySQL adalah **rutinitas** yang **disimpan** yang **mengambil** beberapa argumen (juga disebut parameter) dan mengembalikan nilai. Fungsi dapat digunakan dalam ekspresi dan karenanya mereka dapat digunakan dalam berbagai bagian dari pernyataan SQL.

## Query Untuk Membuat Function

### 1. Query Dasar Function

```
CREATE FUNCTION function_name ([parameter[,...]])  
  
RETURNS type [characteristic ...]  
  
routine_body;
```

#### Penjelasan Query :

- **function\_name** : Nama dari fungsi.
- **parameter** : Daftar parameter yang dipisahkan dengan koma.
- **type** : Tipe data yang dikembalikan oleh fungsi.
- **characteristic** : Karakteristik dari fungsi, bisa DETERMINISTIC, SQL DATA ACCESS, dan lainnya.
- **routine\_body** : Blok kode yang dijalankan saat fungsi dipanggil.

### 2. Contoh Function

```
CREATE FUNCTION jumlah_buku() RETURNS INT  
  
BEGIN  
  
    DECLARE total INT;  
  
    SELECT COUNT(*) INTO total FROM buku;  
  
    RETURN total;  
  
END;
```

```
CREATE PROCEDURE tampilkan_anggota()  
  
BEGIN  
  
    SELECT * FROM anggota;  
  
END;
```

—