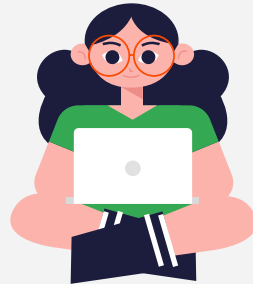


Sqllyze Fun Learning



View, Function dan Trigger

Scroll
Down





Apa itu 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 Dasar View



```
CREATE VIEW nama_view AS
```

```
SELECT kolom1, kolom2, ...
```

```
FROM nama_tabel
```

```
WHERE kondisi;
```

Contoh :

```
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;
```



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



Apa itu 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**.



Query untuk Membuat Trigger



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

Penjelasan

- **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.



Contoh Trigger



```
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;
```



Apa itu 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 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.



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;
```



Database Sistem E-Commerce (lanj.)



6. Menambahkan data ke tabel produk

```
INSERT INTO produk (id, nama_produk, harga, stok)
VALUES (1, 'Laptop XYZ', 15000.00, 10);
```

7. Menambahkan data ke tabel pelanggan

```
INSERT INTO pelanggan (id, first_name, last_name, email, alamat)
VALUES (1, 'Andi', 'Wijaya', 'andi.wijaya@example.com', 'Jl. Kebon Jeruk 123');
```

8. Menambahkan data ke tabel transaksi

```
INSERT INTO transaksi (id, pelanggan_id, produk_id, jumlah, total,
tanggal_transaksi)
VALUES (1, 1, 1, 2, 30000.00, '2023-05-01');
```

9. Menampilkan data dari tabel produk

```
SELECT * FROM produk;
```

10. Menampilkan data dari tabel pelanggan

```
SELECT * FROM pelanggan;
```

11. Menampilkan data dari tabel transaksi

```
SELECT * FROM transaksi;
```



Database Sistem E-Commerce (lanjt.)



Penjelasan Query :

- Membuat database baru dengan nama "e_commerce".
- Menggunakan database "e_commerce" untuk perintah selanjutnya.
- Membuat tabel "produk" dengan kolom id, nama_produk, harga, dan stok. Kolom id merupakan primary key.
- Membuat tabel "pelanggan" dengan kolom id, first_name, last_name, email, dan alamat. Kolom id merupakan primary key, dan kolom email memiliki constraint UNIQUE.
- Membuat tabel "transaksi" dengan kolom id, pelanggan_id, produk_id, jumlah, total, dan tanggal_transaksi. Kolom id merupakan primary key, sedangkan kolom pelanggan_id dan produk_id merupakan foreign key yang merujuk ke tabel pelanggan dan tabel produk.