

TRIGGER

HANI IRMAYANTI, S.KOM

TRIGGER

- ▶ Trigger adalah object di dalam database yang berasosiasi dengan suatu table.
- ▶ Trigger akan diaktivasi ketika table tersebut dikenai *event* tertentu.
- ▶ Event yang dimaksud adalah kejadian yang menimpa table seperti penambahan, perubahan, maupun penghapusan data.
- ▶ Trigger berfungsi untuk mengeksekusi satu atau sekumpulan perintah sql secara otomatis, ketika kita menambah, mengubah, maupun menghapus sebuah baris data dalam table.
- ▶ Trigger dapat menguji validitas data yang akan dimasukkan maupun yang akan digunakan untuk melakukan perubahan.
- ▶ Trigger dapat mengubah nilai kolom pada table lain.

Sebelum membuat trigger, buat table histori_buku, untuk mencatat semua yang transaksi atau aksi yang terjadi pada table buku.

```
create table histori_buku(  
buku_isbn char(13),  
buku_judul varchar (75),  
penerbit_id char(4),  
buku_tglterbit date,  
buku_jmlhalaman int,  
buku_deskripsi text,  
buku_harga decimal (10,0),  
aksi varchar(6),  
Aksi_tgl date);
```

Manfaat Trigger

- ▶ Trigger dapat digunakan untuk menguji validitas data yang akan dimasukkan maupun yang akan digunakan untuk melakukan perubahan.
- ▶ Trigger dapat memperoleh nilai lama dari baris data yang akan dihapus atau diubah (misalnya untuk pencatatan histori data dari suatu tabel)
- ▶ Trigger dapat mengubah nilai kolom pada table lain.

Konsep Trigger

- ▶ Trigger adalah objek database yang berisi kumpulan perintah SQL yang akan dieksekusi atau diaktivasi ketika suatu event terjadi.
- ▶ Dalam satu database, dapat didefinisikan lebih dari satu trigger, asal nama dari masing-masing trigger berbeda.
- ▶ Trigger harus berasosiasi dengan table tertentu sebagai pemicu trigger untuk menentukan kapan trigger bersangkutan akan dieksekusi.
- ▶ Trigger memiliki event : BEFORE, INSERT, AFTER, INSERT, BEFORE UPDATE, AFTER UPDATE, BEFORE DELETE dan AFTER DELETE.
- ▶ Trigger memiliki referensi NEW dan OLD

DAFTAR EVENT

- ▶ Event adalah peristiwa atau kejadian yang menunjukkan kapan suatu trigger akan diaktivasi.
- ▶ Contoh : apabila dalam table BUKU didefinisikan trigger dengan Event AFTER INSERT maka trigger tersebut akan diaktivasi setiap kali user melakukan penambahan data pada table BUKU

EVENT	Keterangan
BEFORE INSERT	Trigger akan diaktivasi sesaat sebelum data ditambahkan ke dalam suatu table
AFTER INSERT	Trigger akan diaktivasi sesaat setelah data ditambahkan ke dalam suatu table
BEFORE UPDATE	Trigger akan diaktivasi sesaat sebelum data dalam suatu table di ubah.
AFTER UPDATE	Trigger akan diaktivasi sesaat setelah data dalam suatu table di ubah.
BEFORE DELETE	Trigger akan diaktivasi sesaat sebelum data dihapus dari suatu table
AFTER DELETE	Trigger akan diaktivasi sesaat setelah data dihapus dari suatu table.

REFERENSI : NEW dan OLD

- ▶ NEW berfungsi sebagai referensi dari baris data yang akan dimasukkan atau yang akan digunakan untuk mengubah data.
- ▶ OLD berfungsi sebagai referensi dari baris data yang ditimpa dengan data baru melalui perintah `update` dan dihapus melalui perintah `delete`.
- ▶ Referensi NEW digunakan pada saat mengeksekusi perintah INSERT, sedangkan pada saat UPDATE referensi NEW dan OLD digunakan dua-duanya, serta perintah DELETE hanya menggunakan referensi OLD.
- ▶ Contoh :
Pada perintah `update`, referensi NEW berguna untuk menyimpan nilai yang baru dilakukan perubahan, sedangkan nilai lama akan bisa diakses dengan referensi OLD.

DELIMITER

- ▶ Pada saat mendefinisikan objek yang tersimpan di database MySQL seperti stored procedure, trigger maupun function, ada problem yang akan muncul yaitu penggunaan titik koma (;) pada statement SQL yang merupakan bagian dari objek tersebut. Karena MySQL secara default menganggap titik koma (;) sebagai delimiter / pembatas akhir dari suatu perintah / statement. Akhirnya pembuatan objek yang memiliki beberapa statement tidak akan berjalan sempurna karena "berhenti di tengah jalan".
- ▶ Solusi terhadap masalah di atas adalah menggunakan delimiter selain tanda titik koma (;) misalkan dengan garis pipa (|), atau (&), atau (#)

Membuat TRIGGER

Bentuk Umum	CONTOH
<pre>CREATE TRIGGER <nama_trigger> {EVENT} ON <nama_tabel> FOR EACH ROW BEGIN <badan_trigger> END</pre>	<pre>DELIMITER \$\$ Create trigger tr_ai_buku after insert on buku for each row begin insert into histori_buku values (new.buku_isbn, new.buku_judul, new.penerbit_id, new.buku_tgl_terbit, new.buku_jmlhalaman, new.buku_deskripsi, new.buku_harga, 'INSERT', NOW()); END\$\$</pre>

Deskripsi Contoh

- ▶ Perintah yang terdapat didalam contoh akan membuat trigger dengan nama `tr_ai_buku` yang akan diaktivasi setiap kali user memasukkan baris baru ke dalam tabel `BUKU`
- ▶ Perinta yang akan dieksekusi adalah perintah untuk memasukkan data baru ke dalam tabel `histori_buku`
- ▶ Nilai kolom yang dimasukkan ke dalam tabel `histori buku` diambil dari nilai yang baru saja dimasukkan ke dalam tabel `BUKU`.
- ▶ Untuk mengambil nilai dari tabel `buku`, harus menggunakan referensi `NEW`. Contoh : `NEW.buku_isbn`

Memeriksa Pengaruh Trigger

```
MariaDB [bukudb]> insert into buku values (
-> '222-22666-3-4','Sistem Basis Data','PB05','2006/03/04',270,'MySQL',67000);
Query OK, 1 row affected (0.11 sec)
```

```
MariaDB [bukudb]> select * from histori_buku;
```

buku_isbn	buku_judul	penerbit_id	buku_tglterbit	buku_jmlhalaman	buku_deskripsi	buku_harga	aksi	aksi_tgl
999-11555-2-2	menguasai microsoft excel 2007	PB04	0000-00-00	200	Excel	60000	INSERT	2017-05-26
999-11555-2-2	menguasai microsoft excel 2007	PB04	0000-00-00	254	Excel	64000	UPDATE	2017-05-26
999-11555-2-2	menguasai microsoft excel 2007	PB05	0000-00-00	254	Excel	64000	DELETE	2017-05-26
222-11444-6-5	Pemrograman Java	PB05	2007-05-06	500	Android	70000	insert	2017-05-26
222-22666-3-4	Sistem Basis Data	PB05	2006-03-04	270	MySQL	67000	insert	2017-05-29

```
5 rows in set (0.00 sec)
```

Menghapus TRIGGER

Untuk menghapus trigger dari database, digunakan perintah `DROP TRIGGER`.

Bentuk umumnya :

```
DROP TRIGGER [<nama_database>].<nama_trigger>;
```

Contoh :

```
DROP TRIGGER bukudb.tr_ai_buku;
```

Atau

```
Drop trigger tr_ai_buku;
```

SELESAI

TERIMA KASIH