

MODUL III

T R I G G E R

TUJUAN

Setelah menyelesaikan modul ini, anda diharapkan dapat :

1. Mengenalkan mahasiswa tentang konsep trigger
2. Mengenalkan mahasiswa tentang penggunaan trigger

DASAR TEORI

Trigger

Trigger dapat didefinisikan sebagai himpunan kode (prosedural) yang dieksekusi secara otomatis sebagai respon atas suatu kejadian berkaitan dengan tabel basis data. Kejadian (event) yang dapat membangkitkan trigger umumnya berupa pernyataan INSERT, UPDATE, dan DELETE.

Membuat Trigger

Trigger dibuat menggunakan sintaks sebagai berikut :

```
CREATE TRIGGER <trigger_name> <trigger_time> <trigger_event>  
ON <table>  
FOR EACH ROW <trigger_body statement>
```

Pada sintaks di atas, trigger_name adalah nama dari trigger yang akan kita buat.

Triggertime adalah saat aktivasi trigger. Parameter trigger_time dapat berisi BEFORE atau AFTER, yang menandakan apakah aktivasi trigger dilakukan sebelum atau sesudah terjadi perubahan pada table.

Trigger_event menunjukkan jenis statement yang mengaktifkan trigger. Trigger_event dapat didefinisikan sebagai salah satu dari :

1. INSERT : trigger diaktifkan ketika sebuah baris baru ditambahkan ke table, sebagai contoh melalui statement INSERT, LOAD DATA, atau REPLACE.
2. UPDATE : trigger diaktifkan ketika sebuah baris dimodifikasi, sebagai contoh melalui statement UPDATE.

3. DELETE : Trigger diaktifkan ketika sebuah baris dihapus, melalui statement DELETE dan REPLACE. Namun demikian, DROP TABLE dan TRUNCATE TABLE tidak mengaktifkan trigger DELETE.

Tidak boleh ada dua buah trigger yang sama pada sebuah table yang memiliki trigger_time dan trigger_event yang sama. Sebagai contoh, kita tidak dapat membuat dua buah BEFORE UPDATE trigger pada satu buah table yang sama, namun kita dapat membuat trigger BEFORE UPDATE dan AFTER UPDATE untuk satu table yang sama.

Trigger_body merupakan definisi statement yang dijalankan ketika sebuah trigger diaktifkan. Jika ada beberapa statement yang ingin dijalankan, statement-statement tersebut dapat didefinisikan diantara BEGIN...AND.

Keyword OLD dan NEW dapat digunakan untuk mereferensikan nilai sebelum dan sesudah trigger dilakukan. Sebagai contoh OLD.nama_kolom menunjukkan nilai kolom sebelum data tersebut dihapus atau diupdate, sedangkan NEW.nama_kolom menunjukkan nilai kolom sebuah data yang akan dimasukkan atau nilai kolom data setelah diupdate.

Before Trigger

Berikut adalah contoh trigger yang dijalankan sebelum data dimasukkan ke dalam table.

```
MariaDB [(none)]> use test;
Database changed
MariaDB [test]> delimiter //
MariaDB [test]> CREATE TRIGGER before_insert BEFORE INSERT ON employee
-> FOR EACH ROW
-> BEGIN
-> IF NEW.salary IS NULL OR NEW.salary = 0 THEN
->   SET NEW.salary = 1000;
-> ELSE
->   SET NEW.salary = NEW.salary + 100;
-> END IF;
-> END //
Query OK, 0 rows affected (0.59 sec)
```

Pada contoh di atas, kita membuat trigger before insert yang akan dijalankan sebelum INSERT dilakukan ke tabel employee. Trigger ini akan menggantikan nilai salary jika salary yang dimasukkan dalam INSERT bernilai NULL atau 0 dan menambahkan 100 jika selainnya.

Berikut adalah contoh memasukkan data setelah didefinisikan trigger.

```
MariaDB [test]> insert into employee(id,first_name,last_name,start_date,end_date,salary,city,description)
-> values (11,'John','Lenon','19960725','20060725',0,'Canberra','Programmer');
Query OK, 1 row affected (0.10 sec)
```

```
MariaDB [test]> select * from employee where first_name='John';
+-----+-----+-----+-----+-----+-----+-----+
| id | first_name | last_name | start_date | end_date | salary | city | description |
+-----+-----+-----+-----+-----+-----+-----+
| 11 | John      | Lenon    | 1996-07-25 | 2006-07-25 | 1000.00 | Canberra | Programmer |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Namun demikian kita tidak dapat mengupdate tabel yang sama dengan tabel yang diasosiasikan dengan trigger menggunakan trigger.

```
MariaDB [test]> delimiter &&
MariaDB [test]> CREATE TRIGGER before_update BEFORE UPDATE ON employee
-> FOR EACH ROW
-> BEGIN
-> UPDATE employee SET salary = salary + (NEW.salary - OLD.salary)
-> WHERE id = NEW.id;
-> END &&
Query OK, 0 rows affected (0.44 sec)
```

After Trigger

Berikut adalah contoh trigger yang dijalankan setelah insert dilakukan terhadap table.

- Buat tabel beli

```
MariaDB [test]> CREATE TABLE beli(
-> id_beli INT(11) NOT NULL AUTO_INCREMENT,
-> kd_barang VARCHAR(5) DEFAULT NULL,
-> nama_barang VARCHAR(20) DEFAULT NULL,
-> jumlah INT(11) DEFAULT NULL,
-> PRIMARY KEY (id_beli))
-> ENGINE = MYISAM DEFAULT CHARSET = utf8;
Query OK, 0 rows affected (0.35 sec)
```

- Buat tabel jual

```
MariaDB [test]> CREATE TABLE jual(
-> id_jual INT(11) NOT NULL AUTO_INCREMENT,
-> kd_pelanggan VARCHAR(10) NOT NULL,
-> kd_barang VARCHAR(20) NOT NULL,
-> jumlah INT(11) DEFAULT NULL,
-> PRIMARY KEY (id_jual))
-> ENGINE = MYISAM DEFAULT CHARSET = utf8;
Query OK, 0 rows affected (0.28 sec)
```

- Buat table stok

```
MariaDB [test]> CREATE TABLE stok(
-> kd_barang VARCHAR(5) NOT NULL,
-> jumlah INT(11) NOT NULL,
-> PRIMARY KEY (kd_barang))
-> ENGINE = MYISAM DEFAULT CHARSET = utf8;
Query OK, 0 rows affected (0.13 sec)
```

- `DEFAULT CHARSET=utf8`: Ini menentukan kumpulan karakter default yang akan digunakan untuk kolom dalam tabel. Menggunakan UTF-8 memastikan database Anda dapat menyimpan dan menangani data teks dalam berbagai bahasa.

- Buat trigger dengan nama beli_barang sebagai berikut :

```
MariaDB [test]> DELIMITER &
MariaDB [test]> CREATE TRIGGER beli_barang AFTER INSERT ON beli
-> FOR EACH ROW BEGIN
-> INSERT INTO stok SET
-> kd_barang = NEW.kd_barang, jumlah=New.jumlah
-> ON DUPLICATE KEY UPDATE jumlah=jumlah+New.jumlah;
-> END &
Query OK, 0 rows affected (0.36 sec)
```

- Lakukan insert pada table beli

```
MariaDB [test]> INSERT INTO beli VALUES (1,'K001','PENSIL',30);
Query OK, 1 row affected (0.11 sec)
```

```
MariaDB [test]> SELECT * FROM BELI;
+-----+-----+-----+-----+
| id_beli | kd_barang | nama_barang | jumlah |
+-----+-----+-----+-----+
|      1 | K001      | PENSIL      |      30 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

```
MariaDB [test]> INSERT INTO beli VALUES (2,'K001','PENSIL',23);
Query OK, 1 row affected (0.00 sec)
```

```
MariaDB [test]> SELECT * FROM BELI;
+-----+-----+-----+-----+
| id_beli | kd_barang | nama_barang | jumlah |
+-----+-----+-----+-----+
|      1 | K001      | PENSIL      |      30 |
|      2 | K001      | PENSIL      |      23 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

- Cek isi table stok

```
MariaDB [test]> SELECT * FROM STOK;
+-----+-----+
| kd_barang | jumlah |
+-----+-----+
| K001      |      53 |
+-----+-----+
1 row in set (0.00 sec)
```

NB : Perintah DELIMITER digunakan untuk mengubah delimiter standar, misalnya di sini dari titik koma (;) menjadi slash ganda (/). Langkah ini umumnya dilakukan ketika isi stored procedure mengandung titik koma – yang merupakan delimiter standar di SQL. Pernyataan di antara BEGIN dan END merupakan badan (body) stored procedure. Perintah DELIMITER di akhir baris digunakan untuk mengembalikan delimiter ke karakter semula.

```

MariaDB [test]> INSERT INTO beli VALUES (3,'K002','BUKU',30);
Query OK, 1 row affected (0.00 sec)

MariaDB [test]> SELECT * FROM STOK;
+-----+-----+
| kd_barang | jumlah |
+-----+-----+
| K001      | 53     |
| K002      | 30     |
+-----+-----+
2 rows in set (0.00 sec)

MariaDB [test]> INSERT INTO beli VALUES (4,'K002','BUKU',15);
Query OK, 1 row affected (0.00 sec)

MariaDB [test]> SELECT * FROM STOK;
+-----+-----+
| kd_barang | jumlah |
+-----+-----+
| K001      | 53     |
| K002      | 45     |
+-----+-----+
2 rows in set (0.00 sec)

```

Contoh di atas dilakukan untuk mengupdate stok barang, saat terdapat barang yang masuk/penambahan barang.

Drop Trigger

Untuk menghapus trigger, maka kita dapat menggunakan perintah :

```
Drop trigger nama_trigger;
```

LATIHAN

1. Dengan menggunakan database perpustakaan dan table yang terdapat dalam database tersebut, buatlah trigger untuk mengurangi stok buku saat dilakukan peminjaman berdasarkan judul buku yang dipinjam.
2. Buatlah trigger untuk menambahkan jumlah buku, saat terjadi pengembalian buku.
3. Buatlah trigger untuk menjumlahkan harga sewa saat terjadi peminjaman buku.