

PETUNJUK PRAKTIKUM PBD

MODUL 5 – FUNCTION & TRIGGER

I. TUJUAN PRAKTIKUM:

1. Mahasiswa mampu untuk membuat function
2. Mahasiswa mampu menjalankan function
3. Mahasiswa dapat mengerti proses dari Trigger.
4. Mahasiswa dapat membuat Trigger

II. LANDASAN TEORI

1. Function

SYNTAK UNTUK FUNCTION:

```
CREATE FUNCTION Function_name ([func_parameter [,...]])  
RETURN type  
[characteristic ..] routine_body
```

Pemanggilan fungsi pada ekspresi SQL dapat dilakukan dengan beberapa cara:

- a. Dipanggil pada perintah SELECT
- b. Dipanggil pada klausa kondisi dari WHERE dan HAVING.
- c. Dipanggil dari clause CONNECT BY, START WITH, ORDER BY dan GROUP BY.
- d. Pada klausa VALUES dari perintah INSERT.
- e. Pada klausa SET dari perintah UPDATE

2. Trigger

SYNTAK TRIGGER:

```
CREATE  
[DEFINER = { user | CURRENT_USER }]  
TRIGGER trigger_name trigger_time trigger_event  
ON tbl_name FOR EACH ROW trigger_stmt
```

Keterangan:

- - *Trigger_time* : kapan trigger terjadi (BEFORE /AFTER)
- - *Trigger_event* : operasi yang menyebabkan trigger (INSERT/UPDATE/DELETE)
- - *Trigger_stmt* : aksi yang terjadi pada saat trigger terjadi

III. LANGKAH PRAKTIKUM

1. Buatlah sebuah fungsi yang digunakan untuk mendapatkan volume sebuah balok. Ujilah fungsi yang anda buat!
2. Berdasarkan tabel rental:
 - a. Buatlah fungsi yang digunakan untuk mendapatkan jumlah inventory yang belum dikembalikan oleh seorang customer ! Ujilah dengan *customer_id=100,200,300*.
 - b. Buatlah sebuah fungsi yang digunakan untuk mencari berapa hari sebuah film dipinjam oleh seorang customer.

- c. Buatlah sebuah fungsi yang digunakan untuk mencari siapa customer yang sedang meminjam sebuah film. Ujilah untuk inventory_id = 106,177,1020

3. Contoh penggunaan Trigger

- a. Buatlah 4 buah tabel dengan struktur sebagai berikut:

```
CREATE TABLE test1(a1 INT);
CREATE TABLE test2(a2 INT);
CREATE TABLE test3(a3 INT NOT NULL AUTO_INCREMENT
PRIMARY KEY);
CREATE TABLE test4(
  a4 INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  b4 INT DEFAULT 0
);
```

- b. Buatlah sebuah trigger sebagai berikut :

```
CREATE TRIGGER testref BEFORE INSERT ON test1
FOR EACH ROW BEGIN
  INSERT INTO test2 SET a2 = NEW.a1;
  DELETE FROM test3 WHERE a3 = NEW.a1;
  UPDATE test4 SET b4 = b4 + 1 WHERE a4 = NEW.a1;
END;
```

Perhatikan trigger tersebut :

- Nama trigger adalah :
- Trigger_time nya :
- Trigger_event :
- Trigger_event terjadi pada tabel
- Trigger_statement melakukan proses :

- c. Lakukan proses penyisipan data sebagai berikut:

c.1.

```
INSERT INTO test3 (a3) VALUES
(NULL), (NULL), (NULL), (NULL), (NULL),
(NULL), (NULL), (NULL), (NULL), (NULL);
```

c.2.

```
INSERT INTO test4 (a4) VALUES
(0), (0), (0), (0), (0), (0), (0), (0), (0), (0);
```

c.3.

```
INSERT INTO test1 VALUES
-> (1), (3), (1), (7), (1), (8), (4), (4);
```

Perhatikan hasil pada tabel test1, test2, test3 dan test4 !

4. Cobalah contoh-contoh yang ada pada modul_7 tentang Trigger !

- a. Buat terlebih dahulu tabel barang.
- b. Isi sesuai dengan contoh.
- c. Buat tabel detail pembelian sesuai dengan struktur yang telah ditetapkan.
- d. Coba buat beberapa trigger untuk event pada tabel detailpembelian baik untuk delete, insert maupun update.
- e. Coba lakukan insert, update dan delete pada tabel detailpembelian tunjukkan perubahan yang terjadi !!

5. Buatlah trigger untuk kasus penjualan. Tabel identik dengan tabel detailpenjualan.