Lembaga Pendidikan & Pengembangan Profesi Terpadu
LP3T NURUL FIKRI
IT Training - IT Education
LP3T-NF

KOTA DEPOK
PARICARA DARMA

TRANSACTION

***Powered by*:**

Over View

Dalam kenyataan sehari-hari kita sering menemukan bisnis proses yang melibatkan banyak operasi manipulasi data seperti menambah, mengubah dan mengurangi atau menghapus data yang harus dilakukan secara berurutan. Contoh bisnis proses seperti ini antara lain:

- Pemindahan dana dari satu rekening ke rekening lain
- Pencatatan penjualan dan pengurangan stok
- Pencatatan penerimaan pembayaran dan pengurangan deposit pelanggan
- Pencatatan status penerimaan dan pemesanan

Transaction Syntax

Syntax :

BEGIN
  perintah manipluasi ke-1
  perintah manipulasi ke-2
  perintah manipulasi ke-N
{COMMIT | ROLLBACK [TO SAVEPOINT nama] }

COMMIT -> seluruh proses disahkan
ROLLBACK -> seluruh proses akan dibatalkan

Contoh Transaction

```
BEGIN;

  update satuan set nama='miligram' where id=1;
  update barang set id_sat=1;

ROLLBACK;
```

Transaction Save Point

```
BEGIN;
  update satuan set nama='miligram' where id=1;
 SAVEPOINT update_barang;
  update barang set id_sat=1;
  update barang set nama='Minyak’;
ROLLBACK TO SAVEPOINT update_barang;
ROLLBACK ;
```

Transaksi Tabel View

View bukan sebuah tabel yang asli namun dia hanya merupakan duplikat/bayangan dari sebuah tabel, view dapat di-select layaknya sebuah tabel namun untuk meng-update ataupun menghapus data yang terdapat di dalamnya haruslah menggunakan ***RULE***.

Jadi dengan cara melekatkan rule pada tabel view tersebut maka proses update dan delete dapat dilakukan sebagaimana mestinya

RULE(1)

Rule berfungsi untuk merubah atau meng-update sebuah tabel, hanya dengan melekatkan rule tersebut pada suatu tabel maka sifat dari tabel tadi dapat diubah. VIEW dalam PostgreSQL sebenarnya telah memiliki SELECT RULE.

Rule memiliki sebuah keyword yang berfungsi untuk mencegah terjadinya penambahan suatu data ke dalam sebuah tabel, keyword yang dimaksud adalah NOTHING.

RULE(2)

Dengan menggunakan RULE kita dapat memanfaatkan berbagai macam default yang telah tersedia pada PostgreSQL, seperti timestamp dan current user. Berikut contoh yang mengimplementasikan default tersebut, misalkan kita membuat dua buah tabel barang berisikan atribut yang sama namun berbeda fungsinya. Nama tabel tersebut adalah barang dan barang_log.

Fungsi dari tabel barang untuk menampung data yang masuk, sedangkan tabel barang_log berfungsi untuk menampung data hasil UPDATE atau DELETE dari tabel barang, artinya jika sebuah data pada tabel barang di-update ataupun dihapus maka data tersebut tidak hilang melainkan secara otomatis akan tersimpan pada tabel barang_log

.

Syntax Rule

Syntax :

```
CREATE RULE name AS
  ON event TO object
  DO [INSTEAD] [action | NOTHING]
```

Contoh :

```
CREATE RULE kependudukan_v_update AS
  ON UPDATE TO kependudukan_v
  DO INSTEAD
  UPDATE penduduk  SET nama = NEW.nama
  WHERE nama = OLD.nama;
```

Fungsi(1)

Aplikasi pemakai database, seringkali memerlukan pengeksekusian beberapa perintah SQL sekaligus untuk menyelesaikan suatu bisnis proses. Beberapa contoh bisnis proses yang demikian adalah:

- Transaksi penjualan dan update stok
- Pencatatan penerimaan barang dan update status po
- Transaksi penjualan dan posting jurnal penjualan (akunting)

Akan lebih mudah bagi programmer jika sekumpulan perintah SQL tersebut dibungkus (encapsulated) dalam satu operasi abstraksi

Fungsi(2)

- User defined function (function) adalah suatu operasi yang dapat didefinisikan oleh pemakai database, dapat terdiri dari beberapa perintah SQL yang dieksekusi sebagai satu kesatuan operasi.
- Definisi operasi ini disimpan dan dieksekusi oleh database server.
- User defined function dikenal juga dengan istilah 'stored procedure‘.

Fungsi(3)

PostgreSQL menyediakan beberapa bahasa pemrograman untuk mengimplementasikan user defined function, antara lain:

- PL (Procedural Language) / pgSQL
- TCL
- Perl
- Python
- C
- Ruby
- Java

Membuat Fungsi(1)

Struktur fungsi (function) terdiri dari 2 bagian:

1. Deklarasi, teridiri atas:
   a) Nama fungsi
   b) Parameter (argument)
   c) Tipe data kembalian (return data type)
   d) Deklarasi variable lokal, jika diperlukan
2. Tubuh fungsi

Membuat Fungsi dengan PL/pgSQL

Untuk dapat mendefinsikan atau membuat fungsi dengan PL/pgSQL maka sebelumnya Anda pastikan dahulu apakah bahsa plpgsql telah terdaftar pada catalog pg_language. Untuk itu coba cek dengan query berikut ini :

SELECT * FROM pg_language

Jika hasil query menyatakan ada plpgsql pada kolom “lanname” maka database Anda sudah mensupport bahsa plpgsql dan Anda siap untuk membuat fungsi dengannya. Jika belum lakukan langkah berikut ini:

CREATE LANGUAGE plpgsql

Syntax Membuat Fungsi

```
CREATE FUNCTION nama_fungsi ([tipe_data_parameter,...])
RETURNS tipe_data_kembalian AS  $$
DECLARE
deklarasi variabel lokal;
[.....]
BEGIN
    statement;
     [....]
END;
$$ LANGUAGE 'plpgsql';
```

Contoh Membuat Fungsi

Contoh membuat fungsi:

```
CREATE FUNCTION tambah (a1 integer, a2 integer)
RETURNS integer AS  $$
DECLARE
        hasil integer;
BEGIN
   hasil = a1 + a2;
   RETURN hasil;
END;
$$ LANGUAGE 'plpgsql';
```

Memanggil fungsi tambah():

```
SELECT tambah(10,12)
UPDATE tabel1 SET kolom1=tambah(2,3) WHERE id=10
```

Ekspresi

Ekspresi adalah perhitungan atau operasi yang menghasilkan nilai dengan tipe data yang hanya dikenal di postgresql. Selain query dinamis, semua ekspresi hanya disiapkan sekali selama siklus hidup sebuah proses server postgresql. Karena ekspresi hanya disiapkan sekali maka nilai-nilai konstanta (misal: now atau current) hanya disiapkan sekali.

Berikut ini memperlihatkan bagaimana suatu konstanta dalam fungsi diproses.

CREATE TABLE logs (id serial primary key, msg text, tgl date, waktu time)

Contoh Ekspresi

Membuat fungsi addlog

```
CREATE OR REPLACE FUNCTION addlog(str text) RETURNS VOID
AS $$
BEGIN
INSERT INTO logs (msg,tgl,waktu) VALUES (str,'today', 'now');
RETURN;
END;
$$ LANGUAGE 'plpgsql';
```

Memanggil fungsi addlog:

SELECT addlog('Percobaan nih'); -- tunggu beberapa detik lalu eksekusi lagi dan kemudian lihatlah isi tabel logs, dan perhatikan nilai kolom tgl dan waktu

Contoh Ekspresi(2)

Fungsi addlog sebelumnya , ternyata ketika dieksekusi berapa kali namun nilai kolom waktu dan tgl pada tabel logs tidak akan pernah berubah. Sebaliknya fungsi addlog2 berikut ini menghasilkan nilai yang diharapkan.

```
CREATE OR REPLACE FUNCTION addlog2(str text) RETURNS VOID AS $$
DECLARE
    hariini date;
    sekarang time;
BEGIN
    hariini := 'today';
   sekarang := 'now';
INSERT INTO logs (msg,tgl,waktu) VALUES (str,hariini, sekarang);
RETURN; END; $$ LANGUAGE 'plpgsql';
```

TRIGGER

Adalah fungsi yang dipanggil secara otomatis ***oleh database pada saat event yang ditentukan oleh pemakai database.***

Untuk menerapkan trigger , perlu langkah berikut ini:

- Membuat fungsi yang akan dipanggil oleh fungsi trigger
- Mendaftarkan fungsi triger, dengan perintah CREATE TRIGGER

Syntax Trigger

CREATE TRIGGER nama_triger { BEFORE | AFTER }
{ event [OR..] }
ON nama_tabel FOR EACH { ROW | STATEMENT }
EXECUTE PROCEDURE nama_fungsi(arguments)

Untuk fungsi trigger nilai kembaliannya (RETURNS data type) adalah “trigger'.

Mendaftar Fungsi Sebagai Trigger

Sebuah fungsi didaftarkan sbg trigger dengan ketentuan sbb:

- Berlaku pada tabel tertentu
- Pada operasi yang mengubah data: INSERT, UPDATE, DELETE
- Dieksekusi sebelum (BEFORE) atau setelah (AFTER) operasi mengubah data
- Dieksekusi untuk setiap baris (ROW) yang dipengaruhi operasi atau hanya sekali untuk setiap perintah (STATEMENT) dari operasi mengubah data.

Contoh Fungsi Trigger

```
CREATE FUNCTION trig_pengeluaran()
RETURNS TRIGGER AS $$
BEGIN
UPDATE barang SET stok = stok - NEW.jml
WHERE idbarang = NEW.idbarang;
RETURN NEW;
END;
$$
LANGUAGE 'plpgsql';
```

Contoh Trigger & Memanggil Fungsi Trigger

```
CREATE TRIGGER trig_pengeluaran
AFTER INSERT ON pengeluaran
FOR EACH ROW
EXECUTE PROCEDURE trig_pengeluaran();
```

Lembaga Pendidikan & Pengembangan Profesi Terpadu
LP3T NURUL FIKRI
IT Training - IT Education
LP3T-NF

KOTA DEPOK
PARICARA DARMA

THANK YOU

OUR QUALITY
YOUR TRUST

www.nurulfikri.com

PT. Nurul Fikri Cipta Inovasi
Jl. Margonda Raya No. 522
Depok Jawa Barat
Telp / Fax : (021) 7874224 / 7874225
Website: www.nurulfikri.com