

Adv. SQL

Ocak 2023

Miraç

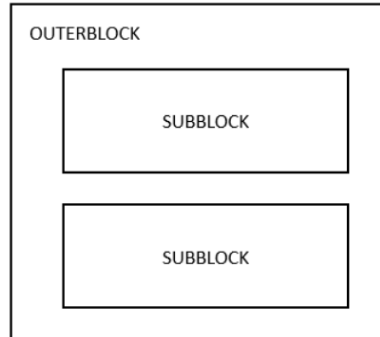
PL/pgSQL Nedir ???

- PL/pgSQL prosedürel bir SQL dilidir.
 - ➔ Prosedürel = Kodların satır satır okunup işlenmesi
- SQL diline kontrol yapıları eklenmiştir, If-Else / For gibi yapılar ...
- PL/pgSQL, SQL'in tüm veri tiplerini, operatör ve fonksiyonlarını kullanabilir.
- PostgreSQL 9.0 ve sonrasında PL/pgSQL varsayılan olarak kurulmaktadır.

PL/pgSQL Blok Yapısı

PL/pgSQL blok yapıları bir dildir. Fonksiyonun gövdesini oluşturan ifadeler blok denilir.

```
[ <<label>> ]  
[ declare  
    declarations ]  
begin  
    statements;  
    ...  
end [ label ];
```



PL/pgSQL Variables

- Değişken tanımlama syntax : expression kısmı opsiyoneldir, default değer verecek isek kullanırız, kullanılmaz ise default olarak Null atanır. Atama yapılırken := veya = kullanılabilir.

```
variable_name data_type [:= expression];
```

- Değişkeni kullanmadan önce, PL/pgSQL block da deklarasyon bölümünde ilgili değişkeni tanımlamalıyız

PL/pgSQL Row Type

Database deki recordların data türünde bir değişken oluşturmak için kullanılır

Syntax :

```
row_variable table_name%ROWTYPE;  
row_variable view_name%ROWTYPE;
```

İlgili datalara ulaşmak istersem :

```
row_variable.field_name
```

PL/pgSQL Record Types

Row-Type gibidir, bazı fieldlar üzerinden recordlara ulaşmak istediğimizde kullanılır

Syntax :

```
declare
    rec record;
begin
    -- select the film
    select film_id, title, length
    into rec
    from film
    where film_id = 200;

    raise notice '% % %', rec.film_id, rec.title, rec.length;
```

PL/pgSQL Constant

Sabit ve değiştirilmesini istemediğimiz bir değişken oluşturabiliriz..

Syntax :

```
constant_name constant data_type := expression;
```

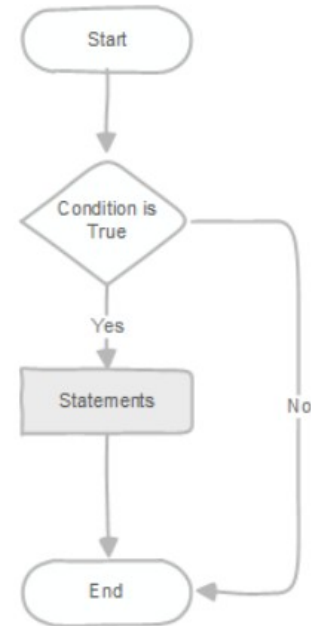
```
do $$  
declare  
    vat constant numeric := 0.1;  
    net_price    numeric := 20.5;  
begin  
    raise notice 'The selling price is %', net_price * ( 1 + vat );  
end $$;
```

PL/pgSQL - IF Statement

Syntax :

```
if condition then  
    statements;  
end if;
```

FlowChart :

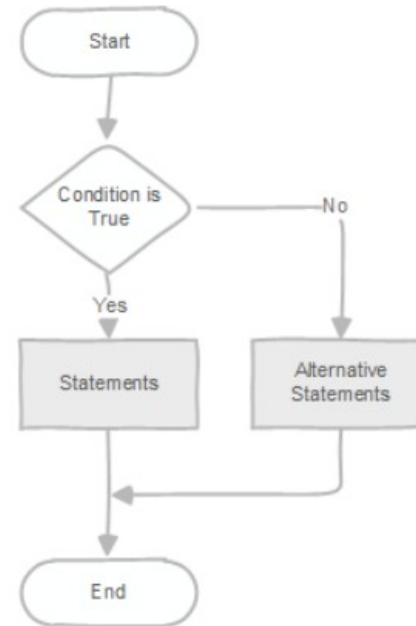


PL/pgSQL- If-Then-Else

Syntax :

```
if condition then  
    statements;  
else  
    alternative-statements;  
END if;
```

FlowChart :

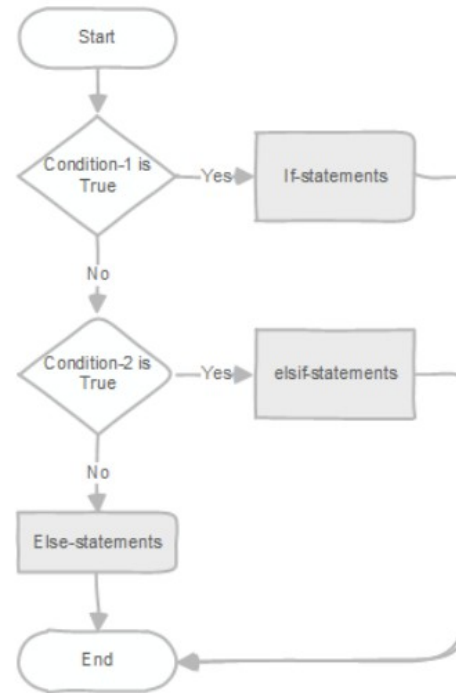


PL/pgSQL - If-Then-Elseif

Syntax :

```
if condition_1 then
    statement_1;
elsif condition_2 then
    statement_2
...
elsif condition_n then
    statement_n;
else
    else-statement;
end if;
```

FlowChart :

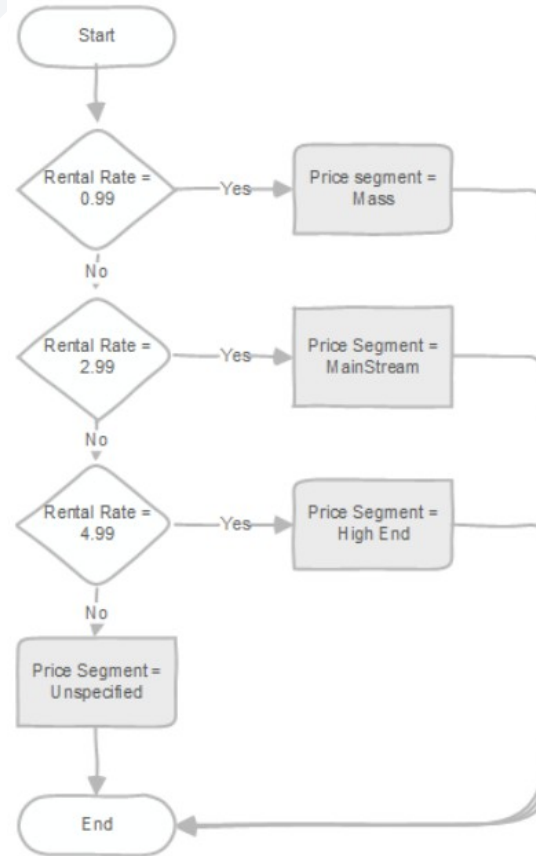


PL/pgSQL Case

Syntax:

```
case search-expression
  when expression_1 [, expression_2, ...] then
    when-statements
  [ ... ]
  [else
    else-statements ]
END case;
```

FlowChart :



PL/pgSQL Loop - If

```
do $$
declare
    n integer:= 10;
    fib integer := 0;
    counter integer := 0 ;
    i integer := 0 ;
    j integer := 1 ;
begin
    if (n < 1) then
        fib := 0 ;
    end if;
    loop
        exit when counter = n ;
        counter := counter + 1 ;
        select j, i + j into i, j ;
    end loop;
    fib := i;
    raise notice '%', fib;
end; $$
```

→ Programlama mantığındaki gibi döngülerimizi if ile şarta bağlayarak sonlandırabiliyoruz.

→ Döngünün herhangi bir iterasyonunda çıkış yapmak için exit keyword ünü kullanabiliriz

PL/pgSQL While Loop

Çıktı :

```
do $$  
declare  
    counter integer := 0;  
begin  
    while counter < 5 loop  
        raise notice 'Counter %', counter;  
        counter := counter + 1;  
    end loop;  
end$$;
```



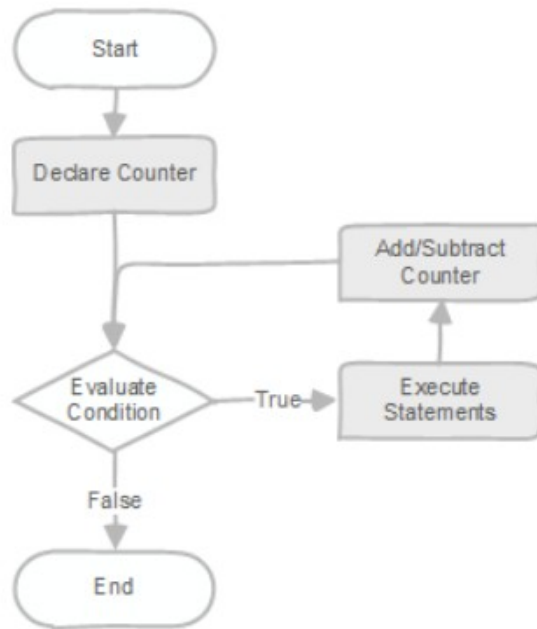
```
NOTICE: Counter 0  
NOTICE: Counter 1  
NOTICE: Counter 2  
NOTICE: Counter 3  
NOTICE: Counter 4
```

PL/pgSQL For Loop

```
do $$  
begin  
  for counter in 1..5 loop  
    raise notice 'counter: %', counter;  
  end loop;  
end; $$
```



```
NOTICE: Counter: 1  
NOTICE: Counter: 2  
NOTICE: Counter: 3  
NOTICE: Counter: 4  
NOTICE: Counter: 5
```



İyi Çalışmalar...