

Exploitation d'une Base de Données

Cours 6 - PL/pgSQL et Curseurs

Anaïs Durand

25 Avril 2023

SELECT ... INTO ...

```
DO$$  
  DECLARE  
    n Colis.num%TYPE;  
    p Colis.poids%TYPE;  
  BEGIN  
    SELECT c.num, c.poids INTO n, p  
    FROM Colis c, Client cl  
    WHERE c.destinataire = cl.num  
           AND cl.nom = 'Petit'  
           AND cl.prenom = 'Jean';  
  
    RAISE NOTICE 'Colis % (% kg)', n, p;  
  END;  
$$;
```

SELECT ... INTO ...

```
DO$$
  DECLARE
    n Colis.num%TYPE;
    p Colis.poids%TYPE;
  BEGIN
    SELECT c.num, c.poids INTO n, p
    FROM Colis c, Client cl
    WHERE c.destinataire = cl.num
           AND cl.nom = 'Petit'
           AND cl.prenom = 'Jean';

    RAISE NOTICE 'Colis % (% kg)', n, p;
  END;
$$;
```

Si le **SELECT** retourne plusieurs lignes, seule la première sera mise dans **n** et **p** puis affichée.

Curseur

Comment faire pour accéder à toutes les lignes retournées par le
`SELECT` ?

⇒ utiliser un *curseur*

Curseur

Comment faire pour accéder à toutes les lignes retournées par le `SELECT` ?

⇒ utiliser un *curseur*

1 Déclarer le curseur :

```
DECLARE  
  nom_curseur cursor FOR SELECT ... FROM ..  
  WHERE ...;
```

Curseur

Comment faire pour accéder à toutes les lignes retournées par le `SELECT` ?

⇒ utiliser un *curseur*

1 Déclarer le curseur :

```
DECLARE  
    nom_curseur cursor FOR SELECT ... FROM ..  
    WHERE ...;
```

2 Ouvrir le curseur :

```
OPEN nom_curseur;
```

3 Parcourir les données lignes par lignes :

```
FETCH nom_curseur INTO var1, var2, var3 ...  
WHILE FOUND LOOP  
    FETCH nom_curseur INTO var1, var2, var3 ...  
    ...  
END LOOP;
```

3 Parcourir les données lignes par lignes :

```
FETCH nom_curseur INTO var1, var2, var3 ...  
WHILE FOUND LOOP  
    FETCH nom_curseur INTO var1, var2, var3 ...  
    ...  
END LOOP;
```

4 Fermer le curseur :

```
CLOSE nom_curseur;
```

Exemple

```
DO$$
DECLARE
  n      Colis.num%TYPE;
  p      Colis.poids%TYPE;
  curs  cursor FOR SELECT c.num, c.poids
                        FROM Colis c, Client cl
                        WHERE c.destinataire = cl.num
                           AND cl.nom = 'Petit'
                           AND cl.prenom = 'Jean';

BEGIN
  OPEN curs;
  FETCH curs INTO n, p;
  WHILE FOUND LOOP
    RAISE NOTICE 'Colis % (% kg)', n, p;
    FETCH curs INTO n, p;
  END LOOP;
  CLOSE curs;
END;
$$;
```

Modification avec curseur

Possibilité de modifier/supprimer la ligne courante pendant un parcours avec curseur :

Modification avec curseur

Possibilité de modifier/supprimer la ligne courante pendant un parcours avec curseur :

- 1 Déclarer le curseur **FOR UPDATE** pour éviter des problèmes d'accès concurrents :

```
DECLARE  
  nom_curseur cursor FOR SELECT ... FROM ...  
  WHERE ... FOR UPDATE;
```

Modification avec curseur

Possibilité de modifier/supprimer la ligne courante pendant un parcours avec curseur :

- 1 Déclarer le curseur **FOR UPDATE** pour éviter des problèmes d'accès concurrents :

```
DECLARE  
    nom_curseur cursor FOR SELECT ... FROM ...  
    WHERE ... FOR UPDATE;
```

- 2 Modification/suppression avec **WHERE CURRENT OF** :

```
UPDATE ...  
SET ...  
WHERE CURRENT OF nom_curseur;  
DELETE FROM ...  
WHERE CURRENT OF nom_curseur;
```

Exemple

```
DO$$
  DECLARE
    n      Colis.num%TYPE;
    p      Colis.poids%TYPE;
    d      Colis.dateExped%TYPE;
    curs   cursor FOR SELECT c.num, c.poids, c.dateExped
              FROM Colis c
              WHERE c.destinataire = 'C001'
              FOR UPDATE;

  BEGIN
    OPEN curs;
    FETCH curs INTO n, p, d;
    WHILE FOUND LOOP
      IF d <= current_date THEN
        DELETE FROM Colis WHERE CURRENT OF curs;
      ELSIF p > 10 THEN
        UPDATE Colis
          SET entrepot = 'B'
          WHERE CURRENT OF curs;
      END IF;
      FETCH curs INTO n, p, d;
    END LOOP;
    CLOSE curs;
  END;
$$;
```