

Cours de bases de données, ,
<http://sql.bdpedia.fr>

Curseurs

Les curseurs

Curseur = **mécanisme de base pour parcourir le résultat d'une requête SQL.**

Permet de charger une ligne à la fois. Evite le chargement d'une très grosse table si résultat volumineux.

Un curseur se déclare. Exemple PL/SQL.

```
CURSOR MonCurseur IS
SELECT * FROM Film, Artiste
WHERE idMES = idArtiste;

-- Une variable pour le curseur
v_monCurseur MonCurseur%ROWTYPE;
```

Les curseurs paramétrés

Autre exemple, avec paramètre

```
CURSOR MonCurseur (p_artiste INTEGER) IS
  SELECT prenom || nom AS nomMES, anneeNaiss, COUNT(*) AS nbFilms
  FROM Film, Artiste
  WHERE idMES = p_artiste
```

```
-- Une variable pour le curseur
v_realisateur MonCurseur%ROWTYPE;
```

Exécution d'un curseur

Un curseur est **toujours** exécuté en trois phases

ouverture du curseur (ordre OPEN) ;

parcours du résultat en itérant des ordres FETCH autant de fois que nécessaire ;

fermeture du curseur (CLOSE).

Bien comprendre : (1) on calcule, (2) on récupère le résultat **ligne à ligne** ; (3) on libère les ressources.

Un exemple complet (1)

```
CREATE OR REPLACE PROCEDURE CurseurFilms (p_annee INT) AS

  -- Curseur
  CURSOR MonCurseur (v_annee INTEGER) IS
    SELECT idFilm, titre, prenom, nom
    FROM Film, Artiste
    WHERE idMES = idArtiste
    AND   annee = v_annee;

  -- Variable du curseur
  v_monCurseur MonCurseur%ROWTYPE;
  -- Variable pour la liste des acteurs
  v_mesActeurs VARCHAR(255);
BEGIN
  -- Ouverture du curseur
  OPEN MonCurseur(p_annee);
```

Un exemple complet (2)

```
-- On prend le premier n-uplet
FETCH MonCurseur INTO v_monCurseur;
-- Boucle sur les n-uplets
WHILE (MonCurseur%FOUND) LOOP
  -- Recherche des acteurs avec la fonction MesActeurs
  v_mesActeurs := MesActeurs (v_monCurseur.idFilm);

  DBMS_OUTPUT.PUT_LINE('Ligne ' || MonCurseur%ROWCOUNT ||
    ' Film: ' || v_monCurseur.titre ||
    ', de ' || v_monCurseur.prenom || ' ' ||
    v_monCurseur.nom || ', avec ' || v_mesActeurs);
  -- Passage au n-uplet suivant
  FETCH MonCurseur INTO v_monCurseur;
END LOOP;
-- Fermeture du curseur
CLOSE MonCurseur;

EXCEPTION
  WHEN OTHERS THEN
    DBMS_OUTPUT.PUT_LINE('Message : ' || sqlerrm);
END;
```

Un exemple complet (3)

Exemple de l'exécution de cette procédure.

```
SQL> set serveroutput on
```

```
SQL> execute CurseurFilms(1992);
```

```
Ligne 1 Film:  Impitoyable, de Clint Eastwood, avec
```

```
                Clint Eastwood, Gene Hackman, Morgan Freeman
```

```
Ligne 2 Film:  Reservoir dogs, de Quentin Tarantino, avec
```

```
                Quentin Tarantino, Harvey Keitel, Tim Roth, Chris Penn
```

On a "encapsulé" un traitement de données complet, que l'on peut ensuite exécuter dans n'importe quel contexte.

Autre syntaxe

Il existe une syntaxe encore plus simple.

```
CREATE OR REPLACE PROCEDURE CurseurMAJ AS

  -- Curseur
  CURSOR CurseurGenre IS
    SELECT * FROM Genre FOR UPDATE;
BEGIN
  -- Boucle FOR directe: pas de OPEN, pas de CLOSE

  FOR v_genre IN CurseurGenre LOOP
    IF (v_genre.code IS NULL) THEN
      DELETE FROM Genre WHERE CURRENT OF CurseurGenre;
    ELSE
      UPDATE Genre SET code=UPPER(code)
        WHERE CURRENT OF CurseurGenre;
    END IF;
  END LOOP;
END;
```

On retrouve ce type d'expression dans des langages interprétés (construction de la variable au moment de l'exécution). Python, PHP...