

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

Curseurs

# Les curseurs

En SQL, le résultat d'une requête est un **ensemble**. Conceptuellement, il existe instantanément.

Un programme accède au résultat nuplet par nuplet

**Curseur** : mécanisme pour parcourir le résultat d'une requête

Le programme prend du temps pour parcourir un curseur : **que devient la base de données pendant ce temps ?**

Dans ce qui suit : illustration avec PL/SQL.

# Un curseur se déclare

Exemple PL/SQL.

```
-- Déclaration d'un curseur  
cursor MonCurseur is  
select * from Voyageur  
where lieu = 'Corse'
```

Autre exemple, avec paramètre

```
cursor MonCurseur (p_lieu varchar) is  
select * from Voyageur  
where lieu = p_lieu
```

Ce peut être n'importe quelle requête SQL

# Exécution d'un curseur

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

- Ouverture du curseur (ordre **open**) : la requête est exécutée
- Parcours du résultat : itération de **next** (ou **fetch**) tant qu'il reste des nuplets
- Fermeture du curseur (**close**)

Séquence typique ;

```
open monCurseur;  
while (nuplet = monCurseur.fetch()) do  
  -- Traitement de nuplet  
end while;  
close monCurseur;
```

# Le résultat d'un curseur est immuable

Le résultat est déterminé au moment de l'exécution

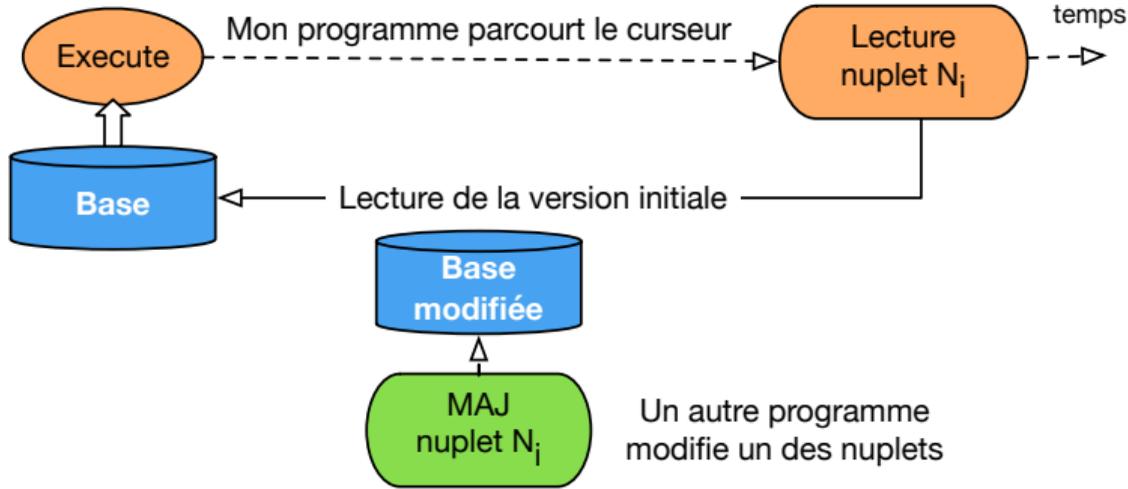
Sinon le curseur suivant ne finirait jamais...

```
-- Un curseur qui s'exécute indéfiniment
open du curseur sur la table T;
while (fetch du curseur) loop
    Insérer un nouveau nuplet dans T;
end loop;
close du curseur;
```

Tout se passe comme si le SGBD maintenait un "cliché" dans une table temporaire. Tant que le curseur existe, ce cliché est conservé.

# Les mises à jour concurrentes ne sont pas visibles

Même principe : le contenu est défini à l'exécution



Première rencontre avec la notion de **concurrency** et **isolation**

# Peut-on modifier les nuplets parcourus ?

Oui, avec la clause `current of` (si accès à une seule table)

```
-- Déclaration du curseur
cursor CurseurGenre is select * from Genre

-- Boucle for intégrant open, fetch et close
for v_genre in CurseurGenre loop
    update Genre set code=UPPER(code)
    where current of CurseurGenre;
end loop;
```

**Notez** : ces mises à jour deviennent visibles par ma session

# À retenir

Les curseurs introduisent un aspect **temporel** dans la gestion des données.

Le résultat d'une requête est défini à l'instant de l'exécution mais exploité sur une période de durée indéterminée.

Soulève des questions sur la **concurrency** avec d'autres sessions.

- Mes mises à jour sont-elles visibles ?
- Les mises à jour des autres sessions sont-elles visibles ?
- Et si on a deux mises à jour **simultanées** du même nuplet... ?

Plus sur ce sujet avec la notion de **transaction**.