

Les séquences

Rappel

Un schéma contient un ensemble d'objets d'une base de données. Un schéma est généralement associé à un utilisateur. Les objets de schéma (schema objects) sont les structures logiques qui se réfèrent directement aux données de la base. Par exemple :

- Tables
- Vues (Views)
- Séquences
- Synonymes
- PL/SQL (Procédures Stockées, paquetage)
- Triggers (Déclencheurs)
- ...

Les séquences

Une séquence est un objet de schéma, tout comme une table par exemple, qui permet de générer des numéros uniques utilisables selon les besoins de l'utilisateur, pour fournir des données pour des colonnes de type numériques par exemple.

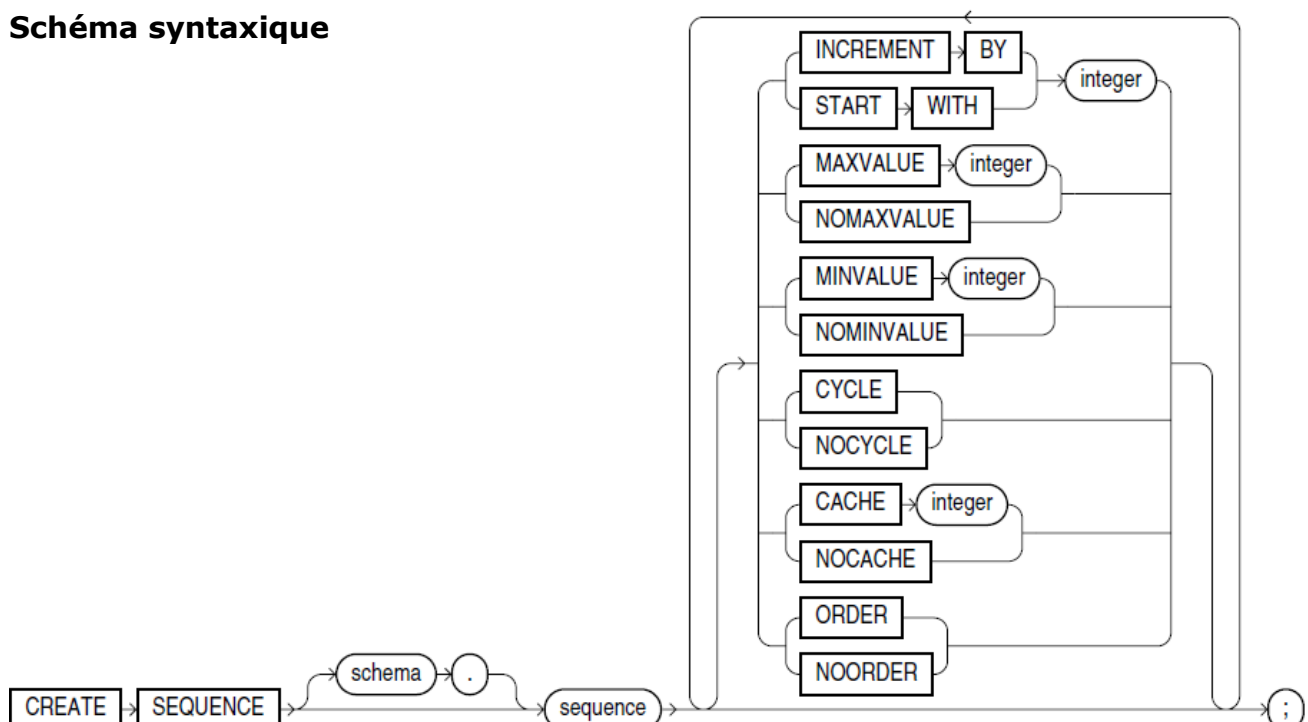
Une séquence peut ainsi être utilisée pour générer des valeurs pour une clé primaire « artificielle » (c'est-à-dire un numéro séquentiel sans aucune signification).

Si on utilise une séquence pour générer le numéro d'employé de la table EMPLOYE, même si deux utilisateurs saisissent un nouvel employé en même temps, il n'est pas nécessaire que l'un attende que l'autre ait terminé pour connaître le numéro suivant disponible, il suffit de demander le numéro suivant à la séquence.

Les séquences sont indépendantes des tables, ce qui signifie que l'on peut utiliser une même séquence pour plusieurs tables si on le désire. De même, une même séquence peut être utilisée par plusieurs utilisateurs.

Création d'une séquence

Schéma syntaxique



Syntaxe simplifiée

```
CREATE SEQUENCE sq_nomSéquence
  [Increment By n]                -- intervalle entre chaque no généré (+/-)
  [Start With n ]                 -- première valeur à générer
  [{ MaxValue n | NoMaxValue }]  -- valeur maximum à générer | pas de max
  [{ MinValue n | NoMinValue }]  -- valeur minimum à générer | pas de min
  [{ Cycle | NoCycle }]          -- on recommence le cycle | ou pas
  [{ Cache | NoCache }] (def 20) -- un certain nb de valeurs en mémoire cache
  [{ Order | NoOrder }];         -- on conserve l'ordre des transactions (important
                                si la séquence est utilisée comme «timestamp»)
```

Comme toujours, les séquences peuvent être modifiées avec `ALTER SEQUENCE` ou supprimées avec `DROP SEQUENCE`.

Exemples :

Séquence croissante :	Séquence décroissante :
<pre>CREATE SEQUENCE sq_exe_employe INCREMENT BY 1 --- par défaut START WITH 100 MINVALUE 100 MAXVALUE 999 NOCYCLE; --- par défaut</pre>	<pre>CREATE SEQUENCE sq_exe_dept_desc INCREMENT BY -1 START WITH 10000 MAXVALUE 10000 MINVALUE 1;</pre>

Utilisation d'une séquence

On peut se référer aux valeurs d'une séquence dans des requêtes SQL grâce à deux pseudo-colonnes :

- **NEXTVAL** : retourne la prochaine valeur
- **CURRVAL** : retourne la valeur actuelle de la séquence (pour autant que la séquence ait déjà été utilisée au moins une fois dans la session)

On accède à ces colonnes via le nom de la séquence :

- Ex : `sq_exe_employe.NEXTVAL`

On peut utiliser les séquences directement dans une commande `INSERT INTO` :

- Ex : `INSERT INTO exe_employe`
`VALUES (sq_exe_employe.NEXTVAL, 'Jean', 'BON', ...`

Les séquences fonctionnent de la façon suivante :

- La toute première fois que l'on se réfère à la pseudo-colonne `NEXTVAL` d'une séquence, celle-ci retourne sa valeur initiale. Les accès suivants à `NEXTVAL` incrémenteront la séquence et retourneront la nouvelle valeur.
- Chaque fois que l'on se réfère à la valeur `CURRVAL` d'une séquence, celle-ci retourne la dernière valeur de cette séquence **DANS LA SESSION COURANTE**. Ceci implique que l'on ne peut pas accéder à `CURRVAL` dans une session avant d'avoir effectué au moins un accès à `NEXTVAL` dans cette même session.
- Dans une seule et même requête, l'utilisation simultanée de plusieurs `CURRVAL` et `NEXTVAL` est possible, néanmoins la séquence ne sera incrémentée qu'une seule fois et `CURRVAL` et `NEXTVAL` retourneront la même valeur – même si l'on fait plusieurs références à `NEXTVAL` !

Exemple

La valeur d'une séquence peut donc être utilisée plusieurs fois dans une session :

On désire insérer un nouveau département (en utilisant `sq_exe_dept.NEXTVAL` pour obtenir un nouveau numéro), puis insérer plusieurs employés dans ce département. On utilisera ensuite `sq_exe_dept.CURRVAL` pour faire référence au département nouvellement créé.

On suppose que les séquences `sq_exe_dept` et `sq_exe_employe` existent déjà dans le schéma concerné.

On va procéder de la façon suivante :

- `INSERT INTO exe_dept` -- obtenir un nouveau n° pour créer un département
`VALUES (sq_exe_dept.NEXTVAL, 'Logistique', 'Genève');`
- `INSERT INTO exe_employe`
`VALUES (sq_exe_employe.NEXTVAL, -- pour obtenir un nouveau n° d'employé`
`'Jean', 'BON', ..., -- autres valeurs de l'employé à fournir`
`sq_exe_dept.CURRVAL); -- numéro du département créé ci-dessus`
- `INSERT INTO exe_employe`
`VALUES (sq_exe_employe.NEXTVAL, -- pour obtenir un nouveau n° d'employé`
`'Yves', 'REMORD', ..., -- autres valeurs de l'employé à fournir`
`sq_exe_dept.CURRVAL); -- numéro du département créé ci-dessus`

Ceci nous permet d'insérer un département puis des employés qui y sont rattachés sans connaître le numéro du nouveau département.

Séquence et PL/SQL

Les séquences trouvent une utilisation particulièrement intéressante dans l'automatisation de la numérotation de valeurs de clés primaires. En effet, pourquoi ne pas demander à Oracle de gérer cet aspect pour nous ?

Ceci peut se faire à l'aide de déclencheurs (triggers) qui réagissent à des événements sur un objet de schéma et qui peuvent utiliser des séquences dans du code PL/SQL.

Dans notre exemple, pour obtenir un numéro de département automatiquement, il faudrait un trigger de type BEFORE INSERT (qui s'exécute avant l'insertion effective du département) qui se chargerait d'incrémenter la séquence lors de l'insertion d'un nouvel enregistrement :

Pour créer ce trigger, il faut exécuter la commande suivante :

```
CREATE OR REPLACE TRIGGER trg_insert_dept
  BEFORE INSERT ON exe_dept
  FOR EACH ROW
BEGIN
  :NEW.dep_no := sq_exe_dept.NEXTVAL;
END;
```

On utilise la prochaine valeur de la séquence

comme clé primaire du nouvel enregistrement

Ce trigger se déclenchera automatiquement à chaque nouvelle insertion d'un département, quelle que soit l'origine de l'insertion (une application, un utilisateur avec un outil comme PL/SQL Dev, etc.)