

# Les Séquences

## Mise en œuvre avec Oracle 12c

Jamal MALKI

Département Informatique  
IUT-LR, La Rochelle Université

Janvier 2022

# Overview

---

1. Introduction
2. Créer une séquence
3. Utiliser une séquence
4. Colonne d'identité
5. Administrer les séquences
6. Métabase : sequences
7. Références

# Introduction

---

## Définition

Une séquence est un objet de la base de données qui génère des nombres (numéros) uniques qui pourront être insérés dans une ou plusieurs colonnes d'une table ou plusieurs tables (comme par exemple une Primary Key).

## Exemple

Deux utilisateurs insèrent simultanément de nouveaux employés dans la table EMP. En utilisant une séquence pour produire des nombres uniques des employés pour la colonne EMPNO, ni l'un ni l'autre ne doit attendre pour entrer le prochain nombre disponible des employés. La séquence produit auto des valeurs correctes pour chaque utilisateur.

# Fonctionnement interne des séquences

---

- Les séquences sont indépendantes des tables, ainsi la même séquence peut être employée pour une ou plusieurs tables. Après création, une séquence peut être consultée par divers utilisateurs pour produire des nombres de séquences réels.
- Une séquence incrémente ou décrémenté un compteur dans une routine Oracle, ce qui génère un gain de temps et de codage.
- Il est possible de stocker les valeurs dans un cache, ce qui accélèrera l'accès aux données de la séquence.

# Créer une séquence

## Syntaxe SQL

```
CREATE SEQUENCE nom_sequence  
    [INCREMENT BY n]  
    [START WITH n]  
    [MAXVALUE n | NOMAXVALUE]  
    [MINVALUE n | NOMINVALUE]  
    [CYCLE|NOCYCLE]  
    [CACHE n | NOCACHE];
```

- Pour créer une séquence, il faut posséder le privilège **CREATE SEQUENCE**.
- Le nom d'une séquence est composé du prefix "SEQ" suivi du nom de la table, à laquelle appartient la colonne qu'elle incrémente, et le nom de la colonne qu'elle incrémente. Cette convention de nommage permet de retrouver et identifier facilement les séquences.

# Créer une séquence I

---

Propriété	Description
INCREMENT BY	Pas d'incrémentation de la valeur de la séquence, par défaut 1
START WITH	Départ du compteur, par défaut 1
MAXVALUE — NOMAXVALUE	Valeur maximale du compteur ou pas de valeur max. La valeur max est $10^{27}$ pour une incrémentation et -1 pour une décrémentation
MINVALUE — NOMINVALUE	Valeur minimale du compteur ou pas de valeur min. La valeur minimale sera de 1 pour une séquence ascendante et de $-10^{26}$ pour une séquence descendante

# Créer une séquence II

---

Propriété	Description
CYCLE — NOCYCLE	Autorise la séquence à reprendre à partir de START WITH une fois qu'elle atteint MAXVALUE ou MINVALUE. À ne pas utiliser pour une séquence sur une colonne PK
CACHE n — NOCACHE	Met ou non des valeurs en cache, par défaut 20

# Exemple

---

Une séquence qui incrémente la colonne "empno" de la table "emp" devra s'appeler "seq\_emp\_empno".

## Séquence seq\_emp\_empno

```
CREATE SEQUENCE seq_emp_empno  
    INCREMENT BY 1  
    START WITH 1  
    MAXVALUE 9999  
    MINVALUE 0  
    NOCYCLE  
    CACHE 25;
```



# Utiliser une séquence

Pour récupérer les valeurs d'une séquence, on utilise les pseudo colonnes : NEXTVAL et CURRVAL.

- NEXTVAL stocke de manière automatique la valeur suivante de la séquence dans CURRVAL.
- NEXTVAL renverra une valeur unique à chaque référence à la séquence. Chaque valeur sera unique quel que soit l'utilisateur qui y fait référence.
- Pour récupérer la valeur suivante d'une séquence, il suffira de faire appel à la pseudo colonne NEXTVAL de la manière suivante : `nom_séquence.NEXTVAL`
- Pour récupérer la valeur actuelle d'une séquence, il suffira de faire appel à la pseudo colonne CURRVAL de la manière suivante : `nom_séquence.CURRVAL`.

## Attention

Il est possible de sélectionner NEXTVAL et CURRVAL mais pas de faire d'insertion, de modification ou d'effacement dans ces pseudo colonnes.

# Utiliser une séquence

---

## INSERT

```
INSERT INTO EMP(empno, lastname, firstname)  
VALUES (seq_emp_empno.NEXTVAL, 'Albert', 'Durand');
```

```
INSERT INTO tel(telno, empno)  
VALUES ('0546450030', seq_emp_empno.CURRVAL)
```

## Remarque importante

Une séquence ne revient jamais en arrière même si des annulations (ROLLBACK) sont exécutées. Les valeurs utilisées ne seront pas remises dans le cache. De même si le système plante, les valeurs, stockées en mémoire, seront alors perdues.

# Utiliser une séquence : valeur par défaut d'une colonne

- Il est possible de spécifier les pseudo-colonnes CURRVAL et NEXTVAL comme valeurs par défaut pour une colonne.
- Dans l'exemple suivant, on peut voir l'effet de la spécification d'une séquence comme valeur par défaut pour une colonne. La valeur par défaut est utilisée uniquement lorsque la colonne n'est pas référencée par un INSERT.

## Exemple : valeur par défaut d'une colonne

```
create sequence seq_t1_col1;  
create table t1 (  
    col1 number default seq_t1_col1.nextval,  
    description varchar2 (30)  
);
```

# Utiliser une séquence : valeur par défaut d'une colonne

---

- Les valeurs par défaut ne sont utilisées que lorsqu'une colonne n'est pas référencée dans une instruction INSERT.
- Si la colonne est référencée, même en fournissant la valeur NULL, la valeur par défaut n'est pas utilisée. Oracle 12c vous permet de modifier ce comportement en utilisant la clause ON NULL dans la définition par défaut.

## Exemple : valeur par défaut d'une colonne ON NULL

```
create sequence seq_t2_col1;  
create table t2 (  
    col1 number default ON NULL seq_t2_col1.nextval,  
    description varchar2 (30)  
);
```

# Colonne d'identité

Oracle 12c introduit la possibilité de définir une clause d'identité sur une colonne de table définie à l'aide d'un type numérique.

## Syntaxe SQL

```
GENERATED  
    [ ALWAYS | BY DEFAULT [ ON NULL ] ]  
AS IDENTITY [ ( identity_options ) ]
```

identity\_options ::= presque toutes les options de l'ordre CREATE SEQUENCE

## Attention

Pour créer une colonne d'identité, il faut posséder le privilège CREATE SEQUENCE.

# Colonne d'identité : cas ALWAYS

Utiliser ALWAYS force l'utilisation de l'identité : si une instruction INSERT fait référence à la colonne identity, même pour spécifier une valeur NULL, une erreur est produite.

## Syntaxe SQL

```
SQL> create table identity_test_table (  
    id number GENERATED ALWAYS AS IDENTITY,  
    description varchar2(30)  
);
```

```
SQL> insert into identity_test_table (id, description)  
    values (1, 'desc1')
```

ERROR at line 1:

ORA-32795: cannot insert into a generated always identity column

# Colonne d'identité : BY DEFAULT

---

- L'utilisation de BY DEFAULT permet d'utiliser l'identité si la colonne n'est pas référencée dans l'instruction INSERT.
- Si la colonne est référencée, la valeur spécifiée sera utilisée à la place de l'identité.
- Tenter de spécifier la valeur NULL dans ce cas entraîne une erreur, puisque les colonnes d'identité sont toujours NOT NULL .

## Syntaxe SQL

```
SQL> create table identity_test_table (  
    id number GENERATED BY DEFAULT AS IDENTITY,  
    description varchar2(30)  
);
```

# Colonne d'identité : BY DEFAULT ON NULL

---

L'utilisation de BY DEFAULT ON NULL permet d'utiliser l'identité si la colonne d'identité est référencée, mais une valeur NULL est spécifiée.

## Syntaxe SQL

```
SQL> create table identity_test_table (  
    id number GENERATED BY DEFAULT ON NULL AS IDENTITY,  
    description varchar2(30)  
);
```



# Administrer les séquences : Modifier une séquence

Il peut parfois être intéressant de modifier les caractéristiques d'une séquence, comme sa valeur maximale (MAXVALUE).

## Syntaxe SQL

```
ALTER SEQUENCE nom_sequence  
    [ INCREMENT BY n ]  
    [ { MAXVALUE n | NOMAXVALUE } ]  
    [ { MINVALUE n | NOMINVALUE } ]  
    [ { CYCLE | NOCYCLE } ]  
    [ { CACHE n | NOCACHE } ] ;
```

## Attention

Il n'est pas possible de changer la valeur de départ (START WITH) grâce à la commande ALTER SEQUENCE. Le seul moyen est de supprimer la séquence et de la recréer.

# Administrer les séquences : Supprimer une séquence

---

Pour pouvoir supprimer une séquence de son propre schéma ou de n'importe quel schéma, il faut posséder les privilèges `DROP SEQUENCE` ou `DROP ANY SEQUENCE`.

## Syntaxe SQL

```
DROP SEQUENCE nom_sequence;
```

# Métabase : sequences

- ALL\_SEQUENCES : describes all sequences accessible to the current user.
- DBA\_SEQUENCES : describes all sequences in the database.
- USER\_SEQUENCES : describes all sequences owned by the current user. This view does not display the SEQUENCE\_OWNER column.

## Syntaxe SQL

```
SQL> desc USER_SEQUENCES
```

```
Nom          NULL ? Type
```

```
-----  
SEQUENCE_NAME NOT NULL VARCHAR2(128)
```

```
MIN_VALUE     NUMBER
```

```
MAX_VALUE     NUMBER
```

```
INCREMENT_BY  NOT NULL NUMBER
```

```
CYCLE_FLAG    VARCHAR2(1)
```

```
...
```

```
SQL>
```

# Références

---



Oracle

Database Online Documentation 12c, Release 1 (12.1) / Database Administration / Database Concepts / 24 - Managing Views, Sequences, and Synonyms



Marie-Christine LAFAYE (2003)

Cours Marie-Christine Lafaye sur les Séquences. (2003-2004)

*IUT-LR, Dept INFO 2003*



ISO/IEC (1992)

Second Informal Review Draft, ISO/IEC 9075:1992, Database Language SQL – July, 30, 1992

*ISO/IEC 1992*



Pierre Delmal (2000)

SQL2 – SQL3 et Applications à Oracle, Pierre Delmal

*De Boeck Université 2000*