Plan du module

- **Partie 1- I. Introduction aux SGBDs**
  - Chapitre 1: Présentation des SGBDs
  - Chapitre 2: Rappel. Définition et Evolution des données
  - Chapitre 3: Contrôle des données
  - **Chapitre 4: Gestion des objets utilisateurs**
    - **(Vues, séquences et Index)**
- **Partie 2- II. Langage procédural: PL/SQL**
- **Partie 3- III. Gestion des Transactions**
-

ENICARTHAGE
المدرسة الوطنية للمهندسين بقرطاج
Ecole Nationale d'Ingénieurs de Carthage

I.4 Séquences, Vues & Index

2ème Ingénieurs info

Année Universitaire 2020-2021

A-Les séquences

Plan

- Introduction
- Définitions et objectifs
- Création d'une séquence
- Utilisation d'une séquence
  - NEXTVAL
  - CURRVAL
- Modification d'une séquence
- Suppression d'une séquence
- Consultation d'une séquence
  - Table système DUAL
  - Vue user_sequences du dictionnaire de données

Introduction

- Tous les SGBDs offrent désormais une facilité pour obtenir des identificateurs sans avoir à accéder à une table.
- Cette facilité permet d'obtenir des valeurs qui sont générées automatiquement par le SGBD.
- Cette facilité n'est malheureusement pas standardisée ; par exemple,:
  - MySQL permet d'ajouter la clause AUTO_INCREMENT à une colonne ;
  - DB2 et SQL Server ont une clause IDENTITY pour dire qu'une colonne est un identifiant ;
  - Oracle et PostgreSQL utilisent des séquences.
    - Oracle depuis sa version 12c utilisent aussi des colonnes générées comme IDENTITY

    Mais qui peuvent décrire cette identité à partir des paramètres des séquences (à voir plus trard das le cours)

1. Définitions et objectifs

- Une séquence est un objet, de la base comme les tables, les vues…
- Les séries (*sequences*) sont un excellent moyen d'avoir une base de données qui génère automatiquement des clés primaires entières uniques qui sont incrémentées ou décrémentées par le serveur Oracle (à chaque fois que l'on consulte).
- Elle est créée par l'utilisateur.
- Stockée et Gérée indépendamment d'une table
- Une séquence peut être partagée par plusieurs utilisateurs.

2. Création séquence

```
CREATE SEQUENCE nom-séquence
[INCREMENT BY(1|valeur)]  ➔ 0 n'est pas autorisé
                          Si un entier négatif est spécifié, la série
                          décroîtra dans l'ordre. Un entier positif
fera croître en ordre.
[START WITH valeur]       ➔ par défaut valeur minimale de
                          la séquence
[MAXVALUE valeur| NOMAXVALUE] -- 10^28-1 pr séquence / et à -10^27+1 pr sq
 [MINVALUE valeur| NOMINVALUE] -- 1 ou -10^27-1
[CYCLE| NOCYCLE]
[CACHE(valeur|20) | NOCACHE]; -- éviter de générer des valeurs en temps réel
[ORDER|NORDER]
```

- **CYCLE** indique que la séquence doit continuer de générer des valeurs même après avoir atteint sa limite. Au-delà de la valeur maximale, la séquence générera la valeur minimale et incrémentera comme cela est défini dans la clause concernée et vice versa.
- **NOCYCLE (par défaut)**. Est une option qui interdit à la série de produire des valeurs au-delà des maximum ou minimum définis. C'est la valeur par défaut.
- **ORDER** garantit que les valeurs de la séquence sont générées dans l'ordre des requêtes. Si vos séquences jouent le rôle d'horodatage (timestamp), vous devrez utiliser cette option. Pour la génération de clés primaires, cette option n'est pas importante

3. Utilisation d'une séquence

- Pour utiliser une séquence, on utilise les pseudo-colonnes.
  - nom_seq.**CURRVAL** (renvoie la valeur courante de la séquence), et
  - nom_seq.**NEXTVAL** (incrémente la séquence et retourne la nouvelle valeur).
  - Lors de la première utilisation d'une séquence, il faut utiliser *NEXTVAL* pour l'initialiser.
- ATTETION !!!:
  - Ne jamais utiliser une séquence avec CYCLE pour générer des valeurs de clé primaire
  - La valeur de la séquence peut être perdue:
    - Lorsqu'un enregistrement est supprimé de la table,
    - Lors de l'insertion d'un enregistrement, s'il y a violation d'une contrainte d'intégrité (l'enregistrement n'a pas été inséré)

3. Utilisation d'une séquence

- Exemple1:

```
CREATE SEQUENCE seq1
INCREMENT BY 10
START WITH 5
MAXVALUE 100;
```

- Insertion d'un enregistrement dans une table en utilisant les séquences:

1.
```
INSERT INTO EmployesInfo (numemp, nom) VALUES
(seq1.nextval, 'Ben Brahim');
```

➔

Numemp	nom
5	Ben Brahim

2.
```
INSERT INTO EmployesInfo (numemp, nom) VALUES
(seq1.nextval, 'Jemii');
```

➔

Numemp	nom
5	Ben Brahim
15	Jemii

4. Modification d'une séquence

```
ALTER SEQUENCE [schéma.] nomSéquence
[INCREMENT BY entier ]
[ { MAXVALUE entier | NOMAXVALUE } ]
[ { MINVALUE entier | NOMINVALUE } ]
[ { CYCLE | NOCYCLE } ]
[ { CACHE entier | NOCACHE } ]
[ { ORDER | NOORDER } ] ;
```

- **La clause START WITH ne peut être modifiée sans supprimer et recréer la séquence.**
- Des contrôles sont opérés sur les limites, par exemple MAXVALUE ne peut pas être affectée à une valeur plus petite que la valeur courante de la séquence.
- Exemple 3:
  - *Supposons qu'on a les deux séquences seqEqp et seqPaS mais qu'on ne devra pas stocker plus de 95 000 passagers et pas plus de 850 equipements. De plus les incréments des séquences doivent être égaux à 5.*
  - *Les instructions SQL à appliquer sont les suivantes : chaque invocation des méthodes NEXTVAL prendra en compte désormais le nouvel incrément tout en laissant intactes les données existantes des tables.*

```
ALTER SEQUENCE seqEqp INCREMENT BY 5 MAXVALUE 850;
ALTER SEQUENCE seqPaS INCREMENT BY 5 MAXVALUE 95000;
```

5. Suppression séquence

```
DROP SEQUENCE nom-séquence
```

- Exemple
  - **DROP SEQUENCE** emp_seq;

5. Consultation des valeurs d'une séquence

A-Utilisation de la pseudo-table DUAL

- Table DUAL
  - C'est une particularité d'Oracle.
  - Elle ne contient qu'une seule ligne et une seule colonne
  - Ne peut être utilisée qu'avec une requête SELECT.
  - Elle permet de faire afficher une expression dont la valeur ne dépend d'aucune table en particulier.
- Pour voir la valeur d'une séquence, on utilise currval avec la table DUAL.

5. Consultation des valeurs d'une séquence

- Exemple 4:

```
1. CREATE SEQUENCE ma_sequence; --nouvelle séquence
2. SQL> SELECT ma_sequence.CURRVAL
        FROM DUAL;
   ERREUR à la ligne 1 : ORA-08002: séquence ma_sequence.CURRVAL
   pas encore définie dans cette session
```

- Exemple5: complet

```
SQL> CREATE SEQUENCE ma_sequence START WITH 1 minvalue 0;
Séquence créée.
SQL> SELECT ma_sequence.nextval FROM DUAL;
NEXTVAL
---------
1
SQL> SELECT 'La valeur courante est ' || ma_sequence.currval
FROM DUAL;
----------------------------------------------------
La valeur courante est 1
SQL> ALTER sequence ma_sequence INCREMENT BY 20;
Séquence modifiée.;
SELECT ma_sequence.nextval FROM DUAL;
NEXTVAL --------- 21
```


5. Consultation des valeurs d'une séquence

- Exemple 5 (suite)

```
SQL> SELECT ma_sequence.NEXTVAL + ma_sequence.NEXTVAL from DUAL;
MA_SEQUENCE.NEXTVAL+MA_SEQUENCE.NEXTVAL
---------------------------------------
82 (41+41)
SQL> ALTER sequence ma_sequence INCREMENT BY -41
MAXVALUE 100 cycle nocache;
Séquence modifiée.
SQL> SELECT ma_sequence.nextval from DUAL;
NEXTVAL
---------
0
SQL> SELECT ma_sequence.nextval from DUAL;
NEXTVAL
---------
100
SQL> SELECT ma_sequence.nextval from DUAL
 NEXTVAL
--------
 59
```

5. Consultation des valeurs d'une séquence

- **Utilisation de la vue USER_SEQUENCES du dictionnaire de données**
  - L'utilisateur peut avoir des informations sur les séquences qu'il a créées en consultant la table USER_SEQUENCES du dictionnaire des données.
  - La table ALL_SEQUENCES lui donne les séquences qu'il peut utiliser (même s'il ne les a pas créées)

5. Consultation des valeurs d'une séquence

- **Utilisation de la vue USER_SEQUENCES du dictionnaire de données**
- Exemple 6:

```
SQL> CREATE SEQUENCE test_seq
START WITH 10
INCREMENT BY 5
MINVALUE 10
MAXVALUE 20
CACHE 2 ORDER;
➔ Sequence created.
SQL>
SQL> SELECT * FROM user_sequences
WHERE sequence_name='TEST_SEQ';

SEQUENCE_NAME MIN_VALUE MAX_VALUE INCREMENT_BY CACHE_SIZE LAST_NUMBER
----------------------------- ---------- ---------- ------------ -
TEST_SEQ      10          20         5          2           10
```

➔Last_number : dernier nombre généré.

Colonne auto-incrémentée

- Depuis la version 12c, il est possible d'utiliser un type numérique {entier} pour définir une colonne auto-incrémentée avec la clause GENERATED AS IDENTITY disponible dans les instructions CREATE TABLE et ALTER TABLE :

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

- ALWAYS (par défaut) utilise le générateur de séquences et interdit qu' une valeur soit explicitement imposée lors d'un INSERT ou UPDATE (erreur ORA- 32795 impossible d'insérer la valeur dans une colonne d'identité...).
- BY DEFAULT utilise le générateur de séquences mais n'interdit pas qu'une valeur soit explicitement imposée d'un INSERT ou UPDATE.
- Avec l'option ON NULL, la séquence est capable d'affecter implicitement une valeur à chaque INSERT ou si un quelconque NULL arrive en lieu et place de la colonne concernée.
- *options_sequence* sont identiques à celles du CREATE SEQUENCE.

Exemple

- Le code suivant présente l'utilisation de cette option pour une clé primaire. Il est à noter que le NOT NULL est implicite sur une telle colonne et que vous ne pouvez disposer que d'un seul auto-incrément par table.

Création de la table et de son auto-incrémentation

```
CREATE TABLE billets (
  id   NUMBER(6)
       GENERATED ALWAYS
       AS IDENTITY,
  vol_id     VARCHAR2(6) NOT NULL,
  jour_vol   DATE DEFAULT SYSDATE NOT NULL,
  pax_nom    VARCHAR2(30) NOT NULL,
  siege_pax CHAR(3) NOT NULL ,
  CONSTRAINT pk_billets PRIMARY KEY(id));
```

Insertions

```
INSERT INTO billets(vol_id,pax_nom,siege_pax)
  VALUES          ('AF6143','Guilbaud','03F');
INSERT INTO billets(vol_id,pax_nom,siege_pax)
  VALUES          ('AF6145','Blanchet','23B');
INSERT INTO billets(vol_id,pax_nom,siege_pax)
  VALUES          ('AF6145','Bruchez','02A');
```

- →

```
SQL> SELECT * FROM billets;

        ID VOL_ID JOUR_VOL PAX_NOM                        SIEGE_PAX
---------- ------ -------- ------------------------------ ----------
         1 AF6143 20/11/14 Guilbaud                       03F
         2 AF6145 20/11/14 Blanchet                       23B
         3 AF6145 20/11/14 Bruchez                        02A
```