Exploitation d'une BD

CM3-1 : SQL, intégrité et LDD

Mickaël Martin Nevot

V1.1.0

Exploitation d'une BD

Rappel : pour tester

- Interpréteur en ligne : https://livesql.oracle.com
- Documentation : https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf

En SQL, les retours à la ligne sont non déterministes

Rappel : convention de nommage

- Conseillée dans le **monde professionnel** :
  - Relations et attributs : *lower snake case*

Snake snake_case

Données enregistrées / resituées en majuscules : nommage préservé, confusions évitées

- Appliquée dans le **cadre de l'enseignement** :
  - Relations : *pascal case*
  - Attributs : *lower case*

Tableau blanc / feuille : plus court à écrire et lisible
Outil numérique : plus facile à sélectionner

Pascal PascalCase

L'usage des *backquotes* n'est pas standard SQL et provoque une erreur

LDD

- Langage de description de données (LDD)
- Spécification du **schéma conceptuel** d'une BD :
  - Base de données
  - Relations
  - Contraintes
- Spécification des **vues** d'une BD

Permet de créer des « objets » SQL

Syntaxe

- **Dénomination** (identifieurs et mots clef) :
  - Insensibles à la casse (majuscules conseillées pour mots clefs)
  - Caractères spéciaux interdits ou à éviter
  - Ne pas utiliser les mots clefs comme identifieurs
- **Commentaires** :
  - Ligne : `--`
  - Bloc : `/* ... */`
- **Chaînes de caractères** :
  - Guillemets simples : `'Ceci est une chaîne'`
- `NULL` :
  - Valeur nulle, information inconnue ou incomplète

Pas `0` ou `""`

Les types de données

Il existe d'autres types

Type		Taille mémoire	À la norme
NUMBER(p, s)	Entier et décimal	1 - 22 octets	✗
LONG	Entier	2^{31} -1 octets	✗
FLOAT(p)	Décimal	1 - 22 octets	✓
DATE	De 4712 av. J.-C. à 9999 ap. J.-C.	7 octets	✓
TIMESTAMP(p) ...	Date + frac. sec.	7 – 13 octets	✓
CHAR(n)	Chaîne (taille fixe)		✓
VARCHAR2(n)	Chaîne (taille var.)		✗
CLOB	Longue chaîne		✗
BLOB	Longue chaîne binaire		✗

Fuseaux horaires possibles

Taille maximale : en nombre de caractères

Pas de UNSIGNED et de ZEROFILL

Quel type de caractères ?

Taille fixe

- Avantages :
  - Accès direct facile (rapide)
  - Pas de fragmentation
  - Réparation facile de table
- Inconvénient :
  - Pas d'économie de place

`CHAR(...)`

Taille variable

- Avantage :
  - Gain de place
- Inconvénients :
  - Insertion plus longue (car calcul de la taille exacte)
  - Fragmentation inévitable
  - Temps d'accès : moindre performance

`VARCHAR2(n), CLOB, BLOB`

Les commandes

- Création de BD
- Destruction de BD
- Création des tables
- Destruction des tables
- Modifications de la structure des tables
- Création de clefs étrangères

Création/destruction de BD

- CREATE DATABASE :

Syntaxe :
```
CREATE DATABASE Db_name [...]
```

- DROP DATABASE :

Syntaxe :
```
DROP DATABASE
```

  - Efface tous les fichiers des tables

Création/destruction de tables

- CREATE : création de table

Syntaxe :
```
CREATE [...] TABLE [...] Tbl_name [...]
([<CREATION_CLAUSE> [, ...]]) [...]
```

<CREATION_CLAUSE> :
```
column_name type [...] [column_constraint [...]]
```

Une table est créée par défaut dans le schéma public (hors cadre du cours)

```
CREATE TABLE Etudiant (
    nom CHAR(32),
    prenom VARCHAR2(32)
);
```

- DROP : destruction définitive d'une table

Syntaxe :
```
DROP TABLE [...] Tbl_name [...]
```

- TRUNCATE : effacement de toutes les données d'une table

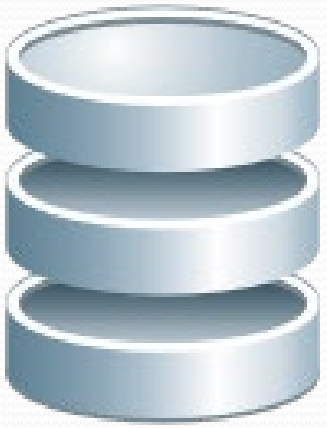
Syntaxe :
```
TRUNCATE TABLE [...] Tbl_name [...]
```

Modification de tables

- ALTER TABLE :

Syntaxe :
```
ALTER TABLE [...] Tbl_name [...] <action>
```

  - Exemple d'<ACTION> :
    - ADD col CREATION_CLAUSE
    - MODIFY col CREATION_CLAUSE DEFAULT value
    - DROP COLUMN col
    - RENAME TO Tbl_name
    - Etc.



Rappel : clefs

- **Clef candidate, potentielle** (rappel) : ensemble des données permettant d'indexer chaque ligne de manière différenciée
- **Clef primaire** : Obligatoire
  - **Une seule par relation** (clef candidate retenue comme primaire)
  - **Simple** (un seul attribut) ou **composée** (plusieurs attributs)
  - **Unique** et **non nulle**
- **Clef étrangère** :
  - Clef primaire d'une autre relation de la BD

Employe			
nume	nome	daten	numd
1	Dupond	1/12/80	1
2	Jacques	21/04/68	1
3	Martin	03/25/52	2

Departement	
numd	nomd
1	ISMIN
2	ICM

Rappel : contraintes

- Types de contraintes :
  - Contraintes d'intégrité :
    - **Clef primaire**
    - **Clef étrangère**
  - Contraintes de valeurs :
    - **Non nullité**
    - **Unicité** (une valeur donnée n'apparait qu'une fois)
- Définitions de contraintes :
  - **Contraintes d'attributs** (spécifiques à un attribut donné)
  - **Contraintes de tables** (portent sur plusieurs attributs)

Les contraintes apportent de la cohérence

Il est possible de nommer une contrainte

Rappel : contraintes d'intégrité

- Intégrité **d'entité** (ou de relation) :
  - Garanti un attribut (donc l'extension) sans doublon
- Intégrité **référentielle** :
  - Impose que toute valeur de la clef est une valeur de clef primaire de la relation associée
- Intégrité **sémantique** :
  - Pas toujours modélisable au niveau du schéma relationnel
  - *Triggers*
- Intégrité **applicative** :
  - Extérieure à la BD : liée à l'application

Des restrictions existent sur les mises à jour

Contraintes d'attributs

```
CREATE TABLE Etudiant (
    ide NUMBER(38) PRIMARY KEY,
    nom VARCHAR2(32) NOT NULL,
    prenom VARCHAR2(20),
    email VARCHAR2(25) UNIQUE,
    daten DATE,
    annee NUMBER(1) CHECK (annee < 4),
    tel CHAR(14)
        CONSTRAINT ck_etu_tel
        CHECK (REGEXP_LIKE(tel, '[0-9][0-9]-[0-9][0-9]-[0-9][0-9]-[0-9][0-9]-[0-9][0-9]')),
    sexe CHAR(1) DEFAULT 'F'
);
```

Clef primaire

Non nullité

Unicité

Vérification, validation

Une contrainte peut être nommée (pk, fk, uq, nn, ck)

Valeur par défaut (si non renseignée)

Ce n'est pas une contrainte, mais la syntaxe est la même

Contraintes de tables

```
CREATE TABLE Convention (
    ide NUMBER(38) NOT NULL,
    ids NUMBER(38) NOT NULL,
    datec DATE,
    date_deb DATE,
    duree NUMBER(38),
    CONSTRAINT pk_con PRIMARY KEY (ide, ids),
    CONSTRAINT fk_con_etu FOREIGN KEY (ide) REFERENCES Etudiant(ide)
        -- Impactée (supprimée) comme la clef primaire.
        ON DELETE CASCADE,
    CONSTRAINT fk_con_soc FOREIGN KEY (ids) REFERENCES Societe(ids)
        -- Mise à nulle lorsque la clef primaire est impactée.
        ON DELETE SET NULL,
    CONSTRAINT uk_con_01 UNIQUE (ide, date_deb),
    CONSTRAINT ck_con_dat CHECK (datec < date_deb)
);
```

Clef primaire (composée)

Clef étrangère

Unicité

Vérification, validation

Exemple de création de table

```
CREATE TABLE Etudiant (
    ide NUMBER(38) NOT NULL PRIMARY KEY,
    nom VARCHAR2(25) NOT NULL,
    prenom VARCHAR2(20),
    email VARCHAR2(25) UNIQUE,
    sexe CHAR(1)
        CONSTRAINT ck_etu_sex
        CHECK (sexe IN ('M', 'F') OR sexe IS NULL),
    daten DATE,
    adresse VARCHAR2(60),
    annee NUMBER(1) DEFAULT 3 CHECK (annee < 4),
    tel CHAR(14)
        CONSTRAINT ck_etu_tel
        CHECK (REGEXP_LIKE(tel, '[0-9][0-9]-[0-9][0-9]-[0-9][0-9]-[0-9][0-9]-[0-9][0-9]')),
    CONSTRAINT uk_etu_01 UNIQUE (nom, prenom)
);
```

Ecriture de données

LMD

- INSERT :

Syntaxe :

```
INSERT [...] INTO Tbl [...] [(col [, ...])] [...] VALUES (expr | DEFAULT [, ...]) [...]
```

```
INSERT INTO Etudiant (ide, nom, prenom) VALUES (1, 'Dupont', 'Leon');
INSERT INTO Etudiant VALUES (2, 'Martin', 'Michel');
INSERT INTO EtudiantClone SELECT * FROM Etudiant; -- La table doit être crée préalablement
```

- UPDATE :

Syntaxe :

```
UPDATE [...] Tbl [...] SET col = expr [, ...] [...] [WHERE condition ...] [...]
```

```
UPDATE Societe SET raisons = 'SA' WHERE raisons = 'SARL';
```

- DELETE :

Syntaxe :

```
DELETE [...] FROM Tbl [...] [WHERE condition ...] [...]
```

```
DELETE FROM Convention WHERE datec < '2020-01-01';
```

Attention : sans critère de filtrage, tous les enregistrements sont concernés !

Auto incrémentation

- Oracle 11- :

Varie grandement d'un SGBD à un autre

Crée une séquence Etudiant_ide_seq (hors cadre du cours)

```
CREATE SEQUENCE Etudiant_ide_seq START WITH 1 INCREMENT BY 1;

INSERT INTO Etudiant (ide, nom) VALUES (Etudiant_ide_seq.nextval, 'Antonio Paz');

-- Change la valeur courante de la séquence.
ALTER SEQUENCE Etudiant_ide_seq INCREMENT BY 42;
```

- Oracle 12+ :

```
CREATE TABLE Etudiant (
    ide NUMBER(38) GENERATED ALWAYS AS IDENTITY(START with 1 INCREMENT by 1),
    nom VARCHAR(64) NOT NULL
);

INSERT INTO Etudiant (ide, nom) VALUES (DEFAULT, 'Lilliana Angelovska');

-- Change la valeur courante de la séquence.
ALTER TABLE Etudiant MODIFY ide GENERATED BY DEFAULT ON NULL AS IDENTITY (START WITH 42);
```

Liens

- Document classique :
  - Laurent Carmignac. *Programmation et administration des bases de données.*

Crédits

Auteur

Mickaël Martin-Nevot
mmartin.nevot@gmail.com

- Laurent Carmignac

Carte de visite électronique

Relecteurs

Cours en ligne sur : www.mickael-martin-nevot.com