Introduction à l'administration de bases de données

DDL (Data Definition Language)

DCL (Data Control Language)

DML (Data Manipulation Language)

Création de tables

- Jusqu'ici on a vu
  - La sélection de tuples d'une table (DML->select)

- Dans ce cours
  - Création, suppression, modification de tables et de contraintes d'intégrités (DDL->CREATE, DROP, ALTER)
  - Droits d'accès aux tables (DCL-> GRANT, REVOKE)
  - Insertion, modification et suppression de tuples (DML->INSERT, DELETE, UPDATE)

DDL
(Data Definition Language)

Définition de schémas

- Un schéma de base de données est composé de tables et de vues

- Création de tables
  - CREATE TABLE <nomTable> (<éléments de la table>$^{+}$)

  - <élément de table> ::= <définition de colonne> |
    <contrainte de table>

  - <définition de colonne> ::= <nomcolonne> <type>
    [<clause défaut>] [<contrainte de colonne>]

  - <contrainte de table> dans la suite...

Exemple

- CREATE TABLE Personnes (
  numsecu NUMBER(20),
  nom VARCHAR2(20),
  prenom VARCHAR2(20),
  age NUMBER(3));

Contraintes d'intégrité

- Une contrainte contraint les valeurs insérées dans une colonne ou un groupe de colonnes d'une table

- Contraintes de colonne
  - Contraintes portant sur une colonne
  - Peuvent être ajoutées dans la définition de la colonne ou à la fin de la création de la table

- Contraintes de table
  - Contraintes portant sur plusieurs colonnes
  - Définies à la fin de la création de la table

- Contrainte CHECK
  - Contrainte qui peut avoir une condition logique à vérifier
  - Peut être contrainte de colonne ou de table

Contraintes d'intégrité

- Les contraintes de colonnes permettent de spécifier de contraintes portant sur un seul attribut
  - Valeur nulle impossible -> NOT NULL
    -> nom VARCHAR2(20) NOT NULL
  - Unicité -> UNIQUE ou PRIMARY KEY
    -> numsecu NUMBER(20) PRIMARY KEY
  - Contrainte référentielle ou clé étrangère -> REFERENCES <table référencée> [colonne référencée]
    - Le nom de la colonne référencée est optionnel s'il est identique à celui de la colonne référençante

    -> numsecu NUMBER(3) REFERENCES Personnes
  - Contrainte générale -> CHECK <condition>
    -> valeur NUMBER(2) CHECK (valeur > 0 and valeur <10)

Contraintes d'intégrité

- Les contraintes de table peuvent porter sur plusieurs attributs
  - Contrainte d'unicité -> UNIQUE ou PRIMARY KEY <attribut>$^{+}$

    ->UNIQUE (numsecu, poste, salaire)
  - Contrainte référentielle -> [FOREIGN KEY (<colonne référençante>$^{+}$)] REFERENCES <table référencée> [(<colonne référencée>$^{+}$)]

    -> num_client number(2) references clients
  - Contrainte générale -> CHECK <condition>

    -> CHECK (nom<>prenom)

Exemples

- CREATE TABLE Personnes (
  numsecu NUMBER(20) PRIMARY KEY,
  nom VARCHAR2(20) NOT NULL,
  prenom VARCHAR2(20),
  age NUMBER(3) DEFAULT 0 CHECK (age BETWEEN 0 AND 140));

- CREATE TABLE Occupation (
  numsecu NUMBER(3) REFERENCES Personnes,
  poste VARCHAR2(20),
  salaire NUMBER(6),
  PRIMARY KEY (numsecu, poste, salaire));

- CREATE TABLE Personnes1 as
  SELECT * FROM Personnes;

Exemples

- CREATE TABLE Taxes(
  id PRIMARY KEY,
  valeur NUMBER(2) CHECK (valeur>0 and valeur<12));

- CREATE TABLE Salaires(
  Numsecu VARCHAR2(30) PRIMARY KEY,
  id_taxe number(2) references Taxes(id));

- CREATE TABLE client(nom varchar2(20) PRIMARY KEY,
  prenom varchar2(20),
  CHECK (nom<>prenom));

Contraintes nommées

- Les contraintes ont toujours un nom
  - Le nom désigné par Oracle
    - SYS_######## par exemple SYS_C000145
    - L'inconvénient est que ce nom ne permet pas à l'utilisateur d'identifier la contrainte facilement
  - Le nom désigné au moment de la création de la contrainte
    - Ce nom, par convention, doit faire référence au nom de la table, à la colonne(s) en question et à la nature de la contrainte
    - CLIENT_PK, PERSONNES_NOM_PRENOM

Exemple

- CREATE TABLE client(
  nom VARCHAR2(20) CONSTRAINT CLIENT_PK PRIMARY KEY,
  Prenom VARCHAR2(20),
  CONSTRAINT CLIENT_NOM_PRENOM CHECK (nom<>prenom));

- CREATE TABLE Taxes(
  id NUMBER(2) CONSTRAINT TAXES_PK PRIMARY KEY,
  valeur NUMBER(2) CONSTRAINT TAXES_VALEUR CHECK (valeur>0 and valeur<12));

- CREATE TABLE Salaires(
  Numsecu VARCHAR2(30) CONSTRAINT SALAIRES_PK PRIMARY KEY,
  id_taxe NUMBER(2) CONSTRAINT SALAIRES_IDTAXE-TAXES_ID REFERENCES Taxes(id));

Suppression de tables

- Les tables peuvent être supprimées avec
  DROP TABLE <nomTable>;

- Si les tables ont de contraintes référentielles, la suppression doit être faite dans l'ordre dicté par les contraintes
  DROP TABLE Occupation;
  DROP TABLE Personnes;

- Il existe la possibilité de forcer la suppression mais cela doit être utilisé avec précaution
  DROP TABLE Personnes CASCADE CONSTRAINTS

Modification d'un schéma

- Différents types d'altération sont possibles

  ALTER TABLE nomTable ALTERATION

  - Activation/désactivation d'une contrainte -> ENABLE/ DISABLE
  - Ajout d'une contrainte -> ADD CONSTRAINT
  - Suppression d'une contrainte -> DROP CONSTRAINT
  - Ajout d'une colonne -> ADD COLUMN
  - Modification d'une colonne -> ALTER COLUMN
  - Suppression d'une colonne -> DROP COLUMN

Désactivation de contraintes

- Les contraintes peuvent être désactivés après leur création
  - DISABLE
    - Désactivation d'une contrainte. ALTER TABLE avec DISABLE

      ALTER TABLE <nomTable>
      DISABLE CONSTRAINT <nomContrainte>;

  - ENABLE
    - Réactivation d'une contrainte. ALTER TABLE avec ENABLE

      ALTER TABLE <nomTable>
      ENABLE CONSTRAINT <nomContrainte>;

- ATTENTION ! La désactivation d'une contrainte peut entraîner des inconsistances dans le contenu des tables concernées par la contrainte.

Création de contraintes après création des tables

```
CREATE TABLE client(
    nom VARCHAR2(20),
    Prenom VARCHAR2(20));

ALTER TABLE client ADD CONSTRAINT CLIENT_PK
    PRIMARY KEY (nom);

ALTER TABLE client ADD CONSTRAINT
    CLIENT_NOM_PRENOM CHECK (nom<>prenom);
```

Exemples

- ALTER TABLE Personnes ADD(
  situationFamille VARCHAR2(10));

- ALTER TABLE nomTable MODIFY(
  situationFamille VARCHAR2(15) DEFAULT ‘celibataire’);

- ALTER TABLE nomTable MODIFY(
  situationFamille VARCHAR2(15) NOT NULL);

- ALTER TABLE Personnes ADD CONSTRAINT checkSituation
  CHECK (situationFamille IN ('celibataire', 'marie', 'pacse'));

- ALTER TABLE Personnes DROP CONSTRAINT checkSituation;

DCL
(Data Control Language)

Droits d'accès

- La gestion des droit d'accès aux tables est décentralisée
  - Pas d'administration globale
  - Chaque créateur de table obtient tous les droits d'accès (SELECT, INSERT, UPDATE, REFERENCES)
  - Le créateur peut ensuite passer ses droits sélectivement ou à tout le monde (PUBLIC)
  - Un droit peut être passé avec le droit de le transmettre (WITH GRANT OPTION) ou non
  - Les droits peuvent aussi être retirés

Droits d'accès

- GRANT <privilèges> ON <nomTable> TO <récepteur>$^{+}$ [WITH GRANT OPTION]

- <privilèges> ::= ALL PRIVILEDGES | <action>$^{+}$

- <action> ::= SELECT | INSERT | UPDATE [(<nom colonne>$^{+}$)] | REFERENCE [(<nom colonne>$^{+}$)]

- <récepteur> ::= PUBLIC | <identifiant d'autorisation>

Suppression de droits d'accès

- Les droits peuvent être retirés avec REVOKE

- REVOKE <privilèges> ON <nomTable> FROM <récepteur>
- Exemples
  - GRANT SELECT ON Personnes TO PUBLIC;
  - GRANT UPDATE ON Personnes TO bio01, bio02, bio03
  - GRANT INSERT ON Personnes TO bio01, bio02, bio03 WITH GRANT OPTION;
  - REVOKE INSERT ON Personnes FROM bio01, bio02, bio03;

DML
(Data Manipulation Language)

Les mises à jour

- Insertion de nouvelles tuples dans une table
  - INSERT INTO <nomTable> [(<nom de la colonne>$^{+}$)] {VALUES (<constante>$^{+}$) | <commande de recherche>}
- Modification de tuples
  - UPDATE <nomTable>
    SET {<nom de la colonne>={<expression de valeur> | NULL}}$^{+}$
    WHERE {<condition de recherche> | CURRENT OF <nom de curseur>}
- Suppression de tuples
  - DELETE FROM <nomTable>
    [WHERE {condition de recherche} | CURRENT OF <nom de curseur>]

Exemples

- INSERT INTO Personnes
  VALUES(123, 'Le Blanc', 'Joseph', 40);

- INSERT INTO Personnes (numsecu, nom)
  VALUES (124, 'Martin');

- INSERT INTO Personnes1
  SELECT numsecu, nom, prenom, situationFamille
  FROM Personnes
  WHERE numsecu=123;

- INSERT INTO Personnes1
  SELECT *
  FROM Personnes
  WHERE numsecu=123;

Exemples

- UPDATE Personnes
  SET (age=45) WHERE numsecu=124;

- DELETE FROM Personnes1;

- DELETE FROM Personnes
  WHERE age > 40;

Comment faire pour créer ces tables ?

Étudiants

noetu	nom	prénom
28936E	Dupont	Franck
46283B	Dupont	Isabelle
86719E	Martin	Adrien
99628C	Robert	Adrien
99321C	Denou	Michelle
99322C	Dupont	Isabelle

Notes

noe	codemat	noteex	notecc
99628C	MIAS2I5	12	15,5
46283B	MIAS2I6	8	11
46283B	MIAS2I5	9,5	2
86719E	IUP2MA	12	5,5
99321C	LIL6	18	16,5
28936E	MIAS2I5	13,5	13,5
86719E	IUP2IS	8,5	10
99628C	MIAS2I6	3	6
99321C	LIL5	15	14,5
99322C	MIAS2I5	12	15,5
28936E	MIAS2I6	12	null

Matières

codemat	titre	responsable	diplôme
MIAS2I5	I5	E238	Deug MIAS
MIAS2I6	I6	E426	Deug MIAS
IUP2MA	Automates	E238	Licence IUP-MIAGE
LIL6	Systèmes	E236	Licence Informatique
IUP2IS	Systèmes	E526	Licence IUP-MIAGE
MIAS2I3	Math-Info	E426	Deug MIAS
LIL5	Algo	E426	Licence Informatique

Description

- Etudiants
  - noetu : type varchar2 et clé
  - nom : type varchar2
  - prenom : type varchar2
- Matieres
  - codemat : type varchar2 et clé
  - titre : type varchar2
  - responsable : type varchar2
  - diplôme : type varchar2
- Notes
  - noe : référence Etudiants.noetu
  - codemat : référence Matieres.codemat

  } Clé composée
  - noteex : type number
  - notecc : type number

Et les données existantes dans la table ?

- Problème
  - Quoi faire des données existantes ne respectant pas les contraintes rajoutées ?

- État des contraintes
  - VALIDATE
    - Les données existantes doivent être conformes à la contrainte
  - NOVALIDATE
    - Les données existantes peuvent ne pas être conformes à la contrainte

- Combinaison avec ENABLE/DISABLE