TP 2 : DECOUVRIR SQL

OBJECTIFS :

Manipuler le langage de définition de données de SQL Oracle avec des contraintes en se basant sur un exemple de base de données.

Partie I: *Présentation d'SQL*

- Le langage SQL (Structured Query Language) en français langage de requêtes structurées, est un langage de gestion de base de données relationnelles intégré dans la majorité des SGBDR.
- SQL est développé par IBM dans les années 70
- Il est basé sur l'algèbre relationnel (opérations ensemblistes et relationnelles).
- SQL est composé de 4 sous ensembles :
  - ***Le langage de définition de données (LDD)*** : Ce langage permet la définition et mise à jour de la structure de la base de données (tables, attributs, index, etc.)
  - ***Le langage de manipulation de données (LMD)*** : Ce langage permet de manipuler les données de la base : insertion, suppression et mise à jour,
  - ***Le langage d'interrogation de données (LID)*** : Ce langage permet de rechercher des informations utiles dans la Base de données en utilisant des critères de recherche.
  - ***Le langage de contrôle de données (LCD)*** : Ce langage permet de définir les droits d'accès pour différents utilisateurs de la base de données, donc il permet gérer la sécurité de la base et de confirmer ou annuler las transactions.

SQL			
LDD	**LMD**	**LID**	**LCD**
Create **Alter** **Drop**	**Insert** **Update** **Delete**	**Select**	**Grant, Revoke** **Commit,Rollback**

Partie II: *LDD*

II.1 Create Table:

Pour créer une table il est nécessaire de posséder le privilège **CREATE TABLE** et d'utiliser la syntaxe de création d'une table:

```
CREATE TABLE nomTable
(
  colonne1 type1 [contrainte],
  colonne2 type2 [contrainte],
  ....,
  [CONSTRAINT nomContrainte1 typeContrainte1]
  ....
) ;
```

Avec Oracle voici les différentes contraintes :

- **NOT NULL :** L'attribut doit avoir une valeur.
- **UNIQUE :** doublon non autorisé
- **PRIMARY KEY :** clé primaire
- **DEFAULT :** permet de définir une valeur par défaut de la colonne.
- **FOREIGN KEY :** établit une relation de clé étrangère entre une colonne et la colonne dans la table référencée.
- **CHECK :** définit la condition que doit satisfaire chaque ligne de la table.

Type :

Type	Description	Commentaire pour une colonne
CHAR(*n* [BYTE | CHAR])	Chaîne fixe de *n* caractères ou octets.	Taille fixe (complétée par des blancs si nécessaire). Maximum de 2000 octets ou caractères.
VARCHAR2(*n* [BYTE|CHAR])	Chaîne variable de *n* caractères ou octets.	Taille variable. Maximum de 4000 octets ou caractères.
NCHAR(*n*)	Chaîne fixe de *n* caractères Unicode.	Taille fixe (complétée par des blancs si nécessaire). Taille double pour le jeu AL16UTF16 et triple pour le jeu UTF8. Maximum de 2000 caractères.
NVARCHAR2(*n*)	Chaîne variable de *n* caractères Unicode.	Taille variable. Mêmes caractéristiques que NCHAR sauf pour la taille maximale qui est ici de 4000 octets.
CLOB	Flot de caractères (CHAR).	Jusqu'à 4 gigaoctets.
NCLOB	Flot de caractères Unicode (NCHAR).	Idem CLOB.
LONG	Flot variable de caractères.	Jusqu'à 2 gigaoctets. Plus utilisé mais fourni pour assurer la compatibilité avec les anciennes applications.
BLOB	Données binaires non structurées.	Jusqu'à 4 gigaoctets.
BFILE	Données binaires stockées dans un fichier externe à la base.	idem.
NUMBER[(*t*,*d*)]	Valeur numérique de *t* chiffres dont *d* décimales.	Maximum pour *t* + *d* : 38. Espace maximum utilisé : 21 octets.
DATE	Date et heure du 1er janvier 4712 avant JC au 31 décembre 4712 après JC.	Sur 7 octets. Le format par défaut est spécifié par le paramètre NLS_DATE_FORMAT.

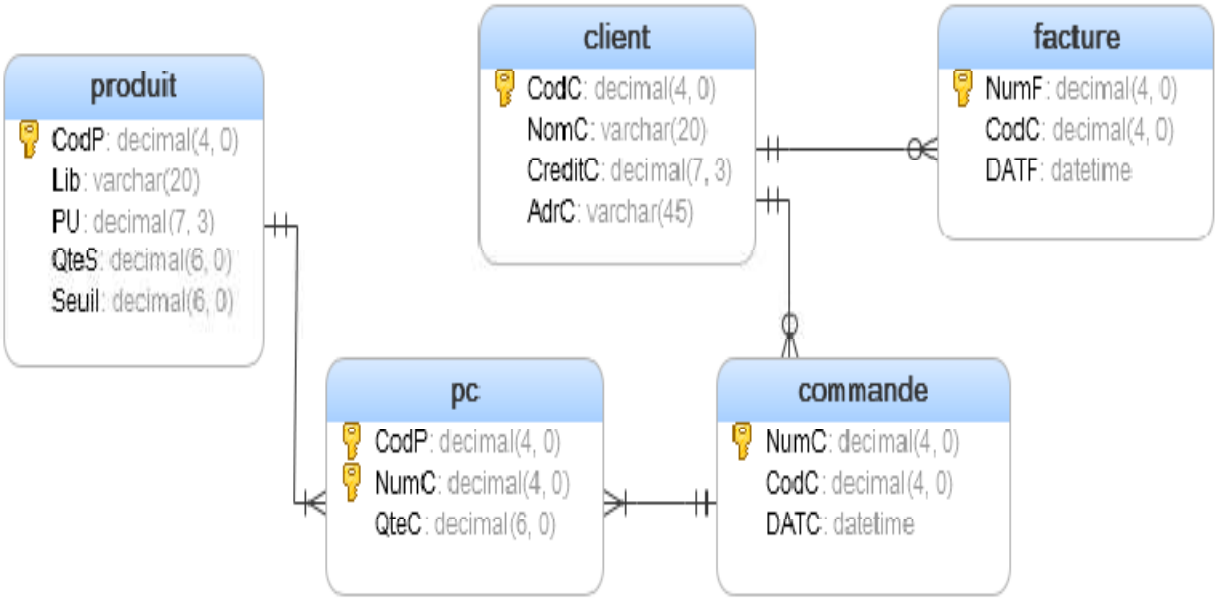
Application:

1. On désire créer une base de données nommée Vente dont le schéma relationnel est le suivant :

   Produit (CodP, Lib, PU, QteS, Seuil)
   Client (CodC, NomC, CreditC, AdrC)
   Commande (NumC, DatC, #*CodC*)
   Facture (NumF, MontF, DatF,#*CodC*)
   PC (#CodP, #NumC, QteC)

   CodC : représente le code client
   NumC : représente le numéro de la commande

La représentation des tables est comme suit :



2. Exécuter les commandes suivantes à partir de l'invite **SQL>** afin de créer les tables et les contraintes sur les clés étrangères (Respecter la syntaxe de chaque commande)

```
-- ============================================================
-- Script pour la création de la base
-- Nom de la base : VenteC
-- Nom de SGBD : ORACLE version 10g
-- Date de création : 16/09/2014 11:15
-- ============================================================
-- Table : Prodit
-- ============================================================
create table Produit
(
   CodP NUMBER(4) not null,
   Lib VARCHAR2(20),
   PU NUMBER(8,3),
   QteS NUMBER(6),
   Seuil NUMBER(4),
   constraint cp_CodP primary key (CodP)
);
  -- ============================================================
  -- Table : Client
  -- ============================================================
  create table Client
  (
     CodC NUMBER(4) not null,
     NomC VARCHAR2(20),
     CreditC NUMBER(8,3),
     AdresseC VARCHAR2(15),
     constraint cp_CodC primary key (CodC)
  );
```

```
-- ============================================================
-- Table : Commande
-- Avec une clé étrangère CodC
-- ============================================================
create table Commande
(
   NumC NUMBER(4) not null,
   CodC NUMBER(4) not null,
   MontC NUMBER(7,3),
   DATC DATE,
   constraint cp_NumC primary key (NumC)
);
-- ============================================================
-- Table : Facture
-- Avec une clé étrangère CodC
-- ============================================================
create table Facture
(
   NumF NUMBER(4) not null,
   CodC NUMBER(4) not null,
   MontF NUMBER(8,3),
   DATF DATE,
   constraint cp_NumF primary key (NumF)
);
-- ============================================================
-- Table : PC
-- ============================================================
create table PC
(
   CodP NUMBER(4) not null,
   NumC NUMBER(4) not null,
   QteC NUMBER(4),
   constraint cp_PC primary key (CodP, NumC)
);
```

3. Vérifier la création de ces tables avec la commande :

```
Select * from tab ;
```

4. Afficher la description de la table Produit :

```
SQL>Desc Produit;
```

II.2 Alter Table:

- **Ajout d'un attribut:**

ALTER TABLE *nom_table*
ADD *attribut* **type;**

- **Modification d'un attribut:**

ALTER TABLE *nom_table*
MODIFY *attribut* **Nouveautype;**

- **Suppression d'un attribut:**

```
ALTER TABLE nom_table
DROP   attribut ;
```

- **Ajout d'une contrainte:**

```
ALTER TABLE nom_table
ADD CONSTRAINT     nom_contrainte
                   Constraint_type (nom_column);
```

- **Suppression d'une contrainte:**

```
ALTER TABLE nom_table
DROP CONSTRAINT     nom_contrainte ;
```

Application:

Soit la base de données Vente représentée par les tables récemment créées:

1. Modifier la structure de la table Commande afin de rajouter la contrainte (ce_Commande_CodC) relative à le clé étrangère CodC. S'assurer que cette clé étrangère possède la propriété "NOT NULL".

```
alter table Commande
add constraint cp_Commande_CodC foreign key (CodC)
references Client (CodC);
```

2. Modifier la structure de la table Commande afin de rajouter la contrainte (ce_Facture_CodC) relative à le clé étrangère CodC. S'assurer que cette clé étrangère possède la propriété "NOT NULL".

```
alter table Facture
add constraint cp_Facture_CodC foreign key (CodC)
references Client (CodC);
```

3. Modifier la structure de la table PC afin de rajouter les contraintes de clés étrangères relatives à NumC et CodP. Soient ce_PC_NumC et ce_PC_CodP les noms respectifs de ces contraintes.

```
alter table PC
add constraint ce_PC_NumC foreign key (NumC)
references Commande (NumC);
```

4. Vérifier la création de ces tables avec la commande :

```
Select * from tab ;
```

5. Copier le fichier VenteC.sql dans votre dossier de travail C:\BD (si le dossier n'est pas créé, il faut le créer)

6. Exécuter maintenant la commande suivante à partir de l'invite SQL>

```
SQL>start C:\BD\VenteC.sql;
```

   Que remarquez vous?

7. Vérifier une deuxième fois la création de toutes les tables avec la commande :

```
SQL>Select * From Tab ;
```

8. Afficher la description de la table **Produit** :

```
SQL>Desc Produit;
```

9. Pour des besoins de suivi des clients d'autres informations s'avèrent utiles. Modifier la structure de la table **Client** en rajoutant les attributs suivants :

Attribut	Type	Longueur	Description
CA	Numérique	10,3	Chiffre d'affaire Client
CredMax	Numérique	10,3	Crédit maximum autorisé Client

10. Modifier l'attribut **AdrC** de la table **Client** afin d'avoir une longueur égale à 50.

11. Ajouter une contrainte (**ck_Cred**) à la table **Client** afin d'assurer un contrôle lors de l'introduction des données de l'attribut **Cred** tel que (**Cred <= CredMax**).

12. Ajouter les contraintes **ck_Qtes**, **ck_Seuil**, **ck_Qtec** aux attributs **Qtes**, **Seuil** et **Qtec** pour que les valeurs introduites soient positives (**>0**).

Partie III: *LMD*

III.1 Insert:

Cette requête permet d'insérer les données dans une table.

INSERT INTO ***nom_de_table***
[(colonne1 [, colonne2] …)]
VALUES (valueur [, valueur …] **);**

III.2 Update:

Il est possible grâce à l'ordre **UPDATE** de modifier des valeurs de colonnes dans les tables.

UPDATE *table*
SET *colonne1= valeur [, colonne2 = valeur …]*
[WHERE *condition***];**

III.3 Delete:

Pour effacer une ou plusieurs lignes d'une table, il suffira d'utiliser la commande DELETE dont voici la syntaxe :

DELETE [FROM] *table*
WHERE *condititon***;**

Application:

Insérer les données suivantes dans les tables correspondantes, en se référant aux commandes suivantes :

```
delete from produit;
Insert into Produit values (2,'Ecran',400,15,5);
Insert into Produit values (5,'Clavier',25,40,10);
Insert into Produit values (3,'CD-ROM',150,20,3);
Insert into Produit values (9,'Souris',5,100,20);
Insert into Produit values (10,'Imprimante',500,50,8);

delete from Client;
Insert into Client values (1250,'Mohamed',50,'Tunis');
Insert into Client values (1360,'Ali',400,'Sousse');
Insert into Client values (1580,'Adel',250,'Adel');
Insert into Client values (1210,'Fatma',20,'Sfax');
Insert into Client values (1000,'Slim',120,'Kef');
Insert into Client values (1200,'Sami',50,'Monastir');
Insert into Client values (1400,'Mahmoud',200,'Zagouan');

delete from Commande;


Insert into Commande values (10,1250,0,to_date('14-07-1999','dd-mm-yyyy'));
Insert into Commande values (220,1210,0,to_date('10-11-2000','dd-mm-yyyy'));
Insert into Commande values (40,1200,0,to_date('15-08-2001','dd-mm-yyyy'));
Insert into Commande values (100,1400,0,to_date('20-10-2003','dd-mm-yyyy'));
Insert into Commande values (300,1250,0,to_date('20-11-2001','dd-mm-yyyy'));
Insert into Commande values (50,1400,0,to_date('12-09-2002','dd-mm-yyyy'));

delete from Facture;
Insert into Facture values (10,1250,NULL,to_date('16-07-1999','dd-mm-yyyy'));
Insert into Facture values (220,1210,NULL,to_date('12-11-2000','dd-mm-yyyy'));
Insert into Facture values (40,1200,NULL,to_date('17-08-2001','dd-mm-yyyy'));
Insert into Facture values (100,1400,NULL,to_date('22-10-2003','dd-mm-yyyy'));
Insert into Facture values (300,1250,NULL,to_date('22-11-2001','dd-mm-yyyy'));
Insert into Facture values (50,1400,NULL,to_date('14-09-2002','dd-mm-yyyy'));

delete from PC;
Insert into PC values (2,10,200);
Insert into PC values (5,10,100);
Insert into PC values (9,10,300);
Insert into PC values (2,220,100);
Insert into PC values (9,40,500);
Insert into PC values (10,40,100);
Insert into PC values (3,40,300);
Insert into PC values (10,100,100);
Insert into PC values (5,300,70);
Insert into PC values (10,300,100);
Insert into PC values (3,50,40);
```

- Afficher le contenu de la table PC.
- Tester quelques requêtes de modification de données.
- Trier les noms de clients selon l'ordre décroissant
- Exécuter maintenant la commande suivante à partir de l'invite SQL>

```
SQL>start C:\BD\VenteR.sql;
```

Ceci va permettre de remplir les différentes tables avec certaines données.

- Vérifier le contenu de chaque table avec la commande select.
- Exécuter les deux commandes SQL suivantes :

```
Select to_char(DatC,'yyyy') from Commande;
Select to_char(sysdate,'HH:MI') from dual;
```

- Que remarquez-vous ?