

## TP2 : Data Definition Langage DDL (créer table, modifier table et supprimer table)

La création de la base de données se fait en utilisant la commande :

```
CREATE DATABASE <NomBase>;
```

CREATE DATABASE `commerce2` ; (pour avoir les simples apostrophes Alt Gr+touche 7)  
On peut également directement écrire :

```
CREATE DATABASE commerce2;
```

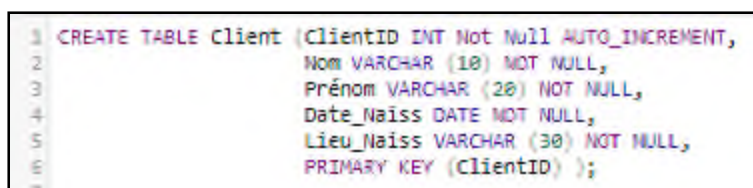
La création de Tables se fait en utilisant la commande de création de table CREATE TABLE qui doit spécifier :

```
CREATE TABLE <NomTable> (Attribut1 Type INT/DATE/YEAR/TEXT/VARCHAR  
(Taille) NOT NULL (si vous n'écrivez pas NOT NULL cet attribut peut avoir la valeur  
NULL) DEFAULT 'value' (la valeur par défaut est mise entre des apostrophes simples  
obtenue par la touche 4) AUTO_INCREMENT (utilisée pour un attribut auto-  
incrémentale), Attribut2 idem,..., Attribut n idem,  
PRIMARY KEY (attribut M, attribut P,..., attribut K),  
FOREIGN KEY (nom de l'attribut clé étrangère) REFERENCES TableRéféréncée (attribut  
clé de la TableRéféréncée) );
```

On peut ajouter la phrase suivante avant le point virgule **ENGINE=InnoDB DEFAULT CHARSET=utf8** pour avoir un moteur de base de données InnoDB pour la table créée et un interclassement utf8 (si vous oubliez cette phrase vous aurez un moteur de base de donnée MyISAM pour votre table créée et vous risquez de ne pas voir les relations entre vos tables dans le concepteur).

1-Table Client (Créée à l'aide de la requête suivante):

```
CREATE TABLE Commerce2.Clients  
(  
ClientID INT NOT NULL,  
Nom VARCHAR(10) NOT NULL,  
Prénom VARCHAR(20) NOT NULL,  
Date_Naiss DATE NOT NULL,  
Lieu_Naiss VARCHAR (30) NOT NULL,  
PRIMARY KEY (ClientID)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 ;
```



```
1 CREATE TABLE Client (ClientID INT Not Null AUTO_INCREMENT,  
2                           Nom VARCHAR (10) NOT NULL,  
3                           Prénom VARCHAR (20) NOT NULL,  
4                           Date_Naiss DATE NOT NULL,  
5                           Lieu_Naiss VARCHAR (30) NOT NULL,  
6                           PRIMARY KEY (ClientID) );
```

**Clients (ClientID, Nom, Prénom, Date\_Naiss, Lieu\_Naiss)**

## 2-Table Fournisseur :

```
CREATE TABLE Commerce2.Fournisseurs
(
FournisseurID INT NOT NULL,
Nom VARCHAR(10) NOT NULL,
Adresse VARCHAR(255) NOT NULL,
Date_Naiss DATE NOT NULL,
Lieu_Naiss VARCHAR(30) NOT NULL,
Telephone INT NOT NULL,
PRIMARY KEY(FournisseurID)
) ENGINE=InnoDB DEFAULT CHARSET= utf8;
```

```
1 CREATE TABLE Fournisseur (FournisseurID INT NOT NULL AUTO_INCREMENT,
2                               NOM VARCHAR (10) NOT NULL,
3                               Adresse VARCHAR (255) NOT NULL,
4                               Date_Naiss DATE NOT NULL,
5                               Lieu_Naiss VARCHAR (30) NOT NULL,
6                               Telephone INT NOT NULL,
7                               PRIMARY KEY (FournisseurID) );|
```

**Fournisseurs (FournisseurID, Nom, Adresse, Date\_Naiss, Lieu\_Naiss, Telephone)**

## 3-Table Articles :

```
CREATE TABLE Commerce2.Articles
(
ArticleID INT NOT NULL,
Nom VARCHAR(20) NOT NULL,
Quantite INT NOT NULL,
Description_du_produit VARCHAR(30) NOT NULL,
FournisseurID_A INT NOT NULL,
PRIMARY KEY(ArticleID),
FOREIGN KEY (FournisseurID_A) REFERENCES Commerce2.fournisseurs (FournisseurID)
) ENGINE=InnoDB DEFAULT CHARSET= utf8;
```

```
CREATE TABLE Articles (ArticleID INT NOT NULL AUTO_INCREMENT,
                          Nom VARCHAR (20) NOT NULL,
                          Quantite INT NOT NULL,
                          Description_du_produit VARCHAR (30) NOT NULL,
                          FournisseurID_A INT NOT NULL,
                          PRIMARY KEY (ArticleID),
                          FOREIGN KEY (FournisseurID_A)REFERENCES Fournisseur (FournisseurID) );
```

**Articles (ArticleID, Nom, Quantite, Description\_du\_produit, #FournisseurID\_A)**

## 4-Table Commandes :

```
CREATE TABLE Commerce2.Commandes
(
NumCommande INT NOT NULL,
```

```

ClientID_C INT NOT NULL,
Type_Livraison VARCHAR(20) NOT NULL,
Date_Livraison DATE NOT NULL,
Etat VARCHAR(30) NOT NULL,
PRIMARY KEY (NumCommande),
FOREIGN KEY (ClientID_C) REFERENCES Commerce2.clients (ClientID)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

1 CREATE TABLE Commandes (NumCommande INT NOT NULL AUTO_INCREMENT,
2                             ClientID_C INT NOT NULL,
3                             Type_Livraison VARCHAR (20) NOT NULL,
4                             Date_Livraison DATE NOT NULL,
5                             Etat VARCHAR (30) NOT NULL,
6                             PRIMARY KEY (NumCommande),
7                             FOREIGN KEY (ClientID_C) REFERENCES Client (ClientID) );
8

```

**Commandes (NumCommande, #ClientID\_C, Type\_Livraison, Date\_Livraison, Etat)**

5-Table Lignes Commandes :

```

CREATE TABLE Commerce2.Ligne_Commandes
(
Cle INT NOT NULL,
NumCommande_L INT NOT NULL,
ArticleID_L INT NOT NULL,
Quantite INT NOT NULL,
prix_vente DECIMAL NOT NULL,
PRIMARY KEY (cle),
FOREIGN KEY (NumCommande_L) REFERENCES Commerce2.commandes (NumCommande),
FOREIGN KEY (ArticleID_L) REFERENCES Commerce2.articles (ArticleID)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

1 CREATE TABLE Lignes_Commandes (Cle INT NOT NULL AUTO_INCREMENT,
2                                 NumCommande_L INT NOT NULL,
3                                 ArticleID_L INT NOT NULL,
4                                 Quantite INT NOT NULL,
5                                 prix_vente DECIMAL,
6                                 PRIMARY KEY (Cle),
7                                 FOREIGN KEY (NumCommande_L) REFERENCES Commandes (NumCommande),
8                                 FOREIGN KEY (ArticleID_L) REFERENCES Articles (ArticleID));
9

```

**Ligne\_Commandes (Cle, #NumCommande\_L, #ArticleID\_L, Quantite, prix\_vente)**

6-Table Entree :

```

CREATE TABLE Commerce2.Entree
(
NumEntree INT NOT NULL,
ArticleID_E INT NOT NULL,
Quantite INT NOT NULL,

```

prix\_entree **DECIMAL NOT NULL**,  
 Date\_E **DATE NOT NULL**,  
 Empolye **VARCHAR(30) NOT NULL**,  
**PRIMARY KEY**(NumEntree),  
**FOREIGN KEY** (ArticleID\_E) **REFERENCES** Commerce2.articles (ArticleID)  
**) ENGINE=InnoDB DEFAULT CHARSET=utf8;**

```

1 CREATE TABLE Entree (NumEntree INT NOT NULL AUTO_INCREMENT,
2                         ArticleID_E INT NOT NULL,
3                         Quantite INT NOT NULL,
4                         prix_entree DECIMAL NOT NULL,
5                         Date_E DATE NOT NULL,
6                         Empolye VARCHAR (30),
7                         PRIMARY KEY (NumEntree),
8                         FOREIGN KEY (ArticleID_E) REFERENCES articles (ArticleID));

```

**Entree (NumEntree, #ArticleID\_E, Quantite, prix\_entree, Date\_E, Empolye)**

7-Table Sortie-Vente:

**CREATE TABLE** Commerce2.Sortie\_Vente  
 (  
 NumVente **INT NOT NULL**,  
 ArticleID\_S **INT NOT NULL**,  
 Quantite **INT NOT NULL**,  
 prix\_vente **DECIMAL NOT NULL**,  
 Date\_V **Date NOT NULL**,  
 Empolye **VARCHAR(30) NOT NULL**,  
**PRIMARY KEY**(NumVente),  
**FOREIGN KEY** (ArticleID\_S) **REFERENCES** Commerce2.articles(ArticleID)  
**) ENGINE=InnoDB DEFAULT CHARSET=utf8;**

```

1 CREATE TABLE Sortie_Vente ( NumVente INT NOT NULL AUTO_INCREMENT,
2                               ArticleID_S INT NOT NULL,
3                               Quantite INT NOT NULL,
4                               prix_vente DECIMAL NOT NULL,
5                               Date_V Date NOT NULL,
6                               Empolye VARCHAR(30) NOT NULL,
7                               PRIMARY KEY (NumVente),
8                               FOREIGN KEY (ArticleID_S) REFERENCES Commerce2.articles(ArticleID) );

```

**Sortie\_Vente (NumVente, #ArticleID\_S, Quantite, prix\_vente, Date\_V, Empolye)**

Modification de la structure d'une table :

Il est possible qu'on veuille modifier la structure d'une table (ajouter, modifier ou supprimer un attribut) après sa création.

#### Ajouter un attribut à une table :

**ALTER TABLE** <NomTable> **ADD** <Attribut Type INT/DATE/YEAR/TEXT/VARCHAR (Taille) NOT NULL (si vous n'écrivez pas NOT NULL cet attribut peut avoir la valeur NULL) **DEFAULT** 'value' (la valeur par défaut est mise entre des apostrophes simples obtenu par la touche 4) **AUTO\_INCREMENT** (utilisé pour un attribut auto-incrémentale) **AFTER** <Attribut-Avant> ;

On prend l'exemple d'ajout de l'attribut « Fiabilité » d'un fournisseur à la table « Fournisseur » qui peut prendre les valeurs {Fiable, moyennement fiable, faiblement fiable, blacklisté}. La valeur par défaut de cet attribut est « Fiable ». Cet attribut est positionné après l'attribut « Lieu\_Naiss» de la table « fournisseur ».

```
ALTER TABLE fournisseur ADD Fiabilité VARCHAR(20) NOT NULL DEFAULT 'Fiable'
AFTER Date_Naiss;
```

On peut également ajouter l'attribut « Lieu\_Résidence » d'un client à la table « Client » qui permet de mentionner où le client réside. Cet attribut est positionné après l'attribut « Prénom» de la table « fournisseur ».

```
ALTER TABLE client ADD Lieu_Résidence VARCHAR (20) NOT NULL AFTER Prénom ;
```

Si vous ne spécifiez pas après quel attribut l'attribut 'Lieu\_Résidence' sera inséré, il sera automatiquement mis comme dernier attribut de la table et vous écrivez seulement :

```
ALTER TABLE client ADD Lieu_Résidence VARCHAR (20) NOT NULL ;
```

#### Modifier l'attribut d'une table :

```
ALTER TABLE client MODIFY Lieu_Résidence VARCHAR (35) NOT NULL;
```

Cette commande modifie la taille de l'attribut « Lieu\_Résidence » dans la table client.

#### Supprimer un attribut dans une table :

```
ALTER TABLE <NomTable> DROP <NomAttribut> ;
```

```
ALTER TABLE client DROP Lieu_Résidence ;
```

Cette commande supprime l'attribut « Lieu\_Résidence » de la table client.

#### Supprimer toute la table :

```
DROP TABLE <NomTable>;
```

```
DROP TABLE client ;
```

Cette commande supprime la table client.

### L'insertion de données dans une table :

L'insertion de données dans une table s'effectue à l'aide de la commande **INSERT INTO**.

#### Exemple 1: insérer un tuple dans la table client

```
INSERT INTO client (ClientID, Nom, Prénom, Date_Naiss, Lieu_Naiss) VALUES (NULL, 'SOULAIMI', 'Fateh', '1995-05-13', 'Alger');
```

Vous remarquez dans la commande d'insertion du tuple, que la valeur attribuée à la colonne **ClientID** est **NULL** car ce champ est **auto-incrémentale**. Pour les attributs qui peuvent être NULL et dans le cas où ces attributs n'ont pas de valeur dans le tuple, la valeur attribuée à ces attributs est mise à **NULL**.

#### Exemple 2: insérer un tuple à la table fournisseur

```
INSERT INTO fournisseur (FournisseurID, NOM, Adresse, Date_Naiss, Fiabilité, Lieu_Naiss, Telephone) VALUES (NULL, 'OUADHI', 'Alger', '1995-12-15', 'Fiable', 'Alger', '0550505050');
```

Vous remarquez dans la commande d'insertion du tuple, que la valeur attribuée à la colonne **FournisseurID** est **NULL** car ce champ est **auto-incrémentale**. Le champ **Fiabilité** de la table fournisseur (qui se trouve avant l'attribut **Lieu\_Naiss** et après l'attribut **Date\_Naiss**) a la valeur par défaut **'Fiable'**. Pour éviter de remplir cet attribut dans la requête et de lui donner sa valeur par défaut automatiquement, il faut enlever l'attribut **Fiabilité** de la commande d'insertion. Ainsi après exécution de la commande d'insertion le champ **Fiabilité** de ce tuple aura la valeur par défaut **'Fiable'** :

```
INSERT INTO fournisseur (FournisseurID, NOM, Adresse, Date_Naiss, Lieu_Naiss, Telephone) VALUES (NULL, 'TEBANE', 'Alger', '1999-06-15', 'Alger', '0550505050');
```

Il est possible d'ajouter plusieurs lignes à un tableau avec une seule requête. Pour ce faire, il convient d'utiliser la syntaxe suivante :

#### Exemple 1: insérer plusieurs tuples à la fois dans table client

```
INSERT INTO client (ClientID, Nom, Prénom, Date_Naiss, Lieu_Naiss)
VALUES
(NULL, 'AMALOU', 'ILYES', '1998-02-01', 'Tizi Ouzou'),
(NULL, 'BOUDERKA', 'MOULOUD', '1990-05-28', 'Alger'),
(NULL, 'ABDELAZIZ', 'AYMEN', '1992-04-16', 'Média'),
(NULL, 'BOUHABEL', 'Moncef', '1995-12-29', 'Alger'),
(NULL, 'BEY', 'Sofiane', '1995-03-20', 'Alger'),
(NULL, 'SALMI', 'RAFIK', '1999-03-01', 'Tizi Ouzou'),
(NULL, 'SLIMANI', 'MEHDI', '1998-05-01', 'Tizi Ouzou'),
(NULL, 'MAHAMMEDI', 'MAHAMMED', '1995-11-01', 'Bejaia');
```

### Exemple 2: insérer plusieurs tuples à la fois dans table articles

```
INSERT INTO articles (ArticleID, Nom, Quantité, Description_du_produit, FournisseurID_A)
VALUES
(NULL, 'Tondeuse', '15', 'Tondeuse Barbe et Cheveux', '1'),
(NULL, 'Aspirateur', '123', 'Appareil destiné au nettoyage des sols', '3'),
(NULL, 'Tapie de sport', '95', 'Tapis roulant pour faire de la gym', '5'),
(NULL, 'Mixeur', '151', 'Robot électrique servant à mélanger, à broyer des aliments', '2');
```

Vous remarquez que l'attribut Quantité et l'attribut FournisseurID\_A sont de type INT et pourtant les valeurs qui leurs sont attribuées sont mises entre des apostrophes simples (obtenus par la touche 4 du clavier).

Dans le cas où le fournisseur qui a le FournisseurID=1 ou 3 ou 5 ou 2 n'existe pas dans la table fournisseur, la commande d'insertion va générer une erreur car les deux tables fournisseur et article sont liées.

### Pour EXPORTER la base de données commerce2 :

- Cliquez sur le nom de la base de données à exporter. La page des informations relatives à la base de données sélectionnée s'affichent.
- Cliquez sur l'onglet « Exporter » pour afficher les options d'exportation de la base de données.
- À partir de la section «Méthode d'exportation», sélectionnez l'option «Exportation Rapide » si ce n'est déjà fait.
- Dans la section «Format », sélectionnez le format « SQL ».
- Cliquez sur le bouton « Exécuter ».
- Votre navigateur vous invitera à enregistrer le fichier sur votre propre disque. Sélectionnez l'emplacement que voulez utiliser pour sauvegarder votre base.

### Pour IMPORTER la base de données commerce2 (Restaurer les tables de la base de données Commerce2):

- Créez une base de données vide, cliquez sur le nom de cette base de données vide. La page des informations relatives à la base de données sélectionnée s'affichent.
- Cliquez sur l'onglet «Importer» pour afficher les options d'importation des données.
- Cliquez sur le bouton «Parcourir les fichiers» de la section « Fichier à importer » (File to import/ choose File). Votre navigateur vous invitera à repérer sur votre ordinateur le fichier de la base de données « commerce2.sql ».
- Une fois le fichier repéré et sélectionné, cliquez sur le bouton « Exécuter » pour avoir toute la structure de la BD créer auparavant (toute la structure de la BD commerce2.sql sera mise dans la base de données vide qui à été créée pour importer la base de données commerce2.sql).