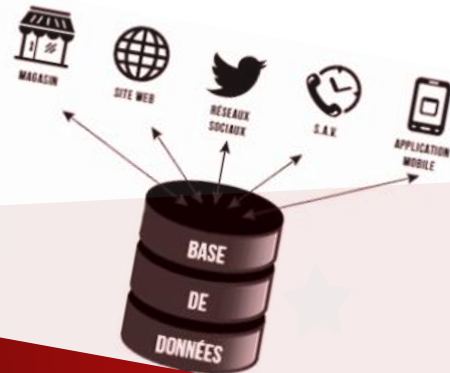


MANIPULER DES BASES DES DONNÉES



MR OUALID FOURI

PARTIE 3 : MANIPULATION DES DONNÉES

CHAPITRE 1 : CRÉER UNE BASE DE DONNÉES

RAPPEL: DÉFINITION D'UNE BASE DE DONNÉES

Une base de données est une structure permettant de stocker un grand nombre d'informations afin d'en faciliter l'utilisation.

RAPPEL: DÉFINITION D'UNE BASE DE DONNÉES

Le fait de structurer les données a pour but d'assurer les fonctions fondamentales suivantes :

- **La fiabilité du stockage de l'information : restituer l'information stockée dans la BD;**
- **La massification : Le traitement de grands volumes de données ;**
- **L'optimisation : En terme de temps de traitement et espace de stockage ;**
- **La sécurisation des accès aux données ;**
- **La qualité des données : Vérification des règles de gestion.**
- **Le partage des données entre plusieurs acteurs (utilisateurs, applications...) et la gestion des concurrences d'accès.**

LES OBJETS D'UNE BASE DE DONNÉES:

- **Les tables : contenant des données;**
- **Les index : servant à retrouver, trier, regrouper rapidement les données ;**
- **Les déclencheurs (triggers) : permettant d'exécuter des opérations particulières lors de l'insertion, la modification ou la suppression de données ;**
- **Les types de données définis par l'utilisateur : servant de référentiel à plusieurs tables ;**
- **Les valeurs par défaut (Defaults).**

LES OBJETS D'UNE BASE DE DONNÉES:

- **Les vues, ou pseudo-tables (Views) : offrant une vue particulière des données aux utilisateurs ;**
- **Les fonctions définies par l'utilisateur : permettant de renvoyer soit une valeur, soit une table ;**
- **Les procédures stockées : exécutées par l'utilisateur pour produire un résultat donné ;**
- **Les diagrammes (Diagrams) : qui visualisent les relations entre les tables.**

SGBD (SYSTÈME DE GESTION DE BASES DE DONNÉES)

- **Un SGBD (Système de Gestion de Bases de Données) est un ensemble logiciel qui permet la structuration, le stockage, et la manipulation d'une base de données.**

SGBD (SYSTÈME DE GESTION DE BASES DE DONNÉES)

En termes d'architecture, et selon le modèle ANSI/SPARC d'un SGBD comporte 3 niveaux :

- ♦ **Niveau interne ou physique :**

Spécification du stockage physique des données (fichiers, disques, etc.) et des méthodes d'accès (index, etc.).

SGBD (SYSTÈME DE GESTION DE BASES DE DONNÉES)

- ◆ **Niveau conceptuel :**

Le niveau central d'un SGBD. Il correspond à une vue générale de toutes les données existant dans l'entreprise.

SGBD (SYSTÈME DE GESTION DE BASES DE DONNÉES)

◆ Niveau externe :

Il s'agit d'une vue externe pour chaque groupe d'utilisateurs sur un sous ensemble de la base ;

Chaque schéma externe est généralement un sous schéma du schéma conceptuel mais peut contenir parfois des informations supplémentaires.

EXEMPLES DE SGBD :

Oracle est un SGBD relationnel et relationnel-objet très utilisé pour les applications professionnelles.

- **PostgreSQL est un SGBD relationnel puissant.**
- **Access est un SGBD relationnel Microsoft.**
- **MongoDb est un SGBD non-relationnel libre (licence Apache) orienté document.**
- **MySQL est un système de gestion de bases de données relationnelles (SGBDR) open source.**

LE LANGAGE DE REQUÊTES SQL

Il s'agit d'un langage d'interrogation des bases de données relationnelles qui permet d'effectuer les tâches suivantes :

- Définition et modification de la structure de la base de données**
- Interrogation et modification non procédurale (c'est à dire interactive) de la base de données**
- Contrôle de sécurité et d'intégrité de la base**
- Sauvegarde et restauration des bases**

LE LANGAGE DE REQUÊTES SQL

Le langage SQL n'est pas sensible à la casse

Le langage de commandes SQL peut être réparti en quatre catégories :

- **DDL (Langage de définition des données) : langage de manipulation des structures de la base.**
- **DML (Langage de manipulation des données) : il permet de consulter et de modifier le contenu de la base de données.**
- **QL (Langage des queries des données) : langage de requêtes sur les données.**
- **CL (Langage de contrôle des données) : il permet de gérer les privilèges et les différents droits d'accès à la base de données.**

CRÉATION D'UNE BASE DE DONNÉES

Pour créer une nouvelle base de données sur MySQL, nous utilisons la commande CREATE DATABASE avec la syntaxe suivante :

```
CREATE DATABASE [IF NOT EXISTS] database_name  
[CHARACTER SET charset_name]  
[COLLATE collation_name]
```

SHOW CHARACTER
SET;

CRÉATION D'UNE BASE DE DONNÉES

On peut utiliser la commande `SHOW CREATE DATABASE` pour examiner la base de données créée :

```
SHOW CREATE DATABASE testdb;
```


CRÉATION D'UNE BASE DE DONNÉES

Enfin, sélectionnez la base de données nouvellement créée avec laquelle on veut travailler en utilisant la commande USE :

```
USE testdb;
```

CRÉATION D'UNE BASE DE DONNÉES

Supprimer une base de données :

DROP DATABASE NOMBD ;

MOTEURS DE STOCKAGE

- **Le moteur de stockage d'un Système de Gestion de Base de Données (SGBD), aussi appelé moteur de table, est l'ensemble d'algorithmes qui permettent de stocker et d'accéder aux données.**
- **les SGBD utilisent un seul moteur de stockage qui est optimisé au mieux pour la lecture, l'écriture et la suppression de données.**
- **MySQL se distingue des autres SGBD par le fait de donner à l'utilisateur le libre choix d'utiliser un moteur de table parmi plusieurs moteurs différents.**

MOTEURS DE STOCKAGE

Afin de consulter la liste des engins mis à notre disposition par MySQL, on peut exécuter la commande :

- **SHOW ENGINES**

MOTEURS DE STOCKAGE

MyISAM : un moteur non transactionnel assez rapide en écriture et très rapide en lecture.	<ul style="list-style-type: none">*Très rapide en lecture*Extrêmement rapide pour les opérations COUNT() sur une table entière*Les index FULLTEXT pour la recherche sur des textes	<ul style="list-style-type: none">*Pas de gestion des relations*Pas de gestion des transactions*Bloquent une table entière lors d'opérations d'insertions, suppressions ou mise à jour des données
InnoDB : un moteur relationnel performant faisant partie de la famille des moteurs transactionnels.	<ul style="list-style-type: none">*Gestion des relations*Gestion des transactions*Verrouillage à la ligne et non à la table	<ul style="list-style-type: none">*Plus lent que MyISAM*Occupe plus de place sur le disque dur*Occupe plus de place en mémoire vive
Memory : un moteur de stockage permettant de créer des tables directement dans la mémoire vive	<ul style="list-style-type: none">*Le moteur le plus rapide*Ne consomme pas de place sur le disque dur	<ul style="list-style-type: none">*Les données sont volatiles : un arrêt du serveur et elles disparaissent*Pas de champs BLOB ou TEXT

TP1

1. Connectez-vous au serveur MySQL avec votre compte root créé lors de l'installation.
2. Affichez les bases de données actuellement disponibles sur le serveur.
3. Créez une base de données avec le nom : bdtest
4. Affichez les bases de données pour vérifier l'existence de la base bdtest.
5. afficher les informastion sur la base de données créer
6. Sélectionnez la base de données nouvellement créée.
7. Supprimez la base de données dbtest
8. sortir du programme.
9. Effectuez les mêmes étapes sur Workbench.

CRÉATION DES TABLES

La création des tables est possible à partir de la ligne de commande.

- Syntaxe : Pour créer une table « nom_table », il suffit d'utiliser la requête suivante:

```
CREATE TABLE nom_table
```

CRÉATION DES TABLES

```
CREATE TABLE [IF NOT EXISTS] table_name(  
    column_1_definition,  
    column_2_definition,  
    ...,  
    table_constraints  
) ENGINE=storage_engine;
```


CRÉATION DES TABLES

La syntaxe de définition d'une colonne est la suivante :

```
nom_colonne type_donnee Liste_contraintes;
```

CRÉATION DES TABLES

Typage des colonnes :

Type	Taille	Utilisation
TINYINT	1 octet	petites valeurs entières/ Boolean
SMALLINT	2 octets	valeur entière
MEDIUMINT	3 octets	valeur entière
INT ou INTEGER	4 octets	valeur entière
BIGINT	8 octets	Valeur maximale entier
FLOAT	4 octets	Simple précision valeurs à virgule flottante
DOUBLE	8 octets	Double-précision valeurs à virgule flottante
DECIMAL	5-17 octets	valeur décimale

Type	Taille (octets)	Utilisation
CHAR	0-255	chaîne longueur fixe
VARCHAR	0-65535	chaînes de longueur variable
TINYBLOB	0-255	Pas plus de 255 caractères dans une chaîne binaire
TINYTEXT	0-255	Courtes chaînes de texte
BLOB	0-65535	données textuelles longues sous forme binaire
TEXTE	0-65535	Longue données de texte
MEDIUMBLOB	0-16777215	forme binaire de longueur moyenne des données de texte
MEDIUMTEXT	0-16777215	longueur moyenne des données de texte
LOBLOB	0-4294967295	Grands données de texte sous forme binaire
LOBTEXT	0-4294967295	Grande données de texte

Type	Taille (Byte)	Format	Utilisation
DATE	3	AAAA-MM-JJ	Les valeurs de date
TIME	3	HH: MM: SS	Valeur temps ou la durée
ANNÉE	1	AAAA	Année Valeur
DATETIME	8	AAAA-MM-JJ HH: MM: SS	Mixage valeurs date et heure
TIMESTAMP	4	AAAAMMJJ HHMMSS	Date de mélange et la valeur temps, un horodatage

CRÉATION DES TABLES

Contraintes d'intégrité :

- Une contrainte d'intégrité est une règle qui définit la cohérence d'une donnée ou d'un ensemble de données de la BD.
- Les contraintes peuvent avoir une portée sur une colonne ou sur une table lorsque la contrainte porte sur plusieurs colonnes.
- Il existe trois types de contraintes d'intégrité

CRÉATION DES TABLES

Intégrité de domaine : (NOT NULL, DEFAULT, UNIQUE...)

- Spécifie un ensemble de valeurs pour une colonne
- Détermine si les valeurs nulles sont autorisées
- Implémentation par contrôle de validité

Intégrité des entités : (PRIMARY KEY)

- Précise la clé primaire de chaque table

Intégrité référentielle : (FOREIGN KEY/REFERENCES)

- Assure l'intégrité des relations entre les clés primaires et les clés étrangères.

CRÉATION DES TABLES

1 - PRIMARY KEY

- ◆ Une clé primaire est une colonne ou un ensemble de colonnes qui identifie de manière unique chacune des ligne de la table.
- ◆ Une clé primaire doit contenir des valeurs uniques.
- ◆ Une colonne de clé primaire ne peut pas avoir de valeurs NULL.

```
CREATE TABLE nom_table(  
    primary_key_colonne Type_donnee PRIMARY KEY,  
    ...  
);
```

CRÉATION DES TABLES

Si la clé primaire est composée de plusieurs colonnes, PRIMARY KEY (contrainte de table) est utilisée avec la syntaxe suivante :

```
CREATE TABLE nom_table(  
    primary_key_colonne1 type_donnee,  
    primary_key_colonne type_donnee,  
    ... ,  
    PRIMARY KEY(liste_colonnes)  
);
```


CRÉATION DES TABLES

2 - FOREIGN KEY

- ◆ Une clé étrangère est une colonne ou un groupe de colonnes d'une table qui renvoie à une colonne ou à un groupe de colonnes d'une autre table.
- ◆ La clé étrangère impose des contraintes sur les données des tables associées, ce qui permet à MySQL de maintenir l'intégrité référentielle.

```
CONSTRAINT nom_contrainte  
FOREIGN KEY (nom_colonne)  
REFERENCES table_parent(nom_colonne_p)  
[ON DELETE reference_option]  
[ON UPDATE reference_option]
```

CRÉATION DES TABLES

3 - NOT NULL

- ◆ La contrainte NOT NULL est une contrainte de colonne qui garantit que les valeurs stockées dans une colonne ne sont pas NULL.

```
nom_colonne type_donnee NOT NULL;
```

CRÉATION DES TABLES

4 - DEFAULT

- ◆ La contrainte MySQL DEFAULT permet de spécifier une valeur par défaut pour une colonne :

```
nom_colonne    type_donnee  DEFAULT valeur_par_defaut;
```

CRÉATION DES TABLES

5- UNIQUE

- ◆ UNIQUE est une contrainte d'intégrité qui garantit que les valeurs d'une colonne ou d'un groupe de colonnes sont uniques.
- ◆ elle peut être une contrainte de colonne ou une contrainte de table.

nom_c

```
CREATE TABLE nom_table(  
    nom_colonne1 definition_colonne,  
    nom_colonne2 definition_colonne,  
    ...,  
    UNIQUE(nom_colonne1, nom_colonne2,...)  
);
```

CRÉATION DES TABLES

6 - CHECK

- ◆ La contrainte CHECK assure que les valeurs stockées dans une colonne ou un groupe de colonnes satisfont à une expression booléenne. Elle est ainsi considérée comme une contrainte de colonne et une contrainte de table.

```
CREATE TABLE Produits (  
    Num_Produit VARCHAR(18) PRIMARY KEY,  
    description VARCHAR(40),  
    cout DECIMAL(10,2) NOT NULL CHECK (cout >= 0),  
    prix DECIMAL(10,2) NOT NULL CHECK (prix >= 0),  
    CONSTRAINT produits_chk_prix_et_cout  
        CHECK (price = cost)  
);
```

SUPPRIMER UNE TABLE

Commande *DROP TABLE* :

La commande DROP TABLE supprime définitivement une table et ses données de la base de données.

```
DROP [TEMPORARY] TABLE [IF EXISTS] nom_table1, nom_table2  
...
```

MODIFIER UNE TABLE

Commande ALTER TABLE :

ALTER TABLE est la commande utilisée pour changer la structure d'une table :

- Ajouter, modifier, renommer et supprimer une colonne
- Renommer une table
- Ajouter et supprimer une contrainte d'intégrité.

MODIFIER UNE TABLE

Commande ALTER TABLE :

Ajouter une ou plusieurs colonnes à une table :

```
ALTER TABLE Produits (  
  
    ADD num_Produit VARCHAR(18)  
  
);
```


MODIFIER UNE TABLE

Commande ALTER TABLE : Modifier une ou plusieurs colonnes d'une table :

Cette modification peut porter sur le type ainsi que les contraintes associées à cette colonne en utilisant la syntaxe suivante :

```
ALTER TABLE Produits (  
    MODIFY num_Produit VARCHAR(20) PRIMARY KEY );  
...;
```

MODIFIER UNE TABLE

Commande ALTER TABLE : Renommer une ou plusieurs colonnes d'une table

Afin de renommer une colonne d'une table, utilise la syntaxe suivante :

```
ALTER TABLE Produits (  
    CHANGE COLUMN cout cout_produit DECIMAL(10,2 ) CHECK (cout >= 0)  
);
```

MODIFIER UNE TABLE

Commande ALTER TABLE : Renommer une table

Pour renommer une table, on utilise la syntaxe suivante :

```
ALTER TABLE Produits  
RENAME TO Articles ;
```

MODIFIER UNE TABLE

Commande ALTER TABLE : Ajouter et supprimer une contrainte de table

PRIMARY KEY :

```
ALTER TABLE nom_table  
  
DROP PRIMARY KEY;
```

MODIFIER UNE TABLE

Commande ALTER TABLE : Ajouter et supprimer une contrainte de table

FOREIGN KEY :

```
ALTER TABLE nom_table
```

```
ALTER TABLE nom_table
```

```
DROP FOREIGN KEY nom_contrainte;
```

```
ALTER TABLE nom_table (column_name, ...)
```

MODIFIER UNE TABLE

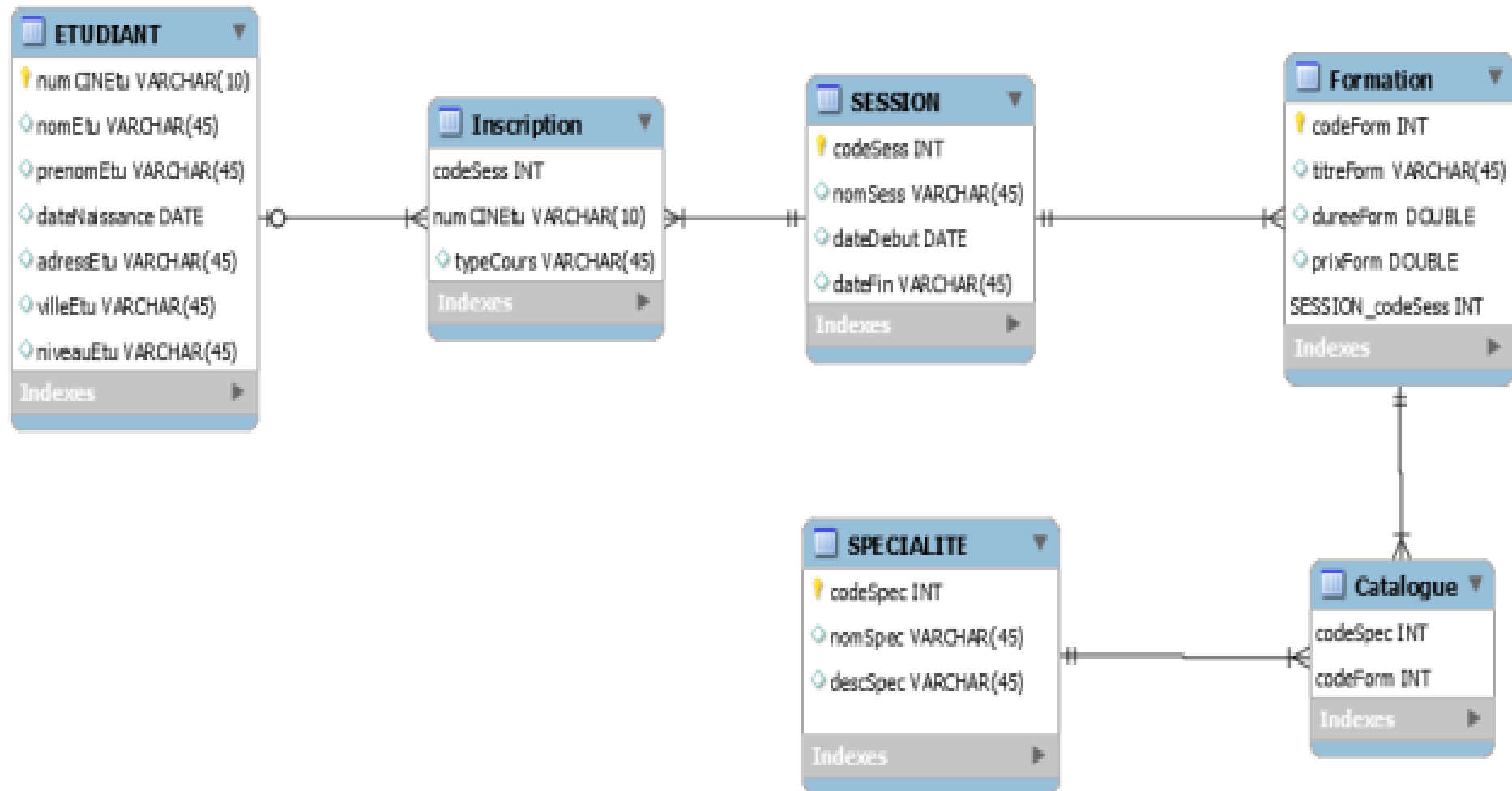
Commande ALTER TABLE : Ajouter et supprimer une contrainte de table

UNIQUE :

```
ALTER TABLE nom_table  
DROP INDEX nom_contrainte;
```

TP2

1. Créer une base de données : « Centre formation »
2. Créer les tables depuis le MLD : « Centre de Formation » (Ne pas oublier les clés primaires et étrangères) voir la figure.



TP2

3. Ajouter une contrainte NOT NULL sur la colonne typeCours de la table Inscription
4. Ajouter une contrainte CHECK dans la table SESSION : dateFin doit être toujours supérieure à dateDebut.
5. Ajouter une colonne « Active » sur la table SPECIALITE.

Remarque : Cette colonne est un flag qui prend la valeur 1 si la spécialité est active, et 0 si elle ne l'est pas.