TP3 : créer une table, modifier la structure d'une table

Objectif :

- Rappel : création d'une table
- Présentation de l'attribut **AUTO_INCREMENT**
- **Modifier la structure d'une table** : la renommer, ajouter un champ à une position précise dans la table

1 compteur : auto_increment

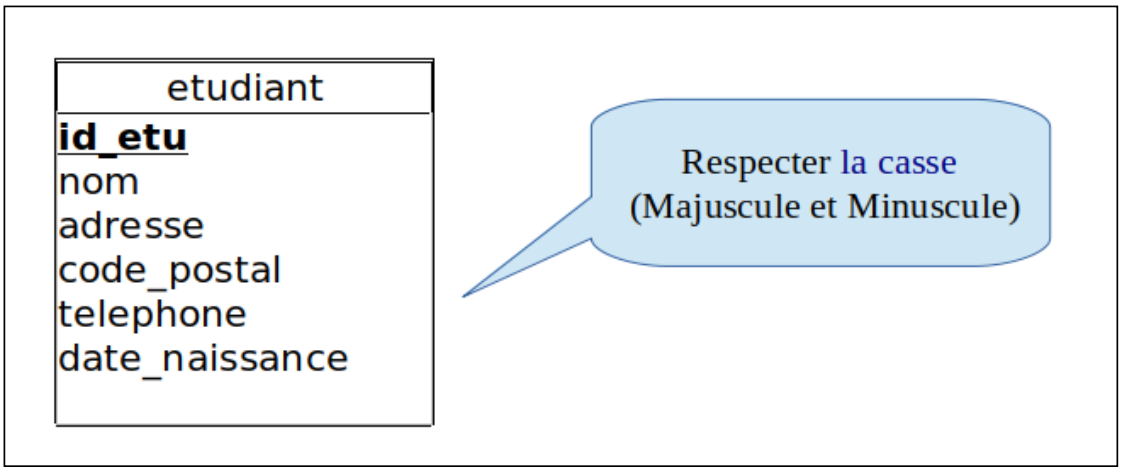
type "Numérique" : entier

- de type INT avec auto-incrémentation à chaque nouvelle entité

Access	MySQL	Oracle	PostgreSQL	SQL server
AUTOINCREMENT	int AUTO_INCREMENT	number(10) + trigger	serial	identity(seed,increment)

documentation mariadb

2 Création de la table



structure de la table (MLD)

2.1 Créer une table etudiant avec comme champs :

- un champ **id_etu** de type entier non signé avec un attribut AUTO_INCREMENT pour générer un identifiant unique. Voir l'exemple avec le lien ci dessous : http://dev.mysql.com/doc/refman/5.7/en/example-auto-increment.html
- un champ **nom** de type **chaîne de caractères variable** de taille 20 caractères maximum.
- un champ **adresse** de type **TEXT**.
- un champ **code_postal** de type **entier** ou de type **VARCHAR(5)**. Le code postal doit toujours être affiché sur 5 caractères (digits) au maximum.
- un champ **telephone** de type **Numérique(10)** ou de type **VARCHAR(10)**. Le numéro de téléphone doit toujours être affiché dans sa totalité (sur 10 digits), c'est à dire sur 10 caractères y compris le premier 0.
- un champ **date_naissance** de type **DATE**.

Remarque : **Digit** : Symbole utilisé pour représenter un des entiers non négatifs plus petit que la base, dans un système de numération. Dans notre cas en base 10 de 0 à 9

2.2 Insérer 4 ou 5 enregistrements

Remarque : dans la documentation de MySql, il est écrit qu'il ne faut pas assigner de valeur à un champ qui a comme attribut **AUTO_INCREMENT**, MySQL assigne automatiquement une séquence de nombres. Il est possible d'assigner une valeur NULL qui est transformée automatiquement en un nombre de la séquence gérée par MySQL. Dans d'autre SGBDR comme sql server, ça ne fonctionne pas exactement de la même manière.

Voici 3 exemples :

```
INSERT INTO etudiant VALUES
    (NULL, 'dupond' , 'rue de chateau', '01000', '0123456789', '1990-12-31');
INSERT INTO etudiant (nom,adresse,telephone,  date_naissance ) VALUES
    ( 'durand' , 'rue de pont', '0123456789', '1990-12-31');
INSERT INTO etudiant  VALUES
    (NULL, 'DAVAL' , 'rue du CHATEAU',90000, 0123456789, '1990-12-31');
```

- afficher tous les enregistrements
- afficher la structure de la table
- afficher la commande pour créer la table

3 opération sur la structure d'une table

3.1 Renommer une table :

Ajouter un « s » au nom de la table

```
RENAME TABLE etudiant TO etudiants;
SHOW TABLES;
```

3.2 Ajouter un champ en fin liste dans la table

Ajouter un champde nom **groupe** et de type « chaîne de 4 caractères » (le champ est par défaut en dernière position). Afficher la structure de la table. Afficher le contenu de la table (les enregistrements).

```
ALTER TABLE  etudiants ADD groupe VARCHAR(20);
DESCRIBE etudiants;
SELECT * FROM etudiants;
```

3.3 Ajouter un champ à une position précise dans la table

Ajouter un champ en troisième position de nom **prenom** et de type « chaîne de 10 caractères ». Afficher la structure de la table. Afficher le contenu de la table (les enregistrements).

```
ALTER TABLE  etudiants ADD prenom VARCHAR(20) AFTER nom;
DESCRIBE etudiants;
SELECT * FROM etudiants;
```

3.4 Supprimer un champ dans la table

Supprimer le champ **groupe**. Afficher la structure de la table. Afficher le contenu de la table (les enregistrements).

```
ALTER TABLE  etudiants  DROP COLUMN groupe;
DESCRIBE etudiants;
SELECT * FROM etudiants;
```

3.5 Modifier un champ dans la table

Modifier le champ **prenom**, renommer le **prenoms** avec un nouveau type . Afficher la structure de la table. Afficher le contenu de la table (les enregistrements).

```
ALTER TABLE  etudiants CHANGE prenom  prenoms VARCHAR(50);
DESCRIBE etudiants;
SELECT * FROM etudiants;
```

exercice : sur le même principe, changer le type numéro de téléphone par une chaîne de caractères fixe sur 13 caractères

conclusion : (la solution n'est pas évidente pour les numéros de téléphone)

Pour les étudiants les plus rapides, essayer d'ajouter une contrainte pour que le numéro de téléphone ne soit composé que de caractères de "0" à "9", exemple

3.6 Supprimer une table

Supprimer la table **etudiant** et la table **etudiants** si elles existent

```
DROP TABLE IF EXISTS etudiant;
DROP TABLE IF EXISTS etudiants;
```

3.7 premier essai avec looping

utiliser looping et créer la table "etudiant"