

1 Introduction : présentation des TPs BDD (objectifs)

- **Partie 1 (8 premières semaines) :**
 - SQL : Être capable de réaliser les structures de données conçues lors des TDs de BDD sur un logiciel => **Créer des tables** .
 - SQL : **Ajouter, modifier, supprimer des données dans ces tables** .
 - SQL : **Afficher (calculs, agrégation...)** des données issues de ces tables .
 - Utiliser “looping” pour **modéliser** (réaliser un MCD), présentation rapide
 - Introduction aux jointures
- **Partie 2 (7 dernières semaines) :**
 - Réaliser des petits scripts en langage **Python** principalement pour afficher, modifier, supprimer ou ajouter des données dans une table.
 - Examen SQL devant machine avec des jointures ?
 - Mini projet : **CRUD sur une table avec une jointure et visualisation de données (fonctions d'agrégation)**

2 Définitions importantes : B.D.D. et S.G.B.D.

BD : (ou BDD) **base de Données** : stock d'informations organisé et structuré de manière à :

- pouvoir être facilement manipulé
- stocker efficacement de très grandes quantités d'informations.
- Conserver l'intégrité des données

SGBD : Les SGBD (**Système de Gestion de Base de Données**) permettent de naviguer dans ces données et d'extraire (ou de mettre à jour) les informations voulues au moyen de *requêtes*. C'est le logiciel qui gère les données.

Remarque : **SGBDR** : le R vient de *relationnel*, notion difficile à comprendre en début de formation, mais l'idée est :

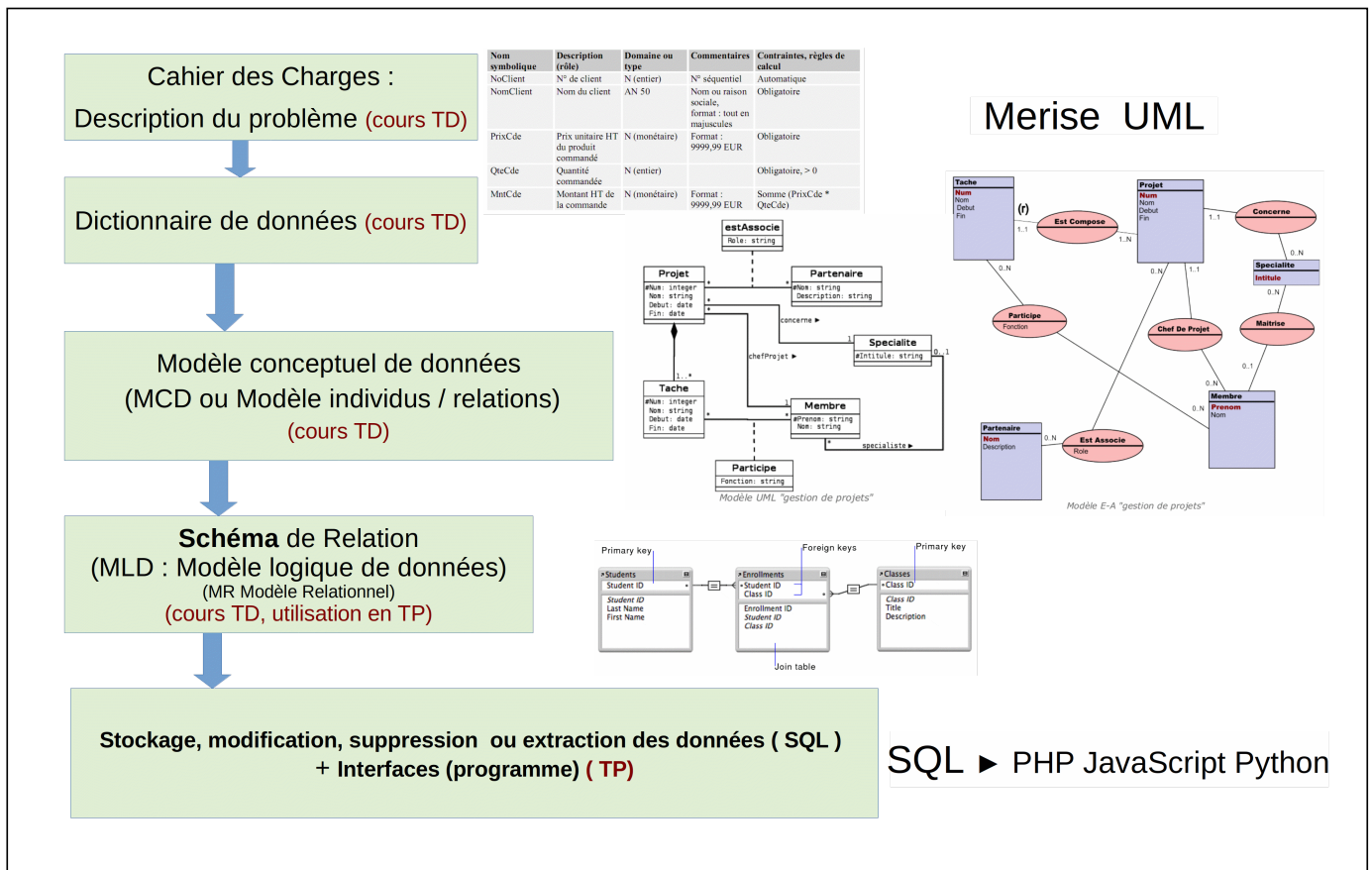
- **minimiser la redondance, et maximiser la cohérence**
- la simplicité d'usage,
- les liaisons entre les données sont facilitées afin d'améliorer les temps d'accès.

Un classement des différents SGBD

DBMS systems in ranking, August 2020

Rank			DBMS	Database Model	Score		
Aug 2020	Jul 2020	Aug 2019			Aug 2020	Jul 2020	Aug 2019
1.	1.	1.	Oracle	Relational, Multi-model	1355.16	+14.90	+15.68
2.	2.	2.	MySQL	Relational, Multi-model	1261.57	-6.93	+7.89
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1075.87	+16.15	-17.30
4.	4.	4.	PostgreSQL	Relational, Multi-model	536.77	+9.76	+55.43
5.	5.	5.	MongoDB	Document, Multi-model	443.56	+0.08	+38.99
6.	6.	6.	IBM Db2	Relational, Multi-model	162.45	-0.72	-10.50
7.	8.	8.	Redis	Key-value, Multi-model	152.87	+2.83	+8.79
8.	7.	7.	Elasticsearch	Search engine, Multi-model	152.32	+0.73	+3.23
9.	9.	11.	SQLite	Relational	126.82	-0.64	+4.10
10.	11.	9.	Microsoft Access	Relational	119.86	+3.32	-15.47
11.	10.	10.	Cassandra	Wide column	119.84	-1.25	-5.37
12.	12.	13.	MariaDB	Relational, Multi-model	90.92	-0.21	+5.96

2.1 Étapes avant de commencer à travailler sur un SGBD comme MySql



exemple de démarche pour étudier et réaliser une base de données

Exemple de cours

3 Modèle relationnel dans un base de données : Création des tables

Une table: c'est un objet abstrait, qui regroupe des données qui ont un rapport très précis entre elles.

Terminologie dans les Bases De Données

Une **table** est composée d'un **nom**, et d'un ensemble de propriétés (que l'on appelle **champs**) dont l'une d'elle (éventuellement plusieurs) a le caractère d'**identifiant** : la **clé primaire**.

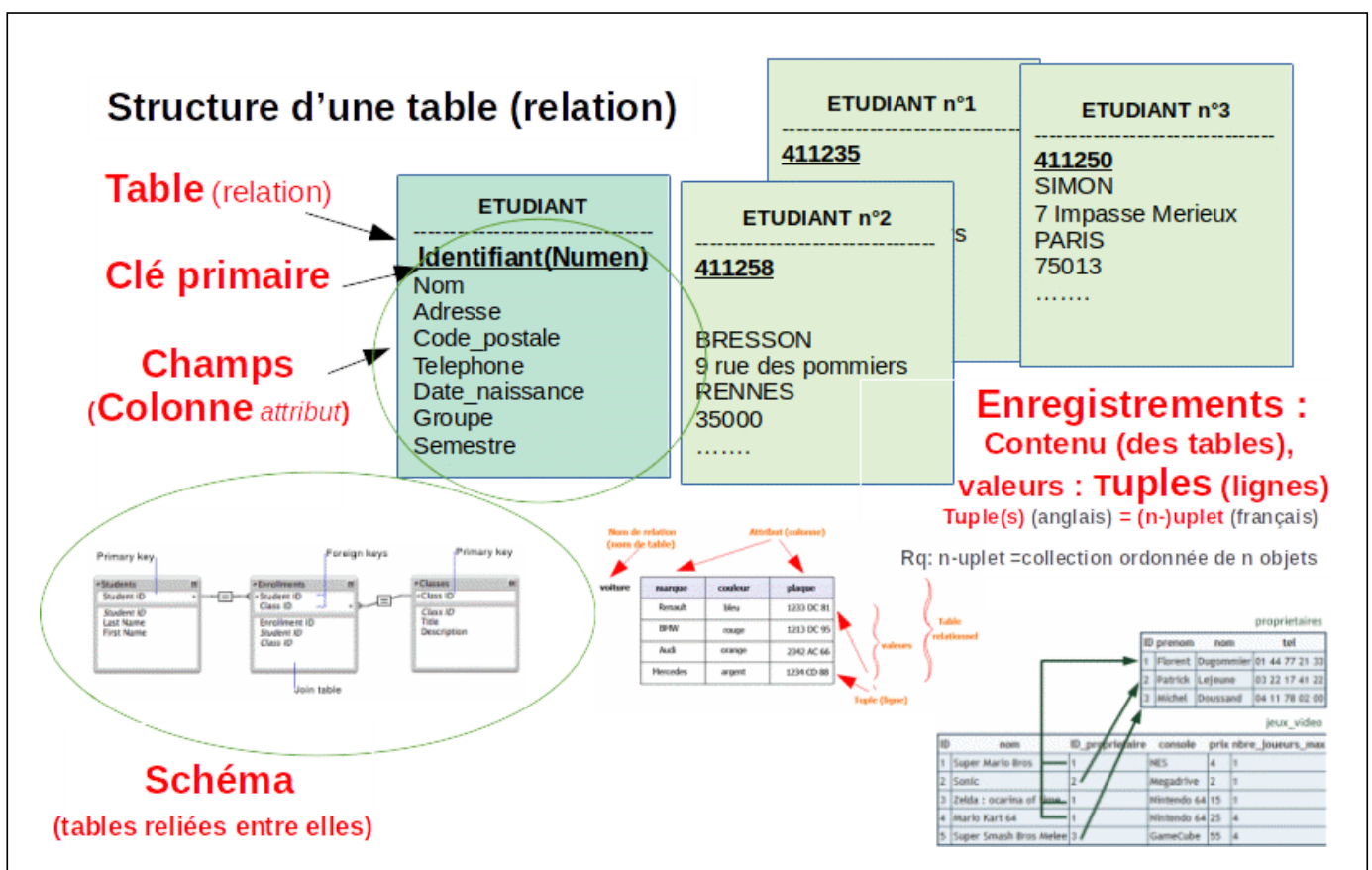
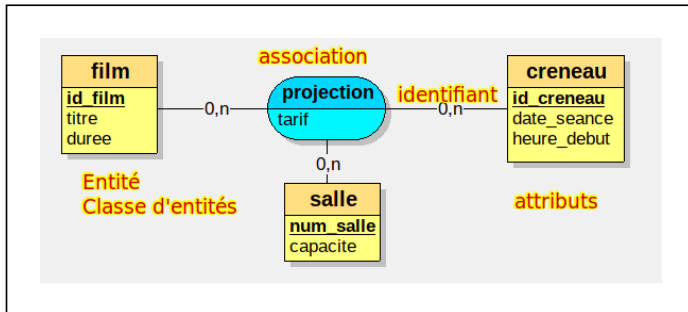


Table , Champs, Clé primaire, Enregistrements

Note : Il est fortement conseillé d'écrire le **nom des champs** et de **la table sans espace, sans caractère spécifique** (lettres accentuées, espaces, virgules, ...). Le code SQL est ainsi plus facilement portable d'une base de données à une autre. (faire comme pour les noms de variables en programmation Java ...).

article : [différence entre une colonne et un champ ? entre une ligne et un enregistrement ?](#)



cours [Modèle Entité-Association](#)

Dans une table, le premier champ (celui qui est souligné) est la **clé primaire** de la table :

- elle permet d'identifier de façon unique chaque enregistrement
- les valeurs prises par la clé primaire sont uniques
- elle est composée d'un ou plusieurs champs

4 introduction et présentation du SQL

4.1 Le SQL :

Définition :

SQL (Structured Query Language) est un langage informatique normalisé qui permet de réaliser des opérations sur des bases de données. La partie langage de manipulation de données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données.

Voici quelques ordres SQL et aspects du langage , mais tous ces ordres ne seront pas abordés ce semestre :

- Définition des données (**LDD** (data definition language)) : **CREATE – ALTER – DROP - RENAME – TRUNCATE**
- Manipulation des données (**LMD**) : **INSERT – UPDATE – DELETE - LOCK TABLE**
- Interrogation des données (**LID**) : **SELECT**
- Contrôle des données (**LCD**) : **GRANT – REVOKE – COMMIT – ROLLBACK – SAVEPOINT - SET TRANSACTION**

5 installation et présentation des outils

- un terminal
- atom / gedit / vscode
- looping (lien à ajouter)
- DataGrip (lien à ajouter)
- Pycharm (lien à ajouter)

```
cd
atom .bashrc
```

Ajouter dans le fichier **.bashrc** de votre **HOME**

```
alias looping='wine /opt/looping-mcd/Looping.exe'
alias datagrip='/opt/DataGrip/bin/datagrip.sh'
alias pycharm='/opt/pycharm/bin/pycharm.sh'
```

6 Découverte du SGBDR MySQL

Remarque : l'objectif n'est pas de connaître toutes les fonctionnalités et subtilités du logiciel **MYSQL** mais d'utiliser toutes les fonctionnalités les plus « classiques » que l'on retrouve dans les logiciels du même type.

6.1 Votre première Table sur Mysql :

- Ouvrir un terminal
- Créer un répertoire : exemple `slbdd`,
- Se déplacer dans ce répertoire
- Ouvrir l'éditeur **SublimText** et créer un fichier texte de nom **tp1.sql**

```
mkdir slbdd
cd slbdd
touch tp1.sql
atom tp1.sql
```

- Depuis le terminal, se connecter au serveur **mysql** et utiliser votre base de données :

```
mysql --user=votreLogin --password=JJMM --host=serveurmysql --database=BDD_votreLogin
## mot de passe: JJMM de votre date de naissance
```

- rajouter au début de votre fichier **tp1.sql**, la commande pour se connecter en commentaire :

[documentation connection à mariadb](#)

et sur mysql

```
mysql --version
mysql --help
-- Usage: mysql [OPTIONS] [database]
-- commande pour se connecter à mysql :
mysql --user=votreLogin --password=JJMM --host=serveurmysql BDD_votreLogin

mysql --user=login --host=serveurmysql.iut-bm.univ-fcomte.fr --database=BDD_login --password=motdepasse
-- autre solution pour se connecter à mysql :
-- mysql -u votreLogin -h serveurmysql -p BDD_votreLogin
-- mot de passe: JJMM
-- solution plus rapide
-- mysql -u votreLogin -h serveurmysql -pJJMM BDD_votreLogin
```

Les commentaires en SQL commence par «**–**» **suivi d'un espace** qui est normalisé ou éventuellement «**#**» qui est admis. **Penser à CTRL MAJ /** sur les éditeurs.

En suivant la documentation sur un [lien de OVH](#), on trouve l'instruction ci-dessous :

```
mysql --host=serveur --user=utilisateur --port=port --password=password nom_de_la_base .
```

Chercher sur internet le **numéro de port** et tester.

Tester avec des ` ou des " autour du mot de passe.

Prendre des Notes (en commentaires) dans vos fichiers SQL (**tp1.sql**) pour garder une trace des TP, vous pourrez utiliser ces TP lors des contrôles devant machine.

6.1.1 Commandes du S.G.B.D.R. MYSQL

```
help
```

- aide pour les commandes MySQL, uniquement sur MySQL (pas d'aide pour le SQL)

```
source tp1.sql
```

- si le fichier existe dans le répertoire courant (du terminal de la connexion sur le serveur mysql), les instructions de ce fichier sont exécutés dans mysql

```
exit
```

- quitter mysql (la connexion sur mysql)

```
show tables;
```

- Afficher les tables de votre base de données (ne pas oublier le caractère ; en fin de ligne)
La commande « show » est une instruction SQL, elle doit se terminer par un ; à la différence des commandes systèmes de MYSQL (qui apparaissent avec la commande help)

```
system ls
```

- exécuter une commande système (shell) : liste le contenu du dossier dans lequel la connexion sur MySQL est ouverte
- Mettre en commentaire ces instructions

```
-- commentaire -- ou #
-- se déplacer dans le bon dossier ( cd S1_BDD)
-- ouvrir un fichier : subl TP1.sql
# mysql -u login -h serveurmysql -p BDD_login
-- mot de passe : jjmm (date de naissance)
# mysql> system ls
-- mysql> source TP1.sql
# mysql> help

-- visualiser les bases de données
# show databases;

-- visualiser les tables dans la BDD
# show tables;
```

script SQL : exemple de commentaires

Ajouter dans le fichier **tp1.sql** les instructions ci-dessous pour créer la table **etudiant** et exécuter ces instructions (*source tp1.sql*)

6.1.2 Premières commandes SQL

Les commandes SQL se terminent par ;. Erreur classique, vous avez oublié le “;” et l’interpréteur SQL de mysql attend la fin de l’instruction SQL

1. Créer une autre table « **etudiant** »
 - INT : entier ,
 - VARCHAR : chaîne de caractères variable

```
CREATE TABLE etudiant (  
    id_etudiant INT  
    , nom_etudiant VARCHAR(25)  
);
```

script SQL

2. Afficher la structure d’une table

```
DESCRIBE etudiant;
```

3. Insérer des enregistrements dans une table:

```
INSERT INTO etudiant (id_etudiant, nom_etudiant) VALUES (1, 'dupond');  
INSERT INTO etudiant VALUES (2, 'durand');  
INSERT INTO etudiant VALUES (1, 'Gauthier');  
INSERT INTO etudiant VALUES (NULL, 'laval');  
INSERT INTO etudiant (nom_etudiant) VALUES ('lepaul');
```

script SQL

NULL : non affecté, non disponible (différent de espace, 0 ou la chaîne de caractères vide “”)

Il est possible de définir les champs que l’on veut remplir (ligne 1 et 5), à défaut il faut donner une valeur à chaque champ.

4. afficher tous les enregistrements d’une table

```
SELECT * FROM etudiant;
```

Remarque : bien faire la différence entre « afficher les enregistrements d’une table » et « afficher la structure d’une table » dans un énoncé (contrôle)

5. supprimer une table **en début de script**

```
DROP TABLE etudiant;
```

Placer cette instruction en début de script pour que vos modifications sur la table soit prises en compte à chaque rechargement du script SQL.

Modifier votre script dans votre fichier à chaque étape, l’objectif est de bien saisir les différences et les points communs entre « NOT NULL » « UNIQUE » et « PRIMARY KEY »

Pour ne plus autoriser un enregistrement avec un identifiant de l’étudiant NULL :

```
CREATE TABLE etudiant (  
    id_etudiant INT NOT NULL  
    , nom_etudiant VARCHAR(25)  
);
```

script SQL

6. Modifier la table en ajoutant la contrainte NOT NULL
7. Insérer les enregistrements précédents dans la table
 - Conclusion : Quelles instructions « INSERT » ne fonctionnent pas et pourquoi?
8. supprimer la table

6.1.3 Clé Primaire, champ unique, champ non “NULL”

9. Modifier la table en ajoutant la contrainte UNIQUE, puis insérer les champs précédents

```
CREATE TABLE etudiant (
    id_etudiant INT NOT NULL UNIQUE
    , nom_etudiant VARCHAR(25)
);
```

script SQL

- Conclusion : Quelles instructions « INSERT » ne fonctionnent pas et pourquoi?

```
CREATE TABLE etudiant (
    id_etudiant INT PRIMARY KEY
    , nom_etudiant VARCHAR(25)
);
```

10. Modifier la table et insérer les champs précédents avec une clé primaire sur l'identifiant

- Conclusion : Quelle différence existe-t'il entre un champ qui a la propriété d'être une clé primaire et un champ qui a comme attribut « UNIQUE NOT NULL » ?

6.1.4 clé primaire : bonne pratique

Définir de préférence les clés primaires à l'aide d'une instruction séparée en fin de table car :

- une **Clé Primaire** peut être composée de plusieurs champs,
- par convention : on définit la clé primaire et les clés étrangères (Semestre 2) en fin de table

```
CREATE TABLE etudiant (
    id_etudiant INT
    , nom_etudiant VARCHAR(25)
    , PRIMARY KEY(id_etudiant)
);
```

script SQL : solution à utilisée par la suite

12. Afficher comment créer la table à l'aide de la commande ci-dessous

```
SHOW CREATE TABLE etudiant;
```

13. Terminer votre script en supprimant la table si elle existe.

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

script SQL : supprimer une table si elle existe

Pour les plus rapides :

14. ajouter un champ dans la table *etudiant* avec la date de naissance de chaque étudiant (nom du champ : *date_naissance*) et insérer quelques valeurs en vous aidant d'internet.

S1 BDD 2019 tp1



<!-- https://pixees.fr/informatiquelycee/term/suj_bac/2022/sujet_02.pdf -
systeme https://pixees.fr/informatiquelycee/term/suj_bac/2022/sujet_05.pdf
https://pixees.fr/informatiquelycee/term/suj_bac/2022/sujet_06.pdf
<https://nsi-snt.ac-normandie.fr/sujets-du-baccalaureat-2022-terminale-nsi> ->