
Département de Mathématiques et Informatique

MIOA301T : BD Avancées

- 1 - Rappels sur les SGBD Relationnels
- 2 - Vues relationnelles
- 3 - Fonctions et procédures stockées
- 4 - Triggers

Rappels sur les SGBD Relationnels

Le modèle relationnel

Le schéma d'une BD relationnelle et ses différentes représentations

Le langage SQL (LDD, LMD, LID)

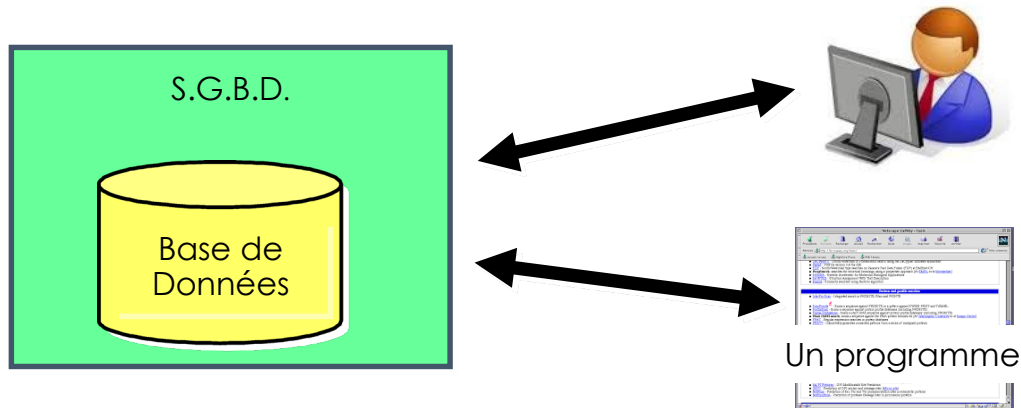
Vus en L1 MIASHS ...

Généralités sur les BD

➤ Une Base de Données (ou **BD**)

Pas BDD !

- Est un ensemble organisé, souvent important, de données gérées comme un tout
- Est mémorisée sur un support rémanent (les données sont persistantes)
- Est gérée par un Système de Gestion de Bases de Données (ou **SGBD**)



Généralités sur les BD relationnelles

- Une BD relationnelle (ou BDR) est une BD structurée conformément au **modèle relationnel**
 - Dans le cadre de ce cours nous utilisons le **SGBDR MariaDB**



- **SQL** (Structured Query Language) est le langage permettant de créer, modifier, interroger une BDR
 - Dans le cadre de ce cours nous utilisons l'**interface web** (interface graphique) fournie par **Adminer** pour transmettre des **requêtes SQL** à MariaDB et visualiser les résultats retournés par MariaDB

Synthèse du modèle relationnel

FOURNISSEURS			
<u>CodeFou</u>	Nom	Ville	Telephone
001	Dupond	Toulouse	05 61 12 13 14
003	Durand	Paris	01 23 45 67 89
004	Dubois	Auch	05 63 45 68 74
006	Bernard	Toulouse	05 61 53 44 88
007	Dupond	Paris	01 32 65 98 55
010	Jacques	Toulouse	05 62 36 59 87

← relation (table)

← attributs (*champs*)← tuple ou n-uplet
(*enregistrement*)schéma
(structure)données
(contenu)

CONDITIONS-ACHATS

<u>CodeFou#</u>	<u>CodePro#</u>	Delai	Prix
001	P01	3	1.00
001	P02	7	3.00
003	P01	1	1.50
003	P02	15	2.50
003	P03	10	25.00
004	P03	12	22.00

clés étrangères (*foreign key*)

PRODUITS

<u>CodePro</u>	Designation	Stock
P01	Crayons	50
P02	Cahiers 50 pages	60
P03	Stylos	10
P04	Gommes	40

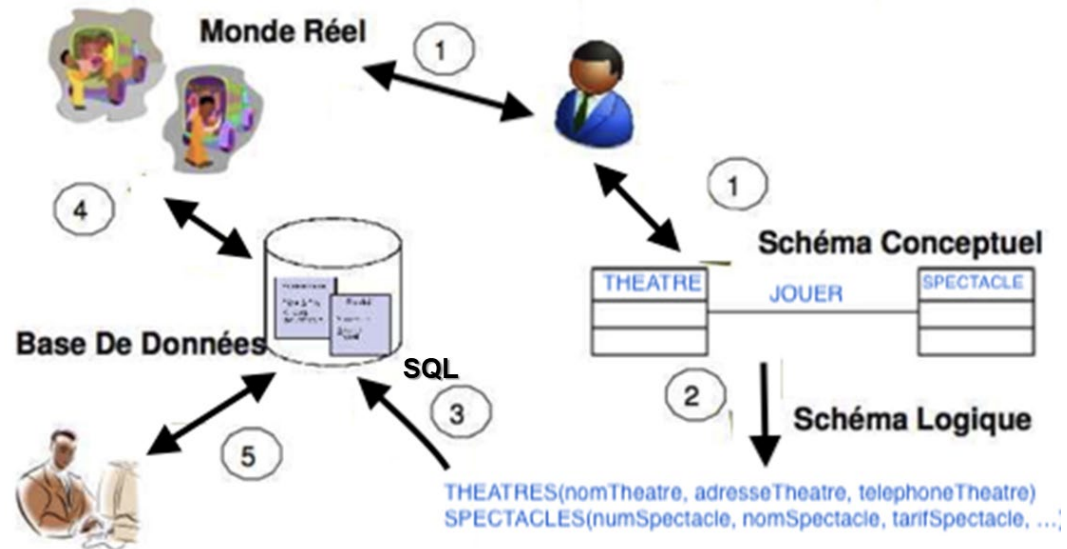
clé primaire (*primary key*)

Le schéma d'une BD et ses différentes représentations

- Le schéma d'une BD décrit en intension son contenu :
 - Il indique ce que peut contenir la B.D. (les **données**) ainsi que certaines **contraintes** permettant d'assurer l'intégrité des données de la base
 - Le schéma d'une B.D. est relativement **stable** dans le temps alors que les données sont soumises à nombreuses mises à jour (insertion, modification, suppression de tuples)

Le schéma d'une BD et ses différentes représentations

- Il y a différents niveaux de description d'un schéma :
 - Conceptuel (MCD)
 - Logique (MLD)
 - Physique (MPD)



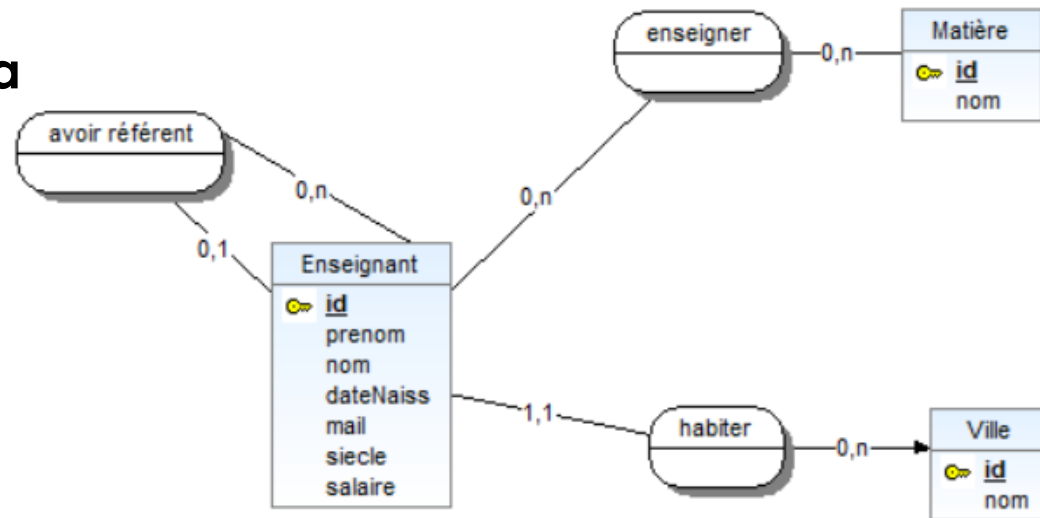
- Illustration à travers un exemple ...

MCD : Illustration à travers une base de données

- Cette représentation (graphique) décrit le **schéma de la BD**. — — — — — →

- Elle le décrit à un niveau conceptuel :

- On parle du schéma conceptuel (ou **MCD**) de la B.D.
- Le formalisme que nous utilisons dans ce cours est le **formalisme Entité-Association**, ou **E-A** (vu en L1)



Rappelez-vous ... et répondez à ces questions :

Quelles données cette BD permet-elle de mémoriser ?

Retrouvez les concepts du modèle E-A de :

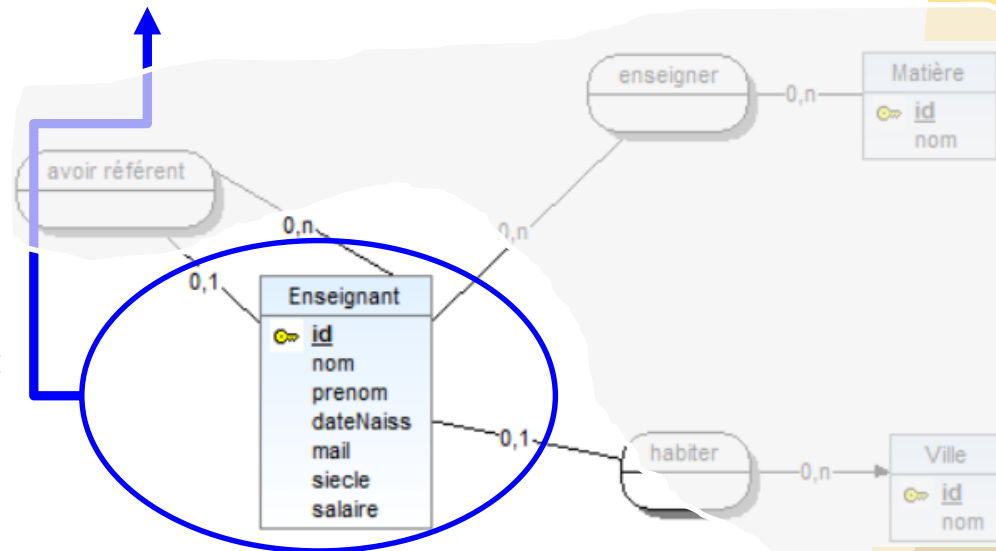
- Type d'entité (TE)
- Type d'association (TA)
- Cardinalités
 - TA « un à plusieurs » (cardinalités x-1 ; y-n)
 - TA « plusieurs à plusieurs » (cardinalités x-n ; y-n)

MLD : Cas de la relation Enseignants

La relation *Enseignants* -->

- Reflète, au niveau logique, le TE *Enseignant* du niveau conceptuel
 - NB : Le modèle pour décrire le niveau logique des relations d'une BDR est le **modèle relationnel**
- La convention de notation que nous utilisons pour le niveau logique dans ce cours est la suivante : **R (A#, B, C, D#)** où :
 - R** est le nom de la relation (un mot commun au pluriel) ; **A**, **B**, **C** et **D** sont ses **attributs**
 - On souligne la **clé primaire**
 - On fait suivre chaque **clé étrangère** du symbole **#** ; on note (A, B)# les clés composées de plusieurs attributs
- Le **schéma logique d'une B.D.** est constitué du schéma logique de chacune des relations de la BD

Enseignants (id, nom, prenom, dateNaiss, mail, siecle, salaire, idVille#, idReferent#)



Retrouvez dans la relation Enseignants les concepts du modèle relationnel de :

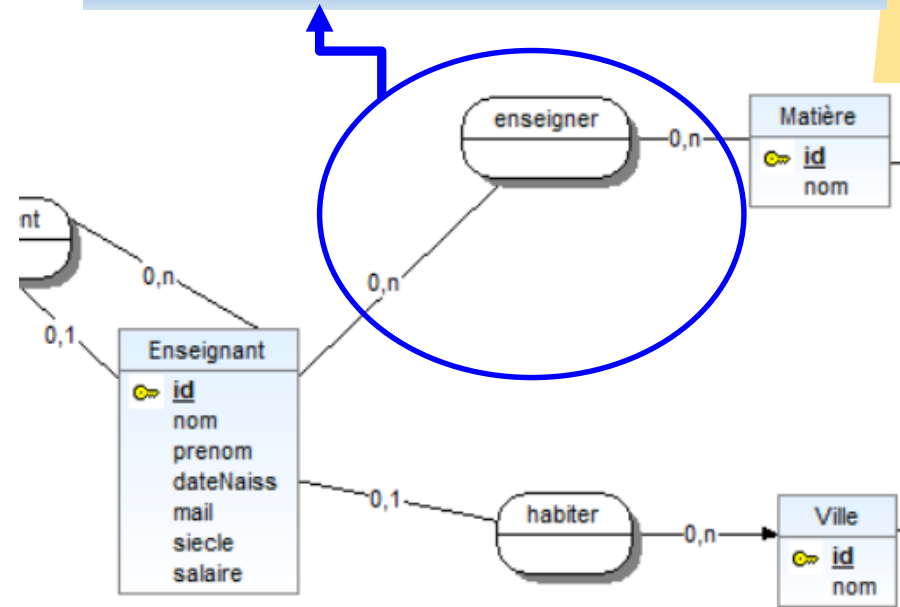
- Attribut**
- Clé primaire**
- Clé étrangère**

Les TA « un à plusieurs » **avoir référent** et **habiter** du schéma E-A ont été traduits sous forme de clés étrangères dans la relation *Enseignants*

MLD : Cas de la relation EnsMatières

La relation *EnsMatières* → **EnsMatières** (idEns#, idMat#)

- Représente, au niveau logique, le type d'association plusieurs-plusieurs enseigner du niveau conceptuel
- On appelle aussi ce genre de relation, une **table de jointure** car :
 - Elle permet de joindre des tuples de la relation *Enseignants* à des tuples de la relation *Matières* (pour mémoriser les matières enseignées par des enseignants), et inversement.



Rappelez-vous ... et répondez à ces questions :

- Combien de clés primaires possède la relation *EnsMatières* ? Combien de clés étrangères ?
- Quel est le type de l'attribut *idEns* ? Est-il **obligatoire** ou **facultatif** ?
- Quel est le type de l'attribut *idMat* ? Est-il obligatoire ou facultatif ?
- Le même enseignant (*idEns*) peut-il apparaître plusieurs fois dans la relation (dans des tuples différents) ? La même matière (*idMat*) peut-elle apparaître plusieurs fois dans la relation (dans des tuples différents) ?

On vient de (re)voir à travers l'exemple que :

- Le nom des relations sont des noms communs, au pluriel
- Toute relation d'une BDR a nécessairement une clé primaire :
 - Il n'y en a qu'une
 - Qui peut être composée de plusieurs attributs
 - Qui est minimale (tous les attributs qui la composent sont nécessaires ; si on retire un attribut, l'ensemble restant ne constitue pas une clé).
Ex. la clé primaire de *EnsMatières*.
- Une relation peut contenir une, voire plusieurs clés étrangères
 - Si la clé primaire référencée est une clé composée de plusieurs attributs, la clé étrangère sera composée du même nombre d'attributs
 - Le nom d'un attribut dans une clé étrangère **peut** être différent de l'attribut de la clé primaire qu'il référence. Ex. les clés étrangères dans *Enseignants*.
 - Le type d'un attribut d'une clé étrangère **doit** être le même que celui de l'attribut auquel il fait référence

MPD - LDD SQL : Cas de la relation Enseignants

- Pour décrire le schéma d'une BD **au niveau physique (MPD)**, en relationnel, on utilise des instructions du **Langage de Description des Données (LDD) SQL**
- **CREATE TABLE** est une instruction qui permet de :
 - créer la **structure** d'une relation,
 - définir des **contraintes d'intégrité** que les tuples de la relation devront respecter lors de la mise à jour des tuples de la BD (vérifications réalisées par le SGBD)

```
CREATE TABLE Enseignants (  
    id                INT,  
    nom               VARCHAR(40),  
    prenom            VARCHAR(40),  
    dateNaiss         DATE,  
    mail              VARCHAR(100),  
    siecle             INT,  
    salaire           INT,  
    idVille           INT,  
    idReferent        INT,  
    primary key (id),  
    foreign key (idReferent)  
        references Enseignants(id)  
);
```

Décrive plus tard

Enseignants (id, nom, prenom, dateNaiss, mail, siecle, salaire, idVille#, idReferent#)

Retrouvez dans cette requête, la définition des éléments suivants :

- Nom de la relation
- Attribut et type d'attribut
- Clé primaire
- Clé étrangère

NB : Une clé étrangère ne peut être définie que lorsque la clé primaire qu'elle référence, a elle-même été définie → La clé étrangère définie sur *idVille* dans la relation *Enseignants* devra être définie après que la relation *Villes* ait été créée

MPD - LDD SQL : Cas de la relation Enseignants (suite)

- Il est possible d'ajouter des contraintes d'intégrité dans la définition des attributs grâce à la **clause CHECK**
- Exemple : comment garantir que le salaire d'un enseignant dans la BD est toujours > 0 ? Que l'adresse mail est de la forme *prenom.nom@ut2j.fr* ?

```
DROP TABLE IF EXISTS Enseignants; -- On supprime la relation au cas où elle existerait
CREATE TABLE Enseignants (        -- On recrée la relation avec les modifications
    id                INT,
    nom               VARCHAR(40),
    prenom            VARCHAR(40),
    dateNaiss         DATE,
    mail              VARCHAR(100) CHECK ( mail = CONCAT(prenom, ".", nom, "@ut2j.fr") ),
    siecle            INT,
    salaire           INT CHECK (salaire > 0),
    ...
);
```

Ou :

```
ALTER TABLE Enseignants -- On modifie la relation déjà créée
    ADD CONSTRAINT email_valid CHECK ( mail = CONCAT(prenom, ".", nom, "@ut2j.fr") );
ALTER TABLE Enseignants
    MODIFY salaire INT CHECK (salaire > 0);
```

MPD – LDD SQL : Cas de la relation Ens-matieres

- D'autres instructions du LDD sont :
 - **ALTER TABLE** pour **modifier** une relation existante (ajouter/modifier/supprimer des attributs ou des contraintes, renommer une table, etc.)
 - **DROP TABLE** pour **supprimer** une relation

Vous pouvez retrouver la syntaxe de ces requêtes sur <https://sql.sh> (faites une recherche sur l'instruction recherchée)

```
DROP TABLE IF EXISTS EnsMatiere;  
CREATE TABLE EnsMatiere (  
    idEns      INT,  
    idMat      INT  
);  
ALTER TABLE EnsMatiere  
    ADD CONSTRAINT PRIMARY KEY (idEns, idMat);  
ALTER TABLE EnsMatiere  
    ADD CONSTRAINT fk_ensmat_ens  
    FOREIGN KEY (idEns)  
    REFERENCES Enseignants (id);
```

EnsMatiere (idEns#, idMat#)

Enseignants (id, nom, prenom , ...)

Ce **code SQL** (dont les instructions/requêtes seront exécutées séquentiellement par le SGBD) permet d'implémenter physiquement la relation *EnsMatiere*. Il consiste à :

1. Supprimer la relation si elle existe déjà
2. (Re)créer la relation avec ses 2 attributs
3. Ajouter la contrainte de clé primaire (clé composée)
4. Ajouter une contrainte de clé étrangère (NB : le nommage *fk_ensmat_ens* est facultatif)

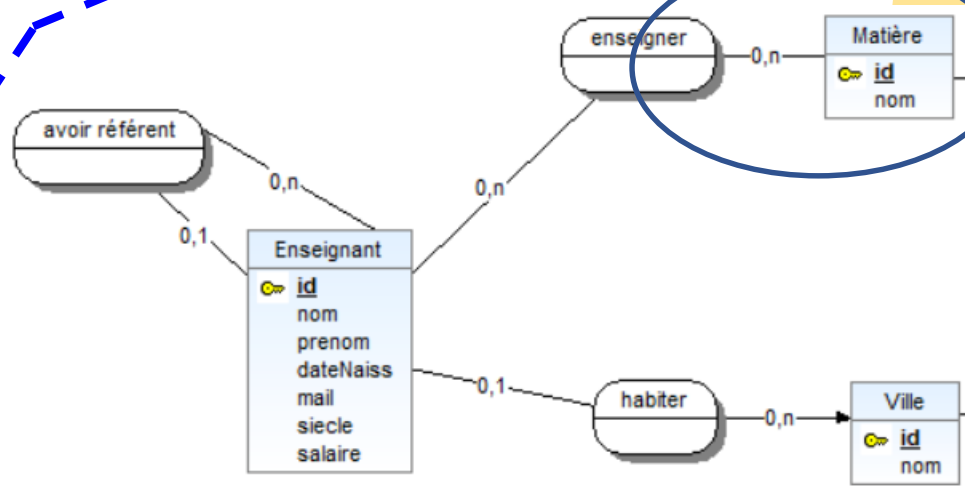
MPD – LDD SQL : Exercice (1)

Les relations Enseignants et EnsMatieres étant définies, il s'agit de modifier la base pour effectuer les modifications signalées en bleu, ci-contre :

Matieres (id, nom)

EnsMatieres (idEns#, idMat#)

- Donnez le code SQL pour implémenter la relation **Matieres** (NB : l'attribut *nom* est obligatoire)
- Donnez le code SQL pour compléter et finaliser (Cf. diapo précédente), la définition de la relation **EnsMatieres**



MPD - SQL LDD : Exercice (2)

- Ecrivez une requête SQL pour ajouter à la relation *Matieres* l'attribut *coef* (coefficient) dont la valeur doit être supérieure à 0 si renseignée,

```
Matieres (id, nom, coef)
```

- Ecrivez une requête SQL pour ajouter la contrainte « le nom d'une matière doit comporter plus de 2 caractères »

MPD - SQL LDD : Adminer (1)

- Les requêtes SQL de création du schéma d'une BD peuvent être exécutées par un SGBDR
- Ex. : Adminer offre une **interface web graphique** pour vous permettre de saisir et exécuter des requêtes SQL

Nom de la base de données

Pour saisir des requêtes SQL

Pour importer et exécuter des requêtes SQL à partir d'un fichier texte

Pour afficher (SELECT) tous les tuples d'une relation

Attention de bien être dans VOTRE Base de Données !

Les relations de la base de données
NB : le nom des **relations des exercices du cours** est préfixé par **C301T_**

MySQL » Serveur » Base de données: 22_ENS_catherine_comparot

Base de données: 22_ENS_catherine_comparot

Modifier la base de données Schéma de la base de données

Tables et vues
Rechercher dans les tables (18)

Rechercher


<input type="checkbox"/>	Table	Moteur?	Interclassement?	Longueur des
<input type="checkbox"/>	C301T_Departements	InnoDB	latin1_swedish_ci	
<input type="checkbox"/>	C301T_Enseignants	InnoDB	latin1_swedish_ci	
<input type="checkbox"/>	C301T_EnsMatières	InnoDB	latin1_swedish_ci	

DB: 22_ENS_catherine_comparot

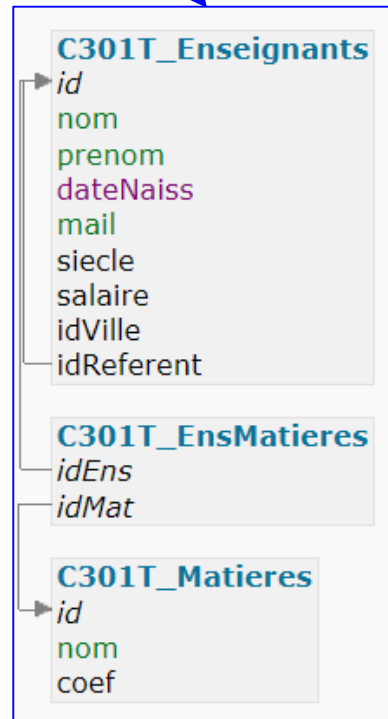
Requête SQL Importer
Exporter
Créer une table

```
select C301T_Departements
select C301T_Enseignants
select C301T_EnsMatières
```

MPD - SQL LDD : Adminer (2)

- Adminer permet de visualiser graphiquement le **modèle logique** du schéma de la BD : pk en italique, flèche fk vers pk ( Schéma de la base de données)

- Exemple :



MLD de la BD :

Enseignants (id, nom, prenom, dateNaiss, mail, siecle, salaire, idVille#, idReferent#)

Matières (id, nom, coef)

EnsMatières (idEns#, idMat#)

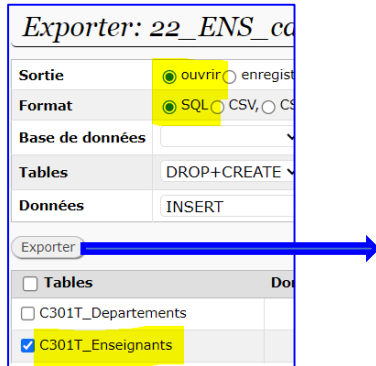
Représentation
Adminer

On vient de (re)voir à travers l'exemple que :

- Il existe différents modèles pour décrire le schéma d'une BD (MCD/MLD/SQL) ; tout dépend à qui la description est destinée
- Connaître les concepts du modèle relationnel pour manipuler le schéma d'une BD, voire les données, à travers une interface graphique (IHM), est un impératif
- Connaître la syntaxe des instructions SQL c'est encore mieux ; cela permet :
 - de travailler plus vite (quelque soit le SGBD, les requêtes sont – quasiment – les mêmes, alors que les IHM diffèrent d'une application à l'autre)
 - de modifier directement du code SQL stocké dans un fichier à importer
 - d'inclure des requêtes SQL dans du code écrit en PL-SQL ou dans un langage hôte tel que Python (PL-SQL sera vu plus tard).


LDD : Mise en œuvre (1)

- Connectez-vous à votre Base MariaDB : <https://mi-mariadb.univ-tlse2.fr/adminer/>
 - Importez et exécutez le code SQL *C1_Enseignants.sql* fourni
 - Vérifiez la création des relations *Enseignants* et *EnsMatières*, et des contraintes de clés primaires et étrangères.
- NB : les contraintes de type *CHECK* d'une relation ne sont visibles qu'avec un **export SQL** du code de création de la relation



```

DROP TABLE IF EXISTS `C301T_Enseignants`;
CREATE TABLE `C301T_Enseignants` (
  `id` int(11) NOT NULL,
  `nom` varchar(40) DEFAULT NULL,
  `prenom` varchar(40) DEFAULT NULL,
  `dateNaiss` date DEFAULT NULL,
  `mail` varchar(100) DEFAULT NULL CHECK (`mail` = concat(`prenom`, '.', `nom`, '@ut2j.fr')),
  `salaire` int(11) DEFAULT NULL CHECK (`salaire` > 0),
  `idVille` int(11) DEFAULT NULL
)
    
```

- Exécutez vos requêtes de ce cours, c'est-à-dire :
 - Création de la table *Matières*, et ajout de la contrainte de clé étrangère sur *idMat* dans *EnsMatières*
 - Ajout du champ *coef* et de la contrainte sur l'attribut *nom* à la relation *Matières*.
- Test de l'ajout de matières ( **Nouvel élément**) : 1, « M », 2 (➔ erreur sur le nom) ; 1, « Maths », -2 (➔ erreur sur le coef); 1, NULL, 2 (➔ l'interface ne permet pas de saisir NULL pour le nom)

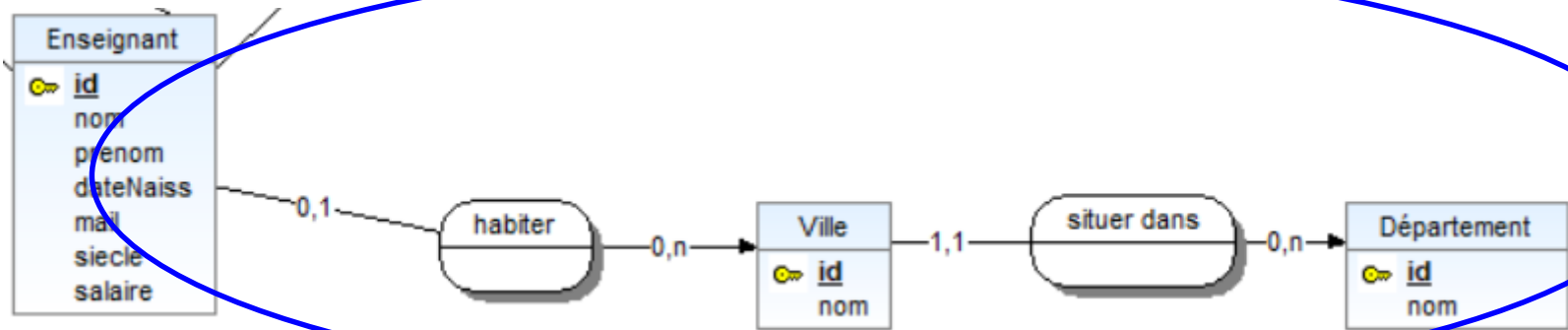
Insérer: C301T_Matières

id		
nom		
coef	NULL	

Enregistrer Enr. et insérer prochain

LDD : Mise en œuvre (2)

- Complétez votre base pour implémenter la portion de schéma E-A suivante :



- Les opérations à réaliser sont les suivantes (les ordonner !):
 1. Ajout de la contrainte de clé étrangère (FK) sur l'attribut idVille dans la relation Enseignants
 2. Création de la relation Villes (id, nom, idDep#) ;
 3. Création de la relation Departements (id, nom)
- Traduisez ces opérations en un code SQL
- Exécutez ce code avec Adminer

LDD : Mise en œuvre (3)

- Exporter le code SQL complet
 - Pour exporter le schéma SQL de la base dans un fichier depuis Adminer :

