

9/10

**Notation :**

- Assez bon tutoriel, du travail
- Le document n'est pas assez structuré (utilisation insuffisante des titres)
- Le centrage des images ne me semble pas être un bon choix
- Confusion entre Serveurs de base de données (Mysql) et solutions Web de regroupement de services (Easyphp, Xampp...)

&lt;/WRAP&gt;

# Création d'un MLD (Modèle Logique des Données) sous MySQLWorkBench

Remarque préalable : Avec le logiciel MySQLWorkBench, il n'est pas possible de créer un MCD. Par conséquent, on crée directement le MLD.

## Préparation du nouveau MLD



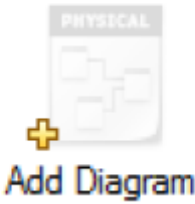
Étapes de la création du MLD

1. Sur la page de démarrage, cliquer sur *Create New EER Model*.

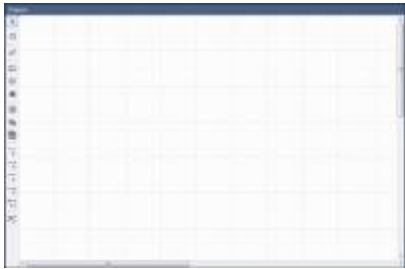


2. Une nouvelle fenêtre s'ouvre. Cliquer sur *ADD Diagramm*.

## Model Overview



3.La feuille de diagramme permettant la saisie du MLD s'affiche.



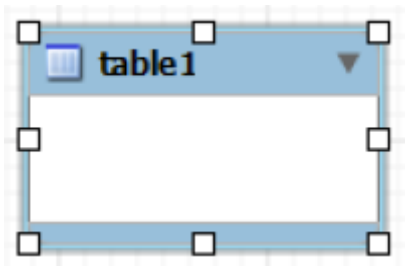
### Création des relations

1.Pour créer une relation (entité dans le MCD), cliquer sur l'icône



puis, cliquer sur la feuille de diagramme pour afficher la relation.

Lorsque la relation est créée, elle se présente sous cette forme. Elle est vide et porte le nom générique Table1.

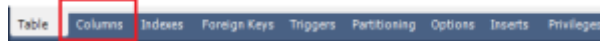


2.Pour ajouter des attributs dans cette relation, double-cliquer dessus.

Un nouvel onglet apparaît alors dans le bas de l'écran.

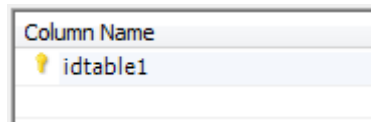


3.Cliquer sur Columns pour ajouter des attributs.



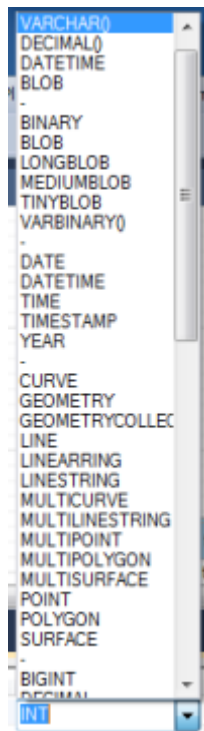
Un onglet permettant la saisie de nouveaux attributs s'affiche.

→Il est possible de donner un nom à l'attribut dans Column Name.

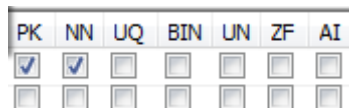


La clé indique que l'attribut (ici, idTable1) est une clé primaire.

→Choisir le type de l'attribut dans DataType.



→Puis, vous pouvez choisir des options pour chacun des attributs.

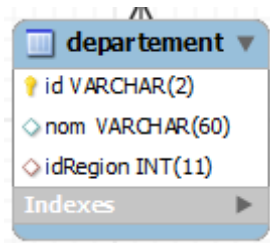


- cocher **PK** (Primary Key) pour que l'attribut soit une clé primaire;
- cocher **NN** (Not Null) pour interdire les valeurs nulles;
- cocher **UQ** (Unique Index) pour interdire les valeurs identiques;
- cocher **BIN** pour que la propriété soit de type booléen;
- cocher **UN** (Unsigned data type) pour que la valeur saisie, pour un attribut de type entier, soit positive;

-cocher **ZF** pour qu'une valeur nulle, d'un attribut de type entier, soit remplacée par un 0;


-cocher **AI** (Auto-Increment) pour que la valeur de la propriété s'incrémente de 1.

Voici un exemple de relation :



Dans ce cas, id est une clé primaire de type texte. L'attribut nom est de type texte. idRegion est la clé étrangère de la relation qui permet une association entre les relations departement et region (non visible ici).

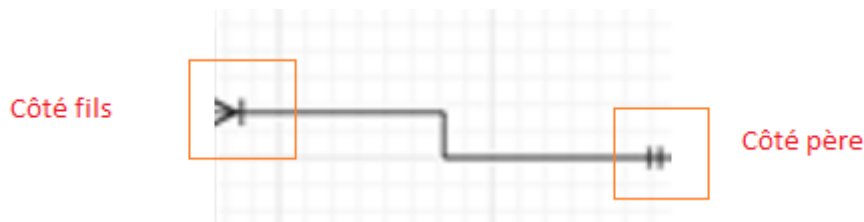
### Création des Contraintes d'Intégrité Fonctionnelles (CIF)

Pour créer une CIF entre deux entités, cliquer sur l'icône .


Puis, cliquer successivement sur les deux relations à associer.

La première relation sur laquelle vous cliquez est la relation fils, la deuxième est la relation père !

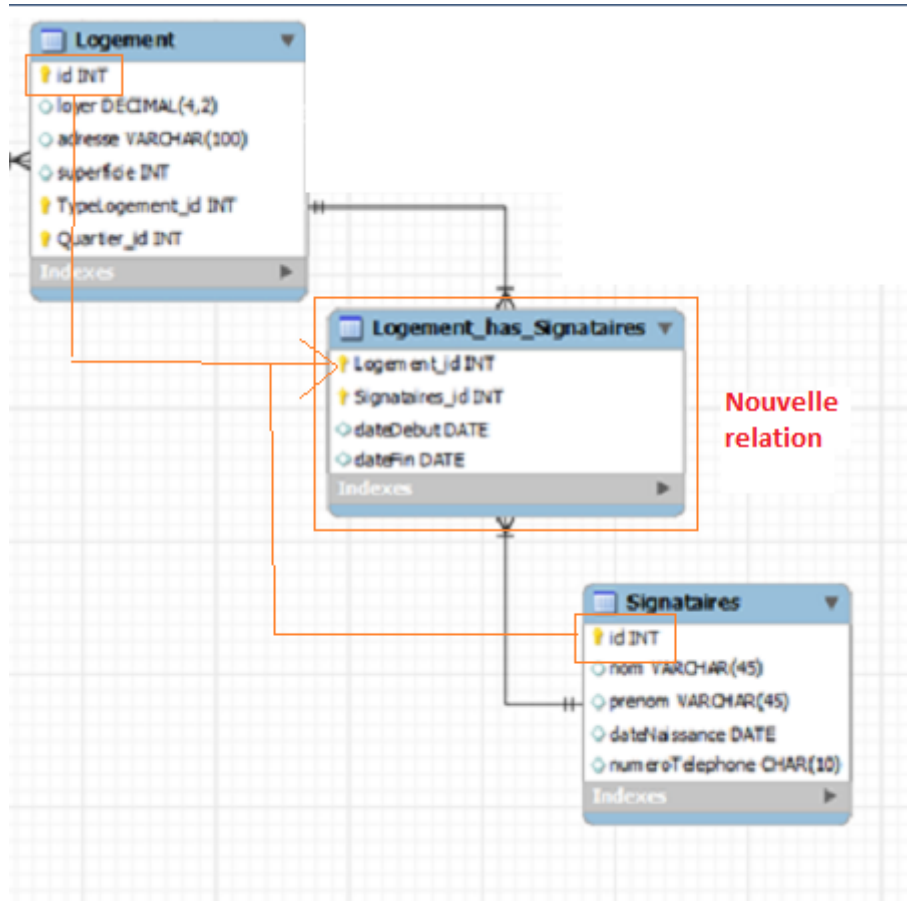
La CIF est créée.



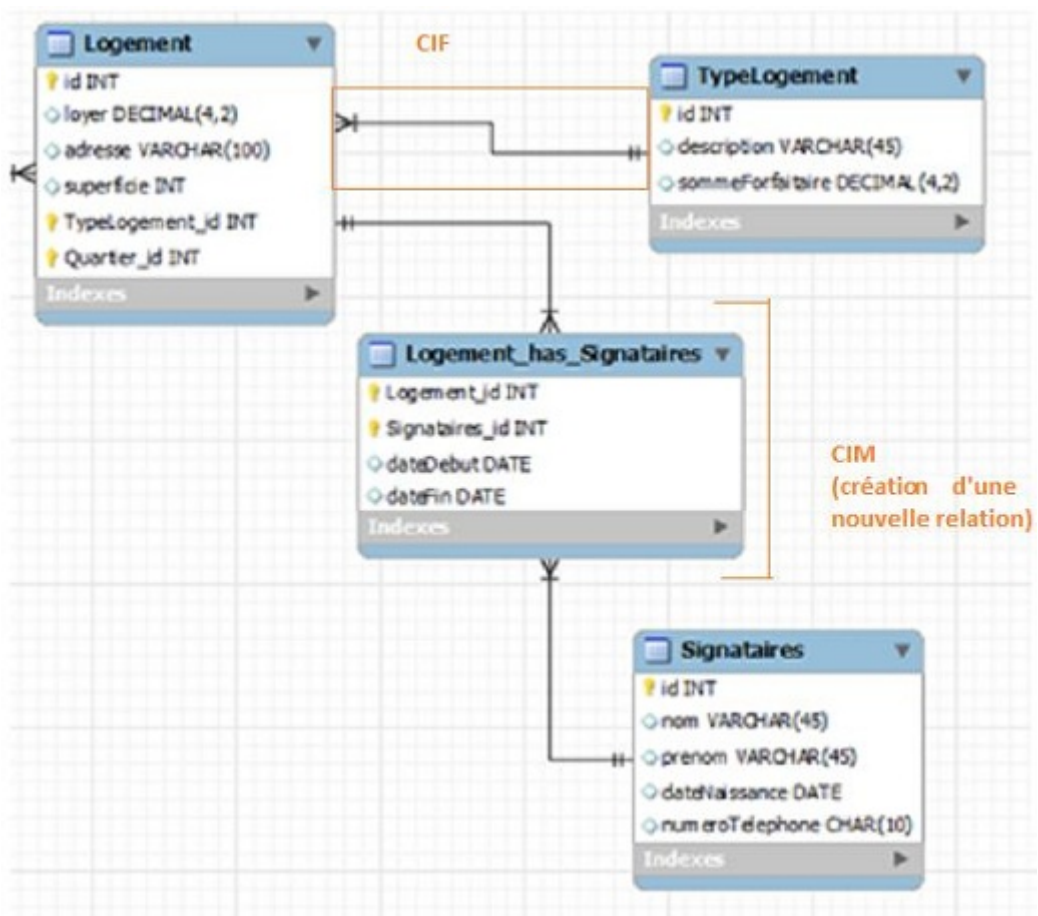
### Création des Contraintes d'Intégrité Multiples (CIM)

Pour créer une CIM entre deux relations, cliquer sur l'icône .

Puis, cliquer successivement sur les deux relations à associer. Une nouvelle relation, ayant pour clé primaire les clés primaires des relations associées, est créée.



Voici un exemple de MLD réalisé sous MySQLWorkBench :

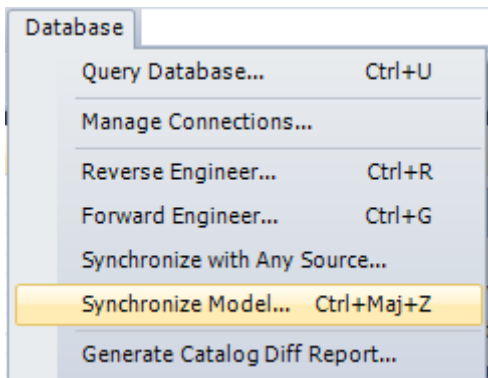


Lorsque la saisie du MCD est terminée, vient l'étape de l'implémentation de ce MLD sous le SGBDR MySQL.

# Génération de la base de données

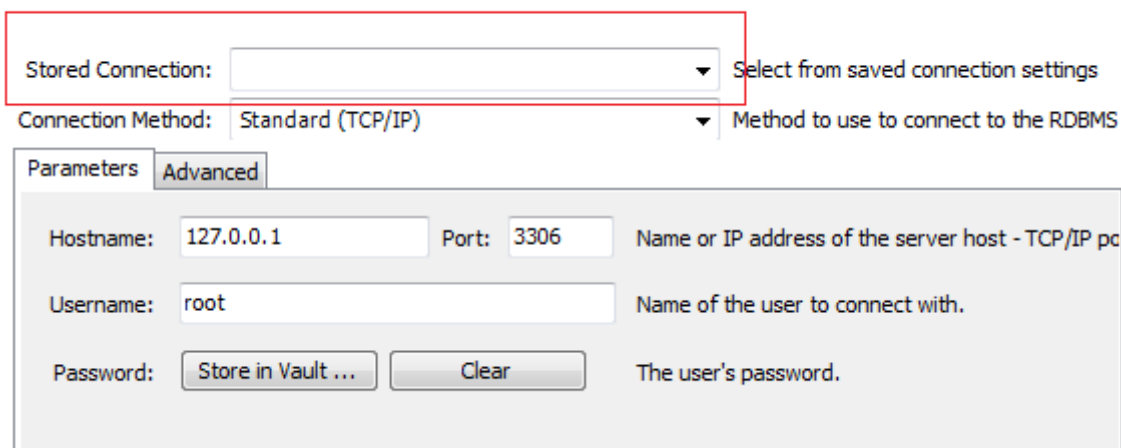
Avant de générer la base de données, ouvrir le SGBDR MySQL, via un serveur tel que Wamp, Xampp ou easyPHP !

Dans MySQLWorkBench, cliquer sur l'onglet Database et choisir *Synchronize Model*.

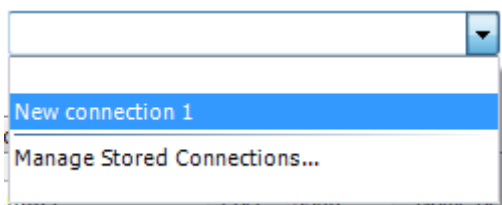


1. Une fenêtre s'ouvre.

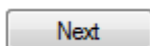
## Set Parameters for Connecting to a DBMS



Choisir la connexion *New Connection 1*.



Puis, cliquer sur le bouton *Next*.



2. Une nouvelle fenêtre s'ouvre.

### Connect to DBMS and Fetch Information

The following tasks will now be executed. Please monitor the execution.  
Press Show Logs to see the execution logs.

- Connect to DBMS
- Retrieve Schema List from Database

Execution Completed Successfully

Fetch finished.

Ne procéder à aucune modification.

Cliquer sur [Next](#).

Next

3. Une troisième fenêtre s'affiche.

### Select Schemata to Synchronize



Select the Schemata to be Processed:

mydb

Sélectionner le schéma à générer en cochant la case (ici, *mydb*)

Cliquer sur le bouton [Next](#).

Next

4. Une quatrième fenêtre s'ouvre.

### Retrieve Object Information

The following tasks will now be executed. Please monitor the execution.  
Press Show Logs to see the execution logs.

- Retrieve Objects from Selected Schemata
- Check Results

Retrieval Completed Successfully

Finished.

Ne modifier aucun élément. Cliquer sur le bouton [Next](#).

Next

5. Une nouvelle fenêtre apparaît. Choisir les relations du schéma que vous souhaitez générer dans la base de données.

### Model and Database Differences

Double click arrows in the list to choose whether to ignore changes, update the model with database changes or vice-versa. You can also apply an action to multiple selected rows.

Model	Update	Source
mydb	→	! N/A
Logement	→	! N/A
TypeLogement	→	! N/A
Quartier	→	! N/A
Commune	→	! N/A
Signataires	→	! N/A
Logement_has	→	! N/A

Cliquer sur le bouton *Next*.

Next

6. Une fenêtre contenant le code SQL exécuté s'affiche.

Ne modifier aucun élément !



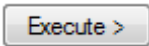
### Preview Database Changes to be Applied

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci ;
USE `mydb`;

CREATE TABLE IF NOT EXISTS `mydb`.`Logement` (
  `id` INT(11) NOT NULL ,
  `loyer` DECIMAL(4,2) NULL DEFAULT NULL ,
  `adresse` VARCHAR(100) NULL DEFAULT NULL ,
  `superficie` INT(11) NULL DEFAULT NULL ,
  `TypeLogement_id` INT(11) NOT NULL ,
  `Quartier_id` INT(11) NOT NULL ,
  PRIMARY KEY (`id`, `TypeLogement_id`, `Quartier_id`),
  INDEX `fk_Logement_TypeLogement` (`TypeLogement_id` ASC),
  INDEX `fk_Logement_Quartier1` (`Quartier_id` ASC),
  CONSTRAINT `fk_Logement_TypeLogement`
    FOREIGN KEY (`TypeLogement_id`)
    REFERENCES `mydb`.`TypeLogement` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_Logement_Quartier1`
    FOREIGN KEY (`Quartier_id`)
    REFERENCES `mydb`.`Quartier` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1
COLLATE = latin1_swedish_ci;
```

Cliquer sur le bouton *Execute*>.



7. Une fenêtre indiquant que la génération s'est correctement déroulée s'affiche.

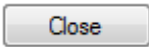
### Progress of Model and Database Synchronization

The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.

- Apply Changes to Database
- Apply Changes to Model

Synchronization Completed Successfully

Cliquer sur le bouton *Close*.



La génération du MLD est terminée. La base de données a été créée.

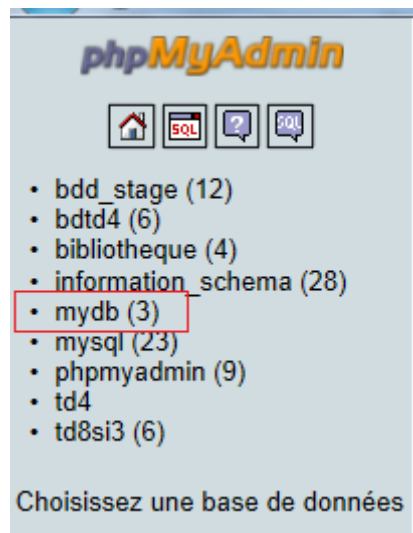
### Consultation de la base de données générée

Le MLD a donc été généré et il vous est possible de consulter la base de données ainsi créée dans le SGBDR

MySQL.



Dans PHPMYAdmin, vous retrouvez la base de données qui vient d'être créée.



Il vous est ainsi possible d'effectuer des manipulations sur cette base de données, en utilisant l'interface graphique ou en ligne de commande (requêtes SQL).

Ce tutoriel a été réalisé par GIROD Pénélope, étudiante en première année de BTS SIO. Version : 1.0(bêta) Date de dernière modification : 16/01/13

From:

<http://slamwiki2.kobject.net/> - **Broken SlamWiki 2.0**

Permanent link:

<http://slamwiki2.kobject.net/si3/etudiants/td7/girod?rev=1358472968>

Last update: **2019/08/31 14:42**

