# Lab 1 : Physical Data Model (PDM) with SQL Data Definition Language (DDL)

**Instructions**

- This TP is based on Chapter 3 - SQL DDL and DML, so please have the course materials available during the lab.
- Create a directory on your computer and store the following in it :
  - a file corresponding to the CDM and LDM modeling
  - the .sql files

**Learning outcomes**

- Produce an MPD from an MLD
- Identify and apply integrity constraints in an MLD and an MPD
- Use the SQL LDD language *(CREATE, ALTER, DROP, CHECK)*
- Manipulate data using SQL LMD operations *(INSERT, UPDATE, DELETE, SELECT)*

## A. Initial design

You have been commissioned by the `Codewarts` school to design and develop a database. At `Codewarts` , students are grouped into houses and compete throughout the year to win the house cup, based on how well they improve in the art of coding. They earn points awarded by teachers for doing well in their subjects, behaving well in class, etc. The purpose of the database is to make it easy to count the points awarded to the different houses and to know how they were awarded.

The school gives you an initial description of its storage needs :

- *Students* identified by an ID, with a first and last name.
- *Houses* with a name, motto, and animal that represents them.
- *Courses* identified by an ID and with a title.
- *Teachers* identified by an ID, with a first and last name.
- A student belongs to a single house, which contains at least one student.
- A student takes at least one class, and a class is taken by at least one student. A grade is assigned to each class taken by a student.
- A course is taught by a teacher, and a teacher teaches one or more courses.
- A teacher assigns points to the different houses. They may not assign any points, and may assign points multiple times. Several different teachers may assign points to a house.

**Questions :**

1. Represent the **CDM** corresponding to these requirements. Use Looping or another modeling tool and save the corresponding file.
2. Use the transformation rules to deduce the **LDM** from this CDM.

# B. Physical Data Model (PDM)

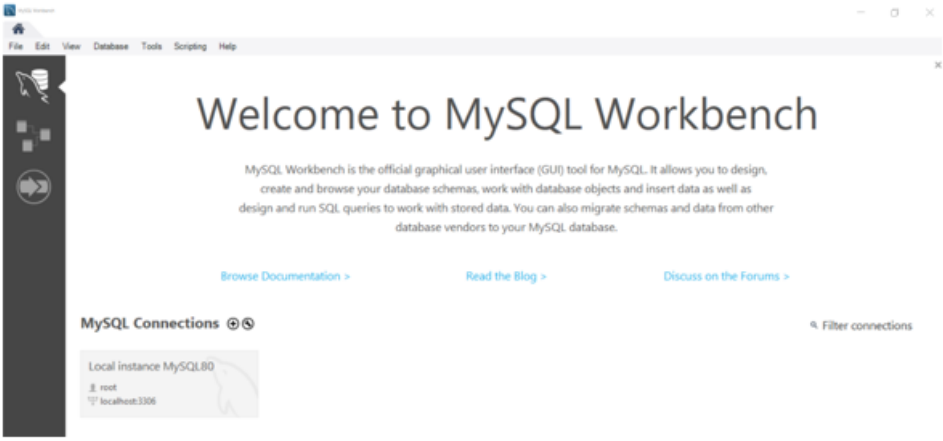
You will now create the database in **MySQL Workbench**.

## B.1. Installation

Install **MySQL Server** and **MySQL Workbench** using the following links :
- on Windows : https ://dev.mysql.com/downloads/windows/installer/8.0.html
Video guide : https ://www.youtube.com/watch ?v=FX0YdJU7Z7E
- on macOS : https ://dev.mysql.com/downloads/installer/
Video guide : https ://www.youtube.com/watch ?v=aVH81OT5XVk

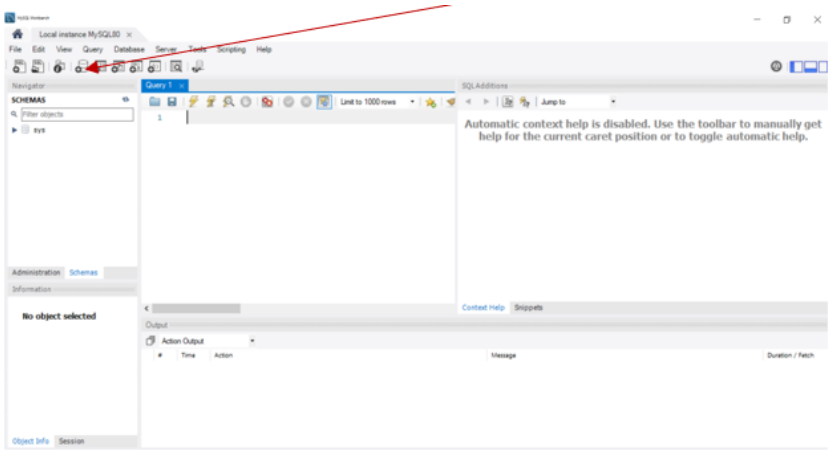
## B.2. Creating the database

### 1. Launch




Connect to the local MySQL server : “local instance MySQL80,” using the credentials defined during installation.

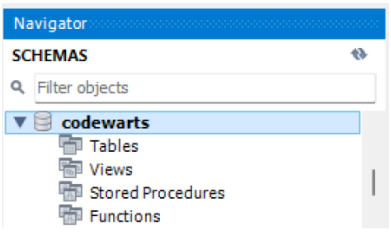
### 2. Creating the schema

1. 

Choose the name of your database schema and click on the schema creation icon. Here, for example, use "codewarts" (note : the name must be in lowercase).


The SQL command corresponding to the creation of the schema is displayed : `create schema codewarts;` You can also type it directly into the code editor and execute it by clicking on the yellow lightning bolt :

2. The schema is displayed on the left tab. You must **double-click** on its name to select it and make it appear **in bold** :

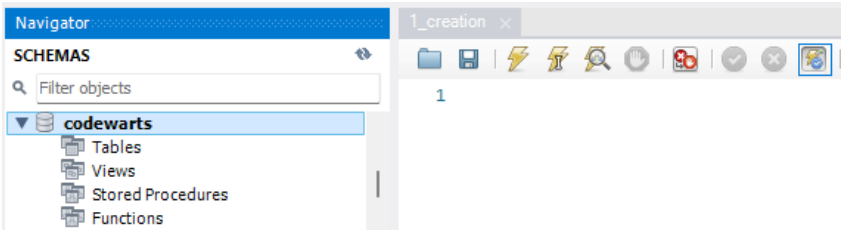



You can also use the command `use codewarts;` to do this.

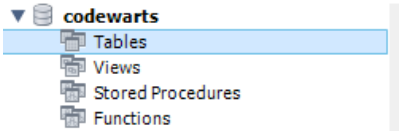
A schema must be **selected** in order to store the tables you are going to create.

## 3. Creating tables

1. Create a file `1_creation.sql` . Use "File > Save Script As" to save any script created with  .




2. Translate the MLD you obtained into MPD using the `CREATE` command.
3. Run the script with your commands by clicking on the lightning bolt.
4. Check that all tables have been added :
   - *With commands* : run the `show tables` query, then the `describe table_name` query, replacing `table_name` with the name of each of the tables created.
   - *Graphically* : by double-clicking on the Tables




**Note :** remember to refresh the diagrams by clicking on

## C. Second design

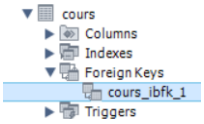
After you submit the first model, `Codewarts` gets back to you because it actually wants to add information to the database.

The school asks you to modify the following information :

- Make it possible for a course to be taught by several teachers (replacement in case of absence),
- Add the teaching field of each teacher,
- Store which teacher is in charge of which house. A teacher can only be in charge of one house.
- Add the gender of each student to be able to verify gender diversity.

1. Modify your CDM to add this information.
2. Generate the new LDM and PDM.
3. To make the changes in the PDM, ***do not change the code in the*** `1_creation.sql` ***file*** but create a new file `2_modification.sql` in which you write the commands allowing the modification/addition of tables with `ALTER`.

   **Note :** here we must delete an attribute that is a foreign key. We must first delete the foreign key constraint before deleting the attribute. To do this :
   1. Retrieve the name of the constraint :




   2. Write a `DROP` for the constraint, for example :
      `ALTER TABLE House DROP FOREIGN KEY prof_fk;` with `prof_fk` replaced by the name you retrieved.

## D. Validation constraints

1. Create a new file `3_constraints.sql`
2. Add the following validation constraints, using `CHECK` :
   - Grades must be letter grades and must be included in `A` , `B` , `C` , `D` , `E` , `F` .
   - It is not permitted to add or remove more than 200 points.
   - Course titles must follow a format such as : `Course name - Course code` . Both pieces of information must be present and separated by `" - "` .

## E. Populate the database

Before `Codewarts` fills the database with its real data, you want to generate test data.

1. Open the file containing the prompt database to be used.
2. Analyze and fill in the prompt :
   - Fill it in with the **relational model** you obtained *(to be done)*.
   - Specify precisely which attributes are the primary keys *(to be done)*.
   - Specify the expected number of rows in each table. Information about the houses is provided by `CodeLard` , the rest is generated by the AI.
   - Specify that there must be a correspondence between the values of the foreign keys and the primary keys to which they refer. The order of insertion is important in order to comply with **integrity constraints**.
   - Copy the **validation constraints** that you have *(to be done)* ready to be executed.
3. Run your finalized prompt on a generative AI of your choice.
4. Save the resulting code in a `4_insertion.sql` script and check that you have no errors when entering the data.

## F. Manipulating data

1. View the data in the different tables with a query such as : `SELECT * from Table_Name` .
2. The students who are taking the retakes are those who obtained an `E` or an `F` . They have all worked hard to improve their grades, which are increased by a `D` . Change all `E` and `F` grades to `D` .
3. The school principal considered that there had been some abuse in certain grade assignments. Delete all assignments corresponding to an addition or removal of more than `150` points.

## To go further...

Can you write the query that will tell you who won the House Cup in 2025 ?