# Utilisation de PDO pour MySQL

1. Créer une base de données, sous MySQL, nommée par exemple SOFTWARE, et exécutez le script fourni « 005 - Script - Brut - Format MySQL - V2.sql »
2. Modifiez le programme **EXO1\_RequeSelectTableJeux\_MySQL.php** en tenant compte de votre contexte (le nom de la BDD que vous avez choisi)

# Utilisation de PDO pour SQL Server

## Installation

1. Installer la DLL pour SQL Server et effectuer l’ajout du driver dans PHP / Laragon comme indiqué dans le cours.
2. Créez un utilisateur SQL Server pouvant accéder à votre base de données
3. Modifiez le programme **EXO1\_RequeSelectTableJeux\_SQLServer.php** et le programme **EXO1\_Repository\_SQLServer.php** en tenant compte de votre contexte (le nom de la BDD que vous avez choisi)

## Analyse du programme

1. Quel type de classe est la classe ***PDORepository***, située dans le fichier : **EXO1\_Repository\_SQLServer.php**
2. Donnez l’ensemble des instructions qu’effectue la ligne de code suivante :



Et expliquez ce qu’elles font

1. Qu’est-ce que le mot clef ***self*** ?

## Affichage de données de la table SOFTWARE.REF\_LIB\_JEUX

Adaptez le code pour récupérer uniquement les code jeux et libellé jeux de la table SOFTWARE.REF\_LIB\_JEUX

2 possibilités : accès par indice ou par libellé. Cf page 6 du cours.

**Donnez les 2 solutions.**

## Lier les colonnes d’un résultat aux attributs d’une classe

Nous disposons d’une table dans SQL Server, située dans le schéma PROCS, nommée FORMATIONELEVES, avec les colonnes suivantes : ID\_ELEVE, NOM\_ELEVE, NB\_HEURES\_COURS\_INFO et CODE\_FILIERE

Nous souhaitons utiliser cette table dans notre programme PHP, et nous avons codé une classe PHP nommée ***FormationEleve*** codée dans le fichier **EXO2\_FormationEleve.php**.

Nous souhaitons pouvoir lier notre application à nos données, puis ensuite ajouter un formulaire de saisie traité par PHP pour ajouter de nouvelles données.

**Partie A : lecture de donnée : modification de la classe EXO2\_FetchClasse.php**

Nous avons commencé à coder une application dans le script : **EXO2\_FetchClasse.php**

1. utiliser les bons paramètres de la méthode setFetchMode (appliquée au « bon » objet) pour lier notre classe *FormationEleve* au résultat de la requête
2. afficher le contenu de l’objet $eleve à chaque itération
3. Quelle est la « *particularité »* de cette façon de faire, quand on pense à la portée des attributs de notre classe *FormationEleve* ? Etes-vous surpris ?

**Partie B : test d’insertion de donnée**

A la fin de votre classe **EXO2\_FetchClasse.php**, vous allez insérer de nouvelles données.

1. Créer un nouvel objet instance de la classe FormationEleve
2. Alimentez ses attributs nom\_eleve, nb\_heures\_cours\_info et code\_filiere
3. Ajoutez le code d’insertion dans la base de données, en prenant exemple sur le cours, soit en utilisant des paramètres nommés, soit des paramètres positionnels

**Partie C : formulaire de saisie**

Créez 2 nouveaux fichiers PHP :

1. Un fichier EXO3\_FormulaireEleve.php, qui contient
   1. un formulaire pointant vers EXO3\_AjoutEleve.php
   2. un champ de type texte pour saisir le nom
   3. un champ de type valeur pour saisir le nombre d’heures
   4. un champ de type texte pour saisir le code filière
2. Un fichier EXO3\_AjoutEleve.php
   1. Ce code récupère les données du formulaire
   2. Ajoute les données saisies à la table PROCS.FORMATIONELEVES