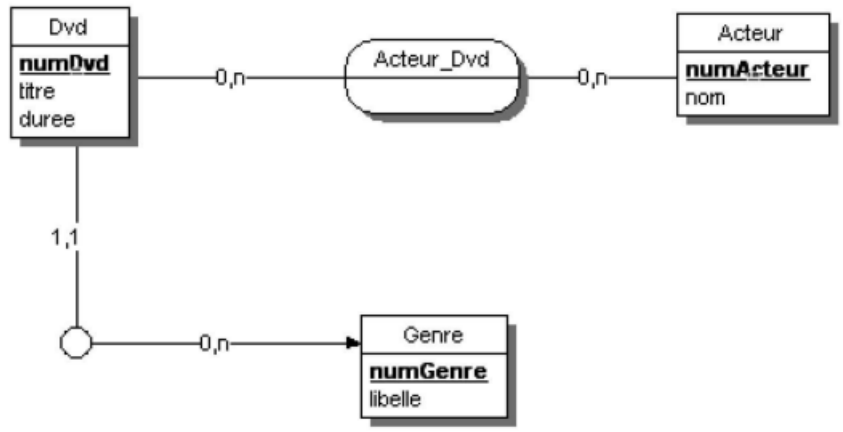
Création d'une application de gestion de DVD en C# :

Première Partie : *Base de données*

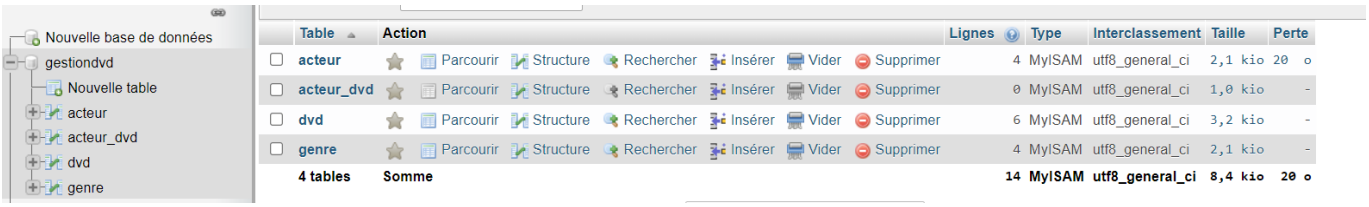
1.a/ Modélisation des tables de la base de données à l'aide de WinDesign :



1.b/ Génération du script de la base de données (Extrait) :

```
# ------------------------------------------------------------------------
#       TABLE : DVD
# ------------------------------------------------------------------------

CREATE TABLE IF NOT EXISTS DVD
 (
   NUMDVD INTEGER(4) NOT NULL AUTO_INCREMENT ,
   NUMGENRE CHAR(4) NOT NULL  ,
   TITRE VARCHAR(30) NULL  ,
   DUREE INTEGER(2) NULL
   , PRIMARY KEY (NUMDVD)
 )
 comment = "";

# ------------------------------------------------------------------------
#       TABLE : GENRE
# ------------------------------------------------------------------------

CREATE TABLE IF NOT EXISTS GENRE
 (
   NUMGENRE CHAR(4) NOT NULL  ,
   LIBELLE VARCHAR(30) NULL
   , PRIMARY KEY (NUMGENRE)
 )
 comment = "";

# ------------------------------------------------------------------------
#       TABLE : ACTEUR
# ------------------------------------------------------------------------

CREATE TABLE IF NOT EXISTS ACTEUR
 (
   NUMACTEUR INTEGER(2) NOT NULL AUTO_INCREMENT ,
   NOM VARCHAR(30) NULL
   , PRIMARY KEY (NUMACTEUR)
 )
 comment = "";

# ------------------------------------------------------------------------
#       TABLE : ACTEUR_DVD
# ------------------------------------------------------------------------
```

2/ Création de la base de données sous phpMyadmin à partie du script généré (Utilisation de WampServer) :



3/ Liaison de la base de données avec Visual Studio :

- ### Étape 1 : installation de MySQL Connector/NET

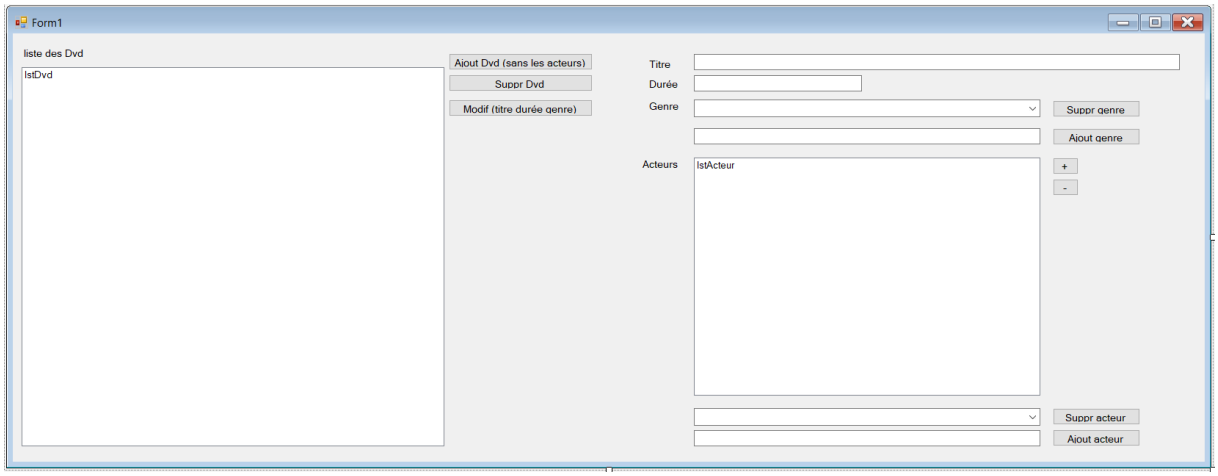
  MySqlConnector est un fournisseur de données ADO.NET pour MySQL Server, MariaDB, Percona Server, Amazon Aurora, Azure Database pour MySQL, Google Cloud SQL pour MySQL.
  Il fournit des implémentations de **DbConnection**, **DbCommand**, **DbDataReader**, **DbTransaction** et les classes nécessaires pour interroger et mettre à jour les bases de données à partir du code managé.

- ### Étape 2 : Configuration sous Visual Studio

  À partir de l'onglet Explorateurs de Serveurs, on choisit l'ajout d'une connexion ainsi on saisira les informations de connexion qui le ServerName, l'UserName et le mot de passe de la base de données.

Deuxième partie : *Implémentation de l'application*

4/ Création de la maquette de l'application



5/ Ajout de la classe ConnexionSql afin de permettre la connexion à la base de données ajoutée : Extrait du code

connexionsql.cs | Form1.cs | Form1.cs [Design]

gestionDvd | gestionDvd.ConnexionSql

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   using System.Data.Sql;
8   using MySqlConnector;
9
10  namespace gestionDvd
11  {
        35 références
12      class ConnexionSql
13      {
14          // propriétés
15          private bool finCurseur=true; // fin du curseur atteinte
16          private MySqlConnection connection; // chaine de connexion
17          private MySqlCommand command; // envoi de la requête à la base de données
18          private MySqlDataReader reader;
19
20
21          // constructeur
                17 références
22          public ConnexionSql(string chaineConnection)
23          {
24              this.connection = new MySqlConnection(chaineConnection);
25              this.connection.Open();
26          }
27
28          // execution d'une requete select
                10 références
29          public void reqSelect(string chaineRequete)
30          {
31              this.command = new MySqlCommand(chaineRequete, this.connection);
32              this.reader = this.command.ExecuteReader();
33              this.finCurseur = false;
34              this.suivant();
```

90 % | Aucun problème détecté

6/ Implémentation des méthodes : Extrait de code

```
1 référence
private void btnAjoutActeur_Click_1(object sender, EventArgs e)
{
    this.ajoutDansCombo(this.txtAjoutActeur, this.cboActeur, "acteur", "nom");
    // place le focus sur la liste des dvd
    this.lstDvd.Select();
}

1 référence
private void lstDvd_SelectedIndexChanged_1(object sender, EventArgs e)
{
    // demande de mise à jour de l'affichage des informations du dvd
    //this.affichInfosDvd();
}

1 référence
private void btnSupprGenre_Click_1(object sender, EventArgs e)
{
    // requete de recherche de lien entre genre que l'on veut supprimer et dvd
    string requete = "select * from dvd d join genre g on (d.numgenre=g.numgenre) where libelle=\"" + this.cboGenre.SelectedItem + "\"";
    // gestion de la suppression du genre, si c'est possible
    this.supprDansCombo(this.cboGenre, requete, "genre", "libelle");
    // place le focus sur la liste des dvd
    this.lstDvd.Select();
}

1 référence
private void btnSupprActeur_Click_1(object sender, EventArgs e)
{
    // requete de recherche de lien entre acteur que l'on veut supprimer et dvd
    string requete = "select * from dvd d join acteur_dvd ad on (d.numdvd=ad.numdvd) join acteur a on (ad.numacteur=a.numacteur) where a.nom=\"" + this.cboActeur.SelectedItem
    // gestion de la suppression de l'acteur, si c'est possible
    this.supprDansCombo(this.cboActeur, requete, "acteur", "nom");
    // place le focus sur la liste des dvd
    this.lstDvd.Select();
}
```

INDEX : *Tableau des fonctionnalités des méthodes (Extrait)*

Clic sur btnAjoutActeur	Contrôler que txtAjoutActeur est bien rempli et que l'acteur saisi dans txtAjoutActeur n'est pas déjà présent dans cboActeur (sinon l'ajout ne doit pas se faire). Ajouter le contenu de txtAjoutActeur dans le combo et dans la base de données.
Clic sur btnSupprActeur	Contrôler qu'un élément est sélectionné dans cboActeur. Contrôler dans la base de données qu'aucun dvd n'est lié à cet acteur (sinon la suppression ne doit pas se faire). Supprimer l'acteur dans la base de données et dans cboActeur.
Clic sur btnAjoutActeurDvd	Contrôler qu'un acteur est bien sélectionné dans cboActeur et que l'acteur n'est pas déjà présent dans lstActeur (les acteurs du film) sinon l'ajout n'est pas possible. Récupérer dans la base de données numacteur correspondant à l'acteur sélectionné. Ajouter le couple (acteur, dvd) dans la table acteur_dvd (pour associer cet acteur avec ce dvd). Ajouter l'acteur dans lstActeur.
Clic sur btnSupprActeurDvd	Contrôler qu'un acteur est bien sélectionné dans lstActeur. Récupérer dans la base de données numacteur correspondant à l'acteur sélectionné. Supprimer le couple (acteur, dvd) dans la table acteur_dvd (pour dissocier cet acteur de ce dvd). Supprimer l'acteur de lstActeur.
Clic sur btnAjoutDvd	Contrôler que le titre, la durée sont bien remplis et un genre sélectionné. Contrôler que le titre n'existe pas déjà (sinon l'ajout ne doit pas se faire). Avertir que les acteurs ne seront pas ajoutés. Récupérer dans la base de données le numgenre correspondant au genre sélectionné. Ajouter le dvd dans la base de données. Réinitialiser complètement la fenêtre (comme lors du chargement).
Clic sur btnSupprDvd	Demander une confirmation de suppression. Supprimer dans la base de données, le dvd sélectionné et les liens avec les acteurs (dans acteur_dvd).