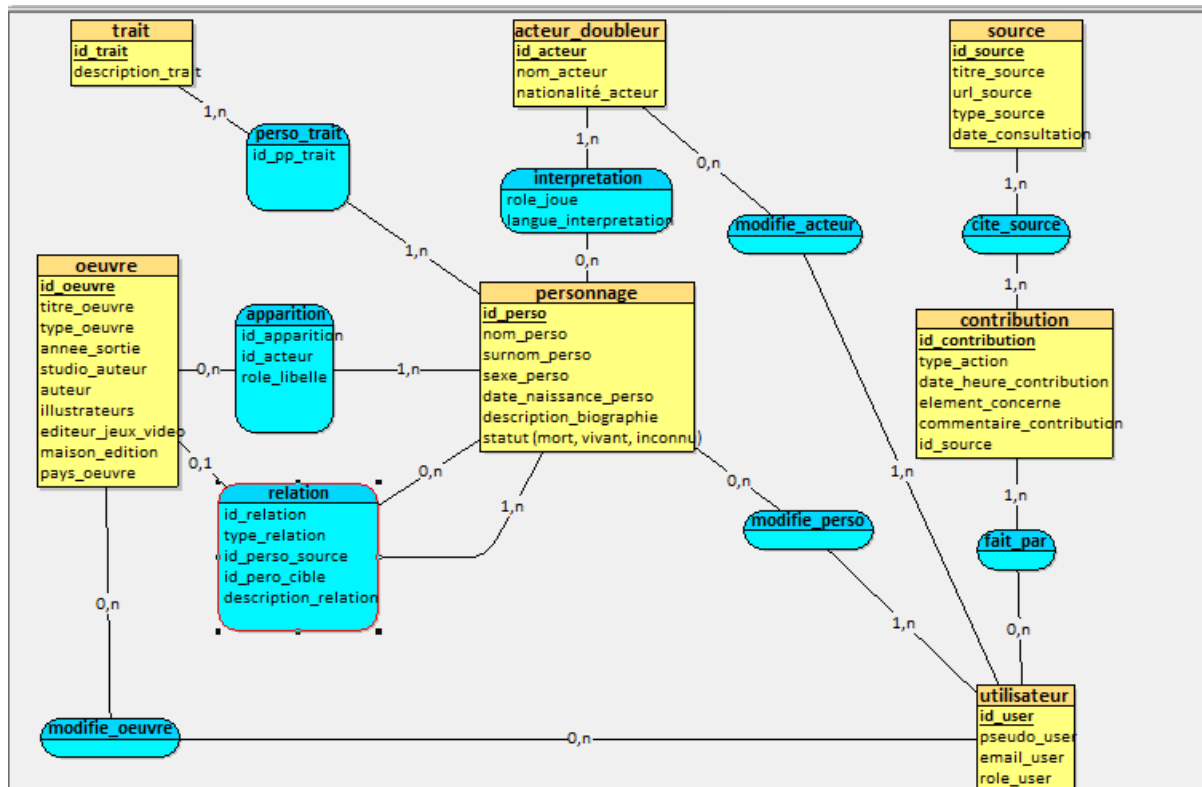


# Mini- projet BDD Partie 2

Naëlle Clara IBOHN MANDJOMBE

## MCD



## MLD

**trait** = (id\_trait VARCHAR(50), description\_trait VARCHAR(50));

**acteur\_doubleur** = (id\_acteur VARCHAR(50), nom\_acteur VARCHAR(50), nationalité\_acteur VARCHAR(50));

**source** = (id\_source VARCHAR(50), titre\_source VARCHAR(50), url\_source VARCHAR(50), type\_source VARCHAR(50), date\_consultation INT);

**oeuvre** = (id\_oeuvre VARCHAR(50), titre\_oeuvre VARCHAR(50), type\_oeuvre VARCHAR(50), annee\_sortie INT, studio\_auteur VARCHAR(50), auteur VARCHAR(50), illustreurs VARCHAR(50), editeur\_jeux\_video VARCHAR(50), maison\_edition VARCHAR(50), pays\_oeuvre VARCHAR(50));

**personnage** = (id\_perso VARCHAR(50), nom\_perso VARCHAR(50), surnom\_perso VARCHAR(50), sexe\_perso VARCHAR(50), date\_naissance\_perso INT, description\_biographie VARCHAR(50), statut\_mort\_vivant\_inconnu VARCHAR(50));

**contribution** = (**id\_contribution** VARCHAR(50), type\_action VARCHAR(50),  
date\_heure\_contribution INT, element\_concerne VARCHAR(50),  
commentaire\_contribution VARCHAR(50), id\_source VARCHAR(50));

**utilisateur** = (**id\_user** VARCHAR(50), pseudo\_user VARCHAR(50), email\_user  
VARCHAR(50), role\_user VARCHAR(50));

**perso\_trait** = (**#id\_trait, #id\_perso**, id\_pp\_trait VARCHAR(50));

**apparition** = (**#id\_oeuvre, #id\_perso**, id\_apparition VARCHAR(50), id\_acteur  
VARCHAR(50), role\_libelle VARCHAR(50));

**relation** = (**#id\_oeuvre**, id\_relation VARCHAR(50), type\_relation VARCHAR(50),  
id\_perso\_source VARCHAR(50), id\_perso\_cible VARCHAR(50), description\_relation  
VARCHAR(50), #id\_perso, #id\_perso\_1);

**modifie\_oeuvre** = (**#id\_oeuvre, #id\_user**);

**interpretation** = (**#id\_acteur, #id\_perso**, role\_joue VARCHAR(50), langue\_interpretation  
VARCHAR(50));

**modifie\_perso** = (**#id\_perso, #id\_user**);

**fait\_par** = (**#id\_contribution, #id\_user**);

**modifie\_acteur** = (**#id\_acteur, #id\_user**);

**cite\_source** = (**#id\_source, #id\_contribution**);

## MPD

Fichier 1\_creation.sql et 2\_contraintes.sql dans mon répertoire Github.

## Insertion des données

### Prompt donné à ChatGPT

Génère un script SQL d'insertion de données volumineux et réaliste pour une base de données sur des personnages fictifs issus de la culture populaire (films, séries, romans, jeux vidéo...), leurs œuvres ainsi que les acteurs/ doubleurs et auteurs. Utilise le modèle relationnel suivant :

```
create table trait(
id_trait varchar(50) primary key,
description_trait varchar(50) not null );
```

```
create table acteur_doubleur(  
  id_acteur varchar(50) primary key,  
  nom_acteur varchar(50) not null,  
  nationalité_acteur varchar(50));
```

```
create table oeuvre(  
  id_oeuvre varchar(50) primary key,  
  titre_oeuvre varchar(50) not null,  
  type_oeuvre varchar(50) not null,  
  annee_sortie int,  
  studio_auteur varchar(50),  
  auteur varchar(50) not null,  
  illustrateurs varchar(50),  
  editeur_jeux_video varchar(50),  
  maison_edition varchar(50),  
  pays_oeuvre varchar(50) not null);
```

```
create table personnage(  
  id_perso varchar(50) primary key,  
  nom_perso varchar(50) not null,  
  surnom_perso varchar(50),  
  sexe_perso varchar(50),  
  date_naissance_perso DATE,  
  description_biographie varchar(50) not null,  
  statut_mort_vivant_inconnu varchar(50));
```

```
create table source(  

```

```
id_source varchar(50) primary key,  
titre_source varchar(50) not null,  
url_source varchar(50),  
type_source varchar(50) not null,  
date_consultation DATE);
```

```
create table contribution(  
id_contribution varchar(50) primary key,  
type_action varchar(50) not null,  
date_heure_contribution int,  
element_concerne varchar(50) not null,  
commentaire_contribution varchar(50),  
foreign key (id_source) references source(id_source));
```

```
create table utilisateur(  
id_user varchar(50) primary key,  
pseudo_user varchar(50) unique,  
email_user varchar(50) not null,  
role_user varchar(50));
```

```
create table perso_trait(  
id_pp_trait varchar(50),  
foreign key (id_trait) references trait(id_trait),  
foreign key (id_perso) references personnage(id_perso));
```

```
create table apparition(  
id_apparition varchar(50) primary key,  
role_libelle varchar(50),
```

```
foreign key (id_oeuvre) references oeuvre(id_oeuvre),  
foreign key (id_perso) references personnage(id_perso),  
foreign key(id_acteur) references acteur_doubleur(id_acteur));
```

```
create table relation(  
id_relation varchar(50) primary key,  
type_relation varchar(50),  
id_perso_source varchar(50),  
id_perso_cible varchar(50),  
foreign key (id_oeuvre) references oeuvre(id_oeuvre));
```

```
create table modifie_perso(  
foreign key (id_perso) references personnage(id_perso),  
foreign key(id_user) references utilisateur(id_user));
```

```
create table fait_par(  
foreign key (id_contribution) references contribution(id_contribution),  
foreign key(id_user) references utilisateur(id_user));
```

```
create table modifie_acteur(  
foreign key (id_acteur) references acteur_doubleur(id_acteur),  
foreign key(id_user) references utilisateur(id_user));
```

```
create table cite_source(  
foreign key (id_source) references source(id_source),  
foreign key(id_contribution) references contribution(id_contribution));
```

```
create table interpretation(  

```

```

role_joue varchar(50) not null,

langue_interpretation varchar(50) not null,

foreign key (id_acteur) references acteur_doubleur(id_acteur),

foreign key(id_perso) references perso(id_perso));


alter table personnage add prenom_perso varchar(50);

alter table personnage add constraint chk_sexe check (sexe_perso in ('homme', 'femme',
'autre', 'inconnu'));

alter table personnage add constraint chk_statut check (statut_mort_vivant_inconnu in
('mort', 'vivant', 'inconnu'));

alter table oeuvre add genre_oeuvre varchar(50);

alter table oeuvre add constraint chk_genre_oeuvre check (genre_oeuvre in ('comédie',
'horreur', 'romantique', 'shojo', 'shonen', 'action', 'autre'));

alter table oeuvre add constraint chk_type_oeuvre check (type_oeuvre in ('film', 'série',
'jeux vidéo', 'livre', 'bd', 'manga', 'autre'));

alter table utilisateur add age_user int;

alter table utilisateur add colonne_fiable varchar(10);

alter table utilisateur add constraint chk_role_fiable check ( (role_user <> 'modérateur')
or (colonne_fiable = 'oui'));

alter table utilisateur add constraint chk_email check (email_user like '%@%.%');

alter table source add constraint chk_type_source check (type_source in ('site web',
'livre', 'article', 'autre'));

alter table contribution add constraint chk_type_action check (type_action in ('ajout',
'modification', 'suppression'));

alter table interpretation add constraint chk_langue check (langue_interpretation in ('fr',
'en', 'jp','autre'));

```

Les clés primaires correspondent aux id et sont identifiés par ‘primary key’ , sauf si autre chose est précisé (quand c'est un attribut composé) les clés étrangères sont identifiées par les ‘foreign key’, et ont le même nom que les clés primaires auxquelles elles font référence.

Il doit y avoir :

30 trait (traits de personnalité/physique)

70 acteur\_doubleur(acteurs/ doubleurs notables)

50 œuvre(films,livres, séries, jeux vidéo...)

80 personnage (personnages principaux et secondaire. Respecter les contraintes chk\_sexe et chk\_statut)

10 source(sources d'information crédibles, site web, livres, articles...)

10 utilisateur

10 contribution

10 relation (relations entre les personnages, amitié, romantique, famille...)

Propose un panel diversifié d'œuvre et de personnage bien réel.

Fournis l'ensemble sous la forme d'un script SQL prêt à être exécuté.

### Réponse :

Fichier 3\_insertion.sql dans mon répertoire Github.

### Interrogation de la BD

#### 1) Scénario d'utilisation

#### **Scénario 1**

#### **CONSULTATION PAR UN FAN**

Le fan n'est pas enregistré sur le site, il l'utilise régulièrement pour effectuer des recherches et avoir des informations sur ses personnages préférés. Il peut consulter la base pour découvrir des œuvres similaires, des relations entre personnages, ou des acteurs ayant interprété plusieurs rôles et peut filtrer ces requêtes par année, genre, pays...

#### Exemple de requête :

Liste des personnages féminins vivants trié par nom

Sélection d'acteur américains ou français

Œuvres sorties entre 1995 et 2025 trié par genre

Liste des œuvres de type 'film' ou 'série' d'horreur

Personnage d'ont le prénom commence par la lettre 'E'

Personnages apparaissant dans des œuvres d'action

## **Scénario 2**

### **MODERATEUR**

Il a accès à plus de données et peu par exemple accepté les modifications des utilisateurs, savoir le nombre exacte d'éléments par catégorie (analyse statistique). Il peut gérer les utilisateurs et de leurs rôles, a accès à toutes les tables et requêtes, détecter les incohérences ou éléments manquants.

Exemple de requête :

Œuvres sans personnages enregistrés

Personnages ayant plus de traits que la moyenne

Nombre de personnages par statut, seulement si plus de 5 personnages

Moyenne d'âge des utilisateurs par rôle

Nombre d'œuvres par genre

Nombre total de personnages par sexe

## **Scénario 3**

### **CONTRIBUTEUR**

C'est un fan enregistré sur le site, il peut faire les mêmes actions que 'consultation par un fan' et aussi créer/ modifié des fiches des personnages en y mettant les informations avec les sources. Ces informations seront vérifiées plus tard par le Modérateur, il peut voir l'avancé de ses modifications ainsi que les fiches dont il s'est occupé/ S'occupe.

Exemple de requête :

Voir ce qu'il a ajouté ou modifié

Suivi personnel des fiches créées

Personnages joués par des acteurs ayant participé à l'œuvre "Avengers"

Acteurs ayant joué dans au moins une œuvre japonaise

Personnages et leurs acteurs associés

Œuvres et leurs personnages principaux



Relations entre personnages

2) Fichier 4\_interrogation.sql dans mon répertoire Github