**Description textuelle de la base de données (MERISE)**

**🎓 Contexte**

La base de données modélise une **plateforme de formation en ligne**, permettant :

* aux enseignants de créer et diffuser des cours,
* aux apprenants de s’inscrire, suivre les leçons, poser des questions et laisser des avis,
* à l’administrateur de gérer les catégories, notifications ….

**🧩 Entités principales et leurs caractéristiques**

**Utilisateur (User)**

* Représente toute personne inscrite sur la plateforme.
* Peut être enseignant ou apprenant.

**Attributs principaux :**

* ID utilisateur
* Nom d’utilisateur, mot de passe, email (hérité du modèle User)

**Profil (Profile)**

* Informations complémentaires sur l’utilisateur.

**Enseignant (Teacher)**

* Est obligatoirement lié à un utilisateur unique.
* Contient les informations personnelles et professionnelles de l’enseignant.

**Attributs :**

* ID enseignant
* Nom complet
* Biographie courte et description détaillée
* Image de profil
* Liens réseaux sociaux
* Pays

**Associations :**

* Un enseignant peut créer plusieurs cours.

**Catégorie (Category)**

* Classement des cours.

**Attributs :**

* ID catégorie
* Titre
* Image
* Slug
* Actif/Inactif

**Cours (Course)**

* Représente une formation.

**Attributs principaux :**

* ID cours
* Titre et description
* Prix
* Langue, niveau
* Statuts de publication (enseignant + plateforme)
* Image et fichier associés
* Slug et identifiant unique
* Date de création
* Indicateur de cours mis en avant

**Associations :**

* Appartient à une catégorie.
* Créé par un enseignant.
* Contient plusieurs variantes (chapitres).
* Peut être suivi par plusieurs utilisateurs.

**Variante (Variant)**

* Correspond à un **chapitre ou module** du cours.

**Attributs :**

* ID variante
* Titre
* Date de création
* Identifiant unique

**Associations :**

* Contient plusieurs éléments de variante (leçons).

**Élément de variante (VariantItem)**

* Correspond à une **leçon individuelle**.

**Attributs :**

* ID élément
* Titre, description
* Fichier associé (vidéo ou ressource)
* Durée
* Indication si la leçon est en prévisualisation
* Date de création

**Question/Réponse (Question\_Answer)**

* Question posée par un utilisateur sur un cours.

**Attributs :**

* ID question
* Titre
* Date de création

**Associations :**

* Peut contenir plusieurs messages.

**Message de Question/Réponse (Question\_Answer\_Message)**

* Message échangé dans le cadre d’une question.

**Attributs :**

* ID message
* Contenu
* Date

**Inscription à un cours (EnrolledCourse)**

* Inscription d’un utilisateur à un cours.

**Attributs :**

* ID inscription
* Date

**Associations :**

* Lien vers le cours, l’utilisateur et l’enseignant.

**Leçon complétée (CompletedLesson)**

* Représente une leçon terminée par l’utilisateur.

**Note personnelle (Note)**

* Note privée prise par l’apprenant sur un cours.

**Avis (Review)**

* Avis laissé par un utilisateur sur un cours.

**Attributs :**

* Texte de l’avis
* Note sur 5 étoiles
* Réponse éventuelle
* Indicateur actif/inactif
* Date

**Notification**

* Notification liée à un évènement (nouvelle évaluation, nouvelle question, etc.).

**Attributs :**

* Type de notification
* Vue ou non vue
* Date

**1️ Liste de souhaits (Wishlist)**

* Liste des cours que l’utilisateur souhaite suivre.

**Pays (Country)**

* Pays avec taux de taxe associé.

**Attributs :**

* Nom
* Taux de taxe
* Actif/inactif

**🧭 Principales associations**

* **User 1⟷1 Teacher**
* **User 1⟷N EnrolledCourse**
* **User 1⟷N Review**
* **User 1⟷N Question\_Answer**
* **Question\_Answer 1⟷N Question\_Answer\_Message**
* **Teacher 1⟷N Course**
* **Course 1⟷N Variant**
* **Variant 1⟷N VariantItem**
* **Course 1⟷N EnrolledCourse**
* **Course 1⟷N Review**
* **Course 1⟷N Question\_Answer**
* **Course 1⟷N CompletedLesson**

**✨ Comment utiliser ce descriptif avec MERISE ?**

1. **MCD (Modèle Conceptuel des Données)**
   * Représente toutes ces entités avec leurs attributs et identifiants.
   * Ajoute les associations cardinalisées.
2. **MLD (Modèle Logique des Données)**
   * Transforme chaque entité en table relationnelle.
   * Spécifie les clés primaires et étrangères.
3. **MPD (Modèle Physique des Données)**
   * S’appuiera sur Django ORM.
4. MCD – **Notation Merise textuelle (formelle)**

Entitiés

ENTITÉ Utilisateur

id\_user : Identifiant

username : Texte

email : Texte

ENTITÉ Profil

id\_profile : Identifiant

// (autres attributs)

ENTITÉ Enseignant

id\_teacher : Identifiant

nom\_complet : Texte

bio : Texte

about : Texte

image : Texte

pays : Texte

facebook : Texte

twitter : Texte

linkedin : Texte

ENTITÉ Catégorie

id\_category : Identifiant

titre : Texte

image : Texte

slug : Texte

actif : Booléen

ENTITÉ Cours

id\_course : Identifiant

titre : Texte

description : Texte

prix : Décimal

langue : Texte

niveau : Texte

statut\_plateforme : Texte

statut\_teacher : Texte

image : Texte

file : Texte

slug : Texte

course\_id : Texte

date\_creation : Date

featured : Booléen

ENTITÉ Variante

id\_variant : Identifiant

titre : Texte

variant\_id : Texte

date\_creation : Date

ENTITÉ ÉlémentVariante

id\_variant\_item : Identifiant

titre : Texte

description : Texte

fichier : Texte

durée : Durée

contenu\_durée : Texte

preview : Booléen

variant\_item\_id : Texte

date\_creation : Date

ENTITÉ Inscription

id\_enrollment : Identifiant

enrollment\_id : Texte

date : Date

ENTITÉ LeçonComplétée

id\_completed : Identifiant

date : Date

ENTITÉ QuestionReponse

id\_qa : Identifiant

titre : Texte

qa\_id : Texte

date : Date

ENTITÉ MessageQA

id\_qam : Identifiant

message : Texte

qam\_id : Texte

date : Date

ENTITÉ Avis

id\_review : Identifiant

review : Texte

rating : Entier

reply : Texte

actif : Booléen

date : Date

ENTITÉ Notification

id\_notification : Identifiant

type : Texte

vu : Booléen

date : Date

ENTITÉ ListeSouhaits

id\_wishlist : Identifiant

ENTITÉ Note

id\_note : Identifiant

titre : Texte

note : Texte

note\_id : Texte

date : Date

ENTITÉ Pays

id\_country : Identifiant

nom : Texte

taux\_taxe : Entier

actif : Booléen

**ASSOCIATIONS**

ASSOCIATION Possède

ENTRE Utilisateur (1,1)

ENTRE Profil (0,1)

ASSOCIATION Est\_enseignant

ENTRE Utilisateur (1,1)

ENTRE Enseignant (0,1)

ASSOCIATION Crée

ENTRE Enseignant (1,1)

ENTRE Cours (0,N)

ASSOCIATION Appartient

ENTRE Cours (1,1)

ENTRE Catégorie (1,1)

ASSOCIATION DéclinéEn

ENTRE Cours (1,1)

ENTRE Variante (0,N)

ASSOCIATION Contient

ENTRE Variante (1,1)

ENTRE ÉlémentVariante (0,N)

ASSOCIATION Inscrit

ENTRE Utilisateur (1,1)

ENTRE Cours (1,1)

ENTRE Enseignant (1,1)

ENTRE Inscription (1,1)

ASSOCIATION Complète

ENTRE Utilisateur (1,1)

ENTRE ÉlémentVariante (1,1)

ENTRE Cours (1,1)

ENTRE LeçonComplétée (1,1)

ASSOCIATION Pose

ENTRE Utilisateur (1,1)

ENTRE Cours (1,1)

ENTRE QuestionReponse (1,N)

ASSOCIATION Message

ENTRE QuestionReponse (1,1)

ENTRE Utilisateur (1,1)

ENTRE Cours (1,1)

ENTRE MessageQA (0,N)

ASSOCIATION Commente

ENTRE Utilisateur (1,1)

ENTRE Cours (1,1)

ENTRE Avis (0,N)

ASSOCIATION Notifie

ENTRE Notification (1,1)

ENTRE Utilisateur (0,1)

ENTRE Enseignant (0,1)

ENTRE Avis (0,1)

ASSOCIATION Souhaite

ENTRE Utilisateur (1,1)

ENTRE Cours (1,1)

ENTRE ListeSouhaits (1,1)

ASSOCIATION PrendNote

ENTRE Utilisateur (1,1)

ENTRE Cours (1,1)

ENTRE Note (1,N)

1. MLD – Modèle Logique de Données (tables SQL)

-- Utilisateur

User(id\_user PK, username, email, ...)

-- Profil

Profile(id\_profile PK, id\_user FK)

-- Enseignant

Teacher(id\_teacher PK, id\_user FK, nom\_complet, bio, about, ...)

-- Catégorie

Category(id\_category PK, titre, image, slug, actif)

-- Cours

Course(id\_course PK, id\_teacher FK, id\_category FK, titre, description, prix, langue, niveau, statut\_plateforme, statut\_teacher, ...)

-- Variante

Variant(id\_variant PK, id\_course FK, titre, variant\_id, date\_creation)

-- Élément de Variante

VariantItem(id\_variant\_item PK, id\_variant FK, titre, description, fichier, ...)

-- Inscription

EnrolledCourse(id\_enrollment PK, id\_user FK, id\_course FK, id\_teacher FK, enrollment\_id, date)

-- Leçon Complétée

CompletedLesson(id\_completed PK, id\_user FK, id\_course FK, id\_variant\_item FK, date)

-- Question/Réponse

Question\_Answer(id\_qa PK, id\_user FK, id\_course FK, titre, qa\_id, date)

-- Message Q/R

Question\_Answer\_Message(id\_qam PK, id\_user FK, id\_course FK, id\_question\_qa FK, message, qam\_id, date)

-- Avis

Review(id\_review PK, id\_user FK, id\_course FK, review, rating, reply, actif, date)

-- Notification

Notification(id\_notification PK, id\_user FK?, id\_teacher FK?, id\_review FK?, type, vu, date)

-- Liste de souhaits

Wishlist(id\_wishlist PK, id\_user FK, id\_course FK)

-- Note

Note(id\_note PK, id\_user FK, id\_course FK, titre, note, note\_id, date)

-- Pays

Country(id\_country PK, nom, taux\_taxe, actif)

script SQL (SQLite3)

* User
* Profile
* Teacher
* Category
* Course
* Variant
* VariantItem
* Question\_Answer
* Question\_Answer\_Message
* CompletedLesson
* EnrolledCourse
* Note
* Review
* Notification
* Wishlist
* Country

Avec :

* les **clés primaires**
* les **clés étrangères**
* les contraintes NOT NULL là où approprié
* les DEFAULT quand nécessaire

**MLD (Modèle Logique de Données)**

* Chaque table et ses colonnes
* Les clés primaires en gras
* Les relations (FK) tracées
* Les cardinalités principales

-- ============================

-- SCRIPT SQLITE3 POUR LA BASE

-- ============================

PRAGMA foreign\_keys = ON;

-- Table User

CREATE TABLE User (

id INTEGER PRIMARY KEY AUTOINCREMENT,

username TEXT NOT NULL UNIQUE

-- Ajoute ici les champs de User selon ton modèle auth (email, password, etc.)

);

-- Table Profile

CREATE TABLE Profile (

id INTEGER PRIMARY KEY AUTOINCREMENT,

user\_id INTEGER,

bio TEXT,

FOREIGN KEY(user\_id) REFERENCES User(id)

);

-- Table Teacher

CREATE TABLE Teacher (

id INTEGER PRIMARY KEY AUTOINCREMENT,

user\_id INTEGER NOT NULL,

image TEXT DEFAULT 'default.jpg',

full\_name TEXT NOT NULL,

bio TEXT,

facebook TEXT,

twitter TEXT,

linkedin TEXT,

about TEXT,

country TEXT,

FOREIGN KEY(user\_id) REFERENCES User(id)

);

-- Table Category

CREATE TABLE Category (

id INTEGER PRIMARY KEY AUTOINCREMENT,

title TEXT NOT NULL,

image TEXT DEFAULT 'category.jpg',

active BOOLEAN DEFAULT 1,

slug TEXT UNIQUE

);

-- Table Course

CREATE TABLE Course (

id INTEGER PRIMARY KEY AUTOINCREMENT,

category\_id INTEGER,

teacher\_id INTEGER,

file TEXT,

image TEXT,

title TEXT,

description TEXT,

price REAL DEFAULT 0.0,

language TEXT,

level TEXT,

platform\_status TEXT DEFAULT 'Published',

teacher\_course\_status TEXT DEFAULT 'Published',

featured BOOLEAN DEFAULT 0,

course\_id TEXT UNIQUE,

slug TEXT UNIQUE,

date TEXT DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY(category\_id) REFERENCES Category(id),

FOREIGN KEY(teacher\_id) REFERENCES Teacher(id)

);

-- Table Variant

CREATE TABLE Variant (

id INTEGER PRIMARY KEY AUTOINCREMENT,

course\_id INTEGER NOT NULL,

title TEXT NOT NULL,

variant\_id TEXT UNIQUE,

date TEXT DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY(course\_id) REFERENCES Course(id)

);

-- Table VariantItem

CREATE TABLE VariantItem (

id INTEGER PRIMARY KEY AUTOINCREMENT,

variant\_id INTEGER NOT NULL,

title TEXT NOT NULL,

description TEXT,

file TEXT NOT NULL,

duration INTEGER,

content\_duration TEXT,

preview BOOLEAN DEFAULT 0,

variant\_item\_id TEXT UNIQUE,

date TEXT DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY(variant\_id) REFERENCES Variant(id)

);

-- Table Question\_Answer

CREATE TABLE Question\_Answer (

id INTEGER PRIMARY KEY AUTOINCREMENT,

course\_id INTEGER NOT NULL,

user\_id INTEGER,

title TEXT,

qa\_id TEXT UNIQUE,

date TEXT DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY(course\_id) REFERENCES Course(id),

FOREIGN KEY(user\_id) REFERENCES User(id)

);

-- Table Question\_Answer\_Message

CREATE TABLE Question\_Answer\_Message (

id INTEGER PRIMARY KEY AUTOINCREMENT,

course\_id INTEGER NOT NULL,

question\_id INTEGER NOT NULL,

user\_id INTEGER,

message TEXT,

qam\_id TEXT UNIQUE,

qa\_id TEXT UNIQUE,

date TEXT DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY(course\_id) REFERENCES Course(id),

FOREIGN KEY(question\_id) REFERENCES Question\_Answer(id),

FOREIGN KEY(user\_id) REFERENCES User(id)

);

-- Table CompletedLesson

CREATE TABLE CompletedLesson (

id INTEGER PRIMARY KEY AUTOINCREMENT,

course\_id INTEGER NOT NULL,

user\_id INTEGER,

variant\_item\_id INTEGER NOT NULL,

date TEXT DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY(course\_id) REFERENCES Course(id),

FOREIGN KEY(user\_id) REFERENCES User(id),

FOREIGN KEY(variant\_item\_id) REFERENCES VariantItem(id)

);

-- Table EnrolledCourse

CREATE TABLE EnrolledCourse (

id INTEGER PRIMARY KEY AUTOINCREMENT,

course\_id INTEGER NOT NULL,

user\_id INTEGER,

teacher\_id INTEGER,

enrollment\_id TEXT UNIQUE,

date TEXT DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY(course\_id) REFERENCES Course(id),

FOREIGN KEY(user\_id) REFERENCES User(id),

FOREIGN KEY(teacher\_id) REFERENCES Teacher(id)

);

-- Table Note

CREATE TABLE Note (

id INTEGER PRIMARY KEY AUTOINCREMENT,

user\_id INTEGER,

course\_id INTEGER NOT NULL,

title TEXT,

note TEXT NOT NULL,

note\_id TEXT UNIQUE,

date TEXT DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY(course\_id) REFERENCES Course(id),

FOREIGN KEY(user\_id) REFERENCES User(id)

);

-- Table Review

CREATE TABLE Review (

id INTEGER PRIMARY KEY AUTOINCREMENT,

user\_id INTEGER,

course\_id INTEGER NOT NULL,

review TEXT NOT NULL,

rating INTEGER,

reply TEXT,

active BOOLEAN DEFAULT 0,

date TEXT DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY(course\_id) REFERENCES Course(id),

FOREIGN KEY(user\_id) REFERENCES User(id)

);

-- Table Notification

CREATE TABLE Notification (

id INTEGER PRIMARY KEY AUTOINCREMENT,

user\_id INTEGER,

teacher\_id INTEGER,

review\_id INTEGER,

type TEXT NOT NULL,

seen BOOLEAN DEFAULT 0,

date TEXT DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY(user\_id) REFERENCES User(id),

FOREIGN KEY(teacher\_id) REFERENCES Teacher(id),

FOREIGN KEY(review\_id) REFERENCES Review(id)

);

-- Table Wishlist

CREATE TABLE Wishlist (

id INTEGER PRIMARY KEY AUTOINCREMENT,

user\_id INTEGER,

course\_id INTEGER NOT NULL,

FOREIGN KEY(course\_id) REFERENCES Course(id),

FOREIGN KEY(user\_id) REFERENCES User(id)

);

-- Table Country

CREATE TABLE Country (

id INTEGER PRIMARY KEY AUTOINCREMENT,

name TEXT NOT NULL,

tax\_rate INTEGER DEFAULT 5,

active BOOLEAN DEFAULT 1

);