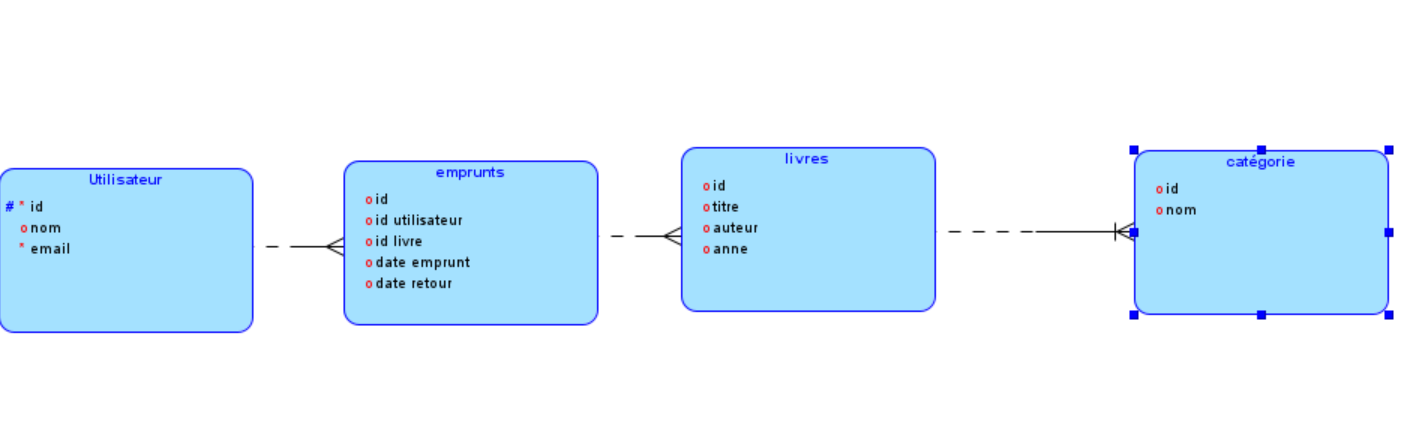
Explication de la nature du projet :

Mon projet consiste à créer un site de gestion de bibliothèque en ligne. Les utilisateurs pourront ajouter, modifier, supprimer et consulter des livres. Les emprunts et retours de livres seront également gérés par le site

Le modèle logique :



Clé de Lecture des Associations du Modèle de Données

## Utilisateurs -> Emprunts

* **Relation** : 1 utilisateur peut avoir plusieurs emprunts.
* **Clé primaire** : **id** (table Utilisateurs)
* **Clé étrangère** : **id\_utilisateur** (table Emprunts)

## Livres -> Emprunts

* **Relation** : 1 livre peut être emprunté plusieurs fois.
* **Clé primaire** : **id** (table Livres)
* **Clé étrangère** : **id\_livre** (table Emprunts)

## Livres -> Catégories

* **Relation** : 1 livre appartient à une catégorie.
* **Clé primaire** : **id** (table Catégories)
* **Clé étrangère** : **id\_categorie** (table Livres)

Croquis des Écrans et Explication

## Page d'Accueil

**Description** : La page d'accueil affiche un tableau de bord contenant un aperçu des différentes sections de l'application. Cette page comprend :

* Un aperçu des utilisateurs (nombre total et récents utilisateurs ajoutés).
* Un aperçu des livres disponibles (nombre total de livres et récemment ajoutés).
* Un aperçu des emprunts en cours (nombre total d'emprunts en cours).
* Un aperçu des catégories de livres (nombre total de catégories).

## Page de Gestion des Utilisateurs

**Description** : La page de gestion des utilisateurs permet de gérer les utilisateurs de la bibliothèque. Cette page comprend :

* Un formulaire pour ajouter un nouvel utilisateur (nom et email).
* Une liste des utilisateurs actuels avec les options suivantes pour chaque utilisateur :
  + Modifier les informations de l'utilisateur.
  + Supprimer l'utilisateur.

## Page de Gestion des Livres

**Description** : La page de gestion des livres permet de gérer les livres de la bibliothèque. Cette page comprend :

* Un formulaire pour ajouter un nouveau livre (titre, auteur, année, catégorie).
* Une liste des livres actuels avec les options suivantes pour chaque livre :
  + Modifier les informations du livre.
  + Supprimer le livre.

## Page de Gestion des Emprunts

**Description** : La page de gestion des emprunts permet de gérer les emprunts et les retours de livres. Cette page comprend :

* Un formulaire pour enregistrer un nouvel emprunt (utilisateur, livre, date d'emprunt).
* Une liste des emprunts en cours avec les options suivantes pour chaque emprunt :
  + Enregistrer le retour du livre (mettre à jour la date de retour).

## Page de Gestion des Catégories

**Description** : La page de gestion des catégories permet de gérer les catégories de livres de la bibliothèque. Cette page comprend :

* Un formulaire pour ajouter une nouvelle catégorie (nom de la catégorie).
* Une liste des catégories actuelles avec les options suivantes pour chaque catégorie :
  + Modifier le nom de la catégorie.
  + Supprimer la catégorie.

Fichiers de Création de la Base de Données et Scripts d’Insertion des Données de Base

## Fichier de Création de la Base de Données (create\_tables.sql)

-- Création de la table Utilisateurs

CREATE TABLE Utilisateurs (

id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

nom VARCHAR2(100) NOT NULL,

email VARCHAR2(100) NOT NULL UNIQUE

);

-- Création de la table Categories

CREATE TABLE Categories (

id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

nom VARCHAR2(100) NOT NULL

);

-- Création de la table Livres

CREATE TABLE Livres (

id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

titre VARCHAR2(255) NOT NULL,

auteur VARCHAR2(255) NOT NULL,

annee NUMBER NOT NULL,

id\_categorie NUMBER,

CONSTRAINT fk\_categorie FOREIGN KEY (id\_categorie) REFERENCES Categories(id)

);

-- Création de la table Emprunts

CREATE TABLE Emprunts (

id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

id\_utilisateur NUMBER,

id\_livre NUMBER,

date\_emprunt DATE,

date\_retour DATE,

CONSTRAINT fk\_utilisateur FOREIGN KEY (id\_utilisateur) REFERENCES Utilisateurs(id),

CONSTRAINT fk\_livre FOREIGN KEY (id\_livre) REFERENCES Livres(id)

);

## Fichier d’Insertion des Données de Base (insert\_data.sql)

-- Insertion de données dans la table Utilisateurs

INSERT INTO Utilisateurs (nom, email) VALUES ('Alice Dupont', 'alice@example.com');

INSERT INTO Utilisateurs (nom, email) VALUES ('Bob Martin', 'bob@example.com');

-- Insertion de données dans la table Categories

INSERT INTO Categories (nom) VALUES ('Science Fiction');

INSERT INTO Categories (nom) VALUES ('Fantasy');

-- Insertion de données dans la table Livres

INSERT INTO Livres (titre, auteur, annee, id\_categorie) VALUES ('Dune', 'Frank Herbert', 1965, 1);

INSERT INTO Livres (titre, auteur, annee, id\_categorie) VALUES ('Le Seigneur des Anneaux', 'J.R.R. Tolkien', 1954, 2);

-- Insertion de données dans la table Emprunts

INSERT INTO Emprunts (id\_utilisateur, id\_livre, date\_emprunt, date\_retour) VALUES (1, 1, TO\_DATE('2024-05-01', 'YYYY-MM-DD'), TO\_DATE('2024-05-15', 'YYYY-MM-DD'));