TP3

Par

Mussallem, Sajed

6262269

Travail présenté à

Mohammed Salim Meflah

Dans le cadre du cours

APPLICATIONS WEB

Groupe 00001

Collège Bois-de-Boulogne

Informatique

26 mai 2024

Table de matières

[Mon projet 3](#_Toc168084854)

[Idée Générale 3](#_Toc168084855)

[Les Pages HTML 3](#_Toc168084856)

[Consignes 3](#_Toc168084857)

[Modèle de données logique et relationnel 4](#_Toc168084858)

[Croquis des écrans 6](#_Toc168084859)

[Script SQL 8](#_Toc168084860)

[Journal d’apprentissage 14](#_Toc168084861)

# Mon projet

## Idée Générale

Mon objectif est de créer et de développer un site web en utilisant les concepts abordés en classe tels que le HTML, le CSS, le JavaScript et le SQL. Ce site permettra aux utilisateurs d'emprunter leurs livres préférés.

## Les Pages HTML

Le site web comprendra une page d'accueil affichant les livres disponibles à l'emprunt, accessible uniquement aux utilisateurs inscrits. Pour s'inscrire, les utilisateurs pourront cliquer sur le bouton "S'inscrire" situé dans le coin supérieur droit de la page d'accueil. Une fois inscrits, ils pourront emprunter des livres. Après avoir sélectionné leurs livres, ils pourront accéder à leur panier en cliquant sur le bouton "Mon Panier", également situé dans le coin supérieur droit de la page d'accueil.

## Consignes

Le projet nécessite de démontrer une compréhension approfondie de la modélisation et de l'implémentation d'une base de données, de l'utilisation de Git pour le contrôle de version, et de la création de pages web interactives connectées à cette base de données. Les consignes demandent la conception d'un site web dynamique comprenant entre 4 et 6 tables dans la base de données, ainsi que 3 à 5 pages HTML. L'utilisation de feuilles de style CSS est obligatoire, et au moins 2 de ces pages doivent inclure du code JavaScript.

# Modèle de données logique et relationnel

A diagram of a computer code

Description automatically generated with medium confidence

A diagram of a computer

Description automatically generated

Le modèle de données logique et relationnel explique les relations entre les entités.

Relation entre Livres et Auteurs:

* Clé étrangère: auteursID dans la table Livres fait référence à id dans la table Auteurs.
* Relation : Un auteur peut avoir écrit plusieurs livres.

Relation entre Emprunts et Livres:

* Clé étrangère : livresID dans la table Emprunts fait référence à id dans la table Livres.
* Relation : Un livre peut être emprunté plusieurs fois.

Relation entre Emprunts et Utilisateurs:

* Clé étrangère: utilisateursID dans la table Emprunts fait référence à id dans la table Utilisateurs.
* Relation : Un utilisateur peut emprunter plusieurs livres.

# Croquis des écrans

***index.html***  
A screenshot of a computer

Description automatically generatedCeci est la page index.html, cette page permet de voir les livres qui sont capables d’être empruntés. En haut, il y a des boutons pour accéder la page d’inscription et la page de ton panier.

***inscription.html***  
A screenshot of a computer

Description automatically generatedCeci est la page inscription.html; c’est cette page qui prend en compte la table Utilisateurs. Ce que je voulais faire, c’est que tu n’as pas le droit d’emprunter un livre sans inscription.

***panier.html***

A screenshot of a computer

Description automatically generated  
Ceci est la page « Mon panier », tu peux voir tous les livres que tu as emprunté. Cette page utilise principalement la table « Emprunts ».

# Script SQL

-- Suppression des tables existantes pour éviter les conflits

DROP TABLE IF EXISTS emprunts CASCADE CONSTRAINTS;

DROP TABLE IF EXISTS livres CASCADE CONSTRAINTS;

DROP TABLE IF EXISTS auteurs CASCADE CONSTRAINTS;

DROP TABLE IF EXISTS utilisateurs CASCADE CONSTRAINTS;

-- Création de la table auteurs

CREATE TABLE auteurs (

id\_auteur NUMBER NOT NULL,

nom VARCHAR2(100) NOT NULL,

date\_naissance DATE,

nationalite VARCHAR2(100),

CONSTRAINT auteurs\_pk PRIMARY KEY (id\_auteur)

);

-- Création de la table livres

CREATE TABLE livres (

id\_livre NUMBER NOT NULL,

titre VARCHAR2(100) NOT NULL,

auteur\_id NUMBER NOT NULL,

genre VARCHAR2(100),

date\_publication DATE,

CONSTRAINT livres\_pk PRIMARY KEY (id\_livre),

CONSTRAINT livres\_fk\_auteur FOREIGN KEY (auteur\_id) REFERENCES auteurs (id\_auteur)

);

-- Création de la table utilisateurs

CREATE TABLE utilisateurs (

id\_utilisateur NUMBER NOT NULL,

nom VARCHAR2(100) NOT NULL,

courriel VARCHAR2(100) NOT NULL,

mot\_de\_passe VARCHAR2(100) NOT NULL,

date\_adhesion DATE DEFAULT SYSDATE,

CONSTRAINT utilisateurs\_pk PRIMARY KEY (id\_utilisateur)

);

-- Création de la table emprunts

CREATE TABLE emprunts (

id\_emprunt NUMBER NOT NULL,

id\_livre NUMBER NOT NULL,

id\_utilisateur NUMBER NOT NULL,

date\_emprunt DATE DEFAULT SYSDATE,

date\_retour DATE,

CONSTRAINT emprunts\_pk PRIMARY KEY (id\_emprunt),

CONSTRAINT emprunts\_fk\_livre FOREIGN KEY (id\_livre) REFERENCES livres (id\_livre),

CONSTRAINT emprunts\_fk\_utilisateur FOREIGN KEY (id\_utilisateur) REFERENCES utilisateurs (id\_utilisateur)

);

-- Insérer des données dans la table auteurs

INSERT INTO auteurs (id\_auteur, nom, date\_naissance, nationalite)

VALUES (1, 'J.K. Rowling', TO\_DATE('1965-07-31', 'YYYY-MM-DD'), 'Britannique');

INSERT INTO auteurs (id\_auteur, nom, date\_naissance, nationalite)

VALUES (2, 'George R.R. Martin', TO\_DATE('1948-09-20', 'YYYY-MM-DD'), 'Américain');

-- Insérer des données dans la table livres

INSERT INTO livres (id\_livre, titre, auteur\_id, genre, date\_publication)

VALUES (1, 'Harry Potter à l\'école des sorciers', 1, 'Fantaisie', TO\_DATE('1997-06-26', 'YYYY-MM-DD'));

INSERT INTO livres (id\_livre, titre, auteur\_id, genre, date\_publication)

VALUES (2, 'Le Trône de fer', 2, 'Fantaisie', TO\_DATE('1996-08-06', 'YYYY-MM-DD'));

-- Insérer des données dans la table utilisateurs

INSERT INTO utilisateurs (id\_utilisateur, nom, courriel, mot\_de\_passe)

VALUES (1, 'John Doe', 'john.doe@example.com', 'motdepasse123');

INSERT INTO utilisateurs (id\_utilisateur, nom, courriel, mot\_de\_passe)

VALUES (2, 'Meryam Lola', 'meryam004@gmail.com', 'abcd123//t');

INSERT INTO utilisateurs (id\_utilisateur, nom, courriel, mot\_de\_passe)

VALUES (3, 'Jana Doha', 'doha004@gmail.com', 'jjj3457//00');

-- Insérer des données dans la table emprunts

INSERT INTO emprunts (id\_emprunt, id\_livre, id\_utilisateur, date\_emprunt, date\_retour)

VALUES (1, 2, 3, TO\_DATE('2023-01-10', 'YYYY-MM-DD'), TO\_DATE('2023-01-20', 'YYYY-MM-DD'));

-- Confirmation des données insérées

SELECT \* FROM auteurs;

SELECT \* FROM livres;

SELECT \* FROM utilisateurs;

SELECT \* FROM emprunts;

BEGIN

ORDS.enable\_schema(

p\_enabled => TRUE,

p\_schema => 'RESTSCOTT',

p\_url\_mapping\_type => 'BASE\_PATH',

p\_url\_mapping\_pattern => 'hr2',

p\_auto\_rest\_auth => FALSE

);

COMMIT;

END;

/

-- Activation de la table auteurs pour accès REST

BEGIN

ORDS.enable\_object (

p\_enabled => TRUE,

p\_schema => 'RESTSCOTT',

p\_object => 'AUTEURS',

p\_object\_type => 'TABLE',

p\_object\_alias => 'auteurs'

);

COMMIT;

END;

/

-- Activation de la table livres pour accès REST

BEGIN

ORDS.enable\_object (

p\_enabled => TRUE,

p\_schema => 'RESTSCOTT',

p\_object => 'LIVRES',

p\_object\_type => 'TABLE',

p\_object\_alias => 'livres'

);

COMMIT;

END;

/

-- Activation de la table utilisateurs pour accès REST

BEGIN

ORDS.enable\_object (

p\_enabled => TRUE,

p\_schema => 'RESTSCOTT',

p\_object => 'UTILISATEURS',

p\_object\_type => 'TABLE',

p\_object\_alias => 'utilisateurs'

);

COMMIT;

END;

/

-- Activation de la table emprunts pour accès REST

BEGIN

ORDS.enable\_object (

p\_enabled => TRUE,

p\_schema => 'RESTSCOTT',

p\_object => 'EMPRUNTS',

p\_object\_type => 'TABLE',

p\_object\_alias => 'emprunts'

);

COMMIT;

END;

/

# Journal d’apprentissage

27 mai

Défi Rencontré: Je n'ai pas beaucoup prêté attention en classe, donc je n'ai pas compris grand-chose au début.

Solution Trouvée: J'ai essayé d'étudier autant que possible.

Apprentissage: La prochaine fois, je devrai prêter plus d'attention en classe.

28 mai

Défi Rencontré: J'avais beaucoup d'examens et de projets à rendre, donc je n'avais pas beaucoup de temps pour étudier.

Solution Trouvée: J'ai essayé de gérer mon temps du mieux possible.

Apprentissage: Je ne devrais pas tout laisser à la dernière minute.

29 mai

Défi Rencontré: J'étais frustré de ne pas pouvoir trouver des idées et j'avais des problèmes de codage.

Solution Trouvée: J'ai consulté des sites internet comme StackOverflow pour trouver des idées.

Apprentissage: Je dois rester calme et réfléchir aux solutions; Internet est une excellente ressource pour cela.

30 mai

Défi Rencontré: Il y avait des choses que je ne comprenais pas bien dans le projet.

Solution Trouvée: J'ai demandé à mes amis de m'aider.

Apprentissage: Je ne devrais pas craindre de demander de l'aide aux autres.

31 mai

Défi Rencontré: Mon plus grand problème était avec le langage SQL; je n'arrivais pas à faire grand-chose. Je n’ai pas pu comprendre les détails plus fins de SQL et de l'intégrer à mon projet.

Solution Trouvée: Je n'ai pas encore réussi à le résoudre, mais je ne vais pas abandonner.

Apprentissage: Ne pas se frustrer et abandonner.