

INTRODUCTION AUX BASES DE DONNÉES AVEC SQL

Enseignant :	Louis RAYNAL	Établissement :	ICES
Contact :	l-raynal@ices.fr	Année :	2025-2026
Public :	L3 Maths		
Ressources :	https://github.com/LouisRaynal/coursSQL		

TP noté 2 (2h)

Nom :

Prénom :

Filière :

Rappels : Toute forme de communication entre vous est interdite. Toute triche résultera en la note de 0.

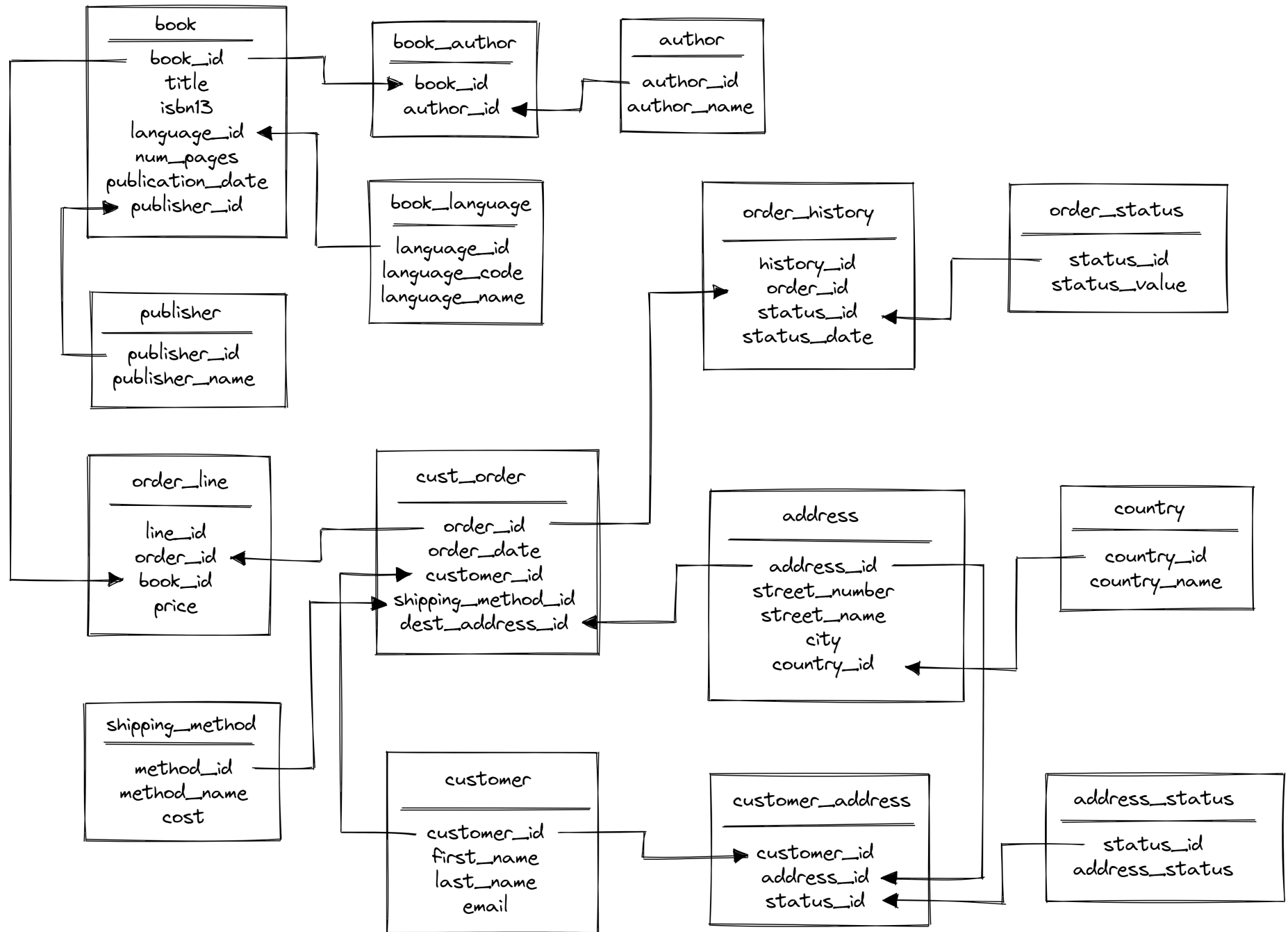
Documents autorisés : Polycopiés et scripts SQL des cours-TP, bases de données utilisées jusqu'à présent, vos notes de cours, fiches mémo, et autres ressources disponibles à l'adresse <https://github.com/LouisRaynal/coursSQL>.

Important : Pour l'exercice ci-dessous, produire un fichier `.sql` comprenant vos requêtes SQL permettant de répondre aux différentes questions posées. Nommez ce fichiers `nom_prénom_exercice_1`, (remplacez `nom` et `prénom` par votre nom et prénom).

A la fin du contrôle, envoyez votre fichier `.sql` par e-mail à l'adresse l-raynal@ices.fr.

Chaque requête doit être exécutable sans modifications de ma part. Ajoutez une ligne de commentaire comme ci-dessous avant chaque requête afin d'indiquer la question concernée.

```
-- Question X
```



Exercice 1

Cet exercice utilise une base de données nommée **bookstore**, stockée dans le fichier **bookstore.db**, téléchargeable depuis le dossier **Examen2** à l'adresse <https://github.com/LouisRaynal/coursSQL>.

En utilisant le navigateur web **Google Chrome**, téléchargez cette base de données **sur votre bureau**, et connectez-vous-y depuis SQLiteStudio.

Le schéma relationnel de cette base de données est représenté en page 2.

Cette base de données correspond aux informations recueillies dans une librairie en ligne. Elle contient les tables suivantes :

- **book** - une liste de tous les livres disponibles dans la librairie ;
- **author** - une liste de tous les auteurs ;
- **book_author** - permet de faire la correspondance entre un livre et son auteur, grâce à un identifiant de livre (**book_id**) et un identifiant d'auteur (**author_id**). Un livre peut avoir plusieurs auteurs, et un même auteur peut avoir écrit plusieurs livres ;
- **book_language** - une liste de toutes les langues possibles de livres ;
- **publisher** - une liste d'éditeurs de livres ;
- **customer** - une liste des clients de la librairie ;
- **address** - une liste des adresses dans le système ;
- **customer_address** - permet de faire le lien entre un client et son adresse, ainsi que le statut de son adresse. Un client peut avoir plus d'une adresse, et une même adresse peut correspondre à plusieurs clients ;
- **address_status** - stocke la nomenclature des statuts des adresses, car une adresse peut être courante (**Active**) ou ancienne (**Inactive**) ;
- **country** - une liste des pays où se trouvent les adresses ;
- **cust_order** - une liste des commandes passées par des clients. Une même commande peut contenir plusieurs livres commandés, c'est la table **order_line** qui détaille le contenu de chaque commande ;
- **order_line** - une liste des livres qui font partie de chaque commande. Le prix d'un livre peut changer d'une commande à l'autre en fonction de diverses promotions ;
- **shipping_method** - une liste des moyens de livraison possibles pour une commande ;
- **order_history** - les historiques pour chaque commande, car une commande peut par exemple être dans un statut *commandé*, *annulé*, *livré*...
- **order_status** - stocke la nomenclature des statuts de commandes.

Questions :

Rédigez des requêtes SQL afin d'accomplir les tâches ci-dessous. Il vous faudra rédiger une requête par question.

1. Affichez les titres des 10 livres les plus anciens selon leurs dates de publication.
2. Affichez le nombre de livres différents dont le titre commence par 'The ' et ne se termine pas par les caractères 'ys'.
3. Certains livres apparaissent plusieurs fois dans la table `book`, parce qu'ils ont été publiés plusieurs fois ou dans des langues différentes. Affichez, pour chaque titre de livre, combien d'exemplaires existent. Utilisez respectivement ***Titre du livre*** et ***Nombre exemplaires*** comme alias de colonne, et ordonnez les résultats par nombre d'exemplaires décroissant.
4. Parmi les livres de la table `book`, affichez les langues différentes de ces livres, ainsi que le code de la langue correspondante. Ordonnez les langues par ordre décroissant grâce au numéro de la colonne correspondante.
5. Affichez toutes les informations sur les livres (table `book`) dont les auteurs sont inconnus.
6. Affichez **tous** les titres de livres, le nombre de pages des livres, les noms d'éditeurs correspondants, les auteurs correspondants lorsqu'ils sont connus, la langue des livres.
7. En utilisant une sous-requête **autonome**, affichez toutes les informations des livres (table `book`) qui n'ont jamais été commandés.
8. En utilisant une sous-requête **corrélée**, affichez les informations sur les clients (table `customer`) ayant une adresse au statut dont l'identifiant est 2.
9. La table `order_history` récapitule l'historique des différents statuts par lesquels une commande est passée. En utilisant une sous-requête **corrélée** ainsi qu'un **EXISTS**, affichez toutes les informations des commandes clients (table `cust_order`), ainsi qu'une colonne supplémentaire qui indiquera 1 si la commande a été livrée (statut 'Delivered') et 0 sinon. Nommez cette colonne ***Statut livré***.
10. De la manière de votre choix, affichez le prénom et le nom des clients ayant acheté un livre en français (**French**) ou en anglais (**English** et ses variantes contenant **English**).