

Bases de données: Examen

Pierre Senellart (pierre.senellart@ens.fr)

27 mai 2025

Les seuls documents autorisés sont 5 feuilles A4 recto-verso (10 pages), avec le contenu de votre choix, lisible à l'œil nu. Cet examen dure deux heures et est constitué d'un unique problème de 10 questions, noté sur 20 points.

On souhaite modéliser le système d'information d'une bibliothèque universitaire. Celle-ci gère un ensemble de *livres*, d'*étudiant·e·s*, d'*emprunts*, et de *réservations*.

Chaque livre a un ou plusieurs auteurs, un titre, un identifiant unique (*cote*), une catégorie (p. ex., roman, manuel, thèse, etc.) et un nombre d'exemplaires. Les exemplaires du même livre partagent la même cote. Les étudiant·e·s (qui ont un nom, une adresse email et un identifiant) peuvent emprunter les livres dans la limite de 5 livres simultanément, et pour une durée maximale de 21 jours par livre. Les étudiant·e·s peuvent aussi *réserver* un livre si tous les exemplaires sont empruntés. Lorsqu'un livre est rendu, le ou la première étudiante en file de réservation est notifié. Chaque emprunt ou réservation est horodaté.

- (2 points) Proposer un schéma entité–association pour ce système d'information, en distinguant les entités, attributs, relations, cardinalités, et contraintes pertinentes.
- (2 points) En déduire un schéma relationnel et l'exprimer à l'aide d'ordres SQL `CREATE TABLE`, en indiquant les contraintes suivantes : `PRIMARY KEY`, `UNIQUE`, `NOT NULL`, `CHECK`, `FOREIGN KEY`.
- (1 point) Identifier une contrainte non exprimable en SQL standard sur ce schéma. Exprimer cette contrainte avec une formule de la logique du premier ordre.
- (3 points) Écrire une requête en algèbre relationnelle, en calcul relationnel, et en SQL qui retourne la liste des adresses emails d'étudiant·e·s ayant réservé un livre qui est actuellement disponible (c'est-à-dire, qu'il en y a au moins un exemplaire non emprunté), avec l'identifiant et le titre de ce livre.
- (2 points) Présenter deux plans d'exécution possibles pour cette requête en commentant de leurs performances en fonction de caractéristiques des tables impliquées dans la requête.
- (2 points) Pour construire un système de recommandation de livres, on considère le graphe de *filtrage collaboratif* dans lequel les sommets sont des livres et il y a une arête entre deux livres si le ou la même étudiante l'a emprunté. Exprimer en Datalog sur le schéma que vous avez défini une requête qui indique si deux livres sont dans la même composante connexe de ce graphe.
- (3 points) On souhaite conserver l'intégralité de l'historique des emprunts et des livres disponibles afin de pouvoir faire des requêtes sur la base telle qu'elle était à un moment donné du passé. Proposer une approche pour ce faire basée sur de la provenance par semi-anneaux, en précisant dans quel semi-anneau se placer, en justifiant qu'il s'agit bien d'un semi-anneau, et en expliquant ce que cette provenance calcule.
- (2 points) Le service informatique souhaite exposer en ligne la liste des livres disponibles, sans afficher les historiques d'emprunt ni les noms des étudiants. Proposer une solution SQL (vues, rôles, permissions) pour garantir cette ségrégation d'accès.

9. (1 point) Un-e étudiant-e développe une application de consultation de ses propres emprunts. Discuter d'un problème de contrôle d'accès mal implémenté qui pourrait permettre à un-e étudiant-e de voir les emprunts d'un autre, et expliquer comment l'éviter (au niveau applicatif ou en base).
10. (2 points) Discuter de comment ajouter un système de sanction pour les étudiant-e-s n'ayant pas rendu un livre emprunté à temps : quelles informations ajouter à la base et comment mettre en place et automatiser la sanction ?