

Feuille de travaux pratiques n° 1

Modèle relationnel, algèbre relationnelle, premiers pas avec DBBrowser for SQLite

Certaines questions sont plutôt de type TD et peuvent être rédigées sur une feuille de papier.

Exercice 1.1 — Une petite librairie

Une librairie scientifique vend quelques livres. Pour modéliser la table **LIVRE**, on dégage la relation suivante :

ISBN	Titre	Auteur1	Auteur2	Prix
0521826462	Modern Computer Algebra	J. von zur Gathen	J. Gerhard	45
2100051164	Le langage de programmation C	D. Ritchie	B. Kernighan	20
2705612963	Corps locaux	J.-P. Serre		28
2705655891	Théorie Algébrique des Nombres	P. Samuel		28
3319902326	Number Fields	D. Marcus		40

1. Quels sont les attributs de cette relation ?
2. Préciser le domaine de chaque attribut.
3. Donner un exemple de n -uplet (**tuple**) pour cette relation.
4. La saisie de données est-elle obligatoire pour chaque attribut ?
5. Peut-on avoir deux lignes identiques ? Proposer une clé primaire pour cette relation.

Exercice 1.2 — Fichier clients

Une entreprise souhaite maintenir une base de données de ses clients. Elle dispose de leur nom, prénom, adresse, éventuellement numéro de téléphone.

1. Quels sont les attributs de la relation **CLIENT** ?
2. Proposer des types de données pour ces attributs.
3. Les attributs permettent-ils de trouver une clé candidate ? Justifier.

On ajoute comme attribut le numéro de client dans la relation, c'est la clé primaire. Pour mieux gérer l'adresse, on crée une relation **ADRESSE** dont les attributs sont les suivants : identifiant, numéro, rue, code postal, ville.

4. À quoi l'attribut *identifiant* sert-il ?
5. Quelles sont les lignes des tables **CLIENT** et **ADRESSE** correspondant à la donnée suivante ?
Louis Pernaut, 0836656565, 23 rue Monsarrat, 33800 Bordeaux.
6. Donner la clé étrangère (ou **externe**).

Exercice 1.3 — Gestion universitaire

On souhaite créer une base de données qui comporte des informations sur l'enseignement en M1 IM. On crée les relations **ÉTUDIANT**, **UE**, **ÉTUDIANT_UE**, **ADRESSE**, **ENSEIGNANT**. Les relations comportent les attributs suivants :

- **ÉTUDIANT** : numéro, nom, prénom, adresse
- **UE** : libellé, code, nombre d'heures associées (cours, TD, TP confondus).
- **ÉTUDIANT_UE** : numéro étudiant, codeUE, la note de CC obtenue, la note d'examen obtenue.
- **ADRESSE** : identifiant adresse, numéro de la rue, nom de la rue, complément ('b' pour bis, 't' pour ter, ' ' si aucun des deux), code postal, nom de la ville.

— **ENSEIGNANT** : numéro, nom, prénom, âge, adresse.

1. Donner les types des données qui vous semblent les plus appropriés.
2. Spécifier si la saisie de données est obligatoire ou non pour un attribut (NULL / NOT NULL).
3. Préciser la clé primaire pour chaque relation.
4. Donner les clés étrangères.

Exercice 1.4 — Algèbre relationnelle : clubs d'échecs

On considère la base de données des clubs d'échecs de la ville de Nice : chaque club est identifié par un numéro, on donne son nom. Pour les joueurs, on enregistre leur numéro d'inscription, ainsi que les nom, prénom, leur club, classement ELO et leur catégorie. Chaque catégorie est identifiée par un numéro et un descriptif (minime, sénior, vétéran, etc.). Pour aider les clubs, la ville de Nice leur prête des pendules pour les tournois. Chaque pendule est identifiée par un numéro et un nom de modèle, elle est accessible à un ou plusieurs clubs.

JOUEUR(Num, Nom, Prénom, ELO, NoCat, NoClub)

CATÉGORIE(NoCat, NomCat)

CLUB(NoClub, NomClub)

PENDULE(NoPend, Modèle)

PRÊT(NoPend, NoClub)

Les données sont les suivantes.

JOUEUR						PRÊT	
Num	Nom	Prénom	NoCat	ELO	NoClub	NoPend	NoClub
1	Allaire	Christabel	2	2100	1	1	1
2	Leblanc	Cloridan	2	1700	1	1	2
3	Beauchemin	Patience	1	1900	2	1	3
4	Flamand	Paulette	1	2200	3	2	1
						2	2
						3	2
						3	3

PENDULE		CLUB		CATÉGORIE	
NoPend	Modèle	NoClub	NomClub	NoCat	NomCat
1	Garde	1	Échiquier Niçois	1	Minimes
2	Garde	2	Club Alekhine	2	Sénior
3	DGT 3000	3	Riviera Chess Club	3	Vétéran

1. Écrire les requêtes correspondantes en algèbre relationnelle et répondre aux questions.
 - (a) Quels sont les joueurs du 1^{er} club ?
 - (b) Quelle est la liste des numéros de catégories avec au moins un joueur ?
 - (c) Afficher toutes les informations du « Club Alekhine » puis uniquement son numéro.
 - (d) Quels sont les numéros des pendules de modèle « Garde » ?
 - (e) Quels sont les prénoms des joueurs de classement ELO supérieur à 2000 ?
2. Traduire en français la requête suivante :
$$\pi_{\text{Nom, Prénom}}(\sigma(\text{NoClub}=2 \text{ et } \text{ELO} > 1800) \text{ ou } (\text{NoClub}=1 \text{ et } \text{ELO} > 2000))(\text{JOUEUR})$$
3. Quelle est la relation JOUEUR \times CATÉGORIE ? Combien a-t-elle de lignes ?
4. Quelle est la jointure naturelle des relations JOUEUR et CATÉGORIE ? Comment peut-on obtenir cette jointure à partir d'un produit cartésien et d'une sélection ?
5. Peut-on calculer l'intersection de JOUEUR et CLUB ? de PRÊT et CLUB ? Expliquer.
6.
 - (a) Afficher une table de tous les joueurs avec leur numéro, nom, prénom, numéro de catégorie, ELO, numéro de club et nom du club.
 - (b) Afficher une table de tous les joueurs avec les 3 colonnes suivantes : nom, prénom et nom de leur club.
 - (c) Afficher une table de tous les joueurs avec les 3 colonnes suivantes : nom, prénom et nom de leur club, en renommant cette dernière colonne « Club ».

- (d) Afficher une table de tous les joueurs avec 5 colonnes : nom, prénom, ELO, nom du club et nom de la catégorie.
7. Donner les noms des clubs qui comptent parmi leurs membres un joueur ayant un classement ELO strictement supérieur à 2000.
8. Donner les prénoms des joueurs de catégorie « Sénior ».
9. Donner les noms des joueurs qui jouent pour un club qui peut emprunter une pendule de modèle « DGT 3000 ».
10. Traduire en français la requête suivante : $\text{PENDULE} \leftarrow \sigma_{\text{Modèle}='Garde'}(\text{PENDULE})$
11. Quelles sont les catégories de joueurs sans aucun joueur ?

Exercice 1.5 — Utilisation de DBML

Pour cet exercice, vous allez utiliser DBML pour obtenir une représentation graphique de vos relations.

1. Allez sur <https://github.com/> pour créer un compte.
2. Le langage DBML a été créé pour définir et documenter des bases de données de façon simple. Il permet aussi de générer des documentations. Allez sur <https://www.dbml.org/home/#intro> pour lire la documentation de base. Vous pouvez aussi consulter <https://www.dbml.org/docs/> pour plus de détails.
3. Allez sur <https://dbdiagram.io/d>, connectez-vous avec votre compte github et créez une nouvelle base de données.
4. Entrez la base de données suivante au format DBML :
 - **ÉTUDIANT** : numéro, nom, prénom, adresse
 - **UE** : libellé, code, nombre d'heures associées (cours, TD, TP confondus).
 - **ÉTUDIANT_UE** : numéro étudiant, codeUE, la note de CC obtenue, la note d'examen obtenue.
 - **ADRESSE** : identifiant adresse, numéro de la rue, nom de la rue, complément ('b' pour bis, 't' pour ter, ' ' si aucun des deux), code postal, nom de la ville.
 - **ENSEIGNANT** : numéro, nom, prénom, âge, adresse.
5. Sauvegardez le fichier DBML obtenu.

Exercice 1.6 — Utilisation de DBBrowser for SQLite

On va ici s'intéresser à la création d'une base de données en utilisant un SGBD (DB Browser for SQLite). Cela nous permettra en particulier d'expérimenter la création d'une base de données relationnelle en utilisant l'interface graphique.

1. Prise en main.

Lancez **DB Browser for SQLite**. S'il n'est pas installé, vous pouvez le trouver à l'adresse suivante : <https://sqlitebrowser.org/dl/>.

Vocabulaire.

L'interface utilise le terme de **table** pour désigner une relation, le terme de **champ (field)** pour désigner un attribut, le terme d'**enregistrement** pour désigner un uplet et le terme de **relation** pour désigner une liaison entre tables.

Note.

Pour éviter d'écrire des capitales tout le temps, avec SQLite, on ne va pas écrire le nom des tables en capitales.

2. Création du schéma.

On souhaite créer une base de données qui comporte des informations sur l'enseignement en informatique. On crée les relations **Étudiant**, **UE**, **ÉtudiantUE**, **Adresse**, **Enseignant**. Les relations comportent les attributs suivants :

- **Étudiant** : numéro étudiant (clé primaire), nom, prénom.
- **UE** : code, libellé, nombre d'heures associées (cours, TD, TP confondus).
- **ÉtudiantUE** : la note de CC obtenue, la note d'examen obtenue.

- **Adresse** : numéro de la rue, complément ('b' pour bis, 't' pour ter, ' ' si aucun des deux), nom de la rue, code postal, nom de la ville. On pourra donner un **numéro à incrément automatique** à chaque adresse qui servira de clé primaire.
- **Enseignant** : numéro (clé primaire), nom, prénom, nombre d'heures enseignées, âge, ville de résidence.

3. Clés étrangères.

On veut maintenant compléter la description de ces relations en introduisant les informations qui nécessitent de lier ces relations entre elles. Pour ce faire, on peut ajouter des attributs à certaines relations et double-cliquer sur la partie **Clé étrangère** : on sélectionne alors la relation et l'attribut qui correspondent.

(a) **Étudiant**. Il faut

- lier la relation **Étudiant** avec la relation **Adresse** afin d'exprimer le fait que tout étudiant a une adresse.
- lier la relation **Étudiant** et la relation **ÉtudiantUE** afin de préciser les UE avec leurs notes qui sont suivies par les étudiants. Eventuellement, un étudiant peut ne suivre aucune UE¹. *Avant de créer une clé étrangère dans une relation il faut qu'elle ait sa clé primaire définie.*

(b) **Enseignant**. Il faut lier la relation **Enseignant** avec la relation **UE** afin d'exprimer le fait que chaque enseignant peut être responsable d'UE (éventuellement, un enseignant peut être responsable d'aucune UE)². À propos de la liaison de la relation **UE** avec la relation **Enseignant**, afin d'exprimer le fait que chaque UE a un enseignant responsable, que pouvez-vous dire ?

(c) **ÉtudiantUE**. Il faut lier la relation **ÉtudiantUE** avec la relation **UE** afin de pouvoir faire le lien entre les notes et l'UE concernée. Avant de créer une clé étrangère dans une relation il faut qu'elle ait sa clé primaire définie. Définir la clé primaire qui vous semble raisonnable (vous ne choisirez pas ici de numéroter les uplets comme on a pu le faire plus haut). À propos de la liaison de la relation **ÉtudiantUE** avec la relation **Étudiant**, afin d'exprimer le fait que les notes sont obtenues par un certain étudiant, que pouvez-vous dire ?

4. Insertion de données. Pour chaque table vous cliquerez sur le bouton ou l'onglet **parcourir les données**, puis **nouvel enregistrement**. L'objectif est de créer les uplets (lignes). **A priori**, il faut opérer dans un ordre logique³ : **Adresse**, **Étudiant**, **Enseignant**, **UE**, **ÉtudiantUE**. Saisir les uplets suivants :

- Adresse(1, 3, 'b', 'Avenue Jean Médecin', '06000', 'Nice')
- Adresse(2, 10, ' ', 'Rue Barla', '06000', 'Nice')
- Adresse(3, 10, ' ', 'Avenue Jean Jaurès', '06200', 'Cagnes-sur-Mer')
- Étudiant(1001, 'Nom1', 'prénom1', 1)
- Étudiant(1002, 'Nom2', 'prénom2', 2)
- Étudiant(1003, 'Nom3', 'prénom3', 3)
- Enseignant(1, 'Diel', 'Roland', 192, 30, 'Nice')
- Enseignant(2, 'Pasquier', 'Nicolas', 192, 39, 'Antibes')
- Enseignant(3, 'Pelleau', 'Marie', 250, 26, 'Nice')
- Enseignant(4, 'Lezowski', 'Pierre', 384, 35, 'Nice')
- Enseignant(5, 'Tichit', 'Laurent', 96, 34, 'Nice')
- UE('SMEMI103', 'Base de Données', 24)
- UE('SMEMI102', 'Analyse de données', 36)
- UE('SMEMI101', 'Séries temporelles', 32)
- UE('SMUMA102', 'EDP et différences finies', 26)
- ÉtudiantUE(1001, 'SMEMI103', 10, 11)
- ÉtudiantUE(1001, 'SMEMI102', 8, 10)
- ÉtudiantUE(1002, 'SMEMI103', 10, 11)
- ÉtudiantUE(1002, 'SMEMI102', 8, 10)
- ÉtudiantUE(1003, 'SMEMI103', 10, 11)

1. En pratique, le nombre d'UE possibles est limité, mais on ne s'en préoccupe pas.

2. Même remarque concernant le nombre que pour la relation **Étudiant**.

3. Expliquez pourquoi cet ordre est logique.

- ÉtudiantUE(1003, 'SMEMI102', 8, 10)
 - ÉtudiantUE(1003, 'SMUMA102', 12, 13)
5. Que se passe-t-il quand vous essayez d'insérer les uplets suivants ? Expliquer.
- Étudiant(1005, 'Nom1', 'Prénom2', 4)
 - Étudiant(1001, 'Nom2', 'Prénom2', 1)
6. **Consultation et sélection simple de données.** Vous pouvez afficher des données des tables créées. En cliquant avec le bouton droit sur une colonne, on peut masquer/afficher une colonne. Afficher les informations suivantes ;
- (a) Numéro et nom de la rue et ville, pour toutes les adresses
 - (b) Nom et prénom de tous les étudiants
 - (c) Le libellé de l'UE et le numéro de l'enseignant responsable pour toutes les UE.
 - (d) code postal et ville, pour toutes les adresses (par ordre croissant de code postal).
- Premières sélections :
- (a) Affichage de tous les étudiants dont le nom est « Nom1 » (toutes les informations)
 - (b) Affichage de tous les numéros d'étudiants qui suivent l'UE « SMIEMI103 »
 - (c) Afficher tous les enseignants qui habitent à « Nice ».
 - (d) Afficher tous les noms de rues de la ville « Nice »
7. **Sélection plus complexe.**
- C'est le moment d'utiliser de manière plus fine les possibilités de SQLite Browser.
- (a) Afficher toutes les UE qui ont strictement plus de 200 heures de cours.
 - (b) Afficher toutes les UE qui ont un nombre d'heures de cours supérieur ou égal à 26.
 - (c) Afficher toutes les UE qui ont un nombre d'heures de cours supérieur ou égal à 26 et dont le nom commence par SMU.
 - (d) Afficher toutes les adresses qui sont dans des avenues.
- Évidemment, l'objectif sera aussi de savoir comment regrouper les informations des différentes relations ; nous verrons comment faire dans la suite du cours.
8. **Modification de données.**
- Utiliser l'onglet **parcourir les données** afin de réaliser les opérations suivantes (ligne par ligne).
- (a) Augmenter toutes les notes d'examen de 2 points pour tous les étudiants et toutes les UE.
 - (b) Pour les étudiants qui suivent l'UE **SL2IBD**, augmenter la note de contrôle continu de 2 points.
9. **Modification du schéma.**
- (a) Modifier la colonne des horaires de la table **Enseignant** pour gérer les chiffres à virgule.
 - (b) Ajouter une colonne « âge » à la table **Étudiant** et compléter son contenu.
 - (c) Supprimer la colonne « âge » de la table **Étudiant**.

Exercice 1.7 — Joueuses de tennis

1. Téléchargez le fichier `wt2021.db` qui contient une base de données des résultats des matches du circuit féminin WTA en 2021, puis ouvrez-le dans **DBBrowser for SQLite**.
2. Utilisez DBML pour décrire les données correspondant à cette base de données.
3. À l'aide de l'onglet **Parcourir les données** de **DBBrowser for SQLite**, répondez aux questions suivantes.
 - (a) Combien de tournois différents sont-ils répertoriés dans la table ?
 - (b) Quel était le premier tournoi de l'année 2021 ?
 - (c) Combien de tournois se sont-ils déroulés en mars 2021 ?
 - (d) Combien de joueuses différentes s'appellent-elles Ostapenko dans la base de données ?
 - (e) Quelle est la plus jeune joueuse française à figurer dans la table ?
 - (f) Combien de joueuses sont-elles nées un 1^{er} avril ?
 - (g) Quel était le classement de Jelena Ostapenko en début d'année ?
 - (h) Combien Ashleigh Barty a-t-elle remporté de tournois en 2021 ?
 - (i) Quelles joueuses ont-elles remporté un tournoi du Grand Chelem en 2021 ?