

## Feuille de travaux pratiques n° 2

### Conception et premières requêtes SQL

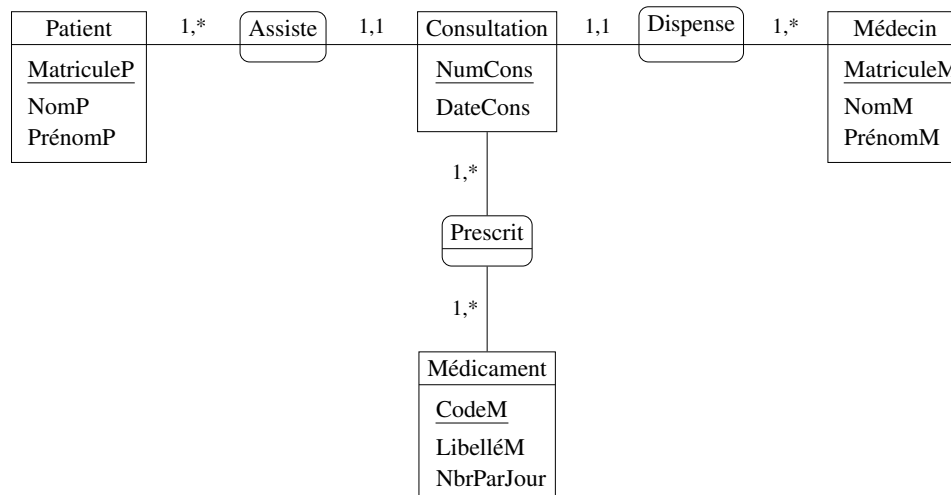
#### Exercice 2.1 — Utilisation de DBML

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 2.2 — Médecins

Le schéma Entités-Associations suivant décrit les consultations de patients auprès de médecins qui prescrivent des médicaments.



Le dictionnaire des données ci-dessous permet de connaître la signification de chaque attribut du schéma Entités-Associations.

Attribut	Type	Description
MatriculeP	integer	Numéro identifiant un patient
NomP	varchar(20)	Nom du patient
PrénomP	varchar(20)	Prénom de l'abonné
NumCons	integer	Numéro de consultation
DateCons	date	Date de la consultation
MatriculeM	integer	Numéro identifiant un médecin
NomM	varchar(20)	Nom du médecin
PrénomM	varchar(20)	Prénom du médecin
CodeM	char(5)	Code identifiant un médicament
LibelléM	varchar(20)	Nom du médicament
NbrParJour	smallint	Nombre de prises par jour du médicament

En fonction de ce schéma, répondez aux questions suivantes.

1. Un patient peut-il effectuer plusieurs visites ?
2. Un médecin peut-il recevoir plusieurs patients dans une même consultation ?
3. Peut-on prescrire plusieurs médicaments dans une même consultation ?
4. Une consultation peut-elle être dispensée par deux médecins différents ?
5. Un même médicament peut-il être prescrit lors de plusieurs consultations ?
6. Le nombre de prises par jour d'un médicament est-il toujours le même ?
7. Traduisez ce schéma Entités-Associations en schéma relationnel.

On souhaite faire évoluer la BD selon les spécifications suivantes :

- Plusieurs médecins peuvent intervenir au cours d'une même consultation.
  - Le nombre de prises par jour d'un médicament peut varier d'une consultation à l'autre. Il dépend donc à la fois du médicament et de la consultation.
8. Modifiez le schéma E-A afin qu'il permette de représenter ces nouvelles spécifications.
  9. Traduisez votre nouveau schéma E-A en schéma relationnel.

### Exercice 2.3 — 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 UE<sup>1</sup>. *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)<sup>2</sup>. 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 numéroté 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 logique<sup>3</sup> : **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)

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(1002, 'SMEMI103', 10, 11)
  - ÉtudiantUE(1002, 'SMEMI102', 8, 10)
  - ÉtudiantUE(1003, 'SMEMI103', 10, 11)
  - 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 2.4 — Joueuses de tennis

1. Téléchargez le fichier `wt_a2021.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) Afficher tous les noms de tournois.
  - (b) Trier les tournois par date, du dernier au premier.
  - (c) Afficher tous les tournois de mars 2021.
  - (d) Afficher toutes les joueuses dont le nom est Ostapenko.

- (e) Trier les joueuses français figurant dans la table, de la plus jeune à la plus ancienne.
  - (f) Afficher toutes les joueuses nées un 1<sup>er</sup> avril.
4. Reprendre les questions précédentes et écrire les requêtes SQL correspondantes.

### Exercice 2.5 — La table des prénoms

Téléchargez l'archive `prenoms.zip` sur Moodle, puis extrayez le fichier `prenoms.db`. Lancez ensuite **DB Browser for SQLite**, puis ouvrez la base de données `prenoms.db`.

#### Présentation

Cette base de données contient une partie de la base de données des prénoms fournie par l'INSEE. Pour chaque prénom, il est indiqué pour chaque année de naissance (de 2000 à 2020) et chaque sexe, le nombre de personnes inscrites à l'état civil sous ce prénom. Pour la table « `prenoms` », la précision est apportée pour chaque département.

#### Les personnes prises en compte

Le champ couvre l'ensemble des personnes nées en France hors Mayotte et enregistrées à l'état civil sur les bulletins de naissance. Les personnes nées à l'étranger sont exclues.

#### Le champ des prénoms retenus

Dans les fichiers de l'état civil, en l'occurrence les bulletins de naissance, les différents prénoms sont séparés par une espace (ou blanc). Ainsi deux prénoms séparés par un tiret constituent un seul prénom composé (exemple : Anne-Laure). Le premier prénom simple ou composé figure en début de liste, et c'est celui qui sera retenu après le traitement de la protection de l'anonymat.

#### Conditions portant sur les prénoms retenus

1. Sur la période allant de 1900 à 1945, le prénom a été attribué au moins 20 fois à des personnes de sexe féminin et/ou au moins 20 fois à des personnes de sexe masculin
2. Sur la période allant de 1946 à 2020, le prénom a été attribué au moins 20 fois à des personnes de sexe féminin et/ou au moins 20 fois à des personnes de sexe masculin
3. Pour une année de naissance donnée, le prénom a été attribué au moins 3 fois à des personnes de sexe féminin ou de sexe masculin

Les effectifs des prénoms ne remplissant pas les conditions 1 et 2 sont regroupés (pour chaque sexe et chaque année de naissance) dans un enregistrement dont le champ prénom (`PREUSUEL`) prend la valeur spéciale `_PRENOMS_RARES`.

#### Documentation (fournie par l'INSEE)

Cette relation prénoms comporte cinq attributs : `sexe`, `preusuel`, `annais`, `dpt`, `nombre`.

Nom	Intitulé	Type	Longueur	Modalité
<code>sexe</code>	sexe	entier	1	1 pour masculin, 2 pour féminin
<code>preusuel</code>	premier prénom	chaîne de caractères	25	
<code>annais</code>	année de naissance	entier	4	2000 à 2020
<code>dpt</code>	département de naissance	entier	3	liste des départements
<code>nombre</code>	fréquence	entier	8	

Vous allez exécuter des requêtes SQL (à l'aide de l'onglet **Exécuter le SQL** pour répondre aux questions suivantes.

**Conseil :** sauvegardez la liste de vos requêtes SQL dans un fichier `.sql`. Vous pouvez inclure des commentaires dans le fichier (lignes commençant par : `--`).

**Attention :** tous les prénoms sont en capitale dans la table.

## Requêtes SQL

**Vous pouvez taper vos requêtes à la suite dans l'onglet « Exécuter le SQL ». Le résultat est celui de la dernière requête. À l'issue du TP, pensez à sauvegarder le fichier SQL**

1. Liste des prénoms de la table.
2. Liste des années de la table.
3. Département et année de naissance des personnes s'appelant Laetitia.
4. Département et année de naissance des garçons prénommés Camille.
5. Liste des prénoms de garçons donnés en Corse en l'an 2000.
6. Liste des prénoms donnés plus de 400 fois dans un département une année donnée, on affichera prénom, département, année.
7. Liste des prénoms donnés strictement moins de 5 fois une année, avec l'année correspondante, dans les Alpes-Maritimes.
8. Liste des prénoms de filles donnés strictement plus de 10 fois entre 2000 et 2004 dans les Alpes-de-Haute-Provence. Proposez deux requêtes.
9. Liste des prénoms donnés exactement 80 fois dans un département dont le numéro est strictement supérieur à 80 en 2005.
10. Liste des prénoms de garçons commençant par Y donnés en 2000 dans les Alpes-Maritimes. Afficher prénom et nombre.
11. Liste des prénoms commençant par « KHL ». Afficher les prénoms, départements et années.
12. Liste des prénoms composés communs entre 2000 et 2003. Afficher les prénoms, départements, années et effectifs.

## Options d'affichage

13. Liste des prénoms avec toutes les informations en renommant les colonnes : prénom, année, département, sexe, effectif.
14. Afficher la liste des prénoms donnés en 2001 en Corse en triant par effectif croissant.
15. Afficher la liste des prénoms donnés en 2001 en Corse en triant par effectif décroissant.
16. Afficher la liste des prénoms distincts de filles qui apparaissent dans la table.
17. Afficher la liste des prénoms distincts commençant par « Al » et terminant par « ce » apparaissant dans la table.
18. Liste des prénoms de filles commençant par « C » donnés en 2000 en Corse. Afficher les prénoms et le nombre, trier par ordre croissant de nombre.
19. Liste des prénoms (distincts) communs donnés entre 2000 et 2020 qui ont deux caractères quelconques, puis terminent par « ice ».
20. Liste des prénoms (distincts) communs entre 2000 et 2020 et qui comportent exactement 6 lettres. Afficher le prénom uniquement.
21. Liste des prénoms (distincts) communs entre 2000 et 2020 et qui comportent exactement 7 lettres, dont un Z. Afficher le prénom uniquement.
22. Liste des prénoms (distincts) non composés communs entre 2000 et 2020 qui commencent par « Ab » et contiennent au moins deux fois la lettre A. Afficher le prénom, le département, l'année et l'effectif. Trié par année décroissante, par effectif croissant, puis par prénom dans l'ordre alphabétique.
23. Afficher la liste des prénoms distincts communs (pas rares) donnés entre 2000 et 2020.
24. Afficher les prénoms communs (non rares) donnés à des garçons en région Provence-Alpes-Côte-d'Azur entre 2002 et 2006. On affichera prénom et nombre, on triera par ordre décroissant de nombre.

## Remarque

Ce fichier est en fait un sous-ensemble de la table fournie par l'INSEE qui recense les prénoms de 1900 à 2020. Sur le site <https://www.insee.fr/fr/statistiques/2540004>, vous pouvez trouver ces données pour la France entière ou par département, sous forme de fichiers csv.

Ce fichier csv peut être importé dans **DB Browser for SQLite**.

## Exercice 2.6 — La base « comparateur de territoires »

Téléchargez `commune.db` sur Moodle. Ce fichier contient une unique table : `comparateur`, qui présente une trentaine d'indicateurs décrivant la population, les logements, les revenus, l'emploi et les établissements au niveau communal.

### Dictionnaire de variables fourni par l'INSEE

- CODGEO : Code du département suivi du numéro de commune ou du numéro d'arrondissement municipal
- LIBGEO : Libellé de la commune ou de l'arrondissement municipal pour Paris, Lyon et Marseille
- REG : Région
- DEP : Département
- P16\_POP : Population en 2016
- P11\_POP : Population en 2011
- SUPERF : Superficie (en km<sup>2</sup>)
- NAIS1116 : Nombre de naissances entre le 01/01/2011 et le 01/01/2016
- DECE1116 : Nombre de décès entre le 01/01/2011 et le 01/01/2016
- P16\_MEN : Nombre de ménages en 2016
- NAISD18 : Nombre de naissances domiciliées en 2018
- DECESD18 : Nombre de décès domiciliés en 2018
- P16\_LOG : Nombre de logements en 2016
- P16\_RP : Nombre de résidences principales en 2016
- P16\_RSECOCC : Nombre de résidences secondaires et logements occasionnels en 2016
- P16\_LOGVAC : Nombre de logements vacants en 2016
- P16\_RP\_PROP : Nombre de résidences principales occupées par propriétaires en 2016
- NBMENFISC16 : Nombre de ménages fiscaux en 2016
- PIMP16 : Part des ménages fiscaux imposés en 2016
- MED16 : Médiane du niveau de vie en 2016
- TP6016 : Taux de pauvreté en 2016
- P16\_EMPLT : Nombre d'emplois au lieu de travail en 2016
- P16\_EMPLT\_SAL : Nombre d'emplois salariés au lieu de travail en 2016
- P11\_EMPLT : Nombre d'emplois au lieu de travail en 2011
- P16\_POP1564 : Nombre de personnes de 15 à 64 ans en 2016
- P16\_CHOM1564 : Nombre de chômeurs de 15 à 64 ans en 2016
- P16\_ACT1564 : Nombre de personnes actives de 15 à 64 ans en 2016
- ETTOT15 : Total des établissements actifs au 31 décembre 2015
- ETAZ15 : Établissements actifs de l'agriculture, sylviculture et pêche au 31/12/2015
- ETBE15 : Établissements actifs de l'industrie au 31/12/2015
- ETFZ15 : Établissements actifs de la construction au 31/12/2015
- ETGU15 : Établissements actifs du commerce, transports et services divers au 31/12/2015
- ETGZ15 : dont Établissements actifs du commerce et réparation automobile au 31/12/2015
- ETOQ15 : Établissements actifs de l'administration publique, enseignement, santé et action sociale au 31/12/2015
- ETTEF115 : Établissements actifs de 1 à 9 salariés au 31 décembre 2015
- ETTEFP1015 : Établissements actifs de 10 salariés ou plus au 31 décembre 2015

### Requêtes SQL

1. Code géographique de Nice.
2. Communes de Côte-d'Or. Afficher code commune, nom, population de 2016.
3. Même question mais en portant la condition sur CODGEO.
4. Communes ne comptant aucun habitant en 2011, ni en 2016. Afficher seulement le nom et le département.

5. Communes comptant au moins deux fois plus d'habitants en 2016 qu'en 2011 (et habitée en 2011). Afficher seulement le nom.
6. Communes dont le nombre d'habitants a été divisé par 2 ou plus de 2011 à 2016. Afficher le nom, la population en 2011 et 2016. Trier par population en 2016 décroissante.
7. Communes des Alpes-Maritimes de strictement plus de 3000 habitants en 2011 dont la population a baissé de 2011 à 2016. Afficher le nom et les nombres d'habitants en 2011 et 2016.
8. Communes de Gironde pour lesquelles le nombre de logements est strictement supérieur à la population en 2016.
9. Communes du Finistère dont le nom contient « Ker ». Afficher le nom et la population en 2016.
10. Communes du Cantal de 1000 habitants au plus en 2016 avec au moins 5 établissement actifs du commerce et de la réparation automobile. Afficher le nom, la population en 2016 et le nombre de tels établissements. Trier par ce dernier nombre dans l'ordre décroissant.
11. Départements comprenant au moins une commune dont le nom commence par « Saint-Maurice ». Afficher les numéros de département sans répétition.
12. Communes des Alpes-Maritimes dont la part de population de 15 à 64 ans en 2016 est supérieure ou égale à 70%. Afficher le nom de la commune, le pourcentage de la population de 15 à 64 ans. Nommer cette dernière colonne « Pourcentage ».