

Chapitre 9

Algèbre relationnelle

Dans ce chapitre, nous allons présenter un modèle de bases de données qui repose sur des relations : *le modèle des bases de données relationnelles*. Un certain niveau d'abstraction étant nécessaire pour pouvoir gérer des bases de données de taille importante, nous introduirons des opérateurs sur ces relations (union, intersection, etc ...), puis nous verrons comment utiliser ces opérateurs pour effectuer des requêtes dans une table.

I Limites des structures de données plates pour la recherche d'informations

On va illustrer sur un exemple la difficulté qu'il peut y avoir à représenter ou à rechercher certains types d'informations à l'aide de structures plates telles que les tableaux. Cet exemple justifiera que nous présentions le modèle relationnel des bases de données.

Le responsable d'une bibliothèque peut choisir de gérer les emprunts par un simple fichier sur un tableur. Les premières lignes de ce fichier pourront alors ressembler à ce qui suit :

index	titre	genre	auteurN	auteurP	nom	prénom	tél	date	retour
1	Salambo	Roman	Flaubert	Gustave	Castel	Claude	0612345112	12-août-10	23-août-10
2	93	Roman	Hugo	Victor	Le Dray	Camille	0422113827	17-sept.-10	03-oct.-10
3	Mme Bovary	Littérature	Flaubert	Gustave	Loutard	Annie	064331282	01-nov.-10	02-déc.-10
1	Salambô	Roman	Flaubert	Gustave	Filâtre	Jean	0432168719	02-sept.-10	14-sept.-10
4	Aphorismes	Littérature	Wilde	Oscar	Castel	Marie	0412324494		
5	L'immoraliste	Roman	Gide	André	Biraud	Michèle	0434578612	05-févr.-11	
3	Madame Bovary	Roman	Flaubert	Gustave	Castel	Claude	0412324494	06-janv.-10	
2	Quatre-vingt-treize	Roman	Hugo	Victor	Filâtre	Jean	0432168719	20-oct.-10	03-nov.-10
1	Salambo	Littérature	Flaubert	gustave	Ledray	Camille	0422113627	16-sept.-10	27-sept.-10
...

FIGURE 9.1 – Utilisation d'un tableur pour gérer une bibliothèque.

Une telle représentation pose alors de nombreux problèmes :

- l'orthographe des différents noms de famille peut être source d'erreurs de saisie (Le Dray ou Ledray)
- l'écriture des titres des ouvrages peut ne pas être uniforme (Mme ou Madame Bovary)
- les numéros de téléphone sont alors saisis à chaque emprunt et des erreurs peuvent apparaître
- l'entrée de la date peut être non cohérente (l'ouvrage Mme Bovary ne peut pas avoir été emprunté une deuxième fois en novembre 2010 puisqu'il apparaît par ailleurs comme emprunté depuis janvier 2010)

On imagine aisément que dans une situation réelle, un tel fichier contiendrait des milliers de lignes. Retrouver alors les prêts de Mme Le Dray peut s'avérer compliqué (du fait de la faute d'orthographe) et coûteux (en temps de calcul). Il peut être également difficile de rechercher le nom des emprunteurs qui ont

des livres à rendre depuis plus de 2 mois (le délai d'emprunt est supposé être de deux mois).

Le fait d'avoir conçu un outil de gestion à structure «plate» nécessite alors une lecture séquentielle complète du tableau pour chaque interrogation.

Pour rechercher le nom des emprunteurs fautifs, il faudrait tout d'abord parcourir chaque ligne, puis au sein de chaque ligne, filtrer les noms des emprunteurs qui ont emprunté un livre il y a plus de 2 mois.

L'invention du modèle relationnel a permis de résoudre ces problèmes d'intégrité et de cohérence des données tout en rendant plus facile et plus efficace la recherche dans une base de données.

Le modèle relationnel repose sur l'algèbre relationnel que nous allons donc présenter par la suite.

II Représentation dans le modèle relationnel

II.1 Introduction sur un exemple

On va représenter l'ensemble des données de la bibliothèque dans 5 tables de manière beaucoup plus exploitable.

Livres			
id	auteur	titre	genre
1	1	Salambô	1
2	2	Quatre-vingt-treize	1
3	3	L'immoraliste	1
4	4	Aphorismes	2
5	1	Madame Bovary	1
...

La table des Livres comporte une colonne identifiant le livre, une autre identifiant l'auteur par son numéro, ainsi qu'une colonne identifiant le genre du livre par un numéro. Les titres des ouvrages ne sont pas répétés ce qui assure la cohérence des données.

Auteurs		
id	nom	prénom
1	Flaubert	Gustave
2	Hugo	Victor
3	Gide	André
4	Wilde	Oscar
...

La table des Auteurs comporte également une colonne identifiant l'auteur.

Emprunts			
qui	quoi	date	rendu
1	1	12-août-10	23-août-10
6	1	02-sept.-10	14-sept.-10
2	1	16-sept.-10	27-sept.-10
2	2	17-sept.-10	03-oct.-10
6	2	20-oct.-10	03-nov.-10
3	5	01-nov.-10	03-déc.-10
1	5	06-janv.-11	NULL
5	3	05-févr.-11	NULL
...

La table des Emprunts comporte une colonne indiquant le nom de l'emprunteur, une autre indiquant l'ouvrage qu'il a emprunté. On notera que si l'ouvrage n'a pas encore été rendu, la date de rendu est alors NULL.

Emprunteurs			
id	nom	prénom	tél
1	Castel	Claude	0612345112
2	Le Dray	Camille	0422113827
3	Loutard	Annie	0654331282
4	Castel	Marie	0412324494
5	Biraud	Michèle	0434578612
6	Filâtre	Jean	0432168719
...

La table des Emprunteurs ne présente pas de répétitions, ce qui garantit la cohérence des données.

Genre	
id	genre
1	Roman
2	Littérature
...	...

II.2 Vocabulaire du modèle relationnel

Définition 1.

Les noms des colonnes de chaque table s'appellent les *attributs*.

Quand on formalise une table quelconque, on note en général les attributs A_1, A_2, A_3, \dots

À chaque attribut est associé son *domaine*, qui est l'ensemble des valeurs qu'il peut prendre : un entier, un flottant, une chaîne de caractères, etc.

On note $\text{Dom}(A_i)$ le domaine de l'attribut A_i .

Exemple 1 Dans le cadre de l'exemple précédent, considérons que la bibliothèque dispose de 10000 livres, écrits par 9000 auteurs et que les livres sont classés selon 20 genres. Dans ce cas, la table **Livres** possède

4 attributs : id, auteur, titre et genre.

$\text{Dom}(\text{id}) = \{1, 2, \dots, 10000\}$ Le domaine de **id** est plus largement celui des entiers naturels.

$\text{Dom}(\text{auteur}) = \{1, 2, \dots, 9000\}$ Le domaine de **auteur** est plus largement celui des entiers naturels.

$\text{Dom}(\text{titre})$ est l'ensemble des titres de livres présents dans la bibliothèque. Le domaine de **titre** est plus largement celui des chaînes de caractères.

$\text{Dom}(\text{genre}) = \{1, 2, \dots, 20\}$ Le domaine de **genre** est plus largement celui des entiers naturels.

Définition 2.

Chaque ligne de la table s'appelle un *n-uplet* (tuple en anglais) : c'est un élément de $\text{Dom}(A_1) \times \text{Dom}(A_2) \times \text{Dom}(A_3) \times \dots$

L'ensemble des *n-uplets* de la table s'appelle une *relation* : une relation *r* est donc une partie de $\text{Dom}(A_1) \times \text{Dom}(A_2) \times \text{Dom}(A_3) \times \dots$

La base de données est donc constituée d'un nombre fini de relations : une par table.

On dit que la relation *r* suit le schéma de relation $R = (A_1, A_2, A_3, \dots)$. L'ensemble des schémas de relation s'appelle le *schéma de la base de données*.

Remarque : une table ayant une seule relation associée, il est assez fréquent de donner le même nom à la table et à la relation.

Exemple 2 La table **Livres** a pour relation $r = \{(1, 1, \text{Salambô}, 1), (2, 2, \text{Quatre-vingt-treize}, 1), \dots\}$.

La relation *r* suit le schéma de relation $R = (\text{id}, \text{auteur}, \text{titre}, \text{genre})$.

Notation 1.

Soit une relation *r* de schéma de relation $R = (A_1, A_2, A_3, \dots, A_n)$.

Si *t* est un *n-uplet* de cette relation, on notera :

- $t[A_i]$ la composante du *n-uplet* *t* associée à l'attribut A_i .
- $t[A_i, A_j]$ le couple extrait du *n-uplet* *t* associé aux attributs A_i et A_j .
- Plus généralement, si K est un sous-ensemble d'attributs ($K \subset \{A_1, A_2, \dots, A_n\}$), on notera $t[K]$ la liste extraite du *n-uplet* associée aux seuls attributs de K .

Exemple 3 Dans la table **Emprunteurs**, si *t* est le *n-uplet*

(5, Biraud, Michèle, 0434578612), on note :

$t[\text{nom}] = (\text{Biraud})$

$t[\text{id}, \text{prénom}] = (5, \text{Michèle})$

III Opérateurs sur le modèle relationnel

III.1 Description des recherches

Une fois les données représentées sous forme relationnelle, il faut pouvoir les exploiter pour y rechercher des informations précises.

Le langage de requêtes SQL va nous permettre d'exprimer ces recherches sous une forme précise. Donnons un aperçu de ce langage que nous étudierons plus tard.

On considère la table **Emprunteurs** et la requête suivante :

```
SELECT nom FROM Emprunteurs WHERE prénom="Claude" OR prénom="Jean"
```

Elle correspond à une recherche précise : *obtenir la liste des noms de tous les emprunteurs se prénommant Claude ou Jean.*

Découpons cette recherche en opérations élémentaires :

- sélectionner dans la table **Emprunteurs** les n -uplets dont l'attribut **prénom** correspond à "Claude"
- sélectionner dans la table **Emprunteurs** les n -uplets dont l'attribut **prénom** correspond à "Jean"
- regrouper ces deux ensembles de n -uplets
- extraire de chacun de ces n -uplets, l'attribut **nom** .

On pourrait donc décomposer cette recherche à l'aide de trois *opérateurs* :

- une *sélection* permettant d'extraire des n -uplets d'une relation qui satisfont à une condition donnée.
- une *union* pour regrouper les n -uplets de deux relations, sous réserve que les deux relations aient le même schéma relationnel
- une *projection* pour ne conserver d'une relation que les composantes associées à certains attributs

Nous allons décrire aux paragraphes suivants les opérateurs les plus usuels. Il sont de deux types :

- opérateurs ensemblistes usuels : union, intersection, différence d'ensembles.
- opérateurs spécifiques à l'algèbre relationnelle : projection, sélection, renommage, jointure, ...

Une fois que ces différents opérateurs seront connus, il vous sera alors nécessaire de traduire vos recherches dans un langage comme SQL à l'aide d'une succession d'opérateurs relationnels. On verra que cette traduction n'est pas unique et qu'il existe même de nombreuses possibilités (plus ou moins coûteuses en mémoire ou en temps de calcul) pour effectuer une même recherche.

III.2 Opérateurs ensemblistes usuels

Tout d'abord, rappelons deux notations utilisées en mathématiques pour définir des ensembles.

Notation 2.

- Si E est un ensemble et si f est une application définie sur E , on note $\{f(x)/x \in E\}$ ou $\{f(x), x \in E\}$, l'ensemble des images des éléments de E par f .
- Si E est un ensemble et si $\mathcal{P}(x)$ est une proposition qui dépend du paramètre x de E , on note $\{x \in E, \mathcal{P}(x)\}$ ou $\{x \in E/\mathcal{P}(x)\}$ l'ensemble des éléments de E qui vérifient \mathcal{P} .

On a vu qu'une relation est un ensemble fini de n -uplets de $D_1 \times \dots \times D_n$ où les D_i sont les domaines de ses attributs.

Si deux relations ont le même schéma, on peut utiliser les opérateurs ensemblistes.

On utilisera notamment : l'union (\cup), l'intersection (\cap) et la différence ensembliste ($-$).

On considère tout au long de cette section deux relations *Auteurs1* et *Auteurs2* ayant le même schéma (id,nom,prénom) sur lesquelles on peut donc appliquer les opérateurs ensemblistes. Ces deux relations peuvent par exemple décrire les noms des auteurs d'ouvrages présents dans deux bibliothèques.

<i>Auteurs1</i>		
id	nom	prénom
1	Flaubert	Gustave
2	Hugo	Victor
3	Gide	André
4	Malraux	André
<i>Auteurs2</i>		
id	nom	prénom
1	Bergson	Henri
2	Hugo	Victor

• **Union :**

On considère une relation r ayant pour schéma $R=(A_1, A_2, A_3, \dots, A_n)$ et une seconde relation s ayant le même schéma $S=R=(A_1, A_2, A_3, \dots, A_n)$.

L'union des deux relations r et s est l'ensemble des n -uplets présents dans r ou dans s . On la note $r \cup s$. Voici le résultat de l'union des deux relations *Auteurs1* et *Auteurs2* :

<i>Auteurs1</i> \cup <i>Auteurs2</i>		
id	nom	prénom
1	Flaubert	Gustave
2	Hugo	Victor
3	Gide	André
4	Malraux	André
1	Bergson	Henri
2	Hugo	Victor

• **Intersection :**

On considère une relation r ayant pour schéma $R=(A_1, A_2, A_3, \dots, A_n)$ et une seconde relation s ayant le même schéma $S=R=(A_1, A_2, A_3, \dots, A_n)$.

L'intersection des deux relations r et s est l'ensemble des n -uplets présents dans r et dans s . On la note $r \cap s$.

Voici le résultat de l'intersection des deux relations *Auteurs1* et *Auteurs2* :

<i>Auteurs1</i> \cap <i>Auteurs2</i>		
id	nom	prénom
2	Hugo	Victor

• **Différence :**

On considère une relation r ayant pour schéma $R=(A_1, A_2, A_3, \dots, A_n)$ et une seconde relation s ayant le même schéma $S=R=(A_1, A_2, A_3, \dots, A_n)$.

La différence $r - s$ est l'ensemble des n -uplets présents dans r mais qui ne sont pas présents dans s .

Voici le résultat de l'intersection des deux relations *Auteurs1* et *Auteurs2* :

<i>Auteurs1</i> - <i>Auteurs2</i>		
id	nom	prénom
1	Flaubert	Gustave
3	Gide	André
4	Malraux	André

III.3 Opérateurs spécifiques

• **Projection :**

On considère une relation r ayant pour schéma $R=(A_1, A_2, A_3, \dots, A_n)$.

La projection permet de récupérer un sous-ensemble des composantes des n -uplets de la relation r .

Si K désigne un sous-ensemble d'attributs ($K \subset \{A_1, A_2, \dots, A_n\}$), la projection de la relation r sur K est définie par :

$$\pi_K(r) = \{t[K], t \in r\}.$$

Considérons à nouveau la relation *Auteurs1* :

<i>Auteurs1</i>		
id	nom	prénom
1	Flaubert	Gustave
2	Hugo	Victor
3	Gide	André
4	Malraux	André

La projection de la relation *Auteurs1* sur les attributs (id,prénom) est alors :

$$\frac{\pi_{(\text{id}, \text{prénom})}(Auteurs1)}{\text{id} \quad \text{prénom}}$$

id	prénom
1	Gustave
2	Victor
3	André
4	André

Une projection ne contient pas forcément autant de n -uplets que la relation de départ. En effet, plusieurs n -uplets peuvent être fusionnés.

Par exemple, la projection $\pi_{\text{prénom}}(Auteurs1)$ ne contient plus 4 n -uplets, mais seulement 3.

$$\frac{\pi_{(\text{prénom})}(Auteurs1)}{\text{prénom}}$$

prénom
Gustave
Victor
André

• **Sélection :**

Une formule de sélection F est une formule construite à partir des constantes, des attributs, des fonctions usuelles, des opérateurs de comparaison, des connecteurs logiques (et, ou, non).

Étant donnée une formule de sélection, on définit alors la sélection d'une relation r selon F par :

$$\sigma_F(r) = \{t \in r, t \text{ satisfait la condition } F\}.$$

On peut ainsi sélectionner les auteurs dont le prénom est André :

$$\frac{\sigma_{\text{prénom}="André"}(Auteurs1)}{\text{id} \quad \text{nom} \quad \text{prénom}}$$

id	nom	prénom
3	Gide	André
4	Malraux	André

On peut également rechercher les auteurs dont l'identifiant est inférieur ou égal à 3 :

$$\frac{\sigma_{\text{id} \leq 3}(Auteurs1)}{\text{id} \quad \text{nom} \quad \text{prénom}}$$

id	nom	prénom
1	Flaubert	Gustave
2	Hugo	Victor
3	Gide	André

On peut enfin sélectionner les auteurs dont le prénom est André ou dont le nom est Flaubert :

$$\frac{\sigma_{\text{prénom}="André" \text{ ou } \text{nom}="Flaubert"}(Auteurs1)}{\text{id} \quad \text{nom} \quad \text{prénom}}$$

id	nom	prénom
1	Flaubert	Gustave
3	Gide	André
4	Malraux	André

• **Renommage :**

Il est possible, souvent pour lever une ambiguïté, de renommer un attribut d'une relation à l'aide d'un opérateur de renommage.

La relation obtenue est alors identique à la relation de départ (les n -uplets ont tous été repris à l'identique). C'est donc le schéma qui est modifié : le nom d'un attribut est modifié.

Le renommage sera utile plus tard, notamment pour réaliser des jointures.

Formellement : soit une relation r ayant pour schéma $R=(A_1, A_2, A_3, \dots, A_n)$ dont on veut renommer un attribut A_{i_0} par l'attribut B . On peut renommer A_{i_0} en B :

$$\rho_{A_{i_0} \leftarrow B}(r) = \{t, \exists u \in r, t[B] = u[A_{i_0}] \text{ et } \forall i \in \llbracket 1, n \rrbracket \setminus \{i_0\}, t[A_i] = u[A_i]\}.$$

Le schéma de la relation $\rho_{A_{i_0} \leftarrow B}(r)$ est alors : $(A_1, A_2, \dots, A_{i_0-1}, B, A_{i_0+1}, \dots, A_n)$.

Par extension, on peut renommer plusieurs attributs. Si une relation r a un schéma $R=(A_1, A_2, A_3, \dots, A_n)$ que l'on souhaite renommer par (B_1, B_2, \dots, B_n) , on notera dans ce cas :

$$\rho_{A_1, \dots, A_n \leftarrow B_1, \dots, B_n}(r) = \{t, \exists u \in r, \forall i \in \llbracket 1, n \rrbracket, t[B_i] = u[A_i]\}$$

Dans la relation $Livres_1$, en renommant **nom** en **LastName** et **prénom** en **FirstName**, on obtient :

$\rho_{\text{nom,prénom} \leftarrow \text{LastName,FirstName}}(\text{Auteurs}_1)$		
id	LastName	FirstName
1	Flaubert	Gustave
2	Hugo	Victor
3	Gide	André
4	Malraux	André

III.4 Algèbre relationnel

Les opérations vus précédemment peuvent être combinées entre elles pour réaliser des opérations plus complexes. L'ensemble des opérations obtenues par composition s'appelle l'*algèbre relationnelle*.

Méthode (Traduire dans le langage de l'algèbre relationnelle des requêtes).

Tout d'abord il faut identifier la relation sur laquelle on souhaite effectuer une requête simple. On peut alors procéder dans l'ordre suivant :

1. Traduire les critères présents dans la requête sous forme de sélection.
2. Si besoin, utiliser des opérations ensemblistes pour combiner les sélections entre elles.
3. Utiliser des projections pour ne conserver que les informations utiles.

Remarque : cet ordre convient dans la plupart des cas, cependant il n'est pas forcément le plus approprié en terme d'efficacité.

Si des attributs ne jouent aucun rôle, il peut être intéressant de projeter en premier sur les attributs utiles pour réduire la taille des données à traiter.

Exercice 1 On considère les relations :

<i>classe</i>			
<i>id</i>	<i>Filière</i>	<i>Numéro</i>	<i>PP</i>
1	PTSI	1	Maths
2	PT	1	SI
3	TSI	1	Maths
4	TSI	2	Physique

<i>élève</i>			
<i>Nom</i>	<i>Prénom</i>	<i>Classe</i>	<i>Note</i>
Mironneau	Marc	1	10.5
Pardo	Eugénie	2	5
Manderscheid	Benjamin	4	12
Illingworth	Thomas	2	11
Lenel	Corinne	1	8
Juhan	Jérôme	3	16
Mironneau	Didier	1	11.5
Lenel	Aude	3	9

Traduire les requêtes suivantes dans le langage de l'algèbre relationnelle :

1. Obtenir la liste des filières proposées dans ce lycée.
2. Obtenir toutes les informations concernant les classes de TSI.
3. Obtenir les prénoms des élèves des classes 1 et 3.
4. Obtenir les noms et les notes des élèves ayant eu une note inférieure ou égale à 10.

Solution :

IV Utilisation d'un gestionnaire de bases de données relationnelles

Les logiciels de gestion de bases de données relationnelles (SGBDR) servent à concevoir informatiquement des relations suivant un schéma relationnel, à définir des valeurs (les n -uplets) et à interroger la base pour rechercher des valeurs particulières.

Une base de données est constituée d'un ensemble de relations, ainsi que de leurs schémas relationnels. Elle est stockée dans un fichier ou dans plusieurs. Un gestionnaire de bases de données va gérer cette base et permettre à des utilisateurs d'y accéder ou de la modifier.

Le gestionnaire joue donc un rôle d'intermédiaire entre la vision idéale, issue du modèle relationnel, et la réalité concrète de la base.

Dans un gestionnaire de bases de données, il est possible d'effectuer les opérations suivantes :

- créer une table à l'aide d'un schéma
- ajouter les n -uplets dans une relation
- modifier les n -uplets d'une relation
- modifier le schéma d'une table : cette opération est délicate et nécessite de donner une valeur par défaut à chaque attribut ajouté.

Les gestionnaires de bases de données (SQLite, MySQL, Oracle, ...) ne fournissent pour la plupart que des outils rudimentaires pour interagir avec eux, à savoir une console dans laquelle on saisit la requête. Pour faciliter les manipulations, on peut utiliser une interface graphique. Il existe de nombreux logiciels de

ce type.

On utilisera pour les TP phpMyAdmin.

IV.1 Description du langage SQL

Le langage SQL est un langage spécifique permettant d'effectuer des requêtes sur une base.

SQL signifie Structured Query Language (*langage structuré de requêtes*). Bien qu'il y ait de nombreux gestionnaires de bases de données, tous comprennent des requêtes écrites dans ce langage.

Les requêtes permettent aussi bien de créer une base que d'effectuer des recherches. **On ne s'intéressera au langage SQL que pour les requêtes de recherche.** Nous utiliserons phpMyAdmin pour créer une petite base de données sans avoir à saisir de requête.

Le langage SQL est conçu pour exprimer la plupart des opérateurs de l'algèbre relationnel. On va indiquer ici

la syntaxe de ce langage pour effectuer des requêtes de recherche en présentant la traduction des opérations décrites précédemment. Toutes les requêtes de recherche commencent par SELECT et se terminent par un point-virgule.

IV.2 Projection

Pour effectuer la projection $\pi_{A_1, A_2, \dots, A_n}(r)$, on saisit la requête :

```
SELECT A_1, ..., A_n FROM r;
```

Il est possible d'obtenir l'intégralité de la relation grâce à l'utilisation du joker * à la place des attributs :

```
SELECT * FROM r;
```

IV.3 Sélection

La commande SELECT admet un paramètre supplémentaire, WHERE, qui nécessite de préciser une condition. Une requête de sélection s'écrit alors :

```
SELECT * FROM r WHERE A=a;
```

La commande SELECT permet donc de réaliser simultanément une projection et une sélection. La syntaxe est donc très simple et tient compte du fait que projection et sélection vont souvent ensemble.

On a vu que si la sélection reposait sur plusieurs conditions, il suffisait bien souvent d'utiliser des connecteurs logiques.

Rappelons que nous avons vu comment sélectionner les auteurs dont le prénom est André ou dont le nom est Flaubert :

$$\sigma_{\text{prénom}=\text{"André"} \text{ ou } \text{nom}=\text{"Flaubert"}}(\text{Auteurs1})$$

Dans le langage SQL, on peut traduire ainsi cette recherche :

```
SELECT * FROM Auteurs1 WHERE (prénom="André" OR nom="Flaubert");
```

IV.4 Opérations ensemblistes

Pour réaliser des opérations ensemblistes entre deux relations r et s , il faut d'abord sélectionner l'intégralité des relations à l'aide d'une projection avec le joker *. Ensuite, on utilise les mots-clés UNION, INTERSECT ou EXCEPT pour réaliser les opérations d'union, d'intersection et de différence.

Pour obtenir $r \cup s$:

```
(SELECT * FROM r) UNION (SELECT * FROM s);
```

Pour obtenir $r \cap s$:

```
(SELECT * FROM r) INTERSECT (SELECT * FROM s);
```

Pour obtenir $r - s$:

```
(SELECT * FROM r) EXCEPT (SELECT * FROM s);
```

Ces parenthèses sont facultatives mais facilitent la lecture d'une requête.

IV.5 Renommage

Pour renommer un attribut, on utilise le mot-clé AS.

Soit une relation r ayant pour schéma $(A_1, \dots, A_n, C_1, \dots, C_n)$. Si l'on souhaite renommer A_1, \dots, A_n par B_1, B_2, \dots, B_n , c'est-à-dire effectuer l'opération $\rho_{A_1, \dots, A_n \leftarrow B_1, \dots, B_n}(r)$, on effectue :

```
SELECT A1 AS B1, ..., An AS Bn, C1, ..., Cn Bn FROM r;
```

On notera qu'il faut saisir tous les attributs.

Dans l'exemple vu précédemment :

```
SELECT id, nom AS LastName, prénom AS FirstName FROM Auteurs1;
```

Méthode (Traduire dans le langage SQL des requêtes).

Les requêtes SQL ne sont pas sensibles à la casse, mais l'usage est de mettre les mots-clés en majuscules et les attributs en minuscules. Les valeurs des attributs non numériques sont saisies entre guillemets simples ou doubles. On note que :

- toutes les requêtes de recherche commencent par le mot-clé SELECT, qui permet d'effectuer une projection en fin de requête.
- la relation sur laquelle on travaille est précisée par FROM.
- les requêtes SQL se terminent par un point-virgule.

SELECT permet donc d'effectuer une projection.

SELECT suivi de la commande WHERE permet d'effectuer une sélection suivie d'une projection.

Exercice 2 On reprend l'exercice précédent.

Traduire cette fois les requêtes suivantes dans le langage SQL :

1. Obtenir la liste des filières proposées dans ce lycée.
2. Obtenir toutes les informations concernant les classes de TSI.
3. Obtenir les prénoms des élèves des classes 1 et 3.
4. Obtenir les noms et les notes des élèves ayant eu une note inférieure ou égale à 10.

Solution :