

1 Problématique

Nous n'avons, pour l'instant, effectué que des requêtes simples et sur une seule table. Mais la puissance d'une base de données est de pouvoir mettre en relation de manière efficiente, des informations multiples.

Comment effectuer des requêtes plus complexes pour mettre en relation plusieurs tables ?

2 Fonctions d'agrégation

Ce sont des fonctions qui vont regrouper les lignes. La plupart des fonctions d'agrégation vont permettre de faire des statistiques sur les données. Le code 1 compte le nombre de lignes (donc d'albums) de la série *Aya de Yopougon*.

```
1 SELECT COUNT(*) FROM Bandes_dessinees WHERE serie = "Aya de Yopougon";
```

Code 1 – Compter des lignes

À retenir

Le code suivant donnerait le même résultat que le code 1.

```
1 SELECT COUNT(titre) FROM Bandes_dessinees WHERE serie = "Aya de Yopougon";
```

Cependant si certaines lignes avaient un titre *NULL* elles ne seraient ici pas comptées.

Il existe de nombreuses fonctions d'agrégation. Citons la moyenne (*AVG*), la somme (*SUM*), le maximum (*MAX*). La documentation en ligne détaillera de manière exhaustive ces fonctions.

Activité 1 :

1. Tester la requête 1.
2. Compter le nombre total de bandes dessinées.
3. Tester la requête 2. Que renvoie-t-elle ?

```
1 SELECT COUNT(DISTINCT id_dessinateur) FROM Bandes_dessinees;
```

Code 2 – Mot clé *DISTINCT*

3 Mettre des tables en relation

3.1 Sous-requêtes

La requête 3 sélectionne toutes les bandes dessinées du genre *Jeunesse*.

```
1 SELECT titre FROM Bandes_dessinees WHERE id_genre = 13;
```

Code 3 – Bandes dessinées jeunesse

Cependant il est nécessaire de connaître l'*id* du genre *Jeunesse* dans la table *Genres*. La requête 4 récupère cet id.

```
1 SELECT id FROM Genres WHERE genre = "Jeunesse";
```

Code 4 – id du genre Jeunesse

Il est possible de combiner ces deux requêtes.

```
1 SELECT titre FROM Bandes_dessinees
2 WHERE id_genre = (SELECT id FROM Genres WHERE genre = "Jeunesse");
```

Code 5 – Sous-requête

Activité 2 :

1. Tester la requête 5.
2. Sélectionner les titres de livres dessinés par Joann Sfar.
3. Compter les titres publiés par l'éditeur Delcourt.
4. Que renvoie la requête 6 ?

```
1 SELECT titre FROM Bandes_dessinees WHERE isbn IN (SELECT isbn
FROM Emprunts WHERE id_emprunteurs = 1);
```

Code 6 – Sous requête

3.2 Jointures

Il existe des informations communes entre certaines tables. Une jointure permet de mettre en relation ces données (figure 1).

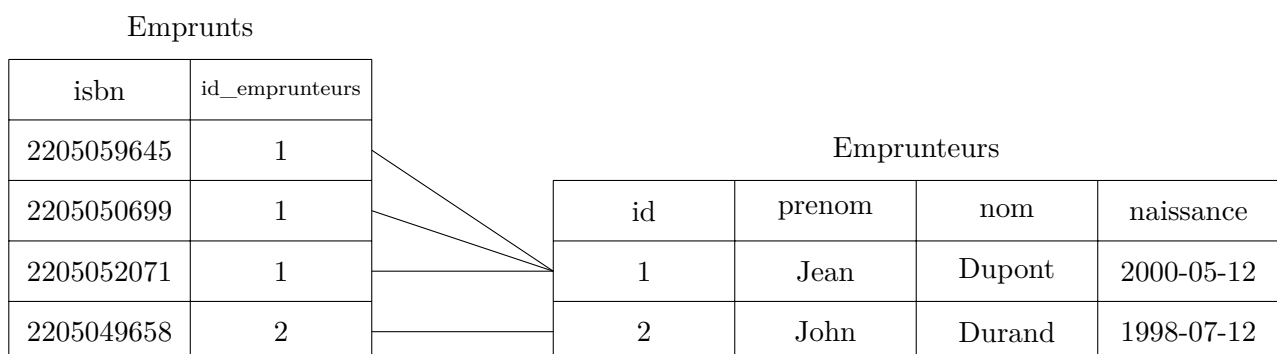


FIGURE 1 – Jointure des tables *Emprunts* et *Emprunteurs*

La jointure crée une *table temporaire* qui contient l'ensemble des données (figure 2).

isbn	id_emprunteurs	id	prenom	nom	naissance
2205059645	1	1	Jean	Dupont	2000-05-12
2205050699	1	1	Jean	Dupont	2000-05-12
2205052071	1	1	Jean	Dupont	2000-05-12
2205049658	2	2	John	Durand	1998-07-12

FIGURE 2 – Jointure des tables *Emprunts* et *Emprunteurs*

Il est alors possible de récupérer n'importe quelle information.

La requête 7 renvoie le nom des emprunteurs associés aux ISBN des bandes dessinées empruntées.

```
1 SELECT Emprunteurs.nom, Emprunts.isbn FROM Emprunts
2 JOIN Emprunteurs ON Emprunts.id_emprunteurs = Emprunteurs.id;
```

Code 7 – Jointure

Remarque

Certaines tables peuvent avoir des attributs qui possèdent le même nom. Pour éviter les ambiguïtés, il est judicieux de nommer les attributs ainsi : *table.attribut*.

Activité 3 :

1. Tester la requête 7.
2. Modifier la requête 7 pour ne renvoyer que les ISBN empruntés par Dupont.
3. Il est possible d'effectuer une jointure avec plus de deux tables. Modifier la requête précédente pour renvoyer le *titre* des bandes dessinées empruntées par Dupont.
4. Le mot clé *ORDER BY* permet de classer les résultats selon le critère donné (requête 8). Classer les résultats de la requête précédente par ordre de titre.

```
1 SELECT editeur FROM Editeurs
2 WHERE editeur LIKE "G%"
3 ORDER BY editeur;
```

Code 8 – Classement des éditeurs commençant par G