

1. toujours sur bd-avec-emprunts.db
2. (version initiale : )
3. y a eu des modif dans cours précédent

## Requêtes avancées

Christophe Viroulaud

Terminale - NSI

**BDD 06**

```
1 INSERT INTO Animaux (nom, age, id_espece) VALUES
2 ("Minou", 15, 2);
```

Code 1 – Une requête peu pratique

Il faut connaître l'id de l'espèce

```
1 INSERT INTO Animaux (nom, age, id_espece) VALUES
2 ("Minou", 15, 2);
```

Code 1 – Une requête peu pratique

Comment mettre en relation plusieurs tables ?

Comment mettre en relation plusieurs tables ?

# Sommaire

## 1. Fonctions d'agrégation

## 2. Mettre des tables en relation

Ce sont des fonctions qui vont regrouper les lignes. La plupart des fonctions d'agrégation vont permettre de faire des statistiques sur les données.

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

Code 2 – Compte le nombre de lignes (donc d'albums) de la série  
*Aya de Yopougon*

## 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.

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

Code 2 – Compte le nombre de lignes (donc d'albums) de la série  
*Aya de Yopougon*

## Remarque

Le code suivant donnerait le même résultat que le code 2.

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

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

## Remarque

Le code suivant donnerait le même résultat que le code 2.

```
1 SELECT COUNT(titre) FROM Bandes_dessinees WHERE  
2 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.

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

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

Code 3 – Mot clé DISTINCT

## Activité 1 :

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

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

Code 3 – Mot clé DISTINCT



# Sommaire

## 1. Fonctions d'agrégation

## 2. Mettre des tables en relation

### 2.1 Sous-requêtes

### 2.2 Jointures

## Requêtes avancées

## └ Mettre des tables en relation

## └ Sous-requêtes

## └ Sous-requêtes

Sous-requêtes

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

```
13;
```

Code 4 – Sélectionner toutes les bandes dessinées du genre  
jeunesse**Remarque**Il est nécessaire de connaître l'id du genre `jeunesse`  
dans la table `Genres`.

## Sous-requêtes

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

```
13;
```

Code 4 – Sélectionner toutes les bandes dessinées du genre  
**jeunesse****Remarque**Il est nécessaire de connaître l'id du genre `jeunesse`  
dans la table `Genres`.

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

Code 5 – Sélectionner l'id du genre jeunesse

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

Code 5 – Sélectionner l'id du genre jeunesse

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

Code 6 – Combiner les requêtes

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

Code 6 – Combiner les requêtes

## Activité 2 :

1. Tester la requête 6.
2. Sélectionner les titres de livres dessinés par Joann Sfar.
3. Compter les titres publiés par l'éditeur Delcourt.
4. Que renvoie la requête 7 ?

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

Code 7 – Sous requête

## Activité 2 :

1. Tester la requête 6.
2. Sélectionner les titres de livres dessinés par Joann Sfar.
3. Compter les titres publiés par l'éditeur Delcourt.
4. Que renvoie la requête 7 ?

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

Code 7 – Sous requête

```
1 SELECT titre FROM Bandes_dessinees
2 WHERE id_dessinateur =
3       (SELECT id FROM Auteurs
4        WHERE nom = "Sfar" and
5          prenom = "Joann");
```

Code 8 – Sélectionner les titres de Joann Sfar

## Correction

```
1 SELECT titre FROM Bandes_dessinees
2     WHERE id_dessinateur =
3           (SELECT id FROM Auteurs
4            WHERE nom = "Sfar" and
5              prenom = "Joann");
```

Code 8 – Sélectionner les titres de Joann Sfar

```
1 SELECT COUNT(titre) FROM Bandes_dessinees
2 WHERE id_editeur =
3     (SELECT id FROM Editeurs
4      WHERE editeur = "Delcourt");
```

Code 9 – Compter les titres d'éditeur Delcourt

```
1 SELECT COUNT(titre) FROM Bandes_dessinees
2 WHERE id_editeur =
3     (SELECT id FROM Editeurs
4      WHERE editeur = "Delcourt");
```

Code 9 – Compter les titres d'éditeur Delcourt

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

Code 10 – Sous requête

La requête renvoie les titres empruntés par l'emprunteur 1.  
La clause IN peut être comparée au mot-clef in en Python.

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

Code 10 – Sous requête

La requête renvoie les titres empruntés par l'emprunteur 1.  
La clause IN peut être comparée au mot-clef in en Python.



# Sommaire

## 1. Fonctions d'agrégation

## 2. Mettre des tables en relation

### 2.1 Sous-requêtes

### 2.2 Jointures

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

isbn	id_emprunteurs
2205059645	1
2205050699	1
2205052071	1
2205049658	2

id	prenom	nom	naissance
1	Jean	Dupont	2000-05-12
2	John	Durand	1998-07-12

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

## Jointures

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

isbn	id_emprunteurs
2205059645	1
2205050699	1
2205052071	1
2205049658	2

id	prenom	nom	naissance
1	Jean	Dupont	2000-05-12
2	John	Durand	1998-07-12

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*

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*

## Requêtes avancées

## Mettre des tables en relation

## Jointures

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

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

Code 11 – Renvoyer le nom des emprunteurs associés aux ISBN des bandes dessinées empruntées

## 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`.

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

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

Code 11 – Renvoyer le nom des emprunteurs associés aux ISBN des bandes dessinées empruntées

## 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 11.
2. Modifier la requête 11 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.

**Activité 3 :**

1. Tester la requête 11.
2. Modifier la requête 11 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.

```
1 SELECT Emprunteurs.nom, Emprunts.isbn FROM Emprunts
2 JOIN Emprunteurs
3 ON Emprunts.id_emprunteurs = Emprunteurs.id
4 WHERE Emprunteurs.nom = "Dupont";
```

Code 12 – Renvoyer les bandes dessinées empruntées par Dupont

## Correction

```
1 SELECT Emprunteurs.nom, Emprunts.isbn FROM Emprunts
2 JOIN Emprunteurs
3 ON Emprunts.id_emprunteurs = Emprunteurs.id
4 WHERE Emprunteurs.nom = "Dupont";
```

Code 12 – Renvoyer les bandes dessinées empruntées par Dupont

```
1 SELECT Emprunteurs.nom, Emprunts.isbn FROM Emprunts
2 JOIN Emprunteurs
3 ON Emprunts.id_emprunteurs = Emprunteurs.id
4 WHERE Emprunteurs.nom = "Dupont";
```

Code 13 – Renvoyer les isbn des bandes dessinées empruntées par Dupont

## Correction

```
1 SELECT Emprunteurs.nom, Emprunts.isbn FROM Emprunts
2 JOIN Emprunteurs
3 ON Emprunts.id_emprunteurs = Emprunteurs.id
4 WHERE Emprunteurs.nom = "Dupont";
```

Code 13 – Renvoyer les isbn des bandes dessinées empruntées par Dupont

```
1 SELECT Bandes_dessinees.titre FROM Bandes_dessinees
2 JOIN Emprunts
3   ON Bandes_dessinees.isbn = Emprunts.isbn
4 JOIN Emprunteurs
5   ON Emprunts.id_emprunteurs = Emprunteurs.id
6 WHERE Emprunteurs.nom = "Dupont";
```

Code 14 – Renvoyer les titres des bandes dessinées empruntées par Dupont

## Correction

```
1 SELECT Bandes_dessinees.titre FROM Bandes_dessinees
2 JOIN Emprunts
3   ON Bandes_dessinees.isbn = Emprunts.isbn
4 JOIN Emprunteurs
5   ON Emprunts.id_emprunteurs = Emprunteurs.id
6 WHERE Emprunteurs.nom = "Dupont";
```

Code 14 – Renvoyer les titres des bandes dessinées empruntées par Dupont



Activité 4 : Le mot clé *ORDER BY* permet de classer les résultats selon le critère donné (requête 15). 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 15 – Classement des éditeurs commençant par G

**Activité 4 :** Le mot clé *ORDER BY* permet de classer les résultats selon le critère donné (requête 15). 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 15 – Classement des éditeurs commençant par G

```
1 SELECT Bandes_dessinees.titre FROM Bandes_dessinees
2 JOIN Emprunts
3   ON Bandes_dessinees.isbn = Emprunts.isbn
4 JOIN Emprunteurs
5   ON Emprunts.id_emprunteurs = Emprunteurs.id
6 WHERE Emprunteurs.nom = "Dupont"
7 ORDER BY Bandes_dessinees.titre;
```

Code 15 – Renvoyer les titres ordonnés.

## Correction

```
1 SELECT Bandes_dessinees.titre FROM Bandes_dessinees
2 JOIN Emprunts
3   ON Bandes_dessinees.isbn = Emprunts.isbn
4 JOIN Emprunteurs
5   ON Emprunts.id_emprunteurs = Emprunteurs.id
6 WHERE Emprunteurs.nom = "Dupont"
7 ORDER BY Bandes_dessinees.titre;
```

Code 16 – Renvoyer les titres ordonnés.