

I Présentation d'une base de données

1 Généralités

Une base de données (*data base*) est un fichier contenant des informations (données), organisé selon une structure permettant de lier certaines informations à d'autres.

Un fichier de base de données est enregistré avec l'extension `.db`

Il est divisé en plusieurs *tables*. On parle aussi de *relations*.

Chaque table contient des *champs*, qui sont présentés en *colonnes*, appelés aussi *attributs*.

Un *enregistrement* est une *ligne* d'une table, contenant une valeur pour chaque colonne.

Il est possible que la valeur soit absente (valeur : NULL).

Les données enregistrées dans chaque *cellule* (intersection d'une ligne et d'une colonne) sont d'un type défini : `INTEGER`, `TEXT`, `REAL` ... qui correspondent en *Python* aux types : `int`, `str`, `float` ...

Les types utilisés sont des *Domaines*. On peut créer de nouveaux domaines en manipulant des types existants et en y ajoutant des contraintes.

Par exemple, on peut créer le domaine `CodePostal` qui sera de type `TEXT` avec la contrainte de contenir exactement 5 chiffres : 35000 ou 02500 seront acceptés, mais pas 2500 ni 'Rennes'.

Exemple de base de données : le fichier `Movie.db`

Tables : `<Film>`, `<Acteur>`, `<Réalisateur>`, `<Genre>`, `<Pays>`.

Attributs : `<Film>` : `ID`, `TitreOriginal`, `TitreFr`, `DateProduction`, `Genre`, `SousGenre`, ...

`<Acteur>` : `Nom`, `Prénom`, `Naissance`, `Décès`.

`<Réalisateur>` : `Nom`, `Prénom`, `Naissance`, `Décès`, `Nationalité`.

`<Genre>` : `Nom`.

`<Pays>` : `Pays`.

Exemple d'enregistrement : `<Film>` 1, `Vertigo`, `Sueurs froides`, 1958, `Policier`, ...

2 Structure des données : les clés

(a) Clé primaire

Dans une table, on peut définir un attribut comme **clé primaire**.

Dans cette table, les enregistrements (lignes) seront alors repérés par cet attribut : leur clé primaire.

On ne pourra pas donner plusieurs fois la même valeur de clé primaire à différentes lignes.

C'est une contrainte de la base de données.

Dans `Movie.db`, l'attribut `ID` de la table `<Film>` est défini comme clé primaire :

chaque ligne est identifiée par la valeur de cette colonne (`ID = identifiant`).

Dans la table `<Acteur>`, la colonne `Nom` est la clé primaire. Si deux acteurs ont le même nom, il faut donc les distinguer : `Allen`, `Woody` ou `Allen(2)`, `Karen` .

(b) Clé étrangère

Un attribut d'une table peut aussi être défini comme une clé étrangère, pointant vers un autre attribut (en général d'une autre table). Dans ce cas, on ne peut enregistrer qu'une valeur déjà présente parmi les valeurs de cet autre attribut.

Dans `Movie.db`, l'attribut `Genre` de la table `<Film>` est défini comme clé étrangère pointant vers l'attribut `Nom` de la table `<Genre>`. Dans la table `<Film>`, pour chaque nouvelle ligne, on devra renseigner dans l'attribut `Genre` une des valeurs pré-existante de la table `<Genre>`.

Contrainte supplémentaire : on ne peut pas effacer une ligne contenant une valeur vers laquelle pointe une clé étrangère. Par exemple, on pourra effacer la ligne `Action` de la table `<Genre>` car aucune ligne de la table `<Film>` n'est référencée par cette valeur, mais on ne pourra pas effacer la ligne `Western`.

II Interrogation d'une base de données

Lorsqu'on interroge une base de données, on fait du *requêtage*.

On utilise pour cela le langage informatique `SQL`, plus adapté que *Python*.

`SQL` = `Structured Query Language` = langage de requête structurée.

Toutes les actions de requête commencent par le mot-clé **SELECT** et finissent par un point-virgule.

1 Sélection d'une table entière :

```
SELECT * FROM <Table> ;
```

 La table toute entière est affichée.

2 Sélection de colonnes (projection) :

```
SELECT <Colonne1, Colonne2, ...> FROM <Table> ;
```

 Les colonnes demandées seront affichées.

Exemple :

```
SELECT TitreOriginal, DateProduction FROM Film ;
```

3 Sélection de lignes :

```
SELECT * FROM <Table> WHERE <Condition> ;
```

La condition permet de sélectionner les lignes à afficher.

Exemples :

```
SELECT * FROM Film WHERE DateProduction > 1980 ;
```

renvoie tous les films produits à partir de 1981,

```
SELECT * FROM Réalisateur WHERE Naissance < 1900 ;
```

renvoie tous les réalisateurs nés avant 1900,

```
SELECT * FROM Réalisateur WHERE Décès IS NULL ;
```

renvoie tous les réalisateurs encore en vie.

Pour exprimer une condition, on peut utiliser :

AND, OR, NOT, +, -, *, /, =, <> (ou !=), <, <=, >, >=

4 Sélection à travers plusieurs tables : les Jointures

L'intérêt de la structure des bases de données est de pouvoir faire des requêtes avec des conditions portant sur des attributs de différentes tables.

```
SELECT <Colonne> FROM <Table1, Table2> WHERE <Condition1> AND <Condition2> ;
```

permet de sélectionner la colonne voulue de la table1, et de n'afficher que les lignes satisfaisant les conditions 1 et 2. Par exemple :

```
SELECT TitreOriginal FROM Film, Réalisateur WHERE Naissance > 1930 AND Réalisateur1 = Nom ;
```

affiche la liste des films dont les réalisateurs sont nés après 1930.

Si on examine cette requête étape par étape :

- 1)

```
SELECT TitreOriginal FROM Film, Réalisateur ;
```


met en relation (et affiche) chaque film avec chaque réalisateur. C'est un produit cartésien d'ensembles.
- 2)

```
SELECT TitreOriginal FROM Film, Réalisateur WHERE Naissance > 1930 ;
```


met en relation (et affiche) chaque film avec chaque réalisateur né après 1930.
- 3)

```
SELECT TitreOriginal FROM Film, Réalisateur WHERE Naissance > 1930 AND Réalisateur1 = Nom ;
```


ne conserve dans la sélection précédente que les réponses dans lesquelles le réalisateur du film (`Réalisateur1`) correspond au réalisateur (`Nom`) dont on a sélectionné la date de naissance.

Autre syntaxe :

```
SELECT TitreOriginal FROM Film JOIN Réalisateur ON Naissance > 1930 AND Réalisateur1 = Nom ;
```

5 Jointure et sélection

```
SELECT TitreOriginal  
FROM Film  
JOIN Réalisateur  
ON Réalisateur1 = Nom AND Nationalité = 'France'  
WHERE Genre = 'Drame' ;
```

affiche les drames français. La sélection **WHERE** est écrite après la jointure.

6 Éviter les ambiguïtés, renommage (aliasing)

Si, à travers certaines tables, plusieurs colonnes portent le même nom, certaines requêtes peuvent devenir ambiguës. On fait alors précéder le nom de la colonne de celui de la table :

Acteur.Nom, Réalisateur.Nom ou Genre.Nom

On peut également renommer une table lors d'une requête : `SELECT R.Nom FROM Réalisateur as R`

Exemple : afficher tous les films réalisés par des français et dont au moins un acteur est né avant 1930.

On veut écrire :

```
SELECT TitreOriginal FROM Film
JOIN Réalisateur ON Réalisateur1 = Nom AND Nationalité = 'France'
JOIN Acteur ON ( Acteur1 = Nom OR Acteur2 = Nom ) AND Naissance < 1930 ;
```

mais cette syntaxe ne fonctionne pas, car la colonne Nom est ambiguë, étant présente à la fois dans la table <Réalisateur> et dans la table <Acteur>, de même pour Naissance. On précise donc :

```
SELECT TitreOriginal FROM Film
JOIN Réalisateur ON Réalisateur1 = Réalisateur.Nom AND Nationalité = 'France'
JOIN Acteur ON ( Acteur1 = Acteur.Nom OR Acteur2 = Acteur.Nom ) AND Acteur.Naissance < 1930 ;
```

On peut renommer les tables, pour éviter d'avoir trop à écrire :

```
SELECT TitreOriginal FROM Film
JOIN Réalisateur AS R ON Réalisateur1 = R.Nom AND Nationalité = 'France'
JOIN Acteur AS A ON ( Acteur1 = A.Nom OR Acteur2 = A.Nom ) AND A.Naissance < 1930 ;
```

On peut aussi renommer les données affichées :

```
SELECT TitreOriginal as Titre FROM Film ;
```

7 Améliorations de l'affichage : les mot-clés DISTINCT et ORDER BY

Si on ne veut afficher que des données différentes, on utilise : `SELECT DISTINCT`.

La requête :

```
SELECT A.Nom, A.Prénom FROM Acteur AS A
JOIN Film ON Acteur1 = A.Nom OR Acteur2 = A.Nom
JOIN Réalisateur AS R ON Réalisateur1 = R.Nom AND Nationalité = 'France' ;
```

permet d'afficher la liste de tous les acteurs jouant dans des films réalisés par un français.

Mais il y a alors des répétitions !

On écrit donc :

```
SELECT DISTINCT A.Nom, A.Prénom FROM Acteur AS A
JOIN Film ON Acteur1 = A.Nom OR Acteur2 = A.Nom
JOIN Réalisateur AS R ON Réalisateur1 = R.Nom AND Nationalité = 'France' ;
```

On peut également vouloir afficher les réponses dans l'ordre alphabétique des noms. On écrit :

```
SELECT DISTINCT A.Nom, A.Prénom FROM Acteur AS A
JOIN Film ON Acteur1 = A.Nom OR Acteur2 = A.Nom
JOIN Réalisateur AS R ON Réalisateur1 = R.Nom AND Nationalité = 'France'
ORDER BY A.Nom ASC ;
```

ASC ou DESC pour *ascendant* ou *descendant* (croissant, décroissant).

8 Les fonctions d'agrégat : MIN, MAX, AVG, SUM, COUNT.

COUNT permet de compter les lignes : `SELECT count(*) from Film ;` renvoie le nombre de films,

`SELECT count(DISTINCT TitreOriginal) from Film ;` renvoie le nombre de films de noms différents,

On peut aussi compter en faisant des regroupements :

```
SELECT Nationalité, count(Nationalité) FROM Réalisateur GROUP BY Nationalité ;
```

renvoie la liste des pays et le nombre de réalisateurs par pays.

MIN, MAX, SUM, AVG renvoient respectivement minimum, maximum, somme et moyenne de la sélection.

```
SELECT MIN(Naissance) FROM Acteur ;
```

 renvoie l'année de naissance du plus vieil acteur.

```
SELECT 2023 - AVG(DateProduction) FROM Film ;
```

 renvoie l'âge moyen des films.

HAVING permet de n'afficher que les réponses répondant à une certaine condition.

```
SELECT Nationalité, count(Nationalité) FROM Réalisateur
GROUP BY Nationalité
HAVING count(Nationalité) > 4 ;
```

affiche les pays et le nombre de réalisateurs originaires de ces pays, s'il y a au moins 5 réalisateurs.

III Travail à réaliser

1. Afficher le titre original et l'année de production de tous les films.
2. Afficher le titre original et l'année de production de tous les films du genre 'Comédie'.
3. Afficher nom et prénom de tous les réalisateurs britanniques.
4. Quelle est l'année de production du plus ancien film ?
5. Combien y a-t-il de films policiers (en tant que genre principal) ?
6. Combien y a-t-il de films policiers (en tant que genre principal ou secondaire) ?
7. Quel est l'âge moyen des réalisateurs encore en vie ?
8. Afficher les titres originaux de tous les films réalisés par des états-uniens.
9. Afficher dans l'ordre alphabétique les titres originaux de tous les films réalisés par des états-uniens.
10. Afficher tous les genres et le nombre de films de ce genre (en genre principal).
11. Même question en classant par nombre de films décroissant, puis par ordre alphabétique des genres.
12. Donner par ordre décroissant le nombre de films par pays.
13. Donner les noms et prénoms des acteurs qui ont joué dans au moins 4 films, et le nombre de films dans lesquels ils ont joué (par ordre décroissant).
14. Dans quels films a joué Jean-Paul Belmondo ? (par ordre alphabétique)
15. Quels sont les réalisateurs qui ont dirigé Jean-Paul Belmondo ? (éviter les répétitions)
16. Combien y en a-t-il ?
17. Donner la liste de tous les réalisateurs ayant travaillé avec Jean-Paul Belmondo (sans répétition) et le nombre de film dans lesquels ils ont collaboré (par nombre décroissant, puis ordre alphabétique).
18. Quels sont les acteurs ayant joué avec Jean-Paul Belmondo ? (sans répétition, par ordre alphabétique)
19. Quels sont les genres des films dans lesquels a joué Jean-Paul Belmondo (et dans combien de films de ce genre a-t-il joué) ?

IV Les réponses

1. Afficher le titre original et l'année de production de tous les films.

```
SELECT TitreOriginal, DateProduction FROM Film ;
```

2. Afficher le titre original et l'année de production de tous les films du genre 'Comédie'.

```
SELECT TitreOriginal, DateProduction FROM Film WHERE Genre = 'Comédie' ;
```

3. Afficher nom et prénom de tous les réalisateurs britanniques.

```
SELECT Nom, Prénom FROM Réalisateur WHERE Nationalité = 'Grande-Bretagne' ;
```

4. Quelle est l'année de production du plus ancien film ?

```
SELECT min(DateProduction) FROM Film ;
```

5. Combien y a-t-il de films policiers (en tant que genre principal) ?

```
SELECT count(*) FROM Film WHERE Genre = 'Policier' ;
```

6. Combien y a-t-il de films policiers (en tant que genre principal ou secondaire) ?

```
SELECT count(*) FROM Film WHERE Genre = 'Policier' OR SousGenre = 'Policier' ;
```

7. Quel est l'âge moyen des réalisateurs encore en vie ?

```
SELECT 2023 - avg(Naissance) FROM Réalisateur WHERE Décès IS NULL ;
```

8. Afficher les titres originaux de tous les films réalisés par des états-unien.

```
SELECT TitreOriginal FROM Film  
JOIN Réalisateur ON Réalisateur1 = Nom AND Nationalité = 'Etats-Unis' ;
```

9. Afficher dans l'ordre alphabétique les titres originaux de tous les films réalisés par des états-unien.

```
SELECT TitreOriginal FROM Film  
JOIN Réalisateur ON Réalisateur1 = Nom AND Nationalité = 'Etats-Unis'  
ORDER BY TitreOriginal ASC;
```

10. Afficher tous les genres et le nombre de films de ce genre (en genre principal).

```
SELECT Genre, count(*) from Film  
GROUP BY Genre ;
```

11. Même question en classant par nombre de films décroissant, puis par ordre alphabétique des genres.

```
SELECT Genre, count(*) from Film  
GROUP BY Genre  
ORDER BY count(*) DESC, Genre ASC ;
```

12. Donner par ordre décroissant le nombre de films par pays.

```
SELECT Pays, count(TitreOriginal) as Nbe_Films FROM Pays, Film  
JOIN Réalisateur  
ON Nom = Réalisateur1 AND Nationalité = Pays  
GROUP BY Pays  
ORDER BY Nbe_Films DESC ;
```

13. Donner les noms et prénoms des acteurs qui ont joué dans au moins 4 films, et le nombre de films dans lesquels ils ont joué (par ordre décroissant).

```
SELECT Nom, Prénom, count(Nom) FROM Acteur  
JOIN Film ON Nom = Acteur1 OR Nom = Acteur2  
GROUP BY Nom  
HAVING count(Nom) > 3  
ORDER BY count(Nom) DESC ;
```

14. Dans quels films a joué Jean-Paul Belmondo ? (par ordre alphabétique)

```
SELECT TitreOriginal from Film  
WHERE Acteur1 = 'Belmondo' OR Acteur2 = 'Belmondo'  
ORDER BY TitreOriginal ASC ;
```

15. Quels sont les réalisateurs qui ont dirigé Jean-Paul Belmondo ? (éviter les répétitions)

```
SELECT DISTINCT Nom, Prénom from Réalisateur  
JOIN Film ON Réalisateur1 = Nom AND (Acteur1 = 'Belmondo' OR Acteur2 = 'Belmondo' )  
ORDER BY Nom ASC ;
```

16. Combien y en a-t-il ?

```
SELECT count(DISTINCT Nom) FROM Réalisateur
JOIN Film
ON ( Nom = Réalisateur1 OR Nom = Réalisateur2 )
AND (Acteur1 = 'Belmondo' OR Acteur2 = 'Belmondo' ) ;
```

17. Donner la liste de tous les réalisateurs ayant travaillé avec Jean-Paul Belmondo (sans répétition) et le nombre de film dans lesquels ils ont collaboré (par nombre décroissant, puis ordre alphabétique).

```
SELECT Nom, Prénom, count(Nom) FROM Réalisateur
JOIN Film
ON ( Nom = Réalisateur1 OR Nom = Réalisateur2 )
AND (Acteur1 = 'Belmondo' OR Acteur2 = 'Belmondo' )
GROUP BY Nom
ORDER BY count(Nom) DESC, Nom ASC ;
```

18. Quels sont les acteurs ayant joué avec Jean-Paul Belmondo ? (sans répétition, par ordre alphabétique)

```
SELECT DISTINCT Nom, Prénom from Acteur
JOIN Film ON Acteur1 = 'Belmondo' OR Acteur2 = 'Belmondo'
WHERE Nom != 'Belmondo' AND (Nom = Acteur1 OR Nom = Acteur2 )
ORDER BY Nom ASC ;
```

19. Quels sont les genres des films dans lesquels a joué Jean-Paul Belmondo (et dans combien de films de ce genre a-t-il joué) ?

```
SELECT Nom, count(*) FROM Genre
JOIN Film ON Acteur1 = 'Belmondo' OR Acteur2 = 'Belmondo'
WHERE Genre = Nom
GROUP BY Genre ;
```