



Point Bac

Attention, l'épreuve écrite du bac contient toujours un exercice portant sur les bases de données. Souvent y sont posées des questions de cours.

1 Système de Gestion de base de Données (SGBD)

En classe de première on s'est intéressé à la manipulation de grande quantité de données sur des fichiers au format .csv en les manipulant à partir d'un script python Vous pourriez revoir ce TD ou au moins le revisualiser au lien suivant

[TD 3 NSI : Opérations sur les Tables](#)

Cette méthode pour gérer des données a son utilité dans un certain cadre. Parfois il faut faire de nombreuses manipulations qui peuvent aller du simple ajout d'un enregistrement à la recherche d'un ensemble d'enregistrement avec des critères très complexes.

Maintenant chaque site web a besoin de d'utiliser une base de donnée. Par exemple pour gérer les membres.

Définition 1

Un Système de Gestion de base de Données (SGBD) est un outil permettant :

- La sauvegarde des données,
- L'interrogation des données,
- La recherche des données,
- La mise en forme des données.

1.1 Les modèles de SGDB

On peut différencier 5 modèles de SGDB :

- **Le modèle hiérarchique** : Les données sont classées hiérarchiquement. On les représente sous la forme d'un arbre avec une arborescence descendante.
- **Le modèle réseau** est une extension du modèle hiérarchique dans lequel l'on rajoute des relations entre les enregistrements.
- **Le modèle relationnel** : les données sont enregistrées dans des tableaux à deux entrées.
- **Le modèle objet** : les données sont stockées sous forme de classe.
- **le modèle déductif** : qui ressemble au modèle relationnel avec une manipulation différente.

2 le modèle relationnel

Le modèle relationnel représente les données sous forme de tableaux à double entrée que l'on appelle **table** ou **relation**.

Par exemple la table *film* suivante :

film		
titre	année	genre
"La Guerre des étoiles"	1977	"Aventure"
"Apocalypse Now"	1979	"Drame"
"Impitoyable"	1992	"Western"
"La vie est belle"	1997	"Drame"
"La vie est belle"	1997	"Drame"
...

Définition 2 (Le vocabulaire)

- Un **enregistrement (ou entité)** correspond à une ligne de la *table*,
- Les **attributs** sont les noms des colonnes,
- Le **degré** d'une *table* est le nombre de ses *attributs*,
- Le **domaine** d'un *attribut* est son type : entier, flottant, chaîne de caractères(string), booléen,...,
- La **clé primaire** est l'*attribut* qui permet d'identifier de manière unique un *enregistrement*,
- Une **clé étrangère** est un *attribut* d'une *table* qui fait référence à la *clé primaire* d'une autre *table*.

Le schéma de la table *film* est

$$\{(\text{titre}, \text{string}), (\text{année}, \text{entier}), (\text{genre}, \text{string})\}$$

2.1 La clé primaire

On voit sur la table *film* que le film "La vie est belle" a été mis deux fois. Une base de données intègre ne doit pas contenir de doublon. Il faut donc rajouter un attribut qui puisse les différencier de manière unique, pour cela on ajoute un attribut appelée clef primaire. En général, cela sera un entier qui s'incrémente à chaque nouvel enregistrement.

film			
idFilm	titre	année	genre
1	"La Guerre des étoiles"	1977	"Aventure"
2	"Apocalypse Now"	1979	"Drame"
3	"Impitoyable"	1992	"Western"
4	"La vie est belle"	1997	"Drame"
5	"La vie est belle"	1997	"Drame"
...

Le schéma de la table *film* devient

$$\{(\text{idFilm}, \text{entier}), (\text{titre}, \text{string}), (\text{année}, \text{entier}), (\text{genre}, \text{string})\}$$

2.2 La clé étrangère

On pourrait rajouter des *attributs* supplémentaires sur une *table* pour avoir le plus d'informations possibles sur chaque *enregistrement*, par exemple le réalisateur et sa nationalité. Comme cela :

film						
idFilm	titre	année	genre	nomReal	prenomReal	nationalitéReal
1	"La Guerre des étoiles"	1977	"Aventure"	"Lucas"	"Georges"	"US"
2	"Apocalypse Now"	1979	"Drame"	"Ford Coppola"	"Francis"	"US"
3	"Impitoyable"	1992	"Western"	"Eastwood"	"Clint"	"US"
4	"La vie est belle"	1997	"Drame"	"Benini"	"Roberto"	"IT"
6	"Le parrain"	1972	"Drame"	"Ford Coppola"	"Francis"	"US"
...			

Cependant on obtient des redondances d'information, il y a deux fois l'information sur la nationalité de Francis Ford Coppola. On fait donc appel à une autre *table*. Cela évite aussi d'avoir à changer de nombreux *enregistrements*, par exemple si un réalisateur a changé de nationalité. Pour cela on fait appel à une *clé étrangère* qui fait référence à cette autre *table*.

On construit la table *realisateur* avec le schéma suivant :

$$\{(\text{idReal}, \text{entier}), (\text{nom}, \text{string}), (\text{prénom}, \text{string})\}$$

et on rajoute un identifiant pour le réalisateur dans la table *film* qui pourra se référer à la table *realisateur*. Son schéma devient

$$\{(\text{idFilm}, \text{entier}), (\text{titre}, \text{string}), (\text{année}, \text{entier}), (\text{genre}, \text{string}), \text{idReal}, \text{entier}\}$$

On alors les deux tables sans redondance d'information :

film				
idFilm	titre	année	genre	idReal
1	"La Guerre des étoiles"	1977	"Aventure"	2
2	"Apocalypse Now"	1979	"Drame"	4
3	"Impitoyable"	1992	"Western"	3
4	"La vie est belle"	1997	"Drame"	5
6	"Le parrain"	1972	"Drame"	4
...

realisateur			
idReal	nom	prénom	nationalité
1	"Capra"	"Frank"	"US"
4	"Francis"	"Ford Coppola"	"US"
3	"Eastwood"	"Clint"	"US"
2	"Lucas"	"Georges"	"US"
5	"Benigni "	"Roberto "	"IT"
...

Définition 3 (Base de données (Database))

Une **base de donnée** est un ensemble de tables dont certaines peuvent être mises en relation. Les **clés étrangères** permettent de gérer des relations entre plusieurs tables, et garantissent la cohérence des données. On peut ainsi modifier des données d'une table sans avoir à modifier plusieurs tables.

2.3 Schématisation

2.3.1 Schéma Entité/association

On schématise une BDD par les structures des tables où sont reliées les clés primaires et étrangères en relation.

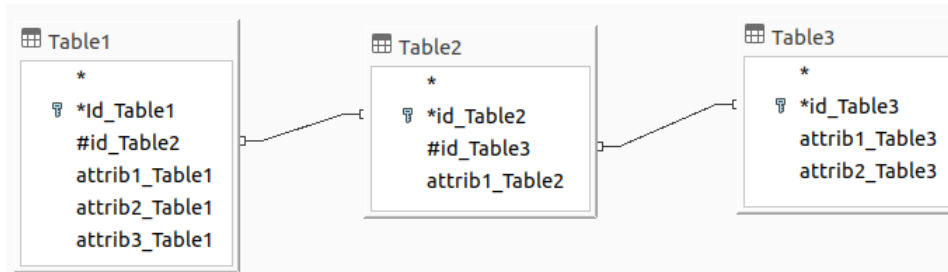


FIGURE 1 – Schéma Entité/Association (E/A)

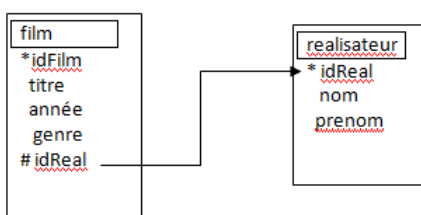


FIGURE 2 – Schéma relationnel (E/A), base de donnée films et réalisateurs

2.3.2 Schéma relationnel

La base de donnée donnée en exemple se schématise aussi par son **schéma relationnel** par

$$realisateur = \{(\underline{idReal}, entier), (nom, string), (prénom, string)\}$$

$$film = \{(\underline{idFilm}, entier), (titre, string), (année, entier), (genre, string), \#idReal, entier)\}$$

Remarquez que l'on souligne les clefs primaires et que l'on met # devant une clef étrangère.

3 Avantages de l'utilisation de plusieurs tables dans une base de données

Utiliser plusieurs tables dans une base de données relationnelle présente plusieurs avantages par rapport à l'utilisation d'une seule table :

- **Réduction de la redondance des données** : En séparant les informations en différentes tables, chaque donnée n'est stockée qu'une seule fois. Par exemple, si vous avez une table pour les clients et une autre pour les commandes, les informations sur chaque client n'ont besoin d'être saisies qu'une seule fois, évitant ainsi les doublons.
- **Clarté et organisation** : Les données sont mieux organisées, ce qui facilite leur gestion et leur compréhension. Chaque table peut se concentrer sur un sujet spécifique, ce qui rend la structure de la base de données plus logique et intuitive.
- **Facilité de mise à jour** : Lorsqu'une information doit être modifiée, il suffit de le faire dans une seule table. Par exemple, si l'adresse d'un client change, vous n'avez qu'à mettre à jour cette information dans la table des clients, plutôt que dans chaque enregistrement de commande.
- **Amélioration de l'intégrité des données** : En utilisant des relations entre les tables, il est plus facile de maintenir l'intégrité des données. Les clés primaires et étrangères permettent de s'assurer que les données restent cohérentes et fiables.
- **Flexibilité et évolutivité** : Une structure de base de données utilisant plusieurs tables est plus facile à adapter aux changements futurs. Si de nouvelles informations doivent être ajoutées, il est souvent plus simple de créer une nouvelle table ou d'ajouter une nouvelle relation entre les tables existantes.
- **Optimisation des performances** : Les requêtes peuvent être plus efficaces car elles peuvent cibler des tables spécifiques plutôt que de parcourir une grande table unique. Cela peut réduire le temps de réponse lors de la recherche ou de la manipulation des données.

En résumé, l'utilisation de plusieurs tables dans une base de données relationnelle permet de mieux organiser les données, d'améliorer leur intégrité et de faciliter leur gestion.

3.1 Intégrité d'une base de donnée

Définition 4 (Contrainte d'intégrité)

Une **contrainte d'intégrité** est une règle qui définit la cohérence d'une donnée ou d'un ensemble de données d'une base de donnée.



Contraintes d'intégrité principales

- La **contrainte de domaine** : chaque attribut doit prendre une valeur dans le domaine de valeurs (entier, flottant etc...).
- La **contrainte d'unicité** : La valeur d'une clé primaire ne doit apparaître qu'une fois dans une table. Elle doit être unique et non nulle.
- La **contrainte d'intégrité référentielle** : Toutes les valeurs d'une clé étrangère d'une table doit correspondre à une valeur existante de la table à laquelle elle fait référence.

Quand les contraintes d'intégrités ne sont pas respectées lors de l'ajout ou de la suppression d'un enregistrement il faut donc donner une règle qui permette de garder l'intégrité. On retiendra les cas suivants :

- PRIMARY KEY : définit l'attribut comme la clé primaire
- UNIQUE : interdit que deux tuples de la relation aient la même valeur pour l'attribut.
- REFERENCES <nom table> (<nom colonnes>) : contrôle l'intégrité référentielle entre l'attribut et la table et ses colonnes spécifiées
- CHECK (<condition>) : contrôle la validité de la valeur de l'attribut spécifié dans la condition dans le cadre d'une restriction de domaine
- FOREIGN KEY (<liste d'attributs>) REFERENCES <nom table>(<nom colonnes>) : contrôle l'intégrité référentielle entre les attributs de la liste et la table et ses colonnes spécifiées

Exemple en SQL

```
CREATE TABLE Personne (
  NoSS CHAR(13) PRIMARY KEY,
  Nom VARCHAR(25) NOT NULL,
  Prenom VARCHAR(25) NOT NULL,
  VilleNaissance CHAR(13),
  Pseudo VARCHAR(25) NOT NULL,
  Age INTEGER(3) CHECK (Age BETWEEN 18 AND 65),
  Mariage CHAR(13) REFERENCES Personne(NoSS),
  FOREIGN KEY VilleNaissance REFERENCES Villes (nom),
  UNIQUE (Pseudo)
);
```

4 Les opérations sur les tables

4.1 Le produit cartésien

Définition 5

Le **produit cartésien** de deux ensembles X et Y , appelé ensemble-produit et noté $X \times Y$, est l'ensemble de tous les couples dont la première composante appartient à X et la seconde à Y .



Exemple

Soit $X = \{1, 2, 3\}$ et $Y = \{4, 5, 6\}$ alors $X \times Y = \{(1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6)\}$



Exemple

Le **produit cartésien** des tables **film** et **realisateur** ressemble à cela :

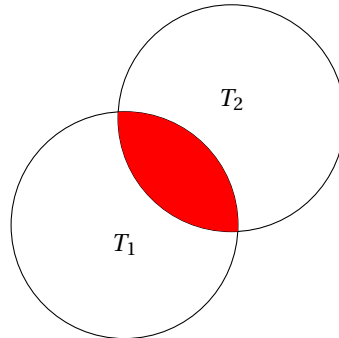
film × réalisateur								
idFilm	titre	année	genre	IdReal	idReal	nom	prénom	nationalité
1	"La Guerre des étoiles"	1977	"Aventure"	2	1	"Capra"	"Frank"	"US"
1	"La Guerre des étoiles"	1977	"Aventure"	2	4	"Francis"	"Ford Coppola"	"US"
1	"La Guerre des étoiles"	1977	"Aventure"	2	3	"Eastwood"	"Clint"	"US"
1	"La Guerre des étoiles"	1977	"Aventure"	2	2	"Lucas"	"Georges"	"US"
1	"La Guerre des étoiles"	1977	"Aventure"	2	5	"Benigni "	"Roberto "	"IT"
2	"Apocalypse Now"	1979	"Drame"	4	1	"Capra"	"Frank"	"US"
2	"Apocalypse Now"	1979	"Drame"	4	4	"Francis"	"Ford Coppola"	"US"
2	"Apocalypse Now"	1979	"Drame"	4	3	"Eastwood"	"Clint"	"US"
2	"Apocalypse Now"	1979	"Drame"	4	2	"Lucas"	"Georges"	"US"
2	"Apocalypse Now"	1979	"Drame"	4	5	"Benigni "	"Roberto "	"IT"
3	"Impitoyable"	1992	"Western"	3	1	"Capra"	"Frank"	"US"
3	"Impitoyable"	1992	"Western"	3	4	"Francis"	"Ford Coppola"	"US"
3	"Impitoyable"	1992	"Western"	3	3	"Eastwood"	"Clint"	"US"
3	"Impitoyable"	1992	"Western"	3	2	"Lucas"	"Georges"	"US"
3	"Impitoyable"	1992	"Western"	3	5	"Benigni "	"Roberto "	"IT"
4	"La vie est belle"	1997	"Drame"	5	1	"Capra"	"Frank"	"US"
4	"La vie est belle"	1997	"Drame"	5	4	"Francis"	"Ford Coppola"	"US"
4	"La vie est belle"	1997	"Drame"	5	3	"Eastwood"	"Clint"	"US"
4	"La vie est belle"	1997	"Drame"	5	2	"Lucas"	"Georges"	"US"
4	"La vie est belle"	1997	"Drame"	5	5	"Benigni "	"Roberto "	"IT"
6	"Le parrain"	1972	"Drame"	4	1	"Capra"	"Frank"	"US"
...

Cette table n'a pas de sens concret mais il faut l'avoir en tête quand on fait une requête.

4.2 La jointure interne

Définition 6

Soient deux tables T_1 et T_2 . La *jointure interne* de ses deux tables sur un attribut donné A est une table formée en ne prenant que les enregistrements de T_1 et T_2 qui ont une valeur de A commune.



Exemple

La jointure interne des tables film et realisateur pour l'attribut idReal

film × realisateur								
idFilm	titre	année	genre	IdReal	idReal	nom	prénom	nationalité
1	"La Guerre des étoiles"	1977	"Aventure"	2	2	"Lucas"	"Georges"	"US"
2	"Apocalypse Now"	1979	"Drame"	4	4	"Francis"	"Ford Coppola"	"US"
3	"Impitoyable"	1992	"Western"	3	3	"Eastwood"	"Clint"	"US"
4	"La vie est belle"	1997	"Drame"	5	5	"Benigni"	"Roberto"	"IT"
6	"Le parrain"	1972	"Drame"	4	4	"Francis"	"Ford Coppola"	"US"
...

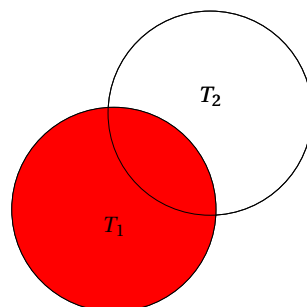
Cette table a plus de sens que la première mais il y a deux colonnes 'idReal', il faudra avoir cela en-tête pour la suite. On peut avoir besoin de distinguer les deux colonnes.

4.3 La jointure externe

Définition 7

Soient X et Y , deux ensembles. La **jointure externe gauche** pour une condition A , est l'ensemble de tous les couples $(x, y) \in X \times Y$ pour la condition A et (x, \emptyset) avec $x \in X$ quand la condition A n'est pas vérifiée.

$$X \times_{\text{gauche}} Y = \{(x, y) \mid (x, y) \in X \times Y \text{ et } A \text{ vraie}\} \cup \{(x, \emptyset) \mid x \in X \text{ et } A \text{ fausse}\}$$



**Exemple**

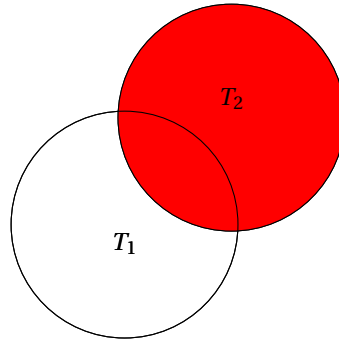
Soit $X = \{1, 2, 3, 4, 5, 6\}$ et $Y = \{1, 4, 6, 9\}$ alors la jointure gauche de X et Y pour $A = "x = y, (x, y) \in X \times Y"$ est

$$X \times_{\text{gauche}} Y = \{(1, 1), (4, 4), (6, 6), (2, \emptyset), (3, \emptyset), (5, \emptyset)\}$$

Définition 8

Soient X et Y , deux ensembles. La **jointure externe droite** pour une condition A , est l'ensemble de tous les couples $(x, y) \in X \times Y$ pour la condition A et (\emptyset, y) avec $y \in Y$ quand la condition A n'est pas vérifiée.

$$X \times_{\text{droite}} Y = \{(x, y) \mid (x, y) \in X \times Y \text{ et } A \text{ vraie}\} \cup \{(\emptyset, y) \mid y \in Y \text{ et } A \text{ fausse}\}$$

**Exemple**

Soit $X = \{1, 2, 3, 4, 5, 6\}$ et $Y = \{1, 4, 6, 9\}$ alors la jointure droite de X et Y pour $A = "x^2 = y, (x, y) \in X \times Y"$ est

$$X \times_{\text{droite}} Y = \{(1, 1), (2, 4), (3, 9), (\emptyset, 6)\}$$

**Exemple**

Rajoutons un enregistrement dans la table *realisateur*

realisateur			
idReal	nom	prénom	nationalité
1	"Capra"	"Frank"	"US"
4	"Francis"	"Ford Coppola"	"US"
3	"Eastwood"	"Clint"	"US"
2	"Lucas"	"Georges"	"US"
5	"Benigni "	"Roberto "	"IT"
77	"Courtois"	"Jerome"	"FR"
...

La **jointure externe droite** des tables film et realisateur pour l'attribut idReal

film \times_{droite} realisateur								
idFilm	titre	année	genre	IdReal	idReal	nom	prénom	nationalité
1	"La Guerre des étoiles"	1977	"Aventure"	2	2	"Lucas"	"Georges"	"US"
2	"Apocalypse Now"	1979	"Drame"	4	4	"Francis"	"Ford Coppola"	"US"
3	"Impitoyable"	1992	"Western"	3	3	"Eastwood"	"Clint"	"US"
4	"La vie est belle"	1997	"Drame"	5	5	"Benigni "	"Roberto "	"IT"
6	"Le parrain"	1972	"Drame"	4	4	"Francis"	"Ford Coppola"	"US"
NULL	NULL	NULL	NULL	NULL	77	"Courtois"	"Jerome"	"FR"
...