

# Bases de données et modèle relationnel

L'ensemble du chapitre est fortement influencé par :

- *Numérique et Sciences Informatique Tle : 24 leçons avec exercices corrigés*, Balabonski, Conchon, Filliâtre, Nguyen, éditions Ellipses
- *Bases de données, de la modélisation au SQL*, Laurent Audibert, éditions Ellipses

## 1. Introduction aux bases de données, types de modèles

### 1.1. Les bases de données

Il est difficile de proposer une définition exacte des bases de données. *Laurent Audibert*<sup>2</sup> propose :

#### Définition générale

Un ensemble organisé d'informations avec un objectif commun.

La définition est vague, mais dès lors que des informations sont **rassemblées** et **stockées** d'une manière **organisée**, on peut parler de *bases de données*, quel que soit le support utilisé (tablettes d'argile, papiers, fichiers, etc...)

Bien entendu l'informatique, c'est-à-dire la science du traitement automatique de l'information, permet le traitement de grandes quantités d'informations, et surtout leur exploitation (ajout, suppression, mise à jour, recherches, ...). Dans le cadre informatique, *Laurent Audibert*<sup>2</sup> propose :

#### Base de données informatisées

Une base de données (**BDD**, en anglais *DataBase* soit **DB**) informatisée est un ensemble structuré de données enregistrées sur des supports accessibles par l'ordinateur, représentant des informations du monde réel et pouvant être interrogées et mise à jour par une communauté d'utilisateurs.

Nous verrons que cette définition impose un certain nombre de contraintes à la mise en place des BDD informatisées. La première est la question de la **cohérence des données**.

## Exemple : Bases de données musicales<sup>2</sup>

Prenons l'exemple d'une base de données contenant des albums musicaux, et qui mémorise pour chaque album son genre, son titre, et le nom du groupe ainsi que le nom du chanteur principal. L'idée de base est de présenter les données sous la forme d'une table, comme dans l'extrait ci-dessous :

Genre	Titre	Groupe	Chanteur
Stoner	Era Vulgaris	Queen Of The Stone Age	Josh Homme
Metal	Master of Puppets	Metallica	
Metal	Them Crooked Vultures	Them Crooked Vultures	Josh Homme
Metal	L'Enfant sauvage	GOJIRA	Jo Duplantier
Stoner	Dying surfer meet his maker	All Them Withches	Charles Michael Parks Jr
Métal	Fortitude	Gojira	Joseph Duplantier
Stoner Rock	Nothing as the ideal	All Them Withches	Charles Michael Parks Jr
Metal Progressif	Magma	Gojira	Joseph DUPLANTIER

La condition *sine qua none* pour garantir la faisabilité et la pertinence d'une recherche de données est la **cohérence de ces données**. Ici on constate plusieurs problèmes de cohérence :

- Le nom du groupe *Gojira* est orthographié de deux manières différentes. Une recherche utilisant la chaîne de caractères `Gojira` ne donnera que deux résultats, alors que trois peuvent être attendus.
- Le genre *Metal* est aussi orthographié de deux manières différentes. On notera qu'une recherche stricte avec la chaîne `Metal`, c'est-à-dire basée sur l'égalité, ne donnera que deux résultats, alors qu'une recherche étendue, c'est-à-dire basée sur l'appartenance de la sous-chaîne `Metal` à la chaîne donnant le genre, renverra elle trois résultats.
- Le nom du chanteur de *Gojira* est écrit de trois manières différentes.
- On peut aussi noter que le nom du chanteur de *Metallica* n'est pas renseigné, ce qui peut-être poser un problème de cohérence.

Ces problèmes de cohérence sont liés (en partie), à un problème de **redondance** des données : en effet dans la table telle qu'elle est présentée, on va devoir saisir plusieurs fois les chaînes telles que `Metal` et `Gojira`, alors qu'elles ne représentent bien qu'une seule *entité* chacune.

Pour régler la question de la **cohérence des données**, il faudra en premier lieu trouver un moyen d'éviter la **redondance** de certaines données.

## 1.2. Historique et problématiques des différents modèles

### Modèles de données et structures de données

Ces deux concepts se placent à différents niveaux d'abstraction. Les **modèles de données** indique quelles caractéristiques d'une entité réelle on souhaite manipuler dans un programme, ainsi que les relations qui lient ces entités entre elles. Par exemple un groupe de musique va produire plusieurs albums, et un chanteur peut chanter dans plusieurs groupes.

Une **structure de donnée** indique la manière dont on va organiser les données **en machine**. Ainsi pour un même modèle il existe plusieurs structures pouvant représenter ce modèle (tableau, listes chaînées, objets...)

Le premier modèle ayant souhaité s'attaquer au problème de la redondance des données est le **modèle hiérarchique**, développé par IBM dans les années 1960, dans le cadre du programme Apollo de la NASA. Ce modèle, très dépendant de la manière dont fonctionnait les ordinateurs de cette époque, enregistre les données à la manière de l'organisation d'un système de fichier, sous la forme d'un arbre (*structure arborescente*), de façon à ce qu'un enregistrement n'ait qu'un seul possesseur. On peut en donner un exemple ci-dessous :



ce type de modèle peut être pratique, mais *il ne répond pas à toutes les problématiques*. En effet, ici nous n'avons pas introduit dans la hiérarchie les leads vocaux. Voici ce que donne le schéma lorsqu'on les introduits :



On constate ici que le choix d'organisation de l'arbre ne permet pas de régler la redondance des données, puisque non seulement un même chanteur peut être lead vocal sur plusieurs albums, mais il peut l'être aussi sur **plusieurs groupes** (comme Josh Homme par exemple). La structure hiérarchique n'est ici pas adaptée.

Une des solutions proposée pour ce problème a été le **modèle réseau**, proposé par Charles William Bachman à la fin des années 1960<sup>2</sup>. C.W. Bachman reçu le Prix Turing en 1973 pour ses contributions sur les technologies des bases de données. On lui doit aussi le modèle OSI (voir le cours de première...). Une représentation de nos albums dans le modèle réseau est la suivante :



Ainsi, la redondance des données n'est plus un problème. Cependant, le modèle réseau n'est pas un modèle toujours efficace, puisque pour retrouver une donnée (un chanteur par exemple), il faut connaître la structure de ce réseau, c'est-à-dire les liaisons existantes, ce qui rend les programmeurs **dépendants de la structure de donnée**.

## 2. Le modèle relationnel

### 2.1. Définition

### 2.2. Modélisation relationnelles des données

### 2.3. Les différentes contraintes

---

1. *Numérique et Sciences Informatique Tle : 24 leçons avec exercices corrigés*, Balabonski, Conchon, Filliâtre, Nguyen, éditions Ellipses ↩

2. *Bases de données, de la modélisation au SQL*, Laurent Audibert, éditions Ellipses ↩ ↩ ↩ ↩