

# Chapitre 7 - Les bases de données relationnelles

## Objectifs :

- ▷ Identifier les concepts définissant le modèle relationnel.
- ▷ Maîtriser les définitions de : relation, attribut, domaine, clé primaire, clé étrangère, schéma relationnel
- ▷ Exprimer les contraintes d'intégrité d'une base de données
- ▷ Savoir réaliser un schéma relationnel
- ▷ Savoir repérer les anomalies dans le schéma relationnel d'une base de données

## 1 Introduction

Les bases de données sont aujourd'hui incontournables, car très peu d'informations restent encore stockées sur du papier. Nous avons vu en classe de Première comment utiliser des données comme celle d'un tableur avec notamment l'utilisation du format `csv`.

En effet, prenons l'exemple d'une table contenant quelques films :

Titre	Réalisateur	Année	Durée
2001 : A Space Odyssey	Stanley Kubrick	1968	149
Alien	Ridley Scott	1979	117
Blade Runner	Ridley Scott	1982	117
Akira	Katsuhiro Ôtomo	1988	124

- Comment peut-on faire si on veut rajouter des informations sur le réalisateur, comme sa nationalité, sa date de naissance ou un deuxième prénom pour le distinguer d'un homonyme ?
- Faut-il le rajouter sur chacun de ses films ?
- Faut-il ajouter de nouvelles entrées ?
- Comment s'y prendre pour trouver facilement tous les films d'un même réalisateur ?

Tous ces problèmes sont démultipliés si on veut rajouter par exemple les scénaristes, les acteurs ou le genre du film. L'utilisation de simples tableaux a donc ses limites.

Pour apporter une solution, on sépare les données en plusieurs tables qui ont un lien (**une relation**) les unes avec les autres. C'est ce que l'on appelle une **base de données relationnelle**. Cela a été théorisé en 1970 par le Britannique *Edgar F. Codd*. Il recevra le *prix Turing* en 1981 pour son travail.

## 2 Définition

### A retenir !

Une **base de données** est un moyen de **stocker des données de manière structurée**, accessibles en lecture et en écriture à l'aide d'un langage.

Les bases de données peuvent être divisées en deux catégories :

- ▷ Les **bases données relationnelles** utilisent des tableaux de données organisés en lignes et colonnes auxquelles on accède grâce au langage **SQL** (pour *Structured Query Language*).
- ▷ Les **bases de données non relationnelles** dites *noSQL* qui peuvent stocker des données de formes diverses : graphes, documents, ..., etc. (hors programme de Terminale)

### 3 Le modèle relationnel

Dans une base de données, plutôt que de stocker toutes les données dans une table, on les stocke dans plusieurs tables reliées entre elles par des clés.

Le **modèle relationnel** créé par *E. F. Codd* en 1970, consiste à voir chaque **table** comme une **relation** sur les données qui la constituent.

Ainsi, dans notre exemple, on va définir une relation (table) pour les films et une autre pour les réalisateurs.

Film	Titre	Réal	Année	Durée
1	2001 : A Space Odyssey	1	1968	149
2	Alien	2	1979	117
3	Blade Runner	2	1982	117
4	Akira	3	1988	124

Réal	Nom
1	Stanley Kubrick
2	Ridley Scott
3	Katsuhiro Ôtomo

#### Les principes de base du modèle relationnel

- ▷ Les données sont regroupées dans différentes **tables** (qu'on appellera plutôt **relations** et qui donnent son nom au modèle). Chaque relation contient des éléments directement en lien avec le sujet général de la table.
- ▷ Autant que possible, des données identiques ne doivent pas se trouver dans des tables différentes : on évite la **redondance** des données.
- ▷ Les données ne doivent pas contenir elles-mêmes d'autres données : on parle d'**atomicité** des données.

### 4 Vocabulaire

#### A retenir !

- ▷ **Relation, ou table** : Tableau à 2 dimensions contenant les données de type texte ou numérique. C'est l'endroit où sont rangées les données. L'ordre des lignes (que l'on appellera des **enregistrements**) n'a pas d'importance.
- ▷ **Enregistrement, ou tuple, ou t-uplet** : cela correspond à une ligne du tableau, et donc un ensemble de valeurs liées entre elles.
- ▷ **En-tête** : c'est l'ensemble des titres des attributs (première ligne de la table).
- ▷ **Attribut** : la table est composée de colonnes qui ont des titres que l'on appelle des attributs. Dans une même table, chaque colonne devra avoir un nom différent.
- ▷ **Domaine** : le domaine d'un attribut donné correspond à un ensemble fini ou infini de valeurs admissibles. Le domaine désigne «le type» (au sens type **Int**, **Float**, **String**).

## 5 Exemple

Prenons l'exemple d'une bibliothèque dont la base de données possède une relation **Livres** suivante :

code	Titre	Auteur	Éditeur	ISBN
834	Tous les hommes [...]	Jean-Paul DUBOIS	l'Olivier	978-2823615166
720	Leurs enfants après eux	Nicolas MATHIEU	Actes Sud	978-2330108717
37	L'Ordre du jour	Eric VUILLARD	Actes Sud	978-2330078973
1563	Chanson douce	Leïla SLIMANI	Gallimard	978-2070196678
486	Boussole	Mathias ENARD	Actes Sud	978-2330081492
1023	Pas pleurer	Lydie SALVAYRE	Seuil	978-2757884003
942	Au revoir là-haut	Pierre LEMAITRE	Albin-Michel	978-2253194613
152	Le sermon sur la chute de Rome	Jérôme FERRARI	Actes Sud	978-2330022808
46	L'Art français de la guerre	Alexis JENNI	Gallimard	978-2070134588
1532	La Carte et le Territoire	Michel HOUELLEBECQ	Flammarion	978-2081246331

Dans notre exemple, nous avons l'**entête** (en bleu), l'**enregistrement** (en rouge) et l'**attribut** (en vert).

Ainsi :

- ▷ Il y a dans notre relation un attribut «**Titre**», un attribut «**Auteur**», ..., etc.
- ▷ L'auteur «**Eric VUILLARD**» a bien écrit le livre «**L'Ordre du jour**». Il est interdit que deux enregistrements soient totalement identiques.
- ▷ L'attribut «**Éditeur**» est une chaîne de caractères, son domaine est donc **String**.
- ▷ Par contre l'attribut «**ISBN**» est un nombre de 13 chiffres, commençant manifestement toujours par 978. Son domaine est donc **Int**.
- ▷ Ici notre schéma serait :

```
((Code, Entier),(Titre, Chaîne de caractères),(Auteur, Chaîne de caractères),(Éditeur,
Chaîne de caractères),(ISBN, Entier))
```

Consulter un résumé en vidéo [ici](#).

## 6 Notions de clé

Afin de comprendre le concept de clé dans une base de donnée relationnelle, consulter cette [vidéo](#).

### 6.1 Clé primaire

#### A retenir !

Une **clé primaire** est un attribut (ou une réunion d'attributs) dont la connaissance suffit à identifier avec certitude un unique enregistrement.

Dans l'exemple de la relation «**Livre**» traité précédemment, l'attribut «**ISBN**» pourrait jouer le rôle de clé primaire car il **identifie de manière unique** n'importe quel ouvrage.

Dans le cas d'une relation contenant des enregistrements de personnes nées en France, la clé primaire pourrait être le **numéro de Sécurité Sociale**.

## 6.2 Clé étrangère

Dans notre exemple, ajoutons maintenant les relations ci-dessous :

▷ Relation «Emprunts»

id_emprunteur	Date	Nom	Prénom	Titre	Auteur	Code
845	12/10/2020	DURAND	Michel	Au revoir là-haut	Pierre LEMAITRE	942
125	13/10/2020	MARTIN	Jean	Pas pleurer	Lydie SALVAYRE	1023
125	13/10/2020	MARTIN	Jean	Boussole	Mathias ENARD	486

▷ Relation «Emprunteurs»

id_emprunteur	Nom	Prénom	Adresse	Téléphone
1	Dupont	Jean	123 Rue de la Libération	01-234-5678
2	Martin	Marie	456 Avenue des Roses	02-345-6789
3	Dubois	Pierre	789 Boulevard de l'Étoile	03-456-7890

L'attribut «id\_emprunteur» est une clé primaire de la relation «Emprunteurs».

«id\_emprunteur» est bien une clé primaire (de la relation «Emprunteurs») mais ne peut pas être une clé primaire de la relation «Emprunts», car une personne peut prendre plusieurs livres à la fois.

On dit alors que c'est une **clé étrangère**.

**A retenir !**

Une clé étrangère est une clé primaire d'une autre relation.

Consulter un résumé en vidéo [ici](#).

## 7 Schéma et diagramme relationnel

### 7.1 Schéma relationnel

**A retenir !**

Le schéma d'une relation est le regroupement de tous les attributs et de leur domaine respectif. On appelle **schéma relationnel** l'ensemble des relations présentes dans une base de données.

Quand on vous demande le schéma relationnel d'une base de données, il est nécessaire de fournir les informations suivantes :

- ▷ Les **noms des différentes relations**
- ▷ Pour chaque relation, la **liste des attributs** avec leur **domaine** respectif
- ▷ Pour chaque relation, la et éventuellement la ou les **clés étrangères**

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

Par exemple, considérons la base de données *Tour de France 2020*, contenant les relations suivantes :

codeEquipe	nomEquipe
ALM	AG2R La Mondiale
AST	Astana Pro Team
TBM	Bahrain - McLaren
BOH	BORA - hansgrohe
CCC	CCC Team
COF	Cofidis, Solutions Crédits
DQT	Deceuninck - Quick Step
EF1	EF Pro Cycling
GFC	Groupama - FDJ
LTS	Lotto Soudal

Le schéma relationnel de cette table sera fréquemment donné sous cette forme :

Equipes ( codeEquipe String, nomEquipe String )

## 7.2 Diagramme relationnel

Souvent, on présentera l'ensemble des renseignements d'un modèle relationnel sous forme d'un **diagramme** qui synthétise la composition des différentes tables et les relations entre elles.

On pourra utiliser l'outil en ligne **QuickDBD** pour créer facilement des schémas relationnels/ Pour cela, on utilise une syntaxe simple pour saisir les attributs, leurs domaines, les clés primaires et étrangères et les relations.

## 8 Les trois contraintes d'intégrité

Dans une base de données, il faut garantir que les données stockées soient :

- **précises**,
- **cohérentes**
- et **conformes** aux exigences du système,

C'est pour cela que l'on a défini des règles que l'on appelle les **contraintes d'intégrité**, à savoir :

### Contrainte de domaine

Tout attribut d'un enregistrement doit respecter le domaine indiqué dans le schéma relationnel.

Attention, certains domaines sont subtils. Par exemple, si une relation possède un attribut "Code Postal", le domaine de cet attribut devra être **String** plutôt que **Entier** . Dans le cas contraire, un enregistrement possédant le code postal 03150 serait converti en 3150 (car pour les entiers, 03150 = 3150). Or le code postal 3150 n'existe pas.

### Contrainte de relation

Tout enregistrement est unique. Cette contrainte est assurée par l'existence obligatoire d'une clé primaire.

### Contrainte de référence

La cohérence entre les différentes tables d'une base de données est assurée par les clés étrangères : dans une table, la valeur d'un attribut qui est clé étrangère doit obligatoirement pouvoir être retrouvée dans la table dont cet attribut est clé primaire.

Pour en savoir plus : <https://www.youtube.com/watch?v=bhtzximjwk8>

## 9 Exercices

**Exercice 1 :** Deux relations modélisent la flotte de voitures d'un réseau de location de voitures.

TABLE 1 – Relation Agences

id_agence	ville	département
1	Paris	75
2	Lyon	69
3	Marseille	13
4	Aubagne	13

TABLE 2 – Relation Voitures

id_voiture	marque	modèle	kilométrage	couleur	id_agence
1	Renault	Clio	12000	Rouge	2
2	Peugeot	205	22000	Noir	3
3	Toyota	Yaris	33000	Noir	3

1. Combien la relation **Voitures** comporte-t-elle d'attributs ?
2. Que vaut son cardinal ?
3. Quel est le domaine de l'attribut **id\_agence** dans la relation **Voitures** ?
4. Quel est le schéma relationnel de la relation **Agences** ?
5. Quelle est la clé primaire de la relation **Agences** ?
6. Quelle est la clé primaire de la relation **Voitures** ?
7. Quelle est la clé étrangère de la relation **Voitures** ?

**Exercice 2 :** Un commerçant utilise plusieurs fichiers pour gérer ses produits. On considère un fichier destiné à gérer des produits frais. Le tableau présenté est un extrait du contenu de ce fichier. Les quatre colonnes contiennent respectivement un identifiant numérique, le nom du produit, son prix et la marque qui le commercialise. Les mêmes noms de marques peuvent apparaître de nombreuses fois dans la colonne marque mais aussi dans les fichiers correspondant à d'autres types de produits.

id	nom	prix	marque
17	Yaourt6	2,52	Yopnone
21	Yaourt12	4,93	Dalait
25	Beurre250	2,27	Croisement
28	Crème50	2,74	Dalait
31	Crème70	3,79	Yopnone

1. A partir de ce fichier, construire une relation **Frais** (pour les produits frais) et une relation **Marques** suivant le modèle relationnel permettant d'éviter la redondance d'informations.
2. Indiquer une clé primaire pour chacune des deux tables et préciser la clé étrangère.

**Exercice 3 :** Concernant les vols qui sont prévus dans un aéroport pendant une journée, on dispose de données, écrites dans un tableur. Nous avons le numéro du vol, les heures de départ et d'arrivée, la provenance pour les vols à l'arrivée, la destination pour les vols au départ, le type d'avion et sa capacité totale de passagers. Chaque vol a un numéro unique.

Extrait du tableau :

vol	hd	ha	provenance	destination	avion	capacité
AF373	8h45	10h05	Toulouse, France	Paris, France	Airbus A320	150
LX529	11h45	12h50	Nice, France	Genève, Suisse	Boeing 747	424

Décrire les 6 relations (**Départs**, **Arrivées**, **Villes**, **Pays**, **Avions**, **Constructeurs**) construites à partir de ce fichier permettant de satisfaire au modèle relationnel.

**Exercice 4 :** Un particulier a un grand nombre de chansons stockées sur son ordinateur.

Il tient à jour un fichier qui contient toutes les chansons enregistrées par des groupes.

Ce fichier contient quatre colonnes où sont notés respectivement le titre de la chanson, le groupe qui l'a enregistré, les membres du groupe et la date d'enregistrement.

Voici une ligne de ce fichier :

titre	groupe	membres	année
Roxane	The Police	Sting, Summers, Copeland	1978

Constituer un modèle relationnel normalisé à l'aide de trois tables nommées : **Chansons**, **Goupes** et **Artistes**.

**Exercice 5 :** On souhaite modéliser un annuaire téléphonique simple dans lequel chaque personne (identifiée par son nom et son prénom) est associée à son numéro de téléphone. Proposer une modélisation relationnelle de cet annuaire.

Vérifier si chacun des ensembles ci-dessous est une relation valide pour la relation **Annuaire** :

1. {}
2. {( 'Titi', 'Toto', '12345678' )}
3. {( 'Titi', 'Toto', '12345678' ), ( 'Dodo', 'Didi', '12345678' )}
4. {( 'Titi', 'Toto', '12345678' ), ( 'Titi', 'Toto', '+332345678674' )}
5. {( 'Titi', 'Toto', '12345678' ), ( 'Dodo', 'Didi' )}
6. {( 'Titi', 'Toto', 5678 )}

**Exercice 6 :** Reprenons la base Tour de France 2020 vue en cours :

TABLE 3 – Relation «Équipes»

codeEquipe	nomEquipe
ALM	AG2R La Mondiale
AST	Astana Pro Team
TBM	Bahrain - McLaren
BOH	BORA - hansgrohe
CCC	CCC Team
COF	Cofidis, Solutions Crédits
DQT	Deceuninck - Quick Step
EF1	EF Pro Cycling
GFC	Groupama - FDJ
LTS	Lotto Soudal

1. Quel temps a réalisé Guillaume MARTIN sur l'étape Sisteron / Orcières-Merlette ?
2. À l'arrivée à Privas, qui est arrivé en premier entre Primož ROGLIČ et Simone CONSONNI ?

TABLE 4 – Relation «Coureurs»

dossard	nomCoureur	prénomCoureur	codeEquipe
141	LÓPEZ	Miguel Ángel	AST
142	FRAILE	Omar	AST
143	HOULE	Hugo	AST
11	ROGLIČ	Primož	TJV
12	BENNETT	George	TJV
41	ALAPHILIPPE	Julian	DQT
44	CAVAGNA	Rémi	DQT
45	DECLERCQ	Tim	DQT
121	MARTIN	Guillaume	COF
122	CONSONNI	Simone	COF
123	EDET	Nicolas	COF

TABLE 5 – Relation «Étapes»

numéroEtape	villeDépart	villeArrivée	km
1	Nice	Nice	156
2	Nice	Nice	185
3	Nice	Sisteron	198
4	Sisteron	Orcières-Merlette	160
5	Gap	Privas	198

TABLE 6 – Relation «Temps»

dossard	numéroEtape	tempsRéalisé
41	2	04 :55 :27
121	4	04 :07 :47
11	5	04 :21 :22
122	5	04 :21 :22