

## CH2 BASES DE DONNEES : LE MODELE RELATIONNEL

Dans la vie courante, les entreprises, l'état, les associations... sont amenés à gérer des quantités de données qui sont difficilement exploitables sous *format* CSV soit de par leur taille trop conséquente soit par leur structure même qui comporte des relations entre différentes données.

Il existe un outil complètement adapté à ce type de données, les **bases de données relationnelles**.



exemple de base : une bibliothèque

#	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generation	Legendary
1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False
2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False
3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False
3	VenusaurMega Venusaur	Grass	Poison	625	80	100	123	122	120	80	1	False
4	Charmander	Fire		309	39	52	43	60	50	65	1	False
5	Charmeleon	Fire		405	58	64	58	80	65	80	1	False
6	Charizard	Fire	Flying	534	78	84	78	109	85	100	1	False
6	CharizardMega Charizard X	Fire	Dragon	634	78	130	111	130	85	100	1	False
6	CharizardMega Charizard Y	Fire	Flying	634	78	104	78	159	115	100	1	False
7	Squirtle	Water		314	44	48	65	50	64	43	1	False
8	Wartortle	Water		405	59	63	80	65	80	58	1	False
9	Blastoise	Water		530	79	83	100	85	105	78	1	False
9	BlastoiseMega Blastoise	Water		630	79	103	120	135	115	78	1	False
10	Caterpie	Bug		195	45	30	35	20	20	45	1	False
11	Metapod	Bug		205	50	20	55	25	25	30	1	False

exemple de base : un classeur csv

**Note :** il existe plusieurs types de bases de données : Hiérarchiques, relationnelles... Ce sont aussi les plus répandues. Dans les bases de données relationnelles, les données sont stockées sous forme de table.

### 1. Les bases de données

Ces bases de données ont vraiment démarré avec l'émergence d'internet et l'apparition du big data, gigantesque collecte de données qu'il faut stocker, modifier, traiter.

D'où les deux aspects :

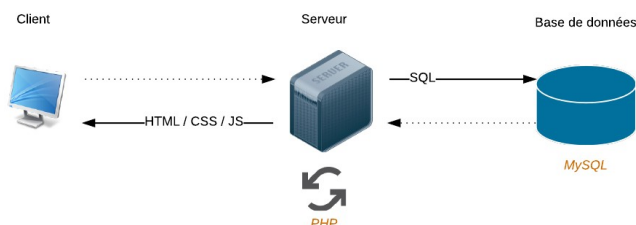
- la **base** en tant que telle, qui contient les données à stocker et répertorier.
- le **SGBD**, ou *Système de Gestion de Bases de Données*.

#### A retenir :

Une **base de données (BD)** est ensemble d'informations structurées mémorisées sur un support persistant.

Un **Système de Gestion de Base de Données (SGBD)** est un système informatique qui assure la gestion de l'ensemble des informations stockées dans une base de données. Il prend en charge, notamment, les deux grandes fonctionnalités :

- Accès aux fichiers** de la base, garantissant leur intégrité, contrôlant les opérations concurrentes, optimisant les recherches et mises à jour.
- Interactions** avec les applications et utilisateurs, grâce à des langages d'interrogation et de manipulation spécifiques. (LMD, ou *Langage de Manipulation de Données*)



SGBD en langage PHP sur un serveur, pour une base de données MySQL

Le SGBD doit permettre d'assurer les fonctions suivantes :

- l'écriture, la lecture et la modification des BD.
- l'indépendance entre le traitement des données et les données directement perceptibles.
- la non-redondance des données.
- le partage des données : elles assurent la possibilité à plusieurs utilisateurs l'accès simultané à la BD.
- la sécurisation de la BD et les niveaux d'accès des utilisateurs : de simples consultants à administrateur.
- la persistance du système en cas de panne. Stockage sur plusieurs supports, chaque modification devant être répercutée sur tous les supports.

### Exemples :

- Voir différents formats sur Pour les bases, on trouvera les formats usuels comme *MySQL, PostgreSQL, SQLite, Oracle Database, Microsoft SQL Server, ...*
- Les SGBD sont également variés ; on s'intéressera au langage généralement utilisé, le *SQL*.
- Le format JSON

```
{
  "squadName": "Super hero squad",
  "homeTown": "Metro City",
  "formed": 2016,
  "secretBase": "Super tower",
  "active": true,
  "members": [
    {
      "name": "Molecule Man",
      "age": 29,
      "secretIdentity": "Dan Jukes",
      "powers": [
        "Radiation resistance",
        "Turning tiny",
        "Radiation blast"
      ]
    },
    {
      "name": "Madame Uppercut",
      "age": 39,
      "secretIdentity": "Jane Wilson",
      "powers": [
        "Million tonne punch",
        "Damage resistance",
        "Superhuman reflexes"
      ]
    },
    {
      "name": "Eternal Flame",
      "age": 1000000,
      "secretIdentity": "Unknown",
      "powers": [
        "Immortality",
        "Heat Immunity",
        "Inferno",
        "Teleportation",
        "Interdimensional travel"
      ]
    }
  ]
}
```

## 2. Structure d'une base de données

### (a) Une façon de présenter les choses

Sur cet exemple, on voit la structure de la **table Livres** d'une base de données ainsi que le vocabulaire associé :

Schéma de la relation	Livres (Auteur, Titre, Année, Pays, Genre)					Entête
	Auteur	Titre	Année	Pays	Genre	
Extension de la relation	Hugo	Hernani	1830	France	Théâtre	Ligne – enregistrement – N-Uplet
	King	Nuit noire, étoile morte	2010	USA	S.F.	
	King	Misery	1999	USA	S.F.	
	De Vigan	No et Moi	2010	France	Roman	
	Claudiel	Le soulier de Satin	1929	France	Littérature	
	Claudiel	L'enquête	2010	France	S.F.	
	Bertholon	Twist	2010	France	Thriller	
	Jackson	Vengeance	2018	USA	Thriller	
	Simmons	Vengeance	2003	USA	Thriller	
Colonne – Attribut (nom et type)			Valeur			

Les **attributs** sont donc, avec leurs types respectifs :

Auteur (str), Titre(str), Année (int), Pays(str), Genre(str)

Le **tuple** (ou *n-uplet*) surligné est donc :

('Claudiel', 'Le soulier de satin', 1929, 'France', 'Littérature')

Pour cette donnée, la **valeur** de l'attribut « Année » est donc 1929.

#### A retenir :

Une **table** est un tableau contenant un ensemble de valeurs réparties en colonnes (rubriques ou libellés)

L'**attribut** est le nom que l'on donne à une rubrique (Auteur, Année, ...)

Le **tuple** (ou *n-uplet*) est une ligne dans la table. On parle aussi de donnée.

Une **base de données relationnelle** est un ensemble de tables dépendantes entre elles.

Une **valeur** est l'information attachée à un attribut pour une donnée (une ligne).

L'ensemble des valeurs possibles pour un attribut (liste finie de valeurs, intervalle de valeurs, ...) est un **domaine**.

Pour représenter de façon schématique la **relation**, on écrit : NomTable(Attribut1, Attribut2, ...)

On parle alors de **schéma relationnel**.

#### Exemples :

- La relation de la table ci-dessus s'écrit : Livres(Auteur, Titre, Année, Pays, Genre)
- La valeur de l'attribut « **Titre** » correspondant à l'auteur « King » vaut soit « *Nuit noire, étoile morte* », soit « *Misery* », suivant le tuple que l'on considère.
- L'attribut « **Genre** » possède 5 valeurs différentes sur cette table, mais en réalité son domaine est plus étendu car il existe d'autres genres

**Remarque :** Le schéma relationnel peut aussi être représenté comme

Nom de la table →

Attributs →

...

LIVRES
Auteur
Titre
Année
Pays
Genre

## (b) La clé, les clés

Prenons en exemple la table suivante, une playlist de radio :

2012-07-13	05H00	Katy Perry	Hot N Cold	2008	F	USA
2012-07-13	05H03	Dry	Ma Melodie	2012	H	France
2012-07-13	05H07	Gotye	Somebody That Use To Know	2012	H	Australie
2012-07-13	05H11	Julian Perretta	Generation X	2012	H	Angleterre
2012-07-13	05H17	Sexion D'assaut	Wati House	2012	Groupe	France
2012-07-13	05H23	Lmfao	Party Rock Anthem	2011	Groupe	USA
2012-07-13	05H27	Jenifer	Sur Le Fil	2012	F	France
2012-07-13	05H29	Of Monsters And Men	Little Talks	2011	Groupe	Islande
2012-07-13	05H33	Carly Rae Jepsen	Call Me Maybe	2012	F	Canada
2012-07-13	05H40	Pitbull	Back In Time	2012	H	USA
2012-07-13	05H43	Katy Perry	Hot N Cold	2008	F	USA
2012-07-13	05H51	Pink	Blow Me (one Last Kiss)	2012	F	USA

La relation s'écrit : **Playlist(Date, Heure, Artiste, Titre, Année, Formation, Pays)**  
 Quels attributs permettent d'identifier précisément et sans ambiguïté chaque tuple ?

### A retenir :

Chaque relation doit comporter une unique **clé primaire**. C'est une information qui permet d'identifier de manière unique une ligne dans une relation.

- Une clé primaire peut être composée de plusieurs attributs. Par exemple, dans une base de donnée regroupant les élèves du lycée, l'ensemble (nom, prénom, date de naissance) est une bonne clé primaire. La probabilité d'avoir deux homonymes parfaits nés le même jour étant très faible.
- Si l'ensemble des données d'une table ne présente pas de clé primaire, il est possible d'ajouter un attribut comprenant un **numéro d'identification (Id)**, généré automatiquement

### Exemples :

Dans une table regroupant des voitures, la plaque d'immatriculation est une bonne clé primaire.

En creusant cet exemple, on pourrait écrire cette relation sous forme de 2 tables :

- Passage d'une part listant les horaires et le morceau
- Morceaux d'autre part listant les morceaux et les infos sur l'artiste

Passage				Morceaux					
1	2012-07-13	05H00	6	1	Carly Rae Jepsen	Call Me Maybe	2012	F	Canada
2	2012-07-13	05H03	2	2	Dry	Ma Melodie	2012	H	France
3	2012-07-13	05H07	3	3	Gotye	Somebody That Use To Know	2012	H	Australie
4	2012-07-13	05H11	5	4	Jenifer	Sur Le Fil	2012	F	France
5	2012-07-13	05H17	11	5	Julian Perretta	Generation X	2012	H	Angleterre
6	2012-07-13	05H23	7	6	Katy Perry	Hot N Cold	2008	F	USA
7	2012-07-13	05H27	4	7	Lmfao	Party Rock Anthem	2011	Groupe	USA
8	2012-07-13	05H29	8	8	Of Monsters And Men	Little Talks	2011	Groupe	Islande
9	2012-07-13	05H33	1	9	Pink	Blow Me (one Last Kiss)	2012	F	USA
10	2012-07-13	05H40	10	10	Pitbull	Back In Time	2012	H	USA
11	2012-07-13	05H43	6	11	Sexion D'assaut	Wati House	2012	Groupe	France
12	2012-07-13	05H51	9						

Diagram illustrating the relationship between the **Passage** and **Morceaux** tables. The **Passage** table has a primary key **Id\_passage** (indicated by a green box around the first column). The **Morceaux** table has a primary key **Id\_morceaux** (indicated by a blue box around the first column). A purple line connects the **Id\_morceaux** column in the **Passage** table to the **Id\_morceaux** column in the **Morceaux** table, showing a one-to-many relationship.

On constate que pour plus de clarté, nous avons ajouter des identifiants : **Id\_passage** et **Id\_morceaux**.

La relation pour chaque table s'écrit :

**Passage**(Id\_passage, Date, Heure, Id\_morceau)

**Morceaux**(Id\_morceaux, Artiste, Titre, Année, Formation, Pays)

**Id\_passage** est donc la clé primaire de **Passage**, et **Id\_morceaux** la clé primaire de **Morceaux**.

On voit apparaître dans la relation le lien entre les 2 tables : **Id\_Morceaux** est présent dans la table **Passage**. **Id\_morceaux** est donc une **clé étrangère** à la table **Passage**, et c'est elle qui relie ces tables.

On la visualise avec un # dans la notation de la relation :

Passage(Id\_passage, Date, Heure, #Id\_morceau)

#### A retenir :

Une **clé étrangère** est un attribut dans une relation dont les données sont prises parmi les données d'une clé primaire d'une autre relation.

Table(Id\_Table, Attribut1, Attribut2,... , #Clé\_Etrangère)

#### Remarque :

*L'intérêt de faire plusieurs tables au lieu d'une seule réside dans la facilité de mise à jour, de correction, mais aussi d'accès à la base.*

#### Exemples :

Dans la base d'une entreprise comportant deux tables, on a les relations suivantes :

Client(Num\_client, Nom\_client, Rue, Ville, Code\_postal, Telephone)

Facture(Num\_facture, Date\_facture, #Num\_client)

- *Num\_client* et *Num\_facture* sont des clés primaires, car l'identification de chaque tuple par ces clés est unique.
- *Num\_client* est aussi utilisé dans la relation **Facture**, donc c'est une clé étrangère.

### (c) Les contraintes d'intégrité

Par exemple, considérons cette version (très) simplifiée de la table Transaction décrivant des opérations bancaires :

Date	Id_Client_Emetteur	Id_Client_Destination	Montant	Type
25/12/2020	0001200263	0001200526	1000	virement
25/12/2020	0001200653	0001300598	250	cheque
25/12/2020	0001200263	0001200526	1000	virement
26/12/2020	0001200698	0001200584	632	virement

Comparer les **tuples** des transactions ligne 2 et ligne 3 : s'agit-il d'une erreur ou bien effectivement de deux transactions différentes ?? ...

Une solution est donc d'ajouter un **nouvel attribut qui identifie de façon unique** la donnée. Le plus simple est en général l'ajout d'un Id.

Id_Transaction	Date	Id_Client_Emetteur	Id_Client_Destination	Montant	Type
1	25/12/2020	0001200263	0001200526	1000	virement
2	25/12/2020	0001200653	0001300598	250	cheque
3	25/12/2020	0001200263	0001200526	1000	virement
4	26/12/2020	0001200698	0001200584	632	virement

Maintenant, nous savons que les deux opérations sont bien distinctes !

### A retenir :

La cohérence des données au sein d'une base est assurée par des **contraintes d'intégrité**. Il s'agit de propriétés que les données doivent vérifier à tout instant.

- **Les contraintes d'unicité** (ou d'entité)

Chaque tuple (ou entité) est identifié de manière unique grâce à la clé primaire.

- **Les contraintes de référence**

Elles garantissent l'association de deux relations : la clé étrangère (attribut de la relation B) utilisée dans la relation A existe.

- **Les contraintes de domaines**

Les valeurs sont restreintes à un domaine. En dehors de ce domaine, la valeur est illégale et le SGBD retourne une erreur.

### Exemples :

Dans la base d'entreprise de l'exemple précédent :

Client(Num\_client, Nom\_client, Rue, Ville, Code\_postal, Telephone)

Facture(Num\_facture, Date\_facture, #Num\_client)

- La contrainte d'unicité assure qu'un client n'est pas en doublon dans la table **Client**.
- La contrainte de référence assure qu'un client existe bien (#Num\_client) dans la table **Client** pour pouvoir saisir une donnée de la table **Facture**.
- La contrainte de domaine rejette une saisie de *Date\_facture* sous la forme « 20\_octobre\_2021 », mais accepte une saisie sous la forme « 20-10-2021 »

## (d)Les associations ( /!\ limites du programme)

Lorsqu'on implémente une BD, on utilise un SGBD qui va permettre d'accéder, de modifier, etc.

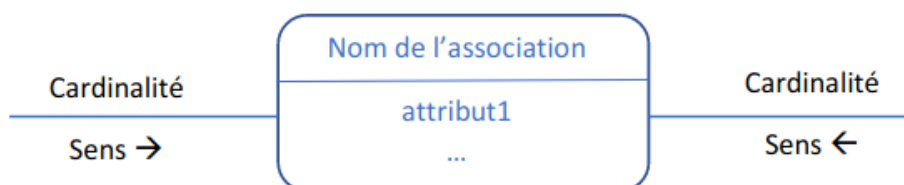
Ce SGBD utilise des schémas d'associations entre les tables, qui lui permet de traduire une partie des règles de gestions qui n'ont pas été satisfaites par la simple définition des entités.

### A retenir :

L'association est à minima caractérisée par :

- **Un nom** : généralement on utilise un verbe définissant le lien entre les entités
- **2 cardinalités** : elles sont présentes sur les deux extrémités du lien. Chaque cardinalité est un couple de valeurs (mini, maxi) qui traduisent 2 règles de gestion (une par sens)

On peut représenter une association par le schéma suivant :



Dans l'exemple du (a), on définit une association "**écrire**" entre les tables Auteurs et Livres. Ce lien possède 2 cardinalités (1,n) et (1,1) qui traduisent les deux règles de gestion suivantes :

- ➔ 1,n : Un auteur peut avoir écrit entre 1 et une infinité (noté n) de livres de la table Livres
- ➔ 1,1 : Un livre a été écrit par un et un seul auteur de la table Auteur

### Remarque :

Les cardinalités les plus répandues sont les suivantes : 0,n ; 1,n ; 0,1 ; 1,1

AUTEURS
Nom_Auteur
Prenom
Date de naissance

**1,n**

Ecrire

**1,1**

LIVRES
Auteur
Titre
Année
Pays
Genre