

# Chapitre 1

## Bases de données

### 1.1 Système de gestion de bases de données (SGBD)

Les données sont le nouvel or de notre société ; elles sont partout, générées par toutes et tous, souvent collectées et de plus en plus traitées. Avec la croissance du numérique et de l'internet, les bases de données et leurs traitements sont devenus informatiques. Toutefois, il faut rester conscient du fait qu'elles ne sont pas forcément liées à l'informatique et internet : institutions, collectivités et entreprises les utilisent pour gérer leurs biens et services, leurs ressources, leurs flux et stocks, etc.

Devant la croissance de leur nombre, leur taille et leur usage, les outils vus en première (table de données) s'avèrent insuffisants à les traiter. À présent, la gestion et l'accès à une base de données sont assurés par un ensemble de programmes qui constituent un **Système de Gestion de Base de Données (SGBD)**. L'information d'une base de données est stockée dans des fichiers et contrairement aux fichiers csv, il n'est pas possible de travailler directement sur ces données avec un simple éditeur de texte, c'est pour cela qu'on a besoin d'un SGBD dont le rôle est de simplifier la gestion des bases de données.

Les SGBD sont des logiciels, fonctionnant en mode client / serveur, assez complexes à mettre en œuvre et à utiliser. Ils sont conçus pour gérer plusieurs millions, voire milliards d'enregistrements de manière fiable et sécurisée. Un SGBD doit permettre de :

- Stocker un grand volume de données, pendant longtemps et avec sécurité ;
- Accéder efficacement aux données ;
- Assurer le respect des règles de cohérence définies sur les données et vérifier les contraintes d'intégrité ;
- Permettre à l'utilisateur de créer de nouvelles bases de données ;
- Permettre à l'utilisateur d'interroger et de modifier les données de la base de données ;
- Contrôler l'accès aux données par plusieurs utilisateurs en même temps (autorisations d'accès), l'action d'un utilisateur ne devant pas en affecter un autre ;
- Assurer la sécurité des données (sauvegardes, synchronisations et maintenance de la base de données sur plusieurs machines / serveurs) et la récupération de la base et le retour à la normale en cas de panne.

Il existe de nombreux SGBD, certains sont payants (et très chers), d'autres gratuits et libres, parmi les plus célèbres on peut citer : Oracle (propriétaire), MySQL (libre), Microsoft SQL Server (propriétaire) ou encore PostgreSQL (libre), SQLite (libre).

Il existe par ailleurs différents modèles de SGDB répondant à des conceptions et des besoins différents :

**Le modèle hiérarchique** : les données sont classées hiérarchiquement. On les représente sous la forme d'un arbre avec une arborescence descendante.

**Le modèle réseau** : c'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** : lequel ressemble au modèle relationnel avec une manipulation différente.

Dans ce cours, nous nous intéresserons au modèle relationnel et aux SGBD associées : les **SGBDR**.

## 1.2 Bases de données relationnelles

Les bases de données relationnelles sont le type de bases que l'on connaît et que l'on utilise le plus dans le monde aujourd'hui. Inventées en 1970 et basées sur l'algèbre relationnelle et les travaux de l'informaticien britannique E.F. Codd (alors chercheur pour IBM) elles permettent de modéliser facilement et sans grandes contraintes les systèmes du monde réel et de créer des bases de données simples à maintenir, à faire évoluer et indépendantes de leur support. C'est la technologie majeure en bases de données depuis les années 1980 (mais elles ne sont pas utilisées pour les bases de données des réseaux sociaux).

Dans ce type de bases de données, les données sont organisées en **tables (ou relations)**. Une table (ou une relation) peut être vue comme un tableau à 2 dimensions qui forme l'ensemble des enregistrements qui existent sur les données. C'est ce que l'on a vu en première.

**Exemple** : la table `pokedex` suivante :

pokedex			
id	nom	type_1	type_2
1	bulbizarre	plante	poison
2	herbizarre	plante	poison
3	florizarre	plante	poison
4	carapuce	eau	None
...	...	...	...

Une table de données comprend diverses caractéristiques que l'on retrouve dans le tableau suivant :

<b>Enregistrement (ou entité)</b>	Ligne de la table.
<b>Attribut</b>	Nom d'une colonne.
<b>Degré d'une table</b>	Nombre d'attributs.
<b>Type d'un attribut</b>	Entier, flottant, chaîne de caractères, booléen, liste...
<b>Domaine d'un attribut</b>	Ensemble des valeurs que peut prendre un attribut (par exemple, pour un âge, un entier entre 0 et 120).
<b>Clé primaire</b>	Attribut permettant d'identifier de manière unique chaque enregistrement
<b>Clé étrangère</b>	Attribut d'une table faisant référence à la clé primaire d'une autre table.

**Exemple :** la table `pokedex` admet la structure suivante :

{id : int, nom : str, type\_1 : str, type\_2 : str}

Les attributs sont `id`, `nom`, `type_1` et `type_2`; le degré est donc quatre. Les attributs `id` et `nom` peuvent jouer le rôle de clé primaire mais pas `type_1` ni `type_2`.

### 1.2.1 Clé primaire

Il est important de noter que la clé primaire identifie de manière unique chaque enregistrement et pas un enregistrement en particulier. Cela a pour objectif d'éviter les doublons, de préserver l'intégrité de la table et de faciliter le traitement des données. Ainsi, la clé primaire sera la plupart du temps un identifiant alphanumérique.

**Exemple :** Dans la table `livres` ci-dessous, l'attribut `ISBN` fait figure de clé primaire. L'ISBN est un code international servant à identifier uniquement les livres, il est donc optimal lorsque l'on souhaite définir une clé primaire pour une base de données traitant de ceux-ci.

livres			
ISBN	Titre	Auteur	Parution
978-2-9155-4943-0	L'Étoile de Pandore T1	Hamilton	2005
978-2-9155-4978-2	L'Étoile de Pandore T2	Hamilton	2006
978-2-3529-4042-5	L'Étoile de Pandore T3	Hamilton	2007
978-2-3529-4090-6	L'Étoile de Pandore T4	Hamilton	2007
2-905158-69-7	Les Guerriers du Silence	Bordage	1993
2-84172-087-X	Terra Mater	Bordage	1994
2-84172-088-8	La Citadelle Hyponéros	Bordage	1995
...	...	...	...

### 1.2.2 Clé étrangère

Il est possible d'ajouter des attributs et des informations dans la table. En reprenant l'exemple précédent, on pourrait ajouter la nationalité de l'auteur par exemple, ce qui donnerait ceci :

livres				
ISBN	Titre	Auteur	Parution	Nationalité de l'auteur
978-2-9155-4943-0	L'Étoile de Pandore T1	Hamilton	2005	UK
978-2-9155-4978-2	L'Étoile de Pandore T2	Hamilton	2006	UK
978-2-3529-4042-5	L'Étoile de Pandore T3	Hamilton	2007	UK
978-2-3529-4090-6	L'Étoile de Pandore T4	Hamilton	2007	UK
2-905158-69-7	Les Guerriers du Silence	Bordage	1993	Fr
2-84172-087-X	Terra Mater	Bordage	1994	Fr
2-84172-088-8	La Citadelle Hyponéros	Bordage	1995	Fr
...	...	...	...	...

Toutefois, ce n'est pas très pratique ni pertinent car on se retrouve avec une grande quantité d'information redondante. Par ailleurs, imaginons une autre table de données où l'on retrouverait nos auteurs et leurs nationalités. Imaginons maintenant que l'un de ces auteurs change de nationalité, il faudrait aller, dans toutes les tables de données où il apparaît, chercher tous les enregistrements où il figure pour les modifier. Cela peut représenter un travail colossal et inutile. En effet, il est beaucoup plus simple de considérer des tables de données séparées **livres** et **auteurs** et de mettre en relation celles-ci si besoin afin d'accéder aux informations souhaitées. On pourrait donc avoir dans ce cas pour tables de données :

livres			
ISBN	Titre	id_Auteur	Parution
978-2-9155-4943-0	L'Étoile de Pandore T1	1	2005
978-2-9155-4978-2	L'Étoile de Pandore T2	1	2006
978-2-3529-4042-5	L'Étoile de Pandore T3	1	2007
978-2-3529-4090-6	L'Étoile de Pandore T4	1	2007
2-905158-69-7	Les Guerriers du Silence	2	1993
2-84172-087-X	Terra Mater	2	1994
2-84172-088-8	La Citadelle Hyponéros	2	1995
...	...	...	...

auteurs			
id_Auteur	Nom	Prénom	Nationalité
1	Hamilton	Peter F.	UK
2	Bordage	Pierre	Fr
...	...	...	...

L'attribut `id_Auteur` est ici la clé primaire de la table **auteurs** et donc une **clé étrangère** de la table **livres**.

Une **base de données** 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.

On retiendra donc les principes généraux pour regrouper les données en plusieurs tables :

- Mettre dans une même table les données relatives à un même sujet, le choix des tables est primordial car il détermine en grande partie la qualité de la base.
- Créer de nouvelles tables pour :
  - éviter la redondance des données ;
  - limiter les incohérences lors des mises à jour ;
  - faciliter le traitement des données.

### 1.2.3 Contraintes

Une base de données obéit à de nombreuses contraintes. Il est donc impératif de les respecter et de bien penser la structure de sa base afin que celle-ci soit exploitable (de nombreuses bases ne le sont pas faute de problèmes de conception). On doit respecter :

- Des contrainte de domaine : type, encadrement, format de valeur, non nullité...
- La contrainte de clé primaire : existence et unicité pour chaque table ;
- Les contraintes de clé étrangère et d'intégrité référentielle : considérons une table T1 dont l'attribut A1 est la clé primaire et une table T2 dont la clé primaire est l'attribut A2 ; de plus, l'attribut A2 est aussi présent dans la table T1 faisant de lui une clé étrangère ; alors toutes les valeurs prises par A2 dans la table T1 doivent exister dans la table T2.

**Exemple :** les deux tables ci-dessous ne respectent pas les principes ci-dessus, notamment celui de l'intégrité référentielle car la valeur d'id\_Auteur 0 n'apparaît pas dans la table `auteurs` (où il est clé primaire) mais dans la table `livres` (où il est clé étrangère).

livres			
ISBN	Titre	id_Auteur	Parution
978-2-9155-4943-0	L'Étoile de Pandore T1	1	2005
978-2-9155-4978-2	L'Étoile de Pandore T2	1	2006
978-2-3529-4042-5	L'Étoile de Pandore T3	1	2007
978-2-3529-4090-6	L'Étoile de Pandore T4	1	2007
2-905158-69-7	Les Guerriers du Silence	0	1993
2-84172-087-X	Terra Mater	0	1994
2-84172-088-8	La Citadelle Hyponéros	0	1995
...	...	...	...

auteurs			
id_Auteur	Nom	Prénom	Nationalité
1	Hamilton	Peter F.	UK
2	Bordage	Pierre	Fr
...	...	...	...

Pour chaque table, il est possible de résumer une partie des contraintes sous la forme d'un tableau afin de mieux les visualiser et de les préciser.

**Exemple :** reprenons la table `livres` et précisons ses contraintes pour chaque attribut.

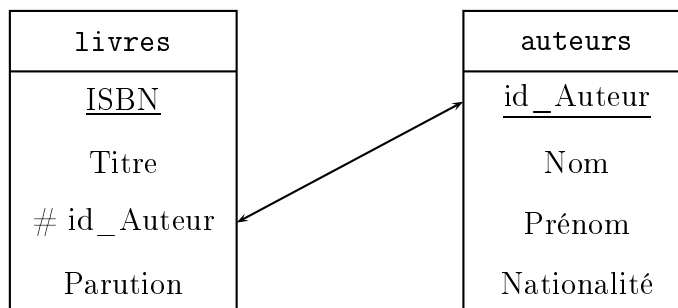
Attribut	Type	Unique	Autres caractéristiques	Non Null
ISBN	str	Vrai	format str-str-str-str-str	Vrai
Titre	str		$1 \leq \text{longueur} \leq 300$	Vrai
id_Auteur	int		$1 \leq$	
Parution	int		$-8000 \leq$	

**Remarques :**

- La colonne Non Null indique si le champ peut être laissé vide ou pas, auquel cas on précisera Vrai (c'est une option présente sur la majorité des SGBD).
- Il est possible d'ajouter d'autres caractéristiques comme une valeur par défaut, un incrément pour chaque enregistrement créé, etc...

### 1.2.4 Schéma relationnel

Il est possible de représenter une base de données sous la forme d'un schéma relationnel dans lequel on fait apparaître les clés primaires soulignées et les clés étrangères précédées du symbole # puis en reliant les tables entre elles au niveau des clés étrangères / primaires.



## 1.3 Ressources supplémentaires

- Vidéo sur les bases de données relationnelles.
- Vidéo sur les SGBDR.

## 1.4 Attendus

- Identifier les concepts définissant le modèle relationnel.
- Savoir distinguer la structure d'une base de données de son contenu. Repérer des anomalies dans le schéma d'une base de données.
- Identifier les services rendus par un système de gestion de bases de données relationnelles : services sans en détailler le persistance des données, gestion des accès concurrents, efficacité de traitement des requêtes, sécurisation des accès.

## 1.5 Exercices

**Exercice 1.1.** Une entreprise a dans sa base de données la table `clients` suivantes :

clients					
id_client	nom	prénom	mail	adresse	téléphone
...	...	...	...	...	...
id_c390922091	Simpson	Marge	marge.simpson@sgfd.org	...	...
id_c390922092	Simpson	Homer	homer.simpson@sgfd.org	...	...
...	...	...	...	...	...

1. Identifier les différents attributs de la table, leurs types et domaines.
2. Lesquels sont susceptibles d'être des clés primaires ?

**Exercice 1.2.** On reprend la base de données de l'exercice précédent et on en considère deux nouvelles tables :

transactions		
id_transaction	id_client	date
...	...	...
id_t683900	id_c390922092	...
...	...	...

détails transactions		
id_transaction	id_produit	quantité
...	...	...
id_t683900	id_p783890	2
...	...	...

1. Identifier les différents attributs des tables, leurs types et domaines.
2. Lesquels sont susceptibles d'être des clés primaires ?
3. Lesquels sont susceptibles d'être des clés étrangères ?
4. Déterminer quelles pourraient être les autres tables de la base ? Quels seraient leurs attributs ?

**Exercice 1.3.** Déterminer les contraintes de la table `clients` de l'exercice 1.1.

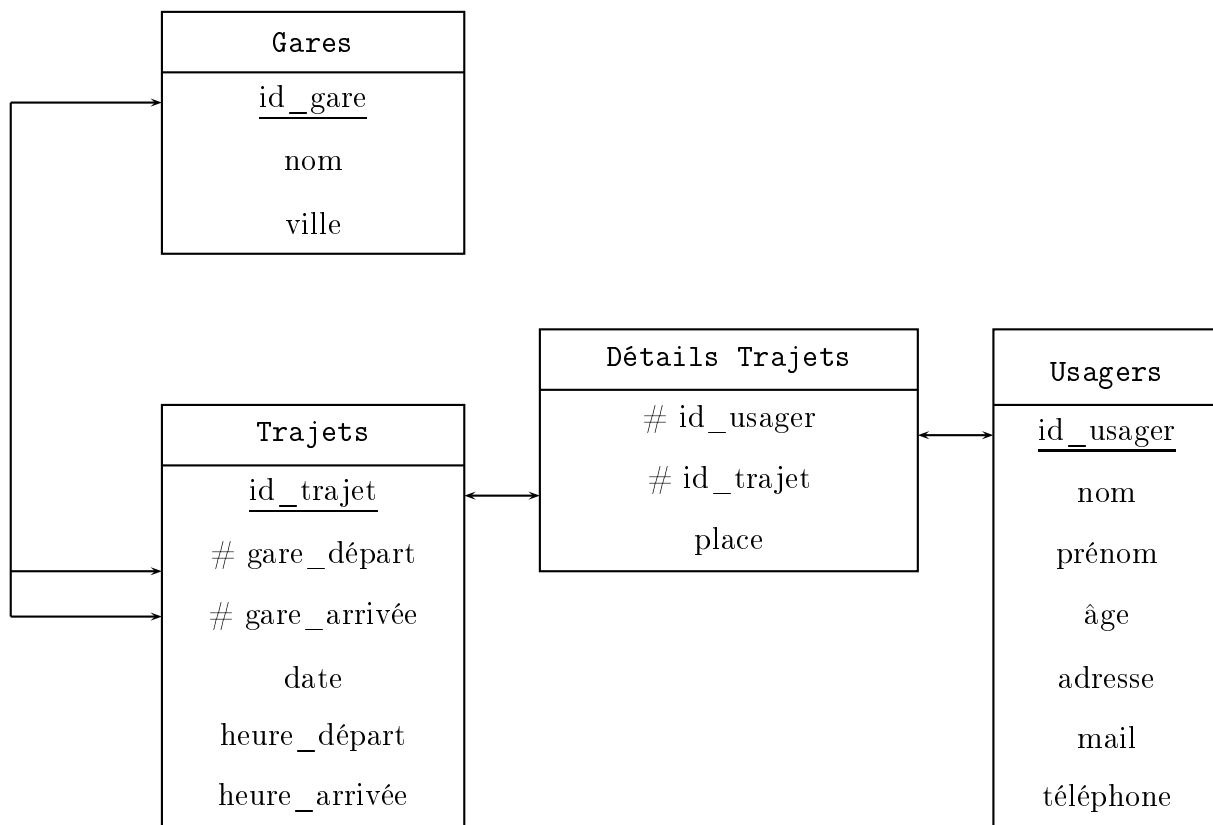
**Exercice 1.4.** Construire le schéma associé à la base de données de l'exercice 1.2.

**Exercice 1.5.** On considère la base de données d'une bibliothèque dont le schéma relationnel est le suivant :

- Abonne(Abonne\_ID, Nom, Prenom, DateNaissance, Adresse, Ville, CodePostal)
- Livre(Livre\_ID, Titre, Auteur, Genre, ISBN)
- Emprunt(Abonne\_ID, Livre\_ID, Livre\_Titre, Livre\_ISBN, DateEmprunt, DateRetourPrevu, DateRetourReel)

Ce schéma est-il correct ? Si non, proposer des corrections.

**Exercice 1.6.** Déterminer les clés primaires et étrangères des tables du schéma relationnel ci-dessous. Quelles pourraient être les autres tables cette base de données ?



**Exercice 1.7.** Construire la base de données du lycée (membres, classes, etc). On précisera ses contraintes et la représentera sous forme d'un schéma.