

# Système de Gestion de Bases de Données Relationnelles

Objectifs principaux du chapitre :

- Maîtriser les concepts de base en Bases de Données Relationnelles
- Savoir réaliser des requêtes SQL de base auprès d'un Système de Gestion de Bases de Données Relationnelles

## 1 Introduction

### 1.1 Basse de données (BD)

Une base de données (BD) est un ensemble de données :

- Représentant une partie du monde réel
- Structurées
- Cohérentes
- Conservées de manière permanente (persistance)
- Manipulables/modifiables/interrogeables
- Partageables
- Sécurisées

**Quelques exemples d'utilisation de BD :**

- Gestion d'entreprises : stocks, personnel, clients, ...
- Gestion bancaire : comptes, emprunts, ...
- Systèmes de réservation : avions, trains, spectacles, ...
- Bibliothèques : ouvrages, emprunteurs, prêts ...

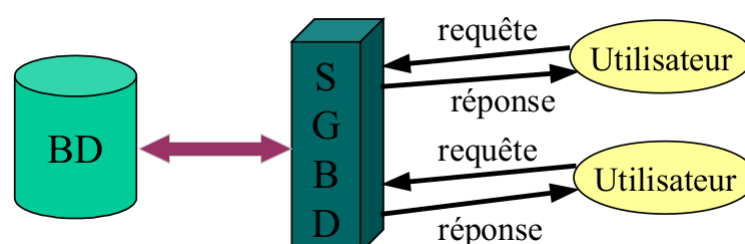
**Exemple de la banque (BNP Paribas) :**

- Grande masse d'informations 8 000 000 clients
- 4 comptes par client, donc 32 000 000 comptes
- 20 écritures par mois par compte, donc 640 000 000 écritures par mois
- Plusieurs utilisateurs simultanément :
  - 2140 agences
  - 31 460 collaborateurs
  - des milliers d'accès internet

**Utilisation d'une base de données relationnelle pour toute application nécessitant la structuration, le stockage, la manipulation et l'interrogation d'un ensemble conséquent d'informations.**

### 1.2 Système de Gestion de Bases de Données (SGBD)

- Un SGBD est un logiciel intermédiaire entre la base de données et l'utilisateur (un humain ou un programme).
- Un SGBD est un ensemble de programmes permettant la création et l'administration d'une BD, la sauvegarde (stockage) et la manipulation (insertion, modification, suppression, interrogation) des données d'une manière efficace.
- Exemples de SGBD : Oracle, PostgreSQL, MySQL, Access, **SQLite**, DB2, H2, ...



## 2 Bases de Données Relationnelles

Représentation des données à l'aide de tableaux à deux dimensions, on parle de tables.

Structure de données de base : la relation (ici c'est une table)

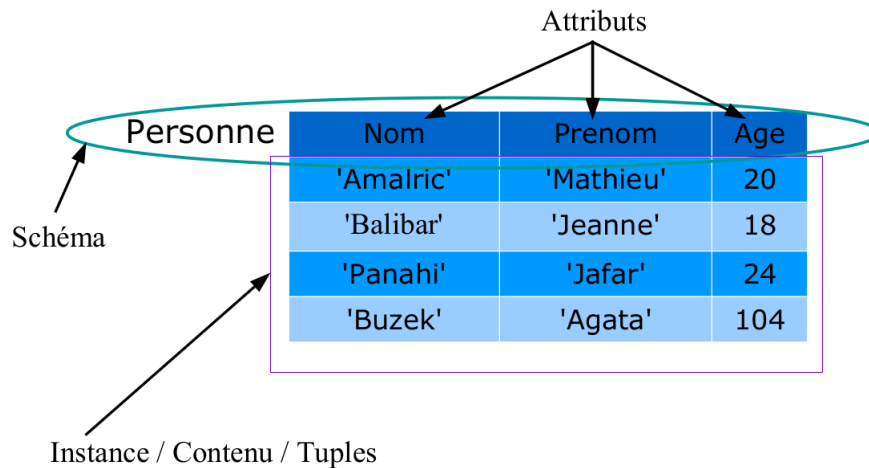
Une relation est caractérisée par :

— son schéma : un nom et un ensemble d'attributs avec leur domaine.

**Personne(Nom : (str), Prenom : (str), age : (int))**

— ses éléments (ses tuples, ses n-uplets), i.e. son contenu (son instance).

**('Amalric','Mathieu',20)**



**Important :** Pour chaque attribut d'une relation, il faut définir un domaine.

Le domaine d'un attribut est l'ensemble ( fini ou pas) des valeurs possibles ( entiers, flottants, chaînes de caractères, dates...).

**Exemple :** Le domaine de l'attribut Age est : l'ensemble des entiers positifs (int) (Il faut rendre impossible l'entrée d'un âge non entier lors de la création d'un n-uplet)

### 2.1 Unicité d'un n-uplet

Dans une relation, deux n-uplets identiques sont non autorisés. On appelle cela une contrainte d'intégrité.

#### Définition :

- Une **contrainte d'intégrités** est une contrainte sur le schéma de la BD permettant en partie de garantir la cohérence de celle-ci.
- Toute relation possède une **clé primaire** c'est à dire un ensemble minimal d'attributs dont les valeurs permettent de distinguer deux tuples de la relation.  
Par convention on souligne les attributs participant aux clés primaires.

#### Remarques :

- Les valeurs de la clé primaire permettent d'identifier de manière unique un tuple de la table.
- Impossible d'avoir deux tuples identiques.

Quelle clef primaire pour le schéma relationnel : **Personne(Nom : (str), Prenom : (str), age : (int)) ?**

#### Exercice n° 1

Y a t-il une clef primaire possible pour les schémas relationnels ci-dessous :

1. Voiture(Couleur,NomModele)
2. Voiture(NumImm,Couleur,NomModele)
3. ModeleVoiture(NomModele,Marque)
4. Livre(Titre,Auteur,AnneePublication,Note)
5. Livre(Titre,Auteur,AnneePublication,Note,ISBN)
6. Plantes(Famille,Genre,Espece,Nom commun, NomScientifique)

## Exercice n° 2

Que pensez vous de cette table relationnelle ?

Voiture	NumImm	Couleur	NomModèle
	123XY34	Jaune	106
	34UV62	Verte	106
	345RT62	Verte	Mégane
	123XY34	Verte	106
	234XU45	Bleue	Clio

## Exercice n° 3

id	titre	auteur	ann_publi	note
1	1984	Orwell	1949	10
2	Dune	Herbert	1965	8
3	Fondation	Azimov	1951	9
4	Le meilleur des mondes	Huxley	1931	9
5	Fahrenheit 451	Bradbury	1953	7
6	Ubik	K.Dick	1969	6
7	Chroniques martiennes	Bradbury	1950	7
8	La nuit des temps	Barjavel	1968	7
9	Blade Runner	K.Dick	1968	9
10	Les Robots	Azimov	1950	10
11	La Planète des singes	Boule	1963	9
12	Ravage	Barjavel	1943	11
13	Le Maître du Haut Château	K.Dick	1962	7
14	La fin de l'éternité	Azimov	1955	9

1. Pourquoi l'attribut **note** ne peut-il pas être une clef primaire ?
2. L'attribut **titre** peut-il servir de clef primaire ?
3. Le couple (**auteur** , **ann\_publi**) peut-il servir de clef primaire ?

**Remarque** : Nous avons créé artificiellement une clé primaire, car aucun des autres attributs ne pouvait convenir.

## 2.2 Redondance des données

En regardant de plus près l'exercice précédent, des informations apparaissent plusieurs fois.

Par exemple : l'auteur "Azimov" apparaît 3 fois.

Dans une base de donnée, on évite autant que possible la redondance d'informations.

Pour cela, on crée une autre relation AUTEURS que l'on lie avec la relation LIVRES.

On a rajouté quelques informations supplémentaires sur les auteurs dans la relation AUTEURS.

Dans la relation AUTEURS, chaque auteur est identifié par l'attribut "id" (clé primaire de la relation)

— Dans la relation LIVRES, on a rajouté un attribut "id\_auteur" qui est la clé primaire de la relation AUTEURS.

— L'attribut "id\_auteur" est ce que l'on nomme **une clé étrangère** de la relation LIVRES, elle permet de faire le lien entre les deux relations.

**Définition** : Une **clé étrangère** est un ensemble d'attributs dont les valeurs appartiennent à l'ensemble des valeurs d'une clé primaire.

Clé primaire

Clé étrangère

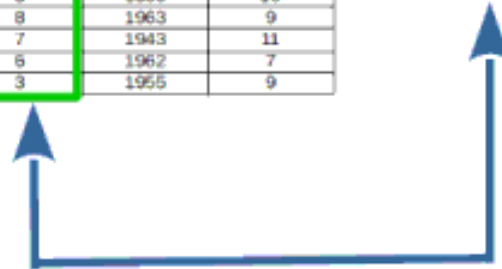
Clé primaire

id	titre	id_auteur	ann_publi	note
1	1984	1	1949	10
2	Dune	2	1965	8
3	Fondation	3	1951	9
4	Le meilleur des mondes	4	1931	9
5	Fahrenheit 451	5	1953	7
6	Ubik	6	1969	6
7	Chroniques martiennes	5	1950	7
8	La nuit des temps	7	1968	7
9	Blade Runner	6	1968	9
10	Les Robots	3	1950	10
11	La Planète des singes	8	1963	9
12	Ravage	7	1943	11
13	Le Maître du Haut Château	6	1962	7
14	La fin de l'éternité	3	1955	9

Relation : LIVRES

id	nom	prenom	ann_naissance	langue_ecriture
1	Orwell	George	1903	anglais
2	Herbert	Frank	1920	anglais
3	Azimov	Isaac	1920	anglais
4	Huxley	Aldous	1894	anglais
5	Bradbury	Ray	1920	anglais
6	K. Dick	Philip	1928	anglais
7	Barjavel	René	1911	français
8	Bouffé	Pierre	1912	français

Relation : AUTEURS



### 2.3 Schéma relationnel

On appelle schéma relationnel l'ensemble des relations présentes dans une base de données. c'est à dire qu'il faut fournir :

- les noms des relations
- la liste des attributs et de leurs domaines
- la clé primaire (on la souligne)
- les clés étrangères (précédées d'un # ou suivies d'un \*)

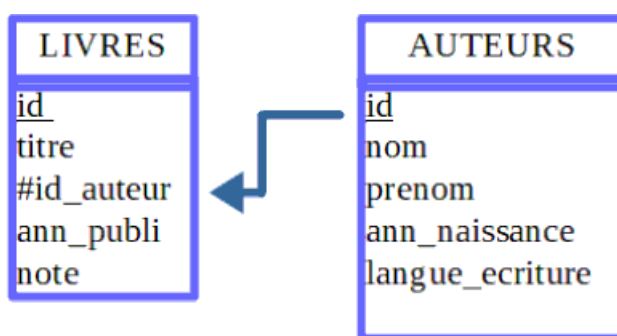
Voici le schéma relationnel de **LIVRES** :

LIVRES ( Id : int , titre : str , #id\_auteur : int , ann\_publi : int , note : int)

Voici le schéma relationnel de **AUTEURS** :

...

Ces relations se représentent graphiquement avec ce schéma :



id, ann\_publi, ann\_naissance, note, id\_auteur sont des entiers positifs  
titre, nom, prenom, langue\_ecriture sont des chaînes de caractères

**Exercice n° 4**

Trouver l'erreur dans cette base de données.

Modele		Voiture		
NomModele	Marque	NumImm	Couleur	#NomModele
106	Peugeot	123XY34	Jaune	106
206	Peugeot	34UV62	Verte	106
306	Peugeot	345RT62	Verte	Mégane
Clio	Renault	123XY36	Verte	106
Espace	Renault	234XU45	Bleue	Clio

**Exercice n° 5**

Voici une base de données constituée des trois tables ci-dessous :

Eleve			Matiere		Moyenne		
Nom	Prenom	Identifiant	Id_matiere	Intitulé	Identifiant	Id_matiere	Note
LASSOU	Pierre	18174125	1	Français	18174125	5	15
DUVANT	Yvan	90054985	2	Mathématiques	90054985	5	9
LIGUILI	Guy	75453058	3	Histoire/Géographie	44850453	4	13
DALORD	Homer	44850453	4	EPS	18174125	1	11
...	...	...	5	Informatique	...	...	...

1. Pour chaque relation, identifier le (ou les) attribut(s) qui sert (servent) de clef primaire.
2. Proposer un schéma relationnel pour chaque relation puis donner une représentation graphique.

**Exercice n° 6**

Considérons la base de données **Tour de France 2020**, contenant les relations suivantes :

Equipes		Coureurs			
codeEquipe	nomEquipe	dossard	nomCoureur	prénomCoureur	codeEquipe
ALM	AG2R La Mondiale	141	LÓPEZ	Miguel-Angel	AST
AST	Astana Pro Team	142	FRAILE	Omar	AST
TBM	Bahrain - McLaren	143	HOULE	Hugo	AST
BOH	BORA - hansgrohe	12	BENNETT	George	TJV
CCC	CCC Team	41	ALAPHILIPPE	Julian	DQT
COF	Cofidis, Solutions Crédits	44	CAVAGNA	Rémi	DQT
DQT	Deceuninck - Quick Step	45	DECLERCQ	Tim	DQT
EF1	EF Pro Cycling	121	MARTIN	Guillaume	COF
GFC	Groupama - FDJ	122	CONSONNI	Simone	COF
LTS	Lotto Soudal	123	EDET	Nicolas	COF
...	...	...	...	...	...

Etapas				Temps		
numéroEtape	villeDépart	villeArrivée	km	dossard	numéroEtape	tempsRéalisé
1	Nice	Nice	156	41	2	04 :55 :27
2	Nice	Nice	185	121	4	04 :07 :47
3	Nice	Sisteron	198	11	5	04 :21 :22
4	Sisteron	Orcières-Merlette	160	122	5	04 :21 :22
5	Gap	Privas	198	41	4	04 :08 :24
...	...	...	...	...	...	...

1. Déterminer la clef primaire pour chacune des relations.
2. Donner le schéma relationnel pour chacune des relations
3. Construire le diagramme de cette base de donnée.

## 2.4 Contraintes d'intégrité

La cohérence des données au sein d'une BD est assurée par des contraintes d'intégrité. Ces dernières sont des invariants, c'est à dire des propriétés logiques que les données doivent vérifier à tout instant. Parmi ces contraintes, on distingue :

### 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 (ou d'entité)

La contrainte de relation impose que tout enregistrement soit unique : cette contrainte est réalisée par l'existence obligatoire d'une clé primaire. Cette clé primaire est souvent créée de manière artificielle (voir id\_auteur dans le paragraphe **Redondance des données**)

### 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.

## Exercice n° 7

1. Quelle est la contrainte non respectée dans l'exercice 2 ?
2. Quelle est la contrainte non respectée dans l'exercice 4 ?

## 3 SGBD

Dans une base de données, l'information est stockée dans des fichiers.

Pour y avoir accès, il faut utiliser un logiciel que l'on appelle "système de gestion de base de données" (SGBD). Il en existe plusieurs, des gratuites, des payantes, des libres et des propriétaires.

- les SGBD permettent de gérer la lecture, l'écriture ou la modification des informations contenues dans une base de données
- les SGBD permettent de gérer les autorisations d'accès à une base de données. Il est en effet souvent nécessaire de contrôler les accès par exemple en permettant à l'utilisateur A de lire et d'écrire dans la base de données alors que l'utilisateur B aura uniquement la possibilité de lire les informations contenues dans cette même base de données.
- les fichiers des bases de données sont stockés sur des disques durs dans des ordinateurs, ces ordinateurs peuvent subir des pannes. Il est souvent nécessaire que l'accès aux informations contenues dans une base de données soit maintenu, même en cas de panne matérielle. Les bases de données sont donc dupliquées sur plusieurs ordinateurs afin qu'en cas de panne d'un ordinateur A, un ordinateur B contenant une copie de la base de données présente dans A, puisse prendre le relais. Tout cela est très complexe à gérer, en effet toute modification de la base de données présente sur l'ordinateur A doit entraîner la même modification de la base de données présente sur l'ordinateur B. Cette synchronisation entre A et B doit se faire le plus rapidement possible, il est fondamental d'avoir des copies parfaitement identiques en permanence. C'est aussi les SGBD qui assurent la maintenance des différentes copies de la base de données.
- plusieurs personnes peuvent avoir besoin d'accéder aux informations contenues dans une base de données en même temps. Cela peut parfois poser problème, notamment si les 2 personnes désirent modifier la même donnée au même moment (on parle d'accès concurrent). Ces problèmes d'accès concurrent sont aussi gérés par les SGBD.

Comme nous venons de la voir, les SGBD jouent un rôle fondamental. L'utilisation des SGBD explique en partie la supériorité de l'utilisation des bases de données sur des solutions plus simples à mettre en oeuvre comme les fichiers au format CSV.

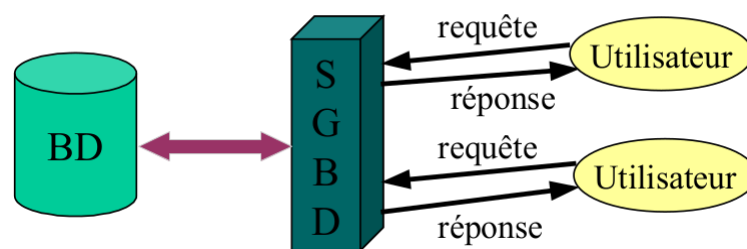
Ce sont des logiciels dont la conception est bien trop complexe pour pouvoir être abordée dans ce cours ; nous nous contenterons d'interagir avec eux par l'intermédiaire de requêtes exprimées dans un langage devenu standard au fil des temps : le langage SQL (pour Structured Query Language).

**Définition : Une requête est une demande que l'on fait à une base de données**

### 3.1 Architecture à deux tiers

Sur un réseau informatique, des informations sont en permanence échangées entre deux machines, un logiciel assurant le traitement des informations sur chacune d'entre elles. On distingue le logiciel client installé sur la machine qui envoie des requêtes au logiciel serveur installé sur la machine qui traite les requêtes.

Ce mode de communication est appelé architecture à deux niveaux. C'est l'une des architectures client-serveur possibles.



### 3.2 Architecture à trois-tiers

L'architecture trois-tiers est une architecture client-serveur qui ajoute un niveau supplémentaire dans l'environnement précédemment décrit. Un serveur de données transmet les informations à un serveur d'application qui, à son tour, transmet les informations traitées vers un client. Ce modèle d'architecture présente plusieurs avantages :

- meilleure prise en compte de l'hétérogénéité des plates-formes ;
- amélioration de la sécurité des données en supprimant le lien entre le client et les données ;
- meilleure répartition des tâches entre les différentes couches logicielles.

