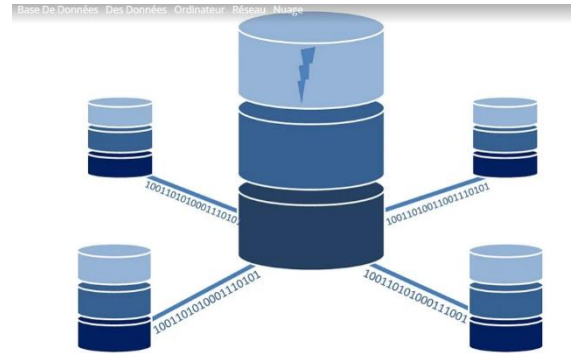


# BASES DE DONNEES

## INTRODUCTION (Généralités)

L'année dernière nous avons eu l'occasion de travailler sur des données structurées en les stockant dans des fichiers au format CSV.

Même si cette méthode de stockage de l'information peut s'avérer pratique dans certains cas précis, il est souvent souhaitable d'utiliser une **base de données pour stocker des données**.



Le terme **base de données** est apparu au début des années 60. C'est l'apparition des disques durs à la fin des années 50 qui a permis d'utiliser les ordinateurs pour stocker et manipuler des données.

Avec l'apparition du Web, la quantité de données à stocker a littéralement explosé.

Aujourd'hui, la plupart des sites internet utilisent des bases de données.

Les bases de données jouent un rôle fondamental dans notre monde devenu numérique où il est extrêmement facile de dupliquer l'information. Voilà pourquoi nous allons cette année les étudier.

**Donnez deux exemples de domaines d'activité où les bases de données jouent un rôle primordial.**

Dans une base de données, l'information est stockée dans des fichiers, mais à la différence des fichiers au format CSV, il n'est pas possible de travailler sur ces données avec un simple éditeur de texte.

Pour manipuler les données présentes dans une base de données (écrire, lire ou encore modifier), il est nécessaire d'utiliser un type de logiciel appelé "**Système de Gestion de Base de Données**" très souvent abrégé en **SGBD**.

Il existe une multitude de SGBD : des **gratuits**, des **payants**, des **libres** ou bien encore des **propriétaires**.

**Donnez le sens du point de vue informatique de ces quatre termes.**

Les SGBD permettent de grandement simplifier la gestion des bases de données.



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, et il est 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.

Ce sont 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 un 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 œuvre, mais aussi beaucoup plus limitées comme les fichiers au format CSV.

# LES BASES DE DONNEES RELATIONNELLES

## (Définitions... à connaître !)

Il existe différents types de bases de données, par exemple, les bases de données hiérarchiques, les bases de données objet, les bases de données NoSQL ou bien encore les **bases de données relationnelles**.

Les bases de données relationnelles ont été mises au point en 1970 par Edgar Franck CODD, informaticien britannique (1923-2003).

Ces bases de données sont basées sur la théorie mathématique des ensembles.

La **notion de relation** est au cœur des **bases de données relationnelles**.

Une relation peut être vue comme un tableau à 2 dimensions, composé d'une **en-tête** et d'un **corps**. Le corps est lui-même composé de **t-uplets** (lignes) et d'**attributs** (colonnes).

L'en-tête contient les intitulés des attributs, le corps contient les données proprement dites.

À noter que l'on emploie aussi le terme "**table**" à la place de "**relation**".

Voici un exemple de relation : Relation "LIVRES"

Le t-uplet encadré en jaune sur le schéma ci-contre contient les éléments suivant : 11, La Planète des singes, Boule, 1963 et 8.

L'attribut "titre" est composé des éléments suivants : 1984, Dune, Fondation, Le meilleur des mondes, Fahrenheit 451, Ubik, Chroniques martiennes, La nuit des temps, Blade Runner, Les Robots, La Planète des singes, Ravage, Le Maître du Haut Château, Le monde des Å, La Fin de l'éternité et De la Terre à la Lune.

id	titre	auteur	ann_publi	note
1	1984	Orwell	1949	10
2	Dune	Herbert	1965	8
3	Fondation	Asimov	1951	9
4	Le meilleur des mondes	Huxley	1931	7
5	Fahrenheit 451	Bradbury	1953	7
6	Ubik	K.Dick	1969	9
7	Chroniques martiennes	Bradbury	1950	8
8	La nuit des temps	Barjavel	1968	7
9	Blade Runner	K.Dick	1968	8
10	Les Robots	Asimov	1950	9
11	La Planète des singes	Boule	1963	8
12	Ravage	Barjavel	1943	8
13	Le Maître du Haut Château	K.Dick	1962	8
14	Le monde des Å	Van Vogt	1945	7
15	La Fin de l'éternité	Asimov	1955	8
16	De la Terre à la Lune	Verne	1865	10

Annotations du schéma :

- en-tête : pointe vers la première ligne (id, titre, auteur, ann\_publi, note).
- corps : pointe vers l'ensemble des lignes de données.
- 1 t-uplet : pointe vers la ligne 11 (La Planète des singes, Boule, 1963, 8).
- l'attribut "titre" : pointe vers la colonne titre.

**Faites la liste des éléments appartenant à l'attribut "auteur".**

Pour chaque attribut d'une relation, il est nécessaire de définir un domaine : Le domaine d'un attribut donné correspond à un ensemble fini ou infini de valeurs admissibles.

Par exemple, le domaine de l'attribut "id" correspond à l'ensemble des entiers (la colonne "id" devra obligatoirement contenir des entiers).

Autre exemple, le domaine de l'attribut "titre" correspond à l'ensemble des chaînes de caractères.

Dernier exemple, le domaine de l'attribut "note" correspond à l'ensemble des entiers positifs.

**Quel est, selon vous, le domaine de l'attribut "auteur" ?**

Au moment de la création d'une relation, il est nécessaire de renseigner le domaine de chaque attribut. Le SGBD s'assure qu'un élément ajouté à une relation respecte bien le domaine de l'attribut correspondant :

- Si par exemple vous essayez d'ajouter une note non entière (par exemple 8,5), le SGBD signalera cette erreur et n'autorisera pas l'écriture de cette nouvelle donnée.

Autre contrainte très importante dans les bases de données relationnelles, une relation ne peut pas contenir 2 t-uplets identiques.

- Par exemple, la situation ci-dessous n'est pas autorisée (ici aussi c'est le SGBD qui veille au grain) :

id	titre	auteur	ann_publi	note
1	1984	Orwell	1949	10
2	Dune	Herbert	1965	8
2	Dune	Herbert	1965	8
3	Fondation	Asimov	1951	9

Afin d'être sûr de respecter cette contrainte des t-uplets identiques, on définit la notion de "**clef primaire**".

Une **clef primaire** est un attribut dont la valeur permet **d'identifier de manière unique** un t-uplet de la relation. Autrement dit, si un attribut est considéré comme clef primaire, on ne doit pas trouver dans toute la relation 2 fois la même valeur pour cet attribut.

**Si on se réfère à l'exemple de la relation ci-dessous :**

### Relation LIVRES

id	titre	auteur	ann_publi	note
1	1984	Orwell	1949	10
2	Dune	Herbert	1965	8
3	Fondation	Asimov	1951	9
4	Le meilleur des mondes	Huxley	1931	7
5	Fahrenheit 451	Bradbury	1953	7
6	Ubik	K.Dick	1969	9
7	Chroniques martiennes	Bradbury	1950	8
8	La nuit des temps	Barjavel	1968	7
9	Blade Runner	K.Dick	1968	8
10	Les Robots	Asimov	1950	9
11	La Planète des singes	Boulle	1963	8
12	Ravage	Barjavel	1943	8
13	Le Maître du Haut Château	K.Dick	1962	8
14	Le monde des Â	Van Vogt	1945	7
15	La Fin de l'éternité	Asimov	1955	8
16	De la Terre à la Lune	Verne	1865	10

**L'attribut "note" peut-il jouer le rôle de clef primaire ?** *Non, car il est possible de trouver 2 fois la même note.*

**L'attribut "ann\_publi" peut-il jouer le rôle de clef primaire ?** *Non, car il est possible de trouver 2 fois la même année.*

**L'attribut "auteur" peut-il jouer le rôle de clef primaire ?** *Non, car il est possible de trouver 2 fois le même auteur.*

**L'attribut "titre" peut-il jouer le rôle de clef primaire ?** *A priori oui, car dans notre exemple, l'attribut "titre" ne comporte pas 2 fois le même titre de roman.*

Mais, ce n'est pas forcément une bonne idée, car il est tout à fait possible d'avoir un même titre pour 2 romans différents. Par exemple, en 2013, l'Américaine Jill Mc CORKLE et l'Anglaise Kate ATKINSON publiaient avec seulement six jours d'écart un livre intitulé "Life After Life" !

Il nous reste donc l'attribut "id". En fait, l'attribut "id" ("id" comme "**identifiant**") a été placé là pour jouer le rôle de **clef primaire**. En effet, à chaque fois qu'un roman est ajouté à la relation, son "id" correspond à l'incrémementation de l'id (**id du nouveau=id de l'ancien+1**) du roman précédemment ajouté.

Il est donc impossible d'avoir deux romans avec le même id.

Ajouter un attribut "id" afin qu'il puisse jouer le rôle de clef primaire est une **pratique courante, non obligatoire** mais **conseillée** dans les bases de données relationnelles.

Dans le cas précis qui nous intéresse, il aurait été possible de ne pas utiliser d'attribut "id", car **chaque livre édité possède un numéro qui lui est propre : l'ISBN**, cet ISBN aurait donc pu jouer le rôle de clef primaire.

### **Voici un extrait d'une relation référençant des films :**

#### **Relation FILMS**

id	titre	realisateur	ann_sortie	note_sur_10
1	Alien, le huitième passager	Scott	1979	10
2	Dune	Lynch	1985	5
3	2001 : l'odyssée de l'espace	Kubrick	1968	9
4	Blade Runner	Scott	1982	10

**Listez les différents attributs de cette relation.**

**Donnez le domaine de chaque attribut.**

**Pour chaque attribut dire si cet attribut peut jouer le rôle de clef primaire, vous n'oublierez pas de justifier vos réponses.**

Nous désirons maintenant un peu enrichir la relation « LIVRES » en ajoutant des informations supplémentaires sur les auteurs, nous obtenons alors :

### Relation LIVRES\_AUTEURS

i d	titre	nom_aut eur	prenom_aut eur	date_nai_aut eur	langue_ecriture_a uteur	ann_pu bli	not e
1	1984	Orwell	George	1903	anglais	1949	10
2	Dune	Herbert	Frank	1920	anglais	1965	8
3	Fondatio n	Asimov	Isaac	1920	anglais	1951	9
4	Le meilleur des mondes	Huxley	Aldous	1894	anglais	1931	7
5	Fahrenh eit 451	Bradbury	Ray	1920	anglais	1953	7
6	Ubik	K.Dick	Philip	1928	anglais	1969	9
7	Chroniq ues martienn es	Bradbury	Ray	1920	anglais	1950	8
8	La nuit des temps	Barjavel	René	1911	français	1968	7
9	Blade Runner	K.Dick	Philip	1928	anglais	1968	8
10	Les Robots	Asimov	Isaac	1920	anglais	1950	9
11	La Planète des singes	Boulle	Pierre	1912	français	1963	8
12	Ravage	Barjavel	René	1911	français	1943	8
13	Le Maître du Haut Château	K.Dick	Philip	1928	anglais	1962	8
14	Le monde des Â	Van Vogt	Alfred Elton	1912	anglais	1945	7
15	La Fin de l'éternité	Asimov	Isaac	1920	anglais	1955	8
16	De la Terre à la Lune	Verne	Jules	1828	français	1865	10

Nous avons ajouté 3 attributs ("prenom\_auteur", "date\_nai\_auteur" et "langue\_ecriture\_auteur").

Nous avons aussi renommé l'attribut "auteur" en "nom\_auteur".

Comme vous l'avez peut-être remarqué, il y a pas mal **d'informations dupliquées**, par exemple, on retrouve 3 fois "**K.Dick Philip 1928 anglais**", même chose pour "**Asimov Isaac 1920 anglais**".

**Cette duplication est-elle indispensable ?** *Non !*

**Est-elle souhaitable ?** *Non plus !*

En effet, dans une base de données, on évite autant que possible de dupliquer l'information.

Si nous dupliquons autant de données inutilement c'est que notre structure ne doit pas être la bonne !

**Mais alors, comment faire pour avoir aussi des informations sur les auteurs des livres ?** *La solution est relativement simple : travailler avec 2 relations (ou tables) au lieu d'une seule et créer un "lien" entre ces 2 relations :*

### Relation AUTEURS

id	nom	prenom	ann_naissance	langue_ecriture
1	Orwell	George	1903	anglais
2	Herbert	Frank	1920	anglais
3	Asimov	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	Boulle	Pierre	1912	français
9	Van Vogt	Alfred Elton	1912	anglais
10	Verne	Jules	1828	français

### Relation LIVRES

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	7
5	Fahrenheit 451	5	1953	7
6	Ubik	6	1969	9
7	Chroniques martiennes	5	1950	8
8	La nuit des temps	7	1968	7
9	Blade Runner	6	1968	8
10	Les Robots	3	1950	9
11	La Planète des singes	8	1963	8
12	Ravage	7	1943	8
13	Le Maître du Haut Château	6	1962	8
14	Le monde des Â	9	1945	7
15	La Fin de l'éternité	3	1955	8
16	De la Terre à la Lune	10	1865	10

Nous avons créé une **relation AUTEURS** et dans la **relation LIVRES**, nous avons remplacé l'attribut "**auteur**" par un attribut "**id\_auteur**".

L'attribut "**id\_auteur**" de la **relation LIVRES** permet de créer un lien avec la **relation AUTEURS**. "**id\_auteur**" correspond à l'attribut "**id**" de la **relation AUTEURS**.

L'introduction d'une **relation AUTEURS** et la mise en place de liens entre cette relation et la **relation LIVRES** permettent d'éviter la redondance d'informations.

Pour établir un lien entre 2 relations  $R_A$  et  $R_B$ , on ajoute à  $R_A$  un attribut  $x$  qui prendra les valeurs de la **clé primaire de  $R_B$** . Cet attribut  $x$  est appelé **clef étrangère**.

Dans l'exemple ci-dessus, l'attribut "**id\_auteur**" de la **relation LIVRES** permet bien d'établir un lien entre la **relation LIVRES** et la **relation AUTEURS**.

"**id\_auteur**" correspond bien à la **clef primaire** de la **relation AUTEURS**, conclusion : "**id\_auteur**" est une **clef étrangère**.

**En partant de la relation FILMS ci-dessous, créez une relation REALISATEURS**

Attributs de la **relation REALISATEURS** : id, nom, prenom et ann\_naissance.  
Recherchez ces informations nécessaires sur le Web.

Modifiez ensuite la **relation FILMS** afin d'établir un lien entre la **relation FILMS** et la **relation REALISATEURS**. Vous préciserez l'attribut qui jouera le rôle de **clef étrangère**.

### **Relation FILMS**

<b>id</b>	<b>titre</b>	<b>realisateur</b>	<b>ann_sortie</b>	<b>note_sur_10</b>
1	Alien, le huitième passager	Scott	1979	10
2	Dune	Lynch	1985	5
3	2001 : l'odyssée de l'espace	Kubrick	1968	9
4	Blade Runner	Scott	1982	10



L'ensemble des relations présentes dans une base de données est appelé **schéma relationnel**.

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

- Les noms des différentes relations (tables)
- La liste des attributs avec leur domaine respectif pour chaque relation.
- La clef primaire et éventuellement la clef étrangère pour chaque relation.

**Exemple : Pour les relations LIVRES et AUTEURS.**

AUTEURS : id, nom, prenom, ann\_naissance, langue\_ecriture  
LIVRES : id, titre, #id\_auteur, ann\_publi, note

Les attributs soulignés (id) sont des clefs primaires, le # signifie que l'on a une clef étrangère.

**Donnez le schéma relationnel de la base de données qui comporte les relation FILMS et REALISATEURS.**