

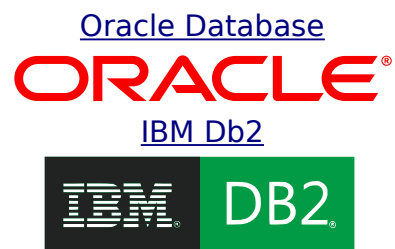
# Les bases de données

Depuis l'avènement de l'informatique, les quantités de données à traiter n'ont cessé de croître. Tout naturellement, il a fallu mettre en place des solutions pour assurer la persistance de ces données tout en les rendant facilement et rapidement accessibles indépendamment des matériels utilisés. Avec le développement des réseaux ces solutions ont dû rendre possible les accès concurrents par plusieurs ordinateurs tout en garantissant l'intégrité des données en cas de modification, autrement dit leur cohérence, leur fiabilité et leur pertinence. Ces évolutions ont donné naissance aux Systèmes de Gestion de Bases de Données Relationnelles (SGBDR).

## Les SGBDR

Les Systèmes de Gestion de Bases de Données Relationnelles sont des logiciels, serveurs réseaux pour la plupart, qui permettent d'accéder à des [bases de données relationnelles](#) rassemblant chacune diverses tables de données associées les unes aux autres.

Parmi les principaux SGBDR on trouve :



La grande majorité des SGBDR sont des logiciels complexes et les données qu'ils gèrent sont souvent cruciales : commandes, factures, caractéristiques des articles mis en ligne... C'est pourquoi ils intègrent des mécanismes de protection en termes de sécurité et de disponibilité :

- authentification des accès par mot de passe ou clé cryptographique,
- chiffrement des échanges réseau,
- administration et création d'utilisateurs,
- séparation des bases de données pour qu'un utilisateur ne puisse accéder qu'à l'une d'entre elles,
- protection du contenu des tables afin d'en interdire la modification,
- duplication des données sur plusieurs serveurs de secours et synchronisation,
- répartition de charge...

Parmi les SGBDR, la situation de SQLite est un peu particulière. En effet, celui-ci repose sur un accès fichier (et non réseau). Il est destiné à gérer de petites bases de données non administrées, à usage individuel. Et puisque son code source a été placé dans le domaine public, il peut s'intégrer facilement à d'autres logiciels. Ceci en fait l'un des SGBDR les plus utilisés dans le monde, puisque à ce jour on peut dénombrer plus d'un milliard de copies et plus de 1000 milliards de bases de données SQLite.

## Le modèle relationnel

Le fonctionnement des SGBDR s'appuie sur le **modèle relationnel** créé par E.F. Codd en 1970, qui est une formalisation mathématique des données en table, similaires à celles rencontrées avec le format [CSV](#) (*Comma-separated values*).

Voici un exemple de table présentant des ouvrages en bibliothèque.

numero	titre	parution	auteur	naissance	mort
142	Le Dernier Jour d'un condamné	1829	Victor Hugo	1802	1885
241	Voyage au centre de la Terre	1864	Jules Verne	1828	1905
323	Les Petites Filles modèles	1858	Comtesse de Ségur	1799	1874
358	Les Misérables	1862	Victor Hugo	1802	1885
419	Les Malheurs de Sophie	1858	Comtesse de Ségur	1799	1874
492	Le Tour du monde en quatre-vingts jours	1872	Jules Verne	1828	1905
677	Mémoires d'un âne	1860	Comtesse de Ségur	1799	1874
760	Vingt mille lieues sous les mers	1869	Jules Verne	1828	1905
806	Notre-Dame de Paris	1831	Victor Hugo	1802	1885
815	De la Terre à la Lune	1865	Jules Verne	1828	1905

Au niveau du modèle relationnel, cette table est la représentation de la **relation** Ouvrage composée de 6 **attributs** (noms de colonnes) : **numero**, **titre**, **date**, **auteur**, **naissance** et **mort**, qui possèdent chacun un **domaine** de valeurs spécifique. Ainsi :

- **numero** est un entier
- **titre** est un texte (une chaîne de caractères plus ou moins longue)
- **parution** est une année
- **auteur** est une chaîne de caractères (de taille limitée par exemple à 128)
- **naissance** est une année
- **mort** est une année

Ces informations sont condensées dans ce qu'on appelle le **schéma relationnel** de la relation Ouvrage afin d'en donner la structure :

Ouvrage(numero:entier, titre:texte, parution:année, auteur:caractères[128], naissance:année, mort:année)

Enfin en mettant de côté l'omniprésence des attributs et des domaines :

- les lignes de cette table, qu'on appelle aussi des enregistrements, sont la représentation de **tuples** ou **6-uplet** (sextuplet) tel que celui-ci :  
(142, "Le Dernier Jour d'un condamné", 1829, "Victor Hugo", 1802, 1885)
- la relation devient quant à elle un ensemble de tuples, et
- la base de données, un ensemble de relations.

**Remarque** Comme ce modèle se fonde sur la théorie des ensembles, il ne peut y avoir deux tuples identiques dans une relation, et donc deux lignes identiques dans une table. Pas plus que d'ordre entre les tuples ou lignes.

Pour résumer, le modèle relationnel apporte une vision plus théorique de la table :

Données en table	Modèle relationnel
table	relation
colonne (titre)	attribut
ligne / enregistrement	tuple / n-uplet
valeurs autorisées dans une colonne (type spécialisé)	domaine d'un attribut

## Conception d'une base de donnée

À y regarder de plus près, la table Ouvrage n'est pas très satisfaisante. On y retrouve le nom des auteurs dupliqué plusieurs fois. Cette redondance d'informations occupe inutilement de la place en mémoire (RAM et disque), et toute correction du nom doit être effectuée partout où celui-ci apparaît. Par ailleurs il y a un risque d'inconsistance lors de la copie des années de naissance et de mort à l'insertion d'un nouvel ouvrage pour un auteur existant. Enfin, il n'est pas possible de conserver les informations sur un auteur dans le cas où l'on supprime tous ses ouvrages.

Cette organisation en une seule table pose donc problèmes. Et compte tenu de l'analyse précédente il convient de concevoir autrement cette base de données en dissociant ouvrage et auteur, tout en maintenant le lien entre les deux.

Voici les tables obtenues en séparant les données.

On rassemble les informations sur les auteurs en créant une table dédiée dans laquelle chaque auteur est identifié par un nombre entier `id_auteur` appelé clé primaire.

id_auteur	nom	naissance	mort
75	Victor Hugo	1802	1885
81	Jules Verne	1828	1905
63	Comtesse de Ségur	1799	1874

numero	titre	parution	id_auteur
142	Le Dernier Jour d'un condamné	1829	75
241	Voyage au centre de la Terre	1864	81
323	Les Petites Filles modèles	1858	63
358	Les Misérables	1862	75
419	Les Malheurs de Sophie	1858	63
492	Le Tour du monde en quatre-vingts jours	1872	81
677	Mémoires d'un âne	1860	63
760	Vingt mille lieues sous les mers	1869	81
806	Notre-Dame de Paris	1831	75
815	De la Terre à la Lune	1865	81

Dans la table des ouvrages on remplace les triplets (auteur, naissance, mort) par la clé d'identification de l'auteur définie dans la table des auteurs. La clé `id_auteur` est qualifiée de clé étrangère car elle se réfère à la clé primaire `id_auteur` de la table des auteurs.

La relation Auteur est composée de 3 **tuples** comportant 4 **attributs** :

- `id_auteur` qui tient le rôle de **clé primaire** en permettant d'identifier un auteur.
- `nom` qui est le nom de l'auteur
- `naissance` qui correspond à l'année de naissance de l'auteur
- `mort` qui correspond à l'année de la mort de l'auteur

Son **schéma relationnel** est :

Auteur(id\_auteur:entier, nom:caractères[128], naissance:année, mort:année)

La relation Ouvrage est toujours composée de 10 **tuples**, mais on ne trouve plus que 4 **attributs** :

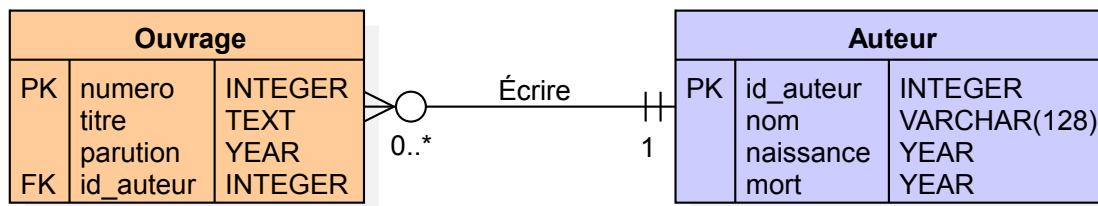
- `numero` qui est la **clé primaire** identifiant de manière unique chaque ouvrage
- `titre` qui est le titre de l'ouvrage
- `parution` qui est l'année de publication de l'ouvrage
- `id_auteur` qui est la **clé étrangère** faisant **référence** à l'auteur de l'ouvrage

Son **schéma relationnel** est :

Ouvrage(numero:entier, titre:texte, parution:année, #id\_auteur:entier)

Ces deux schémas relationnels forment le schéma relationnel de la base de données.

Le diagramme UML suivant illustre ce **schéma relationnel** et l'association entre les deux relations :



Ce diagramme rassemble les différents attributs ainsi que les clés primaire (souligné ou préfixé par PK pour *primary key*) et étrangère (préfixé par # ou par FK pour *foreign key*) de chaque relation. Il précise en outre l'association Écrire qui lie Auteur et Ouvrage. Enfin la présence des valeurs 1 et 0..\*, qu'on appelle des multiplicités dans un diagramme UML (ou cardinalités des relations, dont les positions sont interverties dans un diagramme entité-association), informe qu'un auteur a écrit entre 0 et un nombre quelconque d'ouvrages (\* représentant un entier naturel quelconque), tandis qu'un ouvrage a été écrit par 1 auteur.

## Les contraintes d'intégrité

Afin de maintenir des données valides et cohérentes tout au long de la vie de la base de données, il faut s'assurer en permanence du respect des contraintes d'intégrité suivantes qui sont fixées à la création de la base données :

- L'intégrité de domaine : les valeurs doivent appartenir au domaine fixé pour chaque attribut.
- L'intégrité de relation : chaque tuple est unique et doit être identifié par une clé primaire qui ne peut être nulle.
- L'intégrité de référence entre deux relations : toute clé étrangère doit correspondre à une clé primaire existante.

## Ce qu'il faut retenir

- En apportant une vision plus abstraite de la table avec la notion de relation, le modèle relationnel permet de s'affranchir de la façon dont les données sont organisées et stockées en mémoire.
- Une base de données se compose de plusieurs relations afin de supprimer les redondances, faciliter les mises à jour et permettre les ajouts partiels.
- La structure d'une relation est donnée par son schéma relationnel qui précise le nom des attributs et leur domaine de valeurs.
- L'ensemble des tuples forme le corps de la relation.
- Dans chaque relation, l'un des attributs (parfois un groupe) dont la valeur permet d'identifier de manière unique un tuple, est appelé clé primaire.
- Certaines relations possèdent un attribut qui se réfère à la clé primaire d'une autre relation à laquelle elle est ainsi liée ; un tel attribut est appelé clé étrangère.
- Les contraintes d'intégrité de domaine, de relation et de référence doivent être constamment maintenues afin d'assurer la cohérence des données.