



Les bases de données Synthèse



1. Définitions

➤ Base de données :

- Collection de données **logiquement cohérentes** ;
- Ensemble structuré d'informations **accessibles par une communauté d'utilisateurs** ;
- Ensemble de données **servant de support** à une ou plusieurs applications informatiques (des données + liens entre ces données) ;



➤ Système de Gestion de Bases de Données (SGBD) :

C'est un ensemble d'outils logiciels permettant d'**insérer**, de **modifier** et de **rechercher** efficacement dans une **grande masse d'informations** partagées par de **multiples utilisateurs** (en anglais **DBMS** : **DataBase Management System**).

Séparation entre données et traitements : le logiciel **SGBD** peut évoluer sans affecter les Bases de Données.

➤ Structure d'un SGBD :

Le SGBD peut se décomposer en trois sous-systèmes :

1. Le SGBD externe :

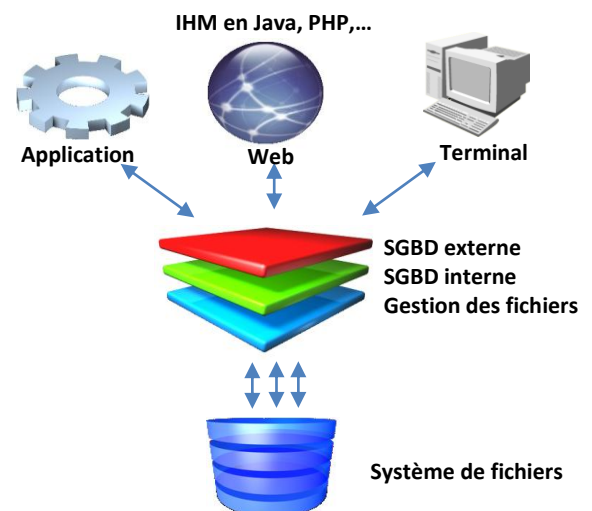
- permet l'accès aux données par l'utilisateur de façon simple par le biais de **requêtes** ;
- **l'interface avec l'utilisateur, IHM** (analyse et interprétation des requêtes) ;

2. Le SGBD interne gère l'organisation des données :

- **Gestion des problèmes** de partage, de cohérence et d'intégrité des données ;
- **Protection des données** contre des accès non autorisés (**droits d'accès**) ;
- **Accès multiples**

3. Le système de **gestions de fichiers** permet le stockage des informations sur un support physique :

- En fonction du **volume des données** et pour **garantir une rapidité d'accès**, les Bases de Données peuvent être réparties sur **plusieurs fichiers** (transparents pour l'utilisateur) ;
- Afin de sécuriser les données en cas de **défaillances matérielles**, ces fichiers peuvent être **dupliqués** sur plusieurs serveurs (synchronisation des contenus) ;



➤ Bases de Données relationnelles et SGBD-R :

Il existe différents types de bases de données : Les bases de données hiérarchiques, réseaux, **relationnelles**, déductives, objets,.... no-sql.

Les **Bases de Données Relationnelles** représentent 66% des bases de données utilisées. Elles s'appuient sur des données en tables, un algèbre relationnel, un langage déclaratif (SQL).

Parmi les **SGBDR** les plus connus, on peut citer MySQL, SQLite, Oracle Database, Microsoft SQL, Microsoft Access.

2. Les Bases de Données Relationnelles

Les Bases de Données Relationnelles s'appuient sur la notion de **Relation** (ou **Table**).

Description

Une **Relation** (ou **Table**) peut être vue comme un tableau à 2 dimensions, composé d'un en-tête et d'un corps.

- L'en-tête contient les **intitulés** des **attributs** (des colonnes) ;
- Le corps contient les **données** proprement dites. Le corps est lui-même composé de **t-uplets** (lignes) et d'**attributs** (colonnes).

Table « Livres » (ou Relation « Livres »)

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

- Le contenu de **chaque ligne** de cette table est appelé **t-uplet** ou **n-uplet** ou **occurrence** et correspond dans notre exemple aux **informations d'un seul livre**.
- Le contenu de **chaque colonne** de cette table est appelé **Attribut**.
 - Chaque **Attribut** est **identifié** par un **nom unique (intitulé)** défini dans l'en-tête de la Table (Relation).
 - Chaque **Attribut** peut prendre un **ensemble de valeurs** (ou éléments) appelé **Domaine**.
- Le **Domaine** définit l'ensemble fini ou infini de valeurs que peut prendre un **attribut**.
Exemples : Entier, Réel, Booléen, Date, Chaînes de caractères..., {0,1,2,3}, {a,b,c}

Cohérence des données d'une Table (domaine et unicité)

- **Contrainte de Domaine** : Le SGBD s'assure qu'un élément ajouté à une **Table** (Relation) respecte bien le **domaine** de l'attribut correspondant.
 - Au moment de la création d'une **Table** (Relation), il est nécessaire de renseigner le **domaine** de chaque **attribut**.
- **Contrainte d'unicité** : Chaque occurrence (T-uplet) d'une Table (Relation) est unique.

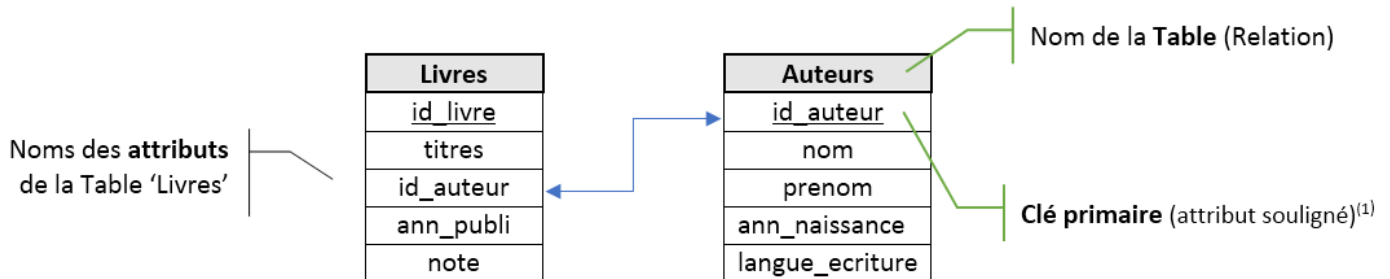
Clé d'une Relation (Table) pour identifier une occurrence, un t-uplet

- **Clé primaire** : permet d'identifier de manière unique un **t-uplet** de la relation.

Schéma relationnel

A des fins d'optimisation de la quantité de données, les informations peuvent se retrouver réparties sur plusieurs Relations (Tables). Des liens permettent de les relier.

Représentation graphique : (exemple avec deux Relations/Tables)



Représentation littérale :

Livres(id_livre, titre, #id_auteur, ann_publi, note)

Auteurs(id_auteur, nom, prenom, ann_naissance, langue_ecriture)

Remarque : Clé primaire : attribut souligné Clé étrangère : attribut précédé du symbole #.

(1) Si plusieurs attributs constituent la clé primaire, tous sont soulignés.

Relations entre Tables (Notion de Clé étrangère)

- **Clé étrangère** : ensemble d'attributs servant de lien entre deux Relations (Tables), clé primaire d'une autre relation.

Dans l'exemple précédent :

L'attribut 'id_auteur' de la relation '**Livres**' est une **clé étrangère** servant de lien vers la relation '**Auteurs**'. Dans la relation '**Auteurs**', 'id_auteur' doit être une **clé minimale** (c'est même ici la clé primaire).

Plusieurs occurrences (livres) de la relation '**Livres**' peuvent référencer le même auteur, mais une occurrence (un livre) de la relation '**Livres**' ne peut référencer qu'un seul auteur (un auteur étant une et une seule occurrence de la relation '**Auteurs**').

Contrainte de clé étrangère

Déclarer un attribut⁽²⁾ **clé étrangère** d'une relation, dans une base de données relationnelle, est une **contrainte** qui garantit l'intégrité référentielle entre deux relations (tables).

Généralement le SGBD effectuera la vérification de cette contrainte de clé étrangère qui garantit :

- que toutes valeurs de l'attribut **clé étrangère** de la relation référençant **existent** dans la relation référencée.
- que cet attribut dans la relation référencée doit faire partie d'une contrainte de clé primaire ou d'une **contrainte d'unicité**.

(2) Une clé étrangère peut porter sur un seul ou plusieurs attributs.

Conséquence.

Nous ne pouvons pas ajouter dans la table 'Livres' un roman dont l'auteur n'est pas encore référencé dans la table 'Auteurs'. Cela introduit une priorité dans l'alimentation des tables d'une base de données.

SQL : Structured Query Language (pour langage de requêtes structurées) est un langage déclaratif permettant d'écrire des requêtes informatiques pour **créer, alimenter, mettre à jour, interroger** les **SGBD-R**.

https://fr.wikipedia.org/wiki/Structured_Query_Language

Le SGBD **MySQL** est conforme à la norme **SQL2** (ou SQL-92) garantissant la prise en compte des requêtes normalisées correspondantes.

Quelques requêtes SQL

Requêtes SQL	Descriptions
SHOW DATABASES;	Visualiser, lister les différentes bases de données du serveur
CREATE DATABASE nombasedonnees ;	Créer une base de données
USE nombasedonnee ;	Sélectionner une base de données
SHOW TABLES;	Afficher la liste des tables contenues dans la base de données sélectionnée
DESCRIBE nomdelatable ;	Afficher la description d'une table
SELECT * FROM nomtable	Afficher tout le contenu d'une table
CREATE TABLE nom_table (nom_attribut1 domaine1, nom_attribut2 domaine2, ... PRIMARY KEY (attribut_x));	Créer une table (voir exemple page suivante) NOT NULL permet de spécifier que l'attribut ne peut avoir la valeur NULL PRIMARY KEY (attribut) permet de désigner l'attribut choisi comme clé primaire
INSERT INTO nom_table VALUES (x, y, z ...);	Insérer des données dans une table Ex : <i>INSERT INTO Films VALUES (1,'Alien, le huitième passager', 'Scott', 1979,NULL);</i> Le SGBD vérifie la cohérence des données insérées dans une table en fonction de la définition de celle-ci.
UPDATE nom_table SET nom_attribut_x = valeur_x, ... nom_attribut_y = valeur_z WHERE condition_de_sélection ;	Mettre à jour une ou des occurrences Les nouvelles valeurs peuvent être le résultat d'une expression Ex : <i>UPDATE Films SET titre= 'Blade Runner 2', note = note+2 WHERE id =4 ;</i> <div>Une virgule (,) ENTRE chaque affectation (ex : nom_attribut_x = valeur_x) Pas de virgule sur la dernière affectation ou si une seule affectation</div>
DELETE FROM nom_table WHERE condition_de_sélection;	Effacer des occurrences en fonction d'une condition de sélection. Ex : <i>DELETE FROM Films WHERE realisateur ='Scott' ;</i> En l'absence de clause WHERE toutes les lignes de la table seront supprimées.
DROP DATABASE nombasedonnee;	Effacer une base de données
DROP TABLE nom_table;	Effacer une table
SOURCE fichier.sql;	Importer et exécuter les requêtes Sql contenues dans le fichier

Condition de sélection

La **condition de sélection** peut porter sur un ou plusieurs attributs. *En l'absence d'une clause WHERE, toutes les lignes sont mises à jour.*

[Quelques opérateurs pour écrire les conditions de la sélection](#) (pour commencer)

Opérateurs et requêtes SQL : Différents opérateurs permettent de définir les conditions

Opérateurs arithmétiques : =, <, >, <=, >=, <> ou !=, ...

Opérateurs logiques : and, or, not, ...

Opérateurs spécifiques : IS NULL, IS NOT NULL

Domaines et Types

L'emploi d'un type (domaine de valeur) adapté permet au SGBD de **vérifier la cohérence** des données mais aussi d'**optimiser l'occupation mémoire**.

https://www.w3schools.com/sql/sql_datatypes.asp

Les types standards

- INTEGER ou INT, SMALLINT *permettent de coder des entiers sur 4 octets ou 2 octets*
- NUMERIC(X) *désigne un entier de X chiffres au maximum.*
- DECIMAL(X,Y) ou NUMERIC(X,Y) *où X et Y sont optionnels et désignent respectivement le nombre de chiffres maximum pouvant composer le nombre, et le nombre de chiffres après la virgule.*
- FLOAT(X), REAL
- CHAR(X) *définit des chaînes de longueur fixe (complétée à droite par des espaces, si la longueur est inférieure à X)*
- VARCHAR(X) *où X est obligatoire et désigne la longueur de la chaîne*
- DATE (AAAA-MM-JJ)
- DATETIME (AAAA-MM-JJ HH:MM:SS)

<https://dev.mysql.com/doc/refman/5.7/en/data-types.html>

La valeur NULL

L'absence de valeur, représentée par la valeur NULL, est une information fondamentale en SQL, qu'il ne faut pas confondre avec une chaîne de caractères qui ne contiendrait que le caractère 'espace' ou bien la valeur 0.

Exemple

```
CREATE TABLE Livres (
  id_livre INT,
  titre VARCHAR(25) NOT NULL,
  auteur VARCHAR(20) NOT NULL,
  ann_publi DECIMAL(4),
  prix DECIMAL(5,2),
  PRIMARY KEY(id_livre)
);
```

Description

Field	Type	Null	Key	Default	Extra
id_livre	int(11)	No	pri		
titre	varchar(25)	No			
auteur	varchar(20)	No			
ann_publi	decimal(4,0)	Yes			
prix	decimal(5,2)	Yes			

Ajout de contraintes

Il est possible d'ajouter des contraintes (CONSTRAINT) à la définition d'une table/Relation permettant d'affiner le contrôle des données saisies (exemple : vérifier qu'une donnée est comprise dans un intervalle de valeurs).

Connexion en mode console à un serveur MySql

```
pi@raspberrypi:/ $ sudo mysql --user=allagui --password=rabia
```

Votre prompt doit être de la forme **mysql>** ou **MariaDB [...]>**