

# Bases de données

## 1. Définition

Une base de données est une collection de données structurées et organisées relatives à un ou plusieurs domaines du monde réel. Elle est exhaustive, non redondante, structurée et persistante.

L'utilisation d'une base de données permet entre autres de :

- centraliser l'information,
- assurer l'indépendance entre les données et les programmes,
- établir des liaisons entre les données,
- partager les données.

## 2. Modèles

Il existe plusieurs modèles de base de données, dont quatre sont présentés schématiquement en figure 4 :

- Le modèle hiérarchique,
- Le modèle réseau,
- Le modèle orienté objet,
- Le modèle relationnel.

Le modèle le plus utilisé est le modèle relationnel.

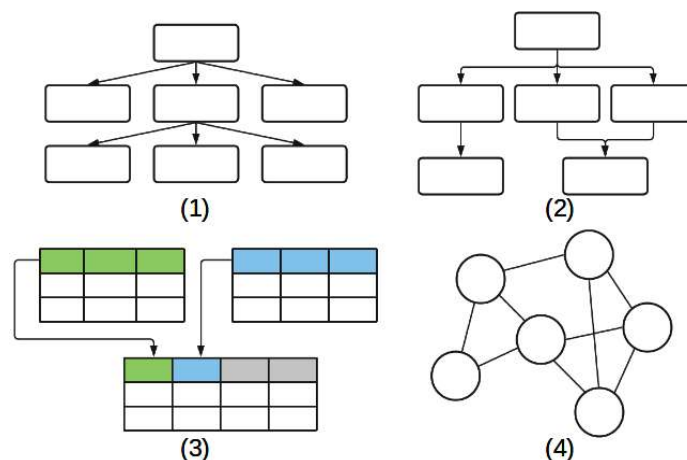


Figure 3: Modèles de bases de données (1) modèle hiérarchique (2) modèle orienté objet (3) modèle relationnel (4) modèle réseau

## 3. Le modèle relationnel

Dans le modèle relationnel, les objets du monde réel sont modélisés à l'aide d'**entités**.

Dans une **entité**, chaque objet est décrit par l'ensemble de ses caractéristiques ou **attributs**.

Les propriétés d'un objet sont définies par les **valeurs** de ses **attributs**.

Lorsqu'une information provient d'une ou de plusieurs entités reliées entre elles on parle d'**association**.

## Exemple :

Entités	Attributs	Valeurs
Qui ? Quoi ?	Quelles sont ses caractéristiques ?	Quelles sont ses propriétés ?
Élève	Nom Prénom Pseudo-nom Date de naissance Genre Téléphone	Soussi Mohamed Essoussi 15/05/2004 G 92605005
Article	Désignation Prix_Unitaire	Pull 15
Paielement	Montant Date_Paiement	20 13/12/2022

Les **entités** définies dans le tableau précédent nécessitent des **associations** pour dire qu'un "Élève achète un Article", ou qu'un "Élève effectue un Paielement".

Souvent, une **association** possède aussi des **attributs**, par exemple le *nombre d'articles achetés* par un Élève et en quelle *date*.

#### 4. Système de Gestion de Bases de Données (SGBD)

Le logiciel qui permet de gérer une base de données est appelé Système de Gestion de Bases de Données (SGBD). Il permet, notamment, de décrire, modifier, interroger et administrer les données d'une base de données. Les bases de données relationnelles sont également appelées SGBDR.

Il est constitué de deux composants principaux, figure 4 :

- Le **moteur** implémente l'ensemble de toutes les fonctionnalités relatives aux bases de données,
- L'**interface** est située entre les applications utilisateur et le SGBD, pour permettre l'accès facile et convivial aux données.

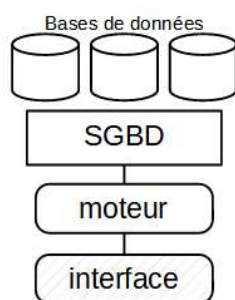


Figure 4: Les composants principaux d'un Système de Gestion de Bases de Données

## Exemples :



Figure 5: Les SGBD les plus connus

Chaque SGBDR, figurant dans la figure 5, est réservé à un usage particulier :

- Lorsqu'on veut une base de données de faible taille qui peut être embarquée dans un smartphone ou intégré à une petite application on peut opter pour **SQLite** et **H2**,
- Lorsqu'on développe une application web en PHP, on peut choisir entre **MariaDB** et **MySQL**, la première est la version open source de la deuxième, leurs fonctionnalités et leurs API sont presque identiques,
- Lorsqu'on travaille dans une PME on peut choisir de payer une licence **MS Access**, qui fait partie de la suite Microsoft Office, et bénéficier, ainsi, de l'ergonomie et la facilité d'utilisation des outils Microsoft. Ou on bien opter pour la base de donnée open source **PostgreSQL** qui est plus performante, mais nécessite des personnes expérimentées pour son utilisation,
- Lorsque le volume des informations à stocker est très important, qu'on veut assurer la sécurité des données, que l'information est répartie dans plusieurs endroits sur un et/ou des serveurs dédiés, on peut choisir entre les base de données **ORACLE** ou **SQL Server**.

## 5. Table

Les données d'une **base de données** sont organisées dans des **tables**.

Une **table** est un ensemble de données relatives à un même sujet (ou entité). Elle est, également, appelée **relation**.

Une **table** est un tableau contenant des **enregistrements**, appelées aussi **lignes**, ou **n-uplets**, ayant une même structure. Elle possède une structure formée par ses **champs** (ses **colonnes**). Et, elle sert à stocker les données dans des **enregistrements** ayant toujours le même nombre de champs.

Exemples :

Colonnes				Lignes
N°	Nom	Prénom	Age	
1	Tounsi	Salah	18	
2	Benzarti	Leïla	22	
3	Gabsi	Ahmed	19	

*Liste des clients*

Code article	Désignation	PU	Quantité
V10	Vis 50x3	40	2500
V20	Vis 20x2	20	1300
B100	Boulon 90x15	450	100
C60	Clou 60x2	5	5000

*Liste des articles*

Figure 6: Exemples de tables

## 6. Enregistrement

Un **enregistrement** est constitué par l'ensemble des **valeurs** attribuées à ses **champs**. Il correspond à une occurrence (ou instance) du sujet représenté par la table. On dit, souvent, qu'il représente un objet du monde réel. Il est appelé, aussi, **n-uplet** ou **ligne**.

Exemples :

Un **client** peut-être représenté comme suit la table  
**Clients** :  
(2, "Benzarti", "Leïla", 22)

Un **article** peut-être représenté comme suit dans la  
table **Articles** :  
(B100, "Boulon 90x15", 5.0, 5000)

## 7. Champ

Un **champ** correspond à une **propriété** élémentaire de l'objet décrit par une **table**. Il est aussi appelé **colonne** ou **attribut**. Un **champ** est décrit par :

- un **nom** ;
- un **type** : numérique, chaîne/texte, date, booléen ;
- une **taille** : elle dépend du **type** de champ :
  - elle représente le nombre maximal de lettres dans un texte,
  - elle représente le nombre maximal de chiffres pour un type numérique ;
- une **valeur par défaut** : la valeur attribuée au champ si aucune valeur n'a été mentionnée ;
- la **présence obligatoire** ou optionnelle d'une valeur pour le champ ;
- les **valeurs autorisées** : indique les valeurs permises pour le champ ;

Exemple :

<b>Nom de la table</b> : Articles <b>Description</b> : Détails des articles vendues par une quincaillerie						
Colonne	Description	Type de données	Taille	Obligatoire	Valeur par défaut	Valeurs autorisées
code_art	Code de l'article	Chaîne	16	Oui		
designation	Désignation de l'article	Chaîne	64	Oui		
PU	Prix unitaire de l'article	Numérique réel	8, 6	Non		$\geq 0$
quantite	Quantité en stock	Numérique entier	4	Non	0	$\geq 0$

Figure 7: Exemple de description des champs d'une table

## 8. Clé primaire

Au moins un **champ** de la **table** doit être défini comme **clé primaire**. La **clé primaire** permet d'identifier un **enregistrement** d'une façon unique. Elle peut être composée de plusieurs **champs**.

Autrement dit, la connaissance de la valeur de la **clé primaire**, permet de retrouver sans aucune ambiguïté les valeurs des autres champs, de l'unique **enregistrement**, de la **table** possédant cette clé.

Exemples :

NomMed	PrénomMed	CodeMed	NomHôpital	AdrHôpital	TelHôpital
Ben Salah	Amine	M10	Sahloul	Sousse	73425001
Amari	Lotfi	M15	Charles Nicolle	Tunis	71236147
Kaïbi	Henda	M27	Fatouma Bourguiba	Monastir	73260871
Amari	Amine	M31	Sahloul	Sousse	73425001
Dhahbi	Samir	M37	Charles Nicolle	Tunis	71236147

Figure 8: Le champ CodeMed est la clé primaire de la table

## 9. Relations

Les **entités** d'une base de données sont souvent reliées entre-elles afin de représenter fidèlement un sujet du monde réel. Le lien entre deux tables est orienté, il part de la **table fille** contenant la **clé étrangère** vers la **table mère** contenant la **clé primaire**.

**Exemple :** Dans un commerce un **client** fait une ou plusieurs **commande(s)** dans une date donnée *dateCmd*, voir figure 9. Le **client** et la **commande** font partie chacun d'une entité indépendante. Pour associer une **commande** à un **client**, on ajoute dans la table **Commandes** une colonne qui correspond à l'identifiant *idClient* du client de la table **Clients** qui a fait la commande. Cette nouvelle colonne est appelée : **clé étrangère**.

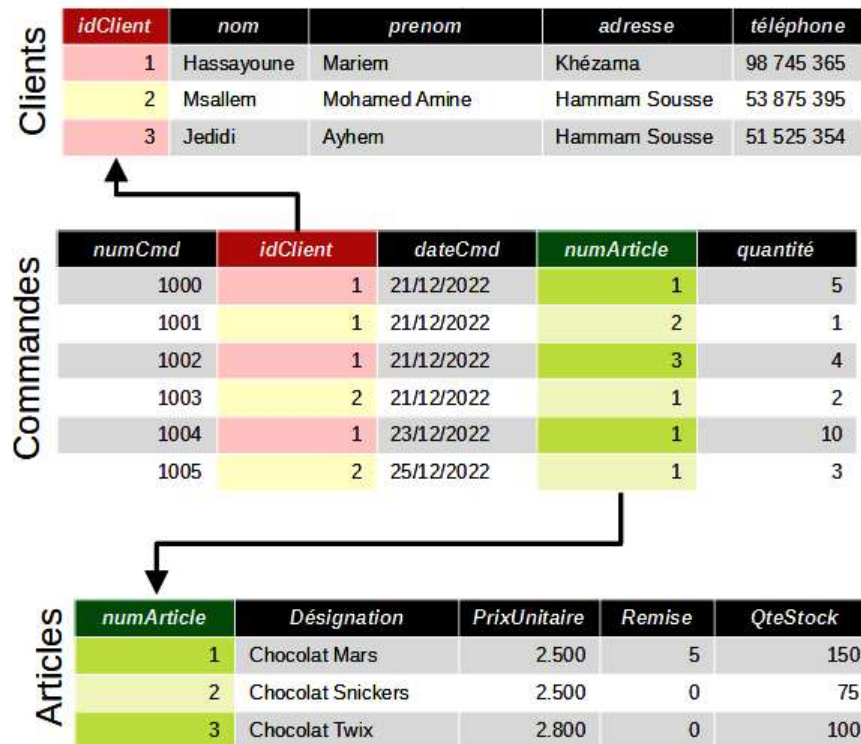


Figure 9: Les liens entre les tables Articles, Commandes, et Clients. *idClient* et *numArticle* sont des clés étrangères dans la table Commandes

On distingue trois types de relations :

- **Relation un à un** : un véhicule, voir figure 10, peut être soit une voiture ou un camion il ne peut pas être les deux à la fois d'où la **relation un à un**.
- **Relation un à plusieurs** : un article peut être commandé plusieurs fois, de même un client peut faire plusieurs commandes, figure 9, d'où la **relation un à plusieurs**.
- **Relation plusieurs à plusieurs** : la relation entre les tables Articles et Clients, figure 9, est une **relation plusieurs à plusieurs** puisqu'on dit : plusieurs clients peuvent commander plusieurs articles de notre boutique.

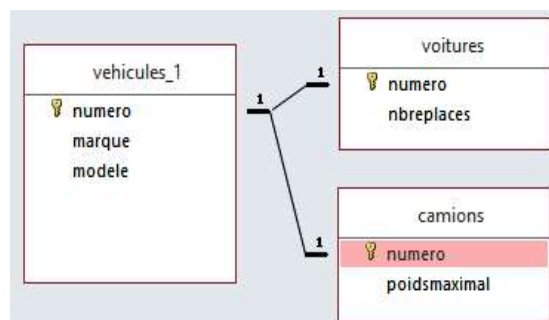


Figure 10: Exemple de relation un à un

## 10. Contraintes d'intégrité

Une **contrainte d'intégrité** est une règle appliquée à une table ou à un champ d'une table afin de garantir la cohérence, la pertinence et la validité des données. Les principaux types de contraintes d'intégrité sont :

- La **contrainte de domaine** : c'est une contrainte appliquée à une colonne d'une table afin d'assurer que la validité des valeurs prises par cette colonne. Toute valeur prise par un champ dans la table doit appartenir à un domaine de valeurs défini pour cet attribut.

**Exemple** : le champ **note** des élèves dans une table **Notes** doit être dans l'intervalle [0, 20].

- La **contrainte de relation** : chaque table doit contenir une clé primaire unique et non nulle.
- La **contrainte de référence** (contrainte d'intégrité référentielle) : Les clés étrangères permettent de relier les tables entre elles. Leurs valeurs (clés étrangères) doivent être incluses dans le domaine des valeurs de la clé primaire dont elles sont issues.

**Remarque** : pour respecter la contrainte d'intégrité référentielle, la clé étrangère devra être de même taille et de même type que la clé primaire.

**Exemple** : *numArticle* la clé étrangère de la table **Commandes** doit être de même type et de même taille que la clé primaire *numArticle* de la table **Articles**. Cette clé étrangère ne doit pas, aussi, prendre des valeurs qui n'existent pas dans la table mère, Articles.

## 11. Représentation de la structure d'une base de données

La structure d'une base de donnée peut être représentée selon deux formalismes :

- **Représentation graphique** : Exemple, voir figure 11.
- **Représentation textuelle** : Exemple, les tables de la figure 11 deviennent :

```
Articles(numArticle, designation, PrixUnitaire, Remise, QteStock)
Commandes(numCmd, idClient#, dateCmd, numArticle#, quantite)
Clients(idClient, nom, prenom, adresse, telephone)
```

**Remarque** : Dans la représentation textuelle, les **clés primaires** doivent être soulignées et les **clés étrangères** sont suivies du signe #.

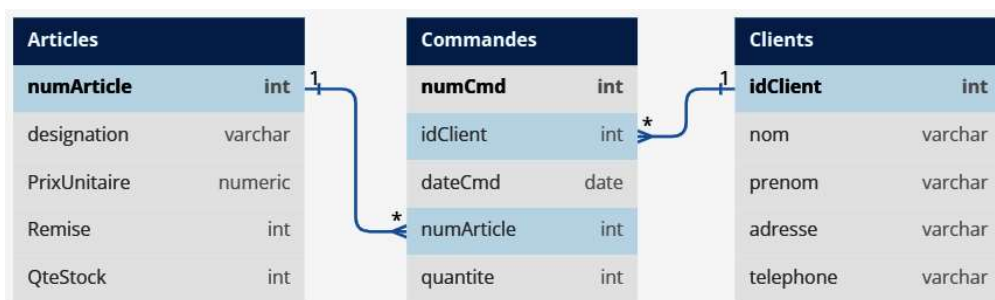


Figure 11: Exemple d'une représentation graphique d'une base de donnée

La représentation graphique dépend de l'outil utilisé. La figure 12, montre un exemple de représentation graphique dans MS Access.

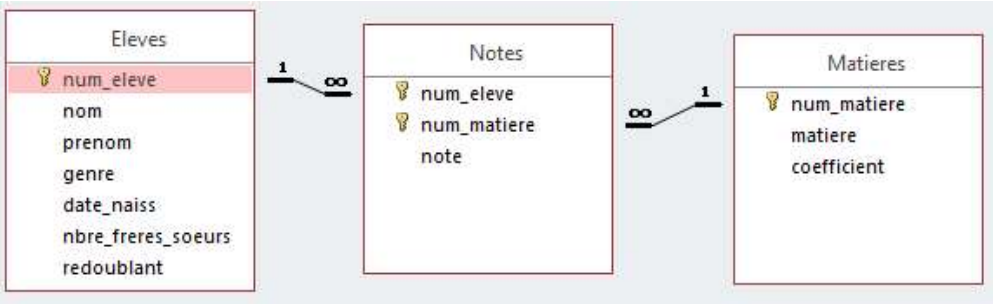


Figure 12: Exemple d'une représentation graphique d'une base de donnée dans MS Access