

# Base de données : Notions fondamentales

## 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** puisqu'elle contient la plupart, voire la totalité, des informations pertinentes et nécessaires pour le domaine ou le sujet traité.
- **Non redondante** afin de maintenir la cohérence des données, d'améliorer les performances lors de la manipulation des données (recherche, insertion, mise à jour et suppression).
- **Structurée** car elle définit la manière de stocker, manipuler et interconnecter les données.
- **Persistante** comme elle garantit que les données sont stockées de manière permanente et durable. De même, elle assure l'intégrité et la disponibilité des données.

## 2. Utilité

Une base de données assure le stockage, la gestion, l'organisation et l'accès efficace aux données de manière cohérente et structurée. Elle renforce, entre autres, les principes suivants :

- **La centralisation** : Les données sont regroupées dans une structure centrale pour éviter les redondances et les incohérences.
- **L'indépendance entre les applications et les données** : L'approche de séparation des préoccupations offre beaucoup d'avantages, notamment, en termes de gestion, de maintenance, de sécurité et d'évolutivité.
- **Les liaisons entre les données** : Les relations entre les données font partie intégrante de la base de données et non pas des applications qui manipulent les données.
- **L'intégrité des données** : C'est un ensemble de règles qui maintient la cohérence des données (unicité, valeur, etc...)
- **Le partage de données (ou concurrence d'accès)** : Plusieurs utilisateurs peuvent accéder simultanément à la base de données.

## 3. Système de Gestion de Base de Données

### a. Définition

Le logiciel qui permet de gérer une base de données est appelé **Système de Gestion de Bases de Données** (abréviation, **SGBD**). Il permet, notamment, de décrire, modifier, interroger et administrer les données d'une base de données.

Les **SGBD** diffèrent selon le modèle de base de données utilisé :

- Modèle relationnel
- Modèle orienté objet
- Modèle hiérarchique

Étant donné que le modèle relationnel largement utilisé dans la conception et dans le développement des applications, les **systèmes de gestion de bases de données relationnelles** (abrévés en **SGBDR**) sont très appréciés dans le domaine professionnel.

## b. Structure

Il est constitué de deux composants principaux, illustrés en figure 1 :

- **Le moteur** implémente l'ensemble de toutes les fonctionnalités relatives aux bases de données : parmi lesquelles on peut citer : la définition, le stockage, l'accès, la sécurité, la réplication des données.
- **L'interface** est située entre les applications utilisateur et le moteur, pour permettre l'accès facile et convivial aux données et aux services en relation.

L'internet regorge de **SGBDR**, la figure 2 présente quelques exemples des logiciels les plus connus.

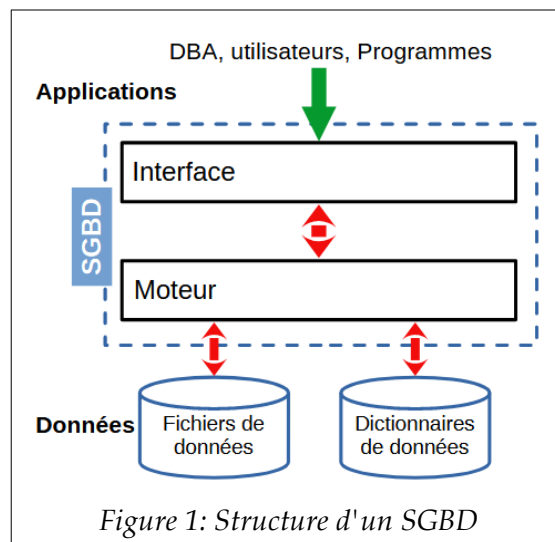


Figure 1: Structure d'un SGBD



Figure 2: Quelques exemples des SGBDR les plus connus

Les SGBD diffèrent par leurs :

- **usages** : embarqué, bureautique, cloud, etc.
- **performances** : volume de données traitées, optimisation des requêtes, consommation mémoire, etc.
- **licences** : open-source, gratuit, payant, etc.
- **évolutivité** : accès réseau, répartition des données sur plusieurs fichiers, partitions, clusters, etc.
- **fonctionnalités** : transactions, procédures stockées, déclencheurs, réplication, gestion des index, etc.
- **compatibilité avec les normes** : norme SQL92, norme SQL:1999, norme SQL:2003, etc.

## c. Fonctionnalités

Parmi les fonctionnalités d'un SGBD, on peut citer :

- **La définition des données** : Le Langage de Description de Données (LDD) permet de décrire les objets d'une base de données : les tables, les relations, les index, etc.
- **La manipulation des données** : Le Langage de Manipulation de Données (LMD) est utilisé pour la recherche, l'insertion, la mise à jour et la suppression des données.
- **La confidentialité et la sécurité des données** : Le Langage de Contrôle de Données (LCD) restreint l'accès aux données aux seules entités autorisées.
- **L'intégrité des données** : Le SGBD veille sur le respect des règles, des contraintes et des normes définies dans le but d'assurer l'exactitude et la cohérence des données.
- **L'accès concurrents** : Le SGBD supporte l'accès simultané aux données.

## d. Intervenants

Les utilisateurs potentiels d'un SGBD sont très variés. Ils peuvent inclure, selon leurs pouvoirs :

- L'**administrateur** (DBA) : configuration, sauvegarde, maintenance, gestion des autorisations d'accès, etc.
- Les **architectes de bases de données** : conception du schéma de la base de données : tables, relations, contraintes, etc.
- Les **développeurs d'applications** : programmation des applications exploitant des données.
- Les **analystes de données** : recherche, extraction, nettoyage et analyse des données pour en extraire des informations utiles.
- Les **utilisateurs finaux** : recherche d'informations, création de rapports d'aide à la décision depuis les données, etc.

## 4. Structure d'une base de données

Les données d'une base de données sont stockées dans les **tables**.

### a. Table

Une **table** est un ensemble de données relatives à un même sujet (ou **entité**). Une table 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**.

Colonnes				Lignes				
N°	Nom	Prénom	Age		Code article	Désignation	PU	Quantité
1	Tounsi	Salah	18	←	V10	Vis 50x3	40	2500
2	Benzarti	Leila	22		V20	Vis 20x2	20	1300
3	Gabsi	Ahmed	19		B100	Boulon 90x15	450	100
Liste de clients					C60	Clou 60x2	5	5000
					Liste d'articles			

Figure 3: Quelques exemples de tables

### b. 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**.

2	Benzarti	Leila	22
---	----------	-------	----

Un **client** est représenté par les valeurs respectives de ses champs  
(2, "Benzarti", "Leila", 22)

B100	Boulon 90x15	450	100
------	--------------	-----	-----

Un **article** est représenté par les valeurs respectives de ses champs  
("B100", "Boulon 90x15", 450.0, 100.0)

Figure 4: Quelques exemples d'enregistrements

### c. Champ

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

- son **nom** ;
- son **type** : numérique, chaîne/texte, date, booléen, etc. ;
- sa **taille** : elle dépend du type de champ : nombre maximal de caractères dans un texte, ou bien nombre maximal de chiffres pour un type numérique ;
- une **valeur par défaut** : c'est la valeur qui sera attribuée au champ si aucune valeur n'a été mentionnée lors d'une opération insertion d'un nouvel enregistrement ;
- la présence **obligatoire** ou optionnelle d'une valeur pour le champ ;
- les **valeurs autorisées** : indiquent les valeurs permises pour le champ ;

<b>Nom de la table</b> : Articles <b>Description</b> : Détails des articles vendus 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 5: Quelques exemples de descriptions des champs d'une table

### d. 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. 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.

La **clé primaire** peut être composée d'un seul champ ou de plusieurs champs.


NomMed	PrénomMed	 CodeMed	NomHopital	AdrHopital	TelHopital
Ben Salah	Amine	M10	Sahloul	Sousse	73425001
Amari	Lotfi	M15	Charles Nicolle	Tunis	71236147
Amari	Amine	M27	Fatouma Bourguiba	Monastir	73260871
Dhahbi	Lotfi	M31	Sahloul	Sousse	73424001
Kaïbi	Henda	M37	Charles Nicolle	Tunis	71236147

Figure 6: Exemple de clé primaire,  **CodeMed** est la clé primaire de la table

### e. Relations

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

Une **clé étrangère** est un attribut d'une table qui fait référence à la **clé primaire** d'une autre table, établissant ainsi une relation entre les deux tables.

L'exemple de la figure 7 illustre la relation entre les deux tables **Clients** et **Factures**. Cette relation indique qu'un client peut être facturé plusieurs fois. Dans cet exemple, la **table mère** est la table **Clients** car elle contient la **clé primaire** : **No client**. La table **Facture** est la **table fille** car elle contient la **clé étrangère** : **Code Client**.

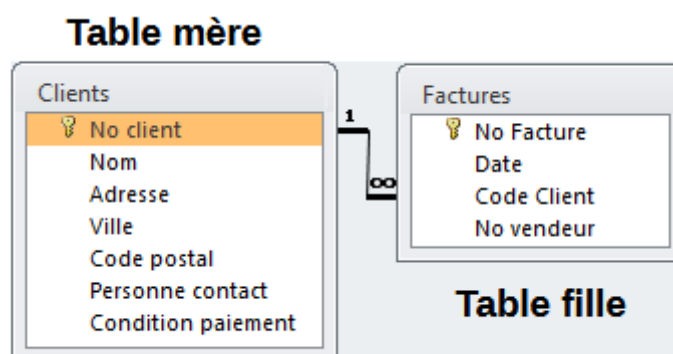


Figure 7: Exemples de relations entre les tables

L'exemple, illustré en figure 8, exprime le fait qu'un client peut commander plusieurs articles, et qu'un article peut faire partie de plusieurs commandes. Effectivement, dans un commerce un **client** fait une ou plusieurs **commande(s)** dans une date donnée **dateCmd**.

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** la colonne **idClient** qui correspond à l'identifiant du client (dans la table **Clients**) qui a fait la commande.

De même une commande est constituée par plusieurs articles et il faut, par conséquence, spécifier dans la table **Commandes** les numéros des articles (attribut **numArticle**).

Les deux attributs **idClient** et **numArticle** de la table **Commandes** dans sont deux **clés étrangères**.

Clients	<b>idClient</b>	<b>nom</b>	<b>prenom</b>	<b>adresse</b>	<b>téléphone</b>
	1	Hassayoune	Mariem	Khézama	98 745 365
	2	Msalle	Mohamed Amine	Hamam Sousse	53 875 395
	3	Jedidi	Ayhem	Hamam Sousse	51 525 354

Commandes	<b>numCmd</b>	<b>idClient</b>	<b>dateCmd</b>	<b>numArticle</b>	<b>quantité</b>
	1000	1	21/12/2022	1	5
	1001	1	21/12/2022	2	1
	1002	1	21/12/2022	3	4
	1003	2	21/12/2022	1	2
	1004	1	23/12/2022	1	10
	1005	2	25/12/2022	1	3

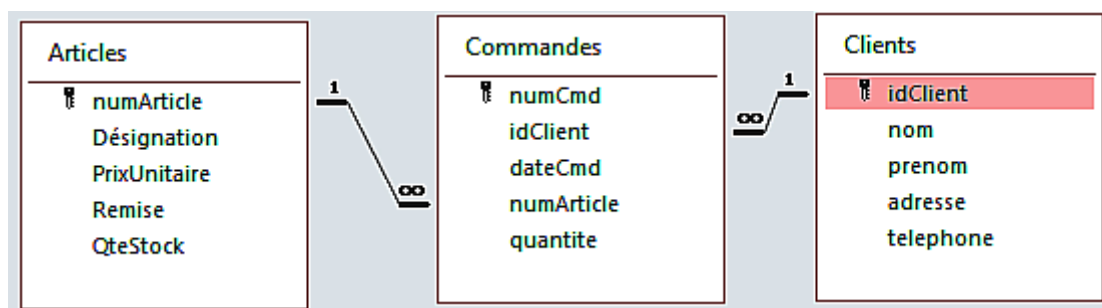
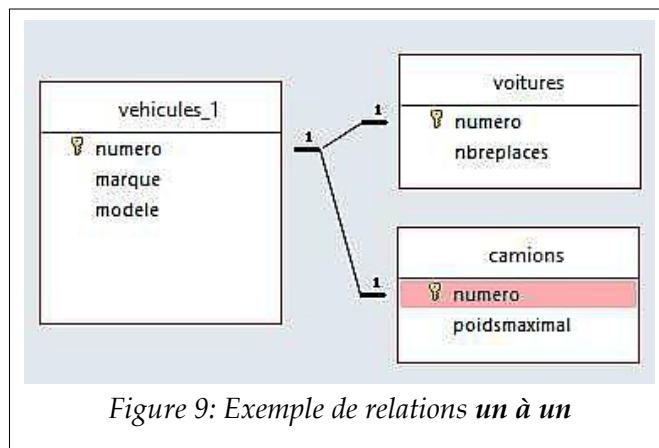
  

Articles	<b>numArticle</b>	<b>Désignation</b>	<b>PrixUnitaire</b>	<b>Remise</b>	<b>QteStock</b>
	1	Chocolat Mars	2.500	5	150
	2	Chocolat Snickers	2.500	0	75
	3	Chocolat Twix	2.800	0	100

Figure 8: Exemple de clés étrangères : **idClient** et **numArticle** sont les clés étrangères de la table **Commandes**

On distingue trois types de relations :

- **Relation un à un** : un véhicule, voir figure 9, 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 10, d'où la relation entre les tables Articles et Commandes est de type un à plusieurs. Il est aussi de même pour la relation entre les tables Clients et Commandes.
- **Relation plusieurs à plusieurs** : la relation entre les tables Articles et Clients, figure 10, est une relation plusieurs à plusieurs puisqu'on dit : Plusieurs clients peuvent commander plusieurs articles de notre boutique.



## 5. 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.

Dans l'exemple de la figure 10, **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**).



## 6. 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 #.

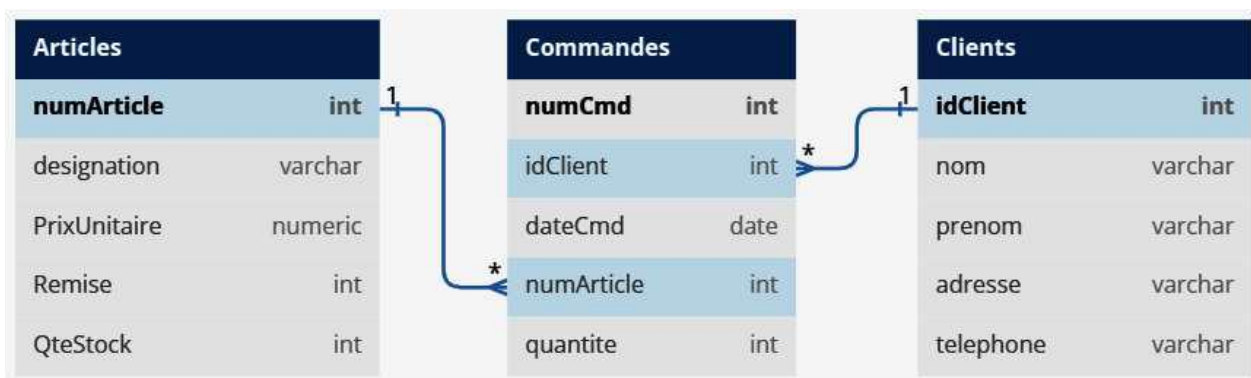


Figure 11: Exemple de représentation graphique du schéma d'une base de données

La représentation graphique dépend de l'outil utilisé. Les figures 9 et 10, montrent des exemples de représentations graphiques dans MS Access.