

Bases de données



Une base de données (BD) est un ensemble structuré d'informations. Dans le langage courant, le terme peut désigner toute source importante de données telle qu'une encyclopédie. Dans le domaine de l'informatique, c'est un ensemble d'informations regroupées sous la forme d'enregistrements stockés sur un système de fichiers structurés et organisées de manière à pouvoir être facilement manipulées.

L'organisation logique des données se fait selon un modèle de données et la structure physique des fichiers comporte des index destinés à accélérer les opérations de recherche et de tri. Le **modèle de données relationnel** est aujourd'hui le plus utilisé parce qu'il permet l'indépendance entre la structure physique et l'organisation logique des données (interface de manipulation).

Le logiciel qui manipule les bases de données est appelé **système de gestion de base de données (SGBD)**. Il permet d'organiser, de contrôler, de consulter et de modifier la base de données.

1. Structuration d'une base de données

Les bases de données sont basées sur une approche de structuration du monde réel qui donne une représentation abstraite (le **schéma**) résultant de l'application d'un modèle de données (data model).

Représentation naïve dans un tableau

Une première idée pour implémenter une base de données consiste à utiliser un tableur.

Prenons pour exemple une base de données stockant les appels téléphoniques passés par les clients d'un opérateur. Les clients peuvent obtenir plusieurs numéros de téléphone auprès de cet opérateur. Pour chacun de ces numéros, l'opérateur enregistrera les informations nécessaires pour établir une facture mensuelle détaillée, tels que les nom, prénom et adresse du client, mais aussi les tarifs de communication et les renseignements sur l'appel.

Une telle base de données représentée dans un simple tableau. Une telle base de données ressemble alors à l'extrait suivant :

Nom	Prénom	Adresse	Numéro de tel.	N° appelé	Date/heure	Durée (s)	Tarif (HT/min)
Cramé	Jacques	2, rue Un Chastrés	0305022545	0033324663 4217	10.2.2019 14:52	455	0,70
Cramé	Jacques	2, rue Un Chastrés	0305022545	0325658954	17.2.2019 14:52	62	0,10
Cramé	Jacques	2, rue Un Chastrés	0305022545	0325658954	18.2.2019 14:52	145	0,10
Cramé	Jacques	2, rue Un Chastrés	0305022545	0652365874	20.2.2019 14:52	302	0,40
****	****	****	****	****	****	****	****

L'avantage de cette méthode est sa simplicité, tout ce qui concerne un appel tient sur une ligne et tous les appels sont répertoriés dans une table. Cette approche a aussi des inconvénients :

- ➔ Des informations sont redondantes, encombrant inutilement la mémoire
- ➔ Pour identifier un client, on a besoin de trois champs.
- ➔ Si un utilisateur change d'adresse, il faut reporter ce changement sur chaque ligne où il apparaît

Représentation dans une base de données relationnelle

Une autre solution est d'utiliser plusieurs tables « reliées » entre elles. On parle alors de base de données relationnelle.

Reprenons l'exemple de stockage d'appels téléphoniques, une solution plus efficace consiste à répartir les données dans trois tables :

- ➔ Une table **Client** contenant les coordonnées des clients
- ➔ Une deuxième table **Numéros** servant à faire le lien entre l'utilisateur et le numéro
- ➔ Une troisième table **Appels** contenant le listing des appels effectués

Cette base de donnée relationnelle est la suivante :

Table CLIENTS				Table NUMEROS	
Id_Client	Nom	Prénom	Adresse	Numéro	Id_Client
156	Cramé	Jacques	48, rue Lin - Travol	0652368412	156
10234	Cramé	Jacques	2, rue Un - Chastrés	0305022545	10234
1457	Crannelle	Line	3, rue As - Freche	06235014111	10234
.....

Table APPELS					
Id_appel	N° appelant	N° appelé	Date et heure d'appel	Durée (s)	Tarif
125478	0305022545	00333246634217	10.2.2019 14:52	455	0,70
125479	0654215896	0214520236	10.2.2019 14:55	30	0,20
125480	0214685232	0214568522	10.2.2019 14:56	1023	0,10
.....

De cette manière les informations semblent moins lisibles car écartées mais cette méthode offrent de nombreux avantages :

- ➔ Les informations ne sont plus redondantes
- ➔ Les informations sont modifiables
- ➔ Les doublons ne portent pas à confusion

2. Le modèle relationnel

Schéma de relation, attribut, domaine

Le modèle relationnel représente sous forme graphique ou textuel les tables qui constituent une base de données.

Cette représentation s'appelle un schéma et une table est aussi dénommée une relation. Le schéma d'une relation est repérée par un nom (noté en majuscule) et est caractérisée par ses attributs (notés en minuscules), comme indiqué sur la figure 1.



A chaque attribut, on associe un ensemble de valeurs appelé domaine que celui-ci est autorisé à parcourir. Un domaine peut par exemple être :

- Un intervalle de valeurs possibles (ex : matricule d'étudiant)
- un ensemble de couleur permises (ex : couleur d'un modèle de voiture)
- les entiers naturels, les nombres réels, les chaînes de caractères ...

Par analogie avec les langages de programmation, le domaine correspond au typage des données.

Le schéma de relation a pour but de stocker les enregistrements qui constituent la base de données.

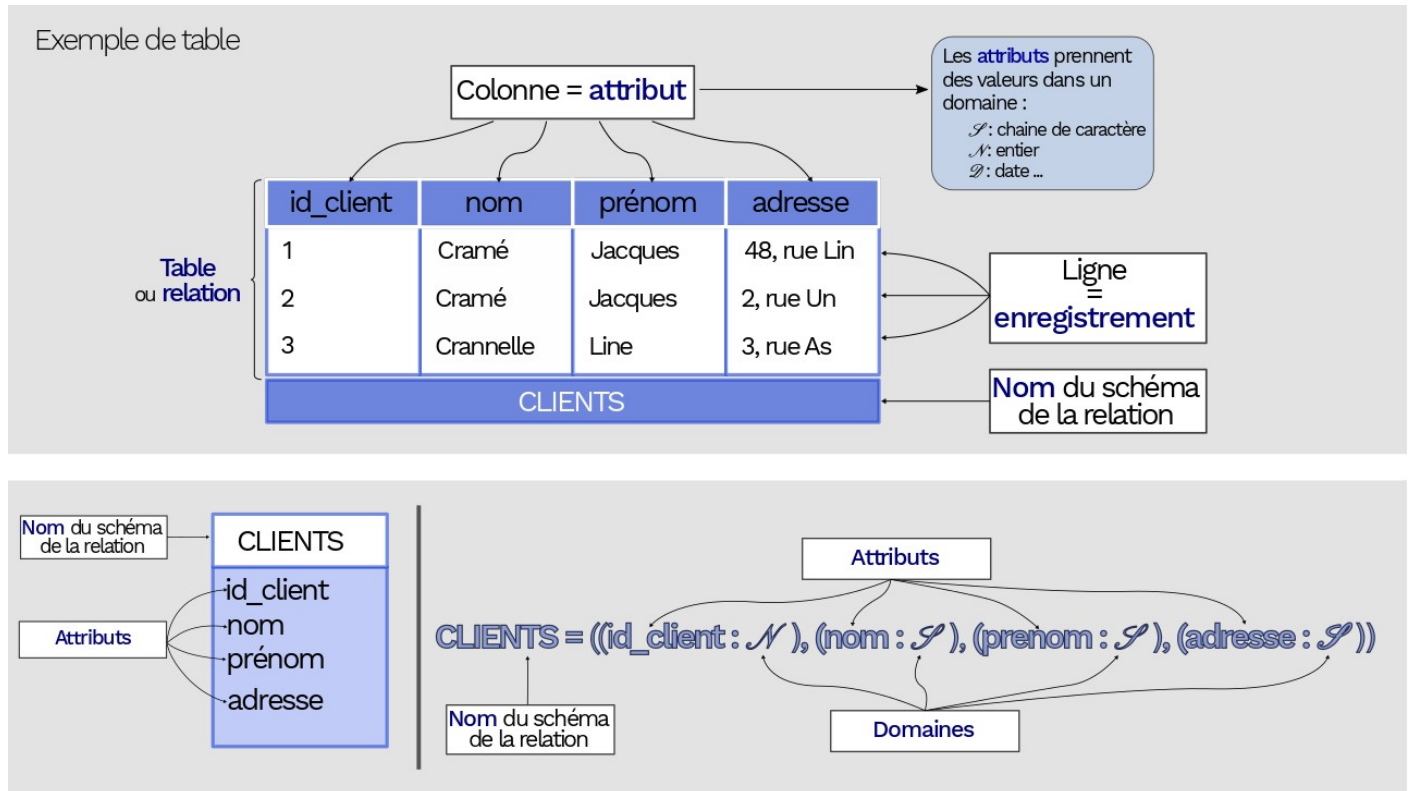


Figure 1: Vocabulaire d'une relation (haut) et représentations graphique (bas gauche) ou textuelle (bas droite) d'un schéma de relation

Remarque : La spécification du domaine dans la représentation textuelle n'est pas obligatoire et est souvent omise.

Normalisation d'une base de données

Le nombre de tables mis en œuvre ainsi que leur définition doit répondre aux deux objectifs suivants :

- **Non redondance des données :** Afin d'éviter les problèmes lors de la mise à jour, chaque donnée ne doit être présente qu'une seule fois dans la base.
- **Cohérence des données :** Les données sont soumises à un certain nombre de contraintes d'intégrité qui définissent un état cohérent de la base.

Contrainte d'intégrité

Une contrainte d'intégrité est une priorité logique, préservée à tout instant par la base de données et qui garantit la cohérence des données. Cette contrainte est donc un invariant de la base de donnée.

On distingue trois grandes catégories de contraintes d'intégrité jouant des rôles complémentaires :



- ➔ **Les contraintes de domaine** : ces propriétés sont toutes assurées par le domaine des attributs. Elles visent à limiter autant que possible la saisie de valeurs illégales ou mal formées.
- ➔ **Les contraintes d'entités** : ces propriétés s'assure que chaque enregistrement est unique afin de l'identifier de manière non ambiguë. Ces contraintes sont remplies par la définition de clé(s) primaire(s) dans chaque table.
- ➔ **Les contraintes référentielles** : Bien construite, une base de données fait appel à des données situées dans différentes relations. Pour que les données restent utilisables et cohérentes, il ne faut pas que l'on puisse détruire les données qui dépendent les unes des autres. C'est le rôle de l'intégrité référentielle de protéger automatiquement ces relation. Cette contrainte est réalisée en définissant des clés étrangères

Les clés primaires et étrangères

Une **clé primaire** a pour but d'assurer l'unicité de chaque enregistrement dans une table. Cette clé est associée à un attribut et s'assure que chaque donnée de l'attribut est unique.

Dans l'exemple précédent, la clé primaire doit être associée à l'attribut `id_client` puisque tous les enregistrements de cet attribut ont des valeurs différentes. Cette clé permet de faire la différence entre les deux homonymes Jacques Cramé.

Une clé primaire est indiquée dans un schéma de relation en repérant :

- ➔ en gras ou par une clé l'attribut dans une représentation graphique
- ➔ par un trait l'attribut dans une représentation textuelle (voir figure 2)

Une **clé étrangère** est un ensemble d'attribut d'une table qui sont une clé primaire dans une autre table. Elle servent de référence à une autre table afin d'établir des liens entre les relations et de contraindre ces relations (contrainte de relation). Ces clés étrangères ont les caractéristiques suivantes :

- x Une clé étrangère ne peut être un attribut qui n'est pas une clé primaire de la table à laquelle elle se réfère
- x Un enregistrement de la table primaire ne peut être effacé s'il possède des enregistrements liés (intégrité référentielle)
- x La clé primaire ne peut être changée dans la table primaire si cet enregistrement possède des enregistrements liés.

Les clés étrangères sont repérées :

- x dans la représentation graphique ➔ en reliant l'attribut avec une clé secondaire à l'attribut avec la clé primaire dans la table primaire.
- x Dans la représentation textuelle ➔ en soulignant en pointillé l'attribut ou en plaçant un hashtag avant l'attribut (voir figure 3)

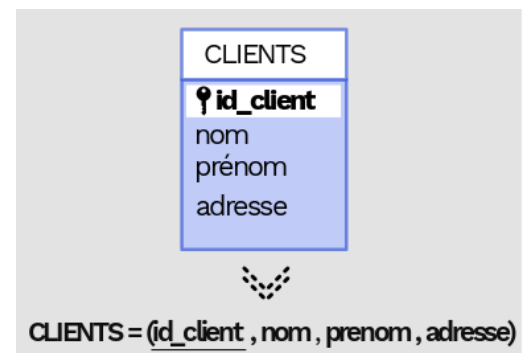


Figure 2: Repérage d'une clé primaire

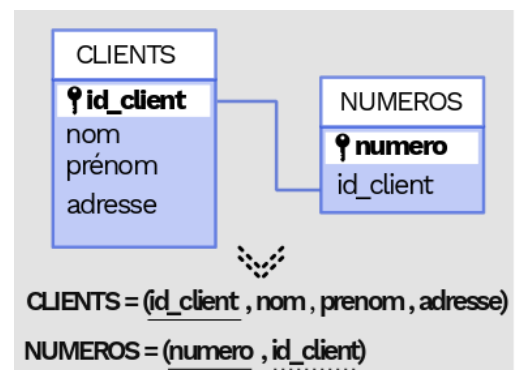


Figure 3: Repérage d'une clé secondaire