

# Base de Données Relationnelles - Définitions

## Base de Données

Une *base de données* est un ensemble de données mémorisé par un ordinateur, organisé selon un modèle et accessible à de nombreuses personnes ou programmes.

Une *base de données* (en anglais *database*), permet de stocker et de retrouver l'intégralité de données brutes ou d'informations en rapport avec un thème ou une activité ; celles-ci peuvent être de natures différentes et plus ou moins reliées entre elles. Ces informations sont généralement très structurées, et la base est localisée dans un même lieu et sur un même support informatisé.

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

Un *SGBD* est un ensemble coordonné de logiciels permettant de décrire, mémoriser, manipuler, traiter, interroger les ensembles de données constituant la base.

## Base de Données Relationnelle

Une *base de données relationnelle* est une base de données où l'information est organisée dans des tableaux à deux dimensions appelés des *relations* ou *tables*, selon le modèle introduit par Edgar F. Codd en 1970.

Selon ce modèle *relationnel*, une base de données consiste en une ou plusieurs relations. Les lignes de ces relations sont appelées des *n-uplets* ou *enregistrements*. Les colonnes sont appelées des *champs* ou *attributs*.

Exemples de SGBD relationnelles: MS Access, SQLite, MySQL, PostgreSQL, Oracle, ...

Une composante *N* d'un *n-uplet* est une *clé primaire* lorsque par nature la valeur de cette composante est différente pour chaque *n-uplet* d'une relation, et que, pour une valeur donnée, il n'existe qu'un seul *n-uplet* dans toute la relation dont la composante a cette valeur.

Une composante *N* d'un *n-uplet* est une *clé étrangère* lorsque les valeurs de cette composante sont des références à une clé primaire.

jointures

## Algèbre Relationnelle et Language SQL

L'*algèbre relationnelle* consiste en un ensemble d'opérations qui prennent en entrée une ou plusieurs relations et retourne une nouvelle relation. L'ensemble de ces opérations forme une structure algébrique au sens mathématique du terme.

Il existe six opérateurs d'algèbre relationnelle (Sélection, Projection, Renommage, Union, Différence et Produit cartésien). À partir de ces opérateurs, des opérateurs dérivés peuvent être conçus comme la jointure, l'intersection ou la division.

*SQL* (sigle de *Structured Query Language*) est le langage informatique le plus répandu utilisé pour formuler des requêtes d'algèbre relationnelle.

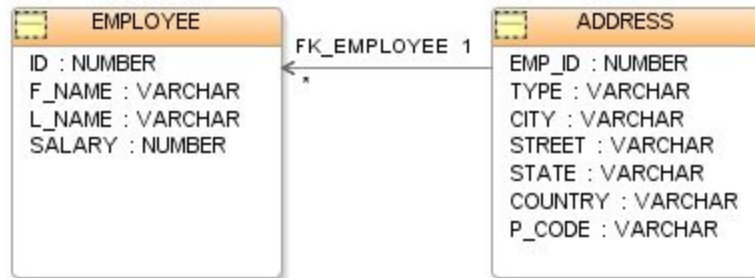
En plus de l'algèbre relationnelle, SQL permet:

- d'utiliser des fonctions d'agrégation (e.g., minimum, maximum, comptage...).
- de trier les n-uplets (le support théorique est la notion d'ensemble et donc sans ordre).
- d'utiliser la récursion (non disponible dans l'algèbre relationnelle car le nombre de jointure est *a priori* non connu à l'avance — comme pour gérer un arbre généalogique et rechercher les ancêtres).
- d'ajouter ou de supprimer du contenu dans la base de données.
- de définir des contraintes d'intégrité définissant des règles à suivre pour garantir la cohérence de la base lors de manipulation autres que la consultation (e.g., vérifier que le nouveau contenu ne casse pas l'intégrité référentielle entre les clés, ou crée de la redondance).

Les *jointures* en SQL permettent d'associer plusieurs tables dans une même requête. Cela permet d'exploiter la puissance des bases de données relationnelles pour obtenir des résultats qui combinent les données de plusieurs tables de manière efficace.

En général, les jointures consistent à associer des lignes de deux tables en associant l'égalité des valeurs d'une colonne d'une première table par rapport à la valeur d'une colonne d'une seconde table.

## Exemples de Requetes SQL Simples



## Modélisation d'une Base de Données Relationnelle

### Le Modèle Entité - Relation

C'est un modèle de représentation de l'information, permettant de comprendre et de visualiser l'organisation des données, mais qui n'est pas destiné directement à l'implémentation de ces données.

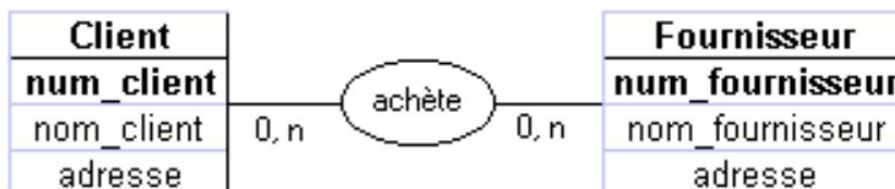
Lors de la conception d'une base de données, on commence par réaliser un modèle entité-association que l'on transforme ensuite en modèle relationnel normalisé, implémentable.

### Terminologie

Une *entité* est un objet ayant une existence propre présentant un intérêt pour l'entreprise. Par exemple, le client Dupuis, le fournisseur Durand, ...

Un *type d'entité* est une classe d'entités ayant en commun un ensemble de propriétés. Par exemple Client, Fournisseur, ...

NB: Une entité est donc une instance d'un type d'entité.



Une *association* est un lien entre entités. Il représente un verbe matérialisant une relation entre entités. Par exemple: le client Dupuis achète au fournisseur Durand.

Un *type d'association* est un lien-type entre types d'entités. Par exemple, “achète” est un type d'association entre les types d'entités Client et Fournisseur.

Une *propriété* est une caractéristique d'un type d'entité ou d'un type d'association. Par exemple, “adresse” est une propriété des types d'entités de type Client et de type Fournisseur.

L'*identifiant* d'un type d'association est obtenu en concaténant les identifiants des types d'entité concernés par l'association. Par exemple, “achète” a pour identifiant (num\_client, num\_fournisseur).

La *cardinalité* d'une entité dans une association est le nombre de fois minimum et nombre de fois maximum qu'une même occurrence de l'entité peut intervenir dans les occurrences de l'association:

- 0: l'occurrence peut ne pas participer
- 1: l'occurrence participe obligatoirement (minimum) ou au plus une fois (maximum)
- n: l'occurrence peut participer plusieurs fois

## **Du Modèle Entité-Association au Modèle Relationnel**

### **Règle 1**

Chaque type d'entité donne naissance à une relation du même nom.

- chaque propriété du type d'entité devient un attribut de la relation.
- l'identifiant du type d'entité devient la clé de la relation.

### **Règle 2**

Si un type d'association n'a aucune patte de cardinalité maximale égale à 1, alors:

- ce type d'association devient une relation
- chaque propriété du type d'association devient un attribut de la relation
- L'identifiant du type d'association devient la clé de la relation.

### **Règle 3**

Si un type d'association a une patte dont la cardinalité maximale est égale à 1, alors:

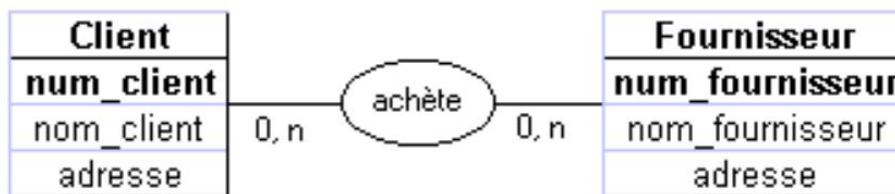
- le type d'association n'est pas transformé en relation, mais est matérialisé par l'ajout d'un attribut dans la relation source de la patte concernée
- cet attribut est la clé de la relation but de la patte concernée

Un *schéma de base de données* représente la configuration logique de tout ou partie d'une base de données relationnelle. Il peut se présenter à la fois sous la forme d'une représentation visuelle et d'un ensemble de formules, appelées « contraintes d'intégrité », qui régissent une base de données.

Le processus de création d'un schéma de base de données s'appelle la *modélisation des données*.

## Exemples

### Achats Fournisseurs



### Enseignements

