LES BASES DE DONNEES

PRESENTATION

La différence entre un fichier (***txt*** ou ***csv*** par exemple) et une base de données est que les informations sont en vrac dans un fichier alors qu'elles sont structurées dans une base de données. Les bases de données sont gérées par des logiciels appelés **Systèmes de Gestion de Bases de Données (SGBD)** qui permettent un accès direct à des données précises.

Les données peuvent être en quantité importante : gigaoctets (10^9 octets), téraoctets (10^{12} octets), voire pétaoctets (10^{15} octets). Les données peuvent être de différents formats : nombres, textes, images, vidéos, etc. Gérer ces données implique de pouvoir les stocker, les retrouver et les mettre à jour de manière efficace.

Une **base de données (BD)** est donc une collection de données structurées reliées entre elles par des relations. Une base de données permet l'interrogation et la manipulation de ces données. Les données présentes sur le **Web** proviennent la plupart du temps de bases de données relationnelles.

LE MODELE RELATIONNEL

Dans le **modèle relationnel**, les données sont stockées dans des tableaux à deux dimensions appelés **tables**. L'ordre entre les colonnes n'a pas d'importance. Par exemple, on ne peut pas parler de la première ou de la deuxième colonne d'une table : chaque colonne est désignée par son nom. On appelle **attribut** le nom donné à une colonne de la table. Une ligne de la table contient des **valeurs** pour chacun des attributs. Chacune de ces lignes est appelée **n-uplet**, ou **tuple**. L'ensemble des n-uplets d'une table s'appelle une **relation**. Une **base de données relationnelle (BDR)** est formée d'un **ensemble de relations**.

Équipes

nom	nomCourt	ville
Paris Saint-Germain	PSG	Paris
Olympique de Marseille	OM	Marseille
Olympique Lyonnais	OL	Lyon

Joueurs

équipe	no	nom	prénom	age
OM	4	KAMARA	Boubacar	20
OM	10	PAYET	Dimitri	33
PSG	1	NAVAS	Keylor	33
PSG	7	MBAPPE	Kylian	21
OL	1	LOPES	Anthony	30
OL	9	DEMBELE	Moussa	24

Rencontres

date	local	visiteur	butl	butv
08/02/21	OM	OL	0	2
21/03/21	OL	PSG	5	1
13/09/20	PSG	OM	0	1

Exemple de base de données relationnelle

À chaque attribut est associé un **domaine**, qui représente l'ensemble de valeurs possibles de l'attribut.

L'ensemble des nombres entiers, l'intervalle de nombres réels [0,0 ; 10000,0] et l'ensemble {rouge, vert, bleu} sont des exemples de domaines.

CLES PRIMAIRES ET SECONDAIRES

Il existe une contrainte fondamentale dans les bases de données relationnelles, une relation ne peut pas contenir 2 t-uplets identiques. Par exemple, la situation ci-dessous n'est pas autorisée :

Joueurs

équipe	no	nom	prénom	age
OM	4	KAMARA	Boubacar	20
OM	4	KAMARA	Boubacar	20
PSG	1	NAVAS	Keylor	33

Pour respecter cette contrainte, on définit la notion de **clef primaire**.

Une clef primaire est un attribut dont la valeur permet d'identifier de manière unique un t-uplet de la relation. Autrement dit, si un attribut est considéré comme clef primaire, on ne doit pas trouver deux fois la même valeur pour cet attribut.

Si aucun attribut ne peut jouer le rôle de clé primaire on peut ajouter un attribut **id** (numéro d'identification unique qui jouera ce rôle).

Un même enregistrement peut très bien avoir plusieurs clés. Dans ce cas, on choisit le plus souvent une clé parmi toutes les autres, qui sera considérée comme une meilleure clé que les autres : c'est la **clé primaire**. Les autres clés seront alors appelées des **clés secondaires** ou clés alternatives.

Équipes

id	nom	nomCourt	ville
1	Paris Saint-Germain	PSG	Paris
2	Olympique de Marseille	OM	Marseille
3	Olympique Lyonnais	OL	Lyon

Joueurs

id	id_equipe	no	nom	prénom	age
1	2	4	KAMARA	Boubacar	20
2	2	10	PAYET	Dimitri	33
3	1	1	NAVAS	Keylor	33
4	1	7	MBAPPE	Kylian	21
5	3	1	LOPES	Anthony	30
6	3	9	DEMBELE	Moussa	24

CLES ETRANGERES

On peut remarquer que l'attribut id_equipe de la relation Joueurs permet de créer un lien avec la relation Equipes. La mise en place de liens entre relations permet d'éviter la redondance d'informations.

Pour établir un lien entre 2 relations RA et RB, on ajoute à RA un attribut x qui prendra les valeurs de la clé primaire de RB. Cet attribut x est appelé clef étrangère (l'attribut correspond à la clé primaire d'une autre table, d'où le nom). Dans l'exemple ci-dessus, id_equipe est une clé étrangère de la relation joueurs. Pour préserver l'intégrité d'une base de données, il est important de bien vérifier que toutes les valeurs de la clef étrangère correspondent bien à des valeurs présentes dans la clef primaire. Ce concept est appelé l'**intégrité référentielle**.

MODELISATION D'UNE BASE DE DONNEES

PREMIERE ETAPE : SCHEMA CONCEPTUEL D'UNE BASE DE DONNEES.

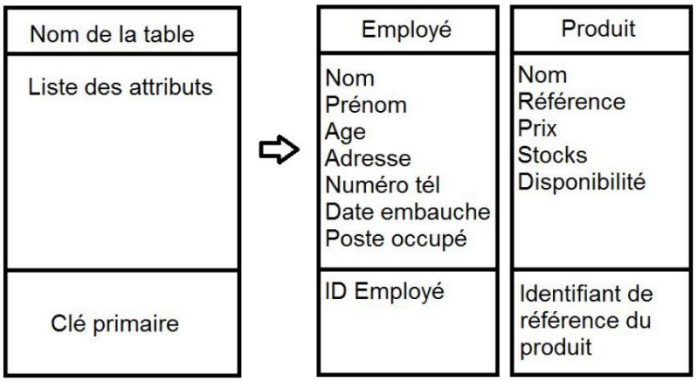
Le schéma conceptuel décrit les tables de la BDD ainsi que leurs liens. La description de la table est relativement simple : elle se borne à leur donner un nom, à décrire chaque colonne (nom et type) et à dire quelles sont les clés primaires. Les liens ne sont pas traduits en clés étrangères, qui ne sont d'ailleurs pas présentes dans ce schéma. Ce schéma conceptuel n'est qu'une première étape dans la modélisation d'une base de données.

DEUXIEME : MODELE LOGIQUE

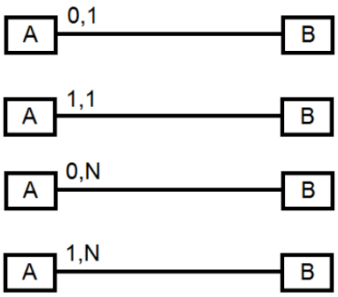
La seconde étape consiste à prendre le schéma et traduire les liens entre tables sous la forme de clés étrangères : on obtient alors le modèle logique de la base de données.

TROISIEME ETAPE : PROGRAMMATION

Une fois cela fait, on peut traduire le tout dans un langage de programmation qui permet de gérer des bases de données (le SQL, par exemple).



Les tables sont représentées par des rectangles, décomposées en trois cadres : un pour le nom de la table, un autre pour les attributs, et un autre pour les clés primaires. Les attributs sont notés à l'intérieur du rectangle, dans le cadre en-dessous du nom de la table : on précise non seulement leur nom, mais aussi leur type.



Les relations entre tables sont indiquées par une flèche ou un trait entre deux tables. La cardinalité d'une association précise le nombre de fois où une instance d'entité peut participer à l'association. Il existe une cardinalité maximale et une minimale.

Le premier cas est celui de la correspondance un vers un : chaque ligne de A pointe vers une ligne de B, et seulement une seule. Dans ce cas, la cardinalité est de 1, que ce soit pour la cardinalité minimale ou maximale.

Le second cas est une petite variante du premier cas : il se peut que des lignes de A ne pointent pas vers une ligne de B. Dans ce cas, la cardinalité maximale ne change pas : elle est de 1. Mais la cardinalité minimale est de 0.

Le troisième cas est celui de la correspondance un vers plusieurs : chaque ligne de A pointe vers au moins une ligne de B, potentiellement plusieurs. Dans ce cas, la cardinalité maximale n'est pas connue, mais on sait qu'elle est supérieure à 1. La cardinalité minimale est de 1.

Le quatrième cas est une variante du cas précédent, sauf qu'il est possible qu'une ligne de A ne pointe pas vers une ligne de B. La cardinalité minimale est 0.

Pour résumer, la cardinalité minimale est soit de 0, soit de 1, tandis que la maximale est soit de 1 soit de plusieurs. Ces cardinalités doivent être indiquées sur le trait de la relation, séparées par une virgule : la cardinalité minimale doit être indiquée avant la maximale. Attention : si on doit indiquer les cardinalités pour la liaison de A vers B, il faut aussi indiquer celle de B vers A sur le trait.