

BASES DE DONNÉES : INTRODUCTION

Sommaire

I	Introduction	1
	I.1 Définition	1
	I.2 Première approche	1
II	Base de données relationnelle	2
	II.1 Exemple	2
	II.2 Vocabulaire	3

I Introduction

I.1 Définition

Une base de données est un ensemble structuré de données. Il peut être vu comme une énorme "bibliothèque" dont chacun des livres stockerait des données telles que des nombres, des noms, des booléens, des dates des mots, etc...

La base est enregistrée sur un support accessible par ordinateur, capable de satisfaire simultanément plusieurs utilisateurs. Une base de données doit pouvoir être lue rapidement et on doit pouvoir en extraire rapidement les données.

I.2 Première approche

Une première idée qu'on pourrait se faire d'une base de données consiste à enregistrer toutes les données dans un même tableau. Par exemple, dans le cas suivant, où on s'intéresse aux notes de Mathématiques d'un groupe d'élèves (le devoir comporte deux parties notées sur 10) :

Nom	Prenom	Devoir	type	coefficient	part_1	part_2
ALBIN	Felix	1	DS	2	1	4
CHERRAJ	Assia	1	DS	2	10	9
GOMES	Quentin	1	DS	2	3	8
LEGER	Maia	1	DS	2	10	5
DECOUPY	Victor	1	DS	2	4	6
GOMIS	Jean-Marie	1	DS	2	1	10
POUBANNE	Lisa	1	DS	2	9	3
HAMELIN	Dimitri	1	DS	2	8	4
HAMELIN	Axel	1	DS	2	5	7
LE GUILLOUS	Cyrielle	1	DS	2	1	4
ALBIN	Felix	2	DM	1	7	3
CHERRAJ	Assia	2	DM	1	9	3
GOMES	Quentin	2	DM	1	9	2
LEGER	Maia	2	DM	1	2	4
DECOUPY	Victor	2	DM	1	4	4
GOMIS	Jean-Marie	2	DM	1	8	1
POUBANNE	Lisa	2	DM	1	4	2
HAMELIN	Dimitri	2	DM	1	6	7
HAMELIN	Axel	2	DM	1	7	5
LE GUILLOUS	Cyrielle	2	DM	1	8	7
ALBIN	Felix	3	Interro	1	5	4
CHERRAJ	Assia	3	Interro	1	9	6
GOMES	Quentin	3	Interro	1	2	3
LEGER	Maia	3	Interro	1	2	8
DECOUPY	Victor	3	Interro	1	2	8

Tableau de données

Ce tableau n'est pas simple à exploiter :

- certains noms peuvent être mal orthographiés à certaines lignes car ils sont souvent répétés ;
- on peut mettre ou non l'accent, à "Félix" par exemple ;
- on peut avoir écrit "DM ou "Devoir maison", "interro" ou "interrogation" et pourtant le devoir est du même type.

S'il y a que très peu de données, il est possible de traiter les données à la main. Par exemple, retrouver toutes les notes de "Félix". Mais si le jeu de données devient trop important ?

Imaginez-vous la situation suivante : un braquage a eu lieu à Saint-Pourçain-sur-Sioule. Les voleurs sont partis dans une voiture. Un témoin est formel : c'est une Renault rouge coupé dont la plaque commençait par un "A" et avec le sigle de la concession automobile "Bony Automobiles". La police appelle Renault pour que la firme lui donne toutes les voitures correspondant à ce signalement. Pensez-vous qu'une personne va regarder dans la base de données les véhicules vendus par Renault un par un pour trouver tous les véhicules correspondants ? Pour rappel, Renault vend plus de 2 millions de véhicules par an...

Une base de données doit avoir plusieurs bonnes propriétés :

- **Intégration** : les données ne doivent pas être redondantes et rester cohérentes (par exemple, le changement du nom d'une personne doit être rapide et se faire partout où cela est nécessaire).
- **Indépendance** : plusieurs propriétés sont regroupées par ce terme : l'indépendance physique : un changement de disque dur pour stocker les données ne doit pas être ressenti ; l'indépendance logique : si on change la logique de l'organisation des données, on n'a pas besoin de changer les programmes d'application ; indépendance du moyen d'accès aux données : l'utilisateur demande simplement d'accéder aux données et on lui renvoie les données demandées.
- **Disponibilité** : les données doivent pouvoir être traitées rapidement et par plusieurs utilisateurs.
- **Sécurité** : protection contre les pannes mais aussi l'incohérence des données (si une lettre est rentrée dans la colonne des dates par exemple et que le format est du type 03/04/12).

Les bases de données sont stockées au sein d'une SGBD (Système de Gestion de Bases de Données). C'est un logiciel visant à simplifier la tâche des usagers en proposant un niveau d'abstraction (c'est-à-dire que le fonctionnement peut s'adapter à différents types de données). Il doit gérer les représentations physique (les "tables" de données) et logique (les liens entre les tables) des données. Pour cela il utilisera un langage dédié à la manipulation des données.

Plusieurs SGBD existent : ORACLE, ACCESS, OPEN OFFICE etc..., mais ils utilisent tous le langage normalisé SQL (Structured Query Language) créé en 1974. Plusieurs SGBD (libres et gratuites...) ont d'ailleurs un nom construit sur cet acronyme : SQLite, MySQL, sqliteman..

II Base de données relationnelle

II.1 Exemple

Dans une base de données relationnelle, les informations seront contenues dans plusieurs tableaux comme suit :

rowid	élèves	Devoir	part_1	part_2
1	1	1	1	4
2	1	2	7	3
3	1	3	5	4
4	1	4	4	9
5	1	5	9	3
6	1	6	3	8
7	1	7	6	6
8	1	8	8	5
9	1	9	9	4
10	1	10	9	7
11	1	11	3	3
12	1	12	4	9
13	1	13	7	9
14	1	14	6	8
15	2	1	10	9
16	2	2	9	3
17	2	3	9	6
18	2	4	3	9
19	2	5	8	1
20	2	6	6	1
21	2	7	7	5
22	2	8	6	10
23	2	9	2	7
24	2	10	1	8
25	2	11	1	9
26	2	12	9	9
27	2	13	5	7
28	2	14	5	8
29	3	1	3	8
30	3	2	9	2
31	3	3	2	3
32	3	4	4	2
33	3	5	6	2
34	3	6	4	5

notes

rowid	Nom	Prenom	année	homme
1	ALBIN	Felix	1998	1
2	CHERRAJ	Assia	1998	0
3	GOMES	Quentin	1997	1
4	LEGER	Maia	1998	0
5	DECOUPY	Victor	1997	1
6	GOMIS	Jean-Marie	1997	1
7	POUBANNE	Lisa	1998	0
8	HAMELIN	Dimitri	1998	1
9	HAMELIN	Axel	1998	1
10	LE GUILLOUS	Cyrielle	1998	0

identité

rowid	type_num	coefficient
1	1	2
2	2	1
3	3	1
4	1	1
5	2	2
6	1	3
7	2	1
8	3	2
9	1	1
10	2	2
11	1	3
12	2	1
13	1	1
14	2	1
15	1	2
16	1	3

devoir_config

rowid	type
1	DS
2	DM
3	Interro

devoir_type

Au lieu d'un seul tableau on en a 4. Ils contiennent des données sur des sujets différents. Un tableau contient les notes des élèves sur chaque partie du devoir. Un autre contient l'identité des élèves et quelques informations sur eux (année de naissance, genre...). Un troisième tableau contient les informations relatives aux devoirs et un dernier les types de devoirs qu'on peut trouver. On remarque tout de suite qu'il n'y a pas plus de redondance : les mots ne sont plus répétés, il n'y a plus de risque de mal les orthographier. Les données prennent moins de place qu'avec la méthode précédente : stocker un entier est moins coûteux que le nom et prénom d'un élève. On appelle ces tableaux **des tables ou des relations**.

Exercice 1.1

Que représente la ligne 22 du tableau "notes"?

E

II.2 Vocabulaire

- **tuple** : les lignes sont les **entrées** ou **tuples** ou encore **entités**. Ces tuples ne sont pas nommés. Exemple : (1,ALBIN,Felix,1998,1).
- **attribut** : les colonnes sont appelées les **champs**, les **attributs** ou encore les **propriétés** : dans

l'exemple précédent, les attributs de la table *identité* sont *rowid* (numéro de la ligne), *Nom*, *Prenom*, *année*, *homme*.

- **table** : tableau contenant les données. C'est un sous-ensemble du produit cartésien de plusieurs attributs. Par exemple, la table *identité* contient les éléments (1,ALBIN,Felix,1998,1), (2,CHERRAJ,Assia,1998,0), etc... On l'appelle également relation (langage algébrique) ou type entité.
- **domaine** : sous SQL, un domaine est décrit par un type (INT,FLOAT,etc...), auquel on peut ajouter des contraintes (tous différents, positif, etc...). Les types les plus usuels sont :
 - Le type VARCHAR (ou CHAR VARYING) permet de coder des chaînes de longueur variables, mais la longueur maximale est fixée. Par exemple VARCHAR(20) désigne le type des chaînes alphanumériques formées d'au plus 20 caractères.
 - CHAR désigne le type des chaînes alphanumériques de longueur fixe, par exemple CHAR(10) désigne le type des chaînes alphanumériques formées de 10 caractères.
 - NUMERIC (ou DECIMAL ou DEC) qui permet de coder des nombres décimaux de façon exacte.
 - Le type FLOAT qui permet de coder des décimaux en base 2 de façon approchée (mais les calculs seront plus rapides qu'avec le type NUMERIC).
 - INT (ou INTEGER) qui permet de coder des entiers.
 - TEXT qui permet de coder des textes (éventuellement longs) de longueur indéterminée.
 - DATE et TIME pour la date et l'heure.

Il faut se souvenir que chaque attribut possède un type qui lui est propre (éventuellement son domaine peut être restreint par des contraintes). Par exemple, dans la relation *identité*, l'attribut *homme* est un booléen (1 si c'est un homme, 0 si c'est une femme).

Remarque 1.2. Les attributs ne sont pas ordonnés, par contre ils possèdent tous un nom et peuvent être appelées par celui-ci. Les lignes ne sont pas ordonnées également mais elles doivent être toutes différentes.

- **clé primaire** : Une clé primaire permet d'identifier de manière univoque chaque tuple d'une relation. C'est un sous-ensemble des attributs de la relation qui permet de distinguer chaque entrée de la relation, mais telle que, si on retire n'importe lequel des attributs de ce sous-ensemble, alors on ne peut plus distinguer certaines entrées de la relation. Par exemple dans la relation *identité* de notre exemple, à laquelle on aurait enlevé l'attribut *rowid*, on aurait pu choisir le sous-ensemble constitué des attributs *Nom* et *Prenom*, mais deux personnes peuvent avoir le même nom dans certains cas. Le sous-ensemble contenant que *rowid* convient dans tous les cas. En pratique la clé est souvent limitée à un seul attribut, qui prend des valeurs distinctes pour chaque entrée de la relation. Cet attribut sera identifié par le terme Cle ou Identifiant ou Id (éventuellement suivi de l'objet à identifier : Villes, Nom, Étudiant, Professeur.)
- **clé étrangère** : Une clé étrangère permet de référencer les entrées de certaines relations avec la clé (primaire) d'une autre relation : dans notre exemple, l'attribut *élèves* de la relation *notes* est une clé étrangère qui référence la clé primaire de la relation *identité*.
- **schéma relationnel** : Le schéma d'une relation est la donnée du nom de la relation et de chacun de ses attributs (avec ou sans mention de leurs domaines). Dans nos exemples, nous avons les deux schémas : *identité(rowid,Nom,Prénom,année,homme)*. Notez que l'on utilisera un symbole pour distinguer les clés primaires. . .

- **schéma de base de données** : est la donnée des différentes relations de la base, chacune avec ses attributs, et des références entre clés étrangères et clés primaires des différentes relations d'une base de données.

 Exercice 1.3

Effectuer le schéma de base de données de l'exemple de ce cours :

E

Définition 1.4: entité

On appelle entité un objet (concret ou abstrait) qui peut être reconnue distinctement et qui est caractérisée par son unicité.

Exemple 1.5

un élève (Albin félix, Poubanne Lisa, etc..), la copie d'un élève à un devoir (la copie de Lisa au devoir 3), etc ...

Définition 1.6: type entité

Un type entité désigne un ensemble d'entités qui possèdent une sémantique et des propriétés communes.

Exemple 1.7

des élèves, des copies de devoir (plus précisément, les notes de ces copies) , des notes d'interrogation, etc..

Définition 1.8: association

On appelle **association** un lien entre plusieurs entités

Exemple 1.9

L'**obtention** de la note de 5/20 par Albin Félix au premier devoir

Définition 1.10: type association

Un type association désigne un ensemble d'associations qui possèdent les mêmes caractéristiques. Le type association décrit un lien entre plusieurs types entités. Les associations de ce type association lient les entités des tables.

Exemple 1.11

L'**obtention des notes aux devoirs** entre le type entité **élèves** et le type entité **notes de devoirs**.

Définition 1.12: cardinalité

On appelle **cardinalité** d'un lien reliant un type association à un type entité le nombre de fois minimal et maximal d'interventions d'une entité du type entité dans une association du type association.

Remarque 1.13. *Même si on connaît le nombre maximal de fois qu'on utilise une entité (par exemple ici, si on sait qu'il y a 12 devoirs, on utilise donc au plus 12 fois un élève) on ne marquera pas ce nombre mais la variable n . En effet, comme les tables ne sont pas figées, ce nombre pourrait changer dans le temps (par exemple si on effectue un 13^e devoir, alors on devrait passer de 12 à 13).*

Remarque 1.14. La cardinalité minimale admise est soit 0 (on peut ne pas utiliser toutes les entités) soit 1 (on doit toutes les utiliser au moins une fois). La cardinalité maximale admise est soit 1 (on l'utilise au plus une fois), soit n (on l'utilise autant de fois qu'on veut). Il y a donc 4 possibilités :

- $0,1$: une entité peut être utilisée au plus qu'une seule fois et peut ne pas être utilisée.
- $0,n$: elle peut être utilisée sans limitation.
- $1,1$: elle peut être utilisée une et une seule fois.
- $1,n$: elle doit être utilisée au moins une fois et sans limitation.

Exemple 1.15

Le type association **obtention des notes aux devoirs** liant le type entité **élèves** au type entité **notes aux devoirs** et de cardinalité $0,n$ pour les élèves, $1,1$ pour les notes de devoirs.