

Chapitre 3 Traitement de données en tables

II Bases de données

Nous avons vu pour l'instant qui produisait des données et comment on pouvait les structurer (dans des tables par exemple). Voyons maintenant comment on traite ces données.

Les données sont donc enregistrées dans des bases de données et leur manipulation est gérée par des logiciels nommés "Systèmes de Gestion de Bases de Données" (SGBD). Les principaux SGBD modernes sont Oracle, DB2 d'IBM, SQL*Server de Microsoft pour les privés et MySQL et PostgreSQL pour les libres. Ils reposent tous sur la gestion de tables de données.

1) Création d'une base de données : un exemple concret

Imaginez que vous gériez une entreprise de bricolage. Vous vendez des produits à des clients. Vos clients vous font des commandes. Une commande avait, avant l'informatisation du monde de l'entreprise, cet aspect :

Commande N° : 30188		Date : 2/1/2009					
Numéro client	B512						
Nom	GILLET						
Adresse	14, r. de l'Eté						
Localité	Toulouse						
N° PRODUIT	LIBELLE PRODUIT	PRIX	QUANTITE	SOUS-TOTAL			
CS464	CHEV. SAPIN 400x6x4	220	180	39600			
PA45	POINTE ACIER 45 (20K)	105	22	2310			
PA60	POINTE ACIER 60 (10K)	95	70	6650			
PH222	PL. HETRE 200x20x2	230	92	21160			
TOTAL COMMANDE				69720			

Figure 1.1 - La pile de bons de commande de la semaine

- une table *détails*

Ces tables se présenteraient ainsi : (figure 1.3) Mais ce découpage en tables n'est pas encore satisfaisant. 2 problèmes subsistent :

- les données n'ont pas de lien entre elles. Il faudrait par exemple associer la commande 30188 au client B512. De même, il faudrait associer un détail à une commande.

- dans une table, on ne met pas de colonnes où des valeurs sont calculées à partir de valeurs déjà présentes dans la base de données.

C'est enregistrer des données pour rien. On fera faire ces calculs par le SGBD. Nous pouvons ainsi supprimer les champs (ou attributs) SOUS-TOTAL (= PRIXx QCOM) et TOTAL-COMMANDE (= somme des valeurs de SOUS-TOTAL).

données des commandes			données des clients			
NCOM	NCLI	DATECOM	NCLI	NOM	ADRESSE	LOCALITE
30188	B512	2/1/2009	B512	GILLET	14, r. de l'Eté	Toulouse
30179	C400	22/12/2008	C400	FERARD	63, r. du Tertre	Poitiers
données des détails						
NCOM	NPRO	QCOM	LIBELLE	PRIX	QCOM	SOUS-TOTAL
30188	CS464	180	CHEV. SAPIN 400x6x4	220	180	39600
30188	PA45	22	POINTE ACIER 45 (20K)	105	22	2310
30188	PA60	70	POINTE ACIER 60 (10K)	95	70	6650
30188	PH222	92	PL. HETRE 200x20x2	230	92	21160
30179	CS262	60	CHEV. SAPIN 200x6x2	75		
30179	PA60	20	POINTE ACIER 60 (10K)	95		

Figure 1.4 - Fichiers de données améliorés

Comment informatiser le traitement de milliers de commande comme celle-ci?

En créant des tables !

Quelles tables devons-nous créer ?

En réfléchissant un peu, on se rend compte qu'il faut créer 3 tables :

- une table *clients*
- une table *commandes*

données de la commande			données du client			
NCOM	DATECOM	TOTAL-COMMANDE	NCLI	NOM	ADRESSE	LOCALITE
30188	2/1/2009	69720	B512	GILLET	14, r. de l'Eté	Toulouse
données des détails						
NPRO	LIBELLE	PRIX	QCOM	SOUS-TOTAL		
CS464	CHEV. SAPIN 400x6x4	220	180	39600		
PA45	POINTE ACIER 45 (20K)	105	22	2310		
PA60	POINTE ACIER 60 (10K)	95	70	6650		
PH222	PL. HETRE 200x20x2	230	92	21160		

Figure 1.3 - Extraction des données du premier bon de commande

Nous complétons donc la structure des données en introduisant dans le fichier des commandes un champ (ou attribut) NCLI indiquant le numéro du client et dans le fichier des détails un champ NCOM spécifiant la commande de chaque détail (figure 1.4). Nous en profitons pour introduire également les données de la commande 30179.

Chapitre 3 Traitement de données en tables

Un dernier problème à régler. Dans la table *détails*, il peut y avoir des commandes différentes avec des libellés et prix identiques (lignes 3 et 6). C'est une répétition inutile et dangereuse parce qu'il faut que tous les libellés d'un même produit soit identique.

La solution est donc de créer une dernière table, la table *produits* :

données des commandes		
NCOM	NCLI	DATECOM
30188	B512	2/1/2009
30179	C400	22/12/2008

données des clients			
NCLI	NOM	ADRESSE	LOCALITE
B512	GILLET	14, r. de l'Eté	Toulouse
C400	FERARD	63, r. du Tertre	Poitiers

données des détails		
NCOM	NPRO	QCOM
30188	CS464	180
30188	PA45	22
30188	PA60	70
30188	PH222	92
30179	CS262	60
30179	PA60	20

données des produits		
NPRO	LIBELLE	PRIX
CS464	CHEV. SAPIN 400x6x4	220
PA45	POINTE ACIER 45 (20K)	105
PA60	POINTE ACIER 60 (10K)	95
PH222	PL. HETRE 200x20x2	230
CS262	CHEV. SAPIN 200x6x2	75

Nous avons retiré " LIBELLE " et " PRIX " de la tables *détails* et ajouté " NPRO " pour créer le lien avec la nouvelle table *produits*.

Voilà, nous venons de construire une base de données bien pensée !

Une base de données est constituée de tables de données ayant des liens entre elles.

2) Utiliser une base de données

Ainsi mises en forme, ces données permettent de répondre simplement à une infinie variété de questions. C'est à l'aide du langage SQL (Structured Query Language) qui est au programme de la terminale NSI, que nous pourrons interagir avec cette base de données, c'est-à-dire créer, consulter, modifier et supprimer les données. Par exemple, en avant-goût, l'extraction des données sur les clients de Toulouse s'obtiendra par l'instruction suivante, dont la compréhension à ce stade ne pose pas de problèmes :

```
select NCLI, NOM, ADRESSE
from CLIENTS
where LOCALITE = 'Toulouse'
```

Cette requête renvoie : B512, GILLET, 14 r. de l'Eté

Que renverrait cette requête ?

```
select NCOM
from COMMANDE
where NCLI in (select NCLI from CLIENT where LOCALITE = 'Toulouse')
```

Une base données renvoie des réponses à des requêtes exprimées dans le langage de programmation SQL. La réponse à une requête envoyée à une base de données peut être :

- des données sélectionnés selon certains critères
- des résultats de calculs à partir des données enregistrées
- d'insérer de nouvelles lignes dans une table
- de supprimer une ligne dans une table

3) Conclusion :

- Chaque ligne d'une table reprend les données relatives à une entité.
- Chaque colonne d'une table décrit une propriété commune des entités.
- Les lignes d'une table sont distinctes : pas de doublon. Il faut qu'un élément soit unique.
- Les lignes d'une table peuvent faire référence chacune à une ligne d'une autre table, où des informations sur une entité associée peuvent être obtenues. Les colonnes de référence s'appellent **clé étrangère** vers cette autre table.
- On évite d'enregistrer des données qu'il est possible de calculer à partir d'autres données enregistrées.