1.3 Travaux Dirigés – Sensibilisation à la problématique des bases de données

1.3.1 Introduction

Objectifs

L'objectif de ce TD est de se faire une idée de l'intérêt de toute la théorie sur la conception des bases de données et de l'intérêt de l'utilisation des systèmes de gestion de base de données. En d'autres termes, nous allons essayer d'apporter des éléments de réponse à la question :

> *« Pourquoi dois-je m'embêter avec toute cette théorie et ces connaissances à assimiler alors que je sais très bien manipuler un fichier, y stocker des informations et les y retrouver avec mon langage de programmation favoris ? »*

Contexte

Supposons que vous ayez à développer une application de gestion d'une bibliothèque. Tous les livres de la bibliothèque possèdent un numéro de livre, un titre, un ou plusieurs auteurs et un éditeur. Lorsqu'une personne emprunte un livre, il faut mémoriser son nom, son prénom, son numéro de téléphone, son adresse, la date de l'emprunt et la date de retour une fois ce dernier réalisé. Toutes les informations doivent être conservées pour garder un historique des emprunts.

1.3.2 Approche naïve

Une solution simple et naïve . . .

Certains d'entre vous ont une expérience des bases de données (il s'agit vraiment de quelque chose d'incontournable aujourd'hui) ou une expérience importante en développement logiciel. Dans le cadre de cet exercice, oubliez toutes vos connaissances et vos réflexions sur le sujet.

1. Votre application va devoir stocker toutes les informations mentionnées dans l'introduction (section *Contexte*), et de manière persistante, donc en utilisant un fichier. Quelle est la solution de stockage des données la plus naïve et la plus naturelle venant immédiatement à l'esprit ?

. . . mais pas sans conséquences

Supposons que nous adoptions la solution naïve et naturelle suivante :
- Nous créons un fichier texte comportant à l'origine une ligne par livre.
- Dans chaque ligne, on trouve les informations *titre*, `auteur`, `éditeur`, *numéro du livre* séparées par une tabulation.
- Quand une personne emprunte un livre, on complète la ligne du livre en question par les champs *nom*, *prénom*, *téléphone*, *adresse* et *date-emprunt* toujours en séparant ces informations par une tabulation.
- Lorsqu'une personne retourne un livre, il suffit d'ajouter un dernier champs *date-retour* sur la ligne du livre en question.
- Quand un livre est emprunté une nouvelle fois, on crée une nouvelle ligne avec toutes les informations concernant le livre et la personne qui l'emprunte. Bien entendu, le bibliothécaire ne ressaisit pas tout, l'application va chercher la plupart de ces informations dans le fichier.

En fait, on peut voir ce fichier texte comme un tableau de chaînes de caractères dont l'entête des colonnes seraient les suivantes :

Titre	Auteur	Éditeur	N°Livre	Nom	Prénom	Téléphone	Adresse	Date-emprunt	Date-retour

Supposons que l'application de gestion de bibliothèque fonctionne correctement et stocke toutes ses données dans un fichier comme celui que nous venons de décrire. Nous allons nous pencher sur les inconvénients et les conséquences inhérentes à une telle approche.

L'application fonctionne maintenant depuis 10 ans. Le nombre de personnes inscrites à la bibliothèque est relativement constant (bien que l'on constate un roulement) et de 5000 personnes en moyenne par an. Un abonné emprunte en moyenne 5 livres par mois.

2. Quel est, approximativement, le nombre de lignes du fichier des données ?
3. Quelle est la taille approximative du fichier sachant que chaque caractère occupe 1 octet et qu'une ligne contient, en moyenne, 150 caractères ?
4. Supposons qu'une personne est abonnée depuis l'origine de l'application. Elle prévient le bibliothécaire que son prénom est mal orthographié. Combien de lignes, approximativement, doivent être modifiées pour corriger cette erreur dans tout le fichier de données ?
5. Lorsqu'un abonné emprunte un livre, le bibliothécaire saisit simplement le numéro du livre et le nom et le prénom de l'abonné. L'application se charge alors de parcourir le fichier pour rechercher les informations manquantes concernant le livre et l'abonné afin d'écrire, à la fin du fichier, la nouvelle ligne concernant l'emprunt. Dans le pire des cas, l'application doit parcourir tout le fichier. Supposons qu'un accès au fichier coûte $10ms$, qu'une lecture de ligne coûte $6ms$ et qu'une recherche sur la ligne pour trouver le numéro du livre ou le nom et le prénom de l'abonné coûte $1ms$. Quel est, dans le pire des cas, le temps mis par l'application pour compléter les informations saisies par le bibliothécaire ?
6. Énumérez ou résumez tous les problèmes que la représentation des données choisie (le fichier de données) semble poser.

1.3.3 Affinement de la solution

Il est évident que la solution naïve décrite dans la section précédente pose de nombreux problèmes. Elle est totalement inacceptable pour une application sérieuse bien qu'elle soit encore largement employée dans des cas de petite taille (comme par exemple, dans la plupart des fichiers bibliographiques LaTeX).

Un premier affinage de la solution de la section précédente consiste à utiliser non pas un fichier unique mais quatre fichiers distincts :

- Un premier fichier est dédié au stockage des informations concernant les livres de la bibliothèque.
- Un second fichier est dédié au stockage des informations concernant les abonnés.
- Les informations stockées dans le troisème fichier vont permettre de faire la correspondance entre les deux premiers pour signifier qu'un livre donné est en cours de prêt par un abonné donné depuis une date donnée.
- Enfin, un dernier fichier va permettre de stocker l'historique des prêts. Il est similaire au troisième fichier, mais il comporte en plus une information relative à la date de retour du livre.

7. Précisez le format et les informations stockées dans chacun de ces quatre fichiers.
8. Quels sont les avantages de cette nouvelle solution ?
9. Intéressons-nous au premier fichier (celui concernant les livres). Quels problèmes diagnostiquez-vous dans ce fichier ?
10. Le format de ce fichier permet-il de prendre en compte des livres co-écrits par plusieurs auteurs ?
11. Quelle solution proposez-vous ?

1.3.4 Que retenir de ce TD ?

Les problèmes les plus courants rencontrés dans des bases de données mal conçues peuvent être regroupés selon les critères suivants :

Redondance des données – Certains choix de conception entraînent une répétition des données lors de leur insertion dans la base. Cette redondance est souvent la cause d'anomalies provenant de la complexité des insertions.

C'est, par exemple, le cas de la première organisation proposée : dès qu'un abonné emprunte un livre, il faut dupliquer toutes les information concernant l'abonné et le livre emprunté ! Au contraire, dans la deuxième solution, seuls les numéros indispensables à la distinction d'un livre et d'un abonné sont répétés dans le cas d'un emprunt.

Incohérence en modification – La redondance de l'information entraîne également des risques en cas de modification d'une donnée car on oublie fréquemment de modifier toutes ses occurrences.

Anomalie d'insertion – Une mauvaise conception peut parfois empêcher l'insertion d'une information, faute de connaître la valeur de tous ses champs. Pour remédier à ce problème, certains SGBD introduisent une valeur non typée qui signifie que la valeur d'un attribut est inconnue ou indéterminée. Cette valeur (appelée usuellement NULL) indique réellement une valeur inconnue et non une chaîne de caractères vide ou un entier égal à zéro.

Dans la première solution proposée, insérer un nouvel abonné qui n'a jamais emprunté de livre peut poser des problèmes. Une solution serait d'insérer des champs vides (suite de tabulations consécutives) au début de la ligne.

Anomalie de suppression – Enfin, une mauvaise conception peut entraîner, lors de la suppression d'une information, la suppression d'autres informations, sémantiquement distinctes, mais indissociables dans la modélisation adoptée.

Par exemple, dans la première solution proposée, si l'on désire supprimer toutes les traces d'un livre dans le fichier de données, on fera complètement disparaître tous les abonnés qui n'ont emprunté que ce livre.

Bien d'autres enjeux, que ceux que nous avons abordés, sont inhérents aux bases de données. Ces enjeux ont été survolés dans la section 1.2.2 et concernent la gestion des bases de données : indépendance physique, indépendance logique, accès aux données, administration centralisée des données, cohérence des données, partage des données, sécurité des données, résistance aux pannes, etc.

La conception des bases de données est donc un problème complexe. La gestion de ces bases constitue également un problème complexe. Or, ces deux problèmes sont extrêmement récurrents puisque les bases de données se trouvent aujourd'hui au cœur de tous les systèmes d'information. C'est pourquoi tout ces problèmes ont été largement étudiés et des solutions fiables et éprouvées ont été trouvées. De nombreux travaux ont ainsi permis de mettre au point une théorie permettant la conception de bases de données *bien formées*. C'est la problématique que nous abordons dans le chapitre 2. La problématique de la gestion des bases de données trouve une solution dans l'utilisation d'un SGBD.

Pour toutes ces raisons, j'espère que l'intérêt la théorie sur la conception des bases de données ainsi que l'intérêt de l'utilisation des systèmes de gestion de base de données deviennent évidant pour vous.