



Syllabus

Catégorie **Économique**

Fichiers et bases de données (UE204)

Introduction

**HELHA – Campus Mons – IRAM
Cours de Mr. Colmant
Année académique 2020-2021**

1) Origine du concept

Informatisation des organisations

L'informatisation des organisations a été réalisée de manière empirique, en automatisant des procédés manuels. Pour des raisons de commodité, ces procédés manuels entraînaient une importante redondance des informations. En effet, celles-ci devaient être présentes sur leur lieu d'utilisation. Par exemple, des informations décrivant des produits devaient être disponibles au même moment dans différents départements (comptabilité, production, commercial...).

Les premières applications de l'informatique ont été l'automatisation de tâches telles que l'édition des fiches de rémunération ou la facturation (traitements répétitifs appliqués à de grandes quantités d'informations).

Sans étude globale, la redondance a été transposée telle quelle dans les solutions informatiques. En conséquence, on a vu se multiplier des applications destinées à des utilisateurs différents mais utilisant les mêmes informations existant plusieurs fois dans différents fichiers.

Par ailleurs, les contraintes techniques de l'époque ont imposé un déplacement des informations vers les supports de l'ordinateur que seuls les informaticiens étaient capables de faire fonctionner.

Il y a donc eu perte de contrôle des utilisateurs sur "leurs " informations. Ce n'était plus eux qui géraient celles-ci mais bien les informaticiens, le manque de dialogue entre les uns et les autres ne faisant qu'ajouter à la frustration.

L'outil informatique a fini par se révéler fort coûteux et finalement peu rentable. Pour résoudre ces difficultés, on a constaté qu'il fallait :

- rendre aux utilisateurs le contrôle sur leurs informations ;
- éliminer les redondances en rapprochant les différents fichiers ;
- résoudre d'autres problèmes également identifiés et liés à des situations de dépendance logique et physique des fichiers.

Notion de dépendance logique

Habituellement, un fichier est géré par l'intermédiaire de programmes d'application. Pour qu'un programme d'application puisse manipuler un fichier, il faut qu'il contienne la description logique (DL) de celui-ci, c'est-à-dire la structure des enregistrements (attributs + représentation) .

Si, pour l'une ou l'autre raison, il est nécessaire de modifier cette description, il faut alors agir dans les textes sources de tous les programmes d'application exploitant le fichier, puis compiler et relier chacun d'eux (produire les exécutables). Tout programme d'application échappant à cette adaptation sera dans le meilleur des cas incapable de

fonctionner. Dans le pire des cas, il produira des résultats erronés. On dit qu'il y a dépendance logique des fichiers vis-à-vis des programmes d'application.

Notion de dépendance physique

Tout fichier a une organisation et doit être implanté sur support. La description d'implantation est réalisée soit dans les programmes d'application soit lors du reliage.

En conséquence, tout changement d'implantation physique entraîne au minimum un reliage pour chaque programme d'application. Tout programme d'application échappant à cette adaptation sera dans le meilleur des cas incapable de fonctionner et dans le pire des cas il produira des résultats erronés. Il y a dépendance physique des fichiers vis-à-vis des supports.

Élimination des dépendances

Les inconvénients présentés par ces deux formes de dépendance, ont conduit à élaborer des outils permettant de les faire disparaître

Élimination de la dépendance logique

Il faut que la description logique (DL) d'un fichier utilisé par un programme d'application ne soit plus présente dans celui-ci.

Dans la situation sans dépendance, les programmes d'application font référence à la DL du fichier pour manipuler celui-ci.

Si on doit modifier la structure du fichier, on adapte la description logique. Pour les programmes d'application, il n'y aura aucune modification, pour autant qu'ils trouvent toujours dans le fichier les informations dont ils ont besoin.

L'ensemble des informations sur support dans un enregistrement (ou plusieurs) constituent un fichier. Ainsi, à côté des fichiers de données, on voit apparaître un fichier dans lequel sont représentées les informations les décrivant. Ce fichier de descriptions de fichiers est appelé **dictionnaire de données (DD)**.

Ce DD doit être géré pour créer, modifier, supprimer et consulter des descriptions de fichiers. Cette gestion est assurée par un programme spécialisé, indépendant des programmes d'application, et appelé **Langage de description de données (LDD)**.

Dans cette présentation aucune hypothèse n'a été faite concernant le nombre d'utilisateurs ou le nombre de programmes d'application (PA) ou le contenu des fichiers. On peut donc très bien imaginer que des utilisateurs différents au moyen de PA différents fassent référence à une même description logique de fichier à laquelle est associée un seul fichier.

Alors une information employée par plusieurs utilisateurs peut être représentée une seule fois dans un seul fichier. En conséquence, on constate que les outils employés pour éliminer la dépendance logique (LDD et DD) permettent d'éliminer la redondance.

On peut donc envisager de constituer un ensemble de fichiers contenant des informations non redondantes et partagées par plusieurs utilisateurs.

Élimination de la dépendance physique

Cette dépendance peut être aisément éliminée. En effet, un fichier étant implanté à partir de sa description logique et d'informations décrivant la méthode d'organisation et le support, on constate que les programmes d'application ne sont plus du tout concernés. L'organisation peut être choisie, par défaut indexée pour la performance, et le système d'exploitation peut se charger des choix concernant le support.

On peut alors concevoir un programme spécialisé capable de produire, à partir d'une description logique de fichiers, une **description d'implantation physique (DIP)** associé à celle-ci et d'installer le fichier. Ce programme sera appelé **Langage de description d'implantation physique (LDIP)**.

Les programmes d'application sont totalement étrangers à ces opérations. Ils feront des demandes de manipulation (requêtes) qui utiliseront les descriptions logiques et les descriptions physiques pour fournir les informations demandées.

Manipulation de fichiers

Les programmes d'application émettent des requêtes de manipulation pour rechercher des enregistrements dans des fichiers en vue de les consulter, les modifier, les supprimer ou en créer.

Comme vu dans la première partie, on peut envisager la mise au point de fonctions standardisées de manipulation : recherche, consultation, création, modification et suppression d'enregistrements. Ces fonctions utilisent comme paramètres le nom du fichier, l'identifiant, les noms et occurrences d'attributs.

L'ensemble de ces fonctions constitue un **Langage de Manipulation de Données (LMD)**.

2) Sécurité des fichiers

Il faut assurer la confidentialité. En effet, on a maintenant un ensemble de fichiers contenant des informations non redondantes utilisables par différents utilisateurs. N'importe lequel d'entre eux n'a pas nécessairement accès à n'importe quel fichier ni à n'importe quelle information dans un fichier.

Il faut assurer l'intégrité. L'ensemble des informations non redondantes doit à tout moment être disponible dans un état correct et connu.

Pour cela, on utilise deux outils :

- Les profils utilisateurs. Pour chacun, on précise les informations accessibles et les manipulations autorisées (lecture, écriture). On a donc un ensemble d'informations qui décrivent ce qu'un utilisateur peut faire. Ces informations sont représentées dans

un fichier profil. Il y a un profil pour chaque utilisateur. Un tel fichier est géré (créé, modifié, supprimé, consulté) par un programme spécialisé appelé **Langage de description de profil (LDP)**.

- L'outil de gestion des accidents graves capable de reconstruire l'ensemble des fichiers à partir des copies (backups) et des logfiles.

Notion d'accès concurrents

Les informations sont non redondantes et mises à la disposition de différents utilisateurs. Lorsque plusieurs utilisateurs veulent utiliser la même information au même moment, il y a accès concurrent et parfois conflit (lors de mises à jour). Il faut donc un outil de gestion des accès concurrents.

3) Notion de base de données et de système de gestion de bases de données

Un ensemble de fichiers contenant des informations non redondantes mises à la disposition d'un ensemble d'utilisateurs en fonction de leurs besoins tel qu'il a été présenté constitue une **Base de Données (BD)**.

Une base de données est gérée par l'intermédiaire d'un ensemble de programmes spécialisés constituant un logiciel appelé **système de gestion de bases de données (SGBD)** dont les composantes sont donc :

- LDD (langage de description de données) gérant un DD ;
- LDIP (langage de description d'implantation physique) ;
- LDP (langage de description de profil) ;
- LMD (langage de manipulation de données) ;
- un outil de gestion des accès concurrents (utilisation de la notion de transaction) ;
- des outils de reconstruction de la base de données.

Les deux premiers outils assurent l'existence de la base de données. Le troisième assure la sécurité des données. Le quatrième permet la manipulation des données.

Notion de transaction

Une transaction (TX) est une unité logique d'échange entre un utilisateur et une base de données, structurée sur le modèle suivant :

- Ouverture de fichier (déclaration de début d'utilisation)
 - Traitements
- Fermeture de fichier (déclaration de fin d'utilisation)

On aura donc la structure générale pour toute transaction :

- Déclaration de début de Tx

- Manipulations
- Déclaration de fin de Tx

Le logiciel SGBD utilise les transactions pour distinguer les unes des autres les demandes multiples de traitement sur une base de données émanant de multiples utilisateurs.

Dans tous les cas, un utilisateur désirant travailler sur une base de données (et disposant d'un profil utilisateur) doit en manifester l'intention au moyen d'une déclaration de début de transaction. En l'absence de celle-ci, aucun traitement n'est possible.

Les traitements via le langage de manipulation des données sont ensuite effectués, en accord avec le profil. Après ceux-ci, l'utilisateur doit faire une déclaration de fin de transaction pour indiquer la fin des traitements. Cette déclaration de fin est indispensable pour donner un caractère définitif aux mises à jour éventuellement effectuées.

Dans tous les cas, l'absence de cette déclaration entraîne l'annulation des mises à jour effectuées. De la sorte, on est certain d'avoir à tout moment une base de données dans un état correct et connu (intégrité).

Transactions et accès concurrents

Présentation du problème

Il y a accès concurrent lorsque deux utilisateurs au moins désirent manipuler en consultation et/ou mise à jour un même enregistrement au même moment.

Lors d'une consultation, il n'y a pas de problème car on ne fait qu'une lecture et chacun des utilisateurs reçoit une copie de l'enregistrement dans son espace de travail en mémoire centrale.

Lors d'une mise à jour, il y a un risque de problème car plusieurs mises à jour simultanées d'un même enregistrement peuvent mettre celui-ci dans un état incohérent. La base de données n'est plus alors dans un état correct et connu.

Première solution

On peut envisager l'emploi de verrous. Un verrou est un indicateur associé à un enregistrement et pouvant avoir deux états : disponible ou occupé. Lorsqu'un utilisateur veut employer un enregistrement, l'état du verrou est testé :

- si l'état est occupé, la demande est rejetée ;
- si l'état est libre, l'enregistrement peut être employé et l'indicateur est basculé dans l'état occupé. L'enregistrement est manipulé puis l'indicateur est remis dans l'état disponible.

Ceci paraît simple mais comporte un risque de blocage complet du système. Considérant comme trop aléatoire un comportement des utilisateurs consistant à libérer volontairement un enregistrement, il a été nécessaire de mettre au point un outil automatique

fonctionnant pour les utilisateurs de manière transparente.

Deuxième solution

Cette solution va utiliser les transactions qui sont obligatoires pour les utilisateurs. Le SGBD est capable d'identifier les différentes transactions déclarées par les utilisateurs (par exemple au moyen d'un numéro de Tx). Lorsqu'une Tx sollicite un enregistrement et que celui-ci est disponible, le SGBD les associe. La Tx est alors active sur l'enregistrement, et toute autre Tx concurrente sur cet enregistrement est suspendue jusqu'à la déclaration de fin de la Tx active.

Transactions et reconstruction de bases de données

Le reconstruction d'une base de données est nécessaire dès lors qu'un accident la laisse dans un état incorrect ou inconnu et donc potentiellement incorrect. L'opération suppose l'existence de copies de sécurité (backups) prises régulièrement.

Il est également nécessaire d'utiliser un logfile dans lequel on trouvera une trace de tous les mouvements réalisés sur la base de données depuis la dernière copie de sécurité. Dans ce fichier logfile, on trouvera le plus simplement des copies des nouveaux états des enregistrements après manipulation par les transactions. A chaque fois qu'un enregistrement est manipulé en mise à jour par une transaction, il est non seulement copié dans la base de données mais il est également copié dans le logfile.