

Chapitre 1

PRESENTATION GENERALE DE MERISE

Il faut ouvrir des cadenas différents avec des clés différentes. (Proverbe chinois)

INTRODUCTION

MERISE est une méthode de développement des projets informatiques de gestion. Elle tire son nom du MERISIER qui est un arbre porte-greffe. De façon analogue, MERISE est le résultat de la greffe de plusieurs méthodes. Elle s'adresse à toutes les applications sur micro, mini-ordinateurs ou grands systèmes informatiques. Par commodité, l'organisme à informatiser sur lequel s'applique la méthode est appelé ici entreprise.

Ce chapitre d'introduction donne une vue globale de la méthode MERISE. Le lecteur pourra aborder dans les chapitres suivants le vocabulaire et les détails de la méthode avec plus de facilité

I - HISTORIQUE

MERISE (Méthode d'Etude et de Réalisation Informatique par Sous Ensembles ; ou aussi Méthode d'Etude et de Réalisation Informatique des Systèmes d'Entreprise) est née dans les années 1978-79 dans le cadre d'un groupement de sociétés de services placées sous l'égide du ministère de l'industrie au Centre Technique Informatique en France.

La méthode MERISE fait donc suite à une consultation nationale lancée en 1977 par le Ministère de l'Industrie dans le but de choisir des sociétés de conseil en informatique afin de définir une méthode de conception de systèmes d'information. Les deux principales sociétés ayant mis au point cette méthode sont le CTI (Centre Technique d'Informatique) chargé de gérer le projet, et le CETE (Centre d'Etudes Techniques de l'Equipement) implanté à Aix-en-provence. La naissance de MERISE marque une date importante dans l'histoire du traitement de l'information. cette naissance résulte d'une part de l'inadéquation des méthodes existantes comme MINOS ou CORIG aux préoccupations actuelles et de la génération des traitements conversationnels faisant suite aux bouleversements techniques des années 70 et , d'autre part , de nombreux travaux sur les bases de données et sur « l'approche systémique »

MERISE est actuellement la méthode la plus répandue dans l'espace francophone. Historiquement, la première version officielle de MERISE date des travaux coordonnés par le Ministère de l'industrie en 1979 ; le groupe de projet comprenait, outre une équipe de recherche dirigée par M. H. TARDIEU, plusieurs sociétés de service. Depuis, plusieurs versions ont été développées. Voici venu le temps des MERISES.

En fait, MERISE a véritablement commencé à être utilisée au début des années 80. A partir de 1986, la méthode MERISE s'est véritablement imposée comme standard de fait avec un nombre d'utilisateurs en forte croissance. Aujourd'hui, MERISE est donc devenue une méthode académique, et est largement utilisée dans les administrations et les entreprises de toutes tailles.

CE QU'APPORTE MERISE

La méthode MERISE apporte une formalisation éclairant les choix à effectuer. Elle est un langage commun de référence centré sur le système d'information et non sur l'informatique appliquée. Elle permet une authentique communication entre le responsable de la stratégie d'entreprise, celui de son informatisation et les utilisateurs finaux. Elle structure les vœux du dirigeant et de l'utilisateur sous forme de *dessins* pour une compréhension facile et de *dossiers* pour une explication complète permettant au responsable de l'informatique de choisir ou d'écrire les programmes. Le choix final d'informatisation sera conforme aux désirs de l'utilisateur et aux possibilités informatiques.

Les dessins illustrent la syntaxe du langage en décrivant l'agencement des informations et l'ensemble des tâches à réaliser. Ainsi, le projet sera mené avec rigueur et cohérence afin d'obtenir les objectifs de fiabilité des données et d'évolution des applications.

Enfin, MERISE est une méthode qui conduit à une réflexion sur l'entreprise et peut aider à modifier son *organisation*, voire d'en créer une nouvelle.

II - LA DEMARCHE MERISE

Parmi les informations qui appartiennent au système d'information, certaines doivent ou peuvent faire l'objet d'un traitement automatisé grâce aux outils informatiques. Pour assurer la cohérence du système d'information, la méthode MERISE propose une démarche d'informatisation comportant les étapes suivantes :

A - LE SCHEMA DIRECTEUR

Son rôle est de définir, de manière globale, la politique d'organisation et d'automatisation du système d'information. Pour ce faire, il est nécessaire de répertorier l'ensemble des applications informatiques existantes à modifier et à développer. Pour rendre contrôlable et modulable ce développement, il est nécessaire de découper le système d'information en sous-ensembles homogènes et relativement indépendants. Ces sous-ensembles sont appelés domaines. *Par exemple, on peut trouver le domaine « Approvisionnement », le domaine « Personnel ».* Les résultats attendus à la fin de cette étape sont une définition précise des domaines, une planification du développement de chaque domaine et un plan détaillé, période par période, des applications qui doivent être réalisées.

B - L'ETUDE PREALABLE PAR DOMAINE

Elle doit aboutir à une présentation générale du futur système de gestion (modèles des données et des traitements) en indiquant les principales novations par rapport au système actuel, les moyens matériels à mettre en œuvre, les bilans, les coûts et les avantages. Cette étude est réalisée en 4 phases qui sont :

- La **phase de recueil d'informations** qui a pour objectif d'analyser l'existant afin de cerner les dysfonctionnements et les obsolescences les plus frappantes du système actuel.
- La **phase de conception** qui a pour objectif de formaliser et hiérarchiser les orientations nouvelles en fonction des critiques formulées sur le système actuel et d'autre part des politiques et des objectifs de la direction générale. Cela revient à modéliser le futur système avec une vue pertinente de l'ensemble.
- La **phase d'organisation** dont l'objectif est de définir le système futur au niveau organisationnel: le qui fait quoi ?
- La **phase d'appréciation** dont le rôle est d'établir les coûts et les délais des solutions définies ainsi que d'organiser la mise en œuvre de la réalisation. A cet effet un découpage en projets est effectué.

C - L'ETUDE DETAILLEE PAR PROJET

Elle consiste d'une part à affiner les solutions conçues lors de l'étude préalable et d'autre part à rédiger, pour chaque procédure à mettre en œuvre, un dossier de spécifications détaillé décrivant les supports (maquettes d'états ou d'écran) ainsi que les algorithmes associés aux règles de gestion... A l'issue de cette étude, il est possible de définir le cahier des charges utilisateurs qui constitue la base de l'engagement que prend le concepteur vis à vis des utilisateurs. Le fonctionnement détaillé du futur système, du point de vue de l'utilisateur, y est entièrement spécifié.

D - LA REALISATION

Son l'objectif est l'obtention des programmes fonctionnant sur un jeu d'essais approuvés par les utilisateurs.

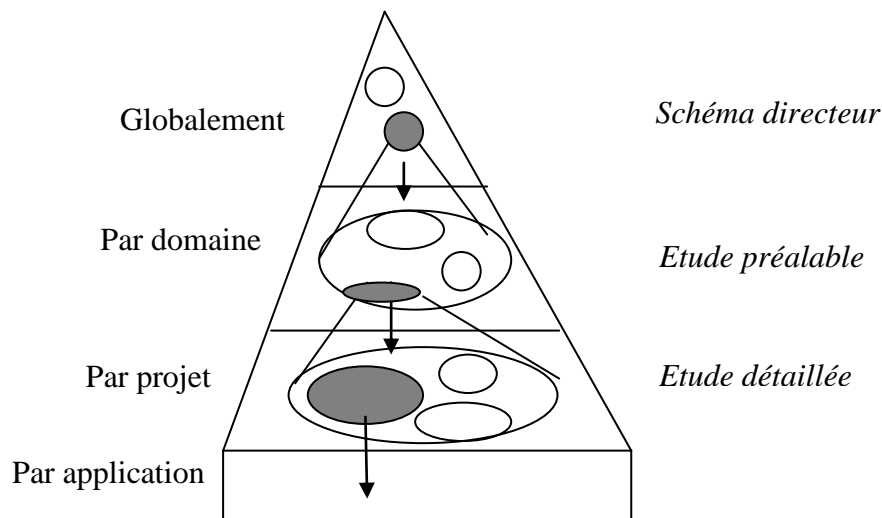
E - LA MISE EN ŒUVRE

Elle se traduit par un changement de responsabilité : l'équipe de réalisation va en effet transférer la responsabilité du produit à l'utilisateur. Cette étape intègre en particulier la formation des utilisateurs. Après une période d'exploitation de quelques mois, la recette définitive de l'application est prononcée.

F - LA MAINTENANCE

Elle consiste à faire évoluer les applications en fonction des besoins des utilisateurs, de l'environnement et des progrès technologiques.

Le schéma suivant, extrait de l'ouvrage « La méthode Merise » reprend les étapes décrites ci-dessus.



Cette démarche lourde et parfois complexe est adaptée à l'automatisation de « gros systèmes d'information ». Pour des informatisations plus modestes, elle peut être perçue comme un carcan, et il convient donc de l'adapter afin de retenir uniquement les concepts et/ou les étapes appropriées aux besoins.

III - LES NIVEAUX DE DESCRIPTION

Le but de cette méthode est d'arriver à concevoir un système d'information. La méthode MERISE est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques. La séparation des données et des traitements assure une longévité au modèle. En effet, l'agencement des données n'a pas à être souvent remanié, tandis que les traitements le sont plus fréquemment.

Même si la méthode MERISE est, avant tout, une méthode de conception de systèmes d'information, et non de systèmes informatiques, il apparaît aujourd'hui que les systèmes d'information sont largement gérés par des applications informatiques. Les modèles MERISE doivent donc être utilisés pour faciliter le développement de ces applications en s'appuyant sur les technologies logicielles actuelles telles que les bases de données relationnelles et/ou l'architecture client-serveur.

MERISE distingue trois niveaux dans la description et un ensemble de modèles pour la représentation des données et des traitements, un formalisme de représentation est associé à chaque modèle. L'ensemble de ces trois niveaux constitue le **cycle d'abstraction** de la conception d'un SI.

A - LE NIVEAU CONCEPTUEL : Les objectifs de l'entreprise

L'informatique consiste à mettre à disposition de l'utilisateur des moyens ou des outils de gestion informatique. Avant de spécifier les moyens informatiques, il est nécessaire de définir le travail de cet ou de ces utilisateurs finaux, de définir l'organisation du travail au sein de l'entreprise. Afin de déterminer cette organisation, l'analyse des objectifs et des fonctions majeures de l'entreprise doit être menée. *Ainsi, l'informatisation est conçue en fonction de l'organisation et l'organisation en fonction des objectifs à atteindre.*

Le **niveau conceptuel** décrit donc la statique et la dynamique du système d'information en se préoccupant uniquement du point de vue du gestionnaire. Ce niveau de préoccupation correspond aux finalités de l'entreprise. Il s'agit de décrire le "quoi" en faisant abstraction des contraintes d'organisation et techniques. Les modèles utilisés pour la description conceptuelle du SI sont :

1 - Le Modèle Conceptuel des Données (MCD)

La structure de mémorisation des informations est représentée sous une forme qui permet un passage aisé vers les "enregistrements informatiques". Il est question ici de la description des données et des relations réalisée à l'aide des concepts et de formalismes

2 - Le Modèle Conceptuel des Traitements (MCT)

Il s'agit de la description de la partie dynamique du SI

B - LE NIVEAU LOGIQUE OU ORGANISATIONNEL : Les postes de travail

Pourquoi une organisation ? Pour réaliser les fonctions de l'entreprise décrites dans la première partie. Cela répond à la question QUI FAIT QUOI ? Dans le cas de développement sur micro-informatique ou dans le cas où l'application ne concerne qu'une seule personne, le niveau organisationnel se ramène à sa plus simple expression, un seul poste de travail. *Conceptuel et organisationnel représentent toute l'entreprise.*

Plus variable, est *la forme* que doit prendre l'outil informatique pour être adapté à l'utilisateur, à son poste de travail. C'est le niveau logique, la maquette des enchaînements d'écrans et la réponse à la question AVEC QUOI ? Le niveau logique est indépendant de l'informatique spécifique, des langages de programmation ou de gestion des données.

Le **niveau organisationnel** décrit la nature des ressources qui sont utilisées pour supporter la description statique et dynamique du système d'information. Ces ressources peuvent être humaines et/ou matérielles et logicielles. Les choix d'organisations sont pris en compte à ce niveau :

- La répartition des traitements entre l'homme et la machine.
- Le mode de fonctionnement : temps réel ou temps différé.
- La répartition géographique des données et des traitements.

1 - Le Modèle Logique des Données (MLD)

Il représente l'ensemble des données par type de site organisationnel.

2 - Le Modèle Organisationnel des Traitements (MOT)

Il permet de représenter par procédures, les phases et les tâches exécutées par chaque poste de travail. En gros, à ce niveau les «qui fait quoi ? » et « où ? » sont décrits

C - LE NIVEAU PHYSIQUE OU OPERATIONNEL : L'informatique spécifique

Le dernier niveau, le plus variable, est l'outil informatique lui-même, les fichiers et les programmes. AVEC QUOI ? Ce niveau est appelé niveau physique. Ce niveau dépend à 100% du système informatique retenu, du type de la base de données et des outils de développement. MERISE est d'un secours précieux dans le cadre des données. La structure "physique" informatique des données tend à être normalisée. Le passage, à l'aide de règles, à ces représentations normalisées est facile. C'est à cet instant que la méthode justifie son utilisation dans le cadre de développement sur micro-ordinateurs. Le modèle conceptuel de données engendre le modèle physique de données.

A ce niveau, les choix techniques sont définis. Ainsi, les organisations physiques de données sont spécifiées au travers du Modèle Physique des Données (MPD) et la description des traitements est réalisée pour chaque transaction (temps réel) ou chaque unité de traitement (temps différé) au travers du Modèle OPérationnel des Traitements (MPT ou MOPT). Ici, le « comment faire » est décrit. Le **niveau opérationnel** est donc celui dans lequel on choisit les techniques d'implantation du système d'information (données et traitements)

La hiérarchisation des préoccupations adoptées par MERISE consiste à intégrer au niveau le plus élevé, les éléments les plus stables ou quasi invariants, c'est-à-dire les éléments qui ne sont modifiés que si l'on bouleverse les règles de gestions fondamentales de l'organisation. La hiérarchie MERISE s'applique tant à la description statique du SI qu'à sa description dynamique.

| Niveaux | Préoccupation | Données | traitements | choix |
|---|--|---|---|----------------|
| CONCEPTUEL | Quoi ? Avec quelles données ? Qu'est ce qu'on va faire ? | MCD Représentation Conceptuelle des données sous formes d'entité et d'association | MCT Représentation concept des traits, sous formes d'opération déclenchées par des événements | DE GESTION |
| LOGIQUE OU ORGANISATIONNEL | Qui fait quoi ? Quand Comment | MLD Description logique en fonction des schémas disponibles (fichiers réseaux, relations,...) | MOT Simulation organisationnelle des traitements, intégration des ressources (hommes et machine en interaction) | D'ORGANISATION |
| PHYSIQUE OU OPERATIONNEL | Avec quels moyens ? | MPD Représentation Logique des données en mémoire en fonction de la fréquence des accès de temps de réponse attendus des choix de répartition | MOPT Application et communication en fonction des architectures, langages, configuration et environnements techniques | TECHNIQUES |

Du fait de ce découpage (qui a été introduit pour faciliter l'analyse d'un problème) seul le premier niveau est réellement indépendant de toute considération technologique : logicielle ou matérielle. *Par exemple, si les données du futur système d'information doivent être gérées par un SGBD, c'est au niveau organisationnel que le choix du type du SGBD (relationnel, réseau ou objets) devra être effectué.* La description statique du système d'information à ce niveau sera donc basée sur l'organisation des bases relationnelles, ou réseau, ou objets. Le troisième niveau est encore plus dépendant de l'aspect technologique puisqu'il cherchera à optimiser l'implantation. Il suppose donc une connaissance très pointue de l'architecture et des fonctions du SGBD qui gèrera le système d'information.

L'étude des technologies logicielles, telles que les types de SGBD ou encore l'architecture client-serveur, sortant du cadre de ce cours, celui-ci se focalisera sur le niveau conceptuel tant au niveau des données que des traitements

Une approche rigoureuse doit être faite dans le processus de conception d'un système ; d'où l'importance de suivre pas à pas le cycle d'abstraction qui est offert par une méthode de conception de système d'information.

Plusieurs méthodes de conception de SI existent. Chacun a un cycle d'abstraction qui lui est propre, mais tous concourent aux même objectifs : l'obtention d'un SI de qualité.

IV - LES METHODES D'ANALYSE

La conception d'un système d'information n'est pas évidente car il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel on va s'appuyer. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels on s'intéresse. Ce type de méthode est appelé *analyse*. Il existe plusieurs méthodes d'analyse, cependant la méthode la plus utilisée dans la francophonie étant la méthode MERISE

A - LES CARACTERISTIQUES DE LA METHODE D'ANALYSE

L'analyse du SI doit permettre, à partir de la constatation d'une situation donnée (structure et fonctionnement du SI actuel) avec ses forces et ses faiblesses, de concevoir un état souhaitable du SI, d'en préparer et soutenir l'éventuelle mise en place. L'analyse donc doit conduire à la conception du SI. La conception n'existe pas sans l'analyse qui n'a de sens que dans la perspective de la conception. Le but avoué de l'analyse est généralement l'informatisation du SI, même si elle ne doit pas en être obligatoirement l'aboutissement. Toute méthode d'analyse se caractérise par le développement plus ou moins poussé des 4 composantes suivantes :

- **Des modèles** (ensemble de concepts et de règles destiné à expliquer et représenter les phénomènes organisationnels).
- **Des langages** : ensemble de constructions qui permettent d'écrire formellement les images du SI, éventuellement en faisant appel à des modèles.
- **Une démarche** : processus opératoire par lequel s'effectue le travail de modélisation, de description et de réalisation du SI.
- **Des outils** : outils logiciels supportant la démarche, outils de documentation, de simulation ou de réalisation du SI.

Exemples : - Méthode privilégiant la démarche : *AXIAL*
- Méthode privilégiant les modèles : *REMORA, MERISE, IDA*
- Méthode ayant peu d'outils : *MERISE, AXIAL, UML*

B - DEMARCHE ASCENDANTE

Elle est basée sur une démarche dont le point de départ est constitué par les données brutes perçues dans le SI étudié pour aboutir grâce à l'analyse des dépendances fonctionnelles à un modèle de données. Elle est particulièrement adaptée quand on est en présence d'une nouvelle application à automatiser (sans existant préalable). Elle consiste à étudier toutes les sorties à obtenir (écran, état imprimé) et à remonter vers les données nécessaires à l'obtention des résultats figurant sur ces sorties.

Ici, le recueil des rubriques se fait à partir des sorties, imprimés et écrans. Il suffit ensuite de remonter aux entrées grâce à la connaissance des algorithmes et règles de gestion utilisées dans l'organisation. Cette méthode paraît mal adaptée pour répondre aux besoins nouveaux. Elle est particulièrement adaptée quand on est en présence d'une nouvelle application à automatiser (sans existant préalable).

C - DEMARCHE DESCENDANTE

Elle est fondée sur l'étude des structures types. Elle est particulièrement adaptée à un existant préalable. Elle consiste à recenser toutes les informations du si rencontrées sur les divers documents en services et à y ajouter les nouvelles données nécessaires aux nouveaux traitements. Elle est la plus intuitive. Elle se fonde sur une vision globale du SI et des fonctions que celui-ci doit assumer. Partant des objectifs sans en connaître tous les détails, l'analyste détermine les voies et moyens permettant d'y parvenir. Toutefois, dans la pratique, un minimum de connaissance est nécessaire, car

il n'est pas souhaitable de tout réinventer. Il s'agit, en fait, de construire une représentation du SI à partir des descriptions faites par les utilisateurs et les décideurs.

Connaissant les informations utilisées et les fonctions à assurer, y compris celles exprimées dans les besoins nouveaux, il est alors possible de rechercher les entités et les liens qui peuvent les unir. Concernant les traitements, c'est la description de la fonction qui amène à redéfinir les sorties, les entrées et les algorithmes qu'elle suppose.

L'avantage d'une telle méthode réside essentiellement dans le fait qu'elle ne reprend à son compte que le minimum de solutions existantes. Ainsi, est-elle à même de proposer le cas échéant des moyens plus efficaces.

C'est la méthode la plus souvent utilisée et conseillée par les promoteurs de MERISE.

Sans entrer dans la tanière du tigre, comment capturer ses petits ? (Proverbe chinois)

INTRODUCTION

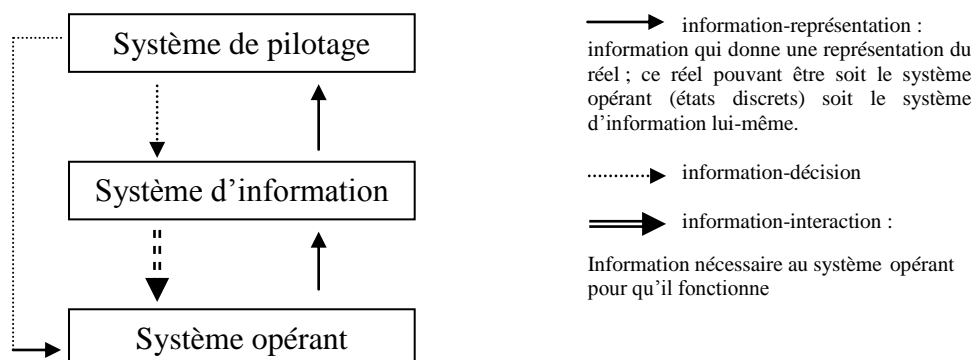
Un système est un ensemble d'éléments matériels ou/et immatériels (hommes, machines, méthodes, règles etc.) en interaction dynamique transformant par un processus des éléments (les entrées) en d'autres éléments (sorties). Appliqué à une organisation sociale (entreprise, association, administration, ...), le système d'information est un ensemble d'éléments en interaction dynamique, organisé en fonction d'un but. Les éléments en interaction sont constitués de l'ensemble des moyens humains, matériels, et immatériels. Les organisations forment un système ouvert (donc en relation avec l'environnement), vivant et évolutif.

Le système d'information (SI) est composé d'éléments divers (employés, ordinateurs, règles et méthodes, ...) chargés de stocker et de traiter les informations relatives au système opérant afin de les mettre à la disposition du système de pilotage.

L'information étant la matière première du traitement informatique, il s'avère indispensable d'aborder dans ce chapitre la notion de système d'information et de flux, base de l'analyse des données et des traitements.

I - PRESENTATION DU SI

A- DESCRIPTION DU SI



Le SI a pour objectif de mettre à la disposition du système de pilotage toutes les informations collectées lui permettant de prendre des décisions et de permettre la transmission de ces informations au système opérant ou à destination de l'environnement

- Le système de pilotage décide des actions à conduire sur le système opérant en fonction des objectifs et des politiques de l'entreprise,
- Le système opérant englobe toutes les fonctions liées à l'activité propre de l'entreprise : facturer les clients, régler les salariés, gérer les stocks, ...

Une telle décomposition prend bien en compte :

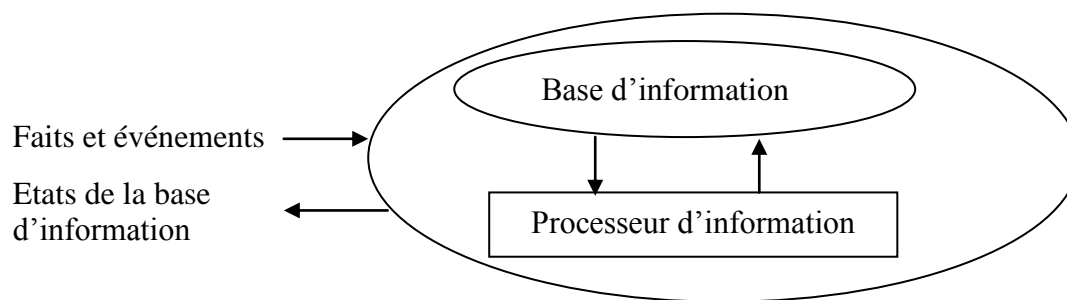
- la différence de besoins en matière d'information des modules opérants et pilotes,
- la nécessité pour le système d'information de ne pas se contenter de transmettre les informations mais d'en changer le niveau de synthèse.

Dans certaines organisations, on peut trouver des formes plus intégrées du système d'information. Cette intégration peut se faire soit au niveau du système opérant, soit au niveau du système de pilotage.

- **Un système d'information intégré au système opérant** ne décrit plus le fonctionnement du système opérant mais il est intégré à ce fonctionnement. Par exemple dans un système de GPAO (Gestion de Production Assistée par Ordinateur), les décisions de pilotage sont directement traduites en des décisions d'exécution de règles incluses dans une gamme opératoire.

- **Un système d'information intégré au système de pilotage** doit permettre d'enregistrer les décisions prises lors de diverses situations afin de rendre le pilotage plus intelligent. Ces Systèmes Interactifs d'Aide à la Décision (S.I.A.D) ont une architecture proche de celle des systèmes experts et font donc largement appel pour leur conception aux techniques de l'intelligence artificielle.

Le système d'information doit décrire (ou représenter) le plus fidèlement possible le fonctionnement du système opérant. Pour ce faire, il doit intégrer une base d'information dans laquelle sera mémorisées la description des objets, des règles et des contraintes du système opérant. Cette base étant sujette à des évolutions, le système d'information doit être doté d'un mécanisme (appelé processeur d'information) destiné à piloter et à contrôler ces changements. Le schéma suivant synthétise l'architecture d'un système d'information.



Le processeur d'information produit des changements dans la base d'information à la réception d'un message. Un message contient des informations et exprime une commande décrivant l'action à entreprendre dans la base d'information. Le processeur d'information interprète la commande et effectue le changement en respectant les contraintes et les règles.

Si le message exprime une recherche sur le contenu de la base d'information, le processeur interprète la commande et émet un message rendant compte du contenu actuel de la base d'information. Dans tous les cas, l'environnement a besoin de connaître si la commande a été acceptée ou refusée. Le processeur émet, à cet effet, un message vers l'environnement.

Relativement à la conception d'un système d'information, l'architecture présentée ci-dessus induit une double conception :

- celle de la base d'information (aspect statique)
- celle du processeur de traitement (aspect dynamique)

Pour aider le concepteur dans ces deux tâches, la méthode MERISE propose un ensemble de formalismes et de règles destinés à modéliser de manière indépendante les données et les traitements du système d'information. Ces modèles ne sont qu'une base de réflexion pour le concepteur et un moyen de communication entre les divers acteurs du système d'information dans l'entreprise.

B - Rôle du SI

Le SI se veut donc être la mémoire de l'organisation et comporte un aspect statique (enregistrement des structures de données, ensemble des données) et un aspect dynamique (mise à jour des données, changement des structures de données, règles,...). L'élément actif du SI constitue le processeur d'information qui traite l'information.

En général, on peut attribuer 4 rôles principaux à un système d'information dans une organisation.

1. Répondre à des obligations légales réclamées par l'environnement socio-économiques

***Exemple** : bulletin de salaire, factures, bilan comptable*

2. Déclencher les décisions programmées.
3. Aider à la prise de décision en fournissant aux décideurs un ensemble d'informations brutes ou modélisées. ***Exemple** : statistique, simulations, tableaux de bord, ...*
4. Assurer la condition des tâches en permettant la communication entre les hommes.

II - SYSTEME D'INFORMATION AUTOMATISABLE

A - ACTIONS PROGRAMMEES

Dans un système les actions "programmées" (avec ou sans ordinateurs) sont des actions qui déterminent de manière unique les sorties à partir des entrées. On dit alors que le système est déterminé. Il s'agit d'un système sans décision (on a alors la fonction $S = f(e)$)

***Exemple** : la connaissance de la capacité d'un magasin détermine de manière unique la quantité de produits à stocker.*

B - SYSTEME AVEC DECISION

Dans un système, une entrée peut conduire à plusieurs sorties possibles. Le choix de la sortie se fait donc par une décision.

***Exemple** : la connaissance du niveau du stock ne détermine pas uniquement les quantités à commander au fournisseur. Il faut ici opérer un choix. Des éléments non formalisables (intuition, intérêts personnels, expériences professionnelles, habitudes, clientèle ...) peuvent intervenir dans un choix.*

En conclusion, on remarque que dans un système, le processus qui transforme les entrées en sorties peut comporter :

- Des actions programmées
- Des choix (décisions)

C - DESCRIPTION D'UN SI AUTOMATISABLE

Pour qu'un système d'information soit automatisable, il doit être formalisable, c'est à dire la connaissance des entrées doit permettre de déterminer les sorties par des règles de transformations explicites. Ainsi seules seront automatisables les parties du SI qui ne contiennent que des actions programmées. Toutefois, les choix peuvent éventuellement être transformés en actions programmées aux moyens d'un modèle, donc deviennent automatisables.

***Exemple** : les décisions de réapprovisionnement peuvent s'effectuer pour utilisation d'un modèle de gestion des stocks tels que : Si **Stock** < 50 alors commander 200*

III – SYSTEME AUTOMATISE D'INFORMATION (SAI)

A - Définition

Un SAI est un sous système d'un système d'information dans lequel toutes les transformations significatives d'informations sont effectuées par des ordinateurs.

B - Objectifs du SAI

Le SAI permet la simplification et l'amélioration du travail administratif d'une part (compatibilité, facturation, paye, etc.) et constitue une aide à la décision d'autre part (simulation, statistique, ...).

C – SOUS-SYSTEMES FONCTIONNELS DU SAI

Dans un SAI le processeur d'information est constituées par un ou plusieurs ordinateurs pilotés par le personnel d'exploitation et le système d'exploitation, en liaison avec l'univers extérieur au moyen des périphériques de communication, des supports de saisie et de personnels de saisie et en liaison avec le modèle et la base d'informations au moyen des périphériques de stockages. Le modèle et la base sont stockées sur des mémoires externes (mémoires magnétiques : disques disquettes, ...). Il se dégage quatre sous - systèmes fonctionnels

❖ Deux systèmes internes au SAI

- Le traitement automatique
- La mémorisation

❖ Deux systèmes interfaces avec l'univers extérieur (traitement de la communication)

- La saisie
- L'accès

1 – Les systèmes internes

a - La mémorisation

C'est la fonction de stockage des informations réalisée sur les mémoires externes.

- Stockage de programmes et de la structure des données
- Stockage des données (fichiers ou base de données)

b - Le traitement automatique (TA)

C'est la fonction consistant à manipuler des données mémorisées ou provenant de l'extérieur (saisie) effectuée par l'ordinateur). Il s'agit :

- **Des contrôles** qui consistent à valider les données saisies respectant des contraintes du modèle (format, masque, ...)
- **Des mises à jour** permettant de transformer les données existantes en données de valeurs nouvelles (ajout de nouvelles données, modification, suppression de données))
- **Des recherches** qui consistent à sélectionner parmi les données de la base d'information celles qui répondent à certains critères.
- **Des calculs** qui permettent d'élaborer des données nouvelles à partir des données saisies ou mémorisées, selon des règles précises.

2 – Les systèmes interfaces

a - La saisie

Il s'agit de communiquer au SAI des informations en provenance d'événements de l'univers extérieur (entrées des données dans l'ordinateur).

b - L'accès (édition)

Il s'agit de transformer des données mémorisées de la base d'information ou des données résultats issues d'un traitement automatique en sorties externes vers l'univers extérieur.

Le SAI est un sous système du SI qui comporte en son sein des parties manuelles (hommes + matériel non automatique) et des parties automatiques (ordinateurs) ou plus exactement des imbrications de parties manuelles et automatiques.

Remarque : un SAI est intégré si une même information n'est saisie qu'une fois en un point du système et est répercutée dans tous les fichiers concernés.

D - LE PARAMETRAGE

Un SAI pour durer doit être adaptable à toute situation. L'application (le logiciel) doit pouvoir résister aux changements sans qu'on ait à modifier les programmes. En d'autres termes, le logiciel doit être fortement paramétrable

Exemple : Au lieu de programmer :

Si CTVA = 1 alors TVA = HT * 18,6%

Sinon si CTVA = 2

Alors TVA = HT * 11,11%

On programmera tout simplement

| Code TVA | taux |
|----------|--------|
| 1 | 18,6% |
| 2 | 11,11% |

Si CTVA = code (1) alors TVA = HT * taux (1)

Et l'ordinateur lira dans le fichier TVA, les valeurs des données des tables code et taux.

Si par exemple le taux 1 change et passe à 16%, il suffira de mettre à jour le fichier TVA en remplaçant 18,6% par 16%, sans modification du programme.

En conclusion, le paramétrage équivaut à une adaptabilité du SAI et à une informatique souple et évolutive.

IV - LE SYSTEME D'INFORMATION DE GESTION (SIG)

A - LES COMPOSANTES DU SIG

Le SIG peut se comprendre à partir de 4 ensembles :

- Les domaines de gestions
- Les données
- Procédures de traitements
- Les règles de gestions

1 - Les domaines de gestions

Ils correspondent chacun à des activités homogènes

Exemple : domaine technique, domaine commercial, domaine financier, etc....

2 - Les données

Elles constituent la matière 1^{ère} du SIG et sont présentées sous toutes les formes (écrites, orales, etc....) et sur tous les types de supports (disque, disquettes, tableau, cassette, télévision, téléphone, papier etc....)

3 - Les modèles

Ils regroupent l'ensemble des procédures propres au domaine

Exemple : *plan comptable (comptabilité- finance)*

4 - Les règles de gestions

Elles permettent d'utiliser, de transformer ou de traiter les données en fonction des objectifs assignés au système.

Exemple : *la quantité de production par heure ; aucune visite aux heures de travail.*

B - DEFINITIONS

1 - Domaines de gestions

- Le domaine de gestion est un sous système qui comme tout système possède un système opérant, un système d'information et un système de décision.

- Le domaine de gestion regroupe un ensemble d'activités ayant le même objectif global ou la même finalité au sein de l'ensemble.

Un domaine de gestion peut donc couvrir de manière transversale, plusieurs départements ou services de l'entreprise.

Exemples : *Gestion des approvisionnements, Gestion des stocks, Gestion des commandes*

2 - projet

Chacun des domaines de gestion a une action propre qui autorise une nouvelle décomposition. On parle alors de **projets**.

Exemple : *Le domaine personnel peut se décomposer en :*

- *Gestion des permanents*
- *Gestions des journaliers*
- *Gestion des salaires*
- *Gestion de la carrière*
- *Gestion de la formation*

3 - Application

La décomposition du projet donne des **applications**

Exemple : *le domaine personnel est décomposé entre autres en projets « gestions de la carrière », « gestion des salaires », et le projet « gestion des salaires » lui même se compose en plusieurs applications :*

- *Salaires cadres supérieurs*
- *Salaires agents de maîtrise*

C - ROLES DU SIG

Les informations proviennent de différentes origines (environnement, différents services...) et apparaissent sous différentes formes (orales, graphiques, tablée...) avant de parvenir au système. Elles sont rarement exploitables en l'état et nécessitent souvent un traitement préliminaire avant d'être exploitées en fonction des objectifs de l'entreprise.

Le rôle du SIG est de collecter l'ensemble des informations, de les traiter et de les acheminer au moment opportun à leur destinataire.

1 - Collecte des informations

Les moyens actuels de diffusion de l'information font que l'organisation se trouve face à une masse d'informations dont il ne faut retenir que celles jugées utiles. Ainsi donc, un travail d'épuration est à effectuer sur les données brutes qui arrivent à l'organisation.

2 - Analyse des informations

Ces informations doivent être 'analysées' pour juger de leur pertinence pour l'organisation et éviter une surcharge inutile et parfois parasite au système. Néanmoins, certaines informations peuvent être conservées en l'état car elles peuvent avoir un intérêt à plus ou moins long terme (salons, coupure de presse, film,...)

3 - Saisie des informations utiles

Les informations utiles pour l'organisation sont structurées afin de permettre une exploitation le plus souvent informatique. Cette phase d'exploitation est précédée d'une saisie (sur papier, sur support magnétique ou optique, ...) pour un traitement informatique ultérieur.

NB : la saisie est une opération délicate qui ne doit pas comporter d'erreurs car elle est à l'origine du traitement.

4 - Le traitement de l'information

Dès lors qu'une information est prise en compte par le SI, le choix consistant à la conserver en l'état, à la saisir sur un support ou à la rejeter constitue déjà un premier traitement. On parle de traitement lorsqu'il s'agit de :

- Procéder à des regroupements préalables
- Mettre à jour des données
- Effectuer des calculs
- Trier (classer)
- Stocker (sauvegarder, mémoriser)

5 - La diffusion de l'information

La diffusion de l'information ou mise à disposition de l'information est la finalité du SI mais elle pose un problème de pouvoir (qui décide de diffuser, vers qui ? et pourquoi ?). Pour être optimale, la diffusion de l'information doit répondre à 3 critères :

Le critère de forme : compte tenu de la rapidité de transmission désirée, du nombre de destinataire, on doit choisir la forme appropriée au moyen de diffusion le plus adéquat.

Exemple : Support papier ; Support électronique ; support oral ; ...

Le critère de temps : de la rapidité avec laquelle une information arrive à son destinataire peut dépendre l'efficacité et la pertinence d'une prise de décision importante.

Le critère de confidentialité : l'information traitée doit parvenir directement à son destinataire et être diffusée plus ou moins largement suivant son importance (bilan social, bulletin de paye, ...).

V - LES QUALITES D'UN SI

L'efficacité d'un SI dépend de 3 qualités : Sa **rapidité**, sa **fiabilité** et la **pertinence** des informations. A chacune des qualités, sont associées des notions informatiques. Ainsi on a **Rapidité** (réseau, microprocesseur etc...), **Fiabilité** (le SGBD : système de gestion des bases de données) et **Pertinence** (analyse informatique)

Remarque : *il n'est obligatoirement pertinent de vouloir tout automatisé. Le choix de l'automatisation dépend des facteurs tels que le volume à traiter, le temps de réponse, la taille de l'entreprise, le gain financier escompté.*

VI - LES DIFFERENTES ARCHITECTURES DE TRAITEMENT

A - ARCHITECTURE CENTRALISEE

L'information est traitée en un seul point de l'organisation, dans ce cas on évite les saisies multiples mais le système s'alourdit par le volume important des informations à traiter et ralentit le temps de réponse.

B - ARCHITECTURE DECENTRALISEE

Les moyens de traitement apparaissent à différents niveaux de l'organisation. Chaque poste de travail peut être équipé d'un poste travaillant avec ses propres données en autonome (réseau local).

C - ARCHITECTURE DISTRIBUEE

Le traitement s'effectue en un site centre alors que la saisie et la diffusion peuvent être décentralisées.

Exercices d'application

Déterminer les différents domaines de gestion des systèmes suivants :

- 1 – Gestion de stocks
- 2 - Gestion du personnel enseignant d'une grande école
- 3 - Gestion des inscriptions dans une grande école

J'écoute et j'oublie. Je lis et je retiens. Je fais et j'apprends. (Proverbe chinois)

INTRODUCTION

L'information est un élément qui permet de compléter notre connaissance sur un objet, un événement, un concept, etc....

Cette information peut se présenter sous diverses formes, les trois principales formes rencontrées en entreprise étant la forme écrites (fiches de papiers, documents écrits, imprimés, ...), la forme symbolique (images, dessins, masques,...) et la forme orale.

Pour une automatisation réussie, l'information doit être cernée et classifiée de manière très précise et doit pouvoir être représentable dans le système informatique. En gros, l'information doit être :

- Structurée
- Représentée
- Extraite d'un support.

I - STRUCTURE D'UNE INFORMATION

A - DEFINITIONS

1 - Information

Une information est un élément de connaissance susceptible d'être représentée à l'aide de convention pour être traitée, conservée ou communiquée.

L'information ou la donnée ou la propriété est l'atome du système d'information et du futur système informatique. L'information est un "renseignement" ou une "connaissance" élémentaire désignée à l'aide d'un mot ou d'un groupe de mots prenant des valeurs.

Exemple : nom de personne, nombre de portes, nom de maison, numéro matricule

Soit l'information *nom de personne*. ABE, LACONE, DIAKITE, SIKAHUE, YAO, sont des exemples de nom de personne.

Une information est donc un ensemble d'éléments dont LACONE est un exemple. Chaque élément de cet ensemble est appelé occurrence. *Une information est un ensemble d'occurrences, de valeurs possibles d'information.*

2 - La propriété

La propriété est une information élémentaire, c'est-à-dire non déductible d'autres informations et qui présente un intérêt pour le domaine étudié.

Exemple : Si on considère le domaine de gestion des commandes d'une société de vente, les données : « référence article », « désignation article », « prix unitaire HT », « taux de TVA », « Quantité commandée » sont des propriétés pertinentes pour ce domaine.

La donnée « Montant HT » n'est, d'après la définition, pas une propriété car ses valeurs peuvent être retrouvées à partir des propriétés «prix unitaire HT » et « Quantité commandée ».

On a : « Montant HT » = « Prix unitaire HT » * « Quantité commandée »

Chaque valeur prise par une propriété est appelée occurrence. *Des occurrences de la rubrique « désignation article » sont par exemple : « râteau », « cahier », « scie », ...*

Une propriété est dite simple ou encore atomique si chacune des valeurs qu'elle regroupe n'est pas décomposable.

La propriété « Adresse », dont des exemples d'occurrences sont donnés ci-dessous, n'est pas élémentaire car elle peut être décomposée en trois propriétés : le code postal, la boîte postale et la ville.

| Adresse |
|----------------------|
| 13 BP 2002 Abidjan13 |
| BP 468 Oumé |

La décomposition d'une propriété en propriétés plus simples ne doit pas être systématique et doit surtout tenir compte de son exploitation dans le système. Si cette exploitation est toujours globale, l'atomisation n'est pas nécessaire, dans les autres cas il faut procéder à l'isolement de chacune des composantes de la propriété et donc introduire de nouvelles propriétés.

| Propriété | Occurrences |
|-----------|--|
| Prénoms | Nyhombo Scéphora ; Laplubelle Rose, Brou Casal |

Une propriété paramètre est une propriété qui, à un instant donné, contient une seule valeur. *Un des exemples les plus classiques pour illustrer les paramètres est la rubrique « Montant Versé ».*

3 - Redondance

Une propriété est redondante si elle est dupliquée plusieurs fois


4 - La synonymie

Deux mots sont synonymes quand ils désignent la même chose (*deux mots sont synonymes s'ils ont la même signification*)

Exemples : Une automobile et une voiture
Un code et une référence

5 - La polysémie

Un mot est polysémique quand il désigne plusieurs choses différentes. Une information dont le nom a plusieurs sens, est un polysème

Exemples : une souris : 

Nom 

Montant peut être le montant de la facture, du contrat, de la commande

Remarque : Il importe d'éclater ces informations en informations n'ayant qu'un seul sens pour éviter toute confusion. Renseigner chaque information identifiée par un texte évitant tout quiproquo.

B - CARACTERISTIQUES D'UNE PROPRIETE

1 - Le type : *C'est l'ensemble dans lequel la propriété prend ses valeurs*

Exemples : *Alphanumérique, Numérique, Date, Logique (Booléen)*
(Lettres, chiffres) (Nombres), (date), (vrai, faux ; oui, non...)

Note : *Une propriété est de type numérique si elle peut être une opérande ou le résultat d'un calcul*

2 - La longueur : *Il s'agit du nombre de caractères utilisé pour représenter l'information.*

3 – La nature : *La nature d'une information est caractérisée par des catégories de données.*

- **Donnée concaténée :** *C'est une donnée qui peut être décomposable en d'autres données élémentaires.*

Exemples : *Adresse = Code postale + Boite postale + Ville*
Date = Jour + Mois + Année.

- **Donnée calculée :** *C'est une donnée obtenue directement à partir d'un calcul*

Exemple : *MHT = Quantité * PU*
Moyenne = (Note1 + Note2 + Note3) / 3

- **Donnée temporaire (ou donnée de situation, mouvement) :** *C'est une donnée dont la valeur varie avec le temps. Il s'agit souvent de données calculées.*

Exemple : *moyenne en informatique ; CA d'un vendeur, température, quantité commandée, nombre de passagers d'un car.*

- **Donnée signalétique (stable ou permanente) :** *C'est une donnée qui ne varie pas dans le temps, c'est une donnée descriptive d'une entité.*

Exemples : *date de naissance; nom; ...*

II - LE DICTIONNAIRE DE DONNEES

Le recensement des informations, le repérage de leur catégorie permet de constituer un document essentiel appelé dictionnaire des données (ou lexique des données).

A - DEFINITION

Le dictionnaire des données est un tableau permettant de recenser l'ensemble des informations rencontrées lors de l'analyse préalable ou permettant de répondre aux objectifs du SI.

Les différentes sources n'engendrent pas toujours explicitement la totalité des propriétés requises à recenser. Pour cela, il faut donc prendre en compte l'ensemble des activités à réaliser puis procéder à l'ajustement (ajout ou suppression) des données. Il est souhaitable de noter les contraintes sémantiques (ou contrainte d'intégrité) relatives aux données

Exemple : toute note de devoir est comprise entre 0et 20.

B - VALIDATION ET EPURATION DU DICTIONNAIRE

Des informations peuvent avoir le même sens, des sens différents ou être liées entre elles par composition ou une règle de calcul. Il importe d'identifier le sens de chaque information en supprimant les ambiguïtés de signification et les liaisons entre informations afin de manipuler un vocabulaire "épuré" qui sera mémorisé.

Dans le dictionnaire des données figurent toutes les propriétés, identifiées par un nom, qui présentent un intérêt pour le domaine à étudier. Ce nom doit être le plus explicite possible : à sa seule lecture on doit donc pouvoir se faire une idée de ce que représente la propriété. En outre, l'identification de chaque propriété consiste à garantir une bijection entre l'ensemble des noms et l'ensemble des propriétés à gérer. On devra donc **exclude** les synonymes qui correspondent à deux noms différents pour identifier la même propriété et les polysèmes qui représentent deux propriétés différentes ayant le même nom. Le principe de non-redondance impose que chaque propriété, correctement identifiée, n'apparaisse qu'une seule fois dans le modèle.

C - LE MODELE DE DICTIONNAIRE DE DONNEES

Le dictionnaire de données est un tableau contenant les propriétés des différentes informations avec leurs codes, signification, type, longueur, nature et observations.

Exemple

| N° ordre | Code | signification | type | longueur | observation |
|----------|--------|--------------------|------|----------|--------------------|
| 01 | MatEtu | Matricule étudiant | AN | 10 | identifiant |
| 02 | NomEtu | Nom étudiant | AN | 25 | |

Lexique : pour expliquer les abréviations : AN ; ...

N° ordre : permet le comptage des données

Code : codification par rapport au système informatique

Type : précise l'ensemble de valeurs de la donnée

Longueur : précise le nombre maximal de caractères à affecter à la donnée

Exercice d'application : cas Gestion des commandes

La société NEHEMIE & Frères achète et vend des consommables informatiques et fournitures de bureau. Pour l'automatisation de ses activités, il vous est demandé d'élaborer le dictionnaire des données.

Aussi rare que les étoiles du matin. (Expression chinoise)

INTRODUCTION

Pour optimiser le SI et les traitements qui suivront lors de la phase de développement de l'application, il va falloir regrouper les données ayant un rapport entre elles, en ensembles cohérents.

Aussi on constate intuitivement que certaines données ont un rapport entre elles et que la connaissance d'une de ces données permet d'avoir la connaissance d'une ou de plusieurs autres données.

Ainsi, la connaissance du matricule d'un étudiant permettra par exemple d'en connaître son nom, son prénom, sa date de naissance, son adresse.

Ces données sont dites alors liées par une dépendance fonctionnelle.

I - DEFINITIONS

- Etablir une dépendance fonctionnelle (DF), c'est le fait de relier de manière unique une propriété ou un ensemble de propriétés à d'autres propriétés.

- Une donnée *B* est en dépendance fonctionnelle (ou dépend fonctionnellement) d'une donnée *A* quand la connaissance d'une valeur de *A* permet de déterminer la connaissance d'au plus une valeur de la donnée *B*.

Exemple : la connaissance du matricule d'un étudiant permet de connaître sans ambiguïté le nom de l'étudiant, son prénom, son adresse, etc. Ainsi on peut dire que connaissant un matricule, on connaît au maximum un nom. Le matricule étudiant (*MatEtu*) est alors dit source de la DF et le nom étudiant (*NomEtu*) est dit BUT

II - LES DIFFERENTES DEPENDANCES FONCTIONNELLES

A - LES DF SIMPLES

Une propriété *B* dépend fonctionnellement d'une propriété *A* quand la valeur de la propriété *A* permet de déterminer au plus une valeur de la propriété *B*.

$A \longrightarrow B$ si $1 A_i \longrightarrow 1 B_j$ } **lecture**
 } A détermine B si la connaissance d'une valeur de A_i permet de
 } Déterminer une et une seule valeur de B_j

Exemple : $MatEtu \longrightarrow NomEtu$

B - LES DF COMPOSITES

Une DF est dite composite si elle comporte dans sa partie gauche (DF source) plusieurs propriétés (identifiants)

$A, B, C \longrightarrow D$ si $1 A_i, 1 B_j, 1 C_k \longrightarrow 1 D_n$

Exemple : $(RefPrdt, NumCde) \longrightarrow QteCde$

C - LES DF ELEMENTAIRES

$A \rightarrow B$ est une DF élémentaire Si et seulement Si :

A n'est pas un sous ensemble de B et si il n'existe pas une partie A' de A telle que $A' \rightarrow B$

Exemple : $REF. Produit \rightarrow Nom produit.$
 $(Code facture, REF produit) \rightarrow QTE facturée$

NB : $(CodeFact, RefPrdt) \rightarrow DesiPrdt$ n'est pas élémentaire car il existe une DF
 $RefPrdt \rightarrow RefPrdt.$

Ainsi donc une DF n'est pas élémentaire quand une partie de la source composée est elle même directement source d'une DF, vers le même but

D - LES DF DIRECTES (REFUS DE TRANSITIVITE)

DF $A \rightarrow B$ est directe s'il n'existe pas une propriété C qui engendrerait une DF transitive de la forme $A \rightarrow C$ s'il existe déjà une DF $B \rightarrow C$

Si $A \rightarrow B$ et $B \rightarrow C$ alors $A \rightarrow C$ ne doit pas exister

Exemple : $CodeClass \rightarrow MatEtu$ et $CodeFil \rightarrow CodeClass$

La DF $CodeFil \rightarrow MatEtu$ est superflue et inutile

NB : les DF à trouver dans le SI doivent être à la fois élémentaires et directes

III - LES PROPRIETES DES DF

Elles permettent, à partir des liens fonctionnels existant de générer de nouveaux liens.
 Soient A, B, C, D, 4 propriétés

A - LA REFLEXIVITE

$A \rightarrow A$ **Exemple :** $RefPrdt \rightarrow RefPrdt$

B - L'AUGMENTATION

$A \rightarrow B \rightarrow C$ Alors $A+C \rightarrow B$

Exemple $RefPrdt \rightarrow PU \rightarrow DesiPrdt$; alors $RefPrdt + DesiPrdt \rightarrow PU$

C - LA TRANSITIVITE

$A \rightarrow b$ et $b \rightarrow c$ alors $a \rightarrow c$

Exemple : $RefPrdt \rightarrow CodeTVA$
 $CodeTVA \rightarrow Taux TVA$ } Alors $RefPrdt \rightarrow TauxTVA$

Les 3 propriétés précédentes sont appelées **axiomes d'Armstrong** et ont pour corollaires les propriétés suivantes beaucoup plus utilisées en pratique.

L'ADDITIVITE (UNION)

$A \longrightarrow B$
 et $A \longrightarrow C$ } alors $A \longrightarrow B+C$

Exemple : $MatEtu \longrightarrow NomEtu$
 $MatEtu \longrightarrow PrénomEtu$ } Alors $MatEtu \longrightarrow NomEtu + PrénomEtu$

LA PSEUDO -TRANSITIVITE

$A \longrightarrow B$
 et $B + C \longrightarrow D$ } Alors $A + C \longrightarrow D$

Exemple : $RefPrdt \longrightarrow Code TVA$
 et $Code TVA + PU \longrightarrow TauxTVA$ } Alors $RefPrdt + PU \longrightarrow TauxTVA$

IV - MODE DE REPRESENTATION DES DF

Le formalisme utilisé pour représenter les DF peut être présenté sous l'une des 2 formes suivantes :

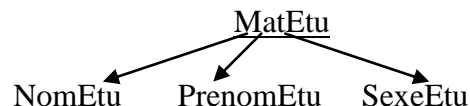
- Le graphe des DF
- La matrice des DF

A - LE GRAPHE DES DF

Il existe deux manières de représenter le graphe des DF 2^{ème} manière

1^{ère} manière

$MatEtu \longrightarrow NomEtu$
 $MatEtu \longrightarrow PrénomEtu$
 $MatEtu \longrightarrow SexeEtu$



B - LA MATRICE DES DF

La matrice des DF consiste à représenter les DF sous formes d'un tableau (matrice) on crée un tableau comportant autant de ligne et de colonnes que de données on met un 1 dans la case intersection des rubriques concernées

| | | a | b | c | d |
|-----------|---|---|---|---|---|
| Matricule | a | 1 | 1 | 1 | 1 |
| Nom | b | | 1 | | |
| Prénom | c | | | 1 | |
| Sexe | d | | | | 1 |

V – ELABORATION DU GRAPHE DES DF

DEMARCHE PRATIQUE D'ELABORATION DU GRAPHE DES DF

- Extraire du dictionnaire de données la liste des propriétés qui ne sont ni calculées ni concaténées
- Recenser les identifiants. A partir de chaque identifiant rechercher les DF simples et les DF composites
 - Placer les propriétés isolées
 - Chercher les DF entre les propriétés isolées. La propriété isolée du graphe est obtenue par concaténation des identifiants indépendants
 - Pour les propriétés isolées ne participant à aucune DF introduire de nouveaux identifiants dans le graphe puis établir les DF avec les propriétés isolées
 - *Sur le graphe obtenu supprimer tous les cycles et toutes les transitivités.*

Le graphe final obtenu est appelé **Structure d'Accès Théorique (SAT)** ou **couverture minimale**.

Définition : La SAT représente tous les chemins d'accès aux données possibles.

Exercice d'application : cas Gestion des commandes

La société NEHEMIE & Frères achète et vend des consommables informatiques et fournitures de bureau. Pour l'automatisation de ses activités, à partir du dictionnaire des données, il vous est demandé d'élaborer la SAT

L'accident de la charrette de devant doit servir à celle qui la suit. (Proverbe chinois)

INTRODUCTION

La modélisation conceptuelle des données du SI d'une organisation se fait au moyen de concepts dont la mise en œuvre aboutit à une représentation appelée Modèle Conceptuel des Données (MCD) (ou Schéma Conceptuel des Données).

Le MCD donne donc une représentation statique du SI de l'entreprise, c'est à dire l'ensemble des propriétés manipulées par l'entreprise ainsi que les relations entre ces différentes propriétés. En effet, ce qui est mis en évidence à ce niveau est la sémantique du système, car chaque mot du langage de l'entreprise a une signification précise et le MCD décrit les rapports qui existent entre les sens de ces mots (concepts).

Une représentation des données du SI doit avoir deux qualités : être complète et être simple.

- La complétude est atteinte si on représente le plus fidèlement possible le réel perçu.

- La simplicité est obtenue en fournissant une représentation permettant le dialogue entre utilisateurs et concepteurs.

Le formalisme adopté par la méthode MERISE pour réaliser cette description est basé sur les concepts « entité - association ». La formalisation des propriétés est le point majeur de la méthode MERISE. Il est important de bien saisir ce chapitre si vous voulez passer pour un Merisien confirmé.

Les informations sont structurées et classées, sans répétition, en deux types d'ensembles (d'informations), les individus et les relations.

I - LES CONCEPTS

A - L'ENTITE

1 - Définitions

Définition 1. : *L'entité (regroupement de propriétés) est un objet (ou individu) du SI pourvu d'une existence propre, conforme au choix de gestion de l'entreprise et porteur de propriétés.*

Définition 2 : *Une entité est la représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire.*

Définition 3 : *Une entité est un regroupement bien pensé, donc sensé, de plusieurs propriétés.*

Exemples : *Un individu est un ensemble d'informations qui a une existence propre.*

L'individu PERSONNE comprend les informations, MatPers, NomPers, PrenomPers, DateNaisPers

L'entité PRODUIT regroupe les propriétés : RefPrdt, DesiPrdt, PUPrdt

2 - Caractéristiques de l'entité

Une entité (ou un individu) est un objet répondant aux conditions suivantes :

1 - être d'intérêt pour l'organisation

2 - être distinguable parmi les autres individus de son type, c'est-à-dire qu'il existera un identifiant par individu (numéro, code, référence...)

3 - avoir une existence propre, "c'est-à-dire une existence concevable sans hypothèse sur l'existence d'autres éléments du réel perçu".

4 - être doté d'un ensemble unique de propriétés. Toutes les informations ont une seule valeur, une seule occurrence pour une occurrence d'individu.

Contre exemple : Tous les pères n'ont pas forcément de 2^{ème} fils

Remarque : un "père" peut ne pas avoir de puîné (2^{ème} fils).

Ceci n'est une entité que pour les pères de 3 fils et 3 fils seulement

| Père |
|--|
| Nom père Prénom père Prénom 1 ^{er} fils Prénom 2 ^{ème} fils Prénom 3 ^{ème} fils |

Définition : On appelle classe d'entité un ensemble composé d'entités de même type, c'est-à-dire dont la définition est la même.

3 - Propriétés de l'entité

Dans notre exemple précédent, NomPers est l'une des propriétés de l'entité PERSONNEL. La propriété est normalement valorisée de manière unique pour chaque occurrence de l'entité.

1 - Une information est dans un seul individu (ou une seule relation).

2 - Une seule occurrence d'information par individu. La valeur d'une information est unique pour un individu (nommé aussi ensemble unique de propriétés d'un individu)

"Prénoms fils" peut exister plusieurs fois

Remarque : "prénoms fils" n'est donc pas une propriété unique pour L'entité "Père"

| Père |
|--|
| Nom père Prénoms père Prénoms fils |

Exemple : pour une école les propriétés suivantes de l'entité ETUDIANT présentent un intérêt pour l'école, existent matériellement (existence propre), sont identifiables, et peuvent être gérées : MatEtu, NomEtu, PrenomEtu, NationEtu, DateNaisEtu, AdresEtu permettent de représenter une même classe d'éléments : LES ETUDIANTS.

4 - L'identifiant de l'entité

Parmi les propriétés que porte l'entité il en existe une qui revêt une importance toute particulière car elle permet, par le biais de sa valorisation, d'identifier de manière unique et non ambiguë une seule occurrence de l'entité, c'est l'IDENTIFIANT.

Définition 1 : Un identifiant est un ensemble de propriétés (une ou plusieurs) permettant de désigner une et une seule entité.

Définition 2 : L'identifiant est une propriété particulière d'un objet telle qu'il n'existe pas deux occurrences de cet objet pour lesquelles cette propriété pourrait prendre une même valeur.

Les attributs d'une entité permettant de désigner de façon unique chaque instance de cette entité sont appelés **identifiant absolu**.

Le modèle conceptuel des données propose de **souligner en trait continu les identifiants** (parfois de les faire précéder d'un #). Ainsi, chaque entité doit posséder au moins un attribut identifiant, et l'ensemble de ses attributs identifiants doivent être renseignés à la création de l'entité.

Formalisme de l'entité

| Nom de l'entité |
|--------------------|
| <u>Identifiant</u> |
| Propriété 2 |
| Propriété 3 |
| Propriété 4 |
| ... |

Exemple

| ETUDIANT |
|---------------|
| <u>MatEtu</u> |
| NomEtu |
| PrenomEtu |
| NationEtu |
| DateNaisEtu |

Le champ du haut contient le libellé (le nom de l'entité). Ce libellé peut être une abréviation pour une raison de simplification de l'écriture. Il s'agit par contre de vérifier qu'à chaque entité correspond un et un seul libellé, et réciproquement. Le champ du bas contient la liste des propriétés de la classe d'entité.

5 - L'occurrence de l'entité

Définition : L'occurrence de l'entité correspond au nombre de fois où cette entité est valorisée.

D'après la définition d'une entité, on sait que la connaissance d'une valeur de la rubrique identifiante détermine la connaissance des valeurs des autres rubriques de l'entité. L'ensemble de ces valeurs est appelé occurrence d'entité.

Exemple : Entité ETUDIANT avec quelques une de ses occurrences :

| ETUDIANT |
|---------------|
| <u>MatEtu</u> |
| NomEtu |
| PrenomEtu |
| NationEtu |
| AdresEtu |
| DateNaisEtu |

| ETUDIANT |
|-------------------|
| <u>IG001/2010</u> |
| YAO |
| Casal |
| Ivoirienne |
| 13 BP 201 Abj |
| 24/01/1992 |

| ETUDIANT |
|-------------------|
| <u>FC094/2010</u> |
| DORO |
| Codjo Alfred |
| Beninoise |
| 01BP1080 Abj |
| 17/08/1990 |

B - L'ASSOCIATION

Contre-exemple : "prénom fils" peut exister plusieurs fois.

MAUVAIS : le prénom du fils n'est pas une propriété unique pour l'entité Père.

| Père |
|--------------|
| Nom père |
| Prénoms père |
| Prénoms fils |

| Fils |
|--------------|
| Prénoms fils |

| Père |
|--------------|
| Nom père |
| Prénoms père |
| Prénoms fils |

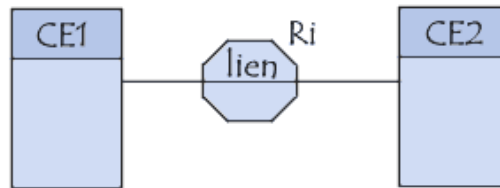
Deux individus "PERE ET FILS" existent

BON : un père peut avoir un ou plusieurs fils.

Il faut pouvoir exprimer le fait que le père est lié au fils. C'est le rôle de la relation.

1 - Définition

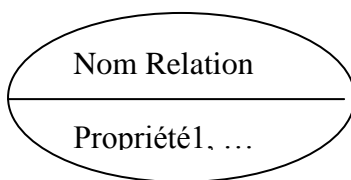
Une association (appelée aussi parfois relation) est un lien sémantique entre plusieurs entités. Une classe de relation contient donc toutes les relations de même type (qui relient donc des entités appartenant à des mêmes classes d'entité).



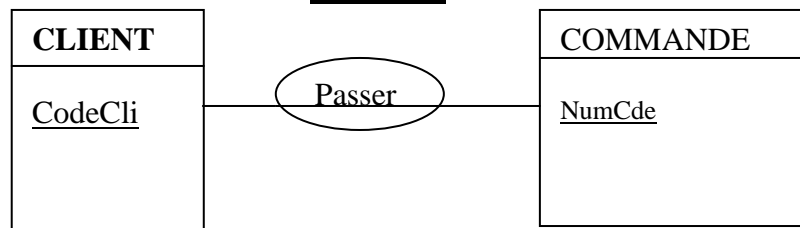
La relation est dépourvue d'existence propre mais elle peut être porteuse de propriétés. Elle n'existe que par les entités qui s'y rapportent. La relation est très souvent traduite par une action ou un verbe d'action.

Une relation n'a pas d'identifiant explicite : l'identifiant ici est créé en concaténant les identifiants des entités reliées.

Formalisme de la relation



Exemple



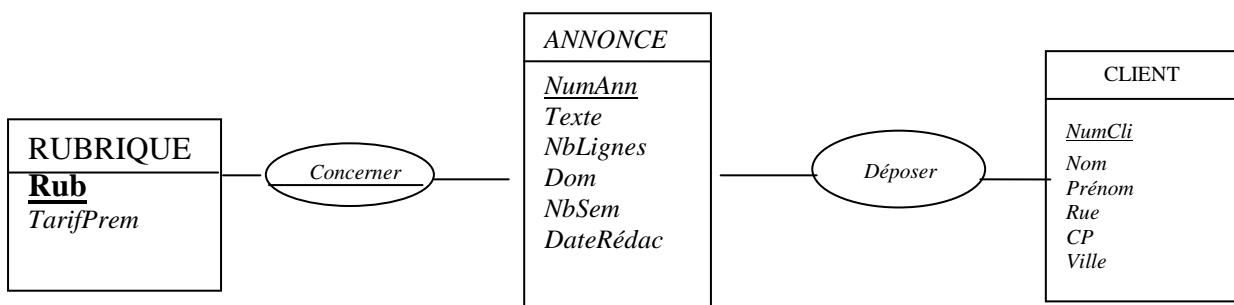
Les relations sont représentées par des hexagones (ou des ellipses) dont l'intitulé décrit le type de relation qui relie les d'entités (généralement un verbe d'action).

Définition : on appelle dimension d'une association, le nombre d'entités auquel est reliée la relation

Une relation peut lier plus de deux entités. Voici les dénominations des classes de relation selon le nombre d'intervenants:

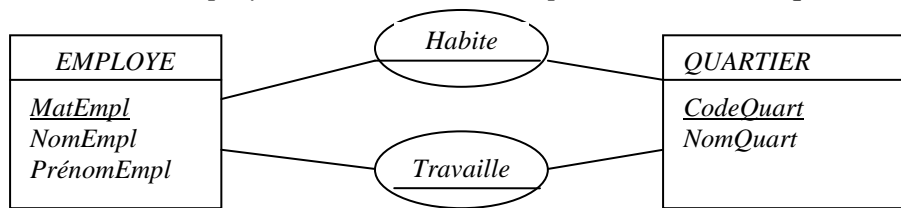
- une relation **récursive** (ou *réflexive*) relie la même d'entité
- une relation **binaire** relie deux d'entités
- une relation **ternaire** relie trois d'entités
- Une relation **n-aire** relie n entités

Par exemple, dans le modèle conceptuel des données relatif à la gestion des annonces on introduit les deux associations « Déposer » et « Concerner » afin d'exprimer les réalités suivantes : un client dépose une annonce et une annonce concerne une rubrique.



Il peut y avoir, entre deux mêmes entités, plusieurs associations qui représentent chacune des réalités différentes.

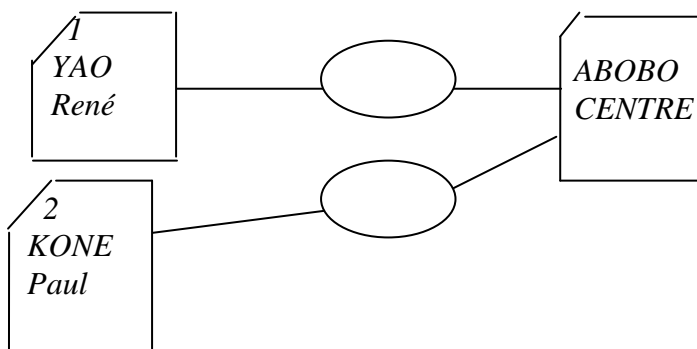
Dans le modèle conceptuel des données ci-dessous, l'association *Habite* indique le département dans lequel l'employé habite et l'association *Travaille* fournit le Quartier dans lequel il travaille. Pour certains employés ces deux Quartiers peuvent être identiques.



L'ensemble des entités participant à une association est appelé collection de cette association. La collection de l'association *HABITE* est formée des entités : *EMPLOYE* et *QUARTIER*.

2 - Occurrence d'association

Une occurrence d'association est un lien particulier qui relie deux occurrences d'entités. Le schéma ci-dessous présente deux exemples d'occurrences de l'association « *Habite* ».



C - CARDINALITE

1 - Définition

La cardinalité représente le nombre d'occurrences minimal et maximal d'une entité par rapport à une relation.

Les cardinalités permettent de caractériser le lien qui existe entre une entité et la relation à laquelle elle est reliée. La cardinalité d'une relation est composé d'un couple comportant une borne maximale et une borne minimale, intervalle dans lequel la cardinalité d'une entité peut prendre sa valeur:

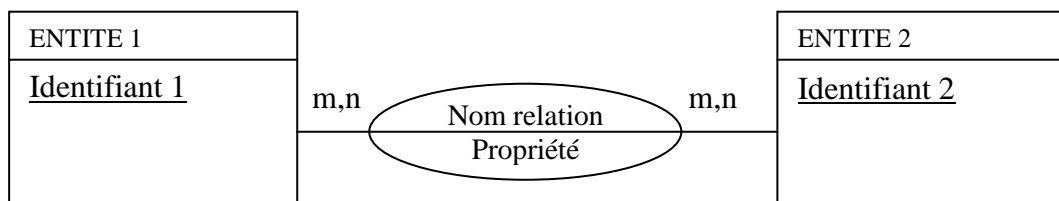
- la borne minimale (généralement 0 ou 1) décrit le nombre minimum de fois qu'une entité peut participer à une relation
- la borne maximale (généralement 1 ou n) décrit le nombre maximum de fois qu'une entité peut participer à une relation

Un couple de cardinalités placé entre une entité E et une association A représente le nombre minimal et maximal d'occurrences de l'association A qui peuvent être « ancrées » à une occurrence de l'entité E. Le tableau ci-après récapitule les valeurs que peut prendre ce couple.

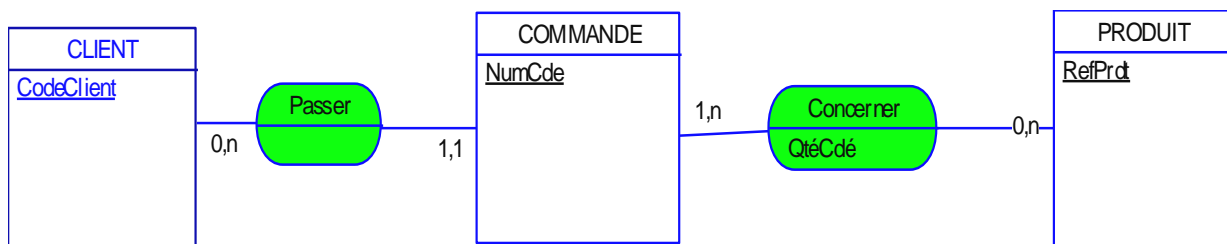
| | |
|--|--|
| | <p>Pour chaque occurrence de E, le modèle admet :</p> <ul style="list-style-type: none"> - soit l'absence de lien - soit la présence d'un seul lien |
| | <p>Pour chaque occurrence de E le modèle admet la présence d'un et un seul lien</p> |
| | <p>Pour chaque occurrence de E le modèle admet la présence d'un seul ou de plusieurs liens</p> |
| | <p>Pour chaque occurrence de E le modèle admet :</p> <ul style="list-style-type: none"> - soit l'absence de lien - soit la présence de plusieurs liens |

Remarque : dans certaines situations, la lettre *n* peut être remplacée par une valeur.

2 - Formalisme



Exemple



Lecture des cardinalités

Association « Passer »

- Un client peut passer 0 ou *n* (plusieurs) commandes
- Une commande est passée par un et un seul client

Association « Concerner »

- Pour une quantité donnée, une commande est concernée par un ou plusieurs produits
- Pour une quantité donnée, un produit peut être concerné par 0 ou plusieurs commandes

D - LA CONTRAINTE D'INTEGRITE FONCTIONNELLE (CIF)

1 - Définition

Quand on détermine, entre une association et une entité, une cardinalité présentant les valeurs (0,1) ou (1,1) l'association est particulière et est appelée contrainte d'intégrité fonctionnelle (ou CIF)

NB : - en principe les CIF sont nommées **CIF**.

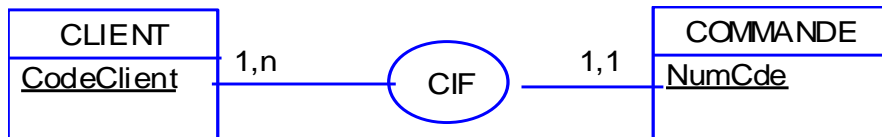
- La **CIF** n'est pas porteuse de propriétés
- Une CIF indique que l'une des entités est totalement déterminée par la connaissance de l'autre

2 - Identification des CIF

0,1 \longrightarrow 0, n ; 0,1 \longrightarrow 1, n

1,1 \longrightarrow 0, n ; 1,1 \longrightarrow 1, n

Exemple



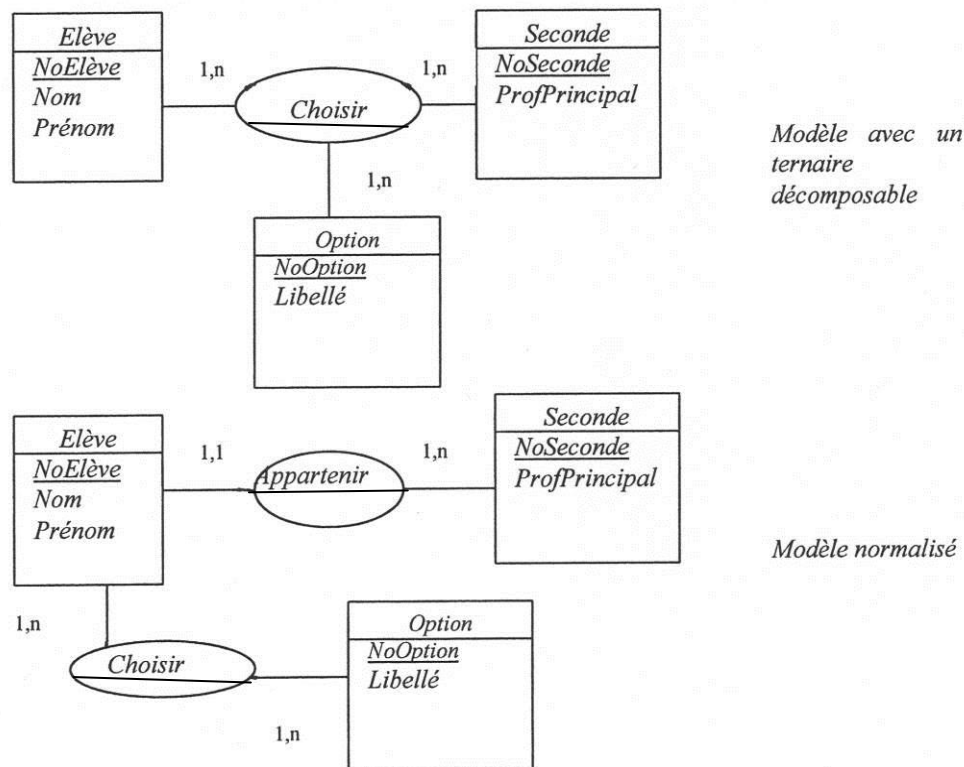
Les CIF sont aussi appelées associations creuses.

E - CARACTERISTIQUES D'UNE ASSOCIATION

Remarques :

- Toute occurrence d'une association de dimension n doit être reliée à n occurrences d'entités. Par exemple, pour une association ternaire dans laquelle participent trois entités « A », « B » et « C », toute occurrence doit être reliée à 3 occurrences des entités respectives A, B et C.

- L'opération de décomposition consiste à éclater une association de dimension n en une ou plusieurs associations de dimension moindre sans perte de sémantique. Dans l'exemple ci-dessous, qui se rapporte à la gestion des options en classe de seconde, l'association ternaire du premier modèle doit être éclatée en deux associations binaires afin d'aboutir au second modèle.



Les associations figurant dans le modèle conceptuel des données devront être non décomposables. Pour ce faire on recherchera en priorité les associations binaires puis les ternaires.

➤ La fonctionnalité d'une association binaire

Les différents types de fonctionnalités d'une association binaire A définie entre deux entités E1 et E2 sont les suivants :

- un à un (**1 – 1**) caractérisé par la cardinalité **maximum** égale à 1 sur les deux segments « E1 -- A » et « E2 – A »,
- un à plusieurs (**1 – n**) caractérisé par la cardinalité **maximum** égale à 1 sur l'un des segments et la cardinalité maximum égale à n sur l'autre,
- plusieurs à plusieurs (**m-n**) caractérisé par la cardinalité **maximum** égale à n sur les deux segments « E1 – A » et « E2 – A ».

➤ Les notions de partialité et de totalité

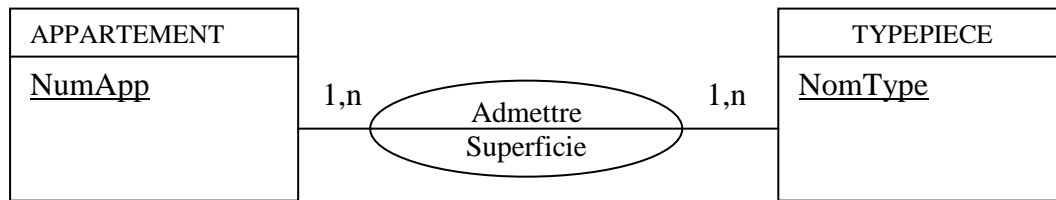
Une association binaire A définie entre les entités E1 et E2 est totale si elle est caractérisée par la cardinalité minimale égale à un sur les deux segments « E1 – A » et « E2 – A ».

Une association binaire A définie entre les entités E1 et E2 est **partielle** si elle est caractérisée par la cardinalité minimum égale à 0 sur l'un des segments.

➤ L'association porteuse

Les propriétés qui dépendent fonctionnellement de plusieurs identifiants d'entités sont portées par les associations entre ces entités. C'est une dépendance fonctionnelle multi-attributs au niveau de la source.

Dans le système d'information suivant la propriété *Superficie* est portée par l'association qui relie l'entité *APPARTEMENT* à l'entité *TYPEPIECE* car pour un appartement et une pièce on a une et une seule superficie.



Le droit d'entrée d'une propriété P dans une association reliant n entités est donc soumis à l'existence de la dépendance fonctionnelle suivante : $I_1, \dots, I_n \rightarrow R$ où I_1, I_2, \dots représentent l'identifiant de chacune des entités qui participent à l'association.

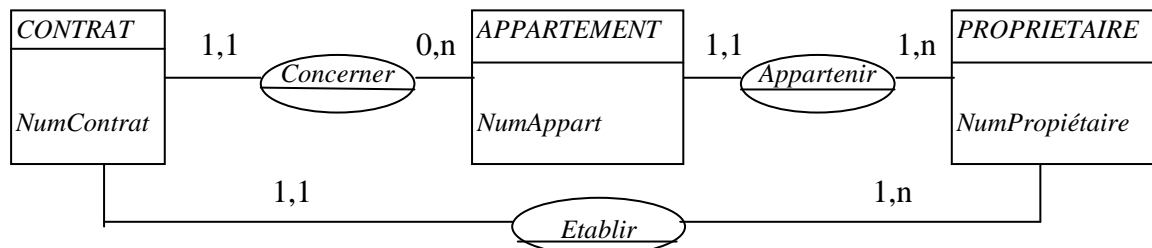
Pour éviter toute redondance, on s'assurera en outre que la dépendance fonctionnelle est élémentaire. **Les associations porteuses sont donc toujours de type (m-n).** On dit qu'une propriété est en dépendance fonctionnelle élémentaire avec une liste de rubriques LR :

- si elle est fonctionnellement dépendante de LR,
- si elle n'est pas fonctionnellement dépendante d'une sous-liste de LR.

La dépendance fonctionnelle suivante : $NoAppart, NomType, NumPropriétaire \rightarrow Superficie$ n'est pas élémentaire car il existe la sous-liste $NumAppart, NomType$ telle que : $NumAppart, NomType \rightarrow Superficie$.

➤ Les associations transitives

Considérons le modèle suivant :



L'association binaire qui relie l'entité « CONTRAT » et l'entité « PROPRIETAIRE » doit être ôtée du modèle car on peut retrouver le propriétaire à partir des associations « Concerner » et « Appartenir ». Il s'agit d'une association transitive. Seules les dépendances fonctionnelles directes entre identifiants d'entités devront donner lieu à des associations binaires de type (1-n).

On veillera à supprimer les dépendances transitives pour ne pas surcharger le MCD car on risque de faire des incohérences.

➤ Les associations réflexives

L'association réflexive est une association binaire qui relie une entité à elle-même. Une occurrence de l'association établit donc un lien entre une occurrence de l'entité et une autre occurrence de cette même entité.

II - LES FORMES NORMALES DES RELATIONS

Pour qu'un SI implanté au moyen d'une base de données relationnelles soit optimisé, il est nécessaire que les relations la composant soient en troisième forme normale de BOYCE_ CODD (du nom de deux chercheurs ayant travaillé sur la théorie mathématique des relations).

A - LA PREMIERE FORME NORMALE (1 FN)

Dans un MCD, chaque entité ou chaque association doit posséder un identifiant et les autres propriétés de l'entité doivent être élémentaires.

Une relation est en 1 FN si les propriétés la composant sont en dépendance fonctionnelle de l'identifiant de cette entité ou cette association.

B - LA DEUXIEME FORME NORMALE

Toute propriété d'une entité ou d'une association doit dépendre de l'identifiant par une dépendance fonctionnelle élémentaire.

C - LA TROISIEME FORME NORMALE

Dans une entité toute propriété doit dépendre de l'identifiant par une DF élémentaire directe

III - REGLES DE VALIDATION DU MCD

R1 : dans un MCD, toute propriété ne doit qualifier qu'une seule entité ou une seule association.

R2 : chaque entité possède un et un seul identifiant.

R3 : dans un MCD, chaque propriété d'une occurrence d'entité ne possède au plus qu'une valeur.

R4 : dans un MCD, toutes les propriétés doivent être élémentaires

R5 : dans un MCD, toute propriété autre que l'identifiant doit dépendre pleinement et directement de l'identifiant.

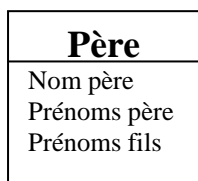
R6 : dans un MCD, à chaque occurrence d'une association correspond une et une seule occurrence de chaque entité participant à l'association.

R7 : dans un MCD, pour une occurrence d'une association il ne doit exister au plus qu'une valeur pour chaque propriété de cette association.

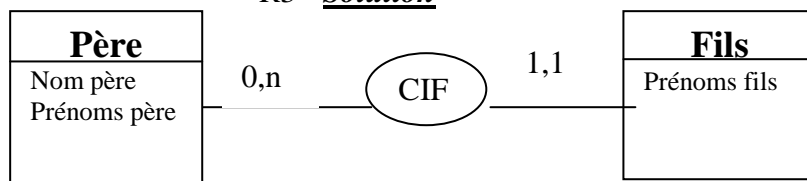
R8 : chaque propriété d'une association doit dépendre pleinement et directement de tout l'identifiant et non pas d'une partie de cet identifiant.

Exemples :

R3- Contre - exemple

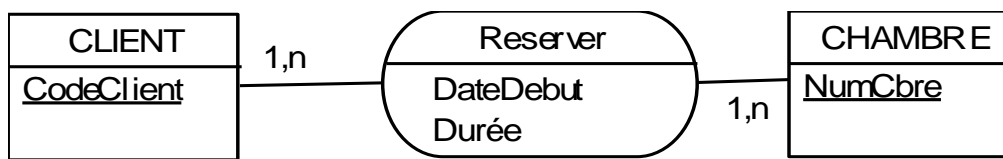


R3 - Solution

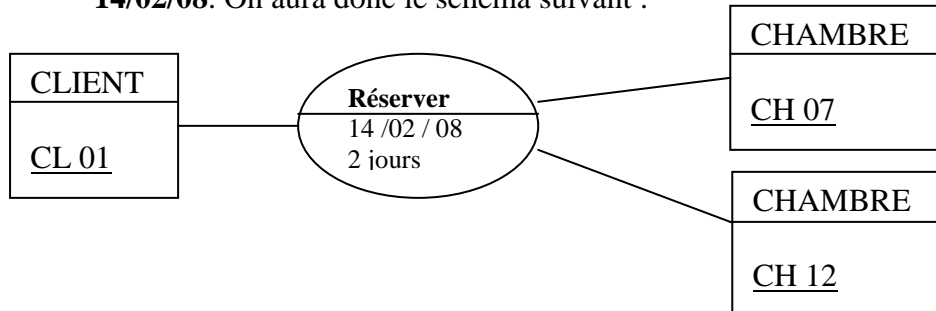


Remarque : Le prénom, en fait, peut prendre plusieurs valeurs si l'employé a plusieurs enfants

Soit le MCD suivant modélisant les réservations de chambres d'un hôtel

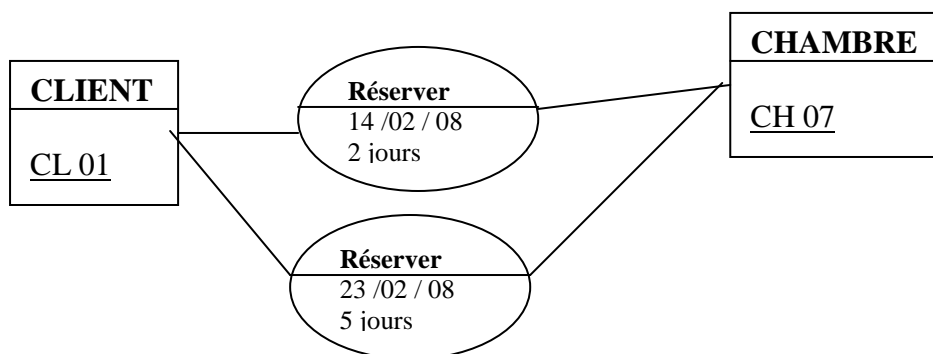


- Supposons que le client **CL 01** souhaite réserver les chambres **CH 07** et **CH 12**, au **14/02/08**. On aura donc le schéma suivant :



(Schéma à éviter car ne respectant pas la règle R6).

- Supposons que le client **CL 01** souhaite réserver les chambres **CH 07** le **14/02/08** pour **2 jours** et le **23/02/08** pour **5 jours**. On aura donc le schéma suivant :

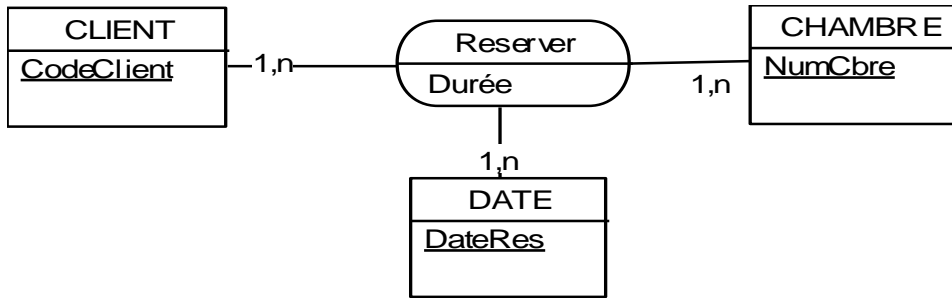


(Schéma à éviter car ne respectant pas la règle R7)

Dans ce cas, on aura plusieurs valeurs identiques d'identifiants « **CL 01 ; CH 07** », ce qui est interdit pour l'identifiant.

Un tel MCD implique donc qu'un client ne puisse faire plusieurs réservations à des dates différentes ou réserver plusieurs chambres à la même date.

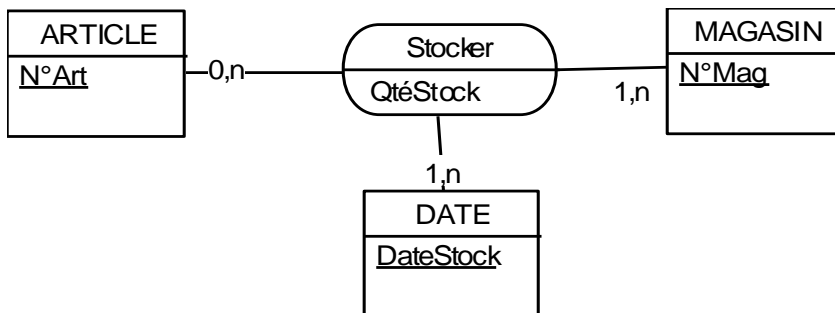
Ainsi le MCD correct est le suivant :



Exercice : A une date donnée, une quantité d'un produit est stocké dans un magasin.
TAF : établir le diagramme des occurrences puis construire le MCD.

Diagramme d'occurrences

| N° article | N° magazine | Quantité | Date |
|------------|-------------|----------|----------|
| A125 | M03 | 18 | 22/02/08 |
| A125 | M03 | 47 | 12/03/08 |
| A098 | M01 | 158 | 12/03/08 |



IV - REGLES DE CONSTRUCTION D'UN M.C.D.

MODELISATION PAR ANALYSE DES DF

Détermination des rubriques « sources »

- On regroupe les rubriques sources des DF qui ont au moins une des caractéristiques suivantes :

- Etre source de DF simple
- Etre membre d'une source de DF composite
- Etre but de plusieurs DF (sans être source de DF)
- Etre une rubrique isolée

Détermination des entités

Chaque rubrique déterminée ci-dessus va générer une entité dont elle sera l'identifiant.

Détermination des CIF

Une DF dont la partie gauche non composée est source et la partie droite est une source indique une association binaire à cardinalité (0,1) ou (1,1) (CIF binaire). (La cardinalité (0,1) ou (1,1) sera placée du côté de l'entité où se trouve la rubrique source de la DF. l'autre cardinalité sera (0, n) ou (1, n))

Détermination des autres associations

Une DF à partie gauche composée détermine une association, quand elle a pour but une rubrique (ou plus) non recensée comme source. Elle va alors générer une association entre les entités qui correspondent aux rubriques constituant la partie gauche en question. (Les cardinalités sont généralement égales à (0, n) ou (1, n).

Détermination des CIF multiples

Chaque DF composite dont le but fait partie de l'ensemble des sources , va générer une CIF multiple , qui part des entités correspondants aux rubriques qui constituent la partie gauche de la DF composite et pointant vers l'entité correspondant au but

Ventilations des rubriques dans les entités ou associations

Il s'agit ici de préciser les propriétés des entités ou associations.

Aménagement au MCD

- Certaines particularités peuvent être rencontrées au niveau du MCD.

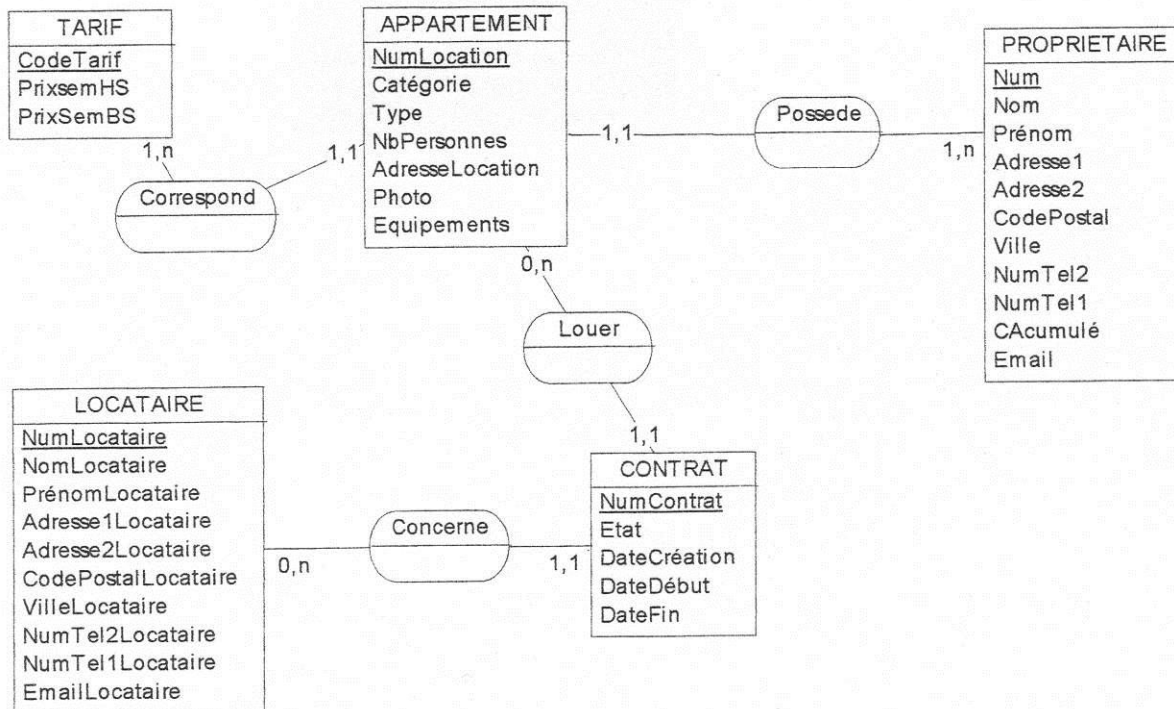
Il se trouve parfois que certaines entités (voir une simple rubrique) ne soient but d'aucune DF, mais indispensable au SI. Dans ce cas, il est nécessaire de créer une association.

- Il possible dans certains cas, de transformer certaines associations en entités. (de telles associations sont toujours creuses et leurs cardinalités sont du type (0, n) ou (1, n)).

Exercices d'application

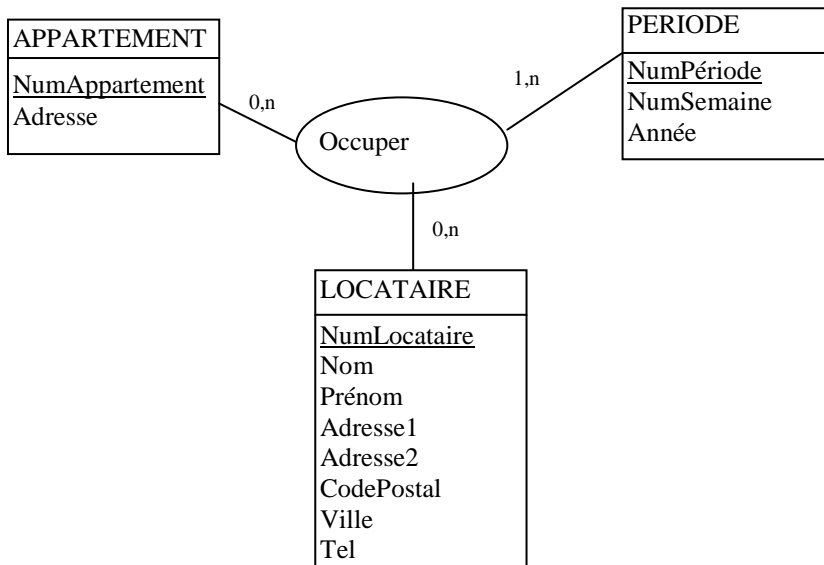
Exo1 : A partir du MCD suivant

- 1 – Elaborer les règles de gestion
- 2 – Dédurre la SAT



Exo2 : En utilisant le MCD précédent et celui qui suit

- 1 – Elaborer les règles de gestion pour un MCD de synthèse
- 2 – Dédire la SAT à partir du MCD de synthèse



Exo3 : cas Gestion des commandes

La société NEHEMIE & Frères achète et vend des consommables informatiques et fournitures de bureau. Pour l'automatisation de ses activités, à partir de la SAT, il vous est demandé d'élaborer le MCD

Changez de chanson quand vous changez de montagne. (Proverbe chinois)

INTRODUCTION

Comme il a été dit dans le chapitre d'introduction, tout système d'information est composé d'une base d'information et d'un processeur d'information qui représente respectivement sa statique et sa dynamique. A l'instar du Modèle Conceptuel des Données (MCD) qui schématise les données du système d'information, le Modèle Conceptuel des Traitements (MCT) décrit les traitements et plus précisément toutes les activités découlant des échanges entre le domaine étudié et le monde extérieur. Il exprime donc ce que fait le domaine sans se poser le problème de savoir qui le fait, quand et comment.

Les traitements décrivent donc les actions à exécuter afin d'obtenir les résultats attendus par l'entreprises. La prise en compte de l'aspect dynamique des règles de gestion se fait à l'aide du modèle basé sur les concepts d'événement, d'opération, de synchronisation, et de processus. Le MCT est mis en œuvre à partir de l'identification des flux d'informations échangés par le domaine étudié avec l'environnement extérieur, et de l'ordre dans lequel ces flux apparaissent.

Le Modèle Conceptuel de Traitement reflète le QUOI de l'activité de l'entreprise et le comment spécifique, les règles de gestion ou les choix de traitement. A ce niveau, la stratégie de l'entreprise n'est pas abordée, mais la *cohérence* de l'ensemble, la *réponse aux différents cas* de situation est étudiée en détail.

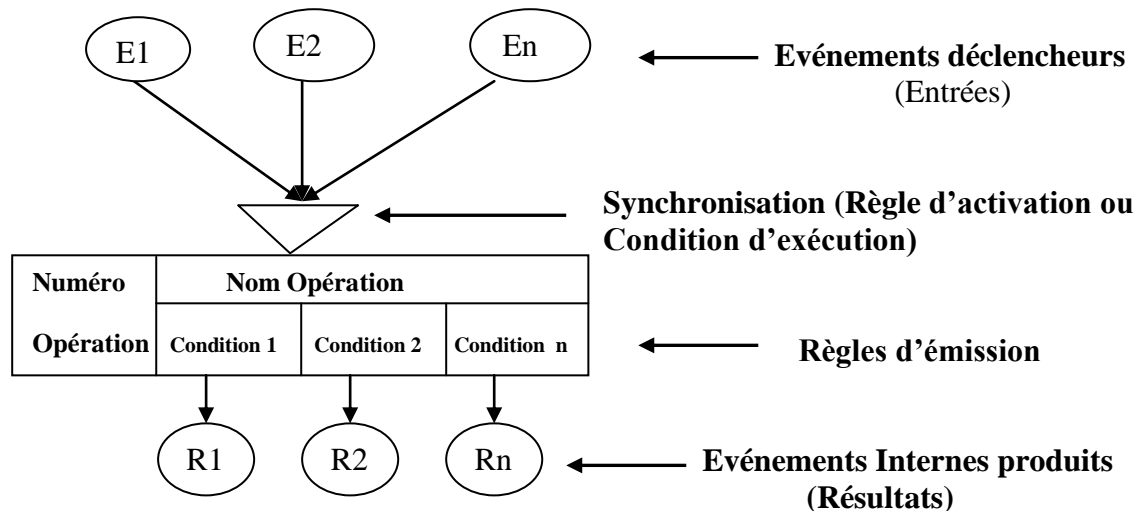
Le MCT vise les objectifs suivants :

- Définir clairement ce qu'on veut faire
- Décrire les actions menées par l'entreprise en réponse aux flux qu'elle reçoit de l'extérieur et qui sont sa raison d'être.
- Formaliser les règles de gestions de l'organisme.

I - LES CONCEPTS

Le MCT exprime ce qu'il faut faire, mais n'indique pas qui doit faire, ni quand il faut le faire, ni où il faut le faire (concept organisationnels). Ni comment il faut le faire (concepts opérationnels). Le MCT est constitué d'un enchaînement d'opérations. Chaque opération est déclenchée soit par un événement unique, soit par plusieurs événements liés par une condition de synchronisation. L'opération exécute alors les traitements et produit un ou plusieurs résultats.

A - FORMALISME DE BASE



B - L'ÉVÉNEMENT

Définition 1 : Un événement est susceptible de déclencher une opération soit seul, soit en synchronisation avec d'autres événements

Définition 2 : Un événement représente un fait nouveau. Il correspond à une sollicitation du système d'information.

L'événement matérialise un fait, qui en se produisant, doit déclencher une réaction du système.

Un événement est extérieur s'il provient de l'univers extérieur et provoque une réaction du SI sous la forme d'une opération.

Un événement est interne s'il apparaît à la suite d'une opération. Il peut soit provoquer une nouvelle réaction du SI, soit constituer un résultat pour l'univers extérieur.

L'événement (au sens du modèle conceptuel des traitements) est porteur d'informations qui doivent être obligatoirement digérées par le système d'information sans quoi il ne répondrait pas à ses objectifs.

L'événement- temps

Le temps figure dans le MCT par les synchronisations qui expriment une attente. Il y a toutefois des attentes qui sont liées non pas à des événements, mais à des contraintes de temps (dates, délais) issues des règles de gestion. Il est alors nécessaires d'assimiler le temps (date, délai) à un événement extérieur participant au déclenchement des opérateurs.

Exemple : pour déclencher un traitement en début d'année civile, on introduira l'événement « Début d'année ». L'événement « Date actuelle est JJ/MM/AAAA » permettra d'exécuter un traitement à une date donnée.

Chaque événement est identifié au moyen d'un libellé générique tel que

- Réception d'un dossier de pré inscription ».
- L'arrivée d'une commande client
- Annonce d'une panne
- Arrivée d'un client.

C - L'OPÉRATION

Définition : *L'opération est un traitement, un travail effectué par un acteur interne afin d'émettre un message et/ou d'en traiter un dans un but précis.*

Seules les opérations effectuées par l'entreprise (les acteurs internes) sont analysées.

La réponse à l'arrivée d'un événement est le déclenchement d'un ensemble de traitements appelé opération.

Exemple : *Le traitement d'enregistrement d'une inscription est une opération déclenchée lors du dépôt de dossier d'inscription.*

Lors de son exécution une opération ne peut pas être interrompue par l'attente d'un événement externe. L'exécution d'une opération se ramène à l'exécution d'actions élémentaires effectuées sur la base d'informations à partir des données portées par le ou les événement(s) déclencheur(s). Ces actions élémentaires portent sur des occurrences d'entités ou d'associations du modèle conceptuel des données.

Remarques :

1 - *La logique d'enchaînement des actions élémentaires n'est pas toujours séquentielle et peut faire intervenir des structures alternatives (Si. Alors ... Sinon) ou itératives (Tant que, Répéter, Pour ...).*

2 - *L'opération peut être différente suivant chaque utilisateur. Une fois le choix de gestion fait par l'utilisateur, l'opération décrit fidèlement son choix.*

Exemples de choix de traitement à faire par l'utilisateur :

- *La facture est émise dès réception de la commande.*
- *La facture est émise dès la livraison.*
- *La facture est émise dès réception d'un document particulier : douane, plan d'implantation particulier, crédit documentaire...*

D - LA SYNCHRONISATION

L'exécution d'une opération est toujours conditionnée par un ou plusieurs événements. La synchronisation d'une opération correspond à la condition d'exécution de l'opération. Elle marque le rendez-vous des événements contributifs qui doivent déclencher l'opération, selon une proposition logique (et, ou) traduisant les règles d'activation (règles de gestion que doivent vérifier les événements contributifs pour déclencher les actions.)

La synchronisation d'une opération est composée de deux éléments :

- d'une part la liste des événements (internes ou externes) qui doivent être arrivés avant de déclencher l'opération.
- et d'autre part la règle sous forme d'une proposition logique qui précise de quelle manière les événements participent au déclenchement de l'opération.

Remarque : *Lorsque l'expression logique de la synchronisation est vérifiée, l'opération est déclenchée et toutes les occurrences d'événements qui ont permis ce déclenchement sont alors consommées par l'opération. Par contre si elle n'est pas vérifiée, les occurrences d'événement restent en attente. Quand une opération est déclenchée par un seul événement, la synchronisation est facultative.*

E - LA REGLE D'EMISSION

La production effective d'un ou de plusieurs événements internes est soumise à une règle d'émission, c'est-à-dire à une proposition logique qui s'applique au contenu de la base d'information après exécution de l'opération. L'événement est produit si la proposition logique est vraie

F - LE RESULTAT

Définition : *Le résultat est un événement produit à la fin d'une opération.*

Le résultat de l'opération dépend de certaines conditions ou, exprimé différemment, une opération peut comporter plusieurs événements en sortie ou résultats de l'opération à des conditions définies. Ces conditions dépendent des informations des événements reçus, d'informations mémorisées ou d'une règle humaine non formalisée.

Exemple : *L'ordre de livraison est émis si le produit est en stock. S'il ne l'est pas, une proposition de produit de substitution est émise ainsi qu'un ordre de réapprovisionnement. L'événement 'facture éditée et émise' est le résultat de l'opération 'établissement de facture'.*

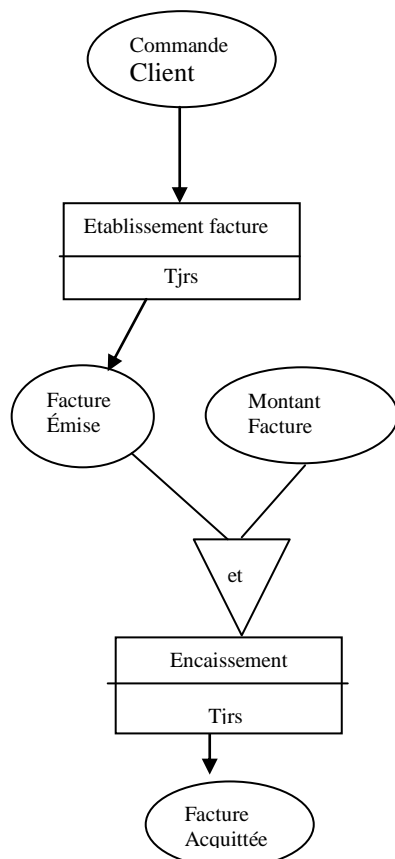
G - LE PROCESSUS

Définition : *Un processus est un ensemble d'opération incluse dans un même domaine d'activité.*

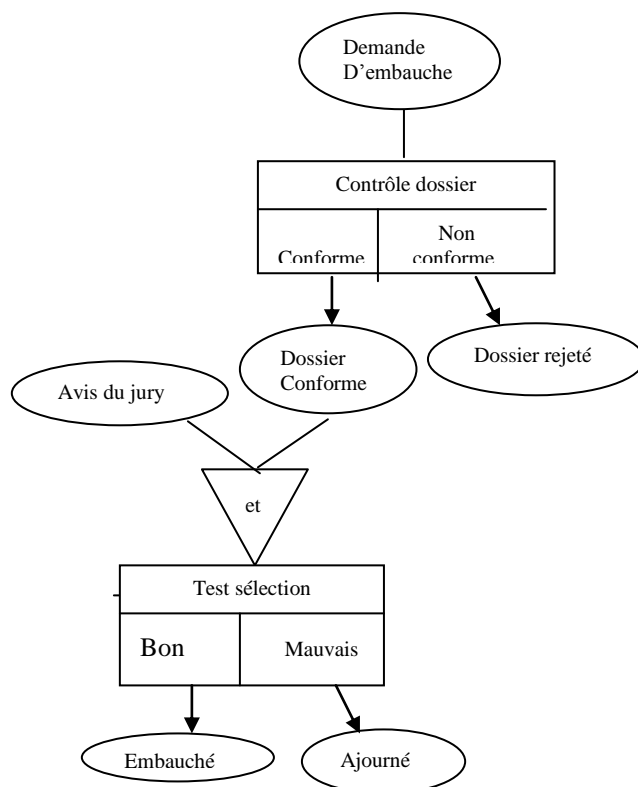
Exemple :- processus facturation
- Processus gestion de commande.

Remarque : *La notion de non interruptibilité suppose que l'opération, au cours de l'exécution du traitement dispose de toutes les données nécessaires à ce traitement; elle ne doit donc pas, en cours de traitement, 'attendre' des données d'un événement qui n'est pas encore introduit dans le SI ; dans ce cas, il faut 'couper l'opération en deux' dès qu'une opération est obligée d'attendre, elle se termine.*

Exemple1 : processus encaissement facture



Exemple 2 : processus d'embauche



II – CONSTRUCTION DU MCT

A - DETERMINATION DU DOMAINE DE L'ETUDE

Définition : *Un domaine est un système de l'entreprise qui a la caractéristique d'avoir une mémoire, un système d'information. Le système d'information sera construit par domaines.*

L'entreprise est décomposée en domaines décomposés en sous domaines, somme de fonctions élémentaires.

Exemple : *l'entreprise est entourée des systèmes tels que Client, Fournisseur, État... les partenaires. Elle est découpée en domaines : vendre, produire, assurer la vie sociale, qui dépendent de ses activités. Ces domaines sont décomposés en systèmes appelés sous domaines (ou champs) : "Produire" peut être découpé en "Maintenir", "Assurer l'approvisionnement des chaînes de production"...*

Le champ de l'étude ou domaine correspond au système d'information étudié. Il est délimité par une définition globale ou par une énumération exhaustive des activités qui le composent. Un domaine doit être défini, au niveau conceptuel, en termes d'activités et non en termes de découpages organisationnels. Une approche théorique est faite avec l'étude des systèmes, la *systémique*. Celle-ci repose sur les principes suivants :

- 1 - Une approche du général au particulier. Tout système se décompose en systèmes.
- 2 - La méthode s'attache à identifier les échanges entre systèmes.
- 3 - La systémique amène à décomposer l'entreprise en systèmes homogènes d'information appelés domaines.

Pour identifier les domaines, deux approches sont possibles :

- 1 - soit ne voir directement que les fonctions majeures qui répondent à la question pourquoi ?
- 2 - soit, si la première démarche n'aboutit pas à un accord général, énumérer des fonctions élémentaires en réunion de "brain storming" et les regrouper en sous domaines, puis en domaines.

Pour cela, répondre toujours à la question pourquoi? Pourquoi gérer des stocks, pourquoi tenir une comptabilité, pourquoi enregistrer une commande ? Ainsi sous domaines et domaines seront découverts.

Exemple : *soit les fonctions analyser la production, produire, gérer les stocks, maintenir, gérer les pièces détachées, suivre les pannes et gérer les réparations.*

L'entreprise est découpée en domaines, sous domaines et fonctions élémentaires

- Pourquoi analyser la production ? Pour produire à moindre coût.
- Pourquoi produire ? Pour vendre.
- Pourquoi gérer les stocks ? Pour approvisionner la production.
- Pourquoi maintenir ? Pour produire.
- Pourquoi gérer les pièces détachées ? Pour approvisionner la maintenance.
- Pourquoi suivre les pannes ? Pour planifier les réparations.
- Pourquoi gérer les réparations ? Pour planifier les réparations.

B - DETERMINATION DES ACTEURS

Définition : *Un acteur est une personne morale ou physique capable d'émettre ou de recevoir des informations.*

Exemple : l'élève de terminale qui souhaite s'inscrire dans une grande école est un acteur du domaine « Gestion des inscriptions ». On distingue deux types d'acteurs :

➤ **les acteurs internes qui appartiennent au système d'information étudié.** Pour le domaine cité ci-dessus, le service des inscriptions ou le service comptabilité de la grande école sont des acteurs internes.

➤ **les acteurs externes qui n'appartiennent pas au système d'information** mais qui sont l'origine ou la destination de flux d'informations reçus ou émanant du système d'information. L'élève de terminale qui effectue une demande d'inscription dans une grande école est un exemple d'acteur externe.

Présentation des acteurs et leurs rôles

| ACTEURS | | Rôles |
|----------|----------|-------|
| Externes | Internes | |
| E1 | | |
| | I1 | |
| | I2 | |

C - LES REGLES DE GESTION DU DOMAINE

Les règles de gestion de domaine sont mises en formes à partir de tous documents (notes, usages, règles orales,...). Les aspects des règles de gestion pris en compte par le MCT sont :

- Les informations échangées entre le domaine et les acteurs.
- L'ordre dans lequel ces informations doivent être traitées.
- Les contraintes éventuelles de temps
- Les règles de traitement des données

Exemple : domaine 'vente '

R1 : à la réception d'une commande provenant soit directement d'un client, soit d'un représentant, la facture est adressée au client ainsi qu'un bon de livraison accompagnant la marchandise.

R2 : le règlement doit être effectué à la réception de la facture. Si le règlement n'est pas parvenu dans un délai de 20 jours, une lettre de relance est adressée au client.

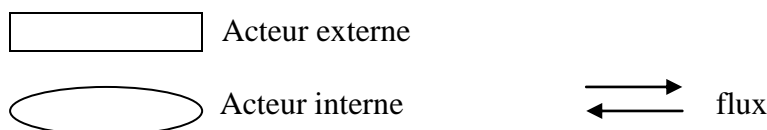
D - LES FLUX D'INFORMATIONS

1 - Définition

Un flux est un échange entre un acteur A et un acteur B

Un flux est un ensemble d'informations faisant l'objet d'échange entre les principaux acteurs du domaine étudié. Les informations sont véhiculées sous forme de messages (annonce, avis, demandes, ...) ou de documents (états et documents imprimés, factures, reçus,...)

Légende :

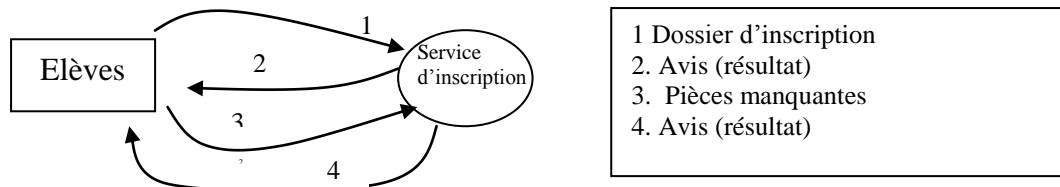


2 - Graphes des flux ou diagrammes des flux

Les acteurs internes - ceux qui appartiennent au système d'information - sont représentés dans des cercles en trait plein,

Les acteurs externes au système d'information sont représentés dans des rectangles. Les flux d'information sont schématisés par des arcs entre acteurs.

Le schéma ci-après représente les flux inhérents à l'inscription dans une grande école.



Ce diagramme met en évidence les événements externes du MCT. Chaque flux échangé d'un acteur externe vers un acteur interne devient en effet un événement déclencheur externe. Les flux à destination d'un acteur externe deviendront des événements internes. Les opérations ainsi que leurs événements résultats ne sont pas aisément déductibles de ce schéma.

3 - Présentation de la matrice des flux

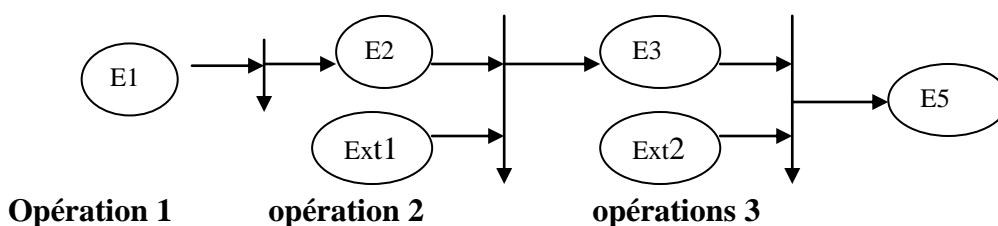
| | E1 | I1 | I2 |
|----|----|----|-----|
| E1 | * | 3 | 1 |
| I1 | 4 | * | |
| I2 | 2 | | 5 * |

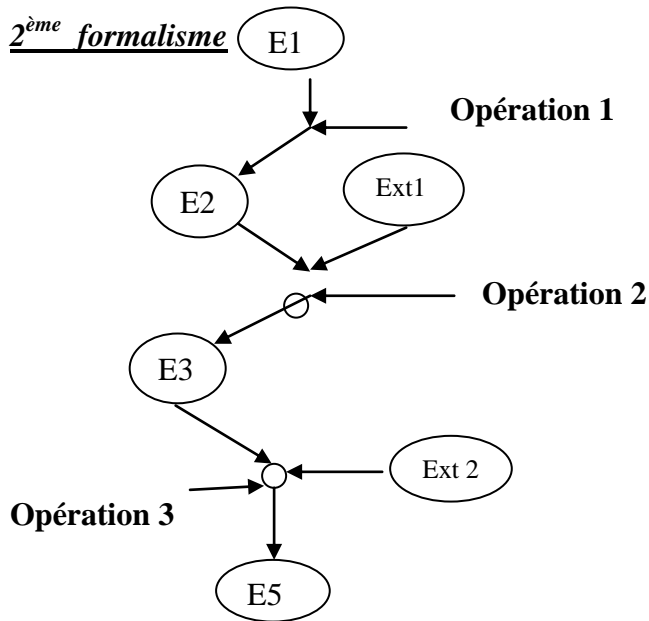
4 - Graphe d'ordonnancement des événements (GOE)

Le GOE ordonne les événements internes en les associant là où il faut, soit aux événements externes, soit aux événements temporels.

Formalisme du GOE

1^{er} formalisme





Exemple de GOE

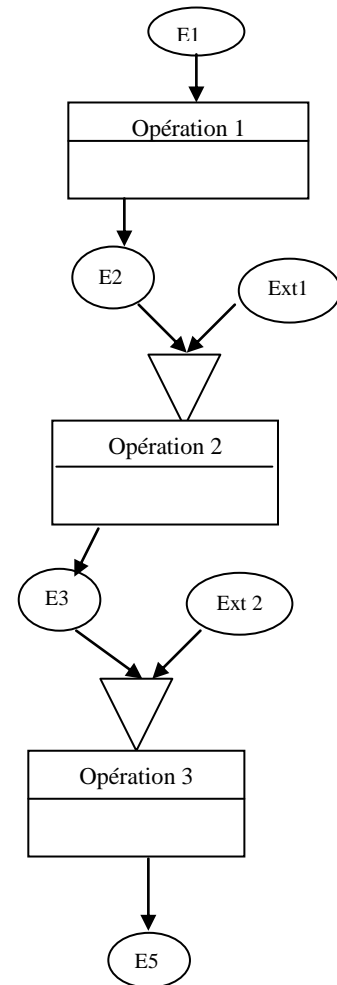


Schéma du MCT déduit du GOE

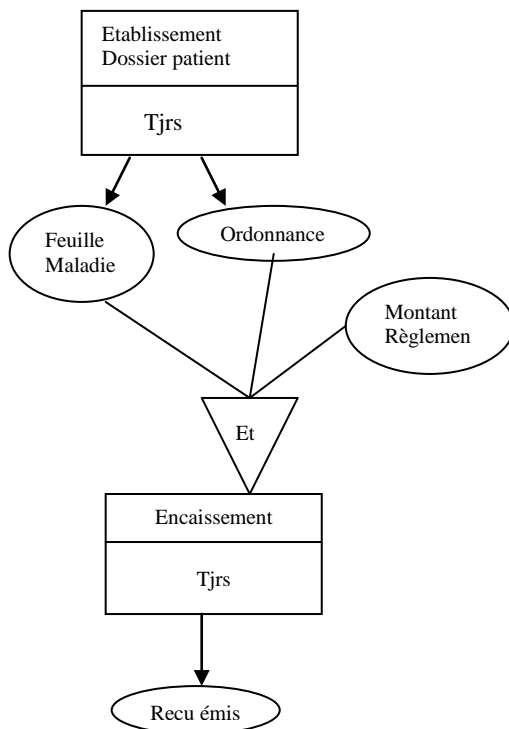
E - LES REGLES DE CONSTRUCTION

Règle de non redondance

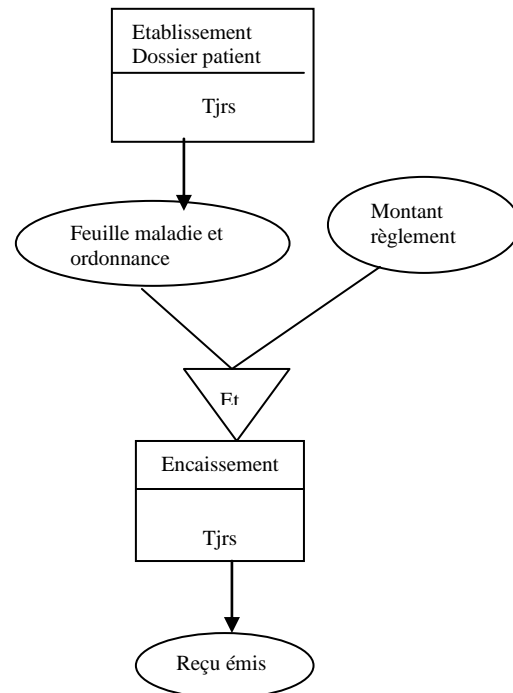
Un MCT ne doit pas comporter des opérations sémantiquement utilisant les mêmes règles de gestion.

Règle relative à un événement

Pour éviter les synchronisations qui n'ont pas lieu d'être, il est important de regrouper dans le même symbole des événements créés simultanément et destinés à une même opération.



MCT à éviter



MCT correct

F - DEMARCHE DE CONSTRUCTION DU MCT

- 1 - Identifier les grands processus.
- 2 - Etablir le MCT pour chaque processus en procédant de la façon suivante :
 - Identifier le champ de l'étude
 - Identifier les acteurs et leurs rôles
 - Etablir le graphe des flux
 - Etablir le GOE
 - Dédurre et vérifier le MCT

Exercices d'application

Exo 1 : Après inventaire chaque 15 du mois en cours, les infirmiers expriment leurs besoins sur des bons de commande. La commande mensuelle concerne les produits tels les antipalustres, anti spasmodiques et anti diarrhéiques. La commande trimestrielle concerne les produits dits matériels tels le coton, les seringues, le sparadrap, les bandes, ...

Chaque fois, après approbation du médecin, les bons sont transmis au service personnel. Le chef du service personnel vérifie la conformité des quotas de produits demandés et transmet les commandes à la DRH.. La secrétaire de la DRH envoie alors une demande de prix aux fournisseurs agréés. 2 jours après l'envoi, la secrétaire du DRH reçoit des factures pro formas. Elle calcule le montant global des commandes et le compare au budget alloué à l'achat des produits médicaux. Si le montant global trouvé est inférieur au budget alloué, elle passe alors la commande définitive. Dans le cas contraire, en accord avec la DRH, elle modifie à la baisse le montant des commandes en diminuant les différentes quantités, puis passe la commande définitive.

TAF : Construire le MCT

Exo 2 : On décrit ci après l'inscription à un établissement supérieur recrutant, sur dossier, des élèves de terminale. L'admission définitive d'un candidat n'est effective que si l'élève a obtenu le baccalauréat. La période des dépôts des dossiers de candidature précède les épreuves du bac. Le candidat commence par déposer un dossier (entre la date d'ouverture et la date de clôture), à partir duquel l'établissement prend une décision soit de refus, soit d'admissibilité (admission sous réserve de l'obtention du bac). Les candidats admissibles sont déclarés admis définitivement lorsqu'ils ont déposés la collante du bac avec paiement des frais d'inscriptions.

TAF : Elaborer le schéma MCT.

LE MODELE LOGIQUE DES DONNEES

C'est pendant qu'il pleut qu'il faut remplir les jarres. (Proverbe chinois)

INTRODUCTION

La description conceptuelle a permis de représenter le plus fidèlement possible les réalités de l'univers à informatiser. Mais cette représentation ne peut pas être directement manipulée et acceptée par un système informatique. Il est donc nécessaire de passer du niveau conceptuel à un second niveau plus proche des capacités des systèmes informatiques. Ce niveau, appelé niveau logique, consiste à choisir l'un des trois modèles suivants :

- modèle hiérarchique (années 80),
- modèle réseau,
- ou modèle relationnel

Chacun de ces modèles repose sur des techniques d'organisation des données particulières que des logiciels seront capables de gérer.

Aujourd'hui, et encore plus demain, il n'existe plus pour le gestionnaire, de réelle alternative entre un modèle de type fichiers et un modèle de type base de données. Le modèle fichier (hiérarchique), trop contraignant dans la mise en œuvre technique, cède la place au modèle relationnel privilégiant l'indépendance données / traitements, permettant par la suite au gestionnaire de s'abstraire des contraintes techniques d'implantation physique. Le modèle relationnel offre le moyen de construire des structures de données dans de multiples domaines d'application de la gestion de l'entreprise. Dans ce chapitre, seul le modèle relationnel sera étudié.

I - LES CONCEPTS

A - L'ATTRIBUT

Définition : *Appelé aussi composant ou caractéristique, l'attribut désigne le plus petit élément d'information manipulable par le concepteur.*

Il est décrit par un "nom d'attribut," prend des valeurs dans un domaine de valeurs et a un sens dans le SI étudié.

Remarque : *L'attribut ici correspond à la propriété dans le MCD*

Exemple : CodeClt
NomEtu

B – LA RELATION

Définition : *Une relation est un ensemble d'attributs significativement associés.*

Exemple : *chaque entité du MCD se traduit en une relation dont les attributs seront les propriétés (rubriques) de l'entité.*

Formalisme d'une table :

NOMRELATION (Clé primaire, Attribut1, Attribut2,..., Clé étrangère)

Exemple : ETUDIANT (MatEtu, NomEtu, PrenomEtu, DateNaisEtu, CodeClasse)

C - CONTRAINTE D'INTEGRITES DE REFERENCE

Elle concerne les valeurs prises par un attribut dans une relation par rapport aux valeurs prises par un attribut, sémantiquement et syntaxiquement équivalent dans une autre relation. Elle contrôle essentiellement les valeurs des clés étrangères.

Exemple : *un étudiant ne peut pas appartenir à une classe qui n'existe pas.*
ETUDIANT (MatEtu, NomEtu, PrenomEtu, StatutEtu, CodeClasse)
CLASSE (CodeClasse, LibClasse)

III - LA MISE EN ŒUVRE DES CONCEPTS

A - DEFINITIONS

Base de données : *Ensemble de données organisées (structurées) relatives à un domaine et gérées par un SGBD.*

SGBD : *C'est un logiciel permettant de décrire, manipuler et traiter les données d'une base de données*

Exemple : D Base, PARADOX, MS-ACCESS

SGBDR : *C'est un logiciel qui permet la création et la modification de la structure des tables à tout moment si nécessaire et qui autorise les opérations relationnelles sur ces tables*

Exemple : MS-ACCESS, PARADOX, SQLBASE, ORACLE.

B - LES REGLES DE PASSAGE DU MCD AU MLDR

La création des bases de données relationnelles s'appuie sur la théorie mathématique des relations, et la restitution des données va être réalisée en utilisant l'algèbre relationnelle.

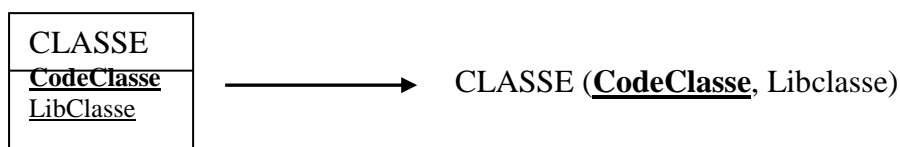
Définition : *une base de données relationnelles est un ensemble de relations regroupant toutes les informations de SI.*

Toute entité devient une relation . L'identifiant de l'entité devient clé primaire de la relation.



C'est à dire que chaque d'entité du modèle conceptuel des données devient une relation dans le modèle logique. L'identifiants de l'entité devient la *clé de la relation*, tandis que les attributs standards deviennent des attributs de la relation.

Exemple :

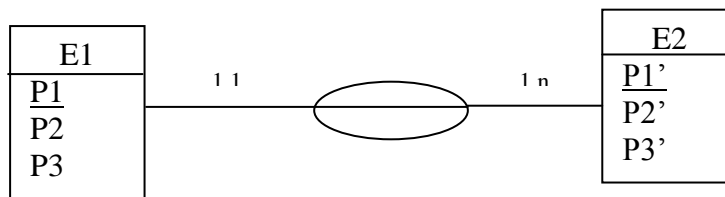


Le passage du modèle conceptuel au modèle logique au niveau des relations se fait selon les cardinalités des entités participant à la relation:

Toute association binaire de type 'Père-Fils' (ou CIF binaire) est caractérisée par l'existence d'une dépendance fonctionnelle entre l'identifiant de l'entité reliée par le segment portant la cardinalité 1,1 ou 0,1 et l'autre entité. Dans le schéma ci-dessous on a la dépendance fonctionnelle suivante : $P1 \rightarrow P1'$

Une telle association entraîne l'intégration de l'identifiant de l'entité but de la dépendance fonctionnelle dans la relation associée à l'entité source de la dépendance fonctionnelle. La propriété ainsi dupliquée devient **clé étrangère** dans la relation source et est marquée par un dièse (#) ou souligné en pointillés.

Une clé étrangère dans une relation est une propriété qui est clé primaire dans une autre relation.

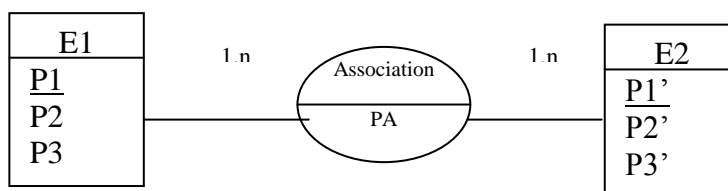


$E1(\underline{P1}, P2, P3, \#P1')$
 $E2(\underline{P1'}, P2', P3')$

Ainsi, on reporte dans la relation traduisant l'entité se trouvant du côté 0,1 ou 1,1 (Fils) de l'association, la clé primaire de la relation traduisant l'autre entité 0,n ou 1,n (Père). Cette clé ainsi reportée est appelée **clé étrangère**. Dans le formalisme, la clé primaire est soulignée d'un trait continu, la clé étrangère est soulignée avec des tirets ou on la repérera par le symbole dièse.

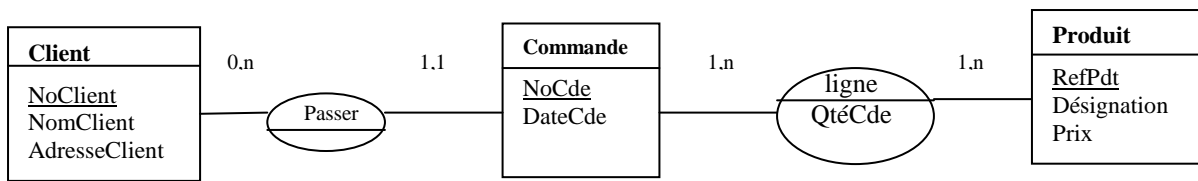
Exemple : ETUDIANT (MatEtu, NomEtu, PrenomEtu, CodeClasse)

Toute association de type (m-n) devient une relation qui hérite des identifiants des entités participants à la relation. Si l'association est porteuse, la relation sera complétée par la liste des propriétés portées. (Une association de ce type devient une table, sa clé primaire est obtenue en concaténant les identifiants des entités qui participent à l'association .si l'association est porteuse de propriétés, celles –ci deviennent des attributs de la table.)



$E1(\underline{P1}, P2, P3)$
 $E2(\underline{P1'}, P2', P3')$
 Association (P1, P1', PA)

L'exemple ci-dessous illustre l'application de ces trois règles sur le modèle conceptuel classique de gestion des commandes suivant :



Modèle relationnel est :

CLIENT(NoClient, NomClient, AdresseClient)

COMMANDE (NoCde, DateCde, NoClient#)

PRODUIT(RefPdt, Désignation, Prix)

LIGNE(NoCde, RefPdt, QtéCde)

CLIENT

| <u>NoClient</u> | NomClient | AdresseClient |
|-----------------|-----------|----------------|
| 1 | Lassiné | Abobo-Marché |
| 2 | Mondakan | Yop-Siporex |
| 3 | Chaupi SA | Plateaux- CCIA |

COMMANDE

| <u>NoCde</u> | DateCde | NoClient# |
|--------------|------------|-----------|
| 100 | 14/04/2008 | 2 |
| 101 | 14/04/2008 | 1 |

PRODUIT

| <u>RefPdt</u> | Désignation | Prix |
|---------------|----------------------|---------|
| VE45 | Vélo | 65000 F |
| VE32 | Kit 2 roues arrières | 38300 F |
| VE21 | Kit éclairage | 21500 F |

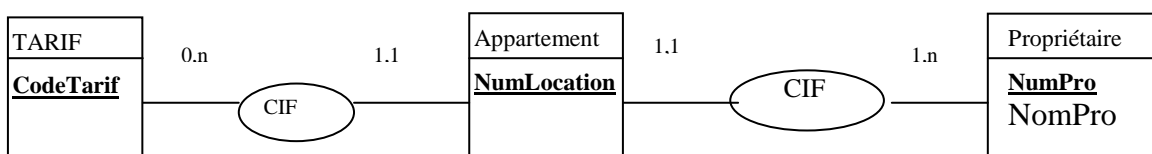
LIGNE

| <u>NoCde#</u> | <u>RefPdt#</u> | Qté |
|---------------|----------------|-----|
| 100 | VE45 | 4 |
| 100 | VE32 | 5 |
| 101 | VE21 | 10 |

On peut remarquer, que dans le modèle relationnel les cardinalités minimales portées par le modèle conceptuel sont perdues. Celles-ci seront en fait représentées au moyen des contraintes de valorisation.

Traduction d'une classe d'agrégation

Dans le cas de la présence d'une classe d'agrégation, la classe d'entité agrégée a comme attributs supplémentaires les attributs de la classe d'entité agrégante

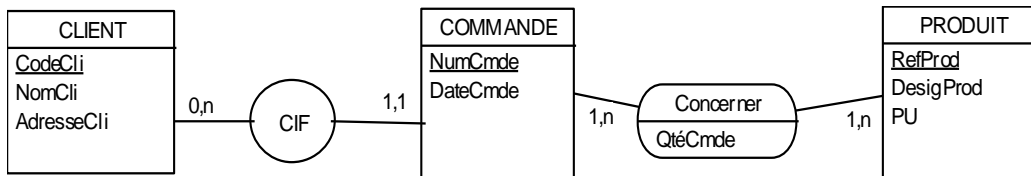


TARIF (CodeTarif, PrixSemHS, PrixSemBS)

APPARTEMENT (NumLocation, ..., CodeTarif, NumPro)

PROPRIETAIRE (NumPro, NomPro, ...)

Remarque : Dans le cas d'une relation de type (0,1 – 0,1) ou (0,1 – 1,1) ou encore (1,1 – 1,1), toutes les propriétés de l'une des entités migrent dans l'autre entité et on choisit une clé issue indifféremment de l'un des identifiants en relation



Résolution

CLIENT (CodeCli, NomCli, AdresseCli)
COMMANDE (NumCmde, DateCmde, CodeCli)
PRODUIT (RefProd, DesigProd, PU)
CONCERNER (NumCmde, RefProd, QtéCmde)

Exercices d'application

Exo1 : cas Gestion des commandes

La société NEHEMIE & Frères achète et vend des consommables informatiques et fournitures de bureau. Pour l'automatisation de ses activités, à partir du MCD, il vous est demandé d'élaborer le MLD relationnel

Exo2 : A partir du MCD de l'exo 2 du chapitre MCD, établir le MLD relationnel

Sans peau, où pousseraient les poils ? (Proverbe chinois)

INTRODUCTION

Le Modèle Organisationnel des Traitements (MOT) s'attache de décrire les propriétés des traitements non traitées par le modèle conceptuel des traitements, c'est-à-dire: le temps, les ressources, le lieu. Le modèle organisationnel des traitements consiste donc à représenter le modèle conceptuel des traitements dans un tableau dont les colonnes sont la durée, le lieu, les responsables et ressources nécessaires à une action.

I - LES CONCEPTS

Les paramètres organisationnels (poste de travail, type du traitement, lieu du traitement, période du traitement) servent de base au découpage en procédures fonctionnelles

A - LE POSTE DE TRAVAIL

Un poste de travail est une cellule d'activité comprenant les moyens humains et matériels nécessaires à la réalisation de cette activité. Cette cellule exerce son activité dans un lieu déterminé. De par cette définition on caractérise le poste de travail par :

Un lieu : *Il représente l'ensemble des lieux où les actions d'une opération pourront s'effectuer*

Exemple : - le magasin est l'ensemble des endroits où s'effectue l'inventaire
- Service comptabilité fournisseur

Un responsable : *Personne ou ensemble de personnes ayant la responsabilité de certaines actions d'une opération*

Exemple : *le chef magasinier est responsable de l'inventaire*

Des ressources : *moyens permettant de réaliser certaines actions d'une opération (personnel logiciel matériel)*

Exemple : - Aide-comptable
- Micro-ordinateur
- Disquette
- Papier

Exemple de poste de travail :

- Chef service informatique
- Directeur général
- Coursier
- Directeur financier

B - LE TYPE (OU NATURE) DU TRAITEMENT

La nature du traitement définit le degré d'automatisation de ce traitement. Il y a trois types de traitements :

Manuel : aucune utilisation de l'informatique

Automatisé conversationnel (ou interactif) : succession de tâches manuelles et de traitements informatiques (dialogue homme – machine) en temps réel : **exemple** : saisie

Remarque : Le type interactif concerne les travaux classiquement effectués à l'aide d'un ordinateur le plus souvent par l'intermédiaire d'un clavier : écran mais à condition qu'il y ait effectivement un dialogue homme machine cela suppose une saisie d'information par le poste de travail.

Exemple : - saisie de données pour la mise à jour
- Saisie de paramètre de sélection

Automatisé batch : traitement entièrement automatisé

Le type automatisé batch concerne tous les traitements collectifs effectués en différé. Ici il n'y a aucune saisie de données supplémentaires mais exécution d'un traitement à partir des données déjà contenues dans le SI. Il n'y a donc aucune intervention humaine en dehors du déclenchement technique du traitement (niveau opérationnel)

Exemple : - Appuyer sur le bouton
- Choisir une commande dans un menu

C - LA PERIODE

Elle indique les conditions temporelles d'activités d'une procédure fonctionnelle

Exemple : plages de temps pendant lesquelles une procédure fonctionnelle peut être exécutée : heure et jour ouvrables

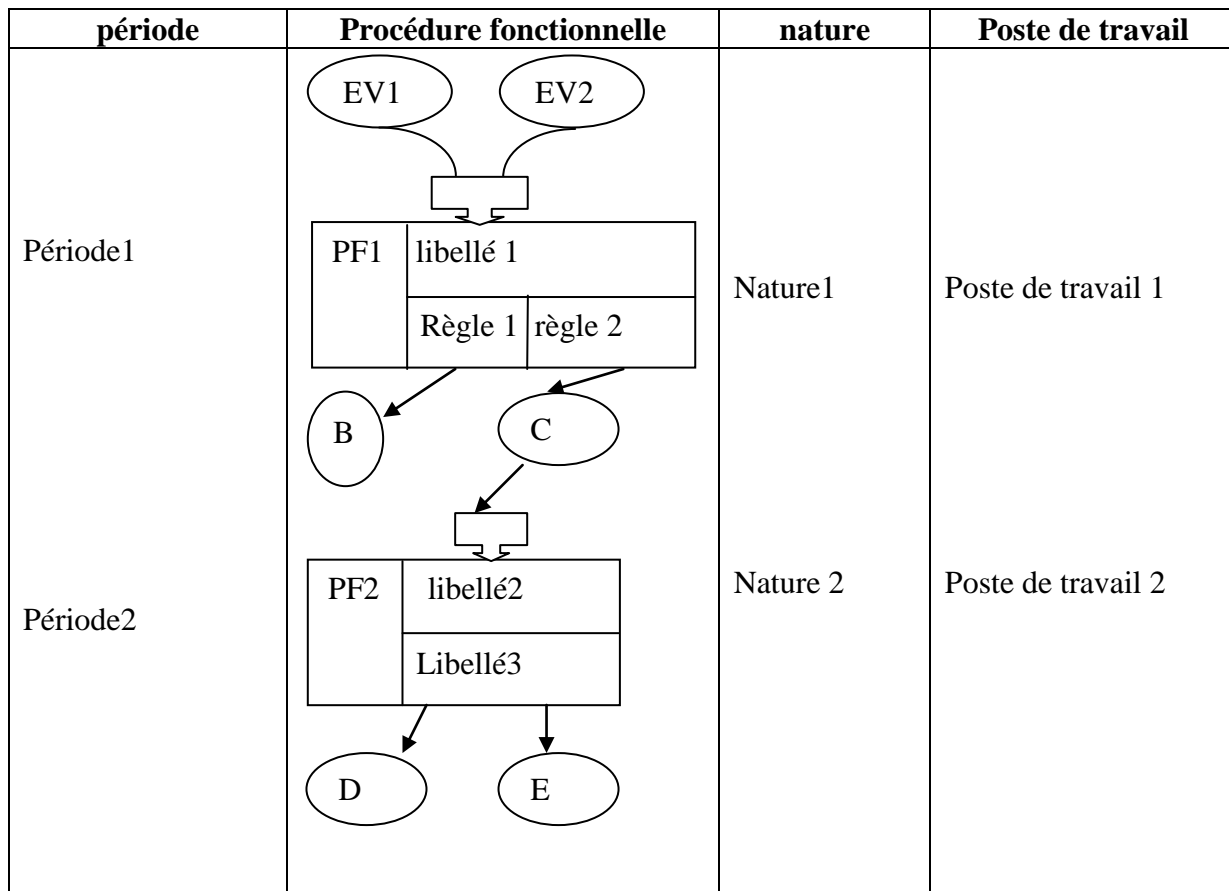
Ces plages concernent essentiellement les procédures fonctionnelles de types manuel ou interactif
Des fréquences de traitements : 2 fois par jours, le dernier vendredi du mois (généralement le type automatique mais également d'autres types éventuellement)

D - LA PROCEDURE FONCTIONNELLE

1 - Définition

Une procédure fonctionnelle est un ensemble de traitements exécutés sans interruption par un même poste de travail utilisant des moyens de traitements d'un type déterminé pendant une période d'activités déterminée

La représentation d'une procédure fonctionnelle est identique à celle d'une opération dans le MCT : les événements en entrée, le numéro de la procédure (PF), la description abrégée du traitement (ou libellé), la synchronisation, les règles d'émission, les événements résultats, auxquels s'ajoutent bien entendu, les paramètres organisationnels caractérisant la procédure fonctionnelle



NB : l'enchaînement des procédures fonctionnelles est réalisé par des événements comme pour le MCT

II – REGLES DE PASSAGE DU MCT AU MOT

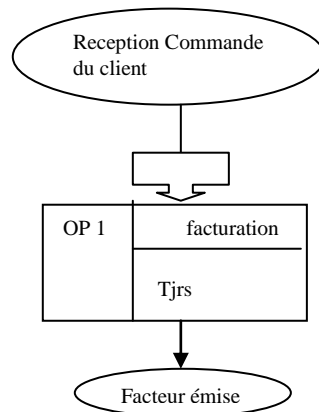
Une opération peut correspondre à une procédure fonctionnelle : c'est quelque fois le cas des applications simples impliquant un poste de travail unique.

La plupart du temps une opération (dans le MCT) peut correspondre à plusieurs procédures fonctionnelles pour les raisons les suivantes :

- *Les traitements de l'opération sont effectués successivement par des postes de travail différents.*
- *Tel ou tel traitement de l'opération est effectué ou non à l'aide de moyens informatique.*
- *Il existe des traitements pratiques de nature spécifiquement organisationnelle (vérification des données) ;*
- *Certains traitements de l'opération sont de natures différentes.*

Exemple : A l'arrivée de la commande d'un client, la facture est émise immédiatement puis transmise au magasin pour livraison au client. En cas de rupture de stock le magasin fait établir un avoir. **TAF** : Etablir le MOT du processus FACTURATION

MCT du processus FACTURATION



MOT du processus FACTURATION

| Période | Procédure fonctionnelle | Nature | Poste de travail |
|------------------------------|-------------------------|---------------------|-----------------------------|
| Jours Heures Ouvrables | | Manuel | S ^{ce} Commercial |
| Jours Heures Ouvrables | | Interactif | S ^{ce} facturation |
| 2 fois par jour | | Automatisé Batch | S ^{ce} facturation |

Commande inexploitable : document inexploitable

Commande rejetée : inexistence du client, des produits, ...

Remarque : Les caractéristiques d'une procédure fonctionnelles

Le MOT ne présente en fait qu'une partie des informations concernant la solution organisationnelle.

- La fiche descriptive de procédure fonctionnelle fournit les informations quantitatives (volume, durée) ainsi qu'une description résumée du traitement.

- La fiche de poste de travail récapitule les procédures fonctionnelles exécutées par le poste de travail

LES SPECIFICATIONS DETAILLEES

Les spécifications détaillées d'une procédure fonctionnelles comprennent :

- *Le découpage du traitement de la procédure fonctionnelle en traitements élémentaires (appelées tâches)*

- *La description détaillée des supports de données associés au traitement (document papier, écrans, supports informatiques, ...)*

Les flux entrant constituent les informations qui doivent être traitées par la procédure et les flux sortant constituent l'ensemble des informations émises lors d'une exécution de la PF et engendrent soit un flux entrant dans une autre PF, soit des résultats pour l'univers extérieur au SI.

Remarque : *le flux se véhicule par un support (papier, disquette)*

Pour décrire de manière détaillée les PF ou utilisera les outils suivants :

Fiche de description de procédure

| Fiche de description de la procédure PF N° |
|--|
| Nature : conversationnelle Objet : MAJ du stock Action sur la base d'information : consultation Données utilisées : Données sorties : |

Description des mouvements portés par les événements

Il s'agit de décrire :

- *les dessins des états de sorties.*
- *les documents de saisie*

Remarque : *les choix de détails concernant la forme générale des documents et des écrans doivent être valides par les utilisateurs il faut donc être très regardant aux problèmes ergonomiques lors de la conception des écrans.*

III – Applications

Reprendre les exercices du chapitre MCT et établir le MOT

Viser l'ensemble et se mettre à l'oeuvre par les détails. (Expression chinoise)

INTRODUCTION

Le Modèle Physique de Données (MPD) est un modèle de la base de données. L'implantation physique, la topographie des enregistrements informatiques ne sont pas définies. *Trois systèmes de gestion de base de données, les modèles hiérarchiques, navigationnels et relationnels servent de modèles.*

Le choix du type de SGBD effectué, deux questions se posent : comment retrouver physiquement un chemin logique, par une information ou par un lien physique ? Faut-il éclater ou regrouper certains enregistrements afin d'accélérer les traitements ?

Un chemin indique le moyen d'arriver à une information à partir d'une information de départ. Deux moyens existent, en informatique, pour retrouver cette information, soit connaître son adresse ou sa position "physique" dans un fichier informatique, soit connaître sa valeur. Connaître son adresse consiste à conserver un chemin "physique", *un lien*. Connaître sa valeur consiste à dupliquer l'information. Un lien physique n'existe qu'entre enregistrements physiques.

Le MPD dépend des logiciels de développement nécessaires à la programmation et à la manipulation des données. La méthode laisse place aux normes du réel. En l'état de l'art, on arrive assez facilement à déduire du MLD des structures de SGBD (ou à partir d'un dessin d'écran, des programmes transactionnels).

Aujourd'hui il n'existe pratiquement pas d'approche normalisée de description et de présentation du niveau physique des données. En effet, la description d'un Modèle Physique des Données (MPD) est étroitement liée aux choix techniques concernant le système de gestion des données.

I - PRINCIPES GENERAUX

Le SGBDR se développe de plus en plus. C'est pourquoi il est plus approfondi. Comme il ne gère pas d'adresse ou de lien, son formalisme est plus facile à comprendre. Dans un modèle relationnel, tous les enregistrements sont composés d'informations et toutes les informations peuvent être accédées indépendamment de leur adresse

Un langage normalisé de création et d'interrogation des enregistrements et des informations existe pour tout SGBD relationnel. Il est appelé *SQL* pour Structured Query Language. Ce langage permet à un utilisateur final d'interroger, après une formation préalable, toutes les informations de la base de données. Cette recherche d'informations peut être menée de toutes les manières possibles, même celles non prévues par des programmes informatiques.

Un enregistrement d'un SGBD relationnel est appelé relation ou table. Ces relations n'ont rien à voir de près avec les relations du modèle conceptuel de données. Le modèle est appelé relationnel car l'établissement d'une navigation entre deux enregistrements nécessite la mise en relation, à valeurs égales, d'informations d'enregistrements. Table ou tableau sont les termes les plus simples spécifiques des SGBD relationnels. Les colonnes d'une table sont les informations et les lignes les occurrences d'enregistrement, les ensembles d'occurrences d'information.

II - ELABORATION DU MPD

En tenant donc compte de l'environnement technique de développement et du langage de programmation choisis pour l'implémentation du logiciel, il s'agit ici de décrire les données à utiliser pour l'atteinte des objectifs assignés. La description du MPD se fera donc dans le langage du système de gestion des accès correspondant à la solution choisie. De plus l'environnement technique de développement influera aussi largement sur la description du niveau physique.

Pour des raisons d'usages pratiques, nous présenterons la ²description d'un système de gestion de base de données de type relationnel. (Voir le cours sur les bases de données). Dans ce cas chaque table du MLD se présentera comme le modèle suivant dans le cas de l'utilisation de MS-ACCESS comme SGBDR. Ce modèle de MPD nous permet de décrire toutes les tables et de spécifier leurs caractéristiques. Les SGBD relationnels doivent respecter certaines conditions (Règles de CODD) ainsi que leurs enregistrements (Normalisations).

Les douze Règles de Codd (plus la zéro).

Ces règles ont été exprimées par M. CODD et servent de modèle pour les SGBD actuels. Elles sont souvent mentionnées et rarement citées. C'est pourquoi elles le sont ici. Une vue est une vue externe telle que définie au niveau conceptuel. Dans le cas de SGBD relationnel, une vue est un ensemble de tables que l'utilisateur manipule. Ces tables manipulées peuvent être différentes des tables définies dans le SGBD.

Règle 0 - SGBD relationnel. Un SGBD relationnel doit pouvoir gérer toutes les bases de données par ses possibilités relationnelles (recherche d'une occurrence par combinaison d'ensembles d'occurrences).

Règle 1 - Règle sur les informations. Toutes les informations, dans une base de données relationnelle, sont dans des tables

Règle 2 - Garantie d'accès. Toute information élémentaire dans une base de données relationnelle est logiquement accessible par combinaison de nom de tables, de clé primaire (index unique), et de nom de colonne.

Règle 3 - Traitement des valeurs vides. Les valeurs vides sont supportées par les SGBD relationnels pour représenter des informations inconnues ou inapplicables de manière systématique.

Règle 4 - Catalogue "relationnel" de la base. La description des bases de données est représentée comme des données ordinaires. Le langage d'interrogation est le même.

Règle 5 - Syntaxe d'un langage concernant les données. Un SGBD relationnel possède au moins un langage

- possédant une syntaxe définie,
- comprenant les fonctionnalités suivantes : définition des données, des vues, manipulation des données (de manière interactive et par programme), contraintes d'intégrité, autorisation (d'accès), commandes de transaction (commencer, valider ou mise à jour logique, revenir à la validation précédente).

Règle 6 - Règles de modification des vues. Toutes les vues théoriquement modifiables (création et suppression) sont modifiables par le SGBD.

Règle 7 - Création, modification et suppression de tables. La possibilité de manipuler une table comme opérande est valable pour la sélection, la mise à jour et la suppression.

Règle 8 - Indépendance vis-à-vis des données physiques. Programmes d'application et transactions ne sont pas modifiés par des changements de stockage ou de méthodes d'accès aux informations.

Règle 9 - Indépendance vis-à-vis des données logiques. Programmes d'application et transactions ne sont pas modifiés par des changements sur les tables sans perte théorique d'information.

Règle 10 - Indépendance des contraintes d'intégrité. Les contraintes d'intégrité sont définies et stockées dans un catalogue (et non dans un programme).

Règle 11 - Indépendance vis-à-vis de la répartition. Un SGBD relationnel est indépendant de la répartition des données.

Règle 12 - Règle de respect des contraintes. Si un SGBD possède un langage "détaillé" (une ligne de table à la fois par opposition à la manipulation des lignes de tables par intersection sur des ensembles de lignes), celui-ci doit tenir compte des contraintes d'intégrité (référentielles et d'intégrité sur la clé) déclarées dans le langage "général".*

Formalisme d'une table

| Nom de la table : CHEQUE Mode d'organisation : relative Mode d'accès : directe Clé primaire : numCheque | | | | | | |
|--|------------------|----------------------|-------------|---------------|---------------|---------------------|
| N° ordre | Code | Signification | Type | Taille | Format | Observation |
| 01 | NumCheque | Numéro du chèque | Texte | 15 | complet | Clé primaire |
| 02 | DateCheque | Date du chèque | Date | 10 | | |
| 03 | MontantCheque | Montant du chèque | Monétaire | 8 | | |

Deux touches pour les yeux et voilà le dragon dessiné qui s'anime. (Expression chinoise)

INTRODUCTION

Le modèle physique des traitements comprend les programmes informatiques et leur environnement d'exploitation, moniteurs temps réel, traitements par lot, temps partagé...

Le but du Modèle OPérationnel des Traitements (MPT ou MOPT) est de décrire l'architecture des logiciels qui devront être réalisés à partir du modèle organisationnel des traitements (MOT) qui a permis de décrire :

- **Les phases "temps réel"** ou automatisé conversationnel "avec la description suivante :
 - Ecrans et leurs enchaînements
 - Traitements associés à chaque écran
 - Des états papiers éventuels
- **Les phases "temps différé"** ou "automatisé batch" avec la description des états des traitements regroupés.

Le MPT permet donc de décrire l'architecture des unités de traitements correspondant aux phases décrites dans le MOT. Les unités de traitements sont les procédures fonctionnelles de nature automatisée.

I - DEFINITIONS

Définition : *Une unité de traitement est une procédure fonctionnelle de nature automatisée.*

Le MPT consiste en l'écriture du programme. Celui-ci peut être généré dans le cadre d'un "atelier de génie logiciel". La finalité de méthodes telles que MERISE est la production de "code" automatique à partir de la conception. La maquette est l'enchaînement des états sans réels calculs. Un prototype exécute des calculs, met à jour des données et l'utilisateur final peut presque dire que la programmation est terminée.

L'outil informatique retenu par l'utilisateur va être réalisé par un programmeur. Celui-ci est libre de découper ses programmes de manière à les réutiliser dans les différents outils informatiques. Les programmes d'édition sont un exemple de programme spécial permettant de créer tout état de sortie en fonction des informations de la base de données. C'est le cas des progiciels sophistiqués.

Définition de la contrainte d'intégrité référentielle. En déclarant l'information "NumClasse" dans la table ETUDIANT comme dépendante de l'information "NumClasse" dans la table CLASSE, la valeur de "NumClasse" dans la table CLASSE *doit exister avant* la valeur "NumClasse" dans la table ETUDIANT. La référence "NumClasse" de ETUDIANT se trouve dans la table CLASSE. IG2A ne peut être enregistré dans la table ETUDIANT que si la CLASSE IG2A existe dans la table CLASSE

Toute colonne d'une table relationnelle peut faire l'objet d'accès pour rechercher une ou des lignes (à l'aide du langage SQL en particulier). Les colonnes d'une table, les informations, peuvent être déclarées comme critère d'accès accéléré ou colonne indexée. La plupart du temps, ces colonnes indexées proviennent des chemins retenus au niveau logique.

La colonne NumClasse peut être indexée dans la table ETUDIANT et conserve ainsi le chemin de CLASSE vers ses étudiants.

Chaque table peut posséder une *clé* permettant d'accéder à la bonne ligne. Cette clé est déclarée comme "*index unique*". Deux Classes ne peuvent posséder le même numéro, la même clé. Un index unique peut être la composition de plusieurs colonnes.

Exemple : le tableau "CLASSE" comprendra la colonne index unique "NumClasse", la colonne index "LibClasse", la colonne index "EffectifClasse" si je veux retrouver toutes les classes ayant le même effectif d'une manière accélérée.

Remarque : Aucune information "*composante*" (la clé) ne peut être vide pour une ligne de table. C'est la "Contrainte d'intégrité sur la clé".

II - LA NORMALISATION DES TABLES

Le résultat du choix des colonnes de table doit respecter certaines règles. Les premières s'appuient sur les informations (formes normales), les secondes (règles de Codd) sont plus larges et définissent l'environnement des tables. Normalisation et règles de Codd sont décrites dans ce paragraphe.

Les rapports entre tables, s'appuyant sur les informations, sont classés en cinq formes dites normales. Ces classes de normalisation des tables permettent de respecter la cohérence des informations entre tables : par exemple, ne pas mettre le prénom du fils dans la table du père. Ces classes de normalisation recoupent les règles de construction du MCD. Un MCD correctement construit entraîne la génération de tables normalisées.

Les règles de CODD permettent de vérifier si un SGBD est relationnel ou non. Les SGBD relationnels actuels ne sont pas actuellement aussi performants en rapidité que les autres du fait de la gestion des informations et non des adresses. Cette performance va en s'améliorant. Tout n'est qu'état de l'art de la technique.

La normalisation des tables consiste à répartir les informations dans les tables en fonction de règles. Seules les clés peuvent être redondées. Cinq étapes de normalisation sont distinguées. A chaque étape, les tables sont déclarées comme étant en première, deuxième... cinquième forme normale. Le but est d'arriver à la dernière étape pour obtenir des tables normalisées. Cette normalisation est obligatoire uniquement si les tables ont été directement construites sans méthode. Ces règles peuvent être rapprochées des règles sur les informations d'individus ou de relation (une seule valeur d'information par individu ou relation par exemple).

Quand le passage s'effectue du MLD au MPD, les tables sont obligatoirement normalisées. Merise évite d'avoir à normaliser les tables.

Dépendance multi-valuée et fonctionnelle sont des termes typiques du modèle relationnel. Une dépendance multi-valuée est un chemin d'une information de départ à *N* informations d'arrivée. A partir d'un "nom de pays", *N* "nom de villes" sont trouvés.

Une dépendance fonctionnelle est un chemin d'une information de départ à *1* information d'arrivée. A partir d'un "nom de ville", 1 "nom de pays" est trouvé, ainsi qu'1 "nombre de citoyens".

L'information du modèle relationnel.

Qu'est-ce que l'information en relationnel ? Une information est-elle dans une table ou dans plusieurs tables ? Le numéro du père dans la table fils est différent du numéro du père dans la table père. Pour manipuler des informations en relationnel, il est nécessaire de spécifier la table de l'information. Chercher le numéro du père dans la table père est différent de chercher le numéro du père dans la table fils. Il peut fort bien n'y avoir aucun rapport entre deux colonnes de tables différentes portant le même nom. Des colonnes telles que adresse, date, numéro, code, montant sont possibles en relationnel sans préciser adresse du client, adresse du fournisseur...

Un ou des enregistrements du MLD peuvent donner naissance à *plusieurs* (éclatement d'enregistrement) ou *un* enregistrement(s) (regroupement d'enregistrements) physiques du MPD. Si les informations des enregistrements sont très différentes, l'éclatement prend en compte moins d'occurrences lors de la manipulation des enregistrements. Le regroupement prend en compte moins d'enregistrements, mais plus d'occurrences. Le choix s'effectue en fonction de la fréquence des programmes manipulant les enregistrements et de la comparaison en gain de temps des solutions.

Éclatement d'enregistrement logique.

Soit un exemple de contrat dont 80% des contrats sont du type 1, 10% du type 2 et le reste (10%) d'au moins cinq types différents. L'enregistrement logique dérivé de l'individu contrat est unique. Les enregistrements physiques peuvent être optimisés et l'enregistrement "contrat" éclaté en Contrat1, Contrat2 et Contrat autre. Les informations de l'enregistrement Contrat1 sont définies et non communes aux autres.

Nota : *Tout programme met à jour ou lit des informations dans des enregistrements physiques et des liens d'enregistrements. Toute information d'un MPD doit être créée et consultée par un programme*

L'objectif de ce paragraphe est d'apprendre à transmettre à un S.G.B.D. R. (Système de Gestion de Bases de Données Relationnelles) la structure d'une base de données relationnelle. Cette déclaration consiste :

➤ à **indiquer les champs** (encore appelés colonnes ou attributs) qui composent chacune des tables du modèle relationnel. Dans cette première étape, la définition d'un champ se limitera à la donnée de son nom et de son type ; ces deux informations étant issues du dictionnaire des données établi lors de la phase d'analyse conceptuelle.

➤ à **implémenter des contraintes d'intégrité** destinées à garantir la cohérence des données mémorisées ; dans cette seconde étape seront mises en place (entre autres) les notions de clé primaire et de clé étrangère représentées sur le modèle relationnel respectivement par un soulignement et un #.

III – LES CONTRAINTES D'INTEGRITE

Considérons par exemple le schéma relationnel suivant :

COMMANDE(NoCommande, DateCommande, DateLivraison, NoClient#)

CLIENT (NoClient, NomClient, TypeClient)

| <i>Rubriques</i> | <i>Type</i> |
|----------------------|--|
| <i>NoCommande</i> | <i>Entier</i> |
| <i>DateCommande</i> | <i>Date</i> |
| <i>DateLivraison</i> | <i>Date</i> |
| <i>NoClient</i> | <i>Entier</i> |
| <i>NomClient</i> | <i>Chaîne de 25 caractères max</i> |
| <i>TypeClient</i> | <i>Chaîne de 15 caractères max, 2 valeurs possibles : PARTICULIER ou PROFESSIONNEL</i> |

La première étape de la déclaration de cette base consiste à transmettre au SGBD R les informations suivantes :

| Table CLIENT | | Table COMMANDE | |
|--------------|-----------------------|----------------|--------|
| NoClient | Entier | NoCommande | Entier |
| NomClient | Chaîne de 25 cars max | DateCommande | Date |
| TypeClient | Chaîne de 15 cars max | DateLivraison | Date |
| | | NoClient | Entier |

Si on limitait la création de cette base à cette étape, le SGBD R pourrait accepter les lignes suivantes :

| CLIENT | | | COMMANDE | | | |
|----------|-----------|---------------|------------|--------------|---------------|----------|
| NoClient | NomClient | TypeClient | N°Commande | DateCommande | DateLivraison | NoClient |
| 1 | | PROFESSIONNEL | 101 | 15/12/1999 | | 3 |
| 2 | DOPE | PARTICULIER | 102 | 17/12/1999 | 16/12/1999 | 1 |
| 2 | DALEBE | ARTISAN | 103 | 17/12/1999 | 22/12/1999 | 1 |

Annotations :

- Tout client a un nom
- Le client n°3 n'existe pas dans la table CLIENT
- 2 clients ne peuvent avoir le même numéro
- Le type client est soit PARTICULIER soit PROFESSIONNEL
- La date de livraison est inférieure à la date de commande

Or, seule la ligne associée à la commande 103 est acceptable. Pour empêcher toutes ces incohérences, il est nécessaire d'enrichir la définition de la base par la mise en place des contraintes d'intégrité suivantes :

| Contraintes | Conséquences |
|--|---|
| Intégrité de relation (encore appelée intégrité d'entité) sur les champs NoClient de la table CLIENT et NoCommande de la table COMMANDE | Cette contrainte garantit des valeurs uniques et obligatoires pour les champs NoClient et NoCommande. Elle permet donc d'implanter la notion de clé primaire. |
| Intégrité de domaine : - sur le champ TypeClient avec la règle : = « PARTICULIER » ou « PROFESSIONNEL » - sur la table COMMANDE avec la règle : DateLivraison > DateCommande | Pour le champ TypeClient, seules les valeurs « PARTICULIER » ou « PROFESSIONNEL » seront tolérées. Pour la table COMMANDE, seules les lignes présentant une date de livraison supérieure à la date de commande seront tolérées |
| Intégrité de valorisation sur les champs NomClient et TypeClient de la table CLIENT et sur les champs DateCommande et NoClient de la table COMMANDE | Cette contrainte impose la présence d'une valeur pour les champs cités. |
| Intégrité référentielle sur le champ NoClient de la table COMMANDE | Cette contrainte impose que toute valeur du champ NoClient de la table COMMANDE appartienne à l'ensemble des valeurs du champ NoClient de la table CLIENT. |

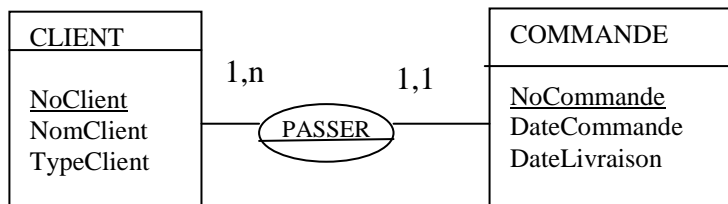
L'intégrité de relation

Attachée à un (ou plusieurs) champ(s) d'une table, cette contrainte garantit des valeurs uniques pour ce (ou cette combinaison de) champ(s). Elle permet donc d'implanter la notion de **clé primaire**.

L'intégrité de valorisation

Attachée à un champ cette contrainte permet d'imposer la saisie d'une valeur pour ce champ. Il n'est pas utile de définir cette contrainte pour un champ clé primaire car l'intégrité d'entité inclut l'intégrité de valorisation.

L'intégrité de valorisation définie sur un champ clé étrangère permet de représenter la cardinalité minimale 1 d'une association de type (1-1) ou (1-n). Ainsi pour indiquer qu'à toute commande est associée un client (ce qui revient à traduire la cardinalité minimale indiquée en gras sur le modèle conceptuel ci-dessous),



L'intégrité de domaine

Cette contrainte permet d'exprimer une condition qui porte sur un ou plusieurs champs de la table. Dans la table Commande pour exprimer la contrainte garantissant la cohérence entre la date de commande et la date de livraison on écrira l'ordre suivant :

(DateLivraison IS NOT NULL AND DateCommande < DateLivraison))

L'intégrité référentielle

Attachée à un champ "clé étrangère" cette contrainte garantit que toute valeur prise par ce champ appartienne à l'ensemble des valeurs de la clé primaire.

L'option ON DELETE CASCADE permet de supprimer une ligne d'une table ainsi que toutes les lignes liées dans une autre table.

Si l'on suppose que le contenu des tables CLIENT et COMMANDE est limité aux lignes suivantes :

| CLIENT | | | COMMANDE | | | |
|----------|-----------|---------------|------------|--------------|---------------|----------|
| NoClient | NomClient | TypeClient | N°Commande | DateCommande | DateLivraison | NoClient |
| 1 | DUBOIS | PROFESSIONNEL | 101 | 15/12/1999 | | 3 |
| 2 | DELAGE | PARTICULIER | 102 | 17/12/1999 | 18/12/1999 | 1 |
| 3 | DUPONT | PROFESSIONNEL | 103 | 17/12/1999 | 22/12/1999 | 1 |

La suppression du client Numéro 1 dans la table CLIENT pourrait :

- soit entraîner la suppression des commandes 102 et 103 dans la table COMMANDE si l'option ON DELETE CASCADE est mentionnée,
- soit entraîner un refus de suppression de ce client si l'option ON DELETE CASCADE n'est pas mentionnée ; dans ce cas il faut préalablement supprimer les commandes 102 et 103 dans la table COMMANDE pour ensuite pouvoir supprimer le client Numéro 1.

Prendre la branche pour la racine (proverbe chinois)

INTRODUCTION

La création des bases de données relationnelles s'appuie sur la théorie mathématique des relations, et la restitution des données est réalisée en utilisant l'algèbre relationnelle.

Une requête est une interrogation de la base de données. C'est en fait une demande de restitution des données. Ceci dit, tous les outils existant pour les requêtes répondent à un même objectif : **restituer à l'utilisateur toutes les informations qu'il souhaite, et uniquement celles là.**

I - L'ALGÈBRE RELATIONNELLE

L'algèbre a été inventée par E. CODD comme un ensemble d'opérations formelles sur les relations.

A - LES OPERATEURS BINAIRES

Ces types d'opérateurs s'appliquent à **deux tables** (ou relations) appelées tables opérandes dont ni la structure ni le contenu n'est modifiés. Ces opérateurs, à partir de deux tables permettent d'en construire une troisième.

1 - L'union

Elle s'applique à deux tables ayant impérativement la même structure. Le résultat est une table de même structure. (2 tables ayant des attributs identiques définis sur le même domaine).

Exemple : CLIENT (CodeCli, NomCli, AdresCli)
FOURNISSEUR (CodFour, NomFour, AdresFour)
TIERS = Union (CLIENT, FOURNISSEUR) = CLIENT **U** FOURNISSEUR.

2 - L'intersection

Elle s'applique à deux tables ayant impérativement la même structure. Le résultat est une table de même structure contenant les tuples qui sont présents dans chacune des deux tables opérandes.

Exemple : Dans une entreprise les tables SYNDICAT et DELEGATION permettent de connaître les employés qui ont été respectivement élus dans le bureau syndical ou dans la délégation du personnel. Elles ont toutes les deux la même structure.

SYNDICAT (MatriculeEmp, NomEmp, PrénomEmp, ServiceEmp, DateEmbaucheEmp)
DELEGATION(MatriculeEmp, NomEmp, PrénomEmp, ServiceEmp, DateEmbaucheEmp)

DEUXMANDATS = Intersection (SYNDICAT, DELEGATION)

DEUX MANDATS = SYNDICAT \cap DELEGATION

3 - La différence

Elle s'applique à deux tables ayant impérativement la même structure. Le résultat est une table de même structure contenant les tuples qui sont présents dans la première des relations opérandes sans être présents dans la seconde.

Exemple : SEULSYNDICAT = Différence (SYNDICAT, DELEGATION)
 SEULSYNDICAT = Minus (SYNDICAT, DELEGATION)

NB : pour cet opérateur, l'ordre dans lequel on écrit les relations opérandes a une importance capitale.

4 - Le produit

Il réalise le produit cartésien et est très peu utilisé dans la réalité, mais il est quelquefois utile. Il s'applique à deux tables quelles que soient leurs structures. Il peut notamment se réaliser sur deux tables n'ayant aucun attribut en commun.

5 - La division

Elle est très rarement employée. Sa mise en œuvre étant très délicate, tous les SGBDR ne l'incluent d'ailleurs pas dans les outils qu'ils comprennent.

6 - La jointure

Elle permet de regrouper les informations de deux tables de structures différentes et ayant un attribut en commun, même si cet attribut commun ne porte pas le "même nom" dans chacune des deux tables.

Exemple :

Table **EMPLOYE**

| CodeEmp | NomEmp | NumVéhicule |
|----------|--------|-------------|
| SI-1256 | YAO | 14 |
| SC-5305 | LACONE | 16 |
| SCP-3528 | DEGAHY | 05 |

Table **VEHICULE**

| NumVéhicule | Marque | Immatriculé |
|-------------|---------|-------------|
| 16 | Renault | 1015 AL |
| 05 | Peugeot | 209 AB |
| 14 | Peugeot | 5025 AC |

Attribut en commun : NumVéhicule

On souhaite connaître l'ensemble des informations du véhicule alloué à chaque employé.
 EMPLOYEEVEHICULE = Jointure (EMPLOYEE, VEHICULE, EMPLOYEE.NumVéhicule = VEHICULE.NumVéhicule)

B - LES OPERATEURS UNAIRES

Ils s'appliquent à *une seule table* opérande qui restera inchangée au niveau de sa structure et de son contenu.

1 - La sélection

Elle consiste à ne conserver que certains tuples d'une table, ceux qui satisfont à certaines conditions exprimées dans l'ordre de sélection.

Exemple 1 : R1 = Sélection (VEHICULE, Marque = "Peugeot ")

Exemple 2 : R2 = Sélection (EMPLOYEE, SalaireEmp <= 450 000 et Marque = "Peugeot ")

2 - La projection

Elle consiste à afficher ou à ne conserver que certains attributs d'une table, c'est-à-dire ceux dont la liste est donnée dans l'ordre de projection.

Exemple *LISTE1 = Projection (ETUDIANT, NomEtu, PrénomsEtu, ClasseEtu)*

3 - Le classement

Il permet de présenter les tuples d'une table dans un ordre déterminé (croissant ou décroissant) des valeurs de un ou plusieurs attributs de cette table.

Exemple : *C1 = Classement (EMPLOYE, NomEmployé ↗)*

4 - Les opérateurs de calcul

Ce sont des opérateurs qui vont permettre des calculs de tuples dans une table. Pour ces opérateurs, la table résultat est composée de tous les attributs de la table opérande auxquels est "ajouté" un attribut supplémentaire qui contiendra le résultat de l'opération.

NB : ces attributs ainsi ajoutés pourront alors servir dans des critères de sélection, à être projetés ou même à servir de base à d'autres calculs.

II - LE LANGUAGE SQL

Le **langage SQL** est une évolution commercialisée par IBM du langage SEQUEL. Il est perçu comme une expression agréable et très complète de séquences d'opérations relationnelles. Le moyen le plus normalisé pour transmettre la structure d'une base à un SGBD R est l'utilisation du **langage S.Q.L.** (Structured Query Language). Certains SGBDR (comme MS - ACCESS) sont dotés d'une interface graphique qui permettent de créer une base de données en s'affranchissant des contraintes syntaxiques du langage.

S.Q.L. est un langage de requête normalisée utilisée pour la gestion d'une base de données relationnelle ; le terme de gestion est très large puisqu'il englobe aussi bien l'administration de la base de données que son exploitation. En fonction de leur finalité, les ordres S.Q.L. sont regroupés dans différentes catégories, les plus connues étant le **L.D.D.** (Langage de Définition des Données) et le **L.M.D.** (Langage de Manipulation des Données). Ainsi pour une table, les ordres permettant de créer, de modifier ou de détruire sa structure appartiendront au L.D.D. alors que les ordres permettant d'interroger ou de modifier son contenu (ajout de nouvelles lignes, destruction de lignes existantes, ...) seront catalogués dans le L.M.D.

Enfin, malgré le fait que le langage S.Q.L. soit normalisé, les syntaxes des instructions diffèrent sensiblement d'un S.G.B.D. à l'autre.

A - LA PROJECTION

Une projection effectue l'extraction d'attributs spécifiés d'une relation puis élimine les tuples en double.

L'option **DISTINCT** permet de ne conserver que des distinctes, en éliminant les doublons. La liste des noms de colonnes indique la liste des colonnes choisies, séparées par des virgules. Lorsque l'on désire sélectionner l'ensemble des colonnes d'une table il n'est pas nécessaire de saisir la liste de ses colonnes, l'option ***** permet de réaliser cette tâche la liste des tables indique l'ensemble des tables (séparées par des virgules) sur lesquelles on opère. La condition logique permet d'exprimer des qualifications complexes à l'aide d'opérateurs logiques et de comparateurs arithmétiques

Syntaxe :

SELECT UNIQUE (« liste des attributs à afficher »)
FROM (« Nom de la table »)

NB : le mot clé **UNIQUE** permet d'éliminer les tuples en double. En fait, SQL n'élimine pas systématiquement les doublons, à moins que ceci soit explicitement demandé par le mot clé **UNIQUE**.

B - LA SELECTION

Une sélection est une combinaison d'une restriction suivie d'une projection.

Syntaxe

SELECT *
FROM. (« Nom de la table »)
WHERE (« Conditions ou critères de sélection »)
AND (autres critères)

NB : -Après le mot clé **SELECT** il est demandé de spécifier les attributs à afficher, ou généralement d'afficher tous les attributs par l'inscription du signe * qui signifie qu'on désigne tous les attributs

- La combinaison avec une projection s'effectue en remplaçant le signe * (qui désigne tous les attributs de la relation) par la liste d'attributs spécifiés de projection.

- De manière générale, la condition suivant la clause **WHERE** peut inclure les opérateurs de comparaison =, <, >, <=, >= pour composer les qualifications qui sont aussi obtenues à l'aide des opérateurs booléens **AND** (et), **OR** (ou), **NOT** (négation) et de parenthèse éventuelles pour indiquer l'ordre d'évaluation.

- Il est également possible de trier les résultats suivant l'ordre ascendant (mot clé **ASC**.) ou descendant (mot clé **DESC**) d'un ou de plusieurs attributs.

Exemple de Syntaxe :

SELECT (« Attribut 1, Attribut 2, Attribut 3, Attribut 4 »)
FROM (« Nom de la table »)
WHERE « Condition de sélection »
AND « Autre condition de sélection »
ORDER BY (Attribut 1 **ASC**, Attribut 3 **DESC**)

Il est possible en SQL d'organiser les tuples fournis en résultat grâce à la clause **ORDER BY**. La clause **ORDER BY** est suivie des mots clés **ASC** ou **DESC**, qui précisent respectivement si le tri se fait de manière croissante (par défaut) ou décroissante. l'exemple de la table voiture :

La sélection de toutes les colonnes de la table triées par ordre croissant de l'attribut Marque se fait par l'instruction :

SELECT *
FROM VOITURE
ORDER BY Marque **ASC**

La sélection de toutes les colonnes de la table triées par ordre croissant de l'attribut Marque, puis par ordre décroissant de l'attribut Compteur, se fait par l'instruction :

SELECT *
FROM VOITURE
ORDER BY Marque **ASC**, Compteur **DESC**

Les prédicats BETWEEN et IN permettent de vérifier respectivement qu'une valeur se trouve dans un intervalle ou qu'une valeur appartient à une liste de valeur :

La sélection de toutes les voitures d'occasion ayant un kilométrage inférieur ou égal à 100 000 km, mais supérieur ou égal à 30 000 km, (effectuée plus haut avec des comparateurs arithmétiques) peut se faire par l'instruction :

```
SELECT *  
FROM VOITURE  
WHERE Compteur BETWEEN 100000 AND 30000
```

La sélection des voitures d'occasion dont la marque est Peugeot ou Ford se fait grâce à l'instruction :

```
SELECT *  
FROM VOITURE  
WHERE Marque IN (Peugeot, Ford)
```

Le prédicat LIKE permet de faire des comparaisons sur des chaînes grâce à des caractères, appelés caractères jokers : Le caractère % permet de remplacer une séquence de caractères (éventuellement nulle). Le caractère - permet de remplacer un caractère (l'équivalent du « blanc » au scrabble ...). Les caractères [-] permettent de définir un intervalle de caractères (par exemple [J- M])

La sélection des voitures dont la marque a un E en deuxième position se fait par l'instruction :

```
SELECT *  
FROM VOITURE  
WHERE Marque LIKE " - E %"
```

Lorsqu'un champ n'est pas renseigné, le SGBD lui attribue une valeur spéciale que l'on note NULL. La recherche de cette valeur ne peut pas se faire à l'aide des opérateurs standards, il faut utiliser les prédicats IS NULL ou bien IS NOT NULL.

La sélection de toutes les voitures d'occasion dont le kilométrage n'est pas renseigné se fait par l'instruction :

```
SELECT *  
FROM VOITURE  
WHERE Compteur IS NULL
```

C – LA JOINTURE

Syntaxe 1

```
SELECT *  
FROM (« Table1, Table2 »)  
WHERE Table1.Clé primaire = Table2.Clé secondaire
```

Mais la jointure peut être exprimée d'une manière plus procédurale avec des blocs imbriqués reliés par l'opérateur IN

Syntaxe 2

```
SELECT *  
FROM ("relation 1")  
WHERE "clé primaire" IN  
    SELECT (clé étrangère)  
    FROM (relation 2)
```

D - LES CALCUL

Il existe aussi des possibilités de calcul avec des fonctions :

COUNT = comptage des valeurs

SUM = somme des valeurs

AVG = moyenne arithmétique des valeurs

MAX = valeur maximum

MIN = valeur minimum

NB : les fonctions peuvent être utilisées dans la clause **SELECT**

Exemple 1: SELECT AVG (Note)

FROM (Matière 2)

WHERE LibMatière = 'MERISE'

Exemple 2

SELECT *

FROM ("Relation")

GROUP BY ("Attribut") **HAVING** (fonction)

E – LE GROUPEMENT

Il peut être intéressant de regrouper des résultats afin de faire des opérations par groupe (opérations statistiques par exemple). Cette opération se réalise à l'aide de la clause **GROUP BY**, suivie du nom de chaque colonne sur laquelle on veut effectuer des regroupements.

Les principales fonctions pouvant être effectuées par groupe sont :

AVG : calcule la moyenne d'une colonne (ou de chaque regroupement si elle est couplée à la clause **GROUP BY**)

COUNT : calcule le nombre de lignes d'une table (ou de chaque regroupement...)

MAX : calcule la valeur maximale d'une colonne (ou de chaque regroupement...)

MIN : calcule la valeur minimale colonne (ou de chaque regroupement...)

SUM : effectue la somme des valeurs d'une colonne (ou de chaque regroupement...)

L'affichage des moyennes des compteurs par marque se fait par l'instruction :

SELECT Marque, **AVG** (Compteur) **AS** Moyenne

FROM VOITURE

GROUP BY Marque

La clause **HAVING** va de pair avec la clause **GROUP BY**, elle permet d'appliquer une restriction sur les groupes créés grâce à la clause **GROUP BY**.

L'affichage des moyennes des compteurs non nulles regroupées par marque se fait par l'instruction :

SELECT Marque, **AVG** (Compteur) **AS** Moyenne

FROM VOITURE

GROUP BY Marque

HAVING Compteur **IS NOT NULL**

Remarque : l'utilisation de **AS** permet de donner un nom à la colonne créée à l'aide de la fonction **AVG**

BIBLIOGRAPHIE

-Hubert Tardieu, Arnold Roch Feld, René Coletti, la méthode MERISE, principe et outils, Edition d'organisation (2000)

-Michel Divine, parlez-vous MERISE, EYROLLES 1994

-www.sousdoues.com

-www.commentcamarche.net