



FACULTÉ DES SCIENCES
JURIDIQUES, ÉCONOMIQUES,
ET SOCIALES - FES

Département Sciences Economie & Gestion

Licence Fondamentale Parcours Gestion

SYSTÈME D'INFORMATION

&

BASE DE DONNÉES

SEMESTRE 5

Professeur : Laila Alami

AN : 2019-2020

SOMMAIRE

Chapitre1 : Introduction aux systèmes d'information	5
1. Introduction	5
1.1 Données et Informations	5
1.2 Définition d'une Information	7
1.3 Définition d'un système.....	8
1.4 Définition d'un système d'information	8
2. Le concept de système d'information.....	9
2.1 Système Opérant.....	9
2.2 Système de pilotage	9
2.3 Système d'information	10
2.4 Exemple de concept de système d'information	11
3. Fonction d'un système d'information	12
3.1 La saisie.....	13
3.2 Le stockage	13
3.3 Le traitement	13
3.4 La restitution.....	13
3.5 Transmission et communication	14
4. Les systèmes d'information et les fonctions de l'entreprise.....	14
4.1 Comptabilité et Gestion des stocks.....	14
4.2 L'informatique bureautique.....	15
4.3 Ventes et Marketing.....	15
4.4 Bureau d'étude (Ingénierie).....	16
4.5 Production et fabrication.....	16
4.6 Logistique et relation client-fournisseur	17
4.7 Management et direction.....	17
Les SI et les fonctions de l'entreprise (Résumé).....	18
Chapitre2 : Initiation à la conception de système d'information.....	19
1. Le besoin de méthode	19

2.	Présentation de la méthode MERISE	19
3.	Les trois cycles de Merise	20
3.1.	Cycle d'abstraction de conception des systèmes d'information.....	20
3.2.	Cycle de vie	23
3.3.	Cycle de décision	23
Chapitre3 : Le modèle conceptuel de données.....		24
1.	Présentation	24
2.	Concepts manipulés.....	24
2.1.	Entité	24
2.2.	Association.....	24
2.3.	Propriété.....	25
2.4.	Dictionnaire de données.....	25
3.	Formalisme Graphique.....	26
4.	Occurrence	27
4.1.	Occurrence d'une propriété	27
4.2.	Occurrence d'une entité.....	27
4.3.	Occurrence d'une association.....	27
5.	Identifiant (ou Clé)	27
6.	Cardinalité	28
7.	Dimension d'une association.....	29
8.	Démarche à suivre	30
Chapitre4 : Le Modèle logique de données (relationnel)		31
1.	Introduction.....	31
2.	Le Modèle relationnel.....	31
3.	Règles de passage du modèle entité - association au modèle relationnel.....	32
3.1.	Une entité se transforme en une relation (table)	32
3.2.	Relation binaire aux cardinalités (X,1) - (X,n), X= {0 ou 1}	32
3.3.	Relation binaire aux cardinalités (X,n) - (X,n), X= {0 ou 1}	33
3.4.	Relation n-aire (quelles que soient les cardinalités).....	34
3.5.	Association Réflexive.....	35
3.6.	Relation binaire aux cardinalités (0,1) - (1,1).....	36
Chapitre5 : Le modèle physique de données		37
1.	Introduction.....	37
2.	Définition d'une Base de Données.....	37

3.	Définition d'un Système de Gestion de Base de Données.....	37
4.	Concepts de base	38
4.1.	Table, attribut et tuple	38
4.2.	Domaine	38
4.3.	Clé d'une relation	39
4.4.	Contrainte d'intégrité	39
Chapitre6 : Le langage de manipulation de données SQL.....		40
1.	Introduction	40
2.	RECHERCHE.....	40
2.1.	Format général d'une requête	40
2.2.	La clause distinct	42
2.3.	Trier un résultat.....	43
2.4.	Fonctions agrégats.....	44
2.5.	Les sous relations	45
2.6.	Les requêtes multi-relations (jointures)	45
2.7.	Les opérateurs ANY, ALL, IN	48
Les travaux dirigés - MCD & MLD		49
	Sujet 1: Gestion d'une Pressing	49
	Sujet 2: Agence immobilière.....	49
	Sujet 3 : Bibliothèque	49
	Sujet : Gestion de la scolarité universitaire.	50
Travaux Pratiques sous ACCESS		51
	Atelier1 : Création d'une Base de Données.....	51
	Atelier2 : Manipulation des données : Le langage SQL.....	57

Chapitre 1 : Introduction aux systèmes d'information

1. Introduction

Dans l'activité de chaque organisation, on utilise une quantité importante d'informations. Pour être utilisables pour la prise des décisions, ces informations doivent être stockées, diffusées, traitées et transformées.

Toute entreprise utilise de l'information pour son fonctionnement ou pour communiquer avec son environnement. Le gestionnaire doit être capable de traiter (ou accompagner le traitement) l'information et la rendre utile et rentable pour l'entreprise.

Ce cours pose les bases de la prise en main de la méthode Merise en passant par des modèles simple de différents cas de gestion.

Ce cours permettra à l'étudiant de faire connaissance avec le concept de *système d'information* et avec des méthodes d'analyse, développement et réalisation d'un système d'information informatisé.

A la fin de ce cours l'étudiant devra être apte à:

- ✓ Comprendre les différents concepts liés à la notion de système d'information,
- ✓ Maîtriser les principales étapes de conception d'une base de données relationnelle,
- ✓ Connaître un progiciel permettant la construction et la gestion d'une base de données relationnelle (Access),
- ✓ Connaître les principales étapes de conception, de développement et de réalisation d'un système d'information.
- ✓ Manipuler une base de données en utilisant des requêtes en langage SQL.

1.1 Données et Informations

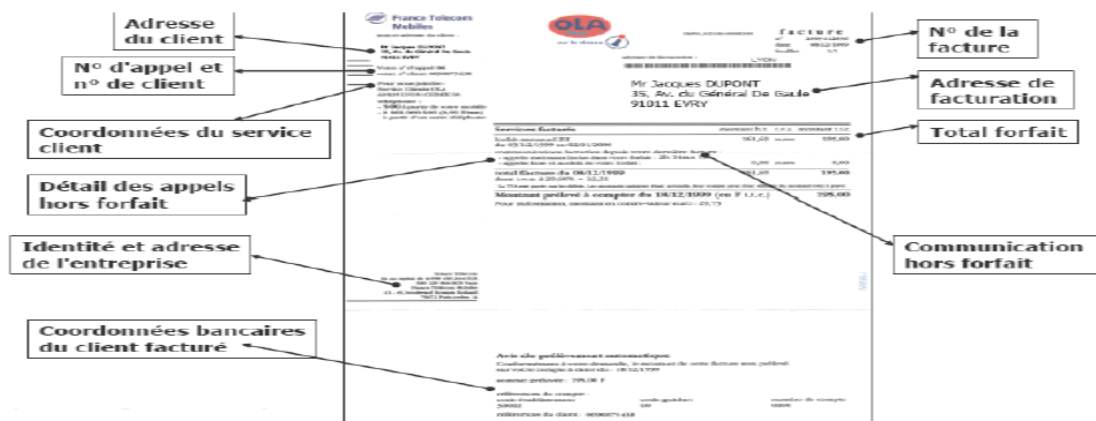
Donnée = signe + Code

Information = Donnée + modèle d'interprétation

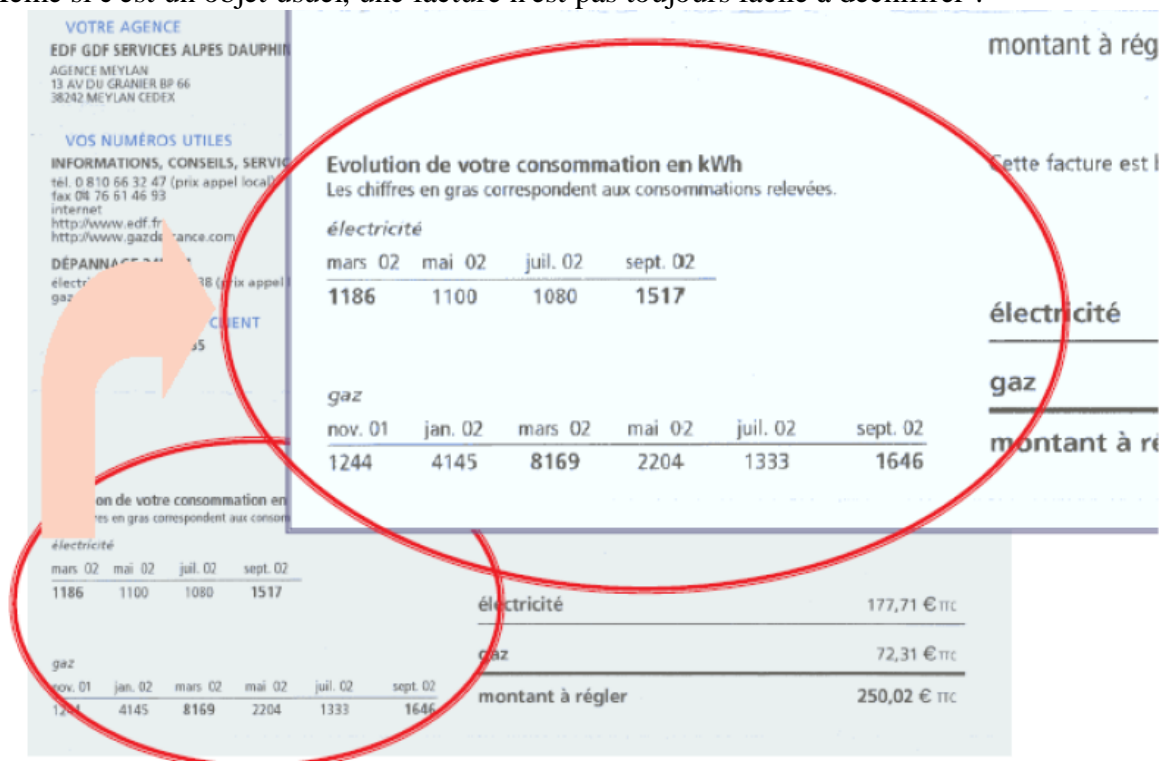
La codification des signes permet de représenter des données. Une série de signes tels que le '1' ou le 'T' deviennent une donnée lorsqu'on connaît la codification correspondante. Une information est quelque chose de beaucoup plus qu'une donnée brute. C'est une donnée qui a un sens, et ce sens vient d'un certain modèle d'interprétation. Une même donnée peut avoir plusieurs sens selon le modèle d'interprétation qui lui est associé. Ainsi, la donnée "10092018" peut être aussi interprété -par un anglo-saxon -comme la date du 09 Octobre 2018 (et non pas le 10 Septembre 2018).

On va voir à travers trois exemples différents comment la notion de modèle d'interprétation peut aller très loin, et qu'une information n'est pas toujours facile à appréhender si nous ne connaissons pas le contexte sémantique auquel réfère le modèle d'interprétation.

Exemples :



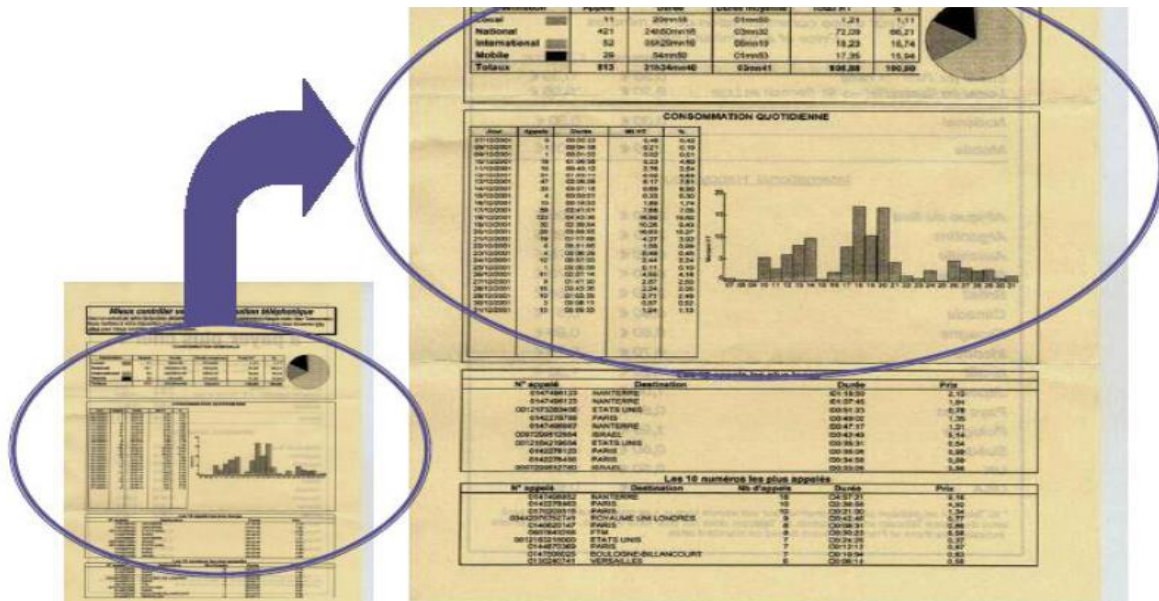
On a ici un exemple un peu plus sophistiqué de ce que peut être un modèle d'interprétation. Même si c'est un objet usuel, une facture n'est pas toujours facile à déchiffrer !



Cette deuxième facture est un peu plus riche que la précédente: on y retrouve non seulement la consommation du client (c'est des kwh chez EDF) mais surtout une information de nature plus décisionnelle, le récapitulatif de la consommation du client sur les 4 dernières factures.

L'objectif est de donner au client une vision complète de son utilisation des services de l'entreprise et toute l'information qui lui permet de mieux consommer. L'information apporte une plus-value évidente dans la communication avec le client, ce qui augmente son taux de satisfaction et sa fidélité.

Le 3ème exemple est une facture encore plus détaillée intégrant des graphiques et des courbes, ce qui est beaucoup plus facile à lire et à interpréter. Cette facture est un exemple typique de l'avantage concurrentiel que peut apporter le SI à une entreprise.



Le développement d'une stratégie commerciale et/ou industrielle nécessite toujours des développements parallèles dans le SI de l'entreprise. Ce qui est bien sûr normal, car le SI est le reflet du fonctionnement présent, passé -et parfois futur -de l'entreprise. Toute modification apportée à ce fonctionnement implique une modification du SI.

1.2 Définition d'une Information

En Latin le terme « Informare » signifie donner une forme, une structure, une signification. On peut dire donc que :

- une information est un élément de connaissance susceptible d'être représenté à l'aide de conventions pour être traité, conservé et communiqué.
- ou encore l'information est une donnée pertinente que le système nerveux central est capable d'interpréter.
- En théorie, l'information diminue l'incertitude.
- En théorie de la décision, on considère même qu'il ne faut appeler information que ce qui est susceptible d'avoir un effet sur nos décisions.

Exemple :

Une bibliothèque possède un grand nombre d'ouvrages, des revues, des livres et des dictionnaires. Nous cherchons un cours complet sur la théorie de l'information. Tout d'abord, il est logique que nous ne trouvions pas ce dossier dans des ouvrages d'arts ou de littérature; nous venons donc d'obtenir une information qui diminuera notre temps de recherche. Nous avons précisé que nous voulions aussi un cours complet, nous ne le trouverons donc ni dans une revue, ni dans un dictionnaire. Nous avons obtenu une information supplémentaire (nous cherchons un livre), qui réduira encore le temps de notre recherche.

Classifications possibles des informations

Selon l'origine (la provenance)

- Informations internes : générées au sein de l'entreprise. Exemple, le montant d'une facture pour un client.
- Informations externes : prises dans l'environnement de l'entreprise et utiles au bon fonctionnement de l'entreprise. Exemple, le taux de TVA, les prix des fournisseurs.....

Selon la durée de vie de l'information

- Informations volatiles : ayant une courte durée de vie. Exemple, la personne X est arrivée à son poste de travail à 8h15 et partie à 16h30.
- Informations persistantes : représente la mémoire de l'entreprise; stockées (historisées) et sécurisées. Exemple, les adresses des clients et des fournisseurs.

Selon le traitement appliqué

- Informations brutes : utilisées sans modification préalable. Exemple, poids d'une pièce fabriquée dans un atelier.
- Informations calculées : à partir des informations brutes ou d'autres informations obtenues par calculs ou règles de gestion. Exemple, les montants des salaires.

1.3 Définition d'un système

Un système est un ensemble d'éléments reliés entre eux en interactions dynamiques et constituant un tout cohérent et fonctionnel compris dans un ensemble plus grand. En Latin et en Grec, le mot « système » veut dire combiner, établir, rassembler. Généralement un système est constitué de composants organisés ensemble dans le but de faciliter le flux d'information, de matière, d'énergie.....

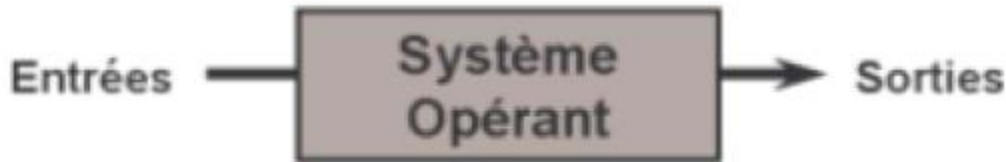
1.4 Définition d'un système d'information

La récolte, le stockage et la diffusion de l'information est devenue une industrie à part entière. Et toute organisation quelle qu'elle soit, doit consacrer une partie de son effort à récolter, traiter, stocker et diffuser l'information issue de son propre fonctionnement. C'est la tâche principale du système d'information, qu'on va tenter de définir dans les pages qui suivent.

2. Le concept de système d'information

2.1 Système Opérant

Représentation abstraite d'une organisation



Système opérationnel où :

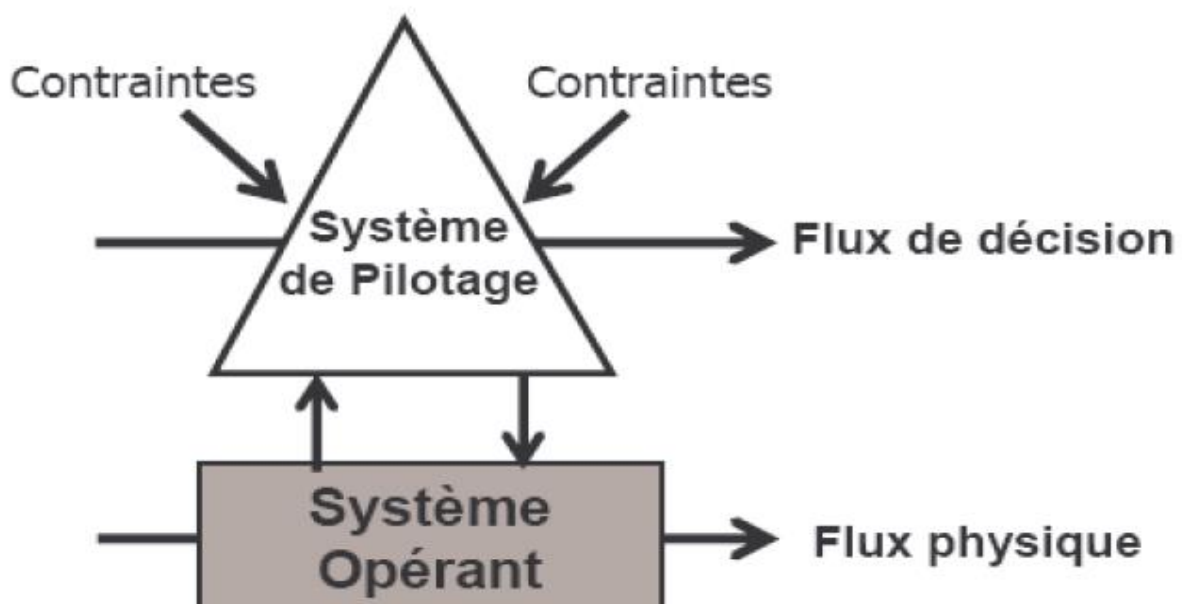
- Les matières premières sont transformées,
- Les produits finaux sont fabriqués.

Pour introduire d'une manière un peu formelle le concept de système d'information, on va recourir à ce qu'on appelle la vision systémique d'une entreprise. On distingue d'abord le système opérant où les produits finaux sont fabriqués à partir d'une certaine matière première.

On réduit l'organisation à une sorte d'usine, qui travaille sur la matière première pour fournir un produit final.

Exemple : Si l'on considère une école de management (qui est un exemple typique d'une organisation), le système opérant est constitué des services et des départements qui organisent le concours d'entrée, les délibérations et les jurys, les cours, les examens, les corrections de copies. Les élèves qui suivent les cours, qui passent des examens ... etc. font aussi partie de ce système opérant. C'est l'obtention du diplôme après cinq ans d'étude qui est - entre autre - le produit final du système opérant à l'école. Le flux physique est donc constitué des élèves, des concours, des cours, des examens, des résultats scolaires, des rapports publiés, ... etc.

2.2 Système de pilotage



La mission de la direction :

- Fixer des objectifs,
- Contrôler leur réalisation
- Corriger (selon des contraintes prédéfinies)

Toute organisation est pilotée par une direction, une équipe dirigeante.

Ce système de pilotage a pour mission de conduire l'organisation vers des objectifs qui lui sont fixés, et de vérifier que ces objectifs ont bien été atteints. Ce qui nécessite souvent un contrôle continu du fonctionnement du système opérant et d'éventuelle modifications (recrutement, investissement, nouveaux développements, ... etc.) à apporter au système opérant.

Parallèlement donc au flux physique, il y a un flux de décision. Ce flux correspond aux décisions prises par la direction de l'organisation pour que celle-ci fonctionne dans les meilleures conditions et puisse atteindre ses objectifs. Et toute organisation est soumise à des contraintes extérieures et intérieures qui contraignent son action et l'empêche d'évoluer librement.

En considérant toujours la même école, les décisions prises par le système de pilotage concernent le nombre d'élèves qui seront admis chaque année à l'école de Management, le seuil d'admissibilité, la définition des programmes des études, la définition des modalités d'examen et le règlement des études, les développements futures de l'école tels que l'augmentation du nombre d'élèves création de nouvelles filières, les ouvertures de postes de recrutement, les investissements en matériel et en logiciel, ... etc.

2.3 Système d'information

Et c'est dans ce contexte qu'apparaît le système d'information. Ce sous-système de l'organisation s'occupe de récolter l'information, de la stocker, de la traiter et de la diffuser dans le système opérant et dans le système de pilotage.

Dans le système opérant, cette information va permettre à celui-ci de fonctionner. Car chaque individu et chaque tâche ont besoin d'être informés sur le flux physique qui la traverse.

En général, cette information est très détaillée, ne concerne qu'un petit élément de l'organisation, et elle est tournée vers le présent.

Dans le système de pilotage, l'information va permettre à celui-ci de prendre les bonnes décisions en étant constamment informé de ce qui se passe dans le système opérationnel.

Cette information a tendance à être très synthétique, elle concerne une grande partie de l'organisation (si ce n'est toute l'organisation, tel que le Chiffre d'Affaire annuel), et elle est tournée vers le passé et/ou le futur.

La tâche principale du SI est donc de fournir un flux d'information qui d'une part, reflète le plus fidèlement possible le flux physique, et d'autre part fournit au système opérationnel les éléments nécessaires pour son fonctionnement quotidien et au système de pilotage les éléments nécessaires à une prise correcte de décision.

Ainsi, le flux d'information est une image du flux physique. Il représente sous une forme plus ou moins réduite, tous les événements survenus dans le système opérant ainsi que tous les éléments d'information qui permettent de traiter ces événements.

Cette image est forcément une réduction de la réalité, elle ne concerne que les aspects pertinents ayant une incidence et/ou un rôle dans le fonctionnement de l'organisation.

En reprenant l'exemple de l'école, on trouvera dans son SI toutes les informations sur les élèves qui y sont inscrits : nom prénom, adresse des parents, date de naissance, parcours scolaire précédent, n° sécurité sociale, ... etc. mais on ne trouvera pas ni la couleur des yeux ni le groupe sanguin (qui est une information non pertinente pour le fonctionnement de l'école).

Plus précisément, on dit que dans le SI il y a des modèles de la réalité organisationnelle. Ces modèles ont été construits par ceux qui mettent en place le SI. La validité et la pertinence de ces modèles sont indispensables au fonctionnement du SI lui-même, et elles garantissent la qualité de l'information fournie.

Pour conclure :

La construction des modèles dans un SI est une phase très importante dans la construction de celui-ci. Cette tâche relève de l'ingénierie des SI, et on utilise le terme "conception de SI".

Un système d'information est un ensemble de moyens humains, techniques et organisationnels pour fournir en temps et en heure toute l'information nécessaire au fonctionnement de l'organisation (autant le système opérant que le système de pilotage).

Pourquoi dit-on que le SI est un artefact (un objet artificiel) ?

Par ce que l'organisation doit consacrer une partie de ses ressources et de son énergie à gérer l'information, alors que ce n'est pas sa tâche initiale.

En reprenant notre exemple, l'école doit consacrer une partie de ses ressources (humaines et techniques) pour gérer l'information concernant son fonctionnement quotidien. Cette tâche occupe plusieurs personnes et consomme des ressources informatiques, mais elle n'est pas le métier original de l'école qui est de former des élèves, de délivrer des diplômes et de mener des activités de recherche.

Cette tâche qui ne fait pas partie du cœur de métier des organisations (sauf pour le cas particulier des organisations spécialisées dans la conception et le développement des SI), est parfois sous-traitée. On appelle cette pratique le "out-sourcing" ou l'infogérance. Ce qui permet à l'organisation de se concentrer sur son cœur de métier, avec quand même le risque de devenir trop dépendante d'un acteur extérieur.

2.4 Exemple de concept de système d'information

Exemple-1 : Ecole de Management

Flux physique

Les élèves, les concours, les cours, les examens, des résultats scolaires, des rapports publiés, les enseignants, les départements ...etc

Flux d'information

Nom, prénom, adresse des étudiants, des parents des enseignants, date de naissance, parcours scolaire, n° de l'étudiant, ...etc

Flux de décision

Nombre d'élèves qui seront admis, le seuil d'amissibilité, la définition des programmes des études, la définition des modalités d'examen et le règlement des études, les développements futures de l'école tels que l'augmentation du nombre d'élèves, création de nouvelles filières, les ouvertures de postes de recrutement, les investissements en matériel et en logiciel, ...etc

Exemple-2 : ONCF (Compagnie des chemins de Fer)

Flux physique

- Les trains, les wagons, les voyageurs, les billets, les réservations, les départs et les arrivés, les trajets (par exemple : Paris - Lyon), le paiement d'un billet, ... etc.

Flux d'information

- Le trajet (n° de trajet, nom ville départ et nom ville d'arrivée, fréquence, heure départ et arrivée), n° de billet, n° de réservation, n° siège réservé, heure effective d'un départ et heure effective d'une arrivée, ... etc.
- Des informations plus synthétiques tels que la répartition du nombre de voyageurs sur un trajet selon les périodes de l'année, la répartition des voyageurs selon les trajets pendant les périodes de Noël des 5 dernières années, la répartition du CA (chiffre d'affaire) d'un trajet selon les jours de la semaine pour les 3 derniers mois, ... etc.

Flux de décision

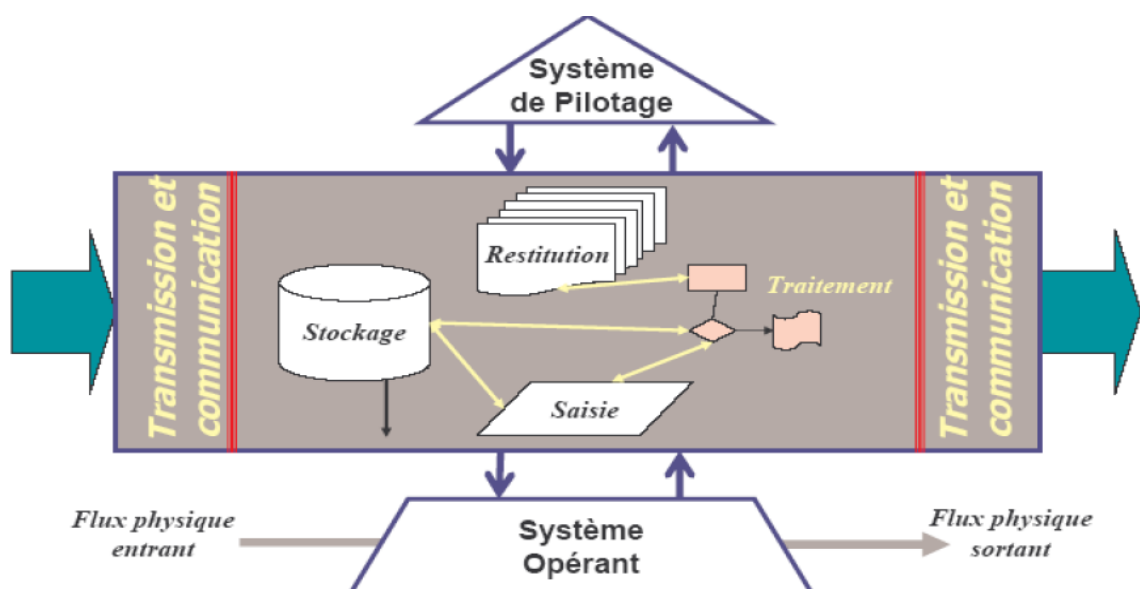
- Définition de la grille tarifaire, définition des trajet pour une période donnée, ouverture de nouveaux trajets (par exemple : un Paris-Berlin hebdomadaire le Vendredi soir à 22h), la fermeture de certains trajets, l'achat de nouveaux équipement, ... etc.

Contraintes

- La réglementation, les cycles économiques, l'inflation, les syndicats, ... etc.

3. Fonction d'un système d'information

Ce schéma résume les 5 fonctions de base d'un système d'information. Remarquez que la communication s'effectue autant avec les systèmes de pilotage qu'avec le système opérant.



3.1 La saisie

C'est la fonction la plus basique. Dans le passé, il y avait les opérateurs de saisie qui passaient leurs journées à saisir des tonnes d'information sur des claviers ... Cela existe encore malgré l'introduction massive des lecteurs de code barre, des logiciels de reconnaissance vocale et de reconnaissance de caractère.

3.2 Le stockage

La technologie a évolué très vite ces 10 dernières années. On peut facilement maintenant graver un CD ou même un DVD, et les disques durs ont des capacités de plus en plus grandes.

3.3 Le traitement

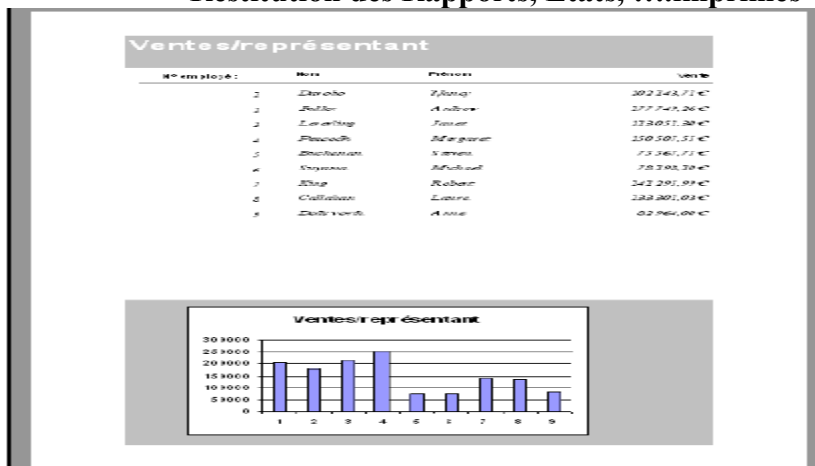
Le principe de fonctionnement des ordinateurs n'a pas beaucoup changé depuis son invention dans les années 40. Mais c'est la vitesse de traitement des processeurs qui atteint des sommets inimaginables il y a seulement quelques années. Les derniers processeurs atteignent la vitesse de traitement de 3 milliard de cycle de base / seconde (Pentium à presque 3 gigahertz) !

Cette étape consiste à créer de nouvelles informations à partir de celles existantes avec des opérations de tri, calculs, regroupements, ...

Cette fonction utilise des ordinateurs, serveurs, systèmes d'exploitation, logiciels d'application,...

3.4 La restitution

Electronique : protocole Web (en Internet, extranet, Intranet) Restitution des Rapports, Etats, ...imprimés



L'impression sur papier reste le moyen le plus commode de restitution des informations. On reçoit tous des relevés mensuels d'activité bancaire, des factures d'abonnement téléphonique, des fiches de paie, ... etc.

Mais les écrans d'ordinateurs ont aussi envahi notre quotidien, de même qu'on voit apparaître des machines qui parlent.

La restitution peut prendre des formes extrêmement sophistiquées à l'aide par exemple des images de synthèse.

3.5 Transmission et communication

Toute transmission nécessite un support (câble coaxial, fibre optique, liaison hertzienne, liaison satellite) et un protocole de communication pour codifier les données sous une forme qui peut être transmise. On distingue ainsi la transmission de la voix et la transmission des données, comme on distingue les protocoles de bas niveau (tel que le TCP/IP) et les protocoles de haut niveau (tels que les formats pour l'échange électronique de donnée).

4. Les systèmes d'information et les fonctions de l'entreprise

Quelque soit l'entreprise (petite, moyen ou grande) et quelque soit son domaine d'activité (production, service, commercialisation), il y a des fonctions communes :

- Les ressources humaines qu'il faut recruter, former, rémunérer, gérer
- La comptabilité et la finance pour calculer les dépenses, les recettes, la rentabilité, le taux d'endettement, etc.
- La production où les produits (voiture, aliments, services bancaires, cours de formation, etc.) sont "fabriqués"; et où on doit planifier, organiser, gérer le stock des produits, les processus de fabrication, la livraison, etc.
- La vente et le marketing, où le contact avec le client a lieu pour le démarcher et lui vendre les produits; et où on doit gérer la relation avec le client et avoir une information précise sur les produits, les tarifs, les promotions, la marge de manœuvre, etc.
- L'ingénierie, où les nouveaux produits sont imaginés, conçus, testés et évalués et où on se préoccupe des processus de fabrication et des méthodes de travail; on a besoin ici d'une information plus spécifique selon la nature du produit conçu.

On va donc regarder pour toutes ces fonctions ce que l'informatique peut apporter et les spécificités des besoins en terme de système d'information.

4.1 Comptabilité et Gestion des stocks

Chaque entreprise doit gérer ses stocks de matières premières ainsi que ses stocks de produits finis. La comptabilité est un élément clé dans toute organisation, elle s'adapte le plus facilement à l'utilisation de l'informatique (pourquoi ?).

Comptabilité et gestion des stocks

C'est la principale porte d'entrée de l'informatique dans l'entreprise.

Les fonctions comptables sont les premières à être automatisées :

- Calcul et édition des fiches de paie.
- Suivi des achats, des livraisons, des ventes.
- Calcul et édition des factures.
- Etats comptables et gestion financière.



Connecticut Mutual Life Insurance : Un entrepôt de la taille d'un stade de football !

L'exemple auquel fait référence la photo est celui d'une entreprise d'assurance dans l'état de Californie aux Etats-Unis. Il y avait chaque jour une navette entre le siège de l'entreprise et l'entrepôt dans lequel sont stockés tous les dossiers de clients, les déclarations de sinistres, les demandes de remboursement, les contentieux non réglés, ... etc. On se demande combien de personnes étaient affecté à la gestion de cette masse énorme d'information !

Cette entreprise a démarré un projet d'informatisation de la gestion des dossiers, avec notamment un archivage sur CD-ROM ou disques dur externes.

4.2 L'informatique bureautique

Elle remplit 3 tâches essentielles :

- Le secrétariat et la production de l'écrit, grâce au logiciels de traitement de texte, aux tableurs, et autres logiciels "Office".
- L'organisation et la gestion des informations locales à un individu.
- La communication grâce principalement à la messagerie électronique, mais aussi le groupware et le travail en groupe.

La micro-informatique a ainsi bouleversé le travail des secrétariats, et le PC est devenu l'outil de travail central de tout cadre dans l'entreprise.

4.3 Ventes et Marketing

Pour un vendeur, il est indispensable de bien connaître :

- les produits, leurs prix et ses possibilités de manœuvre;
- la disponibilité des produits;
- le profil du client : volume d'achat, habitudes de consommation, préférences, ...

Etude de cas

Près de Grenoble, une usine de grenaille de fer fournit plus de 8000 entreprises réparties dans 70 pays différents. Les commerciaux doivent connaître aussi bien la qualité des différentes grenailles que les machines des clients dans lesquelles elles seront utilisées. Résultat : 17000 fiches mises à jour annuellement par 70 vendeurs ... Elles sont traitées sur le mini-ordinateur (AS400 d'IBM) du siège de l'entreprise. La fiche de renseignement est mise au point par le service informatique. Elle comprend les noms et adresses des clients, leur consommation et secteurs d'activité, leurs équipements et leur mode d'utilisation de la grenaille.

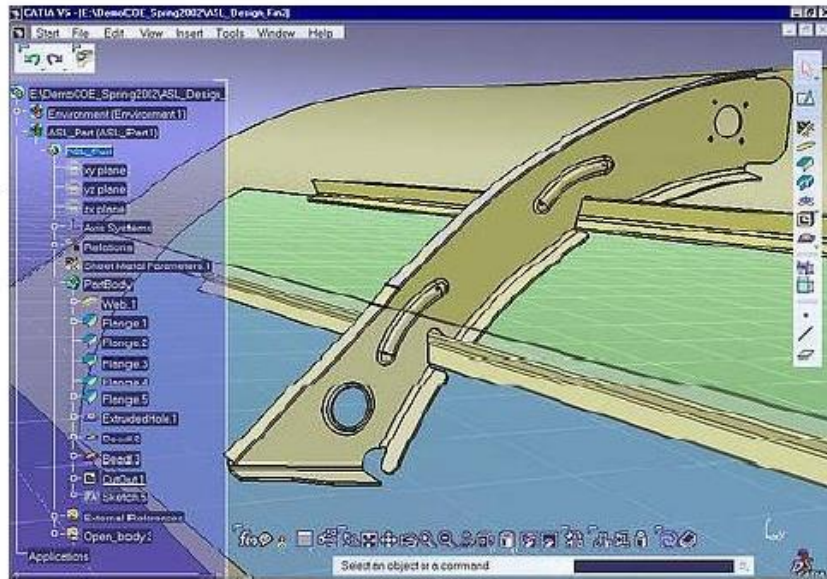
Dans certain cas, notamment pour les vendeurs qui se déplacent chez les clients, on utilise de plus en plus des connections mobiles au système d'information. Cela peut être un ordinateur + un accès par ligne téléphonique mobile, ou plus simplement un assistant électronique (ou PDA en anglais) dans lequel sont chargé régulièrement toutes les informations dont le vendeur a besoin (base clientèle, les produits avec les disponibilités, les prix, les marges de manœuvre ... etc..).

4.4 Bureau d'étude (Ingénierie)

Exemple

Né des besoins précis du constructeur d'avions français, ce logiciel s'adresse dès le début des années 90 aussi bien aux industriels de l'automobile qu'aux architectes. Son premier client est IBM, son deuxième est Boeing ... C'est le logiciel de CAO le plus utilisé au monde par les industriels de l'aéronautique et de l'automobile. En 1990, il équipait plus de 1500 gros ordinateurs et près de 3000 stations de travail IBM !

Depuis, son concepteur a décidé d'en faire un plate-forme de développement pour fabriquer des logiciels plus spécialisés. Devenu un généraliste, CATIA sera utilisé aussi bien par les mécaniciens, les stylistes ou les architectes.



4.5 Production et fabrication

L'ordinateur dans l'atelier et dans l'usine pour :

- Baisser les prix de revient et augmenter la productivité en automatisant et en "robotisant" les tâches de production.
- Fabriquer le bon produit au bon moment.
- Fabriquer de produits en masse et à la demande (selon l'idée de "mass customisation").

Les robots ont envahi les usines depuis longtemps, et il y a certains produits (tels que les voitures ou les microprocesseurs) qui sont en grande partie fabriqués par des machines pilotées par des ordinateurs.

La tendance actuelle est la fabrication de produits sur mesure et à la demande du client. Pour atteindre cet objectif, il faut une interconnexion optimale des différents systèmes d'information de l'entreprise pour que la commande du client puisse arriver le plus vite possible à l'usine qui va fabriquer (en grande partie automatiquement) le produit demandé et l'expédier ensuite chez le client.

La photo suivante illustre une entreprise japonaise qui fabrique en masse des bicyclettes sur mesure. Comme chez le tailleur, le vendeur va prendre une série de mesures et faire essayer le client un prototype qu'on peut ajuster selon la taille, le poids et la position favorite du client.

Ces mesures permettent de construire une maquette numérique du produit demandé, et de transmettre ce "cahier des charges numérique" à une usine robotisée qui va fabriquer ce produit unique pour un client en particulier.

4.6 Logistique et relation client-fournisseur

La relation avec le client est devenue un enjeu majeur pour la majorité des entreprises. Ce qu'on appelle le CRM connaît un très fort engouement, et les entreprises dépensent des sommes énormes pour obtenir ce nouveau or gris : l'information sur le client !

Bien sûr, l'information a toujours été là, mais elle n'était pas exploitable car éparpillée dans des systèmes hétérogènes et des formats incompatibles. Les propositions commerciales des entreprises ne pouvaient pas s'adapter au profil de chaque client. On essayait plus tôt de construire une offre commerciale qui puisse satisfaire le maximum de clients.

Ce n'est plus le cas aujourd'hui. Si vous passez souvent des coups de téléphone à l'étranger (ou à un pays en particulier), votre opérateur va vous proposer rapidement une offre adaptée pour vous fidéliser et augmenter la durée et/ou la fréquence vos consommations.

Si vous achetez des livres ou des CDs chez Amazon.com, au bout de quelques achats le système vous fera des propositions d'ouvrage ou d'œuvres qui auront une grande chance de vous intéresser ...

L'EDI (Echange de Données Informatisée ou Electronic Data Exchange) est une technologie qui existe depuis une dizaine d'années. Elle est apparue dans le secteur de l'industrie textile, ou un format standard d'échange de données est apparu.

Grâce à ce format, les clients et les fournisseurs de la chaîne industrielle du textile ont pu connecter leurs systèmes d'information respectifs. Les échanges de données sont devenus beaucoup plus fluides, ont permis d'économiser de l'argent et surtout beaucoup de temps.

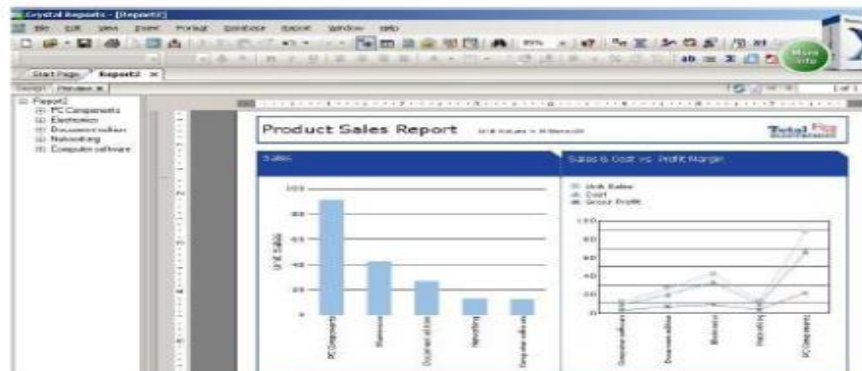
4.7 Management et direction

C'est le système d'information du "patron" :

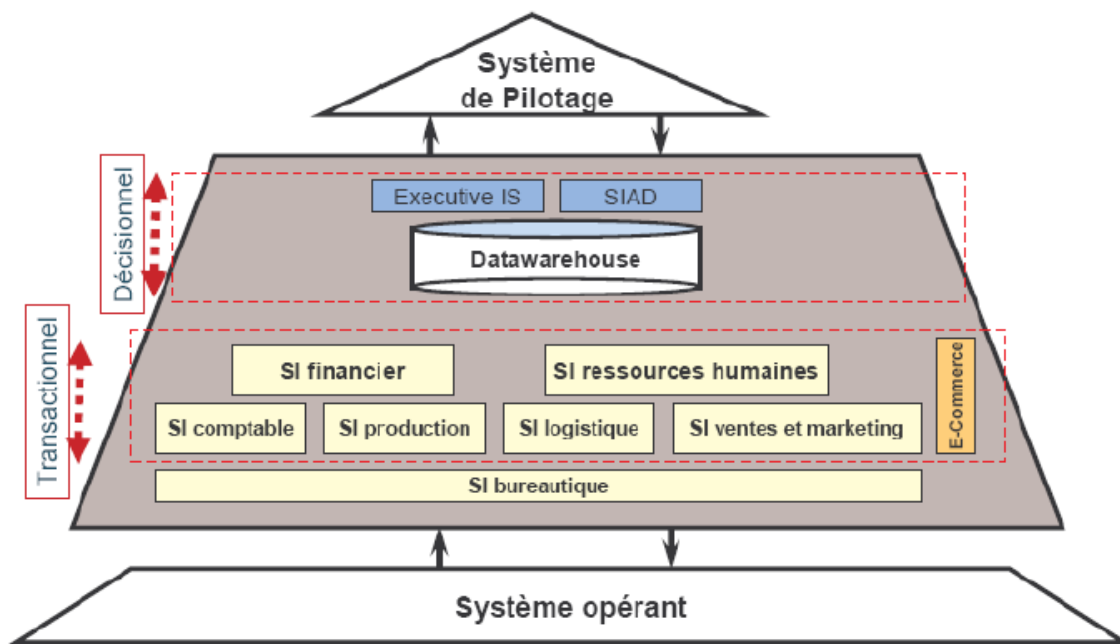
- Tableaux de bord basés sur des indicateurs clés.
- Objectif : Assister et faciliter la prise de décision.
- Diverses terminologies : EIS (*Executive Information System*), SIAD (*Système Interactif d'Aide à la Décision*), *SI stratégique*.

Exemple :

- *Business Objects*



Les SI et les fonctions de l'entreprise (Résumé)



Ce schéma fait la synthèse de la vision fonctionnelle du Système d'Information. On y voit que le SI à deux facettes :

- La première facette est tournée vers le système opérant et dont la finalité est d'assister le fonctionnement opérationnel et la gestion quotidienne de l'organisation. Ce sont les différents SI fonctionnels qu'on a vu, et qui sont accessibles depuis des postes de travail de type bureautique.

Les applications informatiques qui supportent ces SI sont de type transactionnel, et utilisent intensivement des bases de données de production. Elles traitent de grandes masses de données qui sont faiblement agrégées (c.à.d ne concerne qu'un élément atomique de l'organisation tel que la commande d'un client) et qui sont accessibles en quasi temps réel.

Ces applications informatiques sont devenues de plus en plus ouvertes vers l'extérieur pour le commerce électronique (vente et achat en ligne) et le travail collaboratif (intranet, extranet).

- La seconde facette est tournée vers le système de pilotage et dont la finalité est d'assister la direction de l'entreprise et la prise de décision des dirigeants.
- Ce sont des applications informatiques récentes qui font appel à des technologies nouvelles (entrepôts de données ou datawarehouse et SIAD). Elles manipulent des données où le degré de précision est relatif (tel que la variation du panier moyen dans un hypermarché par mois sur les 24 derniers mois), et qui sont fortement agrégées (elles concernent de grand ensemble d'éléments de l'organisation, il faut par exemple calculer la moyenne pour plusieurs centaines de milliers de données).

Ces systèmes sophistiqués sont coûteux, ils permettent aux dirigeants d'avoir une vision synthétique et condensée de l'activité de l'entreprise.

Le but est faciliter les projections et les simulations, en intégrant notamment des données externe au SI de l'entreprise (tel que les données des concurrents, ou la corrélation avec l'indice de dévolution des prix). La finalité ultime de ces systèmes est d'aider le dirigeant dans la prise de décision.

Chapitre2 : Initiation à la conception de système d'information

1. Le besoin de méthode

La conception d'un système d'information n'est pas évidente car il faut réfléchir à l'ensemble des informations 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.

Selon le petit Robert, une méthode est « Un ensemble de démarches raisonnées, suivies, pour parvenir à un but ».

Tel est bien l'objet des méthodes de conception et de développement des SI : présenter une démarche et un ensemble de modèles permettant de définir et mettre en place un nouveau système.

Les avantages d'utilisation d'une méthode sont :

- ▶ Une meilleure gestion et un meilleur suivi du projet en cours.
- ▶ Une facilité de dialogue entre les membres du projet et ensuite avec les utilisateurs du fait qu'elle utilise un formalisme commun et constitue une documentation.

Il existe plusieurs méthodes d'analyse, la méthode la plus utilisée en France étant la méthode **MERISE**.

2. Présentation de la méthode MERISE

MERISE est une méthode de conception, de développement et de réalisation de projets informatiques. 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.

La méthode MERISE date de 1978-1979, et fait 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.

3. Les trois cycles de Merise

L'identification de ces cycles se base sur la trialectique fondamentale de l'**ETRE**, du **FAIRE** et du **DEVENIR**, le sésame de la connaissance de l'objet. En effet, expliquer un concept ou un objet, nous ramène à répondre à trois questions fondamentales : ce qu'il est ? Ce qu'il fait ? Ce qu'il devient ?

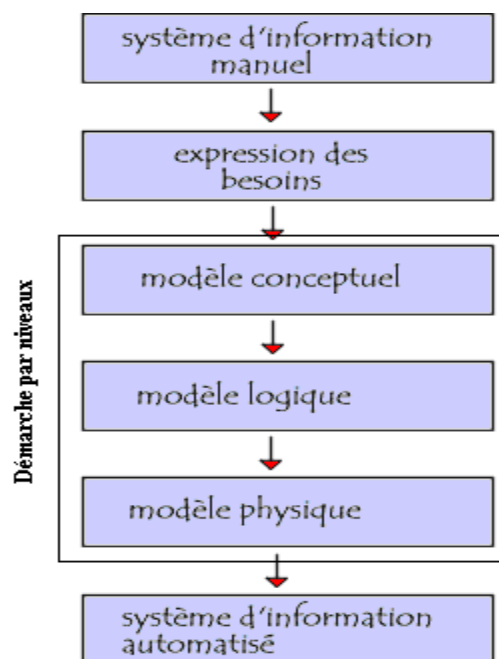
Par conséquent, la définition de l'objet pondère une définition ontologique¹ (*ce que l'objet est* - la dimension de l'ETRE), une définition fonctionnelle (*ce que l'objet fait* - la dimension du FAIRE) et une définition génétique (*ce que l'objet devient* - la dimension du DEVENIR).

En extrapolant aux systèmes d'information, ces trois dimensions se transforment en des cycles. On aura ainsi le **cycle d'abstraction** (*ce que le système d'information est*), le **cycle de décision** (*ce que le système d'information fait*) et le **cycle de vie** (*ce que le système d'information devient*).

3.1. Cycle d'abstraction de conception des systèmes d'information

La conception du système d'information se fait par étapes, afin d'aboutir à un système d'information fonctionnel reflétant une réalité physique. Il s'agit donc de valider une à une chacune des étapes en prenant en compte les résultats de la phase précédente. D'autre part, les données étant séparées des traitements, il faut vérifier la concordance entre données et traitements afin de vérifier que toutes les données nécessaires aux traitements sont présentes et qu'il n'y a pas de données superflues.

Cette succession d'étapes est appelée *cycle d'abstraction pour la conception des systèmes d'information* :



¹ ont - onto tiré du grec *ōn, ontos* - l'être, ce qu'il est, donc *ontologique* = relatif à l'être, en tant que tel

➤ **L'expression des besoins :**

C'est une étape consistant à définir ce que l'on attend du système d'information automatisé, il faut pour cela :

- Faire l'inventaire des éléments nécessaires au système d'information
- Délimiter le système en s'informant auprès des futurs utilisateurs

➤ **Le modèle conceptuel :**

A ce niveau on définit les classes d'objets ou d'événements et les règles de comportement de ceux-ci. Les contraintes liées à l'organisation de l'entreprise ne sont pas prises en considération. On liste et on décrit les règles de gestion en répondant à la question *Quoi faire ?* Au niveau de données on utilise l'approche ENTITE - ASSOCIATION. Au niveau de traitements on utilise l'approche EVENEMENT - RESULTAT.

Dans ce modèle on commence par créer le **MCC** (Modèle conceptuel de la communication) qui définit les flux d'informations à prendre en compte. L'étape suivante consiste à mettre au point le **MCD** (Modèle conceptuel des données) et le **MCT** (Modèle conceptuel des traitements) décrivant les règles et les contraintes à prendre en compte.

➤ **Le modèle logique ou organisationnel :**

A ce niveau, on identifie les ressources employées pour la mise en place de l'ensemble de règles de gestion identifiées au niveau conceptuel (choix du SGBDR pour le système d'information). En répondant aux questions Qui ? Où ? Quand ? On mettra en accord les règles de gestion et la structure organisationnelle de l'entreprise (Exemple : Qui reçoit les commandes passées par les clients ? Qui fait la facturation ? Quand doit-on éditer une facture ? Quel service charger pour éditer la facture ?

Les événements ne sont plus des événements temporels mais des événements à dominante spatiale. Les données se traduisent dans des représentations du type base de données ou en des fichiers classiques.

Le modèle organisationnel consiste à définir le **MOT** (Modèle organisationnel des traitements) décrivant les contraintes dues à l'environnement (organisationnel, spatial et temporel).

➤ **Le niveau physique ou opérationnel.**

A ce niveau, on identifie et on décrit les moyens utilisés (choix matériel) pour gérer les *données* et pour activer les *traitements*. En ce qui concerne les données il y a traduction du modèle logique en un modèle physique particulier en faisant appel à des logiciels et au matériel approprié. En ce qui concerne les traitements, le modèle opérationnel décrit l'architecture de différents programmes et les différentes actions élémentaires jusqu'au niveau

algorithme. Il n'y a pas de programmation effective mais on possède tous les éléments pour qu'il n'y ait pas de difficultés de programmation.

Remarque : Ces trois dernières étapes entre dans le cadre de la **démarche par niveaux** pour la conception d'un nouveau système d'information

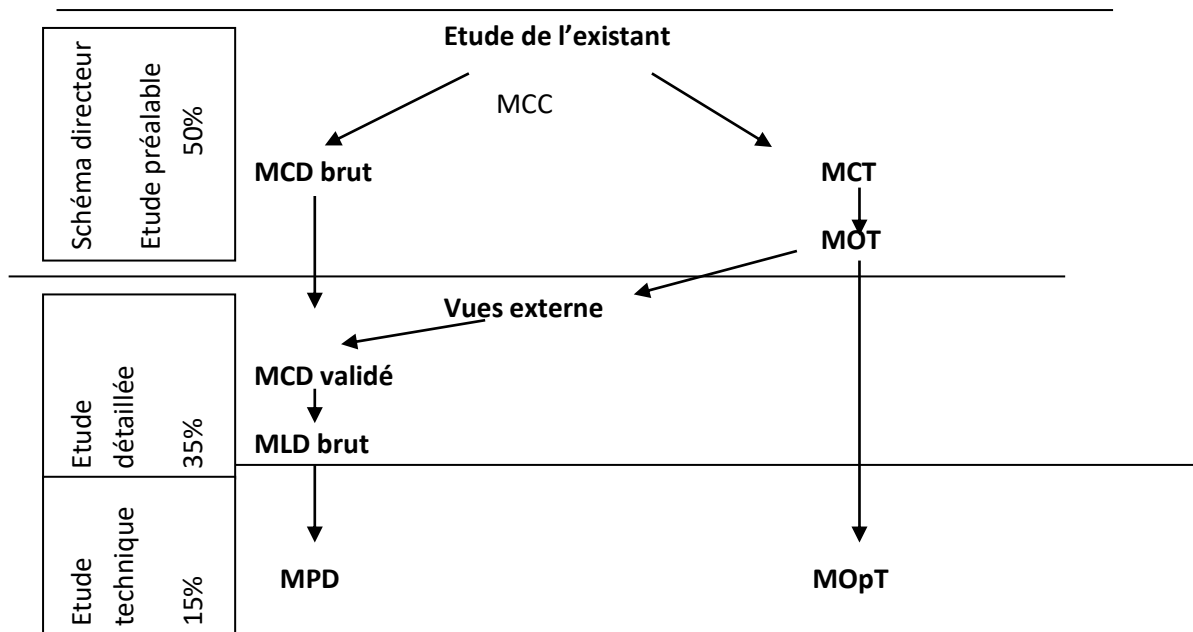
➤ **Système d'information automatisé :**

A ce niveau on procède à l'implantation et la mise à la disposition de l'utilisateur le système développé.

Pour résumer, le tableau suivant présente les différents modèles proposés par Merise à chaque niveau.

Niveau	Préoccupation	Données	Traitements
Conceptuel	Quoi ?	MCD	MCT
Organisationnel ou Logique	Qui fait quoi ? Ou ? Quand ?	MLD	MOT
Physique ou Opérationnel	Avec quels moyens ?	MPD	MopT

Les différents modèles sont organisés de la manière suivante:



3.2. Cycle de vie

Ce cycle décrit la vie du système d'information. Dans l'évolution de tout système on distingue, généralement, un point de départ (la conception), une gestation, une naissance, une croissance, une maturité, une obsolescence et une fin (la mort). En appliquant ces principes à un système d'information on en distingue :

- *la conception* du futur système d'information (descriptions détaillées de spécifications fonctionnelles et techniques).
- *la réalisation* du futur système d'information (production de programmes et de consignes correspondant aux spécifications détaillées).
- *la maintenance* du futur système d'information (adaptation du système aux évolutions de l'environnement).

3.3. Cycle de décision

Ce cycle décrit l'ensemble de décisions et choix pris durant le cycle de vie. Les décisions marquent, généralement, la fin d'une étape et le début d'une autre. Cet ensemble de décisions est hiérarchisé comme il suit :

- décomposition du système d'information en des sous-systèmes et / ou domaines.
- orientations majeures concernant les règles de gestion, les règles d'organisation et les solutions technologiques.
- planification de la réalisation du système d'information
- degré d'automatisation de procédures : procédures manuelles et procédures automatiques
- etc.

Chapitre3 : Le modèle conceptuel de données

1. Présentation

Le modèle conceptuel de données (MCD) est une représentation stable de l'ensemble des données manipulées par l'entreprise ainsi que des relations entre ces données. Le graphe de flux permet souvent de mettre en évidence des individus du MCD.

Le MCD décrit les objets, les événements, les concepts d'une manière abstraite, sans réfléchir, à ce stade, aux possibilités de traitement par l'ordinateur. En ce qui suit, nous étudierons le modèle basé sur le couple entité - association.

Le MCD est précisé lors de l'étude préalable et complété lors de l'étude détaillée du SI.

2. Concepts manipulés.

2.1. Entité

Une entité ou individu est la représentation dans le SI d'un objet concret ou abstrait (matériel ou immatériel) de l'univers de l'entreprise.

Exemples : Client, Compte-bancaire.

Dans le système d'information concernant « **la société au bon beurre** » on peut identifier les entités suivantes : client, commande, bon de livraison, facture.

Remarque. Une entité a une existence propre, sans référence à une autre entité.

On parle d'entité type cad classe d'entité ayant des propriétés analogues.

2.2. Association

Est un lien sémantique de plusieurs entités (indépendamment des traitements) ;

Il est souhaitable de limiter la dimension des relations à deux (relations binaire).

Plus la dimension d'une relation est grande, plus la relation devient complexe et difficile à gérer.

On parle d'association type c'est une association entre plusieurs entités type.

Exemple : Possède est une relation qui lit l'entité Client et l'entité CompteBancaire.

2.3. Propriété

La propriété représente la plus petite quantité d'informations caractérisant une entité ou une association et que l'on peut utiliser d'une manière autonome et qui présente un intérêt pour le domaine à étudier.

Exemples : l'entité Client est caractérisée par les propriétés : NumeroClient, RaisonSociale, adresse et ville.

L'entité CompteBancaire est caractérisée par les propriétés : NumeroCompte, NomBanque et agence

Une **propriété** se décrit par tout ou partie des éléments suivants :

- Son nom: composé de caractères alphanumériques, de préférence une composition de lettres, chiffres et un trait d'union.
- Sa description : ce quelle représente et son intérêt dans le contexte
- Sa nature : son type et/ou domaine,
- Sa longueur : nombre de caractères dans le format

Une propriété peut être :

- Obligatoire ou facultative
- Brut (non calculée)
- Elémentaire (non décomposable)
- Normée par des organismes officiels (Cin , Nimm , Ntel , ...)

2.4. Dictionnaire de données

La liste de toutes les propriétés est définie dans un dictionnaire de données. Celui-ci contient pour chaque propriété tout ou partie des éléments suivants : son nom, son type et sa longueur, sa description et contrainte ou remarques, comme le montre l'exemple suivant.

Les propriétés définies dans un dictionnaire de données doivent être élémentaires.

Nom	Type	Longueur	Description	Remarque	Contrainte
ENum	Numérique	octet	Numéro Enseignant	Identifiant, ne peut être modifié	
ENom	Texte	20	Nom Enseignant		
EPrenom	Texte	15	Prénom		
...

Une propriété définie dans le dictionnaire de données doit être :

- Pertinente : présente un intérêt pour le domaine étudié
- Unique dans le modèle, Deux ou plusieurs entités ne peuvent jamais avoir une même propriété (*Exemple* : Nous avons défini **raisonSociale** comme propriété dans l'entité **Client**, celle – ci ne peut pas être définie dans l'entité **Facture**).

- Brut (non calculée exemple : MTTC représente le montant toute taxe comprise, cette propriété est calculée à partir de MHT, le montant hors taxe et TVA, la valeur de la TVA)
- Atomique: nom décomposable (cette décomposition est relative à son exploitation dans le SI exemple : adresse contenant aussi la ville)
- A une seule signification : si une propriété a plusieurs sens, on parle de polysème, par contre on parle de synonymes quand deux propriétés ont le même sens.

Exemple de Polysème :

Date est une propriété qui peut représenter la date de la commande client et la date de livraison.

Dans cet exemple, il faut utiliser deux propriétés pour représenter chacune des deux dates, ComDate pour la date de la commande et LivDate pour la date de livraison.

Exemple de synonyme :

Libellé et désignation sont deux propriétés qui ont représenté le nom d'un produit, ils ont donc le même sens.

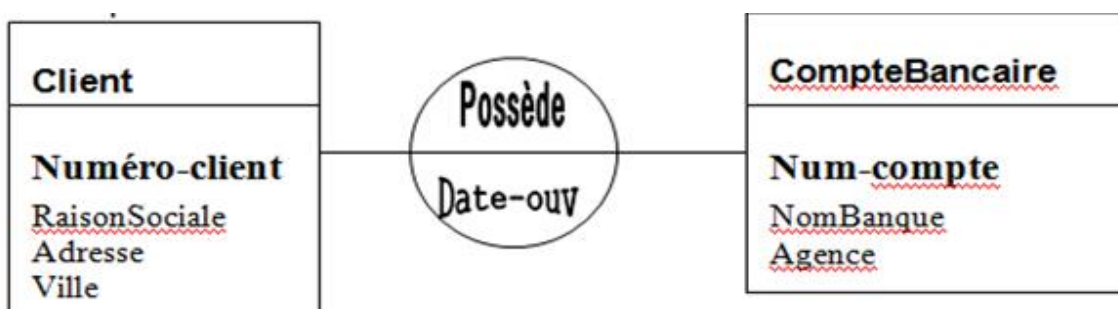
Dans cet exemple, il faut choisir un seul nom pour désigner le nom du produit.

3. Formalisme Graphique

La méthode Merise propose des représentations graphiques pour la plupart de notions qu'elle utilise. Une entité est représentée par un rectangle dans lequel apparaissent son nom et ses propriétés. Une association par une ellipse dans lequel apparaissent son nom et ses propriétés éventuelles.



Exemple:



4. Occurrence

4.1. Occurrence d'une propriété. Toute valeur prise par la propriété définit une occurrence de la propriété.

Exemple : Considérons l'entité **Etudiant** caractérisée par les propriétés : **Numéro** de convocation, **Nom**, **Prénom**, **Date** de **naissance**, Moyenne en **Arabe** et Moyenne en **Anglais**.

Numéro	Nom	Prénom	Date_naiss	Arabe	Anglais
001	SLAOUI	Amine	28/10/96	12,05	10,46
002	BENMOUSSA	Latifa	08/10/95	8,81	10,44
003	HAJAMA	Mohamed	23/01/97	10,31	11,09
004	FARAH	Adil	02/09/95	9,30	13,09
005	KABAJ	Ilham	26/07/94	0,00	0,00
006	TADLAOUI	Rachid	01/05/94	10,83	12,12
007	BENNIS	Karim	22/03/98	10,14	6,89
008	BENNANI	Meryem	21/07/97	10,50	13,60
009	GUENNOUN	Abdelatif	15/12/97	10,50	16,04
010	DAHBI	Jamal	21/06/96	9,23	14,75

Dans le tableau présenté ci-dessus on trouve les valeurs prises par les différentes propriétés. La propriété *Date_naiss* prend les valeurs (donc les occurrences !) : 28/10/96 ; 08/10/95 ; 23/01/97 ; 02/09/95 ; 26/07/94 ; 01/05/94 ; 22/03/98 ; 21/07/97 ; 15/12/97 ; 21/06/96

4.2. Occurrence d'une entité. L'ensemble de valeurs prises par les propriétés caractérisant une entité définit l'occurrence d'une entité.

Exemple : L'ensemble de valeurs suivant est une occurrence de l'entité **étudiant**. Dans le tableau ci-dessus. L'entité **étudiant** a 10 occurrences. (Chaque ligne du tableau est une occurrence de l'entité).

001	SLAOUI	Amine	28/10/96	12,05	10,46
-----	--------	-------	----------	-------	-------

4.3. Occurrence d'une association. Toute réalisation de l'association générée par des occurrences des entités impliquées dans l'association.

5. Identifiant (ou Clé)

Constitué d'une ou de plusieurs propriétés permettant de distinguer sans ambiguïté une occurrence de l'entité. Toute entité doit avoir un identifiant.

Exemples : L'identifiant de l'entité Client est NuméroClient car deux clients ne peuvent pas avoir le même numéro.

L'entité **voiture** ayant les propriétés : NuméroImmatriculation, Couleur, Marque a comme identifiant la propriété **NuméroImmatriculation**.

Un identifiant d'une association est l'ensemble des identifiants des entités reliées à cette relation.

La méthode Merise propose de précéder les propriétés de l'identifiant par le caractère #.

Exemples :

Client	Voiture
#Numéro client raisonSociale Adresse Ville	# NuméroImmatriculation Couleur Marque

Un identifiant est simple quand il est constitué d'une seule propriété.

Un identifiant est composé quand il est constitué de plusieurs propriétés.

6. Cardinalité

Est notée sous la forme (X,Y), X est le nombre minimum d'occurrences de l'entité1 relié à l'entité2 et Y est le nombre maximum d'occurrences de l'entité1 relié à l'entité2.

Exemple : Un client possède au minimum 1 compte bancaire et au maximum plusieurs représenté par N d'ou la cardinalité « 1,N ». Alors qu'un compte bancaire appartient à un client et un seul d'ou la cardinalité « 1,1 ».

Dans la pratique on gère les cardinalités suivantes :

0,1 : Chaque occurrence de l'entité est relié à au plus une occurrence de l'association.

Exemple : Un employé est responsable d'au plus un service.

1,1 : Chaque occurrence de l'entité est reliée exactement à une occurrence de l'association.

Exemple : Un compte bancaire appartient à un client et un seul.

0,N : Chaque occurrence de l'entité est relié à un nombre quelconque d'occurrence de l'association.

Exemple : Une personne possède 0 ou N voitures.

1,N : Chaque occurrence de l'entité est relié à au moins une occurrence de l'association.

Exemple : Un enseignant assure au moins un cours.

Dans le cas des relations binaires, ce qui est fréquent est résumé dans le tableau suivant :

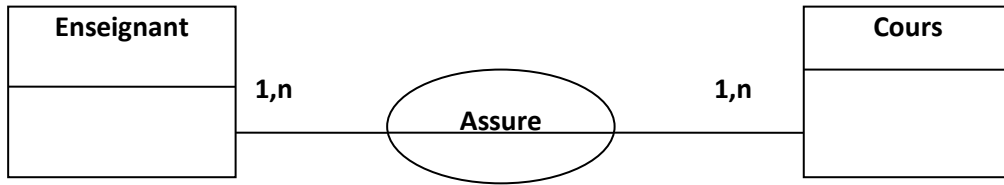
Entité1/Entité2	0-1	1-1	0-N	1-N
0-1	?	?		
1-1	?	*		
0-N				
1-N				

? : Il faut se demander si l'une des entités n'est pas propriété d'une autre.

- : A éviter car il traduit souvent une erreur de conception.

Exemples.

Un enseignant assure un ou plusieurs cours. Un cours peut être assuré par un ou plusieurs enseignants.

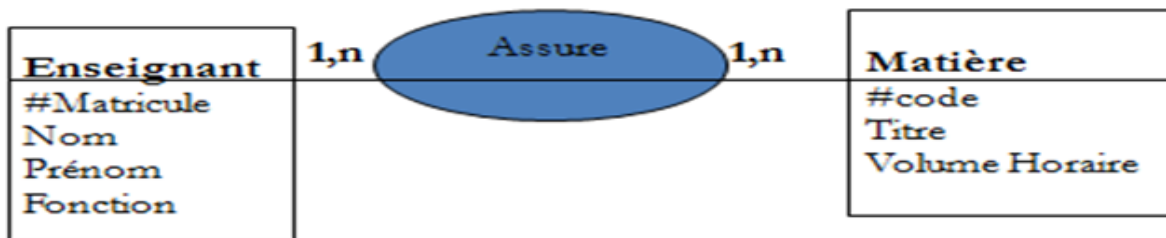


7. Dimension d'une association

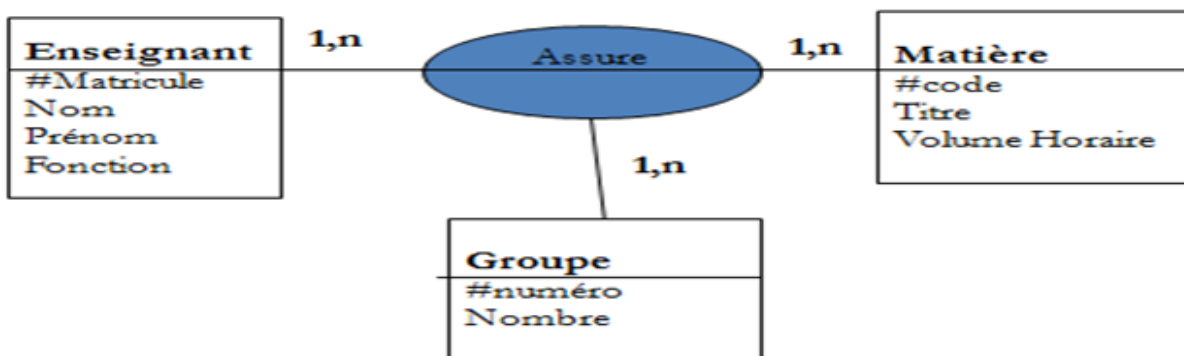
La dimension d'une association est le nombre d'entité qu'elle relie.

- Association binaire : reliant deux entités.
- Association n-aires : reliant plusieurs entités (avec $n > 2$)
- Association réflexive : qui associe les occurrences d'une même entité.
-

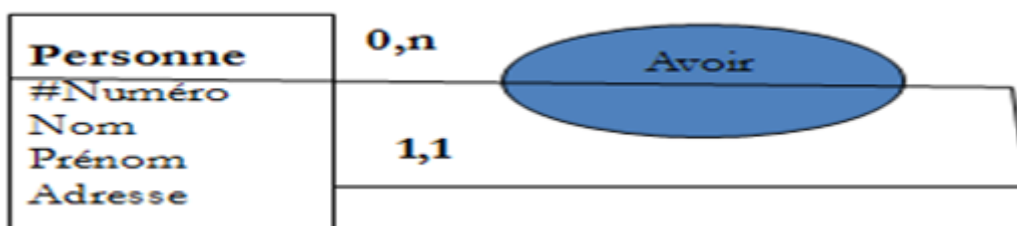
Exemple d'une association binaire :



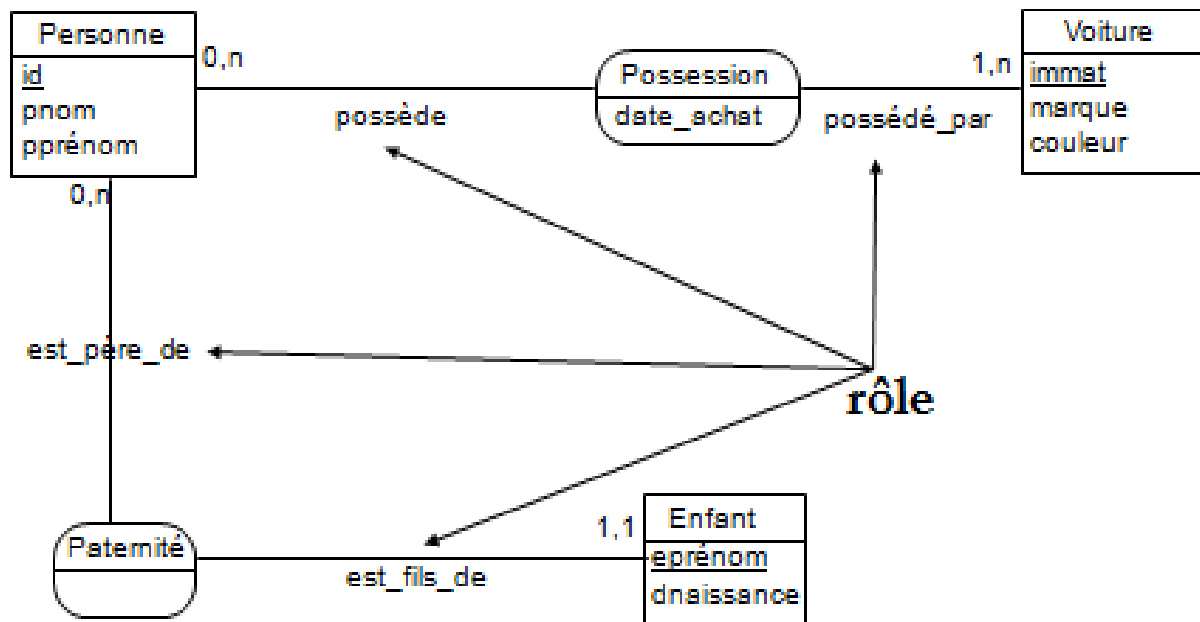
Exemple d'une association n-aire :



Exemple d'une association réflexive :



Il est utile parfois d'ajouter des rôles pour faciliter la lecture de l'association dans un sens et dans l'autre comme le montre l'exemple suivant :



8. Démarche à suivre

La démarche à suivre dans un premier temps pour la conception et la réalisation d'un modèle conceptuel de données :

Généralement, on parcourt les étapes suivantes:

- Etablir un dictionnaire des données (les noms de différentes propriétés, leur nature,.. voir DD).
- Rattacher des propriétés à chaque entité.
- Etablir un identifiant pour chaque entité.
- Définir les associations.
- Rattacher, éventuellement, des propriétés aux associations.
- Construire le schéma du modèle conceptuel de données.
- Calculer les cardinalités.

En résumé, nous pouvons vérifier le MCD obtenu en appliquant les règles suivantes :

Règle1 : Toutes les propriétés doivent être élémentaires.

Règle2 : Chaque entité doit posséder un identifiant et un seul.

Règle3 : L'identifiant détermine d'une manière unique toutes les propriétés de l'entité.

Règle4 : Une propriété ne peut qualifier qu'une seule entité ou qu'une seule association.

Règle5 : Les propriétés d'une association doivent dépendre de la totalité des identifiants des entités participantes.

Chapitre4 : Le Modèle logique de données (relationnel)

1. Introduction

Après avoir conçu le Modèle Conceptuel de Donnée (MCD), l'étape suivante est de le transposer en Modèle Logique de Données Relationnelles (MLDR). Ce MLD est en fait le dernier pas vers le Modèle Physique de donnée (MPD), c'est à dire la description de la base qui va être créée. Après avoir définis les notions de clé primaire et de clé étrangère, nous appliquons les règles nécessaires et suffisantes pour passer d'un MCD à un MLD relationnel.

2. Le Modèle relationnel.

Ce modèle crée par le mathématicien CODD s'inspire du concept mathématique de **relation**.

Une relation \mathfrak{R} est représentée comme un tableau de données, d'où l'appellation de **table**.

Toute colonne d'une relation s'appelle **attribut** (ou encore **champ**). Un attribut est, également, caractérisé par un nom.

Exemple. Considérons les domaines **Couleur** = {Rouge, Vert, Bleu} et **Constructeur** = {Fiat, Renault, Volvo, Opel}. Considérons la relation **Voiture** définie par **Voiture** = {(Rouge, Fiat), (Rouge, Renault), (Vert, Opel), (Bleu, Opel)}. Cette relation peut s'écrire, encore, sous la forme :

Voiture	Couleur_Voiture	Marque
	Rouge	Peugeot
	Rouge	Renault
	Vert	Renault
	Bleu	Mercedes

Cette relation a le nom **Voiture** et les attributs **Couleur_Voiture** et **Marque**. L'attribut **Couleur_Voiture** est un sous - ensemble du domaine **Couleur** et l'attribut **Marque** est un sous - ensemble du domaine **Constructeur**. Une ligne de la table s'appelle **tuplet (ou n-uplet)** ou encore, par abus de langage, **enregistrement** ou **occurrence**.

Une relation est représentée par son nom suivi de la liste de ses attributs. Par exemple la relation Voiture s'écrit : **Voiture(Couleur_Voiture, Marque)**.

Clé. On entend par **clé** un ou plusieurs attributs permettant d'identifier un tuple unique de la relation. Deux tuples ne peuvent pas avoir une même valeur pour la clé.

Exemple. Redéfinissons la relation **Voiture** de la manière suivante :

Voiture	N_Immatriculation	Couleur_Voiture	Marque
	1391-A-15	Rouge	Peugeot
	2900-A-17	Rouge	Renault
	2999-B-22	Vert	Renault
	2800-A-15	Bleu	Mercedes

Cette relation a comme clé l'attribut **N_Immatriculation**. Il est évident qu'il n'existe pas deux voitures ayant un même numéro d'immatriculation.

Une clé peut être constituée de plusieurs attributs.

Exemple :

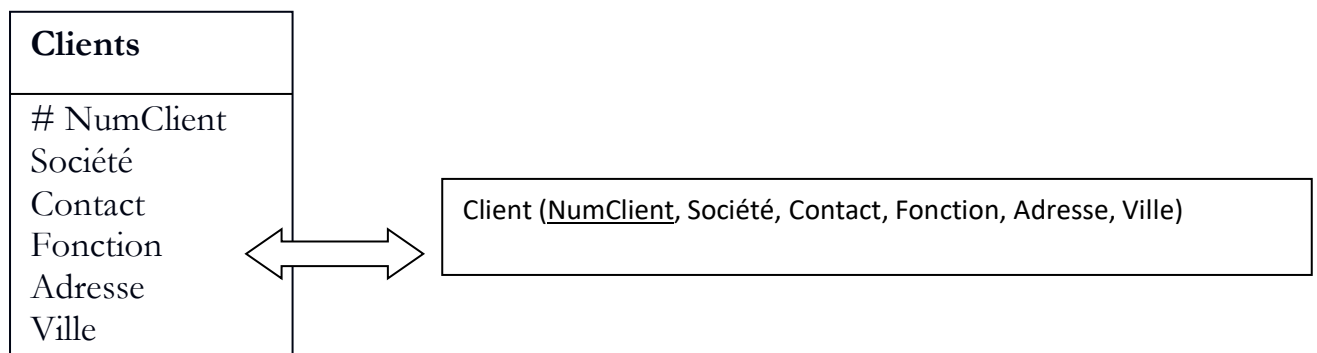
SuivreCours	CodeEtudiant	CodeCours	Date
	C002	01	12/10/2018
	C003	02	16/10/2018
	C002	03	13/10/2018
	C004	01	12/10/2018
	C004	02	16/10/2018

La clé de cette relation est {**CodeEtudiant, CodeCours**} On écrira la relation ainsi : **SuivreCours**(CodeEtudiant, CodeCours, Date). La clé est soulignée.

3. Règles de passage du modèle entité - association au modèle relationnel.

3.1. Une entité se transforme en une relation (table)

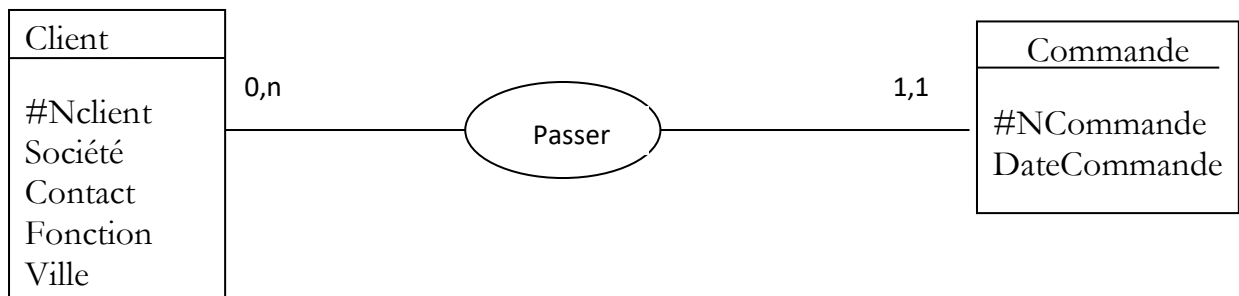
Toute entité du MCD devient une relation du MLDR, et donc une table de la Base de Données. Chaque propriété de l'entité devient un attribut de cette relation, et dont une colonne de la table correspondante. L'identifiant de l'entité devient la **Clé Primaire** de la relation (elle est donc soulignée), et donc la **Clé Primaire** de la table correspondante.



3.2. Relation binaire aux cardinalités (X,1) - (X,n), X= {0 ou 1}

La **Clé Primaire** de la table à la cardinalité (X,n) devient une **Clé Etrangère** dans la table à la cardinalité (X,1) :

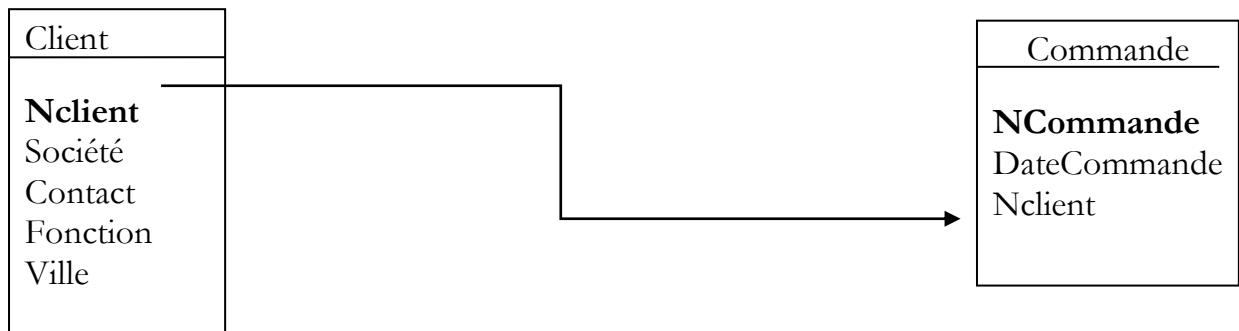
Exemple



MLD : Client (Nclient, Société, Contact, Fonction, Ville)

Commande (NCommande, DateCommande, #Nclient)

Modèle physique de donnée (MPD)

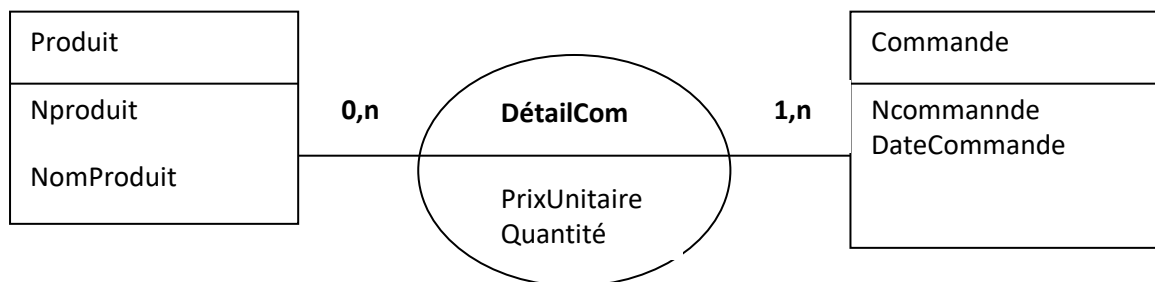


3.3. Relation binaire aux cardinalités (X,n) - (X,n), X= {0 ou 1}

Il y a création d'une table supplémentaire ayant comme **Clé Primaire** une clé composée des **identifiants** des 2 entités. On dit que la **Clé Primaire** de la nouvelle table est la **concaténation** des **Clés Primaires** des deux autres tables.

Si la relation est porteuse de donnée, celles ci deviennent des attributs pour la nouvelle table.

Exemple

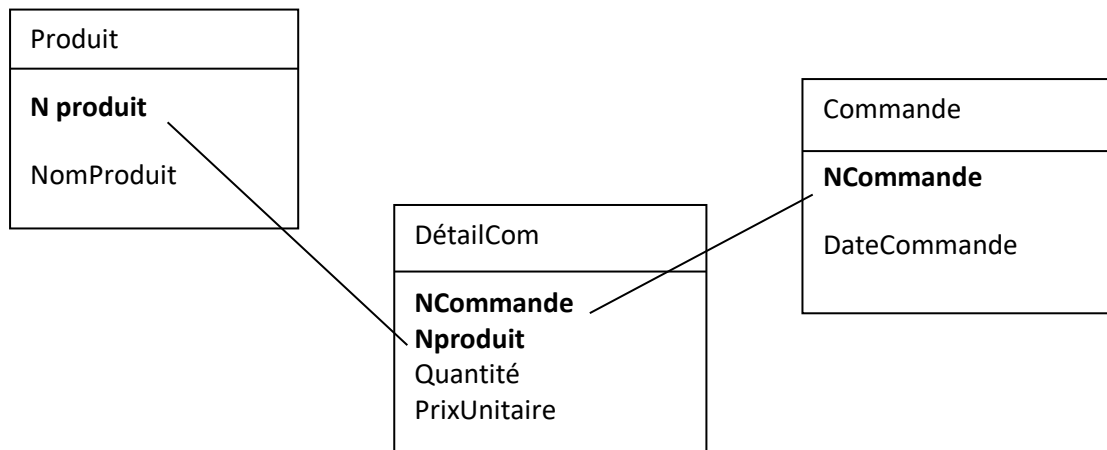


MLD : COMMANDE (NCommande, DateCommande)

PRODUIT (Nproduit, NomProduit)

DétailCom (NCommande, Nproduit, quantité, PrixUnitaire)

MPD

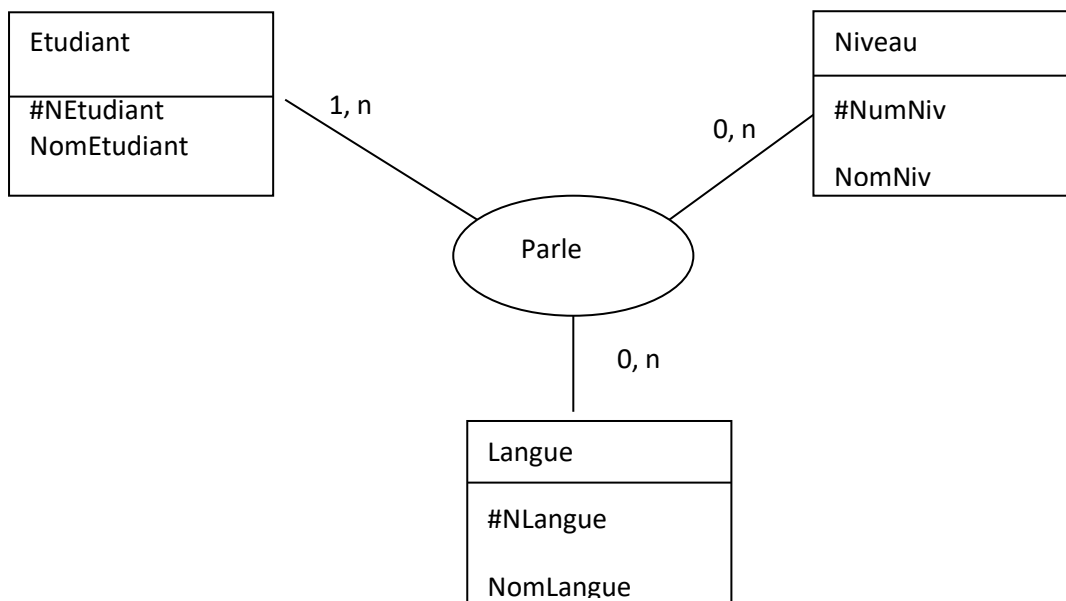


3.4. Relation n-aire (quelles que soient les cardinalités).

Il y a création d'une table supplémentaire ayant comme **Clé Primaire** la **concaténation** des **identifiants** des entités participant à la relation.

Si la relation est porteuse de donnée, celles ci deviennent des attributs pour la nouvelle table.

Exemple



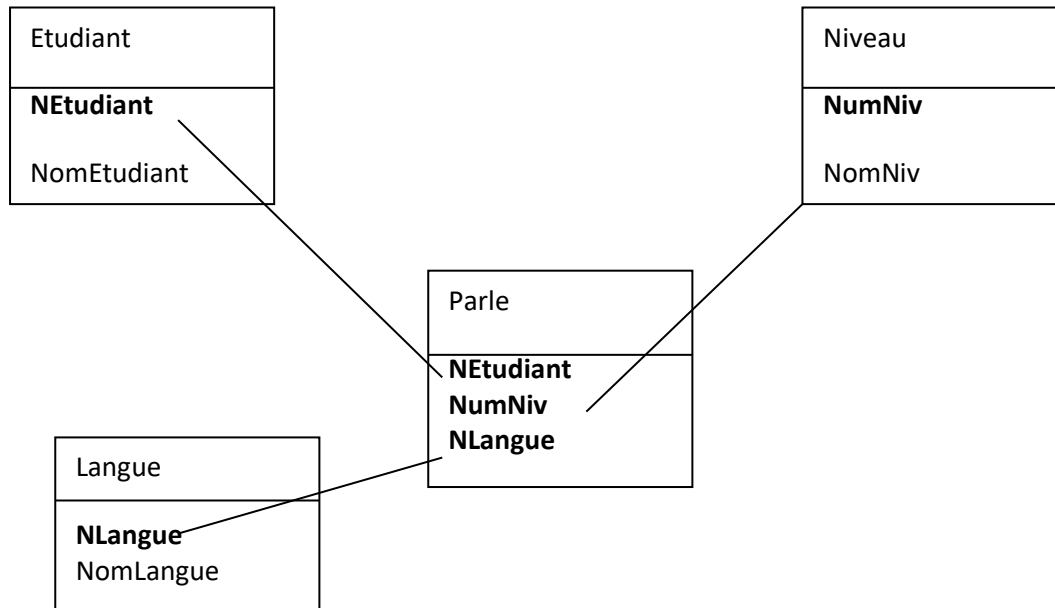
MLD : ETUDIANT (NEtudiant, NomEtudiant)

NIVEAU (NumNiv, NomNiv)

LANGUE (NLangue, NomLangue)

PARLE (NEtudiant, NumNiv, NLangue)

MPD

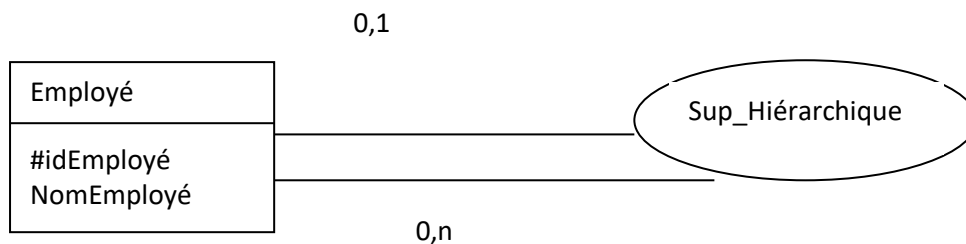


3.5. Association Réflexive.

- Premier cas : cardinalité $(X,1) - (X,n)$, avec $X=\{0 \text{ ou } 1\}$.
La **Clé Primaire** de l'entité se dédouble et devient une **Clé Etrangère** dans la relation ou nouvelle table. Exactement comme si l'entité se dédoublait et était reliée par une relation binaire $(X,1) - (X,n)$ (Cf règle 2).

Exemple

Une société est organisée de manière pyramidale : chaque employé a 0 ou 1 supérieur hiérarchique direct. Simultanément, chaque employé est le supérieur hiérarchique direct de 0 ou plusieurs employés.



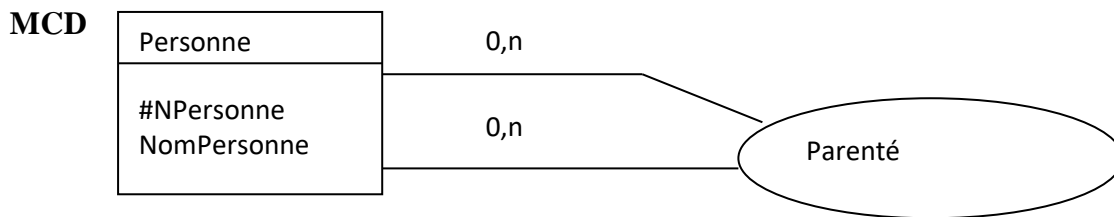
MLD: EMPLOYE (idEmploye, NomEmploye, #idSup)

idSup correspond à l'identifiant (idEmploye) du supérieur hiérarchique direct de l'employé considéré.

- Deuxième cas : cardinalité $(X,n) - (X,n)$, avec $X=\{0 \text{ ou } 1\}$.
De même, tout se passe exactement comme si l'entité se dédoublait et était reliée par une relation binaire $(X,n) - (X,n)$ (règle 3). Il y a donc création d'une nouvelle table.

Exemple

Une personne a 0 ou n descendants directs (enfants), et a aussi 0 ou n descendants indirects (enfants).



MLD : PERSONNE (NPersonne, NomPersonne)

PARENTE (NParent, idEnfant)

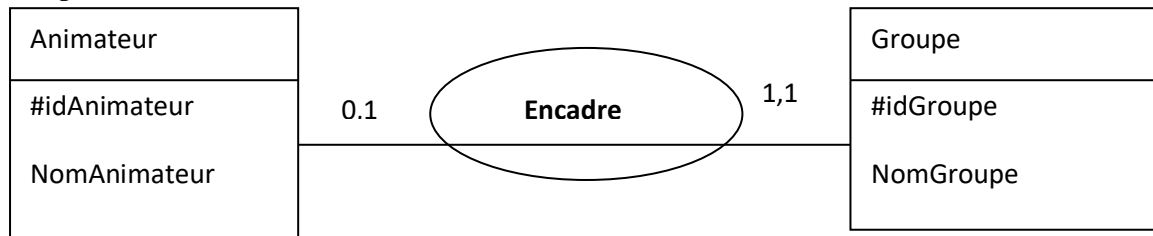
NParent est l'identifiant (NPersonne) d'un ascendant direct de la personne. idEnfant est l'identifiant (NPersonne) d'un descendant direct de la personne.

La table PARENTE sera en fait l'ensemble des couples (parents-enfants) présents dans cette famille.

3.6. Relation binaire aux cardinalités (0,1) - (1,1).

La **Clé Primaire** de la table à la cardinalité (0,1) devient une **Clé Etrangère** dans la table à la cardinalité (1,1):

Exemple



MLD: ANIMATEUR (idAnimateur, NomAnimateur)

GRUPE (idGroupe, NomGroupe, #idAnimateur)

Chapitre5 : Le modèle physique de données

1. Introduction

Cette étape consiste à implémenter le modèle dans un Système de Gestion de base de données (SGBD), il convient donc de créer une base de données relationnelle. Le système utilisé dans le cadre de cette formation : Access.

2. Définition d'une Base de Données

Une base de données (BD) est un ensemble structuré et cohérent de données, stocké sur disque, accessible par un ou plusieurs utilisateurs sur lesquelles on peut développer différentes applications.

De façon informelle, on peut considérer une BD comme une grande quantité de données, centralisées ou non, servant pour les besoins d'une ou plusieurs applications, interrogeables et modifiables par un groupe d'utilisateurs travaillant en parallèle.

Une BD est faite pour :

- ✓ enregistrer les données,
- ✓ les restituer à la demande,
- ✓ tirer des conclusions en rapprochant les faits les uns des autres.

Une BD est un ensemble d'informations sur un sujet qui est :

- ✓ exhaustif,
- ✓ non redondant,
- ✓ structuré,
- ✓ persistant.

La base de données est un schéma relationnel constitué d'un ensemble de relations reliées entre elles.

Dans ce modèle, les données sont stockées dans des tables (relations) , sans préjuger de la façon dont les informations sont stockées dans la machine. Un ensemble de données sera donc modélisé par un ensemble de tables.

De façon informelle, on peut définir le modèle relationnel de la manière suivante :

- Les données sont organisées sous forme de tables à deux dimensions, encore appelées relations et chaque ligne est un n-uplet ou tuple,
- les données sont manipulées par des opérateurs de l'algèbre relationnelle,
- l'état cohérent de la base est défini par un ensemble de contraintes d'intégrité.

3. Définition d'un Système de Gestion de Base de Données

Un système de gestion de base de données (SGBD) est un logiciel qui permet de décrire, modifier, interroger et administrer les données d'une base de données.

Le SGBD peut être vu comme le logiciel qui prend en charge la structuration, le stockage, la mise à jour et la maintenance des données ; c'est, en fait, l'interface entre la BD et les utilisateurs ou leurs programmes.

Un SGBD comprend

- ✓ Un système de gestion de fichiers pour la gestion du stockage physique
- ✓ Un système interne pour le placement et l'accès aux données
- ✓ Et un système externe comme pour le langage de requête élaboré.

4. Concepts de base

4.1. Table, attribut et tuple

Une « table » structurée en colonnes fixes, et en lignes pouvant varier dans le temps en nombre et en contenu, est appelée **Table** (relation dans le MLD).

Exemple : Relation Voiture

IdV	NomM	CatM	NbCh
19202A15	Renault	Clio	7
15422A17	Renault	Megane	8
14878A15	Mercedes	C220	10

Une relation peut être vue comme un tableau à deux dimensions dont chaque colonne (attribut) correspond à un domaine et porte un nom et chaque ligne à un tuple.

Une relation n'est pas définie par des concepts positionnels, les lignes (resp. les colonnes) peuvent être permutées.

Une table instance d'une relation est un ensemble non-ordonné de tuples (lignes). Chaque tuple est composé de valeurs correspondant aux attributs (noms des colonnes) de la relation.

Les règles vérifiées sur la propriété (Vu dans le chapitre MCD) s'appliquent bien entendu à l'attribut dans le MLD et à la colonne (ou champ) dans le MPD.

Une table possède un identifiant (clé primaire) obligatoire et peut avoir une ou plusieurs clé étrangère. La clé étrangère correspond à la clé primaire d'une autre table dans le même schéma relationnel.

4.2. Domaine

Représente l'ensemble de valeurs acceptables pour un composant d'une relation. Un domaine représente le type de l'attribut, sa taille et éventuellement une restriction sur le type.

Ceci permettra de faire un contrôle au moment de la saisie de la donnée.

4.3. Clé d'une relation

Soit $R (A_1, A_2, \dots, A_n)$ un schéma de relation, et X un sous-ensemble de (A_1, A_2, \dots, A_n) , X (groupe d'attributs) est une clé si, et seulement si:

- X est unique, elle détermine tous les attributs (A_1, A_2, \dots, A_n)
- La valeur de X est obligatoire (Not NULL)
- X est minimale.

La clé dans un schéma de relation est soulignée

Exemple1 : Etudiant (cne , nom , prenom , dnais , nogpe)

cne est une clé de la relation Etudiant signifie que le cne est unique cad deux étudiants différents ne peuvent jamais avoir la même valeur de cne.

Exemple2 : Soit le schéma suivant:

Client (noclient, nom , adresse, ville)

Commande(nocom, date, noclient)

Produit(ref, designation prix)

LigneCommande(nocom, ref, quantité)

nocom et Pref est une clé composée dans la relation LigneCommande, elle détermine la quantité achetée.

Une même commande (nocom) peut avoir plusieurs produits. Un produit (ref) peut appartenir à plusieurs commandes. Par contre un produit dans une commande ne se répète pas.

L'attribut noclient dans la table Commande est une clé étrangère, du fait que cet attribut soit clé primaire dans la table Client.

4.4. Contrainte d'intégrité

Il existe différents types de contraintes d'intégrité:

- ✓ liées au modèle (pas de doublons dans une relation);
- ✓ de domaine *Exemple :* une note doit être comprise entre 0 et 20
- ✓ clé primaire dont la valeur est unique dans sa relation.
- ✓ référentielles dites de clé étrangère qui impose que la valeur d'attribut de la relation r_1 apparaît comme valeur de clé dans une autre relation r_2 .

Exemple : Etudiant (cne , nom , prenom , dnais , nodept)
Département (nodept , nom , matricule)

nodept est une clé primaire dans la relation Département, et une clé étrangère dans la relation Etudiant. La valeur de cet attribut n'existe dans la table étudiant que s'il est déjà défini dans la table Département.

Chapitre6 : Le langage de manipulation de données SQL

1. Introduction

Structured Query Language (Langage structuré de requêtes) est un langage de manipulation de données (LMD) relationnelles.

Ce langage permet de spécifier les données à sélectionner ou à mettre à jour à propriétés de valeurs, sans dire comment retrouver les données.

Les opérations utilisées sont en général celles des langages dits assertionnels.

Plusieurs langages permettent de manipuler les bases de données relationnelles ont été proposés, en particulier Quel, QBE et SQL.

Aujourd'hui, le langage SQL est normalisé et est le standard d'accès aux bases de données relationnelles.

Ce langage a pour objectif d'être facilement utilisé par le programmeur, ses fonctionnalités sont plus larges que celles du langage algébrique, il comprend les fonctions :

- Description
- Recherche
- Mise à jour
- Confidentialité et cohérence.

Nous allons étudier dans ce cours la partie Recherche (ou sélection) du LMD SQL.

2. RECHERCHE

C'est la fonction d'interrogation faisant intervenir l'opération de sélection.

Une requête de recherche SQL utilise une ou plusieurs relations en entrée pour afficher le résultat sous forme d'une relation en sortie.

2.1. Format général d'une requête

Rappelons qu'une projection effectue l'extraction de colonnes (attributs) spécifiés d'une relation, puis élimine les attributs en double. SQL n'élimine pas les doubles à moins que ceci soit demandé par le mot clé DISTINCT. La projection sous SQL rend possible l'utilisation des fonctions de calculs sur les colonnes (attributs) extraites. Les fonctions de calculs utilisées sont en particulier les fonctions arithmétiques.

Syntaxe :

```
SELECT [ALL / DISTINCT] { <EXPRESSION DE VALEURS> /* }  
      FROM <LISTE_RELATION >  
      [ WHERE <CONDITIONS> ] ;
```

Une expression de valeurs peut être la liste des attributs ou des expressions arithmétiques.

* indique tous les attributs.

Exemple : Soit la relation Voiture (NIMM, marque, modèle, prix, couleur, achat)

Un extrait de la table voiture est le suivant :

voiture						
NIMM	marque	modèle	prix	couleur	Achat	
124870A15	Renault	Laguna	30	Blanc	12/07/2011	
152140A15	Renault	Clio	15	Rouge	12/05/2010	
156020B10	Peugeot	308	24	GrisSouris	06/08/2012	
156879A12	Peugeot	308	24	GrisClair	01/09/2014	
168790D12	Hyundai	i30	23	Blanc	01/02/2015	
169020A17	Hyundai	i30	21	GrisSouris	02/05/2012	
192020A15	Renault	Megane	26	Blanc	01/10/2013	
196020A17	Peugeot	207	21	GrisClair	01/08/2010	
254001B15	Renault	Clio	18	Noir	25/05/2012	

La requête SQL pour afficher la liste des voitures :

```
SELECT *
FROM VOITURE ;
```

Cette requête affiche à l'écran les 9 tuples de la table voiture.

Il est possible aussi de mentionner certains attributs à afficher et de préciser les tuples vérifiant une condition.

Exemple

La requête SQL pour afficher le prix et le modèle des voitures de marque « Renault » :

```
SELECT PRIX, MODÈLE
FROM VOITURE
WHERE MARQUE = "RENAULT" ;
```

Le résultat affiché à l'écran de cette requête est le suivant :

Requête1	
Prix	modèle
15	Clio
18	Clio
26	Megane
30	Laguna

Une condition peut être simple ou composée.

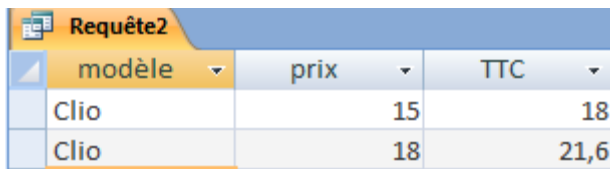
Une condition composée utilise les opérateurs logiques And, Or et/ou Not.

Exemple

La requête SQL pour afficher le modèle, le prix et le prix TTC des voitures de marque « Renault » de prix HT inférieur ou égale à 25 millions.

```
SELECT MODÈLE , PRIX, PRIX * 1.2 AS TTC
FROM VOITURE
WHERE (MARQUE = "RENAULT") AND (PRIX <= 25);
```

Le résultat affiché à l'écran de cette requête est le suivant :



modèle	prix	TTC
Clio	15	18
Clio	18	21,6

Dans la clause Select de cette requête affiche une expression calculée (prix * 1,2) , elle est nommée TTC en utilisant « As ».

La condition de la clause Where est composée, l'utilisation de parenthèses est recommandée.

2.2. La clause distinct

Exemple : Pour afficher les marques des voitures, nous pouvons écrire la requête SQL suivante :

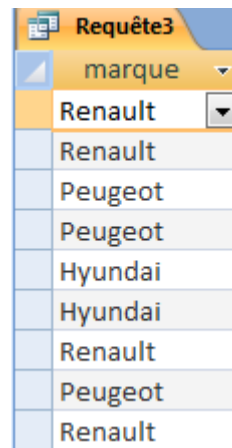
```
SELECT MARQUE  
FROM VOITURE;
```

Ceci est équivalent à la requête suivante :

```
Select ALL (Marque) from voiture ;
```

ALL est pris par défaut

Le résultat affiché à l'écran est le suivant :



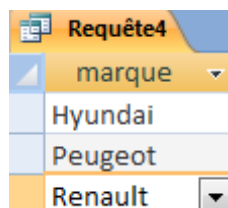
marque
Renault
Renault
Peugeot
Peugeot
Hyundai
Hyundai
Renault
Peugeot
Renault

Pour éliminer les doublons au niveau de l'affichage, nous utilisons la clause distinct.

La requête SQL correspondante est :

```
SELECT DISTINCT ( MARQUE )  
FROM VOITURE;
```

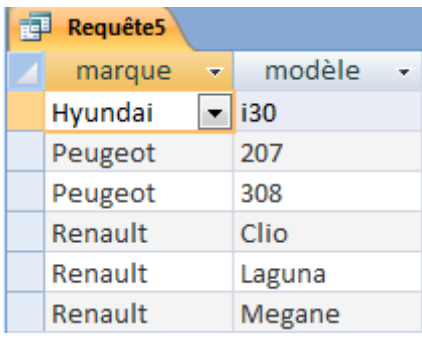
Le résultat affiché est donc :



marque
Hyundai
Peugeot
Renault

Remarque : Distinct s'applique sur le tuple entier de la table voiture.

Exemple : Pour afficher les différentes marques et modèles des voitures

<p>La requête SQL est :</p> <pre>SELECT DISTINCT (MARQUE) , MODÈLE FROM VOITURE;</pre>	<p>Le résultat affiché est le suivant :</p>  <table border="1"><thead><tr><th>marque</th><th>modèle</th></tr></thead><tbody><tr><td>Hyundai</td><td>i30</td></tr><tr><td>Peugeot</td><td>207</td></tr><tr><td>Peugeot</td><td>308</td></tr><tr><td>Renault</td><td>Clio</td></tr><tr><td>Renault</td><td>Laguna</td></tr><tr><td>Renault</td><td>Megane</td></tr></tbody></table>	marque	modèle	Hyundai	i30	Peugeot	207	Peugeot	308	Renault	Clio	Renault	Laguna	Renault	Megane
marque	modèle														
Hyundai	i30														
Peugeot	207														
Peugeot	308														
Renault	Clio														
Renault	Laguna														
Renault	Megane														

Remarque : Les doublons sur la marque ne sont pas éliminés si le modèle est différent.

2.3. Trier un résultat

Pour donner un ordre au tuples affichés, la clause order by peut être utilisée.

Syntaxe : Order by attribut1 [asc/dec], attribut2 [asc/dec],...

L'affichage est effectué selon un ordre défini pour le premier attribut (attribut1), le second attribut (attribut2) est trié par rapport aux valeurs identiques du premier attribut,...

Asc : ordre croissant pris par défaut

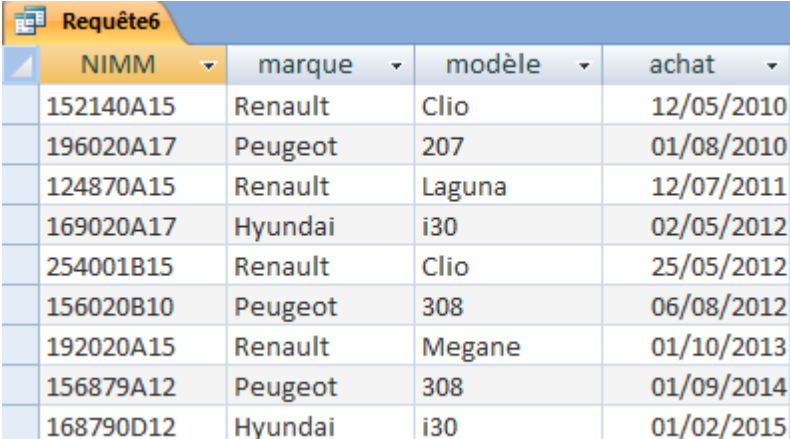
Desc : ordre décroissant.

Exemple : Pour afficher les NIMM, marque et modèle des voitures dans l'ordre de la date d'achat.

La requête SQL est la suivante :

```
SELECT NIMM, MARQUE ,  
MODÈLE , ACHAT  
  
FROM VOITURE  
  
ORDER BY ACHAT;
```

Le résultat affiché de la requête :



NIMM	marque	modèle	achat
152140A15	Renault	Clio	12/05/2010
196020A17	Peugeot	207	01/08/2010
124870A15	Renault	Laguna	12/07/2011
169020A17	Hyundai	i30	02/05/2012
254001B15	Renault	Clio	25/05/2012
156020B10	Peugeot	308	06/08/2012
192020A15	Renault	Megane	01/10/2013
156879A12	Peugeot	308	01/09/2014
168790D12	Hyundai	i30	01/02/2015

Exemple : Pour afficher les marque, modèle , couleur et prix des voitures par ordre croissant de marque et décroissant de prix.

La requête SQL est la suivante :

```
SELECT MARQUE ,
MODÈLE , COULEUR, PRIX

FROM VOITURE

ORDER BY MARQUE ,
PRIX DESC;
```

Le résultat affiché de la requête :

marque	modèle	couleur	prix
Hyundai	i30	Blanc	23
Hyundai	i30	GrisSouris	21
Peugeot	308	GrisClair	24
Peugeot	308	GrisSouris	24
Peugeot	207	GrisClair	21
Renault	Laguna	Blanc	30
Renault	Megane	Blanc	26
Renault	Clio	Noir	18
Renault	Clio	Rouge	15

2.4. Fonctions agrégats

Ce sont les fonctions d'évaluation d'ensembles.

AVG(n) : Donne la moyenne des valeurs de n
 Count(n) : Donne le nombre de tuples
 Max(n) : Donne la plus grande valeur de n
 Min(n) : Donne la plus petite valeur de n
 Sum(n) : Donne la somme des valeurs de n
 Count(*) : Donne le nombre de tuples qui satisfont la condition.

Attention : une fonction agrégat est utilisée dans un select (**Jamais dans un Where**)

Exemple : Pour afficher le nombre de voitures de marque « Peugeot ».

La requête SQL est la suivante :

```
SELECT COUNT(*) AS NOMBREPEUGEOT

FROM VOITURE

WHERE MARQUE = "PEUGEOT";
```

Le résultat affiché de la requête :

NombrePeugeot
3

Exemple : Pour afficher le prix minimal et le prix maximal des voitures.

La requête SQL est la suivante :

```
SELECT MIN(PRIX) AS PMIN, MAX(PRIX) AS
PMAX

FROM VOITURE;
```

Le résultat affiché de la requête :

Pmin	Pmax
15	30

2.5. Les sous relations

Syntaxe: Group by <liste_attributs> [Having <condition>] ;

Group by partitionne la relation en sous relations tel que chacune ait pour l'attribut indiqué par group by la même valeur.

Having a la même fonction que where, ne garde que les sous relations qui satisfont la condition.

Where est appliqué à la relation entière avant la partition (création des sous relations)

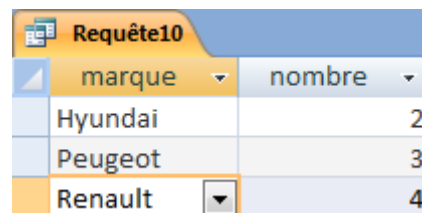
Having est appliqué à chaque sous relation, après donc la partition.

Exemple : Pour afficher le nombre de voitures de chaque marque.

La requête SQL est la suivante :

```
SELECT MARQUE, COUNT(*) AS NOMBRE
FROM VOITURE
GROUP BY MARQUE;
```

Le résultat affiché de la requête :



marque	nombre
Hyundai	2
Peugeot	3
Renault	4

Pour exécuter cette requête, le système crée des sous relations pour chaque marque, une sous relation a pour l'attribut marque la même valeur et ensuite, dans chaque sous relations, il compte le nombre de tuples cad le nombre de voitures d'une même marque.

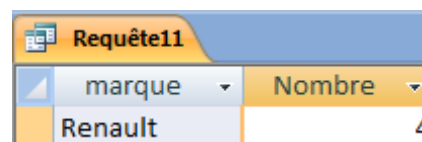
Remarque : un attribut mentionné dans la clause « group by » doit exister dans la clause « Select ».

Exemple : Modifier la requête précédente pour afficher le nombre de voitures de chaque marque ayant un prix minimale inférieur à 20 millions

La requête SQL est la suivante :

```
SELECT MARQUE, COUNT(*) AS NOMBRE
FROM VOITURE
GROUP BY MARQUE
HAVING MIN(PRIX) < 20;
```

Le résultat affiché de la requête :



marque	Nombre
Renault	4

Remarque : il est possible d'utiliser une fonction agrégat dans la clause « Having ».

2.6. Les requêtes multi-relations (jointures)

Une requête multi_relation peut s'écrire sous deux formes :

- Forme procédurale (FP) Des requêtes imbriquées utilisant l'opérateur in. Dans chaque requête, une seule table est utilisée à la fois.
- Forme relationnelle (FR) Une seule requête. Il est possible d'utiliser plusieurs tables dans la clause « From ».

Dans le cas où nous avons une liste de relations, il peut être nécessaire de préciser à quelle relation appartient chacun des attributs. Cette ambiguïté est levée de deux manières : utilisation de variables de désignation ou des noms de relations,

Exemple : soit le schéma relationnel suivant :

Voiture (NIM, marque, modèle , prix , option)

Client (CIN, nom, ville , telephone)

Appartenir (CIN, NIM)

Pour désigner l'attribut NIM de la relation Voiture, on peut écrire Voiture.NIM ou bien utiliser une variable de désignation V pour voiture par exemple et écrire V.NIM.

Exemple : donner le nom des clients ayant des voitures de modèle 307 et de marque Peugeot.

Forme Procédurale :

```
Select nom
From client
Where cin in (select cin
              From Appartenir
              Where NIM in (select NIM
                          From voiture
                          Where (modele="307") and
                                (marque="Peugeot")));
```

Forme relationnelle :

Une seule expression de sélection (1 seul SELECT)

```
Select nom
From client , Appartenir , Voiture
Where (client.cin = Appartenir.cin) and (Appartenir.NIM = Voiture.NIM)
      and (modele="307") and (marque="Peugeot" );
```

Ou bien, en utilisant des variables de désignation :

```
Select nom
From client C , Appartenir A , Voiture V
Where (C.cin = A.cin) and (A.NIM = V.NIM) and (modele="307") and (marque="Peugeot" );
```

Exemple: Soit le schéma relationnel suivant :

Matière (Mcode, Titre, niveau)

VolumH (Mcode, Type, Nbh)

Un extrait de ces tables est le suivant :

matières			
	Mcode	Titre	niveau
+	CP2	Comptabilité	2
+	FR1	FançaisEcrit	1
+	IF1	Informatiquel	1
+	IF3	Informatiquell	2
+	MT1	MathEco	2

VolumeH			
	Mcode	Type	Nbh
	CP2	Cours	30
	CP2	TD	15
	IF1	Cours	20
	IF1	TP	25
	IF3	Cours	15
	IF3	TD	15
	IF3	TP	15

Question : Donnez pour chaque matière: le code, le titre, le niveau et les volumes horaire (cours , td et/ou TP)

Cette requête doit afficher des attributs de la table matière et des attributs de la table VolumeH. Il faut donc réaliser une jointure entre les deux tables.

Forme relationnelle :

<pre>SELECT * FROM MATIÈRES , VOLUMEH WHERE (MATIÈRES.MCODE = VOLUMEH.MCODE);</pre>	<p>Ou bien</p> <pre>SELECT * FROM MATIÈRES M , VOLUMEH V WHERE (M.MCODE = V.MCODE);</pre>
-------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------

Forme procédurale :

```
SELECT *
FROM MATIÈRES
WHERE MCODE IN (SELECT MCODE
FOM VOLUMEH);
```

Le résultat affiché de cette requête :

Requête1					
matières.Mcode	Titre	niveau	volumeH.Mcode	Type	Nbh
IF1	InformatiqueI	1	IF1	Cours	20
IF1	InformatiqueI	1	IF1	TP	25
CP2	Comptabilité	2	CP2	Cours	30
CP2	Comptabilité	2	CP2	TD	15
IF3	InformatiqueIII	2	IF3	Cours	15
IF3	InformatiqueIII	2	IF3	TD	15
IF3	InformatiqueIII	2	IF3	TP	15

Question : Donnez le code des matières réalisées en cours et aussi en TP.

La requête SQL est la suivante :

```
SELECT X.MCODE
FROM VOLUMEH X, VOLUMEH Y
WHERE (X.TYPE = "COURS") AND (Y.TYPE = "TP") AND (X.MCODE = Y.MCODE);
```

2.7. Les opérateurs ANY, ALL, IN

Soit x l'élément (attribut(s) ou expression(s) retournant une valeur simple),

Soit E l'ensemble des valeurs retournées par la sous requête ou une énumération de valeurs.

Soit opc les opérateurs de comparaison.

Syntaxe :

X opc ANY (requête) retourne vrai si $\exists y \in E$ tel que x opc y, retourne faux sinon

X opc ALL(requête) retourne vrai si $\forall y \in E$ tel que x opc y, retourne faux sinon

X in (requête) retourne vrai si $x \neq y \forall y \in E$, retourne faux sinon.

Les travaux dirigés - MCD & MLD

Sujet 1: Gestion d'une Pressing

Un nouveau pressing s'est installé à Fès, il a instauré un système de carte magnétique qui fonctionne de la manière suivante :

Chaque client dispose d'une carte lors de sa première visite. Les clients doivent présenter leurs cartes à chaque visite. Quand un client apporte des vêtements à nettoyer, l'employé du pressing lui demande sa carte afin d'y inscrire le numéro du vêtement apporté et la date de remise. Le client paie le montant dû et l'employé lui remet une facture datée correspondant aux vêtements déposés pour le nettoyage.

Les vêtements sont classés par type (pantalon, veste, jupe, ...) et nettoyés suivant la qualité demandée (standard, luxe, ...). Le prix à payer pour un vêtement dépend du type de vêtement et de la qualité demandée. Le prix peut varier aussi en fonction de la période, tous les six mois le pressing envoie à ses clients ayant au moins utilisé 3 fois le service « qualité luxe » une offre spéciale de réduction de prix d'une durée limitée.

A partir de la date de remise des vêtements du client, quand celui-ci vient les chercher, il glisse sa carte dans l'automate et ses vêtements lui sont automatiquement présentés.

Sujet 2: Agence immobilière

Une agence de location de maisons à Fès désire gérer sa liste de logements. Elle voudrait en effet connaître l'implantation de chaque logement (le quartier par exemple : Sidi Brahim, Saada, Bensouda, ...) ainsi que les personnes qui les occupent (les signataires uniquement). Le loyer dépend d'un logement, mais en fonction de son type (maison, studio, T1, T2...) l'agence facturera toujours en plus du loyer la même somme forfaitaire à ses clients. Par exemple, le prix d'un studio sera toujours égal au prix du loyer + 100 dh de charges forfaitaires par mois. Pour chaque logement, on enregistre l'adresse, la superficie et le loyer. Quant aux individus qui occupent les logements (les signataires du contrat uniquement), on notera seulement leurs noms, prénoms, date de naissance et numéro de téléphone. Pour chaque quartier, on désire connaître le niveau (Haut standing, économique, ...) ainsi que la distance séparant le quartier de l'agence.

On ne gèrera pas l'historique de l'occupation des logements par les individus.

On considère de plus qu'un individu ne peut être signataire que d'un seul contrat.

Sujet 3 : Bibliothèque

Une bibliothèque de prêt dispose d'un certain nombre d'ouvrages, classés par rayon (Littérature, Histoire, Géographie, etc.). Chaque ouvrage est l'œuvre d'un ou plusieurs auteurs, et doit également être référencé selon un certain nombre de mots-clés.

Chaque adhérent peut emprunter jusqu'à 3 livres en même temps, et dispose d'un certain délai passé lequel il doit recevoir des relances puis des pénalités.

La bibliothèque possède plusieurs exemplaires de chaque ouvrage. Elle enregistre uniquement les emprunts présents (il n'y a pas d'historique des emprunts passés).

Travail à faire : Pour chaque sujet ci-dessus,

1. Présenter le dictionnaire des données, préciser les entités et leurs propriétés
2. Elaborer le schéma du modèle conceptuel des données
3. Donner le MLD correspondant à votre MCD, préciser les clés primaires et les clés étrangères.

TD- Manipulation de données, Le langage SQL

Sujet : Gestion de la scolarité universitaire.

Soit le schéma relationnel suivant

Etudiant (CNE, nom , date_naiss, niveau, adresse , ville)

Enseignant (nomens, fonction)

Module (Mcode, titre , nomens)

Liste (filiere , Mcode)

Intervient (Mcode , nomens)

Exam (CNE , Mcode , note1)

Oral (CNE , Mcode , note2)

Alieu (Mcode , jour , heure , salle)

Occupe (nomens , jour , heure , salle)

TRAVAIL A FAIRE : Répondre aux questions suivantes en écrivant des requêtes dans le langage SQL:

1. Donnez la liste des étudiants de niveau 3.
2. Donnez le nom des étudiants qui habitent à Fès.
3. Donnez le nom des enseignants différents qui occupent la salle 4 le Lundi.
4. Donnez les salles ou il y a cours le lundi ou le Jeudi.
5. Donnez le nom des étudiants ayant passé l'examen en programmation ou en module 6.
6. Donnez le nom de la filière comportant le module de code 8.
7. Donnez le nom de la filière comportant le module de titre "Informatique de gestion".
8. Donnez le nom des étudiants ayant obtenu une note supérieure à 12 dans le module de titre "Informatique de gestion".
9. Donnez le nombre d'étudiants différents qui ont passé un oral.
10. Donnez la meilleure note de l'examen du module2.
11. Donnez le nom des étudiants qui ont obtenu la meilleure note à l'examen du module 2.
12. Donnez la liste des filières comprenant plus de 4 modules.
13. Donnez le nombre de salles comportant plus de 4 séances.
14. Donnez la moyenne générale de chaque étudiant à l'examen.

Travaux Pratiques sous ACCESS

Atelier1 : Création d'une Base de Données

I- Création d'une nouvelle base de données :

Dans un dossier portant votre nom, créer une nouvelle base de données

II- Création des tables en mode « Création »

Table1 :

Dans la base de données créée, utiliser le mode création pour créer une Table nommée « Client » avec les champs suivants et avec les types de données appropriés.

Nom du champ	Type de données	Taille	Liste de choix
Nclient	Numérique	octet	
nomclient	Texte	50	
Prefixe	Texte	4	Liste déroulante, valeurs : Mr ; Mme ; Mlle
Contact	Texte	50	
Adresse	Texte	255	
Ville	Texte	10	Liste déroulante, valeurs : (voir plus loin)

Comme le montre la figure suivante :

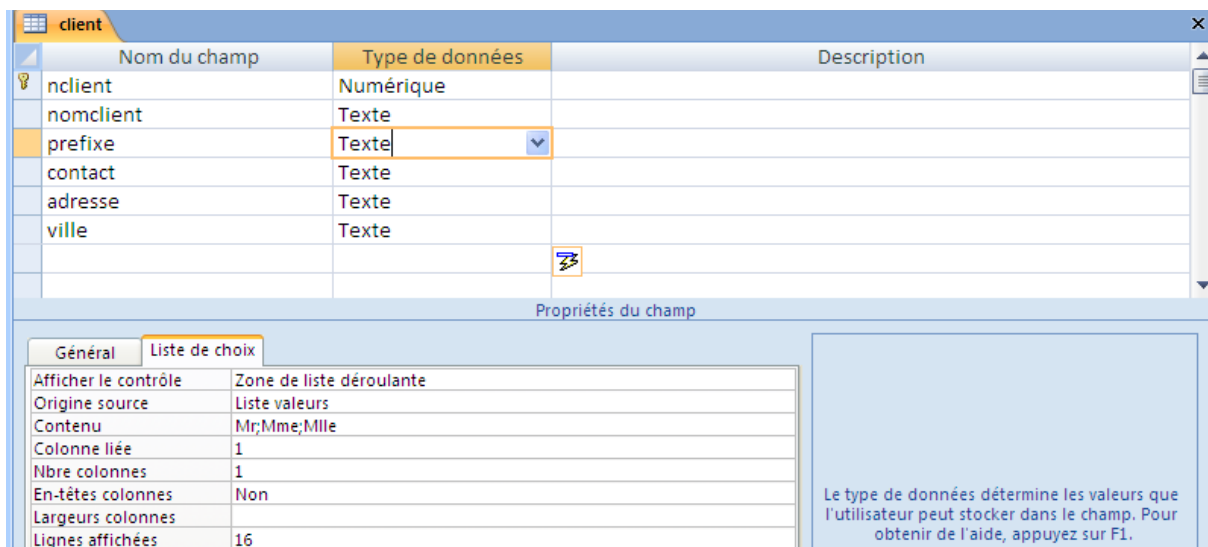


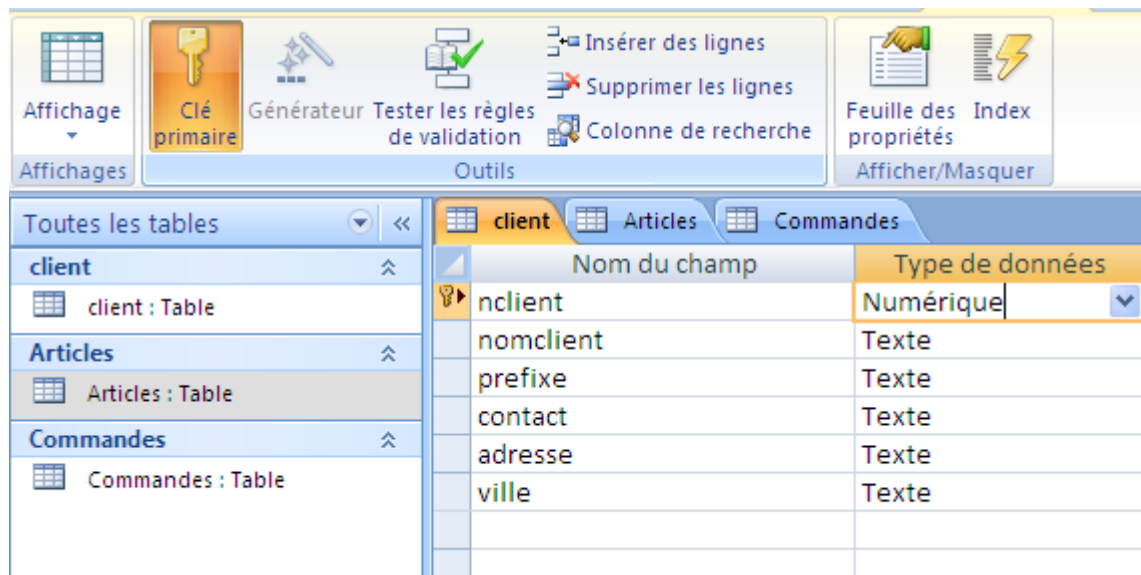
Figure1

Comment créer une liste déroulante ou une liste de choix ?

- En mode création sélectionner le champ en question
- Dans la partie inférieure de l'écran choisir l'onglet « Liste de choix »
- Dans la ligne « Origine source », choisir « Liste valeur »
- Dans la ligne « contenu », saisir les valeurs de votre liste de choix séparées par des points-virgules (Voir Figure-1)
 - o Pour le champ « Préfixe » les valeurs sont : Mr ; Mme ; Mlle
 - o Pour le champ « ville » les valeurs sont : Casablanca; Rabat; Mohammedia; Fès; Tanger; Oujda; Marrakech; Agadir.

III- Création d'une clé primaire

1. La table est ouverte en mode création
2. Sélectionner le champ devant être clé primaire de chaque table de votre base de données
3. Cliquer sur le bouton clé de la barre d'outils comme le montre la *figure_2*



- Figure-2

Remplir ensuite le tableau avec les données suivantes:

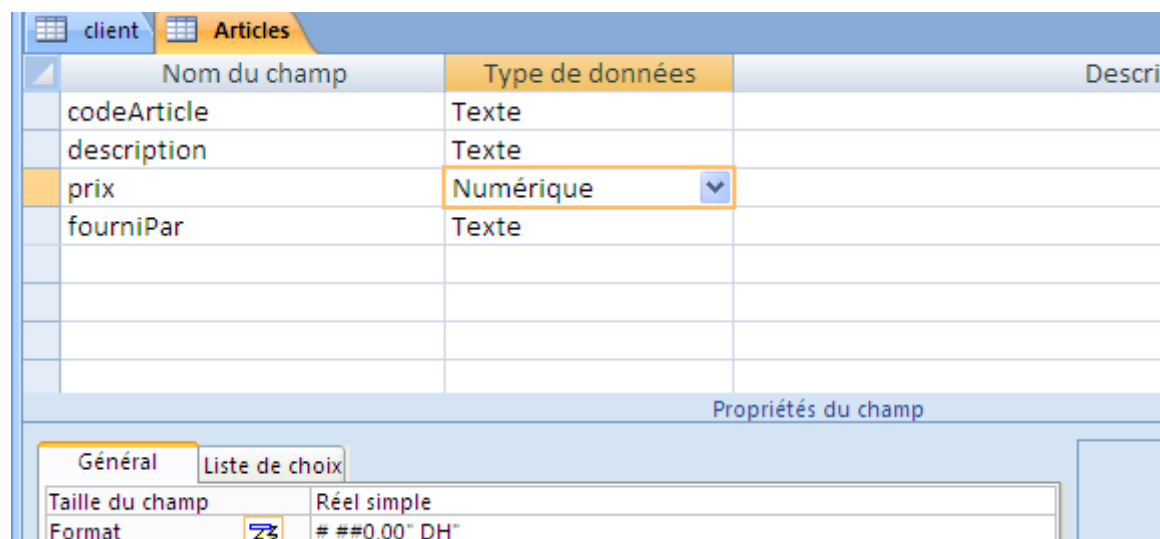
nclient	nomclient	prefixe	contact	adresse	ville
1	MANAGEM	Mr	BENNANI	23 Zone industrielle	Casablanca
10	NEXANS MAROC	Mr	BERRADA	42 Zone industrielle	Casablanca
11	OULMES	Mme	KARIMI	56, Bd Nador	Fès
12	COSUMAR	Mlle	MAHMOUDI	67, Rue de la résistance	Tanger
2	REBAB	Mme	HACHIMI	32, Zone industrielle	Agadir
3	S.M.IMITER	Mme	MANSOURI	21, RUE mohamed V	Rabat
4	AFRIQUIA GAZ	Mr	FAHIMI	7 Avenue des FAR	Mohammedia
5	SAMIR	Mr	FAKHIRI	8 Avenue Hassan II	Rabat
6	ingelwoud MAG.OXYGENE	Mme	SKALLI	7, Zone industrielle	Agadir
7	ALUMINIUM	Mme	CHOUKRI	36 Avenue Mohamed V	Casablanca
8	LAFARGE	Mr	LATIFI	34, Zone industrielle	Casablanca
9	SONASID	Mr	MAHIR	55, Zone industrielle	Rabat
*					

Table2 :

Dans la même base de données, cliquer sur « création de table » dans le menu « créer » pour créer la Table nommée « **Articles** » avec les champs suivants et avec les types de données appropriés.

Nom du champ	Type de données	Taille	Format
codeArticle	Texte	4	
Description	Texte	100	
Prix	Numérique	Réel simple	Monétaire avec DH
fourniPar	Texte	50	

Comme le montre la figure suivante :



Figure_3

Comment personnaliser l'écriture monétaire ? :

- En mode création sélectionner le champ en question
- Dans la partie inférieure de l'écran choisir l'onglet « Général»
- Dans la ligne « Format » choisir la monnaie en question ou bien saisir « # ##0,00" DH" » (Voir Figure-4)

Nom du champ	Type de données
code article	Texte
description	Texte
prix	Monétaire
fourni par	Texte

Général		Liste de choix
Format	# ##0,00" DH	
Décimales	Auto	
Masque de saisie		
Légende		
Valeur par défaut		
Valide si		
Message si erreur		
Null interdit	Non	
Indexé	Non	
Balises actives		
Aligner le texte	Général	

Figure-4

Passer en mode Feuille de données et remplir le tableau

codeArticle	description	prix	fourniPar
h1	Imrimante HP 501	1 500,00 DH	eric electronics
h2	Imprimante HP Laser	2 000,00 DH	eric electronic
h3	IBM, Core 2 Duo	12 000,00 DH	eric electronic
h4	Dell, Core 2 Duo	13 000,00 DH	lesley industries
h5	lecteur CD portable	400,00 DH	lesley industries
k1	USB 8Go	200,00 DH	kitware compagny
k2	USB 4Go	100,00 DH	kitware compagny
k3	HDD Externe	600,00 DH	super info
k4	Ecran LCD 17"	1 000,00 DH	super info
k5	Clavier infra rouge	100,00 DH	super info
*			

Table3 :

Créer dans la même base de données la table nommée « **Commandes** » avec les champs suivants et avec les types de données appropriés.

Nom du champ	Type de données	Taille
ncommande	Numérique	Entier
dateCommande	Date/Heure	
nclient	Numérique	Octet
article	Texte	4
quantité	Numérique	Octet
importé	Oui/Non	

Comme le montre la figure suivante :

Nom du champ	Type de données
nCommande	Numérique
dateCommande	Date/Heure
nClient	Numérique
article	Texte
quantité	Numérique
importé	Oui/Non

Propriétés

Général Liste de choix

Format Oui/Non

Figure_5

Passer en mode Feuille de données et remplir le tableau suivant:

nCommande	dateCommande	nClient	article	quantité	importé
21	14/05/2017	3 h2		2	<input checked="" type="checkbox"/>
22	17/09/2017	5 k4		1	<input checked="" type="checkbox"/>
23	08/10/2017	2 h5		2	<input checked="" type="checkbox"/>
24	13/10/2017	8 h3		2	<input type="checkbox"/>
25	15/05/2018	6 k3		8	<input type="checkbox"/>
26	19/05/2018	4 h4		1	<input checked="" type="checkbox"/>
27	08/09/2018	7 k1		5	<input type="checkbox"/>
28	15/05/2019	4 h1		1	<input type="checkbox"/>
29	22/08/2019	3 h1		2	<input checked="" type="checkbox"/>
30	08/09/2019	3 h4		2	<input type="checkbox"/>
31	20/09/2019	8 h1		7	<input type="checkbox"/>
32	22/11/2019	13 h1		1	<input type="checkbox"/>
33	22/11/2019	1 h1		1	<input type="checkbox"/>
34	07/12/2019	1 h1		2	<input type="checkbox"/>

IV. Création des relations

- ✓ Fermer toutes les tables
- ✓ Dans l'onglet « Outil de base de données » cliquer sur le bouton « Relation » (voir *figure_6*)
- ✓ Dans la boîte de dialogue qui s'affiche, double-cliquez sur la 1ère table à lier puis sur la 2^{ème} table puis sur la 3^{ème} et fermer la boîte.
- ✓ Dans la nouvelle boîte de dialogue affichant les champs des 2 tables, faites glisser le champ comportant la clé primaire sur le champ homologue de l'autre table (clé étrangère).
- ✓ Dans la boîte de dialogue qui apparaît, cocher la case "Appliquer l'intégrité référentielle" et cliquer sur Créer (*figure-7*)

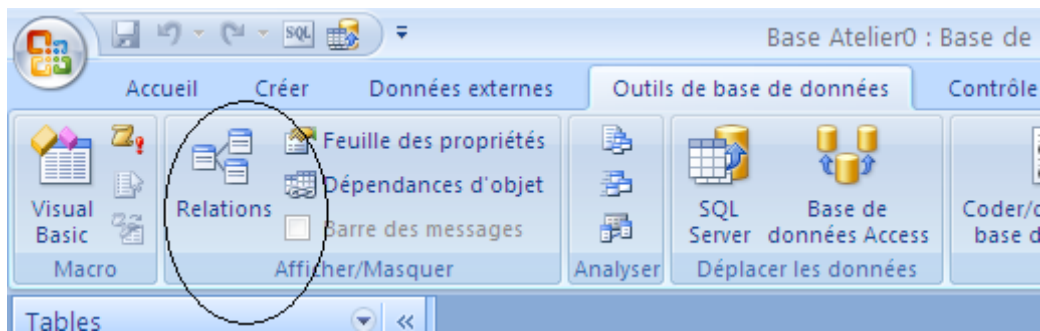


Figure-6

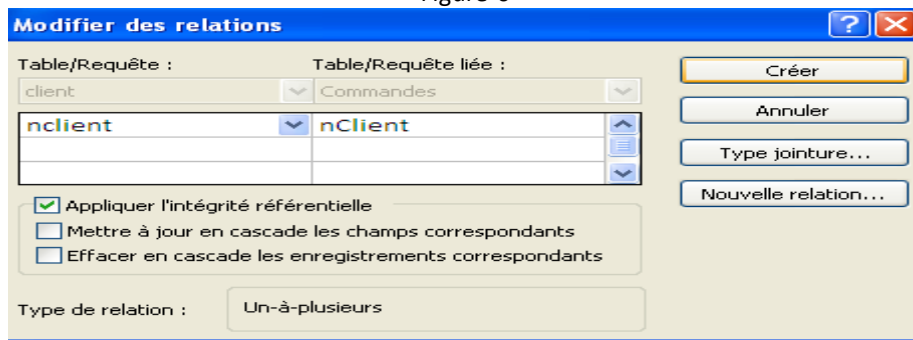


Figure-7

Le résultat des relations est le suivant :

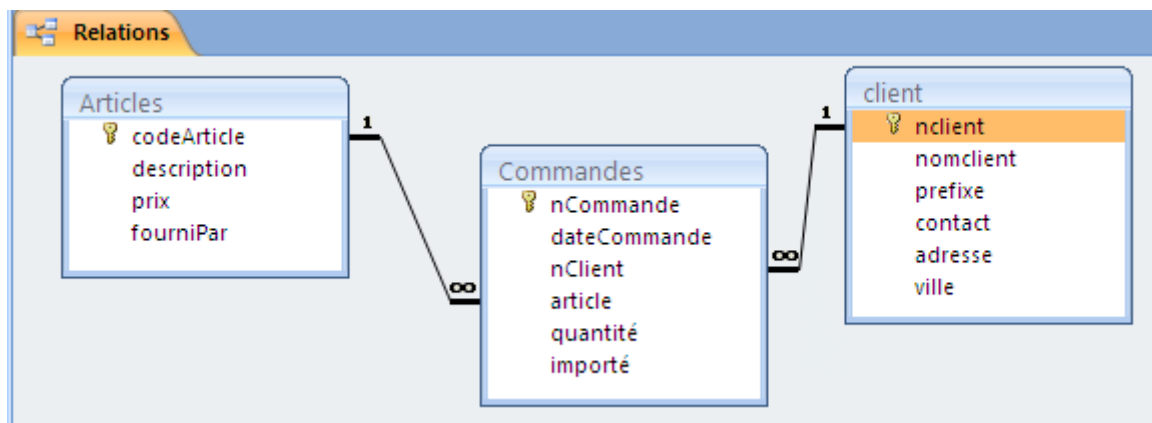


Figure-8

Nb : L'attribut clé primaire et celui de clé étrangère doivent avoir exactement le même type.

Une valeur ne peut apparaître dans la clé étrangère si elle n'existe pas dans la clé primaire.

Atelier2 : Manipulation des données : Le langage SQL

1. Donnez le numéro et nom des clients originaires de « Casablanca » classés par ordre alphabétique croissant (nom).

2. Donnez la description et le code article dont le prix est supérieur à 150 DH.

3. Donnez la liste des articles fournis par « Super info ».

4. Donnez le numéro et la date des commandes du client n°1.

5. Donnez la liste des articles (code, description, prix et prixTTC 20% de TVA) dont le prix est compris entre 200 DH et 500 DH. Afficher le résultat par ordre décroissant de prix.

6. Donnez la liste des commandes (numéro, date et article) contenant des articles importés ayant été passés le mois de septembre.

7. Donnez le nombre de commandes de chaque client.

8. Donnez le numéro des clients ayant passé plus de deux commandes.

9. Donnez la liste des commandes (numéro, date, numéro de client et nom) des clients originaires de « Agadir ».

10. Donnez la liste des articles (code et description) d'un fournisseur donné [paramètre saisi au clavier].

11. Donnez le nom des clients et personne à contacter (avec préfixe) ayant commandés des articles importés.

12. Donnez le prix minimal et le prix maximal des articles commandés.

13. Donnez pour chaque client, son numéro, son nom et le montant à payer pour toutes ses commandes.