CHAPITRE 1 : Introduction aux bases de données

PLAN :

1 INTRODUCTION

Les entreprises et les organisations connaissent depuis longtemps la place prépondérante de l'information et de son rôle crucial dans l'amélioration de leur rentabilité. Le traitement automatique de l'information, grâce à l'informatique a permis de rendre des services considérables à ces entreprises en améliorant leur fonctionnement.
Tout au début de l'utilisation de l'informatique, les méthodes classiques analysaient les collections de données du monde réel pour en faire des fichiers. Chaque application avait son ou ses fichiers, que l'on traitait séparément.

Rappel : Définition d'un fichier
Un fichier est un ensemble d'enregistrements de même descripteur ; un descripteur est une suite de rubriques ou champs représentant des catégories de valeurs ; un enregistrement est une succession de valeurs de ces champs.

L'amélioration des capacités et techniques de stockage par des mémoires plus grandes et plus souples et des techniques d'accès plus performantes ont fait évolué les exigences des utilisateurs. Ceux-ci ne se contentaient plus de l'automatisation d'applications routinières, mais exigeaient des :

- Informations complètes,
- Informations cohérentes,
- Informations structurées,
- Informations disponibles rapidement pour plusieurs utilisateurs.

Très vite les Systèmes de Gestion de Fichiers (SGF), ont prouvé leurs limites par leur :

- Insuffisance dans la gestion des données en mémoire secondaire (données persistantes),
- Incapacité à restituer rapidement l'information

LA SOLUTION : APPARITION DES CONCEPTS DE BASE DE DONNEES (BD) (1962-1963) ET DE SON SYSTEME DE GESTION (SGBD)

2 LES CONCEPT DE BASE DE DONNEES ET SYSTEME DE GESTION DE BD

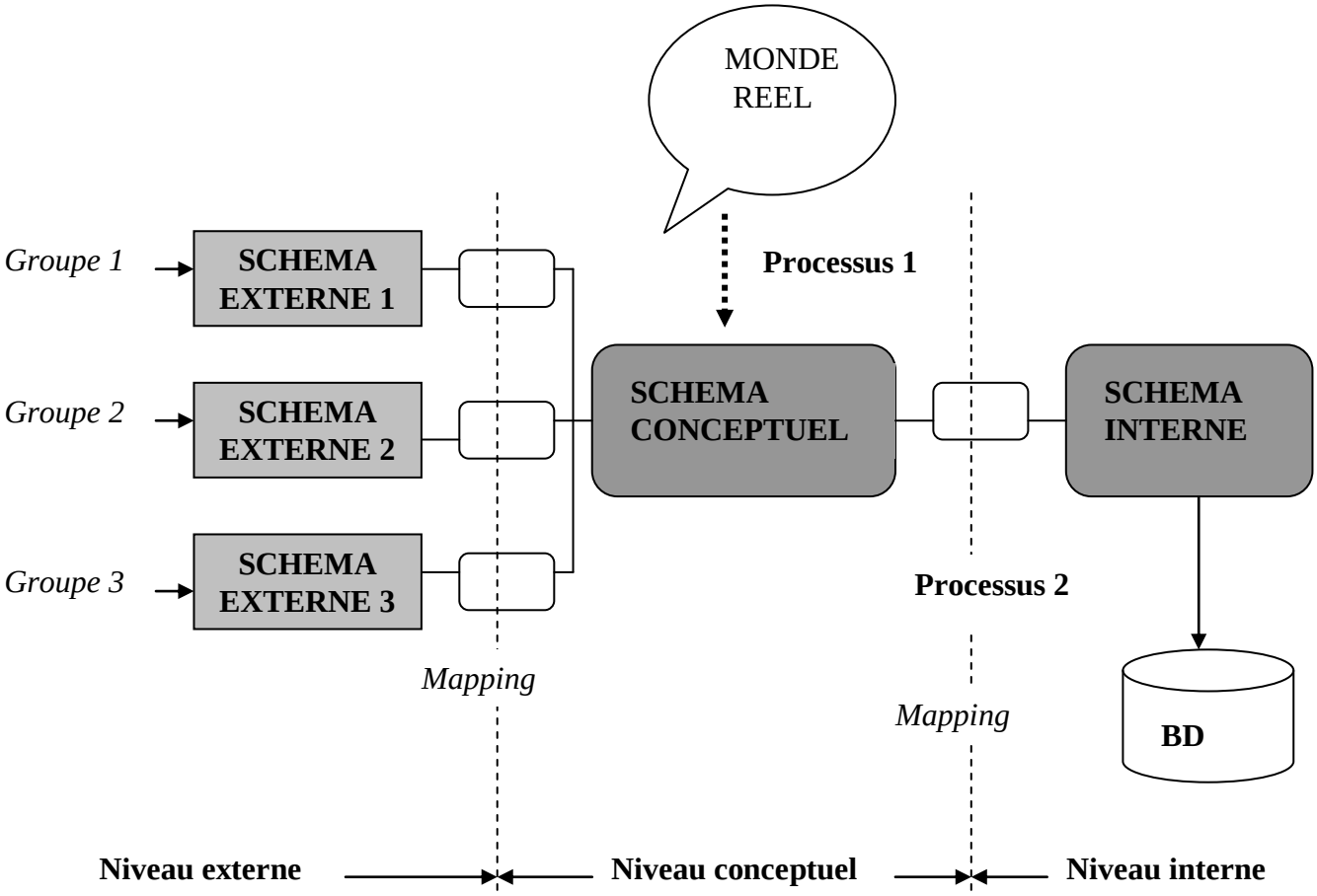
2.1 DEFINITIONS :

- **Une base de données (BD)** est une collection de données, représentations partielles d'aspects pertinents de la réalité organisationnelle sur lesquels on souhaite être renseigné. Ces collections de données, aussi cohérentes que possibles, sont mémorisées sur des supports accessibles par un ordinateur, avec une **redondance calculée** et structurées pour satisfaire simultanément **plusieurs utilisateurs** de façon sélective.
- **Le système de gestion de bases de données (SGBD)** est un logiciel qui interagit avec la BD. Il est complémentaire à un SGF et assure les fonctions sur lesquelles celui-ci est défaillant, entre autres des accès sophistiqués et la prévention des conflits aux accès concurrents.

Avant de créer et de faire fonctionner une base de données, un important travail de conception qui aboutit au schéma de la base doit être réalisé. Ce travail obéit aux différentes étapes suivantes.

2.2 LES DIFFERENTS NIVEAUX DE REPRESENTATION D'UNE BASE DE DONNEES

Nous pouvons schématiser les différents niveaux de représentation d'une BD par la figure suivante :



- La norme ANSI/SPARC du groupe CODASYL (1975) a permis de définir trois niveaux de représentation :EXTERNE, CONCEPTUEL, INTERNE.

- Le processus de développement d'une base de données comprend deux grandes phases :

- Une phase de **conception** dont le résultat est le **SCHEMA CONCEPTUEL**. Au cours de cette phase, la validation s'effectue à travers les **SCHEMAS EXTERNES**
- Une phase de réalisation dont le résultat est le **SCHEMA INTERNE** et la **BASE DE DONNEES**

2.2.1 LE NIVEAU CONCEPTUEL

- Le schéma conceptuel est la « charpente » d'une BD. Il décrit en termes abstraits la réalité organisationnelle et ses règles de gestion.
- Le processus de conception consiste à traduire les objets du monde réel en catégories d'objets suivant des **MODELES** bien définis.
- Il existe plusieurs types de modèles que nous pouvons classer en trois catégories :
    - **Les modèles de 1ère génération : décennie 60**
        - HIERARCHIQUE
        - RESEAU
    - **Les modèles de 2ième génération : les décennies 70 et 80**
        - E/A : ENTITE /ASSOCIATION
        - RELATIONNEL
        - LES RESEAUX SEMANTIQUES
    - **Les modèles de 3ième génération : la décennie 90**
        - LE MODELE OBJET

2.2.2 LE NIVEAU EXTERNE

- Un schéma externe appelé aussi **VUE** dans les systèmes relationnels est une perception des données par un programme d'application.
- Une vue est un **SOUS-SCHEMA** d'un schéma conceptuel.
- Le schéma externe peut contenir des informations complémentaires (par exemple des informations de calcul).
- Les différents schémas externes permettent la validation du schéma conceptuel.
- Il existe une technique de construction du schéma conceptuel à partir de l'intégration de vues.

2.2.3 LE NIVEAU INTERNE

- Le schéma interne est constitué de deux schémas.
    - LE **SCHEMA LOGIQUE** : Il est construit à partir du schéma conceptuel par transformation (mapping) en utilisant des règles de correspondance générales entre structures conceptuelles et structures de stockage et en définissant les chemins d'accès.
    - LE **SCHEMA PHYSIQUE :** Il est construit à partir de la connaissance du SGBD utilisé.

2.3 MISE EN ŒUVRE D'UNE BASE DE DONNEES

La mise en œuvre d'une BD nécessite l'exécution de plusieurs tâches dont les principales sont la définition et la manipulation de la base de données.

TACHE N°1 : DEFINITION DE LA BASE DE DONNEES

- La première tâche consiste à représenter le schéma conceptuel à l'aide d'un langage ; le **LDD L**angage de **D**éfinition de **D**onnées.
- LE LDD est propre à chaque SGBD et permet la représentation :

- Des classes d'entités
- Des classes d'associations entre les entités
- Des contraintes d'intégrité.

- LE LDD permet aussi la spécifications des schémas externes.

EXEMPLE 1: utilisation de SQL

CREATE TABLE VOL (NUMERO : INT, DATE : CHAR(6), SIEGE : INT, VILLEDEP : CHAR(10), VILLEARR : CHAR(10)) ;

CREATE INDEX FOR VOL ON NUMERO

TACHE N°2 : MANIPULATION DES DONNEES DE LA BASE

- Une fois le schéma défini, il faut interroger, mettre à jour la BD grâce au **L**angage de **M**anipulation de **D**onnées : LE **LMD**.
- L'utilisateur interroge la BD à l'aide d'une requête (question).
- Le LMD comporte des instructions de :
  - Recherche
  - Insertion et suppression
  - Mises à jour.
- Le LMD peut être autonome ou inclus dans un langage hôte, c'est-à-dire utiliser des langages de programmation traditionnels dans lesquels des ordres d'appel de procédures fournis par le SGBD permettent de réaliser des opérations spéciales sur les BD.

EXEMPLE 2 : Utilisation de SQL

UPDATE VOL
SET SIEGE = SIEGE +4
WHERE NUMERO = 1023 AND DATE = 'AOUT 99'

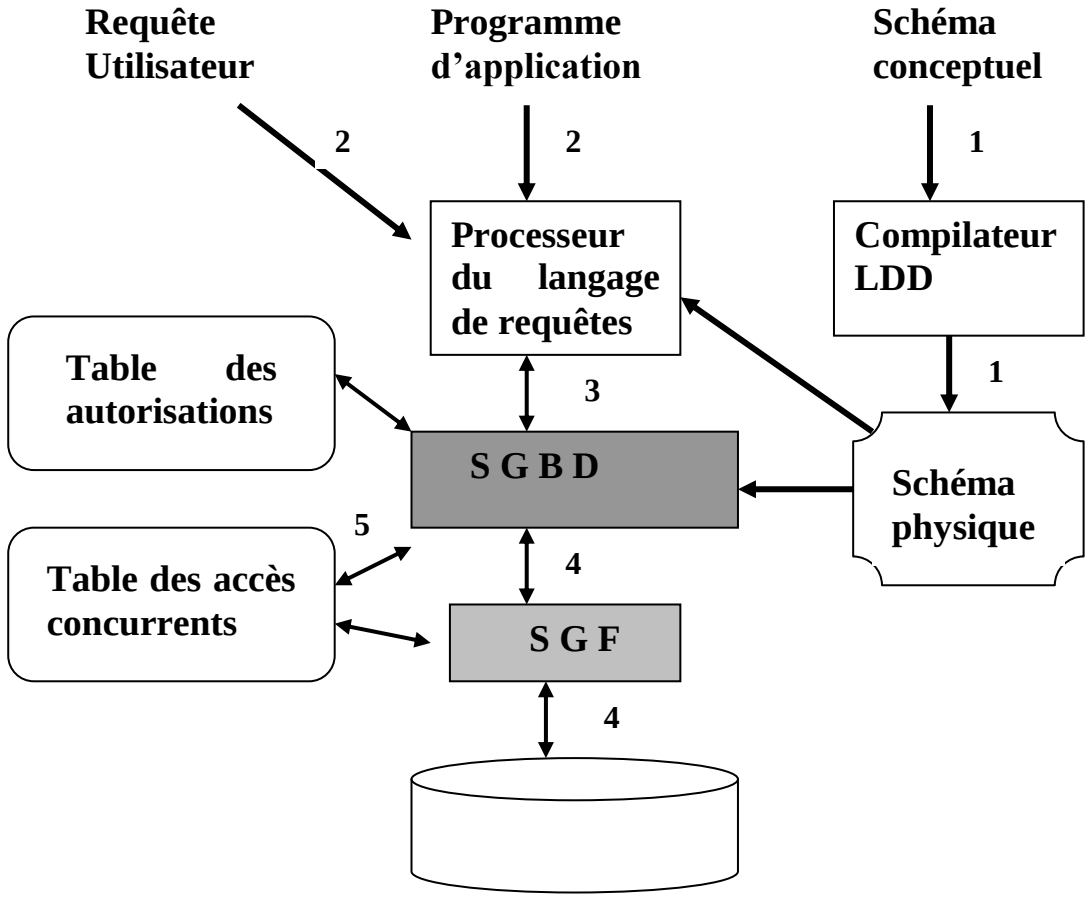
EXEMPLE 3 : utilisation d'un LANGAGE HOTE

…..
CALL GET (B)
…
A := B+1
…
CALL STORE (A)

REMARQUE : souvent un même langage assure la définition et la manipulation de données (par exemple SQL).

3 ARCHITECTURE ET ROLES D'UN SYSTEME DE BD

3.1 ARCHITECTURE



1. Le schéma conceptuel est traduit en une description interne grâce au LDD.

2. Le processeur de langage de requêtes peut recevoir des données de deux sources :
   - D'un terminal : PROGRAMME SQL par exemple
   - D'un programme d'application écrit en C par exemple où figure des ordres d'appel de type CALL.

3. Le SGBD analyse la requête ou le programme d'application pour en vérifier la conformité et la cohérence.

4. Le SGBD traduit les commandes de la requête ou du programme d'application en commandes exécutables par le SGF sur la BD.

5. Le SGBD maintient et accède aux tables des autorisations et des accès concurrents.

3.2 ROLES D'UN SGBDR

Les objectifs d'un SGBD sont les suivants :

1. Indépendance physique

2. Indépendance logique
3. Manipulation des données par des non informaticiens
4. Efficacité des accès aux données
5. Administration centralisée des données
6. Non redondance des données
7. Cohérence des données
8. Partageabilité des données
9. Sécurité et confidentialité des données

Tous ces objectifs sont possibles grâce :

- Au support de **modèles de données**,
- A la définition de **langages de requêtes**,
- A la définition du concept **de transaction**.

3.2.1 OBJECTIF 1 : INDEPENDANCE PHYSIQUE

Réaliser l'indépendance physique c'est réaliser l'indépendance entre structure de stockage (niveau physique) et structures de données, représentations abstraites du monde réel (niveau interne).

- Au niveau logique, les structures de données sont définies sur la base d'entités et liens entre entités.
- Au niveau physique, les structures de stockage associées sont des enregistrements, des fichiers etc. Une entité de niveau logique peut être implémentée par un ou plusieurs enregistrements du niveau physique sans que le programme d'application qui manipule cette entité connaisse cette correspondance.

L'indépendance physique permet de changer l'organisation physique des données en mémoire secondaire sans avoir à modifier les programmes d'applications.

3.2.2 OBJECTIF 2 : INDEPENDANCE LOGIQUE

L'indépendance logique des données désigne l'aptitude à modifier le schéma –en ajoutant des relations, attributs) sans changer les applications. [DATE]. C'est autoriser plusieurs visions différentes de différents groupes de travail sur la même base.

- Ceci est possible grâce à la notion de vue.
- La modification d'une vue n'implique pas la modification de la base.

Il s'agit de l'indépendance des niveaux conceptuel et externe.

3.2.3 OBJECTIF 3 : MANIPULATION DES DONNEES PAR DES NON INFORMATICIENS

Les non informaticiens manipuleront la BD grâce à des langages non procéduraux, c'est-à-dire où les chemins d'accès ne seront pas précisés. Ce sont les langages de quatrième génération et sont les LMD des SGBD.

3.2.4 OBJECTIF 4 : EFFICACITE DES ACCES AUX DONNEES

Les accès aux données seront plus efficaces que dans les SGF grâce notamment :

- Au développement d'index sophistiqués
- A l'existence de plusieurs chemins d'accès à une donnée
- A l'existence de techniques d'optimisation de requêtes qui sélectionnent le chemin optimal à une donnée.

3.2.5 OBJECTIF 5 : ADMINISTRATION CENTRALISEE DES DONNEES

- Administrer des données consiste à :
- Définir des structures de stockage,
- Définir des structures de données,
- Assurer le suivi et le contrôle de leur évolution.
- Pour cela un administrateur de la base sera chargé de ces fonctions.

3.2.6 OBJECTIF 6 : NON REDONDANCE DES DONNEES

La conception intégrée de la BD grâce à la notion de schéma de données (qui est une représentation globale des informations à l'aide de modèles de données pour un ensemble d'applications) évite les redondances constatées dans les SGF.

3.2.7 OBJECTIF 7 : COHERENCE DES DONNEES

Les données de la base obéissent à des règles appelées **contraintes d'intégrité (CI).**
Une CI est une assertion que doit vérifier le SGBD à chaque fois que la donnée sur laquelle elle est définie est sollicitée (déchargement de l'utilisateur lors des opération de création, modification, suppression).

Il existe deux types de CI : CI statique et CI dynamique. Une CI statique est définie sur les aspects structurels de données, alors qu'une CI dynamique est définie sur leurs aspects comportementals.

Une BD cohérente est une BD dont les CI sont toujours vérifiées lors d'opérations sur la BD.

3.2.8 OBJECTIF 8 : PARTAGEABILITE DES DONNEES

Le SGBD doit permettre à plusieurs applications de partager les données. Pour cela, il doit gérer les conflits d'accès (les détecter et les solutionner) .
Ceci est possible grâce à la notion de **TRANSACTION** et de définition **d'ALGORITHMES DE GESTION DE LA CONCURRENCE.**

Une transaction est un ensemble indécomposable d'opérations sur la BD dont l'exécution maintient la BD dans un état cohérent.

3.2.9 OBJECTIF 9 : SECURITE ET CONFIDENTIALITE DES DONNEES

Les données doivent être protégées contre les pannes et contre les accès mal intentionnés.

Protection contre les pannes
On distingue deux types de pannes :

- Pannes simples caractérisées par la perte du contenu de la mémoire centrale
- Pannes catastrophiques caractérisée par la perte du contenu des mémoires secondaires

Dans les deux cas, il est possible de récupérer un état cohérent des données grâce à l'existence et à la gestion de journaux de transactions, qui gardent la trace d'exécutions antérieures.

Protection contre les accès mal intentionnés
La définition de droits d'accès, de mots de passe etc. gérés par le SGBD permet la confidentialité des données de la base et leur inviolabilité.

4 LES MODELES DE DONNEES

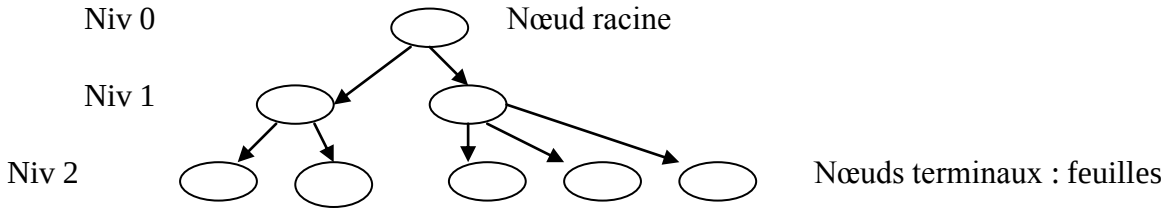
Différents modèles de données existent liés aux différentes générations de bases de données. La première décennie (années 60) caractérise la première génération et est basée sur les modèles hiérarchique et réseau. La deuxième décennie (années 70) a vu l'émergence de modèles plus évolués, à savoir les modèles Entité/Association, Relationnel et réseaux sémantiques qui ont caractérisé la deuxième génération. Les années 90 ont vu apparaître des modèles permettant la modélisation de structures complexes de données, ce sont les modèles objets, qui caractérisent la troisième génération. Nous présentons ici deux exemples de bases de données de première génération : les bases de données hiérarchiques et bases de données réseaux, ainsi qu'un type de base de données de deuxième génération, les bases de données relationnelles.

4.1 LE MODELE HIERARCHIQUE

Une base données hiérarchique est une BD qui utilise une structure de données hiérarchique appelée modèle hiérarchique.

4.1.1 LA STRUCTURE DE DONNEES

La structure de données d'un modèle hiérarchique est une **arborescence** où les nœuds représentent des classes d'entités et les arcs sont des associations entre ces entités**.**



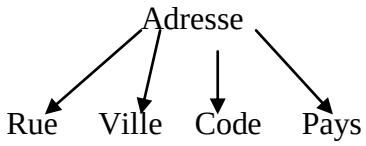
On appelle fils d'un nœud Père, l'ensemble des nœuds à une distance 1 du nœud père.

- Un attribut ou Item ou encore constituant, est la plus petite unité manipulable et prenant des valeurs dans un domaine de définition.

Exemple : nom, rue, numéro

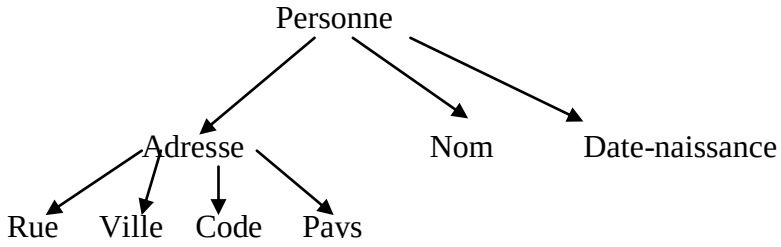
- Un groupe simple est un ensemble d'attributs formant une arborescence.

Exemple :



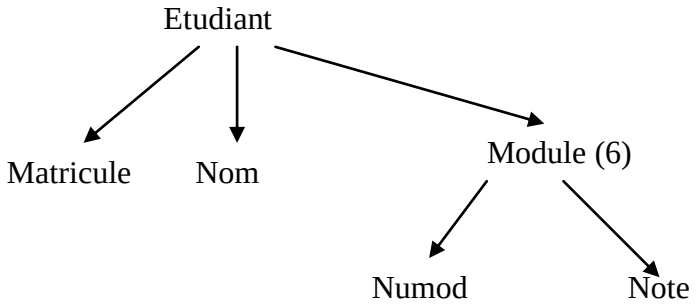
- Un groupe composé est un ensemble tel que chaque attribut est un groupe simple ou un attribut.

Exemple :



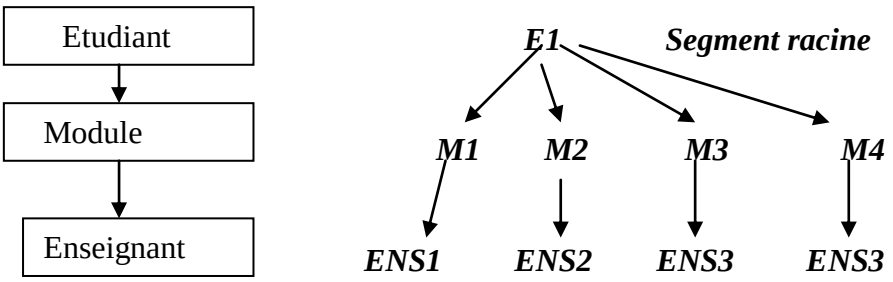
- Un groupe répétitif est un groupe composé dont un groupe peut se répéter plusieurs fois.

Exemple :



- Notion d'arborescence valuée : une arborescence valuée est un arbre dont les nœuds sont valués par des enregistrements logiques appelés **segment.**

Exemple :

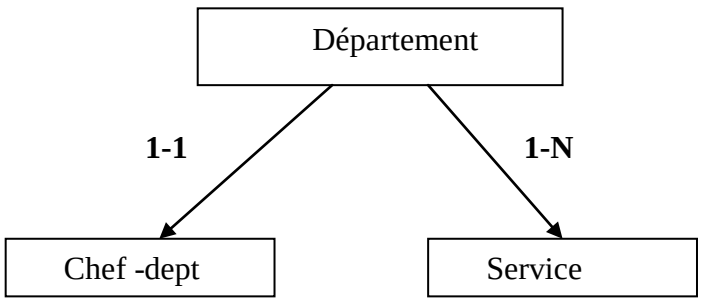


- **Notion de schéma conceptuel hiérarchique :**

Un schéma conceptuel hiérarchique est un ensemble d'objets possédant des propriétés reliés par des liens hiérarchiques. Les liens pris en charge par le modèle hiérarchique sont :

- 1-1 : un père n'a qu'un fils
- 1-N : un père à N fils

Exemple :



4.1.2 IMPLEMENTATION PHYSIQUE D'UNE BD HIERARCHIQUE

Définition : une BD hiérarchique est un ensemble d'arborescences valuées faisant référence à un même schéma.
Il existe deux manières d'implémenter une BD hiérarchique, en utilisant une structure séquentielle ou une structure de liste.

- Structure séquentielle

Une arborescence valuée = un enregistrement de longueur variable.
A l'intérieur de l'enregistrement, les segments sont placés dans un ordre défini.

Exemple : enregistrement1 : *d1--- c1----s1----s2----s3,*

enregistrement2 : *d2---c2---s4—s5----s6-----s7*

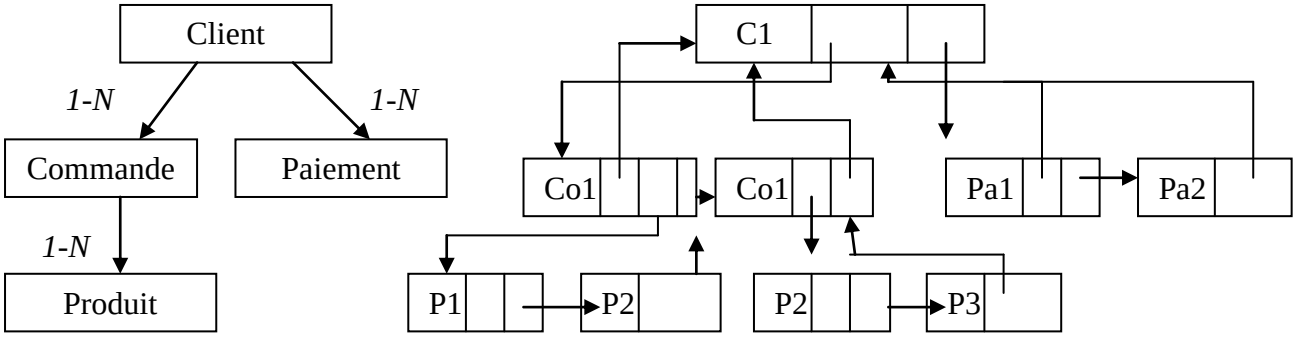
Les enregistrements sont placés les uns à la suite des autres pour obtenir un fichier séquentiel.

- Structure de liste

Elle utilise des pointeurs. Nous distinguons trois types de pointeurs :

- Pointeur Père : pour chaque nœud, un pointeur pointe vers le père.
- Pointeur fils : pour le père, un pointeur pointe vers le premier fils gauche.
- Pointeur frère : tous les nœuds fils d'un même niveau sont chaînés entre eux.

Exemple :



LE LANGAGE DE MANIPULATION DE DONNEES DL/1

Les LMD des modèles hiérarchiques sont des langages de type procédural : ils donnent au programmeur la possibilité de se déplacer dans une arborescence à partir d'une position courante. Le tableau suivant est une synthèse des principales instructions du langage :

Exemple d'utilisation :

Soit la requête suivante : trouver toutes les commandes du client de numéro « C1 »

```
GU client with  NUM = 'C1'
Etiq1 : GN commande FOR THIS client commande FOUND ? IF NOT EXIT
PRINT NUMCOMMANDE
GOTO etiq1
EXIT
```

Opérations DL/1	Signification	Description de l'opération
GU	GET UNIQUE	Accès direct à un segment utilisé pour l'accès à la racine à partir de la clé
GN	GET NEXT	Déplacement dans l'arborescence à partir d'un nœud pour atteindre le nœud successeur (pré-ordre)
GNP	GET NEXT WITHIN PARENT	Déplacement dans des nœuds qui ont le même père
GHU **GHN** **GHNP**	GET HOLD UNIQUE GET HOLD NEXT GET HOLD NEXT WITHIN PARENT	Avec verrouillage du segment accédé pour les suppressions et modifications

REMARQUES :

- L'ajout d'un segment ne peut s'effectuer que si les segments supérieurs dont il dépend existent dans l'arbre.

Exemple : l'ajout d'un produit nécessite l'existence d'un client ayant commandé ce produit.

- La suppression d'un segment entraîne la suppression de tous les segments qui lui sont rattachés.

Exemple : la suppression d'un client entraîne la suppression de toutes les commandes et des produits.

- La modification d'un segment entraîne la recherche de tous les segments qui contiennent l'information modifiée.

Exemple : la modification du prix d'un produit nécessite la recherche de tous les clients ayant commandé ce produit.

Conclusion :

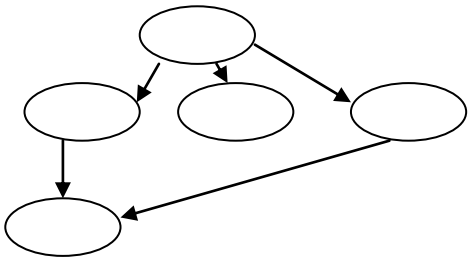
- **La redondance des données dans la base est inévitable du fait de la représentation par des structures arborescentes.**
- **Seuls les liens de type 1-1 sont pris en charge dans le modèle hiérarchique**

4.2 LE MODELE DE DONNEES RESEAUX

4.2.1 LA STRUCTURE DE DONNEES

Un réseau est un ensemble de nœuds et d'arcs. Pour une représentation de données, les nœuds représentent les objets et les arcs, les associations entre ces objets.

Exemple :



- Description des objets :

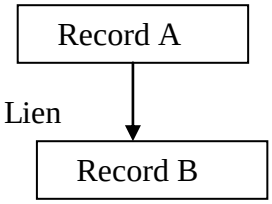
Les concepts définissant les objets sont les suivants : l'atome, l'agrégat, l'article ou enregistrement logique.

Notion1 : un atome (data item) est la plus petite unité de données possédant un nom. Un atome est représenté par une valeur dans la base

Notion 2 : un agrégat (data aggregate) est une collection d'atomes rangés consécutivement dans la base et portant un nom.

Notion 3 : Un article ou enregistrement logique (record) est une collection d'agrégats et d'atomes rangés côte à côte dans la base de données et constituant l'unité d'échanges entre la BD et les applications.

- Description des associations



Nous distinguons trois types de liens dans le modèle réseau :

1-1 : à un objet de A correspond un et un seul objet de B et vice versa

1-N : à un objet de A peut correspondre un à plusieurs objets de B mais inversement , à un objet de B ne peut correspondre qu'un seul objet de A (lien hiérarchique).

M-N : à un objet de A peut correspondre un à plusieurs objets de B et inversement.

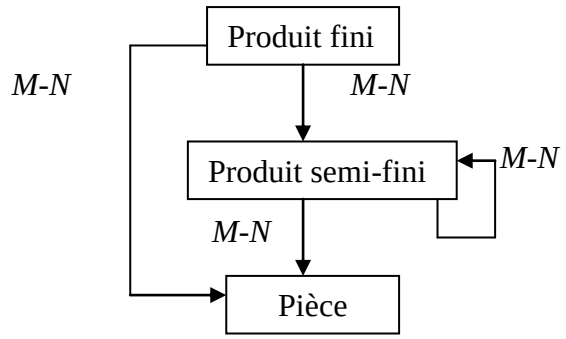
4.2.2 LE MODELE RESEAU CODASYL

En 1969 a eu lieu la conférence appelée CODASYL qui a normalisé et présenté des concepts communs pour la définition de schémas réseaux. Tous les types d'objets représentés par des nœuds sont identifiés par des clés. Une clé est un atome ou un agrégat qui permet d'identifier de manière unique chaque record.

Exemple : le matricule d'un étudiant, le numéro d'un fournisseur.

Les types de liens supportés par la norme CODASYL sont les liens 1-1 et 1-N. Les liens M-N nécessitent des transformations que nous verrons plus loin.

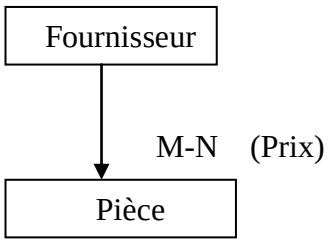
Exemples :



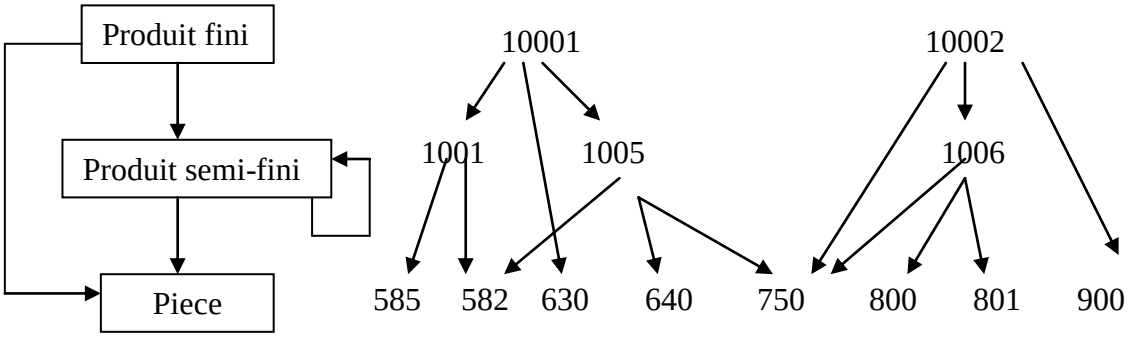
Tous les liens expriment une composition

- Données d'intersection : dans certains cas, le lien possède des propriétés qui relient les objets en relation. On les appelle des données d'intersection.

Exemple1 : le lien de composition entre produit fini, produit semi - fini et pièce possède la propriété de quantité.
Exemple 2



- Instances ou occurrences de schémas

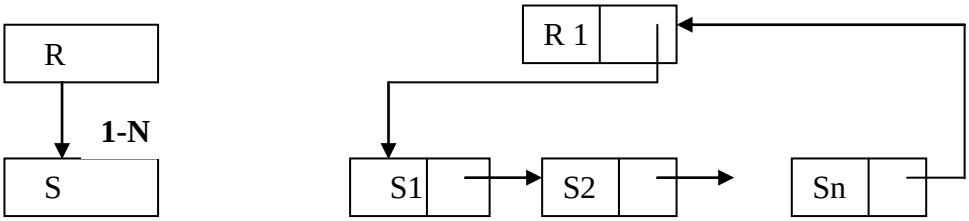


- Le lien CODASYL est appelé SET

Un SET est un lien entre un enregistrement propriétaire appelé OWNER et un ou plusieurs enregistrements membres appelé MEMBER.
Un enregistrement ne peut être à la fois propriétaire et membre d'un même SET (pas de boucle). Cependant un enregistrement peut être propriétaire de plusieurs SET différents et ou membre de plusieurs SET différents.

4.2.3 IMPLEMENTATION PHYSIQUE DES LIENS

Le lien CODASYL est implémenté grâce à la liste circulaire ou anneau.

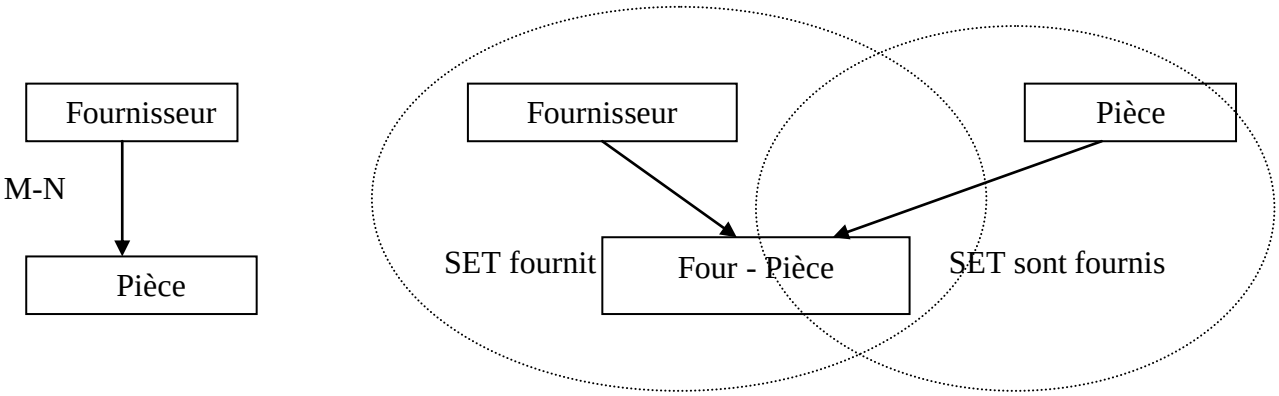


Exemple :

- Transformation du lien M-N

Le lien M-N ne peut être représenté directement par CODASYL, étant donné qu'il ne représente que les liens de type hiérarchique (owner-member).
Un artifice est utilisé qui consiste à créer un enregistrement intermédiaire avec deux liens 1-N.

Exemple :



Le record intermédiaire est crée en utilisant les données d'intersection si elles existent, et ayant pour clé la concaténation des clés des records composant le lien.

4.2.4 LE LANGAGE DE MANIPULATION CODASYL

Le langage de manipulation de données CODASYL est basé sur un ensemble de primitives synthétisé par le tableau ci-après. Cependant, il est bon de savoir que le système gère une table standard contenant :

- un pointeur par ensemble de données (RECORD)
- deux pointeurs par anneau (SET)
  - un pointeur de tête d'anneau
  - un chaînon sur l'anneau
- un pointeur par espace physique contenant la base
- un pointeur programme désignant le dernier enregistrement accédé

Primitives	Descriptions
FIND	Localise un enregistrement et provoque l'initialisation des pointeurs courants
GET	Restitue dans la zone d'E/S l'enregistrement désigné par le pointeur courant
CONNECT	Insère l'enregistrement désigné par le pointeur courant dans un anneau
DISCONNECT	Enlève l'enregistrement désigné par le pointeur courant
RECONNECT	Transfère un record dans un autre anneau
ERASE	Supprime l'enregistrement désigné par le pointeur courant
STORE	Provoque l'écriture de l'enregistrement placé dans la zone d'E/S et initialise le pointeur courant qui désignera cet enregistrement
MODIFY	Provoque la mise-à-jour à partir de la zone d'E/S de l'enregistrement désigné par le pointeur courant

Exemples :

1. Recherche des informations sur le client « CL1 »

MOVE 'CL1' TO numcli IN client
FIND ANY client USING numcli IN client
GET client

2. Modifier l'adresse du client « CL1 »

MOVE 'CL1' TO numcli IN client
FIND ANY client USING numcli IN CLIENT
GET client
MOVE 'nouvelleadresse' TO adresse IN client
MODIFY client

4.3 LE MODELE ENTITE/ASSOCIATION – CHEN (1976)

Il est communément admis que la description de l'aspect statique de la réalité organisationnelle passe par la description de ses entités, de leurs propriétés, des liens entre les entités et des contraintes auxquelles elles sont soumises. Chen est parti de cet aspect descriptif pour proposer un modèle proche de la réalité.

LE MODELE : LES CONCEPTS

Il est basé sur les concepts d'ENTITE TYPE/ ASSOCIATION TYPE et sur deux formes d'abstraction : la classification et l'instanciation.

Entité : elle représente un objet concret ou abstrait de la réalité

Exemple :

Le client « Attaf Mohamed », Le compte numéro 120005, le séminaire de BD etc.

Association : une association d'entités est une combinaison d'entités dans laquelle chacune joue un rôle spécifique

Exemple :

'Attaf' est affecté au service comptabilité

'Attaf' est titulaire du compte numéro 120005

Entité type : ensemble d'entités ayant des caractéristiques communes appelées propriétés

On parle aussi de classe entité et on note :

$E : \{e_1, e_2, …., e_n\}$

Une entité e_i est une instance ou occurrence du type (ou de la classe E)

Exemple :

Tous les clients sont caractérisés par leur nom et leur adresse. Nous définirons une classe CLIENT ayant les attributs NOM et ADRESSE.

Association type : ensemble ou classe d'associations similaires. Une association type est un sous-ensemble du produit cartésien des entités type impliquées.

On note :

$R(E_i, E_j, …, E_p)$

$R : P (E_i \times E_j \times … \times E_p)$

Exemples :

EMPLOYE ***est affecté à*** *PROJET :*

R(EMPLOYE,PROJET)

EMPLOYE ***du*** *DEPARTEMENT* ***est affecté à*** *PROJET :*

R(EMPLOYE, DEPARTEMENT, PROJET)

Une entité ou une association sont caractérisés par le doublet <**attribut, valeur**> : un attribut est une fonction d'une entité type ou d'une association type sur un domaine ou un produit cartésien de domaines.

Exemples :
E:{e_1} : (Nom, 'Attaf',), (Date naissance, '251268'), (Salaire, '40KD')

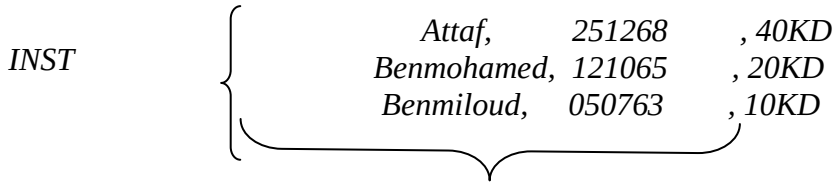
R (E,P) : (DateDeb, '010196'), (DateFin, '010199')

Nom appartient au domaine Char(30)
Date naissance appartient au domaine Int(6)

SCHEMA D'ENTITE TYPE

Il définit l'entité type en intention. C'est le nom de l'entité type suivie des noms d'attributs.

Exemple : EMPLOYE (NOM, DATENAISSANCE, SALAIRE)

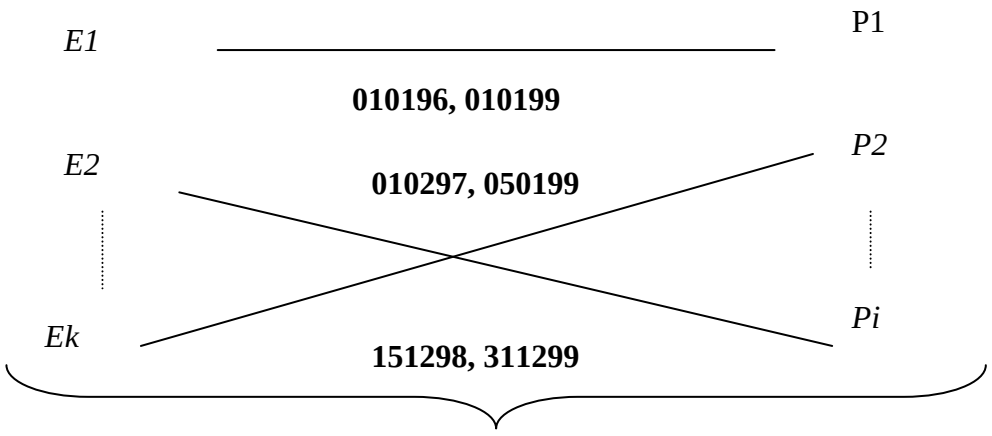


EXTENSION DE LA CLASSE

SCHEMA D'ASSOCIATION TYPE

Il définit l'association type en intention. C'est le nom de l'association type suivie des noms des entités type la composant avec éventuellement les noms d'attributs.

Exemple :
AFFECTATION (EMPLOYE, PROJET : DATEDEB, DATFIN)

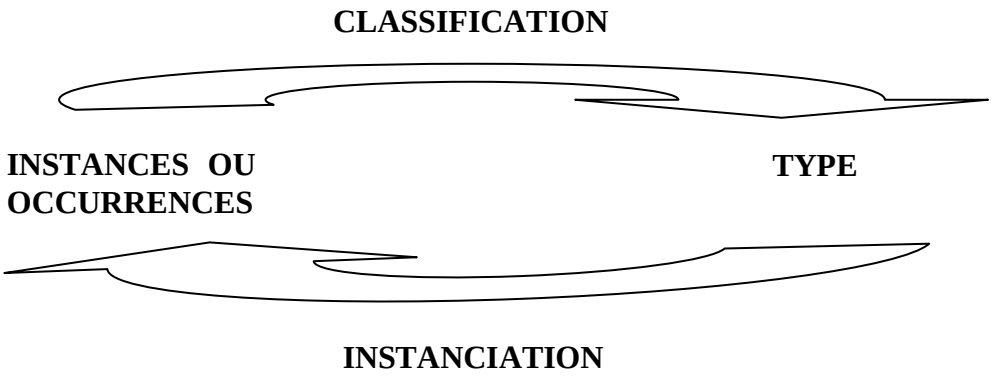


INSTANCES

LES FORMES D'ABSTRACTION

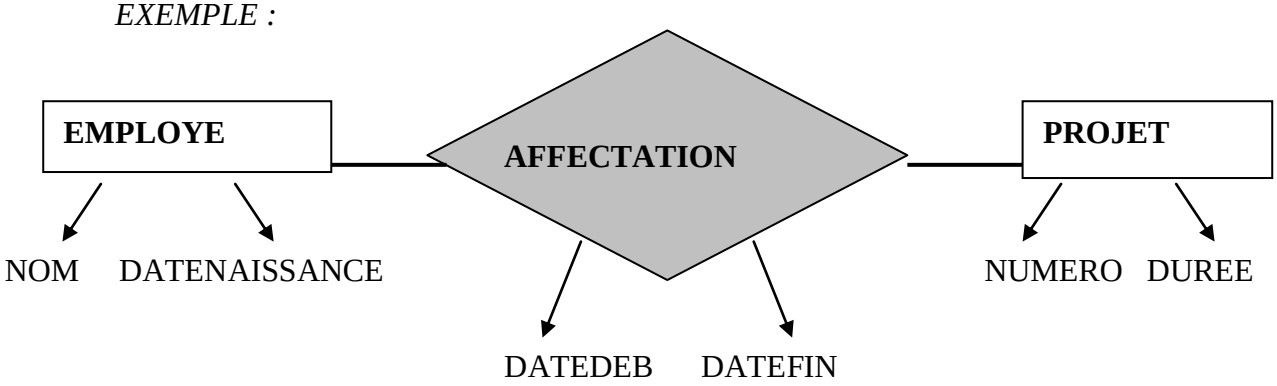
- **Classification :** partir des informations de détail et construire la classe.

- **Instanciation :** partir de la classe et définir ces instances.



- **REPRESENTATION GRAPHIQUE :** une entité type est représentée par un rectangle alors qu'à l'association type est associé le losange.

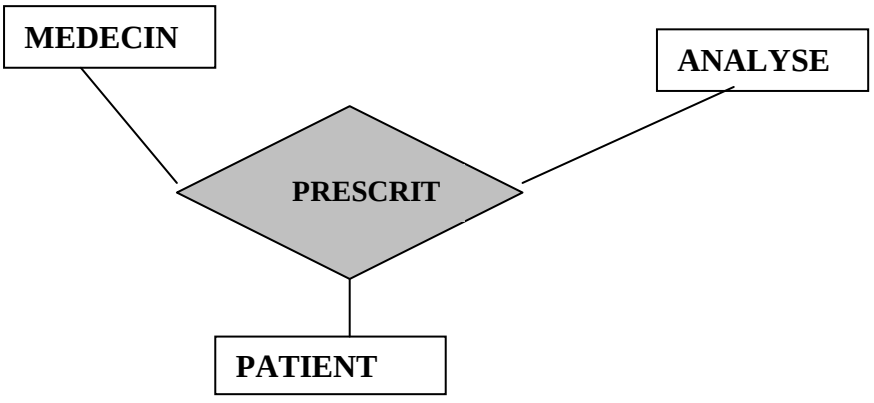
EXEMPLE :



Remarque : Les attributs sont simples et monovalués

- **LES ASSOCIATIONS N-AIRES**

EXEMPLE :

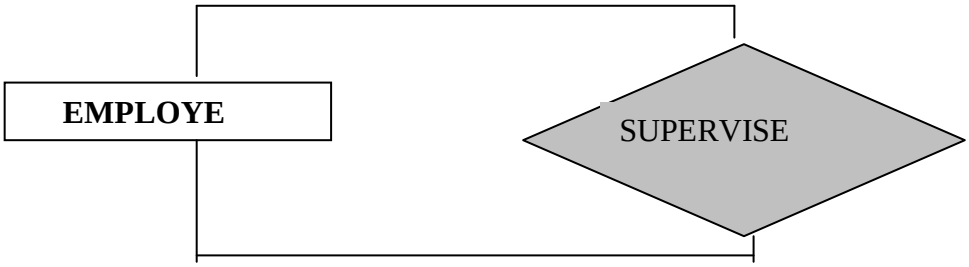


- **ASSOCIATIONS RECURSIVES**

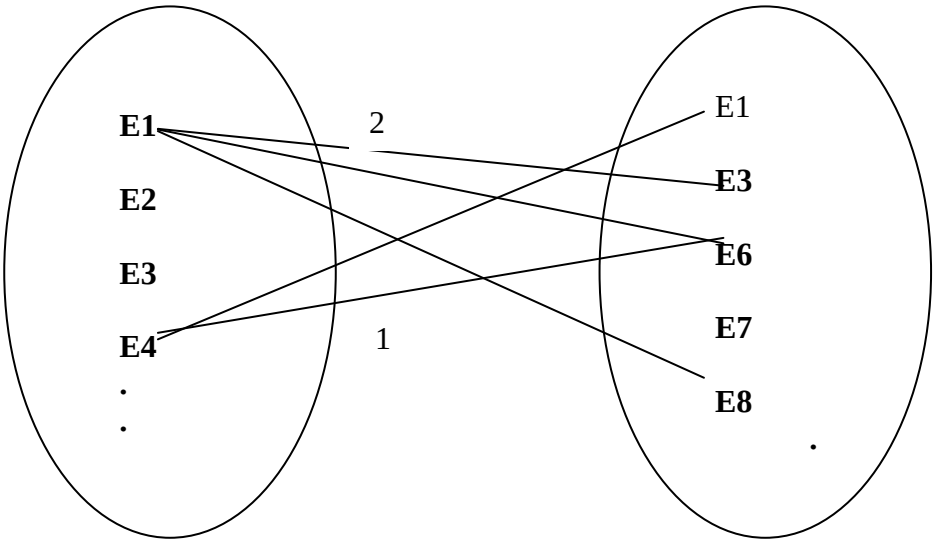
Une entité peut jouer plusieurs rôles dans une association type.

EXEMPLE :

ROLE 1 : EST SUPERVISE



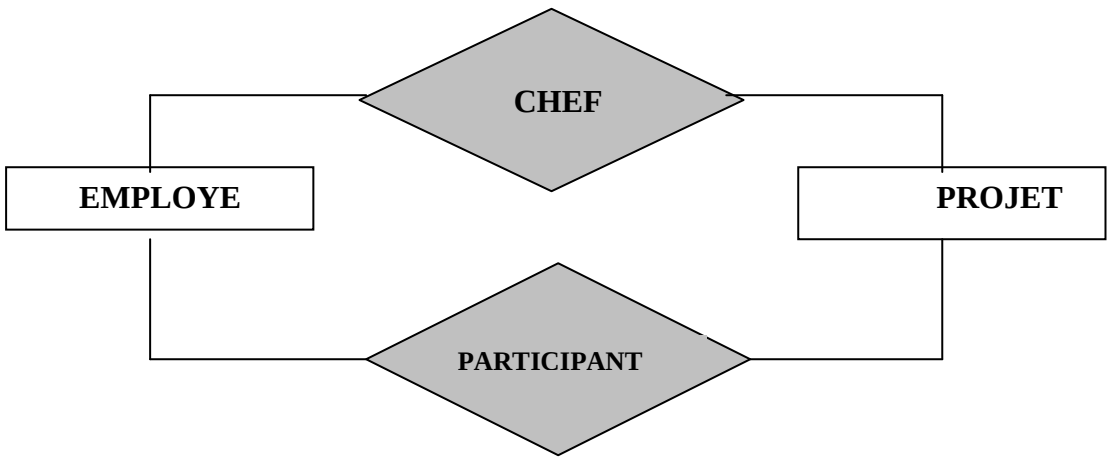
ROLE 2 : SUPERVISEUR

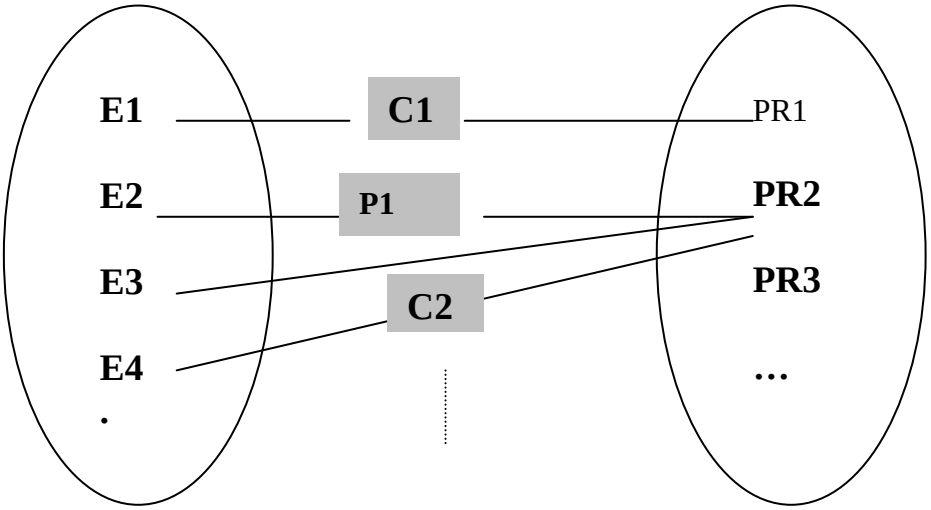


- **ROLES**

Une entité peut jouer plusieurs rôles dans ses relations avec une autre entité type.

EXEMPLE :





CONTRAINTES IMPOSEES A LA MODELISATION

- **CLE**

Une clé est un attribut ou un groupe d'attributs dont les valeurs permettent d'identifier de manière unique les entités d'un même type.

La spécification de la clé d'une entité type implique une contrainte d'unicité sur l'extension de l'entité type.

EXEMPLES :

EMPLOYE : NUMEMP
PROJET : NUMPROJ

La clé d'une association type est la combinaison des clés des entités types participantes.

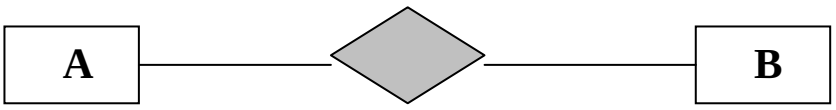
EXEMPLE :

AFFECTATION : NUMEMP, NUMPROJ

- **CARDINALITES**

La cardinalité est le nombre d'associations auxquelles une entité peut participer.

Il existe trois cardinalités prédéfinies :



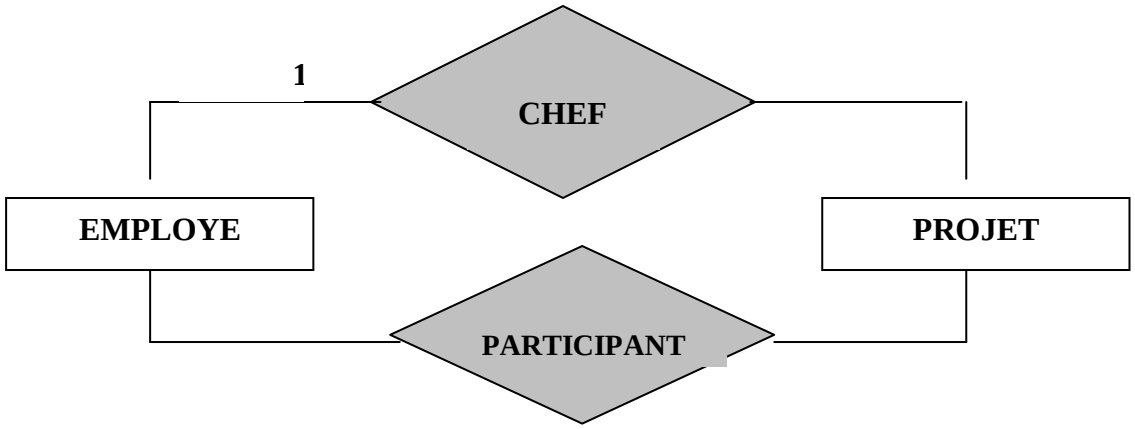
1-1 : à une instance de A est associée une et une seule instance de B.

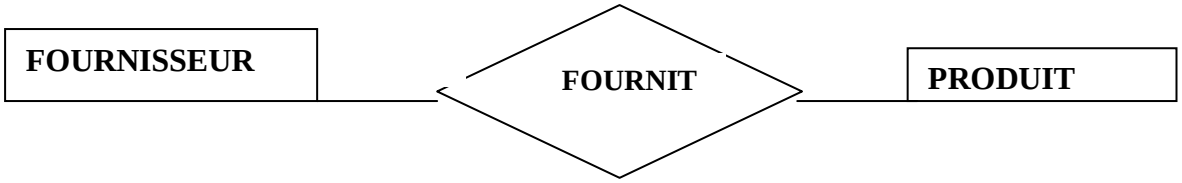
1-N : à une instance de A est associée une ou plusieurs instances de B.

M-N : à une instance de A est associée une ou plusieurs instances de B et réciproquement.

EXEMPLES :


EMPLOYE
AFFECTATION
PROJET


1
CHEF
EMPLOYE
PROJET
PARTICIPANT


FOURNISSEUR
FOURNIT
PRODUIT

4.4 LES BASES DE DONNEES RELATIONNELLES

4.4.1 LE MODELE RELATIONNEL (E.CODD 1970)

- Le modèle relationnel est fondé sur le concept mathématique de relation et inventé par CODD à IBM San José en 1970.
- Il lui est associé la théorie de la normalisation dont l'objectif est l'élimination de certains comportements anormaux des relations lors de mise-à-jour.

4.4.2 LES CONCEPTS DU MODELE

- **Notion mathématique de relation**

Soient D1, D2, …,Dn des ensembles de valeurs (domaines) :
Une relation R (D1,D2,…,Dn) est un sous ensemble du produit cartésien des domaines D1, D2, …,Dn.

R(D1, D2, …, Dn) inclus dans **D1 X D2 X…X Dn**

Un tuple ou n-uplet d'une relation est :

<d1, d2,…,dn> tel que d1 appartient à D1
d2 « « D2
dn « « Dn

- **Transposition à la réalité conceptuelle**

Les ensembles Di sont les domaines de valeurs des propriétés du monde réel.

EXEMPLES :
Char, Char(30), Int(6), {F,M},…

- Notion d'attribut

Un attribut est une colonne de relation caractérisée par un nom.

Les attributs d'une relation doivent être différents. Plusieurs attributs peuvent être définis sur le même domaine.

- Notion de clé
Une clé est un attribut ou groupe d'attributs qui identifie de manière unique un tuple de relation.

- Schéma de relation

Le schéma de relation est le nom de la relation suivie de la liste de ses attributs.

- Schéma relationnel

C'est l'ensemble des schémas de relations qui définit une application.

- Base de données relationnelle

C'est une BD dont le schéma est un ensemble de schémas de relations (schéma relationnel), et dont les occurrences (instances) sont les tuples de ces relations.

EXEMPLES :

INTENTIONS DE RELATIONS

EMPLOYE (NUMEMP, NOME, DATNAIS, SALAIRE)
PROJET (NUMPROJ, DUREE)
DEPARTEMENT (NUMDEP, NOM, NBREMPL)
CLIENT (NUMCLI, NOMC, ADRESSEC, BALANCE)
FOURNISSEUR (NUMFOUR,NOMF, ADRESSEF)
PRODUITFOURNI (NUMPROD, NUMFOUR, PRIX)

Soit la relation EMPLOYE (NUMEMP, NOME, DATNAIS, SALAIRE)

EXTENTION DE LA RELATION

10101 Attaf, 251268 , 40KD
10102 Benmohamed, 121065 , 20KD
10103 Benmiloud, 050763 , 10KD

Représentation sous forme de tableau

EMPLOYE

NUMEMP	***NOME***	***DATNAIS***	***SALAIRE***
10101	*Attaf*	*251268*	*40KD*
10102	*Benmohamed*	*121065*	*20KD*
10103	*Benmiloud*	*050763*	*10KD*