



Web : www.fao.org/fi/copemed

Proyecto FAO COPEMED

Universidad de Alicante
Ramón y Cajal, 4
03001 - Alicante, España

Tel : +34 96 514 59 79

Fax : +34 96 514 59 78



GCP/REM/057/SPA
Email : copemed@ua.es

Formation à l'utilisation des Systèmes de Gestion de Bases de Données Relationnelles

organisée avec la collaboration du

Centre Royal de Télédétection Spatiale (*)

SYSTÈME D'INFORMATION ET BASES DE DONNÉES

Support de Cours (**)

Rabat (Maroc), 28 février – 3 mars 2000

(*) CRTS 16 bis, Avenue de France Agdal, Rabat Maroc Tel. : +212 (7) 776305 Fax : +212 (7) 776300 E-mail : layachi@crts.gov.ma

(**) Société AT TIME 28, Rue Michlifen, Appt. 2 Agdal – Rabat (Maroc) Tel./Fax : +212 (7) 672030 E-mail : atime@mtds.com

SYSTEME D'INFORMATION ET BASES DE DONNEES

- **Système d'Information (SI) de l'entreprise :**

Ensemble des moyens et procédures de recherche, de saisie, de classement, de mémorisation, de traitement et de diffusion de l'information du système et de son environnement.

- **Rôle d'un SI :**

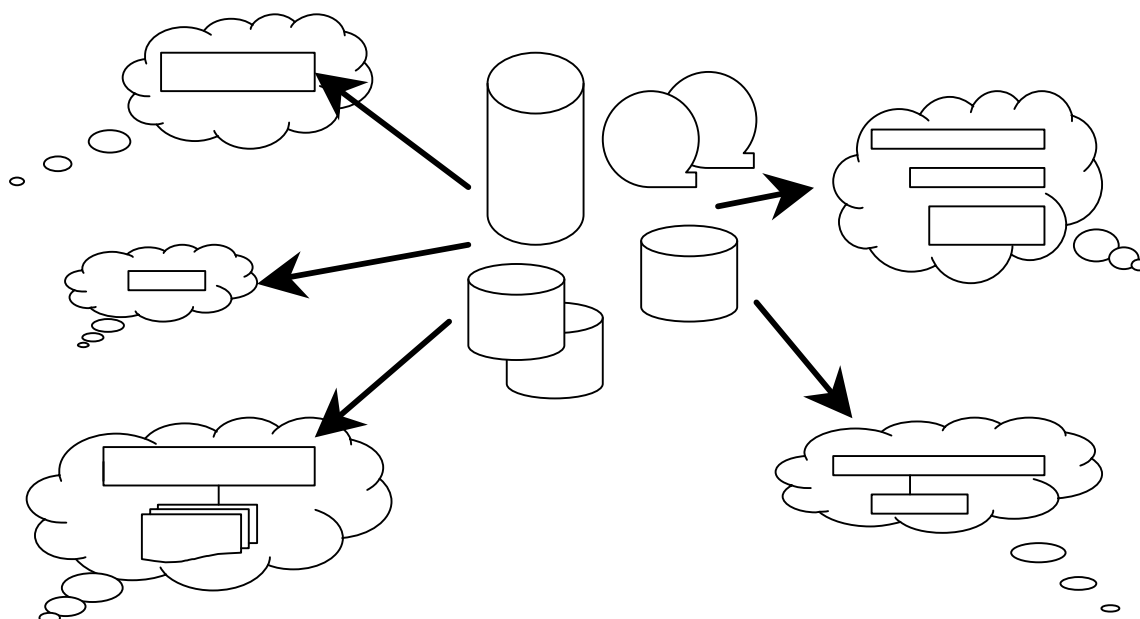
- Assister à la décision.
- Faciliter la coordination des acteurs du système.

- **Objectif de l'automatisation des SI :**

Restituer, aux différents membres de l'entreprise, les informations sous une forme directement utilisable ; au moment opportun, afin de faciliter le bon fonctionnement opérationnel et la prise de décision aux différents niveaux.

- **Environnement basé sur un système de fichiers :**

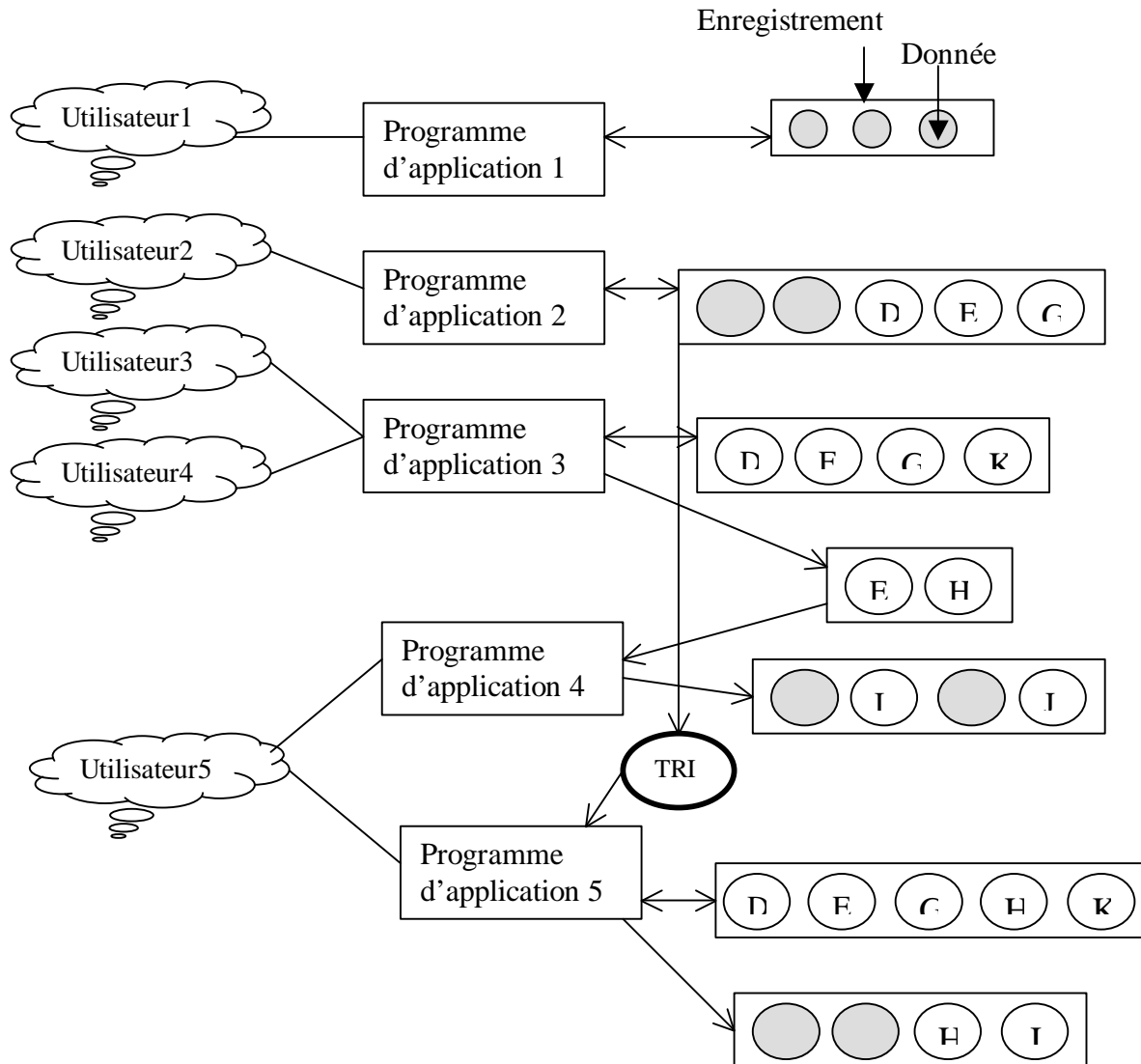
- Plusieurs fichiers.
- Plusieurs programmes créés en différentes dates, avec éventuellement différents langages.



- **Inconvénients de cet environnement :**

- Redondance et inconsistance :

Même information répliquée dans plusieurs fichiers.



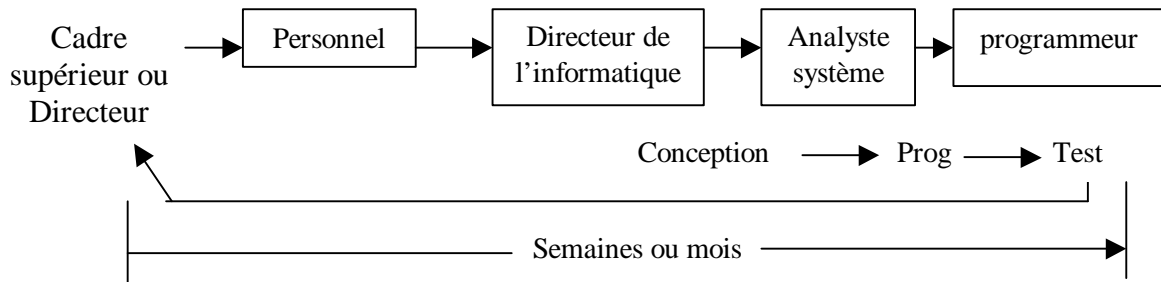
Pour chaque nouvelle application, un programmeur crée de nouveaux fichiers. Le système d'information se retrouve vite composé de nombreux fichiers avec une grande redondance des données.

Ce qui implique :

- Accroissement du volume global des données
- Risque d'inconsistance des données : informations sur plusieurs fichiers, non "rafraichies" au même moment.

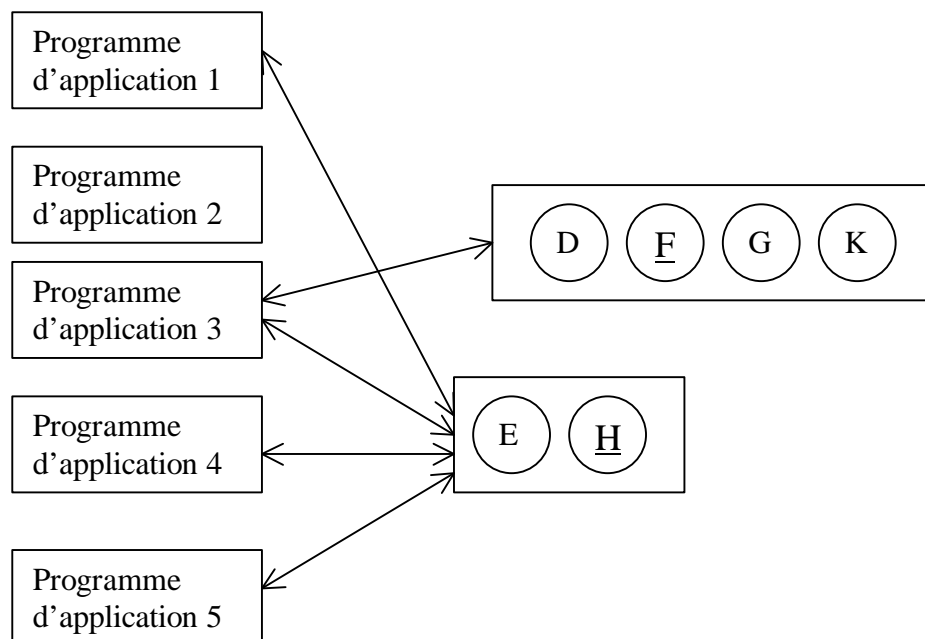
➤ Difficultés d'accès aux données :

Les besoins occasionnels ne peuvent être satisfaits de façon efficace, ce qui nuit donc à l'efficacité du système, qui fait intervenir de nombreux acteurs :



➤ Difficultés dans la maintenance :

Un fichier finit par être utilisé par des applications multiples. Une restructuration de ce fichier provoquera une réaction en chaîne de modifications dans d'autres programmes d'applications.



➤ Difficulté de partage de données :

Supposons un environnement multi-utilisateurs,

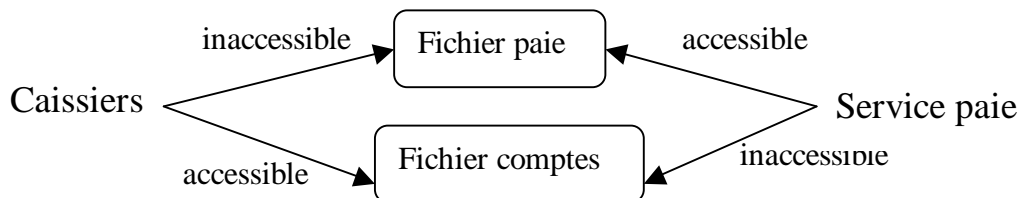
Compte = 500

Utilisateur 1	Tâche	Utilisateur 2
Lire compte	T1	-
-	T2	lire compte
compte := compte -50	T3	compte:=compte -50
écrire compte	T4	-
-	T5	écrire compte

Au moment de l'écriture du compte, on ne sait pas la valeur qui va être sauvegardée.

➤ Problèmes de sécurité :

Les données stockées sur fichiers doivent être protégées contre les accès non autorisés



➤ Problèmes d'intégrité :

Les données doivent satisfaire certaines contraintes.

Exemples :

- Soldes > =25
- Un client référencé dans fichier des commandes, doit figurer d'abord dans le fichier des clients.

Nécessité donc de modules spécifiques chargés de respecter les contraintes au sein des diverses applications.

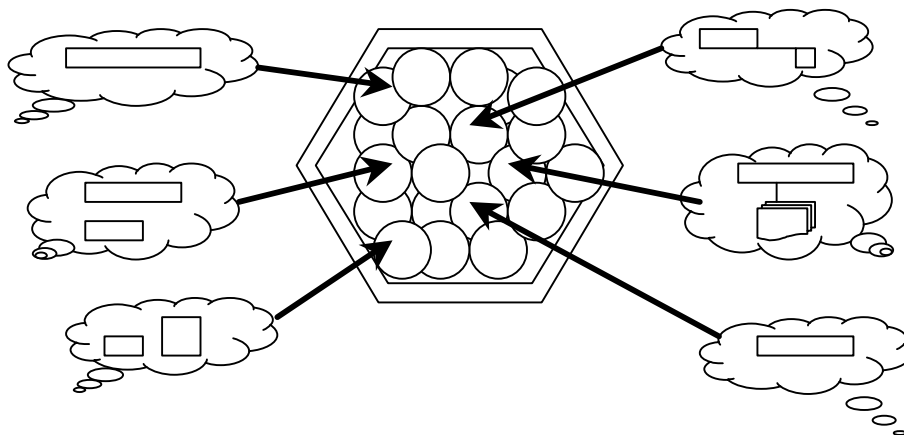
- **Environnement basé sur un système de Bases de Données relationnelles**

Base de données :

Une base de données est une collection de données reliées entre elles, stockées ensemble, aussi peu redondantes que possible pour être utilisées par une ou plusieurs applications d'une façon optimale.

Les données sont stockées de façon à être indépendantes des programmes qui les utilisent.

JAMES MARTIN



Exemple : Une base de données BANQUE : un ensemble de tables (ou relations)

La table Dépôt

<u>AGENCE</u>	COMPTE	CLIENT	POSITION
B. MED. V	020 500	Haji	12000
ZERKTOUNI	110 230	Jalal	14000
IBN SINA	120 560	Radi	20500
MASSIRA	320 001	Jabri	3000

La table Crédit

<u>AGENCE</u>	PRET	CLIENT	MONTANT
B. MED. V	10	Haji	5000
ZERKTOUNI	11	Sami	6500
IBN SINA	12	Cadi	7000
B. MED. V	13	Radi	4500

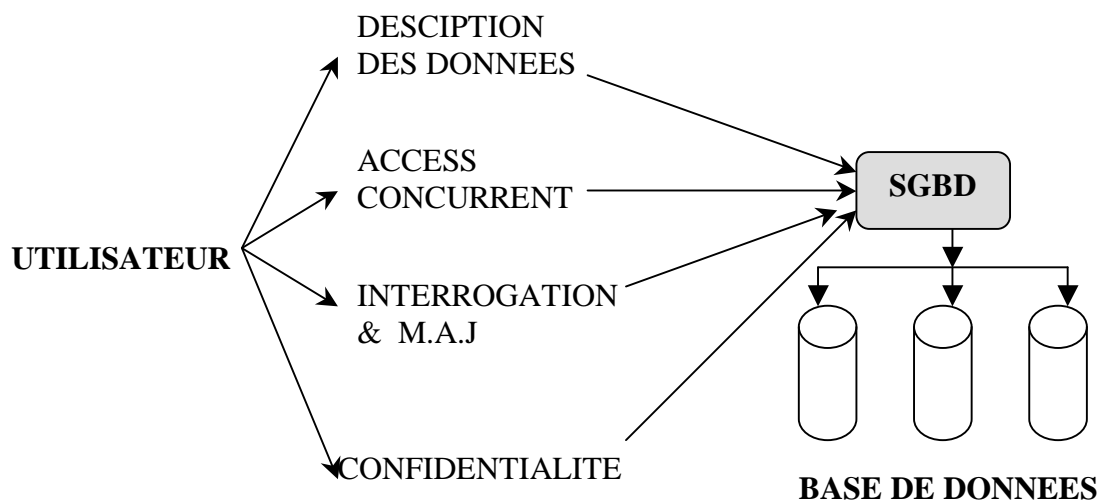
La table Clientèle

<u>CLIENT</u>	ADRESSE	VILLE
Haji	320, Rue Toubkal, Agdal	Rabat
Radi	11, Rue Sanaa, Nasser	Rabat
Sami	85, rue Zerktouni, Appt 12	Casa
Jalal	10, cite Medina, 1 ^{er} etage	Casa
Cadi	55, Rue de Tunis	Casa
Jabri	32, Rue Al jazira	Kenitra

Remarque : On suppose que client est un identifiant de la table clientèle

Système de gestion de base de données :

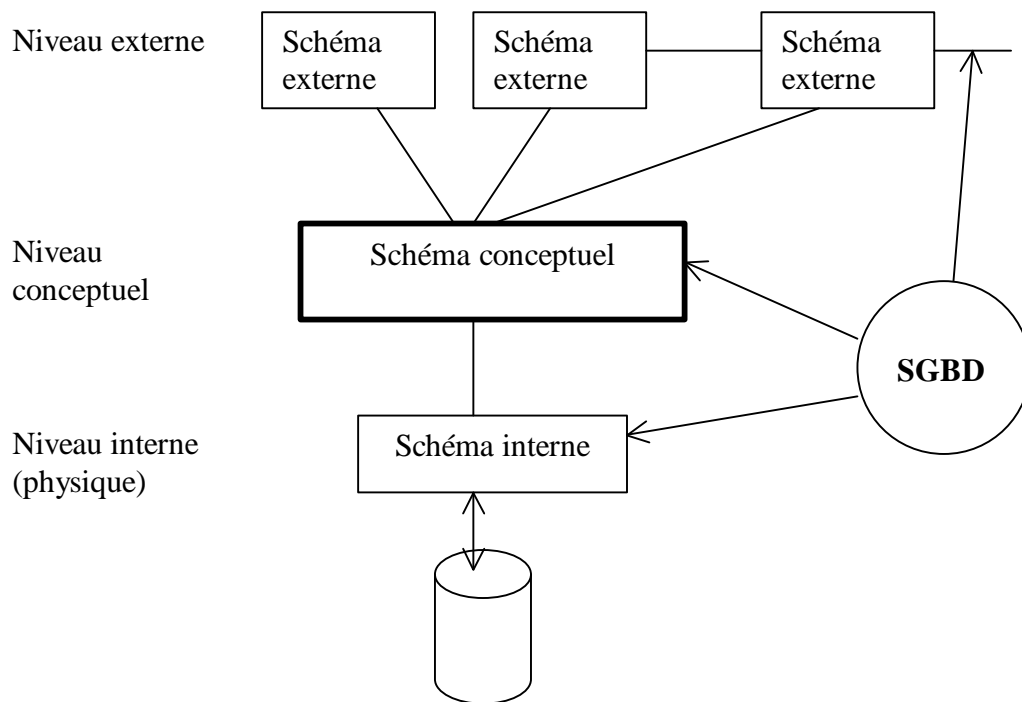
Logiciel qui gère un ensemble de fichiers qui constituent la B.D et qui permet à plusieurs utilisateurs de stocker et d'extraire des données de ces fichiers, dans des conditions d'intégrité et de confidentialité.



Architecture d'une base de données :

Le SGBD présente la B.D sous trois niveaux (niveaux de description de données : architecture ANSI/ SPARC) :

- Niveau interne (physique)
- Niveau conceptuel (logique)
- Niveau externe



OBJECTIF DES SGBD :

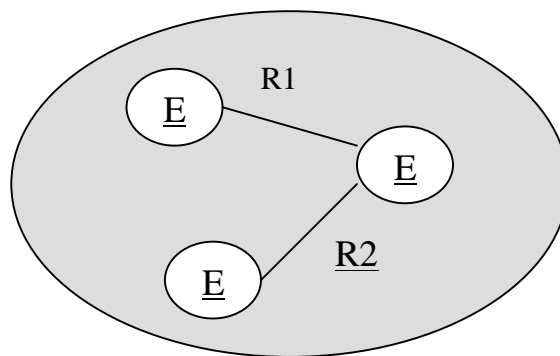
Simplifier et Faciliter l'exploitation des données :

- Indépendance programmes/données
- Faciliter l'accès aux données
- Intégrité des données
- Sécurité des données
- Gérer le partage des données
- Récupération en cas de panne

MODELE DE DONNEES RELATIONNEL

I. LE MODELE ENTITES –RELATIONS (MCD : Modèle Conceptuel de Données)

Modèle développé pour faciliter la conception de la structure logique d'une B.D.



Monde réel

Il repose sur la perception du monde réel sous forme d'un ensemble d'objets "entités" associés au moyen d'un ensemble de "relations" entre ces objets.

1. Entité :

Ensemble d'objets de même type

Défini par un ensemble d'attributs,

À chaque attribut est associé un "domaine" : ensemble de valeurs autorisés.

Exemple :

Clients d'une banque : Nom, adresse, ville

Une entité peut être considérée comme un type qui regroupe des objets et qui a une existence suffisamment autonome pour intervenir dans des rencontres avec d'autres objets.

Une B.D contient une collection d'entités.

Exemple : Agences = (nom-agence, avoir, ville)
 Client = (nom, CIN, adresse, ville)
 Compte =(numéro, agence, position)
 Transaction =(numéroT, date, montant)
 Prêt = (NuméroP, montant)

2. Relation :

Associe (lie) plusieurs entités,

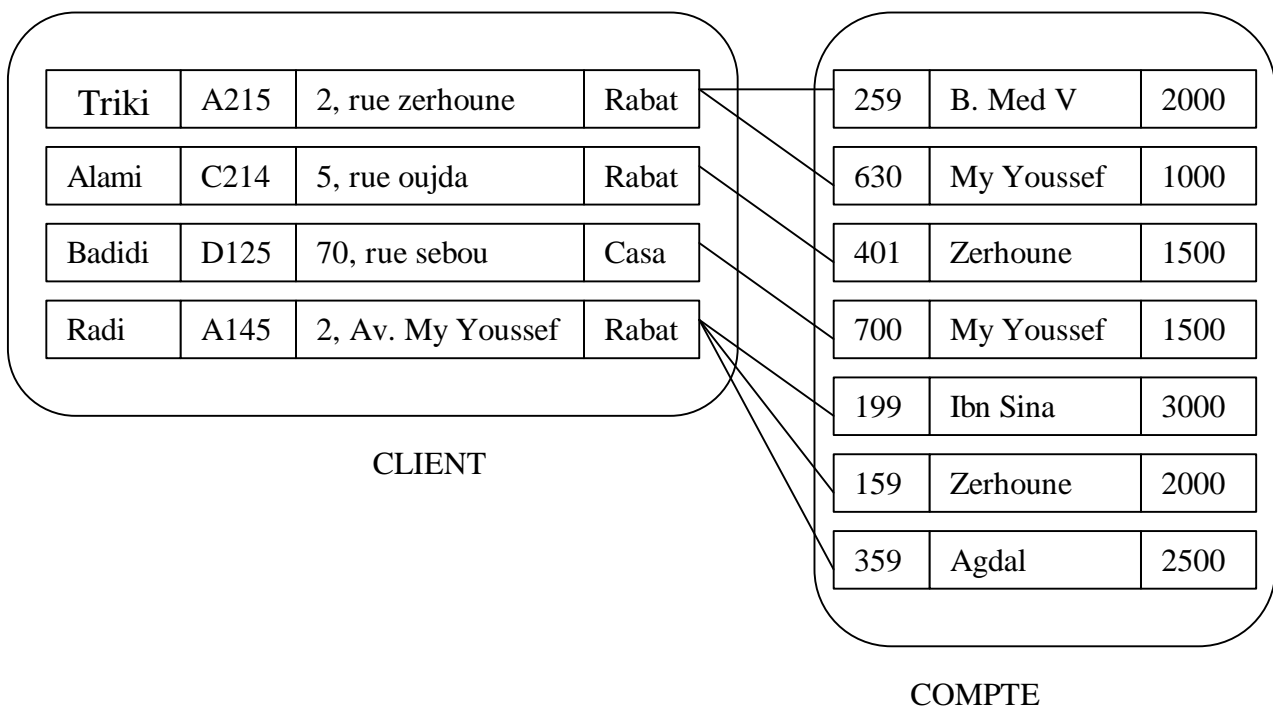
Une relation n'existe que par les entités qu'elle met en correspondance,

Une relation peut être caractérisée par des attributs

Exemple :

Un client détient un compte

Relation : Détient
 Entités liées : Client, compte
 Attribut de la relation : Date dernier mouvement



3. Dimension d'une relation :

Le nombre d'occurrences des entités concernées par une relation. Ce nombre peut être strictement supérieur au nombre d'entités intervenant dans la relation,

Exemples :

Relations binaires :

“Clicom” : un client possède un compte,

"Amortir" : un client amortit un prêt,

“Marié à” : une personne est mariée à une autre,

Relations ternaires :

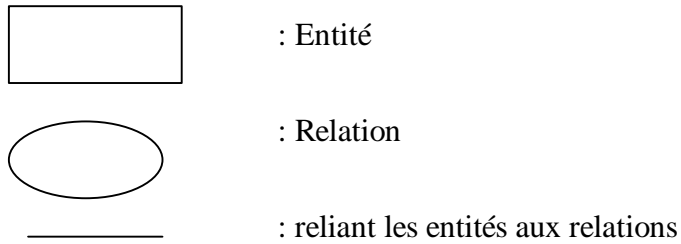
"Ouv-Cpte" : un client ouvre un compte dans une agence.

En pratique, il est rare d'avoir besoin d'une relation de dimension ≥ 4

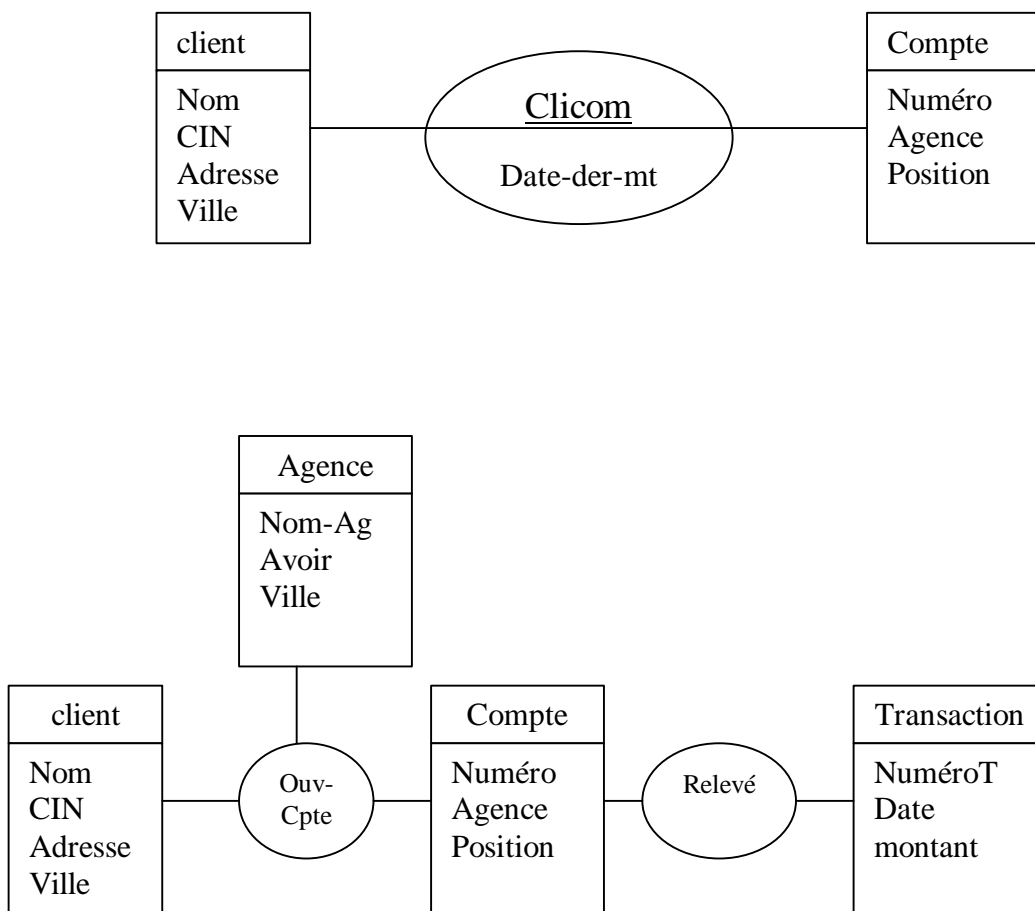
4. Diagrammes Entités – Relations :

La structure d'ensemble d'une B.D peut être représentée graphiquement par un diagramme entité-relation ou un MCD :

Symboles :

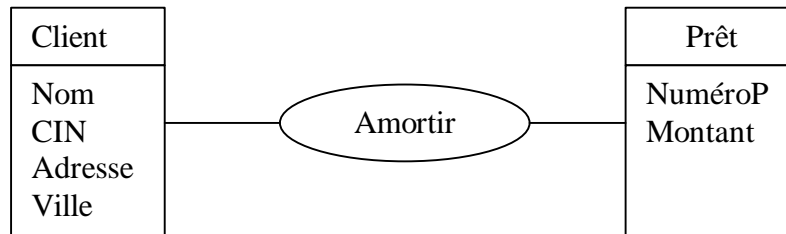


Exemples :



5. Contraintes dans le MCD :

Soit le modèle :



Un client peut-il avoir plusieurs prêts ?,

Combien au maximum ?

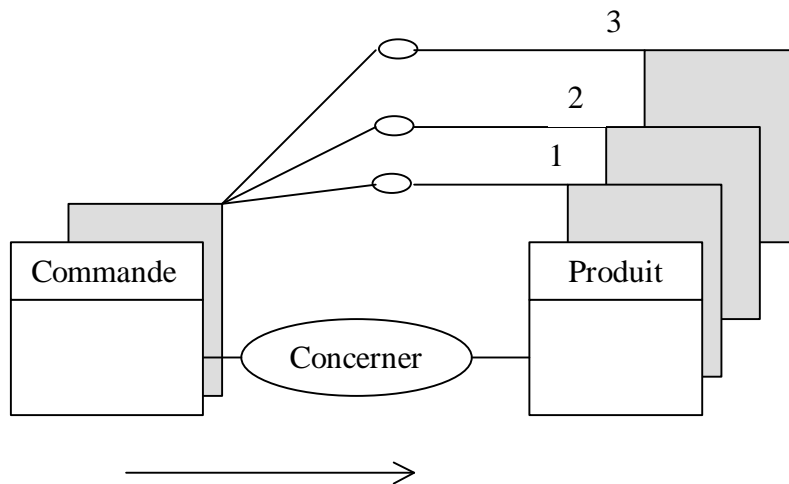
Un prêt peut-il être accordé à plusieurs clients ?

Les réponses peuvent être définies sur un modèle de données, comme contraintes :
complément d'explications ou des restrictions relatives au sens des données,

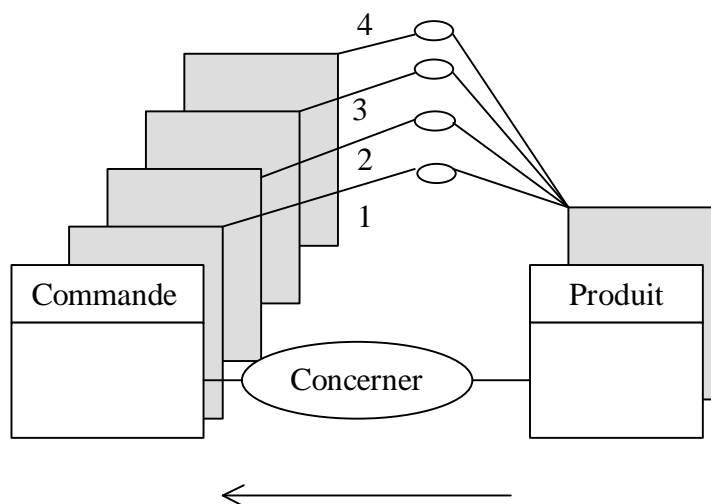
6. Cardinalité d'une relation pour une entité :

La cardinalité d'une relation pour une occurrence entité donnée est le nombre d'occurrences de relations que possède cette occurrence entité via la relation,

Exemple :



Quand je compte le nombre de produits d'une commande, je m'intéresse aux cardinalités de « commande »



Quand je compte le nombre de commandes d'un produit, je m'intéresse aux cardinalités de "produit"

- Cardinalité minimale :

Nombre minimum d'associations que possède une occurrence entité via la relation,

Exemple :

la commande peut-elle ne concerner aucun produit ?

OUI : cardinalité minimale = 0

NON : cardinalité minimale = 1

- Cardinalité maximale :

Nombre maximum d'associations que possède une occurrence entité via la relation,

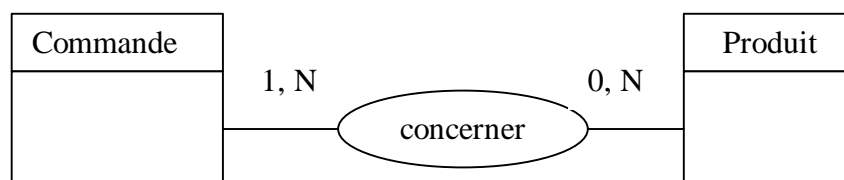
Exemple :

la commande concerne-t-elle un seul produit au maximum ?

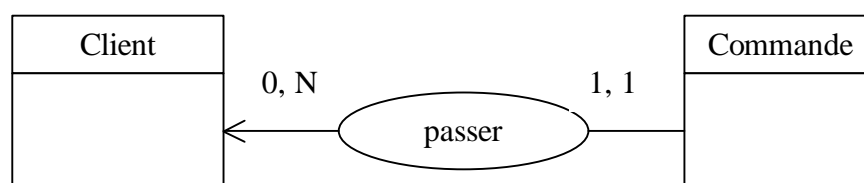
OUI : cardinalité maximale = 1

NON : cardinalité maximale = N

Les cardinalités de notre exemple :



Quand une cardinalité maximale pour une entité = 1, on représente une flèche partant de l'entité, et on parle d'une dépendance fonctionnelle (DF) : Commande dépend du client,



Chaque commande est associée à un client. Si on supprime un client, ses commandes n'ont plus besoin d'être mémorisées, mais on peut annuler les commandes et conserver le client.

7. Identifiant (clé) :

Il est important d'identifier (différencier) les occurrences entités et relations créés

Exemple :

Client = (nom, CIN, adresse, ville)

Nom : n'est pas identifiant

Nom, CIN : identifiant

CIN : identifiant minimal ou identifiant (clé) privilégié ou clé primaire (si elle est retenue comme descripteur de la classe client)

- Clé primaire :
Groupe minimum d'attributs identifiant une occurrence entité unique dans sa classe
- Entité forte : qui possède une clé primaire
- Entité faible :
Entité qui a besoin de se référer à une entité forte (entité dominante) pour identifier chacune de ses occurrences de façon unique.

Exemple :

Transaction =(numéro-transaction, date, montant)

Transaction d'un compte : numéro transaction est identifiant

Transaction de tous les comptes : numéro transaction n'est pas identifiant

Numéro transaction est uniquement un discriminant

Transaction est appelée « entité faible »

- Clé primaire d'une entité faible : s'obtient par :
Clé primaire de l'entité dominante + discriminant de l'entité faible

Exemple :

Clé primaire pour transaction : (numéro compte, numéro transaction)

- Clé primaire d'une relation :

Se compose des clés primaires des entités associées par la relation,

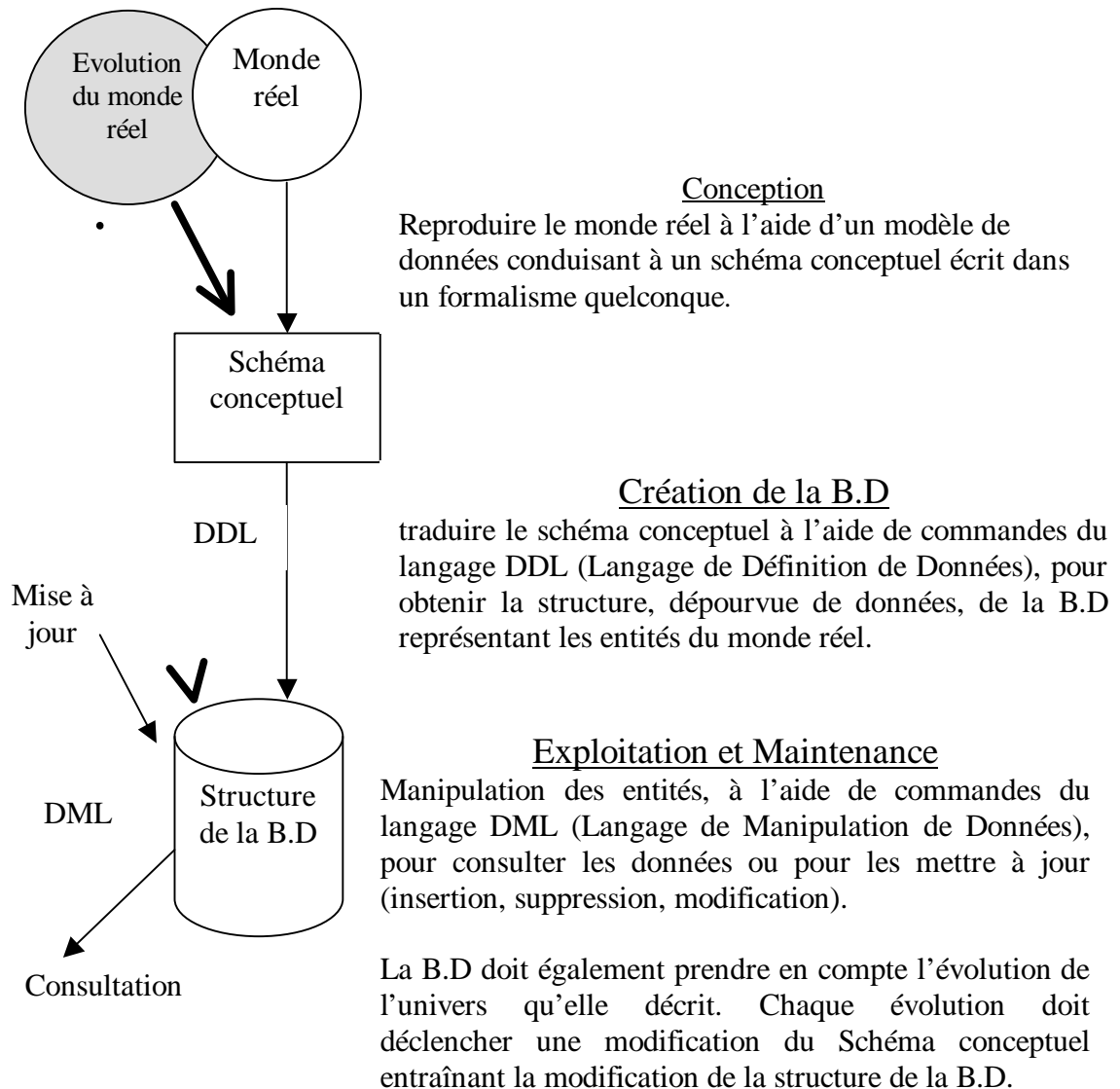
Exemple :

Client = (nom, **CIN**, adresse, ville)

Compte = (**numéro**, agence, position)

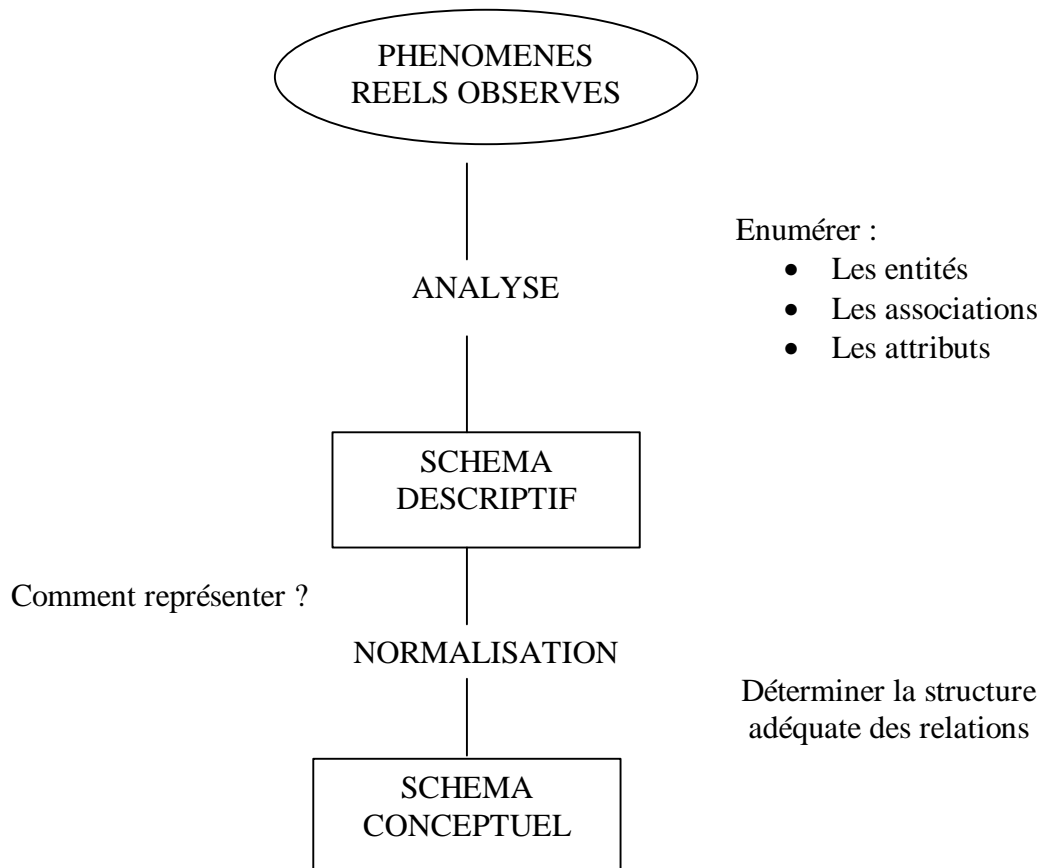
Clé primaire de Clicom : (**CIN**, **numéro**)

II. CYCLE DE VIE D'UNE B. D



Étape de conception

La conception d'une B.D relationnelle consiste à donner une perception tabulaire d'un univers réel : traduire le monde réel en terme de tables (relations) représentant ses entités ainsi que les liens pouvant exister entre elles,



Exemple : Étude des commandes

- Univers du discours :

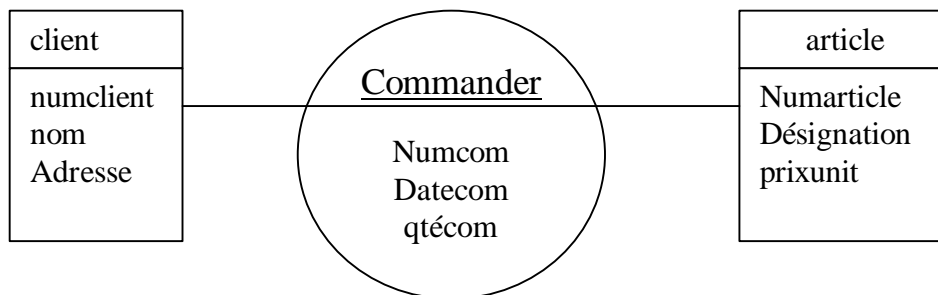
Un client commande un certain nombre d'articles qui seront livrés à son adresse. Une commande indique le numéro du client, son nom, la date de la commande et, par article commandé, son numéro, sa désignation, son prix unitaire et la quantité commandée. Comme les clients et les articles, chaque commande est identifiée par un numéro de commande,

- Analyse du réel :

Inventaire des entités, relations et attributs :

Entités	Relations (associations)	Attributs (Dictionnaire de Données)
Client Article	Commander	Numclient Nom Adresse Datecom Numarticle Désignation Prixunit Qtécom Numcom

- Synthèse de l'analyse :



- Schéma descriptif :

Client = (numclient, nom, adresse)

Article = (numarticle, désignation, prixunit)

Commande = (numclient, numarticle, numcom, datecom, qtécom)

- Normalisation :

Il faut passer ensuite par une phase de normalisation pour obtenir un schéma conceptuel bien conçu.

III. DIFFICULTE D'UNE MAUVAISE CONCEPTION

Une mauvaise conception d'une B.D peut entraîner :

- La répétition des informations
- L'inaptitude à la représentation de certaines informations
- La perte d'informations.

➤ Répétition des informations :

Exemple :

Emprunt = (agence, avoir, ville, prêt, client, montant)

Agence	Avoir	Ville	Prêt	Client	Montant
B. Med. V	900 000	CASA	17	Mr. ALAMI	50 000
MURSULTAN	500 000	CASA	53	BADIDI	20 000
ZERKTOUNI	300 000	CASA	15	TRIKI	15 000
My. YOUSSEF	350 000	FES	14	RADI	05 000
B. Med V	900 000	CASA	17	Mme. ALAMI	50 000
ZERKTOUNI	300 000	CASA	12	TAHRI	15 000

Pour ajouter un prêt : (zerktouni, 31,HAMMADI, 10 000)

Il faut insérer : (zerktouni, 300 000, CASA, 31, HAMMADI, 10 000)

La répétition des attributs : avoir, ville entraînera :

Gaspillage d'espace disque

Complication des MAJ due à la redondance :

- Multiplication des modifications en cas de changement,
- Incohérence de la B.D en cas d'oubli

➤ Inaptitude à la représentation des informations :

Crédit = (agence, prêt, client, montant), prêt est un identifiant.

Dépôt =(agence, compte, client, position), compte est un identifiant.

En combinant crédit et dépôt en une seule table C_D :

C_D =(agence, prêt montant, compte, position, client)

Dans la mesure où le prêt doit être un identifiant, C_D ne peut représenter un client ayant un compte dans une agence sans y amortir un prêt,

➤ Perte d'informations :

Dans la relation Emprunt :

Agence	Avoir	Ville	Prêt	Client	Montant
B. Med. V	900 000	CASA	17	Mr. ALAMI	50 000
MURSULTAN	500 000	CASA	53	BADIDI	20 000
ZERKTOUNI	300 000	CASA	15	TRIKI	15 000
My. YOUSSEF	350 000	FES	14	RADI	05 000
B. Med V	900 000	CASA	17	Mme. ALAMI	50 000

La suppression des prêts d'une agence entraîne la perte des informations relatives à cette agence.

IV. NORMALISATION

- Problème :

Etant donné un ensemble d'informations à représenter dans une B.D :

Comment déterminer la structure logique adéquate de ces données ?

Quelles seront les relations et quels attributs vont elles avoir ?

- Normaliser :

Créer des schéma relationnels sans redondances inutiles, répondant à un standard :

« forme normale ».

- Besoin :

Informations complémentaires sur l'univers modélisé par la B.D : contraintes de

dépendance de données : « dépendances fonctionnelle ».

- Dépendance fonctionnelle :

X et Y deux (ou plusieurs) attributs de la B.D.

$X \rightarrow Y$: Y dépend fonctionnellement de X

La connaissance de la valeur de x entraîne la connaissance de la valeur de Y.

X détermine Y.

Pour une valeur de X, il existe une et une seule valeur de Y.

Exemple :

Agence \rightarrow Ville

Prêt \rightarrow Montant

Id_article \rightarrow désignation

(Numcom, NumLigne) \rightarrow Idarticle

- Formes normales :

- Première forme normale (1NF) :

Domaine atomique :

Ses éléments sont indivisibles (une seule valeur).

Exemple : l'ensemble des entiers est atomique,

Un schéma R est en 1NF Si et seulement si les domaines de tous ses attributs sont atomiques

Exemple :

Clientèle = (IdClient, nom, adresse, tel)

IdClient	Nom	Adresse		Tel
100	Triki	2, rue zerhoune	rabat	776532
101	Alami	5, rue oujda	rabat	756434
102	Badidi	70, rue sebou	casa	346754

Adresse comporte 2 valeurs : adresse et ville, d'où :

Clientèle = (IdClient, nom, adresse, ville, tel) est en 1NF

IdClient	Nom	Adresse	Ville	Tel
100	Triki	2, rue zerhoune	Rabat	776532
101	Alami	5, rue oujda	Rabat	756434
102	Badidi	70, rue sebou	Casa	346754

➤ Deuxième Forme Normale (2FN) :

*Un schéma R est en 2FN :
 Si et seulement si
 Tout attribut de R, n'appartenant pas à la clé
 primaire, est en dépendance fonctionnelle totale de
 la clé primaire*

Exemple :

Soit le schéma crédit :

Crédit = (agence, prêt, client, montant)

Agence	Prêt	Client	Montant
B. Med. V	17	Mr. ALAMI	50 000
MURSULTAN	53	BADIDI	20 000
ZERKTOUNI	15	TRIKI	15 000
My. YOUSSEF	14	RADI	05 000
B. Med. V	17	Mme. ALAMI	50 000

Prêt @ client n'est pas une DF vérifiée sur crédit

Répétition des attributs : agence, prêt, et montant.

En considérant les DF :

Prêt → agence

Prêt → montant

Crédit peut être décomposée en :

Dossier-crédit = (prêt, agence, montant)

Bénéficiaire = (prêt, client)

Dossier-crédit

Prêt	Agence	Montant
17	B. Med. V	50 000
53	MURSULTAN	20 000
15	ZERKTOUNI	15 000
14	My. YOUSSEF	05 000

Bénéficiaire

Prêt	Client
17	Mr. ALAMI
17	Mme. ALAMI
53	BADIDI
15	TRIKI
14	RADI

Exemple :

Considérons le schéma Dépôt :

Dépôt = (agence, compte, client, position)

Avec les DF :

Compte → agence

Compte → position

Compte → client n'est pas imposée,

Dépôt peut être décomposé en :

Dossier_compte = (compte, agence, position)

Déposant = (compte, client)

➤ Troisième forme normale (3NF) :

Un schéma R est en 3NF ssi

- *R est en 2NF,*
- *Aucun attribut ne dépend transitivement de la clé primaire,*
(tout attribut de R, n'appartenant pas à la clé, ne dépend que de la clé),

NumClient → NumFacture => pas de DF
NumFacture → NumClient => DF direct
NumFacture → NomClient => DF indirect

En effet :

NumFacture → NumClient → NomClient => DF par transitivité.

Donc pour : R = (NumFacture, DateFacture, TotalFacture, NumClient, NomClient)

La clé est NumFacture. R est en 2NF.

La solution en 3NF :

R1 = (NumFacture, DateFacture, TotalFacture, NumClient)

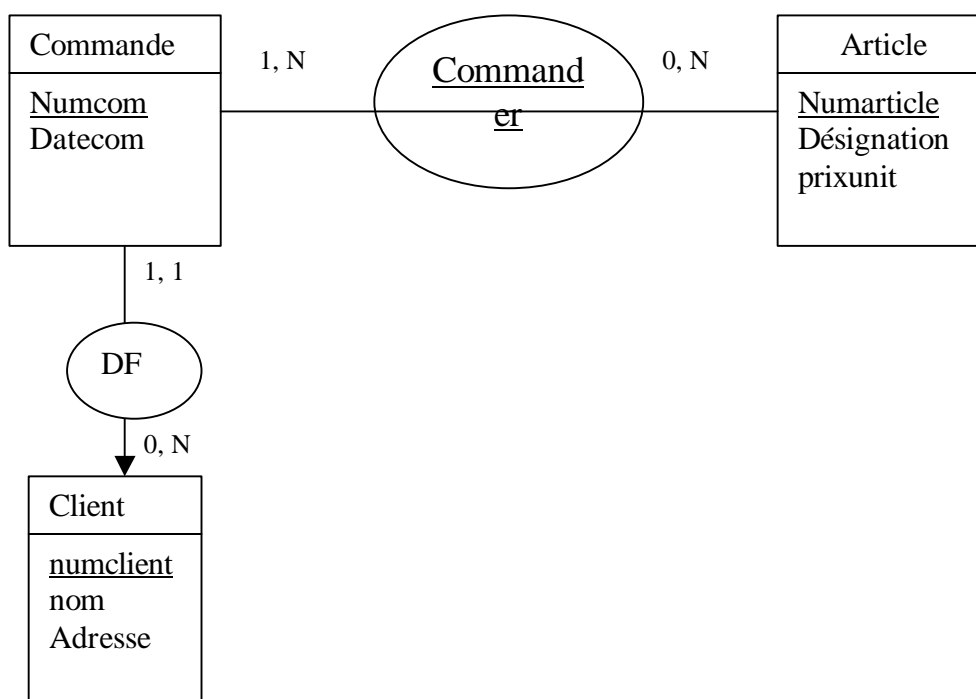
R2 = (NumClient, NomClient)

➤ Conclusion :

Les objectifs à atteindre lors de l'élaboration d'une B.D sont :

- Eviter la redondance (sauf si nécessaire pour assurer la jointure),
- Ne pas perdre l'information,
- Respecter les règles de gestion du système.
 - Ces contraintes sont obtenues en respectant la 3NF lors de l'élaboration de la B.D.
 - La représentation des attributs dans un MCD normalisé (respectant les 3FN) est un passage de conception vivement recommandé avant d'élaborer le modèle logique de données (MLD) de la B.D. conçue.

- Modèle Conceptuel de Données de l'exemple "Étude des commandes"



V. PASSAGE DU MCD AU MLD

- La normalisation permet d'obtenir un MCD respectant les trois formes normales.

- Le passage d'un MCD normalisé à un MLD reflétant la structure d'une base de données bien conçue doit respecter les règles suivantes :
 - Toute entité forte est transformée en une table ayant la même clé que celle de cette entité,

 - Toute entité faible est transformée en une table ayant pour clé son discriminant ajouté à la clé de l'entité forte dont elle dépend,

 - Toute relation « n-n » (ayant pour cardinalités maximales « n » avec les différentes entités) est transformée en une table ayant pour clé le groupe d'attributs formé des différentes clés des entités associées par cette relation,

 - Toute DF entre deux entités (cardinalité maximale égale 1) est transformée par l'ajout de la clé de l'entité dépendante à l'ensemble des attributs de la table correspondante à l'entité dont la première entité dépend. L'attribut ajouté est dit clé étrangère au niveau de cette dernière table.

Exemple :

Modèle Logique de Données de l'exemple "Étude des commandes"

Table **Client**

Clé	Nom
#	NUMCLIENT
	NOM
	ADRESSE

Table **Article**

Clé	Nom
#	NUMARTICLE
	DESIGNATION
	PRIXUNIT

Table **Commande**

Clé	Nom
#	NUMCOM
	DATECOM
	NUMCLIENT

Table **Commander**

Clé	Nom
#	NUMCOM
#	NUMARTICLE
	QTECOM