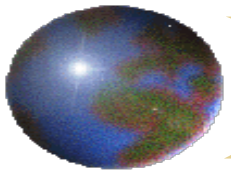


# ***BASE DE DONNEES***

Dr. Coulibaly Tiékoura



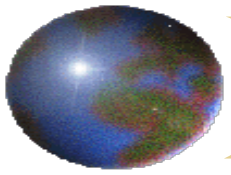
# *Plan Général du cours*

- ✚ **Chapitre 1: Introduction générale**
- ✚ **Chapitre 2: Le modèle relationnel**
- ✚ **Chapitre 3: L'algèbre relationnelle**
- ✚ **Chapitre 4: Le langage SQL**
- ✚ **Chapitre 5: Administration avec l'outil phpMyAdmin**

# ***BASE DE DONNEES***

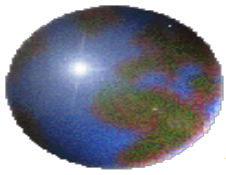
## **Chap 1. Introduction générale**

Dr. Coulibaly Tiékoura

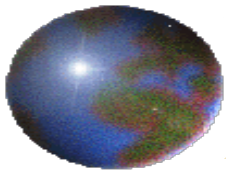


# *Plan du Chapitre 1*

- ✚ **I. Notions intuitives**
- ✚ **II. Objectifs et avantages des SGBD**
- ✚ **III. L'architecture ANSI/SPARC**
- ✚ **IV. Notion de modélisation des données**
- ✚ **V. Les différents modèles de données**
- ✚ **VI. Bref historique, principaux SGBD commercialisés**



# *I. Notions intuitives*



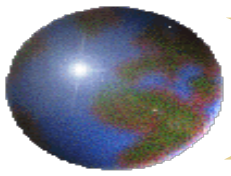
# *I. Notions intuitives*

## ✚ **Base de données**

- ❑ Ensemble structuré de données apparentées qui modélisent un univers réel
- ❑ Une BD est faite pour enregistrer des faits, des opérations au sein d'un organisme (administration, banque, université, hôpital, ...)
- ❑ Les BD ont une place essentielle dans l'informatique

## ✚ **Qualités d'une BDD**

- Données non redondantes
- Données cohérentes
- Consultable et modifiable aisément
- Fort degré d'indépendance entre données et matériel
- Fort degré d'indépendance entre données et programmes



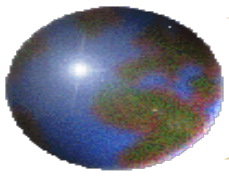
# *I. Notions intuitives*

## ✚ **Système de Gestion de Base de Données (SGBD)**

= DATA BASE MANAGEMENT SYSTEM (DBMS)

- ❑ Système qui permet de gérer une BD partagée par plusieurs utilisateurs simultanément
- ❑ Ensemble de services permettant de gérer des bases de données :
  - décrire et modifier la structure des données
  - manipuler les données : insérer, modifier, supprimer
  - consulter les données de façon simple (requête)
  - assurer sécurité, confidentialité (authentification), intégrité, cohérence
  - permettre des accès simultanés (multi-utilisateurs)
- ❑ **Exemples** :
  - MySQL (Open Source, pour tous systèmes)
  - PostgreSQL (Open Source, université de Berkeley )
  - Oracle
  - DB2 (IBM)
  - Access (Microsoft)
  - SQL Server (Microsoft)





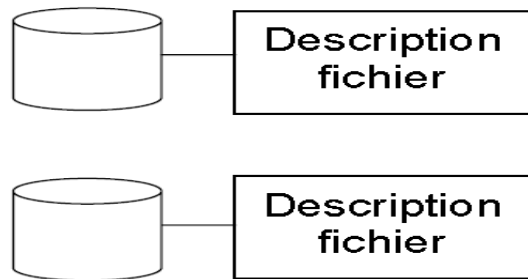
# I. Notions intuitives

## ✚ Des fichiers aux Base de Données

Séparation des données et des programmes

FICHIER

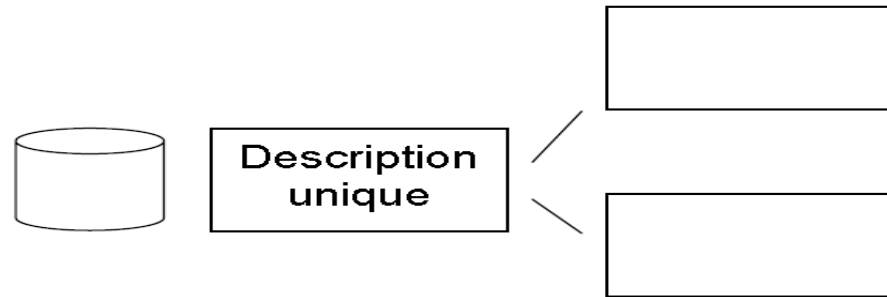
Les données des fichiers sont décrites dans les programmes



Programmes

BASE DE DONNEES

Les données de la BD sont décrites hors des programmes dans la base elle-même

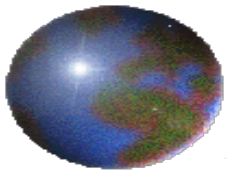


Programmes

La multiplication des fichiers entraînait la *redondance* des données, ce qui rendait difficile les mises à jour.

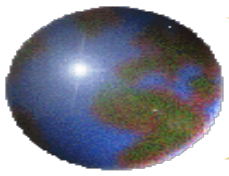
D'où l'idée *d'intégration* et de *partage* des données





# *II*

## *Objectifs et avantages des SGBD*



## II. Objectifs et avantages des SGBD

### ✚ Que doit permettre un SGBD ?

- ❑ **Décrire les données**

indépendamment des applications (de manière intrinsèque)

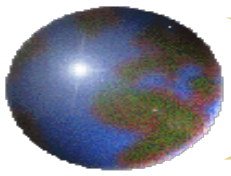
⇒ *langage de définition* des données ou **DATA DEFINITION LANGUAGE (DDL)**

- ❑ **Manipuler les données**

- ❑ Interroger et mettre à jour les données

langage de *requêtes* déclaratif. *ex.: quels sont les noms des produits de prix < 100F ?*

⇒ **langage de manipulation** des données ou **DATA MANIPULATION LANGUAGE (DML)**



## II. Objectifs et avantages des SGBD

### ✚ Que doit permettre un SGBD ?

#### ❑ Contrôler les données

##### intégrité

vérification de contraintes d'intégrité (*ex.: le salaire doit être compris entre 400F et 20000F*)

##### confidentialité

contrôle des droits d'accès, autorisation

⇒ **langage de contrôle des données** ou **DATA CONTROL LANGUAGE (DCL)**

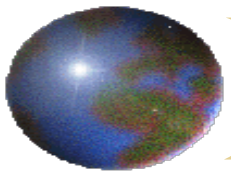
#### ❑ Partage

une BD est partagée entre plusieurs utilisateurs en même temps

⇒ **contrôle des accès concurrents**

notion de **transaction**

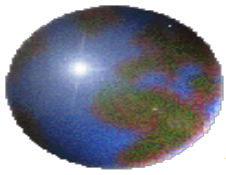
L'exécution d'une transaction doit préserver la cohérence de la BD



## *II . Objectifs et avantages des SGBD*

### ✚ Que doit permettre un SGBD ?

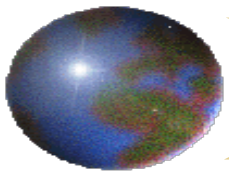
- ❑ **Sécurité:** reprise après panne, journalisation
- ❑ **Indépendance physique**
  - Pouvoir modifier les structures de stockage ou les index sans que cela ait de répercussion au niveau des applications.
- ❑ **Indépendance logique**
  - Permettre aux différentes applications d'avoir des vues différentes des mêmes données
  - Permettre au DBA de modifier le schéma logique sans que cela ait de répercussion au niveau des applications.

A blurred background image of a hand typing on a computer keyboard, centered behind the title text.

# *III*

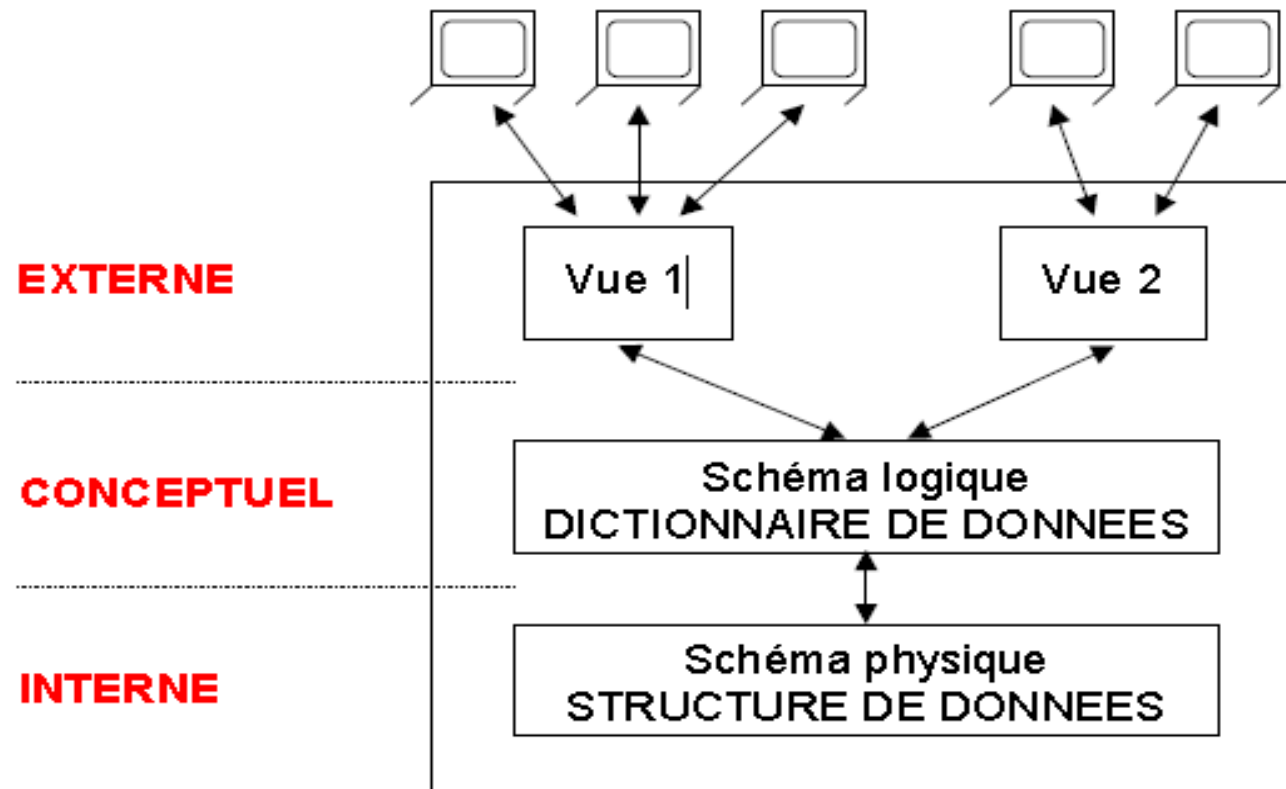
## *L'architecture*

### *ANSI/SPARC*



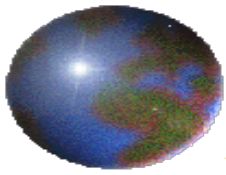
### III. L'architecture ANSI/SPARC

- Proposition en 75 de l' ANSI/SPARC (*Standard Planning And Requirement Comitte*)
- 3 niveaux de représentation des données



SGBD

Niveaux de représentation des données



### *III. L'architecture ANSI/SPARC*

#### ✚ **Le niveau externe**

- ❑ Le concept de *vue* permet d'obtenir l'indépendance logique
- ❑ La modification du schéma logique n'entraîne pas la modification des applications (une modification des vues est cependant nécessaire)
- ❑ Chaque vue correspond à la perception d'une partie des données, mais aussi des données qui peuvent être synthétisées à partir des informations représentées dans la BD (par ex. statistiques)

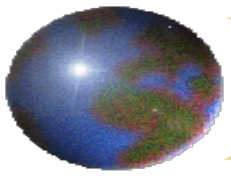
#### ✚ **Le niveau conceptuel**

- ❑ Il contient la description des données et des contraintes d'intégrité (Dictionnaire de Données)
- ❑ le schéma logique découle d'une activité de modélisation.

#### ✚ **Le niveau interne**

- ❑ Il correspond aux structures de stockage et aux moyens d'accès (index)





### III. L'architecture ANSI/SPARC

Pour résumer :

#### ✚ Les fonctions des SGBD

- **DEFINITION DES DONNEES**

⇒ *Langage de définition des données (DDL) (conforme à un modèle de données)*

- **MANIPULATION DES DONNEES**

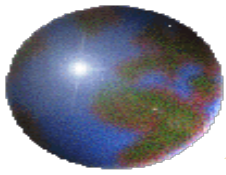
Interrogation; Mise à jour (insertion, suppression, modification)

⇒ *Langage de manipulation des données (DML) (langage de requête déclaratif)*

- **CONTRÔLE DES DONNEES**

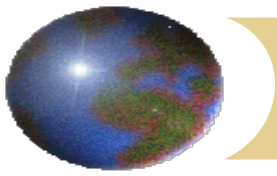
Contraintes d'intégrité; Contrôle des droits d'accès; Gestion de transactions

⇒ *Langage de contrôle des données (DCL)*



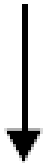
# *IV*

## *Notion de modélisation des données*

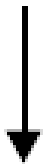


## *IV. Notion de modélisation des données*

**UNIVERS REEL**



**MODELE  
CONCEPTUEL  
MCD**



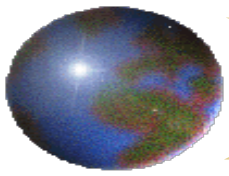
**SCHEMA  
LOGIQUE**

**Modèles sémantiques  
Orientés « conception »  
Entité-Association, Merise ...**

**Modèles de BD  
Hiérarchique, Réseau  
Relationnel ...**

- Les modèles de BD sont souvent trop limités pour pouvoir représenter directement le monde réel

- Méthodologies de conception présentées en ACSI, SGBD2



## *IV. Notion de modélisation des données*

### **Le modèle Entité-Association**

**EA** en français, **ER** en anglais (pour Entity Relationship)

Formalisme retenu par l'ISO pour décrire l'aspect conceptuel des données à l'aide d'entités et d'associations.

#### **Le concept d'entité**

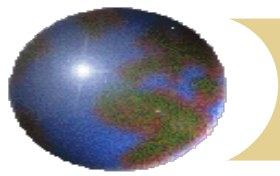
Représentation d'un objet matériel ou immatériel

Par exemple un employé, un projet, un bulletin de paie

Nom de l'entité
Liste des propriétés

☐ Les entités peuvent être regroupées en **types d'entités**

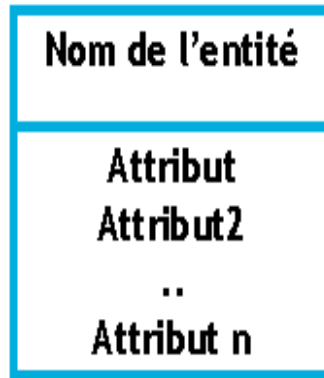
- Par exemple, on peut considérer que tous les employés particuliers sont des **instances** du type d'entité générique EMPLOYE
- Par exemple l'employé nommé DUPONT est une instance ou occurrence de l'entité EMPLOYE.



## IV. Notion de modélisation des données

### Le modèle Entité-Association (propriétés/identifiant)

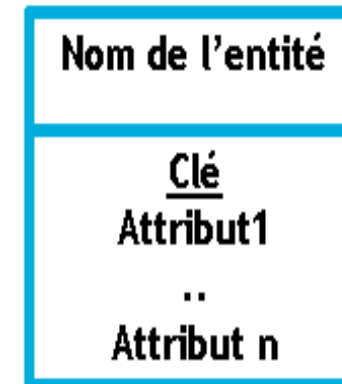
- **Attribut** : Une donnée élémentaire qui décrit une entité



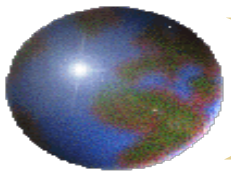
- **Un attribut peut être :**

- Atomique = comporte un seul champ
  - Exemple : Prénom, Nom..
- Composé = comporte plusieurs valeurs
  - Exemple : Date de naissance (jour, mois, année)
- Monovalué = une seule valeur pour une entité donnée
  - Exemple : CIN
- Multivalué : plusieurs valeurs pour une entité donnée
  - Exemple : Auteurs, Diplômes

- On appelle identifiant ou clé primaire d'une entité un attribut dont la valeur est unique pour chaque entité



- Toute entité doit posséder au moins une clé puisque c'est elle qui permet de distinguer une entité d'une autre
- Une clé peut donc être un attribut, un ensemble d'attributs ou un attribut artificiel.
- Les attributs d'une clé doivent toujours avoir une valeur différente de NULL.



## *IV. Notion de modélisation des données*

### **Le modèle Entité-Association: (associations et cardinalités)**

- ❑ Représentation d'un lien entre deux entités ou plus
- ❑ Dans ce lien chaque entité joue un rôle déterminé conformément aux règles de gestion dictées par le système d'informations.

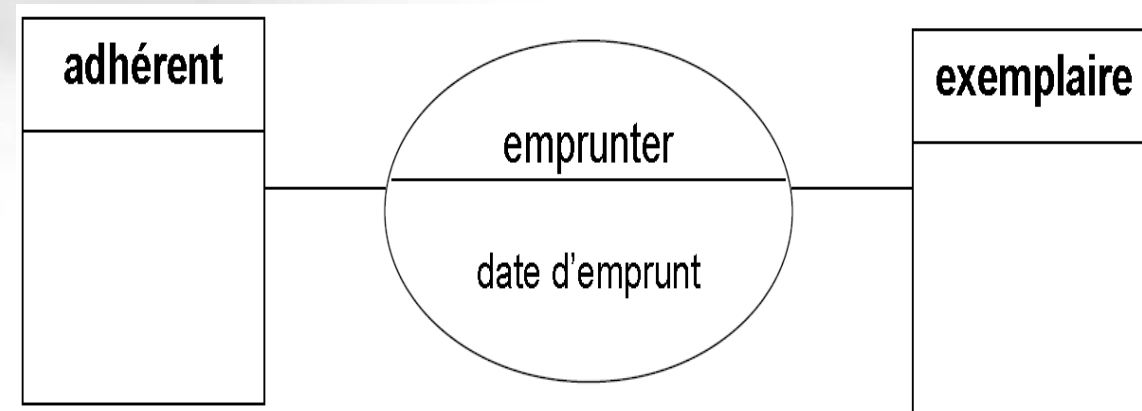
Exemples:

- L'acteur « X » a joué dans le film « Y »
- L'auteur « A » a écrit l'ouvrage de « B »

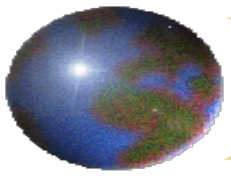


- une association peut avoir des propriétés particulières

Par exemple, la date d'emprunt d'un livre





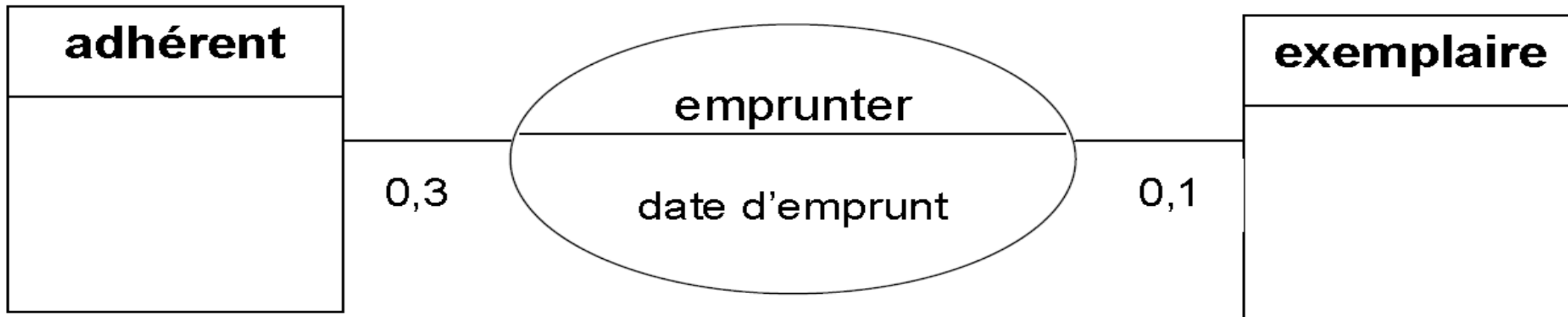


## IV. Notion de modélisation des données

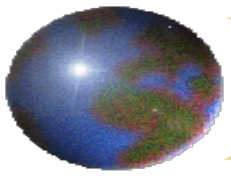
### Le modèle Entité-Association: (associations et cardinalités)

La cardinalité d'une association pour une entité constituante est constituée d'une borne minimale et d'une borne maximale :

- **Minimale** : nombre minimum de fois qu'une occurrence de l'entité participe aux occurrences de l'association, généralement **0 ou 1**
- **Maximale** : nombre maximum de fois qu'une occurrence de l'entité participe aux occurrences de l'association, généralement **1 ou n**







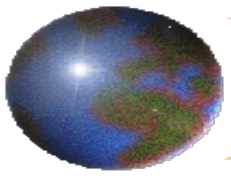
## *IV. Notion de modélisation des données*

### **Le modèle Entité-Association: (associations et cardinalités)**

- La cardinalité 0,3 indique qu'un adhérent peut être associé à 0, 1, 2 ou 3 livres, c'est à dire qu'il peut emprunter au maximum 3 livres.
- A l'inverse un livre peut être emprunté par un seul adhérent, ou peut ne pas être emprunté.
- Les cardinalités maximum sont nécessaires pour concevoir le schéma de la base de données
- Les cardinalités minimums sont nécessaires pour exprimer les contraintes d'intégrité

En notant uniquement les cardinalités maximum, on distingue 3 type de liens :

- **Lien fonctionnel 1:n**
- **Lien hiérarchique n:1**
- **Lien maillé n:m**

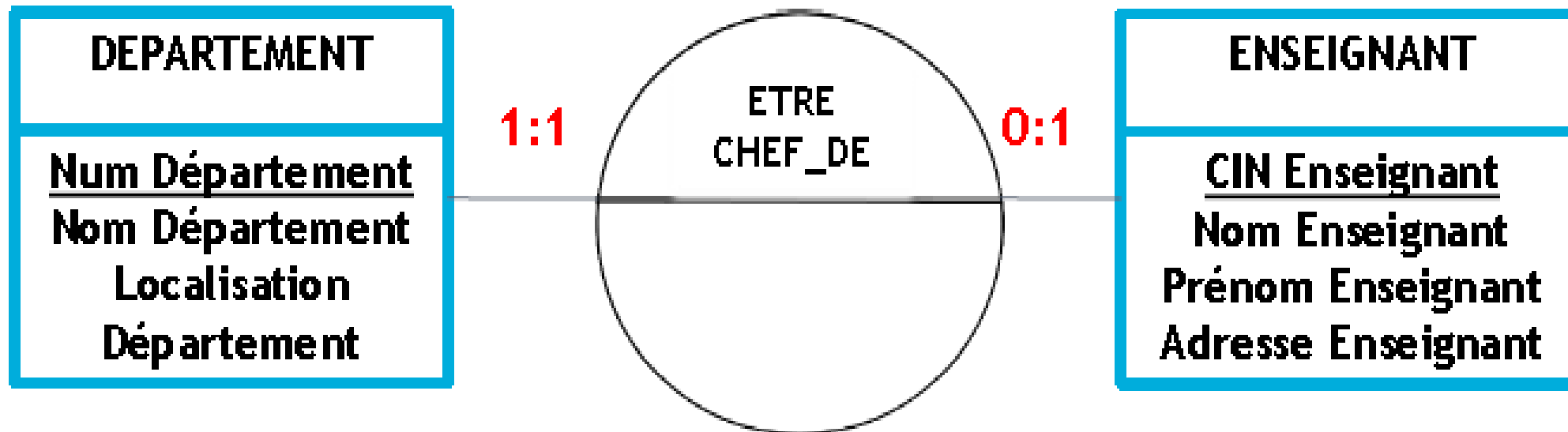


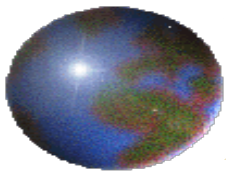
## IV. Notion de modélisation des données

### Le modèle Entité-Association: (Les associations: types de lien)

❑ On distingue 3 types de lien:

❖ **1 : 1** : à chaque occurrence d'une entité ou d'une classe d'entité, il existe **au plus une occurrence** d'entité de l'autre classe d'entité liée par l'association (un DÉPARTEMENT n'a qu'un seul chef et un ENSEIGNANT ne peut être chef que d'un DÉPARTEMENT donc l'association CHEF\_DE est 1 : 1).

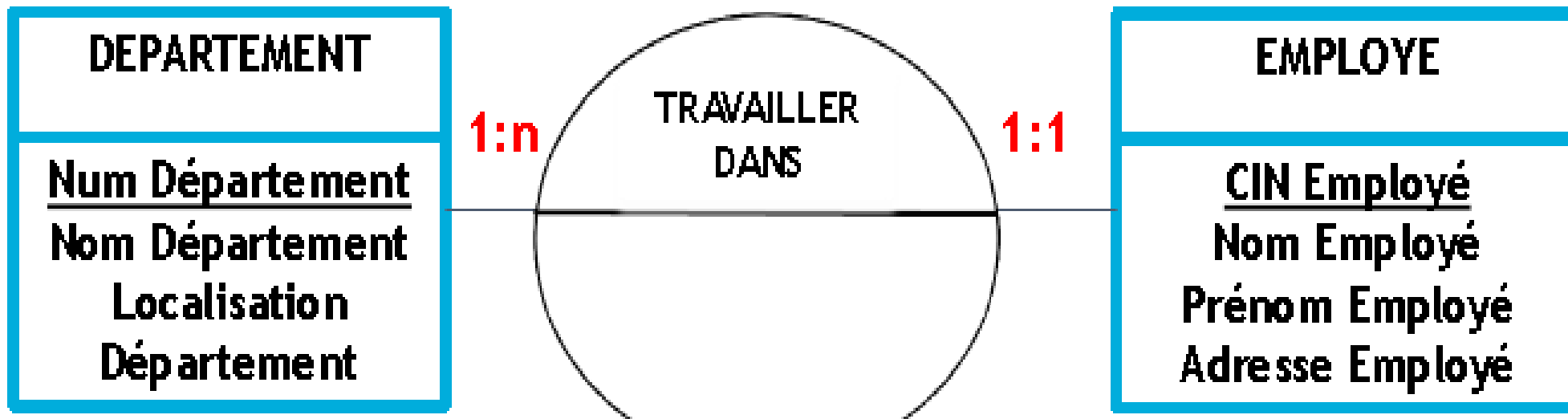


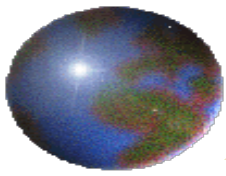


## IV. Notion de modélisation des données

### Le modèle Entité-Association: (Les associations: types de lien)

- ❖ **1 : n** : à toute occurrence d'une entité de E1, il existe **au plus une** occurrence d'entité de E2 liée par l'association **et** à chaque occurrence d'une entité de E2, on associe **0 ou plusieurs** occurrences d'entité de E1 (un EMPLOYÉ travaille dans un unique DÉPARTEMENT et un DÉPARTEMENT emploie plusieurs employés indiquent que l'association TRAVAILLER est 1 : n).

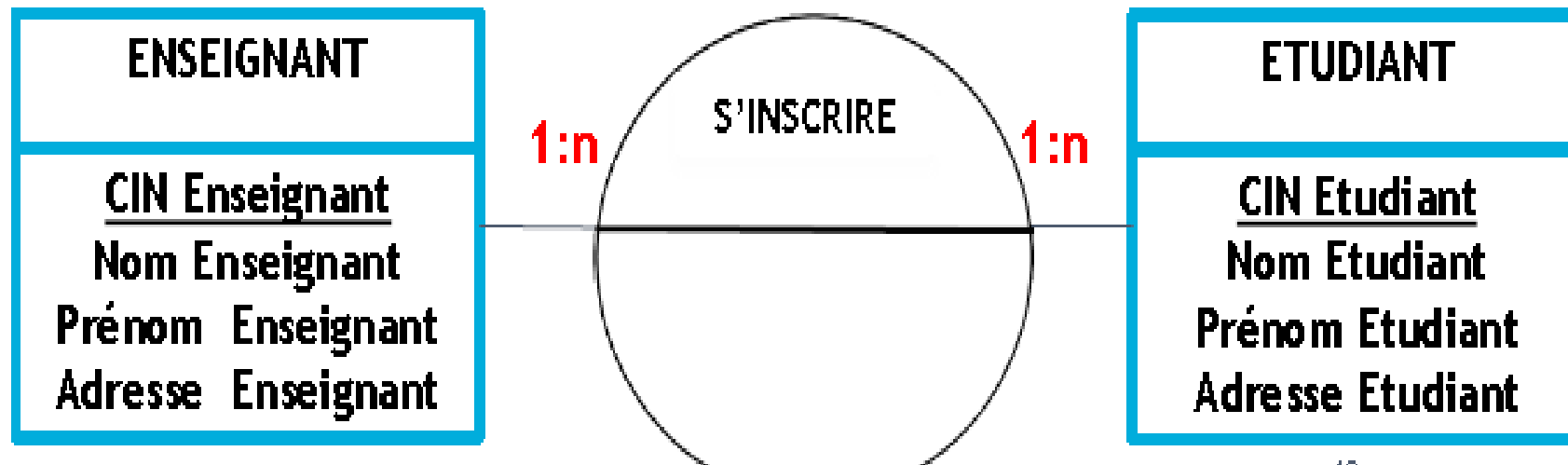


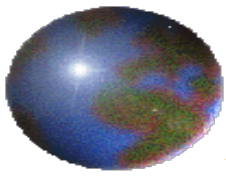


## IV. Notion de modélisation des données

### Le modèle Entité-Association: (Les associations: types de lien)

❖ **n : m** : à toute occurrence d'une entité de E1, il existe **0 ou plusieurs** occurrences d'entité de E2 liée par l'association **et** à chaque occurrence d'une entité de E2, on associe **0 ou plusieurs** occurrences d'entité de E1 (un ÉTUDIANT s'inscrit à plusieurs ENSEIGNEMENT et un ENSEIGNEMENT inscrit plusieurs ÉTUDIANT donc l'association S'INSCRIT est n : m).

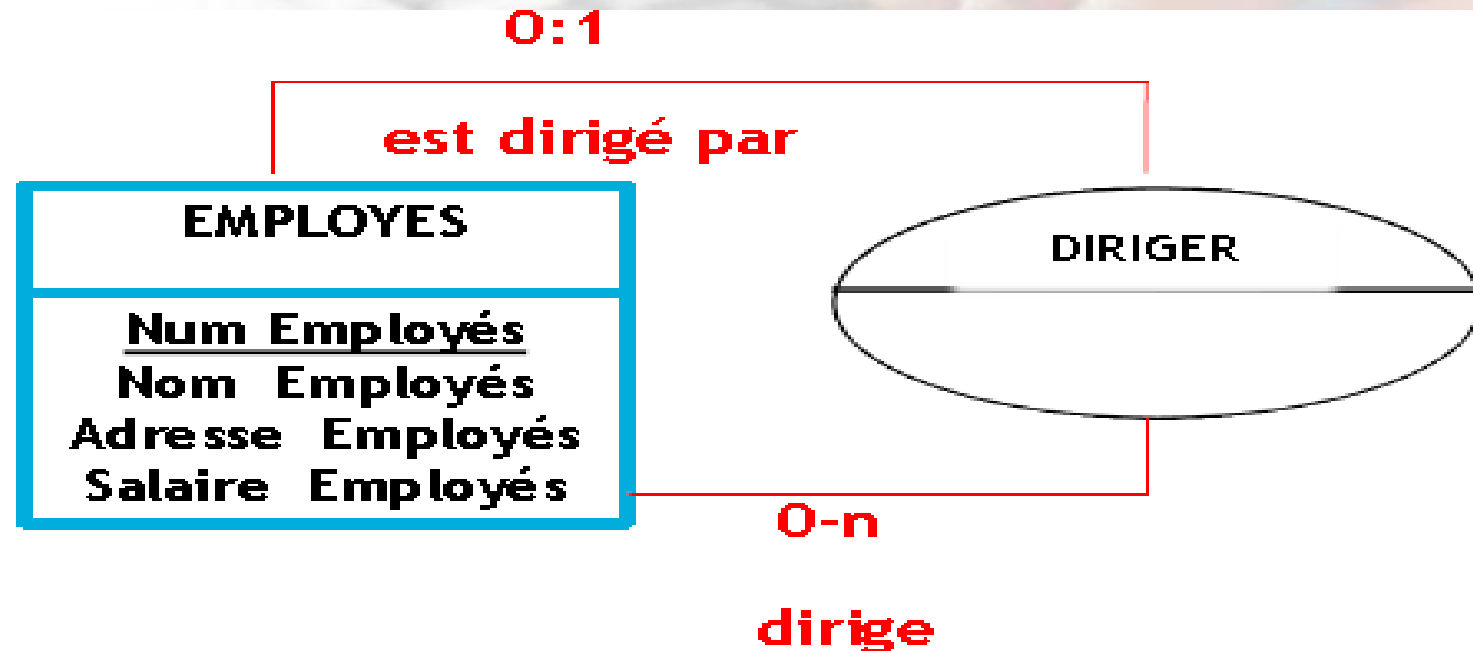


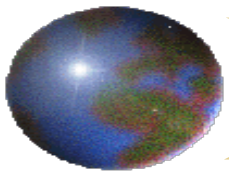


## IV. Notion de modélisation des données

### Le modèle Entité-Association: (Cas particulier: liaison unaire ou réflexive)

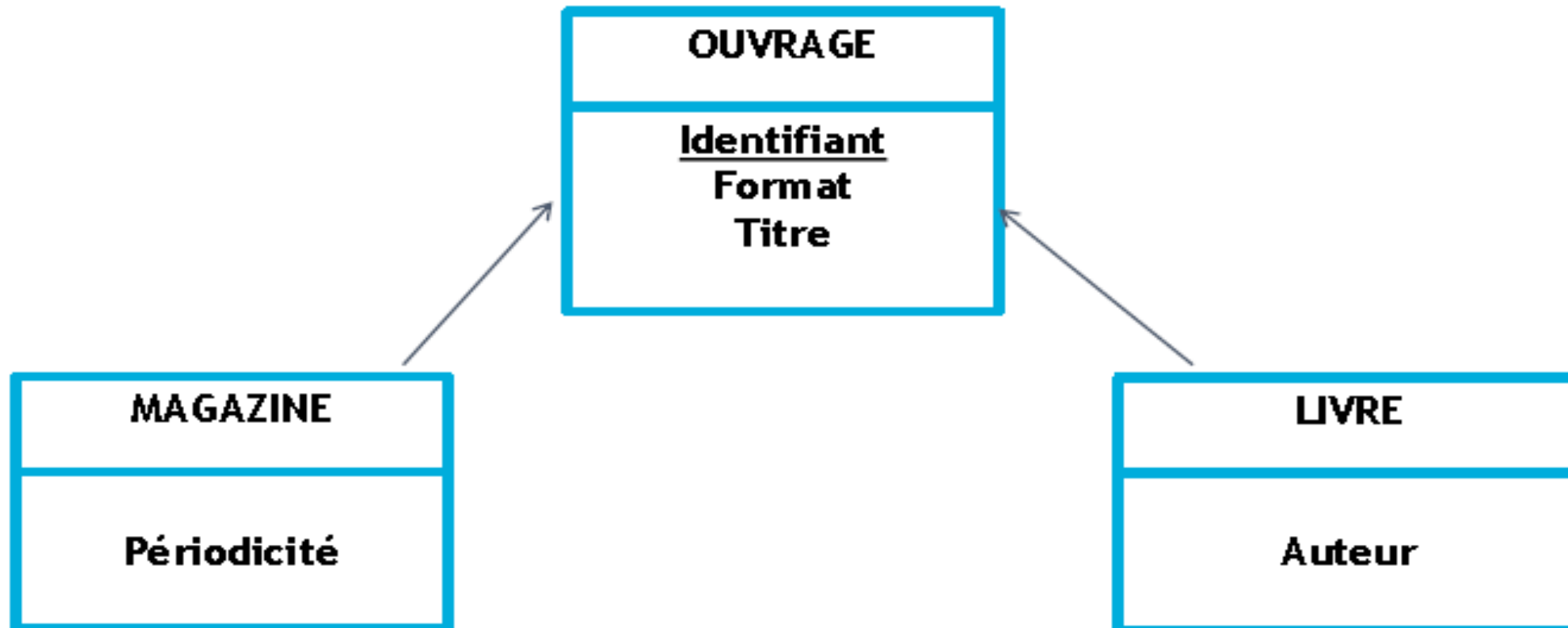
Une association réflexive est une association binaire qui fait intervenir au moins deux fois la même entité

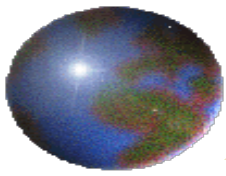




## *IV. Notion de modélisation des données*

**Le modèle Entité-Association:** (Cas particulier: liaison d'inclusion /sous entité)

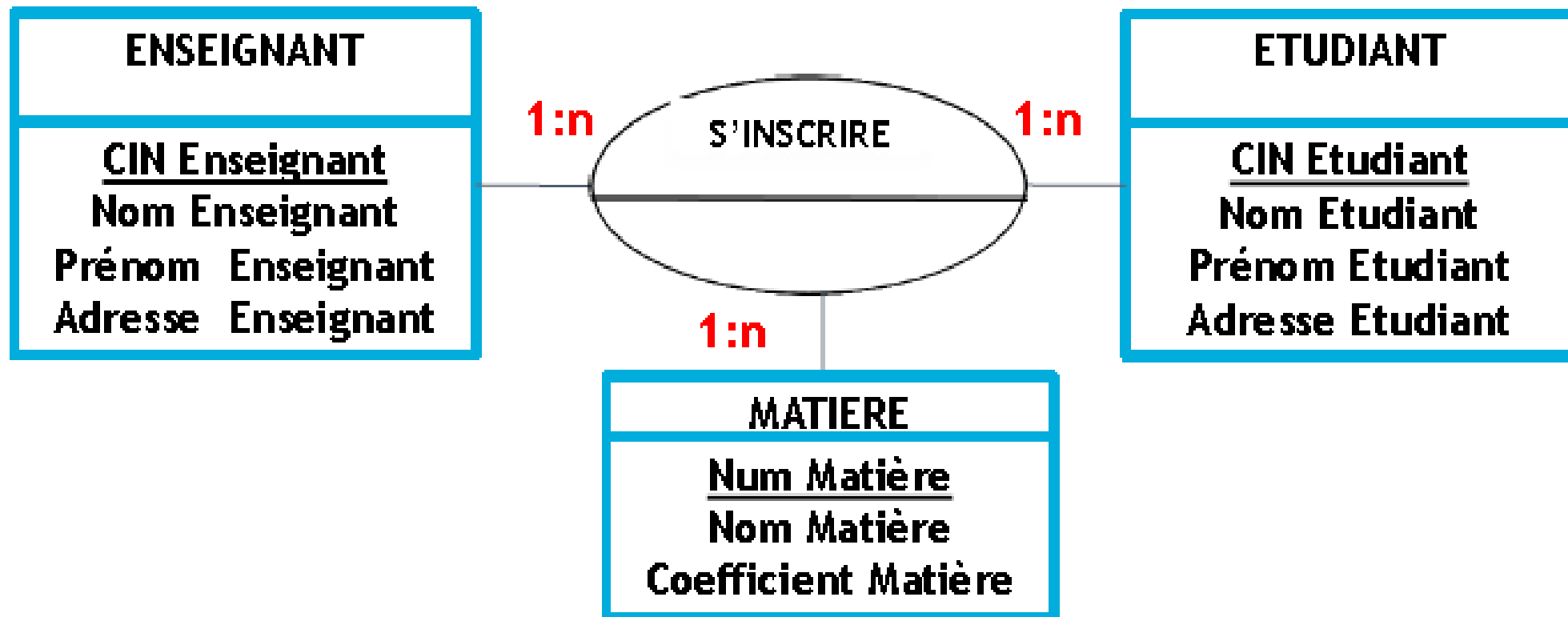




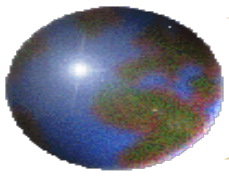
## IV. Notion de modélisation des données

### Le modèle Entité-Association: (Cas particulier: liaison ternaire)

- La liaison ternaire exprime le lien entre une entité et un couple d'entités
- Des attributs peuvent également qualifier cette association.



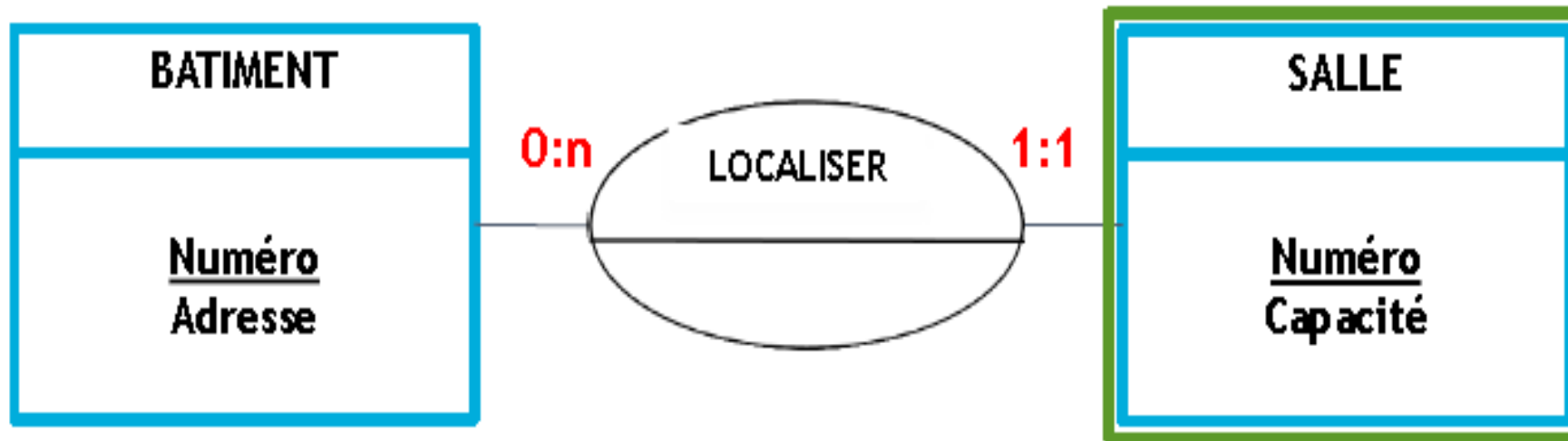


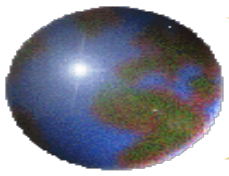


## IV. Notion de modélisation des données

### Le modèle Entité-Association: (Cas particulier: entité faible)

- Formalisée comme une entité mais son identification s'effectue relativement à une autre entité via une association
- L'identifiant de l'entité faible est constituée de celui de l'entité faible et de celui de l'entité forte liée

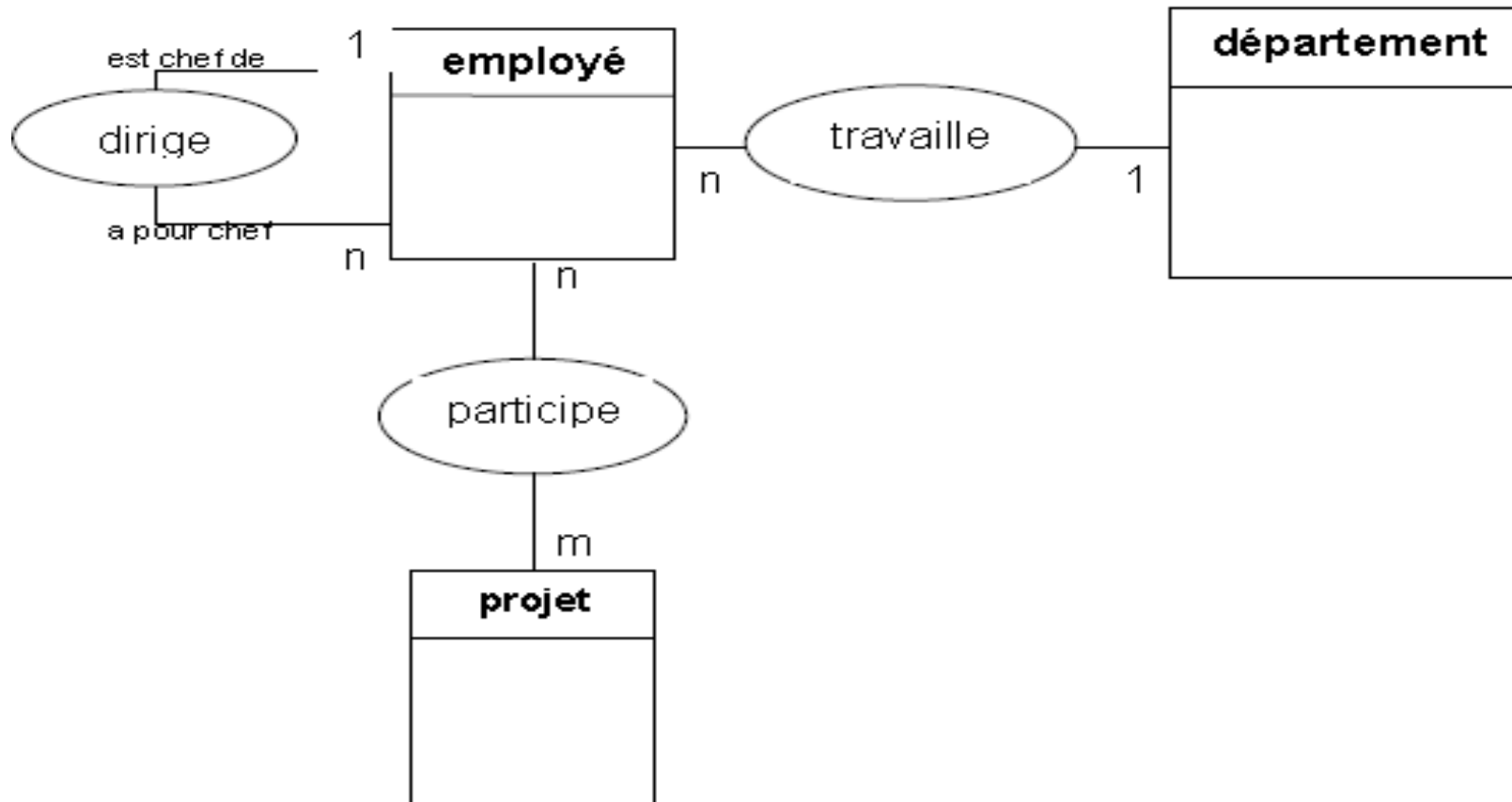


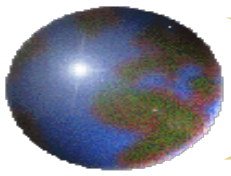


## IV. Notion de modélisation des données

### Le modèle Entité-Association

- ❑ Exemple de diagramme Entité Association (ou Modèle Conceptuel de Données (MCD))



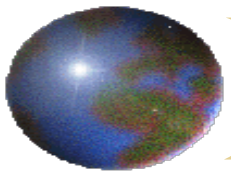


## *IV. Notion de modélisation des données*

### **Le modèle Entité-Association (Démarche générale)**

✚ Définir un modèle entité-association correspond donc à :

- ✚ chercher les classes d'entités et leurs attributs ;
- ✚ chercher les liens et leurs attributs ;
- ✚ analyser les types de liens ;
- ✚ définir les clés ;
- ✚ construire le diagramme résultant.

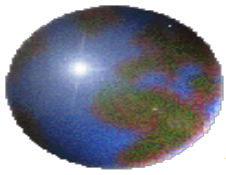


## *IV. Notion de modélisation des données*

### Le modèle Entité-Association (Démarche générale d'un MCD )

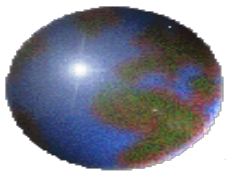
✚ L'élaboration du MCD passe par les étapes suivantes :

1. La mise en place de **règles de gestion** (si celles-ci ne vous sont pas données),
2. L'Analyse de l'existant et élaboration du **dictionnaire des données** (épuré),
3. La recherche des **dépendances fonctionnelles** entre ces données,
4. Dégager les '**entités naturelles**' grâce aux identifiants existants déjà dans l'organisation
5. Rattacher les **propriétés** aux entités
6. Recenser les **associations** entre entités et leur rattacher leurs éventuelles propriétés
7. Déterminer les **cardinalités**
8. Décomposer si possible les **associations n-aires** (cf. règles)
9. S'assurer de la conformité du modèle aux règles de construction (cf. règles)
10. Normaliser le modèle : s'assurer qu'il est en 3FN



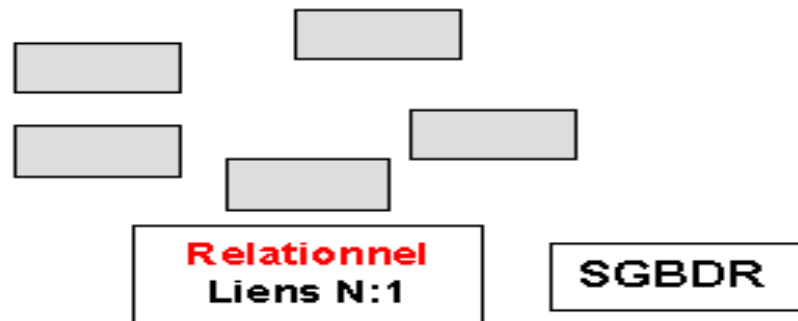
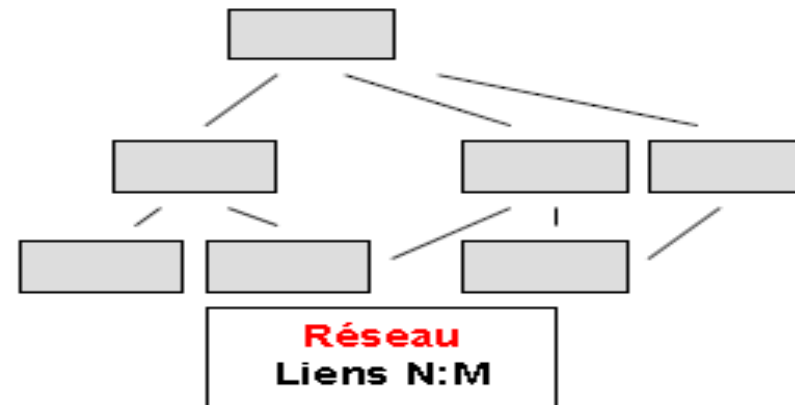
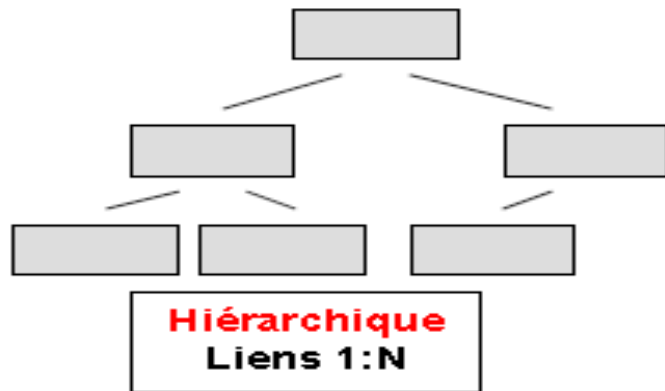
V

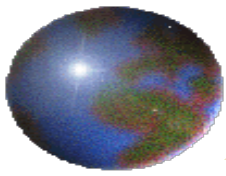
# *Les différents modèles de données*



## V. Les différents modèles de données

- ❑ L'organisation des données au sein d'une BD a une importance essentielle pour faciliter l'accès et la mise à jour des données.
- ❑ Les bases de données sont apparues à la fin des années 60, à une époque où la nécessité d'un système de gestion de l'information souple se faisait ressentir. Il existe cinq modèles de SGBD, différenciés selon la représentation des données qu'elle contient. Nous présenteront les trois (3) principaux:



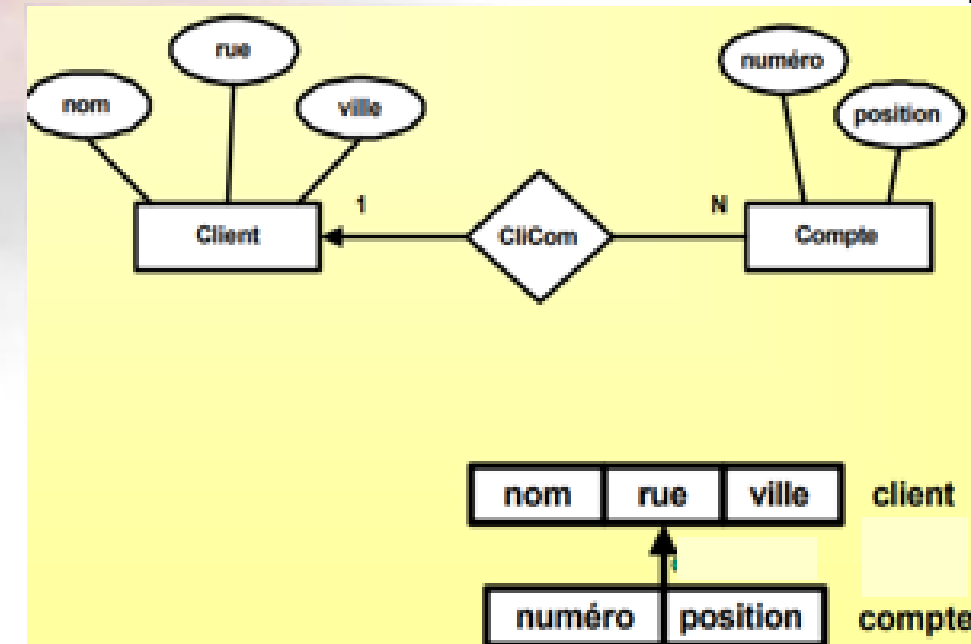
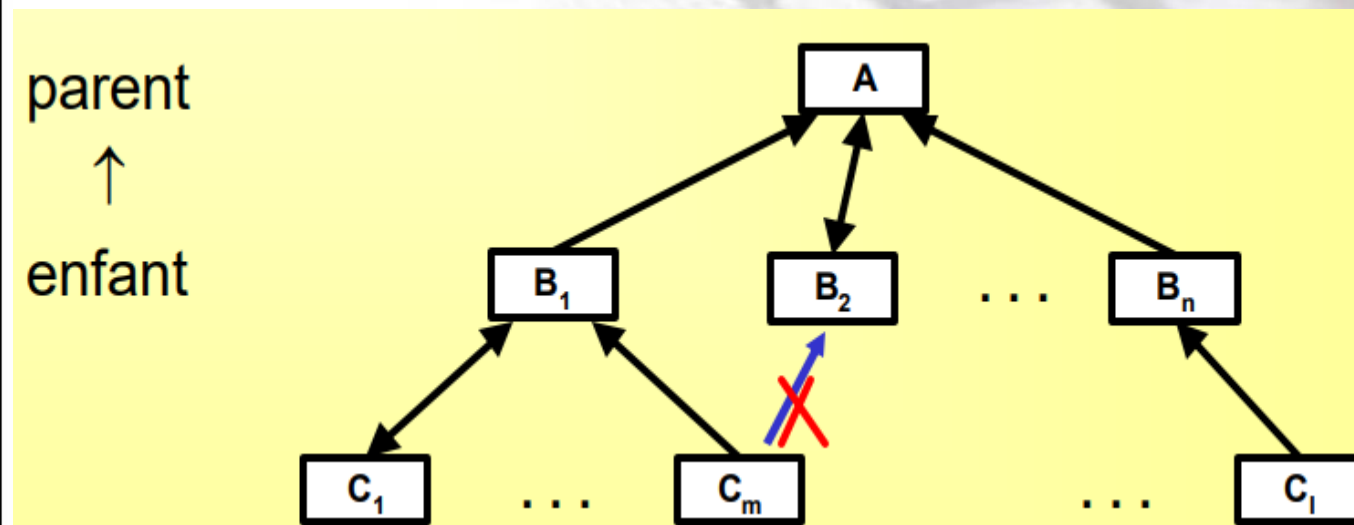


## V. Les différents modèles de données

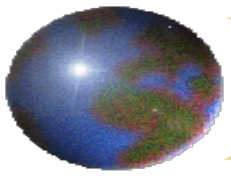
### LE MODÈLE HIÉRARCHIQUE

- ✚ Il s'agit du premier modèle de SGBD. Schéma logique représenté par un **ARBRE**
- ✚ les données sont classées hiérarchiquement, selon une arborescence descendante
- ✚ ce modèle permet de représenter une relation «un à plusieurs» (1:N) ou «un à un» (1:1).
- ✚ les nœuds se connectent les uns aux autres en formant une relation **parent-enfant**. Chaque nœud enfant n'aura **qu'un seul parent**.

- Exemple de schéma hiérarchique:







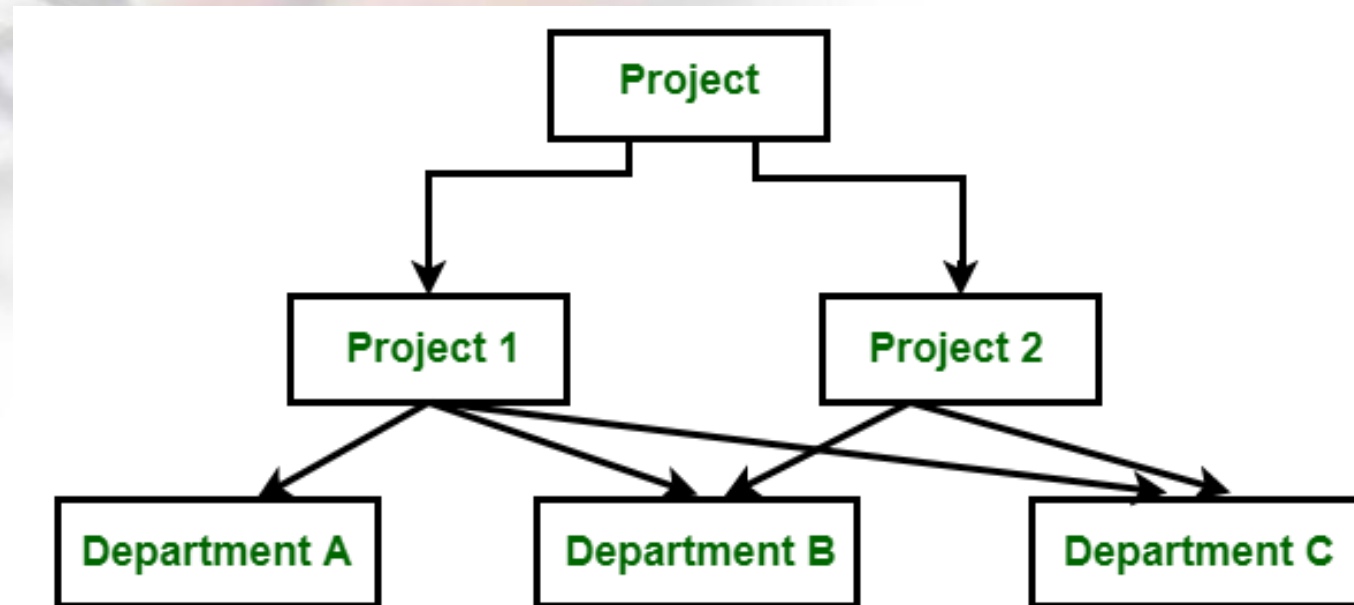
## V. Les différents modèles de données

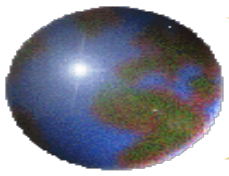
### LE MODÈLE RÉSEAU

- ✚ Le modèle réseau est une extension du modèle hiérarchique, qui organise les données dans une structure similaire à un graphique. Schéma logique représenté par un **GRAPHE**
- ✚ Contrairement au modèle hiérarchique, un nœud du modèle réseau peut avoir plusieurs parents.
- ✚ En plus des relations, **1:1 et 1:n**, il implémente également des relations plusieurs à plusieurs (**n:m**).
- ✚ les données de la BD créées à l'aide du modèle de réseau auront davantage de relations
- ✚ permet d'accéder aux données facilement et beaucoup plus rapidement.

- Exemple de schéma réseau:

Dans cet modèle, un nœud enfant peut avoir plus d'un parent.





## V. Les différents modèles de données

### LE MODÈLE RELATIONNEL

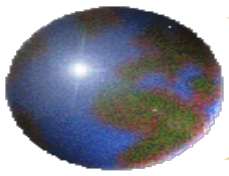
- ✚ Ce modèle est introduit par Edgar Frank Codd en 1970 ainsi que les 12 règles de Codd permettant de vérifier si un SGBD est un modèle de base de données relationnelle.
- ✚ les données sont enregistrées dans des tables à 2 dimensions (lignes et colonnes) appelées relations.
- ✚ La manipulation des données se fait selon le concept mathématique de relation de la théorie des ensembles, c'est-à-dire l'algèbre relationnelle. Les opérations relationnelles permettent de créer une nouvelle relation (table) à partir d'opérations élémentaires sur d'autres tables (par exemple l'union, l'intersection, ou encore la différence).
- ✚ Il implémente les différents types de relations (**1:1** , **1:n**, **n:m**)

- Exemple de schéma relationnel:

Attributs

Marque	Modèle	Série	Numéro
Renault	18	RL	4698 SJ 45
Peugeot	309	Chorus	5647 ABY 82
Ford	Escort	Match	8562 EV 23

Tuples  
(N-uplets)



## V. Les différents modèles de données

### LE MODÈLE RELATIONNEL

- Les relations représentent les **entités** du monde réel (comme des personnes, des objets, etc.) ou les **associations** entre ces entités
- Passage d'un schéma conceptuel E-A à un schéma relationnel.
  - une **entité** est représentée par la relation : **nom\_de\_l'entité** (*liste des attributs de l'entité*)
  - une **association** M:N est représentée par la relation : **nom\_de\_l'association** (*liste des identifiants des entités participantes, liste des attributs de l'association*)

Ex . :

CLIENT (IdCli, nom, ville)

PRODUIT (IdPro, nom, prix, qstock)

VENTE (IdCli, IdPro, date, qte)

➤ Représentation des données sous forme de **tables** :

CLIENT

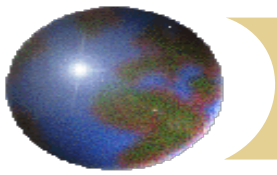
IdCli	Nom	Ville
X	Smith	Paris
Y	Jones	Paris
Z	Blake	Nice

PRODUIT

IdPro	Nom	Prix	Qstock
P	Auto	100	10
Q	Moto	100	10
R	Velo	100	10
S	Pedalo	100	10

VENTE

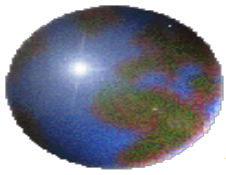
IdCli	IdPro	Date	Qte
X	P		1
X	Q		2
X	R		3
Y	P		4
Y	Q		5
Z	Q		6



## *V. Les différents modèles de données*

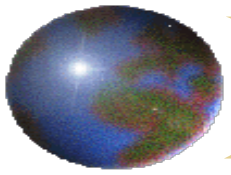
### **Principales différences entres les 3 modèles**

Modèle de données hiérarchique	Modèle de données réseau	Modèle de données relationnelles
Dans ce modèle, pour stocker la méthode de hiérarchie des données est utilisée. C'est la méthode la plus ancienne et elle n'est plus utilisée aujourd'hui.	Il organise les enregistrements les uns par rapport aux autres via des liens ou des pointeurs.	Il organise les enregistrements sous forme de table et les relations entre les tables sont définies à l'aide de champs communs.
Pour organiser les enregistrements, il utilise une structure arborescente.	Il organise les enregistrements sous forme de graphes orientés.	Il organise les enregistrements sous forme de array.
Il implémente les relations 1:1 et 1:n.	En plus de 1:1 et 1:n, il implémente également des relations plusieurs à plusieurs.	En plus de 1:1 et 1:n, il implémente également des relations plusieurs à plusieurs.
Une anomalie d'insertion existe dans ce modèle, c'est-à-dire que le nœud enfant ne peut pas être inséré sans le nœud parent.	Il n'y a pas d'anomalie d'insertion.	Il n'y a pas d'anomalie d'insertion.
Une anomalie de suppression existe dans ce modèle, c'est-à-dire qu'il est difficile de supprimer le nœud parent.	Il n'y a pas d'anomalie de suppression.	Il n'y a pas d'anomalie de suppression.
Ce modèle manque d'indépendance des données.	Il y a une indépendance partielle des données dans ce modèle.	Ce modèle offre une indépendance des données.
Il permet d'accéder à des données complexes et asymétriques.	Il permet d'accéder à des données complexes et symétriques.	Il permet d'accéder à des données complexes et symétriques.



# *VI*

## *Bref historique, principaux systèmes*



## *VI. Bref historique, principaux systèmes*

### ✚ **Années 60 Premiers développements des BD**

- ✚ fichiers reliés par des pointeurs
- ✚ systèmes IDS 1 et IMS 1 précurseurs des SGBD

### ✚ **Années 70 Première génération de SGBD**

- ✚ apparition des premiers SGBD
- ✚ séparation de la description des données de la manipulation de celles-ci par les applications
- ✚ modèles hiérarchique et réseau CODASYL
- ✚ langages d'accès navigationnels
- ✚ SGBD IDMS, IDS 2 et IMS 2

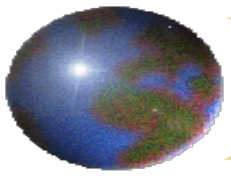
### ✚ **Années 80 Deuxième génération**

- ✚ modèle relationnel
- ✚ les SGBDR représentent l'essentiel du marché BD (aujourd'hui)
- ✚ architecture répartie client-serveur

### ✚ **Années 90 Troisième génération**

- ✚ modèles de données plus riches
- ✚ systèmes à objets OBJECTSTORE, O2





## *VI. Bref historique, principaux systèmes*

- ✚ **Principaux systèmes** (Oracle, DB2 (IBM), Ingres, Informix, Sybase, SQL Server (Microsoft), O2, Gemstone)
- ✚ **Sur micro** (Access, Paradox, FoxPro, 4D, Windev)
- ✚ **Sharewares** (MySQL, MSQL, Postgres, InstantDB)