

BASE DE DONNEES

Contact

- ◆ Olivier Curé
 - ◆ ocure@univ-mlv.fr
 - ◆ <http://igm.univ-mlv.fr/~ocure>
 - ◆ Copernic 4B060

Objectifs du cours

- ◆ Présentation des concepts liés aux bases de données, aux modèles des bases de données et aux systèmes de gestion de base de données.
- ◆ Vous permettre de concevoir des schémas et d'utiliser des base de données au sein d'applications.
- ◆ Comprendre que la conception d'une base de données n'est pas triviale, demande de la flexibilité et de la réflexion.
- ◆ Développer des applications Web avec PHP en profitant des possibilités d'une SGBDOR comme PostgreSQL.

Organisation du cours

- ◆ 4 cours de 2 heures
- ◆ 4 TD de 2 heures
- ◆ SGBD utilisé : PostgreSQL

Lectures recommandées

- ♦ Bases de données
 - ♦ Gardarin
- ♦ Fundamentals of Database Systems, 4th edition
 - ♦ Elmasri and Navathe. Addison Wesley
- ♦ A first course in database
 - ♦ Ullman and Widom. Computer Science Press
- ♦ Intelligent database systems
 - ♦ Bertino, Catania and Zarri. Addison Wesley
- ♦ Des structures aux bases de données
 - ♦ Carrez, Masson

1. Introduction

DATAAllegro
Microsoft



ORACLE

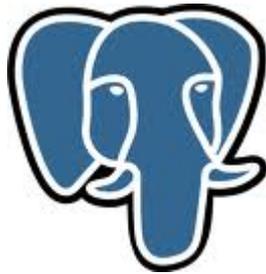


IBM DB2

SYBASE



TERADATA
Raising Intelligence



HyperSQL



ORACLE
BERKELEY DB

VoltDB

VERTICA



INGRES



VERSANT
THE DATABASE FOR DEVELOPERS

aster data
big data. fast insights.



ScaleDB



calpont



db4objects

PARACCEL

INFOBRIGHT



Relational

Object Relational



Embedded



Memory centric



Column stores / data warehouse



Object



XML



Network

IDMS

Hierarchy

IBM IMS

RDBMS market

- ◆ 2007 :
 - ◆ US \$ 18.8billion
 - ◆ 5 vendors take up 90% of the market
 - ◆ Oracle 44%
 - ◆ IBM DB2 21%
 - ◆ Microsoft SQL Server 18.5%
 - ◆ Sybase 3.5%
 - ◆ Teradata 3%

Key Value stores



Voldemort
TokyoCabinet
Scalaris
Amazon Dynamo & SimpleDB

Dynomite

Graph databases



Document databases

Terrastore



Column family databases

Google BigTable



Pourquoi étudier les bases de données

- ◆ Une fondation théorique appliquée à des besoins et problèmes réels et concrets.
- ◆ Regroupe de nombreuses notions informatiques : OS, programmation, logique, multimédia, intelligence artificielle, réseaux.
- ◆ Intérêt grandissant pour les technologies des BD :
 - ◆ Industrie : data mining, data warehouse, customer relationship mgmt, etc..
 - ◆ Scientifique : génôme, digital libraries et le web, etc..
 - ◆ Base de données multimédia, base de données et XML (NXD).
 - ◆ OLTP vs OLAP; OLAP et Web

Base de données (BD)

- ♦ Une base de données (*database*) est un ensemble structuré et cohérent de données enregistrées avec le minimum de redondance pour satisfaire simultanément plusieurs utilisateurs de manière sélective et dans un temps opportun.

Systeme de gestion de base de données (SGBD)

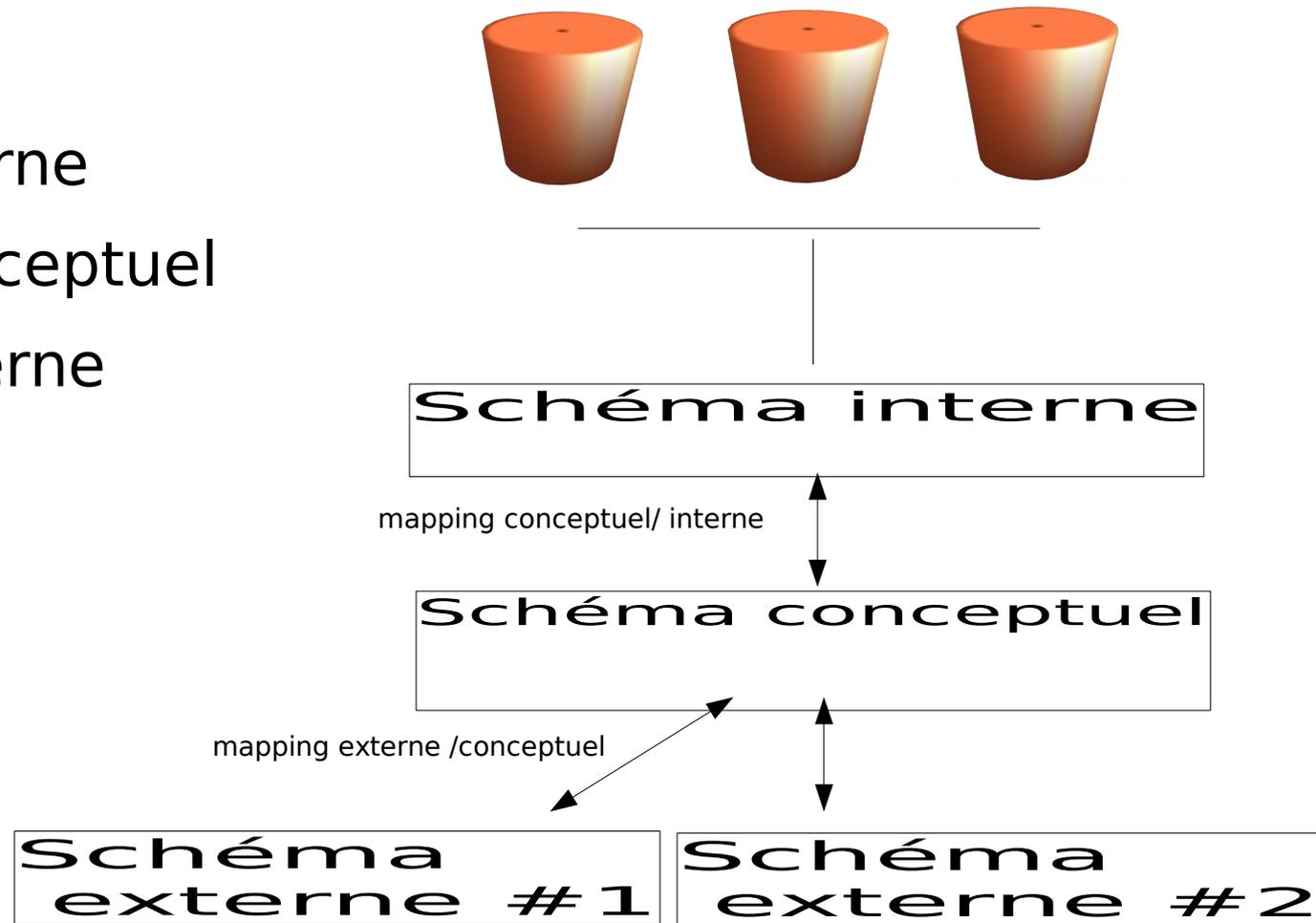
- ♦ Un ensemble de logiciels permettant la création et la maintenance de bases de données.
- ♦ Un SGBD (*database Management System -DBMS*) doit permettre :
 - ♦ la description, l'organisation et la gestion des données sur les mémoires secondaires.
 - ♦ la recherche, la sélection et la modification des données.
 - ♦ la manipulation des représentations abstraites des données, indépendamment de leur organisation et de leur implantation sur les supports physiques.

Caractéristiques des SGBD

- ♦ Structure (types et comportements des données).
- ♦ Persistence des données (mémoire secondaire).
- ♦ Accès aux données (langage déclaratif pour les requêtes, langage procédural de programmation).
- ♦ Performance (accès et stockage des données).
- ♦ Partage (accès concurrentiel et transactions)
- ♦ Volume important de données
- ♦ Sécurité et reprise sur pannes

Architecture ANSI /SPARC

- ◆ Trois niveaux :
 - ◆ Niveau interne
 - ◆ Niveau Conceptuel
 - ◆ Niveau externe



ANSI : American National Standards Institute

SPARC : Standard Planning and Requirements Committee

OlivierCuré [ocure@univ-mlv.fr]

Caractéristiques des SGBDR

- ♦ Un schéma est la structure d'une instance d'une base de données.
- ♦ On dispose de 3 schémas :
 - ♦ Un schéma physique : décrit comme la base de données est stockée sur les disques.
 - ♦ Un schéma conceptuel décrit les données se trouvant dans la base de données.
 - ♦ Un schéma externe spécifiant comme un utilisateur donné voit les données.
- ♦ Le catalogue de la BD contient les descriptions des schémas physique et conceptuel.

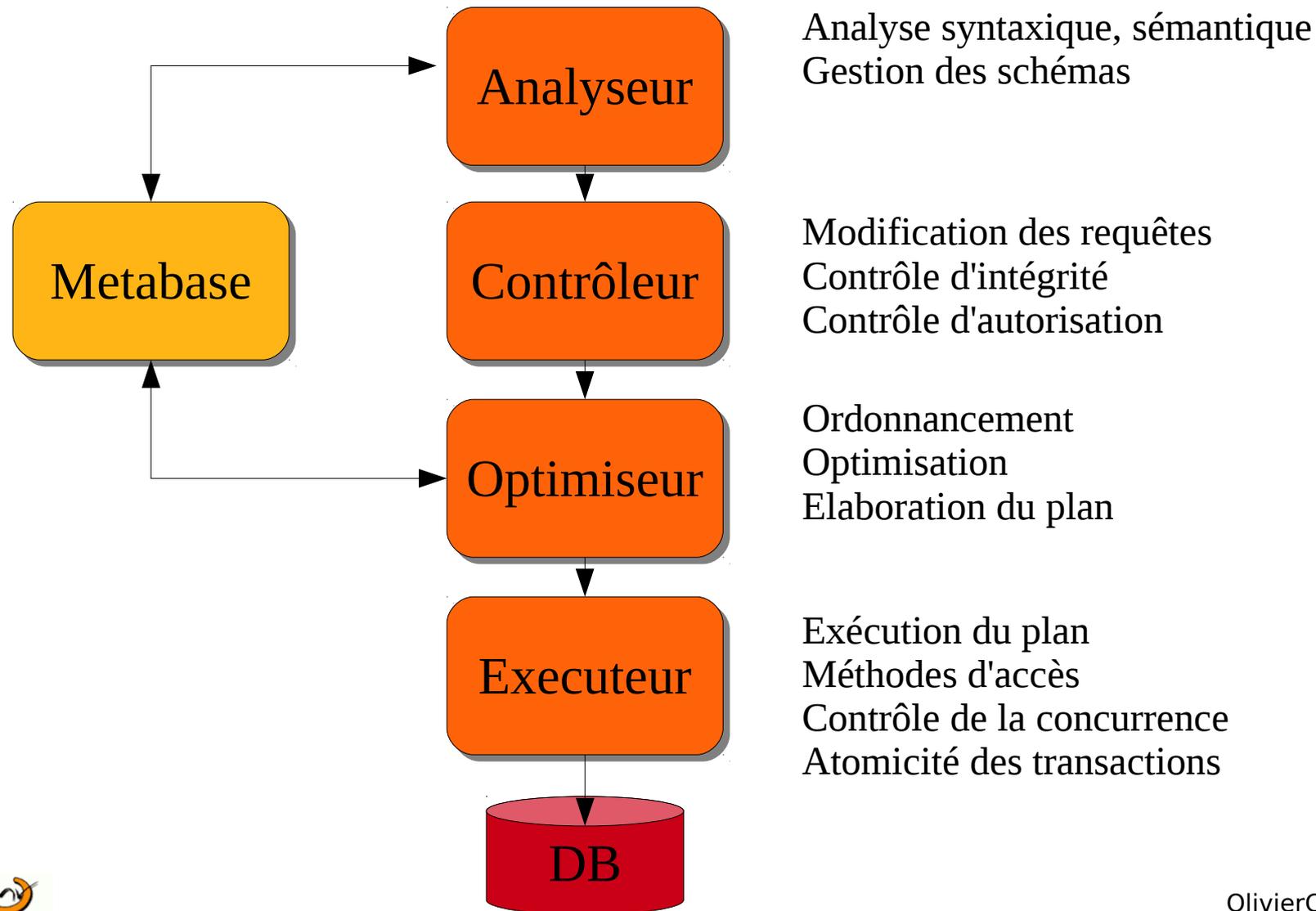
Caractéristiques des SGBDR (2)

- ♦ Un SGBD support l'**indépendance physique des données** si les utilisateurs n'interagissent pas avec le système au niveau physique.
- ♦ Cela supporte 3 avantages :
 - ♦ Utilisation plus facile du système
 - ♦ Optimisation de requêtes
 - ♦ Isolation des changements au niveau physique : la modification de l'organisation physique des données ne doit pas entrainer de modifications dans les programmes accédant aux données

Caractéristiques des SGBDR (3)

- ♦ Un SGBD support l'**indépendance logique des données** si les utilisateurs interagissent avec leur propre schéma externe.
- ♦ Cela supporte 3 avantages :
 - ♦ Customisation de schémas externes,
 - ♦ Meilleure sécurité au niveau des données,
 - ♦ Isolation des changements au niveau conceptuel : une modification du schéma conceptuel ne doit pas entraîner la modification des programmes.

Architecture fonctionnelle d'un SGBD



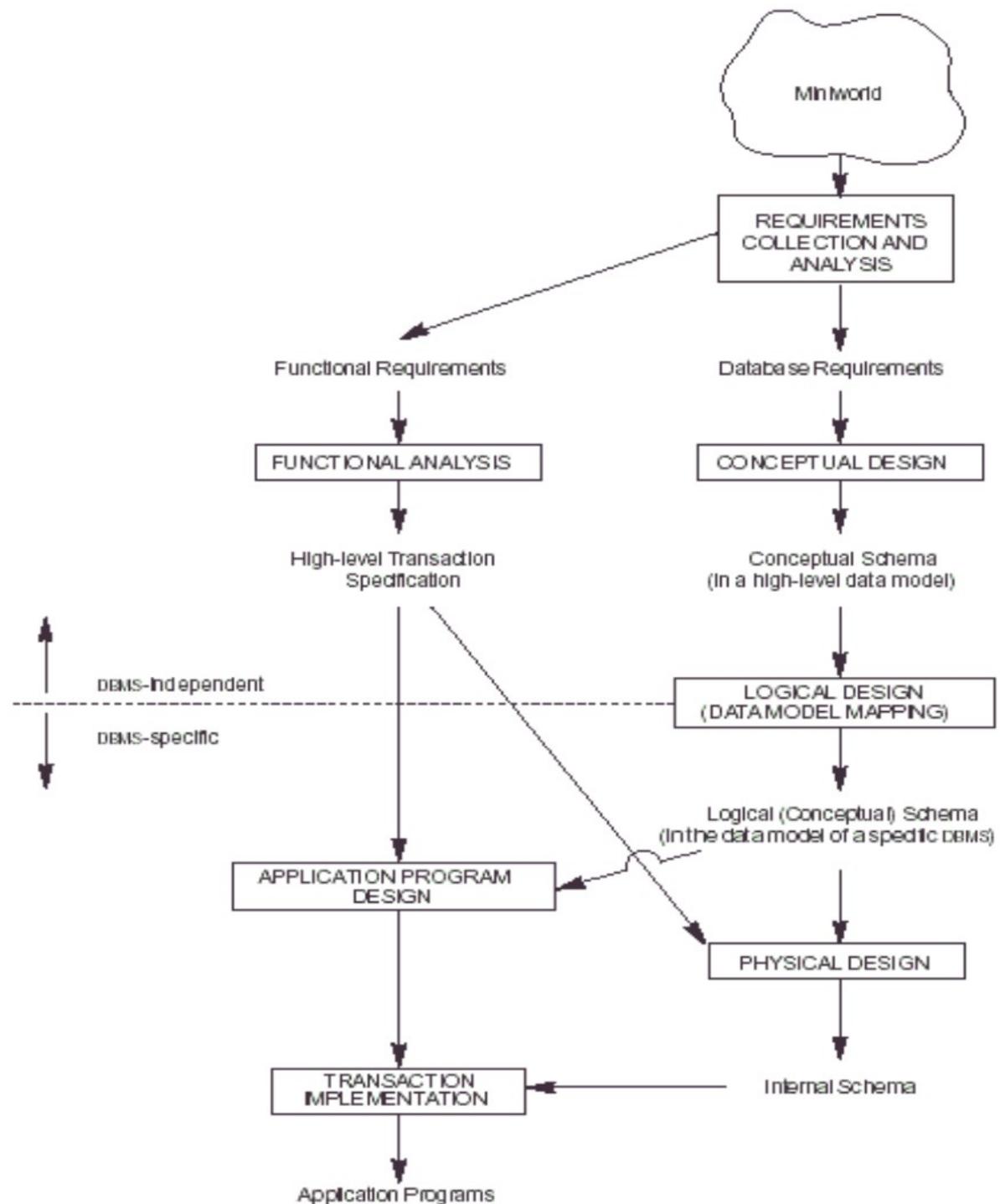
2. Modèle entité- association

Etape de développement d'une base de données

- ♦ Cahier des charges (besoin des utilisateurs, que doit gérer la bd).
- ♦ Schéma conceptuel (le plus souvent dans le modèle EA).
- ♦ Schéma logique (traduire le modèle EA dans le modèle de la base de données).
- ♦ Ajustement de schéma (normalisation, consistance).
- ♦ Schéma physique (indexes, organisation sur le disque).
- ♦ Sécurisation

Etape de développement d'une base de données (2)

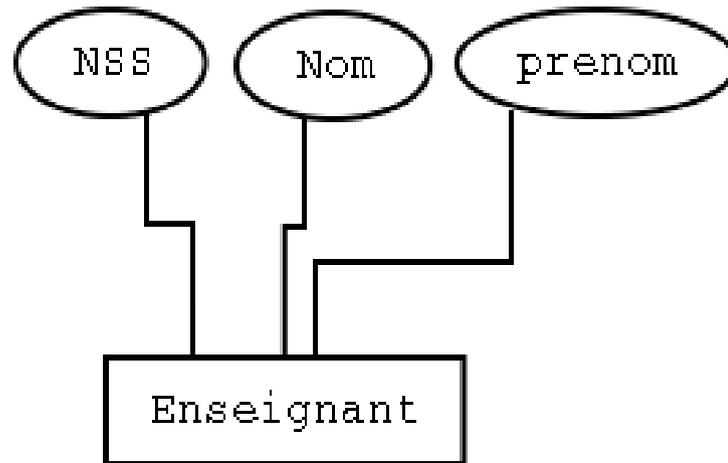
extrait de "Fundamentals of Database Systems 4th edition", Elmasri & Navathe, Addison-Wesley



Entité et propriété

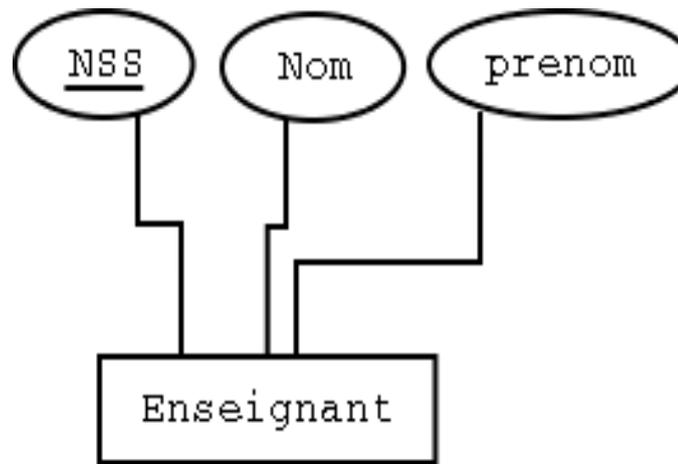
- ♦ Une entité est similaire à la notion d'objet. Elle décrit une « chose » du monde.
- ♦ Un type d'entité est similaire à la notion de classe, elle définit la structure commune à un ensemble d'entités. Exemple : les étudiants.
- ♦ Une propriété correspond à un état. Les valeurs des propriétés décrivent une entité.
- ♦ Représentation graphique dans un diagramme EA :
 - ♦ Une entité dans un rectangle
 - ♦ Une propriété dans une forme ovale.

Diagramme EA



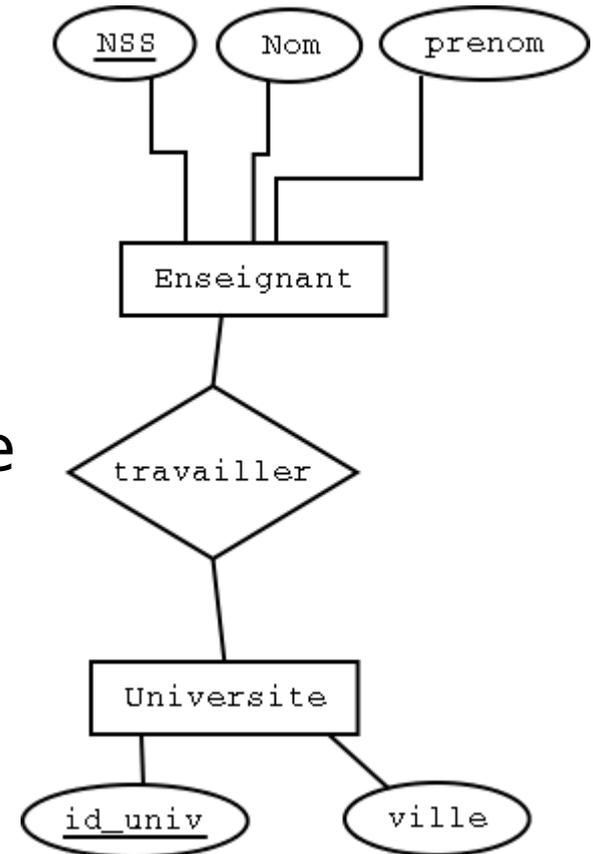
Propriétés identifiantes

- ❖ C'est l'ensemble des propriétés qui ne peuvent appartenir à plusieurs entités.
- ❖ La valeur doit être fixée pour chaque entité.
- ❖ Cette propriété est soulignée dans le diagramme EA.

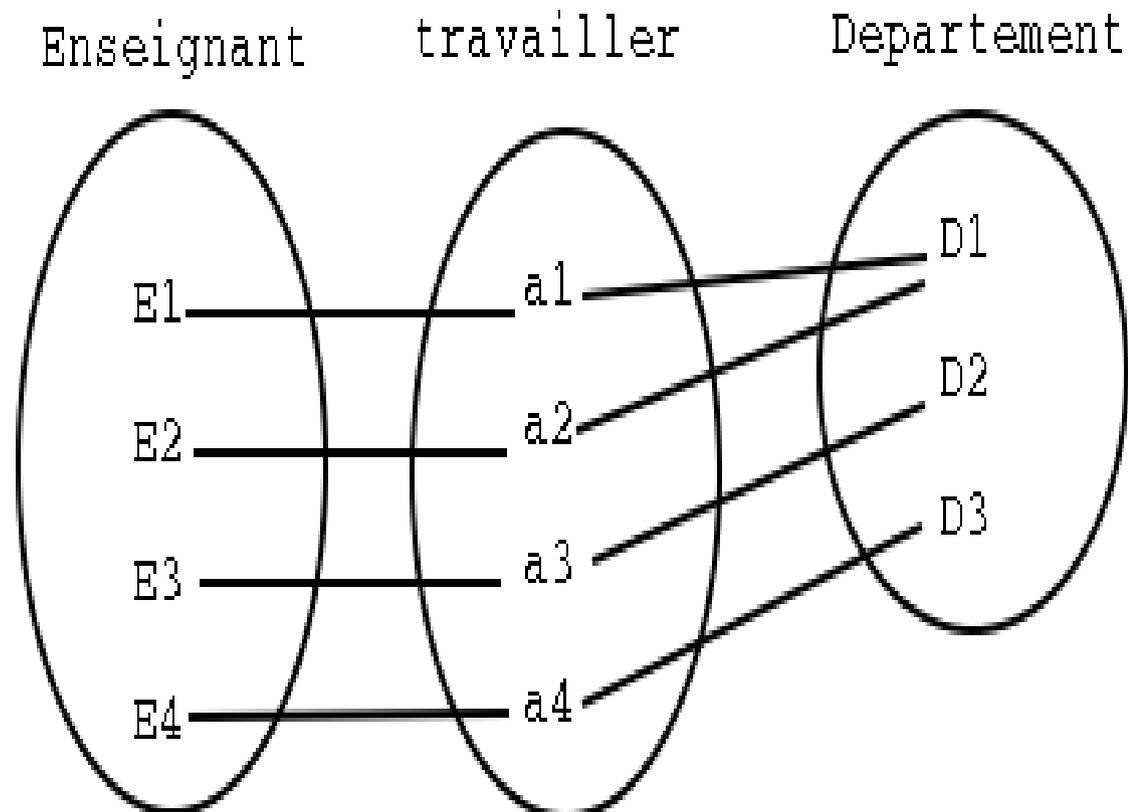


Association

- ❖ Une association est un lien entre deux ou plusieurs entités.
- ❖ Un type d'association est une collection d'associations similaires. Un type d'association A va lier des types d'entités E_1, \dots, E_n de manière que chaque association dans A implique les entités $e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n$.
- ❖ Représentation sous la forme d'un losange.

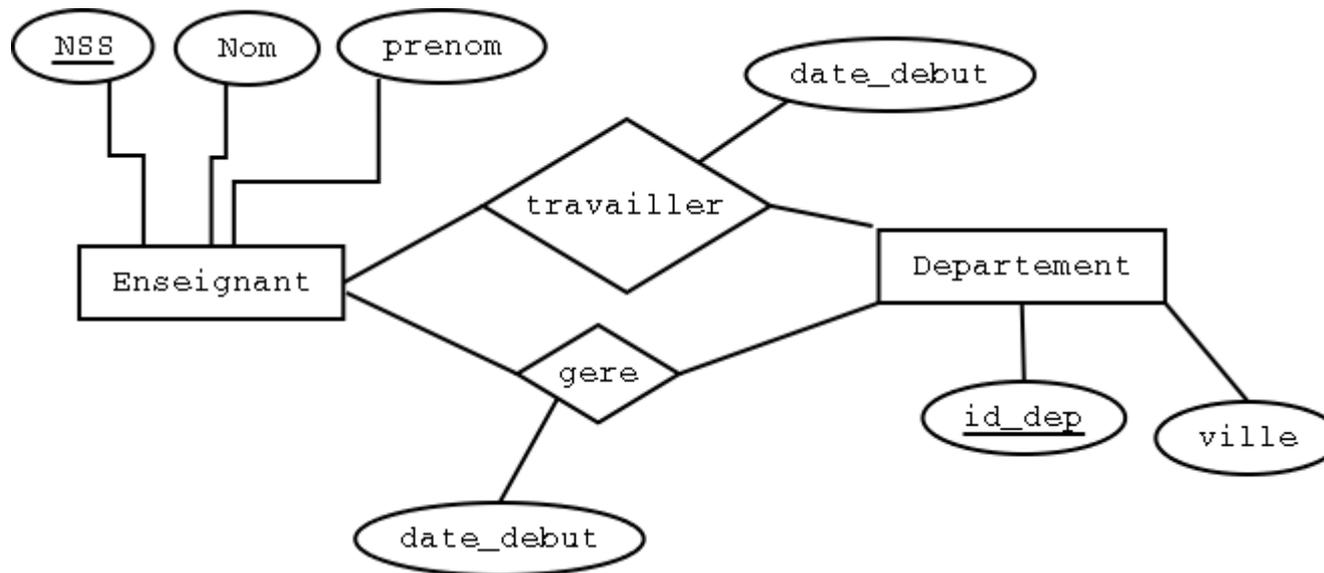


Type d'entité et type d'association



Association(2)

- Un type d'association peut avoir ses propres attributs.

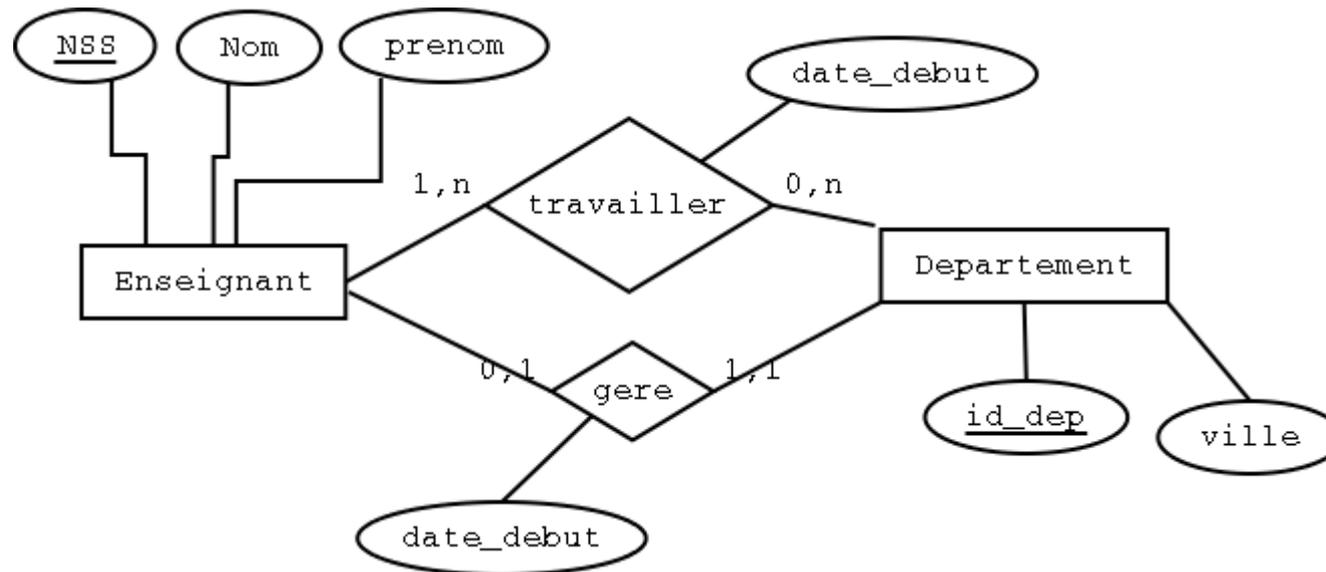


Cardinalité

- ◆ Les cardinalités permettent de mettre en évidence, le nombre minimum (maximum) d'instances d'un type d'entité qui sont en relation avec une instance d'un autre type d'entité.
- ◆ Les cardinalités sont représentées sous la forme d'un couple de valeurs min, max.
- ◆ $\min \in \{0,1\}$ et $\max \in \{1, n\}$

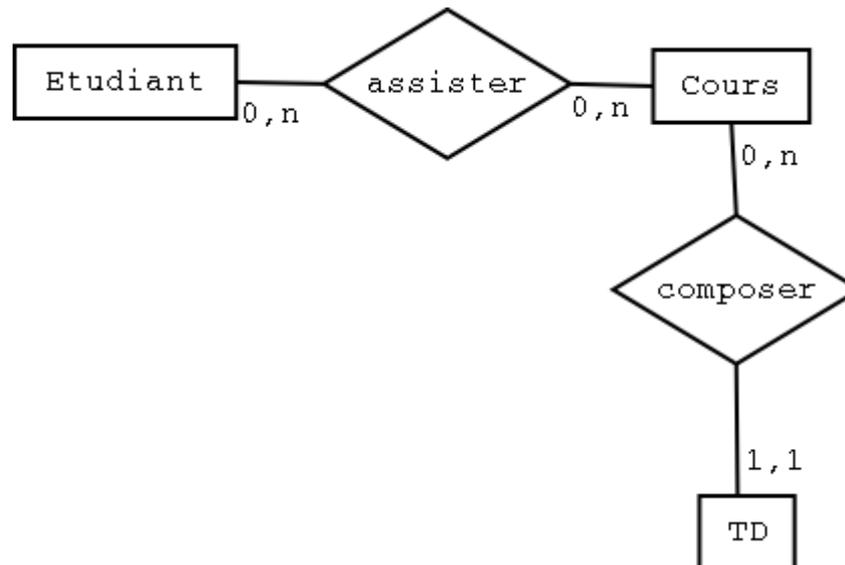
Diagramme ER avec cardinalités

- Un type d'association peut avoir ses propres attributs.



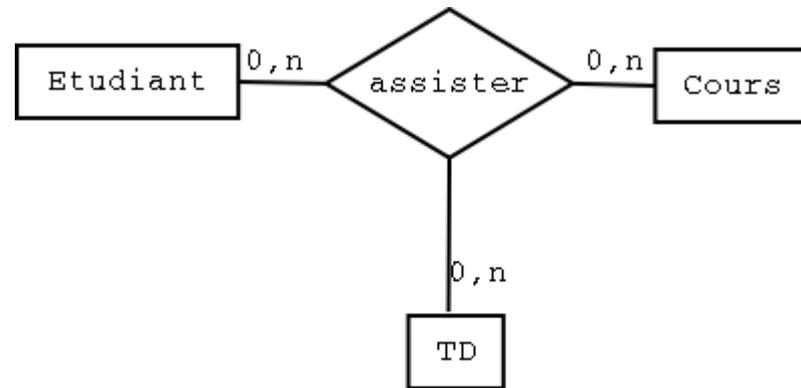
Association ternaire

- Si chaque élève participe à toutes les séances de TD :

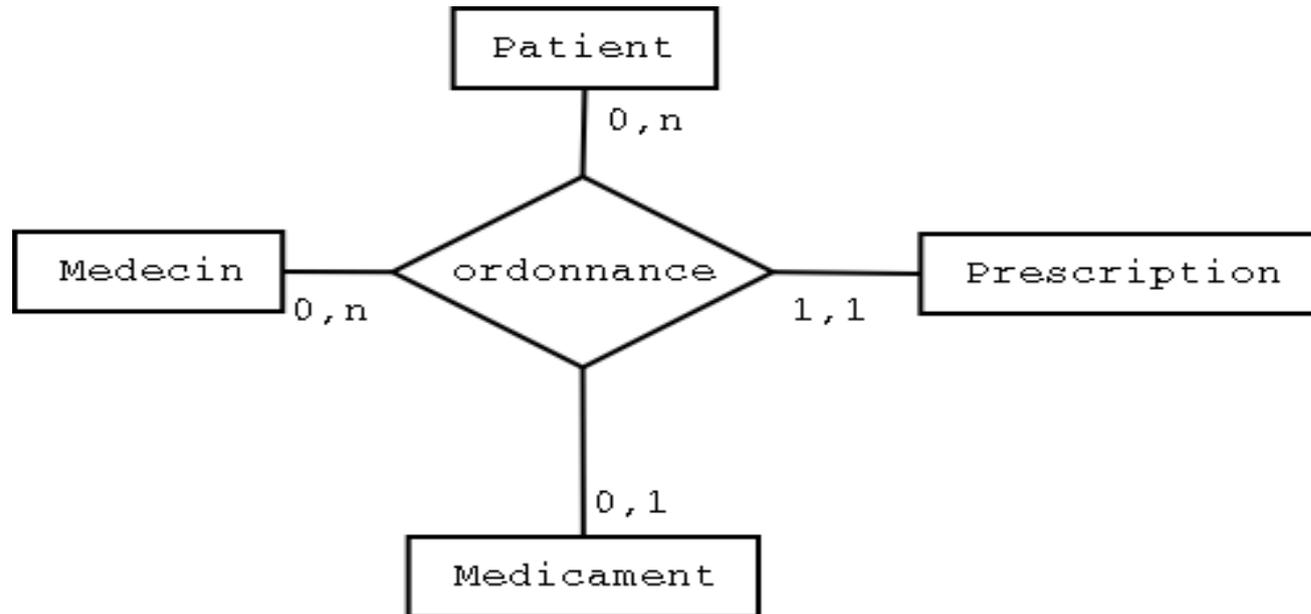


Comment faire lorsque les étudiants ne sont pas tenus de suivre l'intégralité des TD des cours.

Association ternaire (2)



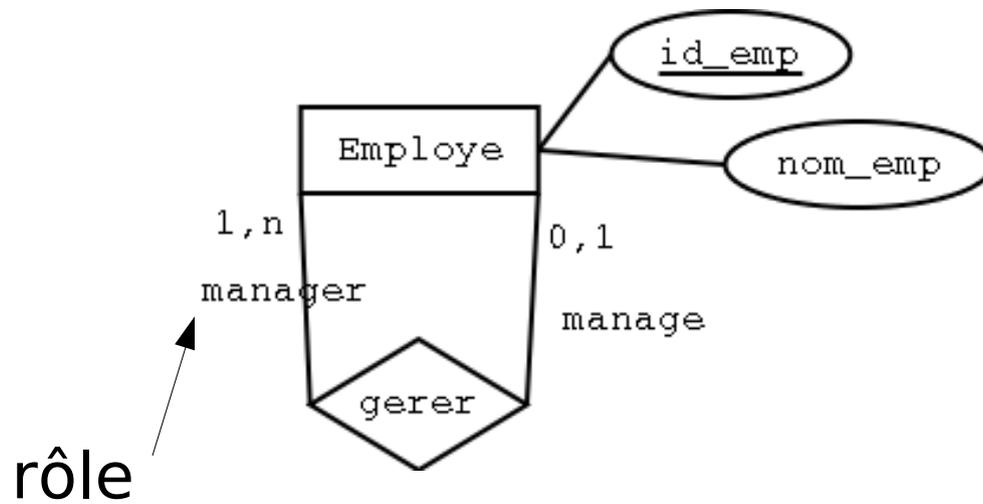
Association n-aire



Un médecin (ou un patient) particulier peut ne pas participer à la relation. Chaque instance de Prescription apparaît une fois et une fois seulement dans une instance de la relation « ordonnance ».

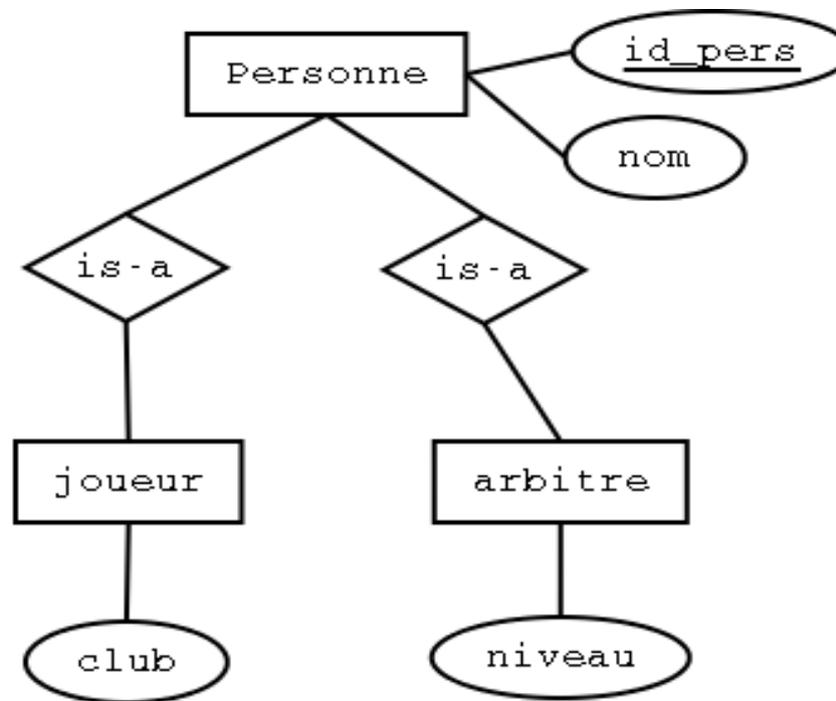
Une instance de médicament n'apparaît au plus qu'une fois par ordonnance.

Association récursive



Association et héritage

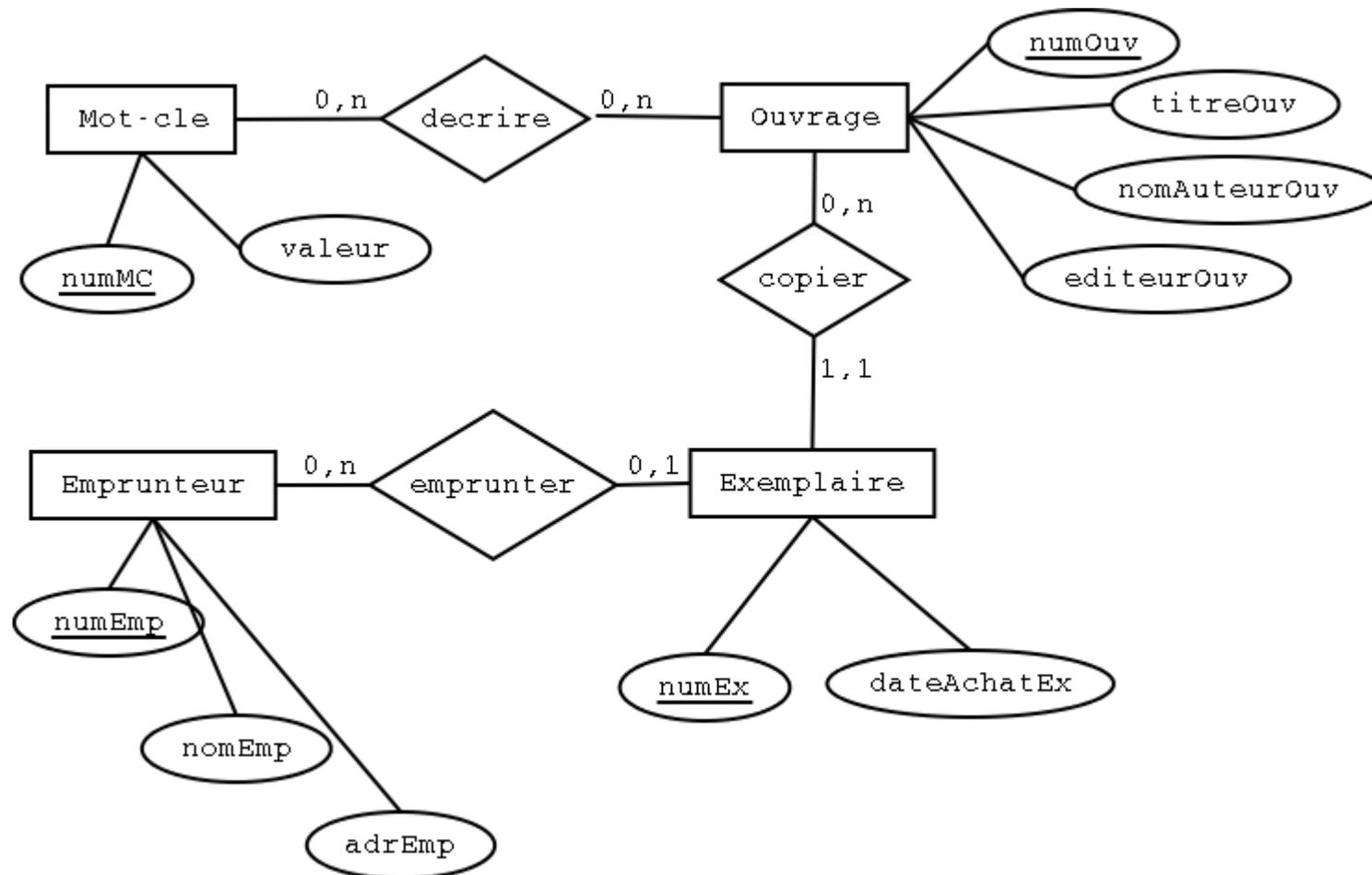
- On définit des hiérarchies au niveau des types d'entité. On utilise alors une association du type « is-a ».



Un exemple

- ◆ Une bibliothèque et sa gestion des emprunts :
 - ◆ Un ouvrage est caractérisé par un numéro unique, un titre, un auteur et un éditeur. Un ouvrage est décrit par un ensemble de mots qui indiquent les sujets qui sont traités.
 - ◆ Le bibliothèque dispose d'un ou plusieurs exemplaires de chaque ouvrage. Un exemplaire est identifié par un numéro et caractérisé par une date d'achat.
 - ◆ Un emprunteur est identifié par un numéro , son nom et son adresse.

Diagramme EA de la gestion d'une bibliothèque



Bilan du modèle EA

- ♦ Avantages :
 - ♦ Uniquement 3 concepts : entité, association et attribut.
 - ♦ Représentation graphique intuitive et rapide.
- ♦ Inconvénients :
 - ♦ Non-déterministe, pas de règle absolue pour déterminer ce qui est entité, attribut ou association.
 - ♦ Absences de contraintes d'intégrité, structures complexes.

3.

Le modèle relationnel

Modèle de données (*data model*)

- ◆ Un ensemble de concepts permettant de décrire la structure d'une base de données :
 - ◆ types de données,
 - ◆ relations,
 - ◆ contraintes,
 - ◆ opérations (langage de définition et de manipulation des données).
 - ◆ sémantique.

Quelques modèles de données

- ◆ Evolution dans le temps :

- ◆ système de fichiers (50s)

- ◆ modèle hiérarchique(60s) :IMS

- ◆ modèle réseau (70s): Codasyl, IDMS

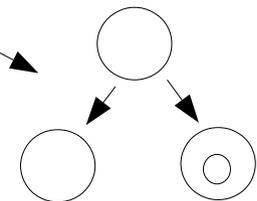
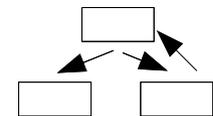
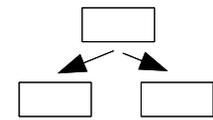
- ◆ modèle relationnel (80s): Oracle, Sybase,DB2,..

- ◆ modèle orienté objet (90s): Versant, Gemstone, O2

- ◆ modèle objet-relationnel (90s) : Oracle, DB2, SQLServer, PostgreSQL

- ◆ modèle semi-structuré (hétérogénéité, XML).

- ◆ Le modèle relationnel est le plus populaire aujourd'hui.



Systeme de fichiers

- ◆ Jusqu'au milieu des années 60.
- ◆ Problèmes :
 - ◆ Redondance des données
 - ◆ Problèmes lors de la mise-à-jour.
 - ◆ Pas de modèle abstrait des données
 - ◆ Nécessite une connaissance sur l'organisation du stockage
 - ◆ Il n'existe pas un langage de requêtes standard.

Bases de données fondées sur le modèle hiérarchique

- ◆ Développement par North American Rockwell et IBM : IMS (Information Management System)
- ◆ Basé sur une structure arborescente
- ◆ Problèmes
 - ◆ Les changements dans les structures de données (de la BD) imposent des modifications dans les programmes qui accèdent aux données.
 - ◆ Familiarisation des programmeurs avec la structure de la base de données.

Bases de données fondées sur le modèle réseau

- ♦ Une extension du modèle de données du modèle hiérarchique.
- ♦ Standardisation en 1971 par le groupe CODASYL (COncference on DAta SYstems Languages).
- ♦ Avantage :
 - ♦ Implémentation des relations n:m.
- ♦ Problème :
 - ♦ La navigation de la structure est encore plus complexe que dans le modèle hiérarchique.

Bases de données fondées sur le modèle relationnel

- ♦ Proposé par E.F. Codd qui travaillait chez IBM
- ♦ IBM se concentre sur IMS et ignore plus ou moins le modèle réseau.
- ♦ IBM lance l'implémentation d'un SGBD sur le modèle relationnel.
- ♦ Entre temps, J.L. Ellison, à partir des articles de Codd, crée une start-up qui va vendre des SGBD basées sur le modèle relationnel : Oracle.

Modèle relationnel

- ◆ Basé sur la notion d'ensemble (*set*)
 - ◆ collection d'élément de même type
 - ◆ pas d'ordre pour les enregistrements
 - ◆ ordonnancement des attributs dans une relation
 - ◆ pas de duplications des enregistrements
- ◆ On parle alors de SGBDR (*RDBMS*)

Domaine de valeurs

- ♦ Un domaine de valeurs correspond à un ensemble de valeurs. Par exemple les entiers, les réels, etc..
- ♦ Le produit cartésien (\times) de n domaines est l'ensemble de toutes les combinaisons des éléments de ces ensembles.
- ♦ Exemple : $D1 = \{ a, b \}$ et $D2 = \{ a, e \}$
- ♦ $D1 \times D2 = \{ (a,a), (a,e), (b,a), (b,e) \}$

Relation

- ◆ Une relation est un sous-ensemble du produit cartésien d'une liste de domaines.
- ◆ Chaque relation est identifiée par un nom.
- ◆ Représentation sous la forme d'une table à 2 dimensions :
 - ◆ des colonnes, attributs.
 - ◆ des lignes, enregistrements ou n-uplets.

Attribut

- ◆ Un attribut prend ses valeurs dans domaine de valeurs. Un attribut est identifié par un nom.
- ◆ Exemple :

titre	année	nomDir	prénomDir	annéeNaissDir
Alien	1979	Scott	Ridley	1943
Reservoir Dogs	1992	Tarantino	Quentin	1963
The thing	1982	Carpenter	John	1948
Volte-face	1997	Woo	John	1946
Pulp Fiction	1995	Tarantino	Quentin	1963

Clés

- ❖ Clé : un ensemble d'attributs dont le regroupement des valeurs identifie sans ambiguïté un n-uplet (*tuple*).
- ❖ Clé candidate (*candidate key*): une clé pour laquelle aucun sous-ensemble n'est une clé. Une clé minimale.
- ❖ Clé primaire (*primary key*): une clé candidate sélectionnée pour être la clé principale d'une relation. Une clé primaire par relation.

Clé externe

- ♦ Une clé externe (*foreign key*) est un ensemble d'attributs dans une relation qui correspondent exactement à la clé primaire d'une autre relation.
- ♦ Le nom des attributs ne doivent pas nécessairement correspondre mais les domaines doivent être identiques.

Un exemple

Une relation sur les films

titre	année	nomDir	prénomDir	annéeNaissDir
Alien	1979	Scott	Ridley	1943
Reservoir Dogs	1992	Tarantino	Quentin	1963
The thing	1982	Carpenter	John	1948
Volte-face	1997	Woo	John	1946
Pulp Fiction	1995	Tarantino	Quentin	1963
Terminator	1984	Cameron	James	1954
Ghosts of Mars	2001	Carpenter	John	1948
Mad Max	1979	Miller	George	1945
Mad Max 2	1981	Miller	George	1945

Quelques problèmes : Supprimer un film, ajouter un réalisateur, modifier la date de naissance d'un réalisateur, Cohérence des données.

Un exemple (2)

Une relation sur les films

Une autre relation sur les réalisateurs

titre	année	idDir	nomDir	prénomDir	annéeNaissDir
Alien	1979	1	Scott	Ridley	1943
Reservoir Dogs	1992	2	Tarantino	Quentin	1963
The thing	1982	3	Carpenter	John	1948
Volte-face	1997	4	Woo	John	1946
Pulp Fiction	1995	2	Cameron	James	1954
Terminator	1984	5	Miller	George	1945
Ghosts of Mars	2001	3			
Mad Max	1979	6			
Mad Max 2	1981	6			

Problèmes précédents : Supprimer un film, ajouter un réalisateur, modifier la date de naissance d'un réalisateur, Cohérence des données.

Schéma d'une relation

- ♦ Le schéma d'une relation correspond à l'ensemble : nom de la relation, liste des attributs (avec domaines) et indication de la clé primaire (souligné).
- ♦ $R : A_1, A_2, \dots, A_n$
- ♦ Schéma d'une base de données est l'ensemble des schémas des relations de la base.

Schémas, exemple

- ◆ Schéma des relations :

film (titre, année, idDir)

realisateur (idDir, nomDir, prenomDir, annéeNaisDir)

- ◆ Schéma de la base de données :

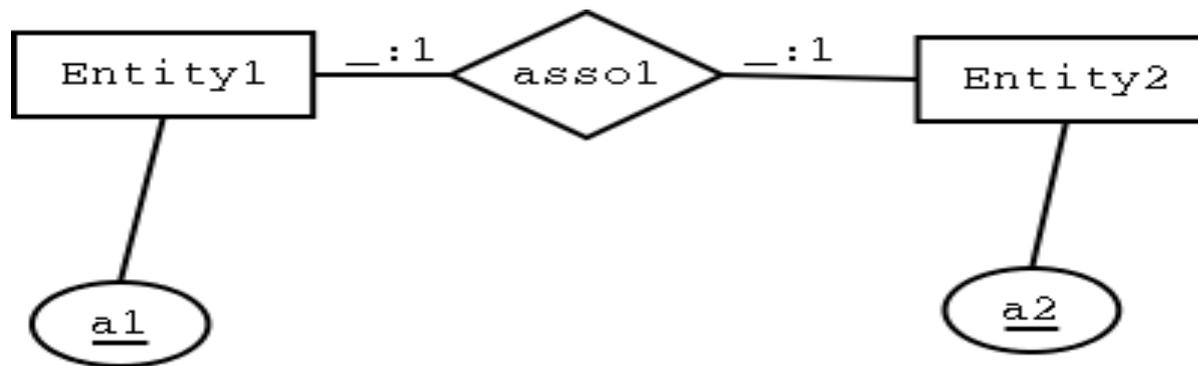
film, realisateur

Mapping EA - relationnel

- ◆ Une type d'entité devient une relation.
- ◆ Une propriété devient un attribut.
- ◆ Une propriété identifiante devient une clé primaire.
- ◆ Une association devient une relation et sa représentation est fonction des cardinalités.

Transformation des associations en relations

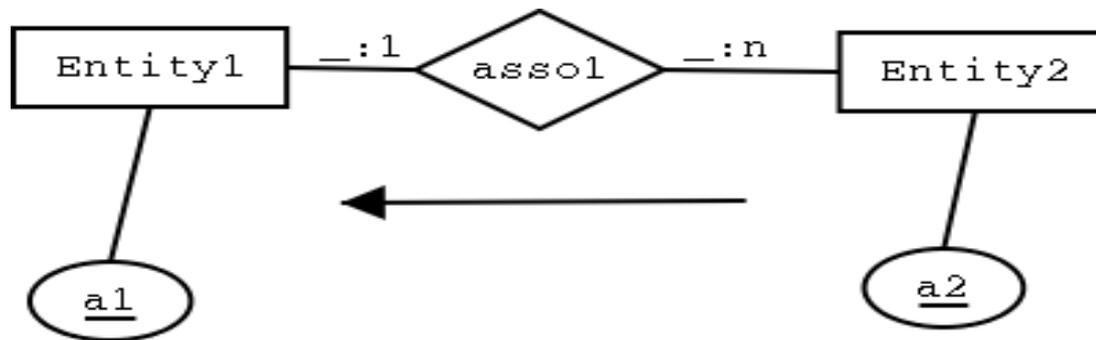
- ◆ Les cardinalités maximales de l'association sont de 1 :
- ◆ L'identifiant d'une entité est utilisé comme clé étrangère de l'autre relation, ou inversement.



Entity1(a1, ...)
Entity2(a2, ..., #a1)
ou
Entity1(a1, ..., #a2)
Entity2(a2, ...)

Transformation des associations en relations (2)

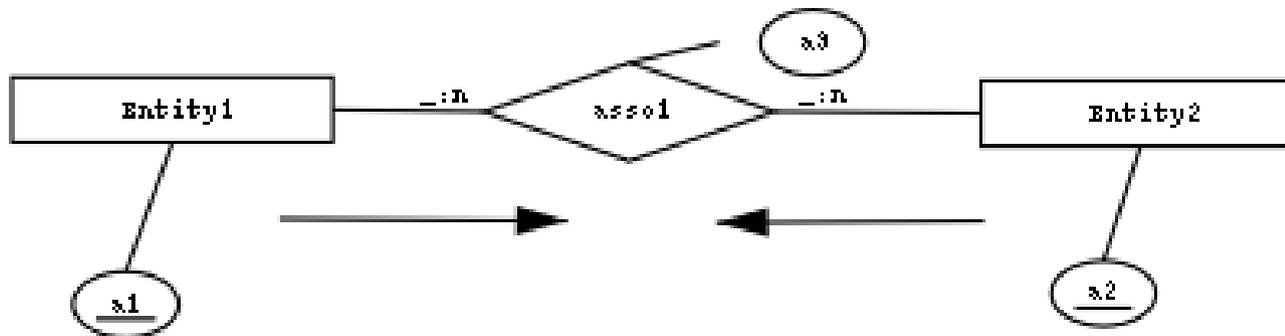
- ♦ La cardinalité maximale d'une association est à n et l'autre à 1 :
- ♦ L'identifiant de l'entité (*:n) devient une clé étrangère de l'entité (*:1).



Entity1(a1, ..., #a2)
Entity2(a2, ...)

Transformation des associations en relations (3)

- ◆ Les cardinalités maximales sont à n :
- ◆ L'association devient une relation et sa clé primaire se compose des identifiants des entités.

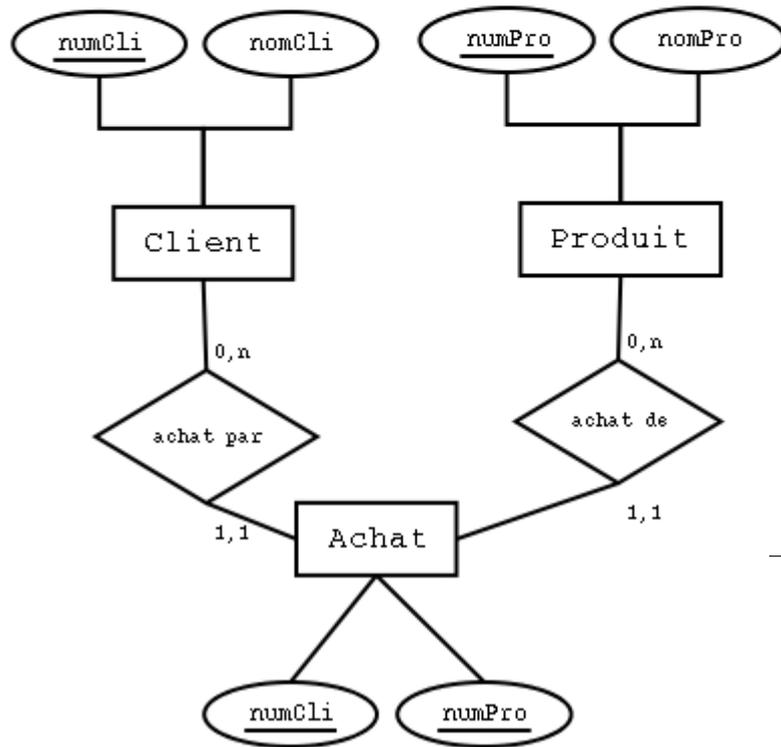
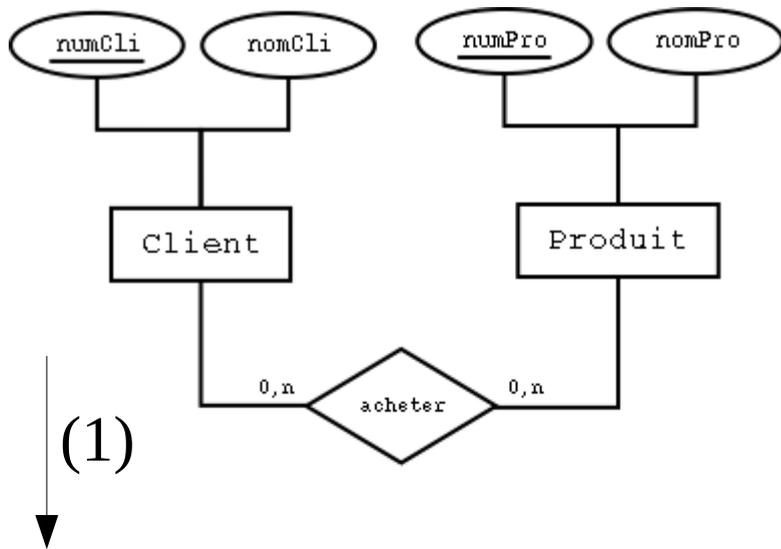


Entity1(a1, ...)

Entity2(a2, ...)

Ass01(#a1, #a2, a3)

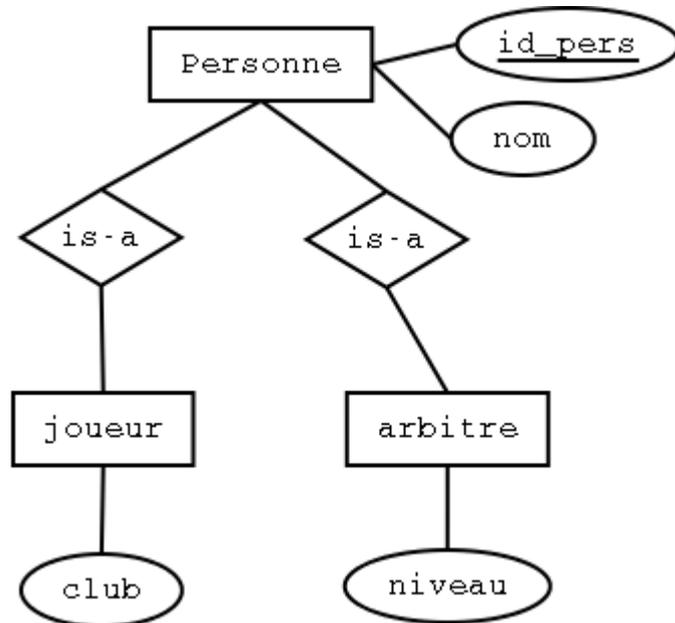
Exemple mapping relations n,m



- ◆ Schéma relationnel
Produit (numPro, nomPro)
Client (numCli, nomCli)
Achat (#numCli, #numPro)

Transformation des associations en relations (4)

- ◆ Dans le cas d'une association « is-a », il est possible de voir la solution avec un point de vue orienté objet (duplication des attributs de la super-classe dans la sous-classe) ou bien entité-association (duplication de l'identifiant de la super-classe dans la sous-classe).



Personne(id_pers, nom)

Vision OO :

joueur(id_pers, club)

arbitre(id_pers, niveau)

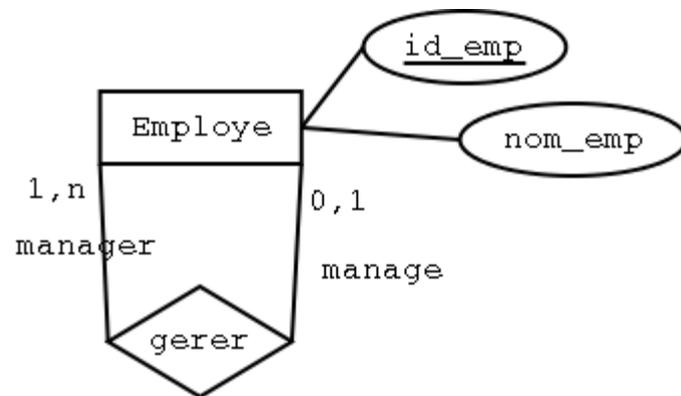
Vision EA :

joueur(id_pers, nom, club)

arbitre(id_pers, nom, niveau)

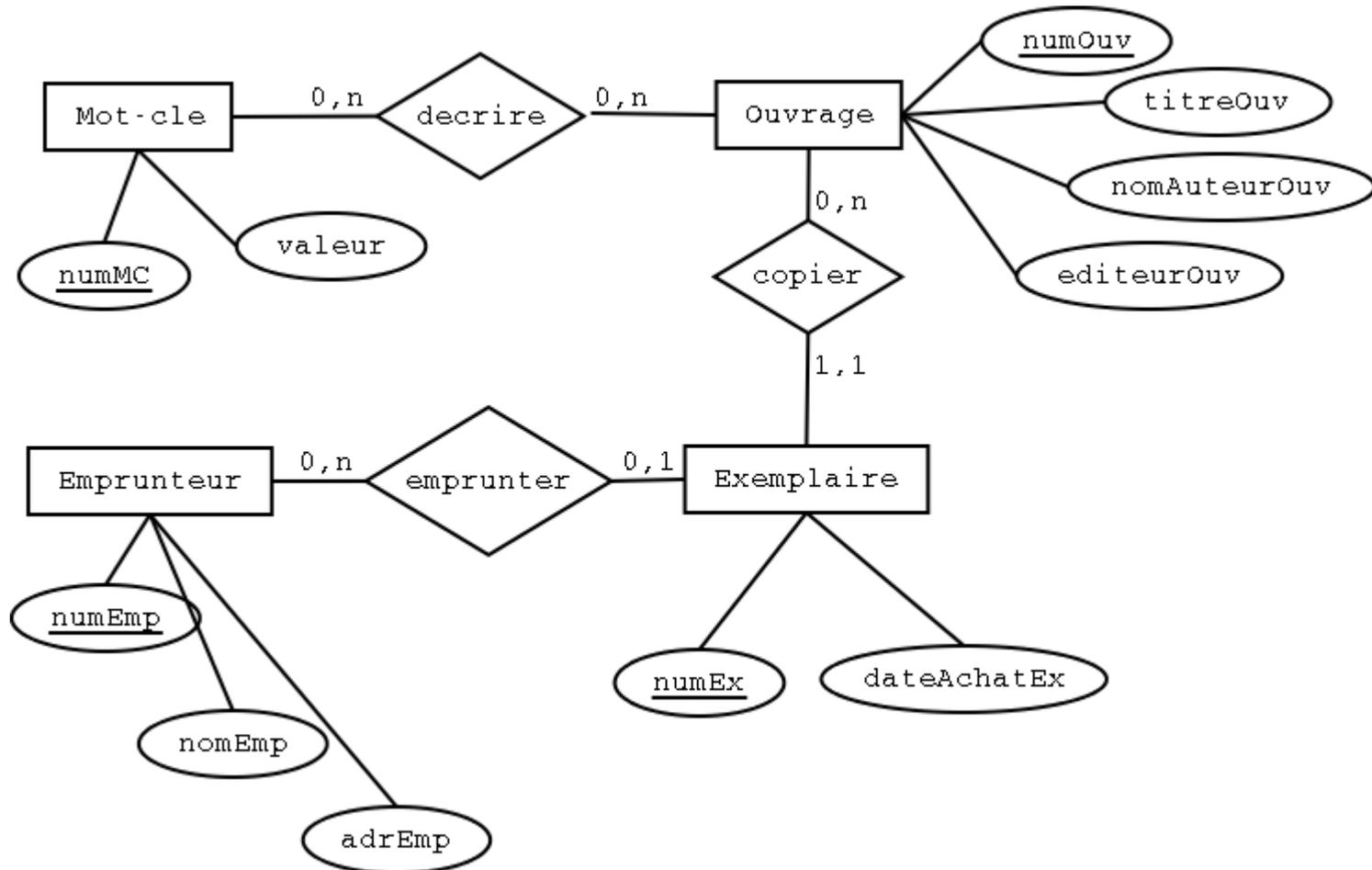
Transformation des associations en relations (5)

- ◆ Dans le cas d'une association récursive, l'attribut clé primaire de la relation est dupliquée sous la forme d'une clé étrangère.



Employe (id_emp, nom_emp, #id_emp_manager)

Exemple Bibliothèque



Schémas de la bibliothèque

♦ Schéma relationnel

Mot_cle (numMC , valeurMC)

Ouvrage (numOuv, titreOuv, nomAuteurOuv, editeurOuv)

Exemplaire (numEx, dateAchat, #numEmp, #numOuv)

Emprunteur (numEmp, nomEmp, adrEmp)

MotCle_Ouvrage (#numMC, #numOuv)

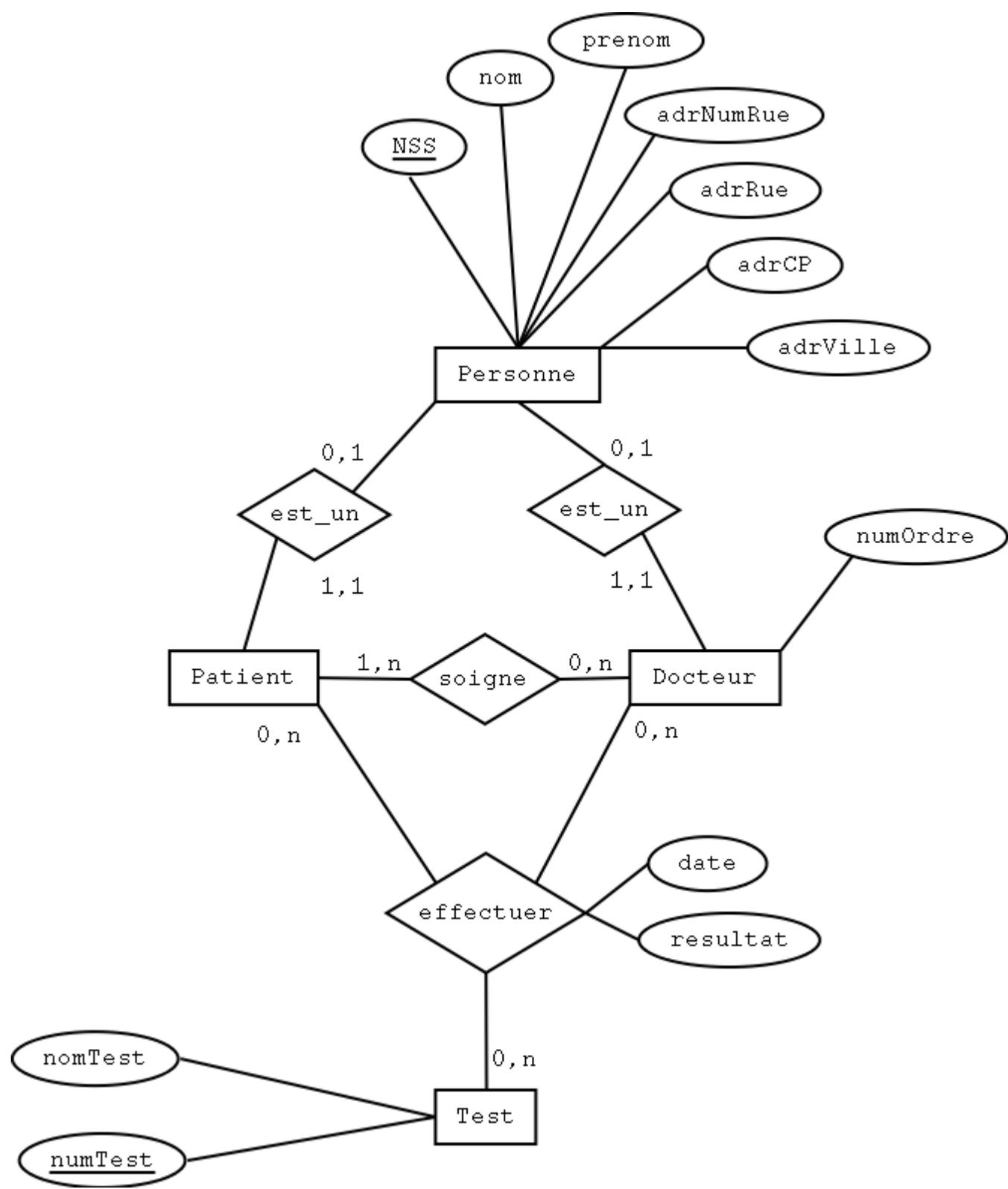
♦ Schéma de la base de données

Mot_cle, Ouvrage, Exemplaire, Emprunteur,
MotCle_Ouvrage

Autre exemple

- ◆ On gère des patients et des docteurs qui sont des personnes. Pour chaque personne, on stocke le NSS, le nom, le prénom, l'adresse. Pour un docteur nous avons en plus un numéro d'ordre.
- ◆ Un docteur soigne un patient et peut lui prescrire un test (portant un nom et identifié par un numéro).
- ◆ On désire stocker la date et le résultat d'un test donné pour un patient et prescrit par un docteur.

Correction exemple



Son schéma relationnel

Personne(NSS, nom, prenom, adrNumRue, adrRue, adrVille)

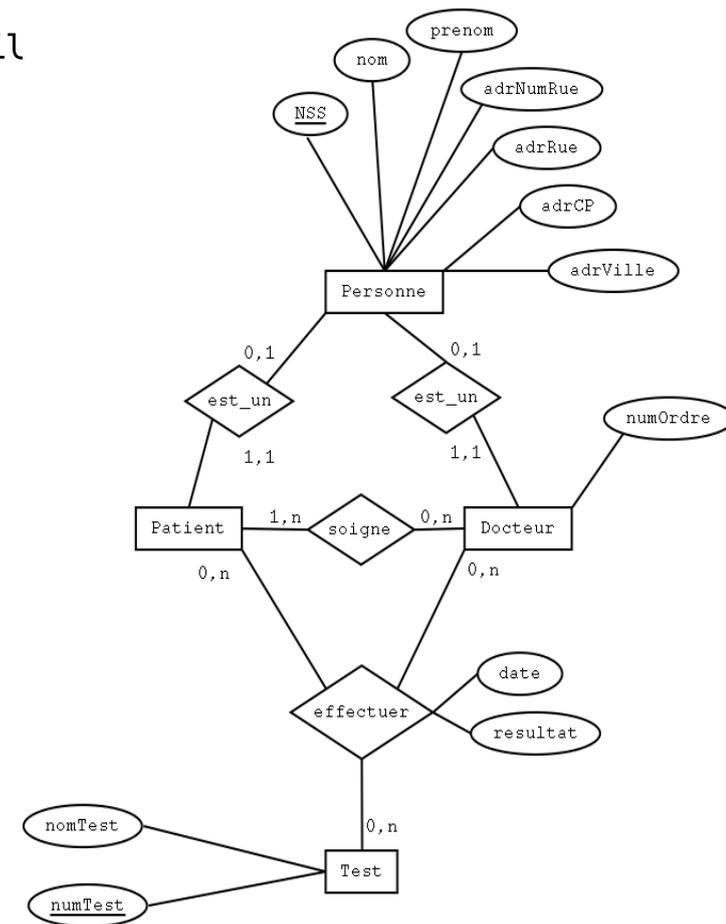
Patient (NSS)

Docteur(NSS, numOrdre)

Test (numTest, nomTest)

Soigne (#NSSPatient, #NSSDocteur)

EffectuerTest (#NSSDocteur, #NSSPatient, #NumTest, date, resultat)



4. Algèbre relationnel