

Pôle informatique 2013/2014

École Nationale Supérieure des Mines de Saint-Étienne

Conception de Systèmes d'Information

Antoine Zimmermann

`antoine.zimmermann@emse.fr`

(adapté du cours de Laurent Vercouter)

Contenu du cours

• Partie « Base de Données »

- Novembre/Décembre 2013
- Objectif : savoir concevoir un modèle de BDD et l'implanter avec un langage de manipulation de données
- Outils : modèle E-A, relationnel, SQL, OpenOffice Base, PHP

• Partie « Développement d'un SI »

- Janvier/Février 2013
- Objectif : savoir appliquer une méthode de développement d'un SI

Evaluation = moyenne de deux notes

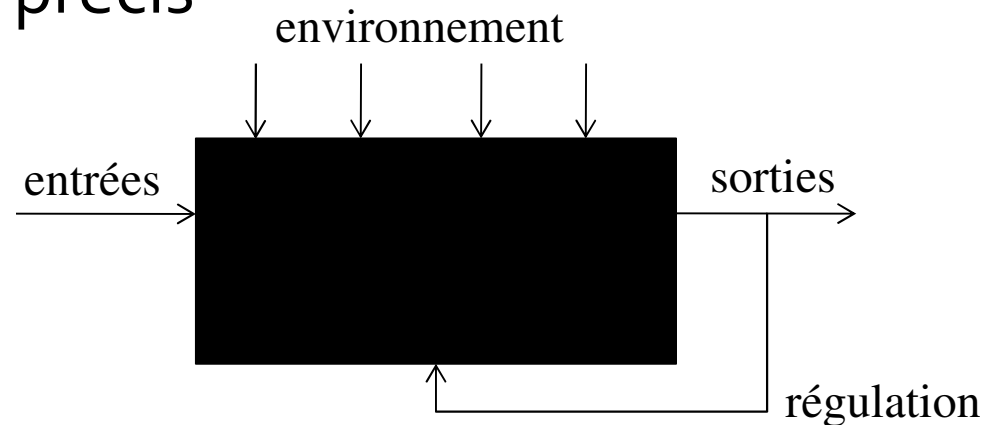
- examen écrit sur les aspects théoriques (individuel)
- mise en œuvre pratique dans un mini-projet (en double binôme)

Plan

- **Introduction sur les Systèmes d'Information**
 - ◆ **Notions de base**
 - ◆ Mise en œuvre technique
- **Le Modèle Entité-Association**
 - ◆ Concepts
 - ◆ Exemple
- **Le modèle relationnel**
 - ◆ Concepts
 - ◆ Algèbre relationnelle

Qu'est ce qu'un Système d'Information

- Système = ensemble auto-réglable et interagissant avec l'environnement qui fonctionne en vue d'un objectif précis



- Information = donnée enrichie d'un modèle d'interprétation

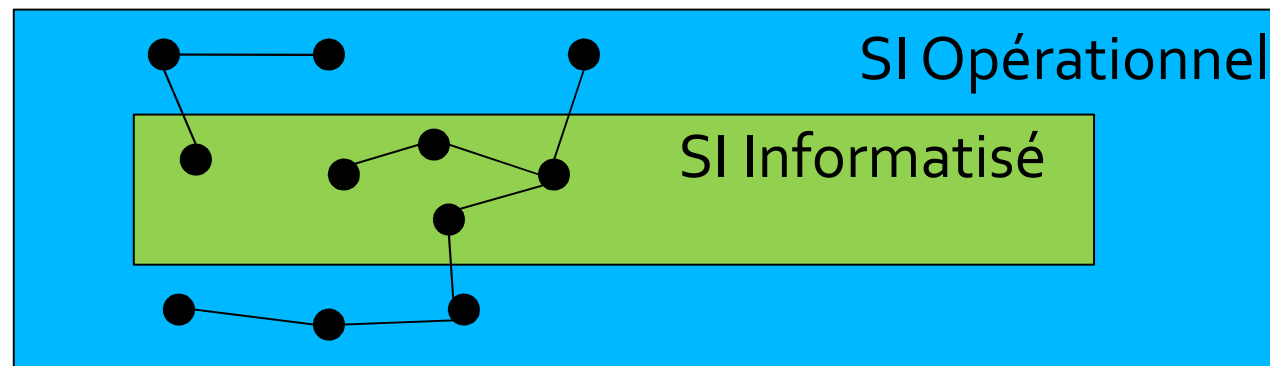
Fonctions du SI

Un SI a deux fonctions principales :

- La production d'information
 - ◆ Collecter des informations
 - ◆ Traiter et transmettre des informations
 - ◆ Mémoriser des informations
- La mise en œuvre d'outils de gestion
 - ◆ Fonction technologiques (matériels, logiciels, méthodes, savoir-faire, ...)
 - ◆ Fonction économique
 - ◆ Fonction sociale

Informatique et SI

L'informatique facilite la gestion d'un SI mais ne le couvre pas dans son ensemble.



- SIO – Système d'Information Opérationnel = toute l'activité autour du SI
- SII – Système d'Information Informatisé = uniquement le contenu informatisé (fichiers, bases, logiciels, ...)

SI sans Informatique

- Les bibliothèques et administrations utilisaient des SI bien avant l'invention de l'ordinateur
- Un SI peut être mis en œuvre à l'aide de cahiers, registres, indexes, etc
- **mais** un SI sans informatique a peu d'intérêt car les tâches ne peuvent être automatisées (le SI doit être exploité par un opérateur humain)

Communication entre SI

- Les SI d'entreprises partenaires (associées, filiales, client/fournisseur) doivent communiquer
 - par des moyens « classiques » (courrier, ...)
 - par des outils informatiques : EDI (Echange de Données Informatisées)
- Le(s) Système(s) d'Information peut atteindre une taille et une envergure gigantesque
 - Facebook, > 1 000 000 000 de comptes, des **centaines de milliards** de mises à jours depuis sa création
 - Google > 100 000 000 000 de documents indexés, on estime à plusieurs **milliers de milliards** de requêtes depuis sa création

Mon projet C est-il un SI ?

- En un sens oui... mais très limité.
- Un SI exige une continuité de l'information (l'information n'est pas perdue quand on quitte l'application)
- Un SI implique généralement de plusieurs applications exploitant la même information
- Exemple du transport ferroviaire

Transport ferroviaire

- La SNCF a besoin de gérer l'information sur les *gares*, les *voies*, les *trains*, les *wagons*, les *places*, les *classes*, les *horaires*, les *tarifs*, les *distances*, les *conducteurs*, les *contrôleurs*, les *réservations*, etc.
- Qui a besoin de l'information ?
 - ♦ **Guichetiers**, et par ce biais, le **client**
 - ♦ **Voyages-sncf.com**, et par ce biais, l'**internaute**
 - ♦ Les **écrans d'affichage** des trains à l'arrivée et au départ
 - ♦ Le système de **gestion d'aiguillage**
 - ♦ Le gestionnaire **comptable**
 - ♦ Le service de **maintenance**

Plan

- **Introduction sur les Systèmes d'Information**
 - ◆ Notions de base
 - ◆ **Mise en œuvre technique**
- Le Modèle Entité-Association
 - ◆ Concepts
 - ◆ Exemple
- Le modèle relationnel
 - ◆ Concepts
 - ◆ Algèbre relationnelle

Stockage des données

- L'informatisation du SI nécessite la mise en place de supports matériels et logiciels de stockage des données.
- L'information est numérisée et conservée sous la forme de fichiers.
- Le moyen d'organiser, d'accéder à et de manipuler ces fichiers est à définir

Système de Gestion de Fichiers

- Système utilisé pour gérer les données d'un disque (disquette, disque dur, CD-Rom, ...)
- Un fichier = une suite ordonnée de données
- 2 types d'accès aux données d'un fichier
 - ♦ Séquentiel (depuis l'adresse de début du fichier)
 - ♦ Accès direct (selon un décalage par rapport à l'adresse de début)
- Système de Gestion de Fichiers (SGF) : gère les accès aux fichiers, application par application
- Inconvénient :
 - ♦ Structure trop simple (applications indépendantes)
 - ♦ Redondance d'information

SGF : inconvénients par l'exemple

Exemple : *Comment stocker les relations d'amitié dans des fichiers?*

- 1 fichier pour toutes les relations
 - ♦ Problème d'efficacité (fichier très gros)
 - ♦ Problème d'accès (comment trouver les relations de Jules ou de Jim seulement)
 - ♦ Problème de droit (qui a accès à quoi ?)
- 1 (petit) fichier par utilisateur
 - ♦ Problème de redondance
 - ♦ Problème de cohérence

Base de Données

- Une Base de Données (BdD) est un ensemble structuré de données ayant un sens et accessibles par l'ordinateur pour satisfaire simultanément plusieurs utilisateurs
 - ♦ nom, tél., mél., adresse, relation sociale
- Une BdD est conçue, construite et remplie avec des données dans un but précis

Base de Données (suite)

- La structure de la BdD dépend du modèle choisi
 - ♦ Hiérarchique ou réseaux (≈ 1960)
 - ♦ Relationnel ($\approx 1970/1980$)
 - ♦ Objet (≈ 1990)
 - ♦ XML (arborescent) (≈ 2000)
 - ♦ Graphe (en particulier RDF)
- Une BdD peut avoir n'importe quelle taille (agenda personnel ≈ 100 entrées ; Facebook $\approx 600+$ millions d'utilisateurs)

Au-delà des fichiers : Système de Gestion de Base de Données

- Un Système de Gestion de Bases de Données (SGBD) est un ensemble d'outils logiciels permettant la manipulation de BdD
- Facilite la manipulation des données pour des non-informaticiens
- Fournit des fonctionnalités d'administration de la base

Principes fondamentaux des SGBD (1)

- Fidélité
 - ♦ image fidèle de la réalité qu'elle modélise
- Unicité
 - ♦ pas de redondance d'information dans la BdD
- Indépendance
 - ♦ indépendant du modèle de stockage
- Concurrence
 - ♦ Gestion d'accès simultanés à une même donnée.
- Performance
 - ♦ temps d'exécution raisonnable

Principes fondamentaux des SGBD (2)

- Confidentialité
 - ♦ Accessibilité des données dépendant de l'utilisateur
- Intégrité
 - ♦ garanties de fiabilité et de cohérence.
- Robustesse
 - ♦ tolérant aux problèmes matériels, logiciels ou humains

Niveaux de représentation d'une BDD

- niveau externe (sous-schéma conceptuels)
 - ♦ définition des « interfaces » d'accès aux données
 - ♦ géré par le concepteur de la BdD et/ou les utilisateurs
- niveau conceptuel (e.g. modèle Entité-Association)
 - ♦ identification des concepts concrets et abstraits de la réalité
 - ♦ géré par le concepteur de la BDD
- niveau logique (e.g. modèle relationnel)
 - ♦ formalisation de la structure des données
 - ♦ géré par le concepteur de la BDD
- niveau physique (e.g. système de fichiers, index)
 - ♦ stockage physique des données
 - ♦ géré par le SGBD

Niveaux de représentation d'une BDD

Dans ce cours,
On ne s'occupe
Que de ça

- niveau externe (sous-schéma conceptuels)
 - ♦ définition des « interfaces » d'accès aux données
 - ♦ géré par le concepteur de la BdD et/ou les utilisateurs
- niveau conceptuel (e.g. modèle Entité-Association)
 - ♦ identification des concepts concrets et abstraits de la réalité
 - ♦ géré par le concepteur de la BdD
- niveau logique (e.g. modèle relationnel)
 - ♦ formalisation de la structure des données
 - ♦ géré par le concepteur de la BdD
- niveau physique (e.g. système de fichiers, index)
 - ♦ stockage physique des données
 - ♦ géré par le SGBD

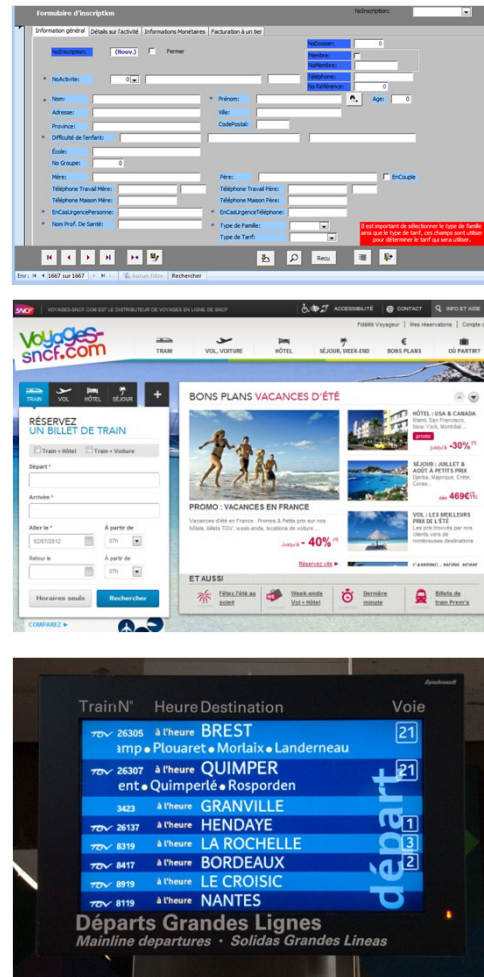
Exemple SNCF


- Plusieurs utilisateurs :


Guichetier



Internaute


Usager




BdD d'entreprise
(employés, salaires, etc)


BdD horaires, trajets


BdD statistiques
(usage, retards, satisfaction, etc)

SGBD SNCF

Plan

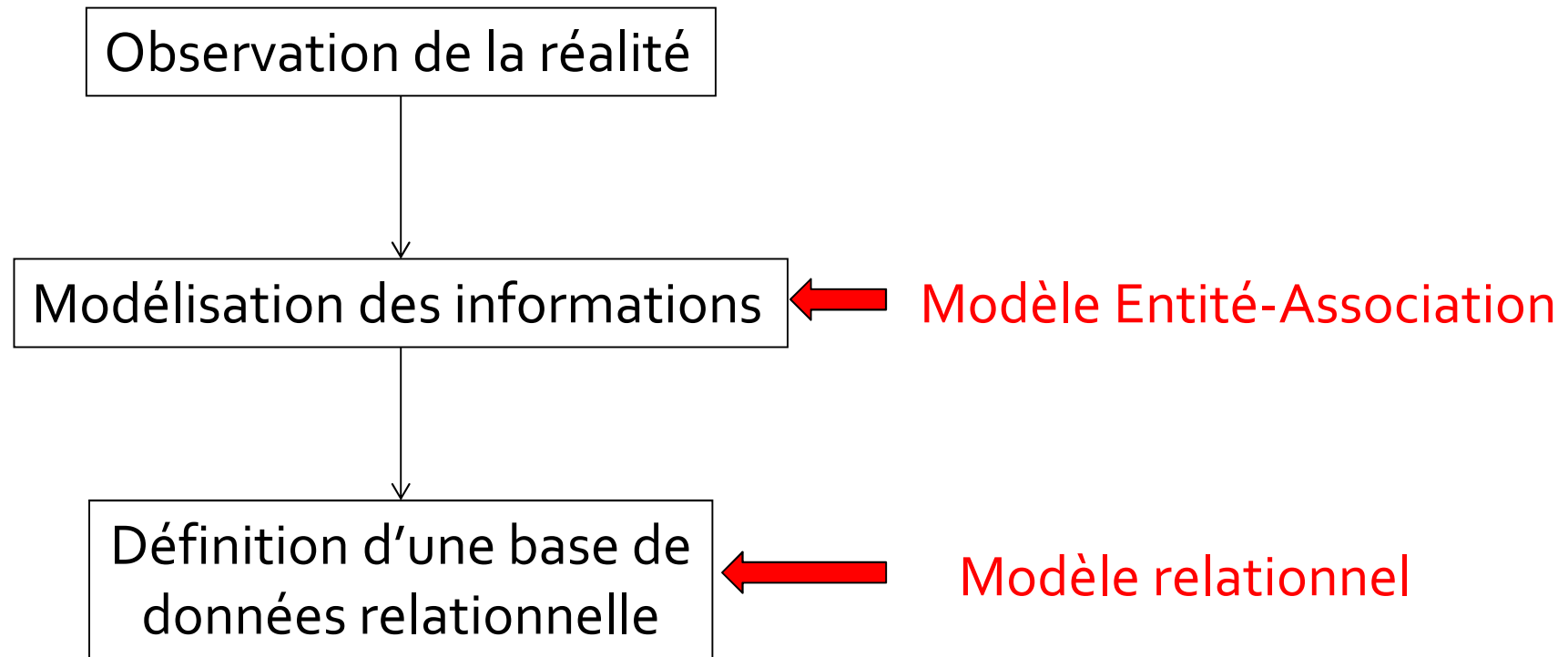
- Introduction sur les Systèmes d'Information
 - ◆ Notions de base
 - ◆ Mise en œuvre technique
- **Le Modèle Entité-Association**
 - ◆ Concepts
 - ◆ Exemple
- Le modèle relationnel
 - ◆ Concepts
 - ◆ Algèbre relationnelle

Le modèle Entité- Association (ou EA)

Origine du modèle Entité-Association

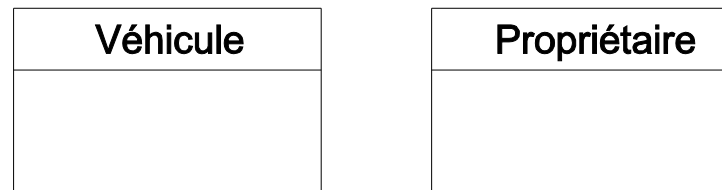
- Proposé par Chen en 1976
- Modèle sémantique pour comprendre et visualiser l'organisation des données
- Également appelé modèle EAR (Entité-Attribut-Relation)
- Objectif : concevoir un Modèle Conceptuel de Données (MCD)

Objectifs & démarche



Concepts de base : Entité & Occurrence

(Type d')entité(s) : type d'objet abstrait ou concret provenant de l'observation du monde réel et pour lequel nous souhaitons enregistrer et connaître des informations



Une **entité** est une instance d'un type d'entités. Souvent, le terme « *entité* » est utilisé à la place de « *type d'entités* », auquel cas on parle d'**occurrence** pour les instances

Ex: la Peugeot 206 immatriculée « 1234 WW 42 » est une occurrence de l'entité Vehicule, et la personne prénommée « Paul Martin » née le 4 fevrier 1980 une occurrence de Propriétaire

Concepts de base : attributs

Attribut : caractéristique d'une entité (ou d'une association) que le concepteur juge nécessaire de répertorier

Véhicule
<i>immatriculation: String</i> <i>marque: String</i>

Propriétaire
<i>nom: String</i> <i>prénom: String</i> <i>naissance: Date</i>

Remarques :

- Les attributs sont typés
- Une entité définit les attributs par leur type
- Une occurrence affecte une valeur à chaque attribut

Concepts de base : clé

Clé : attribut ou un ensemble d'attribut qui permet d'identifier de manière unique une occurrence d'une entité parmi toutes ses occurrences

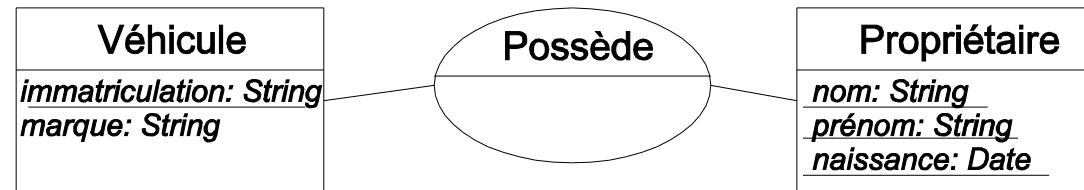
Véhicule
<u>immatriculation: String</u>
marque: String

Propriétaire
nom: String
prénom: String
<u>naissance: Date</u>

- Plus d'une occurrence de Véhicule avec une même immatriculation ne peut pas exister
- Plus d'une occurrence de Propriétaire avec un même nom, prénom et date de naissance ne peut pas exister

Concepts de base : association

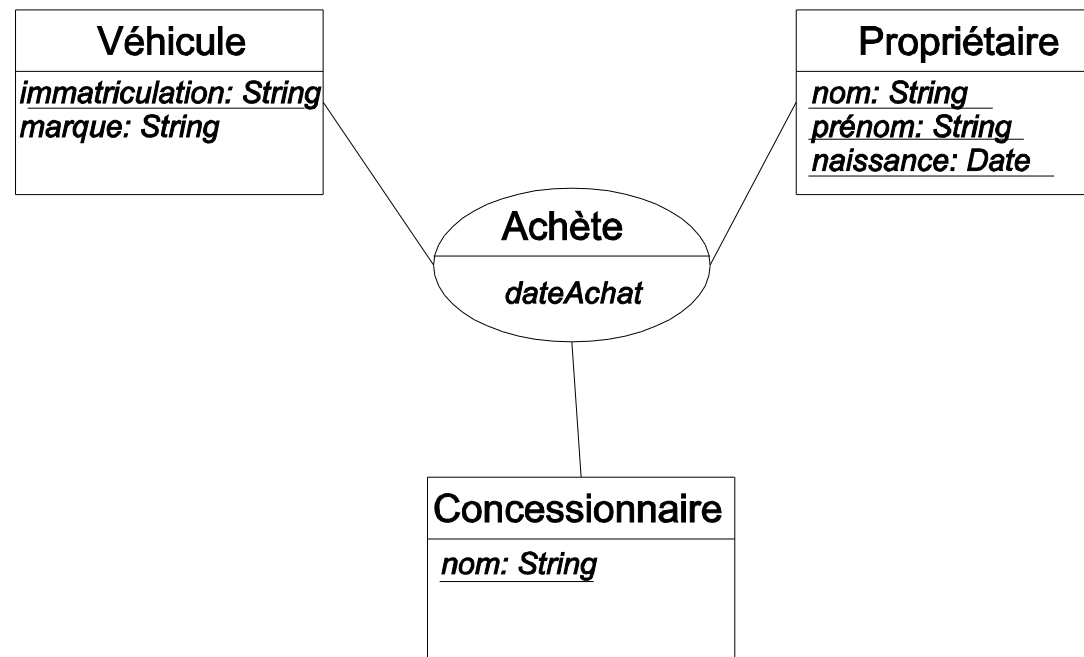
Association : lien entre des entités présentant un intérêt pour la conception que l'on souhaite réaliser



Concepts de base : association (2)

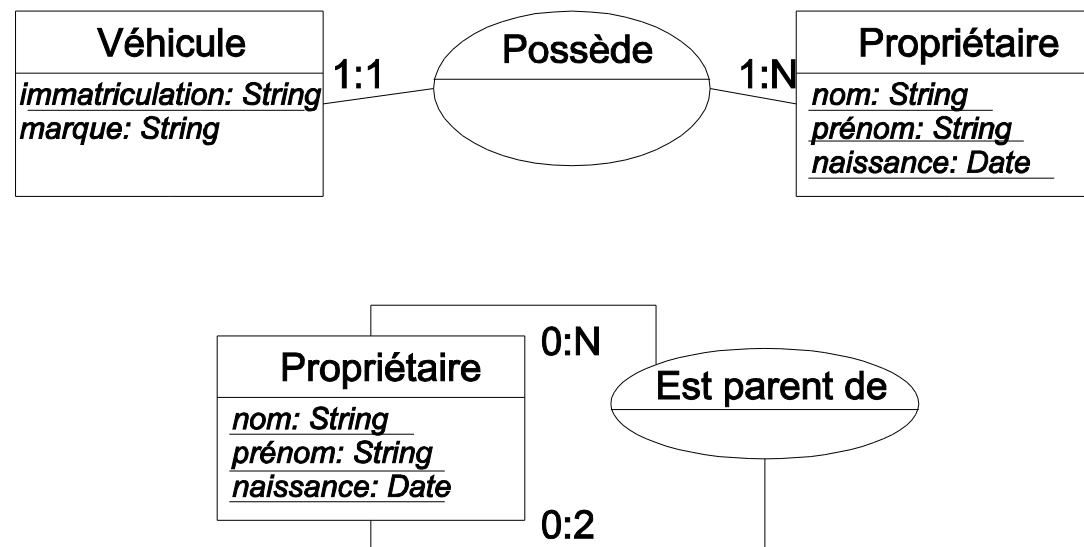
Une association peut :

- relier plus de 2 entités
- bénéficier d'attributs



Cardinalité d'une association

La cardinalité d'une association indique le nombre d'occurrences de chaque entité qui peuvent être impliquées dans une même association



Exemple de modélisation

Entité-Association

On souhaite concevoir le SI correspondant à la gestion de la scolarité de l'EMSE. La scolarité fonctionne de la manière suivante :

- Un élève appartient à une promotion (1A, 2A ou 3A).
- Les élèves d'une promotion suivent plusieurs groupes pédagogiques (GP). Chaque groupe pédagogique est constitué d'unités pédagogiques (UP).
- A chaque GP et UP est affecté un enseignant responsable.
- Un élève obtient une note par UP et par GP qu'il suit.

Exemple : les entités

On représente les entités suivantes :

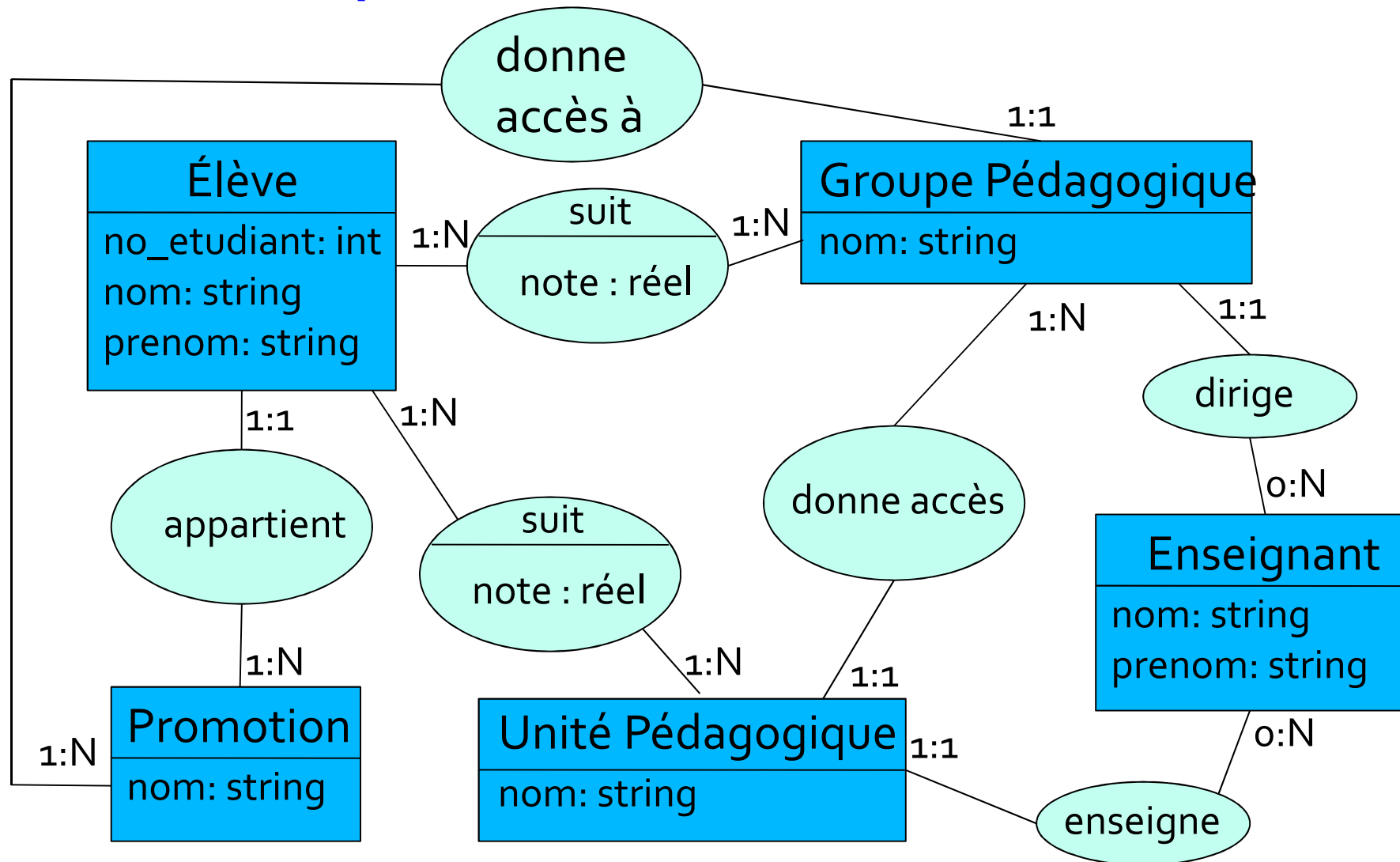
- **Élève**
- **Enseignant**
- **Promotion**
- **Groupe Pédagogique**
- **Unité Pédagogique**

Exemples : les associations

On représente les associations suivantes :

- Un élève **appartient** à une promotion
- Une promotion **donne accès** à certains groupes pédagogiques
- Un élève **suit** plusieurs groupes pédagogiques et y obtient une note
- Un élève **suit** plusieurs unités pédagogiques et y obtient une note
- Les GP et UP sont **enseignés** par un enseignant

Exemple : le modèle E-A



Plan

- Introduction sur les Systèmes d'Information
 - ◆ Notions de base
 - ◆ Mise en œuvre technique
- Le Modèle Entité-Association
 - ◆ Concepts
 - ◆ Exemple
- **Le modèle relationnel**
 - ◆ Concepts
 - ◆ Algèbre relationnelle

Le modèle Relationnel

Querying Databases

- ▶ What is a query?
 - ▶ an **expression** in relational algebra, i.e., **operations** on tables
 - ▶ The result of a query on one or more table(s) is a **table**
- ▶ Operations for filtering:
 - ▶ **Selection** (filter / criteria) = selection of rows
 - ▶ **Projection** = selection of columns
- ▶ Operations for combining 2 tables:
 - ▶ **Cartesian product** (all possibilities)
 - ▶ **Join** (bridging between related rows)
 - ▶ **Union** (union of sets of rows)
 - ▶ **Difference** (elimination of rows)

Relational Algebra Operations

► **Unary** Operations

$$\text{UnOp (Table}_1 \text{)} \rightarrow \text{Table}_2$$

- Selection (or restriction) : choice of rows
- Projection : choice of columns

► **Binary** Operations

$$\text{(Table}_1 \text{) BinOp (Table}_2 \text{)} \rightarrow \text{Table}_3$$

- On tables with the same structure: set-operators
 - Union
 - Intersection
 - Difference
- On tables with different structure
 - Cartesian Product
 - Join

Some definitions

- ▶ A **table** is a set of rows
- ▶ A **row** L is a sequence of values V_i , $L = (V_1, \dots, V_i, \dots, V_n)$
- ▶ The **row** (V_i) built from L is denoted $L(i)$
- ▶ Let be two rows $L_1 = (V_1, \dots, V_n)$ and $L_2 = (W_1, \dots, W_m)$, the **concatenation** of L_1 and L_2 , denoted L_1L_2 is:

$$L_1L_2 = (V_1, \dots, V_n, W_1, \dots, W_m)$$

Selection

Choice of rows according to a Boolean condition

$\sigma_{\text{condition}}$ (Table)

Selection

Choice of rows according to a Boolean condition

$\sigma_{\text{condition}}$ (Table)

Person

Name	Age	Address	Birthplace
James B.	47	London, UK	London, UK
Clark K.	35	Metropolis, USA	Krypton
Lois L.	28	Metropolis, USA	Metropolis, USA
Tarzan	25	London, UK	Forest, Benin

► People living in London:

$\sigma_{\text{Address}=\text{"London, UK"}}(\text{Person})$

Name	Age	Address	Birthplace
James B.	47	London, UK	London, UK
Tarzan	25	London, UK	Forest, Benin

Selection

Choice of rows according to a Boolean condition

$\sigma_{\text{condition}}$ (Table)

Person

Name	Age	Address	Birthplace
James B.	47	London, UK	London, UK
Clark K.	35	Metropolis, USA	Krypton
Lois L.	28	Metropolis, USA	Metropolis, USA
Tarzan	25	London, UK	Forest, Benin

► People living in London:

$\sigma_{\text{Address}=\text{"London, UK"}}(\text{Person})$

Name	Age	Address	Birthplace
James B.	47	London, UK	London, UK
Tarzan	25	London, UK	Forest, Benin

► People living where they were born:

$\sigma_{\text{Address}=\text{Birthplace}}(\text{Person})$

Name	Age	Address	Birthplace
James B.	47	London, UK	London, UK
Lois L.	28	Metropolis	Metropolis

Selection (contd.)

- Persons being less than 30 y.o.:

$\sigma_{\text{Age} < 30}(\text{Person})$

Name	Age	Address	Birthplace
Lois L.	28	Metropolis	Metropolis
Tarzan	25	London, UK	Forest, Benin

this is a table

Selection (contd.)

- Persons being less than 30 y.o.:

$\sigma_{\text{Age} < 30}(\text{Person})$

Name	Age	Address	Birthplace
Lois L.	28	Metropolis	Metropolis
Tarzan	25	London, UK	Forest, Benin

this is a table too!

Selection (contd.)

- ▶ People less than 30 y.o.:

$\sigma_{\text{Age} < 30}(\text{Person})$

Name	Age	Address	Birthplace
Lois L.	28	Metropolis	Metropolis
Tarzan	25	London, UK	Forest, Benin

- ▶ **Composition of selection:** people less than 30 living in London:

$\sigma_{\text{Address} = \text{"London, UK"}}(\sigma_{\text{Age} < 30}(\text{Person}))$

Name	Age	Address	Birthplace
Tarzan	25	London, UK	Forest, Benin

Selection (contd.)

- ▶ People less than 30 y.o.:

$$\sigma_{\text{Age} < 30}(\text{Person})$$

Name	Age	Address	Birthplace
Lois L.	28	Metropolis	Metropolis
Tarzan	25	London, UK	Forest, Benin

- ▶ **Composition of selection:** people less than 30 living in London:

$$\sigma_{\text{Address} = \text{"London, UK"}}(\sigma_{\text{Age} < 30}(\text{Person}))$$

- ▶ **Conjunction of selection:**

$$\sigma_{\text{Address} = \text{"London, UK"} \text{ and } \text{Age} < 30}(\text{Person})$$

equivalent to the composition

Name	Age	Address	Birthplace
Tarzan	25	London, UK	Forest, Benin

Formalism for the selection operation

► Selection / constant:

$\sigma_{i=a}(T)$ is the set of rows L of the table T where $L(i) = a$

$$\sigma_{i=a}(T) = \{ L \mid L \in T \text{ and } L(i) = a \}$$

► Selection / inter-columns:

$\sigma_{i=j}(T)$ is the set of rows L of the table T where $L(i) = L(j)$

$$\sigma_{i=j}(T) = \{ L \mid L \in T \text{ and } L(i) = L(j) \}$$

► Comparison operators \neq , $<$, $>$, \leq , \geq can be used

► Boolean operators **AND**, **OR**

Projection

Choice of columns in a table

$\pi_{\text{column(s)}} (\text{Table})$

Projection

Choice of columns in a table

$\pi_{\text{column(s)}} (\text{Table})$

Person

Name	Age	Address	Birthplace
James B.	47	London, UK	London, UK
Clark K.	35	Metropolis, USA	Krypton
Lois L.	28	Metropolis, USA	Metropolis, USA
Tarzan	25	London, UK	Forest, Benin

► Name and age of people:

$\pi_{\text{Name, Age}}(\text{Person})$

Name	Age
James B.	47
Clark K.	35
Lois L.	28
Tarzan	25

Projection (contd.)

- ▶ Name of people living in London:

$\pi_{\text{Name}}(\sigma_{\text{Address}=\text{"London, UK"}}(\text{Person}))$

Name	Age	Address	Birthplace
James B.	47	London, UK	London, UK
Tarzan	25	London, UK	Forest, Benin

Projection (contd.)

- Name of people living in London:

$\pi_{\text{Name}}(\sigma_{\text{Address}=\text{"London, UK"}}(\text{Person}))$

Name	Age	Address	Birthplace
James B.	47	London, UK	London, UK
Tarzan	25	London, UK	Forest, Benin

Name
James B.
Tarzan

Formalism for the projection operation

► Projection onto columns:

$\pi_{i_1, \dots, i_k}(T)$ is the set of rows of the table T obtained by keeping columns i_1, i_2, \dots, i_k

$$\pi_{i_1, \dots, i_k}(T) = \{ (L(i_1), \dots, L(i_k)) \mid L \in T \}$$



The result of any operation on tables is also a table
 \Rightarrow different operations can be combined



The result of selections or projection
can be **an empty table**

Cartesian Product

Each row of the first table is combined with all the rows of the second table

$$\text{Table}_1 \otimes \text{Table}_2$$

Cartesian Product

Each row of the first table is combined with all the rows of the second table

Table₁ \otimes Table₂

Man

Name	Address
James B.	London
Clark K.	Metropolis
Tarzan	London

Woman

Name	Address
Lois L.	Metropolis
Catwoman	Gotham City

► All man-woman pairs:

Man \otimes Woman

Name ₁	Address ₁	Name ₂	Address ₂
James B.	London	Lois L.	Metropolis
James B.	London	Catwoman	Gotham City
Clark K.	Metropolis	Lois L.	Metropolis
Clark K.	Metropolis	Catwoman	Gotham City
Tarzan	London	Lois L.	Metropolis
Tarzan	London	Catwoman	Gotham City

Cartesian Product (contd.)

Man

Name	Address
James B.	London
Clark K.	Metropolis
Tarzan	London

City

CName	Population	Country
London	7,556,900	UK
Metropolis	8,391,881	USA

► In which country do the men of the Man table live?

1) $\text{Man} \otimes \text{City}$

Name	Address	CName	Population	Country
James B.	London	London	7,556,900	UK
James B.	London	Metropolis	8,391,881	USA
Clark K.	Metropolis	London	7,556,900	UK
Clark K.	Metropolis	Metropolis	8,391,881	USA
Tarzan	London	London	7,556,900	UK
Tarzan	London	Metropolis	8,391,881	USA

Cartesian Product (contd.)

2) Followed by a selection:

$\sigma_{\text{Address=CName}}(\text{Man} \otimes \text{City})$

Name	Address	CName	Population	Country
James B.	London	London	7,556,900	UK
James B.	London	Metropolis	8,391,881	USA
Clark K.	Metropolis	London	7,556,900	UK
Clark K.	Metropolis	Metropolis	8,391,881	USA
Tarzan	London	London	7,556,900	UK
Tarzan	London	Metropolis	8,391,881	USA

Name	Address	CName	Population	Country
James B.	London	London	7,556,900	UK
Clark K.	Metropolis	Metropolis	8,391,881	USA
Tarzan	London	London	7,556,900	UK

Cartesian Product (contd.)

- ▶ Man-woman pairs living in the same city:

$$\sigma_{\text{Address}_1 = \text{Address}_2} (\text{Man} \otimes \text{Woman})$$

Name ₁	Address ₁	Name ₂	Address ₂
James B.	London	Lois L.	Metropolis
James B.	London	Catwoman	Gotham City
Clark K.	Metropolis	Lois L.	Metropolis
Clark K.	Metropolis	Catwoman	Gotham City
Tarzan	London	Lois L.	Metropolis
Tarzan	London	Catwoman	Gotham City

Cartesian Product (contd.)

- ▶ Man-woman pairs living in the same city:

$$\sigma_{\text{Address}_1 = \text{Address}_2}(\text{Man} \otimes \text{Woman})$$

Name ₁	Address ₁	Name ₂	Address ₂
James B.	London	Lois L.	Metropolis
James B.	London	Catwoman	Gotham City
Clark K.	Metropolis	Lois L.	Metropolis
Clark K.	Metropolis	Catwoman	Gotham City
Tarzan	London	Lois L.	Metropolis
Tarzan	London	Catwoman	Gotham City

Name ₁	Address ₁	Name ₂	Address ₂
Clark K.	Metropolis	Lois L.	Metropolis

Formalism for the cartesian product operation

- ▶ $T_1 \otimes T_2$ is the set of all rows that can be obtained by concatenating a row of table T_1 with a row of table T_2

$$T_1 \otimes T_2 = \{ L_1 L_2 \mid L_1 \in T_1 \text{ and } L_2 \in T_2 \}$$



The cartesian product $T_1 \otimes T_2$ is often followed by a selection in which one column from T_1 and one column from T_2 are equalled

- ▶ So much so that there is an operation for this called **Join**

Join

Cartesian product followed by a selection

$\text{Table}_1 \bowtie_{\text{condition}} \text{Table}_2$

Join

Cartesian product followed by a selection

$$\text{Table}_1 \bowtie_{\text{condition}} \text{Table}_2$$

Man

Name	Address
James B.	London
Clark K.	Metropolis
Tarzan	London

City

CName	Population	Country
London	7,556,900	UK
Metropolis	8,391,881	USA

► In which country do the men of the Man table live?

$$\text{Man} \bowtie_{\text{Address=CName}} \text{City} \Leftrightarrow \sigma_{\text{Address=CName}}(\text{Man} \otimes \text{City})$$

Name	Address	CName	Population	Country
James B.	London	London	7,556,900	UK
Clark K.	Metropolis	Metropolis	8,391,881	USA
Tarzan	London	London.	7,556,900	UK

Join (contd.)

- Couple man-woman of the same city

Man

Name	Address
James B.	London
Clark K.	Metropolis
Tarzan	London

Woman

Name	Address
Lois L.	Metropolis
Catwoman	Gotham City

$$\text{Man} \bowtie \text{Woman} \Leftrightarrow \sigma_{\text{Address}_1=\text{Address}_2}(\text{Man} \otimes \text{Woman})$$

Name ₁	Address ₁	Name ₂	Address ₂
Clark K.	Metropolis	Lois L.	Metropolis

Formalism for the join operation

- ▶ $T_1 \bowtie_{i=j} T_2$ is the set of rows obtained by concatenating a row of table T_1 with a row of table T_2 such that $L_1(i) = L_2(j)$

$$T_1 \bowtie_{i=j} T_2 = \{ L_1 L_2 \mid L_1 \in T_1 \text{ and } L_2 \in T_2 \text{ and } L_1(i) = L_2(j) \}$$

- ▶ Link with the Cartesian product and the selection:
 - ▶ Let m be the number of columns of T_1

$$T_1 \bowtie_{i=j} T_2 = \sigma_{i=j+m}(T_1 \otimes T_2)$$



As for selection, comparison operators $\neq, <, >, \leq, \geq$ can be used as well as boolean operators

Union

Man

Name	Address
James B.	London
Clark K.	Metropolis
Tarzan	London

Woman

Name	Address
Lois L.	Metropolis
Catwoman	Gotham City

► The set of people:

$\text{Man} \cup \text{Woman}$

Name	Address
James B.	London
Clark K.	Metropolis
Tarzan	London
Lois L.	Metropolis
Catwoman	Gotham City

Union

Man

Name	Address
James B.	London
Clark K.	Metropolis
Tarzan	London

Woman

Name	Address
Lois L.	Metropolis
Catwoman	Gotham City

- The set of people:

$\text{Man} \cup \text{Woman}$

Name	Address
James B.	London
Clark K.	Metropolis
Tarzan	London
Lois L.	Metropolis
Catwoman	Gotham City

- People living in Metropolis:

$\sigma_{\text{Address}=\text{"Metropolis"}}(\text{Man} \cup \text{Woman})$

Name	Address
Clark K.	Metropolis
Lois L.	Metropolis

Difference

Man

Name	Address
James B.	London
Clark K.	Metropolis
Tarzan	London

Woman

Name	Address
Lois L.	Metropolis
Catwoman	Gotham City

- What are the cities for which we know at least one woman and no man?

$$\pi_{\text{Address}}(\text{Woman}) - \pi_{\text{Address}}(\text{Man})$$

Address
Gotham City

Formalism for these two operations

Let T_1 and T_2 be 2 tables with **same number of columns** and columns have the **same domain**.

► Formalising **union**

$T_1 \cup T_2$ is the set containing the rows of T_1 and those of T_2 .

$$T_1 \cup T_2 = \{ L \mid L \in T_1 \text{ or } L \in T_2 \}$$

► Formalising **difference**

$T_1 - T_2$ is the set containing the rows of T_1 that are not in T_2 .

$$T_1 - T_2 = \{ L \mid L \in T_1 \text{ and } L \notin T_2 \}$$

Query Tree

- ▶ What are all the possible couples with a man from London and a woman from Metropolis?

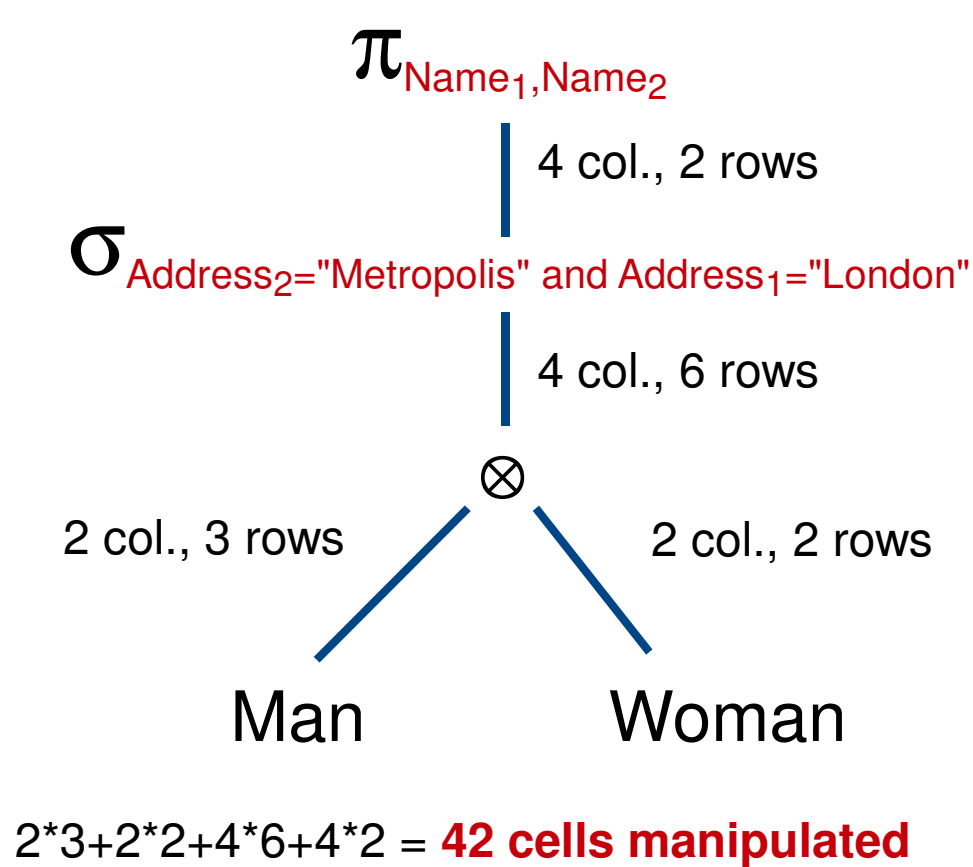
$$\pi_{\text{Name}_1, \text{Name}_2}(\sigma_{\text{Address}_2 = \text{"Metropolis"}}(\sigma_{\text{Address}_1 = \text{"London"}}(\text{Man} \otimes \text{Woman})))$$

OR

$$\pi_{\text{Name}_1}(\sigma_{\text{Address}_1 = \text{"London"}}(\text{Man})) \otimes \pi_{\text{Name}_2}(\sigma_{\text{Address}_2 = \text{"Metropolis"}}(\text{Woman}))$$

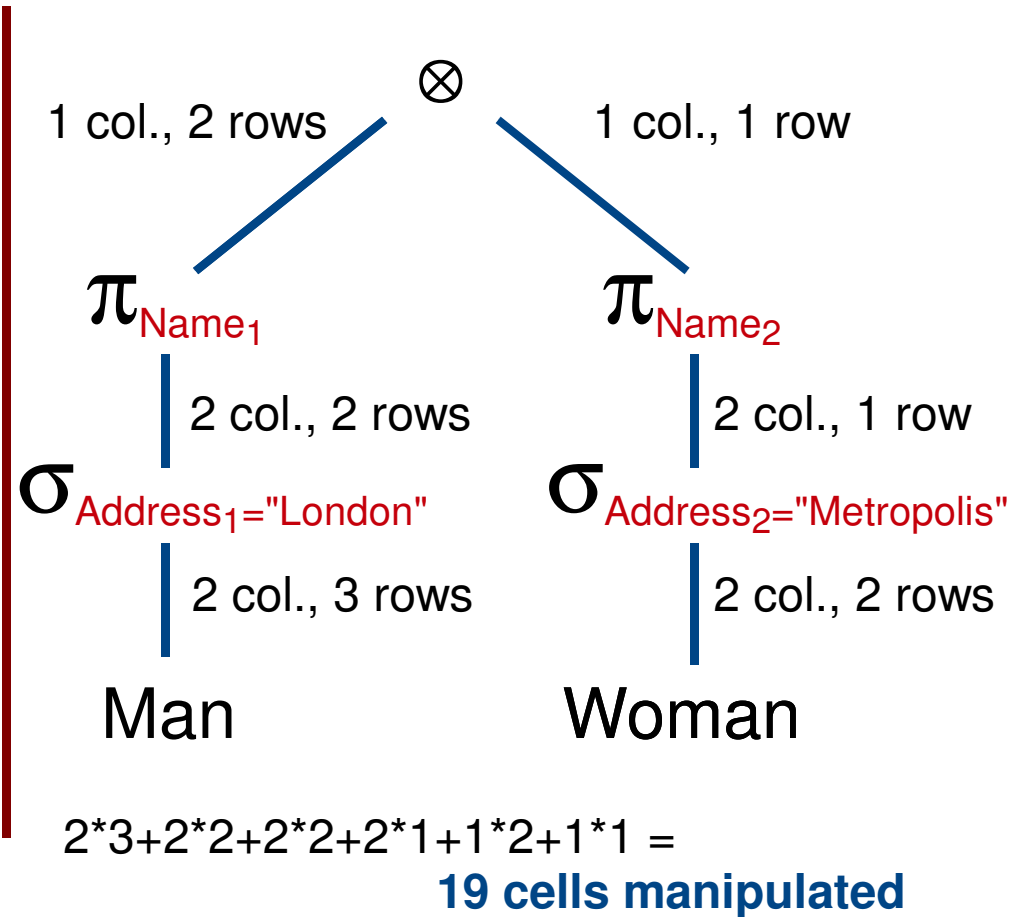
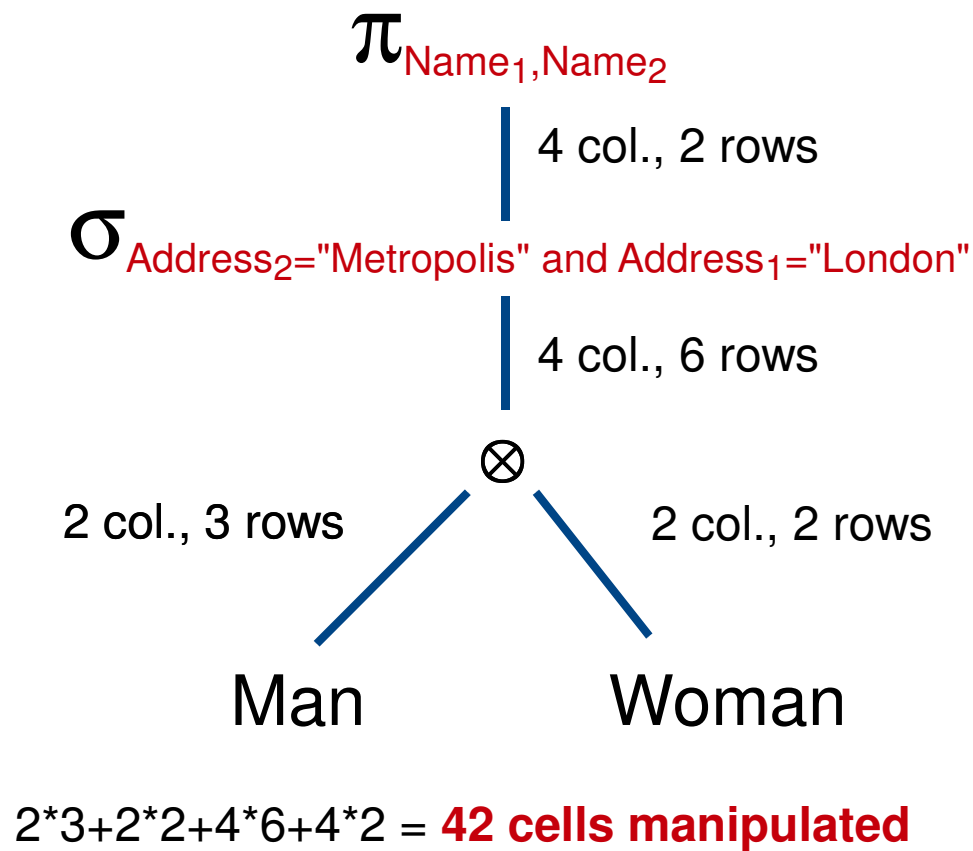
Query Tree

- What are all the possible couples with a man from London and a woman from Metropolis?



Query Tree

- What are all the possible couples with a man from London and a woman from Metropolis?



Du modèle Entité-Association vers le modèle relationnel

- Un modèle Entité-Association peut se traduire simplement en un modèle relationnel
 - ♦ Schéma de données plus formalisé
 - ♦ Possibilité d'utiliser l'algèbre relationnel pour exprimer des requêtes
- 3 règles simples de transformation

Règle 1

Toute entité est traduite par une relation contenant les mêmes attributs et clés que l'entité

Véhicule
<i>immatriculation: String</i>
<i>marque: String</i>

Propriétaire
<i>nom: String</i>
<i>prénom: String</i>
<i>naissance: Date</i>

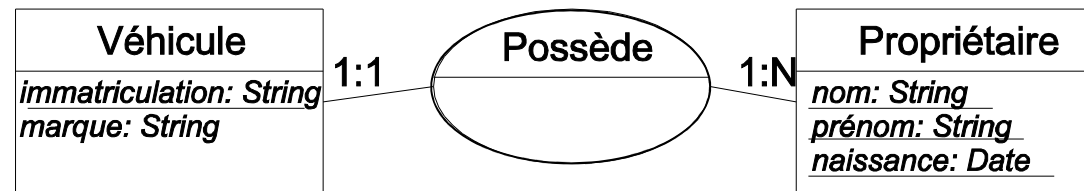
Se traduit par :

Vehicule(immatriculation: string, marque: string)

Proprietaire(nom: string, prenom; string, naissance: date)

Règle 2

Toute association depuis une entité R vers une entité R' ayant une cardinalité 0:1 ou 1:1 se traduit par l'ajout, dans la relation résultat de la traduction de R , de la clé de R' avec le statut d'attributs

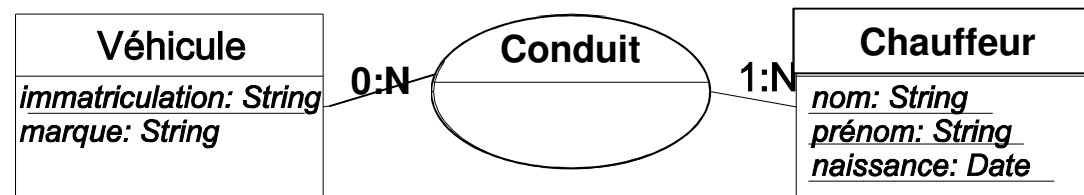


Se traduit par :

Vehicule(immatriculation: string, marque: string, nom_prop: string, prenom_prop; string, naissance_prop: date)

Règle 3

Toute association entre entités ayant des cardinalités $0:N$ ou $1:N$ à chaque extrémité se traduit par la création d'une relation contenant comme attributs les clés des entités associées ainsi que d'éventuels attributs de l'association. La clé de la relation créée est l'ensemble des attributs représentant les clés des entités associées.



Se traduit par :

Achete(immatriculation: string, nom_cond: string, prenom_cond: string, naiss_cond: date)

Introduction de clés numériques

- Quand la clé d'une entité est composée de plusieurs attributs, il peut être efficace d'introduire une clé artificielle numérique
- Cette clé ne correspond à aucun attribut réel mais facilite les jointures et sélection

Ex: *Proprietaire(numero: int, nom: string, prenom: string, naissance: date)*

- ! Le numéro de sécurité sociale serait idéal mais interdit par la CNIL !

Cohérence des données

- Les relations définissent la structure des données mais pas la cohérence de contenu.
- Exemple de contenu incohérent : une même personne enregistrée avec deux dates de naissance différentes
- La gestion de la cohérence des données consiste à rendre impossible la saisie de données incohérente
- 2 concepts clés
 - Dépendances fonctionnelles
 - Formes normales

Dépendances fonctionnelles

(1)

- Une dépendance fonctionnelle (DF) représente le fait qu'à la valeur d'un ou plusieurs attributs, on associe une valeur pour un autre attribut

Soit $R(A_1, A_2, \dots, A_n)$ et une DF de A_i vers A_j se note :

$$A_i \rightarrow A_j$$

- Par exemple, il existe une DF entre le numéro et le nom d'un propriétaire

$$\text{numero} \rightarrow \text{nom}$$

Dépendances fonctionnelles

(2)

- Une DF $A \rightarrow B$ est dite élémentaire si un sous-ensemble de la clé n'est pas source de la DF

$$\forall a \in A, A - \{a\} \not\rightarrow B$$

- Une DF $A \rightarrow B$ est dite directe s'il n'existe pas d'ensemble d'attributs C tel que

$$A \rightarrow C \text{ et } C \rightarrow B$$

Suite du cours

2013

Jeu. 14/11, 10h00 : TD1 modélisation E.-A.

Lun. 18/11, 8h15 : TD2 algèbre rel. / 10h00 : TP1 modèle rel.

Jeudi. 21/11, 8h15 : TD3 SQL / 10h00 : TP2 SQL

Jeu. 05/12, 8h15 : TD modélisation E.-A. projet (3h) PROJET

Jeu. 19/12, 8h15 : TD4 SQL / TP3 SQL

2014

Jeu. 16/01, 8h15 : présentation partie dev. projet PROJET

Jeu. 16/01, 10h00 : implém. modèle projet PROJET

Jeu. 23/01, 8h15 : TP4 dev. PHP / **TP dev. projet PROJET**

Jeu. 30/01, 8h15 : TP dev. projet (3h) PROJET

Lun. 03/02, 10h00 : TP dev. projet (1h30) PROJET

Mer. 05/02, 8h15 : finalisation projet (3h) PROJET

Jeu. 06/02, 10h00 : Exam

Le projet CSI

Concevoir une application de gestion d'hôpital



Le système doit permettre de gérer des informations sur des chambres, lits, patients ainsi que le personnel de l'hôpital. L'implémentation utilisera un BdD relationnel et une interface Web codée en PHP.

Une modélisation détaillée du domaine sera demandée, puis une implémentation partielle des fonctionnalités.

Le projet se fait par quadrinôme à définir d'ici vendredi.

Pour l'implémentation des fonctionnalités, les groupes se diviseront en binômes.

Projet alternatif

Nous avons un projet avec un **vrai client** ayant un besoin en système d'information

Le travail à effectuer est de l'ordre de difficulté d'un projet CSI

Projet tuteuré par Philippe Beaune, qui est en contact avec le client

Seulement quelques groupes **volontaires** peuvent choisir ce projet

Attention : le travail à effectuer n'est pas forcément « équitable » par rapport au projet minesbook mais correspond à une situation réelle d'ingénieur

Tous les supports de ce cours sont
disponibles ici :

<http://www.emse.fr/~zimmermann/Teaching/CSI/>