

Conception de bases de données

Dr. Rania OTHMAN

rania.othman@intervenants.efrei.net



Outils

- **Outils de modélisation :**

- Draw.io (Diagrams.net) : <https://app.diagrams.net/>
- Looping (recommandé) : <https://www.looping-mcd.fr/>
- JMerise : <https://www.jfreesoft.com/JMerise/>
- DB Designer : <https://www.dbdesigner.net/>
- AnalyseSI : <http://www.analysesi.com/>
- Mocodo (en ligne) : <http://www.mocodo.net/>

Outils

- **SGBD : MySQL (Workbench) :**

Windows

Installer : <https://dev.mysql.com/downloads/installer/>

MacOS

Installer : [MySQL :: Download MySQL Community Server \(Archived Versions\)](#)

Workbench: [MySQL :: Download MySQL Workbench](#)

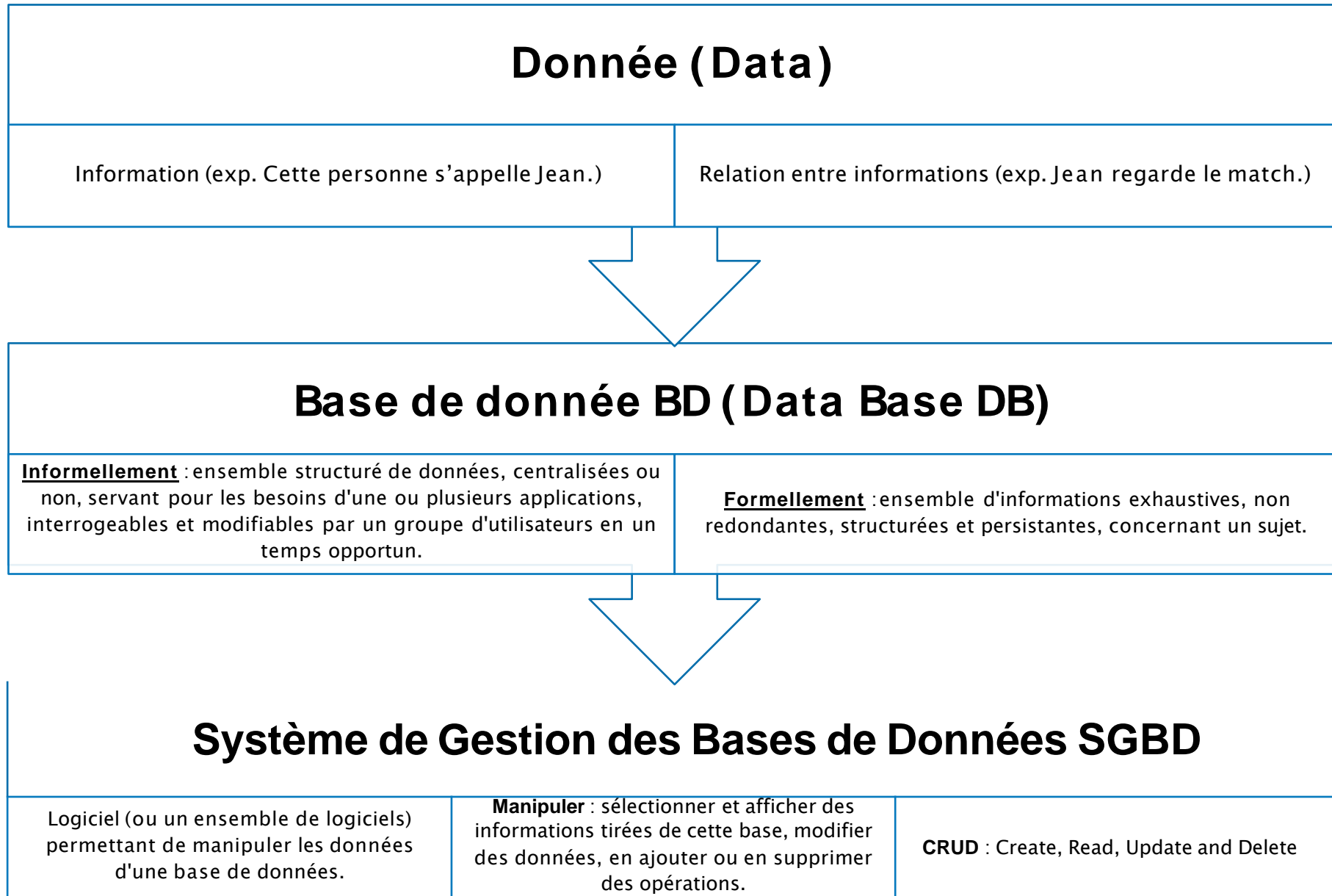
Tuto: [SQL 2.1 - Mac OS - Installer MYSQL & Workbench – YouTube](#)

A decorative graphic consisting of two parallel diagonal stripes, one blue and one orange, running from the bottom left towards the top right of the slide. They appear to be layered, with the orange stripe slightly offset from the blue one.

01

Principaux concepts de Base de données et de SGBD

Notions de base



SGBD : Les composants

LDD : Langage de Définition des Données

- Le résultat de la compilation est un ensemble de tables, stockées dans un fichier spécial (dictionnaire/répertoire) des données

LMD : Langage de Manipulation des Données

- Prenant en charge leur consultation et leur modification de façon optimisée, ainsi que les aspects de sécurité

Sauvegarde et récupération

- Mécanismes permettant de pouvoir revenir à l'état antérieur de la base (Transaction)

Accès concurrents aux données

- En minimisant l'attente des utilisateurs et en garantissant l'obtention de données cohérentes en cas de mises à jours simultanées

02

Conception et modélisation de base de données

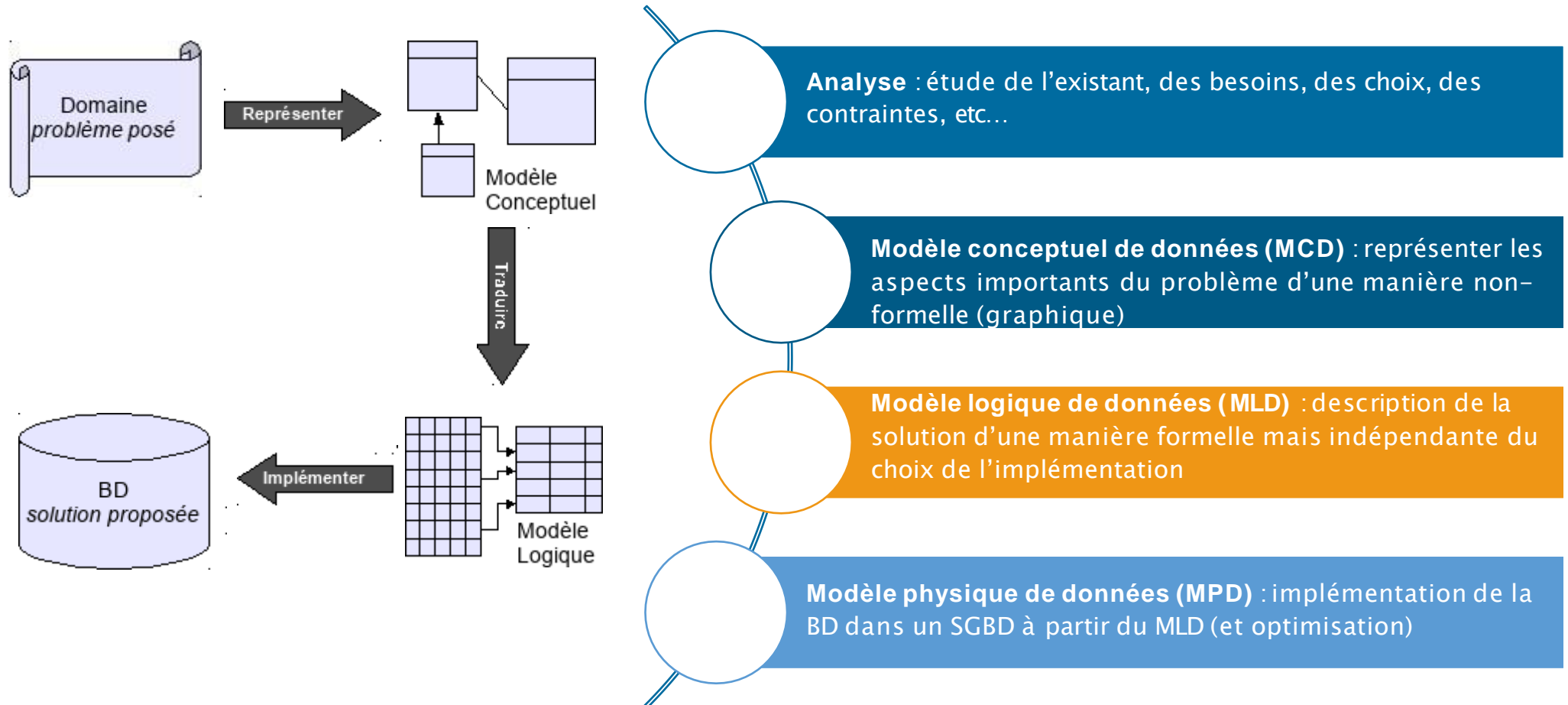


Modèle E/A

Modèle relationnel

Normalisation

Processus de conception d'une base de données



Bien analyser, bien modéliser !



L'étape d'analyse est fondamentale dans le processus de conception

Bien analyser

!

La perception de l'existant et des besoins repose sur l'expertise de l'ingénieur



L'étape de modélisation conceptuelle doit être correctement réalisée

Bien modéliser

!

Passer d'une spécification en langage naturel soumise à interprétation, à une spécification non ambiguë nécessite l'utilisation de formalismes de modélisation tels que E-A ou UML

→ Le modèle logique est déduit de façon systématique du modèle conceptuel et l'implémentation logicielle est réalisée par traduction directe du modèle logique.

Modèle E/A

Le modèle E/A

Conception des bases de données (**relationnelles principalement**)

Mettre en place un schéma correct permettant le développement d'une application viable.

Distinguer les entités qui constituent la base de données, et les associations (relations) entre ces entités.

Ces concepts permettent de donner une structure à la base, ce qui s'avère indispensable.

Simple et suffisamment puissant pour représenter des structures relationnelles.

Il repose sur une représentation graphique qui facilite considérablement sa compréhension.

Souffre de nombreuses insuffisances :

Il n'existe pas d'opération permettant de manipuler les données, et pas (ou peu) de moyen d'exprimer des contraintes

Il mène souvent à certaines ambiguïtés pour des schémas complexes.

Le concepteur doit répondre au mieux aux exigences

Le jargon !

- Une **entité** est similaire à la notion d'objet, elle décrit une «entité» du monde réel.
 - Exemple : un **livre**, un **étudiant**, un **compte**, une **facture**, ...
- Une **association** est un lien entre plusieurs entités
 - Exemple : une facture **contient** plusieurs produits.

E/A : représentation informelle

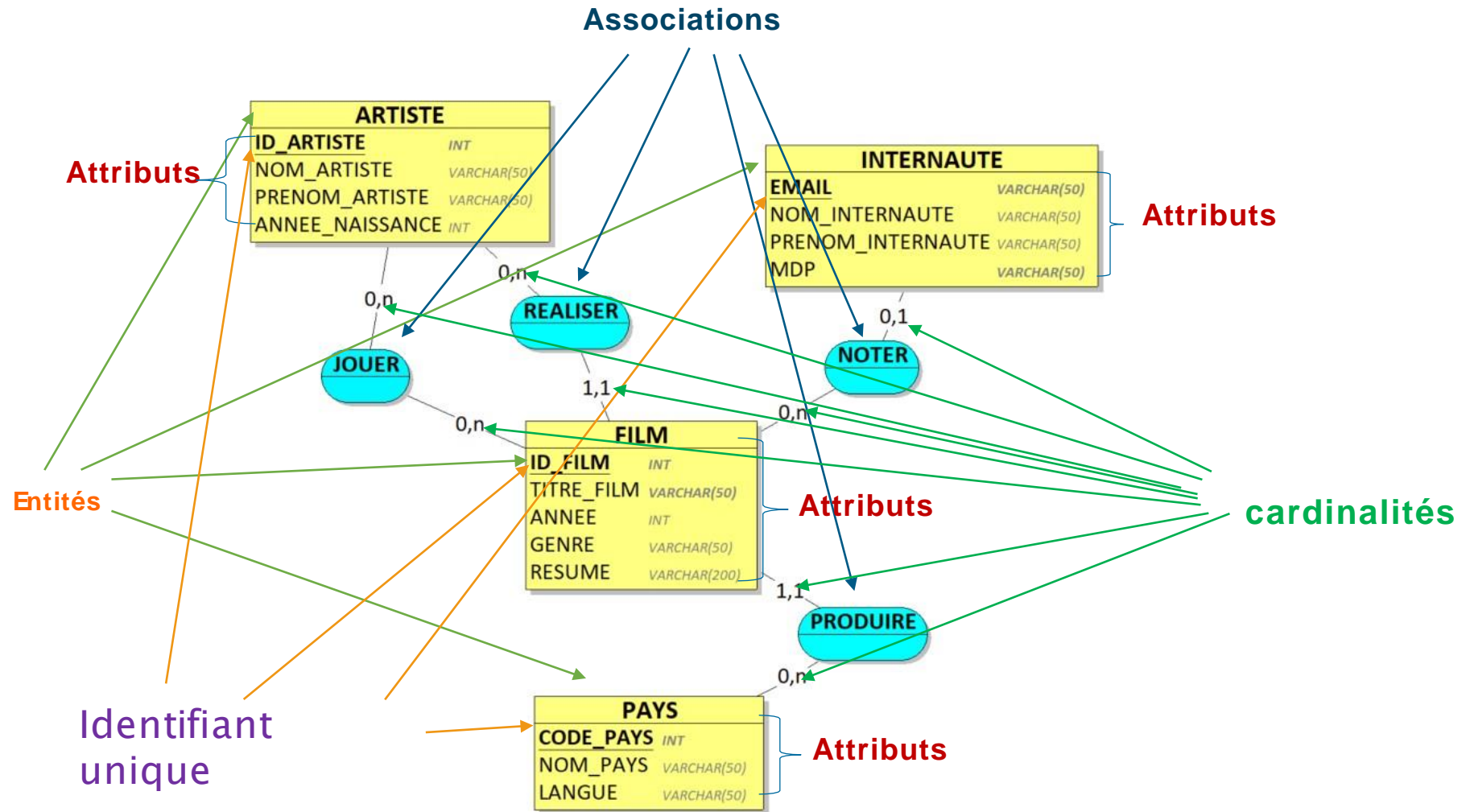
« Soit une base de données décrivant des **films**, avec leur **metteur en scène** et leurs **acteurs**, ainsi que les **cinémas** où passent ces films. Cette BD est accessible sur le Web et des **internauts** peuvent noter les films qu'ils ont vus. »

titre	année	nomMES	prénomMES	annéeNaiss
Alien	1979	Scott	Ridley	1943
Vertigo	1958	Hitchcock	Alfred	1899
Psychose	1960	Hitchcock	Alfred	1899
Kagemusha	1980	Kurosawa	Akira	1910
Volte-face	1997	Woo	John	1946
Pulp Fiction	1995	Tarantino	Quentin	1963
Titanic	1997	Cameron	James	1954
Sacrifice	1986	Tarkovski	Andrei	1932

Même année

Même MES

E/A : représentation formelle



Entité, attribut et type d'entité

Entité : tout objet identifiable et pertinent pour l'application

⇒ exp : Film, Artiste, Internaute...

Attribut : une propriété qui caractérise une entité

⇒ Exp : Titre d'un film, le nom d'un artiste...

⇒ Un attribut est **atomique** : prend une valeur et une seule

⇒ Un attribut peut être **multivalué** : ensemble de valeurs prises dans un même domaine (numéros de téléphone)

⇒ Un attribut peut être **composé** : constitué par agrégation d'autres attributs (adresse : n° rue, nom de la rue, code postal, pays)

Le **type d'une entité** est composé des éléments suivants :

⇒ Son nom;

⇒ La liste de ses attributs avec, – optionnellement – le domaine où l'attribut prend ses valeurs : les entiers, les chaînes de caractères ;

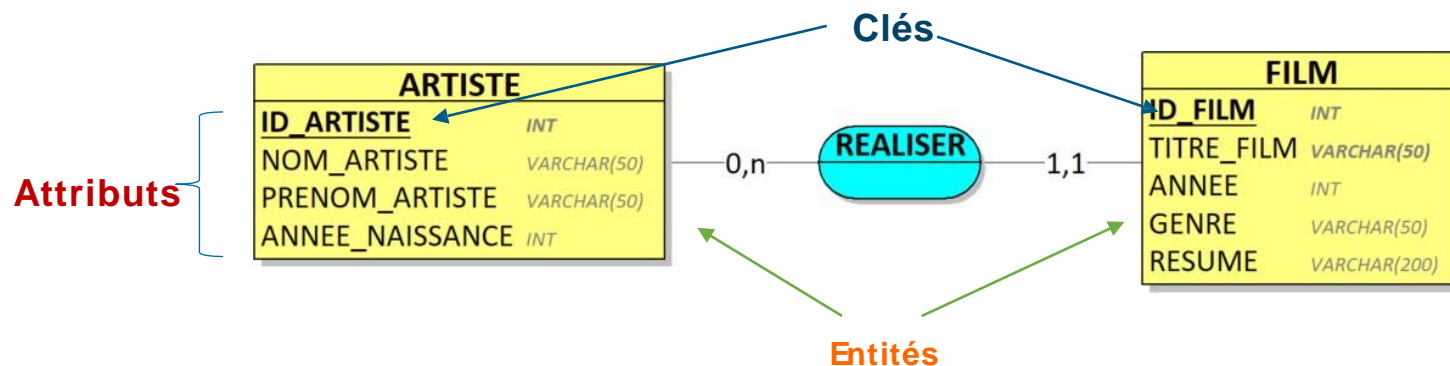
⇒ L'indication du (ou des) attribut(s) permettant d'identifier l'entité : ils constituent la clé.

Identifiant unique : clé

Clé : Soit **E** un type d'entité, **A** l'ensemble de ses attributs, une **clé** de **E** est un **sous-ensemble minimal** de **A**, permettant d'identifier de manière unique une entité parmi n'importe quelle extension de E

- Les caractéristiques d'une bonne **clé** sont :
 - ⇒ sa valeur est connue pour toute entité ;
 - ⇒ on ne doit jamais avoir besoin de la modifier ;
 - ⇒ sa taille de stockage doit être la plus petite possible (performance de stockage)

Exemples : id_Film pour identifier un film, la paire (**email,mdp**) pour identifier un internaute, etc

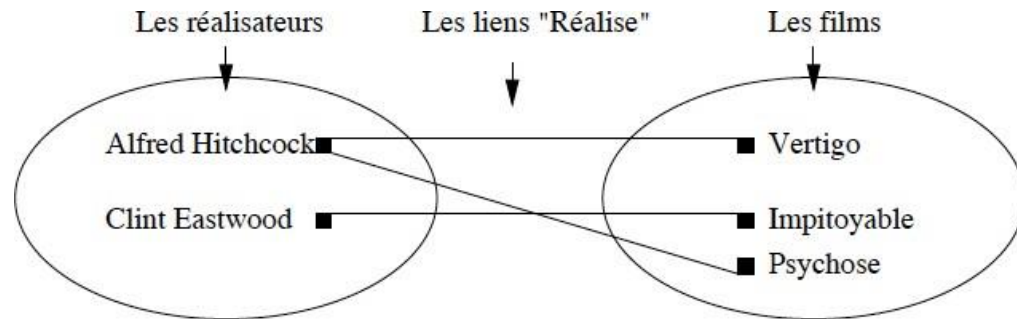


Choix de l'identifiant

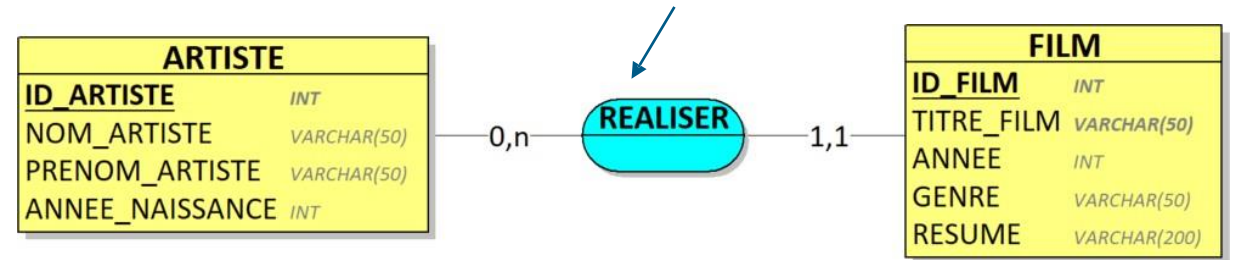
- ❑ Évitez les identifiants composés de plusieurs attributs (comme, par exemple, un identifiant formé par les attributs nom et prénom) :
 - ✓ ils dégradent les performances du SGBD,
 - ✓ mais surtout l'unicité supposée par une telle démarche finit généralement, tôt ou tard, par être démentie
- ❑ Évitez les identifiants susceptibles de changer au cours du temps (comme la plaque d'immatriculation d'un véhicule)
- ❑ Évitez les identifiants du type chaîne de caractère

Association

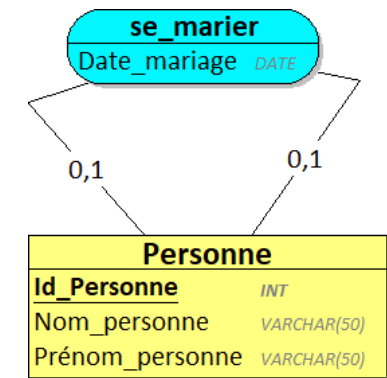
- Une association (appelée aussi parfois relation) représente les liens sémantiques qui peuvent exister entre plusieurs entités



Association



- Une classe d'association peut avoir des attributs (pour réaliser, on peut ajouter date_rea)
- Voici les dénominations des classes d'association :
 - une classe d'association *récursive* (ou réflexive) relie la même classe d'entité
 - une classe d'association *binaire* relie deux classes d'entité
 - une classe d'association *ternaire* relie trois classes d'entité
 - une classe d'association *n-aire* relie n classes d'entité

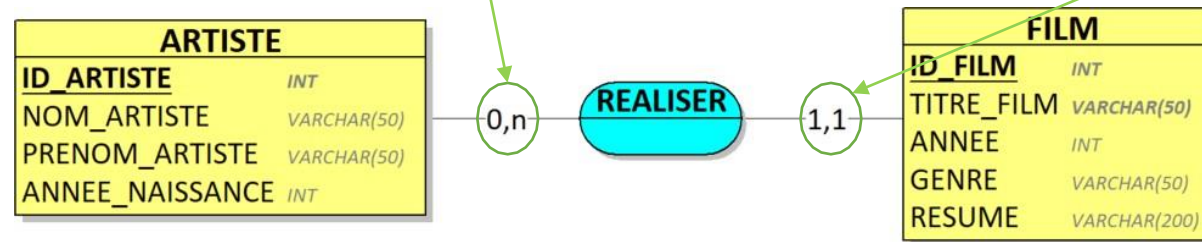


Cardinalité

Soit une association (E_1, E_2) entre deux types d'entités. La cardinalité de l'association pour E_i (E_1 ou E_2) est une paire [Min, Max] telle que :

- **Max** : cardinalité maximale qui désigne le nombre maximal de fois où une instance e_i de E_i peut intervenir dans l'association (1 ou n)
- **Min** : cardinalité minimale qui désigne le nombre minimal de fois où une instance e_i de E_i dans la relation (0 ou 1)

Exemple : Un artiste(MES) réalise **0 ou plusieurs film(s)** et un film est réalisé par **un et un seul artiste(MES)**.

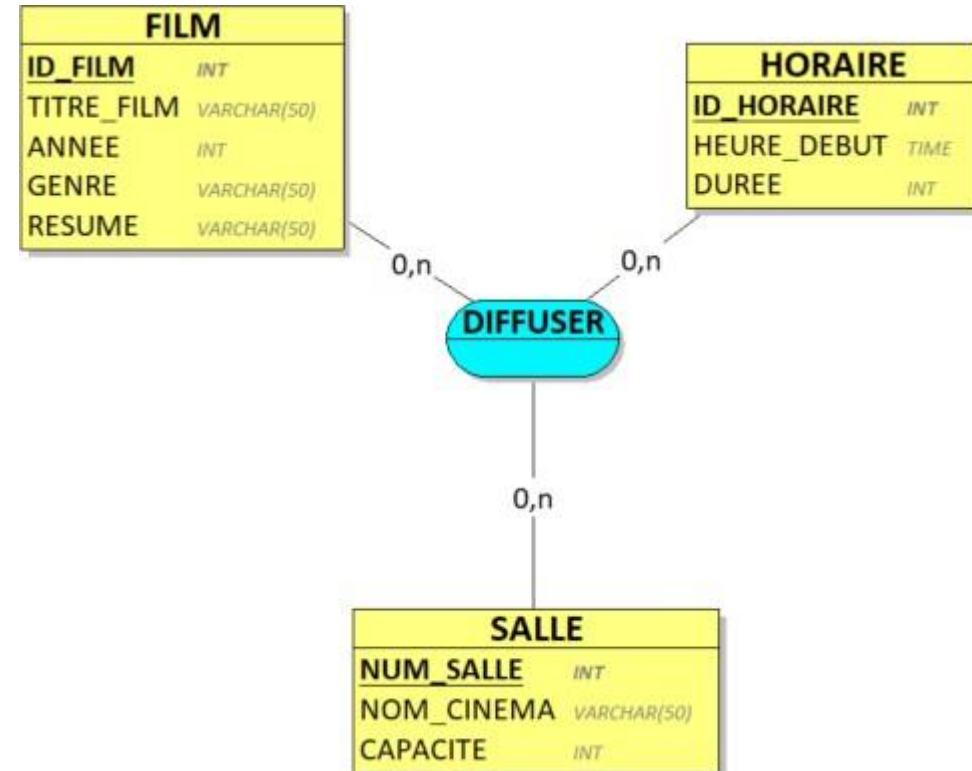


Association n-aire

Relation entre n types d'entités E_1, E_2, \dots, E_n est un ensemble de n -uplets (e_1, e_2, \dots, e_n) où chaque e_i appartient à E_i

Exemple :

- Un film est diffusé à 15h dans la salle numéro 5 et à 18h dans la salle numéro 7
- A 15h, il y aura la diffusion du film «Avengers» dans la salle 9 et du film « Iron man» dans la salle numéro 5.
- Dans la salle numéro 5, ils seront diffusés les films « Avengers » à 15h et « Iron man » à 17h.

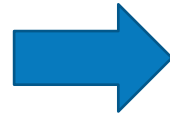


Quelques conseils

- **Normalisation des noms** : Le nom donné à une entité, une association ou un attribut doit être unique
 - Risque d'incohérence si des attributs portent le même nom mais également de redondance, de gaspillage de l'espace mémoire

Internaute	
<u>num</u>	INT
nom	VARCHAR(50)
prenom	VARCHAR(50)

Artiste	
<u>num</u>	INT
nom	VARCHAR(50)
prenom	VARCHAR(50)



Internaute	
<u>num_Internaute</u>	INT
nom_Internaute	VARCHAR(50)
prenom_Internaute	VARCHAR(50)

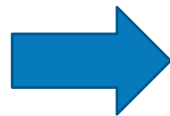
Artiste	
<u>num_Artiste</u>	INT
nom_Artiste	VARCHAR(50)
prenom_Artiste	VARCHAR(50)

Quelques conseils

- **Normalisation des attributs**

- Chaque attribut multiple (structure) doit être transformé en une entité et une association supplémentaire est rajoutée (exemple : plusieurs adresses pour un internaute)
- Idem pour une énumération (exemple pour Film, l'attribut « Genre » = {action, comédie, fiction,...})
- Eviter la redondance d'information des attributs par calcul ou transition : un attribut dérivé d'autres attributs (exemple : date de naissance et âge)

Internaute	
<u>num_Internaute</u>	INT
nom_Internaute	VARCHAR(50)
prenom_Internaute	VARCHAR(50)
Adresse	VARCHAR(50)



Internaute	
<u>num_Internaute</u>	INT
nom_Internaute	VARCHAR(50)
prenom_Internaute	VARCHAR(50)
Adresse	VARCHAR(50)

1,n

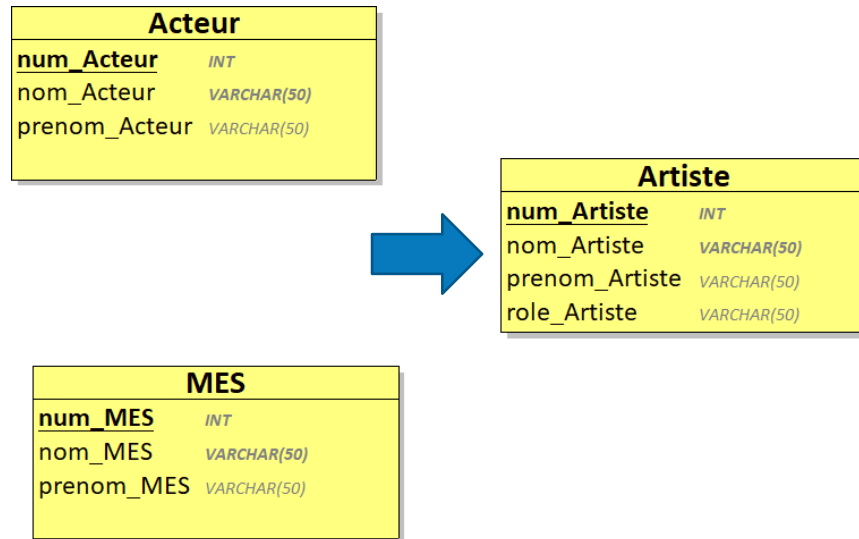


0,n

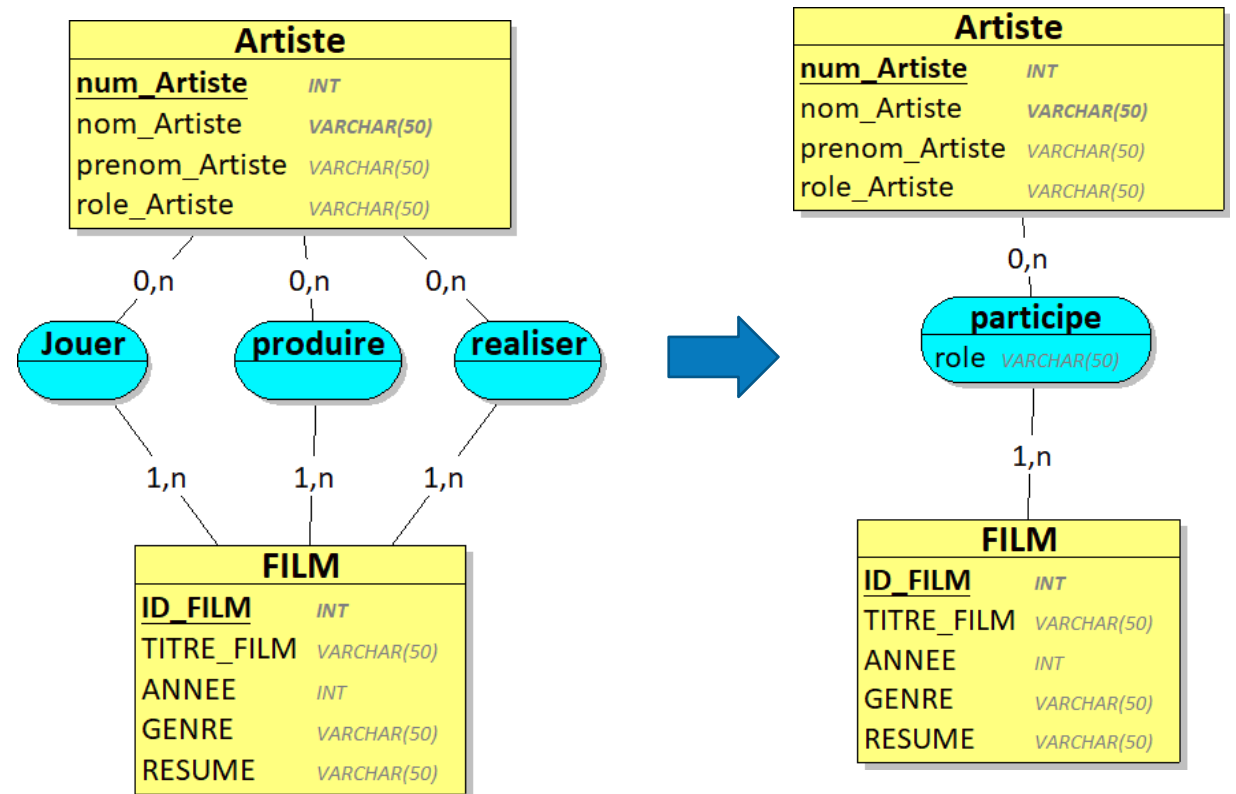
Adresse	
<u>num_adresse</u>	INT
num_rue	INT
nom_rue	VARCHAR(50)
CP	INT

Quelques conseils

- Faire des **fusions** quand c'est possible
- Il faut factoriser les entités et les associations **quand c'est possible**



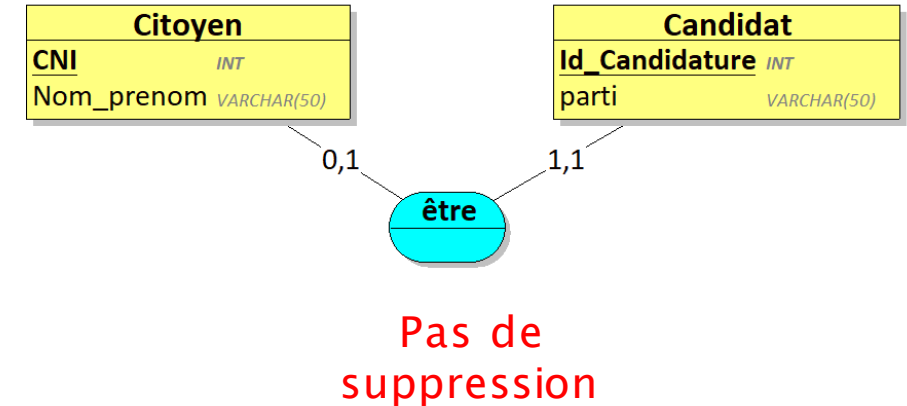
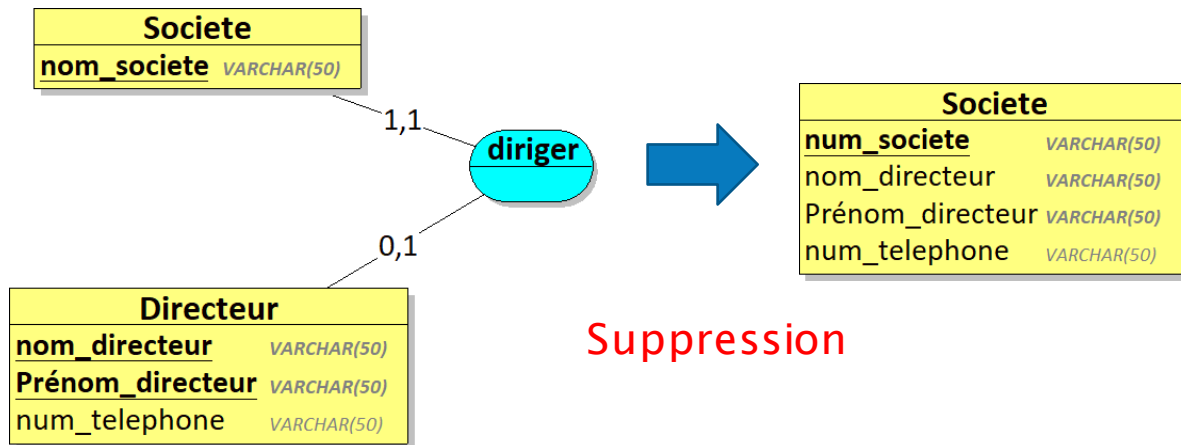
Fusion d'entités



Fusion d'associations

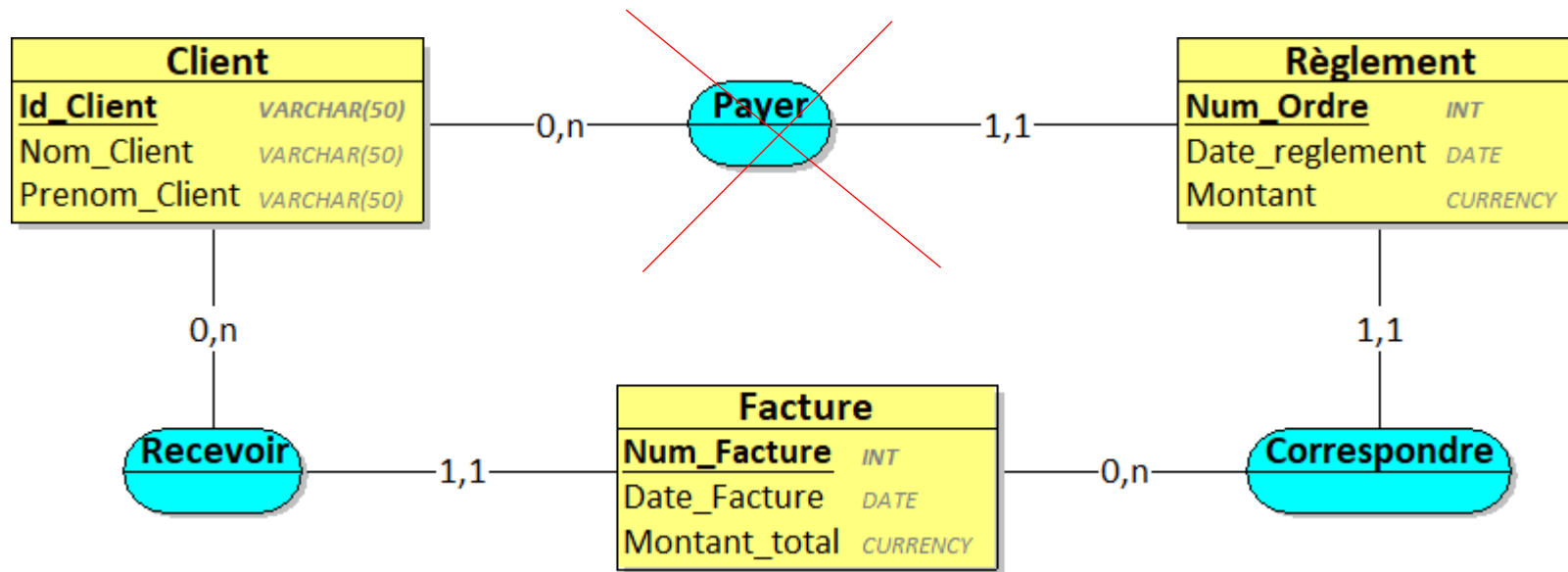
Quelques conseils

- Faire des **suppressions** quand c'est possible
 - Il faut se poser la question de l'intérêt d'une association quand les cardinalités maximales sont toutes égales à 1



Quelques conseils

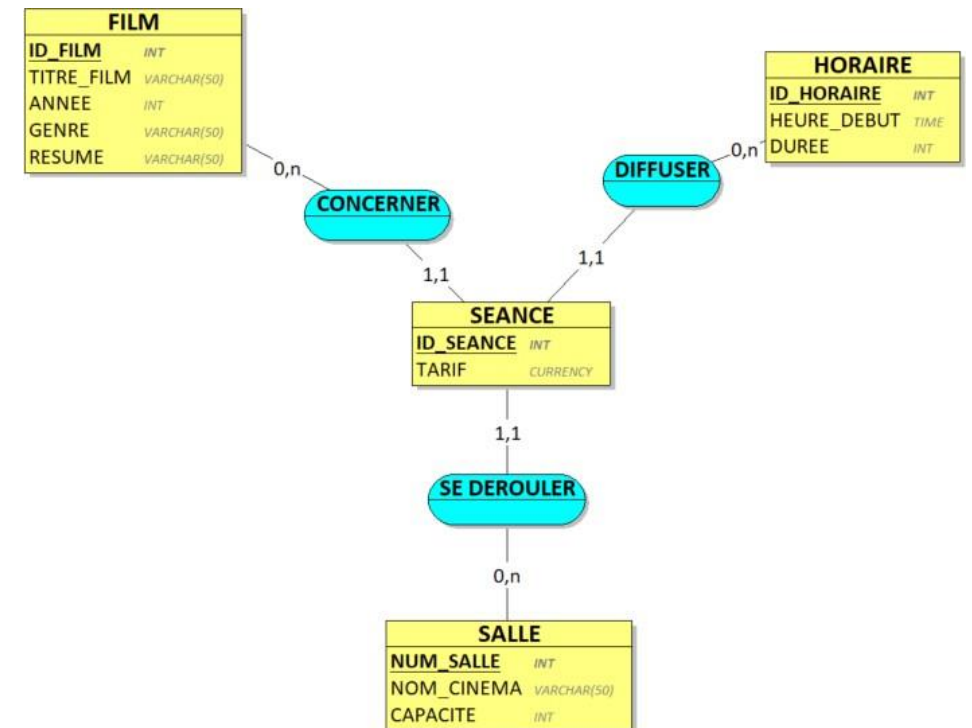
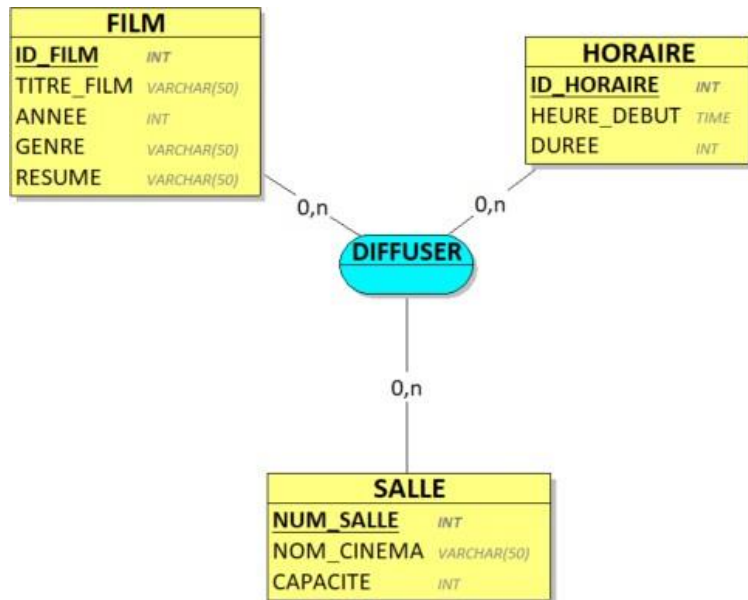
- S'il existe deux chemins pour se rendre d'une entité à une autre, alors ces deux chemins doivent avoir deux significations distinctes. exemple : si un client ne peut pas régler la facture d'un autre client, alors l'association payer est inutile.
- Dans le cas contraire, il faut supprimer le chemin le plus court puisqu'il est déductible des autres chemins



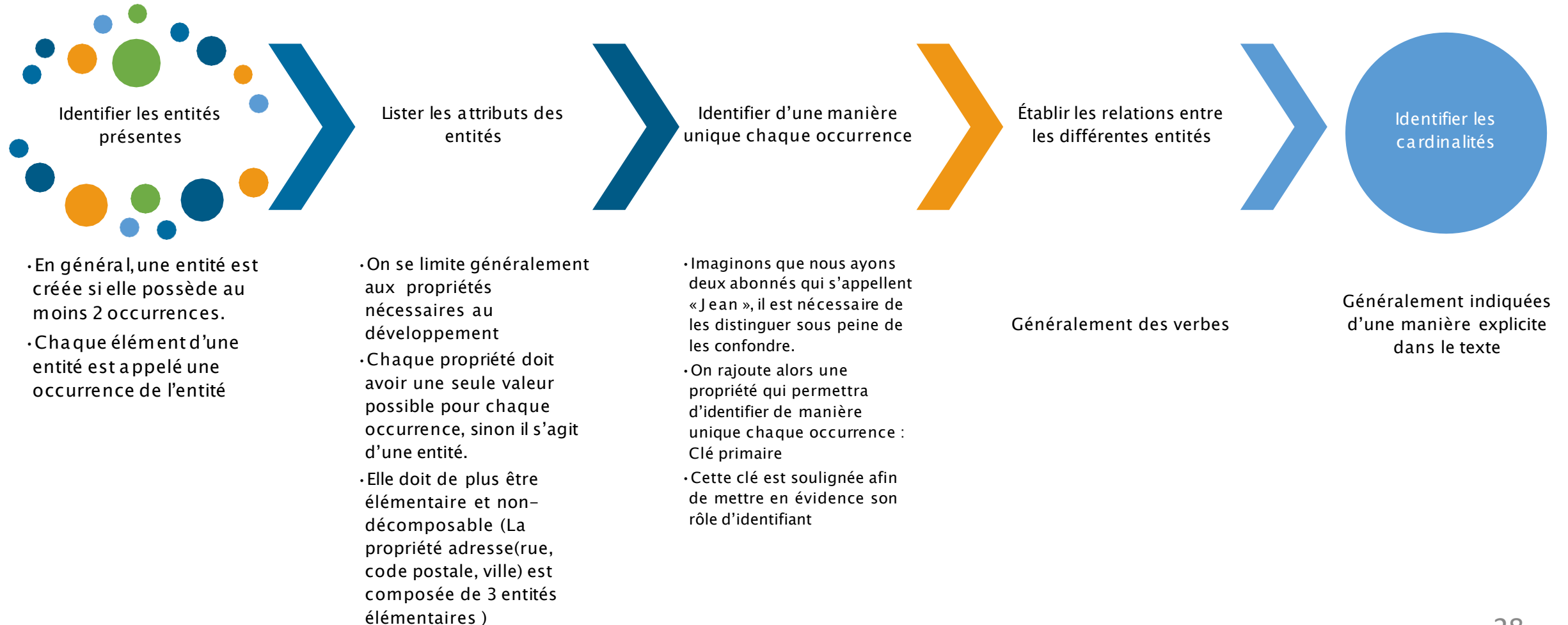
D'une association n-aire vers un type d'entité

Soit A une association entre les types d'entité $\{E_1, E_2, \dots, E_n\}$. La transformation de A en type d'entité s'effectue en trois étapes :

1. Attribution d'un identifiant autonome de A
2. Création d'une association A_i , de type (1..n) entre A et chacun des E_i
3. La contrainte minimale du coté de A est toujours à 1.



En bref



Application : Newsletter

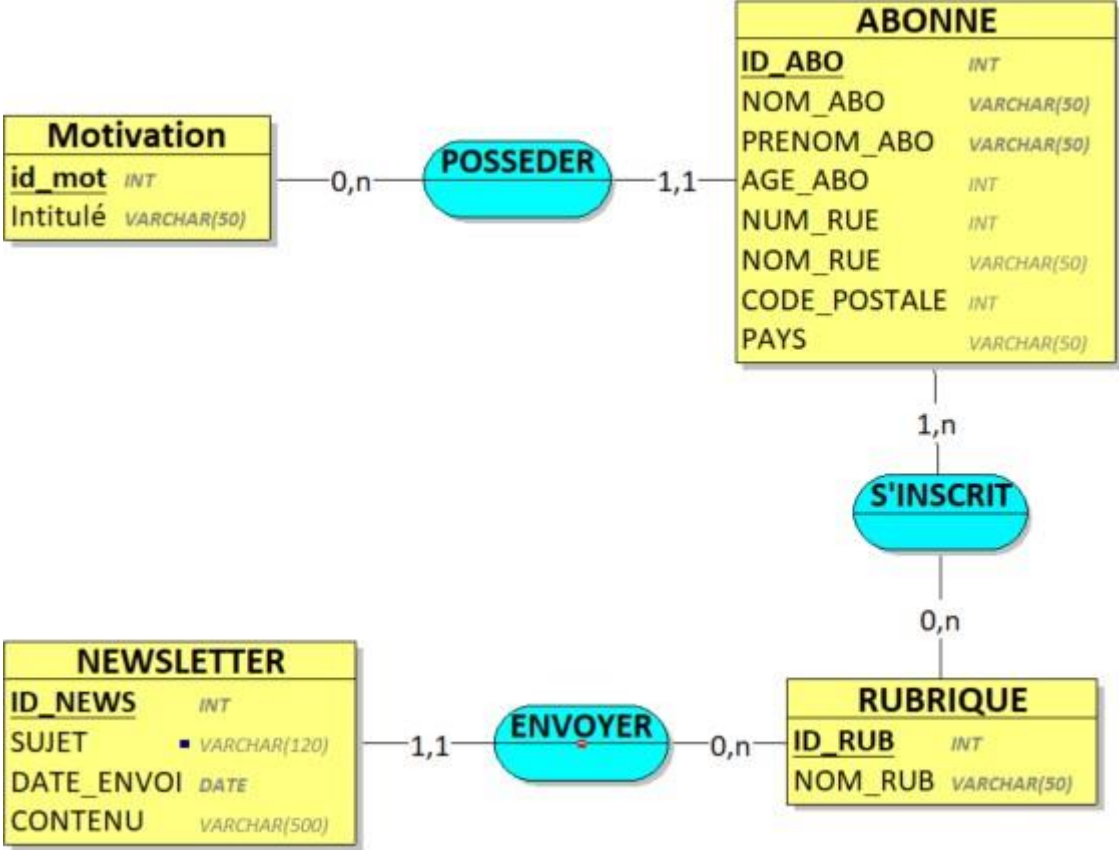
« Un abonné est inscrit à une ou plusieurs rubrique(s). Chaque rubrique envoie une Newsletter chaque semaine aux abonnés de la rubrique correspondante. Un abonné a une motivation d'inscription parmi plusieurs possibles. »

⇒ Ces phrases, si elles sont exactes et validées par le client, sont suffisantes pour concevoir notre premier modèle E/A. Elles contiennent en effet toutes les informations nécessaires.

Application : Newsletter

« Un abonné est inscrit à une ou plusieurs rubrique(s). Chaque rubrique envoie une NewsLetter chaque semaine aux abonnés de la rubrique correspondante. Un abonné a une motivation d'inscription parmi plusieurs possibles. »

Application : Newsletter



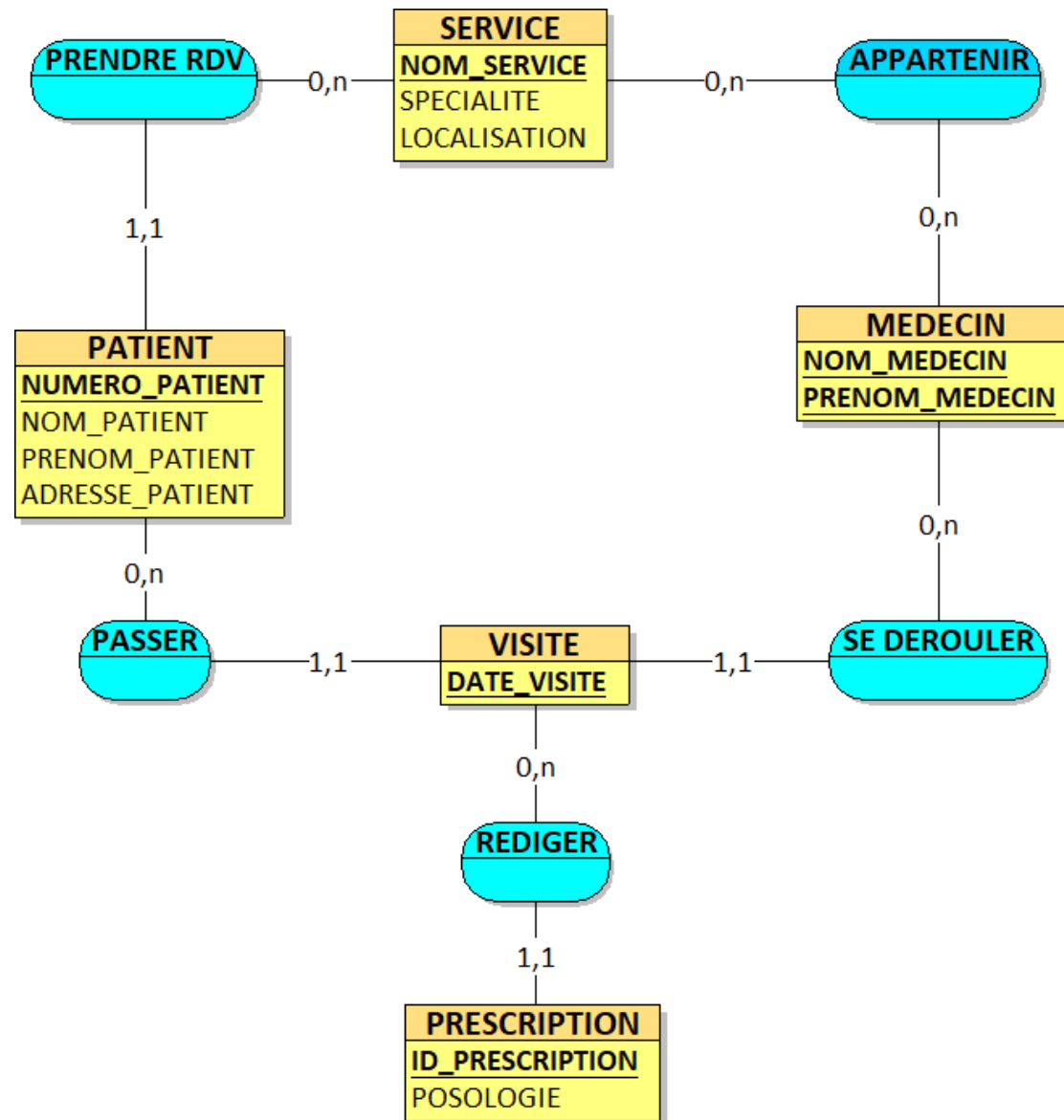
Application : Gestion d'un hôpital

- Les patients de l'hôpital sont répartis dans les services (caractérisés chacun par un nom identifiant, sa localisation, sa spécialité) de ce dernier.
- Chaque médecin appartient à un service. Il est identifié par son nom et son prénom.
- Un patient passe des visites.
- Chaque visite est effectuée chez un médecin à une date déterminée.
- Lors d'une visite, une ou plusieurs prescriptions peuvent être rédigée.
- Chaque prescription mentionne le nom d'un médicament et la posologie à respecter par le patient (posologie = simple ligne de texte).
- Un patient possède un numéro d'inscription (identifiant), un nom, un prénom et une adresse.

Application : Gestion d'un hôpital

- Les **patients** de l'hôpital sont **répartis** dans les **services** (caractérisés chacun par un **nom identifiant**, sa **localisation**, sa **spécialité**) de ce dernier.
- Chaque **médecin** **appartient** à un **service**. Il est **identifié** par son **nom** et son **prénom**.
- Un **patient** **pass**e des **visites**.
- Chaque **visite** est **effectuée** chez un **médecin** à une **date déterminée**.
- Lors d'une **visite**, une ou plusieurs **prescriptions** peuvent être **rédigée**.
- Chaque **prescription** **mentionne** le **nom d'un médicament** et la **posologie** à respecter par le **patient** (posologie = simple ligne de texte).
- Un **patient** possède un **numéro d'inscription** (identifiant), un **nom**, un **prénom** et une **adresse**.

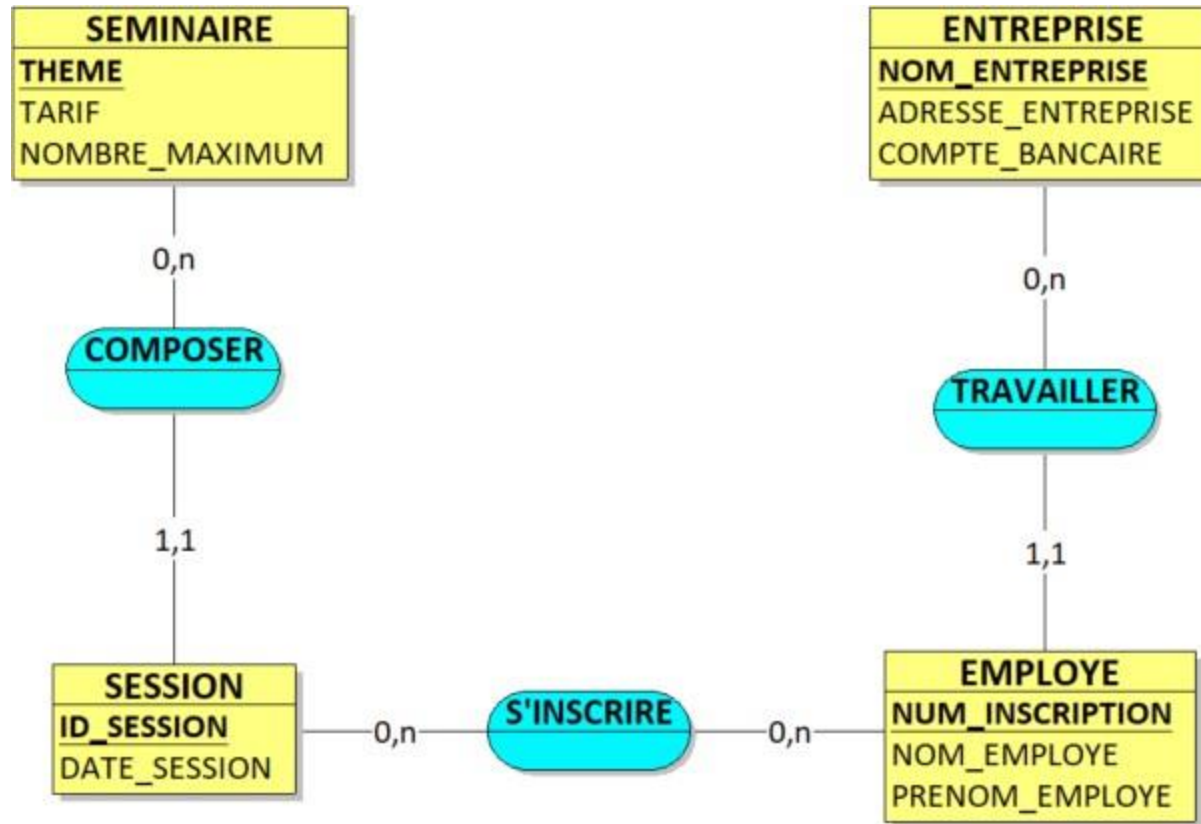
Application : Gestion d'un hôpital



Application : Société de formation

Une société de formation désire informatiser la gestion des inscriptions aux sessions qu'elle organise, ainsi que la facturation. Il existe un certain nombre de séminaires de formation, chacun consacré à un thème différent et facturé à un tarif déterminé. Un séminaire peut être organisé plus d'une fois, ce qui correspond à autant de sessions. Les sessions d'un séminaire se tiennent à des dates différentes. Des entreprises inscrivent certains de leurs employés à certaines sessions. Il existe un nombre maximum de participants pour les sessions de chaque séminaire (quelle que soit la date de la session).

Application : Société de formation



Application : Département informatique

On s'intéresse à la possible utilisation d'un système de gestion de base de données pour la gestion du département Informatique d'une école. En particulier, on veut pouvoir stocker :

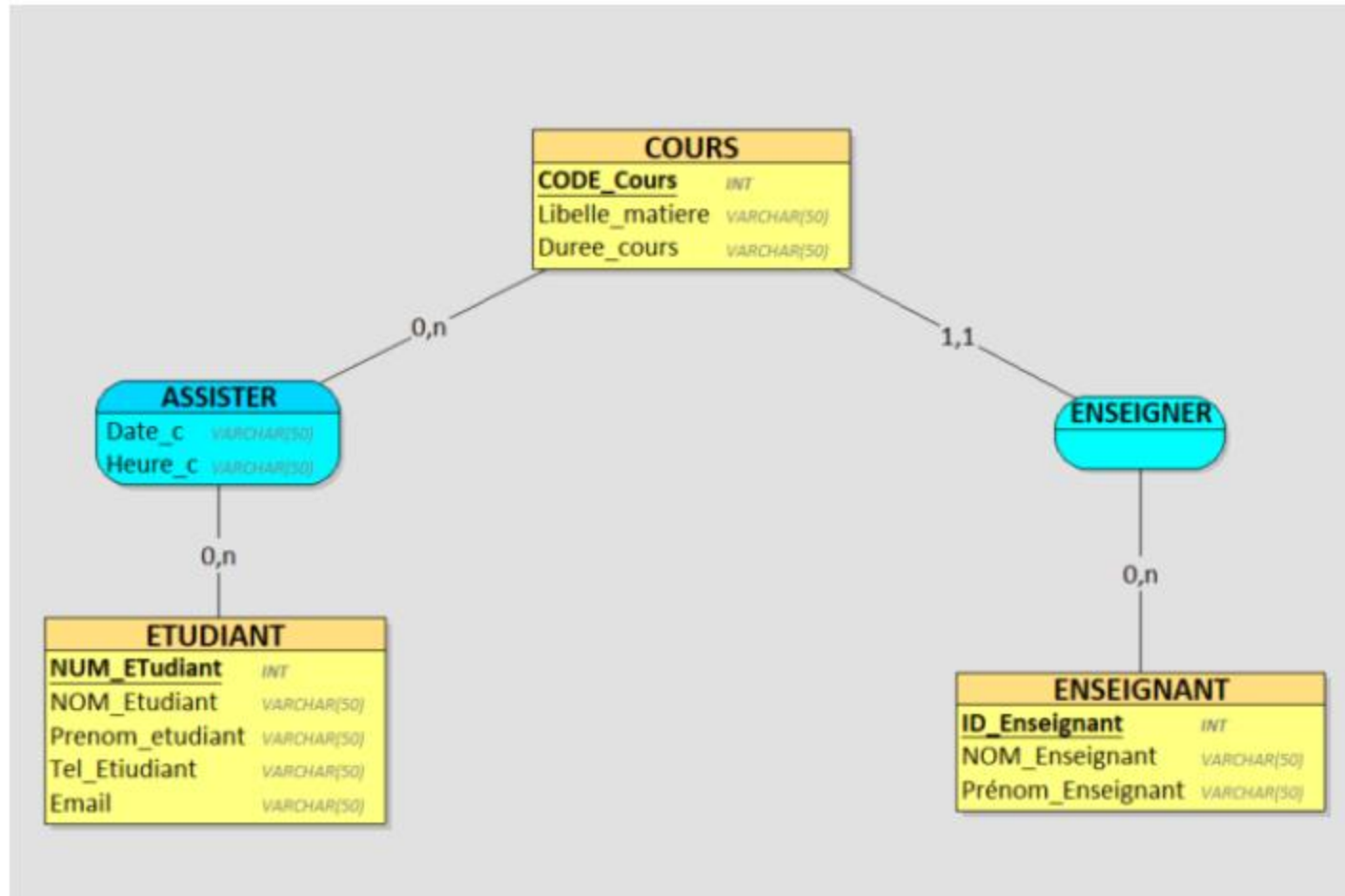
- L'ensemble des matières pouvant être dispensées,
- La liste des étudiants ainsi que celle des enseignants.

Cette base de données doit (entre autres) servir à gérer les cours.

- Un cours correspond à une matière enseignée par un enseignant (sinon, par convention, il s'agit d'un autre cours).
- Pour un cours, on veut pouvoir stocker des informations sur les absences des étudiants.
- Les étudiants peuvent être absents plusieurs fois pour le même cours, à des dates et horaires différents.

Créer un modèle E/A qui servira de base à un système d'information pour la gestion du département.

Application : Département informatique



E/A en résumé !



Simple et pratique :

- Il n'y a que 3 concepts : entités , associations et attributs
- Approprié à une représentation graphique intuitive, même s'il existe beaucoup de conventions.
- Permet de modéliser rapidement des structures pas trop complexes.



Non-déterminisme :

- Pas de règle absolue pour déterminer ce qui est entité, attribut ou relation.

Modèle relationnel

Références

Quelques livres utiles:

- Bases de données par Gardarin
- Introduction aux bases de données par Chris J. Date, traduit par Frédéric Cuppens
- SQL avancé par Joe Celko Thomson International Publishing.
- Bases de données et SGBD, de la conception à la mise en œuvre : démarche pratique. Kramarz Francis.
- SQL2 Initiation / programmation par Christian Marée et Guy Ledant
- Bases de données relationnelles : analyse et comparaison des systèmes. Gardarin et Valduriez.

