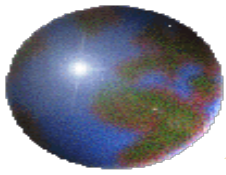


BASE DE DONNEES

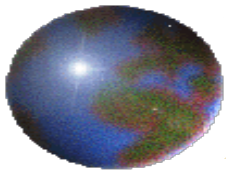
Chap 2. Le modèle relationnel

Dr. Coulibaly Tiékoura

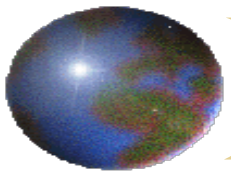


Plan du Chapitre 2

- ✚ **I. LES CONCEPTS**
- ✚ **II. LES DÉPENDANCES FONCTIONNELLES**
- ✚ **III. LES REGLES DE PASSAGE DU MODELE E-A AU MODELE
RELATIONNEL**
- ✚ **IV. LES RÈGLES D'INTÉGRITÉ**
- ✚ **V. LES FORMES NORMALES**



I. LES CONCEPTS



I. Les concepts

✚ LE DOMAINE

❑ ensemble de valeurs atomiques d'un certain type sémantique

Ex. : NOM_VILLE = { Nice, Paris, Rome }

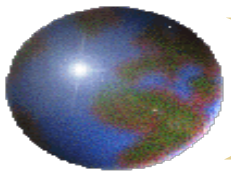
❑ les domaines sont les ensembles de valeurs possibles dans lesquels sont puisées les données

❑ deux ensembles peuvent avoir les mêmes valeurs bien que sémantiquement distincts

Ex. :

NUM_ELV = { 1, 2, ... , 2000 }

NUM_ANNEE = { 1, 2, ... , 2000 }



I. Les concepts

✚ LA RELATION

= sous ensemble du produit cartésien de plusieurs domaines

$R \subset D1 \times D2 \times \dots \times Dn$ avec $D1, D2, \dots, Dn$ sont les domaines de R et n est le degré ou l'arité de R

Ex:

Les domaines :

NOM_ELV = { dupont, durant }

PREN_ELV = { pierre, paul, jacques }

DATE_NAISS = {Date entre 1/1/1990 et 31/12/2020}

NOM_SPORT = { judo, tennis, foot }

La relation ELEVE

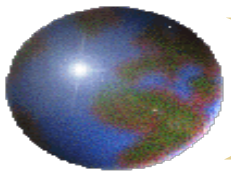
$ELEVE \subset NOM_ELV \times PREN_ELV \times DATE_NAISS$

$ELEVE = \{ (dupont, pierre, 1/1/1992), \\ (durant, jacques, 2/2/1994) \}$

La relation INSCRIPT

$INSCRIPT \subset NOM_ELV \times NOM_SPORT$

$INSCRIPT = \{ (dupont, judo), (dupont, foot), \\ (durant, judo) \}$



I. Les concepts

✚ LES N-UPLETS

= un élément d'une relation est un **n-uplet** de valeurs (tuple en anglais)

- un n-uplet représente un fait

Ex.: « Dupont pierre est un élève né le 1 janvier 1992 »
« dupont est inscrit au judo »

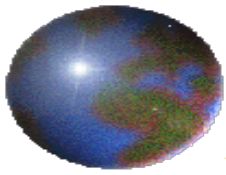
• DEFINITION PRÉDICATIVE D'UNE RELATION

Une relation peut être considérée comme un PRÉDICAT
à n variables

$$\theta(x, y, z) \text{ vrai} \Leftrightarrow (x, y, z) \in R$$

Ex. :

$$\text{est_inscrit}(\text{dupont}, \text{judo}) \Leftrightarrow (\text{dupont}, \text{judo}) \in \text{INSCRIPT}$$



I. Les concepts

✚ LES ATTRIBUTS

Chaque composante d'une relation est un attribut

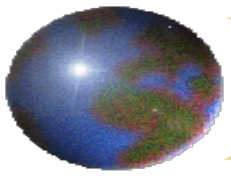
- Le nom donné à un attribut est porteur de sens
- Il est en général différent du nom de domaine
- Plusieurs attributs peuvent avoir le même domaine

Ex. :

La relation **TRAJET** :

$\text{TRAJET} \subset \text{NOM_VILLE} \times \text{NOM_VILLE}$

Dans laquelle la première composante représente la ville de départ VD, la deuxième composante la ville d'arrivée VA d'un trajet.



I. Les concepts

✚ LE SCHÉMA D'UNE RELATION

Le schéma d'une relation est défini par :

- le nom de la relation
- la liste de ses attributs

on note : $R (A1, A2, \dots, An)$

Ex.:

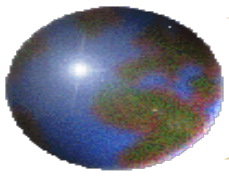
ELEVE (NOM, PRENOM, NAISS)

INSCRIPT (NOM_ELV, SPORT)

TRAJET (VD, VA)

• Extension

- L'extension d'une relation correspond à l'ensemble de ses éléments (n-uplets)
→ le terme RELATION désigne une extension



I. Les concepts

✚ LE SCHÉMA D'UNE BDR

Le schéma d'une base de données est défini par :

- l'ensemble des schémas des relations qui la composent

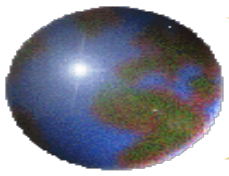
□ Différence entre :

- **le schéma de la BDR** qui dit comment les données sont organisées dans la base
- **l'ensemble des n-uplets de chaque relation**, qui représentent les données stockées dans la base

✚ LA REPRÉSENTATION

□ **1 RELATION = 1 TABLE**

U1	V1	W1	X1	Y1
U2	V2	W2	X2	Y2
U3	V3	W3	X3	Y3



I. Les concepts

LA REPRÉSENTATION

❑ 1 ÉLÉMENT ou n-uplet = 1 LIGNE

* une relation est un ensemble \Rightarrow on ne peut pas avoir 2 lignes identiques

LIGNE \rightarrow
1 élément

U1	V1	W1	X1	Y1

❑ 1 ATTRIBUT = 1 COLONNE

U1				
U2				
U3				

\uparrow
COLONNE

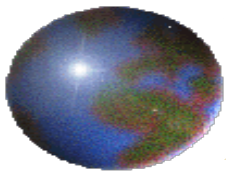
1 attribut ou propriété

❑ EXEMPLE: La relation ELEVE

ELEVE :

élément \rightarrow

NOM	PRENOM	NAISS
dupont	Pierre	1/1/1992
durant	Jacques	2/2/1994
duval	Paul	3/03/81



I. Les concepts

✦ Fenêtre Création de Table sur MySQL (PhpMyAdmin)

phpMyAdmin

Serveur courant : MySQL

Récentes Préférées

- Nouvelle base de données
- atomes
- commande
- departements_france
- ecole
- email_jetable
- etablissement
- film
- fournisseur
- information_schema
- inscription
- mabase_test
- mysql
- pays
- performance_schema
- sys
- test_laravel
 - Nouvelle table
 - classe
 - etudiant

Serveur: MySQL:3308 » Base de données: test_laravel » Table: étudiants

Parcourir Structure SQL Rechercher Insérer Exporter Importer Privilèges Opérations Déclencheurs

Nom de table: étudiants Ajouter 1 colonne(s) Exécuter

Nom	Type	Taille/Valeurs*	Valeur par défaut	Interclassement	Attributs	Null	Index	A_I
	INT		Aucun(e)			<input type="checkbox"/>	--	<input type="checkbox"/>
	INT		Aucun(e)			<input type="checkbox"/>	--	<input type="checkbox"/>
	INT		Aucun(e)			<input type="checkbox"/>	--	<input type="checkbox"/>
	INT		Aucun(e)			<input type="checkbox"/>	--	<input type="checkbox"/>

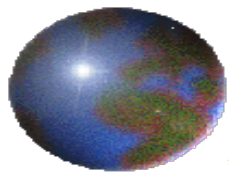
Commentaires de table : Interclassement : Moteur de stockage : MyISAM

Définition de PARTITION :

Partitionner par : (Expression ou liste de co)

Partitions :

Aperçu SQL Enregistrer



I. Les concepts

✚ Affichage d'une table sur MySQL (PhpMyAdmin)

← Serveur: MySQL:3308 » Base de données: etablissement » Table: etudiants

Parcourir

Structure

SQL

Rechercher

Insérer

Exporter

Importer

Privileges

↔T↔

▼

id

nom

prenom

created_at

updated_at

filiere_id

☐

✎ Éditer

✚ Copier

⊖ Supprimer

1

KOFFI

Moussa

2022-03-09 09:42:10

2022-03-09 10:25:36

14

☐

✎ Éditer

✚ Copier

⊖ Supprimer

2

AYA

Virginie

2022-03-09 09:44:39

2022-03-09 10:26:00

13

☐

✎ Éditer

✚ Copier

⊖ Supprimer

3

KOUAO

Yves

2022-03-09 09:50:24

2022-03-09 09:50:24

1

☐

✎ Éditer

✚ Copier

⊖ Supprimer

6

KOUAKOU

Kouamé Yannick

2022-03-10 10:37:09

2022-03-10 10:37:09

5

☐

✎ Éditer

✚ Copier

⊖ Supprimer

5

Diakite

Akissi Solange

2022-03-09 09:52:00

2022-03-09 09:52:00

11

↑

☐ Tout cocher

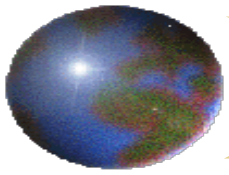
Avec la sélection :

✎ Éditer

✚ Copier

⊖ Supprimer

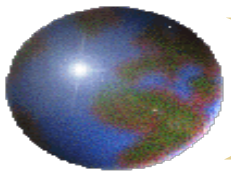
📄 Exporter



II

LES DÉPENDANCES

FONCTIONNELLES



II. les dépendances fonctionnelles

✚ Dépendance fonctionnelle

Soit $R(A_1, A_2, \dots, A_n)$ un schéma de relation. Soit X et Y des sous ensembles de $\{A_1, A_2, \dots, A_n\}$

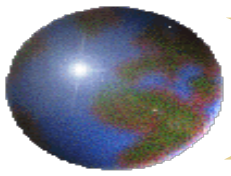
On dit que Y dépend fonctionnellement de X ($X \rightarrow Y$) si à chaque valeur de X correspond une valeur unique de Y .

- on écrit : $X \rightarrow Y$
- on dit que : X détermine Y

❖ Ex.:

PRODUIT (no_prod, nom, prixUHT)	:	no_prod \rightarrow (nom, prixUHT)
NOTE (no_contrôle, no_élève, note)	:	(no_contrôle, no_élève) \rightarrow note

- une dépendance fonctionnelle est une propriété **sémantique**, elle correspond à une contrainte supposée toujours vrai du monde réel



II. les dépendances fonctionnelles

✚ La clé d'une relation

Attribut (ou groupe minimum d'attributs) qui détermine tous les autres.

❑ Ex.:

PRODUIT (**no_prod**, nom, prixUHT)

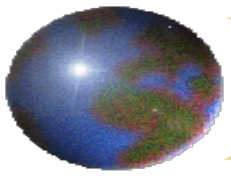
no_prod → (nom, prixUHT)

no_prod est une clé

- Une clé détermine un n-uplet de façon unique
- Pour trouver la clé d'une relation, il faut examiner attentivement les **hypothèses sur le monde réel**
- Une relation peut posséder plusieurs clés, on les appelle clés candidates

❑ Ex.:

dans la relation PRODUIT, nom est une clé candidate (à condition qu'il n'y ait jamais 2 produits de même nom)



II. les dépendances fonctionnelles

✚ Clé primaire

Choix d'une clé parmi les clés candidates.

✚ Clé étrangère ou clé secondaire

Attribut (ou groupe d'attributs) qui fait référence à la clé primaire d'une autre relation

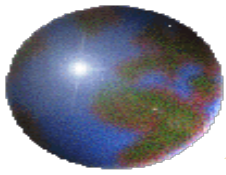
Ex.:

CATEG (**no_cat**, design, tva)

PRODUIT(no_prod, nom, marque, no_cat, prixUHT)

no_cat dans PRODUIT est une clé étrangère

❑ **CLÉ ÉTRANGÈRE** = CLÉ PRIMAIRE dans une autre relation

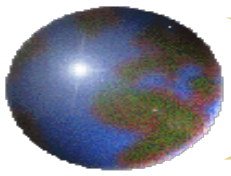


III

REGLES DE PASSAGE

MODELE E-A/ MODELE

RELATIONNEL



III. Règles de passage modèle E-A/ Modèle relationnel

✚ Règle de passage: entités

- Il possède une seule structure qui est la relation
- Une relation est identifiée par son nom.
- Chaque relation se compose d'un ensemble de colonnes désignés par les attributs.
- Un modèle relationnel est constitué par un ensemble de relations.

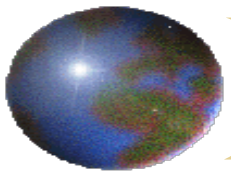
❑ Le schéma d'une relation est constitué du nom de la relation, suivi des attributs.

❑ Pour chaque entité du schéma E/A:

- On crée une relation de même nom que l'entité.
- Chaque propriété de l'entité, y compris l'identifiant, devient un attribut de la relation.
- Les attributs de l'identifiant constituent la clé de la relation.



DEPARTEMENT(Num_département, nom_département, localisation)

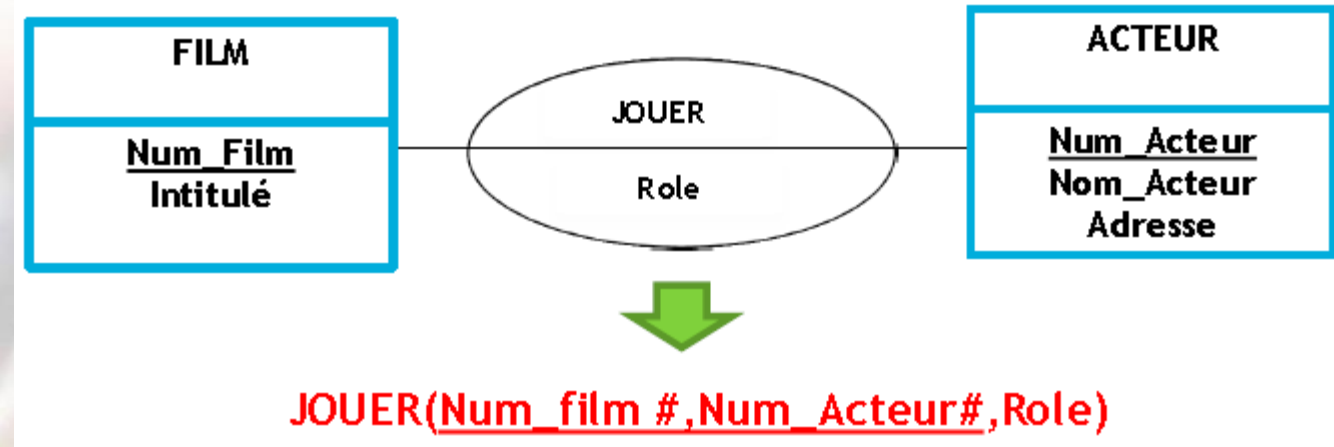


III. Règles de passage modèle E-A/ Modèle relationnel

✚ Règle de passage: associations

⦿ Toute classe d'association est transformée en relation.

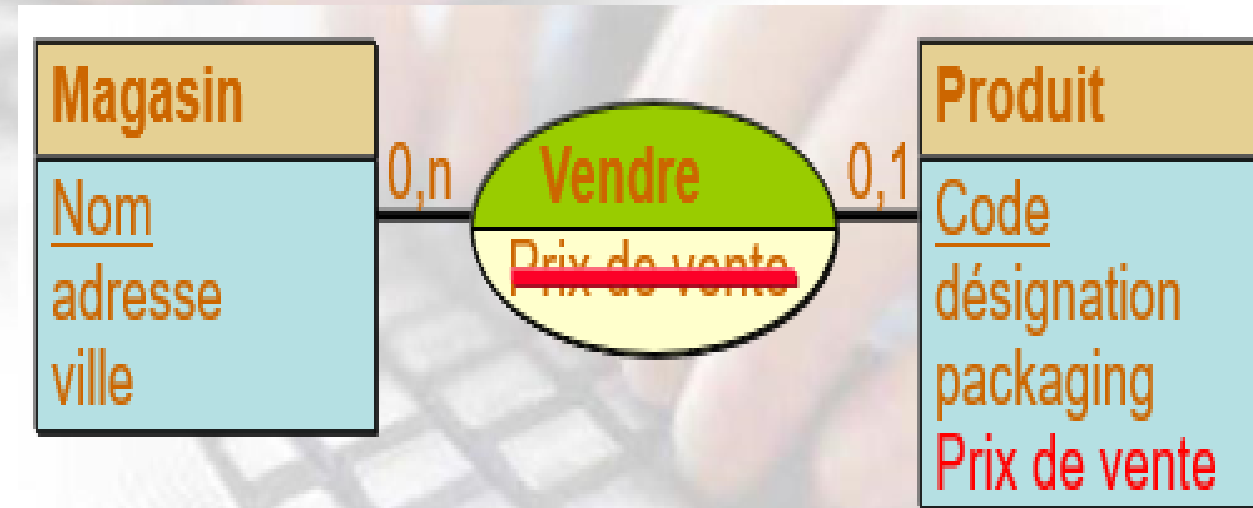
- La clé de cette relation est composée de tous les identifiants des entités participantes.

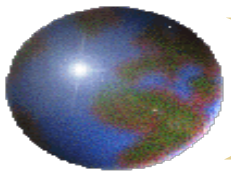


✚ Règle d'optimisation

⦿ Toute association reliée à une entité avec une cardinalité de type **0,1** ou **1,1** peut être fusionnée avec l'entité la reliant.

⦿ Nous devons déplacer les attributs de l'association vers ceux de la relation traduisant l'entité.

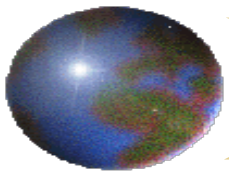




III. Règles de passage modèle E-A/ Modèle relationnel

✚ Règle de passage: associations de 1 à 1 ou de 1 à plusieurs

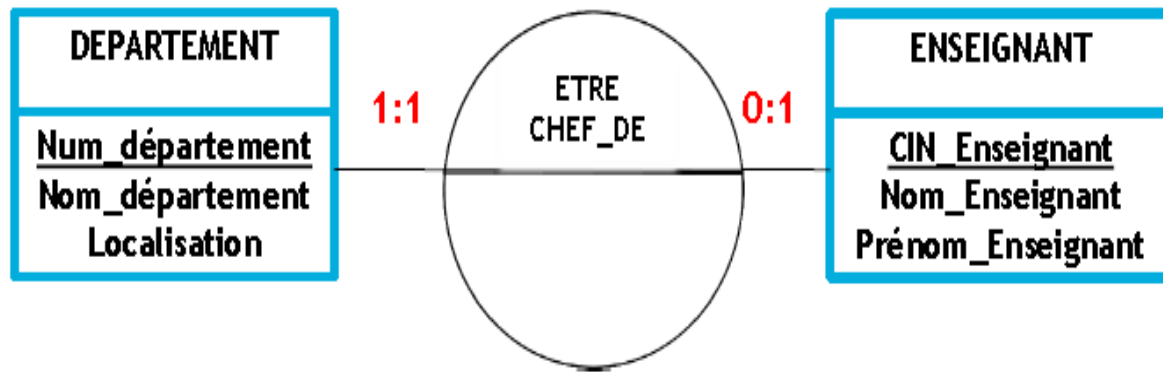
- ◎ Soit une association de un à plusieurs ou de un à un entre A et B. Le passage au modèle logique suit les règles suivantes :
 - On crée les relations Ra et Rb correspondant respectivement aux entités A et B.
 - L'identifiant de B devient un attribut de Ra.
- ◎ Une occurrence de A référence l'occurrence de B qui lui est associée à l'aide d'une *clé étrangère*.
- ◎ Cette référence se fait de manière unique et suffisante à l'aide de l'identifiant.



III. Règles de passage modèle E-A/ Modèle relationnel

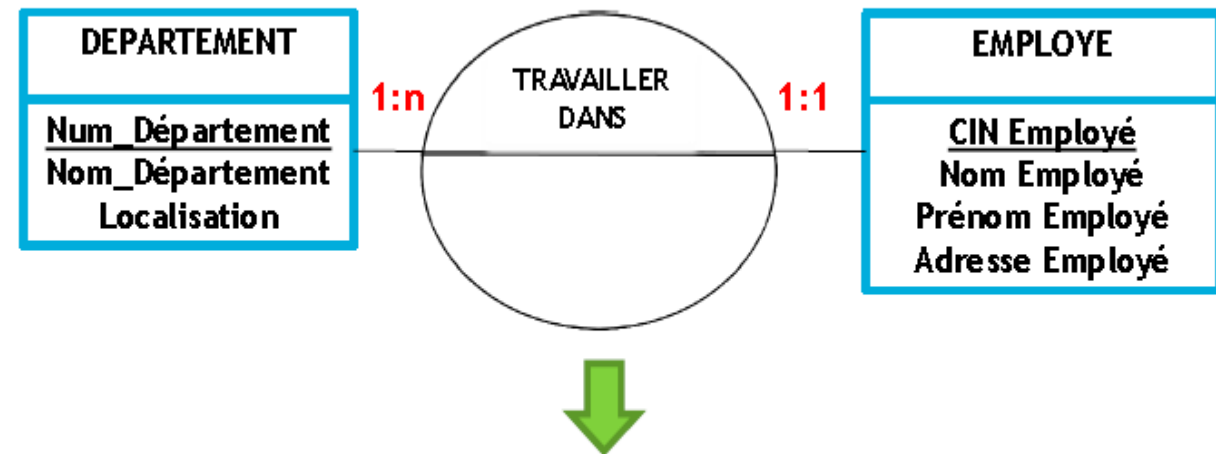
✿ Règle de passage: associations de 1 à 1 ou de 1 à plusieurs

ASSOCIATIONS DE 1 À 1

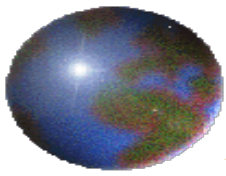


DEPARTEMENT(Num_département, nom_département, localisation
, CIN_Enseignant#)
ENSEIGNANT(CIN_enseignant, Nom_Enseignant, prénom_Enseignant

ASSOCIATIONS DE 1 À PLUSIEURS

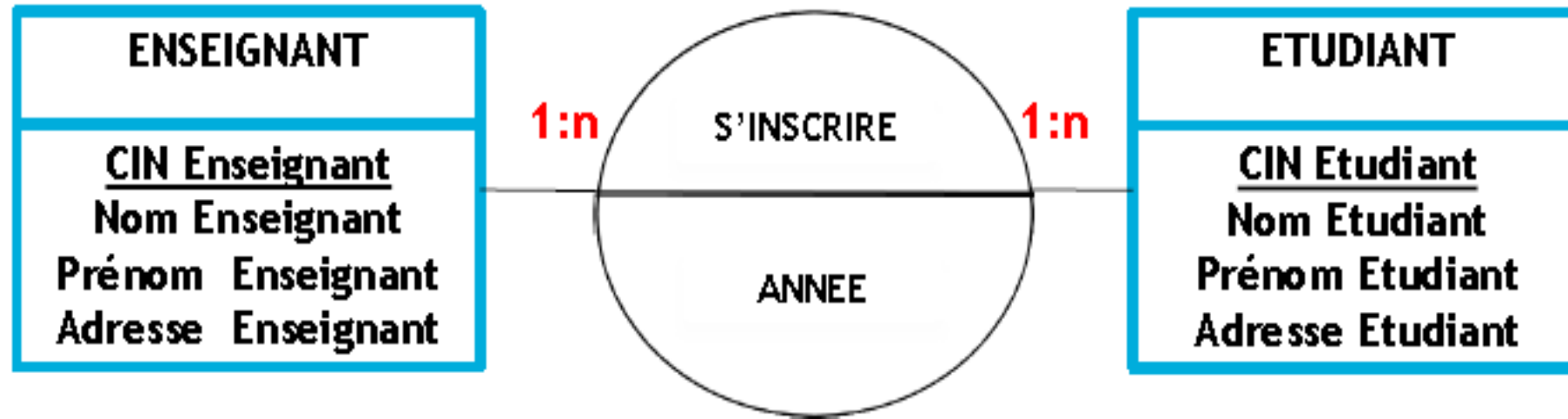


DEPARTEMENT(Num_Département, Nom_Département, Localisation)
EMPLOYE(CIN_Employé, Nom_Employé, Prénom_Employé,
Adresse_Employé, Num_département#)



III. Règles de passage modèle E-A/ Modèle relationnel

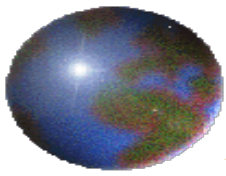
✦ Règle de passage: associations de plusieurs à plusieurs



ENSEIGNANT(CIN_Enseignant, Nom_Enseignant, Prénom_Enseignant, Adresse_Enseignant)

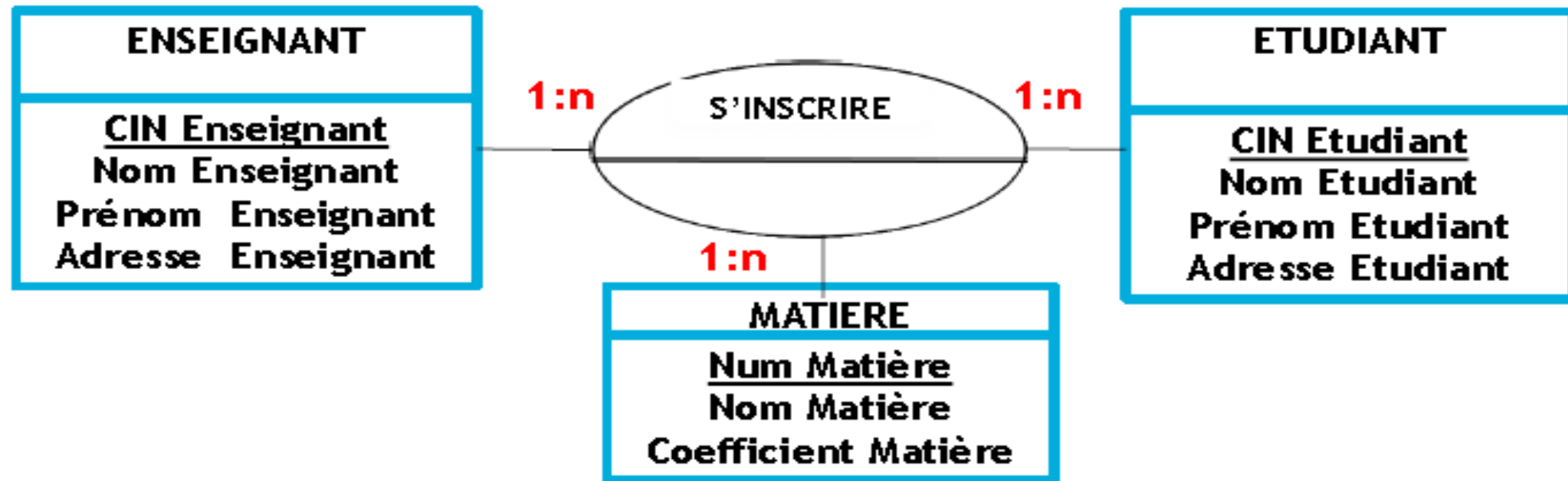
ETUDIANT(CIN_Etudiant, Nom_Etudiant, Prénom_Etudiant, Adresse_Etudiant)

S'INSCRIRE(CIN_Enseignant#, CIN_Etudiant #, Année)



III. Règles de passage modèle E-A/ Modèle relationnel

✚ Règle de passage: associations ternaires

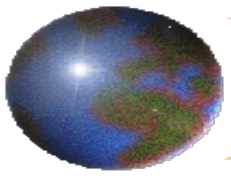


ENSEIGNANT(CIN_Enseignant, Nom_Enseignant, Prénom_Enseignant, Adresse_Enseignant)

ETUDIANT(CIN_Etudiant, Nom_Etudiant, Prénom_Etudiant, Adresse_Etudiant)

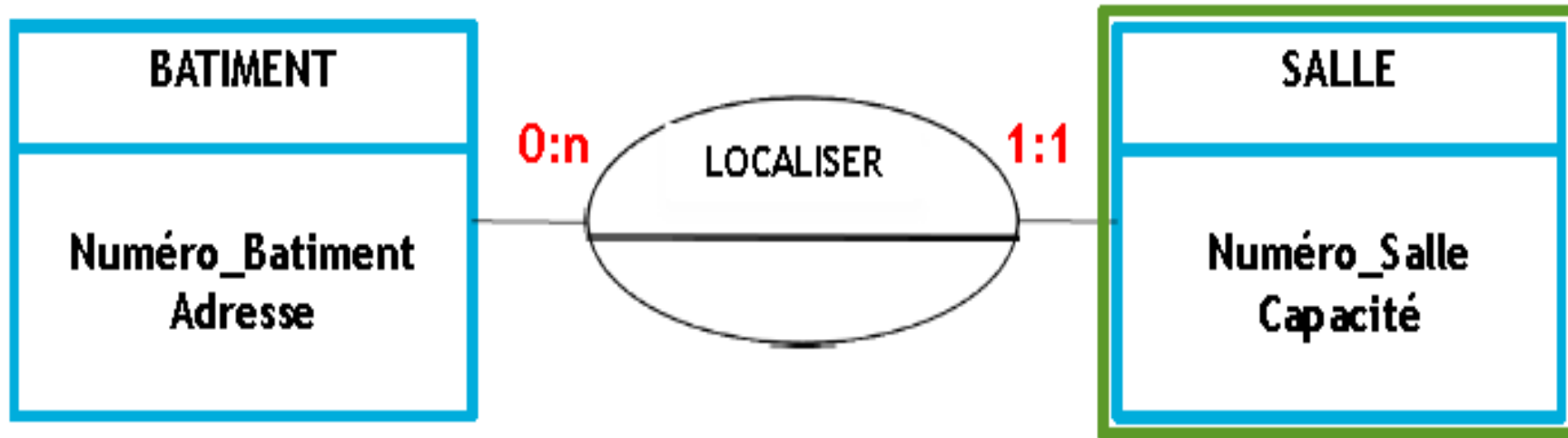
MATIERE (Num_Matière, Nom_Matière, Coefficient_Matière)

S'INSCRIRE(CIN_Enseignant#, CIN_Etudiant #, Num_Matière#, Année)



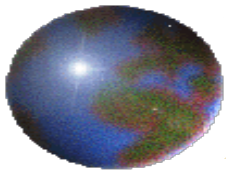
III. Règles de passage modèle E-A/ Modèle relationnel

✚ Règle de passage: associations avec entité faible

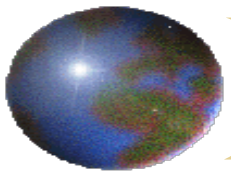


BATIMENT (Numéro_Batiment, adresse)

SALLE (Numéro_Batiment#, Numéro_Salle, capacité)



IV *LES RÈGLES* *D'INTÉGRITÉ*



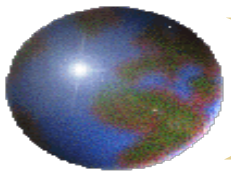
IV. Les règles d'intégrité

Les règles d'intégrité sont les assertions qui doivent être vérifiées par les données contenues dans une base.

Le modèle relationnel impose les contraintes structurelles suivantes :

- ❑ **INTÉGRITÉ DE DOMAINE**
- ❑ **INTÉGRITÉ DE CLÉ**
- ❑ **INTÉGRITÉ RÉFÉRENCIELLE**

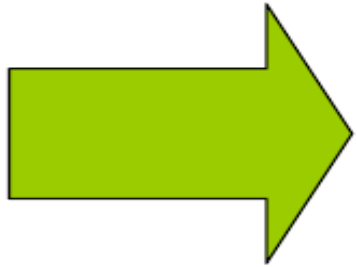
- La gestion automatique des contraintes d'intégrité est l'un des outils les plus importants d'une base de données.
- Elle justifie à elle seule l'usage d'un SGBD.



IV. Les règles d'intégrité

✚ INTÉGRITÉ DE DOMAINE

Les valeurs d'une colonne de relation doivent appartenir au domaine correspondant.

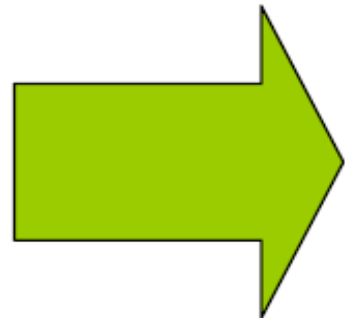


- contrôle des valeurs des attributs
- contrôle entre valeurs des attributs

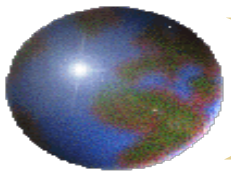
✚ INTÉGRITÉ DE CLÉ

Les valeurs de clés primaires doivent être :

- uniques
- non NULL



- Unicité de clé
- Unicité des n-uplets



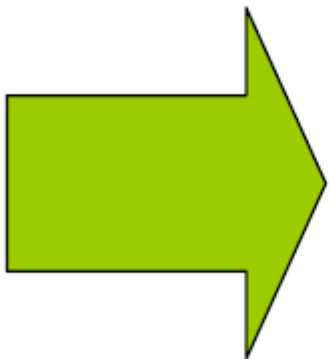
IV. Les règles d'intégrité

- **Valeur NULL:** valeur conventionnelle pour représenter une information **inconnue**
- dans toute extension possible d'une relation, il ne peut exister 2 n-uplets ayant même valeur pour les attributs clés.

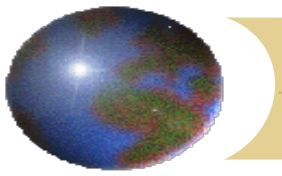
Sinon 2 clés identiques détermineraient 2 lignes identiques (d'après la définition d'une clé), ce qui est absurde

INTÉGRITÉ RÉFÉRENCIELLE

Les valeurs de clés étrangères sont 'NULL' ou sont des valeurs de la clé primaire auxquelles elles font référence



- Relations dépendantes



IV. Les règles d'intégrité

• LES DÉPENDANCES :

Liaisons de un à plusieurs exprimées par des attributs particuliers: **clés étrangères** ou **clés secondaires**

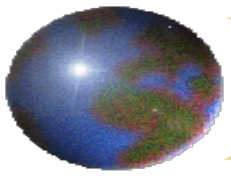
❑ Remarque: *Les contraintes de référence ont un impact important pour les opérations de mises à jour, elles permettent d'éviter les **anomalies** de mises à jour*

Exemple :

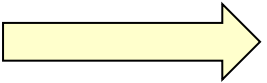
CLIENT (**no_client**, nom, adresse)

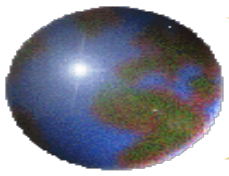
ACHAT (**no_produit**, **no_client**, date, qte)

Clé étrangère **no_client** dans ACHAT



IV. Les règles d'intégrité

- **insertion** tuple **no_client = X** dans **ACHAT**
  vérification si **X** existe dans **CLIENT**
- **suppression** tuple **no_client = X** dans **CLIENT**
 - ➔ soit interdire si **X** existe dans **ACHAT**
 - ➔ soit supprimer en cascade tuple **X** dans **ACHAT**
 - ➔ soit modifier en cascade **X = NULL** dans **ACHAT**
- **modification** tuple **no_client = X** en **X'** dans **CLIENT**
 - ➔ soit interdire si **X** existe dans **ACHAT**
 - ➔ soit modifier en cascade **X** en **X'** dans **ACHAT**



IV. Les règles d'intégrité

✚ Paramétrage des Relations dans MySQL (Ajouter une contrainte d'intégrité référentielle)

Syntaxe MySQL

```
ALTER TABLE nom_table_avec_cle_etrangere
ADD CONSTRAINT nom_contrainte
FOREIGN KEY (cle_etrangere)
REFERENCES table_referencee(cle_primaire);
```

On entrera la requête suivante :

MySQL

```
ALTER TABLE voitures
ADD CONSTRAINT fk_voitures_clients
FOREIGN KEY (client_id)
REFERENCES clients (id);
```

Par exemple, pour appliquer une contrainte d'intégrité référentielle à la table voitures :

id	code	prenom	nomfamille	adresse	ville	telephone
1	COURJACI	Jacinthe	Courtois	23, des Chênes	Victoriaville	819-751-1111
2	FRENMARC	Marc	Frenette	4, Notre-Dame Ouest	Victoriaville	(819) 752-2222
3	GAGNLOUI	Louis	Gagnon	101, Boul. Jutras	Victoriaville	819-758-8888
4	DARGMARC	Marco	D'Argy	266 Principale	Warwick	819-353-3333
5	JACQLINE	Line	Jacques	424 St-Jean-Baptiste	Princeville	819-364-4444
6	BELLMART	Martin	Bellemare	1253, Notre-Dame Est	Victoriaville	819-751-3333

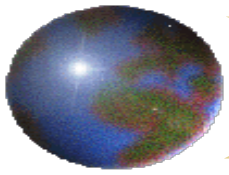
clients

Clé primaire

id	client_id	modele_id	annee
1	3	2	2008
2	1	3	2006
3	1	10	2011
4	2	10	2007
5	4	3	2010
6	4	8	2009

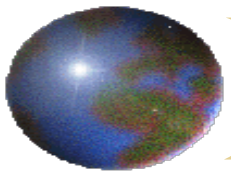
Clé étrangère

voitures



IV. Les règles d'intégrité

- ☐ **client_id** de **voitures** est une clé étrangère qui fait référence à la clé primaire **id** de **clients**
- ☐ Appliquer l'intégrité référentielle signifie que l'on ne pourra pas avoir, à aucun moment, une ligne de **voitures** avec un identifiant **client_id** inexistant dans la table **clients**.
- ☐ Une valeur de clé étrangère peut être Null

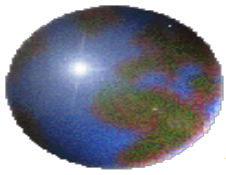


IV. Les règles d'intégrité

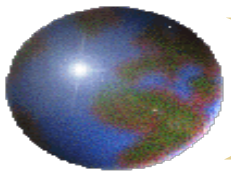
ON UPDATE et ON DELETE

Si l'utilisateur tente de supprimer une ligne d'une table parente, alors qu'une ou plusieurs lignes existent dans une table enfant correspondante à la première, il y a plusieurs possibilités :

- ❑ **CASCADE** efface la ligne de la table parente et supprime automatiquement les lignes correspondants dans la table enfant
- ❑ **SET NULL** supprime la ligne de la table parente et met la (ou les) valeur(s) de la clé étrangère à NULL
- ❑ **SET DEFAULT** supprime la ligne de la table parente et met tous les composants de la clé étrangère à leur valeur par défaut dans la table enfant
- ❑ **NO ACTION** rejette l'opération de suppression dans la table parente



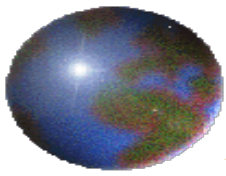
V *LES FORMES NORMALES*



V. Les formes normales

✚ La théorie de la normalisation

- Elle met en évidence les relations "indésirables«
- Elle définit les critères des relations "désirables« appelées **formes normales**
- Propriétés indésirables des relations
 - Redondances
 - Valeurs NULL
- Elle définit le processus de normalisation permettant de **décomposer** une relation non normalisée en un ensemble équivalent de relations normalisées.



V. Les formes normales

✚ La décomposition

Objectif:

- décomposer les relations du schéma relationnel sans perte d'informations
 - obtenir des relations canoniques ou de base du monde réel
 - aboutir au schéma relationnel normalisé
-
- Le schéma de départ est le schéma universel de la base
 - Par raffinement successifs on obtient des sous relations sans perte d'informations et qui ne seront pas affectées lors des mises à jour (**non redondance**).

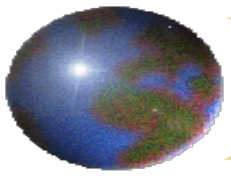
✚ Les formes normales

5 FN, les critères sont de plus en plus restrictifs

$FN_j \Rightarrow FN_i (j > i)$

• Notion intuitive de FN

une « *bonne relation* » peut être considérée comme une **fonction** de la clé primaire vers les attributs restants



V. Les formes normales

✚ 1ère Forme Normale 1FN

Une relation est en 1FN si tout attribut est atomique (non décomposable)

Contre-exemple

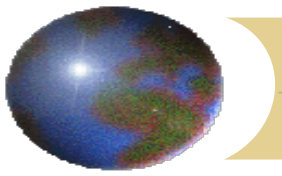
ELEVE (**no_elv**, nom, prenom, liste_notes)

Un attribut ne peut pas être un ensemble de valeurs

➡ Décomposition

ELEVE (**no_elv**, nom, prenom)

NOTE (no_elv, no_matiere, note)



V. Les formes normales

❖ 2ème Forme Normale 2FN

Une relation est en 2FN si:

- elle est en 1FN
- si tout attribut n'appartenant pas à la clé ne dépend pas d'une partie de la clé
- C'est la phase d'identification des clés
- Cette étape évite certaines redondances
- Tout attribut doit dépendre fonctionnellement de la totalité de la clé

Contre-exemple

une relation en 1FN qui n'est pas en 2FN

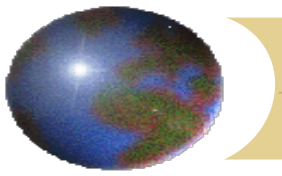
COMMANDE (**date**, **no_cli**, **no_pro**, qte, prixUHT)

elle n'est pas en 2FN car la clé = (**date**, **no_cli**, **no_pro**), et le **prixUHT** ne dépend que de **no_pro**

➡ Décomposition

COMMANDE (**date**, **no_cli**, **no_pro**, qte)

PRODUIT (**no_pro**, prixUHT)



V. Les formes normales

✚ 3ème Forme Normale 3FN

Une relation est en 3FN si:

- elle est en 2FN
- si tout attribut n'appartenant pas à la clé ne dépend pas d'un attribut non clé

Ceci correspond à la non transitivité des D.F. ce qui évite les redondances.

En 3FN une relation préserve les D.F. et est sans perte.

Contre-exemple

une relation en 2FN qui n'est pas en 3FN

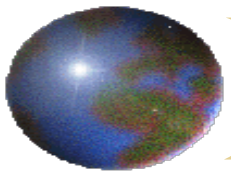
VOITURE (**matricule**, marque, modèle, puissance)

on vérifie qu'elle est en 2FN ; elle n'est pas en 3FN car la clé = **matricule**, et la **puissance** dépend de (marque, modèle)

➡ Décomposition

VOITURE (**matricule**, marque, modèle)

MODELE (**marque, modèle**, puissance)



V. Les formes normales

✚ 3ème Forme Normale de BOYCE-CODD BCNF

Une relation est en BCNF :

- elle est en 1FN et
 - ssi les seules D.F. élémentaires sont celles dans lesquelles une clé détermine un attribut
- BCNF signifie que l'on ne peut pas avoir un attribut (ou groupe d'attributs) déterminant un autre attribut et distinct de la clé
 - Ceci évite les redondances dans l'extension de la relation: mêmes valeurs pour certains attributs de nuplets différents
 - BCNF est plus fin que FN3 : **BCNF \Rightarrow FN3**

Contre-exemple

une relation en 3FN qui n'est pas BCNF

CODEPOSTAL (ville, rue, code)

on vérifie qu'elle est FN3, elle n'est pas BCNF car la clé = (ville,rue) (ou (code, ville) ou (code, rue)), et code \rightarrow ville