

**Evaluation**

- **Contrôle des connaissances :**
  - 1 écrit (CC) de modélisation de 1h ou 1h30 (25%)  
Programmé le **10/10**
  - 1 écrit (CC) de SQL de 1h ou 1h30 (25%),  
soit le **14/11** ou le **28/11**
  - 1 un examen final de 1<sup>ère</sup> session de 2h (50%)
- Deuxième session : 1 examen de rattrapage (100%)

2

# Bases de données

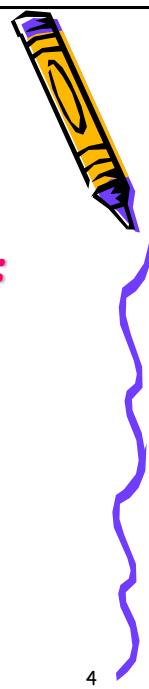


- Définitions
- Glossaire des SGBD relationnels
- Démarche
- Structures physiques



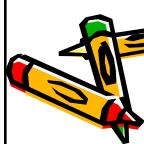
3

## BDD : définitions



**BDD = Bases Des Données :**

- Fichier
- Attribut
- Identifiant
- Bases de données
- SGBD

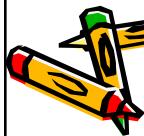


4

## BDD : glossaire SGBDR

**SGBDR = Système de Gestion des Bases de Données Relationnel**

- **Table ou relation**
- **Clé primaire**
- **Clé externe(étrangère)**
- **Champ ou attribut**
- **Domaine**
- **Tuple**



5



## BDD : démarche

- **Conception : modélisation, normalisation**
- **Développement : création, optimisation**
- **Utilisation**
- **Maintenance**



6



## BDD : Conception



- Méthode Merise : MCD, MLD, MPD

MCD = Modèle Conceptuel des Données

MLD = Modèle Logique des Données

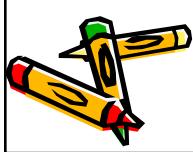
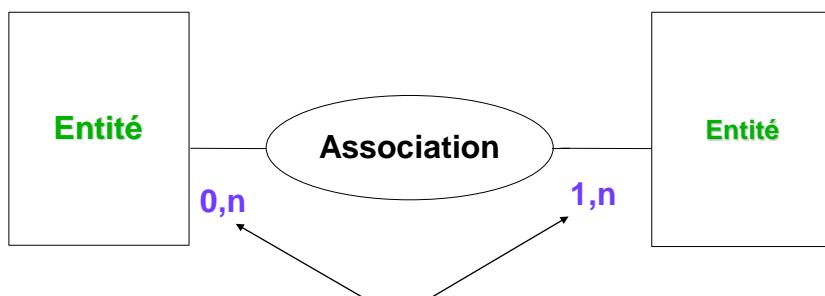
MPD = Modèle Physique des Données

- Normalisation : formes normales

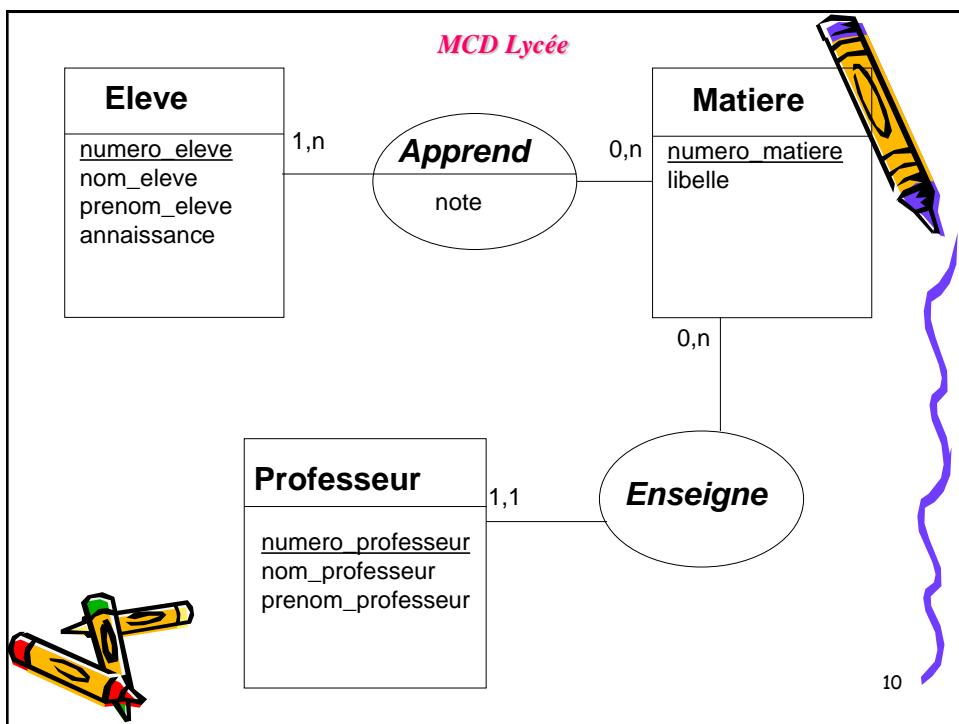
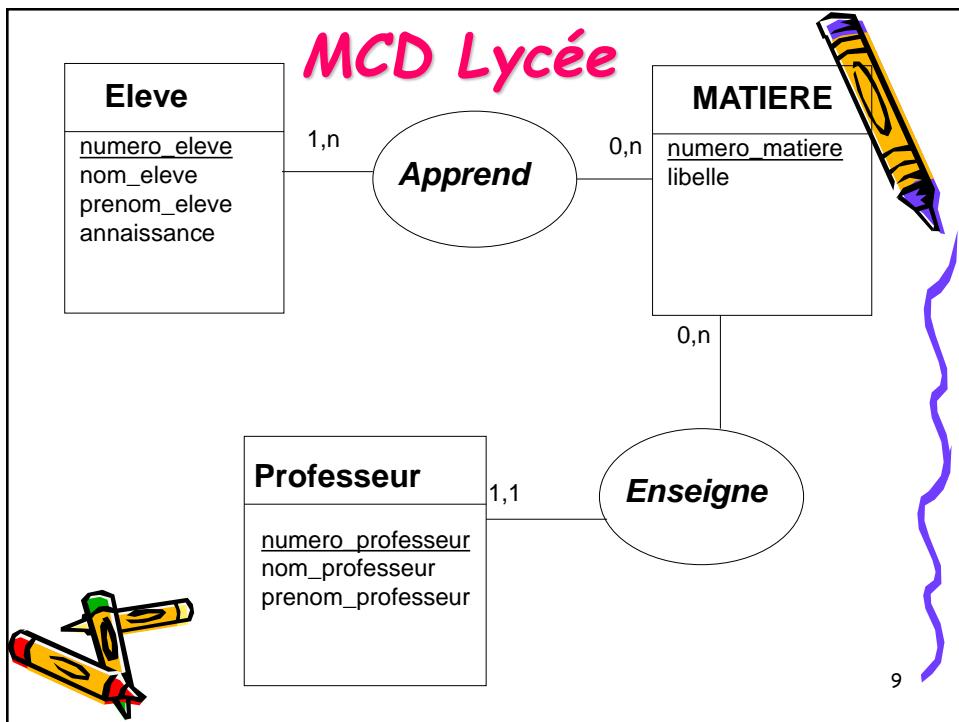


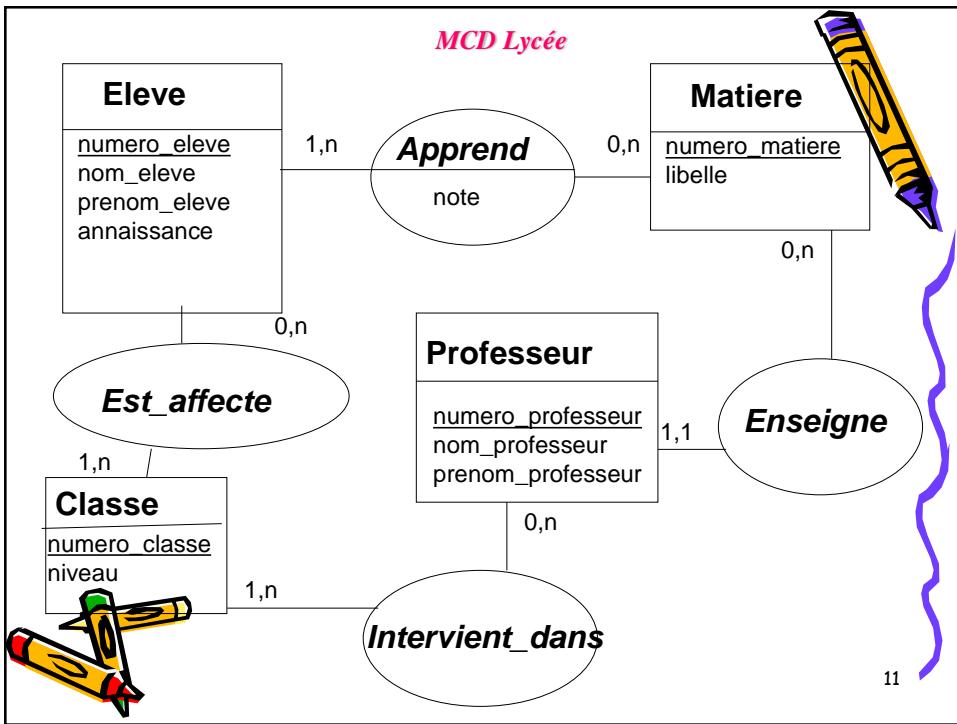
7

## Merise données : MCD



8





## *Merise : MLD relationnel*

- Chaque entité devient une table
- Chaque association porteuse devient une table (associations : apprend, est\_affecté, intervient: 1,n ou 0,n to 0,n ou 1,n)
- Chaque association non porteuse devient soit une table de pointeurs, soit une clé étrangère dans la table qui a la cardinalité 0,1 ou 1,1
- Ici, nous pouvons voir que la cardinalité (1,1) ou (0,1) va nous indiquer l'entité qui va recevoir la clé étrangère.

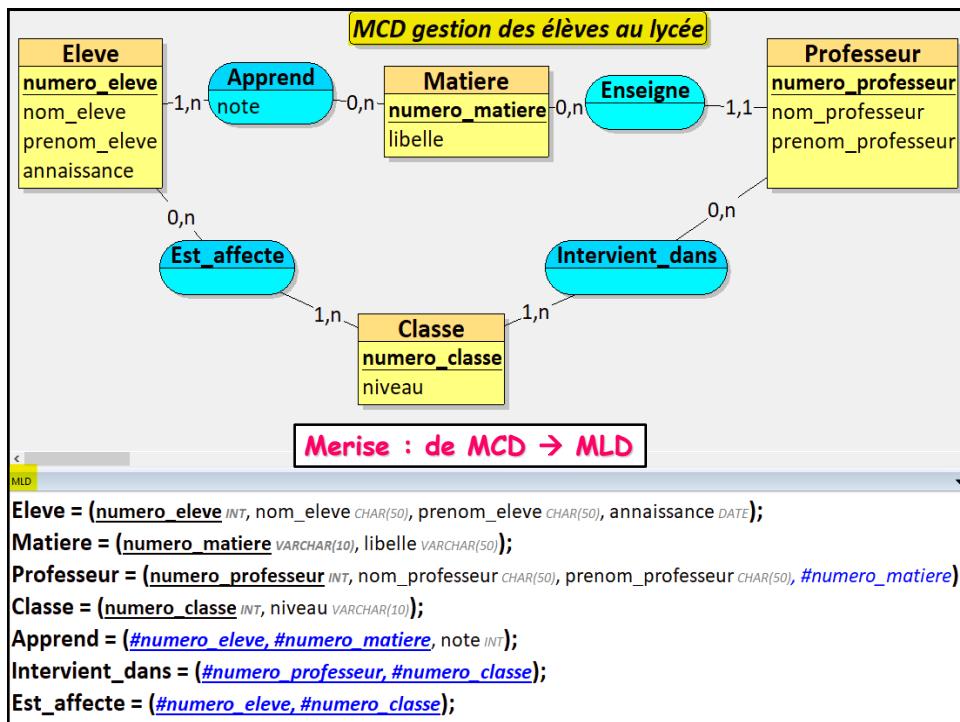
# Merise : MLD relationnel

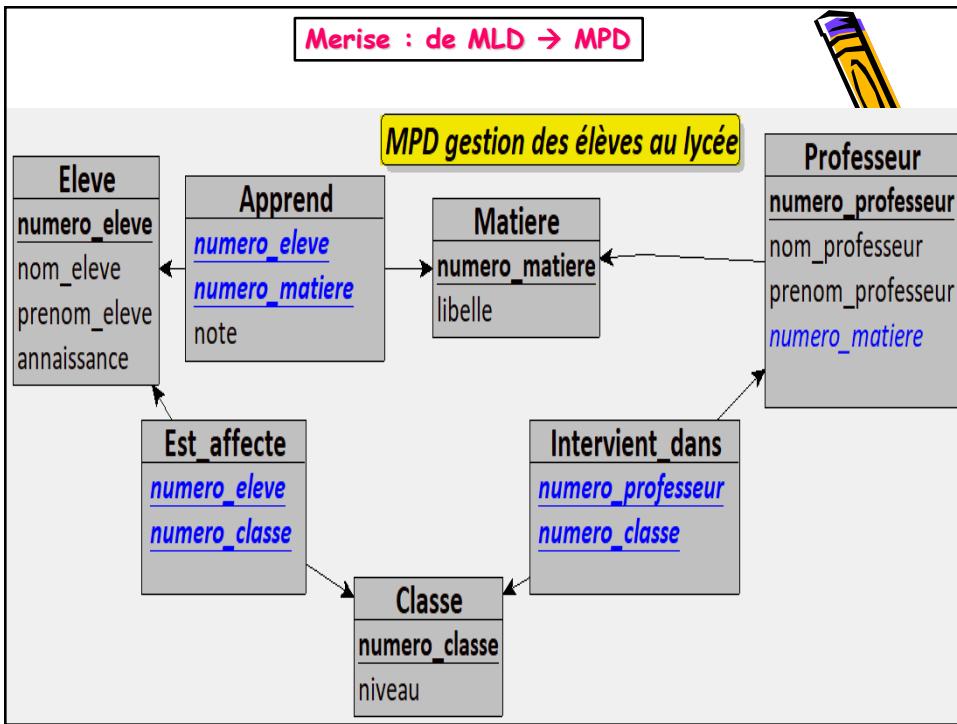
1. **Eleve** (numero\_eleve, nom\_eleve, prenom\_eleve, annaissance)
2. **Apprend** (#numero\_eleve, #numero\_matiere, note)
3. **Matiere** (numero\_matiere, libelle)
4. **Professeur** (numero\_professeur, nom\_professeur, prenom\_professeur, #numero\_matiere)
5. **Classe** (numero\_classe, niveau)
6. **Intervient\_dans** (#numero\_classe, #numero\_professeur)
7. **Est\_affecté** (#numero\_eleve, #numero\_classe)

Le nouvel identifiant (clé primaire) des nouvelles entités

Intervient\_dans, Apprend, Est\_affecte, seront une concaténation des deux clés étrangères.

13





## **Normalisation de Codd : point de départ**

- **Liste des attributs**
- **Dépendances fonctionnelles**
- **Dépendances multivaluées**



17



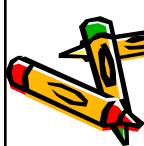
## **Normalisation : processus**

**Normaliser c'est  
regrouper la liste des  
attributs en relations**

**- ayant du sens**

**- cohérentes**

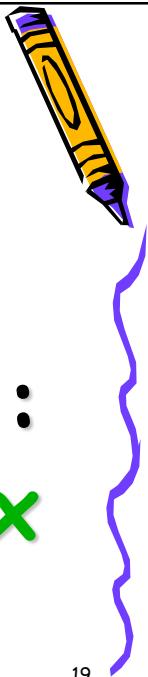
**- sans redondance**



18



Normalisation : Point d'arrivée



On obtient une  
série de tables :  
fichiers et index



19

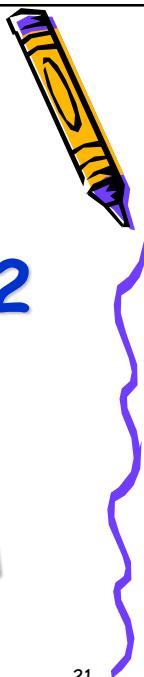
Normalisation : 1FN =  
Forme Normale

- Tout attribut de R  
contient:  
une valeur monovaluée  
et non composée



20

Normalisation : 1FN



• Exemples :

- prenom 1, prenom 2
- Adresse:  
numero\_voie, type  
voie, nom voie, nom  
ville, code postal

21

Normalisation : 1FN



● R (numero\_eleve,  
prenoms, Adresse)  
n'est pas en 1FN



22

**R normalisée en 1FN**

**R = Relation**

**• R (numero\_eleve,  
prenom1, prenom2,  
numero\_voie, Type\_voie,  
Nom\_voie, Nom\_ville,  
Code\_postal)**

23

**Normalisation : 2FN**

**1. R est en 1FN**

**2. Tout attribut non clé  
de R dépend de la  
totalité de la clé**

24

**Normalisation : 2FN**

- R (numero\_eleve,  
numero\_matiere, note,  
nom\_eleve)

**Le Nom\_eleve ne dépend  
Fonctionnellement que du  
numero\_eleve**

25

100	005	12	Dupont
100	001	16	Dupont
100	002	08	Dupont
100	003	10	Dupont
200	004	12	Martin

26

## **Normalisation : 2FN**

● R n'est pas en 2FN

● On normalise :

**R1 (numero\_eleve,  
nom\_eleve)**

**R2 (numero\_eleve,numero\_  
matiere, Note)**



27



## **Normalisation : 3FN**

**1. R est en 2FN**

**2. Tout attribut de R ne  
dépend pas de la clé  
par transitivité**



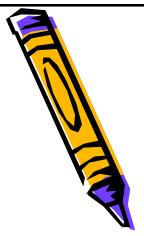
28



## Normalisation : 3FN

● R (numero\_professeur,  
Nom\_professeur,  
numero\_matiere, Libelle  
matiere)

Libelle matière dépend  
fonctionnellement de numero\_  
professeur par transitivité, à  
travers le numero\_matiere



29

100	Duval	001	Anglais
101	Farenc	001	Anglais
102	Gervais	001	Anglais
103	Justin	001	Anglais
104	Loliée	002	Espagnol

30



On normalise en 3FN



R1 (numero\_professeur,  
Nom\_professeur,  
numero\_matiere)

R2 (numero\_matiere,  
Libelle matiere)



31

Normalisation 4FN et 5FN



● On traite les DM

Dépendances multivaluées



32

4FN : la relation ne contient qu'une seule DM

## R (Etudiant, Cours, Sport)

avec

Etudiant DM Cours

Etudiant DM Sport

33

R n'est pas en 4FN

Pierre	Maths	Karaté
Pierre	Anglais	Karaté
Pierre	Physique	Karaté
Jacques	Philo	Judo
Jacques	Philo	Natation
Jacques	Philo	Marathon

34

Normalisation de R en 4FN



**R1** (Etudiant, Cours)

**R2** (Etudiant, Sport)



35

**R1**

Pierre	Maths
Pierre	Anglais
Pierre	Physique
Jacques	Philo



36

**R2**

Pierre	Karaté
Jacques	Judo
Jacques	Natation
Jacques	Marathon

37

**5FN : DM de jointure**

**R (Revendeur, Marque, Type de produit)**

avec

**Revendeur DM Marque**

**Revendeur DM Type de produit**

**Marque DM Type de produit**

38




Pierre	Renault	Voiture
Pierre	Renault	Camion
Pierre	Volvo	Voiture
Paul	Renault	Camion
Jacques	Renault	Voiture
Jacques	Volvo	Voiture

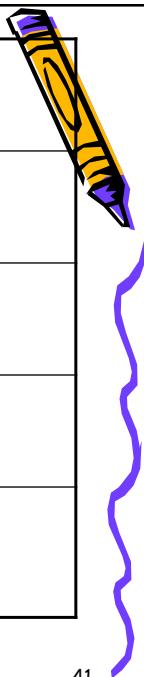
39




Normalisation de R en 5FN	
R1	(Revendeur, Marque)
R2	(Revendeur, Type de produit)
R3	(Marque, Type de produit)

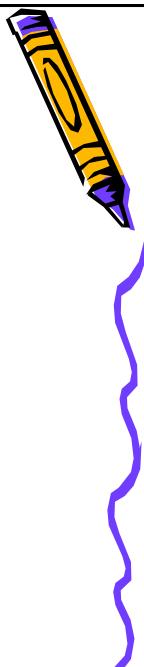
40

Pierre	Renault
Pierre	Volvo
Jacques	Renault
Paul	Renault
Paul	Volvo

 **R1** 

41

Pierre	Voiture
Pierre	Camion
Paul	Camion
Jacques	Voiture

 **R2** 

42

<i>Renault</i>	<i>Voiture</i>
<i>Renault</i>	<i>Camion</i>
<i>Volvo</i>	<i>Voiture</i>

*R3*

43

**DF contient 3FN**

**5FN**

**4FN } DM**

44

## NORMALISATION : préparation

- numero\_eleve = A
  - nom\_eleve = B
  - prenom\_eleve = C
  - annaissance = D
  - numero\_matiere = E
  - Libelle\_matiere = F
  - Note = G
  - numero\_professeur = H
  - Nom\_professeur = I
  - prenom\_professeur = J
  - numero\_classe = K
  - Niveau = L
- A DF B, C, D, K
  - E DF F
  - AE DF G
  - H DF I, J, E
  - K DF L
- A DM E
  - H DM K
  - E DM H



45



## NORMALISATION : exécution

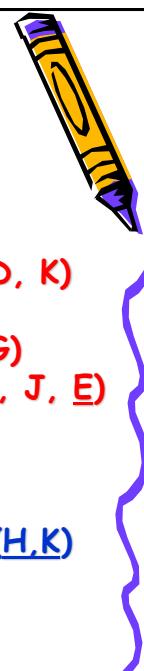
- A DF B, C, D, K
  - E DF F
  - AE DF G
  - H DF I, J, E
  - K DF L
- A DM E
  - H DM K
  - E DM H



- 1) DF
  - Elève (A, B, C, D, K)
  - matière (E, F)
  - Apprend (A, E, G)
  - Professeur (H, I, J, E)
  - Classe (K, L)

- 2) DM
  - Intervient dans (H, K)

46



## *Création & manipulation des BDD : requêtes SQL*



- « Structured Query Language » 1986

- Opérateurs de l'algèbre relationnelle



47

« LDD » : langage de définition des données

```
CREATE TABLE eleve  
(numero_eleve NUM(6) [ UNIQUE] [NOT  
NULL] [PRIMARY KEY],  
nom_eleve CHAR(30)[NOT NULL],  
prenom_eleve CHAR(20) [NOT NULL],  
annaissance NUM(4)...)
```



48

« LDD » langage de définition des données

**CREATE TABLE** apprend  
**(numero\_eleve [ UNIQUE] [NOT NULL]  
[FOREIGN KEY],**  
**numero\_matiere [ UNIQUE] [NOT NULL]  
[FOREIGN KEY],**  
**note NUM(2,2) [VALEUR: 0,00 à 20,00])**

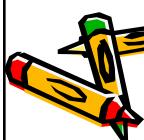


49

## □ INDEX : structures d'accès

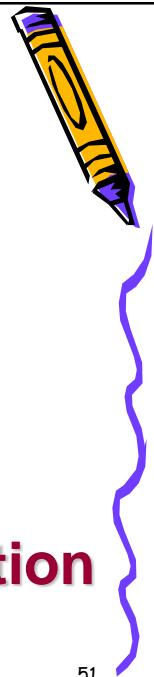
**CREATE [UNIQUE] INDEX i1  
ON eleve (numeroélève)**

**CREATE INDEX i2 ON eleve  
(annaissance)**



50

ALTER TABLE matiere  
ADD COLUMN coefficient  
NUM(1)

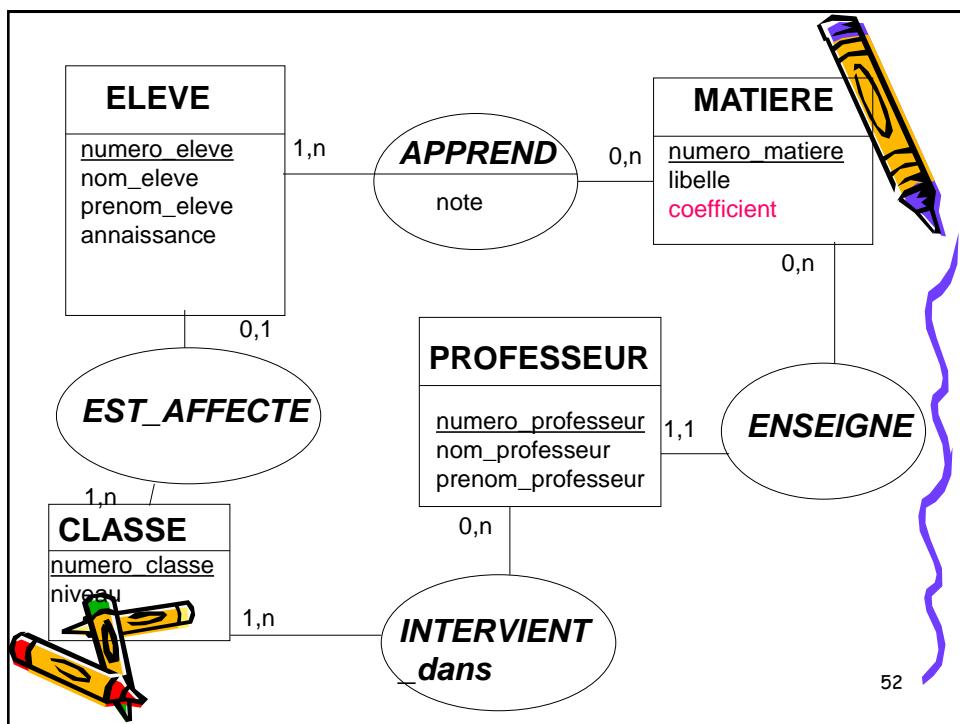


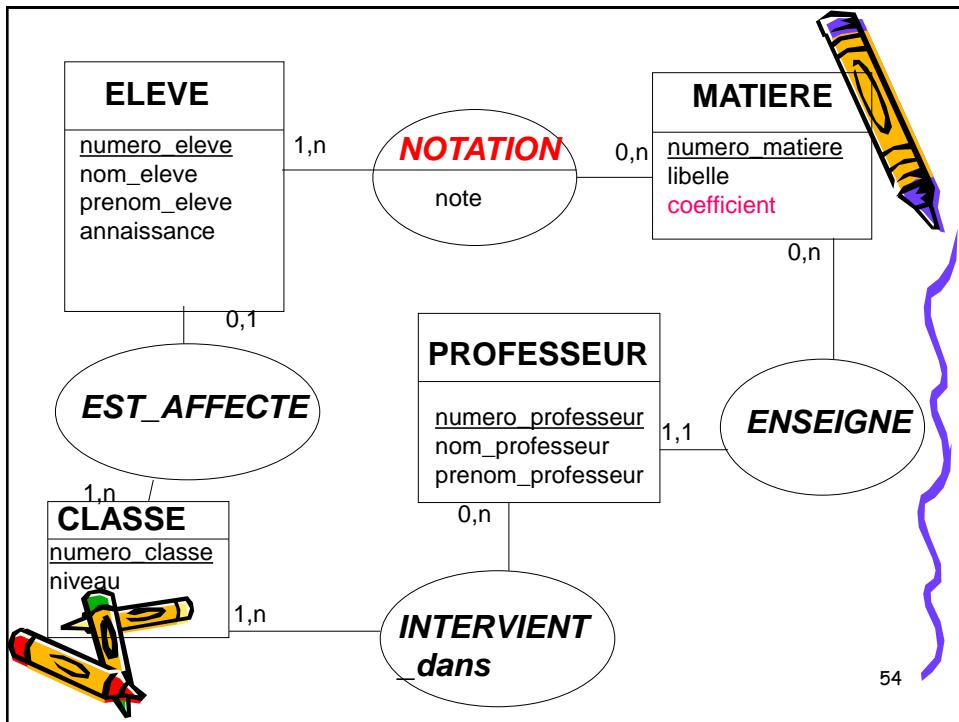
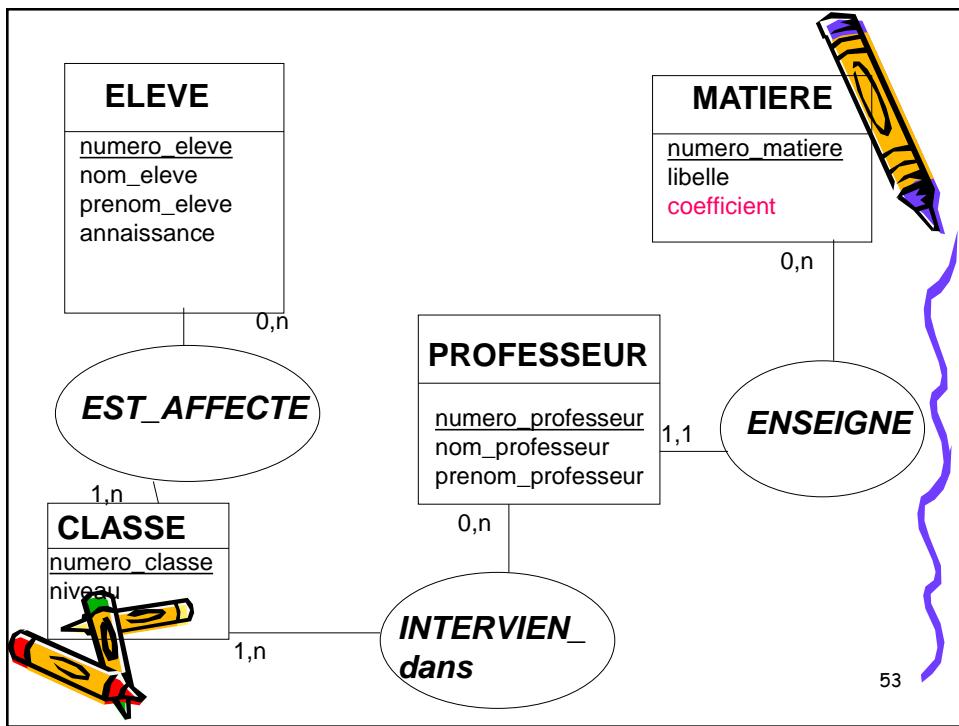
DROP TABLE apprend

RENAME apprend to notation



51





« Les autorisations d'accès »

## GRANT & REVOKE



GRANT *privilege* ON *objet* TO *public*

**Qui (public) a le droit de faire  
quoi (privilège)  
à quoi (objet)**

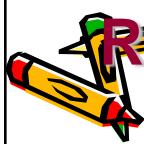


55

GRANT INSERT  
ON *apprend*  
TO *professeur*



GRANT SELECT  
ON *apprend*  
TO PUBLIC



REVOKE ALL ON *apprend*  
FROM PUBLIC

56

« LMD » : langage de manipulation des données

**SELECT**



**INSERT**

**UPDATE**



**DELETE**

57

**SELECT**

**SELECT**

**nom\_eleve, prenom\_eleve**

**FROM eleve**

**WHERE annaissance = 1994**

**ORDER BY nom\_eleve**



58

**SELECT**  
**SELECT nom\_eleve, prenom\_eleve,**  
**note**  
**FROM eleve, apprend**  
**WHERE annaissance = 1994**  
**AND eleve.numero\_eleve=apprend.numero\_eleve**  
**AND note NOT NUL**  
**ORDER BY nom\_eleve ASC**



59

**□INSERT**  
**INSERT INTO eleve**  
**(numero\_eleve, numero\_classe,**  
**nom\_eleve, prenom\_eleve,**  
**annaissance)**  
**VALUES (105, C217, Durant,**  
**Alexis, 2000)**



60

## UPDATE

UPDATE apprend

SET note = 14

WHERE numero\_eleve = 106

AND numero\_matiere = 5



61



## DELETE

DELETE FROM

eleve

WHERE

numero\_eleve = 78



62



# Les agrégats

## □ SUM

**GROUP BY** pour regrouper les données appartenant à la même entité(obligatoire dans SELECT).

**SELECT numero\_client, SUM(montant)**

**FROM commande**

**GROUP BY numero\_client**

63

## □ AVG

2 requêtes, Est-ce possible de les regroupées !

**SELECT numero\_eleve, nom\_eleve  
FROM eleve**

**SELECT numero\_eleve, AVG(note)**

**FROM apprend**

**GROUP BY numero\_eleve**

**OU :**

**SELECT numero\_eleve, nom\_eleve, AVG(note)**

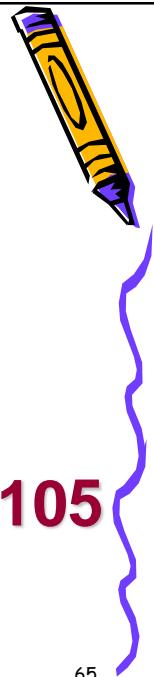
**FROM eleve, apprend**

**WHERE eleve.numero\_eleve=apprend.numero\_eleve**

**GROUP BY AVG(note) DESC**

64

MIN ou MAX



**SELECT MAX(note)**  
**FROM apprend**  
**WHERE numero\_eleve = 105**



65

COUNT

**SELECT COUNT(\*)**  
**FROM eleve**  
**SELECT COUNT(\*) FROM**  
**apprend**  
**WHERE numero\_eleve = 102**



66

## Opérateurs ensemblistes

EXISTS

UNION

INTERSECT

MINUS



67

### EXISTS

la commande EXISTS s'utilise dans une clause conditionnelle pour savoir s'il y a une présence ou non de lignes lors de l'utilisation d'une sous-requête.

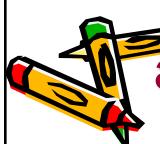
**SELECT nom\_eleve FROM eleve**

**WHERE EXISTS**

**(SELECT \* FROM apprend**

**WHERE note = 20**

**AND eleve.numero\_eleve =  
apprend.numero\_eleve)**



68

## NOT EXISTS

**SELECT nom\_eleve FROM eleve**

**WHERE NOT EXISTS**

**(SELECT \* FROM matiere,apprend**

**AND eleve.numero\_eleve =**  
**apprend.numero\_eleve**

**AND matiere.code\_matiere =**  
**apprend.code\_matiere)**



69



## UNION

Permet de concaténer les résultats de 2 requêtes ou plus. Pour l'utiliser il est nécessaire que chacune des requêtes à concaténer retournent le même nombre de colonnes, avec les mêmes types de données et dans le même ordre.

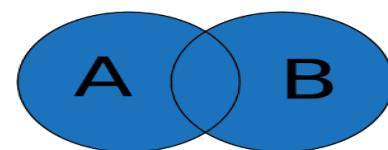
**apprend(numero\_eleve, code\_matiere)**

**pratique(numero\_eleve, code\_sport)**

**SELECT numero\_eleve FROM apprend**

**UNION**

**SELECT numero\_eleve FROM pratique**



70



## INTERSECT

Permet d'obtenir l'intersection des résultats de 2 requêtes. Cette commande permet donc de récupérer les enregistrements communs à 2 requêtes. Cela peut s'avérer utile lorsqu'il faut trouver s'il y a des données similaires sur 2 tables distinctes.

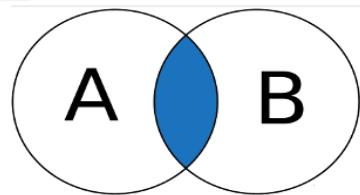
**SELECT numero\_eleve**

**FROM apprend**

**INTERSECT**

**SELECT numero\_eleve**

**FROM pratique**



71

## EXCEPT/MINUS

EXCEPT s'utilise entre 2 instructions pour récupérer les enregistrements de la première instruction sans inclure les résultats de la seconde requête. Si un même enregistrement devait être présent dans les résultats des 2 syntaxes, ils ne seront pas présent dans le résultat final.

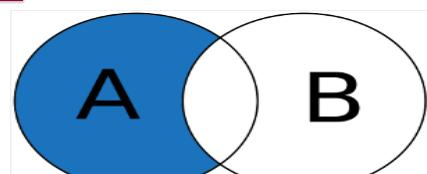
**SELECT numero\_eleve**

**FROM apprend**

**MINUS**

**SELECT numero\_eleve**

**FROM pratique**



72

## VUE : table virtuelle

Non stockée (temporaire)  
créée par une requête  
sur une ou plusieurs tables  
(permanentes)



73



- **CREATE VIEW** France\_Custom

**AS**

```
SELECT CustomerName, ContactName  
FROM Customers  
WHERE Country = 'France';
```

- **SELECT \* FROM** France\_Custom;



74



# DF : Dépendances Fonctionnelles



## Exercice 1 :

Considérons le schéma de la relation suivante : R(A, B, C, D, E)

Cette relation est définie en extension par les tuples suivants :

A	B	C	D	E
a1	b2	c2	d3	e2
a1	b2	c2	d1	e1
a2	b3	c2	d1	e5
a2	b4	c5	d1	e5

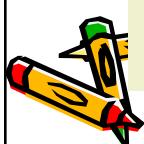
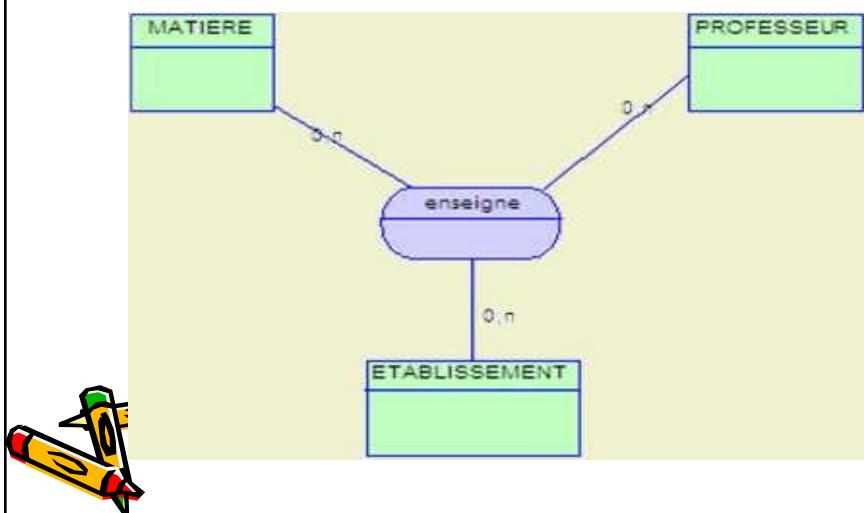
- 1) Parmi les dépendances fonctionnelles suivantes, lesquelles s'appliquent à l'extension de R ?

- E→D       E→A       {A, E}→C
- D→E       B→C       {A, E}→D
- C→A       B→D       {A, D}→E
- E→B       B→A       {A, B}→A



75

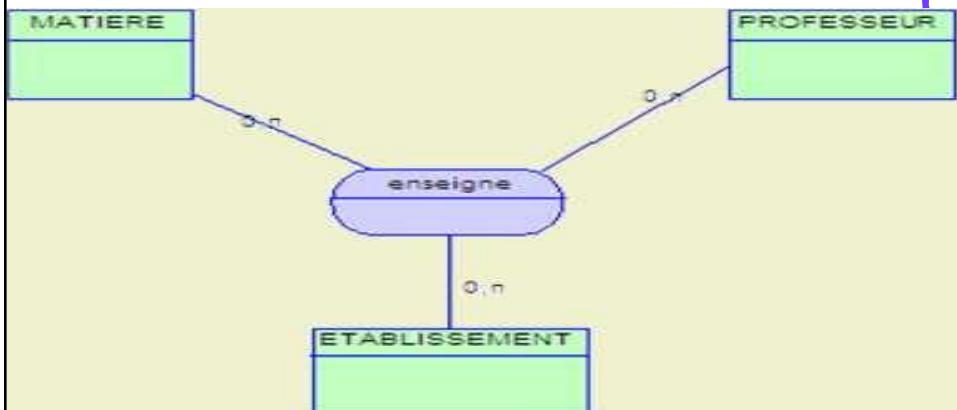
## Modèle Conceptuel de Données Les associations ternaires



76

## Intérêt d'une association ternaire

Pour savoir quelle matière est enseignée par tel professeur dans tel établissement, il faut relier les 3 entités par une association ternaire



## Associations binaires ou ternaires ?

Quel serait le MCD dans cet exemple ?

Les professeurs enseignent dans des établissements dans certaines matières. Il est possible qu'un professeur enseigne des matières différentes dans les établissements de l'enseignement supérieur.

Décomposons cet exemple en phrases simples :

Un professeur peut enseigner dans plusieurs établissements supérieurs

Un établissement peut avoir plusieurs professeurs

Une matière est enseignée par plusieurs professeurs

Un professeur peut enseigner plusieurs matières

Un établissement propose plusieurs matières

Une matière est proposée dans plusieurs établissements

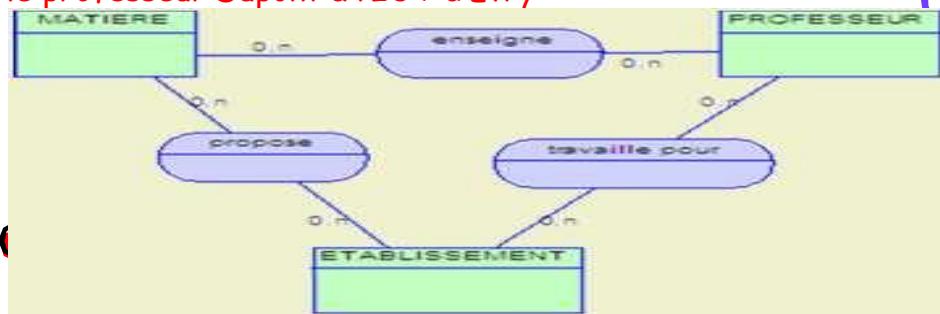


## Associations binaires ? Problème...

Avec cette modélisation, nous savons qu'un professeur enseigne certaines matières et travaillent dans certains établissements.

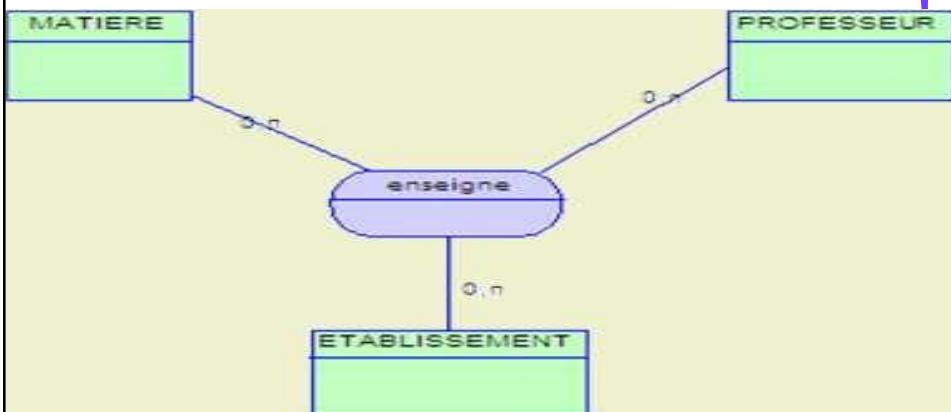
Le professeur Dupont enseigne les mathématiques et l'informatique.

Le professeur Dupont travaille à l'université D'Evry et à l'IUT  
Mais nous ne savons pas quelles matières sont enseignées par le professeur Dupont à l'IUT d'Evry



## Intérêt d'une association ternaire

Pour savoir quelle matière est enseignée par tel professeur dans tel établissement, il faut relier les 3 entités par une association ternaire



## Associations n-aires



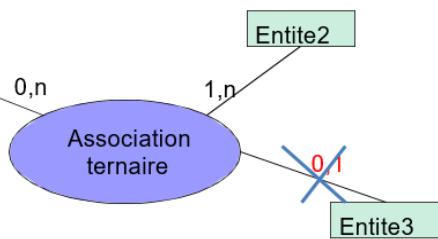
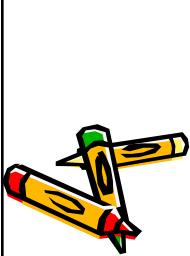
- Il peut exister des quaternaires, voire des n-aires mais c'est extrêmement rare.
- Il y a peut être une erreur de conception si le nombre d'entités reliées (arité) est supérieur ou égale à 4.



81

## Association ternaire et cardinalité maximale

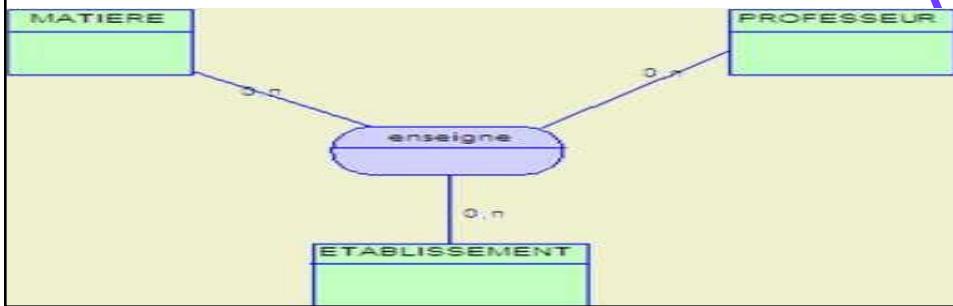
- Dans une association ternaire,
  - Les cardinalités minimales peuvent être à 0 ou 1 selon le contexte (participation facultative ou obligatoire de l'entité à l'association).
  - Toutes les cardinalités maximales sont obligatoirement à n.  
La cardinalité maximale à n représente le fait qu'une occurrence de l'entité peut participer n fois à l'association.  
Si l'une des cardinalités maximales n'est pas à n, il y a une erreur de conception



82

## Associations ternaires : limites

- Une association ternaire met en jeu les 3 entités.  
Dans l'association ternaire « enseigne », on ne peut pas mémoriser des professeurs qui enseignent des matières en dehors d'un établissement car dans ce cas, l'entité « établissement » ne serait pas concernée par la relation. Par contre, on peut mémoriser des professeurs qui n'enseignent pas, car la cardinalité minimale est à zéro.



## Association ternaire et agrégat

Si un professeur peut enseigner plusieurs matières mais si un professeur ne peut enseigner qu'une seule matière par établissement, nous sommes en présence d'une "fausse ternaire".

Ce cas particulier s'appelle un agrégat et sera vu plus en détail ultérieurement.

Il est possible de le traiter comme une ternaire dans le MCD mais cela aura une répercussion sur la clé de la nouvelle table dans le MLD.

## Association ternaire et réflexive

L'étude concerne un seul établissement.  
Un professeur peut enseigner plusieurs matières  
à d'autres professeurs.

Il y a une association ternaire avec :

- 1 lien vers l'entité Matière
- 1 lien vers l'entité Professeur  
(rôle de professeur-formateur)
- 1 autre lien vers l'entité Professeur  
(rôle de professeur-formé)



85

## Interprétation d'un MCD : cas Sport

Nous allons proposer plusieurs cas qui modélisent  
un même contexte avec 3 entités : Adhérent,  
Centre, Sport.

Chaque modélisation a une signification différente.  
Indiquez en français ce que modélise chaque  
schéma.

Essayez de voir les limites de chaque schéma.

Il est fortement conseillé d'essayer de rédiger la  
signification de chaque schéma avant de regarder  
la correction



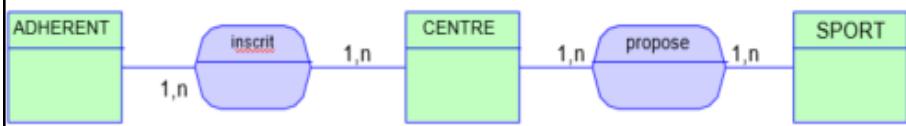
86

## Interprétation d'un MCD - cas 1

Des adhérents sont inscrits dans des centres.

Les centres proposent des sports.

On ne sait pas quel(s) sport(s) pratique un adhérent.



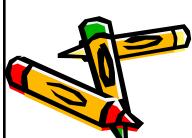
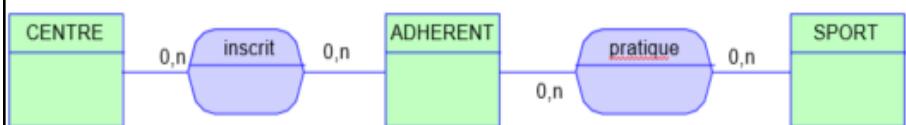
87

## Interprétation d'un MCD - cas 2

Des adhérents sont inscrits dans des centres.

Les adhérents pratiquent certains sports.

On ne sait pas quel(s) sport(s) propose un centre.

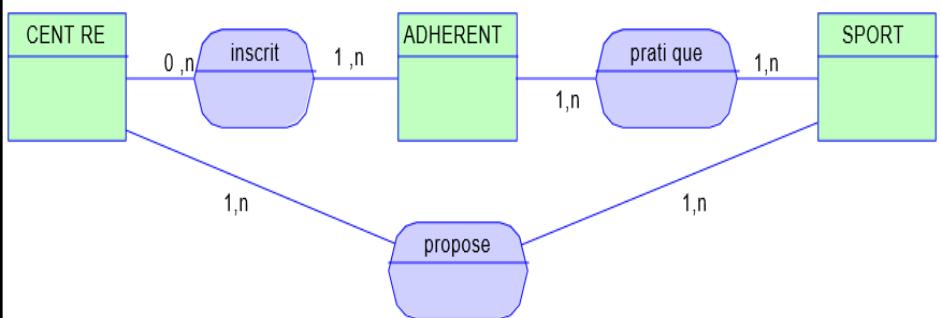


88

## Interprétation d'un MCD - cas 3



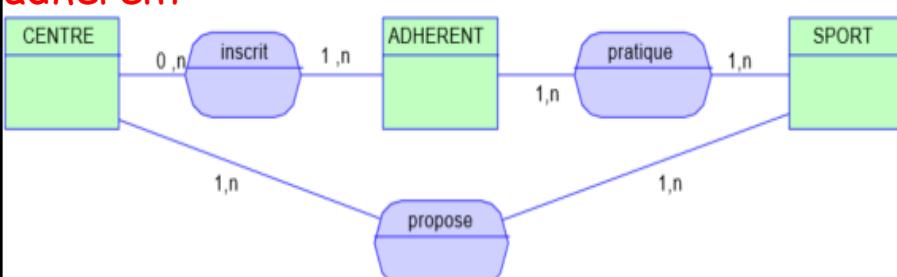
Des adhérents sont inscrits dans des centres.  
Les adhérents pratiquent des sports.  
Les centres proposent des sports.  
**On ne sait pas dans quel(s) centre(s) un adhérent pratique un sport donné.**



## Interprétation d'un MCD - cas 4



Des adhérents pratiquent des sports dans des centres.  
**On ne sait pas quel sport est proposé par un centre si ce sport n'est pratiqué par aucun adhérent**

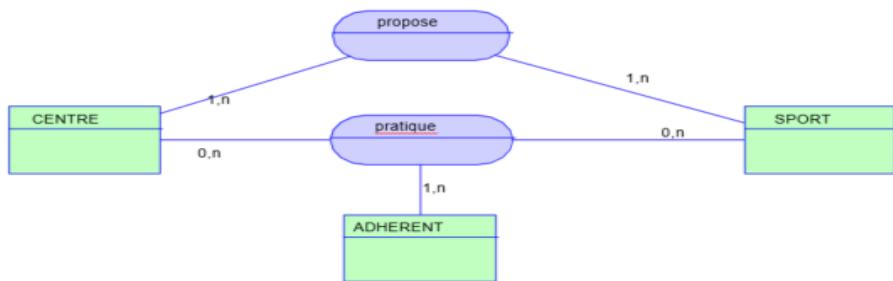


## Interprétation d'un MCD - cas 5

Des adhérents sont inscrits dans des centres pour pratiquer des sports.

Les centres proposent des sports à pratiquer.

On ne connaît pas les sports pratiqués par les adhérents hors de ces centres (ex : sports individuels).

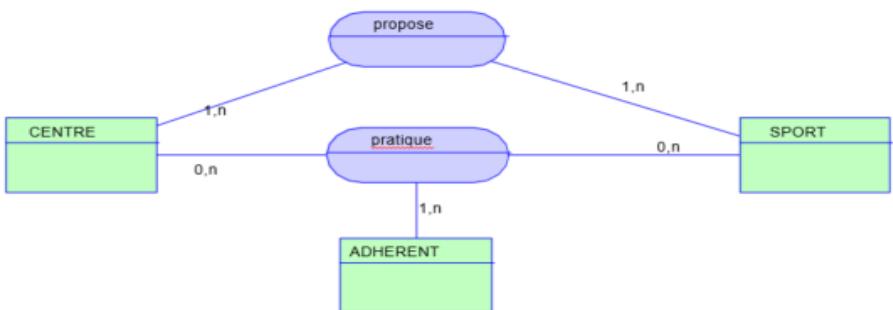


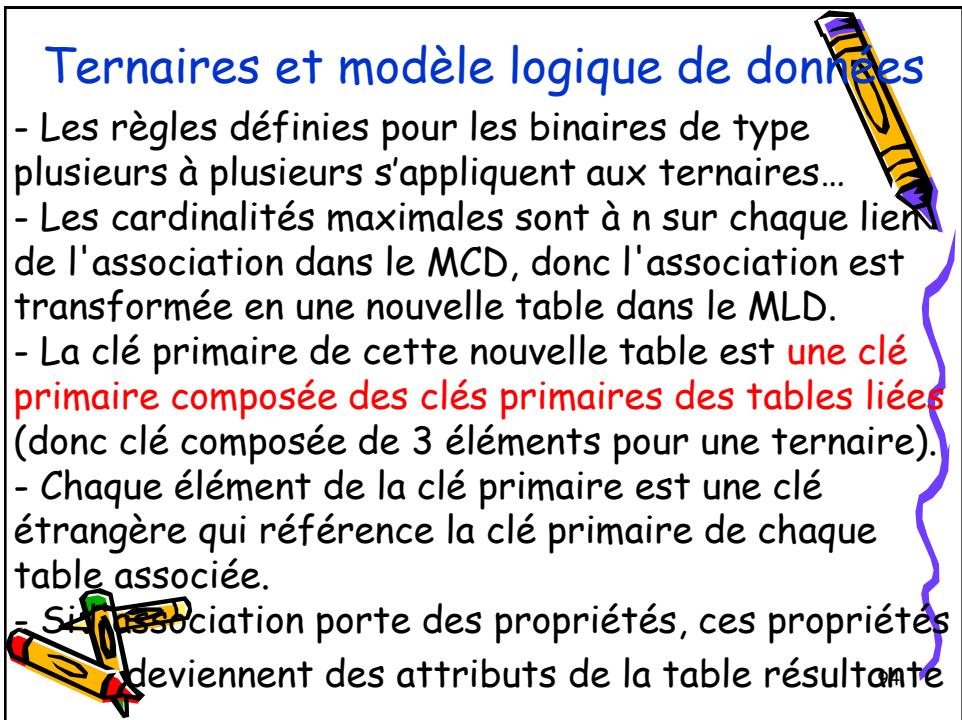
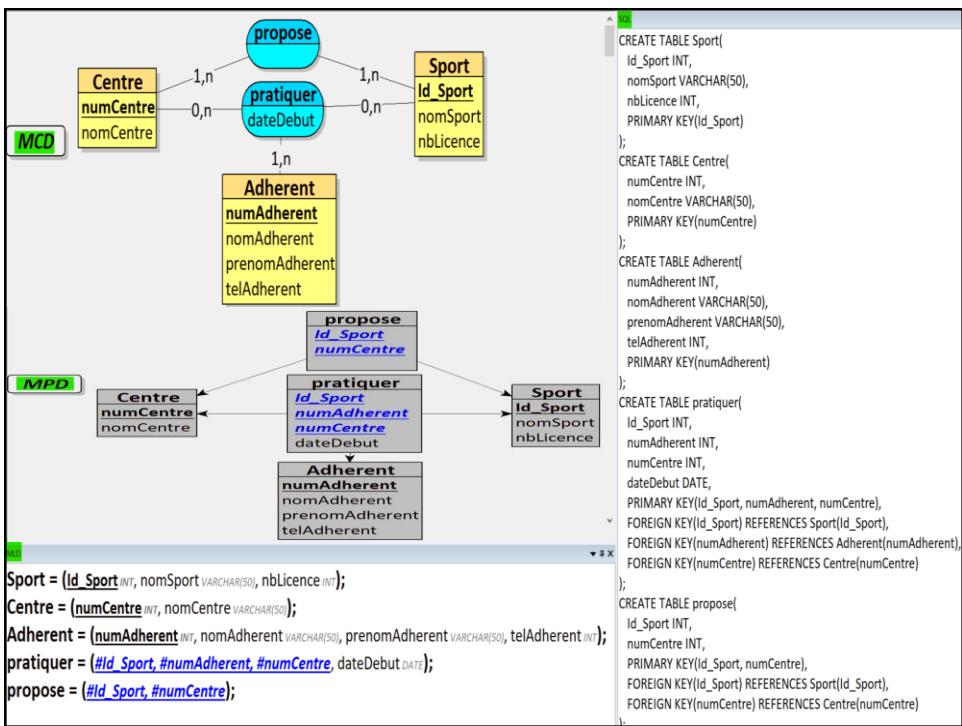
## Interprétation d'un MCD - cas 6

Des adhérents sont inscrits dans des centres pour pratiquer des sports.

Les centres proposent des sports à pratiquer.

Proposez une modification pour connaître les sports pratiqués par les adhérents hors des centres.





# Passage au MLD d'une ternaire

Table Centre(numCentre, nomCentre)

numCentre: clé primaire de la table Centre

Table Sport(idSport, nomSport, nbLicences)

idSport: clé primaire de la table sport

Table Adherent(numAdherent, nomAdherent, prenomAdherent, telAdherent)

numAdherent: clé primaire de la table Adherent

Table Pratique(numCentre, idSport, numAdherent, dateDebut)

numCentre, idSport, numAdherent: clé primaire composée de la table Pratique

numCentre clé étrangère qui référence numCentre de la table Centre

idSport clé étrangère qui référence idSport de la table Sport

numAdherent clé étrangère qui référence numAdherent de la table Adherent

