

EMD
Module BDD 2LMD
Durée 1h30. Documents non autorisés

Exo1 : (3 points)

Soit $R1(A, B, C, D, E, F)$ une relation avec l'ensemble des dépendances suivantes :
 $AB \rightarrow C, D, E$; $B \rightarrow C$; $D \rightarrow E$; $F \rightarrow B$; $F \rightarrow C$

- Donner le graphe minimum des dépendances. Quelle est la clé de $R1$?
- Proposer une décomposition en 3^{ème} forme normale de $R1$.

Exo 2 : (5 points)

Soit la relation Sinistre (Num-sinistre, Imm-vehicule, modèle-vehi, ann-circulation, NSS-client, nom, prénom, adresse, date-sinistre, montant-dégât, NSS-conducteur, nom-Cond, prénom-Cond, part-responsabilité-cond)

Modélisant la gestion des sinistres (accidents) d'une compagnie d'assurances :

Les clients sont possesseurs d'un ou plusieurs véhicules actuellement assurés. Chaque client est identifié par son numéro NSS et possède un nom et prénom et une adresse. Chaque véhicule est identifié par son numéro d'immatriculation et possède un modèle et une année de mise en circulation. Chaque véhicule a fait l'objet d'un certain nombre de sinistres (accidents). Chaque sinistre est numéroté et possède une date, le nom et le numéro NSS du conducteur, le montant des dégâts et la part de responsabilité du conducteur. Ce qui est à noter est que le conducteur peut être différent du propriétaire du véhicule.

Questions :

- donner les dépendances fonctionnelles de cette description,
- résoudre le problème des attributs multi-valués (1FN)
- donner le Graphe minimum des DF. Quelle est la clé de cette relation?
- décomposer la relation Sinistre en relations sans redondances (3^{ème} forme normale)

Problème : (12 points)

Soit la base de données relationnelle de gestion des notes des étudiants :

etudiant (numEtud, NSS, nomEtud, prénomEtud, dateNaiss, Sexe, Département*, cycle) cycle $\in \{L1, L2, L3, M1, M2\}$
enseignant (numEns, NSS, nomEns, grade, département*) grade prend ces valeurs dans $\{« A », « MA », « CC »\}$
Cours (numCours, Intitulé, numEns, cycle, département*) cycle $\in \{L1, L2, L3, M1, M2\}$
Notes (NumEtud, NumCours, Note)
département (numDep, Libellé-Dép)

- Écrire en LDD de SQL les commandes permettant de créer ces relations en prenant soin d'indiquer les contraintes d'intégrité (les clés primaires, les clés secondaires, les clés étrangères, les contraintes sur les valeurs ...)
- Soient les requêtes suivantes :
 - ✓ + a) Quels sont les enseignants qui enseignent des cours dans des départements différents de leur département d'origine (auquel ils sont affectés)
 - ✓ + b) donner les noms et prénoms des étudiants avec leurs notes dans le cours base de données
 - ✗ + c) donner la liste des étudiants inscrits en L3 et qui ont eu tous les modules
 - ✗ d) donner le nombre d'étudiants par département
 - ✓ + e) donner les noms des enseignants qui n'ont jamais donné de cours

Écrire en algèbre relationnelle les requêtes a), b), c), e)

Écrire en SQL les requêtes a), b), d), e)

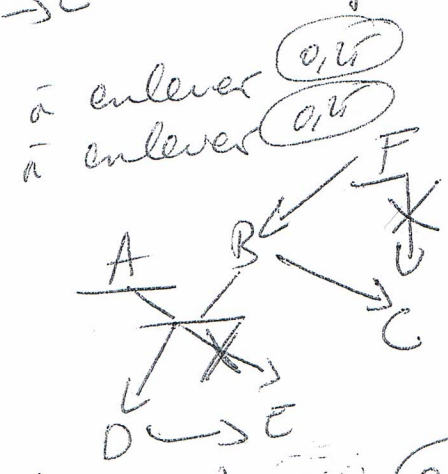
Bon courage

NB : pour vous retrouver dans les requêtes procéder par étapes

Exo1:

- a) $AB \rightarrow C \not\Rightarrow AB \rightarrow C$
 $B \rightarrow C$
 $AB \rightarrow E$ suite
 $F \rightarrow C$ "

DF augmentée à enlever (0,2)



GN (0,25)

Clé (F, A) (0,25)

b) Décomposition

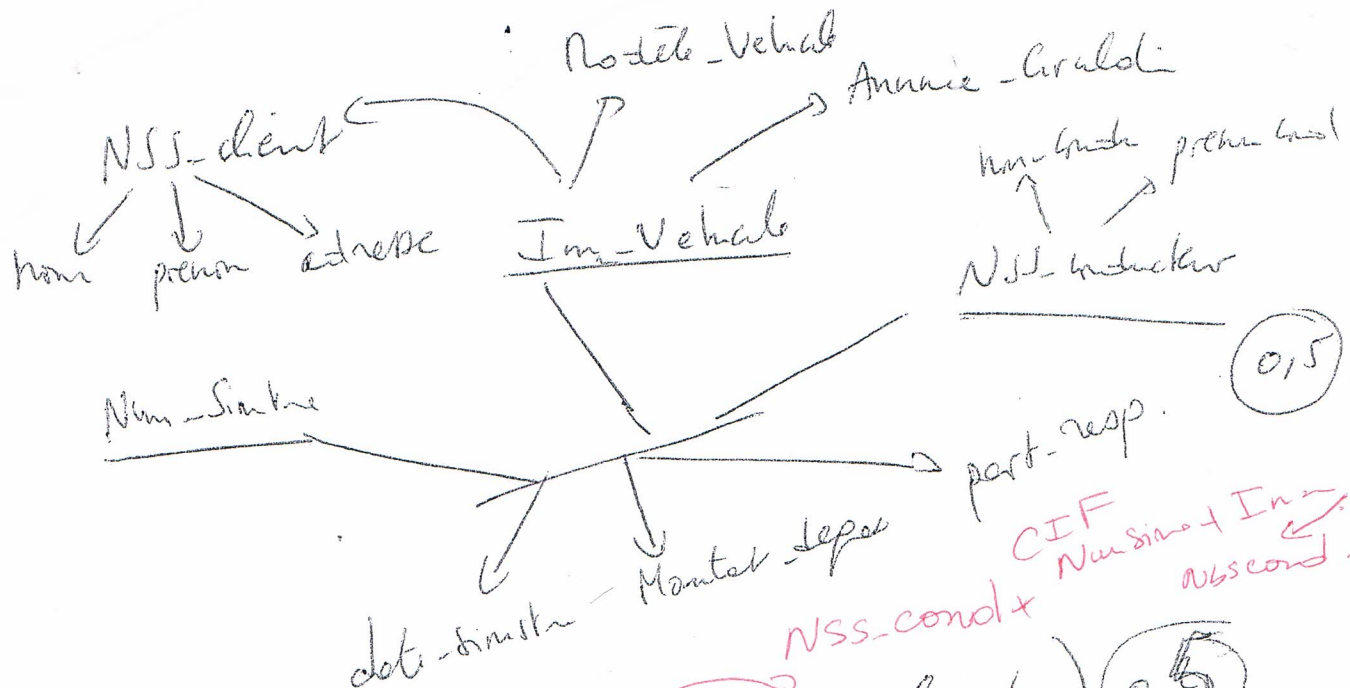
- $R_1(\underline{AB}, D)$ (0,5)
 $R_2(\underline{D}, E)$ (0,5)
 $R_3(\underline{F}, B)$ (0,25)
 $R_4(\underline{B}, C)$ (0,5)

Exo2 Imm-vehicule \rightarrow NSS-client

- 1/5 ① DF: NSS-client \rightarrow nom, prenom, adresse
 Imm-vehicule \rightarrow No de véhicule, ann-circulation
 Num-sinistre, Imm-vehicule, NSS-conducteur \rightarrow
 Date-sinistre, Montant-dépôt, part-resp.
 NSS-conducteur \rightarrow Nom-conducteur, prenom-conducteur

- ①② Attributs Multivaleés:
 un client peut avoir plusieurs véhicules
 client-vehicule (NSS-client, Imm-vehicule)
 un véhicule a fait l'objet d'un certain nombre de sinistres
 sinistre-vehicule (Imm-vehicule, Num-sinistre)

(3)



clé (Num-Sinistre, Imm-Vehab, NSS-client) 0,5

NSS-client (NSS-client, nom, prenom, ^{adresse}) 0,5

Vehab (Imm-Vehab, No-télé, Ann-Garabdi) 0,5

clat_Vehab (NSS-client, Imm-Vehab) 0,5

conducteur (NSS-conducteur nom-conduc, prenom-conduc) 0,5

Sinistre-veh (Imm-Vehab, Num-Sinistre) 0,5

P-Sinistre (Num-Sinistre, Imm-Vehab, NSS-conducteur, ~~date-sinistre~~, ~~Montant-depôt~~, part-resp) 0,5

~~Sinistre (Num-Sinistre, Imm-Vehab, NSS-conducteur, date-sinistre, Montant-depôt)~~

Num-Sinistre + Imm-Vehab → NSS-conducteur

problème

① CREATE Table Departement (
Num Dep CHAR(4) primary Key, 0,25
libelle_Dep VARCHAR(30));

CREATE Table Etudiant (
Num Etud CHAR(12) primary Key, 0,15
NSS CHAR(20) UNIQUE, 0,15
Nom Etud VARCHAR(20) NOT NULL,
prenom Etud VARCHAR(20) NOT NULL,
date Naiss Date,
Sexe CHAR(1) check in {'F', 'M'} 0,15
Departement CHAR(4) References Departement 0,15
Cycle CHAR(2) check in {'L1', 'L2', 'L3', 'M1', 'M2'}); 0,15

CREATE Table Enseignant (
Num Ens CHAR(4) primary Key, 0,15
NSS CHAR(20) UNIQUE, 0,15
Nom Ens VARCHAR(20) NOT NULL,
prenom Ens VARCHAR(20) NOT NULL,
grade CHAR(1) check in {'A', 'MA', 'CC'} 0,15
departement CHAR(4) References Departement); 0,15

CREATE Table Note (
0,15 Num Etud CHAR(12) References Etudiant,
0,15 Num Cours CHAR(4) References Cours,
0,15 Note Integer check between 0 and 20,
0,15 primary Key (Num Etud, Num Cours));

CREATE Table Cours (

EATE Table Constraints	
Non Cons	CHAR(4) primary Key, (0,1)
Intitile	VARCHAR(30) NOT NULL,
Non Enr	CHAR(1) Reference Enkript (Non Enr),
Cycle	CHAR(2) check in { 'L1', 'L2', 'L3', 'M1', 'M2' }
Department	CHAR(4) Reference Department; (0,1)

② Algebra Rielmann

a) Algebre Relativelle

e) $ENS1 = \alpha(N_{\text{non}ENS}, N_{\text{ENS}}, \text{departement: Dep})$ Enseignement.
cours

~~ENS1~~ ~~Thema~~ ~~Jointure avec~~ ~~cours~~

En 51 Theta Join

~~II~~ ~~En 51~~ ~~Dep~~ ~~En 51~~

~~II~~ ~~En 51~~ ~~Dep~~ ~~En 51~~

$\Pi_{\text{NomEn}, \text{preEn}}(\text{Cours})$

$\text{Num En} = \text{N En}$
 $\text{departent} \neq \text{Dep}$

En 51

①

b) $R_2 = \left(\text{Course} \begin{array}{c} \diagup \quad \diagdown \\ \text{NucCourse} = \text{NucCourse} \end{array} \left(\text{Elect} \begin{array}{c} \diagup \quad \diagdown \\ \text{NucElect} = \text{NucElect} \end{array} \text{Vote} \right) \right) \quad (1)$

$R2 = \Pi_{\text{NomEhd, prehu, Note}} \left(\text{Intituli} = \text{"Base de données"} \right) R1$

~~e) $R_1 = \prod_{\text{Nodes}} (\text{Cycle} = L_3' \text{ Cms})$

$R_2 = \prod_{\text{Nodes} \geq 10} (\text{Nodes} \times R_1)$

$R_3 = \prod_{\text{Nm Endors.}}$~~

$$e) R_1 = \pi_{\text{NumCours}} (\sigma_{\text{Payée} = 'L3'} (\text{Cours}))$$

$$R_2 = \pi_{\text{NomEld}, \text{NumCours}} \left(\sigma_{\text{Note} \geq 10} \left(\text{Notes} \bowtie R_1 \right) \right)$$

$$R_3 = R_2 / R_1$$

$$R_4 = \pi_{\text{NomEld}, \text{prenomEld}} \left(\text{Eldersent} \bowtie R_3 \right)$$

$$e) R_1 = \pi_{\text{NumEns}} (\sigma_{\text{Inhibé} = "BDD"} (\text{Cours}))$$

$$R_2 = \pi_{\text{NumEns}} (\text{Enseignant}) - R_1$$

$$R_3 = \pi_{\text{NumEns}} \left(\text{Enseignant} \bowtie R_2 \right)$$

$\text{NumEns} = \text{NumEns.}$

4. SQL

a) `SELECT NomEns, prenomEns`
`FROM Enseignant E, Cours C`
`where E.NumEns = C.NumEns and`
`E.departement != C.departement;`

b) `SELECT NomEld, prenomEld`
`FROM EldersentE, NotesN, Cours C`
`where E.NumEld = N.NumEld and`
`C.NumCours = N.NumCours and`
`Inhibé = "Base de données";`

1) SELECT department, Count (NumEnlist)
FROM Enlistment
Group By department ;

1

e) SELECT NumEnls
FROM Enlistment
Where NumEnlist NOT IN (SELECT NumEnlist
FROM Course
Where ~~NumEnls~~
Intitle = "BAD") ;

1

Soit la base de données sur les matchs de tennis. La définition des tables est la suivante (les clé primaires sont soulignées) :

Joueur (NJ, nomj, prénomj, âge, nationalité)

Tournoi (NT, nomtournoi, pays)

Rencontre (NJgagnant, NJperdant, NT, année, score)

NJgagant référence Joueur(NJ)

NJperdant référence Joueur(NJ)

NT référence Tournoi(NT)

Sponsor (NS, noms)

Sponsorisé (NS, NT, année, montant)

NS référence Sponsor(NS)

NT référence Tournoi(NT)

Ecrire en algèbre relationnelle les requêtes suivantes :

1. Afficher nom et prénom des joueurs de plus de 30 ans qui ont participé au tournoi de Roland Garros
2. Afficher le nom et l'âge des joueurs ayant gagné à 'Roland Garros', peu importe l'année,
3. Afficher les noms des sponsors ayant sponsorisé un tournoi ayant lieu en 'France',
4. Afficher les noms des joueurs qui n'ont jamais participé au tournoi de 'Roland Garros'
5. Afficher le nom et prénom des joueurs ayant gagné un tournoi sponsorisé par 'BNP-Paribas',
6. Afficher le nom des joueurs qui ont la même nationalité,
7. Afficher les couples (nom, prénom et nationalité du joueur gagnant, nom, prénom et nationalité du joueur perdant) des joueurs de nationalités différentes ayant eu au moins une rencontre (match) au tournoi de 'Roland Garros' de l'année 2014.
8. Afficher les noms et prénoms des joueurs ayant gagné au moins un match à 'Roland Garros' et 'Flushing Meadow' pendant l'année 2014,
9. Afficher les noms des sponsors qui sponsorisent le tournoi de 'Roland Garros' mais non le tournoi de 'Flushing Meadow',
10. Afficher les noms de sponsor ayant sponsorisé tous les tournois de 'France'.

Solution Mino Interropehon BDD

① $R_1 = \text{ThomTournoi} = \text{"Roland Garros"} \left(\text{Tournoi} \bowtie_{NT=NT} \text{Rencontre} \right)$ ①

② $R_2 = \Pi_{\text{nomj}, \text{prenomj}} \left(\text{Tape} > 30 \text{ Tournoi} \right) \bowtie_{NS=NS \text{ gagnant}} R_1$ ①pt

② $R_1 = \text{ThomTournoi} = \text{"Roland Garros"} \left(\text{Tournoi} \bowtie_{NT=NS \text{ gagnant}} \text{Rencontre} \right)$ ①pt

$R_2 = \Pi_{\text{nomj}, \text{ape}} \left(\text{Tournoi} \bowtie_{NS=NS} R_1 \right)$ ①pt

③ $\Pi_{\text{nomj}} \left(\text{Sponsor} \bowtie_{NS=NS} \left(\text{Sponsor} \bowtie_{NT=NT} \left(\text{Sponsor} = \text{France} \left(\text{Tournoi} \right) \right) \right) \right)$ ①pt

④ $R_1 = \Pi_{NS \text{ gagnant}} \left(\text{Rencontre} \bowtie_{NT=NT} \left(\Pi_{NT} \left(\text{ThomTournoi} = \text{"Roland Garros"} \right) \right) \right)$ ①pt

$R_2 = \Pi_{NS \text{ perdant}} \left(\text{rencontre} \bowtie_{NT=NT} \left(\Pi_{NT} \left(\text{ThomTournoi} = \text{"Roland Garros"} \right) \right) \right)$ ①pt

$R_3 = R_1 \cup R_2$ ①pt

$R_4 = \Pi_{NS} \text{ Tournoi} - R_3$ ①,75pt

⑤ $R_1 = \Pi_{NS} \left(\text{Thom} = \text{"BNP Paribas"} \text{ Sponsor} \right)$ ①pt

$R_2 = \Pi_{NT} \left(\text{Sponsor} \bowtie_{NS=NS} R_1 \right)$ ①pt

$R_3 = \Pi_{NS \text{ gagnant}} \left(\text{Rencontre} \bowtie_{NT=NT} R_2 \right)$ ①pt

$R_4 = \Pi_{\text{nomj}, \text{prenomj}} \left(R_3 \bowtie_{NT=NS \text{ gagnant}} \text{Tournoi} \right)$ ①pt

⑥ Renommage et Jointure sur une même table (Joueur)

- 2 solutions possibles ① renommage de tous les attributs de la table Joueur sauf nationalité. \Rightarrow Jointure Naturelle \neq
- ② renommage de tous les attributs de la table Joueur \Rightarrow Theta Jointure.

Solution avec Jointure Naturelle.

$$\text{Joueur 1} = \alpha(NJ: NJ1, Nomj: Nomj1, prenomj: prenomj1, \text{âge: âge1}) \text{ Joueur.}$$

$$\pi_{Nomj, prenomj, \text{Nationalité}} \left(\rho_{NJ1 \neq NJ} \left(\text{Joueur 1} \bowtie \text{Joueur} \right) \right) \quad (2 \text{ pts})$$

$\text{Nationalité} = \text{Nationalité}$

Solution avec Theta Jointure

$$\text{Joueur 1} = \alpha(NJ: NJ1, Nomj: Nomj1, prenomj: prenomj1, \text{âge: âge1, nationalité: nationalité1}) \text{ Joueur}$$

$$\pi_{prenomj, nomj1, nationalité} \left(\text{Joueur} \left[\begin{array}{l} \text{Nationalité} = \text{Nationalité1} \\ NJ \neq NJ1 \end{array} \right] \text{ Joueur} \right)$$

⑦ (2)

$$\text{Joueur 1} = \alpha(NJ: NJ1, Nomj: Nomj1, prenomj: prenomj1, \text{âge: âge1, Nationalité: Nat1})$$

(pts)

$$R_1 = \pi_{NJ \text{ gagnant}} \left(\sigma_{\text{Année} = 2014 \text{ and } \text{Nom Tournoi} = "Poland Games"} \left(\text{Rencontre} \bowtie \text{Tournoi} \right) \right)$$

$NJ = NJ$

(pts)

$$R_2 = \left(\text{Joueur 1} \bowtie \left(R_1 \bowtie \text{Joueur} \right) \right)$$

$Nj1 \neq Njperdant$ $NJgagnant = NJperdant$

$$R_2 = \pi_{Nomj, prenomj, Nationalité} \left(\text{Joueur 1} \left[\begin{array}{l} NJ1 = NJperdant \\ Nat1 \neq Nationalité \end{array} \right] \left(R_1 \bowtie \text{Joueur} \right) \right)$$

$NJgagnant = NJ$

$Nomj2, prenomj2, Nat1$

(8) 2 $R_2 = \prod_{NT} \left(\sigma_{\text{non Tournoi}} = \text{"Roland Garros" ou non Tournoi} = \right) \text{Tournoi}$

(circled) $R_1 = \prod_{N_{jg1}, \text{non Tournoi}} \left(\sigma_{\text{non Tournoi}} = \text{"RG" or "Flushing Meadow"} \right) \text{Tournoi}$
 $\text{non Tournoi} = \text{"RG" or "Flushing Meadow"}$
 $\text{non Tournoi} = \text{"FN"}$
 $NT = NT$

(circled) $R_2 = \alpha \left(N_{jg1}, \text{non Tournoi}, N_{\text{non Tournoi}} \right) R_1$
 $N_{\text{non Tournoi}}: N_{\text{non Tournoi}} 2, \text{premier Tournoi}: \text{premier Tournoi}$

(circled) $R_3 = \prod_{\text{non Tournoi}, \text{premier Tournoi}} R_1 \left[\begin{array}{l} N_{jg1} = N_{\text{premier Tournoi}} \text{ and } \\ \text{non Tournoi} = \text{"Roland Garros" and } \\ \text{non Tournoi} = \text{"Flushing Meadow"} \end{array} \right] \text{ } \cancel{R_2}$
lower

2 (9) chercher d'abord les sponsors qui sponsorisent le tournoi de Roland Garros et enlever dans l'ensemble trouvé ceux qui sponsorisent le Tournoi de Flushing Meadow:

(circled) $R_1 = \prod_{NS} \left(\sigma_{\text{non Tournoi}} = \text{"Roland Garros"} \right) \left(\text{Sponsorisé} \right) \text{Tournoi}$
 $NT = NT$

(circled) $R_2 = \prod_{NS} \left(\sigma_{\text{non Tournoi}} = \text{"Flushing Meadow"} \right) \left(\text{Sponsorisé} \right) \text{Tournoi}$
 $NT = NT$

(circled) $R_3 = R_2 - R_1$

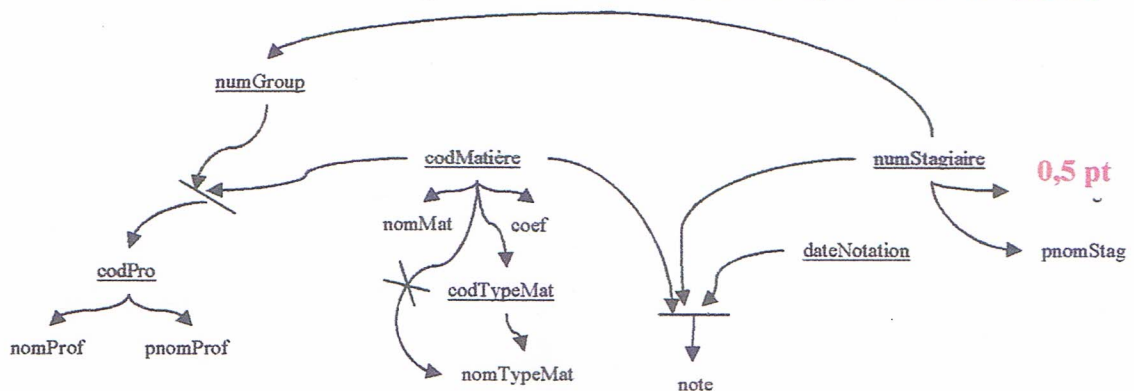
9 (10) chercher les NT de Tournoi de France
 (circled) $R_1 = \prod_{NT} \left(\sigma_{\text{pays}} = \text{France} \right) \text{Tournoi}$

(circled) $R_2 = \prod_{NS, NT} \text{Sponsorisé} / R_1$
 (circled) $R_3 = \prod_{\text{non Tournoi}} \left(R_2 \right) \text{Sponsor}$
 $NS \quad NS$

EXO 1 :

- 1) Anomalies : redondances on cite : le coefficient de la matière, le nom du type de la matière, le nom de la matière, le nom et le prénom du stagiaire, le nom et le prénom du prof. **0,75 point**
Ces anomalies de redondances induisent des mises à jour complexes et sont donc sujets à erreurs d'où non cohérence.

- 2) Graphe min des DFs : $\text{codMat} \rightarrow \text{nomTypeMat}$ est déduite (transitive), on l'élimine. **(0,5 pt)**



*à vérifier
paquet
?*

- 3) Pour l'identifiant de la relation 'Ecole', on prend les clés concaténées suivantes:

(numGroup, codMatière, codProf, numStagiaire, dateNotation) **0,75 pt**

C'est l'ensemble minimum qui détermine tous les attributs de la relation 'Ecole'

- 4) Normalisation :

1FN ok car la relation école possède un identifiant et il n'existe pas d'attributs multi-valués

2FN non ok chaque attribut doit dépendre pleinement de la clé. Or, nous avons

a) nomMat dépend seulement d'une partie de la clé à savoir codMatière (partie de la clé):

$\text{codMatière} \rightarrow \text{nomMat}, \text{coef}$ **0,25 pt**

b) nomStag, pnomStag et numGroup dépendent seulement de numStagiaire

$\text{numStagiaire} \rightarrow \text{nomStag}, \text{pnomStag}, \text{numGroup}$ **0,25 pt**

c) nomProf, pnomPro dépendent d'une partie de la clé codProf **0,25 pt**

$\text{codProf} \rightarrow \text{nomProf}, \text{pnomProf}$

d) note dépend aussi d'une partie de la clé numStagiaire, codMatière, dateNotation **0,25 pt**

$\text{numStagiaire}, \text{codMatière}, \text{dateNotation} \rightarrow \text{note}$

e) codProf dépend d'une partie de la clé numGroup, codMatière **0,25 pt**

$\text{numGroup}, \text{codMatière} \rightarrow \text{codProf}$

On décompose donc en 2FN et on aura :

Matière (codMatière, nomMat, coef, codTypeMat, nomTypeMat) **0,25 pt**

Stagiaire (numStagiaire, nomStag, pnomStag, numGroup*) 3FN OK **0,25 pt**

Professeur (codProf, nomProf, pnomProf) 3FN OK **0,25 pt**

Notation (numStagiaire, codMatière, dateNotation, note) 3FN OK **0,25 pt**

Enseigner (numGroup, codMatière, codProf) 3FN OK **0,25 pt**

3FN (toutes les DFs doivent dépendre de la clé) : **non ok pour la relation 'Matière' en effet : 0,5 pt**

La DF codTypeMat → nomTypeMat la source ne fait pas partie de la clé. Pour rendre la relation matière en 3FN on décompose la relation 'Matière' comme suit :

Matière2 (codMatière, nomMat, coef, codTypeMat*) **0,25 pt**

TypeMatière (codTypeMat, nomTypeMat) **0,25 pt**

Remarque : une décomposition directe en 3FN sans faute (clé primaire soulignée) mais en ne donnant pas d'explication sera pénalisée par 1.5pt au total pour cette question l'étudiant aura 2/3,5)

Problème :

- 6.5 1) LDD : Comme les tables se font références mutuellement, certaines tables verront les attributs clé étrangères créés après la création des tables parents. On utilisera donc la commande ALTER TABLE <nom de la table> ADD CONSTRAINTS...

Cependant on acceptera la solution des étudiants avec une pénalité de **1 point**. On acceptera aussi bien les contraintes en ligne que les contraintes nommées.

~~CREATE TABLE~~ DEPARTEMENT (NUM-DEPARTEMENT **NUMBER(3)**,
NOM-DEPARTEMENT **VARCHAR(20) NOT NULL**,
VILLE **VARCHAR(15) NOT NULL**,
MATRICULE-CHEF-DEPART **CHAR(9)**, // References employé (matricule) **0,25 pt (acceptée avec 0,5 de pénalité, solution avec alter table omise ****)**

~~CONSTRAINT~~ PK_DEP **PRIMARY KEY** (NUM-DEPARTEMENT) **0,5 pt** ;

~~CREATE TABLE~~ EMPLOYE (MATRICULE **CHAR(9)**,
NSS **CHAR(15)**,
NOMEMP **VARCHAR(20) NOT NULL**,
PRENOM **VARCHAR(20) NOT NULL**,
SEXE **ENUM('M', 'F')**,
DATE-NAISS **DATE**,
SALAIRE **NUMBER(6)**,
MATRICULE-SUPERVISEUR **CHAR(9)**, //References employé (matricule) **0,25 pt (acceptée avec 0,5 de pénalité solution avec alter table omise ****)**
NUM-DEPARTEMENT **NUMBER(3)**,

CONSTRAINT PK_EMPLOYE **PRIMARY KEY** (MATRICULE), **0,5 pt**

CONSTRAINT FK_EMP_NUMDEP_DEP **FOREIGN KEY** NUM-DEPARTEMENT **REFERENCES** DEPARTEMENT (NUM-DEPARTEMENT) **ON DELETE SET NULL ON UPDATE CASCADE**, **0,5 pt**

CONSTRAINT UN_NSS **UNIQUE**(NSS) **0,5 pt** ;

******ALTER TABLE EMPLOYE ADD CONSTRAINT FK_EMPL_MATSUP_EMPL FOREIGN KEY MATRICULE-SUPERVISEUR REFERENCES EMPLOYE(MATRICULE) ON DELETE SET NULL, ON UPDATE CASCADE;**
0,75 pt (solution idéale)

****ALTER TABLE DEPARTEMENT ADD CONSTRAINT FK_DEP_MATCHEFDEP_EMPL FOREIGN KEY MATRICULE-CHEF-DEPART REFERENCES EMPLOYE (MATRICULE) ON DELETE SET NULL ON UPDATE CASCADE; 0,75 pt (solution idéale)

CREATE TABLE PROJET (NUMPROJ CHAR(15),
NOM-PROJET VARCHAR(15) NOT NULL,
VILLE VARCHAR(15) NOT NULL,
NUM-DÉPARTEMENT NUMBER(3) REFERENCES DEPARTEMENT (NUM-DÉPARTEMENT) ON DELETE SET NULL ON UPDATE CASCADE, 0,5 pt
PRIMARY KEY (NUMPROJ) 0,5 pt);

CREATE TABLE TRAVAILLE-SUR-PROJET (MATRICULE CHAR(9) REFERENCES EMPLOYE ON DELETE CASCADE ON UPDATE CASCADE, 0,5 pt
NUM-PROJET INT(3) REFERENCES PROJET ON DELETE CASCADE ON UPDATE CASCADE, 0,5 pt
NBRE-HEURE NUMBER(3),
PRIMARY KEY (MATRICULE, NUM-PROJET) 0,5);

Remarque : les réponses sans ON delete et on update seront acceptées avec une pénalité de 0,25 au total de la question.

- 2) En règle générale, les tables sont créées dans un ordre qui privilégie d'abord la création des tables parents ensuite des tables enfants. Dans le cas où les tables se font référence mutuellement telles que les tables employés et département, on différera la création des clés étrangères après la création des tables en utilisant la commande alter table qui ajoute les contraintes des clés étrangères. (voir corrigé question 1) 0,5 pt.

Algèbre relationnelle :

- a) Donner les noms des employés ayant comme superviseur l'employé « Bouguerra ».

$R1 = \Pi_{\text{Matricule}} (\sigma_{\text{NOMEMP} = \text{'Bouguerra'}} \text{EMPLOYE})$ 0,25pt

$R2 = R1 \bowtie \text{EMPLOYE}$ 0,5 pt
MATRICULE = MATRICULE-SUPERVISEUR

$R3 = \Pi_{\text{NOMEMP}} R2$ 0,25 pt

$\Pi_{\text{NOMEMP}} (\text{EMPLOYE} \bowtie (\Pi_{\text{Matricule}} (\sigma_{\text{NOMEMP} = \text{'Bouguerra'}} \text{EMPLOYE})))$
MATRICULE = MATRICULE-SUPERVISEUR

- C) Donner les Matricules des employés qui sont dans tous les projets situés à « ALGER ».

$R1 = \Pi_{\text{NUMPROJ}} (\sigma_{\text{VILLE} = \text{'ALGER'}} \text{PROJET})$ 0,25 pt

$R2 = \Pi_{\text{MATRICULE}, \text{NUMPROJ}} (\text{TRAVAIL-SUR-PROJET})$ 0,25 pt

$R3 = R2 \div R1$ 0,5pt

$\Pi_{\text{MATRICULE}, \text{NUMPROJ}} (\text{TRAVAIL-SUR-PROJET}) \div \Pi_{\text{NUMPROJ}} (\sigma_{\text{VILLE} = \text{'ALGER'}} \text{PROJET})$

SQL