

LICENCES MATHÉMATIQUES ET INFORMATIQUE
 3ÈME ANNÉE - FORMATION INITIALE ET PAR APPRENTISSAGE
 BASES DE DONNÉES RELATIONNELLES
 RECUEIL D'EXERCICES ET ANNALES DE SUJETS DE
 PARTIEL ET D'EXAMENS
 2024-2025

Maude Manouvrier

La reproduction de ce document par tout moyen que ce soit est interdite conformément aux articles L111-1 et L122-4 du code de la propriété intellectuelle.

Table des matières

1 Exercices de TD	3
1.1 Modèle relationnel	3
1.2 Passage au relationnel	7
1.3 Algèbre relationnelle, calcul relationnel à variable nuplet et SQL	17
1.4 Dépendances fonctionnelles, décomposition de schéma et formes normales	31
2 Annale de sujets d'examens	37
2.1 Examen de Janvier 2022	37
2.1.1 Exercice 1 - Algèbre relationnelle et SQL (<i>13 points soit 43%</i>)	37
2.1.2 Exercice 2 - Dépendances fonctionnelles et SQL (<i>3 points soit 10%</i>)	39
2.1.3 Exercice 3 - Passage au relationnel (<i>4 points soit 13%</i>)	40
2.2 Partiel d'Octobre 2021	41
2.3 Examen de Juin 2021	44
2.3.1 Exercice 1 - Algèbre relationnelle, calcul relationnel à variable nuplet et SQL (<i>20 points soit 50%</i>)	44
2.3.2 Exercice 2 - Passage au relationnel (<i>7 points soit 17.5%</i>)	46
2.3.3 Exercice 3 - Dépendances fonctionnelles et normalisation (<i>13 points soit 32.5%</i>)	47
2.4 Examen de Janvier 2021	48
2.4.1 Exercice 1 - Algèbre relationnelle et SQL (<i>25 points soit 50%</i>)	48
2.4.2 Exercice 2 - Passage au relationnel et SQL (<i>5 points soit 10%</i>)	50
2.4.3 Exercice 3 - Dépendances fonctionnelles et Décomposition de schéma (<i>20 points soit 40%</i>)	50
2.5 Partiel de Décembre 2020	53
2.6 Examen de Janvier 2020	56
2.6.1 Exercice 1 - Algèbre relationnelle et SQL (<i>21 points soit 52.5%</i>)	56
2.6.2 Exercice 2 - Passage au relationnel et SQL (<i>6 points soit 15%</i>)	58
2.6.3 Exercice 3 - Décomposition de schéma (<i>13 points soit 32.5%</i>)	59
2.7 Partiel d'Octobre 2019	61
2.8 Partiel de Mars 2019	64

2.9	Examen de Mai 2019	66
2.9.1	Exercice 1 - Algèbre relationnelle et SQL (<i>23 points soit 57,5%</i>)	66
2.9.2	Exercice 2 - Dépendances fonctionnelles, formes normales et décomposition (<i>11 points soit 27,5%</i>)	69
2.9.3	Exercice 3 - Requêtes en algèbre relationnelle (<i>6 points soit 15%</i>)	69
2.10	Examen de Janvier 2019	70
2.10.1	Exercice 1 - Passage au relationnel (<i>13 points soit 32,5%</i>)	70
2.10.2	Exercice 2 - Algèbre relationnelle, calcul relationnel à variable nuplet et SQL (<i>19 points soit 47,5%</i>)	71
2.10.3	Exercice 3 - Dépendances fonctionnelles, formes normales et décomposition (<i>8 points soit 20%</i>)	73
2.11	Examen de Mars 2018	74
2.11.1	Exercice 1 - Passage au relationnel (<i>20 points soit 50%</i>)	74
2.11.2	Exercice 2 - Modèle relationnel et requêtes en algèbre relationnelle et calcul relationnel à variable nuplet (<i>20 points soit 50%</i>)	75
2.12	Examen de Mai 2018	77
2.12.1	Exercice 1 - Algèbre relationnelle, calcul relationnel à variable nuplet et SQL (<i>30 points soit 60%</i>)	77
2.12.2	Exercice 2 - Dépendances fonctionnelles, formes normales et décomposition (<i>10 points soit 20%</i>)	79
2.12.3	Exercice 3 - Requêtes en algèbre relationnelle (<i>10 points soit 20%</i>)	80
2.13	Examen de Juin 2017	81
2.13.1	Exercice 1 - Modèle relationnel et requêtes en algèbre relationnelle, en calcul relationnel à variable nuplet et en SQL (<i>25 points soit 62,5%</i>)	81
2.13.2	Exercice 2 - Dépendances fonctionnelles et décomposition (<i>15 points soit 27,5%</i>)	83
2.14	Partiel d'Octobre 2018	84
2.15	Examen Mai 2012	86
2.15.1	Exercice 1 - Requêtes d'interrogation (<i>44 points soit 55%</i>)	86
2.15.2	Exercice 2 - Dépendances fonctionnelles (<i>7 points soit 8,5%</i>)	86
2.15.3	Exercice 3- Formes normales (<i>10 points soit 12,5%</i>)	87
2.15.4	Exercice 4- Décomposition de schéma (<i>19 points soit 24%</i>)	87
2.16	Partiel de Mars 2012	88
2.16.1	Exercice 1 - Passage au relationnel (<i>11 points soit 27,5%</i>)	88
2.16.2	Exercice 2 - Requêtes en algèbre relationnelle (<i>18 points soit 45%</i>)	88
2.16.3	Exercice 3 - Requêtes en calcul relationnel à variable nuplet (<i>11 points soit 27,5%</i>)	89

1 Exercices de TD

1.1 Modèle relationnel

Exercice 1 :

Soient 2 relations dont 2 exemples d'instances sont données ci-dessous.

TABLE 1 – Une instance de la relation **Personne**

Num	Nom	Prenom	DateNaissance	Sexe
1	GAMOTTE	Albert	25/05/1972	M
2	GAMOTTE	Cyril	14/06/2008	M
3	AUBOISDORMAN	Abel	12/04/1942	M
4	AUBOISDORMAN	Jacques	23/05/1967	F
5	LO	Ella	12/08/1991	F
6	LO	Samantha	21/09/1989	F

TABLE 2 – Une instance de la relation **Parente**

IDParent	IDEnfant
1	2
3	4
4	5
4	6

1. Définir la(les) clé(s) primaire(s) que pourrait¹ avoir chacune des relations, en analysant uniquement les instances (sans analyser la sémantique du schéma).
2. Définir la(les) clé(s) primaire(s) de chacune des relations, en analysant uniquement la sémantique sous-jacente du schéma.
3. Définir, lorsqu'elles existent, les clés étrangères de chacune des relations, en analysant uniquement la sémantique sous-jacente du schéma.
4. Définir la contrainte de domaine associée à l'attribut **Sexe**.
5. Définir une contrainte entre les 2 attributs de la relation **Parente**, en analysant la sémantique sous-jacente du schéma.

Exercice 2 :

On souhaite créer une bases de données gérant les données relatives à des étudiants, des cours et à des notes d'étudiants. Un étudiant peut suivre plusieurs cours et obtenir plusieurs notes (note de CC, note d'Examen à la première session, note d'Examen à la deuxième session par exemple) pour un même cours. Pour chaque cours, un étudiant a un seul triplet de notes (note de CC, note d'Examen 1ère Session, note d'Examen 2ème Session²).

1. Rappel : Une clé est définie pour un schéma et doit être valable pour toutes les instances possibles pouvant être associées au schéma.

2. Cette note peut ne pas avoir de valeur si l'étudiant n'a pas passé de 2ème session pour l'enseignement concerné.

Parmi les choix ci-dessous : (i) Quel est à votre avis le "meilleur" modèle relationnel pour répondre aux besoins du paragraphe précédent ? (j) Expliquer les avantages et inconvénients, lorsqu'il y en a, de chaque solution proposée.

1. Mettre toutes les informations sur les étudiants, les cours (intitulé, niveau, enseignant, etc.) et sur les notes dans une même relation.
2. Créer une relation **Etudiant**, une relation **Cours** et mettre toutes les informations sur les notes dans la relation **Etudiant**. Préciser les clés étrangères.
3. Créer trois relations **Etudiant**, **Cours** et **Notes** et définir des clés étrangères dans la relation **Notes** faisant référence aux deux autres relations.

Exercice 3 :

Soit le schéma de base de données ci-dessous contenant deux relations et permettant de gérer des clients et leurs comptes en banque :

Client(ClientID, Nom, Prenom, Adresse)
de clé primaire **ClientID**

Compte(CompteID, ClientID, Solde, TypeCompte)
de clé primaire **CompteID**
et de clé étrangère **ClientID** faisant référence à la clé primaire de la relation **Client**

Les tableaux 3 et 4, ci-dessous, représentent respectivement une instance de chacune de ces relations :

TABLE 3 – Une instance de la relation **Client**

ClientID	Nom	Prenom	Adresse
1	GAMOTTE	Albert	10 rue des Alouettes 75017 Paris
2	DEBECE	Aude	26 av. des blés 91000 Corbeil
3	PABIEN	Yvon	145 allée du Général 92000 Boulogne

TABLE 4 – Une instance de la relation **Compte**

CompteID	ClientID	Solde	TypeCompte
1	1	1200	PEL
2	1	-600	Compte Courant
3	2	3400	Compte Courant
4	3	3400	Compte Courant

Indiquer ce que va faire le SGBD si un utilisateur insère le nuplet suivant (5, 4, 1200, 'Compte Courant') dans la relation **Compte**.

Exercice 4 :

Soit la relation **Enseignant** dont une instance est présentée dans le tableau 5. La clé primaire de la relation est **EnseignantID**. L'attribut **DepartementID** est une clé étrangère faisant référence à la clé primaire de la relation **Departement** (non représentée dans l'énoncé de l'exercice).

TABLE 5 – Une instance de la relation **Enseignant**

EnseignantID	DepartementID	Nom	Prenom	email
1	1	Bertrand	Patrice	Patrice.Bertrand@dauphine.fr
2	1	Hoffmann	Marc	Marc.Hoffmann@dauphine.fr
3	1	Belhajjame	Khalid	Khalid.Belhajjame@dauphine.fr
4	1	Manouvrier	Maude	manouvrier@lamsade.dauphine.fr
5	1	Negre	Elsa	negre@lamsade.dauphine.fr

TABLE 6 – Une instance de la relation **Master**

MasterID	NomMaster	ResponsableID	DepartementID
1	ISF	1	1
2	Actuariat	2	1
3	MIAGE-SITN App.	3	1

Soit la relation **Master** dont une instance est présentée dans le tableau 6. La clé primaire de la relation est **MasterID**. L'attribut **ResponsableID** est une clé étrangère faisant référence à la clé primaire de la relation **Enseignant**. L'attribut **DepartementID** est une clé étrangère faisant référence à la clé primaire de la relation **Departement** (non représentée dans l'énoncé de l'exercice).

1. Peut-on insérer un nuplet (3,'MIAGE-IF',2,1) dans la relation **Master** ? Expliquer.
2. Peut-on insérer un nuplet (5,'MIAGE-ID',4,1) dans la relation **Master** ? Expliquer.
3. Peut-on supprimer le nuplet correspondant à l'enseignant 'Khalid Belhajjame' ?

Exercice 5 :

Soit la relation **Cours**, dont le schéma est composé de 6 attributs³, et dont un exemple d'instance est présenté ci-dessous :

CoursID	MasterID	Intitulé	Description	Obligatoire	EnseignantID
1	1	BD	Introduction aux BD	NON	3
2	3	C++	Programmation	OUI	4
3	1	C++	Programmation	NON	4

- En observant les nuplets (donc uniquement pour cette instance) ci-dessus, la clé primaire⁴ de cette relation **Cours** peut-elle être (expliquer) :
 - L'attribut **EnseignantID** ?
 - L'attribut **MasterID** ?
 - L'attribut **Intitulé** ?
 - Le couple d'attributs (**EnseignantID**, **MasterID**) ?
 - Les 6 attributs ?
- On veut créer une relation **Reservation** permettant de connaître les différentes salles réservées pour chaque enseignement avec les dates associées.
 - Donner le schéma de cette relation. Vous préciserez bien les clés primaires et les clés étrangères le cas échéant de la relation.
 - Si on suppose que l'on a choisi une clé artificielle **ReservationID**. Comment peut-on s'assurer qu'une même salle ne puisse pas être réservée plusieurs fois à la même date et une même heure de début pour le même enseignement ?

Exercice 6 ⁵ :

Une association d'étudiants dispose d'une base de données pour réaliser le suivi de ses anciens élèves. Un membre de l'association est soit un ancien étudiant soit un étudiant actuel. Certains membres de l'association sont des membres actifs car ils occupent une fonction particulière : trésorier, président, secrétaire, ...

La base de données est représentée par le schéma relationnel suivant :

```

Membre(CodeMb, NomMb, PrénomMb, AnnéePromotionMb, AdresseMb, CpMb, VilleMb, TelMb)
  de clé primaire : CodeMb
Entreprise(CodeEnt, LibelléEnt, AdresseEnt, VilleEnt, CpEnt, SecteurEnt)
  de clé primaire : CodeEnt
Travailler(CodeMb, CodeEnt, PosteOccupé)
  de clé primaire (CodeMb, CodeEnt)
  de clé étrangère CodeMb, faisant référence à la clé primaire de Membre.
  et de clé étrangère CodeEnt, faisant référence à la clé primaire Entreprise.
Fonction(CodeF, LibelléF, CodeMb)
  de clé primaire CodeF
  de clé étrangère CodeMb, faisant référence à la clé primaire de Membre.
```

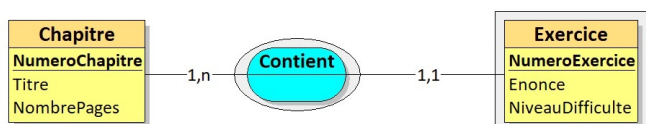
-
- On suppose qu'il existe une relation **Enseignant**(**EnseignantID**, **Nom**, **Prenom**, ...).
 - Rappel : une clé primaire doit être définie pour toutes les instances possibles de la relation. Dans cet exercice, pour débiter, vous définirez la clé primaire uniquement pour l'instance donnée en exemple.
 - Repris et adapté de ecogest.info.free.fr/Cours/GSI/Rev-ExoMR.doc

Un utilisateur de la base de données constate qu'il ne peut pas stocker tous les postes occupés par un membre dans la même entreprise. Expliquer la raison de ce problème et proposer une solution pour résoudre le problème.

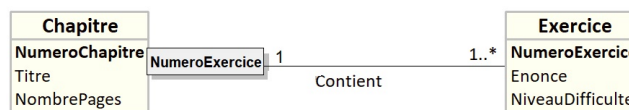
1.2 Passage au relationnel

Les exercices de cette section sont issus d'anciens sujets d'examen. Les schémas de modélisation en couleur ont été réalisés sous le logiciel <https://www.looping-mcd.fr/>.

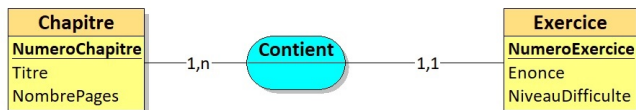
Exercice 7 :



Modèle 1 en modélisation Entité / Association



Modèle 1 en modélisation UML



Modèle 2 en modélisation Entité / Association



Modèle 2 en modélisation UML

Les schémas de modélisation ci-dessus (représentés en Entité/Association à gauche et en UML à droite) représentent deux modèles différents, le modèle 1 et le modèle 2. Le premier attribut souligné de chaque ensemble d'entités ou le premier attribut en gras de chaque classes correspond à l'identificateur.

1. Indiquer, en français, ce que représente chaque modèle et indiquer leurs différences.
2. Donner le modèle relationnel correspondant à chacun des deux modèles 1 et 2 (vous préciserez bien les clés primaires et les clés étrangères des différentes relations).

Pour chacun des exercices 8 à 17, déduire le schéma relationnel de la base de données correspondante à chacun des énoncés suivants.

Vous préciserez les clés primaires des relations en les soulignant ainsi que les clés étrangères en les signalant par un # et en précisant à quoi elles font référence.

Dans votre schéma relationnel, chaque relation doit être spécifiée de la manière suivante : $R_{Nom}(att_1, \dots, att_n)$ où R_{Nom} est le nom de la relation et att_1, \dots, att_n sont des noms d'attributs. Le nom de la relation doit obligatoirement avoir un lien avec les noms des ensembles d'entités (classes) ou des associations. Vous donnerez des explications claires et concises du passage au relationnel. Vous préciserez notamment pourquoi et comment vous créez ou modifiez certaines relations (1 ligne d'explication maximum par relation).

Exercice 8 : Voir question en haut de la page 7

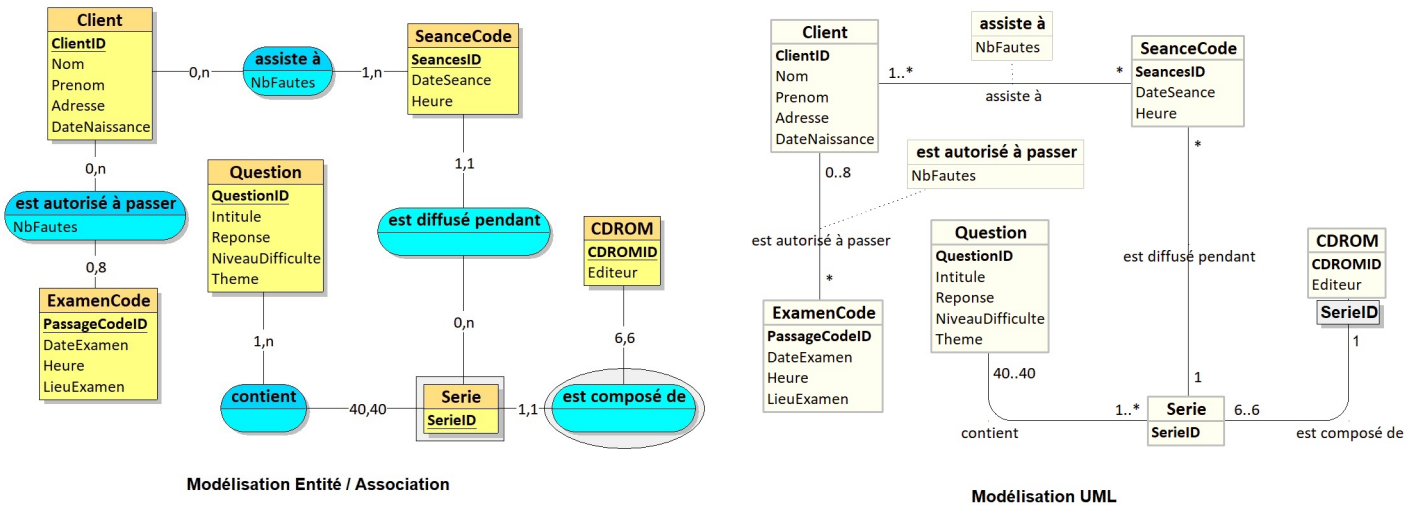


FIGURE 1 – Schémas de modélisation de l'exercice 8.

Le schéma de modélisation de la figure 1 (représenté en Entité/Association à gauche et en UML à droite) modélise la base de données des examens théoriques du code de la route d'une auto-école. Chaque client de l'auto-école assiste à plusieurs séances de code (autant qu'il le souhaite). A chaque séance de code, le directeur de l'auto-école choisit une série de questions sur un CD-ROM. Chaque CD-ROM est composé de 6 séries, numérotées de 1 à 6. Chaque série est composée de 40 questions (numérotées de 1 à 40). Une même question peut apparaître dans plusieurs séries avec un numéro d'ordre pour chaque série ; par exemple une même question peut apparaître comme question N°2 de la série 5 du CD-ROM 15 et comme question N°12 de la série 3 du CD-ROM 4. Une même série peut être projetée plusieurs fois à des séances différentes. Lorsqu'un élève assiste à une séance, il obtient le nombre de fautes (une note sur 40) qu'il a fait pour la série passée pendant la séance. Lorsqu'un élève a obtenu, au cours des quatre dernières séances auxquelles il a assistées, un nombre de fautes inférieur ou égal à 5, le directeur de l'auto-école l'autorise à passer l'examen théorique du code de la route à une date donnée (un seul examen pour une date donnée). L'auto-école ne peut présenter que 8 élèves maximum à chaque date d'examen. Les élèves ayant obtenu plus de 5 fautes à l'examen sont recalés et doivent assister de nouveau à des séances de code avant de pouvoir se représenter à l'examen. **Les identificateurs des ensembles d'entités/classes sont les attributs se terminant par ID, ils sont soulignés en E/A et en gras en UML.**

Exercice 9 : Voir question en bas de la page 7

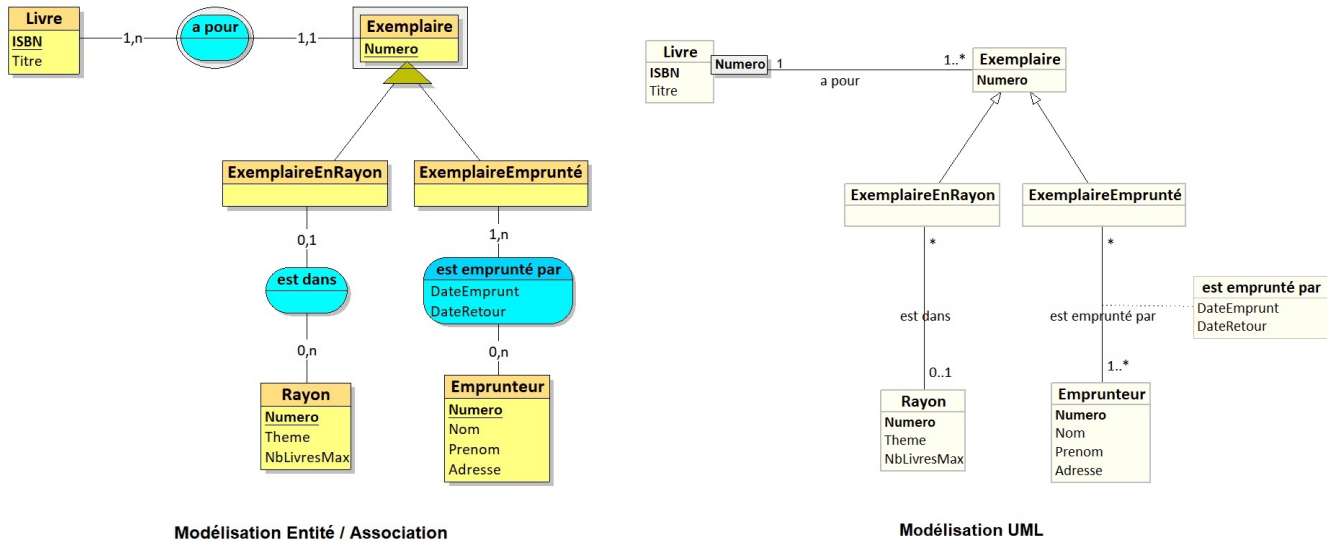


FIGURE 2 – Schémas de modélisation de l'exercice 9.

Le schéma de modélisation de la figure 2 (représenté en Entité/Association à gauche et en UML à droite) modélise une base de données gérant les ouvrages d'une bibliothèque. Chaque livre de la bibliothèque est identifié par un ISBN et possède un titre. A chaque livre correspond plusieurs exemplaires, numérotés à partir de 1. Un exemplaire d'un livre donné peut être en rayon ou être emprunté. Lorsqu'un exemplaire est emprunté, la base de données doit pouvoir indiquer par qui, à quelle date et quelle est la date de retour. Pour chaque emprunteur (identifié par un numéro de carte de bibliothèque unique), la base de données stocke le nom, le prénom et l'adresse. Chaque rayon de la bibliothèque est identifié par un numéro, et est caractérisé par un thème et un nombre de livres maximum. **Les identificateurs sont soulignés en E/A et sont en gras en UML.**

Exercice 10 : Voir question en bas de la page 7

Le schéma de modélisation de la Figure 3 (représenté en Entité/Association à gauche et en UML à droite sur la page suivante) modélise une base de données gérant des revues et les articles de ces revues. Chaque revue paraît sous la forme de numéros, identifiés par un nombre relatif à la revue et à l'année en cours (ex. le numéro N°12 de *Linux Magazine* en 2009 est différent du numéro N°12 de *Linux Magazine* en 2010). Un article peut faire référence à d'autres articles (en précisant le numéro et la revue dans lesquels l'article référencé a été publié). **Les attributs soulignés en E/A et en gras en UML correspondent aux identificateurs.** Le rectangle en gras dans le schéma entité/association représente un agrégat.

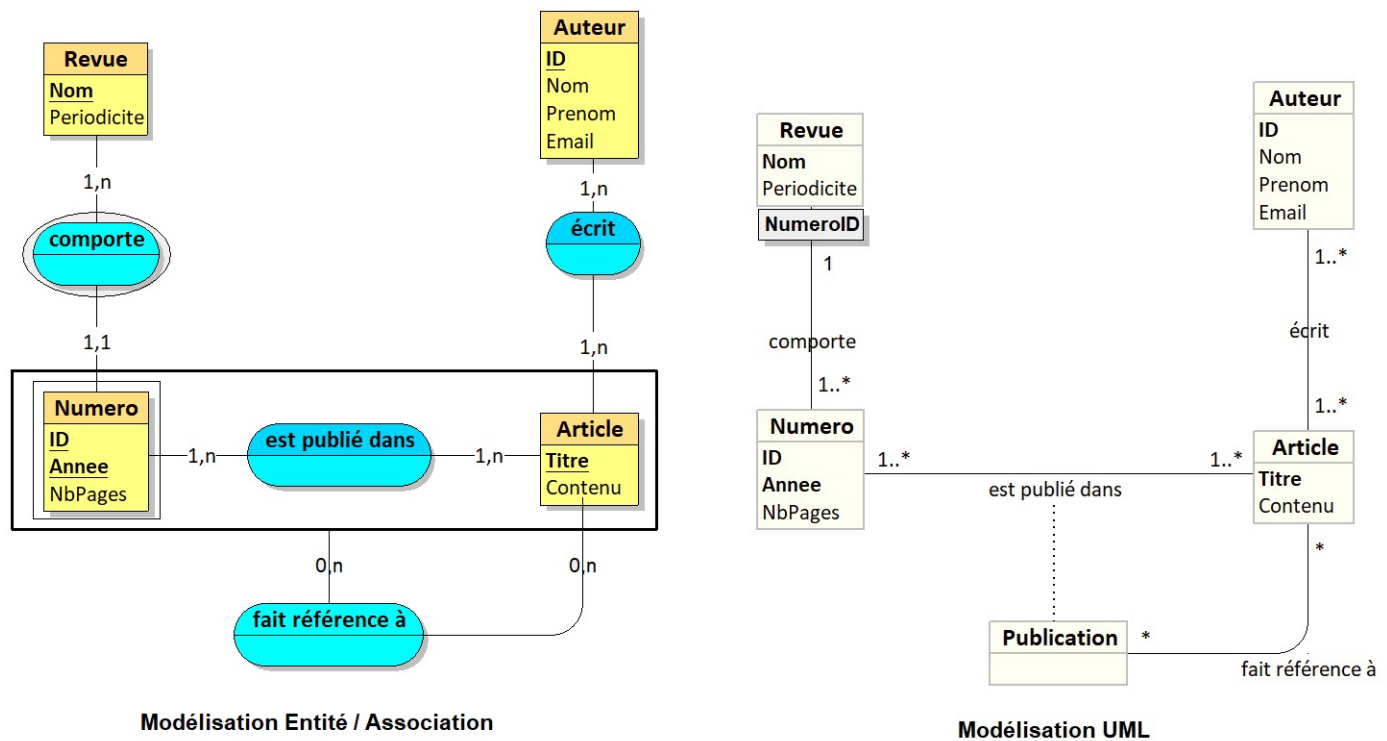


FIGURE 3 – Schémas de modélisation de l'exercice 10.

Exercice 11 : Voir question en bas de la page 7

Le schéma de modélisation de la figure 4 (représenté en Entité/Association à gauche et en UML à droite) modélise une base de données permettant de gérer des participants à des courses à pied et leurs résultats respectifs à différentes courses. Un participant est caractérisé par un nom, un prénom, un sexe, une adresse, un téléphone, un email et une date de naissance. Un participant peut être licencié ou non. Lorsqu'il est licencié, il possède un numéro de licence et appartient à un club sportif. Un club sportif est caractérisé par un nom et une adresse. Chaque participant participe à une ou plusieurs courses. Une course est caractérisée par un nom et une date (ex. Marathon de Paris - 9 avril 2006) ainsi qu'un numéro d'édition⁶, un nombre de kilomètres, un lieu de départ et un lieu d'arrivée. A chaque course courue par un participant, on enregistre, dans la base de données, le numéro de dossard du participant ainsi que le temps réalisé. Les identificateurs sont soulignés en E/A et en UML⁷.

La base de données doit permettre de répondre à des requêtes telles que : Quels sont les noms et prénoms des participants ayant couru la course "Paris-Versailles" du 25 Septembre 2005 ? Quel était le numéro de dossard de Yvon Kourire à la course "20 Kilomètres de Paris" du 16 octobre 2009 ? En combien de temps, Ella Gagnet a-t-elle couru la course "les Foulées du Luxembourg" du 2 octobre 2009 ? Quels sont les participants licenciés du club "Les Bled Runners" ? A quelle date a eu lieu la première édition de la course "Marseille-Cassis" ?

6. Par exemple, le Marathon de Paris qui a eu lieu le 9 avril 2006 correspondait à la 30ème édition.

7. Attention, il s'agit d'une petite liberté prise avec UML, les attributs soulignés ici ne correspondent pas à des attributs dérivés mais aux identificateurs (pour ne pas les oublier lors du passage au relationnel!!).

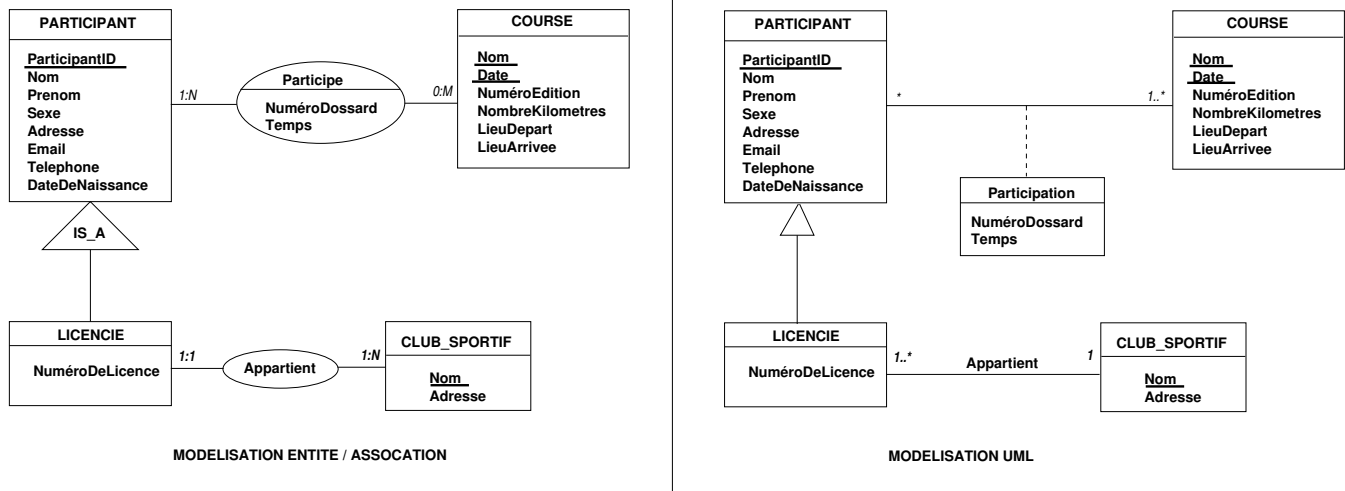
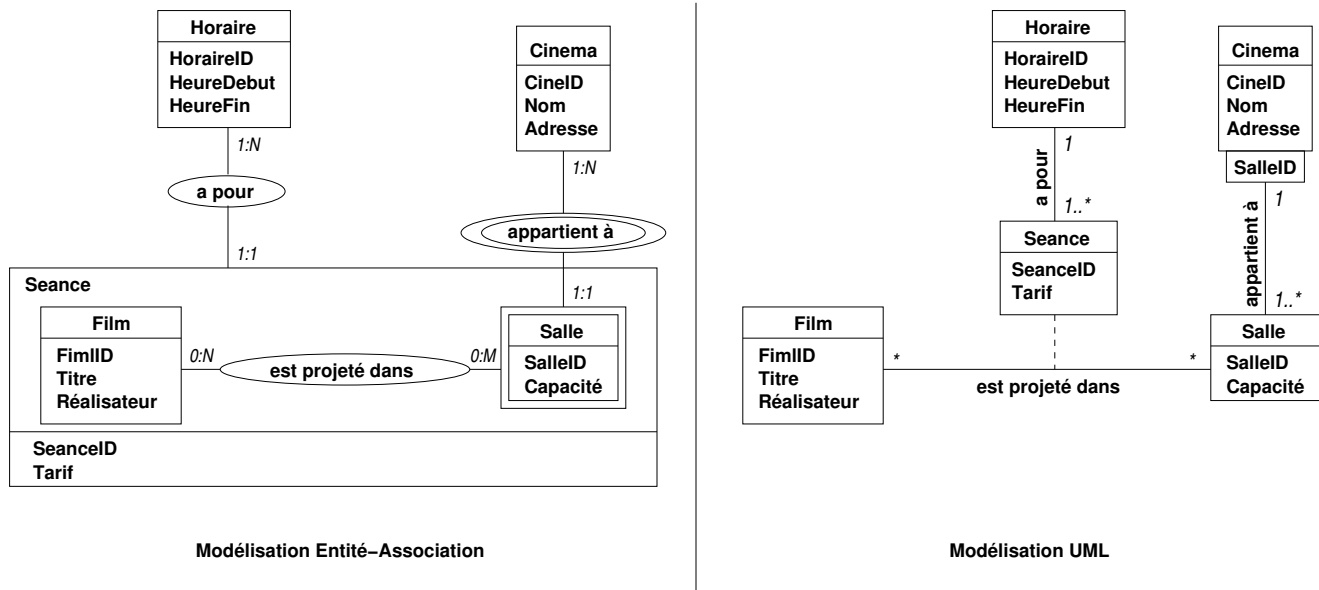


FIGURE 4 – Schémas de modélisation de l'exercice 11.

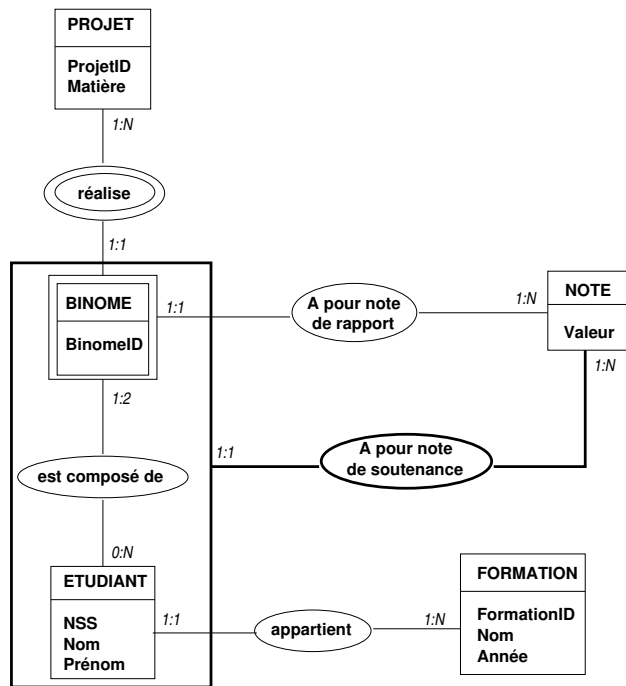
Exercice 12 : Voir question en bas de la page 7



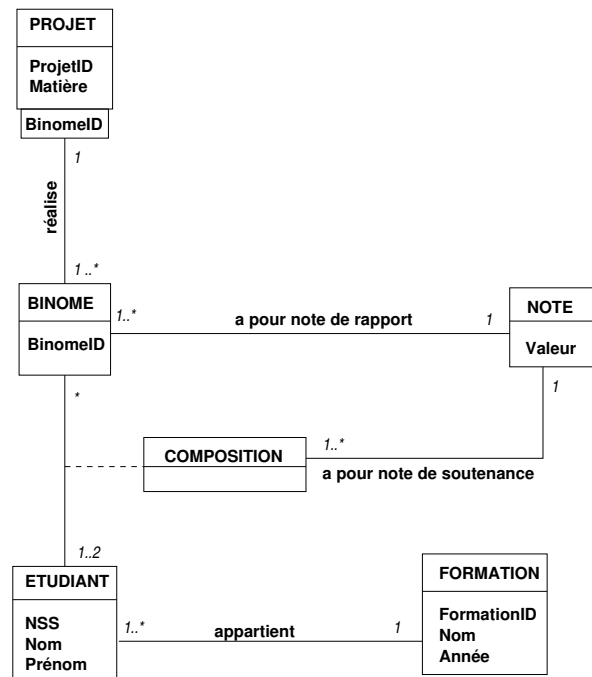
On souhaite définir une base de données permettant de gérer des séances de cinéma, dont le schéma de modélisation est présenté ci-dessus (modélisation en entité-association à gauche et en UML à droite). Les attributs dont le nom se terminent par ID correspondent aux identifiants. Les films sont décrits par leur titre et le nom de leur réalisateur. Ils sont projetés dans des salles dont le numéro (*SalleID*) est relatif au cinéma (il y a une salle numéro 1 au 'MK2 bibliothèque' et une salle numéro 1 au 'Pathé Wepler', par exemple). Chaque séance possède un identifiant et un tarif et représente un couple (*Film*,*Salle*) qui est associé à un et un seul horaire.

Exercice 13 : Voir question en bas de la page 7

Le schéma de modélisation qui suit (représenté en Entité/Association à gauche et en UML à droite) modélise la base de données des projets d'une Université. Les étudiants appartiennent à des formations (ex. MIAGE 2ème année, MIAGE Apprentissage 2ème année etc.) et réalisent des projets (ex. un projet de Bases de Données) en binôme. Un binôme peut être composé d'un ou deux étudiants. Les binômes sont identifiés par des numéros relatifs au projet (le binôme numéro 1 du projet de Bases de Données peut être différent du binôme numéro 1 du projet de Recherche Opérationnelle par exemple). Chaque binôme se voit attribuer une note correspondant à la note du rapport du projet et chaque membre du binôme se voit de plus attribuer une note personnelle de soutenance de projet. **Le premier attribut de chaque ensemble d'entités ou classes est l'identificateur.**



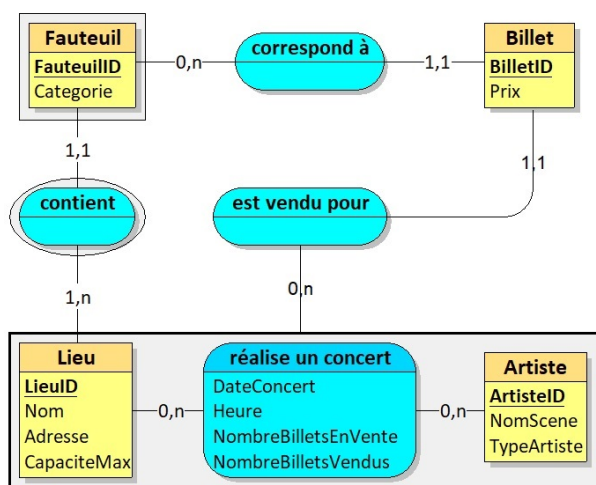
Modélisation Entité / Association



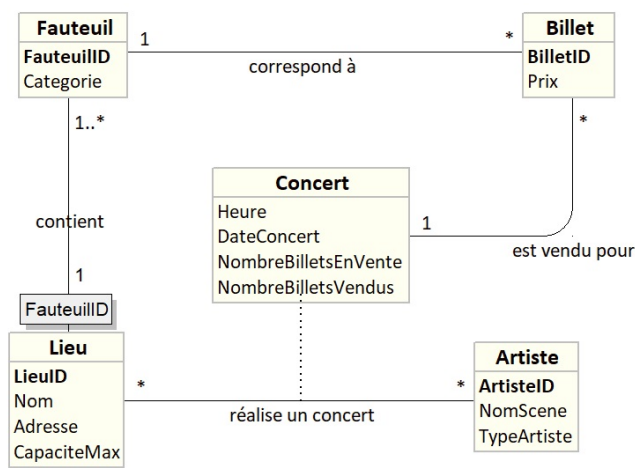
Modélisation UML

Exercice 14 : Voir question en bas de la page 7

Le schéma de modélisation ci-dessous (en Entité/Association et en UML) modélise une base de données permettant de gérer des concerts et des salles de concerts. Des artistes réalisent des concerts dans des lieux, à des dates et des heures donnés. Chaque lieu contient des fauteuils dont les numéros sont relatifs au lieu (ex. le fauteuil 1 du Zenith de Paris et le fauteuil 1 de l'Olympia). Des billets sont vendus pour chaque concert, un billet étant associé à un fauteuil donné. Il ne peut pas y avoir plus de billets vendus que le nombre de billets en vente. La valeur de l'attribut **NombreBilletsEnVente** d'un concert est fixe. En revanche, la valeur de l'attribut **NombreBilletsVendus** d'un concert est calculée en fonction du nombre de billets réellement vendus. Le domaine de l'attribut **TypeArtiste** peut être par exemple⁸ Chanteur, Musicien, Groupe, Slameur. **Les attributs dont le nom se termine par ID sont les identificateurs d'ensembles d'entités ou de classes.** Ces attributs pourront être des entiers incrémentés automatiquement. *Vous pouvez ajouter des attributs, si vous le souhaitez et afin d'améliorer la modélisation proposée.* La base de données doit permettre de répondre à des requêtes telles que : Quel artiste a réalisé un concert à 'La Cigale' le '18 Octobre 2006' ? Combien y a-t-il eu de billets vendus pour le concert de 'Ayo' au 'Nouveau Casino' le '12 Septembre 2006' ? Combien reste-t-il de billets à vendre pour le concert de 'Grand Corps Malade' au 'Zenith' ? Quel artiste n'est jamais passé au 'Stade de France' ? etc.



Modélisation Entité / Association

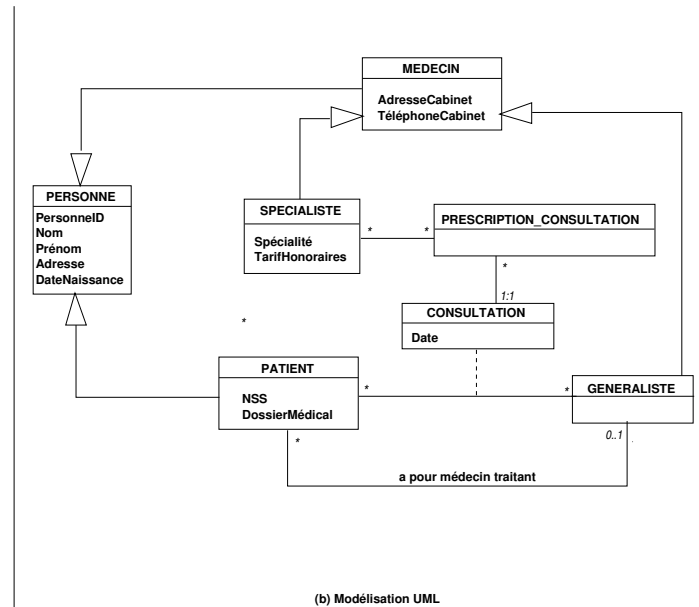
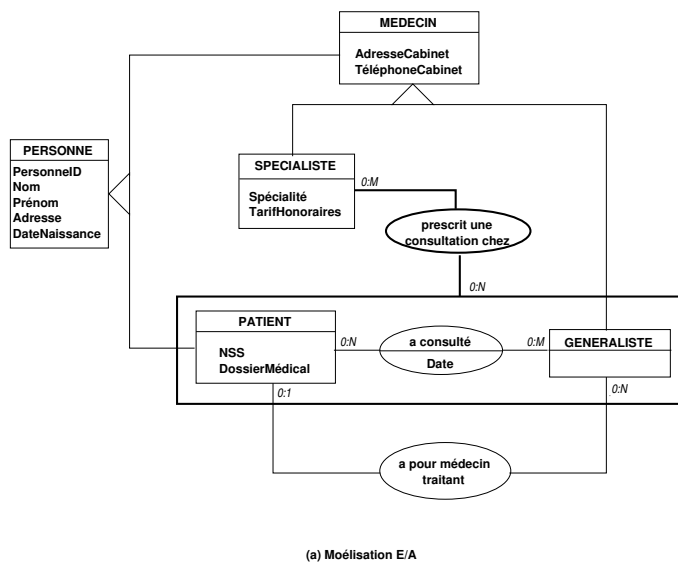


Modélisation UML

8. Vous pouvez modifier le domaine de cet attribut à votre guise.

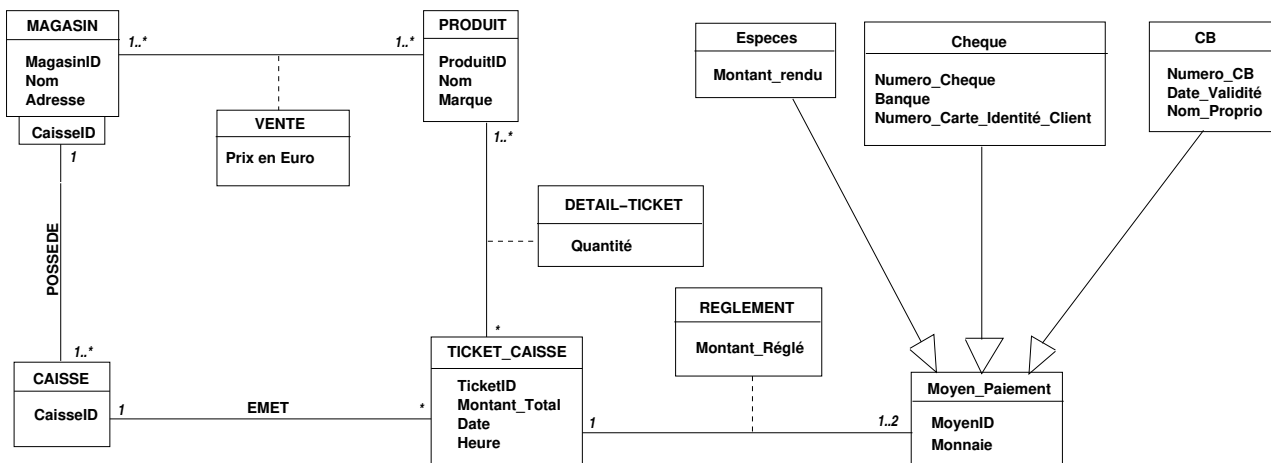
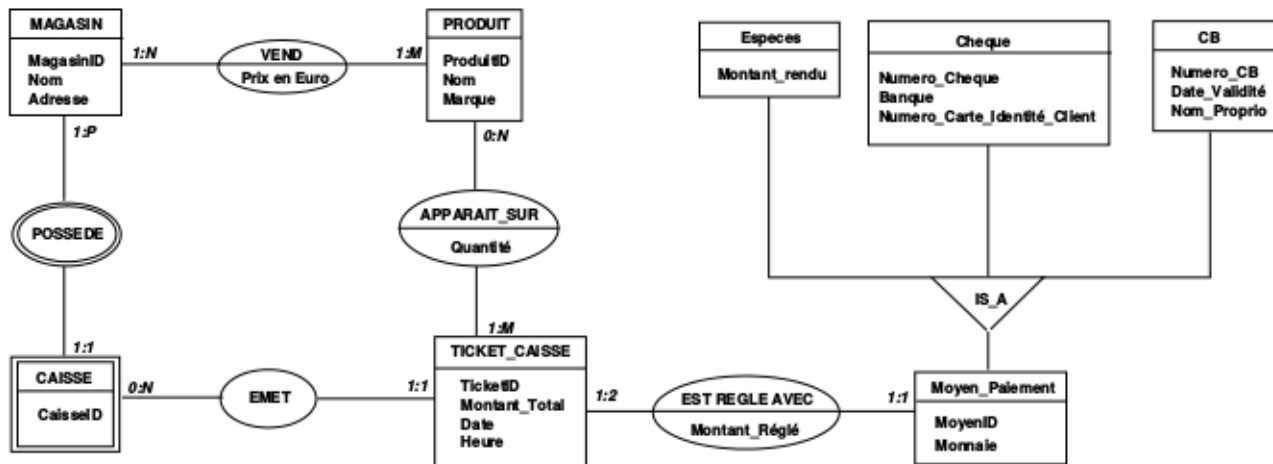
Exercice 15 : Voir question en bas de la page 7

Le schéma de modélisation ci-dessous (représenté en Entité/Association à gauche et en UML à droite) modélise une base de données gérant des patients, leur médecin traitant et des médecins spécialistes. Une personne est soit un patient, soit un médecin, un médecin pouvant être malade et devenir patient. Un médecin est soit généraliste, soit spécialiste, un généraliste ne pouvant pas être spécialiste et inversement. Un patient a choisi ou non son médecin traitant. Un patient consulte, à une date donnée, un généraliste. Lors des consultations, le généraliste peut ou non prescrire une ou plusieurs consultations chez un spécialiste. Le rectangle en gras dans le schéma entité/association représente un agrégat. **L'attribut PersonneID est un identificateur.**

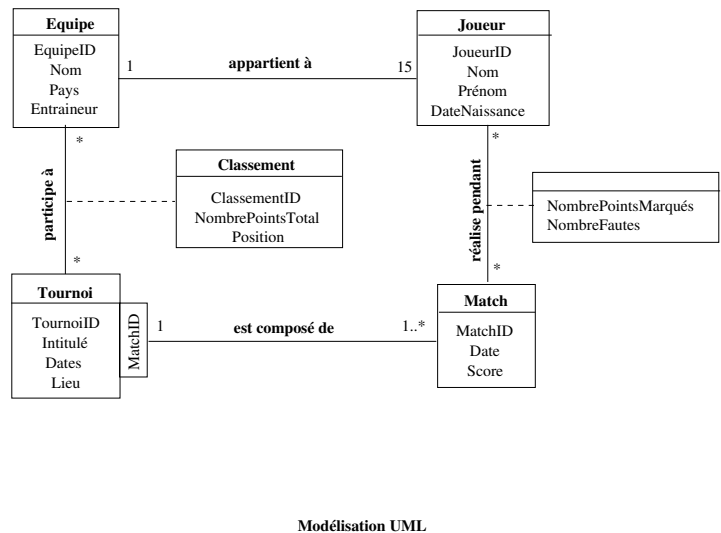
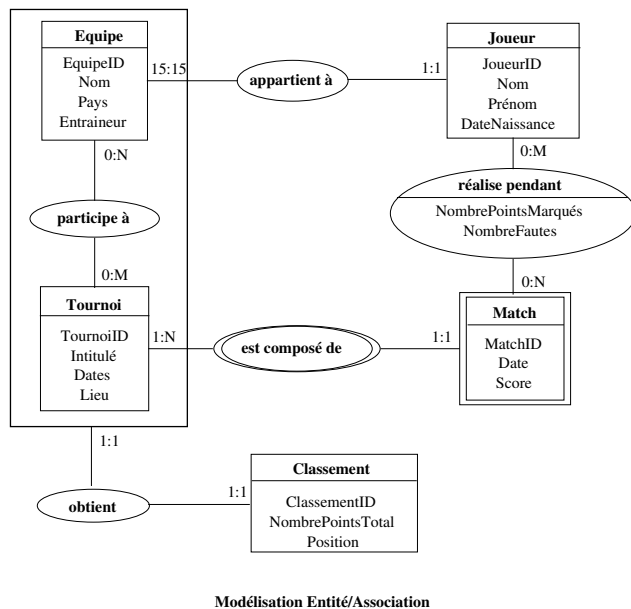


Exercice 16 : Voir question en bas de la page 7

Le schéma de modélisation ci-dessous modélise la base de données d'une chaîne de supermarchés. La sémantique des schémas est claire. Il est à noter néanmoins, que les supermarchés permettent aux clients de régler leurs achats par deux moyens de paiement différents.

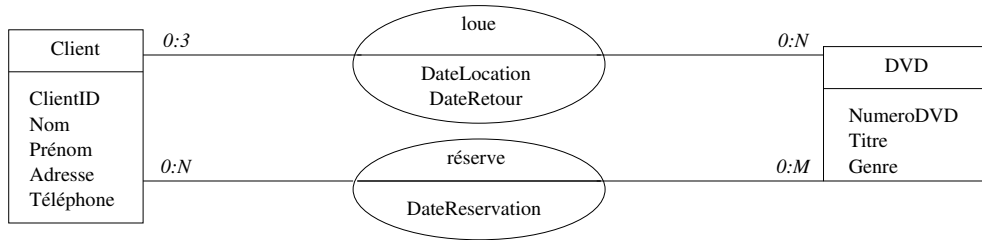


Exercice 17 : Voir question en bas de la page 7

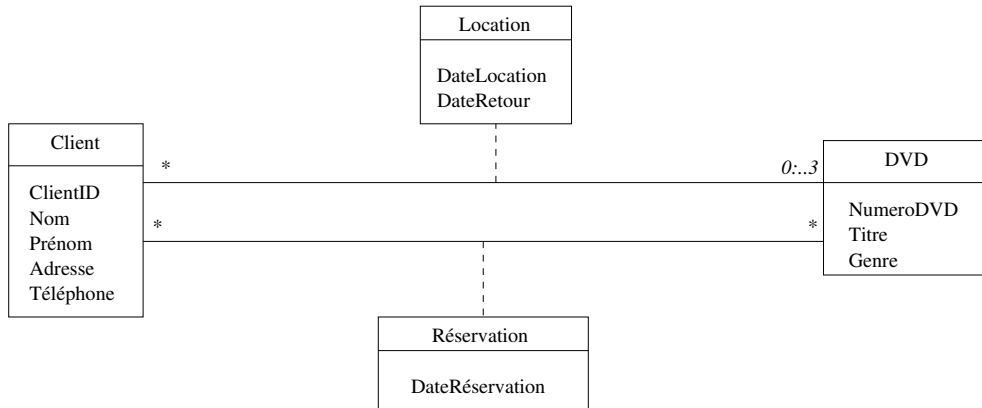


On souhaite définir une base de données permettant de gérer des tournois de rugby, dont le schéma de modélisation est présenté ci-dessus (modélisation en entité-association à gauche et en UML à droite). **Les attributs dont le nom se terminent par ID correspondent aux identificateurs.** Les équipes sont composées de 15 joueurs. Chaque équipe, lorsqu'elle participe à un tournoi, obtient un classement indiquant le nombre de points total marqués par les joueurs de l'équipe et sa position dans le classement général du tournoi. Chaque joueur, lorsqu'il participe à un match, obtient un nombre de points marqués et un nombre de fautes (pouvant être égal à zéro). L'identificateur de chaque match est relatif à chaque tournoi (par exemple il existe le match numéro 1 du "tournoi des 6 nations" et le match numéro 1 de la "coupe du monde").

Exercice 18 : Voir question en haut de la page 7



(a) Modélisation Entité-Association



(b) Modélisation UML

Le schéma de modélisation ci-dessus (représenté en Entité/Association à haut et en UML en bas) modélise la base de données d'une boutique de location de DVD. L'attribut **ClientID** identifie un client. L'attribut **NumeroDVD** identifie un DVD.

1.3 Algèbre relationnelle, calcul relationnel à variable nuplet et SQL

Les exercices 22 à 27, 29, 33 et 35 à 38 et 32 et 39 sont issus d'anciens sujets d'examen. Merci à A. Doucet, S. Gançarski, H. Garcia Molina, G. Jomier, J. Ullman, P. Valduriez, J. Widom, R. Ramakrishnan et J. Gehrke qui retrouveront certaines de leurs propositions adaptées dans les autres exercices de cette section.

Exercice 19 :

Soient deux relations $R(A, B, C)$ et $S(B, C, D)$ avec $a \in \text{dom}(A)$ et $b \in \text{dom}(B)$. Les expressions suivantes sont-elles bien formées ?

1. $r \cup s$
2. $\Pi_B(r) - \Pi_B(s)$
3. $\sigma_{B=b}(r)$
4. $\sigma_{(A=a) \wedge (B=b)}(r)$

Exercice 20 :

Soient les deux relations suivantes r et s dont des instances sont données dans les tables 7 et 8, avec $Dom(Cours) = \{Math, Physique, Latin\}$; $Dom(Etudiant) = \{Toto, Lulu\}$; $Dom(Note) = \{A, B, C\}$ et $Dom(Prof) = \{Martin, Dupond, Durand\}$.

Calculer le résultat des requêtes suivantes :

TABLE 7 – Une instance r de l'exercice 20

Cours	Etudiant	Note
Math	Toto	A
Math	Lulu	B
Latin	Toto	C
Physique	Toto	A

TABLE 8 – Une instance s de l'exercice 20

Cours	Prof
Math	Martin
Latin	Durand
Physique	Dupond

1. $\sigma_{Cours='Math'}(r)$
2. $\Pi_{Cours}(r) - \Pi_{Cours}(s)$
3. $\Pi_{Cours}(s) - \Pi_{Cours}(r)$
4. $\Pi_{Etudiant, Cours}(r) \div \Pi_{Cours}(s)$
5. $\Pi_{Etudiant}[r \div \Pi_{Cours}(s)]$
6. $\Pi_{Prof}[\sigma_{(Note='A') \wedge (Etudiant='Toto')}(r \bowtie s)]$

Exercice 21 :

Soit une base de données contenant de relations de schéma $R(A, B, D)$ et $S(A, B)$ (avec tous les attributs de même domaine) et d'instances r (voir table 9) et s (voir table 10) :

1. Quel est le schéma et quels sont les n-uplets de $j = \Pi_D(r \bowtie s)$ où la jointure est naturelle, c'est-à-dire qu'elle porte sur tous les attributs de même nom dans les deux relations? Caractérisez en français les n-uplets de j par rapport à ceux de r et de s .
2. Quel est le schéma et quels sont les n-uplets de $u = r \div s$? Caractérisez en français les n-uplets de u par rapport à ceux de r et de s .
3. Quel est le schéma et quels sont les n-uplets de $r - (s \times u)$?
4. Quel est le schéma et quels sont les n-uplets de $r - (u \times s)$?

TABLE 9 – Une instance r de l'exercice 21

A	B	D
a_1	b_1	d_1
a_1	b_1	d_2
a_1	b_2	d_1
a_2	b_2	d_1
a_3	b_1	d_2
a_2	b_2	d_2
a_2	b_2	d_3
a_1	b_1	d_4

TABLE 10 – Une instance s de l'exercice 21

A	B
a_1	b_1
a_2	b_2

Exercice 22 :

Soit le schéma de base de données ci-dessous contenant deux relations et permettant de gérer des clients et leurs comptes en banque :

Client(ClientID, Nom, Prenom, Adresse)
de clé primaire **ClientID**

Compte(CompteID, ClientID, Solde, TypeCompte)
de clé primaire **CompteID**
et de clé étrangère **ClientID** faisant référence à la clé primaire de la relation **Client**

Les tableaux 11 et 12, ci-dessous, représentent respectivement une instance de chacune de ces relations :

TABLE 11 – Une instance de la relation **Client**

ClientID	Nom	Prenom	Adresse
1	GAMOTTE	Albert	10 rue des Alouettes 75017 Paris
2	DEBECE	Aude	26 av. des blés 91000 Corbeil
3	PABIEN	Yvon	145 allée du Général 92000 Boulogne

1. Donner le résultat des requêtes⁹ suivantes :

- (a) $\Pi_{Nom, Prenom}(Client) - \Pi_{Nom, Prenom}[Compte \bowtie Client]$
- (b) $\Pi_{Nom, Prenom, TypeCompte}[Client \bowtie Compte] \div \Pi_{TypeCompte}(Compte)$

9. Certaines requêtes peuvent retourner une relation sans nuplet. Vous donnerez uniquement dans ce cas le schéma du résultat.

TABLE 12 – Une instance de la relation **Compte**

CompteID	ClientID	Solde	TypeCompte
1	1	1200	PEL
2	1	-600	Compte Courant
3	2	3400	Compte Courant
4	3	3400	Compte Courant

2. Exprimer en algèbre relationnelle et en calcul relationnel à variable nuplet les requêtes suivantes :
 - (a) "Quel est le solde du compte courant du client nommé Yvon PABIEN ?"
 - (b) "Quels sont les noms et les prénoms de clients n'ayant pas de compte en banque de type PEL ?"
 - (c) "Quels sont les types de compte en banque détenus par tous les clients de la banque ?"
3. Exprimer en calcul relationnel à variable nuplet la requête suivante :
 "Quels sont les noms et prénoms de clients dont le solde du compte courant est le plus élevé (i.e. est supérieur aux soldes de tous les autres comptes courants) ?"

Exercice 23 :

Une base de donnée relationnelle comporte les relations suivantes :

Apprécie(*Personne*, *Sujet*), **Etudie**(*Personne*, *Sujet*), **Déteste**(*Personne*, *Sujet*)

dont la sémantique est claire. Les clés primaires sont composées des couples d'attributs (*Personne*, *Sujet*).

Exprimer en algèbre relationnelle, en calcul relationnel à variable nuplet et en SQL, les requêtes suivantes :

1. Quelles personnes étudient les Bases de Données et Java ?
2. Quelles personnes étudient les Bases de Données ou Java ?
3. Quelles personnes apprécient Java mais pas le C++ ?
4. Quelles personnes étudient au moins un sujet qu'elles détestent ?
5. Quelles personnes apprécient au moins deux sujets ?
6. Quelles personnes apprécient uniquement deux sujets ?
7. Quelles personnes étudient le plus grand nombre de sujets ? (*en SQL uniquement*)
8. Quelles personnes détestent tous les sujets qu'apprécie Charles ?
9. Quelles personnes apprécient Java et détestent le C ?
10. Quelles personnes apprécient tous les sujets qu'elles étudient ?
11. Quelles personnes étudient tous les sujets qu'elles apprécient ?

Exercice 24 :

On suppose qu'une bibliothèque gère une base de données dont le schéma est le suivant (les clés primaires des relations sont soulignées) :

Emprunt(*Personne*, *Livre*, *DateEmprunt*, *DateRetourPrevue*, *DateRetourEffective*)
Retard(*Personne*, *Livre*, *DateEmprunt*, *PenalitéRetard*)

Exprimer, lorsque cela est possible, les requêtes suivantes en algèbre relationnelle, en calcul à variables nuplets et en SQL.

1. Quelles sont les personnes ayant emprunté le livre "Recueil Examens BD" ?
2. Quelles sont les personnes n'ayant jamais rendu de livre en retard ?
3. Quelles sont les personnes ayant emprunté tous les livres (empruntés au moins une fois) ?
4. Quels sont les livres ayant été empruntés par tout le monde (i.e. tous les emprunteurs) ?
5. Quelles sont les personnes ayant toujours rendu en retard les livres qu'elles ont empruntés ?

Exercice 25 :

Un organisme de gestion de spectacles, de salles de concert et de vente de billets de spectacles gère une base de données dont le schéma relationnel est le suivant :

Spectacle (*Spectacle_ID*, *Titre*, *DateDéb*, *Durée*, *Salle_ID*, *Chanteur*)

Concert (*Concert_ID*, *Date*, *Heure*, *Spectacle_ID*)

Salle (*Salle_ID*, *Nom*, *Adresse*, *Capacité*)

Billet (*Billet_ID*, *Concert_ID*, *Num_Place*, *Catégorie*, *Prix*)

Vente (*Vente_ID*, *Date_Vente*, *Billet_ID*, *MoyenPaieement*)

Les attributs soulignés sont les attributs appartenant à la clé primaire. Ils sont de type entier. L'attribut *Salle_ID* de la relation *Spectacle* est une clé étrangère qui fait référence à l'attribut de même nom de la relation *Salle*. L'attribut *Spectacle_ID* de la relation *Concert* est une clé étrangère qui fait référence à l'attribut de même nom de la relation *Spectacle*. L'attribut *Concert_ID* de la relation *Billet* est une clé étrangère qui fait référence à l'attribut de même nom de la relation *Concert*. L'attribut *Billet_ID* de la relation *Vente* est une clé étrangère qui fait référence à l'attribut de même nom de la relation *Billet*.

Exprimez, lorsque cela est possible, les requêtes suivantes en algèbre relationnelle, en calcul relationnel à variables nuplets et en SQL.

1. Quelles sont les dates du concert de Corneille au Zenith ?
2. Quels sont les noms des salles ayant la plus grande capacité ?
3. Quels sont les chanteurs n'ayant jamais réalisé de concert à la Cigale ?
4. Quels sont les chanteurs ayant réalisé au moins un concert dans toutes les salles ?
5. Quelles sont les dates et les identificateurs des concerts pour lesquels il ne reste aucun billet invendu ?

Exercice 26 :

Soit la relation de schéma ci-dessous, permettant de gérer des enseignements :

Enseignement (*EnseignementID*, *Titre*, *Semestre*, *Enseignant*)

Deux solutions, parmi les requêtes en algèbre relationnelle ci-dessous permettent de répondre à la question "Quels enseignants enseignent les "BD" ou la "RO". (1) Indiquer quelles sont ces deux réponses à la requête et (2) préciser, la requête qui, selon vous, sera la plus rapide à exécuter (expliquer pourquoi).

1. $[\Pi_{Enseignant} (\sigma_{(Titre='BD')}(Enseignement))] \bowtie [\Pi_{Enseignant} (\sigma_{(Titre='RO')}(Enseignement))]$
2. $[\Pi_{Enseignant} (\sigma_{(Titre='BD')}(Enseignement))] \cup [\Pi_{Enseignant} (\sigma_{(Titre='RO')}(Enseignement))]$
3. $[\Pi_{Enseignant} (\sigma_{(Titre='BD')}(Enseignement))] \cap [\Pi_{Enseignant} (\sigma_{(Titre='RO')}(Enseignement))]$
4. $\Pi_{Enseignant} [(\sigma_{(Titre='BD')}(Enseignement)) \bowtie (\sigma_{(Titre='RO')}(Enseignement))]$
5. $\Pi_{Enseignant} [\sigma_{(Titre='BD') \vee (Titre='RO')}(Enseignement)]$

Exercice 27 :

Soit le schéma de bases de données ci-dessous contenant trois relations et permettant de gérer des clips vidéo, des chaînes de télévision et les passages des clips sur ces chaînes.

$Clip(\underline{ClipID}, Titre, NomDArtiste, DuréeEnMinutes)$ $ChaineTV(\underline{ChaineTVID}, Nom)$

$Diffusion(\underline{DiffusionID}, \#ClipID, \#ChaineTV)$

Les attributs soulignés sont les clés primaires des relations. L'attribut $\#ClipID$ dans la relation *Diffusion* est une clé étrangère qui fait référence à la clé primaire de la relation *Clip*. L'attribut $\#ChaineTV$ dans la relation *Diffusion* est une clé étrangère qui fait référence à la clé primaire de la relation *ChaineTV*.

Exprimer les requêtes suivantes en algèbre relationnelle et en calcul relationnel à variables nuplets.

1. Sur quelles chaînes de télévision passent les clips de "Petite Soeur" de "Laam" ? (*vous donnerez le nom des chaînes de télévision*)
2. Quels sont les titres et les noms des artistes des clips vidéo n'étant diffusés sur aucune chaîne de télévision ?
3. Quels sont les titres des clips vidéo diffusés sur toutes les chaînes ?
4. Quels sont les titres des clips vidéo ayant la plus grande durée ? (*A faire en calcul à variables nuplets uniquement.*)

Exercice 28 :

Soit le schéma de bases de données simplifié suivant :

$R_1(\underline{Parent}, \underline{Enfant})$, $R_2(\underline{Personne}, Age, Sexe)$ et $R_3(\underline{Enfant}, Ecole)$

Aucun attribut ne peut prendre la valeur NULL.

Attention : Un attribut **Age** a été introduit ici pour simplifier les requêtes. Dans une vraie base de données, un tel attribut ne doit pas apparaître. On doit utiliser plutôt un attribut **DateDeNaissance**.

Exprimez les requêtes suivantes en algèbre relationnelle, en calcul relationnel à variables nuplets et en SQL.

1. Quels sont les enfants de Pierre ?
2. Quel âge ont les enfants de Marie ?
3. Combien Paul a-t-il d'enfants de sexe féminin ? (*en SQL uniquement*)
4. Quel est l'âge moyen des personnes répertoriées dans la base ? (*en SQL uniquement*)
5. Qui sont les grands-parents de Jacques ?
6. Quels parents ont au moins un enfant de sexe masculin ?
7. Quels parents n'ont que des enfants de sexe masculin ?
8. Quels couples de parents ont au moins deux enfants ensemble ?
9. Quelle est la liste des parents et des écoles de leurs enfants ?
10. Quelle est la liste des parents qui ont au moins un enfant dans chacune des écoles présentes dans la base ?
11. Quels sont les oncles (personnes de sexe masculin, frères d'un des parents) de Tristan ?
12. Qui n'a pas d'enfant scolarisé ?
13. Quels sont les parents qui ont au moins un fils plus âgé qu'une fille ?
14. Quels sont les arrière-grands-parents de Bérénice ?

Exercice 29 :

Un enseignant utilise une base de données pour gérer ses sujets d'examen et les notes de ses étudiants. Cette base de données est composée de 4 relations : une contenant le planning des examens, une deuxième contenant les sujets des exercices (le numéro de l'exercice étant relatif à l'examen), une troisième contenant la liste des étudiants (identifiés par le numéro de leur carte d'étudiant) et une dernière contenant les notes obtenues par chaque étudiant à chaque exercice. Le schéma de ces relations est le suivant :

Examen(ExamID, Date, Heure, Salle, Formation) de clé primaire ExamID ;

Exercice(NuméroExo, ExamID, Sujet, Solution, Barème) de clé primaire (NuméroExo, ExamID) et de clé étrangère ExamID faisant référence à la clé primaire de la relation Examen ;

Etudiant(NumCarteEtudiant, Nom, Prénom, Formation) de clé primaire (NumCarteEtudiant) ;

RelevéNotes(NuméroExo, ExamID, NumCarteEtudiant, Note) de clé primaire le triplet (NuméroExo, ExamID, NumCarteEtudiant) et de clés étrangères (NuméroExo, ExamenID) faisant référence à la clé primaire de la relation Exercice et NumCarteEtudiant faisant référence à la clé primaire de la relation Etudiant.

Le domaine de l'attribut **Barème** de la relation **Exercice** est l'ensemble des entiers compris entre 0 et 100 (le barème correspondant à un pourcentage). Le domaine de l'attribut **Note** de la relation **RelevéNotes** est l'ensemble des réels compris entre 0 et 20.

1. Exprimer les requêtes suivantes en algèbre relationnelle, en calcul relationnel à variables nuplets et en SQL. Vous pourrez abréger le nom des relations par *Exam*, *Exo*, *E* et *RN*, ainsi que les noms des attributs *ExamID* par *EID*, *NuméroExo* par *NumE*, *NumCarteEtudiant* par *NCE*, *Nom* par *N*, *Prénom* par *P* et *Barème* par *B*.
 - (a) Quel est le barème de l'exercice numéro 1 de l'examen de la formation 'FC MIAGE' du '07/02/06' ?
 - (b) Quels sont les noms et prénoms des étudiants n'ayant pas de note ?
 - (c) Quels sont les exercices réalisés par tous les étudiants ? (i.e. les exercices pour lesquels tous les étudiants ont obtenu une note)
 - (d) Quels sont les noms et prénoms des étudiants ayant eu la moyenne à tous leurs exercices ? *Le domaine de l'attribut Note étant [0,20], la moyenne est égale à 10.*
2. Définir, en SQL uniquement, une vue de schéma **Vue1**(Nom, Prénom, ExamID, NoteFinale), indiquant pour chaque étudiant (en précisant son nom et son prénom) et chaque examen passé par cet étudiant, sa note finale. *La note finale de chaque examen est calculée en faisant la somme de notes de chaque exercice de l'examen, chaque note de chaque exercice étant pondérée par le barème de l'exercice.*
3. En utilisant la vue définie à la question précédente, exprimer, en calcul relationnel à variables nuplets et en SQL, la requête suivante : Quels sont les noms et prénoms des meilleurs étudiants (i.e. ayant obtenu la meilleure note finale) de chaque examen ?

Exercice 30 :

Soit le schéma de bases de données simplifié suivant :

$R_1(P, S)$, $R_2(P, T)$ et $R_3(T, S)$ avec $P = Personne$, $S = Sport$ et $T = Centre_Sportif$.

Exprimez, lorsque cela est possible, les requêtes suivantes en algèbre relationnelle, en calcul relationnel à variables nuplets et en SQL.

1. Quels centres sportifs proposent au moins un sport pratiqué par Pierre ?
2. Quels centres sportifs proposent tous les sports pratiqués par Henri ?
3. Quels centres sportifs proposent tous les sports pratiqués par chacun de leurs membres ?
4. Quels sports offerts par l'association ne sont pas proposés par le Centre *Les Joyeux Musclés* ?

5. Quels centres sportifs proposent au moins deux sports pratiqués par Louis ?
6. Quels centres sportifs proposent les sports pratiqués par Jacques et Jean ?
7. Donnez les couples de personnes tels que chaque personne du couple pratique au moins un sport que l'autre pratique et au moins un sport que l'autre ne pratique pas.

Exercice 31 :

Soit le schéma de la base de données est le suivant :

Salle(Nom, Horaire, Titre)
Film(Titre, Realisateur, Acteur)
Vu(Spectateur, Titre)
Produit(Producteur, Titre)
Aime(Amateur, Titre)

Exprimez, lorsque cela est possible, les requêtes suivantes en algèbre relationnelle, en calcul relationnel à variable nuplet et en SQL.

1. Où et à quelles heure peut-on voir le film “Out of Africa” ?
2. Quels sont les films réalisés par Marcel Carmé ?
3. Quels sont les acteurs de Mission ?
4. Où peut-on voir les films où joue Arletty ?
5. Quels sont les acteurs qui ont produit un film ?
6. Quels sont les acteurs qui produisent un film dans lequel il joue ?
7. Quels sont les acteurs qui jouent dans les films d’Alain Renais ?
8. Quels acteurs jouent dans tous les films ?
9. Quels acteurs jouent dans tous les films de Tati ?
10. Qui produit tous les films de Tarkowski ?
11. Quels spectateurs ont vu tous les films ?
12. Quels spectateurs aiment tous les films qu’ils ont vus ?
13. Où peut-on voir C.Lambert après 16h ?
14. Quels films ne passent dans aucune salle ?
15. Qui produit un film qui ne passe dans aucune salle ?
16. Quels sont les producteurs qui ont vu tous les films qu’ils produisent ?
17. Quels producteurs on vu tous les films de Coline Serreau ?
18. Quel spectateur aime un film qu’il n’a pas vu ?
19. Qui n’aime aucun film ?
20. Qui ne produit aucun Film d’Alain Cavalier ?
21. Quels sont les acteurs qui produisent au moins un film qu’ils ont réalisé ?
22. Quels sont les acteurs qui produisent tous les films qu’ils ont réalisés ?
23. Quels sont les producteurs qui n’ont vu que les films qu’ils produisent ?
24. Quels acteurs ont joué dans au moins deux films ?
25. Quels producteurs ont produit exactement trois films ?

Exercice 32 :

La base de données *J'MleCiné* stocke de l'information destinée aux cinéphiles et comprend les relations suivantes dont la sémantique est évidente :

Festival (*NomF*, *Ville*, *AnnéeF*), *AnnéeF* représente l'année du festival

Film (*Titre*, *Réalisateur*, *Pays*, *AnnéeR*), *AnnéeR* représente l'année de réalisation du film

Casting (*Acteur*, *Titre*, *Rôle*)

Programme (*NomF*, *AnnéeF*, *Titre*)

Palmarès (*NomF*, *AnnéeF*, *Titre*, *Personne*, *Prix*)

On suppose que : *NomF* identifie un festival de cinéma qui se produit au plus une fois par an ; *Titre* identifie un film ; *Personne*, *Réalisateur* et *Acteur* identifient des personnes, voire des êtres vivants ; *Prix* est une chaîne de caractères qui, associée à *Festival* identifie un prix décerné lors d'un Festival de cinéma (oscar, ours, palme ...). Il y a au moins un festival de cinéma par an.

1. Exprimez les requêtes suivantes en algèbre relationnelle, calcul relationnel et SQL. Si l'expression de tout ou partie de la requête ne vous apparaît pas possible dans un langage indiquez pourquoi. Pour faciliter l'expression de certaines requêtes vous pouvez utiliser des vues.

- Quels sont les réalisateurs des films primés au Fespaco de Ouagadougou 2009 ?
- Quels films ont été primés à la Mostra de Venise 2008 ou à la Berlinale de Berlin 2008
- Qui est acteur et réalisateur ?
- Qui réalise un film où il est aussi acteur ?
- Quels réalisateurs n'ont jamais été acteurs ?
- Liste des réalisateurs avec le nombre de films qu'ils ont présentés au Festival de Cannes depuis 2000, limitée à un nombre de films supérieur à 3, et ordonnée par nombre de films puis par nom de réalisateur
- Quels réalisateurs ont réalisé au moins un film chaque année depuis 2005 ?
- Quels acteurs ont été primés au Festival de Cannes 2009 mais pas à la Mostra 2009 ?
- Quels pays présentant au moins un film au Fespaco 2009 n'ont eu aucun prix ?
- Qui joue dans tous les films réalisés par Hitchcock ?

2. Exprimez en français les requêtes suivantes.

- (a) $\{t.Realisateur \mid Film(t) \wedge \neg [\exists u Programme(u) \wedge (u.Titre = t.Titre) \wedge (\exists s Festival(s) \wedge (u.NomF = s.NomF) \wedge (s.Ville = 'Cannes'))]\}$

- (b) $\prod_{Realisateur}(Film) - \prod_{Personne}(Palmares)$

- (c)

```
SELECT  f.Réalisateur
FROM    Film
WHERE   NOT EXISTS ( SELECT  t.Titre
                     FROM    Film t
                     WHERE   f.Realisateur = t. Realisateur
                     AND NOT EXISTS ( SELECT c.Titre
                                     FROM Casting
                                     WHERE c.Titre = t.Titre
                                     AND c.Acteur = f.Realisateur
                                   )
                   )
```

Exercice 33 :

Soit la base de données dont le schéma relationnel est le suivant :

```
SérieTV(SérieID, Titre, Saison, Descriptif, NbEpisodes)
ChaîneTV(Nom, NumeroCanal, Président)
Programme(ProgID, Chaîne, Série, Heure, Période)
Production(ProdID, Chaîne, Série)
```

Les clés primaires **SérieID**, **ProgID** et **ProdID** des relations **SérieTV**, **Programme** et **Production** sont de type entier. La clé primaire **Nom** de la relation **ChaîneTV** est de type chaîne de caractères. L'attribut **Série** de la relation **Programme** est une clé étrangère qui fait référence à l'attribut **SérieID** de la relation **SérieTV**. L'attribut **Chaîne** de la relation **Programme** est une clé étrangère qui fait référence à l'attribut **Nom** de la relation **ChaîneTV**. L'attribut **Série** de la relation **Production** est une clé étrangère qui fait référence à l'attribut **SérieID** de la relation **SérieTV**. L'attribut **Chaîne** de la relation **Production** est une clé étrangère qui fait référence à l'attribut **Nom** de la relation **ChaîneTV**.

Exprimer les requêtes suivantes en algèbre relationnelle et en SQL.

1. Quelles sont les noms des chaînes de télévision qui programment la série *Friends*?
2. Quelles sont les noms des chaînes de télévision qui programment la série *Kamelott* mais pas la série *PJ*?
3. Quelles sont les titres des séries qui ne passent sur aucune chaîne de télévision (ne sont pas programmées) ?
4. Quelles sont les chaînes de télévision qui ont programmé toutes les saisons¹⁰ de la série *Friends*?
5. Quelles sont les chaînes de télévision qui ne programment que les séries qu'elles produisent ?
6. Quelles sont les chaînes de télévision qui ne programment aucune des séries qu'elles produisent ?

Exercice 34 (à faire en TP) :

Soit le schéma de base de données suivant :

```
Succursale(NomSucc, Actif, VilleSucc)
Client(Nom_Client, Rue, Ville_Client)
Compte(Nom_Succ, Numero_Compte, Nom_Client, Solde)
Emprunt(Nom_Succ, Numero_Emprunt, Nom_Client, Montant)
```

Exprimer les requêtes suivantes en SQL.

1. Trouver les noms des succursales ayant des comptes client, avec ou sans élimination des doublons.
2. Trouver les noms des clients ayant un compte à la succursale de nom *Rivoli*.
3. Trouver les noms des clients ayant un compte à la succursale de nom *Rivoli* ou un emprunt à la succursale de nom *Opéra*.
4. Trouver les noms des clients ayant un compte à la succursale de nom *Rivoli* et un emprunt à la succursale de nom *Opéra*. Proposer trois solutions
5. Trouver les noms des clients ayant un compte à la succursale de *Rivoli* mais pas d'emprunts. Proposer trois solutions.
6. Trouver les noms des clients ayant un compte dans la ville où ils habitent. Créer une vue.
7. Trouver les noms des clients ayant un compte à *Etoile* avec la ville où ils habitent. Créer une vue.
8. Trouver les noms des clients ayant un compte dans la succursale où Pierre a un compte. Donnez au moins deux solutions dont une avec des variables nuplets.

10. S'il y a x saisons dans une série, alors la relation **SerieTV** contient x nuplets ayant la même valeur pour **Titre** et ayant pour valeur de l'attribut **Saison** un entier variant de 1 à x .

9. Trouver les succursales qui ont un actif plus élevé que celui d'une succursale d'Aurillac. Deux solutions.
10. Trouver les succursales qui ont un actif plus élevé que celui de toutes les succursales d'Aurillac. Trois solutions.
11. Trouver les noms des clients ayant un compte dans toutes les succursales de Conflans-Saint-Honorine.
12. Donner la liste par ordre alphabétique des emprunteurs de la succursale d'Orsay.
13. Donner pour chaque succursale le solde moyen des comptes clients.
14. Donner pour le solde moyen des comptes clients pour les succursales ayant un solde moyen supérieur à 5000.
15. Combien de clients habitent Paris ?
16. Combien de clients ayant un compte à la succursale de nom *Bastille* n'ont pas leur adresse dans la relation Client ?
17. Insérer le nuplet (Paul, Victor Hugo, Paris) dans la relation Client.
18. Diminuer l'emprunt de tous les clients habitant Ajaccio de 5%.
19. Créditer le compte de Paul de 1000 euros.
20. Fermer le compte de Thomas.
21. Supprimer de Succursale tous les succursales sans client.
22. Créer la relation VIP(Nom_Client, Numero_Compte).
23. Exprimer la contrainte en SQL "Un client ne peut emprunter que s'il n'est pas débiteur sur son solde courant".

Exercice 35 :

Soit la base de données de schéma ci-dessous, permettant de gérer les serveurs et logiciels des départements d'une entreprise

Serveur(NomServeur, TypeServeur)
Département(DeptID, NomDept, Budget, #NomServeur)
Logiciel(NomLogiciel, Editeur)
Installation(#NomLogiciel, #DeptID, DateInstall)

Les attributs soulignés sont les clés primaires des relations. Tous les attributs sont de type chaînes de caractères excepté les attributs *DeptID* et *DateInstall* qui sont respectivement de type entier et date. L'attribut #NomServeur, dans la relation *Département*, est une clé étrangère qui fait référence à l'attribut de même nom de la relation *Serveur*. Les attributs #NomLogiciel et #DeptID, dans la relation *Installation*, sont des clés étrangères qui font respectivement référence aux attributs de même nom des relations *Logiciel* et *Département*.

Exprimez, les requêtes suivantes en algèbre relationnelle, en calcul relationnel à variable nuplet et en SQL.

1. Quel est le nom du serveur du département 'R&D' ?
2. Quels logiciels sont installés à la fois dans le département 'R&D' et dans le département 'RH' ?
3. Quels logiciels sont installés à la fois dans le département 'R&D' ou dans le département 'RH' ?
4. Quels éditeurs produisent au moins deux logiciels différents ?
5. Quels sont les noms de départements ayant le plus gros budget ?
6. Quels logiciels ne sont pas installés ?
7. Quels logiciels sont installés dans tous les départements ?

8. Quels sont les noms des départements ayant installé tous les logiciels ?
9. A quelle date y a t'il eu le plus d'installations ? (*vous pouvez utiliser une vue*)

Exercice 36 :

Soit la base de données de schéma ci-dessous, permettant de gérer des documents et les mots-clé qu'ils contiennent :

Document(DocID, Titre, NbPages, DateCréation)

Mot-Clé(#DocID, Mot, NbOccurrences)

Les attributs soulignés sont les clés primaires des relations. L'attribut #DocID dans la relation Mot-Clé est une clé étrangère qui fait référence à la clé primaire de la relation Document. Les attributs DocID, NbPages et NbOccurrences sont de type entier. L'attribut NbOccurrences correspond au nombre de fois où un même mot apparaît dans le document associé. Par exemple, le mot-clé "SQL" apparaît 864 fois dans le document intitulé "Cours de Bases de Données". L'attribut DateCréation est de type date. L'attribut Titre est unique (i.e. il ne peut pas y avoir deux documents différents de même titre).

Le tableau 36 donne quelques exemples de nuplets de la relation *Document* et le tableau 36 donne quelques exemples de nuplets de la relation *Mot-Clé*.

Pour simplifier, dans les requêtes, vous pouvez abréger le nom des relations et des attributs par leur initiales (par exemple *D* pour *Document*, *MC* pour *Mot-Clé*, *DID* pour *DocID* etc.).

DocID	Titre	NbPages	DateCréation
1	Cours de Bases de Données	100	01/09/2007
2	TD de Bases de Données	87	07/09/2007

TABLE 13 – Partie de l'instance de la relation *Document* de l'exercice 36.

DocID	Mot	NbOccurrences
1	SQL	865
1	Relation	1200
2	SQL	456

TABLE 14 – Partie de l'instance de la relation *Mot-Clé* de l'exercice 36.

1. Exprimez, les requêtes suivantes en algèbre relationnelle et en calcul relationnel à variable nuplet.
 - (a) Quels sont les mots-clés du document numéro 1 ?
 - (b) Quels sont les titres des documents où apparaît le mot-clé "SQL" OU le mot-clé "Relation" ?
 - (c) Quels sont les titres des documents où apparaît le mot-clé "SQL" ET le mot-clé "Relation" ?
 - (d) Quels sont les titres des documents où apparaît le mot-clé "SQL" MAIS PAS le mot-clé "Relation" ?
 - (e) Quels sont les mots-clés apparaissant dans tous les documents ?
2. Exprimez, la requête suivante en calcul relationnel à variable nuplet uniquement : "Quels sont les titres des documents ayant le plus grand nombre de pages ?"

Exercice 37 :

Soit la relation **Auteur**(#DocID, Nom, Prénom) ajoutée à la base de données de l'exercice précédent. Cette relation, dont la clé primaire est composée de 3 attributs, permet d'indiquer le nom et le prénom des différents auteurs d'un document, un document pouvant être écrit par plusieurs auteurs et un même auteur pouvant écrire plusieurs documents.

La requête en algèbre relationnelle suivante permet-elle de calculer le résultat de la requête "Quels documents ont pour auteurs 'Albert Gamotte' ET 'Yvon Pabien' ?" ?

$$\Pi_{Titre} \left[\sigma_{(Nom='Gamotte') \wedge (Prenom='Albert') \wedge (Nom='Pabien') \wedge (Prenom='Yvon')} (Auteur \bowtie_{DocID=\#DocID} Document) \right]$$

Si la réponse à la question est positive, expliquer pourquoi et donner un exemple de résultat en utilisant les nuplets de la relation **Document** ci-dessus et en donnant quelques exemples de nuplets de la relation **Auteur**. Si la réponse à la question est négative, expliquer pourquoi et proposer une correction de la requête.

Exercice 38 :

Soit la base de données de schéma ci-dessous, permettant de gérer des inscriptions en crèches :

Creches (CrecheID, Nom, Adresse, Telephone, NombreDePlacesAuTotal)
Enfant (EnfantID, Nom, Prénom, TelephoneMere, TelephonePere, Adresse, DateDeNaissance)
DemandeInscription (DemandeID, #EnfantID, #CrecheID, DateDemande)
Inscription (InscriptionID, #EnfantID, #CrecheID, DateAccord)

Les attributs soulignés sont les clés primaires des relations. L'attribut #EnfantID dans les relations **DemandeInscription** et **Inscription** est une clé étrangère qui fait référence à la clé primaire de la relation **Enfant**. L'attribut #CrecheID dans les relations **DemandeInscription** et **Inscription** est une clé étrangère qui fait référence à la clé primaire de la relation **Creche**.

Pour simplifier, dans les requêtes, vous pouvez abréger le nom des relations et des attributs par leurs initiales (par exemple C pour Creche, DI pour DemandeInscription, CID pour CrecheID etc.). Vous pouvez également omettre les #.

- Exprimez, les requêtes suivantes en algèbre relationnelle, en calcul relationnel à variable nuplet et en SQL.
 - Quelle est la date de naissance de l'enfant ALBERT GAMOTTE ?
 - Dans quelles crèches (en donnant le nom des crèches) n'y-a-t-il eu aucune demande d'inscription ?
 - Quels sont les noms et prénoms des enfants pour lesquels il y a une demande d'inscription dans toutes les crèches de la base ?
- Exprimez, la requête suivante en calcul relationnel à variable nuplet et en SQL : "Quel est le nom de la plus grande crèche (en nombre de places) ?"
- Exprimez, la requête suivante en SQL uniquement :
 - Créer une vue permettant de connaître le nombre d'enfants inscrits dans chaque crèche (en donnant le nom des crèches).
 - Quel est le nombre de place restante dans chaque crèche ? (NB : Le nombre de places restantes se calcule en faisant une soustraction entre le nombre de places au total et le nombre d'inscrits)
 - Vous pouvez réutiliser la vue créée précédemment. On supposera que le nom des crèches est unique (il n'existe pas deux crèches avec le même nom).

Exercice 39 :

La base de données *OLYMPIX* contient les résultats sportifs de l'ensemble des Jeux Olympiques "modernes" dans les relations suivantes dont la sémantique est évidente :

Résultats (*Id-Sportif*, *Discipline*, *Classement*, *Id-Jeux*)

Jeux (*Id-Jeux*, *Ville*, *Pays*, *Année*)

Sportif (*Id-Sportif*, *Nom*, *Prénom*, *Pays*)

Comme certaines villes ont accueilli plusieurs fois les Jeux Olympiques modernes (recréés en 1896), une instance des Jeux Olympiques est identifiée par le nom de la ville et l'année où ils se déroulent.

Exprimez les requêtes suivantes en algèbre relationnelle (éventuellement étendue), calcul relationnel et SQL. Si l'expression de tout ou partie de la requête ne vous apparaît pas possible dans un langage indiquez pourquoi. Pour faciliter l'expression de certaines requêtes vous pouvez utiliser des vues.

1. Nom et prénom des sportifs grecs ayant eu au moins 2 médailles d'or ou 1 médaille d'or et une d'argent à Athènes 2004 ?
2. Nom et prénom des sportifs ayant été premiers à la "course 100m" et au "relais 4×100m" dans les mêmes jeux.
3. Pour les Jeux d'Athènes 2004, donnez la liste des pays et leur nombre de médailles d'or, classée par ordre décroissant des nombres de médailles d'or et en cas d'égalité par ordre alphabétique de pays. Les pays n'ayant pas eu de médaille d'or figureront en fin de liste.
4. Nom et prénom des sportifs ayant eu au moins 3 médailles (or, argent ou bronze) à Sydney 2000, avec leur nombre de médailles de chaque catégorie.
5. Nom et prénom des sportifs ayant eu des médailles d'or à Athènes 2004 et à Pékin 2008.
6. Quelles disciplines ont été pratiquées à tous les Jeux Olympiques modernes.
7. Nom et prénom des sportifs qui étaient à Pékin 2008 mais pas à Athènes 2004. 8
8. Quelles villes ont accueilli plusieurs fois les Jeux Olympiques depuis 1896 ?

1.4 Dépendances fonctionnelles, décomposition de schéma et formes normales

Les exercices 45, 50, 57 et 41 sont issus d'anciens sujets d'examen. Merci à A. Doucet, S. Gançarski, H. Garcia Molina, J. Ullman, P. Valduriez, J. Widom, R. Ramakrishnan et J. Gehrke qui retrouveront certaines de leurs propositions adaptées dans les autres exercices de cette section.

Exercice 40 :

Soit une relation R de schéma $R(A, B, C, D, E)$ et une instance r de cette relation ci-dessous.

Quelles dépendances fonctionnelles vérifie l'instance r ?

Attention : Constater les dépendances fonctionnelles que vérifie r ne prouve pas que ces dépendances fonctionnelles existent sur R (et donc sont vérifiées par toutes les instances possibles de R).

TABLE 15 – Une instance r de la relation $R(A, B, C, D, E)$.

A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_1	b_2	c_2	d_2	e_1
a_2	b_1	c_3	d_3	e_1
a_2	b_1	c_4	d_3	e_1
a_3	b_2	c_5	d_1	e_1

Exercice 41 :

Cet exercice est construit à partir du schéma de la base de données *OLYMPIX* de l'exercice n°39 du chapitre 1.3.

1. Qu'est ce qu'une dépendance fonctionnelle (définition) ?
2. Qu'est ce qu'une dépendance fonctionnelle triviale (= évidente) ?
3. Exprimez en les justifiant les dépendances fonctionnelles non triviales qui doivent être vérifiées pour chaque relation du schéma de la base de données *OLYMPIX*
4. Déduisez en les clés minimales des relations

Exercice 42 :

Soit $F = \{AB \rightarrow C, B \rightarrow D, CD \rightarrow E, CE \rightarrow GH, G \rightarrow A\}$

A-t-on $AB \rightarrow E$? $BG \rightarrow C$? $AB \rightarrow G$?

Exercice 43 :

Soit un univers U et X, Y, Z et W des sous-ensembles d'attributs de U . A t'on les implications logiques suivantes ? Si oui, le montrer, si non donner un contre-exemple.

1. $\{X \rightarrow Y, Z \rightarrow W\} \Rightarrow XZ \rightarrow YW$
2. $\{XY \rightarrow Z, Z \rightarrow X\} \not\Rightarrow Z \rightarrow Y$
3. $\{X \rightarrow Y, Y \rightarrow Z\} \Rightarrow X \rightarrow YZ$
4. $\{X \rightarrow Y, W \rightarrow Z\}$ et $Y \subseteq W \not\Rightarrow X \rightarrow Z$
5. $\{W \rightarrow Y, X \rightarrow Z\} \Rightarrow WX \rightarrow Y$
6. $\{X \rightarrow Y\}$ et $Z \subseteq Y \Rightarrow X \rightarrow Z$
7. $\{X \rightarrow Y, X \rightarrow W, WY \rightarrow Z\} \Rightarrow X \rightarrow Z$
8. $\{XY \rightarrow Z, Y \rightarrow W\} \not\Rightarrow XW \rightarrow Z$
9. $\{X \rightarrow Y, XY \rightarrow Z\} \Rightarrow X \rightarrow Z$

Exercice 44 Chacune des assertions suivantes est-elle vraie ? Si oui, le démontrer. Si non, proposer un contre-exemple.

1. Si $A \rightarrow B$ Alors $B \rightarrow A$?
2. Si $A \rightarrow B$ et $AB \rightarrow C$ Alors $A \rightarrow C$?
3. Si $AB \rightarrow C$ Alors $A \rightarrow C$ et $B \rightarrow C$?

Exercice 45 :

Soit un schéma de bases de données contenant les relation suivantes :

Bureau(*NumBureau*, *NumTelephone*, *Taille*) avec
 $F_{Bureau} = \{ NumBureau \rightarrow NumTelephone, Taille; NumTelephone \rightarrow NumBureau; \}$
Occupant(*NumBureau*, *PersonneID*) avec $F_{Occupant} = \{ NumBureau \rightarrow PersonneID \}$
Materiel(*NumBureau*, *NumPC*) avec $F_{Materiel} = \{ NumPC \rightarrow NumBureau \}$

1. Les contraintes ci-dessous sont-elles vérifiées par ce schéma de bases de données ? Si la réponse est positive, expliquer pourquoi. Si la réponse est négative, indiquez quelle(s) dépendance(s) fonctionnelle(s) il faut ajouter/supprimer ou modifier pour que la contrainte soit vérifiée.
 - (a) "Un bureau peut contenir plusieurs postes téléphoniques."
 - (b) "Il y a une et une seule personne par bureau."
 - (c) "Un bureau contient un seul ordinateur."
2. A partir des familles de dépendances fonctionnelles initiales données dans l'énoncé, indiquez quelles sont les clés minimales possibles de chaque relation.

Exercice 46 :

Soit R une relation dont le schéma est le suivant :

$R(UtilisateurID, Nom, Prénom, AdresseEmail, Login, Passwd, ServeurMail).$

1. Exprimer, à l'aide de dépendances fonctionnelles, les contraintes suivantes que doivent vérifier les instances de la relation R :
 - (a) "On peut déduire le nom et le prénom d'un utilisateur à partir de son identificateur."
 - (b) "Un utilisateur (identifié par son identificateur) possède un seul login et un seul password par serveur de mails."
 - (c) "Une adresse email est associée à un et un seul identificateur d'utilisateur."
 Attention : un utilisateur peut avoir plusieurs adresses de mails.
 - (d) "Une adresse email est associée à un et un seul serveur de mails."
2. Indiquer, à partir de la famille de dépendances fonctionnelles, issue de la question 1, quelles sont les clés minimales de R .
3. Indiquer, à partir de la famille de dépendances fonctionnelles, issue de la question 1, en quelle forme normale est la relation R .

Exercice 47 :

Soient r et s deux instances de schéma R ayant comme clé K . Parmi les relations suivantes, lesquelles ont comme clé K ? Montrez-le ou trouvez des contre-exemples.

1. $r \cup s$
2. $r \cap s$
3. $r - s$
4. $\Pi_K(r)$

Exercice 48 :

Soient $R(A, B, C, D)$ et $F = \{A \rightarrow B, B \rightarrow C\}$

1. Quelles sont les clés minimales de R ?
2. Quelle est la fermeture de A ?

Exercice 49 :

Soient $F = \{A \rightarrow B, CE \rightarrow H, C \rightarrow E, A \rightarrow CH\}$ et $G = \{C \rightarrow EH, A \rightarrow BC\}$

1. Les deux familles sont-elles équivalentes?
2. F est-elle minimale?
3. Peut-on déduire les dépendances fonctionnelles $CE \rightarrow B$ et $AB \rightarrow C$?

Exercice 50 :

On a les relations suivantes :

$Compte(NumCompte, NumClient, Solde)$ avec
 $F_{Compte} = \{ NumCompte \rightarrow Solde; NumClient \rightarrow NumCompte; \\ NumCompte \rightarrow NumClient \}$

$Localisation(NumCompte, NumAgence)$ avec
 $F_{Localisation} = \{ NumCompte \rightarrow NumAgence \}$

$Agence(NumAgence, Nom, Adresse, Directeur)$ avec
 $F_{Agence} = \{ NumAgence \rightarrow Nom, Adresse, Directeur; Directeur \rightarrow NumAgence; \\ Nom, Adresse \rightarrow NumAgence; \}$

1. Les contraintes ci-dessous sont-elles vérifiées?
 - (a) "Un client (représenté par $NumClient$) peut avoir plusieurs comptes en banques."
 - (b) "Un directeur ne peut pas diriger plusieurs agences bancaires."
 - (c) "Un même numéro de compte peut apparaître dans différentes agences."
2. Quelles sont les clés minimales des relations?

Exercice 51 :

Soit la relation $R(A, B, C, D, E)$. Pour chaque instance de R , précisez dans quels cas (pour quelles valeurs de a) la dépendance fonctionnelle $BC \rightarrow D$ est violée et dans quels cas, elle ne l'est pas.

1. $\{ \}$ (instance vide)
2. $\{(a, 2, 3, 4, 5); (2, a, 3, 5, 5)\}$
3. $\{(a, 2, 3, 4, 5); (2, a, 3, 4, 5); (a, 2, 3, 6, 5)\}$
4. $\{(a, 2, 3, 4, 5); (2, a, 3, 7, 5); (a, 2, 3, 4, 6)\}$

Exercice 52 :

Soit la relation $R(A, B, C, D)$. Pour chaque famille de dépendances fonctionnelles proposées :

1. Identifier les clés minimales de R .
2. Indiquer si la décomposition est sans perte d'information et/ou sans perte de dépendances.

Les familles de dépendances fonctionnelles et les propositions de décompositions sont les suivantes :

1. $F = \{B \rightarrow C, D \rightarrow A\}$,
la relation R est décomposée en $R_1(B, C)$ et $R_2(A, D)$.
2. $F = \{AB \rightarrow C, C \rightarrow A, C \rightarrow D\}$,
la relation R est décomposée en $R_1(A, C, D)$ et $R_2(B, C)$.
3. $F = \{A \rightarrow BC, C \rightarrow AD\}$,
la relation R est décomposée en $R_1(A, B, C)$ et $R_2(A, D)$.
4. $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow D\}$,
la relation R est décomposée en $R_1(A, B)$ et $R_2(A, C, D)$.
5. $F = \{A \rightarrow B, B \rightarrow C, C \rightarrow D\}$,
la relation R est décomposée en $R_1(A, B)$, $R_2(A, D)$ et $R_3(C, D)$.

Exercice 53 :

1. Trouver un exemple de décomposition SPI et \neg SPD
2. Trouver un exemple de décomposition \neg SPI et SPD

Exercice 54 :

Pour chaque relation, (1) déterminer les clés minimales, (2) s'il y a de la redondance et (3) si la décomposition proposée est SPI et SPD.

1. $R(\text{Cours}, \text{Etudiant}, \text{DateNaissance}, \text{Note})$ munie de la famille de dépendances fonctionnelles $F = \{C, E \rightarrow N; E \rightarrow DN\}$.
La relation est décomposée en $R_1(C, E, N)$ et $R_2(E, DN)$
2. $R(\text{Etudiant}, \text{Examen}, \text{Date})$ munie de la famille de dépendances fonctionnelles $F = \{D, Et \rightarrow Ex; Ex \rightarrow D\}$.
La relation est décomposée en $R_1(Et, Ex)$ et $R_2(Ex, D)$
3. $R(\text{NomResident}, \text{Adresse}, \text{NumAppart}, \text{Tel})$ munie de la famille de dépendances fonctionnelles $F = \{A, NumA \rightarrow NomR; NomR, A \rightarrow T\}$.
La relation est décomposée en $R_1(NomR, A, NumA)$ et $R_2(NomR, A, Tel)$

Exercice 55 :

Soit $R(A, B, G, D, E)$ munie de la famille de dépendances fonctionnelles $F = \{DE \rightarrow A; B \rightarrow D; AG \rightarrow E; AGE \rightarrow D; BD \rightarrow A; ABD \rightarrow G\}$

1. La décomposition de R en $R_1(A, G, D, E)$, $R_2(D, E)$, $R_3(B, D)$ et $R_4(A, G, E)$ est-elle SPI ?
2. La dépendance $BD \rightarrow E$ fait-elle partie de F^+ ?
3. Calculer la fermeture de B et de AG .
4. Trouver une famille équivalente à F et minimale.

Exercice 56 :

Soient un schéma de relation $R(A, B, C, D, E, G)$ et une famille de dépendances fonctionnelles $F = \{AB \rightarrow C; AC \rightarrow B; AD \rightarrow E; B \rightarrow D; BC \rightarrow A; E \rightarrow G\}$.

Les décompositions suivantes de R sont-elles (a) sans perte d'information et (b) sans perte de dépendances ?

Expliquer.

1. $R_1(A, B)$, $R_2(B, C)$, $R_3(A, B, D, E)$ et $R_4(E, G)$
2. $R_1(A, B, C)$, $R_2(A, C, D, E)$ et $R_3(A, D, G)$
3. $R_1(A, B, C)$, $R_2(A, D, E)$, $R_3(B, D)$ et $R_4(E, G)$

Exercice 57 :

Soit une relation $R(\text{NomFichier}, \text{Taille}, \text{Repertoire}, \text{DateCr  at}, \text{HeureCr  at}, \text{Login}, \text{AutAcc  s}, \text{DateAcc  s}, \text{HeureAcc  s}, \text{TypeAcc  s})$, dont une partie d'une de ses instances est donn  e par le tableau 16.

Pour simplifier, on abr  gera les attributs par NF , T , R , DC , HC , L , A , DA , HA et TA .

NomFichier	Taille	Repertoire	DateCr��at	HeureCr��at	Login	AutAcc��s	DateAcc��s	HeureAcc��s	TypeAcc��s
SujetExamFC.pdf	49002	/home/manouvri	23/01/06	14 :44	manouvri	Lecture/Ecriture	01/02/06	10 :06	Ecriture
CoursBD.pdf	647369	/home/manouvri/www	23/01/01	15 :34	manouvri	Lecture/Ecriture	19/12/05	18 :57	Ecriture
CoursBD.pdf	647369	/home/manouvri/www	23/01/01	15 :34	gamotte	Lecture Seule	01/02/06	11 :23	Lecture

TABLE 16 – Instance de la relation de l'exercice 57.

Soit F , une famille minimale de d  pendances fonctionnelles associ  e    la relation R :

$$F = \{ \begin{array}{l} NF, R \longrightarrow T ; \\ NF, R \longrightarrow DC ; \\ NF, R \longrightarrow HC ; \\ L, NF, R \longrightarrow A ; \\ L, NF, R, DA, HA \longrightarrow TA \end{array} \}$$

1. A-t-on les contraintes suivantes ? *Expliquer*.
 - (a) Un nom de fichier est unique. Il ne peut pas se r  p  ter dans plusieurs r  p  toires diff  rents.
 - (b) Un m  me utilisateur (identifi   par son login) peut acc  der plusieurs fois au m  me fichier    la m  me date, mais pas    la m  me heure.
 - (c) Un utilisateur (identifi   par son login) a une autorisation d'acc  s (de type Lecture/Ecriture ou Lecture Seule) par fichier.
2. D  terminer les cl  s minimales de R . *Expliquer*.
3. En quelle forme normale est la relation R ? *Expliquer*.
4. La relation R contient-elle de la redondance d'information ? *Expliquer*.
5. Proposer une d  composition de R qui soit sans perte d'information et sans perte de d  pendances, contenant le moins de relations possible et dont le sch  ma des relations est BCNF. *Expliquer et montrer que votre d  composition est SPI et SPD*.

Exercice 58 :

1. Donner une famille de d  pendances fonctionnelles pour la relation $R(A, B, C, D)$ de cl   AB telle que R soit $1NF$ mais pas $2NF$
2. Donner une famille de d  pendances fonctionnelles pour la relation $R(A, B, C, D)$ de cl   AB telle que R soit $2NF$ mais pas $3NF$

Exercice 59 :

Pour chaque relation ci-dessous, (a) donnez la liste des cl  s de la relation, (b) pr  cisez en quelle forme normale est la relation (vous donnerez des explications claires et concises de 3 lignes maximum) et (c) proposez une d  composition BCNF.

1. $R_1(A, B, C, D, E)$ avec $F_1 = \{A \longrightarrow B; C \longrightarrow D\}$
2. $R_2(A, B, C)$ avec $F_2 = \{B \longrightarrow C\}$
3. $R_3(A, B)$ avec $F_3 = \emptyset$

Exercice 60 Soit $R(A, B, C, D)$, pour chaque famille de dépendances fonctionnelles, déterminer en quelle forme normale est la relation R :

1. $F = \{A \longrightarrow D\}$
2. $F = \{A \longrightarrow D; D \longrightarrow A\}$

Exercice 61 :

Soit une relation R composée de quatre attributs A, B, C et D . Pour chaque famille de dépendances fonctionnelles : (a) identifier les clés potentielles de la relation R , (b) identifier en quelle forme normale est R , (c) si R n'est pas BCNF, proposez une décomposition qui préserve les dépendances.

1. $C \longrightarrow D, C \longrightarrow A, B \longrightarrow C$
2. $B \longrightarrow C, D \longrightarrow A$
3. $ABC \longrightarrow D, D \longrightarrow A$
4. $A \longrightarrow B, BC \longrightarrow D, A \longrightarrow C$
5. $AB \longrightarrow C, AB \longrightarrow D, C \longrightarrow A, D \longrightarrow B$

Exercice 62 :

Pour chaque relation ci-dessous, indiquez : (1) si la relation est redondante, (2) quelles sont les clés minimales et (3) si le schéma de la relation n'est pas BCNF, appliquer l'algorithme de décomposition BCNF, (4) la décomposition est-elle SPD ?

1. $R(\text{Heure}, \text{Cours}, \text{Professeur}, \text{Note}, \text{Salle}, \text{Etudiant})$ une relation et F une famille de dépendances fonctionnelles $\{\text{Cours} \longrightarrow \text{Professeur}; \text{Cours}, \text{Etudiant} \longrightarrow \text{Note}; \text{Heure}, \text{Salle} \longrightarrow \text{Cours}; \text{Etudiant}, \text{Heure} \longrightarrow \text{Salle}, \text{Cours}\}$
2. $R(\text{Fournisseur}, \text{Adresse}, \text{RaisonSociale}, \text{NumProduit}, \text{Libelle}, \text{Quantite}, \text{Prix}, \text{NumCommande}, \text{Delai}, \text{Date})$ et F une famille de dépendances fonctionnelles $\{\text{NumCommande} \longrightarrow \text{Fournisseur}, \text{Delai}, \text{Date}; \text{NumProduit}, \text{Fournisseur} \longrightarrow \text{Prix}; \text{Fournisseur} \longrightarrow \text{RaisonSociale}, \text{Adresse}; \text{NumCommande}, \text{NumProduit} \longrightarrow \text{Quantite}; \text{NumProduit} \longrightarrow \text{Libelle}\}$
3. $R(\text{NumVoiture}, \text{Marque}, \text{Modele}, \text{Puissance}, \text{Couleur})$ et F une famille de dépendances fonctionnelles $\{\text{NumVoiture} \longrightarrow \text{Modele}, \text{Couleur}; \text{Modele} \longrightarrow \text{Marque}, \text{Puissance}\}$

Exercice 63 :

Soit $R(A, B, C, D, E)$ une relation et F une famille de dépendances fonctionnelles $\{DE \longrightarrow C; C \longrightarrow B; B \longrightarrow E\}$

1. La famille F est-elle minimale ?
2. Quelles sont les clés minimales de R ?
3. Quelles redondances sont susceptibles d'apparaître dans R ?
4. En quelle forme normale est R ?
5. Si R est décomposée en $R_1(C, D, E)$ et $R_2(A, B, D, E)$. La décomposition proposée est-elle SPI et/ou SPD ?
6. Proposer une décomposition de R qui soit SPI et SPD.

2 Annale de sujets d'examens

2.1 Examen de Janvier 2022

2.1.1 Exercice 1 - Algèbre relationnelle et SQL (13 points soit 43%)

Soit le schéma d'une base de données, décrit en SQL ci-dessous, permettant de gérer des séances d'entraînement d'un coach sportif.

```
CREATE TABLE planning
( PID serial PRIMARY KEY,
  lieu varchar(25) NOT NULL,
  jour varchar(8) NOT NULL,
  heure varchar(10) NOT NULL,
  CONSTRAINT UN_planning UNIQUE (lieu,jour)
);

CREATE TABLE seance
( SID serial PRIMARY KEY,
  PID integer NOT NULL,
  dateSeance date NOT NULL,
  nbInscrits integer DEFAULT 0,
  CONSTRAINT UNE_seance UNIQUE (PID,dateSeance),
  CONSTRAINT FK_seance FOREIGN KEY (PID) REFERENCES planning(PID)
);

CREATE TABLE athlete
( AID serial PRIMARY KEY,
  nom varchar(25) NOT NULL,
  prenom varchar(25) NOT NULL,
  dateNaissance date NOT NULL,
  CONSTRAINT UNE_Personne UNIQUE (nom,prenom,dateNaissance)
);

CREATE TABLE inscription
( IID serial PRIMARY KEY,
  SID integer NOT NULL,
  AID integer NOT NULL,
  CONSTRAINT UNE_inscription UNIQUE (AID,SID),
  CONSTRAINT FK1_inscription FOREIGN KEY (AID) REFERENCES personne(AID),
  CONSTRAINT FK2_inscription FOREIGN KEY (SID) REFERENCES seance(SID)
);
```

Le coach organise des séances d'entraînement selon un planning stocké dans la relation **planning**, dont une instance est donnée ci-dessous :

PID	lieu	jour	heure
1	Boulogne	jeudi	19h
2	Boulogne	dimanche	10h30
3	Vincennes	dimanche	14h
4	Vincennes	lundi	13h

Chaque semaine, le coach indique les séances qui suivent ce planning. Par exemple, dans l'instance de la relation **seance** ci-dessous, il a inséré les séances des 2 premières semaines de janvier 2022 :

SID	PID	dateseance	nbinscrits
1	1	06/01/2022	1
2	1	13/01/2022	3
3	2	09/01/2022	1
4	2	16/01/2022	1
5	3	09/01/2022	0
6	3	16/01/2022	0
7	4	03/01/2022	0
8	4	10/01/2022	0

Les athlètes, dont il est le coach, sont répertoriées dans la relation **athlete**, dont une instance est donnée ci-dessous :

AID	nom	prenom	datenaissance
1	Noel	Pierre	25/12/2000
2	Zarella	Maude	12/05/1977
3	Gamotte	Albert	15/02/2011
4	Pabien	Yvon	14/12/1973

Chaque athlète peut s'inscrire, en fonction de ses disponibilités, aux différentes séances. Une instance de la relation **inscription**, ci-dessous, donne les informations sur les inscriptions de la 1ère semaine de janvier 2022.

IID	SID	AID
1	1	1
2	2	1
3	3	1
4	4	1
5	2	2
6	2	3

Dans la suite, vous pourrez abréger le nom des relations par *P*, *S*, *A* et *I*, ainsi que les noms des attributs par leurs initiales (ex. *lieu* par *L*, *jour* par *J*, etc.).

1. **Créer, en SQL, une vue**, nommée **PlanningSeance**, permettant de connaître pour chaque séance les informations de planning associé. *Sur les instances données précédemment, la requête **SELECT * FROM PlanningSeance** doit renvoyer le résultat suivant (l'ordre des nuplets et des attributs n'a aucune importance) :*

PID	SID	dateseance	nbinscrits	lieu	jour	heure
1	1	06/01/2022	1	Boulogne	jeudi	19h
1	2	13/01/2022	3	Boulogne	jeudi	19h
2	3	09/01/2022	1	Boulogne	dimanche	10h30
2	4	16/01/2022	1	Boulogne	dimanche	10h30
3	5	09/01/2022	0	Vincennes	dimanche	14h
3	6	16/01/2022	0	Vincennes	dimanche	14h
4	7	03/01/2022	0	Vincennes	lundi	13h
4	8	10/01/2022	0	Vincennes	lundi	13h

2. **Exprimer les requêtes suivantes en algèbre relationnelle et en SQL.** *Aucune explication demandée. Vos requêtes doivent fonctionner sur les instances des relations données dans l'énoncé du sujet de l'examen mais aussi sur toutes les autres instances possibles. Vous pouvez utiliser la vue créée précédemment en la nommant *PS*.*

- (a) Qui (en précisant le nom et le prénom) est inscrit aux séances ayant lieu à Boulogne OU à celles ayant lieu à Vincennes ?
- (b) Qui (en précisant le nom et le prénom) est inscrit aux séances ayant lieu à Boulogne ET à celles ayant lieu à Vincennes ?
- (c) Qui (en précisant le nom et le prénom) est inscrit aux séances ayant lieu à Boulogne MAIS PAS à celles ayant lieu à Vincennes ?
- (d) Qui (en précisant le nom et le prénom) est inscrit à toutes les séances de la base de données ?
- (e) Quel jour de la semaine y-a-t-il au moins 2 séances de programmées ?

3. **Exprimer en SQL seulement les requêtes suivantes :** (*Aucune explication demandée*)

- (a) A combien de séances assiste chaque personne de la base (en indiquant zéro quand la personne n'assiste à aucune séance) ? *Vous précisez le nom et le prénom de chaque personne.*
- (b) Quelles sont (en donnant leur nom et prénom) les personnes les plus jeunes de la base ?

2.1.2 Exercice 2 - Dépendances fonctionnelles et SQL (*3 points soit 10%*)

1. Traduire (a) en français et (b) en SQL, sous la forme d'une contrainte, la dépendance fonctionnelle suivante, associée à la relation **planning** de l'exercice précédent : jour, heure \rightarrow lieu
2. Soit la relation **Voiture** dont le schéma est le suivant :

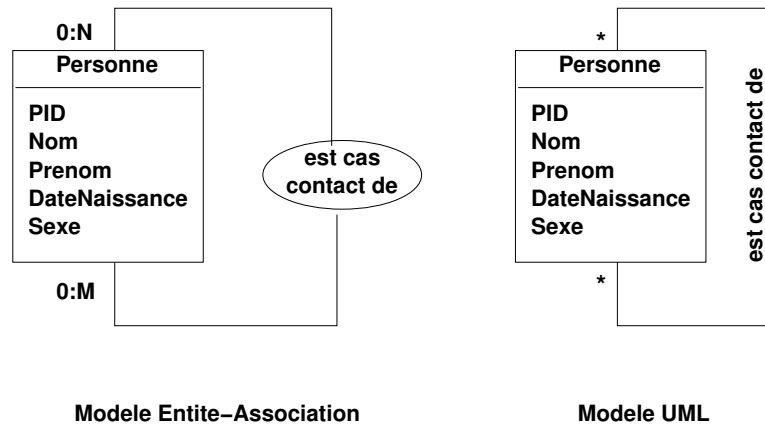
```
CREATE TABLE voiture
( immatriculation varchar(20) PRIMARY KEY,
  type varchar(25) NOT NULL,
  marque varchar(25) NOT NULL,
  dateCreation date NOT NULL DEFAULT CURRENT_DATE
);
```

On souhaite définir un déclencheur en SQL affichant un message d'erreur lorsque la dépendance fonctionnelle **Type \rightarrow Marque**, associée à la relation **Voiture**, n'est pas vérifiée. **Complétez le code du déclencheur ci-dessous en indiquant quelle requête SELECT ce déclencheur doit exécuter.**

```
CREATE OR REPLACE Function FonctionTypeMarque() returns trigger as
'DECLARE
  imm Voiture.Immatriculation%TYPE;
BEGIN
  SELECT INTO imm ... -- REQUÊTE A COMPLÉTER
  IF FOUND THEN
    RAISE EXCEPTION ''Dependance non verifiee'';
  ELSE RETURN NEW;
  END IF;
END;'
LANGUAGE 'plpgsql';
```

```
CREATE TRIGGER DFTTypeMarque BEFORE INSERT ON Voiture
FOR EACH ROW
EXECUTE procedure FonctionTypeMarque() ;
```

2.1.3 Exercice 3 - Passage au relationnel (4 points soit 13%)



On souhaite définir une base de données permettant de gérer des cas contacts. Une personne est identifiée par un attribut nommé PID et est caractérisée par un nom, un prénom, une date de naissance et un sexe. Une personne peut être cas contact d'une ou plusieurs personnes. La modélisation Entité/Association et celle en UML sont représentées sur la figure page 82.

Déduisez le schéma relationnel de la base de données correspondante.

Vous préciserez les clés primaires (artificielles) des relations en les soulignant ainsi que les clés étrangères en les signalant par un # et en précisant à quoi elles font référence. *Aucune explication n'est demandée.*

Dans votre schéma relationnel, chaque relation doit être spécifiée de la manière suivante : $Nom(\underline{att_1}, \dots, att_n)$ où Nom est le nom de la relation et att_1, \dots, att_n sont des noms d'attributs. Le nom de la relation doit obligatoirement avoir un lien avec les noms des ensembles d'entités (classes) ou des associations du schéma de modélisation.

Exercice 4 - Dépendances fonctionnelles et normalisation (10 points soit 34%)

Soit une relation $R(VaccinID, NomVaccin, Fabricant, PersonneID, Nom, Prenom, DateNaissance, DateVaccin, NumeroDose)$, avec pour famille de dépendances fonctionnelles associée :

$$F = \{ VaccinID \longrightarrow NomVaccin, Fabricant; \\ NomVaccin \longrightarrow Fabricant; \\ PersonneID \longrightarrow Nom, Prenom, DateNaissance; \\ VaccinID, NomVaccin, PersonneID, DateVaccin \longrightarrow NumeroDose; \\ \}$$

Pour simplifier, vous pourrez abréger les attributs par $VID, NV, F, PID, N, P, DN, DV$ et ND .

1. La contrainte "Deux fabricants peuvent avoir le même nom de vaccin" est-elle vérifiée par ce schéma de bases de données ? Si la réponse est positive, expliquer pourquoi. Si la réponse est négative, indiquez quelle(s) dépendance(s) fonctionnelle(s) il faut ajouter/supprimer ou modifier pour que la contrainte soit vérifiée.

A partir de la famille F de dépendances fonctionnelles initiale donnée dans l'énoncé :

2. Indiquer quelles sont la (ou les) clé(s) minimale(s) de la relation R . Aucune explication demandée.
3. Peut-on supprimer la dépendance fonctionnelle $VaccinID \rightarrow Fabriquant$ de F et la retrouver à partir des autres dépendances fonctionnelles de F ? Expliquez.
4. Peut-on déduire la dépendance fonctionnelle $VaccinID, PersonneID, DateVaccin \rightarrow NumeroDose$ de F ? Expliquez.
5. La famille F est-elle minimale? Montrez-le ou, si F n'est pas minimale, proposez une famille minimale équivalente. Expliquez.
6. Indiquer en quelle forme normale est la relation R . Expliquez.
7. Proposer une décomposition de R , en relations BNCF, qui soit sans perte d'information et sans perte de dépendances et contenant le moins de relations possible. Expliquer.

2.2 Partiel d'Octobre 2021

Soit la base de données de schéma ci-dessous, permettant de gérer des recettes de cuisine :

Ingredient(IID, Nom)

Recette(RID, Titre, Description, Categorie)

Composition(RID, IID, Quantite, Unite)

Les attributs soulignés sont les clés primaires des relations. Les attributs IID et RID sont des clés artificielles. L'attribut IID dans la relation **Composition** est une clé étrangère qui fait référence à la clé primaire de la relation **Ingredient**. De même, l'attribut RID dans la relation **Composition** est une clé étrangère qui fait référence à la clé primaire de la relation **Recette**. Les attributs IID, RID et Quantite sont de type entier. Les autres attributs sont de type chaînes de caractères. Seul l'attribut Unite peut prendre la valeur NULL.

Le tableau suivant donne un exemple d'instance de la relation **Ingredient** :

<u>IID</u>	<u>Nom</u>
1	farine
2	sucré
3	œuf
4	beurre
5	huile

Le tableau suivant donne un exemple d'instance de la relation **Recette** (l'attribut **Description** n'est pas représenté en totalité) :

RID	Titre	Description	Categorie
1	Cookies	Pour 20 cookies environ ...	Goûter
2	Béchamel	Pour réaliser une sauce béchamel ...	Sauce
3	Gâteau au yaourt	Pour 6 personnes ...	Dessert

Le tableau ci-dessous donne un exemple d'instance de la relation **Composition** :

RID	IID	Quantite	Unite
1	1	350	g
1	2	250	g
1	3	2	
1	4	250	g

*Pour simplifier, dans la suite, vous pouvez abréger le nom des relations et des attributs par leur initiales (par exemple *I* pour *Ingredient*, *R* pour *Recette*, *N* pour *Nom* etc.).*

1. Quelle(s) contrainte(s) faut-il ajouter dans la base de données pour interdire la création de 2 recettes de même titre? Expliquez.
2. Faut-il ajouter l'attribut **Quantite** dans la clé primaire de **Composition**? Expliquez.
3. Indiquez si les insertions suivantes sont possibles ou non dans les instances précédentes. Expliquez.
 - (a) Insertion du nuplet (2,6,150,'ml') dans l'instance de la relation **Composition**.
 - (b) Insertion du nuplet (1,2,200,'g') dans l'instance de la relation **Composition**.
 - (c) Insertion du nuplet (2,4,50,'g') dans l'instance de la relation **Composition**.
4. **A partir des instances, données précédemment** (sans tenir compte des insertions de la question précédente) :
 - donnez le résultat ¹¹ des requêtes suivantes
 - et donnez l'intitulé des requêtes en français.

Aucune explication demandée.

(a) $\Pi_{Nom}(Ingredient \bowtie [\Pi_{IID}(Ingredient) - \Pi_{IID}(Composition)])$

(b) $\{r.Titre / Recette(r) \wedge (\forall i Ingredient(i) \implies [\exists c Composition(c) \wedge (c.IID = i.IID) \wedge (r.RID = c.RID)])\}$

11. Si le résultat est vide, vous préciserez au minimum le schéma de la table résultat

5. **Exprimez, les requêtes¹² suivantes en algèbre relationnelle et en calcul relationnel.**

Aucune explication demandée.

Vous pouvez utiliser une requête précédente en nommant la requête R_i , où i est le numéro de la requête (a, b, etc.).

- (a) Quels sont les ingrédients (en donnant leur nom, leur quantité et leur unité) nécessaires pour réaliser des cookies (i.e. la recette dont le Titre est Cookies) ?
 - (b) Quelles recettes (en précisant leur titre) peut-on réaliser avec moins de 300g de farine ?
 - (c) Quelles recettes (en précisant leur titre) contiennent du beurre OU de l'huile ?
 - (d) Quelles recettes (en précisant leur titre) contiennent de la farine ET du beurre ?
 - (e) Quelles recettes (en précisant leur titre) ne contiennent PAS d'œuf ?
 - (f) Quels ingrédients (en donnant leur nom) apparaissent, avec la même quantité, dans toutes les recettes de la base de données ?
6. On suppose qu'une instance de la relation **Composition** contient \mathcal{C} nuplets et qu'une instance de la relation **Ingredient** en contient \mathcal{I} .

Indiquez quel est le nombre de nuplets résultat de chaque requête ci-dessous.

Expliquez.

Lorsque vous ne pouvez pas préciser le nombre exact de nuplets résultat, vous devez préciser une borne supérieure (ex. nombre de nuplets résultats $\leq y$ où y est une borne que vous devez définir).

- (a) $Ingredient \times Composition$
 - (b) $Ingredient \bowtie Composition$
 - (c) $\Pi_{IID, RID}(Composition) \div \Pi_{IID}(Ingredient)$
7. **Exprimez la requête de la question 4(a) en calcul relationnel à variable nuplet.**
Aucune explication demandée.
8. **Exprimez la requête de la question 4(b) en algèbre relationnelle.**
Aucune explication demandée.

12. Vos requêtes doivent pouvoir être exécutées sur toutes les instances possibles, pas uniquement celles données en exemple.

2.3 Examen de Juin 2021

2.3.1 Exercice 1 - Algèbre relationnelle, calcul relationnel à variable nuplet et SQL (20 points soit 50%)

Soit le schéma d'une base de données, décrit en SQL ci-dessous, permettant de gérer des feuilles d'émargement d'examens. Lorsqu'un(e) étudiant(e) signe une feuille d'émargement pour un examen donné, l'attribut **Presence** a pour valeur **TRUE**. On supposera qu'il n'existe pas d'homonyme¹³.

```
CREATE TABLE Examen
( ExID    serial,
  DateEx  Date NOT NULL,
  Lieu    varchar(100) NOT NULL,
  CONSTRAINT UN_Examen UNIQUE (DateEx,Lieu),
  CONSTRAINT PK_Examen PRIMARY KEY(ExID)
);

CREATE TABLE Etudiant
( EtID    serial,
  Nom     varchar(100) NOT NULL,
  Prenom  varchar(100) NOT NULL,
  CONSTRAINT UN_Etudiant UNIQUE (Nom,Prenom),
  CONSTRAINT PK_Etudiant PRIMARY KEY(EtID)
);

CREATE TABLE Emargement
( ExID    integer,
  EtID    integer,
  Present boolean DEFAULT FALSE,
  CONSTRAINT PK_Emargement PRIMARY KEY (ExID, EtID),
  CONSTRAINT "FK_Emargement_Examen" FOREIGN KEY (ExID)
    REFERENCES Examen (ExID) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT "FK_Emargement_Etudiant" FOREIGN KEY (EtID)
    REFERENCES Etudiant (EtID) ON UPDATE RESTRICT ON DELETE RESTRICT
) ;
```

Dans la suite, vous pourrez abréger le nom des relations par *Ex*, *Et*, *Em*, ainsi que les noms des attributs par leurs initiales (ex. *Nom* par *N*, *DateEx* par *D*, etc.).

1. Pour chaque requête ci-dessous, que se passe-t-il lorsqu'on exécute la requête sur les instances données page 45 ?
 - (a) `INSERT INTO Examen VALUES (2,'22/06/21','Amphi 4');` *Expliquer.*
 - (b) `INSERT INTO Emargement VALUES (1,3,False);` *Expliquer.*
 - (c) `INSERT INTO Etudiant(Nom,Prenom) VALUES ('Zarela', 'Maude');` *Expliquer.*
2. **Exprimer les requêtes suivantes en algèbre relationnelle, en calcul relationnel à variable nuplet et en SQL.** *Aucune explication demandée. Vos requêtes doivent fonctionner sur les instances des relations données à la fin de cette page, mais aussi sur toutes les autres instances possibles. Vous pouvez utiliser une requête précédente en nommant la requête R_i , où i est un numéro de la requête.*
 - (a) Quels étudiant(e)s (en donnant leur nom et leur prénom) doivent passer un examen en 'Amphi 4' OU un examen en 'Amphi 8'?

13. i.e. il n'existe pas 2 étudiant(e)s avec le même nom et le même prénom.

- (b) Quels étudiant(e)s (en donnant leur nom et leur prénom) doivent passer un examen en 'Amphi 4' ET un examen en 'Amphi 8' ?
- (c) Quels étudiant(e)s (en donnant leur nom et leur prénom) ont été absent(e)s (i.e. sont associé(e)s à **False** sur la feuille d'émargement) à au moins un examen ?
- (d) Quels étudiant(e)s (en donnant leur nom et leur prénom) ont toujours été présent(e)s à tous les examens (i.e. apparaissent présent(e)s (**True** sur la feuille d'émargement) pour tous les examens de la base de données) ?
3. **Exprimer en algèbre relationnelle et en calcul relationnel à variable nuplet, la requête suivante** (*Aucune explication demandée*) :

```
SELECT D,L FROM Examen ex
WHERE NOT EXISTS (SELECT * FROM Etudiant et
WHERE NOT EXISTS (SELECT * FROM Emargement em
WHERE ex.ExID=em.ExID AND et.EtID=em.EtID))
```

4. **Créer une vue en SQL** permettant d'afficher, pour chaque étudiant(e) (en donnant son nom et son prénom), le nombre d'examens qu'il(elle) doit passer (i.e. le nombre de fois où il(elle) apparaît sur une feuille d'émargement). *Aucune explication demandée.*
5. **Exprimer en calcul relationnel à variable nuplet et en SQL la requête suivante :**
 Quels sont les étudiant(e)s (en donnant leur nom et leur prénom) devant passer le plus grand nombre d'examens ?
Vous pouvez utiliser ou supposer existante la vue de la question précédente. Aucune explication demandée.

TABLE 17 – Une instance de la relation **Examen** de l'exercice 1

ExID	DateEx	Lieu
1	22/06/2021	Amphi 4

TABLE 18 – Une instance de la relation **Etudiant** de l'exercice 1

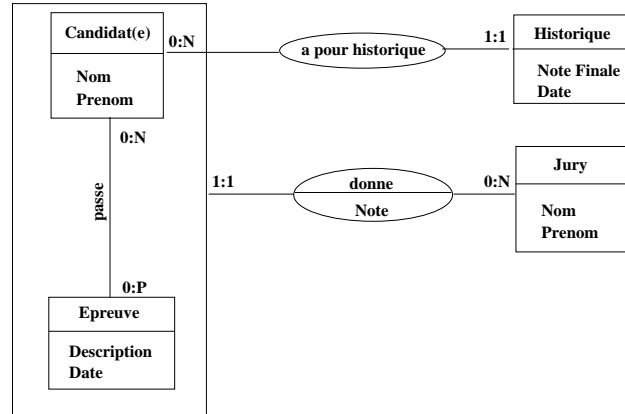
EtID	Nom	Prenom
1	Computing	Claude
2	Gamotte	Albert

TABLE 19 – Une instance de la relation **Emargement** de l'exercice 1

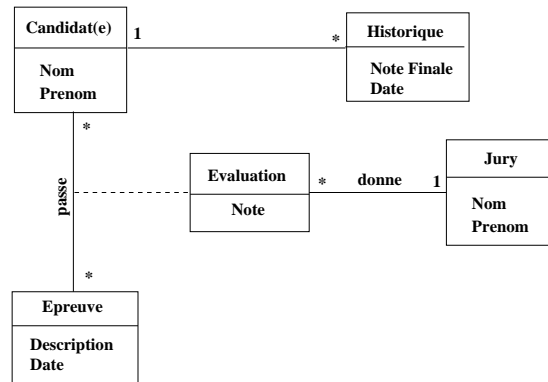
ExID	EtID	Present
1	1	TRUE
1	2	FALSE

2.3.2 Exercice 2 - Passage au relationnel (7 points soit 17.5%)

On souhaite définir une base de données permettant de gérer des jurys de concours (ex. concours de danse, de chant etc.). La modélisation Entité/Association et celle en UML sont représentées sur la figure page 46. Un(e) candidat(e) peut passer plusieurs épreuves. Pour chaque épreuve et chaque candidat(e), un membre du jury attribue une note. Un historique des notes (comprenant une note finale calculée à une certaine date à partir des notes affectées au candidat) est conservé pour chaque candidat(e) après chaque épreuve. Les identifiants ne sont pas représentés sur le schéma. Chaque ensemble d'entités ou classe possède un attribut unique (non représenté sur le schéma) de type entier dont la valeur s'incrémente automatiquement.



(a) Modelisation Entite/Association



(b) Modelisation UML

FIGURE 5 – Schémas de modélisation de l'exercice 2.

Déduisez le schéma relationnel de la base de données correspondante.

Vous préciserez les clés primaires (artificielles) des relations en les soulignant ainsi que les clés étrangères en les signalant par un # et en précisant à quoi elles font référence. *Aucune explication n'est demandée.*

Dans votre schéma relationnel, chaque relation doit être spécifiée de la manière suivante : $Nom(\underline{att_1}, \dots, att_n)$ où Nom est le nom de la relation et att_1, \dots, att_n sont des noms d'attributs. Le nom de la relation doit obligatoirement avoir un lien avec les noms des ensembles d'entités (classes) ou des associations du schéma de modélisation.

2.3.3 Exercice 3 - Dépendances fonctionnelles et normalisation (13 points soit 32.5%)

Soit une relation $R(NomEtudiant, PrenomEtudiant, NumCarteEtudiant, Formation, Cours, Note)$, dont une partie d'une de ses instances est donnée par le tableau ci-dessous.

Pour simplifier, vous pourrez abréger les attributs par NE , PE , $NumC$, F , E , C , et N .

NomEtudiant	PrenomEtudiant	NumCarteEtudiant	Formation	Enseignant	Cours	Note
GAMOTTE	Albert	12345	L3 MIDO	M. Manouvrier	BD	15
COMPUTING	Claude	45678	L3 MIDO	M. Manouvrier	BD	12
ZARELA	Maude	34567	L3 MIDO	C. Murat	RO	14
COMPUTING	Claude	45678	L3 MIDO	C.Murat	RO	10

- Déterminer la famille de dépendances fonctionnelles F , associée à R , sachant qu'on a les contraintes suivantes (à une contrainte correspond une DF) :
 - Un(e) étudiant(e) est identifié(e) par le numéro de sa carte d'étudiant(e). Mais, il peut y avoir des homonymes (i.e. des étudiant(e)s ayant le même nom et le même prénom).
 - Un(e) étudiant(e) est dans une et une seule formation.
 - Un cours, dans une formation donnée, est enseigné par un(e) et un(e) seul(e) enseignant(e). Mais un même cours, dans deux formations différentes, peut être donné par deux enseignant(e)s différent(e)s.
 - Un(e) étudiant(e) a une et une seule note par cours, mais suit plusieurs cours de sa formation.
- A partir de la famille F , déduite de la question précédente, déterminer les clés minimales de R . Aucune explication demandée.
- En quelle forme normale est la relation R ? Expliquer.
- Proposer une décomposition de R qui soit sans perte d'information et sans perte de dépendances, contenant le moins de relations possible et dont les relations sont toutes BCNF. Expliquer.

2.4 Examen de Janvier 2021

2.4.1 Exercice 1 - Algèbre relationnelle et SQL (25 points soit 50%)

Soit le script SQL suivant, permettant de gérer les cas contacts dans le cadre d'une épidémie.

```
CREATE TABLE Personne(  
    PID serial PRIMARY KEY,  
    nom varchar(25) NOT NULL,  
    prenom varchar(25) NOT NULL,  
    dateNaissance date NOT NULL,  
    sexe varchar(1) NOT NULL,  
    CONSTRAINT UK_Personne UNIQUE(nom,prenom,dateNaissance),  
    CONSTRAINT CK_Sexe CHECK (sexe IN ('M','F','N'))  
);  
  
CREATE TABLE Contact(  
    PID1 integer,  
    PID2 integer,  
    CONSTRAINT PK_Contact PRIMARY KEY (PID1,PID2),  
    CONSTRAINT FK_Contact_PID1 FOREIGN KEY (PID1) REFERENCES Personne(PID),  
    CONSTRAINT FK_Contact_PID2 FOREIGN KEY (PID2) REFERENCES Personne(PID)  
);  
  
CREATE TABLE Test(  
    TID serial PRIMARY KEY,  
    PID integer NOT NULL,  
    positif boolean,  
    dateTest date NOT NULL DEFAULT CURRENT_DATE,  
    CONSTRAINT UK_Test UNIQUE(PID,dateTest),  
    CONSTRAINT FK_Test_PID FOREIGN KEY (PID) REFERENCES Personne(PID)  
);
```

Lorsqu'une personne, identifiée par x , est en contact avec une personne, identifiée par y , alors 2 nuplets (x, y) et (y, x) sont insérés de la relation *Contact*.

Des instances sont présentées dans les tables 1 à 3 de la dernière page.

Dans la suite, vous pourrez abrégier le nom des relations par P , C et T , ainsi que les noms des attributs par leurs initiales (ex. *nom* par N , *dateNaissance* par D , etc.).

NB : Les questions qui suivent sont indépendantes les unes des autres.

1. Indiquez, en analysant leur schéma ci-dessus, pour chacune des relations de la base de données, **quelle(s) est(sont) les clés minimales possibles**? *Aucune explication demandée.*
2. **Exprimez, les requêtes¹⁴ suivantes en algèbre relationnelle, en calcul relationnel à variable nuplet et en SQL.**
Aucune explication demandée.

Vous pouvez utiliser une requête précédente en nommant la requête R_i , où i est un numéro de la requête.

14. Vos requêtes doivent pouvoir être exécutées sur toutes les instances possibles, pas uniquement celles données en exemple. Les requêtes peuvent renvoyer une table vide sur les instances exemples.

Vous pouvez également abréger les valeurs et par exemple mettre 'G' au lieu de 'Gamotte', 'A' au lieu de 'Albert', 'Z' au lieu de 'Zarela', 'M' au lieu de 'Maude', etc.

- (a) Qui (en donnant le nom et le prénom) a réalisé au moins un test ?
- (b) Qui (en donnant le nom et le prénom) a réalisé au moins 2 tests ?
- (c) Qui (en donnant le nom et le prénom) est **n'est pas** en contact avec Albert GAMOTTE ?
- (d) Qui (en donnant le nom et le prénom) est en contact avec Albert GAMOTTE **OU** Maude ZARELA ?
- (e) Qui (en donnant le nom et le prénom) est en contact avec Albert GAMOTTE **ET** Maude ZARELA ?
- (f) Qui (en donnant le nom et le prénom) est en contact avec toutes les personnes testées positives ?

3. Soit le déclencheur suivant :

```
CREATE OR REPLACE Function FonctionTrigger() returns trigger as
'DECLARE
  p1 Contact.pid1 %TYPE;
  p2 Contact.pid2 %TYPE;

BEGIN
  SELECT INTO p1,p2 pid1,pid2
  FROM Contact
  WHERE (pid1 =NEW.pid2)
  AND (pid2 =NEW.pid1);

  IF NOT FOUND THEN
    INSERT INTO Contact(pid1,pid2) VALUES(NEW.pid2,NEW.pid1);
  END IF;

  RETURN NEW;

END;'
```

```
LANGUAGE 'plpgsql';
```

```
CREATE TRIGGER MonTrigger AFTER INSERT ON Contact
FOR EACH ROW
EXECUTE procedure FonctionTrigger() ;
```

- (a) **Que se passe-t-il si on exécute la requête suivante : INSERT INTO Contact VALUES (1,3); ? Expliquez.**
- (b) **Que se passe-t-il si on remplace le AFTER INSERT par BEFORE INSERT et que l'on exécute la requête suivante : INSERT INTO Contact VALUES (3,4); ? Expliquez.**

2.4.2 Exercice 2 - Passage au relationnel et SQL (5 points soit 10%)

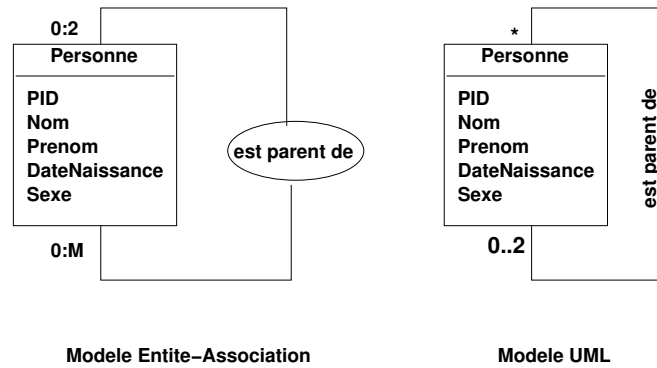


FIGURE 6 – Modèle Entité-Association et modèle UML de l'exercice 2

On souhaite définir une base de données permettant de gérer des liens de parenté et de filiation entre personnes, chaque personne étant identifiée par l'attribut PID. La modélisation Entité/Association et la modélisation UML de cette base de données sont représentées sur la figure 5 (page 82).

Déduisez le script SQL permettant de définir le schéma relationnel de la base de données de la figure 1.

Vous préciserez bien les clés primaires (artificielles ou non) des relations, les clés étrangères et toutes les contraintes de domaine que vous jugerez nécessaire d'ajouter. *Aucune explication n'est demandée - vous pouvez vous inspirer du script SQL de l'exercice précédent.*

Dans votre schéma relationnel, le nom des relations et les noms des attributs doivent obligatoirement avoir un lien avec les noms des ensembles d'entités (ou classes), des attributs ou des associations du schéma de modélisation.

2.4.3 Exercice 3 - Dépendances fonctionnelles et Décomposition de schéma (20 points soit 40%)

Soit une relation $R(NomEtudiant, NomProf, PrenomEtudiant, PrenomProf, EtudiantID, ProfID, UE, Note, Commentaire, AdresseMail)$, avec pour famille de dépendances fonctionnelles associée :

$$F = \{ \begin{array}{l} EtudiantID \longrightarrow NomEtudiant, PrenomEtudiant; \\ ProfID \longrightarrow NomProf, PrenomProf; \\ ProfID \longrightarrow AdresseMail; \\ NomProf, PrenomProf \longrightarrow AdresseMail; \\ EtudiantID, UE, ProfID \longrightarrow Commentaire, Note; \\ EtudiantID, NomEtudiant, PrenomEtudiant, NomProf, PrenomProf, UE, ProfID \longrightarrow Commentaire, Note \end{array} \}$$

Dans la suite, vous pourrez abréger le nom des relations par *NomEtudiant* par *NE*, *PrenomEtudiant* par *PE*, *EtudiantID* par *EID*, etc.).

1. **Les contraintes ci-dessous sont-elles vérifiées par ce schéma de bases de données ?** Si la réponse est positive, expliquer pourquoi. Si la réponse est négative, indiquez quelle(s) dépendance(s) fonctionnelle(s) il faut ajouter/supprimer ou modifier pour que la contrainte soit vérifiée.

- (a) "Il y a un seul professeur par UE."
- (b) "Un étudiant a une seule note et un seul commentaire par UE."
- (c) "Un étudiant a une seule note et un seul commentaire par UE et par professeur."

A partir des familles de dépendances fonctionnelles initiales données dans l'énoncé :

2. **Indiquez quelles sont la (ou les) clé(s) minimale(s) de la relation R . Aucune explication demandée.**
3. **Pour chacune des dépendances fonctionnelles suivantes, peut-on la supprimer de F et la retrouver à partir des autres dépendances fonctionnelles de F ? Expliquez.**
 - (a) $ProfID \rightarrow AdresseMail$
 - (b) $EtudiantID, NomEtudiant, PrenomEtudiant, NomProf, PrenomProf, UE, ProfID \rightarrow Commentaire, Note$
4. **La famille F est-elle minimale ? Montrez-le ou, si F n'est pas minimale, proposez une famille minimale équivalente. Expliquez.**
5. **Indiquez en quelle forme normale est la relation R . Expliquez.**
6. **La décomposition de R en R_1 et R_2 , ci-dessous, est-elle sans perte d'information et sans perte de dépendances ? Expliquez.**

$R_1(ProfID, NomProf, PrenomProf, AdresseMail)$

$R_2(NomEtudiant, PrenomEtudiant, EtudiantID, ProfID, UE, Note, Commentaire)$

7. **Proposez une décomposition de R qui soit sans perte d'information (SPI) et sans perte de dépendances (SPD), dont les relations issues de la décomposition sont en forme normale BCNF et qui contienne le moins de relations possibles. Expliquez clairement et donnez les détails de votre décomposition.**

Les instances de l'exercice 1 :

TABLE 20 – Une instance de la relation *Personne*

PID	Nom	Prenom	DateNaissance	Sexe
1	Gamotte	Albert	1991/04/24	M
2	Pabien	Yvon	1975/05/11	M
3	Computing	Claude	2000/01/24	M
4	Zarela	Maude	1991/12/24	F

TABLE 21 – Une instance de la relation *Contact*

PID1	PID2
2	3
3	2
2	1
1	2
2	4
4	2

TABLE 22 – Une instance de la relation *Test*

TID	PID	positif	dateTest
1	1	False	2020/11/15
2	1	True	2020/11/30
3	2	NULL	2020/12/01

2.5 Partiel de Décembre 2020

Le script SQL ci-dessous modélise une base de données permettant de gérer les informations d'un concours de chant.

Un candidat est décrit par un nom d'artiste et un sexe. On supposera que chaque nom d'artiste est unique. Les candidats sont répartis dans des équipes, chacune gérée par un coach. Un candidat peut changer d'équipe (on stocke dans la base la date d'entrée dans chaque équipe pour chaque candidat). Les candidats sont évalués au sein de *battles*. Lors d'une *battle*, deux candidats s'opposent et seul un des deux est vainqueur, l'autre étant éliminé du concours.

```
CREATE TABLE Candidat (  
  CID serial,  
  NomArtiste varchar(25) NOT NULL UNIQUE, -- Nom d'artiste du candidat  
  Sexe varchar(1)  
  CONSTRAINT CK_Sexe CHECK (Sexe IN ('M','F','N')),  
  Elimine boolean DEFAULT False, -- True si le candidat est éliminé du jeu False sinon  
  CONSTRAINT PK_Candidat PRIMARY KEY(CID)  
);  
  
CREATE TABLE Equipe (  
  EID serial,  
  Coach varchar(25) NOT NULL UNIQUE, -- Nom du Coach  
  CONSTRAINT PK_Equipe PRIMARY KEY(EID)  
);  
  
CREATE TABLE Repartition (  
  RID serial,  
  CID integer NOT NULL, -- ID du candidat  
  EID integer NOT NULL, -- ID de l'équipe  
  dateR date DEFAULT CURRENT_DATE NOT NULL, -- Date d'entrée dans l'équipe  
  CONSTRAINT PK_Repartition PRIMARY KEY(RID),  
  CONSTRAINT FK_Repartition_CID FOREIGN KEY (CID) REFERENCES Candidat(CID),  
  CONSTRAINT FK_Repartition_EID FOREIGN KEY (EID) REFERENCES Equipe(EID),  
  CONSTRAINT UN_Repartition UNIQUE(EID,CID,dateR)  
);  
  
CREATE TABLE Battle (  
  BID serial,  
  CID1 integer NOT NULL, -- ID du candidat 1  
  CID2 integer NOT NULL, -- ID du candidat 2  
  dateB date DEFAULT CURRENT_DATE NOT NULL, -- Date de la battle  
  Vainqueur integer NOT NULL, -- ID du candidat ayant remporté la battle  
  CONSTRAINT PK_Battle PRIMARY KEY(BID),  
  CONSTRAINT FK_Battle_CID1 FOREIGN KEY (CID1) REFERENCES Candidat(CID),  
  CONSTRAINT FK_Battle_CID2 FOREIGN KEY (CID2) REFERENCES Candidat(CID),  
  CONSTRAINT FK_Battle_Vainqueur FOREIGN KEY (Vainqueur) REFERENCES Candidat(CID),  
  CONSTRAINT UN_Battle UNIQUE(CID1,CID2,dateB)  
);  
  
CREATE TABLE Perdant (  
  PID serial,  
  CID integer NOT NULL, -- ID du candidat  
  BID integer NOT NULL, -- ID de la battle pendant laquelle le candidat a été éliminé  
  CONSTRAINT PK_Perdant PRIMARY KEY(PID),  
  CONSTRAINT FK_Perdant_CandidatID FOREIGN KEY (CID) REFERENCES Candidat(CID),  
  CONSTRAINT FK_Perdant_BID FOREIGN KEY (BID) REFERENCES Battle(BID),  
  CONSTRAINT UN_Perdant UNIQUE(CID,BID)  
);
```

Des instances des 3 premières relations sont présentées dans les tables 1 à 3 de la dernière page. L'instance de la relation *Perdant* (non représentée) ne contient aucun n-uplet.

Dans la suite, vous pourrez abréger le nom des relations par *C*, *E*, *R*, *B*, et *P*, ainsi que les noms des attributs par leurs initiales (ex. *NomArtiste* par *NA*, *Sexe* par *S*, etc.).

1. Indiquez, pour chacune des relations de la base de données, **quelle(s) est(sont) la(les) clé(s) minimale(s) possible(s)** ? *Aucune explication demandée.*
2. **A partir des instances de la dernière page, indiquez si les insertions** ¹⁵ **suivantes sont possibles ou non.** *Expliquer. (3,5 points)*
 - (a) INSERT INTO Perdant(CID,BID) VALUES (2,1) ;
 - (b) INSERT INTO Perdant(CID,BID) VALUES (7,1) ;
 - (c) INSERT INTO Repartition(CID,EID,dateR) VALUES (1,3,'04/08/2020') ;
3. **Exprimez, les requêtes** ¹⁶ **suivantes en algèbre relationnelle, en calcul relationnel à variable nuplet et en SQL.**
Aucune explication demandée.
Vous pouvez utiliser une requête précédente en nommant la requête R_i , où i est un numéro de requête.
 - (a) Quels artistes (en donnant le nom d'artiste) composent l'équipe du coach nommé *Amel* ?
 - (b) Quels artistes (en donnant le nom d'artiste) composent l'équipe du coach nommé *Amel* OU celle du coach *Pascal Obispo* ?
 - (c) Qui (en donnant le nom d'artiste) n'a gagné aucune *battle* ?
 - (d) Qui (en donnant le nom d'artiste) a participé à au moins une *battle* ?
 - (e) Qui (en donnant le nom d'artiste) a participé à au moins 2 *battles* ?
 - (f) Qui (en donnant le nom d'artiste) a gagné toutes les *battles* auxquelles il/elle a participé ?
4. **Exprimez, en SQL uniquement, les requêtes suivantes** (*Aucune explication demandée*) :
 - (a) Quel est le nombre de participations aux *battles* par candidat (en donnant le nom d'artiste) et par date (en donnant la date de la *battle*) ?
 - (b) Qui (en donnant le nom d'artiste) a participé au plus grand nombre de *battles* lors de la dernière date ¹⁷ ? (*Vous pouvez utiliser la requête précédente*)
 - (c) Quel est le nombre de *battles* perdues et de *battles* gagnées par candidat (en donnant le nom d'artiste) ?
 - (d) Le candidat n°2 est éliminé du jeu. Donner la (ou les) requête(s) de mise à jour du n-uplet correspondant.

15. On supposera que tous les domaines des attributs sont corrects.

16. Vos requêtes doivent pouvoir être exécutées sur toutes les instances possibles, pas uniquement celles données en exemple. Les requêtes peuvent renvoyer une table vide sur les instances exemples.

17. i.e. la date la plus grande/récente.

Les instances de l'exercice :

TABLE 23 – Une instance de la relation *Candidat*

CID	NomArtiste	Sexe	Elimine
1	Abi	M	False
2	Gustine	F	False
3	Tom Rochet	M	False
4	Cheyenne	F	False
5	Baby J	F	False
6	Antoine Delie	M	False

TABLE 24 – Une instance de la relation *Equipe*

EID	Coach
1	Lara Fabian
2	Amel
3	Pascal Obispo
4	Marc Lavoine

TABLE 25 – Une instance de la relation *Repartition*

RID	CID	EID	DateR
1	1	3	04/02/2020
2	5	3	04/02/2020
3	2	1	04/02/2020
4	4	1	04/02/2020
5	3	2	04/02/2020
6	6	4	04/02/2020
7	1	2	05/03/2020

TABLE 26 – Une instance de la relation *Battle*

BID	CID1	CID2	DateB	Vainqueur
1	1	2	04/10/2020	1
2	1	3	04/10/2020	1
3	1	4	05/02/2020	1
4	1	5	05/02/2020	1
5	1	2	05/02/2020	1

2.6 Examen de Janvier 2020

2.6.1 Exercice 1 - Algèbre relationnelle et SQL (21 points soit 52.5%)

Soit le script SQL suivant, permettant de gérer des formations, des étudiants (dont on connaît la formation courante) et les candidatures d'étudiants à des formations.

```
CREATE TABLE Formation
( FID      serial,
  NomF     varchar(50) NOT NULL UNIQUE,
  Niveau   varchar(2) NOT NULL,
  CONSTRAINT PK_Formation PRIMARY KEY (FID),
  CONSTRAINT CK_Formation CHECK (Niveau IN ('L1','L2','L3','M1','M2'))
);

CREATE TABLE Etudiant
( EID      serial,
  NomE     varchar(25) NOT NULL,
  Prenom   varchar(25) NOT NULL,
  DateNaissance date NOT NULL,
  FID int4 NOT NULL, -- Formation actuelle de l'étudiant
  CONSTRAINT PK_Etudiant PRIMARY KEY (EID),
  CONSTRAINT UN_Etudiant UNIQUE (NomE,Prenom,DateNaissance),
  CONSTRAINT FK_Etudiant_Formation FOREIGN KEY (FID)
    REFERENCES Formation (FID) ON UPDATE RESTRICT ON DELETE RESTRICT
);

CREATE TABLE Candidature
( CID      serial,
  EID      int4 NOT NULL,
  FID      int4 NOT NULL, -- formation où le candidat souhaite s'inscrire
  Statut   varchar(25) NOT NULL DEFAULT 'Non Vue',
  CONSTRAINT PK_Candidature PRIMARY KEY (CID),
  CONSTRAINT FK_Candidature_Etudiant FOREIGN KEY (EID)
    REFERENCES Etudiant (EID) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT FK_Candidature_Formation FOREIGN KEY (FID)
    REFERENCES Formation (FID) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT UN_Candidature UNIQUE (EID,FID),
  CONSTRAINT CK_Candidature CHECK (Statut IN ('Acceptee','Refusee','Non Vue', 'Liste attente'))
);

CREATE TABLE ListeAttente
( LID      serial,
  EID      int4 NOT NULL,
  FID      int4 NOT NULL,
  Rang     int4,
  CONSTRAINT PK_ListeAttente PRIMARY KEY (LID),
  CONSTRAINT FK_ListeAttente_Etudiant FOREIGN KEY (EID)
    REFERENCES Etudiant (EID) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT FK_ListeAttente_Formation FOREIGN KEY (FID)
    REFERENCES Formation (FID) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT UN_ListeAttente UNIQUE (EID,FID)
);
```

Des instances de 3 premières relations sont présentées dans les tables 1 à 3 de la dernière page. L'instance de la relation `ListeAttente` (non représentée) ne contient aucun nuplet.

Dans la suite, vous pourrez abréger le nom des relations par *F*, *E*, *C* et *LA*, ainsi que les noms des attributs par leurs initiales (ex. *NomF* par *NF*, *Niveau* par *N*, etc.).

- Indiquez, pour chacune des relations de la base de données, **quelle(s) est(sont) les clés minimales possibles**? *Aucune explication demandée.*

- Exprimez, les requêtes¹⁸ suivantes en algèbre relationnelle et en SQL.**

Aucune explication demandée.

Vous pouvez utiliser une requête précédente en nommant la requête R_i , où i est un numéro de la requête.

- Qui (en donnant le nom et le prénom) candidate à une formation de même niveau que sa formation actuelle? (*Par exemple, dans les instances exemples, l'étudiant numéro 3 candidate dans une formation (identifiée par 6) de niveau M2 alors que sa formation actuelle (identifiée par 4) est également de niveau M2.*)
 - Qui (en donnant le nom et le prénom) candidate dans une formation de niveau M2 OU M1?
 - Qui (en donnant le nom et le prénom) candidate dans une formation de niveau M2 ET dans une formation de niveau M1?
 - Qui (en donnant le nom et le prénom) candidate dans toutes les formations (y compris sa formation actuelle)?
- Exprimez, en SQL uniquement,** les requêtes suivantes (*Aucune explication demandée*) :
 - Quel est le nombre de candidats par formation? (*en précisant le nom de la formation et uniquement pour les formations où il y a au moins une candidature*)
 - Quel est le nombre de candidats par formation en affichant zero pour les formations sans candidat? (*en précisant le nom de la formation*)
 - Qui (en donnant le nom et le prénom) est le plus jeune étudiant?
 - Soit le déclencheur suivant :

```
CREATE OR REPLACE FUNCTION FunctionTriggerCandidature() RETURNS trigger AS
'DECLARE
    f Formation.FID%TYPE;
    nb int4;

BEGIN
    SELECT INTO f FID FROM Etudiant WHERE EID=NEW.EID ;

    SELECT INTO nb COUNT(*) FROM Candidature WHERE EID=NEW.EID ;

    IF f = NEW.FID THEN
        RAISE EXCEPTION ''Inscription impossible'';
    END IF;

    IF nb=3 THEN
        RAISE EXCEPTION ''Inscription impossible'';
    END IF;

    RETURN NEW;
END;'
LANGUAGE 'plpgsql';
```

18. Vos requêtes doivent pouvoir être exécutées sur toutes les instances possibles, pas uniquement celles données en exemple. Les requêtes peuvent renvoyer une table vide sur les instances exemples.

```
CREATE TRIGGER InsertionCandidature
BEFORE INSERT ON Candidature
FOR EACH ROW
EXECUTE PROCEDURE FunctionTriggerCandidature();
```

- (a) **Que se passe-t-il si on exécute la requête suivante : INSERT INTO Candidature VALUES (6,1,2); ? Expliquez.**
- (b) **Que se passe-t-il si on exécute la requête suivante : INSERT INTO Candidature VALUES (7,1,6); ? Expliquez.**
- (c) **Que se passe-t-il si on remplace le BEFORE INSERT par AFTER INSERT et que l'on exécute la requête suivante : INSERT INTO Candidature VALUES (8,2,6); ? Expliquez.**

5. Soit le déclencheur suivant :

```
CREATE OR REPLACE FUNCTION FunctionTriggerListeAttente() RETURNS trigger AS
'DECLARE
  m int4;

BEGIN
  SELECT INTO m MAX(Rang) FROM ListeAttente WHERE FID=NEW.FID ;

  IF m IS NULL THEN
    INSERT INTO ListeAttente(FID,EID,Rang) VALUES(NEW.FID,NEW.EID,1);
  END IF;

  IF m IS NOT NULL THEN
    INSERT INTO ListeAttente(FID,EID,Rang) VALUES(NEW.FID,NEW.EID,m+1);
  END IF;

  RETURN NEW;
END;'
```

```
LANGUAGE 'plpgsql';

CREATE TRIGGER InsertionListeAttente
AFTER UPDATE ON Candidature
FOR EACH ROW
WHEN (NEW.Statut = 'Liste attente')
EXECUTE PROCEDURE FunctionTriggerListeAttente();
```

Quels nuplets vont être mis à jour et/ou ajoutés après avoir exécuté les 2 requêtes suivantes :

```
UPDATE Candidature SET Statut = 'Liste attente' WHERE FID=3 AND EID=1;
UPDATE Candidature SET Statut = 'Liste attente' WHERE FID=3 AND EID=2;
```

Expliquez.

2.6.2 Exercice 2 - Passage au relationnel et SQL (6 points soit 15%)

On souhaite définir une base de données permettant un réseau social (i.e. des liens d'amitié entre personnes), chaque personne étant identifiée par l'attribut PID. La modélisation Entité/Association et la modélisation UML de cette base de données sont représentées sur la figure 7 (page 59).

Déduisez le script SQL permettant de définir le schéma relationnel de la base de données correspondante.

Vous préciserez bien les clés primaires (artificielles ou non) des relations, les clés étrangères et toutes les contraintes de domaine que vous jugerez nécessaire d'ajouter. Aucune explication

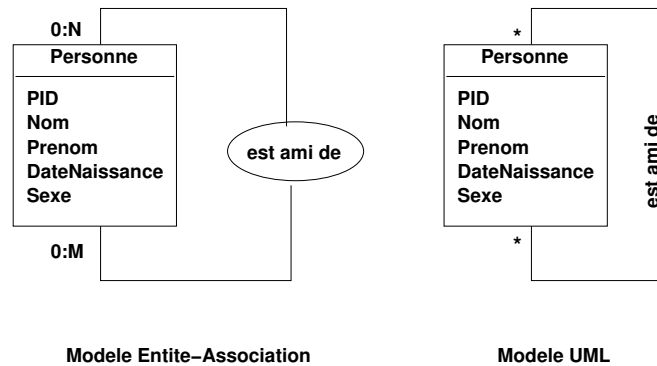


FIGURE 7 – Modèle Entité-Association et modèle UML de l'exercice 2

n'est demandée - vous pouvez vous inspirer du script SQL de l'exercice précédent.

Dans votre schéma relationnel, le nom des relations et les noms des attributs doivent obligatoirement avoir un lien avec les noms des ensembles d'entités (ou classes), des attributs ou des associations du schéma de modélisation.

2.6.3 Exercice 3 - Décomposition de schéma (13 points soit 32.5%)

Soit une relation $R(NumCom, DateCom, NumCli, AdrCli, NumProd, Prix, Qte)$, avec pour famille de dépendances fonctionnelles associée :

$$F = \{ \begin{array}{l} NumCom \longrightarrow DateCom, NumCli, AdrCli; \\ NumCom, NumProd \longrightarrow Prix, Qte; \\ NumCli \longrightarrow AdrCli; \\ NumProd \longrightarrow Prix \end{array} \}$$

1. Quelles sont la (ou les) clé(s) minimale(s) de la relation R ? Expliquez.
2. Indiquez, pour chaque contrainte ci-dessous, quelle DF permet de vérifier la contrainte ou quelle DF il faudrait ajouter ou supprimer pour permettre de vérifier la contrainte. Expliquez
 - (a) Un produit a un et un seul prix.
 - (b) Un client a plusieurs adresses.
3. Indiquez en quelle forme normale est la relation R . Expliquez.
4. La famille F est-elle minimale? Montrez-le ou, si F n'est pas minimale, proposez une famille minimale équivalente. Expliquez.
5. Proposez une décomposition de R qui soit sans perte d'information (SPI) et sans perte de dépendances (SPD), dont les relations issues de la décomposition sont en forme normale BCNF et qui contienne le moins de relations possibles. Expliquez clairement et donnez les détails de votre décomposition.

Les instances de l'exercice 1 :

TABLE 27 – Une instance de la relation *Formation*

FID	NomF	Niveau
1	Licence MIDO	L3
2	1ere année de Master Informatique	M1
3	MIAGE IF	M2
4	MIAGE ID	M2
5	MIAGE SITN	M2
6	IASD	M2

TABLE 28 – Une instance de la relation *Etudiant*

EID	NomE	Prenom	DateNaissance	FID
1	COMPUTING	Claude	'04/12/1997'	2
2	ZARELA	Maude	'03/04/1996'	2
3	PABIEN	Yvon	'03/04/1993'	4

TABLE 29 – Une instance de la relation *Candidature*

CID	EID	FID	Statut
1	1	3	Non Vue
2	1	4	Non Vue
3	2	3	Non Vue
4	2	5	Non Vue
5	3	6	Non Vue

2.7 Partiel d'Octobre 2019

Soit la base de données de schéma ci-dessous, permettant de gérer les emprunts d'une bibliothèque :

Personne(PID, Nom, Prenom, DateNaissance)

Livre(ISBN, Titre, Resume)

Emprunt(ISBN, PID, DateEmprunt, DateRetour)

Les attributs soulignés sont les clés primaires des relations. L'attribut **PID** dans la relation **Emprunt** est une clé étrangère qui fait référence à la clé primaire de la relation **Personne**. De même, l'attribut **ISBN** dans la relation **Emprunt** est une clé étrangère qui fait référence à la clé primaire de la relation **Livre**. L'attribut **PID** est de type entier, les attributs **DateEmprunt** et **DateRetour** sont de type date. Les autres attributs sont de type chaîne de caractères. Seul l'attribut **DateRetour** peut prendre la valeur NULL.

Le tableau suivant donne un exemple d'instance de la relation **Personne** :

PID	Nom	Prenom	DateNaissance
1	GAMOTTE	Albert	31/01/2000
2	DEBECE	Aude	23/11/1999
3	COMPUTING	Claude	05/12/2003
4	SUFFIT	Sam	01/07/1998
5	PABIEN	Yvon	04/05/2001

Le tableau suivant donne un exemple d'instance de la relation **Livre** (l'attribut **Resume** n'est pas représenté en totalité) :

ISBN	Titre	Resume
1	SQL	Ce livre parle de SQL ...
2	UML	Ce livre parle de UML ...
3	Les BD c'est la vie!	Cet ouvrage encense les bases de données...

Le tableau ci-dessous donne un exemple d'instance de la relation **Emprunt** :

ISBN	PID	DateEmprunt	DateRetour
1	1	29/09/19	10/10/19
1	2	01/10/19	
3	1	01/10/19	
2	3	01/10/19	15/10/19

*Pour simplifier, dans la suite, vous pouvez abréger le nom des relations et des attributs par leur initiales (par exemple **P** pour **Personne**, **L** pour **Livre**, **N** pour **Nom** etc.).*

1. Quelle(s) contrainte(s) faut-il ajouter dans la base de données pour interdire la création de 2 livres de même titre ? Expliquez.
2. Expliquer la différence (en donnant des exemples de nuplets éventuellement) entre choisir comme clé primaire de Emprunt, le couple (ISBN, PID) ou le triplet (ISBN, PID, DateEmprunt) ? Expliquez.
3. Indiquez si les insertions suivantes sont possibles ou non dans les instances précédentes. Expliquez.
 - (a) Insertion du nuplet (1,6,'28/10/19',NULL) dans l'instance de la relation Emprunt.
 - (b) Insertion du nuplet (1,2,'01/10/19','10/10/19') dans l'instance de la relation Emprunt.
 - (c) Insertion du nuplet (1,3,'28/10/19',NULL) dans l'instance de la relation Emprunt.
4. **A partir des instances, données précédemment** (sans tenir compte des insertions de la question précédentes), **donnez le résultats des requêtes suivantes.** *Aucune explication demandée.*
 - (a) $\Pi_{Nom,Prenom}(Personne \bowtie [\Pi_{PID}(Personne) - \Pi_{PID}(Emprunt)])$
 - (b) $\{l.Titre / Livre(l) \wedge [\exists e1 Emprunt(e1)(e1.ISBN = l.ISBN)] \wedge \neg [\exists e2 Emprunt(e2)(e2.ISBN = l.ISBN) \wedge (e1.PID \neq e2.PID)]\}$
5. **Exprimez, les requêtes**¹⁹ **suivantes en algèbre relationnelle et en calcul relationnel.** *Aucune explication demandée.*
Vous pouvez utiliser une requête précédente en nommant la requête R_i , où i est le numéro de la requête.
 - (a) Quelle est la date de naissance d'Albert GAMOTTE ?
 - (b) Quel livre (en précisant son titre) n'a été emprunté par personne ?
 - (c) Quelles sont les titres des livres empruntés par chaque personne ? *Votre requête devra afficher l'identificateur, le nom et le prénom des personnes et les titres des livres empruntés (une personne et un livre par nuplet). Par exemple, sur les instances données dans le sujet, la requête doit renvoyer la table 30 de la page suivante.*

PID	Nom	Prenom	Titre
1	GAMOTTE	Albert	SQL
1	GAMOTTE	Albert	Les BD c'est la vie !
2	DEBECE	Aude	SQL
3	COMPUTING	Claude	UML

TABLE 30 – Le résultat de la requête (c) sur les instances données en exemple.

- (d) Qui (en précisant le nom et le prénom) a emprunté le livre intitulé 'SQL' OU le livre intitulé 'UML' ?
- (e) Qui (en précisant le nom et le prénom) a emprunté le livre intitulé 'SQL' ET le livre intitulé 'UML' ?

19. Vos requêtes doivent pouvoir être exécutées sur toutes les instances possibles, pas uniquement celles données en exemple.

- (f) Qui (en précisant le nom et le prénom) a emprunté le livre intitulé 'SQL' MAIS PAS le livre intitulé 'UML' ?
 - (g) Qui (en précisant le nom et le prénom) a emprunté tous les livres ?
6. On suppose qu'une instance de la relation **Emprunt** contient N nuplets et qu'une instance de la relation **Personne** en contient M .

Indiquez quel est le nombre de nuplets résultat de chaque requête ci-dessous.

Expliquez.

Lorsque vous ne pouvez pas préciser le nombre exact de nuplets résultat, vous devez préciser une borne supérieure (ex. nombre de nuplets résultats $\leq y$ où y est une borne que vous devez définir).

- (a) $Emprunt \times Personne$
 - (b) $Emprunt \bowtie Personne$
 - (c) $\Pi_{ISBN,PID}(Emprunt) \div \Pi_{PID}(Personne)$
7. **Exprimez la requête de la question 4(a) en calcul relationnel à variable nuplet.**
Aucune explication demandée.
8. **Exprimez la requête de la question 4(b) en algèbre relationnelle.**
Aucune explication demandée.

2.8 Partiel de Mars 2019

Soit la base de données de schéma ci-dessous, permettant de gérer des locations d'appartements :

Personne(PID, Nom, Prenom, DateNaissance, Age)

Appartement(AID, Adresse, proprioID, prixParJour)

Location(AID, PID, dateDebut, dateFin)

Les attributs soulignés sont les clés primaires des relations. L'attribut **proprioID** dans la relation **Appartement** est une clé étrangère qui fait référence à la clé primaire de la relation **Personne**. L'attribut **PID** dans la relation **Location** est une clé étrangère qui fait référence à la clé primaire de la relation **Personne**. De même, l'attribut **AID** dans la relation **Location** est une clé étrangère qui fait référence à la clé primaire de la relation **Appartement**. Les attributs **PID**, **AID**, **proprioID**, **prixParJour** et **Age** sont de type entier. Les attributs **DateNaissance**, **dateDebut** et **dateFin** sont de type Date. La valeur de l'attribut **age** est calculée à partir de l'attribut **DateNaissance**. Les attributs **Nom**, **Prenom** et **Adresse**²⁰ sont de type chaîne de caractères. Aucun attribut ne peut prendre la valeur NULL sauf l'attribut **dateFin**.

Le tableau suivant est une instance de la relation **Personne** :

PID	Nom	Prenom	DateNaissance	Age
1	GAMOTTE	Albert	18/05/1973	45
2	DEBECE	Aude	20/03/1998	21
3	COMPUTING	Claude	01/07/1968	51
4	SUFFIT	Sam	21/04/1985	34

Le tableau suivant est une instance de la relation **Appartement** :

AID	Adresse	proprioID	prixParJour
1	21 rue ...	2	65
2	158 avenue ...	1	100
3	4 allée ...	2	70

Le tableau ci-dessous est une instance de la relation **Location** :

AID	PID	dateDebut	dateFin
3	3	02/03/19	10/03/19
1	3	05/04/19	10/04/19
2	4	21/07/19	25/07/19

1. Quelle(s) contrainte(s) faut-il ajouter dans la base de données sur les dates de la location d'un appartement ? (0,5 point)
2. Expliquer la différence (en donnant des exemples de nuplets éventuellement) entre choisir comme clé primaire de Location, le couple (PID,AID) ou le triplet (PID,AID,dateDebut) ? *Expliquer.* (1 point)
3. A partir des instances de la page précédente, indiquer si les insertions²¹ suivantes sont possibles ou non. *Expliquer.* (3,5 points)
 - (a) Insertion du nuplet (3,3,'03/08/2019') dans l'instance de la relation Location.

20. Pour des raisons de place, la valeur de l'attribut **Adresse** n'est pas complètement représentée.

21. On supposera que tous les domaines des attributs sont corrects.

- (b) Insertion du nuplet (4,2,'02/07/2019') dans l'instance de la relation **Location**.
- (c) Insertion du nuplet (4,'33 rue ...',5,55) dans l'instance de la relation **Appartement**.
- 4. A partir des instances de la page précédente (sans tenir compte des insertions de la question précédente), **donner le résultats des requêtes suivantes**. *Aucune explication demandée.*
(1 point)
 - (a) $\Pi_{Nom, Prenom, AID}(Personne \bowtie Location) \div \Pi_{AID}[(\sigma_{prixParJour < 90}(Appartement))]$
 - (b) $\{p.Nom, p.Prenom \mid Personne(p) \wedge \neg[\exists a \text{ Appartement}(a) \wedge (a.proprioID = p.PID)]\}$

*Pour simplifier, dans la suite, vous pouvez abréger le nom des relations et des attributs par leur initiales (par exemple P pour **Personne**, A pour **Appartement**, N pour **Nom** etc.). En algèbre relationnelle, vous pouvez également utiliser une requête précédente en nommant la requête R_i , où i est le numéro de la requête.*

- 5. **Traduire la requête 4(a) en calcul relationnel à variable nuplet.**
Aucune explication demandée. (1 point)
- 6. **Traduire la requête 4(b) en en algèbre relationnelle.***Aucune explication demandée. (1 point)*
- 7. **Exprimer, les requêtes²² suivantes en algèbre relationnelle et en calcul relationnel à variable nuplet.** *Aucune explication demandée. Attention : vos requêtes doivent fonctionner sur toutes les instances possibles de la base, pas uniquement celles de la page précédente. (11 points)*
 - (a) Quelles sont les adresses des appartements d'Albert GAMOTTE ?
 - (b) Qui (en donnant le nom et le prénom) a loué au moins 2 appartements différents ?
 - (c) Qui (en donnant le nom et le prénom) a loué un appartement d'Albert GAMOTTE OU un appartement d'Aude DEBECE ?
 - (d) Qui (en donnant le nom et le prénom) a loué un appartement d'Albert GAMOTTE ET un appartement d'Aude DEBECE ?
 - (e) Quel appartement (en donnant son adresse) n'a jamais été loué ?
 - (f) Qui (en précisant le nom et le prénom) a loué tous les appartements de la base ?
- 8. **Exprimer, la requête suivante en calcul relationnel à variable nuplet** (*Aucune explication demandée*) : Qui (en précisant le nom et le prénom) est la personne la plus âgée de la base ? (1 point)

22. Vos requêtes doivent pouvoir être exécutées sur toutes les instances possibles, pas uniquement celles du sujet.

2.9 Examen de Mai 2019

2.9.1 Exercice 1 - Algèbre relationnelle et SQL (23 points soit 57.5%)

Soit le schéma d'une base de données, décrit en SQL ci-dessous, permettant de gérer des étudiants et leur notes (des instances de ces relations sont données sur la page suivante).

```
CREATE TABLE Etudiant
( EtudiantID      serial,
  Nom              varchar(25) NOT NULL,
  Prenom          varchar(25) NOT NULL,
  Date_Naissance  date NOT NULL,
  CONSTRAINT PK_EtudiantID PRIMARY KEY (EtudiantID),
  CONSTRAINT UN_Etudiant UNIQUE (Nom,Prenom,Date_Naissance)
);

CREATE TABLE Binome
( BinomeID        serial,
  IDMembre1       int4 NOT NULL UNIQUE,
  IDMembre2       int4,
  CONSTRAINT PK_Binome PRIMARY KEY (BinomeID),
  CONSTRAINT "FK_Binome_Etudiant" FOREIGN KEY (IDMembre1) REFERENCES Etudiant (EtudiantID)
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT "FK2_Binome_Etudiant" FOREIGN KEY (IDMembre2) REFERENCES Etudiant (EtudiantID)
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT "CK_Binome_Etudiant" CHECK (IDMembre1!=IDMembre2),
  CONSTRAINT "UN_Binome_Etudiant" UNIQUE (IDMembre1,IDMembre2)
) ;

CREATE TABLE TypeEval
( TypeID          serial,
  Intitule        varchar(25) NOT NULL UNIQUE,
  CONSTRAINT PK_TypeEval PRIMARY KEY (TypeID));

CREATE TABLE ReleveNotes
( ReleveID        serial,
  EtudiantID      int4 NOT NULL,
  TypeID          int4 NOT NULL,
  Note            float NOT NULL,
  CONSTRAINT PK_Releve PRIMARY KEY (ReleveID),
  CONSTRAINT "FK_RN_Etudiant" FOREIGN KEY (EtudiantID) REFERENCES Etudiant (EtudiantID)
    ON UPDATE CASCADE ON DELETE CASCADE,
  CONSTRAINT "FK_RN_Type" FOREIGN KEY (TypeID) REFERENCES TypeEval (TypeID)
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT "UN_ReleveNotes_Etudiant" UNIQUE (EtudiantID,TypeID)
) ;
```

etudiantid integer	nom character varying (25)	prenom character varying (25)	date_naissance date
1	Gamotte	Albert	2001-01-21
2	Computing	Claude	1999-03-15
3	Zarella	Maude	2000-09-19
4	Deblouze	Agathe	2001-03-02

FIGURE 8 – Une instance de la relation Etudiant.

binomeid integer	idmembre1 integer	idmembre2 integer
2	1	3
3	2	[null]
4	4	[null]

FIGURE 9 – Une instance de la relation Binome.

typeid integer	intitule character varying (25)
1	Examen
2	TP
3	Partiel
4	CC

FIGURE 10 – Une instance de la relation TypeEval.

releveid integer	etudiantid integer	typeid integer	note double precision
1	4	1	20
3	4	3	20
4	3	1	15
5	2	1	8

FIGURE 11 – Une instance de la relation ReleveNotes.

1. **Indiquer toutes les clés minimales possibles de chaque relation.** *Aucune explication demandée.*
2. **Que vérifie la contrainte nommée CK_Binome_Etudiant ?** Expliquer en français et donner un exemple de nuplet qui ne vérifierait pas cette contrainte.
3. **Exprimer les requêtes suivantes en algèbre relationnelle (non étendue) et en SQL.**
Aucune explication demandée. Vos requêtes doivent fonctionner sur les instances des relations données en fin de sujet d'examen (en renvoyant éventuellement une table vide) mais aussi sur toutes les autres instances possibles.
Vous pourrez abréger le nom des relations par E, B, T, R, ainsi que les noms des attributs par leurs initiales (ex. EtudiantId par EID, Nom par N, etc.).
 - (a) Quelle est la note de partiel (évaluation d'intitulé 'Partiel') d'Albert Gamotte ?
 - (b) Qui (en donnant son nom et son prénom) n'a pas de note de partiel (évaluation d'intitulé 'Partiel') ?
 - (c) Qui (en donnant son nom et son prénom) a eu une note supérieure à 10 pour le partiel (évaluation d'intitulé 'Partiel') OU le TP (évaluation d'intitulé 'TP') ?
 - (d) Qui (en donnant son nom et son prénom) a eu une note supérieure à 10 pour le partiel (évaluation d'intitulé 'Partiel') ET le TP (évaluation d'intitulé 'TP') ?
 - (e) Pour quel type d'évaluation (en donnant son intitulé) y-a-t-il une note pour tous les étudiants ?
4. **Exprimer en SQL les requêtes suivantes :**
 - (a) Quelle est la moyenne par type d'évaluation (en donnant son intitulé) ?
 - (b) Qui (en donnant son nom et son prénom) a la meilleure note à l'examen ?
5. Soit le code SQL du déclencheur suivant :


```

1.  CREATE OR REPLACE FUNCTION FunctionTriggerReleveNote() RETURNS trigger AS
2.  ' DECLARE
3.      eId Etudiant.EtudiantID%TYPE;

4.  BEGIN
5.      SELECT INTO eId IDMembre2 FROM Binome WHERE (IDMembre1=NEW.EtudiantID);
6.      IF eId IS NOT NULL THEN
7.          INSERT INTO ReleveNotes(EtudiantID,TypeID,Note) VALUES (eId,2,NEW.Note);
8.      END IF;
9.      RETURN NEW;
10.  END;'
11. LANGUAGE 'plpgsql';

12. CREATE TRIGGER InsertionNoteTPNote
13.  AFTER INSERT ON ReleveNotes
14.  FOR EACH ROW
15.  WHEN (NEW.TypeID=2)
16.  EXECUTE PROCEDURE FunctionTriggerReleveNote();
      
```

 - (a) **Décrire en français (5 lignes maximum) le code SQL du déclencheur .**
Vous pouvez vous aider des numéros des lignes sans pour autant forcément toutes les décrire.
 - (b) **Expliquer ce qu'il se passe après l'exécution de la requête suivante :**
`INSERT INTO ReleveNotes(EtudiantID,TypeID,Note) VALUES(1,2,18).`

2.9.2 Exercice 2 - Dépendances fonctionnelles, formes normales et décomposition (11 points soit 27.5%)

On souhaite créer une base de données permettant de gérer des stations de métro. Pour cela, la relation R suivante a été créée : R (IdStation, NumeroLigne, NomStationDepart, NomStationArrivee, NomStation, PositionDepuisDepart).

Le nuplet $(1, 2, 'PorteDauphine', 'Nation', 'Ternes', 4)$ a pour signification que la ligne numéro 2, qui a pour station de départ 'Porte Dauphine' et station d'arrivée 'Nation', a la station nommée 'Ternes' en position 4 depuis la station de départ. Le nuplet $(2, 2, 'Nation', 'PorteDauphine', 'Ternes', 22)$ signifie que la station 'Ternes' est en position 22 sur la ligne 2 quand la station de départ est 'Nation'.

Soit F , la famille de dépendances fonctionnelles associée à R :

```
F = { IdStation → NumeroLigne, NomStationDepart, NomStationArrivee, NomStation, PositionDepuisDepart ;
      NumeroLigne, NomStationDepart → NomStationArrivee ;
      NumeroLigne, NomStationArrivee → NomStationDepart ;
      NumeroLigne, NomStationDepart, NomStationArrivee, NomStation → PositionDepuisDepart, IdStation ;
    }
```

Dans la suite de l'exercice, vous pourrez abréger le nom des attributs par leurs initiales : IS pour IdStation, NL pour NumeroLigne, NSD pour NomStationDepart, NSA pour NomStationArrivee, NS pour NomStation, et PDD pour PositionDepuisDepart.

A chaque réponse, vous donnerez des explications claires et concises.

1. Quelle(s) est(sont) la(les) clé(s) minimale(s) de la relation R ? Il peut n'y avoir qu'une seule clé minimale.
2. En quelle forme normale est la relation R ? (A défaut, indiquer au minimum en quelle forme normale elle n'est pas).
3. La famille F est-elle minimale ? Si F est minimale, expliquez. Si elle ne l'est pas, trouver une famille minimale équivalente.
4. Proposer une décomposition sans perte d'information et sans perte de dépendances de la relation R telle que toutes les relations issues de la décomposition soient en forme normale BCNF et telle qu'il y ait le moins de relations possibles dans la décomposition. (Vous préciserez bien quel algorithme de décomposition vous avez utilisé.)

2.9.3 Exercice 3 - Requêtes en algèbre relationnelle (6 points soit 15%)

Soient les deux relations suivantes :

- $R(A, B, C, D)$ d'instance r contenant N nuplets, de clé minimale (A, B, C)
- $S(A, B, C)$ d'instance s contenant M nuplets, de clé minimale (A, B, C) . (Les attributs A, B et C de S sont de même domaine que les attributs de même nom dans R).

Pour chacune des requêtes ci-dessous, numérotées de (a) à (c) :

- (a) $r \times s$
- (b) $r \bowtie s$
- (c) $r \div s$

Indiquer le nombre de nuplets résultat. *Expliquer.* Lorsque vous ne pouvez pas préciser le nombre exact de nuplets résultat, vous devez préciser une borne supérieure (ex. nombre de nuplets résultats $\leq y$ où y est une borne que vous devez définir).

2.10 Examen de Janvier 2019

2.10.1 Exercice 1 - Passage au relationnel (13 points soit 32,5%)

Les 2 figures qui suivent représentent la modélisation entité-association (figure 12) et UML (figure 13) d'une base de données gérant les affectations d'avions et d'équipages, d'une compagnie aérienne, sur ses différents vols.

Un équipage et un avion sont affectés à chaque vol. Un vol est caractérisé par sa date, son heure de début, sa durée, mais également son trajet (allant d'un terminal d'aéroport de départ à un terminal d'aéroport d'arrivée). A un aéroport, il est possible d'associer plusieurs terminaux. Un équipage est constitué d'un pilote, d'un copilote, et de plusieurs personnels de cabine. La compagnie dispose d'une flotte d'avions, décrits par leur type, la taille de l'équipage devant leur être affecté, et leur autonomie de vol (durée de vol maximale que l'avion peut couvrir sans ravitaillement). Un même avion peut être affecté à plusieurs équipages et un même équipage peut être affecté à plusieurs avions. Un membre d'équipage est identifié par son nom, son prénom et sa date de naissance. Il ne peut pas exister 2 aéroports de même nom. Le nom d'un terminal est relatif à son aéroport de rattachement (par exemple il y a un "Terminal 1" à l'aéroport Paris-CDG et un terminal de même nom à l'aéroport JFK).

Chaque ensemble d'entités ou classe possède un attribut identificateur (*AeroportID*, *TerminalID*, *Mem-breEquipageID* etc.) de type entier, qui n'est pas représenté sur les schémas.

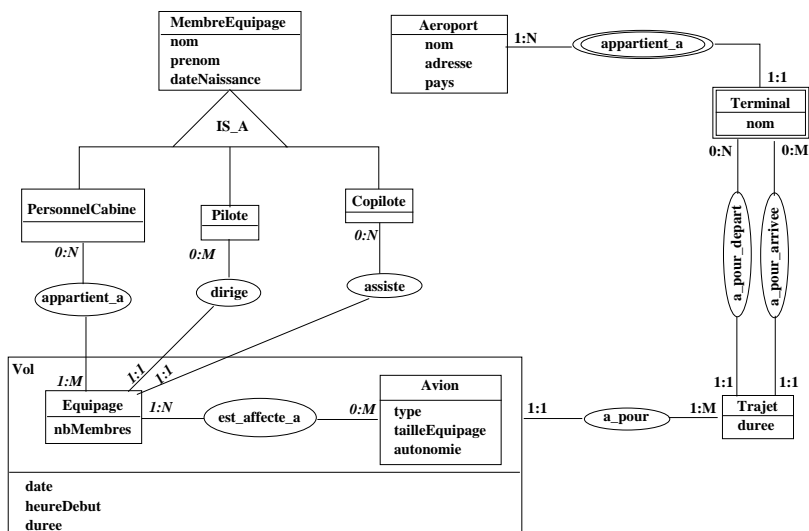


FIGURE 12 – Modélisation Entité-Association de l'exercice 1.

Déduisez le schéma relationnel de la base de données correspondante.

Vous préciserez les clés primaires (artificielles ou non) des relations en les soulignant ainsi que les clés étrangères en les signalant par un # et en précisant à quoi elles font référence. Vous préciserez également les contraintes d'unicité et si les attributs peuvent ou non prendre la valeur NULL.

Aucune explication n'est demandée.

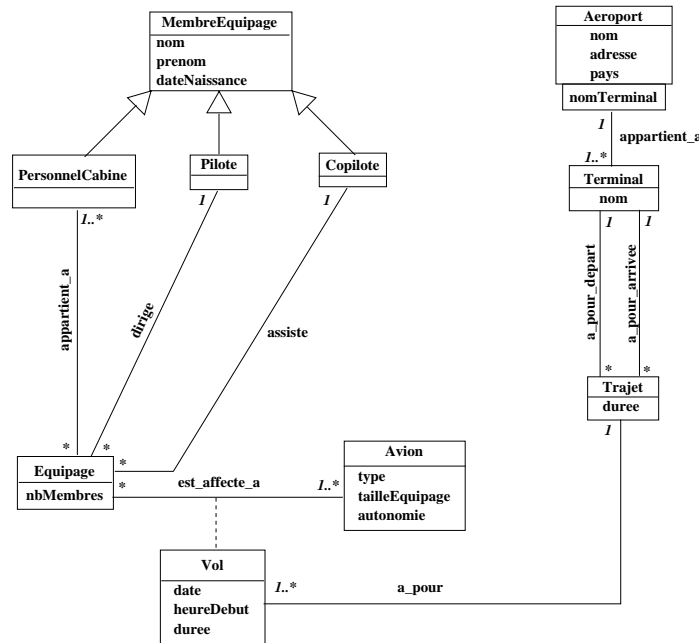


FIGURE 13 – Modélisation UML de l'exercice 1.

Dans votre schéma relationnel, chaque relation doit être spécifiée de la manière suivante :

$Nom(att_1, \dots, att_n)$ où Nom est le nom de la relation et att_1, \dots, att_n sont des noms d'attributs. Le nom de la relation doit obligatoirement avoir un lien avec les noms des ensembles d'entités (classes) ou des associations du schéma de modélisation.

2.10.2 Exercice 2 - Algèbre relationnelle, calcul relationnel à variable nuplet et SQL (19 points soit 47,5%)

Soit le schéma d'une base de données, décrit en SQL ci-dessous, permettant de gérer des commandes de produits.

Dans la suite, vous pourrez abréger le nom des relations par Cl , P , Com , D ainsi que les noms des attributs par leurs initiales (ex. Nom par N , $Prenom$ par P , etc.).

```
CREATE TABLE client
( cid SERIAL,
  nom varchar(20) NOT NULL,
  prenom varchar(20) NOT NULL,
  adresse varchar(40) NOT NULL,
  ville varchar(20) NOT NULL,
  CodePostal varchar(5) NOT NULL,
  categorie varchar(2),
  CONSTRAINT PK_Client PRIMARY KEY (cid),
  CONSTRAINT CK_Client_Unique UNIQUE (nom,prenom,adresse,ville)
);
```

```

CREATE TABLE produit
( pid SERIAL ,
  libelle varchar(20) NOT NULL,
  prix integer NOT NULL,
  stock integer NOT NULL,
  CONSTRAINT PK_Produit PRIMARY KEY (pid),
  CONSTRAINT CK_Libelle_Unique UNIQUE (libelle)
);

CREATE TABLE commande
( comId SERIAL,
  cid integer NOT NULL,
  dateC date NOT NULL,
  CONSTRAINT PK_commande PRIMARY KEY (comId),
  CONSTRAINT "FK_Commande_Client" FOREIGN KEY (cid) REFERENCES client (cid)
    ON UPDATE RESTRICT ON DELETE RESTRICT
);

CREATE TABLE detailsCommande
( comId integer NOT NULL,
  pid integer NOT NULL,
  quantité integer NOT NULL,
  CONSTRAINT PK_detailscommande PRIMARY KEY (commId,pid),
  CONSTRAINT "FK_detailsCommande_Commande" FOREIGN KEY (commId) REFERENCES commande (commId)
    ON UPDATE RESTRICT ON DELETE RESTRICT
  CONSTRAINT "FK_detailsCommande_Produit" FOREIGN KEY (pid) REFERENCES produit (pid)
    ON UPDATE RESTRICT ON DELETE RESTRICT
);

```

1. **Exprimer les requêtes suivantes en algèbre relationnelle (non étendue) et en SQL (pas en calcul relationnel!).** *Aucune explication demandée.*
 - (a) Quels sont les produits (en donnant leur libellé) apparaissant dans au moins 2 commandes ?
 - (b) Quels sont les produits (en donnant leur libellé) jamais commandés ?
 - (c) Quels sont les produits (en donnant leur libellé) commandés par des clients habitant 'Paris' OU par des clients habitant 'Marseille' ?
 - (d) Quels sont les produits (en donnant leur libellé) apparaissant dans toutes les commandes ?
 - (e) Dans quelles villes a-t-on commandé tous les produits dont le prix est inférieur à 100 euros ?
2. **Exprimer la requête suivante en calcul relationnel à variable nuplet et en SQL** (*Aucune explication demandée*) : Quels sont les produits (en donnant leur libellé) les plus chers ?
3. **Écrire en SQL une vue, TotalCommande, permettant de calculer le prix total de chaque commande (i.e la somme des prix de tous les produits apparaissant dans chaque commande) ?** *Aucune explication demandée.*

4. En supposant que la vue précédente existe, et que la requête `SELECT * FROM TotalCommande;` a pour résultat la Table ci-dessous :

comId	PrixTotal
1	120
2	1400
3	88
4	79

Pour chacune des requêtes suivantes : (a) Donner le résultat de la requête et (b) exprimer la requête en calcul relationnel à variable nuplet. *Aucune explication demandée*

- (i) `SELECT comId FROM TotalCommande WHERE PrixTotal >= ALL(SELECT PrixTotal FROM TotalCommande);`
(ii) `SELECT comId FROM TotalCommande WHERE PrixTotal > ANY(SELECT PrixTotal FROM TotalCommande);`
5. Si on suppose que la relation `commande` contient N nuplets et que la relation `client` contient M clients, **quel est le nombre de nuplets résultat de chacune des requêtes suivantes :**
- (a) `SELECT * FROM client, commande`
(b) `SELECT * FROM client, commande WHERE client.cid=commande.cid`

Expliquez.

2.10.3 Exercice 3 - Dépendances fonctionnelles, formes normales et décomposition (8 points soit 20%)

Soient la relation ²³ `Cine`(Film, , Ville, Salle, Producteur, Delege) et la famille F , contenant les 3 dépendances fonctionnelles associées à la relation R , suivante :

`Salle` \longrightarrow `Ville`

`Film, Ville` \longrightarrow `Salle, Producteur`

`Producteur` \longrightarrow `Delege`

Pour simplifier vous pouvez abréger le nom des attributs par leurs initiales (par exemple F pour `Film`, V pour `Ville`, etc.).

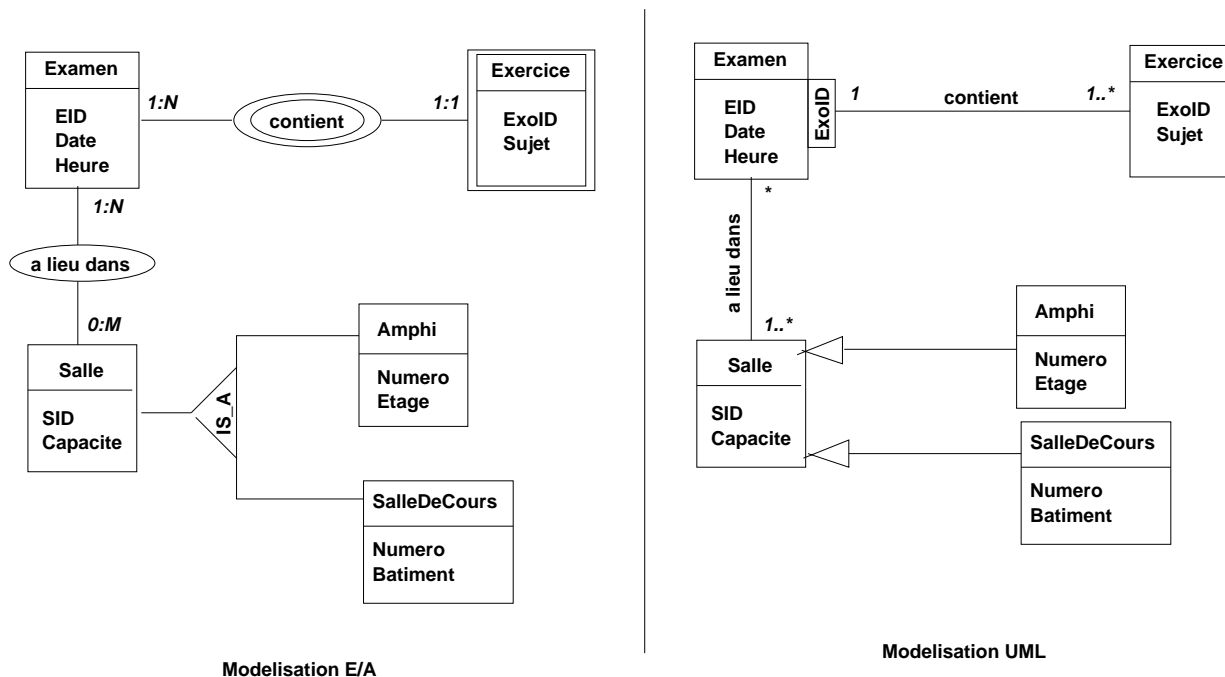
1. **Quelle(s) est (sont) la (les) clé(s) minimale(s) de la relation `Cine`? *Aucune explication demandée.***
2. **En quelle forme normale est le schéma de la relation `Cine`? Expliquez.**
3. On choisit de décomposer la relation `Cine` de l'énoncé en 3 relations :
`R1`(Salle, Film, Ville)
`R2`(Film, Ville, Producteur)
`R3`(Producteur, Delege)
 - (a) **En quelle forme normale est le schéma de chaque relation? Expliquez.**
 - (b) **La décomposition proposée est-elle sans perte d'information? Expliquez.**
 - (c) **La décomposition proposée est-elle sans perte de dépendance? Expliquez.**

23. Exercice adapté du livre de Jean-Luc Hainaut, Bases de données - 4e édition Concepts, utilisation et développement, InfoSup, Dunod, 2018.

2.11 Examen de Mars 2018

2.11.1 Exercice 1 - Passage au relationnel (20 points soit 50%)

On souhaite définir une base de données permettant de gérer des examens. La modélisation Entité/Association et celle en UML sont représentées sur la figure ci-dessus. Chaque examen est composé d'énoncés d'exercices. Un examen peut avoir lieu dans une ou plusieurs salles (en fonction des effectifs des étudiants devant passer l'examen). Une salle peut être soit un amphi soit une salle de cours. Les identifiants sont les attributs dont le nom se terminent par *ID*.



Déduisez le schéma relationnel de la base de données correspondante.

Vous préciserez les clés primaires (artificielles ou non) des relations en les soulignant ainsi que les clés étrangères en les signalant par un # et en précisant à quoi elles font référence. Vous préciserez également les contraintes d'unicité et si les attributs peuvent ou non prendre la valeur NULL. *Aucune explication n'est demandée.*

Dans votre schéma relationnel, chaque relation doit être spécifiée de la manière suivante :

$Nom(\underline{att_1}, \dots, att_n)$ où Nom est le nom de la relation et att_1, \dots, att_n sont des noms d'attributs. Le nom de la relation doit obligatoirement avoir un lien avec les noms des ensembles d'entités (classes) ou des associations du schéma de modélisation.

2.11.2 Exercice 2 - Modèle relationnel et requêtes en algèbre relationnelle et calcul relationnel à variable nuplet (20 points soit 50%)

Soit la base²⁴ de données de schéma ci-dessous, permettant de gérer des articles et leurs fournisseurs :

Articles (NoArt, Libelle, Stock, PrixInvent) de clé primaire NoArt.

Fournisseurs (NoFour, NomFour, AdrFour, VilleFour) de clé primaire NoFour.

Ventes(NoFour, NoArt, PrixAchat, Delai) de clé primaire (NoFour, NoArt), de clé étrangère NoFour faisant référence à la clé primaire de Fournisseurs et de clé étrangère NoArt faisant référence à la clé primaire de Articles.

Seuls les attributs AdrFour et VilleFour peuvent prendre les valeurs NULL. Des exemples d'instances sont données dans les figures 8 à 11.

NB : Les questions qui suivent sont indépendantes les unes des autres.

1. A partir des instances de relations présentées dans les figures 14 à 16 en fin de sujet :
 - (a) **Indiquez si les insertions suivantes sont possibles.** On supposera que les valeurs appartiennent bien aux domaines des attributs. Expliquez.
 - i. Insertion du nuplet (14,109,24.90,12) dans la relation Ventes.
 - ii. Insertion du nuplet (14,108,10,24) dans la relation Ventes.
 - iii. Insertion du nuplet (14,107,50.90,5) dans la relation Ventes.
 - (b) Quelle contrainte faut-il ajouter pour qu'il ne soit pas possible d'insérer le nuplet suivant dans la relation Fournisseurs : (20, 'PIMS', '50 BLD DES ALOUETTES', 'GRENOBLE')
 - (c) **Calculez le résultats des requêtes suivantes sur les instances situées en fin de sujet** (Aucune explication demandée) :
 - i. $\Pi_{NoArt}(Articles) - \Pi_{NoArt}(Ventes)$
 - ii. $\{a_1.Libelle \mid Articles(a_1) \wedge \neg(\exists a_2 Articles(a_2) \wedge (a_2.PrixInvent > a_1.PrixInvent))\}$
2. Dans la suite de l'exercice, vous pouvez utiliser des abréviations pour le noms des relations et des attributs. Par exemple A pour Articles, F pour Fournisseurs, V pour Ventes, NA pour NoArt, etc.

Écrivez les requêtes suivantes en algèbre relationnelle et en calcul relationnel à variable nuplet.
Aucune explication demandée.

- (a) Quels sont les numéros et libellés des articles dont le stock est inférieur à 10 ?
- (b) Quels sont les noms des fournisseurs fournissant au moins 2 articles ?
- (c) Quels sont les libellés des articles vendus par les fournisseurs 10 OU 11 ?
- (d) Quels sont les libellés des articles vendus par les fournisseurs 15 ET 12 ?
- (e) Quels sont les libellés des articles vendus par aucun fournisseur ?
- (f) Quels sont les libellés des articles vendus par tous les fournisseurs ?
- (g) Quels sont les libellés des articles toujours vendus au même prix ?

24. Exercice repris et adapté de <http://webtic.free.fr/sql/exint/q1.htm>

NOART	LIBELLE	STOCK	PRIXINVENT
100	HP Deskjet 930C	10	181.41
101	Scanner Epson Perfection	12	150.92
102	Zip 250Mo USB	8	272.88
103	App Photo Numerique	5	836.95
104	Fax Modem V92	20	75.00
105	Cam Sony DCR	4	2230.66
106	Cable USB	80	8.38
107	Hub 4 ports USB	35	48.78
108	Papier couche 100 A4	110	12.00

FIGURE 14 – Une instance de la relation **Articles**.

NOFOUR	NOMFOUR	ADRFOUR	VILLEFOUR
10	STE LIGER	13 RUE DES ALLIES	PARIS
11	ORKIS	22 BLD DES BELGES	RENNES
12	STE IMPEC	3 BIS IMPASSE DES CLERCS	MARSEILLE
13	STE LE DUC	101 AVENUE MARECHAL FOCH	BORDEAUX
14	BUROTIC		
15	KBSS	42 COURS VITTON	LYON
16	COOP ACHAT	54 RUE DE LA PAIX	NIORT
17	PRINT42	32 AVENUE DES PLATANES	VALENCE
18	STE PROTEC	92 RUE DE LA REPUBLIQUE	TOULOUSE
19	PIMS	50 BLD DES ALOUETTES	GRENOBLE

FIGURE 15 – Une instance de la relation **Fournisseurs**.

NOFOUR	NOART	PRIXACHAT	DELAI
10	104	76.26	14
11	105	2199.00	2
11	108	11.50	5
12	100	185.20	40
12	106	8.38	20
13	103	835.00	23
14	102	260.50	3
14	108	12.00	4
15	100	180.00	23
15	107	45.47	2
16	101	150.20	17

FIGURE 16 – Une instance de la relation **Ventes**.

2.12 Examen de Mai 2018

2.12.1 Exercice 1 - Algèbre relationnelle, calcul relationnel à variable nuplet et SQL (30 points soit 60%)

Soit le schéma d'une base de données, décrit en SQL ci-dessous, permettant de gérer des inscriptions d'enfants à des ateliers d'informatique.

1. **Décrire le schéma de la base de données dont le script SQL est ci-dessous** (des instances de ces relations sont données dans les figures 17 à 19).

Vous préciserez bien pour chaque relation, son schéma (i.e. nom de la relation, nom des attributs, clé primaire et clé(s) étrangère(s)), et à quoi fait(font) référence la(es) clé(s) étrangère(s). Aucune explication demandée.

Vous pourrez abréger le nom des relations par A, E, I, ainsi que les noms des attributs par leurs initiales (ex. atelierId par AID, nom par N, etc.).

```
CREATE TABLE atelier
( atelierId serial NOT NULL,
  nom varchar(20) NOT NULL UNIQUE,
  description varchar(100) NOT NULL,
  ageMin int NOT NULL,
  ageMax int NOT NULL,
  CONSTRAINT pk_atelier PRIMARY KEY (atelierId),
  CONSTRAINT ck_atelier CHECK (ageMin<ageMax)
);

CREATE TABLE enfant
( enfantId serial NOT NULL,
  nom varchar(20) NOT NULL,
  prenom varchar(10) NOT NULL,
  age int NOT NULL,
  CONSTRAINT pk_enfant PRIMARY KEY (enfantId),
  CONSTRAINT un_enfant UNIQUE (nom,prenom,age)
);

CREATE TABLE inscription
( enfantId int NOT NULL,
  atelierId int NOT NULL,
  CONSTRAINT pk_inscription PRIMARY KEY (enfantId, atelierId),
  CONSTRAINT fk_inscription_enfant FOREIGN KEY (enfantId)
    REFERENCES enfant (enfantId) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT fk_inscription_atelier FOREIGN KEY (atelierId)
    REFERENCES atelier (atelierId) MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT
);
```

atelierid [PK] integer	nom character varying (20)	description character varying (100)	agemin integer	agemax integer
1	Scratch Junior	Ecriture d'histoire sur tablette	4	7
2	Scratch	Initiation à la programmation	7	10
3	Python	Programmation	10	99

FIGURE 17 – Une instance de la relation atelier.

enfantid [PK] integer	nom character varying (20)	prenom character varying (10)	age integer
1	GAMOTTE	Albert	5
2	PABIEN	Yvon	8
3	SUFFIT	Sam	11

FIGURE 18 – Une instance de la relation enfant.

enfantid [PK] integer	atelierid [PK] integer
1	1
2	2
3	3

FIGURE 19 – Une instance de la relation inscription.

2. **Que se passe-t-il si on exécute les requêtes suivantes sur les instances données en fin de sujet ? Expliquez (chaque requête est indépendante).**
 - (a) INSERT INTO atelier VALUES (4, 'BD', 'Initiation aux bases de données',99,10);
 - (b) INSERT INTO enfant VALUES (4, 'GAMOTTE', 'Albert',5);
 - (c) INSERT INTO inscription VALUES (4,3);
3. **Exprimer les requêtes suivantes en algèbre relationnelle (non étendue), en calcul relationnel à variable nuplet et en SQL. Aucune explication demandée. Vos requêtes doivent fonctionner sur les instances des relations données précédemment (en revoyant éventuellement une table vide) mais aussi sur toutes les autres instances possibles.**
 - (a) Quels enfants (en donnant leur nom et leur prénom) sont inscrits à l'atelier numéro 1 OU à l'atelier numéro 2?
 - (b) Quels enfants (en donnant leur nom et leur prénom) sont inscrits à l'atelier numéro 1 ET à l'atelier numéro 2?
 - (c) Quels enfants (en donnant leur nom et leur prénom) ne sont inscrits à aucun atelier?
 - (d) Quels enfants (en donnant leur nom et leur prénom) sont inscrits à tous les ateliers?
4. **Créer une vue en SQL permettant d'afficher, pour chaque atelier, le nombre d'enfants inscrits à chaque atelier (en précisant le nom de l'atelier). Aucune explication demandée.**
5. **Exprimer en calcul relationnel à variable nuplet et en SQL la requête suivante :**
 Quels sont les ateliers (en donnant leur nom) ayant le moins d'inscrits ?
 Vous pouvez utiliser ou supposer existante la vue de la question précédente. Aucune explication demandée.

6. Soit le code SQL du déclencheur suivant :

```
1. CREATE OR REPLACE FUNCTION FunctionTriggerInscription() RETURNS trigger AS
2. ' DECLARE
3.     id atelier.atelierId%TYPE;
4. BEGIN
5.     SELECT INTO id atelierId FROM atelier WHERE NEW.age BETWEEN ageMin AND ageMax;
6.     IF FOUND THEN
7.         INSERT INTO inscription VALUES (New.enfant_id,id);
8.     END IF;
9.     RETURN NEW;
10. END;'
11. LANGUAGE 'plpgsql';

12. CREATE TRIGGER InsertionAtelier
13. AFTER INSERT ON enfant
14. FOR EACH ROW
15. EXECUTE PROCEDURE FunctionTriggerInscription() ;
```

(a) Décrire en français (5 lignes maximum) le code SQL du déclencheur suivant et expliquez ce qu'il se passe après l'insertion du nuplet (4,'SLATABLE','Deborah',9) dans la relation enfant. *Expliquez.*

Vous pouvez vous aidez des numéros des lignes sans pour autant forcément toutes les décrire.

(b) Indiquer ce qui se passe lorsque l'on remplace la ligne 13 par BEFORE INSERT ON enfant. *Expliquez.*

7. Le schéma de la relation atelier doit vérifier la dépendance fonctionnelle suivante : $nom \rightarrow atelierId$.

Indiquez ce que cette dépendance signifie en français et quelle(s) contrainte(s) en SQL il faudrait ajouter pour qu'elle soit vérifiée.

Pour rappel, des contraintes sont définies dans le script SQL de création de la base de la question

1. Une contrainte peut également être exprimée par une assertion, de la manière suivante :

```
CREATE ASSERTION Nom_Assertion
BEFORE INSERT INTO Nom_Relation
CHECK (Condition) ;
```

2.12.2 Exercice 2 - Dépendances fonctionnelles, formes normales et décomposition (10 points soit 20%)

On souhaite créer une base de données permettant de gérer des votes à une élection. Pour cela, la relation R suivante a été créée : R (NomCandidat, PrenomCandidat, NomParti, NomElecteur, PrenomElecteur, NumBureauVote, AdresseBureauVote, NbVoix, DateScrutin, AVoté).

Un nuplet $(nc, pc, np, ne, pe, nbv, abv, nv, ds, av)$ a pour signification que l'électeur de nom ne et prénom pe a voté ($av = True$) dans le bureau de vote numéro nbv , situé à l'adresse abv , à la date de scrutin ds , et que, dans ce bureau de vote, le candidat de nom nc et prénom pc , du parti np , a obtenu nbv voix. Lorsque l'électeur de nom ne et prénom pe , devant voter dans le bureau de vote numéro nbv , situé à l'adresse abv , n'a pas voté à la date de scrutin ds , alors $av = False$.

Soit F , la famille de dépendances fonctionnelles associée à R :

```
F = { NomElecteur, PrenomElecteur → NumBureauVote, AdresseBureauVote, AVoté, DateScrutin;  
      NomCandidat, PrenomCandidat → NomParti;  
      NumBureauVote → AdresseBureauVote;  
      NomCandidat, PrenomCandidat, NomParti, NumBureauVote, AdresseBureauVote, DateScrutin → NbVoix  
    }
```

A chaque réponse, vous donnerez des explications claires et concises.

1. **La famille F permet-elle de déduire pour quel candidat a voté chaque électeur, i.e. la dépendance fonctionnelle**
 $\text{NomElecteur, PrenomElecteur, DateScrutin} \rightarrow \text{NomCandidat, PrenomCandidat}$
est-elle vérifiée ? Si vous pensez que cette dépendance est vérifiée, prouvez-le. Si vous pensez que cette dépendance n'est pas vérifiée, donner un contre-exemple.
2. **Quelle(s) est(sont) la(les) clé(s) minimale(s) de la relation R ?** Il peut n'y avoir qu'une seule clé minimale.
3. **En quelle forme normale est la relation R ?** (A défaut, indiquer au minimum en quelle forme normale elle n'est pas).
4. **La famille F est-elle minimale ?** Si F est minimale, expliquez. Si elle ne l'est pas, trouver une famille minimale équivalente.
5. **Proposer une décomposition sans perte d'information et sans perte de dépendances de la relation R telle que toutes les relations issues de la décomposition soient en forme normale BCNF et telle qu'il y ait le moins de relations possibles dans la décomposition.** (Vous préciserez bien quel algorithme de décomposition vous avez utilisé.)

2.12.3 Exercice 3 - Requêtes en algèbre relationnelle (10 points soit 20%)

Soient les deux relations suivantes :

- $R(A,B,C,D)$ d'instance r contenant N nuplets, de clé minimale (A,B,C)
- $S(A,B,C)$ d'instance s contenant M nuplets, de clé minimale (A,B,C) . (Les attributs A, B et C de S sont de même domaine que les attributs de même nom dans R).

Pour chacune des requêtes ci-dessous, numérotées de (a) à (c) :

- (a) $r \times s$
- (b) $r \bowtie s$
- (c) $r \div s$

1. **Indiquer le nombre de nuplets résultat.** Expliquez. Lorsque vous ne pouvez pas préciser le nombre exact de nuplets résultat, vous devez préciser une borne supérieure (ex. nombre de nuplets résultats $\leq y$ où y est une borne que vous devez définir).
2. Indiquer quelle est la clé minimale associée à la relation résultat de la requête. Expliquez.

2.13 Examen de Juin 2017

2.13.1 Exercice 1 - Modèle relationnel et requêtes en algèbre relationnelle, en calcul relationnel à variable nuplet et en SQL (25 points soit 62.5%)

Soit le schéma d'une base de données suivant, permettant de gérer des joueurs et leurs convocations à des matches.

```
CREATE TABLE Match
( MID integer,
  DateMatch Date DEFAULT CURRENT_DATE,
  Heure integer NOT NULL,
  Lieu varchar(100) NOT NULL,
  CONSTRAINT PK_Match PRIMARY KEY (MID)
);
```

```
CREATE TABLE Joueur
( JID integer,
  Nom varchar(25) NOT NULL,
  Prenom varchar(25) NOT NULL,
  AnneeNaissance int NOT NULL,
  CONSTRAINT PK_Joueur PRIMARY KEY (JID)
);
```

```
CREATE TABLE Convocation
( MID integer,
  JID integer,
  CONSTRAINT PK_Convocation PRIMARY KEY (MID,JID),
  CONSTRAINT "FK_Convocation_MID" FOREIGN KEY (MID) REFERENCES Match (MID)
    ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT "FK_Convocation_JID" FOREIGN KEY (JID) REFERENCES Joueur (JID)
    ON UPDATE RESTRICT ON DELETE RESTRICT
);
```

Des exemples d'instances des relations sont donnés sur la dernière page du sujet. *Vous pourrez abréger le nom des relations par M, J, C, ainsi que les noms des attributs par leurs initiales (ex. Nom par N, AnneeNaissance par AN, etc.).*

1. A partir des instances données à la fin du sujet :

- Peut-on insérer le nuplet ('1','1') dans la relation **Convocation**? *Expliquez.*
- Peut-on insérer le nuplet ('3','1') dans la relation **Convocation**? *Expliquez.*
- Quelle contrainte faudrait-il ajouter, dans la relation **Joueur**, afin que le nuplet ('3','Computing','Claude','2005') ne puisse pas être inséré? (i.e. afin qu'on ne puisse pas insérer 2 fois le même joueur) *Expliquez.*

2. En supposant que la relation **Match** contient m nuplets, la relation **Joueur**, j nuplets, et la relation **Convocation**, c nuplets. **Pour chacune des requêtes ci-dessous, numérotées de (a) à (d) : indiquer le nombre de nuplets résultat. Expliquez.** Lorsque vous ne pouvez pas préciser le nombre exact de nuplets résultat, vous devez préciser une borne supérieure (ex. nombre de nuplets résultats $\leq y$ où y est une borne que vous devez définir).

(a) $Convocation \bowtie Joueur$

(b) $\Pi_{JID}(Joueur) - \Pi_{JID}(Convocation)$

(c) $Convocation \div \Pi_{MID}(Match)$

Dans les questions qui suivent, vos requêtes doivent fonctionner sur les instances des relations données en fin de sujet d'examen mais aussi sur toutes les autres instances possibles.

3. **Exprimer les requêtes suivantes en algèbre relationnelle, en calcul relationnel à variable nuplet et en SQL.** Aucune explication demandée.

(a) Quels joueurs (en précisant leur nom et prénom) n'ont été convoqués à aucun match ?

(b) Quels joueurs (en précisant leur nom et prénom) ont été convoqués à au moins un match ?

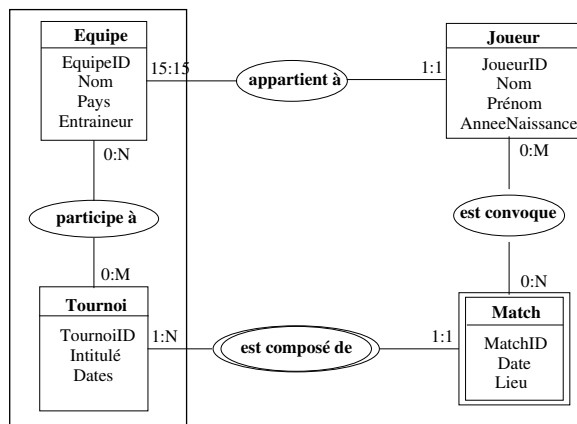
(c) Quels joueurs (en précisant leur nom et prénom) ont été convoqués à un match à Boulogne ET à un match à Asnières ? (Boulogne et Asnières sont des villes)

(d) Quels joueurs (en précisant leur nom et prénom) ont été convoqués à un match à Boulogne OU à un match à Asnières ?

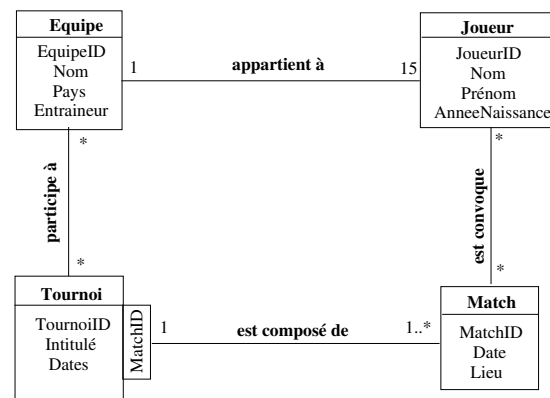
(e) Quels joueurs (en précisant leur nom et prénom) ont été convoqués à tous les matchs ?

4. **Exprimer en calcul relationnel à variable nuplet et en SQL, la requête suivante :** "Quels joueurs (en précisant leur nom et prénom) sont les plus jeunes (i.e. ont la plus grande année de naissance) ?". Aucune explication demandée.

5. **Compléter et modifier le modèle relationnel de la base de données, de telle sorte qu'il corresponde au schéma de modélisation ci-dessous. Expliquez.**



Modélisation Entité/Association



Modélisation UML

TABLE 31 – Une instance de la relation **Match** de l'exercice 1

MID	DateMatch	Heure	Lieu
1	"21/01/17"	17h	Boulogne
2	"22/01/17"	15h	Asnières

TABLE 32 – Une instance de la relation **Joueur** de l'exercice 1

JID	Nom	Prenom	AnneeNaissance
1	Comptuting	Claude	2005
2	Debece	Gilles	2008

TABLE 33 – Une instance de la relation **Convocation** de l'exercice 1

MID	JID
1	2
2	2

2.13.2 Exercice 2 - Dépendances fonctionnelles et décomposition (15 points soit 27.5%)

Soit $R(A, B, C, D)$ une relation à laquelle est associée $F = \{AC \rightarrow D; B \rightarrow D; A \rightarrow B; D \rightarrow A\}$.

A chaque réponse, vous donnerez des explications claires et concises.

- Quelle(s) est(sont) le(les) clé(s) minimale(s) de la relation R ? *Expliquez.*
- En quelle forme normale est le schéma de la relation R ? *Expliquez.*
- Parmi les familles de dépendances fonctionnelles suivantes, quelle(s) est(sont) la(les) famille(s) équivalente(s) à F . *Expliquez.*
 - $F_1 = \{AC \rightarrow D; A \rightarrow B; D \rightarrow A\}$
 - $F_2 = \{B \rightarrow D; A \rightarrow B; D \rightarrow A\}$
 - $F_3 = \{B \rightarrow D; B \rightarrow A; D \rightarrow A\}$
 - $F_4 = \{D \rightarrow B; B \rightarrow A; A \rightarrow D\}$
 - $F_5 = \{D \rightarrow B; A \rightarrow B; A \rightarrow D\}$
- Proposer une décomposition sans perte de dépendance et sans perte d'information de R , qui contienne le moins de relations possible, et dont le schéma de chaque relation issue de la décomposition soit BCNF.
Expliquez.

2.14 Partiel d'Octobre 2018

Soit la base de données de schéma ci-dessous, permettant de gérer des personnes sur un réseau social et les liens entre ces personnes :

Personne(PID, Nom, Prenom, AnneeNaissance, Pseudo)

Amis(P1ID, P2ID)

Les attributs soulignés sont les clés primaires des relations. L'attribut P1ID dans la relation Amis est une clé étrangère qui fait référence à la clé primaire de la relation Personne. De même, l'attribut P2ID dans la relation Amis est une clé étrangère qui fait référence à la clé primaire de la relation Personne. Les attributs PID, P1ID et P2ID sont de type entier. L'attribut AnnéeNaissance est de type date. Les attributs Nom, Prenom et Pseudo sont de type chaîne de caractères.

Le tableau suivant est une instance de la relation Personne :

PID	Nom	Prenom	AnneeNaissance	Pseudo
1	GAMOTTE	Albert	1992	gam
2	DEBECE	Aude	1991	audeD
3	COMPUTING	Claude	1968	ccomp
4	SUFFIT	Sam	1985	sam
5	PABIEN	Yvon	1975	yvon

Le tableau ci-dessous est une instance de la relation Amis :

P1ID	P2ID
1	2
2	1
2	4
4	2
1	4
4	1

Un nuplet (x, y) dans la relation Amis signifie que la personne identifiée par x est ami(e) avec la personne identifiée par y sur le réseau social. Pour simplifier les requêtes, lorsque deux personnes, respectivement identifiées par x et y , sont ami(e)s sur le réseau social, 2 nuplets, (x, y) et (y, x), sont insérés dans la relation Amis.

1. Quelle(s) contrainte(s) faut-il ajouter dans la base de données pour interdire :

- (a) La création d'un pseudonyme déjà existant? *Expliquer.*
- (b) Une relation d'amitié d'une personne avec elle même? *Expliquer.*

2. Indiquer si les insertions²⁵ suivantes sont possibles ou non. Expliquer.

- (a) Insertion du nuplet (1,6) dans l'instance de la relation Amis.

25. On supposera que tous les domaines des attributs sont corrects.

- (b) Insertion du nuplet (1,2) dans l'instance de la relation **Amis**.
 - (c) Insertion du nuplet (1,3) dans l'instance de la relation **Amis**.
3. A partir des instances de la page précédente (sans tenir compte des insertions de la question précédente), **donner le résultats des requêtes suivantes**. *Aucune explication demandée.*

(a) $\Pi_{Nom, Prenom}(Personne \bowtie [\Pi_{PID}(Personne) - \Pi_{PID}(Amis)])$

(b) $\{p1.Nom, p1.Prenom \mid Personne(p1) \wedge [\forall p2 \text{ } Personne(p2) \wedge (p2.AneeNaissance > 1990) \wedge (p1.PID \neq p2.PID) \implies (\exists a \text{ } Amis(a) \wedge (a.P1ID = p.PID) \wedge (a.P2Id = p2.PID))]\}$

4. **Traduire la requête 3(a) en calcul relationnel à variable nuplet**.

Aucune explication demandée.

5. **Traduire la requête 3(b) en algèbre relationnelle**. *Aucune explication demandée.*

6. **Exprimer, les requêtes²⁶ suivantes en algèbre relationnelle et en calcul relationnel à variable nuplet**. *Aucune explication demandée.*

*Pour simplifier, dans la suite, vous pouvez abréger le nom des relations et des attributs par leur initiales (par exemple P pour **Personne**, A pour **Amis**, N pour **Nom** etc.). En algèbre relationnelle, vous pouvez également utiliser une requête précédente en nommant la requête R_i , où i est le numéro de la requête.*

- (a) Quelle est l'année de naissance d'Albert GAMOTTE ?
- (b) Qui (en précisant le nom et le prénom) est ami avec Albert GAMOTTE ?
- (c) Qui (en précisant le nom et le prénom) n'est pas amis avec Claude COMPUTING ?
- (d) Qui (en précisant le nom et le prénom) est ami avec Albert GAMOTTE OU Claude COMPUTING ?
- (e) Qui (en précisant le nom et le prénom) est ami avec Albert GAMOTTE ET Claude COMPUTING ?
- (f) Qui (en précisant le nom et le prénom) est ami avec Albert GAMOTTE MAIS PAS avec Claude COMPUTING ?
- (g) Qui (en précisant le nom et le prénom) est ami avec toutes les personnes de la base ?

7. **Exprimer, la requête suivante en calcul relationnel à variable nuplet** (*Aucune explication demandée*) : Qui (en précisant le nom et le prénom) est la personne la plus jeune ?

26. Vos requêtes doivent pouvoir être exécutées sur toutes les instances possibles, pas uniquement celles du sujet.

2.15 Examen Mai 2012

2.15.1 Exercice 1 - Requêtes d'interrogation (44 points soit 55%)

Soit la base de données (simplifiée) de schéma ci-dessous, permettant de gérer des résultats d'élections :

Candidat(CandidatID, Nom, Prénom, Groupe) de clé primaire CandidatID;

Scrutin(ScrutinID, Date, Tour) de clé primaire ScrutinID;

BureauVote(BureauID, Adresse) de clé primaire BureauID;

Electeur(ElecteurID, Nom, Prénom, #BureauID) de clé primaire ElecteurID et de clé étrangère #BureauID faisant référence à la clé primaire de BureauVote;

Vote(VoteID, #ScrutinID, #CandidatID, #ElecteurID) de clé primaire VoteID et de clés étrangères #ScrutinID faisant référence à la clé primaire de Scrutin, #CandidatID faisant référence à la clé primaire de Candidat et #ElecteurID faisant référence à la clé primaire de Electeur.

*Vous pourrez abréger le nom des relations par C, S, B, E et V, ainsi que les noms des attributs par leurs initiales (ex. **CandidatID** par CID, **Nom** par N, etc.).*

1. Exprimer les requêtes suivantes en algèbre relationnelle (non étendue), en calcul relationnel à variable nuplet et en SQL. *Aucune explication demandée.*
 - (a) Quels électeurs (en donnant leur nom et leur prénom) n'ont pas voté au premier tour du scrutin de la date 06/05/12?
 - (b) Quels électeurs (en donnant leur nom et leur prénom) ont voté à tous les scrutins enregistrés dans la base?
 - (c) Quels électeurs (en donnant leur nom et leur prénom) ont toujours voté pour des candidats du même groupe?
2. Exprimer les requêtes suivantes En SQL uniquement. *Aucune explication demandée*
 - (a) Combien il y a-t-il eu de votants au total lors du premier tour du scrutin du 06/05/12? (*Vous créerez une vue pour cette requête nommée NbVotants*).
 - (b) Quel est le nombre de votes obtenus lors du premier tour du scrutin du 06/05/12 pour chaque candidat (en donnant leur nom et leur prénom)? (*Vous créerez une vue pour cette requête nommée NbVotesParCandidat*).
 - (c) Quel est le pourcentage de votes obtenus lors du premier tour du scrutin du 06/05/12 pour chaque candidat (en donnant leur nom et leur prénom)? (*Vous pourrez utiliser les deux vues précédemment créées*).
3. Exprimer la requête suivante en SQL et en calcul relationnel à variable nuplet : Quels sont les candidats (en donnant leur nom et leur prénom) ayant obtenu le plus grand nombre de votes lors du premier tour du scrutin du 06/05/12? (*Vous pourrez utiliser une des vues précédemment créées*). *Aucune explication demandée.*

2.15.2 Exercice 2 - Dépendances fonctionnelles (7 points soit 8,5%)

En reprenant le schéma de la base de données de l'exercice précédent :

1. Exprimez en calcul à variable nuplet et SQL la dépendance fonctionnelle suivante, associée à la relation **Vote** : **#ScrutinID, #ElecteurID** \rightarrow **VoteID**. *Expliquez.*
Pour rappel, une contrainte peut être exprimée en SQL par une assertion, de la manière suivante :

```
CREATE ASSERTION Nom_Assertion
BEFORE COMMIT
CHECK (Condition) ;
```
2. Que pouvez-vous en déduire sur le couple d'attributs (**#ScrutinID, #ElecteurID**) de la relation **Vote**? *Expliquez.*

2.15.3 Exercice 3- Formes normales (10 points soit 12,5%)

Exercice repris et adapté de http://www-inf.int-evry.fr/cours/BD/BD_REL/EXOS/td-df.html et de http://cs.ulb.ac.be/public/_media/teaching/infoh303/bd2002ex2s.pdf.

Soit la relation $R(\text{Occupant}, \text{NbPersonnes}, \text{NoApt}, \text{Adresse}, \text{NbPièces}, \text{Propriétaire})$ et la famille F contenant les dépendances fonctionnelles associées à la relation R :

$\text{Occupant} \longrightarrow \text{Adresse}, \text{NoApt}, \text{nbPersonnes}$

$\text{Adresse}, \text{NoApt} \longrightarrow \text{Propriétaire}, \text{Occupant}, \text{NbPièces}$

Pour simplifier vous pouvez abréger le nom des attributs par leurs initiales (par exemple O pour Occupant , NbP pour NbPersonnes , NA pour NoApt etc.).

1. Quelle(s) est(sont) la(les) clé(s) minimale(s) de la relation R ? *Aucune explication demandée.*
2. En quelle forme normale est la relation R ? *Expliquez.*

2.15.4 Exercice 4- Décomposition de schéma (19 points soit 24%)

Exercice repris et adapté de <http://tice.agrocampus-ouest.fr/file.php/318/2012/DependencesFonctionnelles.pdf>.

Soit la relation $R(\text{NumEtudiant}, \text{NumCours}, \text{Intitulé}, \text{Note})$ et la famille F contenant les dépendances fonctionnelles associées à la relation R :

$\text{NumCours} \longrightarrow \text{Intitulé}$

$\text{NumCours}, \text{NumEtudiant} \longrightarrow \text{Note}$

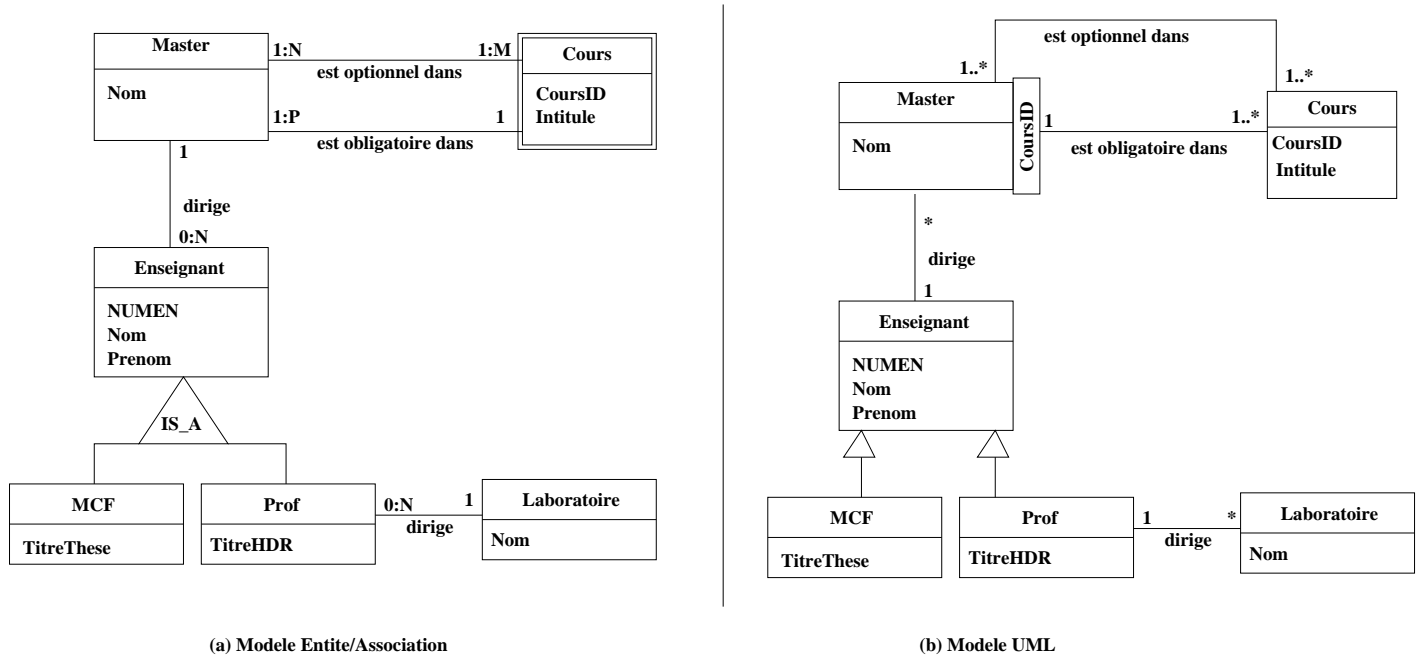
Soit une instance r de cette relation :

NumEtudiant	NumCours	Intitulé	Note
1	1	BD	12
2	1	BD	16
3	1	BD	8
1	2	R0	10

1. On propose de décomposer R en $R_1(\text{NumCours}, \text{Intitulé})$ et $R_2(\text{NumEtudiant}, \text{Note})$.
 - (a) Quelle(s) est(sont) la(les) clé(s) minimale(s) de R_1 et R_2 ? *Expliquez.*
 - (b) En quelle forme normale est chacune des relations R_1 et R_2 ? *Expliquez.*
 - (c) Calculer les instances r_1 et r_2 issues de la décomposition de r . *Aucune explication demandée.*
 - (d) Calculer la jointure de r_1 et r_2 . Que constatez-vous? *Expliquez.*
 - (e) La décomposition proposée est-elle SPI et/ou SPD? *Expliquez.*
2. Proposez une décomposition SPI et SPD de R . *Expliquez comment vous trouvez cette décomposition et pourquoi elle est SPI et SPD.*

2.16 Partiel de Mars 2012

2.16.1 Exercice 1 - Passage au relationnel (11 points soit 27,5%)



On souhaite définir une base de données permettant de gérer des Masters, dont le schéma de modélisation est présenté ci-dessus (modélisation en entité-association à gauche et en UML à droite). Les attributs **CoursID**, **Nom**, et **NUMEN** correspondent aux identificateurs. L'identificateur de chaque cours est relatif à chaque Master (par exemple il existe le cours numéro 1 du Master MIAGE-IF et le cours numéro 1 du Master MIAGE-SITN).

Déduisez le schéma relationnel de la base de données correspondante.

Vous préciserez les clés primaires des relations en les soulignant ainsi que les clés étrangères en les signalant par un # et en précisant à quoi elles font référence.

Dans votre schéma relationnel, chaque relation doit être spécifiée de la manière suivante :

$Nom(att_1, \dots, att_n)$ où Nom est le nom de la relation et att_1, \dots, att_n sont des noms d'attributs. Le nom de la relation doit obligatoirement avoir un lien avec les noms des ensembles d'entités (classes) ou des associations du schéma de modélisation de la question 1.

Vous donnerez des explications claires et concises du passage au relationnel. Vous préciserez notamment pourquoi et comment vous créez ou modifiez certaines relations (1 ligne maximum par relation).

2.16.2 Exercice 2 - Requêtes en algèbre relationnelle (18 points soit 45%)

Soient les deux relations suivantes :

- $R(A, B, C, D)$ d'instance r contenant N nuplet,
- $S(A, B, C)$ d'instance s contenant M nuplets. (Les attributs A, B et C de S sont de même domaine que les attributs de même nom dans R).

Pour chacune des requêtes ci-dessous, numérotées de (a) à (e), indiquez :

1. Si la requête est possible ou non en expliquant pourquoi.
2. Lorsque la requête est possible, indiquez :
 - 2.1 Quel est le schéma du résultat de la requête.
 - 2.2 Quel est le nombre de nuplets résultat. *Expliquez. Lorsque vous ne pouvez pas préciser le nombre exact de nuplets résultat, vous devez préciser une borne supérieure (ex. nombre de nuplets résultats $\leq y$ où y est une borne que vous devez définir).*

$$(a) r \cup s \quad (b) \Pi_{A,B,C}(r) - s \quad (c) r \times s \quad (d) r \bowtie s \quad (e) r \div s$$

2.16.3 Exercice 3 - Requêtes en calcul relationnel à variable nuplet (11 points soit 27,5%)

Soit la base de données de schéma ci-dessous, permettant de gérer des étudiants et leurs notes :

Etudiant(EtudiantID, Nom, Prenom, DateNaissance)

ReleveNotes(#EtudiantID, IntituleCours, Note)

Les attributs soulignés sont les clés primaires des relations. L'attribut **#EtudiantID** dans la relation **ReleveNotes** est une clé étrangère qui fait référence à la clé primaire de la relation **Etudiant**. Les attributs **EtudiantID** et **Note** sont de type entier. L'attribut **DateNaissance** est de type date. Les attributs **Nom**, **Prenom** et **IntituleCours** sont de type chaîne de caractères.

Le tableau suivant donne quelques exemples de nuplets de la relation **Etudiant** :

EtudiantID	Nom	Prénom	DateNaissance
1	GAMOTTE	Albert	01/09/1992
2	DEBECE	Aude	07/03/1991

Le tableau ci-dessous donne quelques exemples de nuplets de la relation **ReleveNote** :

EtudiantID	IntituleCours	Note
1	BD	19
1	Java	2
2	BD	10,5

*Pour simplifier, dans les requêtes, vous pouvez abréger le nom des relations et des attributs par leur initiales (par exemple **E** pour **Etudiant**, **RN** pour **ReleveNote**, **EID** pour **EtudiantID** etc.).*

Exprimez, les requêtes suivantes en calcul relationnel à variable nuplet. Aucune explication n'est demandée.

1. Qui (en précisant le nom et le prénom) a obtenu une note en BD ET en Java ?
2. Qui (en précisant le nom et le prénom) a obtenu une note en BD MAIS PAS en Java ?
3. Qui (en précisant le nom et le prénom) a la meilleure note en BD ?