

Conception de bases de données normalisées (décomposition en 3NF)

Table des matières

| | |
|--|-----------|
| I - Cours | 3 |
| 1. Conception de bases de données normalisées | 3 |
| 1.1. Exercice : Étapes de la conception d'une base de données..... | 3 |
| 1.2. Algorithme de décomposition 0NF->1NF | 3 |
| 1.3. Exemple de décomposition 0NF->1NF (attribut multivalué)..... | 4 |
| 1.4. Algorithme de décomposition 1NF->3NF | 4 |
| 1.5. Exemple de décomposition 1NF->3NF | 5 |
| 1.6. Normalisation par transformation d'attributs en méthodes | 6 |
| 1.7. Synthèse : Processus de conception d'une base de données normalisée | 7 |
| 2. Exemple de synthèse : MCD-Relationnel-Normalisation-SQL | 7 |
| 2.1. Première étape : Modélisation conceptuelle | 8 |
| 2.2. Deuxième étape : Modèle relationnel | 9 |
| 2.3. Troisième étape : Dépendances fonctionnelles | 10 |
| 2.4. Quatrième étape : Formes normales | 10 |
| 2.5. Cinquième étape : Décomposition | 10 |
| 2.6. Sixième étape : Implémentation SQL LDD | 11 |
| II - Exercices | 12 |
| 1. Exercice : À l'école II | 12 |
| 2. Exercice : Project manager | 12 |
| 3. Exercice : Objectifs II | 13 |
| 4. Exercice : Jeu de construction | 14 |
| 5. Exercice : Codes normalisés | 15 |
| 6. Exercice : À Deux Mains | 15 |
| Abréviations | 17 |
| Index | 18 |



1. Conception de bases de données normalisées

Objectifs

Savoir créer des schémas relationnels en troisième forme normale.

1.1. Exercice : Étapes de la conception d'une base de données

Mettre dans l'ordre les étapes de conception suivantes.

1. Rétro-conception d'un modèle UML normalisé
2. Décomposition des relations jusqu'à arriver en 3NF en préservant les DF
3. Énonciation de la forme normale avant normalisation
4. Établissement des DF sous la forme de la fermeture transitive pour chaque relation du modèle
5. Modélisation conceptuelle en UML ou E-A
6. Création du code SQL LDD pour un SGBDR ou un SGBDRO
7. Analyse de la situation existante et des besoins
8. Élaboration du modèle logique en relationnel ou relationnel-objet

Réponse :

1.2. Algorithme de décomposition ONF->1NF

Attributs composés



Méthode

Soit R une relation. Si R contient un attribut non atomique portant sur des valeurs hétérogènes (correspondant à un attribut composé au niveau conceptuel), alors l'attribut est décomposé en plusieurs attributs $a_1 \dots a_n$.

$R(\#pk, a, b, \dots)$ avec a non atomique se décompose en : $R(\#pk, a_1, a_2, \dots, a_n, b, \dots)$

Attributs multivalués

Soit R une relation avec la clé primaire pk . Si R contient un attribut non atomique portant sur des valeurs homogènes (correspondant à un attribut multivalué au niveau conceptuel), alors R est décomposée en R_1 et R_2 , tel que :

- R_1 est R moins l'attribut a
- R_2 est composé de pk et de a , avec (pk, a) clé primaire de R_2 et $R_2.pk$ clé étrangère vers $R_1.pk$

$R(\#pk, a, b, \dots)$ avec a non atomique se décompose en :

- $R_1(\#pk, b, \dots)$
- $R_2(\#pk \Rightarrow R_1, \#a)$

1.3. Exemple de décomposition 0NF->1NF (attribut multivalué)

Situation initiale



```
1 Élève(#login, nom, prénom, uvs)
```

Exemple d'enregistrement :

```
1 ('tpassoir', 'Passoire', 'Toto', ('SY02', 'NF26', 'NF17'))
```



L'attribut `uvs` est multivalué (donc non atomique). On va donc créer deux relations, la première sans l'attribut `uvs`, et la seconde qui va contenir autant d'enregistrements que de valeur par enregistrement dans l'enregistrement initial.

Situation finale



```
1 Élève(#login, nom, prénom)
2 UVÉlève(#login=>Élève, #uv)
```

En reprenant le même exemple, on va avoir l'enregistrement suivant dans `Élève` :

```
1 ('tpassoir', 'Passoire', 'Toto')
```

et les enregistrements suivants dans `UVÉlève` :

```
1 ('tpassoir', 'SY02')
2 ('tpassoir', 'NF26')
3 ('tpassoir', 'NF17')
```

1.4. Algorithme de décomposition 1NF->3NF

Décomposition > 1NF



Pour les NF supérieures à 1, afin de normaliser une relation R on réalise une décomposition en R_1 et R_2 pour chaque DFE responsable d'un défaut de normalisation tel que :

- la partie gauche de la DFE :
 1. devient la clé primaire de R_2
 2. devient une clé étrangère de R_1 vers R_2
- la partie droite de la DFE

1. est enlevée de R1
2. est ajoutée comme attributs simples de R2

Décomposition 1NF->2NF



Soit R une relation comportant une clé composée de k_1 et k_1' . Si R contient une DFE de k_1' vers des attributs n'appartenant pas à la clé, alors R est décomposée en R1 et R2, tel que :

- R1 est R moins les attributs déterminés par k_1' et avec k_1' clé étrangère vers R2
- R2 est composée de k_1' et des attributs qu'elle détermine, avec k_1' clé primaire de R2

$R(\#pk, k_1, k_1', a, b, c, \dots)$ avec (k_1, k_1') clé et $k_1' \rightarrow a, b$ se décompose en :

- $R1(\#pk, k_1, k_1' \Rightarrow R2, c, \dots)$
- $R2(\#k_1', a, b)$

Décomposition 2NF->3NF



Soit R une relation comportant une DFE de a vers b qui n'appartiennent pas à une clé, alors R est décomposée en R1 et R2, tel que :

- R1 est R moins les attributs déterminés par a et avec a clé étrangère vers R2
- R2 est composée de a et des attributs qu'elle détermine, avec a clé primaire de R2

$R(\#pk, a, b, c, \dots)$ avec $a \rightarrow b$ se décompose en

- $R1(\#pk, a \Rightarrow R2, c, \dots)$
- $R2(\#a, b)$

1.5. Exemple de décomposition 1NF->3NF

Situation initiale



1 Resultat ($\#pknum, knumetu, kuv, prenom, nom, credits, resultat, obtenu$) avec ($knumetu, kuv$) clé

DF :

- $pknum \rightarrow knumetu, kuv, prenom, nom, credits, resultat, obtenu$
- $knumetu, kuv \rightarrow pknum, prenom, nom, credits, resultat, obtenu$
- **$knumetu \rightarrow prenom, nom$**
- **$kuv \rightarrow credits$**
- **$resultat \rightarrow obtenu$**

La relation est en 1NF.

Exemple d'instance

| pknum | knumetu | kuv | prenom | nom | credits | resultat | obtenu |
|-------|---------|------|----------|-------|---------|----------|--------|
| 1 | X01 | NF17 | Pierre | Alpha | 6 | A | oui |
| 2 | X01 | NF26 | Pierre | Alpha | 6 | B | oui |
| 3 | X02 | NF17 | Alphonse | Béta | 6 | F | non |

knumetu → prenom,nom

```
1 Resultat (#pknum,knumetu=>Etudiant,kuv,credits,resultat,obtenu)
2 Etudiant (#knumetu,prenom,nom)
```

kuv → credits

```
1 Resultat (#pknum,knumetu=>Etudiant,kuv=>Uv,resultat,obtenu)
2 Etudiant (#knumetu,prenom,nom)
3 Uv(#kuv,credits)
```

resultat → obtenu

```
1 Resultat (#pknum,knumetu=>Etudiant,kuv=>Uv,resultat=>Note)
2 Etudiant (#knumetu,prenom,nom)
3 Uv(#kuv,credits)
4 Note(#resultat,obtenu)
```

1.6. Normalisation par transformation d'attributs en méthodes

Il arrive que la fonction sous-jacente à la DF soit une fonction simple, que l'on peut calculer. Par exemple pour la DF : $ddn \rightarrow age(ddn*)$, on peut calculer *age* en fonction de *ddn* par le calcul : $age = today() - ddn$.

**Méthode**

Chaque fois que c'est possible, on remplacera un attribut par une méthode si cela permet de supprimer une DF sans perdre d'information.

Cette solution est à privilégier *a priori* sur une décomposition.

En relationnel, on supprimera l'attribut de la table et on ajoutera une vue permettant de le retrouver.

Existence d'une fonction de calcul simple**Exemple**

Soit la relation en 2NF :

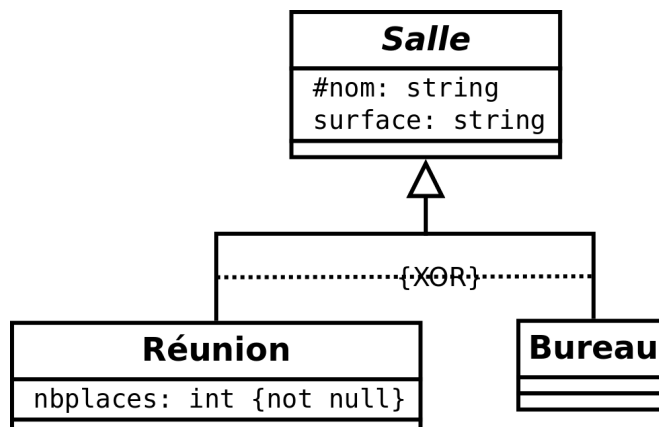
```
1 personne (#numsecu, nom, prenom, ddn, age) avec ddn → age
```

On remplacera cette relation par une relation en 3NF et une vue :

```
1 Personne (#numsecu, nom, prenom, ddn)
2 vPersonne (#numsecu, nom, prenom, ddn, age) avec age = today() - ddn
```

Cas de transformation d'un héritage exclusif par la classe mère**Exemple**

Soit le schéma UML suivant :



Une transformation de l'héritage par la classe mère donnera :

```
1 Salle (#nom:string, surface:string, nbplaces:int, type:{Réunion|Bureau})
2 DF : nom → surface, type, nbplaces et nbplaces → type
```

On peut donc supprimer l'attribut type et aboutir à la relation en 3NF :

```
1 Salle (#nom:string, surface:string, nbplaces:int)
2 vSalle (nom, surface, nbplaces, type) avec SI nbplace IS NULL ALORS type=Bureau
   SINON type=Réunion
```

1.7. Synthèse : Processus de conception d'une base de données normalisée

1. Analyse et clarification du problème posé
2. Modélisation conceptuelle en UML
Résultat : MCD₁
3. Traduction en relationnel en utilisant les règles de passage UML vers relationnel
Résultat : MLD₁
4. Établissement des DF sous la forme de la fermeture transitive pour chaque relation du modèle
Résultat : MLD₁ avec F+
5. Établissement de la forme normale
Résultat : MLD₁ avec F+ et NF
6. Décomposition des relations jusqu'à arriver en 3NF en préservant les DF
Résultat : MLD₂ avec F+ et 3NF
7. Rétro-conception du modèle UML correspondant
Résultat : MCD₂
8. Implémentation en SQL du modèle MLD₂

La normalisation permet de décomposer un schéma relationnel afin d'obtenir des relations non redondantes.

La 3NF* est souhaitable car toujours possible à obtenir, sans perte d'information et sans perte de DF*. La BCNF* est également indiquée, car elle est un peu plus puissante, et plutôt plus simple que la 3NF.

La BCNF* n'est pas encore suffisante pour éliminer toutes les redondances. Il existe pour cela les 4NF* et 5NF* qui ne sont pas abordées dans ce cours. Notons également que les cas de non-4NF et de non-5NF sont assez rares dans la réalité.

2. Exemple de synthèse : MCD-Relationnel-Normalisation-SQL

Problème posé

Soit un modèle conceptuel représentant :

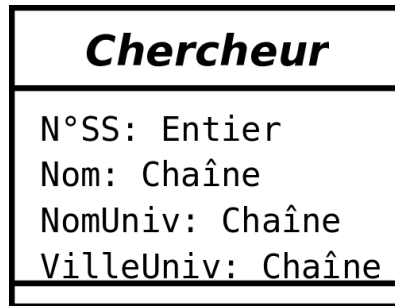
- un type d'entité "chercheur", identifié par le numéro de sécurité sociale, et possédant les autres propriétés suivantes : le nom, le nom de l'université à laquelle il appartient, la ville dans laquelle est basée cette université.
- un type d'entité "professeur", héritant de "chercheur"
- un type d'entité "doctorant", héritant de "chercheur"
- une association de type "encadrement" entre professeur et doctorant (un professeur pouvant encadrer plusieurs doctorants et un doctorant n'ayant qu'un et un seul directeur de thèse).

Afin de réaliser le modèle de données :

1. Dessiner le modèle conceptuel
2. Traduire le modèle conceptuel en modèle logique relationnel.
3. Après avoir identifié les DF, normaliser le modèle relationnel en BCNF.
4. Ecrire les instructions SQL de création d'un tel modèle.

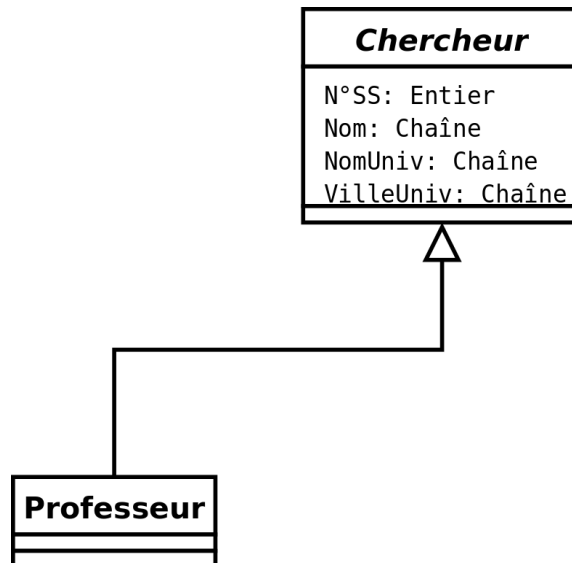
2.1. Première étape : Modélisation conceptuelle

Modélisation conceptuelle : Entité Chercheur



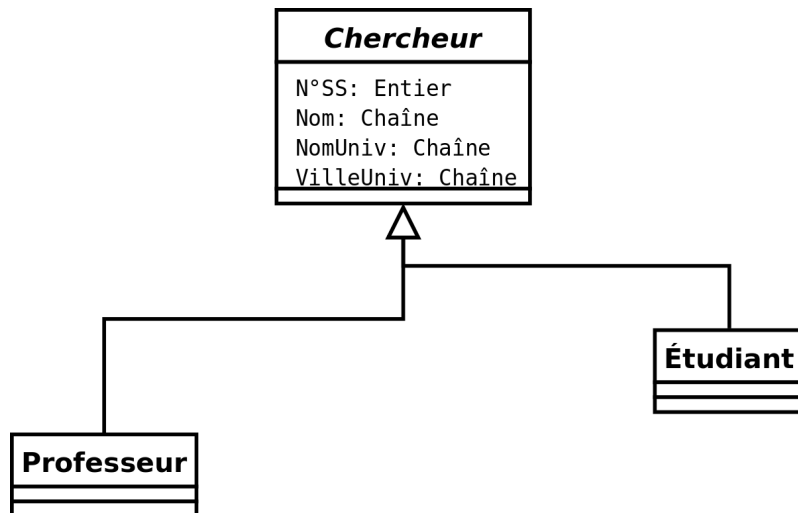
Conception UML (1/4)

Modélisation conceptuelle : Entité Professeur



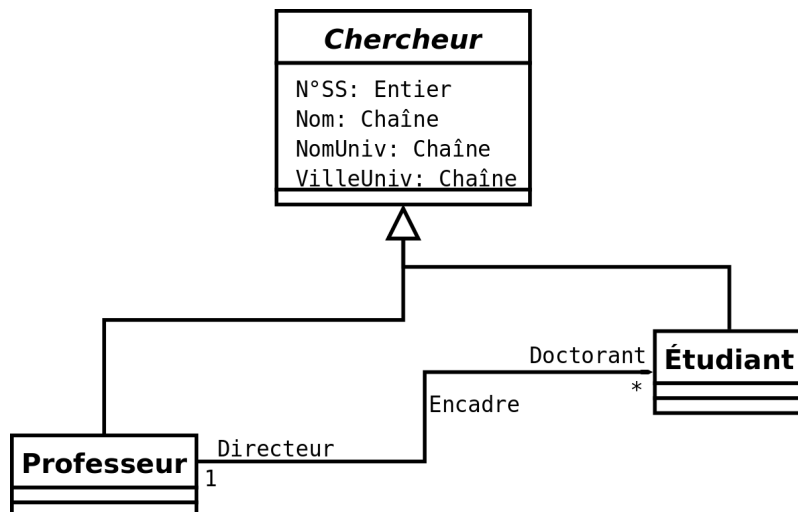
Conception UML (2/4)

Modélisation conceptuelle : Entité Doctorant



Conception UML (3/4)

Modélisation conceptuelle : Association Encadrement



Conception UML (4/4)

2.2. Deuxième étape : Modèle relationnel

Modèle relationnel : Héritage



Choix de transformation de l'héritage : L'héritage est exclusif (les professeurs ne sont plus doctorants), mais pas complet, **car l'association Encadre n'est pas symétrique**. On choisit donc un héritage par les classes filles (Chercheur étant par ailleurs abstrait).

Modèle relationnel : Entité Professeur



```
1 Professeur (#N°SS:int(13), Nom:char(20), NomUniv:char(50), VilleUniv:char(20))
```

Modèle relationnel : Entité Doctorant



```
1 Professeur (#N°SS:int(13), Nom:char(20), NomUniv:char(50), VilleUniv:char(20))
2 Doctorant (#N°SS:int(13), Nom:char(20), NomUniv:char(50), VilleUniv:char(20))
```

Modèle relationnel : Association EncadréPar**Méthode**

```

1 Professeur (#N°SS:int(13), Nom:char(20), NomUniv:char(50), VilleUniv:char(20))
2 Doctorant (#N°SS:int(13), Nom:char(20), NomUniv:char(50), VilleUniv:char(20),
  EncadrePar=>Professeur)

```

2.3. Troisième étape : Dépendances fonctionnelles**Dépendances fonctionnelles : DF évidente (professeur)****Méthode**

- Professeur.N°SS → Nom, NomUniv, VilleUniv

Dépendances fonctionnelles : DF évidente (doctorant)**Méthode**

- Professeur.N°SS → Nom, NomUniv, VilleUniv
- Doctorant.N°SS → Nom, NomUniv, VilleUniv, EncadrePar

Dépendances fonctionnelles : DF déduites du sens des propriétés**Méthode**

Connaissant l'université, on connaît la ville, donc :

- Professeur.N°SS → Nom, NomUniv, VilleUniv
- Professeur.NomUniv → VilleUniv
- Doctorant.N°SS → Nom, NomUniv, VilleUniv, EncadrePar
- Doctorant.NomUniv → VilleUniv

2.4. Quatrième étape : Formes normales**Forme normale : 1NF****Méthode**

Le schéma est en 1NF (clés et attributs atomique).

Forme normale : 2NF**Méthode**

Le schéma est en 2NF (la clé est composée d'un seul attribut)

Forme normale : 3NF**Méthode**

La schéma n'est pas en 3NF : NomUniv → VilleUniv

2.5. Cinquième étape : Décomposition**Normalisation en BCNF : Résultat****Méthode**

```

1 Professeur (#N°SS:int(13), Nom:char(20), NomUniv=>Univ)
2 Doctorant (#N°SS:int(13), Nom:char(20), NomUniv=>Univ, EncadrePar=>Professeur)
3 Univ (#NomUniv:char(50), VilleUniv:char(20))

```

Normalisation en BCNF : Vérification**Méthode**

Le modèle est bien en BCNF, toutes les DF ont pour source une clé.

Normalisation en BCNF : Conservation des DF**Méthode**

La transformation préserve les DF car :

- N°SS → NomUniv et Univ.Nom → Ville
- Donc N°SS → Univ.Ville (par transitivité)

2.6. Sixième étape : Implémentation SQL LDD**Implémentation SQL : Professeur****Méthode**

```
1 CREATE TABLE Professeur (
2   Numss VARCHAR(13) PRIMARY KEY,
3   Nom CHAR(20) NOT NULL);
```

Implémentation SQL : Doctorant**Méthode**

```
1 CREATE TABLE Professeur (
2   Numss VARCHAR(13) PRIMARY KEY,
3   Nom CHAR(20) NOT NULL);
4
5 CREATE TABLE Doctorant (
6   Numss VARCHAR(13) PRIMARY KEY,
7   Nom CHAR(20) NOT NULL,
8   EncadrePar VARCHAR(13) REFERENCES Professeur(Numss));
```

Implémentation SQL : Univ**Méthode**

```
1 CREATE TABLE Univ (
2   Nom CHAR(50) PRIMARY KEY,
3   Ville CHAR(20) );
4
5 CREATE TABLE Professeur (
6   Numss VARCHAR(13) PRIMARY KEY,
7   Nom CHAR(20) NOT NULL,
8   NomUniv CHAR(50) REFERENCES Univ(Nom));
9
10 CREATE TABLE Doctorant (
11   Numss VARCHAR(13) PRIMARY KEY,
12   Nom CHAR(20) NOT NULL,
13   NomUniv CHAR(50) REFERENCES Univ(Nom),
14   EncadrePar VARCHAR(13) REFERENCES Professeur(Numss));
```

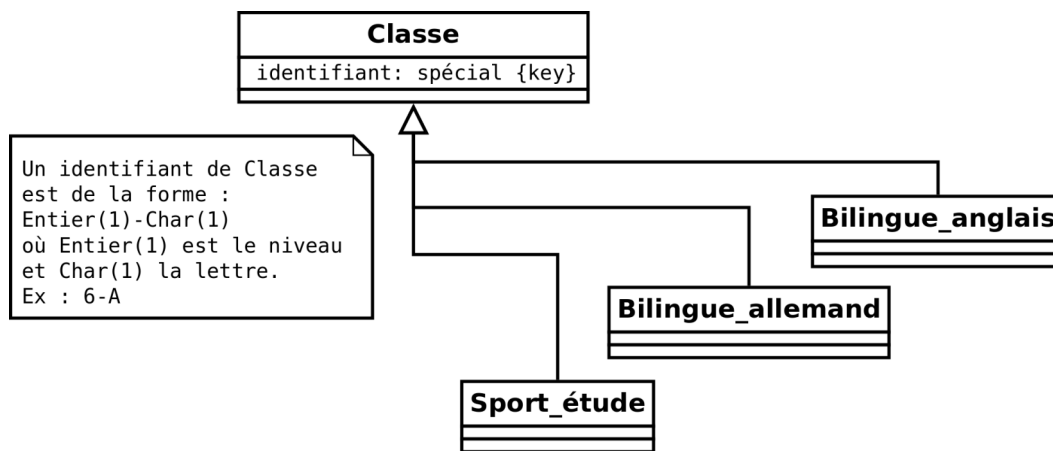
Exercices



1. Exercice : À l'école II

[10 min]

Une école est dotée d'une base de données pour gérer ses classes. On dispose des modèles UML et relationnel suivant décrivant cette base de données.



```
1 classe (#id:entier(1)-char(1),
    type{normal|sport_étude|bilingue_anglais|bilingue_allemand})
```

Question

Montrer que ce schéma n'est pas en première forme normale, et proposer une correction du schéma relationnel.

2. Exercice : Project manager

[20 min]

Soit la table suivante (Emp signifie *Employee*, Pro signifie *Project* et Man signifie *Manager*).

| Emp | EmpName | Pro | ProName | Man | ManName |
|-----|---------|-----|---------|-----|---------|
| E01 | John | P1 | Eco | M1 | Becky |
| E02 | Mary | P2 | Admin | M2 | Joe |
| E03 | Mark | P2 | Admin | M2 | Joe |
| E04 | Travis | P3 | Educ | M1 | Becky |

Soit les dépendances fonctionnelles suivantes :

- Emp → EmpName
- Emp → Pro
- Pro → ProName
- Pro → Man
- Pro → ManName

- Man → ManName

Question 1

Montrer que ce modèle n'est pas en 3NF.

Question 2

Proposer un modèle équivalent en 3NF.

Question 3

Réécrivez les données en respectant ce nouveau schéma.

3. Exercice : Objectifs II

[25 min]

Vous êtes le président de l'association "Objectifs", dont l'objet est d'aider ses étudiants membres à mener des projets dans le cadre de leurs études. Pour le moment cette association gère tous ces projets avec un logiciel de type tableur.

| Numéro | Projet / Tâche | Spécialités | Chef de projet | Participant tâche | Dates | Partenaire | |
|--------|-------------------------|-------------|-------------------------------------|-------------------------------------|---------------|----------------------------|----------------------------------|
| 1 | La nuit du Picolo | Logistique | Paul Densmore ; Spécialiste Musique | | 25 décembre | 1666 (Brasseur) | Bières offertes |
| 1.1 | Gestion | | | Barbara Krieger ; Spécialiste Sport | | | |
| 2 | Escalade de l'Everest | Voyages | Paul Manzarek | | Intersemestre | Pentathlon 1666 (Brasseur) | Fourniture de matériel Publicité |
| 3 | Tournoi de Volley-Ball | Sport | Paul Manzarek | | 15/5 - 23/5 | | |
| 3.1 | Recrutement des équipes | | | Barbara Krieger ; Spécialiste Sport | | | |
| 3.2 | Gestion | | | Paul Densmore ; Spécialiste Musique | | | |

Exemple de fichier de gestion des projets de l'association Objectifs

La concision et la clarté de la rédaction, ainsi que la bonne mobilisation des concepts et de la terminologie du domaine des bases de données seront intégrées à l'évaluation.

Question 1

On considérera ce tableau comme une relation, et on admettra que cette extension est suffisante pour l'interprétation des données :

1. **Démontrer** que cette relation ne peut admettre qu'une seule clé.
2. **Démontrer** que la 1NF* n'est pas respectée à plusieurs reprises (on peut trouver jusqu'à six causes empêchant la 1NF, proposez-en un maximum).

Question 2

On propose une première décomposition pour que le modèle soit en 1NF.

- 1 Projet (#Num, Nom, Debut, Fin, Specialite, CDP-Prenom, CDP-Nom, CDP-Specialite)
- 2 Tache (#NumProjet=>Projet, #NumTache, Nom, Participant-Prenom, Participant-Nom, Participant-Specialite)
- 3 Partenaire (#Nom, Description)
- 4 Partenariat (#Partenaire=>Partenaire, #Projet=>Projet, role)

1. Montrer que ce modèle n'est pas en 3NF et lister l'ensemble des DFE responsables.
2. Proposer une décomposition en 3NF.

4. Exercice : Jeu de construction

[40 min]

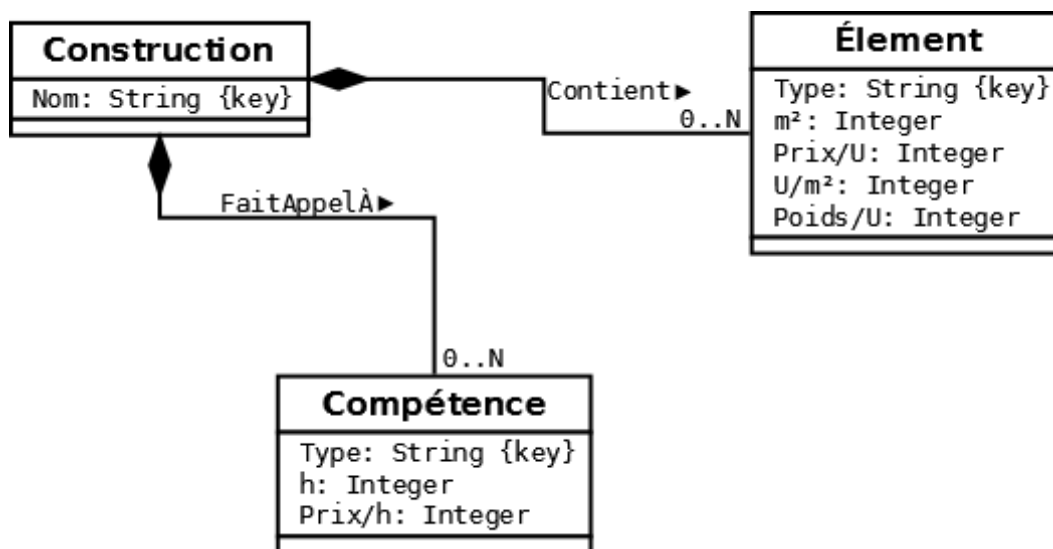
Soit le diagramme UML ci-après décrivant des **constructions** composées d'**éléments** et faisant appel à des **compétences** pour leur montage.

Par exemple une "Maison de type I" contiendra :

- 100 m² de "Brique traditionnelle" (Type de l'élément), pour un prix de 2 euros par unité (Prix/U), 12 briques par m² (U/m²) et 100g / brique (Poids/U)
- 125 m² de "Tuile plate" (Type de l'élément), pour un prix de 4 euros par unité (Prix/U), 24 tuile par m² (U/m²) et 75g / tuile (Poids/U)

et fera appel à :

- 500h de travail de "Maçon" (Type de compétence), pour un tarif horaire de 20€/h
- 425h de travail de "Couvreur" (Type de compétence), pour un tarif horaire de 30€/h
- 75h de "Chef de chantier" (Type de compétence), pour un tarif horaire de 45€/h



Modèle UML

Question 1

Effectuer le passage au relationnel de ce modèle.

Question 2

Sachant que deux mêmes types d'élément ont **toujours** les mêmes caractéristiques (Prix/U, U/m², Poids/U) et que deux mêmes types de compétence ont **toujours** les mêmes prix (Prix/h), établir la fermeture transitive des DFE et la forme normale de ce modèle (justifier).

Donner un exemple de données redondantes.

Indice :

Deux types de maison ont évidemment des quantités d'éléments (m²) et de compétences (h) différentes.

Question 3

Normaliser le modèle en 3NF (justifier, et montrer que la transformation est sans perte).

Question 4

En repartant de la forme normalisée en 3NF, rétro-concevoir un modèle UML qui aurait permis d'aboutir directement à ce schéma non redondant.

5. Exercice : Codes normalisés

[15 min]

On considère la classe Ville ci-après et on renomme les attributs comme suit :

- code_insee : ci
- code_postal : cp
- code_pays_alpha-2 : a2
- code_pays_alpha-3 : a3
- code_pays_un : un

| Ville |
|--|
| code_insee : entier {unique frozen} code_postal : entier {frozen} code_pays_alpha-2 : char {frozen} code_pays_alpha-3 : char {frozen} code_pays_un : entier {frozen} |

On pose les DF suivantes :

- ci → cp, a2, a3, un
- a2 → a3, un
- a3 → a2, un
- un → a2, a3

Question 1

Exprimez la fermeture transitive et trouvez les clés.

Question 2

Le modèle est-il en 3NF ? Justifiez et si nécessaire, normalisez.

6. Exercice : ÀDeuxMains

[30 min]

Une entreprise de maintenance de matériel industriel

La société ÀDeuxMains est en charge de la maintenance de machines industrielles. La société gère plusieurs agences ; chaque agence est affectée à exactement une région dans le monde.

Agences

On dispose d'un fichier CSV `region.csv` correspondant au schéma suivant :

```
1 Agence (numéro_agence:integer, nom_région:varchar, population_région:integer,
  surface_région:integer, densité_région:real, nom_pays:varchar, pib_pays:integer)
```

On dispose des informations complémentaires suivantes :

- `numéro_agence` est unique et non null
- `nom_région` est unique et non null
- `densité_région` = `population_région` / `surface_région`
- `nom_pays` détermine `pib_pays`

On ne fera aucune hypothèse de dépendance fonctionnelle complémentaire.

Question 1

Montrez que ce modèle n'est pas en 3NF (rappelez les clés candidates et énoncez la liste des DFE sous la forme d'une F+). Proposez une solution de normalisation **en relationnel** (sans perte).

Proposez un modèle UML correspondant à la version normalisée.

Machines

On dispose de la liste des types de machines gérés par ÀDeuxMains dans un fichier de tableur `types.ods`.

| #machine | spécialités |
|-----------------|-------------------------------------|
| Compressor 08 | Électricité / Mécanique |
| Aérienne | Électricité |
| Mixte | Électricité / Plomberie |
| Aquatique | Électricité / Plomberie / Mécanique |
| Full Fuel | Plomberie |
| Naturéo Carbone | Plomberie |
| Flash | Électricité |

On sait que les noms des machines sont des clés.

Question 2

Montrez que ce modèle n'est pas en 3NF. Proposez une solution de normalisation **en relationnel** (sans perte).

Proposez un modèle UML correspondant à la version normalisée.

Abréviations



1NF : First Normal Form

3NF : Third Normal Form

4NF : Fourth Normal Form

5NF : Fifth Normal Form

BCNF : Boyce-Codd Normal Form

DDN : Date De Naissance

DF : Dépendance Fonctionnelle

Index



| | |
|---------------------|---|
| Décomposition | 3 |
| Dépendance | 3 |
| Logique | 3 |
| Modèle | 3 |
| Normalisation..... | 3 |
| Redondance | 3 |
| Relationnel | 3 |