

# ***R1.05***

## **Introduction aux Bases de Données**

### **Partie 1**

### **TD4**

### **Contraintes textuelles annexes**

**BUT Informatique S1-1A**

***R. Fleurquin, A. Ridard***

Septembre 2022

#### **Contenu du TD**

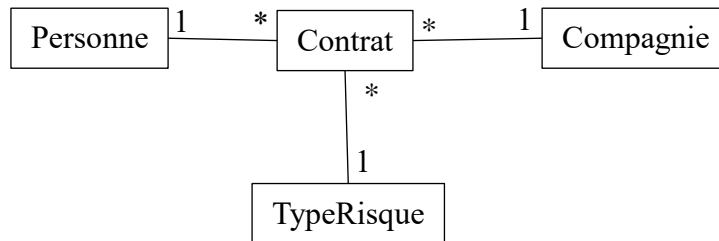
- |  |
|--|
| <input type="checkbox"/> Intérêt et documentation des contraintes textuelles en annexe des diagrammes de classes |
|--|

## Exercice 1 : Intérêt des contraintes textuelles

On considère le domaine ci-dessous.

« Une personne peut souscrire des assurances liées à certains types de risque (décès, infirmité, habitation, voiture, etc.) auprès de compagnies d'assurance. Un contrat d'assurance ne porte que sur un seul type de risque, une seule personne et dans une seule compagnie. Une personne peut souscrire plusieurs assurances chez des assureurs différents pour le même type de risque ou non. Les compagnies couvrent plusieurs types de risque. Une personne peut donc également souscrire plusieurs assurances au sein d'une même compagnie sous réserve que cela porte sur des types de risque différents. »

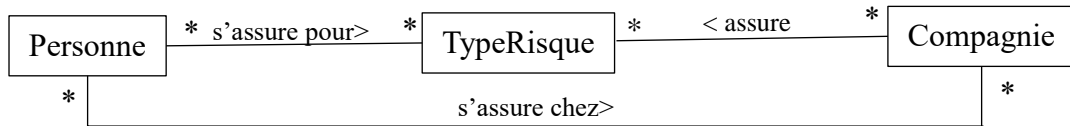
Un développeur propose de modéliser ce domaine avec le diagramme ci-dessous.



1. Est-il possible pour une personne de souscrire plusieurs assurances pour le même risque dans des compagnies différentes ? Peut-il souscrire plusieurs assurances au sein de la même compagnie pour des risques différents ? Peut-il souscrire, bien que cela soit interdit, plusieurs assurances au sein de la même compagnie pour le même type de risque ? Justifiez systématiquement votre réponse par un argument ou par un diagramme d'objet compatible.

*Le problème est qu'ici avec uniquement des associations binaires il est impossible de poser l'unicité d'un triplet d'instances de 3 classes différentes : donc de respecter la dernière phrase du texte décrivant le domaine. Il existe un concept langagier UML qui permet de représenter exactement ce domaine : l'association n-aire (ici ternaire) UML qui généralise le concept d'association binaire entre 2 classes à  $n > 2$  classes. Nous n'introduisons pas dans ce cours ce type d'association. Elles sont peu fréquentes dans les diagrammes, délicates à maîtriser et donc trop souvent mal utilisées par les développeurs. On essaye donc de se débrouiller avec uniquement les associations binaires éventuellement complétées de contraintes textuelles en annexe lorsqu'elles se révèlent insuffisantes. C'est d'autant plus justifiable qu'au final dans le schéma relationnel résultant d'une modélisation UML les associations n-aires ne sont implantables structurellement que comme une traduction de  $n$  associations binaires. On perd donc de toute manière la puissance d'expression des n-aires à l'implantation. Il faut alors par du code supplémentaire (procédures, triggers) forcer le respect des contraintes non prises en charge.*

2. Complétez le diagramme de classes par une contrainte textuelle pour qu'il modélise parfaitement le domaine.
3. On ajoute au domaine initial la contrainte qui suit : « une personne ne peut s'assurer qu'une seule fois au plus pour un type de risque donné ». Un développeur propose alors le diagramme qui suit. Dans ce diagramme de classes UML : i) une personne peut-elle s'assurer plusieurs fois pour un même type de risque? ii) une personne peut-elle s'assurer pour des risques différents dans la même compagnie ou dans des compagnies différentes ? iii) une personne peut-elle s'assurer pour un risque qui n'est pas pris en charge par les compagnies dans lesquelles elle s'assure ? iv) est-il toujours possible de savoir dans quelle compagnie s'assure une personne pour un risque donné ? Peut-on proposer une contrainte textuelle en annexe qui corrige tout ou partie des problèmes relevés ?



*Le problème est qu'ici avec uniquement des associations binaires il est impossible de poser une dépendance fonctionnelle élémentaire impliquant 3 classes : soit ici pour un couple (Personne, typeRisque) on ne veut qu'au plus une seule compagnie soit possible. Là encore, seul l'usage d'une association n-aire (ternaire ici) permettrait de décrire parfaitement ce domaine. L'ajout de contraintes textuelles en annexe d'un diagramme de classes pallie le manque de puissance des associations binaires. Ainsi complété, on obtient une solution correcte et simple.*

4. On ajoute au domaine initial la double contrainte qui suit : « une personne ne peut s'assurer qu'une seule fois au plus pour un type de risque donné. Un type de risque n'est assuré que par une seule compagnie (mais plusieurs types de risques peuvent être couverts par la même compagnie) ». Proposez un diagramme de classes modélisant ce domaine. Est-il nécessaire de lui adjoindre des contraintes textuelles en annexe ?
5. Tous les diagrammes dessinés dans les questions précédentes permettent-ils de conserver les types de risque qu'assurent chaque compagnie en l'absence de contrats ? Modifiez les diagrammes concernés pour ce faire. Quel problème se pose alors ? Réglez-le par une contrainte textuelle.

## Exercice 2 : documentation des contraintes textuelles

On considère le domaine ci-dessous d'un journal qui souhaite conserver des renseignements sur les articles qui ont été publiés dans ses colonnes. Après interview du directeur du journal, on a rassemblé les informations qui suivent. Son souhait est d'informatiser cette connaissance et de permettre en particulier de retrouver des articles traitant de tel ou tel sujet, de trouver un journaliste compétent sur tel ou tel sujet, etc.

*« La rédaction du journal compte plusieurs pôles : actualité, économie, société, culture et sport. Chaque pôle est en charge prioritairement d'alimenter certaines rubriques particulières du journal mais de manière non exclusive, d'autres pôles pouvant abonder ces mêmes rubriques de temps en temps. Par exemple le pôle « société » alimente souvent aussi bien la rubrique « actualité locale » avec les autres pôles, mais prioritairement les rubriques « éducation » ou « justice » et « nécrologie » du journal. On souhaite conserver le nombre des rédacteurs de chaque pôle.*

*La rédaction emploie uniquement des journalistes sous deux statuts distincts : les rédacteurs en chef qui sont responsables d'un et d'un seul pôle et les rédacteurs. Tous ces journalistes sont affectés à un et un seul pôle. Les journalistes sont identifiés par leur nom et prénom. Chaque journaliste a une carte de presse avec un numéro unique. Les rédacteurs disposent d'un grade qui est un entier compris entre 1 et 10. Les rédacteurs en chef gèrent tous les rédacteurs de leur pôle.*

*Un article est rédigé par un ou plusieurs journalistes (pas nécessairement tous du même pôle). Chaque article est associé à un thème ou plusieurs thèmes dont l'un est identifié comme thème principal. Il est caractérisé par sa date de rédaction, sa date de parution et bien évidemment la rubrique dans laquelle il a été publié. La rubrique de publication est conforme au thème principal de l'article.*

*Le journal organise ces thèmes sous une forme arborescente : un thème est décomposé en sous-thèmes, eux-mêmes décomposables en sous thèmes, etc. Un thème est identifié de manière unique par son intitulé. Un thème est associé systématiquement à une et une seule rubrique du journal. Pour chaque journaliste on conserve les thèmes sur lesquels il a des compétences particulières (généralement suite à la rédaction d'articles sur ce thème mais pas toujours). Pour chaque thème on souhaite connaître le nombre des articles qui ont traité de ce thème. »*

Un développeur (qui n'est pas diplômé du DUT de Vannes) propose à son chef de projet de modéliser ce domaine avec le diagramme ci-dessous. Son chef de projet (ancien DUT Info de Vannes lui) le félicite de ce travail (il faut toujours encourager les efforts et valoriser les actions, c'est cela aussi être un bon chef... 😊) mais il lui indique (gentiment) qu'il n'a fait là que la moitié du chemin : il manque encore les contraintes qui vont permettre à ce modèle de représenter le domaine le plus fidèlement possible.

**Sans modifier ce diagramme**, rajoutez des contraintes soit prédéfinies dans le langage UML ou admises par l'IUT directement sur ce diagramme, soit des contraintes textuelles en annexe de ce diagramme pour qu'il se rapproche le plus possible du domaine à modéliser.

Les contraintes tolérées directement sur ce diagramme sont : abstract, xor, subset, attribut ou association dérivable, identifiant.

Les contraintes en annexe porteront notamment sur :

- les domaines de valeur (syntaxe dom(attribut){...}
- la résolution des problèmes de cycles, etc.

Les contraintes en annexe respecteront le format qui suit :

Nom des classes ou associations concernées.

Contraintes 1

Contraintes 2...

