

## Exercice 1 : Vue logique d'une BD relationnelle et Datalog (9 pts)

On considère une base de données relationnelle sur les films dont le schéma est le suivant :

Film [IDFilm, Titre, Année, Durée]  
 Personne [IDPersonne, Nom, DateNaissance]  
 Casting [IDFilm, IDPersonne, Role, Personnage]

Question 1. Traduisez ce schéma en un ensemble de prédicats logiques, dont vous préciserez l'arité.

Question 2. Combien de faits sur chacun des prédicats obtenez-vous avec l'extrait de BD ci-dessous ? On ne vous demande pas de lister les faits.

FILM

IDFilm	Titre	Année	Durée
t84	Star Wars:Episode V	1980	121
t90	Star Wars:Episode VI	1983	131
t96	Star Wars:Episode VII	2015	138
t36	Star Wars:Episode VIII	2017	152
t38	Star Wars:Episode IX	2019	142

PERSONNE

IDPers	Nom	DateN
n984	Irvin Kershner	1923
n434	Mark Hamill	1951
n459	Daisy Ridley	1992

CASTING

IDFilm	IDPers	Rôle	Personnage
t84	n984	réalisateur	
t84	n148	acteur	Han-solo
t38	n434	acteur	Luke S.
t96	n459	actrice	Rey
t36	n459	actrice	Rey
t96	n434	acteur	Luke S.

Question 3. Dans la relation CASTING, l'attribut Rôle a 4 valeurs possibles : réalisateur, compositeur, acteur et actrice. Exprimez la requête suivante sous la forme d'une requête en logique du premier ordre utilisant les prédicats de la question 1 (ainsi que le prédicat binaire  $\neq$  si besoin) :

Donnez les noms d'acteurs ou actrices N1 et N2 et les titres de film T tels que N1 et N2 ont joué ensemble dans le film T.

Votre requête est-elle une requête conjonctive ou une union de requêtes conjonctives (UCQ) ? Si ce n'est pas le cas, pourrait-elle s'exprimer sous cette forme ? (Expliquez).

Question 4. On ajoute au vocabulaire logique 4 prédicats de façon à pouvoir exprimer des requêtes de façon plus commode :

- *joueDans* qui lie une personne (identifiée par IDPersonne) à un film (identifié par IDFilm) dans lequel cette personne est acteur ou actrice ;
- *aPourNom* qui lie une personne (IDPersonne) à son nom ;
- *aPourTitre* qui lie un film (IDFilm) à son titre ;
- *joueAvecDans* qui lie deux personnes (identifiées par IDPersonne) et un film (IDFilm) tel que ces deux personnes jouent dans ce film.

Définissez ces prédicats par des règles Datalog (en utilisant notamment les prédicats du schéma de BD).

Question 5. Supposons qu'on sature la base de faits de la question 2 avec les règles de la question 4 : quels faits de prédicat *joueDans* et *joueAvecDans* sont obtenus ? On ne vous demande pas de donner toute la base de faits saturée.

Question 6. Exprimez la requête de la question 3 comme une requête Datalog en vous appuyant sur les règles de la question 4.

## Exercice 2 : Traductions logiques de RDFS (8 pts)

On considère l'ensemble de 13 triplets RDFS ci-dessous, sur le thème de Star Wars. On utilise une notation simplifiée pour les IRI's prédéfinies, c'est-à-dire sans préciser le préfixe `rdf:` (pour `rdfs:type`) ou `rdfs:` (pour `rdfs:subclass`, `rdfs:subproperty`, `rdfs:domain` et `rdfs:range`).

```
<:Luke :aPourArme _b1>
<_b1 type :SabreLaser>
<:Luke :pilote :XWing1>
<:XWing1 type :Vaisseau>
<:Rey :pilote :Speeder1>
<:Speeder1 type :Vaisseau>
<:Vaisseau subclass :Véhicule>
<:Véhicule subclass :Objet>
<:SabreLaser subclass :Objet>
<:pilote subproperty :utilise>
<:aPourArme subproperty :utilise>
<:pilote domain :Personnage>
<:pilote range :Véhicule>
```

**Question 1.** Donnez deux façons de traduire ces triplets en une base de connaissances logique, composée d'une base de faits  $F$  et d'une base de règles Datalog  $\mathcal{R}$ .

On rappelle ci-dessous les règles d'implication de RDFS. Si vous avez besoin de faire référence à ces règles, il n'est pas nécessaire de donner leur traduction logique complète : donnez juste la traduction de la première règle (on doit pouvoir clairement distinguer entre les variables et les constantes).

```
(p, domain, o), (s1, p, o1) → (s1, type, o)
(p, range, o), (s1, p, o1) → (o1, type, o)
(p1, subproperty, p2), (p2, subproperty, p3) → (p1, subproperty, p3)
(p1, subproperty, p2), (s, p1, o) → (s, p2, o)
(s, subclass, o), (s1, type, s) → (s1, type, o)
(s, subclass, o), (o, subclass, o1) → (s, subclass, o1)
(p, domain, o), (o, subclass, o1) → (p, domain, o1)
(p, range, o), (o, subclass, o1) → (p, range, o1)
(p, subproperty, p1), (p1, domain, o) → (p, domain, o)
(p, subproperty, p1), (p1, range, o) → (p, range, o)
```

**Question 2.** Pour chacune des deux traductions, peut-on exprimer en logique les requêtes suivantes ? Si oui, donnez la requête obtenue, ainsi que l'ensemble de réponses à cette requête sur la base de connaissances de la question 1. Si non, expliquez brièvement pourquoi la requête n'est pas exprimable.

1. Quels sont les personnages  $P$  et objets  $O$  tels que  $P$  utilise  $O$  ?
2. Quels sont les personnages  $P$  et types  $T$  tel que  $P$  utilise un objet de type  $T$  et  $T$  est une sous-classe (stricte) de `Objet` ?

## Exercice 3 : Correspondances entre Datalog et UCQ (3 pts)

**Question 1.** Traduire l'UCQ suivante  $Q$  en une requête (ou ensemble de règles) Datalog équivalente. On nommera *ans* le prédicat réponse.

$$Q(x, y) = (\exists z. p(x, z) \wedge r(x, y, z)) \vee (p(x, y) \wedge x = y) \vee s(x, y)$$

**Question 2.** Toute UCQ peut-elle se traduire en une requête Datalog équivalente ? Justifiez votre réponse.

**Question 3.** Toute requête Datalog peut-elle se traduire en une UCQ équivalente ? Justifiez votre réponse.