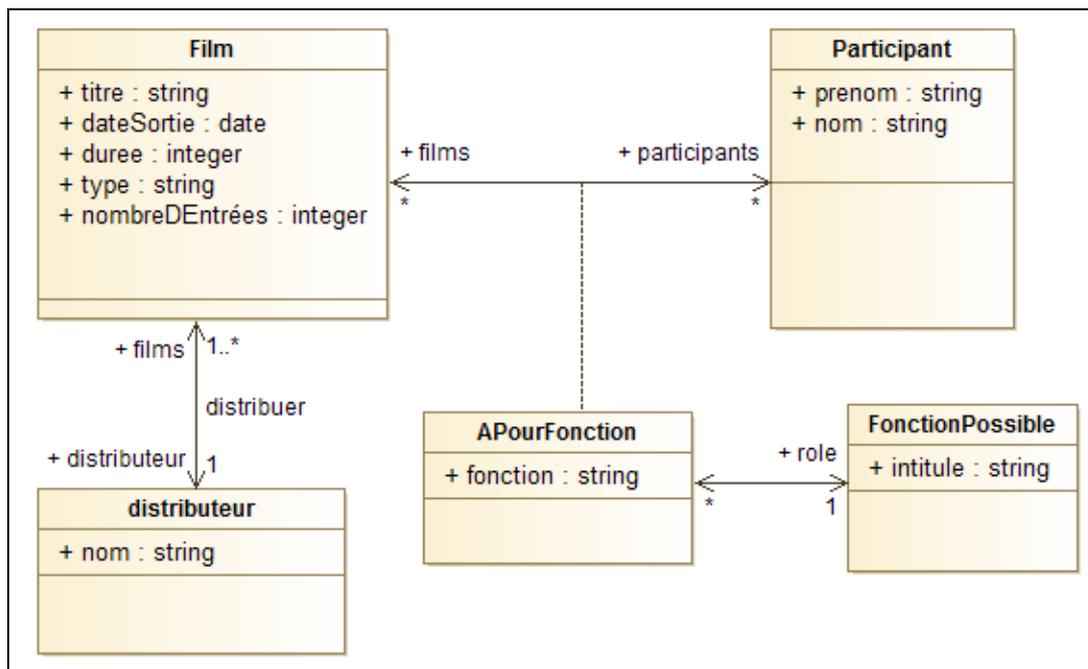


NFP107**EXAMEN – 1^{ère} session – Juin 2018****(Tout document écrit autorisé – Téléphones et ordinateurs portables interdits)**

Nous désirons réaliser une application permettant de gérer une filmothèque. Tous les films nécessitent un ensemble de personnes ayant différentes fonctions : acteur, figurant, compositeur, script, réalisateur, etc. Tout film est distribué par un et un seul distributeur. Un film est caractérisé par son titre, sa date de sortie, sa durée (en minutes), son type : drame, comédie, policier, science fiction et animation. Pour chaque film on mentionnera aussi le nombre d'entrées qu'il a réalisé.



Le modèle relationnel correspondant au diagramme ci-dessus est le suivant :

- FILM (IDF, TITREF, DATESF, DUREEF, TYPEF, NBENTF, #IDD)
- DIST (IDD, NOMD)
- PART (IDP, NOMP, PRENOMP)
- FONC (INTIF)
- APFONC (#IDF #IDP, #FONCTION)

Formats des rubriques :

- ☑ IDF, IDD, IDP : 6 caractères
- ☑ NOMP, PRENOMP, TYPEF : 20 caractères maximum
- ☑ TITREF, NOMD, INTIF, FONCTION : 50 caractères maximum
- ☑ DUREEF : 3 chiffres maximum
- ☑ NBENTF : 8 chiffres maximum
- ☑ DATESF : date

Remarques :

- ☑ Pour chaque réponse, vous veillerez à fournir une solution optimale ;
- ☑ On considère que toutes les relations sont implantées et que les contraintes de clef primaire et de clefs étrangères sont correctement positionnées hormis pour **la relation APFONC qui n'a aucune clef étrangère positionnée** ;
- ☑ **La seule syntaxe acceptée est celle d'Oracle.**

I. Compréhension du schéma (4,50 pts)

1) Ajoutez les clefs étrangères à la relation APFONC. **(1,00 pt)**

alter table APFONC **(0,25 pt)**

add constraint FK_APFONC_IDP foreign key (IDP) references PART (IDP) ;

alter table APFONC **(0,25 pt)**

add constraint FK_APFONC_IDF foreign key (IDF) references FILM (IDF) ;

alter table APFONC **(0,50 pt)**

add constraint FK_APFONC_FONCTION foreign key (FONCTION) references FONC (INTIFP) ;

2) Implantez les contraintes suivantes : **(2,00 pts)**

- Tout film a un et un seul distributeur **(0,50 pt)**

alter table FILM **(0,50 pt)**

modify IDD constraint NN_FILM_IDD not null ;

- TYPEF ne peut prendre que les valeurs **drame, comédie, policier, science fiction** ou **animation** **(0,50 pt)**

alter table FILM **(0,50 pt)**

add constraint CHK_FILM_TYPEF check (TYPEF in ('drame', 'comédie', 'policier', 'science fiction', 'animation')) ;

- TITREF dans la relation FILM est une clef candidate mais non primaire. **(1,00 pt)**

alter table FILM **(0,50 pt)**

modify TITREF constraint NN_FILM_TITREF not null ;

alter table FILM **(0,50 pt)**

add constraint UNQ_FILM_TITREF unique (TITREF) ;

3) Ecrivez la requête SQL qui permet de positionner à 0 la valeur par défaut de l'attribut **NBENTF** de **FILM**. **(0,50 pt)**

alter table FILM

modify NBENTF default 0 ;

- 4) Expliquez en quelques mots pourquoi l'attribut FONCTION fait parti de la clef primaire de la relation APFONC et pourquoi il ne peut jamais être *null*. **(0,50 pt)**

Un participant peut avoir plusieurs rôles dans un film. Par exemple, il peut être à la fois acteur et réalisateur. Il ne peut jamais être null car il fait parti de la clef primaire de la relation APFONC.

- 5) Expliquez en quelques mots l'intérêt de la relation FONC qui est constituée d'un seul et unique attribut. On s'intéressera aussi aux raisons techniques qui peuvent justifier cette relation. **(0,50 pt)**

Cette relation sert uniquement à définir un domaine primaire permettant ainsi de contraindre les valeurs de l'attribut FONCTION. Cette relation fournit les valeurs possibles pour l'attribut FONCTION et évite les erreurs de saisie.

II. Donnez les requêtes suivantes dans le langage SQL (15,50 pts)

- 6) Donnez le titre du ou des films vus par le plus grand nombre de spectateurs **(1,50 pt)**

```
select TITREF (0,50 pt)
from FILM
where NBENTF = ( select max(NBENTF) (1,00 pt)
                from FILM) ;
```

- 7) Donnez l'identifiant des distributeur n'ayant jamais distribué de *comedie*. Vous fournirez une requête pour chaque opérateur : **not in, not exists et minus**. Pour la requête avec **not in**, vous expliquerez clairement pourquoi il faut utiliser une condition **is not null** ou pourquoi elle est inutile ? **(2,00 pts)**

```
select IDD
from DIST
where IDD not in ( select distinct IDD
                  from FILM
                  where TYPEF = 'comedie') ; (0,50 pt)
```

D'après le schéma conceptuel tout film possède obligatoirement un et un seul distributeur. Donc IDD dans la relation FILM n'est jamais null. **(0,25 pt)**

```
select IDD
from DIST
where not exists ( select *
                  from FILM
                  where TYPEF = 'comedie'
                    and DIST.IDD = FILM.IDD) ; (0,75 pt)
```

```
select IDD
from DIST
minus
select distinct IDD
```

from FILM

where TYPEF = 'comedie';

(0,50 pt)

- 8) Ecrivez la requête permettant d'extraire pour chaque film le nom et prénom de son ou de ses scénaristes, son titre et le nom de son distributeur. **Vous n'utiliserez pour cette requête que l'opération de jointure.** Vous donnerez en plus de la version SQL la version en algèbre relationnelle. Vous veillerez à optimiser cette dernière. (2,00 pts)

```
select DIST.NOMD, FILM.TITREF, PART.NOMP, PART.PRENOMP
```

```
from FILM, APFONC, PART, DIST
```

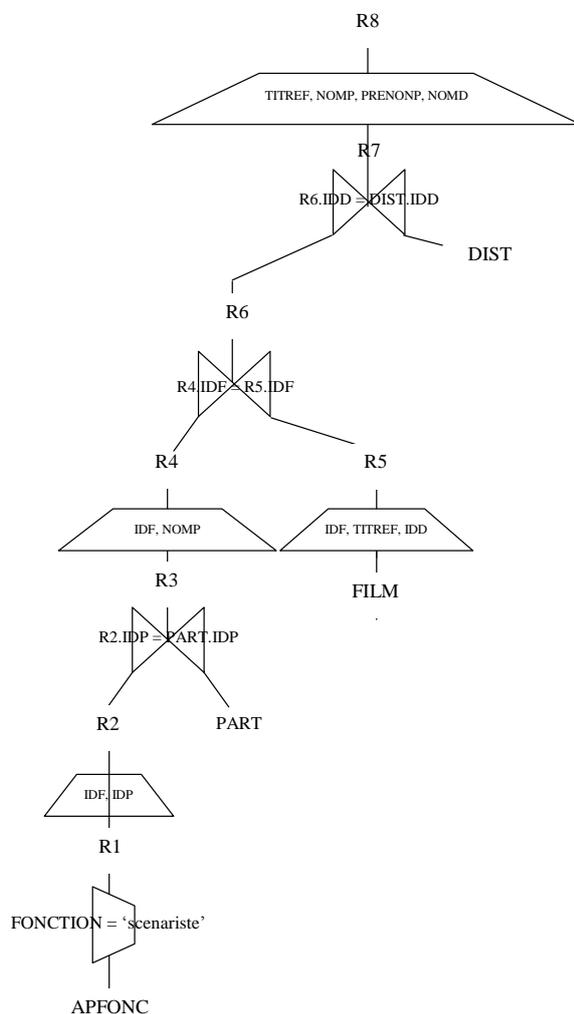
```
where FILM.IDD = DIST.IDD
```

```
and FILM.IDF = APFONC.IDF
```

```
and APFONC.IDP = PART.IDP
```

```
and FONCTION = 'scenariste';
```

(1,00 pt)



(1,00 pt)

- 9) Donnez le nombre moyen de spectateurs pour les comédies (0,75 pt)

```
select avg(NBENTF)
```

```
from FILM
```

```
where TYPEF = 'comedie';
```

10) Donnez l'identifiant et le titre des films ayant les mêmes scénaristes que le film d'identifiant « A10100 ». **(1,50 pts)**

```
select IDF, TITREF (0,50 pt)
```

```
from FILM
```

```
where not exists ( select IDP (0,50 pt)
```

```
from APFONC
```

```
where FONCTION = 'scenariste'
```

```
and IDF = 'A10100'
```

```
minus
```

```
select IDP (0,50 pt)
```

```
from APFONC
```

```
where APFONC.IDF = FILM.IDF
```

```
and FONCTION = 'scenariste')
```

```
and IDF != 'A10100'
```

11) Donnez l'identifiant du film ayant la plus longue distribution (celui qui a le plus de participants). **(1,50 pt)**

```
select IDF (0,50 pt)
```

```
from APFONC
```

```
group by IDF
```

```
having count(*) = ( select max(count(*)) (1,00 pt)
```

```
from APFONC
```

```
group by IDF) ;
```

12) Donnez pour chaque type de film le nombre moyen de spectateurs. **(1,00 pt)**

```
select TYPEF, avg(NBENTF)
```

```
from FILM
```

```
group by TYPEF ;
```

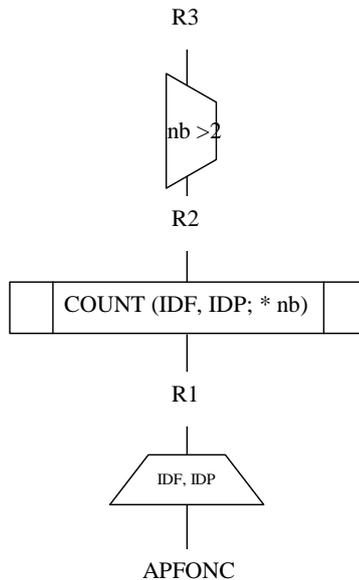
13) Donnez l'identifiant des participants ayant eu plusieurs fonctions distinctes dans un même film. Par exemple, Clint Eastwood peut être à la fois acteur et réalisateur de film *Million Dollar Baby*. Cette requête retourne aussi l'identifiant du ou des films correspondants. **Vous donnerez en plus de la version SQL la version en algèbre relationnelle.** Vous veillerez à optimiser cette dernière. **(2,25 pt)**

```
select IDF, IDP
```

```
from APFONC
```

```
group by IDF, IDP
```

```
having count(*) > 1 ; (1,25 pt)
```



(1,00 pt)

14) Quels sont les index positionnés automatiquement dans les cinq relations. Vous justifierez votre réponse. **(1,50 pt)**

- Dans FILM : IDF car c'est une clef primaire ; **(0,25 pt)**
- TITREF car contrainte unique **(0,25 pt)**
- Dans DIST : IDD car c'est une clef primaire ; **(0,25 pt)**
- Dans APFONC : (IDF, IDP, FONCTION) car c'est une clef primaire ; **(0,25 pt)**
- Dans PART : IDP car c'est une clef primaire. **(0,25 pt)**
- Dans FONC : INTIF car c'est une clef primaire ; **(0,25 pt)**

15) Considérons que nous utilisons Oracle 10g. Dans le plan d'exécution d'une requête exprimée sur des relations, on peut en particulier trouver les actions : sort, hash, hash join. Le plan d'exécution des requêtes suivantes contient-il ces actions ? **(1,50 pt)**

Mettre O (Oui) ou N (Non) dans CHAQUE case	sort	hash	hash join	
<i>select INTIF from FONC where not exists (</i> <i>select * from APFONC where INTIF = FONCTION)</i> Attention : on suppose qu'aucun index n'est positionné	N	N	O	(0,25 pt)
<i>select INTIF from FONC, APFONC</i> <i>where INTIF = FONCTION (+) and IDF is null</i> Attention : on suppose qu'aucun index n'est positionné	N	O	O	(0,50 pt)
<i>select INTIF from FONC</i> <i>minus</i> <i>select FONCTION from APFONC</i>	O	N	N	(0,25 pt)
<i>select INTIF from FONC, APFONC</i> <i>where INTIF = FONCTION (+) and IDF is null</i> Attention : on suppose que tous les index utiles et nécessaires sont correctement positionnés	N	O	N	(0,25 pt)

- Quel est le résultat des requêtes proposées dans le tableau ci-dessus ?

Toutes ces requêtes extraient les FONCTION qui n'apparaissent dans aucun films. **(0,25 pt)**