TP MySQL N° 4

Licence 2 Informatique

Université du Littoral Côte d'Opale

Document à rendre :

un rapport de TP qui contient

les commandes utilisées,

avec description et références à l'énoncé,

et commentaires éventuels.

Introduction à l'algèbre relationnelle et au SQL - TPSQL4

1 Objectifs d'apprentissage

1. mettre en application
   (a) les jointures externes
   (b) requêtes imbriquées, en général, et appliquées à
      i. l'intersection
      ii. la différence
      iii. la division
   (c) requêtes union

Attention
Lisez attentivement les explications et les consignes de travail.

2 Préparer l'environnement de travail

(se reporter aux supports TP1)

Attention
Remarque : vous conserverez une trace numérique de toutes les actions réalisées dans le fichier de scripts. Cela vous permettra de relancer des ensembles de commandes en une seule fois. Vous remettrez le fichier à la fin de chaque séance.

3 Jointure internes : non-equi jointures

3.1 A réaliser

En utilisant le support SQL au chapitre traitant des ordres DQL, plus précisément celui relatif aux jointures externes, proposer les requêtes permettant de répondre aux demandes suivantes :

1. equi-jointure : lister pour chaque nom de réalisateur, son année de naissance, les noms et l'année de naissance des acteurs qui sont nés la même année, classé par nom de réalisateur et année de naissance d'acteur croissante

r1	nom_real	annais_real	nom_act	annais_act
	Andreas Dresen	1963	Olivier Gourmet	1963
	Antonello Grimaldi1955	1955	Isabelle Adjani	1955
	...	...	...	...
	Clint Eastwood	1930	Clint Eastwood	1930
	Joann Sfar	1971	Guillaume Depardieu	1971
	Ursula Meier	1971	Guillaume Depardieu	1971

nombres de lignes : 8

2. non-equi-jointure : lister pour chaque nom de réalisateur, son année de naissance, les noms et l'année de naissance des acteurs qui sont plus jeunes que lui, classé par nom de réalisateur et année de naissance d'acteur croissante

r2	nom_real	annais_real	nom_act	annais_act
	Andreas Dresen	1963	Isabelle Ferrari	1964
	Andreas Dresen	1963	Robert Downer jr	1965
	...	...	...	...
	Ursula Meier	1971	Salomé Stéverin	1985
	Ursula Meier	1971	Melissa Rodrigues	1988
	Ursula Meier	1971	Leora Barbara	1996

nombres de lignes : 221

4 Jointures externes

4.1 A réaliser

En utilisant le support SQL au chapitre traitant des ordres DQL, plus précisément celui relatif aux jointures externes, proposer les requêtes permettant de répondre aux demandes suivantes :

1. jointure interne : lister les réalisateurs et les titres des films qu'ils ont réalisés, classé par titre de film

r4	nom_real	titre_film
	Antonello Grimaldi1955	Caos Calmo
	René Féret	Comme une étoile dans la nuit
	...	...
	Edward Zwick	The Last Samurai
	Ben Stiller	Tonnere sous les tropiques
	Hong Sang-soo	Woman on the beach

nombres de lignes : 15

2. combien de réalisateurs différents ont réalisé des films ?

r4	count(distinct num_real)
	13

nombres de lignes : 1

3. compter les réalisateurs dans la table 'realisateur' : d'où provient l'écart avec le comptage précédent ?

r5	NbreDeRealisateurs
	14

nombres de lignes : 1

Remarque

La jointure interne n'a donc repris que les réalisateurs qui ont pu être joints avec un film (un réalisateur n'a pas réalisé de film : on ne le retrouve pas dans la liste ; certains réalisateurs ont réalisé plus d'un film : on les retrouve plusieurs fois)

4. jointure externe : lister TOUS les réalisateurs et (éventuellement) les titres des films qu'ils ont réalisés, classé par titre de film

r6	nom_real	titre_film
	Joann Sfar	null
	Antonello Grimaldi1955	Caos Calmo
	...	...
	Edward Zwick	The Last Samurai
	Ben Stiller	Tonnere sous les tropiques
	Hong Sang-soo	Woman on the beach

nombres de lignes : 16

5. idem. précédent mais avec l'indication '– pas de film pour ce realisateur –' dans le cas où, pour un réalisateur, le titre de film n'existe pas (est nul)
   — Indice : la fonction `coalesce` peut vous aider

r7	nom_real	letitre
	Joann Sfar	(– pas de film pour ce realisateur –)
	Antonello Grimaldi1955	Caos Calmo
	...	...
	Edward Zwick	The Last Samurai
	Ben Stiller	Tonnere sous les tropiques
	Hong Sang-soo	Woman on the beach

nombres de lignes : 16

Remarque

La jointure externe conserve

- les lignes des 2 tables qui ont pu être jointes (=jointure interne),
- mais conserve aussi les lignes qui n'ont pu être jointes d'une des 2 tables (gauche ou droite, ou des 2 tables si c'est une jointure externe complète)

5 Jointures internes et jointures externes

5.1 A réaliser

En utilisant le support SQL au chapitre traitant des ordres DQL, plus précisément celui relatif aux jointures externes, proposer les requêtes permettant de répondre aux demandes suivantes :

1. version 1 : lister tous les noms des acteurs et éventuellement les titres des films dans lesquels ils ont joué, classé par titre et par nom
   — Indice : cette requête ne produit pas le bon résultat : saurez-vous trouver l'anomalie et apporter une correction pour la version 2 plus bas ?
   — Indice : 1 + 2 * 3, (1+2)*3 ou 1+(2*3) ? (par défaut c'est la 2ème mais ça n'est pas forcément ce qui est souhaité)

r11	nom_act	letitre
	Alessandro Gassman	Caos Calmo
	Nanni Moretti	Caos Calmo
	...	...
	Jack Black	Tonnere sous les tropiques
	Robert Downer jr	Tonnere sous les tropiques
	Tom Cruise	Woman on the beach

nombres de lignes : 32

2. version 2 : lister tous les noms des acteurs et éventuellement les titres des films dans lesquels ils ont joué, classé par titre et nom d'acteur

r12	nom_act	letitre
	Clint Eastwood	(– pas de film pour cet acteur –)
	Isabelle Adjani	(– pas de film pour cet acteur –)
	...	...
	Jack Black	Tonnere sous les tropiques
	Robert Downer jr	Tonnere sous les tropiques
	Tom Cruise	Woman on the beach

nombres de lignes : 35

3. lister tous les noms des acteurs et éventuellement les numéros des salles dans lesquelles on a pu les voir, classé par salle puis nom d'acteur

r13	nom_act	lasalle
	Clint Eastwood	(– jamais vu cet acteur –)
	Isabelle Adjani	(– jamais vu cet acteur –)
	...	...
	Melissa Rodrigues	5
	Olivier Gourmet	5
	Tom Cruise	5

nombres de lignes : 71

6 Sous-requêtes indépendantes et sous-requêtes corrélées

En utilisant le support SQL au chapitre traitant des ordres DQL, plus précisément celui relatif aux requêtes imbriquées, proposer les requêtes permettant de répondre aux demandes suivantes :

6.1 Sous-requêtes : principe général

6.1.1 A réaliser

1. pour lister les acteurs qui sont plus jeunes que Sophie Marceau :
   (a) d'abord, récupérer *l'année de naissance de Sophie Marceau* :

r20	anNais_act
	1966

nombres de lignes : 1

   (b) puis, lister les acteurs dont l'année de naissance est supérieure à *l'année de naissance de Sophie Marceau* :

r21	num_act	nom_act	anNais_act
	3	Judith Henry	1968
	4	Angelina Jolie	1975
	...	...	...
	21	Melissa Rodrigues	1988
	22	Benjamin Biolay	1973
	23	Guillaume Depardieu	1971

nombres de lignes : 10

(c) on peut aussi répondre à cette question en utilisant une non-équi-auto jointure !

r22	num_act	nom_act	anNais_act
	3	Judith Henry	1968
	4	Angelina Jolie	1975
	...	...	...
	21	Melissa Rodrigues	1988
	22	Benjamin Biolay	1973
	23	Guillaume Depardieu	1971

nombres de lignes : 10

2. lister les films dont le réalisateur est né en 1971

(a) d'abord, récupérer *le numéro du réalisateur né en 1971* :

r23	num_real
	4
	12

nombres de lignes : 2

(b) puis, lister les films dont le numéros de réalisateur est *le numéro d'un réalisateur né en 1971* :

```
select *
  from film
  where num_real = (
    select num_real
     from realisateur
     where anNais_real = 1971
    )
;
```

(c) la requête produit une erreur :

Attention

Le résultat envoyé par une requête imbriqué doit être conforme à l'opérateur qui va utiliser le résultat de cette requête

- un des opérateurs de comparaison : la sous-requête doit renvoyer 1 seule valeur, ou `between`
- plusieurs valeurs uniques : `in` (ou un opérateur de comparaison associé à `all` ou `any`)
- un jeu de résultat variable : opérateur `exists`

(d) proposer la requête et l'opérateur adéquat pour la question précédente

r25	num_film	titre_film	anSortie_film	budget_film	genre_film	num_real	resume_film
	7	Home	2008	3000000	film documentaire	4	

nombres de lignes : 1

(e) essayer la même requête avec 2000 comme année de naissance

r25b	num_film	titre_film	anSortie_film	budget_film	genre_film	num_real	resume_film

nombres de lignes : 0

6.2 Sous-requêtes indépendantes

6.2.1 A réaliser

En utilisant le support SQL au chapitre traitant des ordres DQL, plus précisément celui relatif aux requêtes imbriquées, proposer les requêtes permettant de répondre aux demandes suivantes :

1. Lister les acteurs et les films dans lesquels ils ont joués, avec le nom d'acteur, le titre du film, le pourcentage du cachet perçu par rapport au total général des cachets, classé par nom d'acteur et numéro de film :
   (a) d'abord, récupérer le *total des cachets perçus* :

r37	sum(cachet_jouer)
	1758995

nombres de lignes : 1

   (b) puis, lister les noms d'acteurs, les titres de films, le rapport du cachet au *total des cachets perçus* :

r38	nom_act	titre_film	cachet_jouer	pourcentageDuTotal
	Alessandro Gassman	Caos Calmo	12770	0.7260
	André Dussolier	Le crime est notre affaire	41578	2.3637
	...	...	...	...
	Tom Cruise	Woman on the beach	90487	5.1442
	Ursula Werner	Le septieme ciel	47557	2.7036
	Valeria Golino	Caos Calmo	30526	1.7354

nombres de lignes : 32

6.3 Sous-requêtes corrélées

6.3.1 A réaliser

En utilisant le support SQL au chapitre traitant des ordres DQL, plus précisément celui relatif aux requêtes imbriquées, proposer les requêtes permettant de répondre aux demandes suivantes :

1. Lister les acteurs et les films dans lesquels ils ont joués, avec le nom d'acteur, le titre du film, le pourcentage du cachet perçu par rapport au total général des cachets, classé par nom d'acteur et numéro de film :
   (a) d'abord, récupérer le *total des cachets perçus de l'acteur ?...* (? devra être remplacé par une valeur de la requête principale...) :

```
select sum(cachet_jouer)
  from jouer
  where num_act = ?;
```

   (b) puis, lister les noms d'acteurs, les titres de films, le rapport du cachet au *total des cachets perçus de l'acteur* :

r40	nom_act	titre_film	cachet_jouer	totalCachetActeur
	Alessandro Gassman	Caos Calmo	12770	100.0000
	André Dussolier	Le crime est notre affaire	41578	100.0000
	...	...	...	...
	Tom Cruise	Woman on the beach	90487	39.8421
	Ursula Werner	Le septieme ciel	47557	100.0000
	Valeria Golino	Caos Calmo	30526	100.0000

nombres de lignes : 32

7 Intersection et requêtes imbriquées

7.1 A réaliser

En utilisant le support SQL au chapitre traitant des ordres DQL, plus précisément celui relatif aux requêtes imbriquées (et à l'intersection), proposer les requêtes permettant de répondre aux demandes suivantes :

7.1.1 avec l'opérateur `in`

1. lister les réalisateurs s'il existe des films de ces réalisateurs qui ont été projetés salle 1
   (a) d'abord : lister *les numéros des réalisateurs dont les films ont été projetés salle 1*, classé par numéro

r41	num_real
	1
	2
	...
	9
	10
	11

nombres de lignes : 6

   (b) puis : lister réalisateurs dont le numéro se trouve dans la liste *des numéros des réalisateurs dont les films ont été projetés salle 1*, classé par numéro

r42	num_real	nom_real	anNais_real
	1	Edward Zwick	1952
	2	Clint Eastwood	1930
	...	...	...
	9	Hong Sang-soo	1960
	10	Anna Novion	1979
	11	Antonello Grimaldi1955	1955

nombres de lignes : 6

7.1.2 avec l'opérateur `exists`

1. lister les réalisateurs s'il existe des films de ces réalisateurs qui ont été projetés salle 1
   (a) d'abord : lister *les projections de films du réalisateur ? en salle 1*

```
select *
    from film inner join projeter
        using (num_film)
    where num_salle = 1
       and num_real = ?
;
```

   (b) puis : lister réalisateurs s'il existe *des projections de films du réalisateur ? en salle 1*, classé par numéro

r44	num_real	nom_real	anNais_real
	1	Edward Zwick	1952
	2	Clint Eastwood	1930
	...	...	...
	9	Hong Sang-soo	1960
	10	Anna Novion	1979
	11	Antonello Grimaldi1955	1955

nombres de lignes : 6

8 Différence et requêtes imbriquées

8.1 A réaliser

En utilisant le support SQL au chapitre traitant des ordres DQL, plus précisément celui relatif aux requêtes imbriquées (et à la différence), proposer les requêtes permettant de répondre aux demandes suivantes :

8.1.1 avec l'opérateur `not in`

1. lister les réalisateurs dont aucun de leurs films n'a été projeté salle 1
   (a) d'abord : lister *les numéros des réalisateurs dont les films ont été projetés salle 1*

r45	num_real
	1
	2
	...
	9
	10
	11

nombres de lignes : 6

   (b) puis : lister les réalisateurs dont le numéro NE se trouve PAS dans la liste *des numéros des réalisateurs dont les films ont été projetés salle 1*, classé par numéro

r46	num_real	nom_real	anNais_real
	3	Ben Stiller	1965
	4	Ursula Meier	1971
	...	...	...
	12	Joann Sfar	1971
	13	Claude Pinoteau	1925
	14	Georges Lucas	null

nombres de lignes : 8

8.1.2 avec l'opérateur `not exists`

1. lister les réalisateurs dont aucun de leurs films n'a été projeté salle 1
   (a) d'abord : lister *les films du réalisateur X qui ont été projetés salle 1*

```
select *
    from film inner join projeter
        using (num_film)
    where num_salle = 1
      and num_real = ?
;
```

(b) puis : lister les réalisateurs s'il N'existe PAS *des films du réalisateur X qui ont été projetés salle 1*, classé par numero

r48	num_real	nom_real	anNais_real
	3	Ben Stiller	1965
	4	Ursula Meier	1971
	...	...	...
	12	Joann Sfar	1971
	13	Claude Pinoteau	1925
	14	Georges Lucas	null

nombres de lignes : 8

9 Division et requêtes imbriquées

9.1 A réaliser

En utilisant le support SQL au chapitre traitant des ordres DQL, plus précisément celui relatif aux requêtes imbriquées (et à la division), proposer les requêtes permettant de répondre aux demandes suivantes :

1. lister les films qui ont été projetés dans toutes les salles (où sont projetés des films...)

   (a) lister les films tel qu'il n'existe pas de salles (dans lesquelles des projections ont eu lieu) dans lesquelles ils n'y ont pas été projetés

r49	num_film	titre_film	anSortie_film	budget_film	genre_film	num_real	resume_film
	1	The Last Samurai	2003	8000000	épopée	1	
	9	L'échange	2008	5000000	Drame	2	

nombres de lignes : 2

   (b) lister les films tels que le nombre de salles différentes dans lesquelles ils ont été projetés correspond au nombre total de salles (dans lesquelles des projections ont lieu)

r50	num_film	titre_film	anSortie_film	budget_film	genre_film	num_real	resume_film
	1	The Last Samurai	2003	8000000	épopée	1	
	9	L'échange	2008	5000000	Drame	2	

nombres de lignes : 2

10 Union

10.1 A réaliser

En utilisant le support SQL au chapitre traitant des ordres DQL, plus précisément celui relatif à l'opération d'union, proposer les requêtes permettant de répondre aux demandes suivantes :

1. lister les noms des réalisateurs et les noms des acteurs

r51	noms
	Alessandro Gassman
	André Dussolier
	...
	Ursula Meier
	Ursula Werner
	Valeria Golino

nombres de lignes : 41

2. lister les noms des réalisateurs et les noms des acteurs, en les distinguant

r52	noms	type
	Alessandro Gassman	acteur
	André Dussolier	acteur
	...	...
	Ursula Meier	realisateur
	Ursula Werner	acteur
	Valeria Golino	acteur

nombres de lignes : 43

3. d'où provient cette différence de comptage ?

11 Les vues stockées

Une vue est une requête pré-enregistrée en tant qu'objet dans une base de données (tout comme une table). Son contenu est réactualisé à chaque interrogation.

11.1 A réaliser

Testez la création et l'utilisation d'une vue avec MySQL.

11.1.1 Créer une vue

Créer la vue suivante :

```
CREATE VIEW pilotesAvecVol
AS
SELECT DISTINCT numpil
 FROM pilote INNER JOIN vol USING (numpil)
;
```

11.1.2 Utiliser une vue

Utilisez la vue créée précédemment :
Lister les numéros des pilotes ayant volé :

```
SELECT *
  FROM pilotesAvecVol
  ORDER BY numpil;
```

Lister les pilotes ayant volé :

```
SELECT *
  FROM pilote
  WHERE numpil IN (
     SELECT numpil FROM pilotesAvecVol
  )
  ORDER BY numpil;
```

12 Fin

En cas de soucis, envoyez vos questions à adeel.ahmad@univ-littoral.fr

vous devez commencer à sentir sous vos doigts la puissance de SQL...mais ils faut dompter cette puissance en comprenant parfaitement les mécanismes sous-jacents ! (ici c'est votre intelligence qui entre en action)