



Module M4

Base de données

TD n°5
Correction

SQL
Requêtes Imbriquées

Auteurs : Laura Monceaux / Véronique Laime



ATTENTION aux Requêtes imbriquées : vérifier toujours ce que retourne la requête imbriquée (une valeur, un ensemble de valeurs, un t-uple, un ensemble de t-uple, 0 ou n t-uples), information utile pour choisir l'opérateur que vous pouvez utiliser.

Question 1 Les acteurs (nom, prénom, âge) qui ont joué dans un film pour un salaire supérieur à celui de *Nicole Kidman* dans *Australia*

Il faut chercher dans un premier temps le salaire de Nicole Kidman dans *Australia* (RI) puis comparer les salaires des acteurs à ce salaire.

La RI retourne une valeur (utilisation possible de la condition `att > (valeur)`).

```
select distinct nom, prenom, age
from Acteur Natural Join Jouer
where salaire > (select salaire
                 from (Acteur Natural Join Jouer) Join Film on id=id_film
                 where nom='Kidman' and prenom='Nicole' and titre='Australia');
```

remarque : On suppose que N. Kidman n'a joué que dans un film de titre *Australia* (1 seul salaire)

Question 2 Les films (titre, genre) dont le salaire d'un de ces acteurs est le plus élevé (2 versions)

1ère solution : on recherche d'abord le salaire le plus élevé de la base (RI) puis on compare chaque salaire à ce nombre. La RI retourne une valeur.

```
select distinct id, titre, genre
from Jouer join Film on id=id_film
where salaire > (select max(salaire) from jouer);
```

2ème solution : on compare chaque salaire à l'ensemble des salaires (RI) et on ne le garde que s'il est plus grand ou égal à tous les autres. La RI retourne un ensemble de valeurs.

```
select distinct id, titre, genre
from Jouer join Film on id=id_film
where salaire >= ALL (select salaire from jouer);
```

Question 3 Films ayant été diffusé dans tous les cinémas de Loire Atlantique

On regarde dans combien de cinéma est passé le film et on compare ce nombre au nombre de cinéma (RI – une seule valeur retournée).

```
select id_film
from Projection
group by id_film
having count(distinct id_cine) = (select count(*) from Cinema);
```

Question 4 Cinémas (nom, ville) ayant fait le plus d'entrées en 2008

Pour chaque cinéma, on calcule le nombre d'entrées réalisées en 2008 et on compare ce nombre à l'ensemble des entrées de chaque cinéma (RI – ens de valeurs).

```

select id_cine, nom, ville
from Projection Natural Join Cinema
where extract YEAR from dateP = 2008
group by id_cine, nom, ville
having sum(nbentrees) >= all (select sum(nbentrees)
                             from Projection
                             group by id_cine);

```

Question 5 Acteurs ayant un cachet inférieur à la moyenne en 2008. On affichera également les acteurs n'ayant rien touché en 2008

Attention pour pouvoir afficher les acteurs n'ayant rien touché : il faut utiliser une jointure externe --- ne pas oublier de faire la sélection avant sur l'année du film.

Remarque : NV(att,val) permet d'affecter la valeur val à att quand att est égal à NULL.

```

select distinct A.id_act
from Acteur A Left Outer Join
  (select * from Jouer join Film on id=id_film where annee=2008) J
on A.id_act = J.id_act
where NVL(salaire,0) < (select AVG(salaire) from Jouer Join Film on id=id_film where
annee=2008);

```

Question 6 Films ayant plus de recettes que de dépenses. On ne considère ici que les salaires des acteurs comme dépense et les entrées dans les cinémas comme recettes (on comptera tarif réduit pour toutes les entrées)

Tout d'abord on calcule dans la requête principale les recettes de chaque film que l'on doit comparer aux dépenses réalisées pour LE MEME FILM (nécessité d'une corrélation dans la RI : P.id_film = J.id_film). On sélectionne dans la requête imbriquée les tuples concernant le groupe pour lequel on fait la comparaison.

```

select id_film
from Projection P Natural Join Cinema
group by id_film
having sum(nbentrees * tarifR) > (select sum(salaire)
                                 from Jouer J
                                 where P.id_film = J.id_film);

```

Question 7 Films dans lequel jouent au moins 10 acteurs

1ere solution : la RI retourner l'ensemble des films ayant moins de 10 acteurs.

```

select id
from Film
where id in (select id_film from Jouer group by id_film having count(id_act) > 9);

```

2ème solution : pour chaque film de la table FILM, on évalue la RI --- obligatoirement corrélée (id=id_film) – on retourner les identifiants des films pour lesquels la RC retourne un résultat (au moins un tuple).

```

select id
from Film
where exists (select id_film from Jouer where id=id_film having count(id_act) > 9);

```

Question 8 Villes dans lesquelles aucun cinéma n'a diffusé le film *Australia* (2 versions)

1ère solution : la RI retourne un ensemble de valeurs (noms de villes)

```
select distinct ville
from Cinema
where ville not in (select distinct ville
                    from (Projection Natural Join Cinema) join Film on id=id_film
                    where titre ='Australia' );
```

2ème solution : Requête corrélée avec utilisation d'un not exists – Pour chaque Cinéma (C1), on calcule la RI et on ne garde que les cinémas pour lesquels la RI n'a pas retourné de résultats.

```
select distinct ville
from Cinema C1
where not exists (select *
                  from (Projection Natural Join Cinema C2) join Film on id=id_film
                  where titre ='Australia' and C1.ville = C2.ville);
```

Question 9 Acteurs (id_act, nom, prénom) qui ont joué dans un des films réalisées en 2008 ayant fait le plus d'entrées et pour lequel ils ont reçu leur plus gros salaire

Pour chaque film joué pour un acteur, on regarde si ce film a fait la plus d'entrées (1ère RI) et on regarde si l'acteur a obtenu son plus gros salaire dans ce film (2ème RI : requête corrélée).

```
select id_act, nom, prenom
from Acteur A Natural Join Jouer
where id_film in (select id_film
                  from Projection Join Film on id=id_film
                  where annee=2008
                  group by id_film
                  having sum(nbentrees) >= all (select sum(nbentrees)
                                                from Projection Join Film on id=id_film
                                                where annee=2008
                                                group by id_film))
and salaire >= all (select salaire from Jouer J where A.id_act = J.id_act);
```

Question 10 Pour chaque jour du mois de mars 2009, retournez le cinéma (ou les cinémas) qui a (ou ont) fait le plus d'entrées ce jour là. (résultat trié par date, puis par cinéma)

```
select dateP, id_cine
from Projection P1
where extract MONTH from dateP = 3 and extract YEAR from dateP = 2009
group by dateP, id_cine
having sum(nbentrees) >= all (select sum(nbentrees)
                              from Projection P2
                              where P1.dateP=P2.dateP
                              group by id_cine)
order by dateP, id_cine;
```

Question 11 Films qui ont coûté les plus chers (en terme de salaire) et qui n'ont pas été projetés

dans un des cinémas de Nantes

```
select id_film
from Jouer
where id_film not in (select distinct id from Projection Natural Join Cinema where ville='Nantes')
group by id_film
having sum(salaire) >= (select sum(salaire) from Jouer group by id_film);
```