

Révisions Bases de données — Correction

I. Les films, leur année et leur note

Q1. Listez les films de l'année 1980 qui ont obtenu une note ≥ 7 .

```
SELECT * FROM movie WHERE yr = 1980 AND score >= 7
```

Q2. Combien de films dans la table movie ont une note ≥ 7 ?

```
SELECT COUNT(*) FROM movie WHERE score >= 7
```

Q3. Donner les cinq années qui ont le plus de films répertoriés.

```
SELECT yr, COUNT(*) AS c FROM movie
GROUP BY yr ORDER BY c DESC LIMIT 5
```

Q4. Dressez la liste des films dont le titre contient le mot ' < world > '.

```
SELECT title FROM movie WHERE title LIKE '%world%'
```

Q5. Dressez la liste des 10 films au titre le plus court (utiliser LENGTH).

```
SELECT title FROM movie ORDER BY LENGTH(title) limit 10
```

Et si on veut en plus les classer par année on peut faire :

```
SELECT title, yr FROM movie ORDER BY LENGTH(title) ASC, yr DESC LIMIT 10
```

Q6. Quels sont les 10 films les mieux notés de tous les temps?

```
SELECT title,score,yr FROM movie ORDER BY score DESC LIMIT 10
```

Mais si on ne veut pas oublier les ex-aequo :

```
SELECT title,score,yr FROM movie WHERE score IN
(SELECT score FROM movie ORDER BY score DESC LIMIT 10 )
ORDER BY score DESC
```

Q7. Quel est l'année du plus ancien film répertorié? et du plus récent?

```
SELECT MIN(yr), MAX(yr) FROM movie
```

Q8. Quel est le film sur lequel le plus de personnes a voté? et le moins de personnes?

```
SELECT title,votes FROM movie ORDER BY votes DESC LIMIT 1
SELECT title,votes FROM movie ORDER BY votes LIMIT 1
```

et pour afficher les ex-aequo éventuels :

```
SELECT title,votes FROM movie WHERE votes = (SELECT MAX(votes) FROM movie)
```

Q9. Quelle est la note du film le moins apprécié?

```
SELECT MIN(score) FROM movie
```

Q10. Quel(s) film(s) a obtenu cette plus mauvaise note ?

```
SELECT title,score FROM movie ORDER BY score LIMIT 1
```

et si il y a des ex-aequo :

```
SELECT title, score FROM movie WHERE score = (SELECT min(score) FROM movie )
```

Q11. Pour chacune des années, donnez le ou les meilleurs films (le ou les films qui ont obtenu le meilleur score).

```
SELECT yr,title,score FROM movie WHERE (yr,score) IN
(SELECT yr,max(score) FROM movie GROUP BY yr) ORDER BY yr
```

II. Les réalisateurs

Q12. Combien de réalisateurs (différents) ont produit les films de la table movie ?

```
SELECT COUNT(*) FROM (SELECT * FROM movie GROUP BY director)
SELECT COUNT(DISTINCT director) FROM movie
```

Q13. Listez tous les films réalisés par Steven Spielberg.

```
SELECT * FROM movie JOIN actor ON movie.director=actor.id
WHERE actor.name='Steven Spielberg'
```

ou bien :

```
SELECT * FROM movie, actor
WHERE movie.director=actor.id AND actor.name='Steven Spielberg'
```

Q14. Listez les 10 réalisateurs les plus prolifiques (les noms, et non pas les numéros!).

```
SELECT COUNT(*) AS c,actor.name FROM movie
JOIN actor ON movie.director = actor.id
GROUP BY movie.director ORDER BY c DESC LIMIT 10
```

ou bien :

```
SELECT COUNT(*) AS c,actor.name FROM movie, actor
WHERE movie.director = actor.id
GROUP BY movie.director ORDER BY c DESC LIMIT 10
```

Si l'on veut les ex-aequo, cela se complique :

```
SELECT COUNT(*) AS c, actor.name FROM movie , actor
WHERE movie. director = actor.id GROUP BY movie.director
HAVING c IN
(SELECT COUNT(*) AS c1 FROM movie GROUP BY director ORDER BY c1 DESC LIMIT 10)
```

Q15. Qui est le meilleur réalisateur des années 70 ? (celui qui a le meilleur score moyen). Attention : il n'est pas pertinent de faire la moyenne des scores, puisqu'il n'y a pas le même nombre de votants pour chaque film. Faire plutôt une moyenne pondérée, en considérant le nombre de votants.

```
SELECT SUM(score*votes)/SUM(votes) AS S,actor.name FROM movie, actor
WHERE movie.director = actor.id AND 1970<=yr AND yr<1980
GROUP BY movie.director ORDER BY S DESC LIMIT 1
```

III. Les acteurs

Q16. Combien d'acteurs dans la base de données? (un acteur est une personne de la table actor qui a joué dans au moins un film)?

```
SELECT COUNT(*) FROM (SELECT * FROM casting GROUP BY actorid)
SELECT COUNT(DISTINCT actorid) FROM casting
```

Q17. Donner, pour chaque film répertorié, le nom de l'acteur qui a le premier rôle.

```
SELECT movie.title, actor.name FROM movie
JOIN casting ON casting.movieid=movie.id
JOIN actor ON casting.actorid=actor.id
WHERE casting.ord=1
```

ou :

```
SELECT movie.title, actor.name FROM movie, casting, actor
WHERE casting.movieid=movie.id AND casting.actorid=actor.id AND casting.ord=1
```

Q18. Donnez, pour chaque acteur, l'année du dernier film.

```
SELECT a.name, max(yr) FROM actor AS a, casting AS c, movie AS m
WHERE c.actorid=a.id AND m.id=c.movieid
GROUP BY a.id, a.name
```

Q19. Quel sont les 10 acteurs qui ont joué dans le plus de films? (les noms, pas les numéros!)

```
SELECT actor.name, COUNT(*) AS C FROM actor, casting
WHERE actor.id=casting.actorid
GROUP BY actorid ORDER BY C DESC LIMIT 10
```

Q20. Quels sont les acteurs les plus endurants? (les acteurs dont le délai écoulé entre le premier film et le dernier film est le plus long) .

```
SELECT name, max(yr)-min(yr) AS d, casting.actorid AS i2 FROM movie, casting, actor
WHERE movie.id = casting.movieid AND actor.id=casting.actorid
GROUP BY i2 ORDER BY d DESC LIMIT 10
```

Q21. Donnez la liste des acteurs qui ont tourné entre 1940 et 1950.

```
SELECT actor.name FROM actor
JOIN casting ON actor.id=casting.actorid
JOIN movie ON movie.id=casting.movieid
WHERE yr<=1950 AND yr>=1940
GROUP BY actor.id
```

Q22. Dressez la liste des films dans lesquels a joué Gérard Depardieu (orthographié sans accents).

```
SELECT * FROM movie
JOIN casting ON movie.id = casting.movieid
JOIN actor ON casting.actorid = actor.id
WHERE actor.name = 'Gerard Depardieu'
```

ou plus simplement a priori :

```
SELECT * FROM movie, casting, actor
WHERE movie.id = casting.movieid AND casting.actorid = actor.id
AND actor.name = "Gerard Depardieu"
```

Cependant, il apparaît que cette façon de faire la jointure consomme trop de temps sur certains serveurs...

Q23. Listez les films où Sylvester Stallone a le premier rôle.

```
SELECT title FROM movie
JOIN casting ON casting.movieid=movie.id
JOIN actor ON casting.actorid=actor.id
WHERE actor.name="Sylvester Stallone"
AND casting.ord=1
```

Q24. Listez les acteurs dont le nom contient 5 fois la lettre « a », et pour chacun, donnez le nombre de films où il a joué.

```
SELECT name, count(*) FROM actor
JOIN casting ON actor.id=casting.actorid
WHERE name LIKE "%a%a%a%a%a%"
GROUP BY actor.id
```

Q25. Quels sont les acteurs pour lesquels le score moyen (voir plus haut) des films dans lesquels ils ont joué est le plus favorable ?

```
SELECT actor.name, SUM(movie.score*movie.votes)/SUM(movie.votes) AS S FROM movie
JOIN casting ON casting.movieid=movie.id
JOIN actor ON casting.actorid=actor.id
GROUP BY actor.id ORDER BY S DESC LIMIT 10
```

Q26. Donnez la liste des acteurs capables de jouer dans des navets comme dans des chefs-d'œuvre. (on pourra considérer qu'un navet a une note ≤ 3 et un chef-d'œuvre une note ≥ 8 .)

```
SELECT name, max(score) AS maximum, min(score) AS minimum FROM actor
JOIN casting ON casting.actorid = actor.id
JOIN movie ON movie.id = casting.movieid
GROUP BY actor.id
HAVING minimum<3 AND maximum>8
```

Q27. Donnez la liste des acteurs qui ont joué dans au moins 20 films.

```
SELECT name, count(*) AS C FROM actor
JOIN casting ON actor.id = casting.actorid
GROUP BY actor.id HAVING C>=20
```

Q28. Quel est le nombre moyen de films par acteur ?

```
SELECT 1.0*COUNT(*)/( SELECT COUNT(DISTINCT actorid) FROM casting) FROM CASTING
```

Attention à bien faire les calculs en flottant, sinon c'est la division entière qui est effectuée!

Q29. Quels sont les acteurs ayant le plus de premiers rôles ?

```
SELECT actor.name, count(*) AS C FROM actor
JOIN casting ON casting.actorid=actor.id
WHERE casting.ord=1
GROUP BY actor.id ORDER BY C DESC LIMIT 10
```

Q30. Donner la liste des acteurs qui ont déjà eu un rôle d'ordre > 5 .

```
SELECT name FROM actor
JOIN casting ON actor.id = casting.actorid
WHERE ord>5 GROUP BY actor.id
```

ou bien :

```
SELECT DISTINCT(actorid),name FROM actor
JOIN casting ON actor.id = casting.actorid
WHERE ord>5
```

Q31. Quel pourcentage d'acteurs a déjà eu un rôle d'ordre > 5 ?

```
SELECT 100.0*COUNT(DISTINCT actorid)/(SELECT COUNT(DISTINCT actorid) FROM casting)
FROM actor JOIN casting ON actor.id = casting.actorid
WHERE ord>5
```

Q32. Donner la liste des acteurs qui n'ont eu que des rôles d'ordre > 5.

```
SELECT DISTINCT(actorid) AS c,name FROM actor
JOIN casting ON actor.id = casting.actorid
WHERE NOT EXISTS ( SELECT * FROM casting WHERE actorid=c AND ord<=5)
```

Q33. Quel est le seul film où ont joué ensemble Brad Pitt et Edward Norton ?

```
SELECT title FROM movie, casting AS c1, actor AS a1, casting AS c2, actor
AS a2
WHERE c1.movieid=movie.id AND c1.actorid=a1.id
AND c2.movieid=movie.id AND c2.actorid=a2.id
AND a1.name="Edward Norton" AND a2.name="Brad Pitt"
```

Q34. Dresser la liste des acteurs qui ont joué avec John Wayne (identifiant 170), avec les films communs (un acteur qui a joué deux fois avec John Wayne apparaîtra 2 fois, une fois avec chaque film.

```
SELECT actor.name, movie.title FROM casting AS C1
JOIN casting AS C2 ON C1.movieid = C2.movieid
JOIN movie ON movie.id = C1.movieid
JOIN actor ON actor.id = C2.actorid
WHERE C1.actorid = 170 AND C2.actorid <> 170
```

Si on ne donne pas l'identifiant de John Wayne :

```
SELECT actor.name, movie.title FROM casting AS C1
JOIN casting AS C2 ON C1.movieid = C2.movieid
JOIN movie ON movie.id = C1.movieid
JOIN actor ON actor.id = C2.actorid
JOIN actor as a2 ON a2.id=C1.actorid
WHERE a2.name='John Wayne' AND C2.actorid <> C1.actorid
```

Q35. Pour chaque acteur qui a joué avec John Wayne, donner le nombre de films communs.

```
SELECT actor.name, COUNT(*) AS c
FROM casting AS C1
JOIN casting AS C2 ON C1.movieid = C2.movieid
JOIN movie ON movie.id = C1.movieid
JOIN actor ON actor.id = C2.actorid
WHERE C1.actorid = 170 AND C2.actorid <> 170
GROUP BY actor.id,actor.name ORDER BY C DESC
```

IV. Réalisateurs et acteurs

Q36. Listez les personnes de la table actor qui ne sont jamais réalisateur, ni jamais acteur.

```
SELECT name, id AS i FROM actor
WHERE NOT EXISTS (SELECT * FROM movie WHERE director=i)
AND NOT EXISTS (SELECT * FROM casting WHERE actorid=i)
```

Q37. Trouvez tous les films réalisés par Martin Scorsese (id = 78), et dans lesquels joue Robert De Niro (id = 3).

```
SELECT title FROM movie AS m
JOIN casting AS c ON m.id=c.movieid
WHERE m.director=78 AND c.actorid=3
```

Si l'on ne donne pas les identifiants :

```
SELECT title FROM movie AS m
JOIN casting AS c ON m.id=c.movieid
JOIN actor AS a1 ON c.actorid=a1.id
JOIN actor AS a2 ON m.director=a2.id
WHERE a2.name="Martin Scorsese" AND a1.name="Robert De Niro"
```

Q38. Quels acteurs ont le plus tourné avec Stanley Kubrick ?

```
SELECT a.name, count(*) AS c FROM actor AS a
JOIN casting ON casting.actorid=a.id
JOIN movie ON movie.id=casting.movieid
JOIN actor AS a2 ON movie.director=a2.id
WHERE a2.name="Stanley Kubrick"
GROUP BY a.id order BY c desc LIMIT 10
```

Q39. Dressez la liste des acteurs qui ne sont pas Woody Allen, qui ont joué au moins 3 fois dans un film réalisé par Woody Allen.

```
SELECT A1.name, count(*) AS c FROM actor AS A1
JOIN casting ON A1.id = casting.actorid
JOIN movie ON movie.id = casting.movieid
WHERE A1.name!="Woody Allen"
GROUP BY A1.id HAVING C>2 order BY C desc
```

Q40. Combien de réalisateurs sont aussi acteurs (c'est-à-dire ont joué dans au moins un film) ?

```
SELECT COUNT(DISTINCT director) FROM movie, casting
WHERE movie.director=casting.actorid
```

Q41. Combien de réalisateurs ont déjà joué dans un film réalisé par eux-mêmes ?

```
SELECT COUNT(distinct actor.id) FROM actor, casting, movie
WHERE actor.id=casting.actorid AND casting.movieid=movie.id
AND movie.director=actor.id
```

Q42. Combien de réalisateurs ont déjà tourné dans un film réalisé par quelqu'un d'autre ?

```
SELECT COUNT(distinct actor.id) FROM actor, casting, movie AS m1, movie AS m2
WHERE actor.id=casting.actorid AND actor.id=m1.director
AND m2.id=casting.movieid AND m2.director != actor.id
```

Q43. Listez les personnes qui ont réalisé un film qui a eu une note ≥ 8.2 , et joué dans un film qui a obtenu un note ≥ 8.2 (pas forcément le même film).

```
SELECT actor.name
FROM movie AS m1
JOIN casting ON casting.actorid=m1.director
JOIN movie AS m2 ON m2.id = casting.movieid
JOIN actor ON actor.id=m1.director
WHERE m1.score>=8.2 AND m2.score>=8.2
GROUP BY m1.director
```

Q44. Listez les personnes qui ont réalisé et joué dans un film (le même!) qui a obtenu une note ≥ 8.2 .

```
SELECT actor.name
FROM movie
JOIN casting ON casting.actorid=movie.director AND movie.id = casting.movieid
JOIN actor ON actor.id=movie.director
WHERE score>=8.2
GROUP BY movie.director
```

★ ★ ★ ★
★ ★ ★
★ ★
★