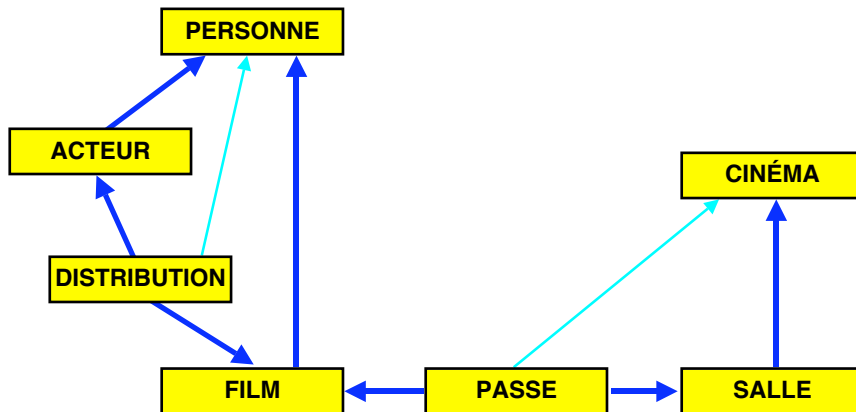


TD : langage assertienel – SQL

Corrigé

Liste des requêtes à exprimer en SQL.

Tracer le graphe des clés étrangères avant de commencer à répondre aux requêtes.



Requêtes élémentaires

Requête 1 Donner la liste de tous les films.

```
SELECT *  
FROM FILM
```

Requête 2 Retrouver la liste des films dont la longueur dépasse 180 min.

```
SELECT *  
FROM FILM  
WHERE LONGUEUR > 180
```

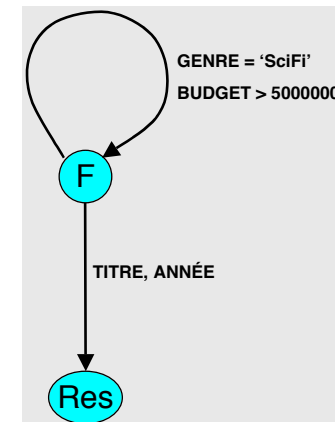
Requête 3 Donner la liste de tous les genres de film.

```
SELECT DISTINCT GENRE  
FROM FILM
```

Requête 4 Trouver le titre et l'année des films de science fiction dont le budget dépasse 5.000.000 \$.

```
SELECT TITRE, ANNÉE  
FROM FILM  
WHERE GENRE = 'SciFi'  
AND BUDGET > 5000000
```

Graphe de la requête :



Requête 5 Donner le nombre de films par genre.

```
SELECT GENRE, COUNT (*)  
FROM FILM  
GROUP BY GENRE
```

Requête 6 Donner le nombre de films de 1960 par genre.

```
SELECT GENRE, COUNT (*)  
FROM FILM  
WHERE ANNÉE = 1960  
GROUP BY GENRE
```

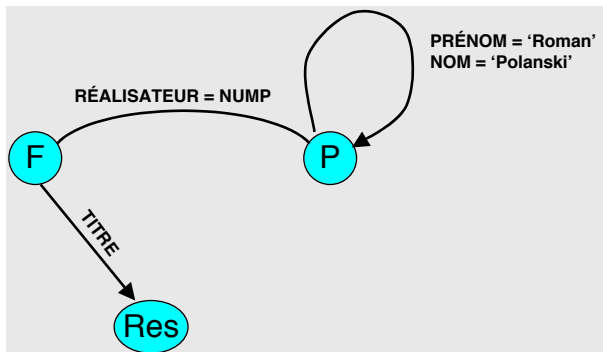
Requêtes faciles

Requête 1 Trouver le titre des films réalisés par Roman Polanski.

Forme plate

```
SELECT F.TITRE
FROM FILM F, PERSONNE P
WHERE F.RÉALISATEUR = P.NUMP
AND P.PRÉNOM = 'Roman'
AND P.NOM = 'Polanski'
```

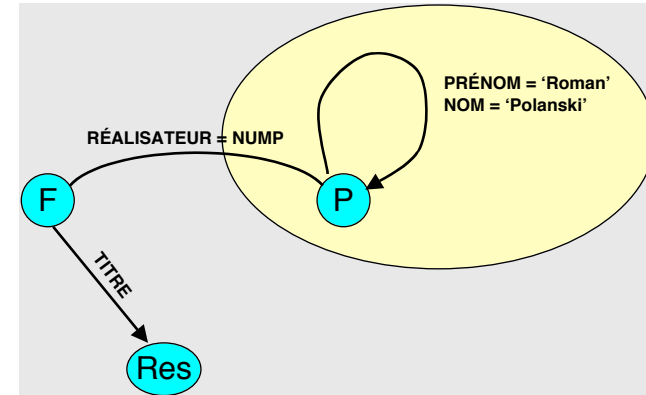
Graphe de la requête :



Forme imbriquée

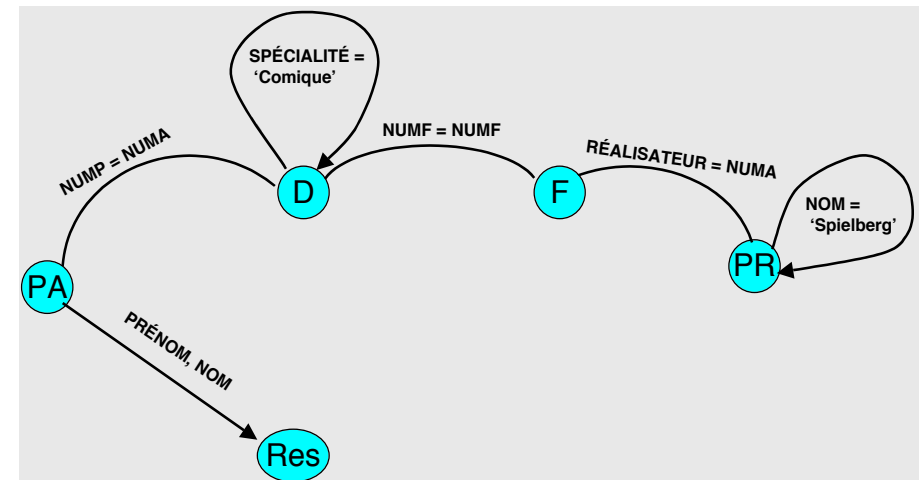
```
SELECT TITRE
FROM FILM
WHERE RÉALISATEUR IN (
  SELECT NUMP
  FROM PERSONNE
  WHERE PRÉNOM = 'Roman'
  AND NOM = 'Polanski' )
```

Graphe de la requête : la requête imbriquée ne contribue pas au résultat



Requête 2 Quels sont les acteurs comiques (nom, prénom) qui ont joué dans un film de Spielberg.

Graphe de la requête :



Forme plate :

```
SELECT PA.PRÉNOM, PA.NOM
FROM PERSONNE PA, DISTRIBUTION D, FILM F, PERSONNE PR
WHERE PA.NUMP = D.NUMA
```

```

AND D.SPÉCIALITÉ = 'Comique'
AND D.NUMF = F.NUMF
AND F.RÉALISATEUR = PR.NUMP
AND PR.NOM = 'Spielberg'

```

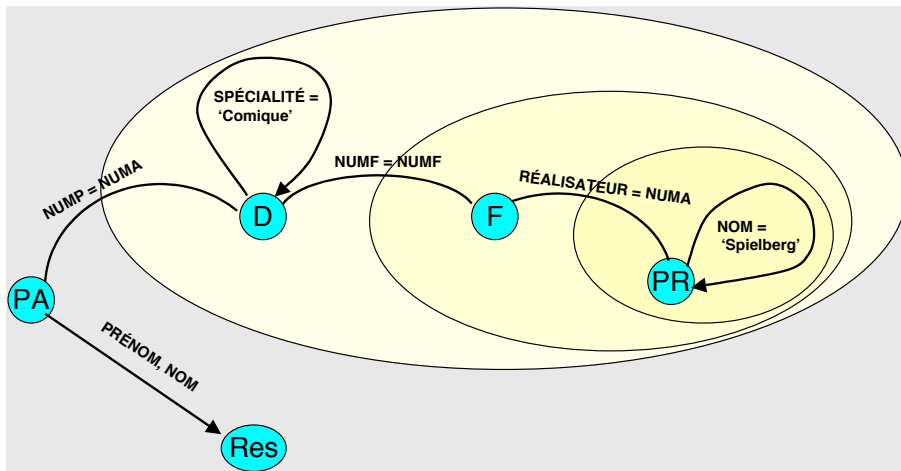
Forme imbriquée

```

SELECT PRÉNOM, NOM
FROM PERSONNE
WHERE NUMP IN (
  SELECT NUMA
  FROM DISTRIBUTION
  WHERE SPÉCIALITÉ = 'Comique'
  AND NUMF IN (
    SELECT NUMF
    FROM FILM
    WHERE RÉALISATEUR IN (
      SELECT NUMP
      FROM PERSONNE
      WHERE NOM = 'Spielberg' ) ) )

```

Graphe de la requête : seule la variable PA contribue au résultat, on obtient trois niveaux d'imbrication



Requête $\square\square$ *Trouver le titre et l'année du film le plus long.*

Forme imbriquée

```

SELECT TITRE, ANNÉE
FROM FILM
WHERE LONGUEUR = (
  SELECT MAX (LONGUEUR)
  FROM FILM )

```

Requêtes de difficulté moyenne

Requête $\square\square\square$ *Nom et prénom des acteurs qui ont joué Gavroche dans les différentes versions des $\square\square$ Misérables $\square\square$ avec les dates correspondantes.*

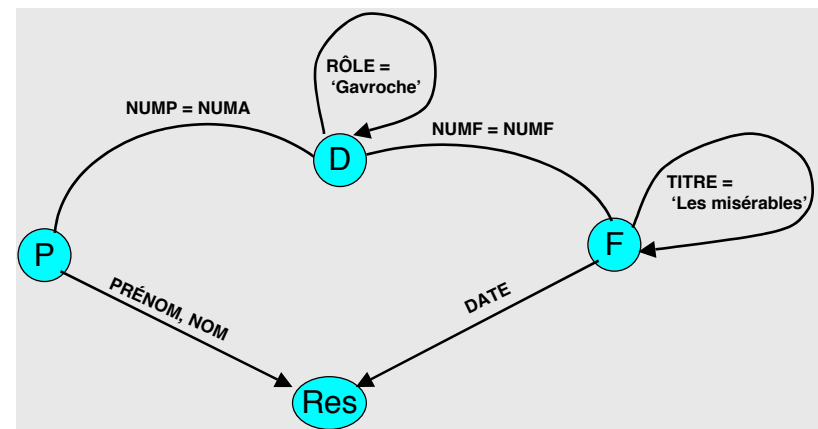
Forme plate :

```

SELECT P.PRÉNOM, P.NOM, F.DATE
FROM PERSONNE P, DISTRIBUTION D, FILM F
WHERE P.NUMP = D.NUMA
AND D.RÔLE = 'Gavroche'
AND G.NUMF = F.NUMF
AND F.TITRE = 'Les misérables'

```

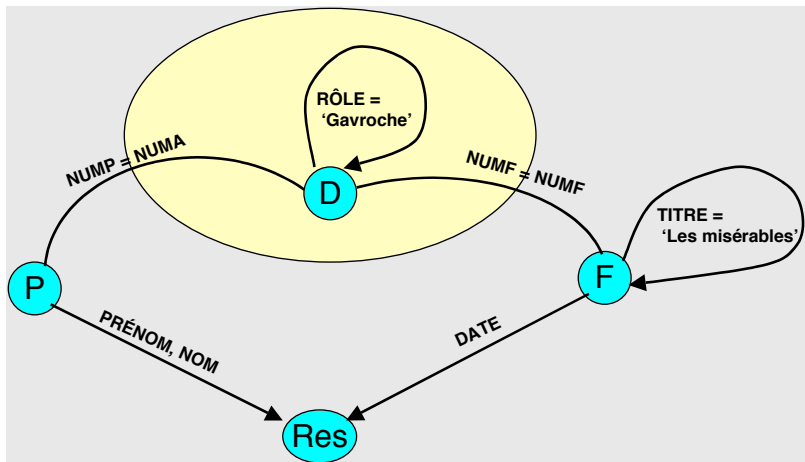
Graphe de la requête :



Forme imbriquée SQL-92

```
SELECT P.PRÉNOM, P.NOM, F.DATE
FROM PERSONNE P, FILM F
WHERE F.TITRE = 'Les misérables'
AND (P.NUMP, F.NUMF) IN (
    SELECT NUMA, NUMF
    FROM DISTRIBUTION
    WHERE RÔLE = 'Gavroche')
```

Graphe de la requête : la requête imbriquée ne contribue pas au résultat et est liée deux fois à la requête globale



Requête 11 Donner le nom et le prénom des réalisateurs qui ont joué dans au moins un de leurs propres films.

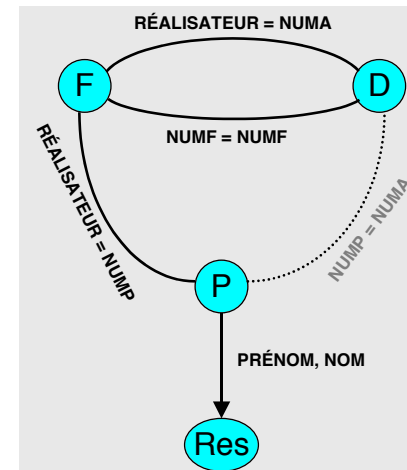
Forme plate :

```
SELECT DISTINCT P.PRÉNOM, P.NOM
FROM PERSONNE P, FILM F, DISTRIBUTION D
WHERE P.NUMP = F.RÉALISATEUR
AND F.NUMF = D.NUMF
AND D.NUMA = F.RÉALISATEUR
```

Forme imbriquée SQL-92

```
SELECT DISTINCT PRÉNOM, NOM
FROM PERSONNE
WHERE NUMP IN (
    SELECT RÉALISATEUR
    FROM FILM
    WHERE (RÉALISATEUR, NUMF) IN (
        SELECT NUMA, NUMF
        FROM DISTRIBUTION))
```

Graphe de la requête :



Requête 12 Quel est le total des salaires des acteurs du film «Nuits blanches à Seattle».

Forme plate :

```
SELECT SUM (D.SALAIRE)
FROM DISTRIBUTION D, FILM F
WHERE DISTRIBUTION.NUMF = F.NUMF
AND F.TITRE = 'Nuits blanches à Seattle'
```

Forme imbriquée :

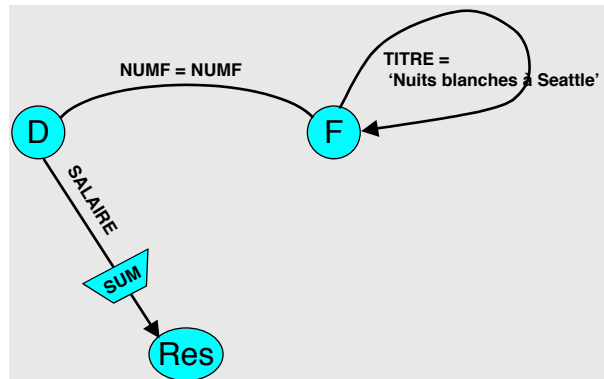
```
SELECT SUM (SALAIRE)
```

```

FROM DISTRIBUTION
WHERE NUMF IN (
  SELECT NUMF
  FROM FILM
  WHERE TITRE = 'Nuits blanches à Seattle' )

```

Grphe de la requête :



Requête 3 Donner la moyenne des salaires des acteurs par film, avec le titre et l'année correspondants.

Forme SQL-89 :

```

SELECT F.TITRE, F.ANNÉE, AVG (D.SALAIRE)
FROM FILM F, DISTRIBUTION D
WHERE F.NUMF = D.NUMF
GROUP BY F.TITRE, F.ANNÉE

```

Forme SQL-92 :

```

SELECT F.TITRE, F.ANNÉE, AVG (D.SALAIRE)
FROM FILM F, DISTRIBUTION D
WHERE F.NUMF = D.NUMF
GROUP BY F.NUMF
-- Si NUMF a bien été déclaré clé primaire de la relation FILM

```

Requête 4 Trouver le genre des films des années 80 dont le budget moyen dépasse 200.000 \$.

```

SELECT GENRE
FROM FILM
WHERE ANNÉE BETWEEN 1980 AND 1989
GROUP BY GENRE
HAVING AVG (BUDGET) > 200000

```

Requêtes plus complexes

Requête 5 Pour chaque film de Spielberg (titre, année), donner le total des salaires des acteurs.

Forme plate :

```

SELECT F.TITRE, F.ANNÉE, SUM (D.SALAIRE)
FROM FILM F, DISTRIBUTION D, PERSONNE P
WHERE F.NUMF = D.NUMF
AND F.RÉALISATEUR = P.NUMP
AND P.NOM = 'Spielberg'
GROUP BY F.TITRE, F.ANNÉE

```

Forme imbriquée SQL-89 :

```

SELECT F.TITRE, F.ANNÉE, SUM (D.SALAIRE)
FROM FILM F, DISTRIBUTION D
WHERE F.NUMF = D.NUMF
AND F.RÉALISATEUR IN (
  SELECT NUMP
  FROM PERSONNE
  WHERE NOM = 'Spielberg' )
GROUP BY F.TITRE, F.ANNÉE

```

Forme imbriquée SQL-92 :

```

SELECT F.TITRE, F.ANNÉE, X.SUMSAL
FROM FILM F, (
  SELECT NUMF, SUM (SALAIRE) AS SUMSAL
  FROM DISTRIBUTION

```

```

GROUP BY NUMF ) AS X
WHERE F.NUMF = X.NUMF
AND F.RÉALISATEUR IN (
    SELECT NUMP
    FROM PERSONNE
    WHERE NOM = 'Spielberg' )

```

Requête 6 Lister les cinémas dont la taille moyenne d'écran est supérieure à 40 mètres carrés.

Forme plate :

```

SELECT C.NOM, C.VILLE
FROM CINÉMA C, SALLE S
WHERE C.NUMC = S.NUMC
GROUP BY C.NUMC, C.NOM, C.VILLE
HAVING AVG (S.TAILLE_ÉCRAN) > 40 )

```

Forme imbriquée SQL-92 :

```

SELECT NOM, VILLE
FROM CINÉMA
WHERE NUMC IN (
    SELECT NUMC
    FROM SALLE
    GROUP BY NUMC
    HAVING AVG (TAILLE_ÉCRAN) > 40 )

```

Requête 7 Quels sont les cinémas Parisiens de la Fox, avec le film correspondant, qui passent un film d'Elia Kazan avant 22 heures dans une salle d'au moins 200 places et d'écran de taille supérieure à 30 carrés.

Forme plate :

```

SELECT DISTINCT C.NOM, F.TITRE
FROM CINÉMA C, SALLE S, PASSE P, FILM F, PERSONNE P
WHERE C.COMPAGNIE = 'Fox'
AND C.VILLE = 'Paris'

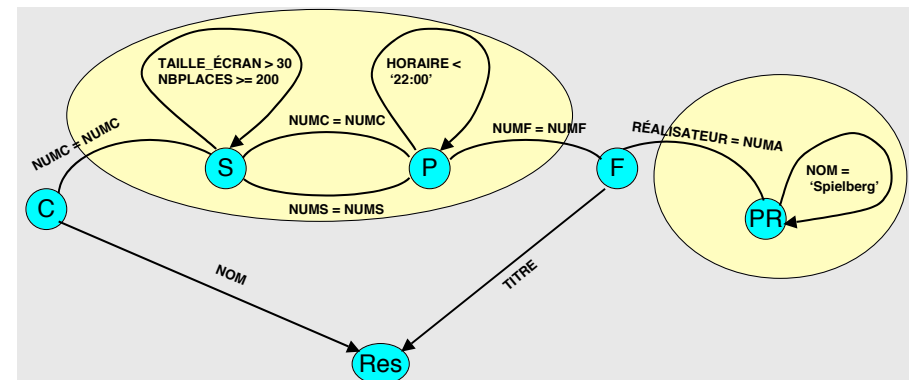
```

```

AND C.NUMC = S.NUMC
AND S.NBPLACES >= 200
AND S.TAILLE_ÉCRAN > 30
AND S.NUMC = P.NUMC
AND S.NUMS = P.NUMS
AND P.HORAIRE < '22:00'
AND P.NUMF = F.NUMF
AND F.RÉALISATEUR = P.NUMP
AND P.PRÉNOM = 'Elia'
AND P.NOM = 'Kazan'

```

Grappe de la requête :



Forme imbriquée SQL-89 :

```

SELECT DISTINCT C.NOM, F.TITRE
FROM CINÉMA C, SALLE S, PASSE P, FILM F
WHERE C.COMPAGNIE = 'Fox'
AND C.VILLE = 'Paris'
AND C.NUMC = S.NUMC
AND S.NBPLACES >= 200
AND S.TAILLE_ÉCRAN > 30
AND S.NUMC = P.NUMC
AND S.NUMS = P.NUMS
AND P.HORAIRE < '22:00'
AND P.NUMF = F.NUMF

```

```

AND F.RÉALISATEUR IN (
  SELECT NUMP
  FROM PERSONNE
  WHERE PRÉNOM = 'Elia'
  AND NOM = 'Kazan' )

```

Forme imbriquée SQL-92

```

SELECT DISTINCT C.NOM, F.TITRE
FROM CINÉMA C, FILM F
WHERE C.COMPAGNIE = 'Fox'
AND C.VILLE = 'Paris'
AND (C.NUMC, F.NUMF) IN (
  SELECT S.NUMC, P.NUMF
  FROM SALLE S, PASSE P
  WHERE S.NBPLACES >= 200
  AND S.TAILLE_ÉCRAN > 30
  AND S.NUMC = P.NUMC
  AND S.NUMS = P.NUMS
  AND P.HORAIRE < '22:00' )
AND F.RÉALISATEUR IN (
  SELECT NUMP
  FROM PERSONNE
  WHERE PRÉNOM = 'Elia'
  AND NOM = 'Kazan' )

```

Requête 8 Trouver le titre des films qui ne passent à aucun cinéma de la compagnie FOX.

On commence par poser la requête inverse les films qui passent dans un (au moins) cinéma de la Fox.

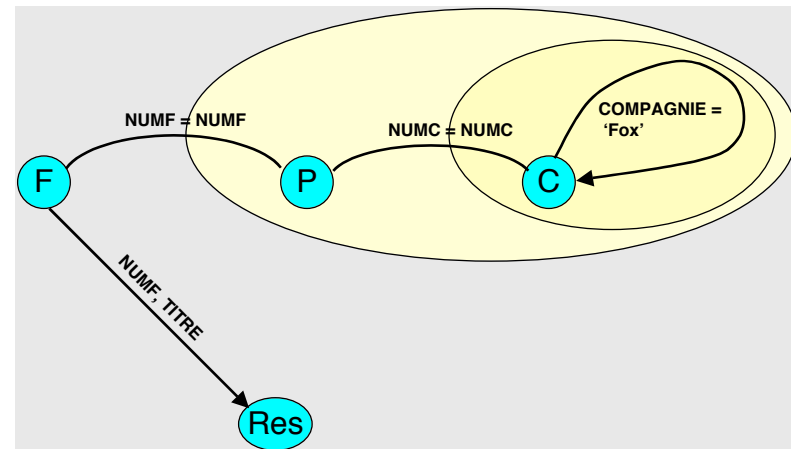
Forme plate : pour trouver ceux qui passent dans un cinéma de la Fox

```

SELECT DISTINCT F.NUMF, F.TITRE
FROM FILM F, PASSE P, CINÉMA C
WHERE F.NUMF = P.NUMF
AND P.NUMC = C.NUMC
AND C.COMPAGNIE = 'Fox'

```

Grphe de la requête :



Forme imbriquée 1 – prédicat IN : pour trouver ceux qui passent dans un cinéma de la Fox

```

SELECT DISTINCT NUMF, TITRE
FROM FILM
WHERE NUMF IN (
  SELECT NUMF
  FROM PASSE
  WHERE NUMC IN (
    SELECT NUMC
    FROM CINÉMA
    WHERE COMPAGNIE = 'Fox' ) )

```

Forme imbriquée 2 – prédicat EXISTS : toujours pour trouver ceux qui passent dans un cinéma de la Fox

```

SELECT DISTINCT NUMF, TITRE
FROM FILM F
WHERE EXISTS (
  SELECT *
  FROM PASSE P
  WHERE P.NUMF = F.NUMF
  AND EXISTS (

```

```

SELECT *
FROM CINÉMA C
WHERE C.NUMC = P.NUMC
AND COMPAGNIE = 'Fox' )

```

La négation de ces deux dernières formes permet d'exprimer la requête initiale ☐ les films qui ne passent à aucun des cinémas de la Fox.

Forme imbriquée 1 – prédicat **NOT IN** : pour trouver ceux qui ne passent dans aucun cinéma de la Fox

```

SELECT DISTINCT NUMF, TITRE
FROM FILM
WHERE NUMF NOT IN (
  SELECT NUMF
  FROM PASSE
  WHERE NUMC IN (
    SELECT NUMC
    FROM CINÉMA
    WHERE COMPAGNIE = 'Fox' ) )

```

Forme imbriquée 2 – prédicat **NOT EXISTS** : pour trouver ceux qui ne passent dans aucun cinéma de la Fox

```

SELECT DISTINCT NUMF, TITRE
FROM FILM F
WHERE NOT EXISTS (
  SELECT *
  FROM PASSE P
  WHERE P.NUMF = F.NUMF
  AND EXISTS (
    SELECT *
    FROM CINÉMA C
    WHERE C.NUMC = P.NUMC
    AND COMPAGNIE = 'Fox' ) )

```

Pour finalement arriver à la forme la plus simple, où seul le prédicat NOT EXISTS provoque un niveau d'imbrication.

Forme 3 – prédicat **NOT EXISTS** uniquement :

```

SELECT DISTINCT NUMF, TITRE
FROM FILM F
WHERE NOT EXISTS (
  SELECT *
  FROM PASSE P, CINÉMA C
  WHERE F.NUMF = P.NUMF
  AND P.NUMC = C.NUMC
  AND COMPAGNIE = 'Fox' )

```

Si on est très pointilleux, on peut remarquer que la question « ☐ qui ne passent à aucun ☐ peut être comprise comme ne devant retourner que les films actuellement programmés, mais pas dans un cinéma de la Fox. Si la relation FILM comporte tous les films qui ont été programmés, on peut modifier la requête comme suit ☐

Forme complète :

```

SELECT DISTINCT NUMF, TITRE
FROM FILM F
WHERE NUMF IN (
  SELECT NUMF
  FROM PASSE )
AND NOT EXISTS (
  SELECT *
  FROM PASSE P, CINÉMA C
  WHERE F.NUMF = P.NUMF
  AND P.NUMC = C.NUMC
  AND COMPAGNIE = 'Fox' )

```

Requête ☐9☐ *Trouver le nom et le prénom des acteurs qui ont eu un salaire plus important dans un film particulier que le salaire du réalisateur du même film.*

En supposant pour simplifier que les acteurs n'ont joué qu'un seul rôle dans le film.

Forme plate :

```

SELECT PA.PRÉNOM, PA.NOM
FROM PERSONNE PA, DISTRIBUTION D, FILM F
WHERE PA.NUMP = D.NUMA

```

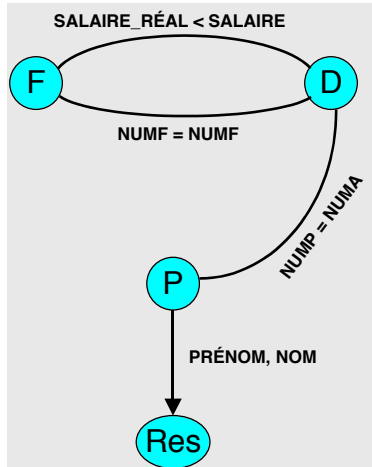


```

AND    D.NUMF = F.NUMF
AND    D.SALAIRE > F.SALAIRE_RÉAL

```

Graphe de la requête :



Forme imbriquée 1 :

```

SELECT PRÉNOM, NOM
FROM PERSONNE
WHERE NUMP IN (
    SELECT D.NUMA
    FROM DISTRIBUTION D, FILM F
    WHERE D.NUMF = F.NUMF
    AND D.SALAIRE > F.SALAIRE_RÉAL )

```

Forme imbriquée 2

```

SELECT PRÉNOM, NOM
FROM PERSONNE
WHERE NUMP IN (
    SELECT NUMA
    FROM DISTRIBUTION D
    WHERE D.SALAIRE > (
        SELECT F.SALAIRE_RÉAL
        FROM FILM F

```

```

WHERE D.NUMF = F.NUMF ) )

```

En tenant compte de la possibilité que les acteurs aient joué plusieurs rôles dans le film, et en supposant qu'ils touchent un salaire pour chaque rôle dans ce cas. (NB: cela ne reflète pas forcément les pratiques de l'industrie du cinéma, mais pour faire plus proche de la réalité, il faudrait compliquer sensiblement le schéma de la base).

Forme imbriquée SQL-89 :

```

SELECT DISTINCT PA.PRÉNOM, PA.NOM
FROM PERSONNE PA, DISTRIBUTION D
WHERE PA.NUMP = D.NUMA
GROUP BY D.NUMA, D.NUMF, PA.PRÉNOM, PA.NOM
HAVING SUM (SALAIRE) > (
    SELECT SALAIRE_RÉAL
    FROM FILM F
    WHERE D.NUMF = F.NUMF ) )

```

Forme imbriquée SQL-92 :

```

SELECT PRÉNOM, NOM
FROM PERSONNE
WHERE NUMP IN (
    SELECT NUMA
    FROM DISTRIBUTION D
    GROUP BY NUMA, NUMF
    HAVING SUM (SALAIRE) > (
        SELECT SALAIRE_RÉAL
        FROM FILM F
        WHERE D.NUMF = F.NUMF ) )

```

Requêtes difficiles

Requête: Trouver les couples acteur-réalisateur (noms et prénoms) tels que l'un a dirigé l'autre sur un film et vice-versa sur un autre.

Forme plate :

```

SELECT DISTINCT P1.PRÉNOM, P1.NOM, P2.PRÉNOM, P2.NOM
FROM PERSONNE P1, PERSONNE P2, FILM F1, FILM F2,

```

DISTRIBUTION D1, DISTRIBUTION D2

```

WHERE P1.NUMP > P2.NUMP
-- PERMET D'ÉLIMINER DEUX PROBLÈMES☐
-- LISTER UN COUPLE (P1, P2) UNE SEULE FOIS
-- DANS UN SEUL ORDRE
-- ÉLIMINER LE CAS DES RÉALISATEURS QUI ONT JOUÉ
-- DANS LEUR PROPRE FILM
AND P1.NUMF = F1.RÉALISATEUR
AND P2.NUMF = F2.RÉALISATEUR
AND F1.NUMF = D1.NUMF
AND D1.NUMA = F2.RÉALISATEUR
AND F2.NUMF = D2.NUMF
AND D2.NUMA = F1.RÉALISATEUR
    
```

```

-- DANS UN SEUL ORDRE
-- ÉLIMINER LE CAS DES RÉALISATEURS QUI ONT JOUÉ
-- DANS LEUR PROPRE FILM
AND F1.NUMF = D1.NUMF
AND D1.NUMA = F2.RÉALISATEUR
AND F2.NUMF = D2.NUMF
AND D2.NUMA = F1.RÉALISATEUR )
    
```

Requête☐☐ Trouver le nom, le prénom, le numéro des acteurs qui ont joué dans tous les films de Lelouch, s'il y en a.

On peut rephraser logiquement cette requête en remarquant que l'on cherche les acteurs des films de Lelouch pour lesquels pour chaque film de Lelouch on peut trouver un tuple dans distribution qui indique que l'acteur a bien joué dans ce film.

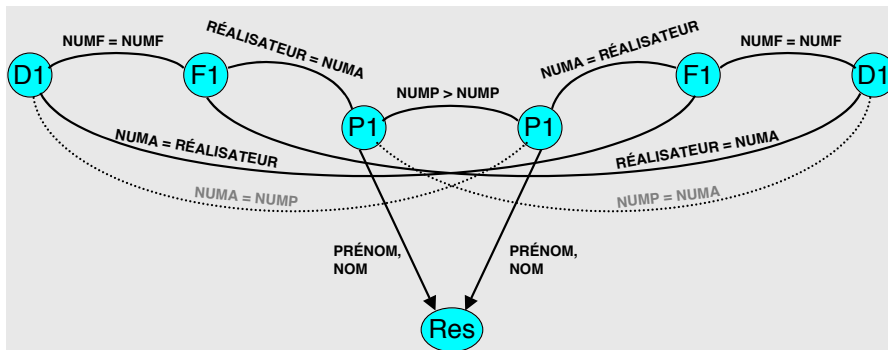
{acteur☐☐ film-de-Lelouch, ☐ distribution tq acteur-a-joué-dans-film }

SQL ne comporte pas d'équivalent du quantificateur universel ☐. On doit donc utiliser une transformation logique – introduction d'une double négation – qui produit la négation d'un quantificateur existentiel ☐☐ après descente d'une des deux négations.

{acteur☐☐☐ film-de-Lelouch, ☐☐ distribution tq acteur-a-joué-dans-film }

Le plus simple est de d'abord écrire la requête sans négation en utilisant une forme imbriquée avec des prédicats EXISTS, puis d'introduire une double négation. On commencera donc par chercher le nom, le prénom, le numéro des acteurs qui ont joué dans un des films de Lelouch.

Graphe de la requête :



Forme imbriquée SQL-92☐

```

SELECT DISTINCT P1.PRÉNOM, P1.NOM, P2.PRÉNOM, P2.NOM
FROM PERSONNE P1, PERSONNE P2
WHERE (P1.NUMP, P2.NUMP) IN (
    SELECT F1.RÉALISATEUR, F2.RÉALISATEUR
    FROM FILM F1, FILM F2, DISTRIBUTION D1,
        DISTRIBUTION D2
    WHERE F1.RÉALISATEUR > F2.RÉALISATEUR
    -- PERMET D'ÉLIMINER DEUX PROBLÈMES☐
    -- LISTER UN COUPLE (RÉAL1, RÉAL2) UNE SEULE FOIS
    
```

Forme imbriquée – prédicat EXISTS : «☐dans un des films☐

```

SELECT NOM, PRÉNOM
FROM PERSONNE P
WHERE EXISTS (
    SELECT *
    FROM FILM F
    WHERE RÉALISATEUR IN (
        SELECT NUMP
        FROM PERSONNE
        WHERE NOM = 'Lelouch' )
    AND EXISTS (
    
```

```

SELECT *
FROM DISTRIBUTION D
WHERE D.NUMF = F.NUMF
AND D.NUMA = P.NUMP )

```

Il ne reste plus alors qu'à introduire les deux négations pour obtenir la requête souhaitée.

Forme imbriquée – prédicat **NOT EXISTS** : «dans tous les films»

```

SELECT NOM, PRÉNOM, NUMP
FROM PERSONNE P
WHERE NOT EXISTS (
  SELECT *
  FROM FILM F
  WHERE RÉALISATEUR IN (
    SELECT NUMP
    FROM PERSONNE
    WHERE NOM = 'Lelouch' )
  AND NOT EXISTS (
    SELECT *
    FROM DISTRIBUTION D
    WHERE D.NUMF = F.NUMF
    AND D.NUMA = P.NUMP ) )

```

Requête 22 Pour chaque film de Bergman, trouver le nom et le prénom de l'acteur qui a eu le plus gros salaire.

Il faut également rephraser cette requête pour l'exprimer de façon logique. La requête consiste à chercher les acteurs pour lesquels, quels que soient les autres acteurs du même film, leur salaire est plus élevé.

Pour simplifier on va d'abord se limiter au cas où un acteur ne joue qu'un rôle par film.

{ film, acteur film film-de-Bergman, acteur acteur-ayant-joué-dans-ce-film, autre-acteur-ayant-joué-dans-ce-film distribution tq acteur-a-joué-dans-film-avec-plus-gros-salaire-que-autre-acteur }

Soit après introduction d'une double négation pour éliminer le :

{ film, acteur film film-de-Bergman, acteur acteur-ayant-joué-dans-ce-film, autre-acteur-ayant-joué-dans-ce-film distribution tq acteur-a-joué-dans-film-avec-plus-gros-salaire-que-autre-acteur }

Forme imbriquée – prédicat **NOT EXISTS** : un seul rôle par acteur

```

SELECT F.TITRE, PA.PRÉNOM, PA.NOM
FROM FILM F, DISTRIBUTION D1, PERSONNE PA
WHERE F.NUMF = D1.NUMF
AND D1.NUMA = PA.NUMP
AND RÉALISATEUR IN (
  SELECT NUMP
  FROM PERSONNE
  WHERE NOM = 'Bergman' )
AND NOT EXISTS (
  SELECT *
  FROM DISTRIBUTION D2
  WHERE D2.NUMF = D1.NUMF
  AND D2.SALAIRE > D1.SALAIRE )

```

En tenant compte maintenant de la possibilité pour un acteur de jouer plusieurs rôles dans un film, on obtient la requête suivante.

Forme imbriquée **SQL-89** – prédicat **> ALL** : possibilité de plusieurs rôles pour un même acteur

```

SELECT F.TITRE, PA.PRÉNOM, PA.NOM
FROM FILM F, DISTRIBUTION D1, PERSONNE PA
WHERE F.NUMF = D1.NUMF
AND D1.NUMA = PA.NUMP
AND RÉALISATEUR IN (
  SELECT NUMP
  FROM PERSONNE
  WHERE NOM = 'Bergman' )
GROUP BY D1.NUMF, D1.NUMA, F.TITRE, PA.PRÉNOM, PA.NOM
HAVING SUM (SALAIRE) > ALL (
  SELECT SUM (SALAIRE)
  FROM DISTRIBUTION D2
  WHERE D2.NUMF = D1.NUMF

```

```
AND D2.NUMA <> D1.NUMA
GROUP BY D2.NUMA )
```

Forme imbriquée SQL-92 : possibilité de plusieurs rôles pour un même acteur

```
SELECT F.TITRE, PA.PRÉNOM, PA.NOM
FROM FILM F, PERSONNE PA
```

```
WHERE (F.NUMF, PA.NUMP) IN (
  SELECT D1.NUMF, D1.NUMA
  FROM DISTRIBUTION D1
  WHERE D1.NUMF IN (
    SELECT NUMF
    FROM FILM
    WHERE RÉALISATEUR IN (
      SELECT NUMP
      FROM PERSONNE
      WHERE NOM = 'Bergman' ) ) )
GROUP BY D1.NUMF, D1.NUMA
HAVING SUM (D1.SALAIRE) = (
  SELECT MAX (
    SELECT SUM (D2.SALAIRE)
    FROM DISTRIBUTION D2
    WHERE D2.NUMF = D1.NUMF
    GROUP BY D2.NUMA ) )
```


On peut aussi simplifier en repartant de la requête valable seulement si un acteur ne joue qu'un rôle par film et en utilisant une vue groupée à la place de distribution.


Utilisation d'une vue groupée SQL-92

```
CREATE VIEW SALAIRE_TOTAL_ACTEUR_FILM
(NUMA, NUMF, SALAIRE_TOTAL)
AS SELECT NUMA, NUMF, SUM (SALAIRE)
FROM FILM
GROUP BY NUMA, NUMF
```

```
SELECT F.TITRE, PA.PRÉNOM, PA.NOM
FROM FILM F, SALAIRE_TOTAL_ACTEUR_FILM D1, PERSONNE PA
```


```
WHERE F.NUMF = D1.NUMF
AND D1.NUMA = PA.NUMP
AND RÉALISATEUR IN (
  SELECT NUMP
  FROM PERSONNE
  WHERE NOM = 'Bergman' )
AND NOT EXISTS (
  SELECT *
  FROM SALAIRE_TOTAL_ACTEUR_FILM D2
  WHERE D2.NUMF = D1.NUMF
  AND D2.SALAIRE_TOTAL > D1.SALAIRE_TOTAL )
```

Requête  Donner le nom et le prénom des réalisateurs qui ont eu le plus gros salaire sur un de leurs films (par comparaison avec ceux des acteurs).

Hypothèse 1  chaque acteur ne joue qu'un rôle, le réalisateur ne joue pas dans le film correspondant.

Forme imbriquée :

```
SELECT PRÉNOM, NOM
FROM PERSONNE
WHERE NUMP IN (
  SELECT RÉALISATEUR
  FROM FILM F
  WHERE SALAIRE_RÉAL > (
    SELECT MAX (SALAIRE)
    FROM DISTRIBUTION D
    WHERE D.NUMF = F.NUMF ) )
```

Hypothèse 2  un acteur peut jouer plusieurs rôles, le réalisateur ne joue pas dans le film correspondant.

Forme imbriquée :

```
SELECT PRÉNOM, NOM
FROM PERSONNE
WHERE NUMP IN (
  SELECT RÉALISATEUR
```

```

FROM FILM F
WHERE SALAIRE_RÉAL > ALL (
  SELECT SUM (SALAIRE)
  FROM DISTRIBUTION D
  WHERE D.NUMF = F.NUMF
  GROUP BY NUMA ) )

```

Hypothèse 3 ☐ Tout acteur – y compris le réalisateur s’il joue dans le film correspondant – peut jouer plusieurs rôles.

Forme imbriquée SQL-92 :

```

SELECT PRÉNOM, NOM
FROM PERSONNE
WHERE NUMP IN (
  SELECT RÉALISATEUR
  FROM FILM F
  WHERE SALAIRE_RÉAL
    + (
      SELECT SUM (SALAIRE)
      FROM DISTRIBUTION D1
      WHERE D1.NUMF = F.NUMF
      AND D1.NUMA = F.RÉALISATEUR )
    > (
      SELECT MAX (
        SELECT SUM (SALAIRE)
        FROM DISTRIBUTION D2
        WHERE D2.NUMF = F.NUMF
        GROUP BY D2.NUMA ) ) )

```

Schéma complémentaire

```

RÉCOMPENSE (NUMR, CATÉGORIE, FESTIVAL)
RÉCOMPENSE_FILM (NUMF, ANNÉE, NUMR)
RÉCOMPENSE_ACTEUR (NUMA, NUMF, ANNÉE, NUMR)

```

Pour répondre aux questions suivantes, il faut noter que lorsqu'un acteur reçoit une récompense, le film en reçoit une indirectement. Ce schéma complémentaire conduit à utiliser une union dans les requêtes.

Requête de difficulté moyenne

Requête ☐4☐ Donner le titre des films qui ont été primés au moins une fois (y compris les récompenses des acteurs jouant dans le film).

Forme plate SQL-89 :

```

SELECT DISTINCT F.TITRE, F.ANNÉE
FROM FILM F, RÉCOMPENSE_FILM RF
WHERE F.NUMF = RF.NUMF
UNION
SELECT DISTINCT F.TITRE, F.ANNÉE
FROM FILM F, RÉCOMPENSE_ACTEUR RA
WHERE F.NUMF = RA.NUMF

```

Forme imbriquée SQL-92 :

```

SELECT TITRE, ANNÉE
FROM FILM
WHERE NUMF IN (
  SELECT NUMF
  FROM RÉCOMPENSE_FILM
UNION
  SELECT NUMF
  FROM RÉCOMPENSE_ACTEUR )

```

Requêtes difficiles

Requête ☐5☐ Lister les cinémas qui ont exclusivement passé des films primés.

Il faut rephraser cette requête pour l'exprimer de façon logique. La requête consiste à chercher les cinémas pour lesquels, quels que soient les films programmés, ces films ont obtenu au moins une récompense. Comme on ne peut pas utiliser d'union dans une sous-requête, il faut donc utiliser une disjonction (ou).

```

{ cinéma ☐ ☐ film-programmé-dans-ce-cinéma ( ( ☐ récompense-film tq film-a-été-récompensé ) ( ☐ récompense-acteur tq film-a-été-récompensé ) ) }

```

Soit, après introduction d'une double négation pour éliminer le ☐, la disjonction se transforme en conjonction (et) :

{ cinéma □ □ film-programmé-dans-ce-cinéma ((□ □ récompense-film tq film-a-été-récompensé) □ (□ □ récompense-acteur tq film-a-été-récompensé)) }

Forme imbriquée SQL-89 :

```
SELECT NOM, VILLE
FROM CINÉMA C
WHERE NOT EXISTS (
  SELECT *
  FROM PASSE P
  WHERE P.NUMC = C.NUMC
  AND NOT EXISTS (
    SELECT *
    FROM RÉCOMPENSE_FILM RF
    WHERE RF.NUMF = P.NUMF )
  AND NOT EXISTS (
    SELECT *
    FROM RÉCOMPENSE_ACTEUR RA
    WHERE RA.NUMF = P.NUMF ) )
```

En SQL-2 par contre, on peut utiliser une union dans une sous-requête, ce qui simplifie son expression.

Forme imbriquée SQL-92 – prédicat NOT EXISTS :

```
SELECT NOM, VILLE
FROM CINÉMA C
WHERE NOT EXISTS (
  SELECT *
  FROM PASSE P
  WHERE P.NUMC = C.NUMC
  AND NOT EXISTS (
    SELECT *
    FROM (
      SELECT NUMF
      FROM RÉCOMPENSE_FILM
      UNION
      SELECT NUMF
      FROM RÉCOMPENSE_ACTEUR ) AS R
    WHERE R.NUMF = P.NUMF ) )
```

WHERE R.NUMF = P.NUMF))

Forme imbriquée SQL-92 – prédicat NOT IN :

```
SELECT NOM, VILLE
FROM CINÉMA
WHERE NUMC NOT IN (
  SELECT NUMC
  FROM PASSE
  WHERE NUMF NOT IN (
    SELECT R.NUMF
    FROM (
      SELECT NUMF
      FROM RÉCOMPENSE_FILM
      UNION
      SELECT NUMF
      FROM RÉCOMPENSE_ACTEUR ) AS R ) ) )
```

Requête 26 □ Donner le titre des films qui ont reçu au moins trois récompenses.

Forme imbriquée SQL-92 :

```
SELECT TITRE, ANNÉE
FROM FILM
WHERE NUMF IN (
  SELECT R.NUMF
  FROM (
    SELECT NUMF
    FROM RÉCOMPENSE_FILM
    UNION
    SELECT NUMF
    FROM RÉCOMPENSE_ACTEUR ) AS R
  GROUP BY R.NUMF
  HAVING COUNT (*) >= 3 )
```

Requête 27 *Noms et prénoms des acteurs qui ont reçu plus de récompenses qu'aucun acteur qui a joué dans "Casablanca" n'en a eu.*

Forme imbriquée SQL-92 :

```
SELECT PRÉNOM, NOM
FROM PERSONNE
WHERE NUMP IN (
  SELECT NUMA
  FROM RÉCOMPENSE_ACTEUR
  GROUP BY NUMA
  HAVING COUNT (*) > (
    SELECT MAX (
      SELECT COUNT (*)
      FROM RÉCOMPENSE_ACTEUR
      WHERE NUMA IN (
        SELECT NUMA
        FROM DISTRIBUTION
        WHERE NUMF IN (
          SELECT NUMF
          FROM FILM
          WHERE TITRE = 'Casablanca' ) )
    GROUP BY NUMA ) ) )
```