LÊ NGUYỄN MINH HƯƠNG

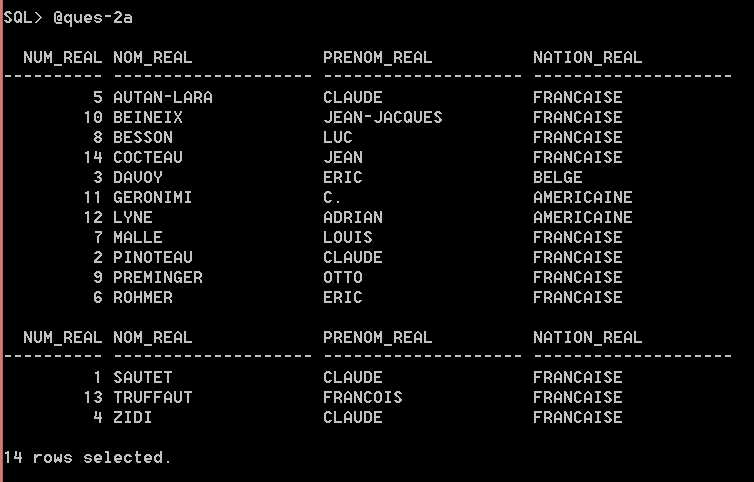
# 2) Requête SQL :

## a. Toutes les informations de tous les réalisateurs triés en ordre alphabétique sur le nom

select numero\_realisateur as num\_real, nom\_realisateur as nom\_real, prenom\_realisateur as prenom\_real, nation\_realisateur as nation\_real

from realisateur

order by nom\_realisateur asc;

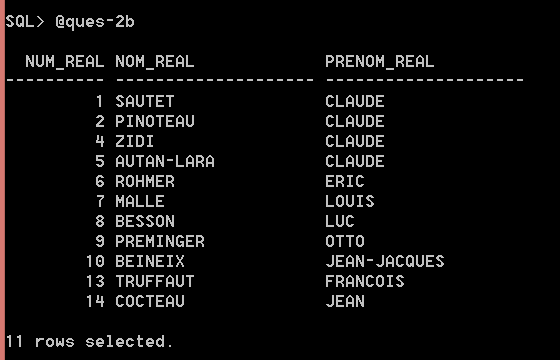


## b. Les numéros, noms et prénoms de tous les réalisateurs de nationalité 'FRANCAISE'

select numero\_realisateur as num\_real, nom\_realisateur as nom\_real, prenom\_realisateur as prenom\_real

from realisateur

where nation\_realisateur = 'FRANCAISE';



## c. Les numéros, noms et prénoms des réalisateurs ayant réalisé au moins un film (sans répétition)

select distinct numero\_realisateur as num\_real, nom\_realisateur as nom\_real, prenom\_realisateur as prenom\_real

from realisateur

where numero\_realisateur in

(

select numero\_realisateur

from film

);



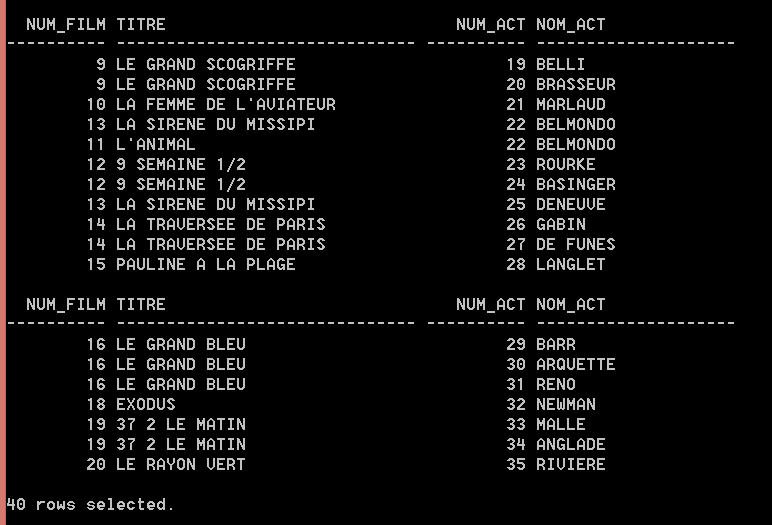
## d. Les numéros et les titres de tous les films avec le numéro et le nom des acteurs

select f.numero\_film as num\_film, f.titre\_film as titre, a.numero\_acteur as num\_act, a.nom\_acteur as nom\_act

from film f, acteur a, role ro

where ro.numero\_acteur = a.numero\_acteur

and ro.numero\_film = f.numero\_film;



## e. Les numéros et les titres des films dans lesquels il y a au moins deux acteurs qui ont eu un rôle,

select f.numero\_film as num\_film, titre\_film as titre, count(\*) as nb\_acteurs

from film f, role ro

where ro.numero\_film = f.numero\_film

group by f.numero\_film, titre\_film

having count(\*) >= 2

order by f.numero\_film asc;



## f. Les numéros, noms et prénoms des réalisateurs dont le prénom est aussi un prénom d'acteur,

select numero\_realisateur as num\_real, nom\_realisateur as nom\_real, prenom\_realisateur as prenom\_real

from realisateur

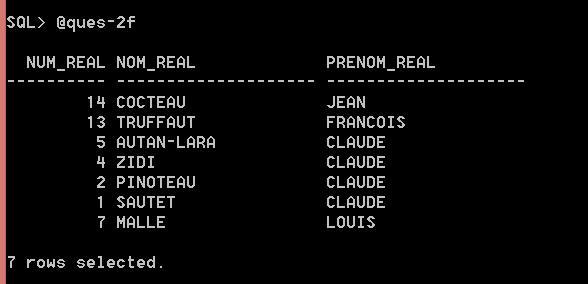
where prenom\_realisateur in

(

select prenom\_acteur

from acteur

);



## g. Les numéros, noms et prénoms des réalisateurs ayant réalisé entre 2 et 5 films,

select re.numero\_realisateur as num\_real, nom\_realisateur as nom\_real, prenom\_realisateur as prenom\_real, count(\*)

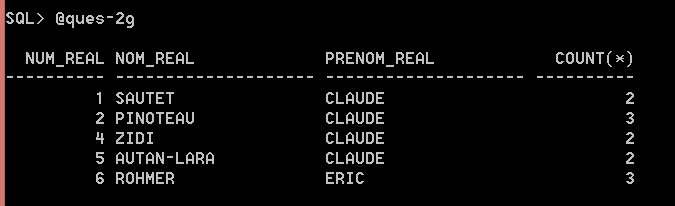
from realisateur re, film f

where re.numero\_realisateur = f.numero\_realisateur

group by re.numero\_realisateur, nom\_realisateur, prenom\_realisateur

having count(\*) between 2 and 5

order by re.numero\_realisateur;



## h. Les numéros, noms et prénoms des acteurs français triés sur le nom, et dont le prénom est un prénom de réalisateur peu répandu (n'apparaissant qu'une seule fois).

select numero\_acteur as num\_act, nom\_acteur as nom\_act, prenom\_acteur as prenom\_act

from acteur

where nation\_acteur = 'FRANCAISE'

and prenom\_acteur in

(

select prenom\_realisateur

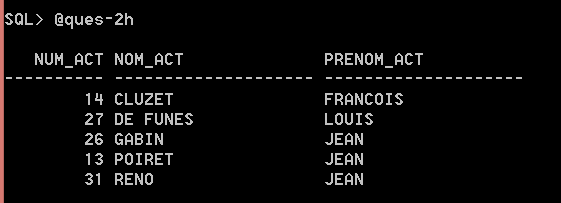
from realisateur

group by prenom\_realisateur

having count(\*) = 1

)

order by nom\_acteur;



## i. Le nombre d'acteurs jouant dans le film 'GARCON'

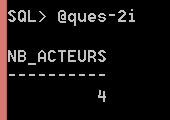
select nom\_acteur as nom\_act, count(\*)

from acteur a, role ro, film f

where ro.numero\_acteur = a.numero\_acteur

and ro.numero\_film = f.numero\_film

and titre\_film = 'GARCON';



## j. Pour chaque acteur (nom, prénom), le nombre de films dans lesquels il a joué,

select nom\_acteur, prenom\_acteur, count(\*) as nb\_films

from acteur a, role ro

where ro.numero\_acteur = a.numero\_acteur

group by nom\_acteur, prenom\_acteur

order by nom\_acteur asc;



## k. Pour chaque acteur (nom, prénom), le premier film dans lequel il a eu un rôle (date\_de\_sortie la plus ancienne).

select nom\_acteur as nom\_act, titre\_film as titre, date\_de\_sortie

from acteur a, film f\_res, role ro

where ro.numero\_acteur = a.numero\_acteur

and ro.numero\_film = f\_res.numero\_film

and date\_de\_sortie =

(

select min(date\_de\_sortie)

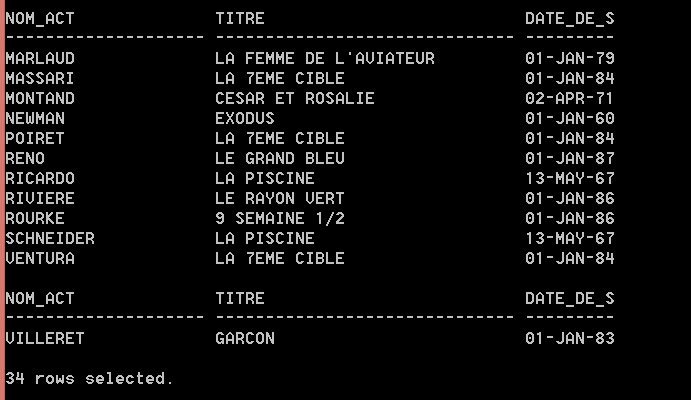
from role ro\_bis, film f\_bis

where ro\_bis.numero\_film = f\_bis.numero\_film

and ro\_bis.numero\_acteur = a.numero\_acteur

)

order by nom\_acteur;



## l. Les prénoms des acteurs et des réalisateurs, sans répétition, sur une seule colonne1,

select distinct prenom\_acteur

from acteur

union

(

select distinct prenom\_realisateur

from realisateur

);



## m. Les prénoms communs aux réalisateurs et aux acteurs,

select prenom\_acteur

from acteur

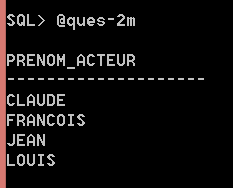
intersect

(

select prenom\_realisateur

from realisateur

);



## n. Les prénoms de réalisateurs qui ne sont pas des prénoms d'acteurs,

select prenom\_realisateur

from realisateur

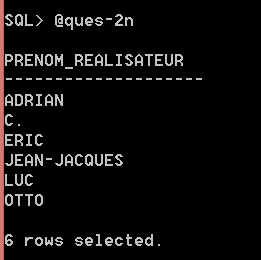
minus

(

select prenom\_acteur

from acteur

);



## o. Affichez les numéros et noms des réalisateurs n'ayant réalisé aucun film en utilisant :

### – une requête imbriquée

select numero\_realisateur as num\_real, nom\_realisateur as nom\_real

from realisateur

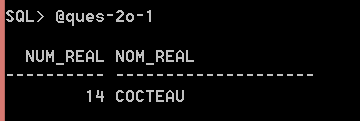
where numero\_realisateur not in

(

select numero\_realisateur

from film

);



### – une opération ensembliste

select numero\_realisateur as num\_real, nom\_realisateur as nom\_real

from realisateur

where numero\_realisateur in

(

select numero\_realisateur

from realisateur

minus

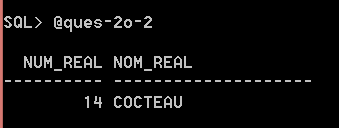
(

select numero\_realisateur

from film

)

);



### – une jointure externe

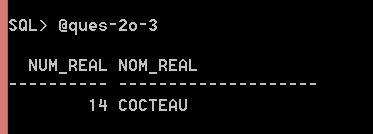
select re.numero\_realisateur as num\_real, nom\_realisateur as nom\_real

from realisateur re

left outer join film f

on re.numero\_realisateur = f.numero\_realisateur

where f.numero\_realisateur is null;



# 3) Saisissez le script & expliquez le rôle des ordres SQLPLUS utilisés :

**a. REMARK :** utilisé pour écrire des commentaires

**b. SET (PAUSE, PAGESIZE, LINESIZE) :** affectation des valeurs

+ SET PAUSE 'text' : l'utilisateur doit appuyer sur Entrée pour continuer l'exécution du programme, dans ce cas la phrase '...' s'affichant le bouton Entrée

+ SET PAUSE ON : afficher le texte encadré dans '...' chaque fois que le système SQL s'arrête

+ SET PAGESIZE : préciser le nombre de lignes affichées sur chaque page

+ SET LINESIZE : préciser le nombre de caractères affichés sur une ligne avant d'en passe à une autre

**c. TTITLE, BTITLE, SKIP, CENTER, RIGHT, FORMAT, SQL.PNO, COLUMN, HEADING,JUSTIFY**

+ TTITLE : top title de chaque page

+ BTITLE : bottom title de chaque page ( utilisation idem TTITLE )

+ CENTER, RIGHT, LEFT : la position de chaque texte, utilisés par TTITLE | BTITILE

+ SKIP : sauter à une nouvelle ligne

+ FORMAT : formatter des éléments

+ SQL.PNO : compter le nombre de pages automatiquement

+ COLUMN : la colonne dans la table

+ HEADING : le titre de la colonne ( l'entête ) ('|' pour sauter à la ligne suivante)

+ JUSTIFY : l'alignement du texte

**d. BREAK ON … ON REPORT, COMPUTE NUM OF … ON … :**

+ BREAK ON [colonne] ON REPORT : spécifier la place òu la commande BREAK s'exécute (comme un report)

+ BREAK ON GENRE ON REPORT : BREAK exécute des commandes spécifiées dans COMPUTE avant de passer à une autre valeur sur la colonne GENRE ; ON REPORT affiche le résultat ( généralement à la fin des résultats )

+ COMPUTE [fonction d'agrégat] OF [colonne] ON REPORT : exécuter la fonction spécifiée ( NUM, SUM, AVG... ) en utilisant des données de la colonne indiquée, ON REPORT affiche le résultat

+ Dans certains cas, BREAK ON & COMPUTE sont utlisés ensemble

**e. CLEAR (BREAKS, COMPUTES, SCREEN)**

+ CLEAR BREAKS, CLEAR COMPUTES : supprimer toutes les commandes BREAK et COMPUTE

+ CLEAR SCREEN : effacer l'écran

# 4) Tapez la commande SELECT \* FROM ACTEUR; et commentez :

Les informations des acteurs sont affichées plus clairement, parce que les commandes dans le script de la question 3 a formatté l'affichage des résultats.

# 6) Tapez de nouveau la commande SELECT \* FROM ACTEUR; et commentez :

Les résultats s'affichent normalement comme au début, car tous les nouveaux formats ont été effacés par le script login.sql

# 7) Affichez les noms des réalisateurs avec les titres de leurs films, en présentant « joliment » le résultat de la requête (une page cadrée, dans la fenêtre de session, avec en-têtes, rupture par réalisateur, etc...) :

REMARK \* Affichez les noms des réalisateurs avec les titres de leurs films \*

REMARK \* 1 page cadrée, dans la fenêtre de session, avec en-têtes, rupture par réalisateur ...\*

set pause 'Tapez pour continuer'

set pause on

set pagesize 35

set linesize 100

clear breaks

clear computes

break on nom\_realisateur skip 2

compute number label 'Nombre de films :' of titre\_film on nom\_realisateur

TTITLE SKIP -

SKIP -

CENTER 'Resultats' -

RIGHT 'page:' FORMAT 99 SQL.PNO -

SKIP -

CENTER '----------------------' -

SKIP -

SKIP

BTITLE SKIP SKIP

column nom\_realisateur heading 'Nom des|realisateur' justify center

column titre\_film heading 'Titre du film' justify center

select nom\_realisateur, titre\_film

from realisateur re, film f

where re.numero\_realisateur = f.numero\_realisateur

order by nom\_realisateur;



# 8) Saisissez, exécutez et commentez la requête suivante :

À partir d'un nombre saisi par l'utilisateur, le programme choisit un réalisateur dont le numéro y convient.

# 9) Affichez toutes les informations concernant les acteurs dont le prénom est choisi lors de l'exécution de votre requête :

accept nom char prompt 'Entrer le prenom de l''acteur (en majuscule) : '

column numero\_acteur heading 'Numero|acteur' justify center

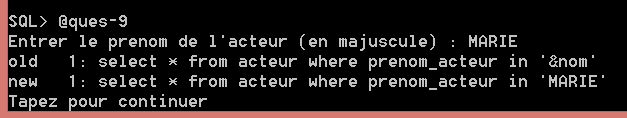
column nom\_acteur heading 'Nom|acteur' justify center

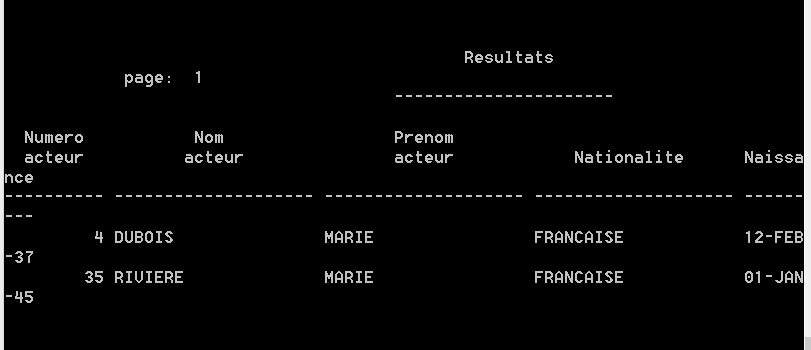
column prenom\_acteur heading 'Prenom|acteur' justify center

column nation\_acteur heading 'Nationalite' justify center

column date\_de\_naissance heading 'Naissance' justify center

select \* from acteur where prenom\_acteur in '&nom';





# Compréhension de requêtes :

## 01.

select Genre , count(NUMERO\_FILM) ,

avg(Duree)\*count(Duree)-sum(Duree)

from FILM

group by Genre ;

avg(Duree)\*count(Duree)-sum(Duree) : résultat toujours égale (ou quasiment égale) à 0.

## 02.

select R.NUMERO\_REALISATEUR,

NOM\_REALISATEUR,

count(\*) as nb\_films,

count(NUMERO\_FILM) as nb\_films,

count(distinct F.NUMERO\_REALISATEUR) as nb\_real

from REALISATEUR R

left outer join FILM F

on R.NUMERO\_REALISATEUR = F.NUMERO\_REALISATEUR

where NATION\_REALISATEUR = 'FRANCAISE'

group by R.NUMERO\_REALISATEUR , NOM\_REALISATEUR , PRENOM\_REALISATEUR;

Pour chaque réalisateur (numéro + nom), le nombre de films qu'il a réalisé.

## 03.

select \*

from ACTEUR

where NUMERO\_ACTEUR <> all (select NUMERO\_ACTEUR from ROLE) ;

Chercher l'acteur qui n'a joué dans aucun film.

<> : différent que ( != )

## 04.

select distinct PRENOM\_REALISATEUR

from REALISATEUR

where exists

(

select \*

from ACTEUR

where PRENOM\_ACTEUR = PRENOM\_REALISATEUR

);

Chercher tous les acteurs dont le prenom est le même que celui des réalisateurs.

exists : retourner 'true' si la sous-requête trouve au moins une ligne.

## 05.

select count(\*) compte , R.\*

from REALISATEUR R , REALISATEUR R\_DEUX

where R.NATION\_REALISATEUR = R\_DEUX.NATION\_REALISATEUR

group by R.NUMERO\_REALISATEUR, R.NOM\_REALISATEUR,

R.PRENOM\_REALISATEUR, R.NATION\_REALISATEUR

order by compte , R.NUMERO\_REALISATEUR ;

Chercher et grouper les réalisateurs de même nationalité, donner le nombre de réalisateurs dans chaque groupe.

## 06-a.

select NUMERO\_ACTEUR , NUMERO\_FILM , 1

from ROLE

union

select NUMERO\_ACTEUR , null , count(\*)

from ROLE

group by NUMERO\_ACTEUR

union

select null , NUMERO\_FILM , count(\*)

from ROLE

group by NUMERO\_FILM

union

select null , null , count(\*)

from ROLE

order by 1 asc nulls last , 2 asc nulls last ;

- 1 : retourne une colonne avec toute valeur = 1

- nulls last 🡪 toutes les valeurs nulles sont placées après les valeurs non-nulles

- order by <entier> : la colonne <entier> dans la requête SELECT

🡪 **OBJECTTIF** : pour chaque acteur, afficher le nombre de films qu'il a eu un rôle, puis une ligne dernière (de chaque acteur) retourne le nombre total de leurs films ( des tables crossées )

Explication en ordre :

- 1e sous-enquête : le numéro des acteurs et le numéro des films avec une colonne dont toute valeur = 1 --> pour chaque valeur de (num\_act, num\_film), on affecte un compteur = 1

- 2e sous-enquête : pour chaque acteur, le nombre total de films òu il a eu un rôle (la ligne dernière de chaque acteur avant d'en passer à un autre)

- 3e sous-enquête : pour chaque film, le nombre d'acteurs qui ont un rôle (le numéro des acteurs se met à NULL)

- 4e sous-requête : la ligne servant à compter le nombre total de lignes

## 06-b.

select NUMERO\_ACTEUR , NUMERO\_FILM , count(\*) as Total

from ROLE

group by cube( NUMERO\_ACTEUR , NUMERO\_FILM )

order by 1 asc nulls last , 2 asc nulls last ;

Même requête que la question précédente en utilisant GROUP BY CUBE(<attributs>)

## 06-c.

select NUMERO\_ACTEUR , NUMERO\_FILM , 1

from ROLE

order by 1 asc nulls last , 2 asc nulls last ;

Pour chaque acteur, le numéro de films qu'il a joué, la 3e colonne se sert d'un compteur

## 06-d.

select NUMERO\_ACTEUR , NUMERO\_FILM , count(\*)

from ROLE

group by grouping sets ( NUMERO\_ACTEUR , NUMERO\_FILM , () )

order by 1 asc nulls last , 2 asc nulls last ;

Pour chaque acteur, le nombre total de films qu'il a joué

Pour chaque film, le nombre total d'acteurs

**grouping sets** est différent que **cube** : il n'y a pas de valeurs "crossées"

## 07-a.

select REALISATEUR.NUMERO\_REALISATEUR,

ACTEUR.NUMERO\_ACTEUR,

count(FILM.NUMERO\_FILM)

from REALISATEUR

join FILM on REALISATEUR.NUMERO\_REALISATEUR = FILM.NUMERO\_REALISATEUR

join ROLE on FILM.NUMERO\_FILM = ROLE.NUMERO\_FILM

join ACTEUR on ROLE.NUMERO\_ACTEUR = ACTEUR.NUMERO\_ACTEUR

where REALISATEUR.NUMERO\_REALISATEUR <= 3

and ACTEUR.NUMERO\_ACTEUR <= 3

group by REALISATEUR.NUMERO\_REALISATEUR, ACTEUR.NUMERO\_ACTEUR

order by REALISATEUR.NUMERO\_REALISATEUR, ACTEUR.NUMERO\_ACTEUR;

Pour chaque réalisateur, le numéro de l'acteur / des acteurs avec qui il travaille, ainsi que le nombre de films que ce réalisateur et ce (ces) acteurs ont réalisé ensemble

## 07-b.

select REALISATEUR.NUMERO\_REALISATEUR,

ACTEUR.NUMERO\_ACTEUR,

count(FILM.NUMERO\_FILM)

from REALISATEUR

join FILM on REALISATEUR.NUMERO\_REALISATEUR = FILM.NUMERO\_REALISATEUR

join ROLE on FILM.NUMERO\_FILM = ROLE.NUMERO\_FILM

join ACTEUR on ROLE.NUMERO\_ACTEUR = ACTEUR.NUMERO\_ACTEUR

where REALISATEUR.NUMERO\_REALISATEUR <= 3 and ACTEUR.NUMERO\_ACTEUR <= 3

group by cube( REALISATEUR.NUMERO\_REALISATEUR , ACTEUR.NUMERO\_ACTEUR )

order by REALISATEUR.NUMERO\_REALISATEUR nulls last, ACTEUR.NUMERO\_ACTEUR nulls last ;

Même résultat que la requête précédente, mais il existe des valeurs nulles ( group by cube (...) )

## 08-a.

select GENRE , count(\*)

from FILM

group by GENRE

order by count(\*) desc;

Le nombre de films appartenant à un genre précisé.

## 08-b.

select \*

from ( select GENRE , count(\*) from FILM group by GENRE order by count(\*) desc )

where rownum <= 4 ;

Rownum : le nombre de lignes.

Le nombre de films appartenant à un genre précisé (comme la requête précédente) mais cette fois on prend seulement 4 lignes dans le résultat.

## 08-c.

select GENRE , NbFilmsDuGenre, RangNbFilmsDuGenre

from ( select GENRE , NbFilmsDuGenre ,

rank() over(order by NbFilmsDuGenre desc) RangNbFilmsDuGenre

from ( select GENRE , count(\*) NbFilmsDuGenre

from FILM

group by GENRE

order by NbFilmsDuGenre desc ) )

where RangNbFilmsDuGenre <= 4 ;

RANK : le rang n'est pas forcément une chaîne de nombres consécutifs.

Chercher les genres selon leur rang, RANK pas le même que ROWNUM (= le nombre de lignes).

## 09.

select row\_number() over(order by NATION\_ACTEUR) row\_number\_NATION\_ACTEUR ,

rank() over(order by NATION\_ACTEUR) ,

NUMERO\_ACTEUR , NOM\_ACTEUR , PRENOM\_ACTEUR , NATION\_ACTEUR

from ACTEUR

where substr(PRENOM\_ACTEUR,0,1) in ( 'B' , 'K' , 'M' , 'P' )

order by row\_number\_NATION\_ACTEUR ;

row\_number = rownum = le nombre de lignes

Afficher : les informations de l'acteur, le rang selon la nation ( rank() over(order by NATION\_ACTEUR) ), le numéro selon la ligne de la nation (row\_number() over(order by NATION\_ACTEUR) ).

## 10.

select GENRE,

rank() over(partition by GENRE order by DUREE)

NUMERO\_FILM, TITRE\_FILM, DUREE,

count(\*) over(partition by GENRE),

count(\*) over() as nombre,

min(Duree) over(partition by GENRE),

min(Duree) over() as min\_result,

max(Duree) over(partition by GENRE),

max(Duree) over()

from FILM

where trim(GENRE) not like '% %'

order by GENRE , DUREE , NUMERO\_FILM ;

The OVER clause specifies the partitioning, ordering & window "over which" the analytic function operates.

rank() over(partition by GENRE order by DUREE) : donner le rang selon la durée de chaque genre (à chaque nouveau genre, le rang est renouvelé)

count(\*) over(partition by GENRE) : compter le nombre de films dans un genre

count(\*) over() : compter le nombre de total des résultats

min(Duree) over(partition by GENRE) : la durée minimale dans l'ensemble des films de chaque genre

## 11.

select NUMERO\_ACTEUR ,

trim(NOM\_ACTEUR) || ' ' || ltrim(PRENOM\_ACTEUR),

DATE\_DE\_NAISSANCE ,

NATION\_ACTEUR ,

case when NATION\_ACTEUR = 'FRANCAISE' then 'francaise'

else 'etrangere'

end FR\_OU\_ETR

from ACTEUR

where extract(year from DATE\_DE\_NAISSANCE) between 1955 and 1960

or regexp\_like(PRENOM\_ACTEUR,'^[D-F]')

order by NOM\_ACTEUR ;

Chercher des acteurs :

- qui sont nés entre 1955 et 1960

- dont le prénom commence de ‘d’ à ‘f’ (regexp\_like(PRENOM\_ACTEUR,'^[D-F]') )

- afficher leur nationalité ( soit française, soit étrangère ),

## 12.

select NUMERO\_ACTEUR , NOM\_ACTEUR ,

nullif(NATION\_ACTEUR,'FRANCAISE') as Nationalite,

coalesce(cast(DATE\_DE\_NAISSANCE as char(15)),'inconnue') as Naissance

from ACTEUR

where substr(NOM\_ACTEUR,3,1) in ('A','E','I','O','U','Y') ;

Chercher des acteurs tels que :

- nation = null si nation = 'francaise', sinon afficher la nationalité

- coalesce( cast(DATE\_DE\_NAISSANCE as char(15)) ,'inconnue') : si DATE\_DE\_NAISSANCE est null, alors affichez 'inconnue'

- le 3e caractère dans le nom\_acteur se trouve dans un de ces caractères : ('A','E','I','O','U','Y')

## 13.

select NUMERO\_ACTEUR , NOM\_ACTEUR , DATE\_DE\_NAISSANCE ,

extract(year from DATE\_DE\_NAISSANCE) ANNEE\_DE\_NAISSANCE,

width\_bucket(1900+mod(extract(year from DATE\_DE\_NAISSANCE),100),1950,1970,4) WB

from ACTEUR

where DATE\_DE\_NAISSANCE is not null ;

- width\_bucket : WIDTH\_BUCKET lets you construct equiwidth histograms, in which the histogram range is divided into intervals that have identical size.

- mod(extract(year from DATE\_DE\_NAISSANCE),100) : extraire l'année à partir de la date de naissance, calculer le reste par le quotient extract(...) par 100

- width\_bucket(1900+mod(extract(year from DATE\_DE\_NAISSANCE),100),1950,1970,4) : créer un histogram dont l'expression donne l'année de naissance, min 1950, max 1970, histogram divisé en 4 parties de largeur identique, la fonction retourne un nombre entier qui indique dans quelle partie se trouve la valeur calculée par l'expression.

## 14.

select NUMERO\_ACTEUR , NOM\_ACTEUR , DATE\_DE\_NAISSANCE ,

( DATE\_DE\_NAISSANCE - interval '99' YEAR ) DATE\_DE\_NAISSANCE\_XXeme

from ACTEUR

where DATE\_DE\_NAISSANCE is not null

and ( DATE\_DE\_NAISSANCE - interval '59' YEAR, DATE\_DE\_NAISSANCE - interval '49' YEAR ) overlaps

( CURRENT\_DATE - interval '50' YEAR , CURRENT\_DATE - interval '30' YEAR );

- interval '99' YEAR : l'intervalle entre 99 années 0 mois

- ( DATE\_DE\_NAISSANCE - interval '99' YEAR ) : prend l'année dans DATE\_DE\_NAISSANCE – (minus) 99 années ( ex : 1921 – 21 )

- CURRENT\_DATE : la date courante

- overlaps : tester s'il y a un chevauchement de temps sur l'intervalle de temps prédéfinie

Ex : 1900 - 2000 overlaps 1950 - 1994 --> résultat : oui (1950 – 1994 )

Chercher des acteurs tels que :

- sa date de naissance non nulle

- il y a un chevauchement de temps pendant la durée (année\_naissance - 59, année\_naissance - 49) et (année\_courant - 50, année\_courante - 30)

## 15.

select stddev(NUMERO\_FILM), variance(NUMERO\_FILM),

stddev(NUMERO\_REALISATEUR), variance(NUMERO\_REALISATEUR),

corr(NUMERO\_FILM,NUMERO\_REALISATEUR)

from FILM ;

stddev : độ lệch (deviation)

variance : phương sai

corr : độ lệch tương quan (correlation coefficient)