

SQL*Plus d'Oracle : Interrogation de la base de données « Cinéma »

Objectifs : Initiation au client d'Oracle et Révision des requêtes SQL

Connexion à SQL*Plus d'Oracle (interpréteur SQL du SGBDR Oracle) sous Linux :

- connectez vous à la machine « alphascorpii » en utilisant le script <http://dept-info.labri.fr/Teaching/BDS2PUF/TD2/connect.sh> (login : oracle, mot de passe : oracle)
- lancez un shell,
- pour cette première séance, créez un répertoire de travail personnalisé « mkdir NOM_PRENOM » et allez à la racine de ce répertoire « cd NOM_PRENOM ». Ce répertoire servira à sauvegarder vos requêtes SQL dans des fichiers *.sql.
- Copiez dans votre répertoire le script ~oracle/oracle_etd.sh (« cp ~oracle/oracle_etd.sh ~oracle/NOM_PRENOM/. ») qui charge (pour la première fois dans la session les variables d'environnement nécessaires à SQL*Plus.
- Lancez l'interpréteur SQL*Plus en vous loggant en temps que etd (mot de passe : etd) : « sqlplus etd/etd »

Quelques commandes SQL*Plus :

quit	quitte la session ;
help <ordre_sql>	donne de l'aide sur la commande SQL <ordre_sql> ;
start <fichier>	charge et exécute le script SQL du fichier <fichier>.sql ;
save <fichier>	sauve le script SQL de la requête en cours dans le fichier <fichier>.sql ;
edit <fichier>	ouvre l'éditeur par défaut.

Il existe un éditeur intégré à SQL*Plus, mais il est plutôt recommandé d'utiliser votre éditeur habituel. Pour appeler votre éditeur depuis SQL*Plus, il faut d'abord définir la variable EDITOR : SQL>DEFINE_EDITOR=emacs .

L'éditeur est ensuite appelé par la commande SQL>edit <fichier>.

Vous allez utiliser la base de données « Cinéma » restreinte aux quatre relations concernant les réalisateurs, les films, les rôles et les acteurs et aux contraintes associées :

```
REALISATEUR(NUMERO_REALISATEUR, NOM_REALISATEUR, PRENOM_REALISATEUR, NATION_REALISATEUR)
FILM(NUMERO_FILM, TITRE_FILM, DATE_DE_SORTIE, DUREE, GENRE, NUMERO_REALISATEUR)
ROLE(NUMERO_ACTEUR, NUMERO_FILM, NOM_DU_ROLE)
ACTEUR(NUMERO_ACTEUR, NOM_ACTEUR, PRENOM_ACTEUR, NATION_ACTEUR, DATE_DE_NAISSANCE)
```

```
REALISATEUR.NUMERO_REALISATEUR est la clé de REALISATEUR
FILM.NUMERO_FILM est la clé de FILM
(ROLE.NUMERO_FILM, ROLE.NUMERO_ACTEUR) est la clé de ROLE
ACTEUR.NUMERO_ACTEUR est la clé de ACTEUR
valeurs (FILM.NUMERO_REALISATEUR) in valeurs (REALISATEUR.NUMERO_REALISATEUR)
valeurs (ROLE.NUMERO_FILM) in valeurs (FILM.NUMERO_FILM)
valeurs (ROLE.NUMERO_ACTEUR) in valeurs (ACTEUR.NUMERO_ACTEUR)
attributs facultatifs : PRENOM_REALISATEUR, NATION_REALISATEUR, DATE_DE_SORTIE,
                        NOM_DU_ROLE, PRENOM_ACTEUR, NATION_ACTEUR, DATE_DE_NAISSANCE
```

- 1) Donnez le schéma entités-associations correspondant au schéma relationnel ci-dessous
- 2) Requetes SQL : Ecrivez et sauvegardez dans des fichiers « .sql » les requêtes permettant d'effectuer les traitements suivants (Attention d'être bien situé dans votre répertoire NOM_PRENOM) :

- a. Toutes les informations de tous les réalisateurs triés en ordre alphabétique sur le nom,
- b. Les numéros, noms et prénoms de tous les réalisateurs de nationalité 'FRANCAISE',
- c. Les numéros, noms et prénoms des réalisateurs ayant réalisé au moins un film (sans répétition),
- d. Les numéros et les titres de tous les films avec le numéro et le nom des acteurs,
- e. Les numéros et les titres des films dans lesquels il y a au moins deux acteurs qui ont eu un rôle,
- f. Les numéros, noms et prénoms des réalisateurs dont le prénom est aussi un prénom d'acteur,
- g. Les numéros, noms et prénoms des réalisateurs ayant réalisé entre 2 et 5 films,
- 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).
- i. Le nombre d'acteurs jouant dans le film 'GARCON'
- j. Pour chaque acteur (nom, prénom), le nombre de films dans lesquels il a joué,
- k. Pour chaque acteur (nom, prénom), le premier film dans lequel il a eu un rôle (date_de_sortie la plus ancienne).

- l. Les prénoms des acteurs et des réalisateurs, sans répétition, sur une seule colonne¹,*
- m. Les prénoms communs aux réalisateurs et aux acteurs,*
- n. Les prénoms de réalisateurs qui ne sont pas des prénoms d'acteurs,*
- o. Affichez les numéros et noms des réalisateurs n'ayant réalisé aucun film en utilisant :*
 - une requête imbriquée
 - une opération ensembliste
 - une jointure externe

3) *Saisissez et exécutez le script donné ci-dessous :*

```
REMARK *****
REMARK * Script SQL*Plus d'Oracle : liste des films, avec mise en page *
REMARK *****

SET PAUSE 'Taper [RETURN] pour continuer'
SET PAUSE ON
SET PAGESIZE 35
SET LINESIZE 100
CLEAR BREAKS
CLEAR COMPUTES
CLEAR SCREEN
BREAK ON GENRE ON REPORT
COMPUTE NUM OF DUREE ON GENRE
COMPUTE NUM OF DUREE ON REPORT
TTITLE SKIP -
      SKIP -
      CENTER 'Les films par genre...' -
      RIGHT 'page:' FORMAT 99 SQL.PNO -
      SKIP -
      CENTER '-----' -
      SKIP -
      SKIP
BTITLE SKIP SKIP
COLUMN TITRE_FILM      HEADING 'Titre du|film' JUSTIFY CENTER
COLUMN GENRE          HEADING 'Genre'          JUSTIFY CENTER
COLUMN DUREE          HEADING 'Duree'          JUSTIFY CENTER FORMAT 999
COLUMN DATE_DE_SORTIE HEADING 'Date de|sortie' JUSTIFY CENTER
SELECT TITRE_FILM, GENRE, DUREE, DATE_DE_SORTIE
FROM FILM
WHERE NUMERO_FILM <= 20
ORDER BY GENRE, DATE_DE_SORTIE
```

*... et expliquez le rôle des ordres SQL*Plus suivants :*

- a. REMARK,
 - b. SET (PAUSE, PAGESIZE, LINESIZE)
 - c. TTITLE, BTITLE, SKIP, CENTER, RIGHT, FORMAT, SQL.PNO, COLUMN, HEADING, JUSTIFY
 - d. BREAK ON ... ON REPORT, COMPUTE NUM OF ... ON ...,
 - e. CLEAR (BREAKS, COMPUTES, SCREEN)
- 4) *Tapez la commande SELECT * FROM ACTEUR; et commentez*
- 5) *Créez le fichier login.sql et exécutez ce fichier à l'aide de la commande start (ou en vous déconnectant puis en vous reconnectant de nouveau à SQL*Plus) :*

```
SET PAGESIZE 20
SET LINESIZE 32767
SET PAUSE OFF
TTITLE OFF
BTITLE OFF
SET HEADING OFF
CLEAR BREAKS
CLEAR COLUMNS
CLEAR COMPUTES
CLEAR SCREEN
```

- 6) *Tapez de nouveau la commande SELECT * FROM ACTEUR; et commentez*
- 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...)*
- 8) *Saisissez, exécutez et commentez la requête suivante :*

```
ACCEPT nr NUMBER PROMPT 'Entrez le numéro du réalisateur choisi : '
SELECT * FROM REALISATEUR WHERE NUMERO_REALISATEUR = &nr;
```
- 9) *Affichez toutes les informations concernant les acteurs dont le prénom est choisi lors de l'exécution de votre requête.*

¹ Les opérations ensemblistes de SQL*Plus d'Oracle sont respectivement union, intersect et minus pour l'union, l'intersection et la différence (except)

Compréhension de requêtes.

1. Saisissez, exécutez et commentez la requête suivante :

```
select Genre , count(NUMERO_FILM) , avg(Duree)*count(Duree)-sum(Duree)
from FILM
group by Genre ;
```

2. Saisissez, exécutez et commentez la requête suivante :

```
select R.NUMERO_REALISATEUR , NOM_REALISATEUR ,
       count(*) , count(NUMERO_FILM) , count(distinct F.NUMERO_REALISATEUR)
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 ;
```

3. Saisissez, exécutez et commentez la requête suivante :

```
select *
from ACTEUR
where NUMERO_ACTEUR <> all ( select NUMERO_ACTEUR
                           from ROLE
                           ) ;
```

4. Saisissez, exécutez et commentez la requête suivante :

```
select distinct PRENOM_REALISATEUR
from REALISATEUR
where exists ( select 74552575275
               from ACTEUR
               where PRENOM_ACTEUR = PRENOM_REALISATEUR ) ;
```

5. Saisissez, exécutez et commentez la requête suivante :

```
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 ;
```

6. Saisissez, exécutez, comparez et commentez les deux requêtes suivantes :

- ```
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 ;
```
- ```
select NUMERO_ACTEUR , NUMERO_FILM , count(*)
from ROLE
group by cube( NUMERO_ACTEUR , NUMERO_FILM )
order by 1 asc nulls last , 2 asc nulls last ;
```

Saisissez, exécutez, comparez et commentez les deux requêtes suivantes :

- ```
select NUMERO_ACTEUR , NUMERO_FILM , 1
from ROLE
order by 1 asc nulls last , 2 asc nulls last ;
```
- ```
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 ;
```

7. Saisissez, exécutez, comparez et commentez les deux requêtes suivantes (qui diffèrent uniquement par leurs clauses group by et order by):

- ```
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 ;
```
- ```
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 ;

```

8. Saisissez, exécutez, comparez et commentez les deux requêtes suivantes (la sous-requête de la seconde requête étant identique à la première requête) :

- select GENRE , count(*) from FILM group by GENRE order by count(*) desc ;
- select *
from (select GENRE , count(*) from FILM group by GENRE order by count(*) desc)
where rownum <= 2 ; -- ou 4

Selon vous, peut-on ainsi afficher les ex-æquo ? Effectuez un essai en remplaçant le 2 par un 4 dans la seconde requête. Saisissez, exécutez et commentez la requête suivante (la sous-requête la plus imbriquée étant quasiment identique à la première requête, le count(*) ayant juste été renommé en NbFilmsDuGenre). Effectuez un essai en remplaçant le 2 par un 4.

```

select GENRE , NbFilmsDuGenre
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 <= 2 ; -- ou 4

```

9. Saisissez, exécutez et commentez la requête suivante :

```

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 ;

```

10. Saisissez, exécutez et commentez la requête suivante :

```

select GENRE , rank() over(partition by GENRE order by DUREE) ,
       NUMERO_FILM , TITRE_FILM , DUREE ,
       count(*) over(partition by GENRE) , count(*) over() ,
       min(Duree) over(partition by GENRE) , min(Duree) over() ,
       max(Duree) over(partition by GENRE) , max(Duree) over()
from FILM
where trim(GENRE) not like '% %'
order by GENRE , DUREE , NUMERO_FILM ;

```

11. Saisissez, exécutez et commentez la requête suivante :

```

select NUMERO_ACTEUR , trim(NOM_ACTEUR) || ' ' || ltrim(PRENOM_ACTEUR) ,
       DATE_DE_NAISSANCE ,
       NATION_ACTEUR , case when NATION_ACTEUR = 'FRANCAISE' then 'française'
                          else 'étrangère'
                          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 ;

```

12. Saisissez, exécutez et commentez la requête suivante :

```

select NUMERO_ACTEUR , NOM_ACTEUR , nullif(NATION_ACTEUR,'FRANCAISE') ,
       coalesce(cast(DATE_DE_NAISSANCE as char(8)),'inconnue')
from ACTEUR
where substr(NOM_ACTEUR,3,1) in ('A','E','I','O','U','Y') ;

```

13. Saisissez, exécutez et commentez la requête suivante :

```

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 ;

```

14. Saisissez, exécutez et commentez la requête suivante :

```

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 '99' YEAR ,
            DATE_DE_NAISSANCE - interval '99' YEAR ) overlaps
            ( CURRENT_DATE - interval '50' YEAR , CURRENT_DATE - interval '30' YEAR ) ;

```

15. *Saisissez, exécutez et commentez la requête suivante :*

```
select stddev(NUMERO_FILM) , variance(NUMERO_FILM) ,  
       stddev(NUMERO_REALISATEUR) , variance(NUMERO_REALISATEUR) ,  
       corr(NUMERO_FILM,NUMERO_REALISATEUR)  
from FILM ;
```