



Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Informatique

Spécialité : MASTER 1 Big Data Analytics

Sujet:
Rapport de TP1 ARBD

Travail présenté à Madame Kahina KESSI

Travail présenté par :
AISSANI Anouar
161835024828

TP1:

Partie IV : Langage d'interrogation de données

16. On veut connaître pour chaque employé le nom et le prénom de son supérieur hiérarchique.

```
SQL> select e1.nom, e1.prenom, e2.nom, e2.prenom
  2  from employe e1, employe e2
  3  where e1.SUPERIEUR_HIERARCHIQUE=e2.noemp;
```

NOM	PRENOM	NOM	PRENOM
Leverling	Janet	Fuller	Andrew
Peacock	Margaret	Fuller	Andrew
Buchanan	Steven	Fuller	Andrew
Emery	Patrick	Buchanan	Steven
Suyama	Michael	Emery	Patrick
Davolio	Nancy	Fuller	Andrew
Callahan	Laura	Fuller	Andrew
Dodsworth	Anne	Buchanan	Steven
Suyama	Jordan	Emery	Patrick

9 rows selected.

17. Pour chaque catégorie, donner le nombre de commandes.

```
SQL> select c.codecateg, c.nomcateg, count (d.nocom) as nb_commande
  2  from categorie c, produit p, detailcommande d
  3  where c.codecateg = p.codecateg
  4  and p.refprod = d.refprod
  5  group by c.codecateg, c.nomcateg
  6  order by c.codecateg;
```

CODECATEG	NOMCATEG	NB_COMMANDE
1	Boissons	3
2	Condiments	6
3	Desserts	11
4	Produits laitiers	10
5	Pâtes et céréales	4
6	Viandes	9
7	Produits secs	2
8	Poissons et fruits de mer	6

8 rows selected.

18. Calculer le chiffre d'affaire pour chaque produit commandé par les clients.

```
SQL> select p.refprod, p.nomprod, sum(d.prixunit*qte-(d.prixunit*d.qte*d.remise)) as chiffreaffaire
2  from produit p, detailcommande d
3  where p.refprod=d.refprod
4  group by p.refprod, p.nomprod
5  order by chiffreaffaire desc;
```

REFPROD	NOMPROD	CHIFFREAFFAIRE
29	Thüringer Rostbratwurst	31630.5
62	Tarte au sucre	13790
27	Schoggi Schokolade	8800
61	Sirop déarable	8721
18	Carnarvon Tigers	8625
60	Camembert Pierrot	6732
17	Alice Mutton	4680
69	Gudbrandsdalsost	4492.8
11	Queso Cabrales	4271.4
16	Pavlova	4175.5
64	Wimmers gute Semmelknödel	3990
2	Aniseed Syrup	2907
44	Gula Malacca	2496
55	Pâté chinois	2433.6
49	Maxilaku	2400
19	Teatime Chocolate Biscuits	2160
77	Original Frankfurter grüne Soß	2080
10	Ikura	1984
59	Raclette Courdavault	1584
57	Ravioli Angelo	1560
74	Longlife Tofu	1520
31	Gorgonzola Telino	1487.5
68	Scottish Longbreads	1485
71	Fløtemysost	1462
53	Perth Pasties	1310
1	Chai	918
63	Vegie-spread	880
50	Valkoinen suklaa	877.5
21	Sir Rodneys Scones	800
23	Tunnbröd	756
52	Filo Mix	560
54	Tourtière	414
24	Guaraná Fantástica	409.5
36	Inlagd Sill	364.8
46	Spegesild	72
13	Konbu	24

36 rows selected.

TP2

Partie I : Création des TableSpaces et des utilisateurs

1. Créer deux TableSpaces GYMNASSE_TBS et GYMNASSE_TempTBS

```
SQL> create tablespace GYMNASSE_TBS datafile 'c:\GYMNASSE_TBS.dat' size 100M autoextend on online;
Tablespace created.

SQL> create temporary tablespace GYMNASSE_TempTBS tempfile 'c:\GYMNASSE_TempTBS.dat' size 100M autoextend on;
Tablespace created.
```

2. Créer un utilisateur DBAGYMNASSE en lui attribuant les deux tablespaces créés précédemment.

3. Donner tous les privilèges à cet utilisateur.

```
SQL> create user DBAGYMNASSE identified by psw default tablespace GYMNASSE_TBS temporary tablespace GYMNASSE_TempTBS;
User created.

SQL> grant all privileges to DBAGYMNASSE;
```

Partie II : Langage de définition de données

4. Créer les relations de la base de données avec toutes les contraintes d'intégrité plus tableerreurs.

```
SQL> CREATE TABLE SPORTIFS(  
  2     IDSPORTIF INTEGER,  
  3     NOM VARCHAR(50),  
  4     PRENOM VARCHAR(50),  
  5     Sexe VARCHAR(1),  
  6     AGE INTEGER,  
  7     IDSPORTIFCONSEILLEUR INTEGER,  
  8     constraint pk_SPORTIFS PRIMARY KEY (IDSPORTIF),  
  9     constraint fk_SPORTIFS FOREIGN KEY (IDSPORTIFCONSEILLEUR) references SPORTIFS (IDSPORTIF) on delete cascade,  
 10     constraint check_SEXE CHECK (SEXE IN('F','M'))  
 11 );  
  
Table created.  
  
SQL>  
SQL> CREATE TABLE SPORTS(  
  2     IDSPORT INTEGER,  
  3     LIBELLE VARCHAR(50),  
  4     constraint pk_SPORTS PRIMARY KEY (IDSPORT)  
  5 );  
  
Table created.  
  
SQL>  
SQL> CREATE TABLE GYMNASES(  
  2     IDGYMNASE INTEGER,  
  3     NOMGYMNASE VARCHAR(50),  
  4     ADRESSE VARCHAR(50),  
  5     VILLE VARCHAR(50),  
  6     SURFACE NUMBER,  
  7     constraint pk_GYMNASE PRIMARY KEY (IDGYMNASE)  
  8 );  
  
Table created.
```

```

SQL> CREATE TABLE ARBITRER(
  2     IDSPORTIF INTEGER,
  3     IDSPORT INTEGER,
  4     constraint fk_SPORTIFS_AR FOREIGN KEY (IDSPORTIF) references SPORTIFS(IDSPORTIF) on delete cascade,
  5     constraint fk_SPORTS_AR FOREIGN KEY (IDSPORT) references SPORTS(IDSPORT) on delete cascade,
  6     constraint pk_ARBITRER PRIMARY KEY (IDSPORTIF, IDSPORT)
  7 );

Table created.

SQL>
SQL> CREATE TABLE ENTRAINER(
  2     IDSPORTIFENTRAINEUR INTEGER,
  3     IDSPORT INTEGER,
  4     constraint fk_SPORTIFS_EN FOREIGN KEY (IDSPORTIFENTRAINEUR) references SPORTIFS(IDSPORTIF) on delete cascade,
  5     constraint fk_SPORTS_EN FOREIGN KEY (IDSPORT) references SPORTS(IDSPORT) on delete cascade,
  6     constraint pk_ENTRAINER PRIMARY KEY (IDSPORTIFENTRAINEUR, IDSPORT)
  7 );

Table created.

SQL>
SQL> CREATE TABLE JOUER(
  2     IDSPORTIF INTEGER,
  3     IDSPORT INTEGER,
  4     constraint fk_SPORTIFS_JOU FOREIGN KEY (IDSPORTIF) references SPORTIFS(IDSPORTIF) on delete cascade,
  5     constraint fk_SPORTS_JOU FOREIGN KEY (IDSPORT) references SPORTS(IDSPORT) on delete cascade,
  6     constraint pk_JOUER PRIMARY KEY (IDSPORTIF, IDSPORT)
  7 );

Table created.

```

```

SQL> CREATE TABLE SEANCES(
  2     IDGYMNASE INTEGER,
  3     IDSPORT INTEGER,
  4     IDSPORTIFENTRAINEUR INTEGER,
  5     JOUR VARCHAR(10),
  6     HORAIRE NUMBER,
  7     DUREE INT,
  8     constraint fk_IDGYMNASE FOREIGN KEY (IDGYMNASE) references GYMNASES(IDGYMNASE) on delete cascade,
  9     constraint fk_SPORTS_SE FOREIGN KEY (IDSPORT) references SPORTS(IDSPORT) on delete cascade,
 10     constraint fk_SPORTIFS_SE FOREIGN KEY (IDSPORTIFENTRAINEUR) references SPORTIFS(IDSPORTIF) on delete cascade,
 11     constraint pk_SEANCES PRIMARY KEY (IDGYMNASE, IDSPORT, IDSPORTIFENTRAINEUR, JOUR, HORAIRE),
 12     constraint check_JOUR CHECK (JOUR IN ('Samedi', 'Dimanche', 'Lundi', 'Mardi', 'Mercredi', 'Jeudi', 'Vendredi'))
 13 );

```

5. Ajouter l'attribut DATECREATION de type Date dans la relation GYMNASES.

```
SQL> ALTER TABLE GYMNASES ADD DATECREATION DATE;
```

Table altered.

```
SQL> DESC GYMNASES;
```

Name	Null?	Type
IDGYMNASE	NOT NULL	NUMBER(38)
NOMGYMNASE		VARCHAR2(50)
ADRESSE		VARCHAR2(50)
VILLE		VARCHAR2(50)
SURFACE		NUMBER
DATECREATION		DATE

6. Ajouter la contrainte not null pour les attributs SEXE et AGE de la relation SPORTIF.

```
SQL> ALTER TABLE SPORTIFS MODIFY SEXE NOT NULL;
```

```
Table altered.
```

```
SQL> ALTER TABLE SPORTIFS MODIFY AGE NOT NULL;
```

```
Table altered.
```

```
SQL> DESC SPORTIFS;
```

Name	Null?	Type
-----	-----	-----
IDSPORTIF	NOT NULL	NUMBER(38)
NOM		VARCHAR2(50)
PRENOM		VARCHAR2(50)
SEXE	NOT NULL	VARCHAR2(1)
AGE	NOT NULL	NUMBER(38)
IDSPORTIFCONSEILLEUR		NUMBER(38)

7. Modifier la longueur de l'attribut PRENOM (agrandir, réduire).

```
SQL> ALTER TABLE SPORTIFS MODIFY PRENOM VARCHAR(100);
```

Table altered.

```
SQL> DESC SPORTIFS;
```

Name	Null?	Type
-----	-----	-----
IDSPORTIF	NOT NULL	NUMBER(38)
NOM		VARCHAR2(50)
PRENOM		VARCHAR2(100)
SEXE	NOT NULL	VARCHAR2(1)
AGE	NOT NULL	NUMBER(38)
IDSPORTIFCONSEILLEUR		NUMBER(38)

```
SQL> ALTER TABLE SPORTIFS MODIFY PRENOM VARCHAR(50);
```

Table altered.

```
SQL> DESC SPORTIFS;
```

Name	Null?	Type
-----	-----	-----
IDSPORTIF	NOT NULL	NUMBER(38)
NOM		VARCHAR2(50)
PRENOM		VARCHAR2(50)
SEXE	NOT NULL	VARCHAR2(1)
AGE	NOT NULL	NUMBER(38)
IDSPORTIFCONSEILLEUR		NUMBER(38)

8. Supprimer la colonne DATECREATION dans la table GYMNASE. Vérifier la suppression.

```
SQL> ALTER TABLE GYMNASES DROP COLUMN DATECREATION;
```

```
Table altered.
```

```
SQL> desc GYMNASES;
```

Name	Null?	Type
-----	-----	-----
IDGYMNASE	NOT NULL	NUMBER(38)
NOMGYMNASE		VARCHAR2(50)
ADRESSE		VARCHAR2(50)
VILLE		VARCHAR2(50)
SURFACE		NUMBER

9. Renommer la colonne ADRESSE dans la table GYMNASE par ADRESSEGYM. Vérifier.

```
SQL> ALTER TABLE GYMNASES RENAME COLUMN ADRESSE TO ADRESSEGYM;
```

```
Table altered.
```

```
SQL> DESC GYMNASES;
```

Name	Null?	Type
-----	-----	-----
IDGYMNASE	NOT NULL	NUMBER(38)
NOMGYMNASE		VARCHAR2(50)
ADRESSEGYM		VARCHAR2(50)
VILLE		VARCHAR2(50)
SURFACE		NUMBER

10. Ajouter la contrainte suivante : L'attribut LIBELLE de la table SPORTS prend ses valeurs dans le domaine {'Basket ball','Volley ball','Hand ball','Tennis', 'Hockey', 'Badmington','Ping pong','Football', 'Boxe'}.

```
SQL> ALTER TABLE SPORTS ADD constraint check_LIBELLE
      2 CHECK (LIBELLE IN('Basket ball','Volley ball','Hand ball','Tennis', 'Hockey', 'Badmington','Ping pong','Football', 'Boxe'));

Table altered.

SQL>
SQL> select constraint_name, constraint_type from user_constraints where table_name=upper('SPORTS');

```

CONSTRAINT_NAME

C
-
CHECK_LIBELLE
C
PK_SPORTS
P

Partie III : Langage de manipulation de données

11. Remplir toutes les tables par les instances représentées ci-dessus en exécutant le script insert.sql.
Quels sont les problèmes rencontrés.

Le problème rencontré: les sportifs numéro 37 et 91 n'ont pas été insérés dans la table SPORTIFS parce que leurs conseillers (**les sportifs qui ont une clé étrangère sur eux**) n'ont pas encore été insérés.

La solution: On peut les insérer après l'insertion des sportifs numéro 44 et 98.

```
149 rows selected.

SQL> select nom from sportifs where IDSPORTIF = 37;

no rows selected

SQL> select nom from sportifs where IDSPORTIF = 91;

no rows selected

SQL> INSERT INTO Sportifs VALUES(91,'BATERAOUI','Zinedine','M',30,98);

1 row created.

SQL> INSERT INTO Sportifs VALUES(37,'LAZARI','Malika','F',25,44);

1 row created.

SQL> SELECT COUNT(*) FROM SPORTIFS;

  COUNT(*)
-----
        151
```

12. Supposons que le sportif LACHEMI Bouzid a changé son conseiller par CHAADI Mourad. Que faut-il faire ?

Il faut chercher le **IDSPORTIF** du **CHAADI Mourad** puis l'ajouter au sportif **LACHEMI Bouzid** comme un **IDSPORTIFCONSEILLEUR**.

```
SQL> UPDATE SPORTIFS
2 SET IDSPORTIFCONSEILLEUR =
3   (
4     SELECT IDSPORTIF
5     FROM SPORTIFS
6     WHERE nom = 'CHAADI' and prenom = 'Mourad')
7 WHERE nom = 'LACHEMI' AND prenom = 'Bouzid';

1 row updated.

SQL> select * from sportifs where nom = 'LACHEMI' AND prenom = 'Bouzid';

IDSPORTIF|NOM                                |PRENOM                                |S|AGE|IDSPORTIFCONSEILLEUR
-----|-----|-----|-----|-----|-----
4|LACHEMI                                |Bouzid                                |M|32|65

SQL> select * from sportifs where nom = 'CHAADI' and prenom = 'Mourad';

IDSPORTIF|NOM                                |PRENOM                                |S|AGE|IDSPORTIFCONSEILLEUR
-----|-----|-----|-----|-----|-----
65|CHAADI                                |Mourad                                |M|30|

SQL>
```

13. Ajouter les deux sports 'Natation' et 'Golf'. Désactiver la contrainte pour autoriser la modification. Réactiver la contrainte.

On ne peut pas modifier une contrainte directement alors, après la désactivation de la contrainte **check_LIBELLE**, on la supprime puis recrée en ajoutant les deux sports 'Natation' et 'Golf' cette fois, enfin on l'active.

```
SQL> ALTER TABLE SPORTS DISABLE constraint check_LIBELLE;

Table altered.

SQL>
SQL> ALTER TABLE SPORTS
2 DROP CONSTRAINT check_LIBELLE;

Table altered.

SQL> ALTER TABLE SPORTS
2 ADD CONSTRAINT check_LIBELLE
3 CHECK (LIBELLE IN ('Basket ball','Volley ball','Hand ball','Tennis', 'Hockey', 'Badmington','Ping pong','Football', 'Boxe', 'Natation', 'Golf'));

Table altered.
```

```
SQL> alter table sports enable constraint check_LIBELLE;
Table altered.
SQL> select constraint_name, constraint_type from user_constraints where table_name=upper('SPORTS');

```

CONSTRAINT_NAME	CONSTRAINT_TYPE
CHECK_LIBELLE	C
PK_SPORTS	P

L'insertion des deux nouveaux sports 'Natation' et 'Golf' :

```
SQL> INSERT INTO Sports VALUES(10,'Natation');
1 row created.
SQL> INSERT INTO Sports VALUES(11,'Golf');
1 row created.
SQL>
```

14. Supprimer tous les gymnases dont la superficie est supérieure à 400 m². Quels sont les problèmes rencontrés.

Problème: On peut pas supprimer les gymnases dont la superficie est **supérieure à 400 m²** parce que la table **SEANCES** a une clé primaire sur la table **GYMNASES**.

Solution: Pour supprimer les gymnases dont la superficie est supérieure à 400 m², il faut d'abords supprimer les seances de celles gymnases, désactiver la clé étrangère ou bien ajouter "**on delete cascade**" a la fin du contrainte du clé étrangère durant la création du table **GYMNASES**.

```
SQL> DELETE FROM GYMNASSES WHERE SURFACE > 400;
DELETE FROM GYMNASSES WHERE SURFACE > 400
*
ERROR at line 1:
ORA-02292: integrity constraint (DBAGYMNASE.FK_IDGYMNASE) violated - child record found
```

Partie IV : Langage d'interrogation de données

15. Quels sont les sportifs (identifiant, nom et prénom) qui ont un âge entre 20 et 30 ans ?

```
SQL> SELECT IDSPORTIF, NOM, PRENOM, AGE FROM SPORTIFS WHERE AGE BETWEEN 20 AND 30;
```

IDSPORTIF	NOM	PRENOM	AGE
1	BOUTAHAR	Abderahim	30
2	BOUROUBI	Anis	28
3	BOUZIDI	Amel	25
5	AAKOUB	Linda	22
6	ABBAS	Sophia	30
7	HADJ	Zouhir	25
8	HAMADI	Hani	30
9	ABDELMOUMEN	Nadia	23
10	ABAD	Abdelhamid	23
11	ABAYAHIA	Amine	24
12	ABBACI	Riad	24
13	ABBACI	Mohamed	22
14	ABDELOUAHAB	Lamia	24
15	ABDEMEZIANE	Majid	25
16	BENOUADAH	Lamine	24
17	ACHAIBOU	Rachid	22
18	HOSNI	Leila	25
19	ABERKANE	Adel	25
20	AZOUG	Racim	25
21	BABACI	Mourad	22
22	BAKIR	Ayoub	25
23	BEHADI	Youcef	24
24	AMARA	Nassima	23
25	AROUEL	Lyes	23
26	BAALI	Leila	23
27	BADI	Hatem	23
29	ROUSSELI	Lamice	22
30	CHIKHI	Nidal	24
31	SETIHA	Moustapha	22
32	COTERI	Daouad	23
33	RAMELI	Sami	23
34	LEHIRACHE	Oussama	24
35	TERIKI	Yacine	24
36	DJELOUDANE	Zinedine	28
38	MESSOUNI	Ismail	24
39	MORELI	Otheman	24
40	FATAHI	Majid	23

116	BELIFA	Samia	30
117	FERRIRA	Manel	30
118	IGHOLI	Lyes	30
119	GUEMEZ	Jaouad	30
120	LECOM	Aissa	30
121	HOUAT	Aziz	30
122	BEQUETA	Aicha	30
123	RATENI	Walid	30
124	TOUAT	Yasmine	30
125	JALONI	Aimad	30
126	DEBOUBA	yasser	30
127	GASTAB	Chouaib	30
128	GIroni	Younes	30
129	DABONI	Rachid	30
130	LACHOUBI	Kamel	30
131	GALLOI	Nadira	30
132	DORONI	Yanis	30
133	LENOUCHI	Youcef	30
134	LERICHE	Hadi	30
135	MANSOUR	Lamine	30
136	LABOULAIS	Fadia	26
137	DOUDOU	Faiza	26
138	MAALEM	Lamia	26
139	BESNARD	Salma	26
140	BELHAMID	Hadjer	26
141	BOUAAZA	Asma	26
142	CORCHI	Melissa	26
143	BELAID	Jaouida	26
144	GASMI	Souad	26
145	LAAMARA	Maria	25
146	DABOUB	Ramezi	25
147	HASSINI	Nadia	25
148	KALOUNE	Maria	27
149	BELHAOUA	Besma	27
150	BELAID	Fouad	25
151	HENDI	Mouad	25
91	BATERAQUI	Zinedine	30
37	LAZARI	Malika	25

143 rows selected.

16. Quels sont les sportifs qui sont des conseillers ?

```
SQL> SELECT DISTINCT s.IDSPORTIF, s.NOM, s.PRENOM  
2 FROM SPORTIFS s, SPORTIFS p  
3 WHERE s.IDSPORTIF = p.IDSPORTIFCONSEILLEUR ORDER BY s.IDSPORTIF;
```

IDSPORTIF	NOM	PRENOM
1	BOUTAHAR	Abderahim
2	BOUROUBI	Anis
3	BOUZIDI	Amel
4	LACHEMI	Bouzid
5	AAKOUB	Linda
6	ABBAS	Sophia
7	HADJ	Zouhir
8	HAMADI	Hani
9	ABDELMOUMEN	Nadia
14	ABDELOUAHAB	Lamia
21	BABACI	Mourad
44	ADIBOU	Ibrahim
65	CHAADI	Mourad
98	RIADI	Walid

17. Quels entraîneurs n'entraînent que du hand ball ou du basket ball ?

```
SQL> SELECT IDSPORTIF, NOM, PRENOM FROM SPORTIFS WHERE IDSPORTIF IN (
  2  SELECT DISTINCT IDSPORTIFENTRAINEUR
  3  FROM ENTRAINEUR
  4  WHERE IDSPORT IN
  5  (SELECT IDSPORT
  6  FROM SPORTS
  7  WHERE LIBELLE = 'Hand ball' OR LIBELLE = 'Basket ball'));
```

IDSPORTIF	NOM	PRENOM
1	BOUTAHAR	Abderahim
2	BOUROUBI	Anis
3	BOUZIDI	Amel
4	LACHEMI	Bouزيد
7	HADJ	Zouhir
149	BELHAOUA	Besma
151	HENDI	Mouad

7 rows selected.

18. Quels sont les sportifs les plus jeunes?

```
SQL> SELECT
  2  IDSPORTIF, NOM, PRENOM, AGE
  3  FROM
  4  SPORTIFS
  5  WHERE
  6  AGE = (SELECT
  7  MIN(AGE)
  8  FROM
  9  SPORTIFS);
```

IDSPORTIF	NOM	PRENOM	AGE
5	AAKOUB	Linda	22
13	ABBACI	Mohamed	22
17	ACHAIBOU	Rachid	22
21	BABACI	Mourad	22
29	ROUSSELI	Lamice	22
31	SETIHA	Moustapha	22
41	DELHOUME	Elina	22
50	KALI	Yasser	22
52	DANDOUR	Rami	22
53	DEMMERA	Houcine	22
72	MARTALI	Bouزيد	22
92	HADJI	Jamel	22
114	MICHALIKH	Asma	22

13 rows selected.

19. Calculer la superficie moyenne des gymnases, pour chaque ville.

```
SQL> SELECT VILLE, AVG(SURFACE)
      2 FROM GYMNASES
      3 GROUP BY VILLE;
```

VILLE	AVG(SURFACE)
El Achour	600
Baba hassen	500
Chéraga	510
Bordj El Bahri	450
Mohammadia	600
Alger centre	280
Hussein Dey	500
Saoula	350
Bir Mourad Raïs	480
Dely Brahim	620
Birkhadem	450
Alger	400
Béni Messous	520
Sidi Mhamed	500
El Biar	620
El Harrach	400
Belouizdad	400
Hydra	310
Les sources	450
El Mouradia	350
Kouba	400
Bordj el kiffan	450

22 rows selected.