

DEVOIR MAISON TP 1, TP 2 OLAP :

TP 1 partie 1 :

1- Création des tables

```
SQL> CREATE TABLE analyse
2  (
3    id_analyse number(10) NOT NULL,
4    nom_analyse varchar2(50) NOT NULL
5  );
```

Table créée.

```
SQL> CREATE TABLE typeAnalyse
2  (
3    typeAnalyse varchar2(50) NOT NULL,
4    description varchar2(50) NOT NULL
5  );
```

Table créée.

```
SQL> CREATE TABLE wilaya
2  (
3    code_wilaya number(10) NOT NULL,
4    nomWilaya varchar2(50) NOT NULL,
5    CONSTRAINT wilaya PRIMARY KEY (code_wilaya)
6  );
```

Table créée.

```
SQL> CREATE TABLE specialite
2  (
3    id_spec number(10) NOT NULL,
4    nom_specialite varchar2(50) NOT NULL,
5    CONSTRAINT specialite PRIMARY KEY (id_spec)
6  );
```

Table créée.

```
SQL> CREATE TABLE grade
2  (
3    titre varchar2(50) NOT NULL,
4    description varchar2(50) NOT NULL,
5    CONSTRAINT grade PRIMARY KEY (titre)
6  );
```

Table créée.

```

SQL> CREATE TABLE hopital
  2  (
  3    code_hopital number(10) NOT NULL,
  4    nom_hopital varchar2(50) NOT NULL,
  5    id_wilaya number(10) NOT NULL,
  6    CONSTRAINT hopital PRIMARY KEY (code_hopital),
  7    CONSTRAINT fk_wilaya
  8      FOREIGN KEY (id_wilaya)
  9      REFERENCES wilaya(code_wilaya)
 10 );

```

Table créée.

```

SQL> CREATE TABLE service
  2  (
  3    code_service number(10) NOT NULL,
  4    nom_service varchar2(50) NOT NULL,
  5    code_hopital number(10) NOT NULL,
  6    CONSTRAINT service PRIMARY KEY (code_service),
  7    CONSTRAINT fk_hopital
  8      FOREIGN KEY (code_hopital)
  9      REFERENCES hopital(code_hopital)
 10 );

```

Table créée.

```

SQL> CREATE TABLE ville
  2  (
  3    codeVille number(10) NOT NULL,
  4    nom_ville varchar2(50) NOT NULL,
  5    id_wilaya number(10) NOT NULL,
  6    CONSTRAINT ville PRIMARY KEY (codeVille),
  7    CONSTRAINT fk_wilaya2
  8      FOREIGN KEY (id_wilaya)
  9      REFERENCES wilaya(code_wilaya)
 10 );

```

Table créée.

2- Remplir les tables :

```

SQL> DECLARE
  2  l char(10);
  3  d char(40);
  4  i number;
  5  begin
  6  for i in 1..4000 loop
  7  Select dbms_random.string('U', 8) into l from dual;
  8  Select dbms_random.string('U', 8) into d from dual;
  9  insert into typeAnalyse values(l,d);
 10  end loop;
 11  commit;
 12  end;
 13  /

```

Procédure PL/SQL terminée avec succès.

```

SQL> DECLARE
  2  l char(10);
  3  i number;
  4  begin
  5  for i in 1..58 loop
  6  Select dbms_random.string('U', 8) into l from dual;
  7  insert into wilaya values(i,l);
  8  end loop;
  9  commit;
 10  end;
 11  /

```

Procédure PL/SQL terminée avec succès.

```
SQL> select * from wilaya;
```

```
CODE_WILAYA  NOMWILAYA
```

```

-----
1 AYPQPUKM
2 UWOGXRPJ
3 VESTAUYU

```

Procédure PL/SQL terminée avec succès.

```
SQL> DECLARE
  2 d date; nbj number; tarif number; pat number; serv number; med number; i number;
  3 BEGIN
  4 FOR i IN 1..710314 LOOP
  5 SELECT TO_DATE( TRUNC( DBMS_RANDOM.VALUE(TO_CHAR(DATE '2014-01-01','J')
  6
  7 ,TO_CHAR(DATE '2019-12-31','J') )
  8 ),'J'
  9 ) into d FROM DUAL;
 10 SELECT trunc(DBMS_RANDOM.VALUE(5000, 100000 ),2) into nbj FROM DUAL;
 11 SELECT floor(DBMS_RANDOM.VALUE(1, 3000 ) ) into tarif FROM DUAL;
 12 SELECT floor(DBMS_RANDOM.VALUE(1, 110000 ) ) into pat FROM DUAL;
 13 SELECT floor(DBMS_RANDOM.VALUE(1, 2212) ) into serv FROM DUAL;
 14
 15 SELECT floor(DBMS_RANDOM.VALUE(1, 12300) ) into med FROM DUAL;
 16 INSERT INTO hospitalisation VALUES (i, d,nbj, tarif, pat, serv, med);
 17 END LOOP;
 18 COMMIT;
 19 END;
 20 /
```

Procédure PL/SQL terminée avec succès.

<

TP 1 Partie 2 :

1-

SQL> select * from service where code_hopital = 1 ;

CODE_SERVICE	NOM_SERVICE	CODE_HOPITAL
1542	FYSILHZH	1
94	UTYITKKQ	1
270	OFNKBKST	1
448	YOILNZPA	1
534	MXUEQZHY	1
661	QJVBTHHV	1
715	WKEDBQRL	1
745	BRCXEFMU	1
754	KSRLALBW	1
884	POOKWEDN	1
978	ZUBGKUQT	1

CODE_SERVICE	NOM_SERVICE	CODE_HOPITAL
1152	MUOHEDAQ	1
1170	TFYCULRG	1
1670	FRXKFDYX	1
1786	ZGPFTHVV	1
1933	QRJKDWBF	1
2067	PTQLUCMV	1
2072	CLKGDGWW	1
2160	QDNCLPNS	1
2173	RUEFOKAI	1

20 ligne(s) sélectionnée(s).

```
SQL> update service set nom_service = 'Cardiologie' where code_hopital = 1 AND code_service= 1542;
```

1 ligne mise à jour.

```
SQL> SELECT codePat,nomPat FROM patient p
2      left outer Join hospitalisation h ON p.codePat = h.id_patient
3      left outer join service s on s.code_service = h.id_service
4      left outer join hopital h2 on s.code_hopital = h2.code_hopital
5      WHERE h.date_hospitalisation >= TO_DATE('01/01/15', 'DD/MM/YY') and h.date_hospitalisation
<= TO_DATE('30/05/18', 'DD/MM/YY')
6      and s.nom_service = 'Cardiologie'
7
```

```
SQL> set timing on
```

```
SQL> set autotrace on explain
```

```
SQL> SELECT codePat,nomPat FROM patient p
2      left outer Join hospitalisation h ON p.codePat = h.id_patient
3      left outer join service s on s.code_service = h.id_service
4      left outer join hopital h2 on s.code_hopital = h2.code_hopital
5      WHERE h.date_hospitalisation >= TO_DATE('01/01/15', 'DD/MM/YY') and h.date_hospitalisation
<= TO_DATE('30/05/18', 'DD/MM/YY')
6      and s.nom_service = 'Cardiologie'
7      and h2.nom_hopital = 'hopital Mustapha';
```

aucune ligne sélectionnée

2-

```
SQL> update wilaya set nomWilaya = 'Oran' where code_wilaya = 31;
```

1 ligne mise à jour.

Ecoulé : 00 :00 :00.03

Plan d'exécution

Plan hash value: 2348040995

	Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
	0	UPDATE STATEMENT		1	40	2 (0)	00:00:01
	1	UPDATE	WILAYA				
*	2	INDEX UNIQUE SCAN	WILAYA	1	40	1 (0)	00:00:01

Predicate Information (identified by operation id):

```
SQL> SELECT COUNT(*) FROM patient p Left outer Join
2      hospitalisation h ON p.codePat = h.id_patient
3      left outer join service s on s.code_service = h.id_service
4      WHERE p.codeWilaya = 31
5      and s.code_hopital = 4;
```

COUNT(*)

76

Ecoulé : 00 :00 :01.46

Plan d'exécution

Plan hash value: 954394061

3-

```
SQL> SELECT h.code_hopital,h.nom_hopital,COUNT(s.code_service) AS nbr
2 FROM hopital h LEFT JOIN service s ON h.code_hopital = s.code_hopital
3 GROUP BY h.nom_hopital,h.code_hopital
4 order by h.code_hopital;
```

CODE_HOPITAL	NOM_HOPITAL	NBR
1	PZBLCGOY	20
2	LBGHTUVM	20
3	LRIBLNJH	19
4	AOURYOJO	10
5	UWJGZOOH	25
6	KPVUTPZH	21
7	AYHEMRAF	18
8	PUQECPEC	21
9	BAZELAID	25
10	RABTRWUN	19
11	PVBGVUET	17

CODE_HOPITAL	NOM_HOPITAL	NBR
12	MORGVOUE	12
13	AJCIWMTU	15
14	TLIRTXRL	32
15	FFTJRASP	15
16	UGEZUAZE	22
17	NIPYAGCV	21
18	HEAPMBNX	18
19	YSYBECQT	17
20	XTPRFCNQ	20
21	EIBGQUAI	18
22	BJDZYNZB	28

CODE_HOPITAL	NOM_HOPITAL	NBR
23	LEBEWTWU	11
24	PVULUUDU	20
25	QZBDDOFQ	18
26	BXUZDJES	22
27	EBCCULPL	18
28	WEERQYOA	22
29	YRUTGUNW	25
30	KTNIQWVG	24
31	AGLEXISX	22

4-

2211 ligne(s) sélectionnée(s).

Ecoulé : 00 :00 :53.85

Plan d'exécution

Plan hash value: 1197111787

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		89091	10M	918 (5)	00:00:12
1	SORT GROUP BY		89091	10M	918 (5)	00:00:12
* 2	HASH JOIN		89091	10M	905 (4)	00:00:11
* 3	HASH JOIN		2212	200K	9 (12)	00:00:01
4	TABLE ACCESS FULL	HOPITAL	113	4520	3 (0)	00:00:01
5	TABLE ACCESS FULL	SERVICE	2212	114K	5 (0)	00:00:01
* 6	TABLE ACCESS FULL	HOSPITALISATION	89131	3046K	895 (4)	00:00:11

5-

112 ligne(s) sélectionnée(s).

Ecoulé : 00 :00 :01.81

Plan d'exécution

Plan hash value: 484082663

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		306K	33M	1056 (18)	0:00:13
1	SORT GROUP BY		306K	33M	1056 (18)	0:00:13
2	NESTED LOOPS		306K	33M	1012 (14)	0:00:13
* 3	HASH JOIN		306K	29M	912 (4)	0:00:11
* 4	HASH JOIN		2212	142K	9 (12)	0:00:01
5	TABLE ACCESS FULL	HOPITAL	113	4520	3 (0)	0:00:01
6	TABLE ACCESS FULL	SERVICE	2212	57512	5 (0)	0:00:01
* 7	TABLE ACCESS FULL	HOSPITALISATION	306K	10M	898 (4)	0:00:11
* 8	INDEX UNIQUE SCAN	PATIENT	1	13	0 (0)	0:00:01


```

SQL> set timing on
SQL> set autotrace on explain
SQL> SELECT h.code_hopital,h.nom_hopital,count(ho.id_hospitalisation)as nmbrhost
  2 FROM hopital h JOIN service s on h.code_hopital = s.code_hopital
  3 JOIN hospitalisation ho on ho.ID_service = s.code_service
  4 JOIN patient p on p.CodePat=ho.ID_patient
  5 WHERE ho.date_hospitalisation >='01/01/2015' and ho.date_hospitalisation<'30/05/2018'
  6 GROUP BY h.code_hopital,h.nom_hopital
  7 HAVING COUNT(ho.id_hospitalisation)>10000
  8 ORDER BY h.code_hopital;

```

aucune ligne sélectionnée

Ecoulé : 00 :00 :00.86

Plan d'exécution

Plan hash value: 3791328530

```

-----
| Id | Operation          | Name          | Rows | Bytes | Cost (%CPU)|
Time |                    |               |      |      |             |
-----
|  0 | SELECT STATEMENT    |               |  306K|  33M|  1056 (18)|
00:00:13 |
|*  1 | FILTER              |               |      |      |             |
|      |                    |               |      |      |             |
|  2 | SORT GROUP BY       |               |  306K|  33M|  1056 (18)|
00:00:13 |

```

7-

```

SQL> SELECT h.code_hopital,h.nom_hopital,max(ho.id_hospitalisation)as maxhost
  2 FROM hopital h JOIN service s on h.code_hopital = s.code_hopital
  3 JOIN hospitalisation ho on ho.ID_service = s.code_service
  4 JOIN patient p on p.CodePat=ho.ID_patient
  5 WHERE ho.date_hospitalisation >='01/01/2015' and ho.date_hospitalisation<'30/05/2018'
  6 GROUP BY h.code_hopital,h.nom_hopital
  7 HAVING COUNT(ho.id_hospitalisation)>10000
  8 ORDER BY h.code_hopital;

```

aucune ligne sélectionnée

Ecoulé : 00 :00 :00.63

Plan d'exécution

Plan hash value: 3791328530

```

-----
| Id | Operation          | Name          | Rows | Bytes | Cost (%CPU)|
Time |                    |               |      |      |             |
-----
|  0 | SELECT STATEMENT    |               |  306K|  37M|  1056 (18)|
00:00:13 |
|*  1 | FILTER              |               |      |      |             |
|      |                    |               |      |      |             |

```

7-

```
SQL> SELECT h.code_hopital,h.nom_hopital,max(ho.id_hospitalisation)as maxhost
2 FROM hospital h JOIN service s on h.code_hopital = s.code_hopital
3 JOIN hospitalisation ho on ho.ID_service = s.code_service
4 JOIN patient p on p.CodePat=ho.ID_patient
5 WHERE ho.date_hospitalisation >='01/01/2015' and ho.date_hospitalisation<'30/05/2018'
6 GROUP BY h.code_hopital,h.nom_hopital
7 HAVING COUNT(ho.id_hospitalisation)>10000
8 ORDER BY h.code_hopital;
```

aucune ligne sélectionnée

Ecoulé : 00 :00 :00.63

Plan d'exécution

Plan hash value: 3791328530

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		306K	37M	1056 (18)
* 1	FILTER				

8-

Ecoulé : 00 :00 :00.01

```
SQL> CREATE VIEW q8 as
2 SELECT h.code_hopital,h.nom_hopital,max(ho.id_hospitalisation)as maxhost
3 FROM hospital h JOIN service s on h.code_hopital = s.code_hopital
4 JOIN hospitalisation ho on ho.ID_service = s.code_service
5 JOIN patient p on p.CodePat=ho.ID_patient
6 WHERE ho.date_hospitalisation >='01/01/2015' and ho.date_hospitalisation<'30/05/2018'
7 GROUP BY h.code_hopital,h.nom_hopital
8 HAVING COUNT(ho.id_hospitalisation)>10000
9 ORDER BY h.code_hopital;
```

Vue créée.

Ecoulé : 00 :00 :00.25

SQL> |

TP2 :

1-

```
Connecté à :  
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production  
With the Partitioning, OLAP and Data Mining options
```

```
SQL> set timing on  
SQL> set autotrace on explain  
SQL> CREATE MATERIALIZED VIEW VM1  
2  BUILD IMMEDIATE  
3  REFRESH COMPLETE ON DEMAND  
4  AS SELECT ho.id_hospitalisation,ho.date_hospitalisation  
5  FROM hospitalisation ho  
6  WHERE ho.id_service ='16';
```

Vue matérialisée créée.

```
Ecoulé : 00 :00 :08.16  
SQL> |
```

2-

```
SQL> CREATE MATERIALIZED VIEW LOG ON scott.hospitalisation  
2  TABLESPACE users  
3  WITH PRIMARY KEY  
4  INCLUDING NEW VALUES;
```

Journal de vue matérialisée créé.

```
Ecoulé : 00 :00 :00.26  
SQL> CREATE MATERIALIZED VIEW VM2  
2  BUILD IMMEDIATE  
3  REFRESH FAST ON DEMAND  
4  AS SELECT ho.id_hospitalisation,ho.date_hospitalisation  
5  FROM hospitalisation ho  
6  WHERE ho.id_service ='16';
```

Vue matérialisée créée.

```
Ecoulé : 00 :00 :03.11
```

3-

INSERTION :

```
SQL> INSERT INTO Hospitalisation values (710315,'21/02/2022',30,15000,20,16,55);
```

1 ligne créée.

Ecoulé : 00 :00 :00.15

Plan d'exécution

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	
0	INSERT STATEMENT		1	34	1 (0)	00:00:01	

```
SQL>
```

```
SQL> execute DBMS_MVIEW.REFRESH('UM1');
```

Procédure PL/SQL terminée avec succès.

Ecoulé : 00 :00 :02.97

```
SQL> execute DBMS_MVIEW.REFRESH('UM2');
```

Procédure PL/SQL terminée avec succès.

Ecoulé : 00 :00 :00.52

SUPPRESSION:

```
SQL> DELETE FROM Hospitalisation where ID_hospitalisation=710315;
```

1 ligne supprimée.

Ecoulé : 00 :00 :00.03

Plan d'exécution

Plan hash value: 657308471

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time	

0	DELETE STATEMENT		1	19	3 (0)	00:00:01	
1	DELETE	HOSPITALISATION					
* 2	INDEX UNIQUE SCAN	HOSPITALISATION	1	19	2 (0)	00:00:01	

Predicate Information (identified by operation id):

2 - access("ID_HOSPITALISATION"=710315)

SQL>

SQL> execute DBMS_MVIEW.REFRESH('VM1');

Procédure PL/SQL terminée avec succès.

Ecoulé : 00 :00 :00.25

SQL> execute DBMS_MVIEW.REFRESH('VM2');

MODIFICATION :

```
SQL> UPDATE Hospitalisation set ID_Service = 15 WHERE date_hospitalisation='21/02/2022';
```

```
0 ligne(s) mise(s) à jour.
```

```
Ecoulé : 00 :00 :00.06
```

```
Plan d'exécution
```

```
Plan hash value: 1098132565
```


Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time

0	UPDATE STATEMENT		206	3502	900 (4)	00:00:11
1	UPDATE	HOSPITALISATION				
* 2	TABLE ACCESS FULL	HOSPITALISATION	206	3502	900 (4)	00:00:11


```
Predicate Information (identified by operation id):
```

```
2 - filter("DATE_HOSPITALISATION">=TO_DATE('2022-02-21 00:00:00',
'yyyy-mm-dd hh24:mi:ss'))
```

```
SQL>
```

```
SQL> execute DBMS_MVIEW.REFRESH('VM1');
```

```
Procédure PL/SQL terminée avec succès.
```

```
Ecoulé : 00 :00 :00.20
```

```
SQL> execute DBMS
```

4- La machine n'a pas exécuté

```
SQL> CREATE MATERIALIZED VIEW VM3
```

```
2 BUILD DEFERRED
```

```
3 REFRESH COMPLETE ON COMMIT
```

```
4 AS SELECT ho.id_hospitalisation,ho.date_hospitalisation
```

```
5 FROM hospitalisation ho JOIN service s ON ho.id_service = s.code_service
```

```
6 WHERE s.code_hopital='2';
```

```
WHERE s.code_hopital='2'
```

```
*
```

```
ERREUR à la ligne 6 :
```

```
ORA-12054: impossible de définir l'attribut de régénération ON COMMIT pour la
vue matérialisée
```

5-

```
SQL> CREATE MATERIALIZED VIEW UM4
  2  BUILD IMMEDIATE
  3  REFRESH COMPLETE ON COMMIT
  4  AS
  5  SELECT s.code_service,count(ho.id_hospitalisation)as nmbrhost
  6  FROM service s JOIN hospitalisation ho ON ho.id_service = s.code_service
  7  GROUP BY code_service;
```

Vue matérialisée créée.