Tp3

Exercice 1 :

1) Créer la table « members » suivante sachant que la clé primaire de cette table est « member_id » :

```
SQL> connect sys as sysdba
Enter password:
Connected.
SQL> create table members(
  2  member_id number primary key,
  3  first_name varchar2(10),
  4  last_name varchar2(10),
  5  gender varchar2(5),
  6  DOB date,
  7  email varchar2(20));

Table created.
SQL> INSERT INTO members
(member_id,first_name,last_name,gender,dob,email) VALUES (1,
'pepi','elice','F','04/03/84','peice0@trellian.com');

1 row created.
SQL> INSERT INTO members
(member_id,first_name,last_name,gender,dob,email) VALUES (2,
'barr','wabersich','M','04/08/76','bwabersich1@china.co');

1 row created.
SQL> INSERT INTO members
(member_id,first_name,last_name,gender,dob,email) VALUES (3,
'gretal','grassick','F','15/08/84','ggrassick@delici.com');

1 row created.
```

2)La table "members" a comme clé primaire "member_id". Oracle va donc créer un index par défaut sur cette colonne. Donner la requête nécessaire pour afficher cet index.

```
SQL> select index_name from dba_indexes where table_name='MEMBERS';

INDEX_NAME
------------------------------
SYS_C008139
```

3)On suppose, qu'un utilisateur de cette table souhaite chercher les membres par nom (last_name) et trouve l'exécution de la requête assez lente. On peut créer un index sur la colonne " last_name " afin d'accélérer la recherche. Donner la requête nécessaire.

SQL> create index last_name_ind on members(last_name);

Index created.

4)Réafficher les index de la table "members".

SQL> select index_name from dba_indexes where table_name='MEMBERS';

INDEX_NAME

SYS_C008139
LAST_NAME_IND

5) Chercher les membres qui ont le nom "Poole".

SQL> select * from members where last_name='elice';

 MEMBER_ID FIRST_NAME LAST_NAME GENDE DOB   EMAIL
---------- ---------- ---------- ----- -------- --------------------
         1 pepi   elice   F   04-03-84 peice0@trellian.com

6)Vérifier si la requête utilise l'index "members_last_name_i" dans la recherche en utilisant la clause " EXPLAIN PLAN FOR ".

SQL> EXPLAIN PLAN FOR select * from members where last_name='elice';

Explained.

SQL> select plan_table_output from table(DBMS_XPLAN.DISPLAY);

PLAN_TABLE_OUTPUT

Plan hash value: 617060928

| Id | Operation | Name | Rows | Bytes | Cost (%CPU
)| Time |


```
PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |              |     1 |    52 |     1   (0
)| 00:00:01 |

|   1 |  TABLE ACCESS BY INDEX ROWID| MEMBERS      |     1 |    52 |     1
(0
)| 00:00:01 |

|*  2 |   INDEX RANGE SCAN          | LAST_NAME_IND |     1 |       |     1   (0
)| 00:00:01 |

--------------------------------------------------------------------------------
-------------

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------


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

   2 - access("LAST_NAME"='elice')

Note
-----
   - dynamic sampling used for this statement (level=2)

18 rows selected.
```

7)Supprimer l'index "last_name_ind".

```
SQL> drop index last_name_ind;

Index dropped.
```

8)Créer un nouveau index appelé « members_name_i » sur les colonnes nom (last_name) et prénom (first_name)

```
SQL> create index members_name_i on members(last_name,first_name);

Index created.
```

9)Chercher les membres dont le nom commence par la lettre 'w' et le prénom par 'b'

```
SQL> select * from members where last_name like 'w%' and first_name like 'b
%
' ;

 MEMBER_ID FIRST_NAME LAST_NAME  GENDE DOB      EMAIL
---------- ---------- ---------- ----- -------- --------------------
         2 barr       wabersich  M     04-08-76 bwabersich1@china.co
```

10)Donner le plan d'exécution de la requête précédente pour vérifier l'utilisation du l'index « members_name_i ».

```
SQL> EXPLAIN PLAN FOR select * from members where last_name like 'a%' and fi
rst_name like 'b%' ;

Explained.

SQL> select plan_table_output from table(DBMS_XPLAN.DISPLAY);

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------
Plan hash value: 1332024799

--------------------------------------------------------------------------------
--------------

| Id  | Operation                    | Name          | Rows  | Bytes | Cost (%CP
U)| Time     |

--------------------------------------------------------------------------------
--------------


PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |               |     1 |    52 |     1   (
0)| 00:00:01 |

|   1 |  TABLE ACCESS BY INDEX ROWID| MEMBERS       |     1 |    52 |     1
(
0)| 00:00:01 |
```

```
|*  2 |   INDEX RANGE SCAN          | MEMBERS_NAME_I |     1 |       |     1
(
0)| 00:00:01 |

-------------------------------------------------------------------------------
-------------

PLAN_TABLE_OUTPUT
-------------------------------------------------------------------------------


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

   2 - access("LAST_NAME" LIKE 'a%' AND "FIRST_NAME" LIKE 'b%')
       filter("LAST_NAME" LIKE 'a%' AND "FIRST_NAME" LIKE 'b%')

Note
-----
   - dynamic sampling used for this statement (level=2)

19 rows selected.
```

Exercice 2 :

1. Quel est la structure de la table « EMPLOYEES » ?

```
DESC hr.employees;
```

2. Quel est le nombre de lignes de la table « EMPLOYEES » ?

```
SELECT COUNT(*) FROM hr.employees;
```

3. Vérifier le plan d'exécution de la commande : select * from hr.employees;

Commencer par créer le plan d'exécution

```
EXPLAIN PLAN FOR SELECT * FROM hr.employees;
```

Consulter le plan d'exécution créé

```
SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);
```

Interpréter les résultats obtenus pour les colonnes Operation, Object, Rows et Time.

Signification	Colonne
Type d'opération exécutée (TABLE ACCESS FULL, INDEX RANGE SCAN, etc.)	Operation
Nom de l'objet (table, index) utilisé	Object
Nombre estimé de lignes traitées	Rows
Temps estimé pour l'opération	Time

Si l’operation est TABLE ACCESS FULL, cela signifie que toute la table est scannée

4. Pour chacune des instructions suivantes, vérifier à chaque fois les plans d’exécution et interpréter les résultats :

select count(*) from hr.employees ;

EXPLAIN PLAN FOR SELECT COUNT(*) FROM hr.employees;
SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);

select first_name, last_name from hr.employees;

Même logique que précédemment. S’il n’y a pas d’index sur ces colonnes, ce sera aussi TABLE ACCESS FULL.

EXPLAIN PLAN FOR
SELECT first_name, last_name FROM hr.employees;

SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);

select first_name from hr.employees where salary>9000;

Avant d’avoir un index, tu verras un TABLE ACCESS FULL.

EXPLAIN PLAN FOR

SELECT first_name FROM hr.employees WHERE salary > 9000;

SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);

Créer un index pour la colonne « salary ».

CREATE INDEX idx_salary ON hr.employees(salary);

Réexécuter la commande précédente.

EXPLAIN PLAN FOR SELECT first_name FROM hr.employees WHERE salary > 9000;
SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY);