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

Rapport de TP1 ENDO

Travail présenté à Monsieur SELMOUNE Nazih

Travail présenté par :
AISSANI Anouar
161835024828

Année Universitaire: 2021/2022

PARTIE 01

1- Créer un compte utilisateur Master avec tous les privilèges.

```
SQL> create user Master identified by psw;

User created.

SQL> grant all privileges to Master;

Grant succeeded.
```

2- Donnez le modèle relationnel associé au diagramme de classe.

Wilaya(CodeWilaya, NomWilaya)
Ville(CodeVille, NomVille, CodeWilaya*)
Client(NumClient, NomClient, SexeClient, codeVille*)
TypeLigne(CodeTypeLigne, TypeLigne)
Ligne(NumeroLigne, NumClient*, CodeTypeLigne*)
Destinataire(CodeOperateurDstinataire, NomOperateurDstinataire)
TypeAppel(CodeTypeAppel, TypeAppel)
Appel(CodeAppel, DureeAppel, DateaAppel Date, NumeroLign*, CodeOperateurDstinataire*, CodeTypeAppel*)

3- Utiliser le compte Master pour créer le modèle physique associé (tables, contraintes de clés primaires et étrangères).

```
SQL> connect Master/psw
Connected.
SQL> CREATE TABLE Wilaya (
  2       CodeWilaya number(10),
  3       NomWilaya varchar(10),
  4       constraint pk_wilaya PRIMARY KEY (codeWilaya)
  5  );

Table created.
```

```
SQL> CREATE TABLE Ville(
  2      CodeVille number(10),
  3      NomVille varchar(10),
  4      CodeWilaya number(10),
  5      constraint pk_ville PRIMARY KEY (CodeVille),
  6      constraint fk_wilaya FOREIGN KEY(CodeWilaya)
         references Wilaya(CodeWilaya)
  7  );

Table created.

SQL> CREATE TABLE Client(
  2      NumClient number(10),
  3      NomClient varchar(10),
  4      SexeClient varchar(1),
  5      codeVille number(10),
  6      constraint pk_client PRIMARY KEY (NumClient),
  7      constraint fk_ville FOREIGN KEY (CodeVille)
         references Ville(CodeVille),
  8      constraint checkSexeClient Check (sexeClient in ('F','H'))
  9  );

Table created.

SQL> CREATE TABLE TypeLigne(
  2      CodeTypeLigne number(10),
  3      TypeLigne varchar(10),
  4      constraint pk_typeLigne PRIMARY KEY (CodeTypeLigne)
  5  );

Table created.
```

```
SQL> CREATE TABLE Ligne(
  2      NumeroLigne number(10),
  3      NumClient number(10),
  4      CodeTypeLigne number(10),
  5      constraint pk_ligne PRIMARY KEY(NumeroLigne),
  6      constraint fk_client FOREIGN KEY (NumClient)
         references Client(NumClient),
  7      constraint fk_typeLigne FOREIGN KEY (CodeTypeLigne)
         references TypeLigne(CodeTypeLigne)
  8  );

Table created.

SQL> CREATE TABLE Destinataire(
  2      CodeOperateurDstinataire number(10),
  3      NomOperateurDstinataire varchar(50),
  4      constraint pk_destinataire PRIMARY KEY (CodeOperateurDstinataire)
  5  );

Table created.

SQL> CREATE TABLE TypeAppel(
  2      CodeTypeAppel number(10),
  3      TypeAppel varchar(20),
  4      constraint pk_typeTable PRIMARY KEY (CodeTypeAppel)
  5  );

Table created.

SQL> CREATE TABLE Appel(
  2      CodeAppel number(10),
  3      DureeAppel number(10),
  4      DateaAppel Date,
  5      NumeroLigne number(10),
  6      CodeOperateurDstinataire number(10),
  7      CodeTypeAppel number(10),
  8      constraint pk_codeAppel PRIMARY KEY(CodeAppel),
  9      constraint fk_ligne FOREIGN KEY (NumeroLigne)
```

```
        references Ligne(NumeroLigne),
 10     constraint fk_destinataire FOREIGN KEY (CodeOperateurDstinataire)
        references Destinataire (CodeOperateurDstinataire),
 11     constraint fk_codeTypeAppel FOREIGN KEY (CodeTypeAppel)
        references TypeAppel (CodeTypeAppel)
 12  );

Table created.
```

4- Remplir en utilisant du code PL/SQL par des valeurs aléatoires mais en respectant les contraintes, les tables Ville(547 instances), Wilaya (58 instance), Client(1065566 instances).

4.1- Adopter une codification numérique séquentielle simple pour toutes les clés.

- **Wilaya:** de 1 à 58
- **Ville:** de 1 à 547
- **Client:** de 1 à 1065566

4.2- La longueur des clés

- On donne la longueur **10** pour toutes les clés.

4.3- L'attribut Sexe doit prendre ses valeurs dans l'ensemble{'F', 'H'}

```
SQL> CREATE TABLE SEXE(
  2  IDSEXE number(1),
  3  SEXE varchar(1),
  4  constraint pk_sexe PRIMARY KEY (IDSEXE)
  5  );

Table created.

SQL> INSERT INTO SEXE VALUES(1, 'F');

1 row created.

SQL> INSERT INTO SEXE VALUES(2, 'H');

1 row created.
```

4.4- Remplir les tables Wilaya, Ville, Client:

```
SQL> DECLARE
  2      Wilaya char(10);
  3      CodeW number;
  4
  5      begin
  6          for CodeW in 1..58 loop
  7              SELECT dbms_random.string('U', 8) into Wilaya from dual;
  8              INSERT INTO Wilaya VALUES(codeW, Wilaya);
  9          end loop;
 10          commit;
 11      end;
 12      /

 PL/SQL procedure successfully completed.

SQL> DECLARE
  2      Ville char(10);
  3      CodeW number;
  4      CodeV number;
  5
  6      begin
  7          for CodeV in 1..547 loop
  8              SELECT dbms_random.string('U', 8) into Ville from dual;
  9              SELECT floor(dbms_random.value(1, 58.9)) into CodeW from dual;
 10              INSERT INTO Ville VALUES(codeV, Ville, codeW);
 11          end loop;
 12          commit;
 13      end;
 14      /

 PL/SQL procedure successfully completed.
```

```
SQL> DECLARE
  2      Client char(10);
  3      CodeV number;
  4      NumClient number;
  5      SexeClient varchar(1);
  6
  7      begin
  8          for NumClient in 1..1065566 loop
  9              SELECT dbms_random.string('U', 10) into Client from dual;
 10              SELECT floor(dbms_random.value(1, 547.9)) into CodeV from dual;
 11              Select SEXE into SexeClient from SEXE where IDSEXE = (SELECT
                 TRUNC(DBMS_RANDOM.value(1,2.9)) from dual);
 12              INSERT INTO Client VALUES(NumClient, Client, SexeClient,
                 codeV);
 13          end loop;
 14          commit;
 15      end;
 16      /

 PL/SQL procedure successfully completed.
```

PARTIE 02

1- Remplir table TypeAppel (2 instances: Nationale, Internationale).

```
SQL> INSERT INTO TypeAppel VALUES(1, 'Nationale');

1 row created.

SQL> INSERT INTO TypeAppel VALUES(2, 'Internationale');

1 row created.
```

2- Remplir en utilisant du code PL/SQL par des valeurs aléatoires mais en respectant les contraintes, les tables TypeLign (10 instances), Ligne (1500255 instance), Destinataire (522 instances), Appel (35002200 instances).

```
SQL> DECLARE
  2      TypeL char(10);
  3      CodeTL number;
  4
  5      begin
  6          for CodeTL in 1..10 loop
  7              SELECT dbms_random.string('U', 10) into TypeL from dual;
  8              INSERT INTO TypeLigne VALUES(CodeTL, TypeL);
  9          end loop;
 10          commit;
 11      end;
 12      /

 PL/SQL procedure successfully completed.
```

```
SQL> DECLARE
  2
  3      numLigne number;
  4      NumClient number;
  5      CodeTypeLigne number;
  6
  7      begin
  8          for numLigne in 1..1500255 loop
  9              SELECT floor(dbms_random.value(1, 1065566.9)) into numClient
                 from dual;
 10              SELECT floor(dbms_random.value(1, 10.9)) into CodeTypeLigne from
                 dual;
 11              INSERT INTO Ligne VALUES(numLigne, NumClient, CodeTypeLigne);
 12          end loop;
 13          commit;
 14      end;
 15      /

 PL/SQL procedure successfully completed.

SQL> DECLARE
  2
  3      codeOD number;
  4      nomOD char(10);
  5
  6      begin
  7          for codeOD in 1..522 loop
  8              SELECT dbms_random.string('U', 10) into nomOD from dual;
  9              INSERT INTO Destinataire VALUES(codeOD, nomOD);
 10          end loop;
 11          commit;
 12      end;
 13      /

 PL/SQL procedure successfully completed.
```

```
SQL> DECLARE
  2       CodeApp number;
  3       Duree number;
  4       DateApp date;
  5       codeTA number;
  6       NumL number;
  7       CodeOD number;
  8   BEGIN
  9       FOR CodeApp IN 1.. 3500220 LOOP
 10           SELECT floor(dbms_random.value(1, 60.9)) into Duree from dual;
 11           SELECT TO_DATE(trunc(dbms_random.value(to_char(date
              '2020-01-01','J'), to_char(date '2021-12-31','J'))),'J') into
              DateApp from dual;
 12           SELECT floor(dbms_random.value(1,2.9)) into codeTA from dual;
 13           SELECT floor(dbms_random.value(1,1500255.9)) into NumL from dual;
 14           SELECT floor(dbms_random.value(1,522.9)) into codeOD from dual;
 15           INSERT INTO Appel VALUES (CodeApp, Duree, DateApp, NumL, codeOD,
              CodeTA);
 16       END LOOP;
 17       COMMIT;
 18       END;
          /
 PL/SQL procedure successfully completed.
```

4- Ecrire et exécuter les requêtes et donner le temps d'excécution.

4.a- Quel est le nombre d'appel effectués de chaque wilaya entre (01/01/2021, et 30/01/2021)?

Temps d'excécution: 00:00:05.01

```
SQL> SELECT w.CodeWilaya, count(a.CodeAppel) AS NOMBRE_APPEL
  2  FROM Wilaya w, Ville v, Client c, Ligne l, Appel a
  3  WHERE w.CodeWilaya = v.CodeWilaya AND v.CodeVille = c.CodeVille
  4  AND C.NumClient = l.NumClient AND l.NumeroLigne = a.NumeroLigne
  5  AND TO_DATE(DateaAppel) BETWEEN (TO_DATE('01-01-2021', 'dd/mm/yyyy'))AND
     (TO_DATE('30-01-2021', 'dd/mm/yyyy'))
  6  GROUP BY w.CodeWilaya
  7  Order by w.CodeWilaya;
```

CODEWILAYA	NOMBRE_APPEL
1	1563
2	1046
3	2615
4	3155
5	2520
6	1391
7	3706
8	1882
9	2597
10	2613
11	2313
12	1553
13	2587
14	1343
15	3465
16	2839
17	3155
18	2362
19	463
20	3981
21	2625
22	1813
23	2928
24	2011
25	3246
26	1544
27	1869
28	2662
29	3983
30	2301
31	2995
32	3435
33	1788
34	3443

```
        35 |         2833
        36 |         3140
        37 |         3150
        38 |         2007
        39 |         4742
        40 |         1203
        41 |         2109
        42 |         2112
        43 |         2660
        44 |         3395
        45 |         2408
        46 |         2283
        47 |         2040
        48 |         1628
        49 |         3458
        50 |         1844
        51 |         1754
        52 |         2923
        53 |         2106
        54 |         1820
        55 |         1858
        56 |         3161
        57 |         2925
        58 |         2037

58 rows selected.

Elapsed: 00:00:05.01
```

```
Execution Plan
----------------------------------------------------------
Plan hash value: 1964707393

-------------------------------------------------------------------------------------
| Id  | Operation             | Name   | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT      |        |    58 |  2494 |  7118   (5)| 00:00:01 |
|   1 |  SORT GROUP BY        |        |    58 |  2494 |  7118   (5)| 00:00:01 |
|*  2 |   HASH JOIN           |        |  8751 |   367K|  7116   (5)| 00:00:01 |
|*  3 |    TABLE ACCESS FULL  | VILLE  |   547 |  3829 |     3   (0)| 00:00:01 |
|*  4 |    HASH JOIN          |        |  8751 |   307K|  7113   (5)| 00:00:01 |
|*  5 |     HASH JOIN         |        |  8751 |   222K|  5995   (6)| 00:00:01 |
|*  6 |      TABLE ACCESS FULL| APPEL  |  8751 |   119K|  4905   (6)| 00:00:01 |
|   7 |      TABLE ACCESS FULL| LIGNE  |  1500K|    17M|  1079   (3)| 00:00:01 |
|   8 |     TABLE ACCESS FULL | CLIENT |  1065K|    10M|  1110   (2)| 00:00:01 |
-------------------------------------------------------------------------------------

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

   2 - access("V"."CODEVILLE"="C"."CODEVILLE")
   3 - filter("V"."CODEWILAYA" IS NOT NULL)
   4 - access("C"."NUMCLIENT"="L"."NUMCLIENT")
   5 - access("L"."NUMEROLIGNE"="A"."NUMEROLIGNE")
   6 - filter(TO_DATE(INTERNAL_FUNCTION("DATEAAPPEL"))>=TO_DATE('
              2021-01-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss') AND
              TO_DATE(INTERNAL_FUNCTION("DATEAAPPEL"))<=TO_DATE(' 2021-01-30
              00:00:00', 'syyyy-mm-dd hh24:mi:ss'))
```

4.b- Quel est le nombre d'appel effectués par type d'appel par année?

Temps d'excécution: 00:00:01.28

```
SQL> SELECT CodeTypeAppel, EXTRACT(YEAR FROM DateaAppel) AS YEAR, count(*) AS
mombre_Appel
  2  FROM Appel
  3  GROUP BY CodeTypeAppel, EXTRACT(YEAR FROM DateaAppel)
  4  ORDER BY EXTRACT(YEAR FROM DateaAppel), CodeTypeAppel;

CODETYPEAPPEL |       YEAR | MOMBRE_APPEL
------------- | ---------- | ------------
            1 |       2020 |       922453
            2 |       2020 |       830804
            1 |       2021 |       919992
            2 |       2021 |       826971

Elapsed: 00:00:01.28

Execution Plan
----------------------------------------------------------
Plan hash value: 1475727400

-------------------------------------------------------------------------------
| Id  | Operation          | Name  | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |       |  1033 | 11363 |  4973   (7)| 00:00:01 |
|   1 |  SORT GROUP BY     |       |  1033 | 11363 |  4973   (7)| 00:00:01 |
|   2 |   TABLE ACCESS FULL| APPEL |  3500K|    36M|  4711   (2)| 00:00:01 |
-------------------------------------------------------------------------------
```

4.c- Quelle est la wilaya dont le nombre d'appels est maximal en 2020?

Temps d'excécution: 00:00:03.09

```
SQL> SELECT w.CodeWilaya, count(a.CodeAppel) AS MAX_NBR_APPEL
  2  FROM Wilaya w, Ville v, Client c, Ligne l, Appel a
  3  WHERE w.CodeWilaya = v.CodeWilaya AND v.CodeVille = c.CodeVille
  4  AND C.NumClient = l.NumClient AND l.NumeroLigne = a.NumeroLigne
  5  AND EXTRACT(YEAR FROM a.DateaAppel) = 2020
  6  GROUP BY w.CodeWilaya
  7  Order by count(a.CodeAppel) desc
  8  FETCH FIRST 1 ROWS ONLY;

CODEWILAYA | MAX_NBR_APPEL
---------- | -------------
        39 |         58479

Elapsed: 00:00:03.09

Execution Plan
----------------------------------------------------------
Plan hash value: 2797103045

-------------------------------------------------------------------------------------------
| Id  | Operation                | Name   | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT         |        |     1 |    52 |  7060   (4)| 00:00:01 |
|   1 |  SORT ORDER BY           |        |     1 |    52 |  7060   (4)| 00:00:01 |
|*  2 |   VIEW                   |        |     1 |    52 |  7059   (4)| 00:00:01 |
|*  3 |    WINDOW SORT PUSHED RANK|        |    58 |  2494 |  7059   (4)| 00:00:01 |
|   4 |     HASH GROUP BY        |        |    58 |  2494 |  7059   (4)| 00:00:01 |
|*  5 |      HASH JOIN           |        | 35002 |  1469K|  7054   (4)| 00:00:01 |
|*  6 |       TABLE ACCESS FULL  | VILLE  |   547 |  3829 |     3   (0)| 00:00:01 |
|*  7 |       HASH JOIN          |        | 35002 |  1230K|  7050   (4)| 00:00:01 |
|*  8 |        HASH JOIN         |        | 35002 |   888K|  5932   (5)| 00:00:01 |
|*  9 |         TABLE ACCESS FULL | APPEL  | 35002 |   478K|  4841   (5)| 00:00:01 |
|  10 |         TABLE ACCESS FULL | LIGNE  |  1500K|    17M|  1079   (3)| 00:00:01 |
```

```
|  11 |        TABLE ACCESS FULL  | CLIENT |  1065K|    10M|  1110   (2)| 00:00:01 |
-------------------------------------------------------------------------------------

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

   2 - filter("from$_subquery$_006"."rowlimit_$$_rownumber"<=1)
   3 - filter(ROW_NUMBER() OVER ( ORDER BY COUNT(*) DESC )<=1)
   5 - access("V"."CODEVILLE"="C"."CODEVILLE")
   6 - filter("V"."CODEWILAYA" IS NOT NULL)
   7 - access("C"."NUMCLIENT"="L"."NUMCLIENT")
   8 - access("L"."NUMEROLIGNE"="A"."NUMEROLIGNE")
   9 - filter(EXTRACT(YEAR FROM INTERNAL_FUNCTION("A"."DATEAAPPEL"))=2020)
```

- Quelle est la wilaya dont le nombre d'appels est maximal en 2021?

Temps d'excécution: 00:00:03.03

```
SQL> SELECT w.CodeWilaya, count(a.CodeAppel) AS MAX_NBR_APPEL
  2  FROM Wilaya w, Ville v, Client c, Ligne l, Appel a
  3  WHERE w.CodeWilaya = v.CodeWilaya AND v.CodeVille = c.CodeVille
  4  AND C.NumClient = l.NumClient AND l.NumeroLigne = a.NumeroLigne
  5  AND EXTRACT(YEAR FROM a.DateaAppel) = 2021
  6  GROUP BY w.CodeWilaya
  7  Order by count(a.CodeAppel) desc
  8  FETCH FIRST 1 ROWS ONLY;

CODEWILAYA | MAX_NBR_APPEL
---------- | -------------
        39 |         57742

Elapsed: 00:00:03.03

Execution Plan
----------------------------------------------------------
Plan hash value: 2797103045

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

```
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT           |         |     1 |    52 |  7060   (4)| 00:00:01 |
|   1 |  SORT ORDER BY             |         |     1 |    52 |  7060   (4)| 00:00:01 |
|*  2 |   VIEW                     |         |     1 |    52 |  7059   (4)| 00:00:01 |
|*  3 |    WINDOW SORT PUSHED RANK|         |    58 |  2494 |  7059   (4)| 00:00:01 |
|   4 |     HASH GROUP BY          |         |    58 |  2494 |  7059   (4)| 00:00:01 |
|*  5 |      HASH JOIN             |         | 35002 |  1469K|  7054   (4)| 00:00:01 |
|*  6 |       TABLE ACCESS FULL    | VILLE   |   547 |  3829 |     3   (0)| 00:00:01 |
|*  7 |       HASH JOIN            |         | 35002 |  1230K|  7050   (4)| 00:00:01 |
|*  8 |        HASH JOIN           |         | 35002 |   888K|  5932   (5)| 00:00:01 |
|*  9 |         TABLE ACCESS FULL | APPEL   | 35002 |   478K|  4841   (5)| 00:00:01 |
|  10 |         TABLE ACCESS FULL | LIGNE   |  1500K|    17M|  1079   (3)| 00:00:01 |
|  11 |        TABLE ACCESS FULL  | CLIENT |  1065K|    10M|  1110   (2)| 00:00:01 |
-------------------------------------------------------------------------------------

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

   2 - filter("from$_subquery$_006"."rowlimit_$$_rownumber"<=1)
   3 - filter(ROW_NUMBER() OVER ( ORDER BY COUNT(*) DESC )<=1)
   5 - access("V"."CODEVILLE"="C"."CODEVILLE")
   6 - filter("V"."CODEWILAYA" IS NOT NULL)
   7 - access("C"."NUMCLIENT"="L"."NUMCLIENT")
   8 - access("L"."NUMEROLIGNE"="A"."NUMEROLIGNE")
   9 - filter(EXTRACT(YEAR FROM INTERNAL_FUNCTION("A"."DATEAAPPEL"))=2021)
```