

# Rapport TP 1 + TP 2

Compréhension, préparation et  
alimentation de la source de données



Réalisé en binôme par :

BENSALAH Kawthar / ABBACI Khaled

Numero du binôme : 22

Master 2 IL - Groupe 1

USTHB 2019/2020

## **Introduction**

À travers ce TP, nous allons préparer la source de données ce qui permet, par la suite, de limiter les erreurs qui peuvent survenir lors du traitement et de faciliter l'accès des utilisateurs à l'intégralité des données.

# TP 1

## Objectif du TP

*Compréhension et Préparation de la source de données*

## Réponse 1 :

- *Création d'un tablespace par défaut :*

```
SQL> CREATE TABLESPACE DefaultTBS
2      DATAFILE 'C:\DefaultTBSFile.dat'
3      SIZE 100M AUTOEXTEND ON ONLINE;

Tablespace créé.

SQL>
```

- *Création d'un tablespace temporaire :*

```
SQL> CREATE TEMPORARY TABLESPACE TempTBS
2      TEMPFILE 'C:\TempTBSFile.dat'
3      SIZE 100M AUTOEXTEND ON;

Tablespace créé.

SQL>
```

- *Création de l'utilisateur Master en lui attribuant les deux tablespaces créés précédemment :*

```
SQL> CREATE USER Master
2      IDENTIFIED BY psw
3      DEFAULT TABLESPACE DefaultTBS
4      TEMPORARY TABLESPACE TempTBS;

Utilisateur créé.

SQL>
```

- *Attribution de tous les privilèges à l'utilisateur Master :*

```
SQL> GRANT ALL PRIVILEGES TO Master;
Autorisation de privilèges (GRANT) acceptée.
SQL>
```

## Réponse 2 :

- *Le modèle relationnel associé au diagramme Entité-Association donné :*

Pour passer du modèle Entité-Association au modèle relationnel :

- Nous avons traduit les entités en relations.
- Pour les traits connectés à l'entité et qui sont de cardinalité [0,1] ou [1,1], nous avons ajouté les identifiants des entités connectées à l'autre bout en tant que clefs étrangères.

**Client** ( NumClient, NomClient , AdrClient , TelClient, DNClient ) ;

**Wilaya** ( CodeWilaya , NomWilaya ) ;

**Ville** ( CodeVille , NomVille, DependWilaya\* ) ;

**Banque** ( CodeBanque, NomBanque ) ;

**Agence** ( NumAgence , NomAgence , TelAgence, AppartienBanque\*, SeTrouveVille\* ) ;

**Type\_Compte** ( CodeType , LibType ) ;

**Compte** ( NumCompte , DateOuverture, PossedeClient\*, est\_domicileAg\*, AppartienType\_Compte\* ) ;

**Operation** ( CodeOp, DateOp, HeurOp , TypeOp , Montant, VersementCompte\*, RetraitCompte\* ) ;

### Réponse 3 :

- *Création du modèle physique associé :*
- *Se connecter en tant que Master :*

```
SQL> Disconnect;  
Déconnecté de Oracle Database 11g Enterprise Edition Release 11.2.0.2.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options  
SQL> Connect Master/psw;  
Connecté.  
SQL>
```

- *Création de la table Client :*

```
SQL> CREATE TABLE Client (  
2   NumClient Number(6) PRIMARY KEY,  
3   NomClient VARCHAR2(50),  
4   AdrClient VARCHAR2(100),  
5   TelClient VARCHAR2(10),  
6   DNClient DATE  
7   );  
  
Table créée.
```

- *Création de la table Wilaya :*

```
SQL> CREATE TABLE wilaya (  
2   Codewilaya Number(6) PRIMARY KEY,  
3   NomWilaya VARCHAR2(50)  
4   );  
  
Table créée.  
  
SQL>
```

- *Création de la table Ville :*

```
SQL> CREATE TABLE Ville (  
2   CodeVille Number(6) PRIMARY KEY,  
3   NomVille VARCHAR2(10),  
4   DependWilaya Number(6),  
5   CONSTRAINT FK_ville  
6       FOREIGN KEY (DependWilaya)  
7       REFERENCES Wilaya(Codewilaya)  
8   );
```

Table créée.

SQL>

- *Création de la table Banque :*

```
SQL> CREATE TABLE Banque (  
2   CodeBanque Number(6) PRIMARY KEY,  
3   NomBanque VARCHAR2(50)  
4   );
```

Table créée.

SQL>

- *Création de la table Agence :*

```
SQL> CREATE TABLE Agence (  
2   NumAgence Number(6) PRIMARY KEY,  
3   NomAgence VARCHAR2(50),  
4   TelAgence VARCHAR2(10),  
5   AppartientBanque Number(6),  
6   SeTrouveVille Number(6),  
7   CONSTRAINT FK_Agence1  
8       FOREIGN KEY (AppartientBanque)  
9       REFERENCES Banque(CodeBanque),  
10  CONSTRAINT FK_Agence2  
11      FOREIGN KEY (SeTrouveVille)  
12      REFERENCES ville(Codeville)  
13  );
```

Table créée.

SQL>

- *Création de la table Type\_Compte :*

```
SQL> CREATE TABLE Type_Compte (  
2   CodeType Number(1) PRIMARY KEY,  
3   LibType VARCHAR2(50)  
4       check(LibType IN ('Epargne','courant'))  
5  );
```

Table créée.

SQL>



- *Création de la table Compte :*

```
SQL> CREATE TABLE Compte (  
2   NumCompte Number(6) PRIMARY KEY,  
3   DateOuverture DATE,  
4   PossedeClient Number(6),  
5   est_domicileAg Number(6),  
6   AppartientType_Compte Number(6),  
7   CONSTRAINT FK_Compte1  
8       FOREIGN KEY (PossedeClient)  
9       REFERENCES Client(NumClient),  
10  CONSTRAINT FK_Compte2  
11  FOREIGN KEY (est_domicileAg)  
12  REFERENCES Agence(NumAgence),  
13  CONSTRAINT FK_Compte3  
14  FOREIGN KEY (AppartientType_Compte)  
15  REFERENCES Type_Compte(CodeType)  
16 );  
  
Table créée.
```

- *Création de la table Operation :*

```
SQL> CREATE TABLE Operation (  
2   CodeOp Number(6) PRIMARY KEY,  
3   DateOp DATE,  
4   HeurOp Varchar2(5),  
5   TypeOp VARCHAR2(1)  
6       check(TypeOp IN ('1','2')),  
7   Montant number(10,2),  
8   VersementCompte Number(6),  
9   RetraitCompte Number(6),  
10  CONSTRAINT FK_OP1  
11  FOREIGN KEY (VersementCompte)  
12  REFERENCES Compte(NumCompte),  
13  CONSTRAINT FK_OP2  
14  FOREIGN KEY (RetraitCompte)  
15  REFERENCES Compte(NumCompte)  
16 );  
  
Table créée.  
  
SQL>
```

- *Vérification de la création des tables :*

```
SQL> desc Client;
Nom                                     NULL ?   Type
-----
NUMCLIENT                             NOT NULL NUMBER(6)
NOMCLIENT                             VARCHAR2(50)
ADRCLIENT                             VARCHAR2(100)
TELCLIENT                             VARCHAR2(10)
DNCLIENT                             DATE

SQL> desc wilaya;
Nom                                     NULL ?   Type
-----
CODEWILAYA                             NOT NULL NUMBER(6)
NOMWILAYA                             VARCHAR2(50)

SQL> desc Ville;
Nom                                     NULL ?   Type
-----
CODEVILLE                             NOT NULL NUMBER(6)
NOMVILLE                             VARCHAR2(10)
DEPENDWILAYA                           NUMBER(6)

SQL> desc Banque;
Nom                                     NULL ?   Type
-----
CODEBANQUE                             NOT NULL NUMBER(6)
NOMBANQUE                             VARCHAR2(50)

SQL> desc Agence;
Nom                                     NULL ?   Type
-----
NUMAGENCE                             NOT NULL NUMBER(6)
NOMAGENCE                             VARCHAR2(50)
TELAGENCE                             VARCHAR2(10)
APPARTIENTBANQUE                       NUMBER(6)
SETROUVEVILLE                         NUMBER(6)

SQL> desc Type_Compte;
Nom                                     NULL ?   Type
-----
CODETYPE                               NOT NULL NUMBER(1)
LIBTYPE                               VARCHAR2(50)
```

```
SQL> desc Compte;
Nom                                     NULL ?   Type
-----
NUMCOMPTE                             NOT NULL NUMBER(6)
DATEOUVERTURE                         DATE
POSSEDECLIENT                         NUMBER(6)
EST_DOMICILEAG                        NUMBER(6)
APPARTIENTTYPE_COMPTE                 NUMBER(6)

SQL> Desc Operation;
Nom                                     NULL ?   Type
-----
CODEOP                                NOT NULL NUMBER(6)
DATEOP                                DATE
HEUOP                                 VARCHAR2(5)
TYPEOP                                VARCHAR2(1)
MONTANT                               NUMBER(10,2)
VERSEMENTCOMPTE                       NUMBER(6)
RETRAITCOMPTE                         NUMBER(6)

SQL>
```

# TP 2

## Objectif du TP

*Alimentation des sources de données*

## Réponse 1 :

- *Remplissage de la table Client :*

```
SQL> DECLARE
  2  nom char(10);
  3  adr char(100);
  4  tel char(10);
  5  dn DATE;
  6  i number;
  7  begin
  8      for i in 1..100000 loop
  9          select dbms_random.string('U',8)
10             into nom
11             from dual;
12          select dbms_random.string('U',30)
13             into adr
14             from dual;
15          select dbms_random.string('U',10)
16             into tel
17             from dual;
18          select TO_DATE(TRUNC(dbms_random.value(
19              TO_CHAR(DATE '2015-01-01', 'J'),
20              TO_CHAR(DATE '2018-12-31', 'J'))), 'J')
21             into dn
22             from dual;
23          insert into Client
24             values (i,nom,adr,tel,dn);
25      end loop
26  COMMIT;
27  end;
28  /

Procédure PL/SQL terminée avec succès.

SQL>
```

- *Remplissage de la table Wilaya :*

```
SQL> DECLARE
  2  nom char(10);
  3  I number;
  4  begin
  5      for i in 1..48 loop
  6          select dbms_random.string('U',8)
  7              into nom
  8              from dual;
  9          insert into wilaya
 10              values (i,nom);
 11      end loop
 12  COMMIT;
 13  end;
 14  /

Procédure PL/SQL terminée avec succès.

SQL>
```

- *Remplissage de la table Ville :*

```
SQL> DECLARE
  2  v char(10);
  3  w number;
  4  I number;
  5  begin
  6      for i in 1..330 loop
  7          select dbms_random.string('U',8)
  8              into v
  9              from dual;
 10          select FLOOR(dbms_random.value(1,48.9))
 11              into w
 12              from dual;
 13          insert into ville
 14              values (i,v,w);
 15      end loop
 16  COMMIT;
 17  end;
 18  /

Procédure PL/SQL terminée avec succès.
```

- *Remplissage de la table Banque :*

```
SQL> DECLARE
  2  nom char(10);
  3  I number;
  4  begin
  5      for i in 1..10 loop
  6          select dbms_random.string('U',8)
  7              into nom
  8              from dual;
  9          insert into Banque
 10              values (i,nom);
 11      end loop
 12  COMMIT;
 13  end;
 14  /

Procédure PL/SQL terminée avec succès.
```

- *Remplissage de la table Agence :*

```
SQL> DECLARE
  2  nom char(10);
  3  tel char(10);
  4  ab number;
  5  stv number;
  6  I number;
  7  begin
  8      for i in 1..12300 loop
  9          select dbms_random.string('U',8)
10             into nom
11             from dual;
12          select dbms_random.string('U',8)
13             into tel
14             from dual;
15          select FLOOR(dbms_random.value(1,10.9))
16             into ab
17             from dual;
18          select FLOOR(dbms_random.value(1,330.9))
19             into stv
20             from dual;
21          insert into Agence
22             values (i,nom,tel,ab,stv);
23      end loop
24  COMMIT;
25  end;
26  /

Procédure PL/SQL terminée avec succès.

SQL>
```

- *Remplissage de la table Type\_Compte :*

```
SQL> begin
  2      insert into Type_Compte
  3      values (1,'Epargne');
  4      insert into Type_Compte
  5      values (2,'Courant');
  6  COMMIT;
  7  end;
  8  /

Procédure PL/SQL terminée avec succès.

SQL>
```

- *Remplissage de la table Compte :*

```
SQL> DECLARE
  2  do DATE;
  3  pc number;
  4  eda number;
  5  atc number;
  6  I number;
  7  begin
  8      for i in 1..200000 loop
  9          select TO_DATE(TRUNC(dbms_random.value(
10              TO_CHAR(DATE '2016-01-01', 'J'),
11              TO_CHAR(DATE '2018-12-31', 'J'))), 'J')
12              into do
13              from dual;
14          select FLOOR(dbms_random.value(1,100000.9))
15              into pc
16              from dual;
17          select FLOOR(dbms_random.value(1,12300.9))
18              into eda
19              from dual;
20          select FLOOR(dbms_random.value(1,2.9))
21              into atc
22              from dual;
23          insert into Compte
24              values (i,do,pc,eda,atc);
25      end loop
26  COMMIT;
27  end;
28  /
```

Procédure PL/SQL terminée avec succès.

SQL>



- *Remplissage de la table Operation :*

```
SQL> DECLARE
2  d DATE;
3  heure char(5);
4  typeop char(1);
5  montant number;
6  cpt number;
7  i number;
8  begin
9      for i in 1..610314 loop
10         select TO_DATE(TRUNC(dbms_random.value(
11             TO_CHAR(DATE '2015-01-01', 'J'),
12             TO_CHAR(DATE '2018-12-31', 'J'))), 'J')
13             into d
14             from dual;
15         select trunc(dbms_random.value(5000,100000),2)
16             into montant
17             from dual;
18         select dbms_random.string('U',5)
19             into heure
20             from dual;
21         select FLOOR(dbms_random.value(1,2.9))
22             into typeop
23             from dual;
24         select FLOOR(dbms_random.value(1,200000))
25             into cpt
26             from dual;
27         insert into operation
28         values (i,d,heure,typeop,montant,
29             decode(typeop,1,cpt,null),
30             decode(typeop,2,cpt,null));
31     end loop
32 COMMIT;
33 end;
34 /
```

Procédure PL/SQL terminée avec succès.

- *Vérifier que les tables sont bien remplies :*

```
SQL> select count(*) from Client;

COUNT(*)
-----
100000

SQL> select NumClient,NomClient,DNClient
2      from Client
3      where NumClient<5;

NUMCLIENT  NOMCLIENT  DNCLIENT
-----
1  WUMPJPRI  03/09/17
2  GWEZPYML  03/07/16
3  QLLQENUW  18/04/17
4  JVGGNFRV  12/09/17

SQL> select count(*) from Wilaya;

COUNT(*)
-----
48

SQL> select Codewilaya,NomWilaya
2      from Wilaya
3      where Codewilaya<5;

CODEWILAYA  NOMWILAYA
-----
1  YGGKNBNK
2  RNCWYPJX
3  JYASYBGM
4  HRTEZWQH
```

```
SQL> select count(*) from ville;

COUNT(*)
-----
330

SQL> select CodeVille,NomVille
2      from Ville
3      where CodeVille<5;

CODEVILLE  NOMVILLE
-----
1  ADGHQHNA
2  PXPILUVB
3  ZBVKIZJZ
4  YEYSHAHT

SQL> select count(*) from Banque;

COUNT(*)
-----
10

SQL> select CodeBanque,NomBanque
2      from Banque
3      where CodeBanque<5;

CODEBANQUE  NOMBANQUE
-----
1  OJFFVSER
2  JLTEQZBJ
3  LLVBBFPC
4  DCEJWOND
```

```

SQL> select count(*) from Agence;

COUNT(*)
-----
12300

SQL> select NumAgence,AppartientBanque,SeTrouveVille
2      from Agence
3      where NumAgence<5;

NUMAGENCE APPARTIENTBANQUE SETROUVEVILLE
-----
1          2              74
2          10             34
3          3              314
4          5              221

SQL> select * from Type_Compte;

CODETYPE LIBTYPE
-----
1 Epargne
2 Courant

SQL> select count(*) from Compte;

COUNT(*)
-----
200000

SQL> select NumCompte,AppartientType_Compte
2      from Compte
3      where NumCompte<5;

NUMCOMPTE APPARTIENTTYPE_COMPTE
-----
1          2
2          2
3          1
4          2

SQL> select count(*) from Operation;

COUNT(*)
-----
610314

SQL> select count(*) from Operation where typeop='1';

COUNT(*)
-----
321290

SQL> select count(*) from Operation where typeop='2';

COUNT(*)
-----
289024

SQL> select codeop,typeop,VersementCompte,RetraitCompte
2      from Operation
3      where codeop<5;

CODEOP T VERSEMENTCOMPTE RETRAITCOMPTE
-----
1 2 114632
2 2 127604
3 2 83264
4 1 132469

```

## **Conclusion**

Après la réalisation des TP 1 et 2, nous avons conçu une base de données bien structurée et alimentée, ce qui facilite la phase des traitements et d'analyse par la suite.