

# Rapport de TPs

## Création, Modification du schéma insertion

### Création des relations de la base

Voici les instructions pour pouvoir créer les différentes tables avec leurs clés.

1)

```
CREATE TABLE CLI(CodeCli number(5) Constraint PK_CLI PRIMARY  
KEY,NomCli char(20),Pays char(30));
```

```
CREATE TABLE PROD(NumProd number(5) Constraint PK_PRO PRIMARY  
KEY,NomProd char(30),TypeProd char(10));
```

```
CREATE TABLE COM(NumCom number(5) Constraint PK_COM PRIMARY  
KEY,CodeCli Number(5) Constraint COM_REF_CLI REFERENCES Cli  
,FraisPort number(4),AnCom number(4));
```

```
CREATE TABLE DET(NumCom number(5) Constraint DET_REF_COM  
REFERENCES Com,NumProd number(5) Constraint DET_REF_PRO  
REFERENCES Prod,Qte number(5),Remise number(5),Constraint PK_DET  
PRIMARY KEY (NumCom, NumProd));
```

2)

Il y a un ordre de création à respecter: c'est par rapport aux clés étrangères. Par exemple la table CLI doit être créée avant COM car COM possède une clé étrangère qui référence la table CLI.

### Modification des tables

5)

```
ALTER TABLE CLI MODIFY NomCli char(30);
```

6)

```
ALTER TABLE CLI ADD Telephone char(30);
```

7)

```
ALTER TABLE PROD ADD PrixUnit number(5);
```

8)

```
ALTER TABLE PROD ADD Constraint NN_PROD check NomProd NOT NULL;
```

### **Insertion de données**

9)

```
INSERT INTO CLI(CodeCli,NomCli,Pays) VALUES (1,'bibì','galsen');
```

```
INSERT INTO CLI(CodeCli,NomCli,Pays) VALUES (2,'laye','galsen');
```

```
INSERT INTO CLI(CodeCli,NomCli,Pays) VALUES (3,'laly','france');
```

```
INSERT INTO PROD(NumProd,NomProd,TypeProd) VALUES  
(1,'chips','aliment');
```

```
INSERT INTO PROD(NumProd,NomProd,TypeProd) VALUES  
(2,'jean','vetement');
```

```
INSERT INTO COM(NumCom,CodeCli,FraisPort,AnCom) VALUES  
(1,2,1000,2010);
```

```
INSERT INTO COM(NumCom,CodeCli,FraisPort,AnCom) VALUES  
(2,1,5000,2009);
```

```
INSERT INTO COM(NumCom,CodeCli,FraisPort,AnCom) VALUES  
(5,3,5000,2009);
```

```
INSERT INTO COM(NumCom,CodeCli,FraisPort,AnCom) VALUES  
(4,2,5000,1995);
```

```
INSERT INTO DET(NumCom,NumProd,Qte,remise) VALUES (1,2,2,0);
```

```
INSERT INTO DET(NumCom,NumProd,Qte,remise) VALUES (2,1,5,10);
```

11)

```
update CLI set NomCli=UPPER(NomCli);
```

13)

```
delete from CLI where Pays='france';
```

```
delete from DET where DET.NumCom IN(select DET.NumCom FROM
COM,DET WHERE DET.NumCom=COM.NumCom AND COM.AnCom<1996);
delete from COM WHERE AnCom<1996;
delete from CLI WHERE NomCli IN (select CLI.NomCli from CLI,COM
WHERE CLI.CodeCli=COM.CodeCli AND COM.AnCom<1996);
```

```
select * from CLI;
select * from COM;
```

suppression des données : à voir  
delete from CLI;

16)  
Il y un ordre à suivre pour supprimer les tables: il faut  
supprimer par rapport aux dépendances  
drop table DET;  
drop table COM;  
drop table CLI;  
drop table PROD;

### **Exécution à partir de fichiers**

17)  
Pour exécuter le fichier create on tape la commande:  
start CREATE;

Pour loader les fichiers sql:  
quit ou exit :pr sortir de sdlplus  
sqlldr /: pr entrer ds sqlloader

Pour loader les fichiers direct il y a un ordre on met:

Pour loader le fichier CLI.sql:  
sqlldr / /users/mpass-2009/20502811/CLI /users/mpass-  
2009/20502811/cliLog

Pour loader le fichier COM.sql:  
sqlldr / /users/mpass-2009/20502811/COM /users/mpass-  
2009/20502811/ComLog

19)

Il y a un ordre de chargement à respecter sinon il y a des  
erreurs avec les contraintes sur les clés étrangères

Lors de l'exécution il y a eu deux erreurs : la première sur l'année et l'autre sur un numéro de client avec un format incorrect. Il faut modifier le fichier puis ne pas recharger tout le fichier sinon il y aura une erreur, juste rajouter les deux lignes modifiées.

```
sqlldr / /users/mpass-2009/20502811/FOU /users/mpass-2009/20502811/FouLog
```

```
sqlldr / /users/mpass-2009/20502811/PRO /users/mpass-2009/20502811/ProLog
```

```
sqlldr / /users/mpass-2009/20502811/DET /users/mpass-2009/20502811/DetLog
```

## SQL et Programmation SQL

1)

```
SELECT NOMCLI  
FROM CLIENT;
```

2)

```
SELECT NOMCLI  
FROM CLI  
WHERE PAYS='France'  
ORDER BY NOMCLI ASC;
```

3)

```
SELECT NOMCLI  
FROM CLI  
WHERE PAYS='France' OR Pays='Royaume-Uni';
```

4)

```
SELECT NomPro , NomCli  
FROM PRO, DET, COM , CLI  
WHERE PRO.NumPro=DET.NumPro  
AND DET.NumCom=COM.NumCom AND COM.NumCli= CLI.NumCli  
AND CLI.NomCli='Antonio';
```

```
SELECT DISTINCT NomPro  
FROM PRO, DET, COM , CLI  
WHERE PRO.NumPro=DET.NumPro  
AND DET.NumCom=COM.NumCom
```

```
AND COM.NumCli= CLI.NumCli  
AND CLI.NomCli='Antonio';
```

```
5)  
SELECT NomFou  
FROM FOU , PRO  
WHERE FOU.NumFou = PRO.NumFou  
AND PrixUnit<50;
```

```
6)  
SELECT DISTINCT NomFou  
FROM FOU , PRO  
WHERE FOU.NumFou = PRO.NumFou  
AND PrixUnit<50 OR PrixUnit>400;
```

```
7)  
SELECT DISTINCT NomCLi  
FROM COM, CLI  
WHERE CLI.NumCli=COM.NumCli  
AND COM.AnCom=1992 OR COM.AnCom=1993;
```

```
8)  
SELECT NomFou, PrixUnit FROM FOU, PRO  
WHERE FOU.NumFou=PRO.NumFou  
AND NomPro='Chef' ORDER BY PrixUnit ASC;
```

```
9)  
SELECT NomFou FROM FOU, PRO WHERE FOU.NumFou=PRO.NumFou AND  
NomPro='Chef' AND (SELECT Min(PrixUnit) FROM PRO WHERE  
NomPro='Chef' ) = PRO.PrixUnit ;
```

```
10)  
SELECT NomFou, NomPro FROM PRO, FOU WHERE FOU.NumFou=PRO.NumFou  
ORDER BY NomFou;
```

```
11)  
SELECT NomFou FROM FOU, CLI, PRO, COM, DET  
WHERE FOU.NumFou=PRO.NumFou  
AND PRO.NumPro=DET.NumPro  
AND DET.NumCom=COM.NumCom  
AND COM.NumCli =CLI.NumCli  
AND CLI.NomCli=FOU.NomFou;
```

```
12)  
SELECT DISTINCT NomFou  
FROM FOU, PRO  
WHERE FOU.NumFou=PRO.NumFou
```

```
AND PRO.PrixUnit<=50
AND NOT PRO.PrixUnit>50 ;
```

```
13)
SELECT DISTINCT NomFou
FROM FOU,PRO
WHERE FOU.NumFou=PRO.NumFou
AND PRO.PrixUnit>50 AND NOT PRO.PrixUnit<=50;
```

```
14)
SELECT NomPro, NomFou, PrixUnit
FROM FOU,PRO
WHERE FOU.NumFou=PRO.NumFou
AND PRO.PrixUnit<=50 ;
```

Lister tous les produits de moins de 50 euros , et voir si toutes cette liste est chez un fournisseur. SI OUI l'afficher.

```
15)
SELECT DISTINCT NomCli
FROM CLI,COM
WHERE CLI.NumCli NOT IN ( SELECT NumCli FROM COM);
```

```
16)
SELECT numcli, nomcli
FROM CLI
WHERE CLI.nomcli in (select nomcli from cli
GROUP BY nomcli
HAVING (count(nomcli)=1));
```

```
17)

SELECT COUNT (NumPro)
FROM PRO;
```

```
18)
SELECT TypePro, AvG(PrixUnit), MAX(PrixUnit)
FROM PRO
GROUP BY typePro;
```

```
19)

SELECT TypePro, AvG(PrixUnit), MAX(PrixUnit)
FROM PRO
GROUP BY typePro
HAVING (COUNT(NumPro)>10) ;
```

## Partie PL/SQL

### Boucle for:

```
DECLARE
    x NUMBER:=100;
BEGIN
    FOR i IN 1..10 LOOP
        IF MOD(i,2)=0 THEN --i is even
            INSERT INTO temp VALUES (i,x,'i is even');
        ELSE
            INSERT INTO temp VALUES(i,x,'i id odd');
        END IF;
        x:=x+100;
    END LOOP;
    COMMIT;
END;
/
```

### Curseur

```
DECLARE
    CURSOR c1 is
        select ename,empno,sal from emp ORDER BY sal DESC;
    my_ename emp.ename%TYPE;
    my_empno emp.empno%TYPE;
    my_sal emp.sal%TYPE;
BEGIN
    OPEN c1;
    FOR i IN 1..5 LOOP
        FETCH c1 INTO my_ename,my_empno,my_sal;
        EXIT WHEN C1%NOTFOUND;
        INSERT INTO tempVALUES(my_ename,my_empno,my_sal);
        COMMIT;
    END LOOP;
    CLOSE c1;
END;
/
```

## Expérimentations sur l'optimisation

1)

On crée un nouveau fichier CreateDet.sql contenant la procédure:

```
Drop TABLE DET;
```

```
CREATE TABLE DET(
```

```
    NumCom number(5) Constraint DET_REF_COM REFERENCES Com ,  
    NumPro number(5) Constraint DET_REF_PRO REFERENCES Pro,  
    Qte number(5),  
    Remise number(5),  
    Constraint PK_DET PRIMARY KEY (NumCom, NumPro));
```

```
set timing on
```

```
DECLARE
```

```
BEGIN
```

```
.....
```

```
END ;
```

Puis on met toutes les instructions pour remplir la tables DET dans un premier temps avec celles du fichier Clean.sql (avec index) et après CreateNoInd(sans index).

Le set timing on permet de voir le temps écoulé après exécution.

Sans index on obtient 7 secondes.

Avec index on obtient 4 secondes.