

Phakeovilay Andrew

Qualité BD Compte rendu

TP#01 Création d'une BD normalisée en Oracle

Table des matières

1 Introduction.....	3
2 Création d'une BD normalisée en Oracle.....	3
2.1 Proposer un schéma relationnel 3NF de la BD en justifiant par la théorie de la normalisation (couverture minimale et algorithme de normalisation).....	3
2.2 Implanter cette BD dans Oracle sans que les attributs correspondent aux spécifications.....	5
2.3 Créez trois séquences pour gérer les valeurs des clés primaires des clients, des produits et des commandes.....	6
2.4 Insérez-vous comme client, puis insérez un produit et enfin effectuez une commande en votre nom de ce produit.....	7
3 Conclusion.....	7

PL/SQL Compte rendu

TP#01 Création d'une BD normalisée en Oracle

1 Introduction

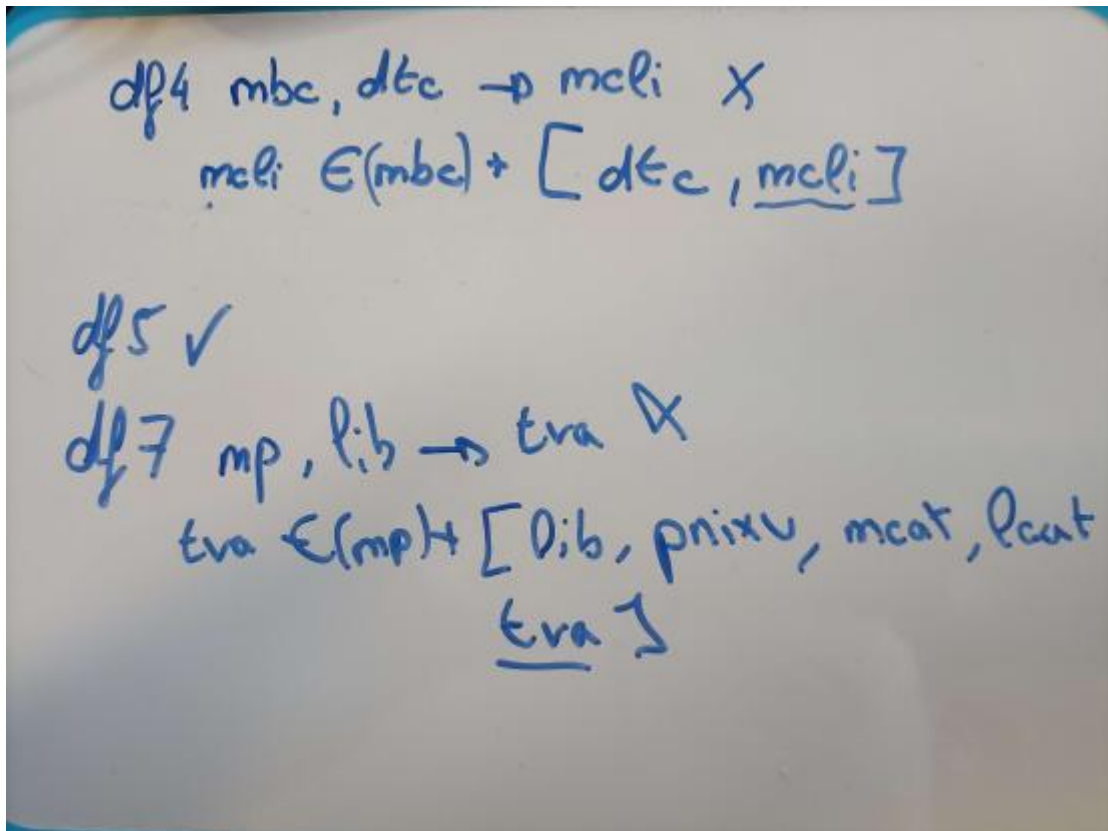
Ce TP permet de mettre en œuvre le principe de normalisation 3NF des bases de données relationnelles.

2 Création d'une BD normalisée en Oracle

2.1 Proposer un schéma relationnel 3NF de la BD en justifiant par la théorie de la normalisation (couverture minimale et algorithme de normalisation).

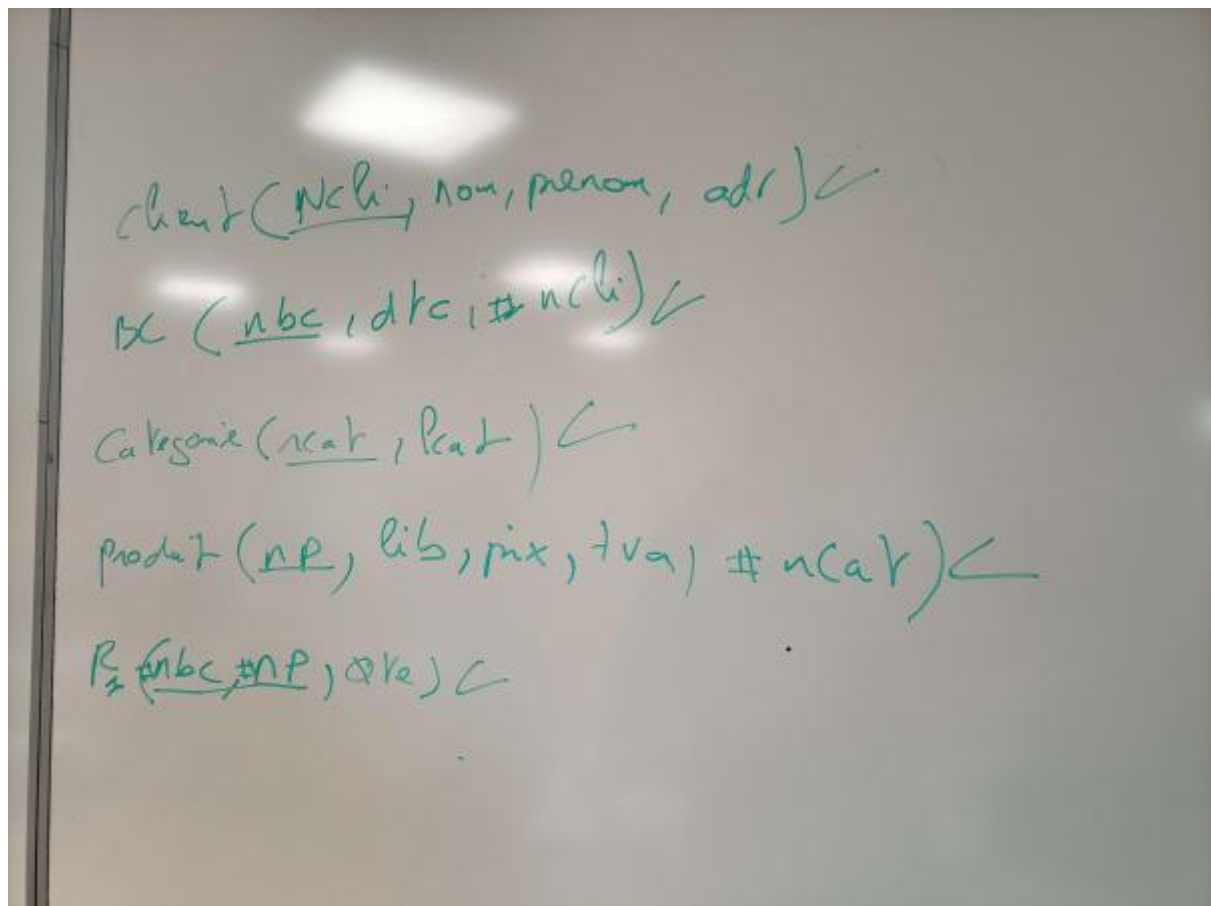
La théorie de la normalisation est d'avoir des dépendances fonctionnelles élémentaires et directes

Nous observons et modifions les dépendances fonctionnelles :



il y a une transition
df 6 $mp \rightarrow lib, prix_u, \underline{mcat}, \underline{pcat}, tra$
df 8 $mcat \rightarrow pcat$
donc
df 6' $mp \rightarrow lib, prix_u, mcat, tra$

Nous écrivons ensuite le schéma relationnel de la BD en fonction des dépendances fonctionnelles :



2.2 Planter cette BD dans Oracle sans que les attributs correspondent aux spécifications

```
CREATE TABLE Client(  
    ncli DECIMAL PRIMARY KEY,  
    prenom VARCHAR(30),  
    nom VARCHAR(30),  
    adr VARCHAR(100)  
);
```

```
CREATE TABLE Categorie(  
    ncat DECIMAL PRIMARY KEY,  
    lcat VARCHAR(50)  
);
```

```
CREATE TABLE BC(  
    nbc DECIMAL PRIMARY KEY,  
    dtc DATE,  
    ncli DECIMAL,  
    FOREIGN KEY (ncli) REFERENCES Client(ncli)  
);
```

```
CREATE TABLE Produit(  
    np DECIMAL PRIMARY KEY,  
    lib VARCHAR(50),  
    prixu DECIMAL(8,2) CHECK (prixu > 0),  
    ncat DECIMAL,  
    tva DECIMAL(5,2) CHECK ( tva IN (20,10,5.5,2.1)),  
    CONSTRAINT fk_ncat_categorie FOREIGN KEY (ncat) REFERENCES Categorie (ncat)  
);
```

```
CREATE TABLE R1(  
    nbc DECIMAL,  
    np DECIMAL,  
    qte DECIMAL(5) CHECK (qte>0),  
    CONSTRAINT pk_R1_nbc_np PRIMARY KEY (nbc,np),  
    CONSTRAINT fk_nbc_bc FOREIGN KEY (nbc) REFERENCES BC (nbc),  
    CONSTRAINT fk_np_produit FOREIGN KEY (np) REFERENCES Produit (np)  
);
```

2.3 Créez trois séquences pour gérer les valeurs des clés primaires des clients, des produits et des commandes.

The screenshot shows a 'Query Builder' window with a 'Feuille de calcul' (Worksheet) tab. The SQL script in the editor is as follows:

```
1 CREATE SEQUENCE seq_client_ncli
2 START WITH 1 INCREMENT BY 1;
3
4 CREATE SEQUENCE seq_produit_np
5 START WITH 1 INCREMENT BY 1;
6
7 CREATE SEQUENCE seq_bc_nbc
8 START WITH 1 INCREMENT BY 1;
```

Below the editor, a 'Sortie de script' (Script Output) window displays the results of the execution:

```
Sequence SEQ_CLIENT_NCLI créé(e) .

Sequence SEQ_PRODUIT_NP créé(e) .

Sequence SEQ_BC_NBC créé(e) .
```

2.4 Insérez-vous comme client, puis insérez un produit et enfin effectuez une commande en votre nom de ce produit

The screenshot shows a 'Query Builder' window with a 'Feuille de calcul' (Worksheet) tab. The SQL script in the editor is as follows:

```
1 INSERT INTO Client VALUES(seq_client_ncli.nextval, 'Andrew', 'Phakeovilay', '24 route de la taillade blanche');
2
3 INSERT INTO Produit VALUES(seq_produit_np.nextval, 'Spider-man', 10, NULL, 20);
4
5 INSERT INTO BC VALUES(seq_bc_nbc.nextval, SYSDATE, seq_client_ncli.currval);
6
7 INSERT INTO R1 VALUES(seq_bc_nbc.currval, seq_produit_np.currval, 1);
```

3 Conclusion

Nous avons vu la mise en place de 3NF et rappel de la création de table, de séquence et insertion par rapports aux dépendances fonctionnelles et le schéma relationnel.