

**Phakeovilay Andrew**

Qualité BD Compte rendu

**TP#04 Modélisation Non First Normal Form**

---

## Table des matières

1 Introduction.....	3
2 Création de relations par "désimbrication" .....	3
2.1 Donnez la relation obtenue suite aux opérations de désimbrication suivantes.....	3
2.1.1 UNNEST(MATCH ; joueur1) = MATCH_1.....	3
2.1.2 UNNEST(MATCH_1 ; joueur2) = MATCH_2.....	3
2.1.3 UNNEST(MATCH_2 ; equipe1) = MATCH_3.....	3
2.1.4 UNNEST(MATCH_3 ; equipe2) = MATCH_4.....	3
2.2 Donnez les commandes SQL de création de la table MATCH_4 obtenue.....	3
2.3 Insérez dans la table MATCH_4 les enregistrements issus du match nm=30 présenté dans la table initiale.....	4
2.4 Proposez une forme 3NF de la relation MATCH_4 en considérant les dépendances fonctionnelles suivantes.....	4
– df1 : nm → stade,ne1,ne2,scr1,scr2.....	4
– df2 : ne → nome.....	4
– df3 : prej, nomj → ne.....	4
– df4 : nm, prej, nomj → maillot, ne.....	4
2.5 Donnez les commandes SQL de création des nouvelles tables obtenues.....	5
2.6 Insérez dans les tables obtenues les enregistrements issus du match nm=30 présenté dans la table initiale.....	6
3 Conclusion.....	6

# Qualité BD Compte rendu

## TP#04 Modélisation Non First Normal Form

---

### 1 Introduction

Ce TP permet de transformer les relations NF2 en relation normalisée avec le SGBD Oracle.

### 2 Création de relations par "désimbrication"

#### 2.1 Donnez la relation obtenue suite aux opérations de désimbrication suivantes

```
Match(nm, stade, e1:[ne, nome, scr, joueur:[prej, nomj, maillot]], e2:[ne, nome, scr, joueur:[prej, nomj, maillot]])
```

##### 2.1.1 UNNEST(MATCH ; joueur1) = MATCH\_1

```
Match_1(nm, stade, e1:[ne, nome, scr, prej, nomj, maillot], e2:[ne, nome, scr, joueur:[prej, nomj, maillot]])
```

##### 2.1.2 UNNEST(MATCH\_1 ; joueur2) = MATCH\_2

```
Match_2(nm, stade, e1:[ne, nome, scr, prej, nomj, maillot], e2:[ne, nome, scr, prej, nomj, maillot])
```

##### 2.1.3 UNNEST(MATCH\_2 ; equipe1) = MATCH\_3

```
Match_3(nm, stade, ne1, nome1, scr1, prej1, nomj1, maillot1, e2:[ne, nome, scr, prej, nomj, maillot])
```

##### 2.1.4 UNNEST(MATCH\_3 ; equipe2) = MATCH\_4

```
Match_4(nm, stade, ne1, nome1, scr1, prej1, nomj1, maillot1, ne2, nome2, scr2, prej2, nomj2, maillot2)
```

#### 2.2 Donnez les commandes SQL de création de la table MATCH\_4 obtenue.

```
CREATE TABLE Match_4(
    nm NUMBER,
    stade VARCHAR(50),
    ne1 CHAR(3),
    nome1 VARCHAR(20),
    scr1 NUMBER,
    prej1 VARCHAR(20),
    nomj VARCHAR(20),
    maillot NUMBER,
    ne2 CHAR(3),
    nome2 VARCHAR(20),
    scr2 NUMBER,
    prej2 VARCHAR(20),
    nomj2 VARCHAR(20),
    maillot2 NUMBER,
    CONSTRAINT PK_Match_4 PRIMARY KEY (nm, ne1, prej1, nomj, ne2, prej2, nomj2)
);
```

2.3 Insérez dans la table MATCH\_4 les enregistrements issus du match nm=30 présenté dans la table initiale.

```
INSERT INTO MATCH_4 VALUES(
    30, 'Stade de France', 'FRA', 'France', 41, 'Thomas', 'Ramos', 15, 'WAL', 'Pays de Galles', 28, 'Dan', 'Biggar', 10
);
INSERT INTO MATCH_4 VALUES(
    30, 'Stade de France', 'FRA', 'France', 41, 'Thomas', 'Ramos', 15, 'WAL', 'Pays de Galles', 28, 'Georges', 'North', 13
);
INSERT INTO MATCH_4 VALUES(
    30, 'Stade de France', 'FRA', 'France', 41, 'Romain', 'Ntamack', 10, 'WAL', 'Pays de Galles', 28, 'Dan', 'Biggar', 10
);
INSERT INTO MATCH_4 VALUES(
    30, 'Stade de France', 'FRA', 'France', 41, 'Romain', 'Ntamack', 10, 'WAL', 'Pays de Galles', 28, 'Georges', 'North', 10
);
INSERT INTO MATCH_4 VALUES(
    30, 'Stade de France', 'FRA', 'France', 41, 'Antoine', 'Dupont', 9, 'WAL', 'Pays de Galles', 28, 'Dan', 'Biggar', 10
);
INSERT INTO MATCH_4 VALUES(
    30, 'Stade de France', 'FRA', 'France', 41, 'Antoine', 'Dupont', 9, 'WAL', 'Pays de Galles', 28, 'Georges', 'North', 10
);
```

2.4 Proposez une forme 3NF de la relation MATCH\_4 en considérant les dépendances fonctionnelles suivantes

- df1 : nm → stade, ne1, ne2, scr1, scr2
- df2 : ne → nome
- df3 : prej, nomj → ne
- df4 : nm, prej, nomj → maillot, ne

On retire ne de la df4 car ne est défini par une partie (df3, prej, nomj → ne), on enlève cela afin de permettre une forme 2NF, on se retrouve avec comme relations

R1(nm, stade, ne1, ne2, scr1, scr2)

R2(ne, nome, prej, nomj)

R3(prej, nomj, ne)

R4(nm, prej, nomj, maillot)

2.5 Donnez les commandes SQL de création des nouvelles tables obtenues.

```
CREATE TABLE Equipe(  
    ne CHAR(3) PRIMARY KEY,  
    nome VARCHAR(20)  
);  
  
CREATE TABLE Joueur(  
    prej VARCHAR(20),  
    nomj VARCHAR(20),  
    ne CHAR(3),  
    CONSTRAINT pk_joueur PRIMARY KEY (prej, nomj),  
    CONSTRAINT fk_ne_joueur FOREIGN KEY(ne) REFERENCES Equipe(ne)  
);  
  
CREATE TABLE Match(  
    nm NUMBER,  
    stade VARCHAR(50),  
    ne1 CHAR(3),  
    ne2 CHAR(3),  
    scr1 NUMBER,  
    scr2 NUMBER,  
    CONSTRAINT pk_match PRIMARY KEY(nm),  
    CONSTRAINT fk_match_equipe1 FOREIGN KEY(ne1) REFERENCES Equipe(ne),  
    CONSTRAINT fk_match_equipe2 FOREIGN KEY(ne2) REFERENCES Equipe(ne)  
);
```

```
CREATE TABLE Composition(  
    nm NUMBER,  
    prej VARCHAR(20),  
    nomj VARCHAR(20),  
    maillot NUMBER,  
    CONSTRAINT pk_nm_prej_nomj_composition PRIMARY KEY (nm, prej, nomj),  
    CONSTRAINT fk_nm_composition FOREIGN KEY(nm) REFERENCES Match(nm),  
    CONSTRAINT fk_prej_composition FOREIGN KEY(prej, nomj) REFERENCES Joueur(prej, nomj)  
);
```

- 2.6 Insérez dans les tables obtenues les enregistrements issus du match nm=30 présenté dans la table initiale.

```
INSERT INTO Equipe VALUES('FRA', 'France');
INSERT INTO Equipe VALUES('WAL', 'Pays de Galles');

INSERT INTO Joueur VALUES('Thomas', 'Ramos', 'FRA');
INSERT INTO Joueur VALUES('Dan', 'Biggar', 'WAL');
INSERT INTO Joueur VALUES('Georges', 'North', 'WAL');
INSERT INTO Joueur VALUES('Romain', 'Ntamack', 'FRA');
INSERT INTO Joueur VALUES('Antoine', 'Dupont', 'FRA');

INSERT INTO Match VALUES(30, 'Stade de France', 'FRA', 'WAL', 41, 28);

INSERT INTO Composition VALUES(30, 'Thomas', 'Ramos', 15);
INSERT INTO Composition VALUES(30, 'Romain', 'Ntamack', 10);
INSERT INTO Composition VALUES(30, 'Antoine', 'Dupont', 9);
INSERT INTO Composition VALUES(30, 'Dan', 'Biggar', 10);
INSERT INTO Composition VALUES(30, 'Georges', 'North', 13);
```

### 3 Conclusion

Nous avons vu comment transformer la fonction UNNEST en SQL, effectuer une forme 3NF en SQL.