**Université Dauphine**

**Mini-Projet Bases de données relationnelles**

**Licence MIDO troisième année par alternance**

**Francisco Javier Martínez Lago**

1. **Passage de la modélisation au relationnel**

La base de données a été créée en utilisant le logiciel d'administration de bases de données pgAdmin4 sur un ordinateur Macintosh. 3 requêtes d'interrogation et un trigger sont demandés car il s'agit d'un travail individuel.

Le sport choisi est Football. On a créé 8 relations au total, 5 à partir des classes du diagramme, et 3 à partir des relations de type \* - \*.

* 1. *Match* ***(****matchID, date*match, heure\_debut, heure\_fin)

J’ai créé un attribut additionnel appelé MatchID qui sera la clé primaire et une contrainte pour que l'heure fin ne puisse être introduite que si l'heure de début et la date existent.

* 1. Equipe (nomequipe, numJoueurs)

Le nom de l’équipe sera la clé primaire.

* 1. Joueur (numero, #nomequipe, nom, prenom, datenaissance)

Un jouer est identifié par sa clé primaire composée de son numéro et du nom de son équipe – le nom d’équipe étant une clé étrangère référençant la clé primaire de la relation équipe.

* 1. Gardien (#numero, #nomequipe)

Le gardien est aussi un jouer, dont on peut simplement garder une référence à la clé primaire de joueur dans gardien. Un nuplet dans gardien ne pourra exister que s'il existe un nuplet correspondant dans la relation Joueur.

* 1. But (butID, heure, #matchID, #numero, #nomequipe)

Un but est associé à un seul match et à un joueur qui marque et 0 ou 1 joueur qui assiste. On crée une clé primaire – butID – pour cette relation. On met des clés étrangères dans But référençant les clés primaires des relations \_Match et Joueur. Pour la relation "assiste" \* - \* avec Jouer, on crée une nouvelle relation "assiste".

* 1. Assiste (#butID, #numero\_joueur, #nomequipe)

Par les règles de relations \* - \*, on met dans "assiste" les clés primaires de Joueur et But en tant que clés étrangères référençant les clés primaires ces tables et aussi en tant que clés primaires de "assiste".

* 1. Est\_Dans\_Le\_Buts (heure\_debut, heure\_fin, nb\_buts\_encaisses, #numero\_joueur, #nomequipe, #matchID)

Relation créé à partir de la relation \* - \* entre Match et Gardien. La clé primaire est composé de clé étrangères référençant les clés primaires de relations associées. Un trigger a été créé pour empêcher la création d'un nuplet dans cette relation si comme résultant on obtient plus de 3 gardiens associés un match et à un equipe.

* 1. Joue (#matchID, score, classement, #nomequipe)

Relation créé à partir de la relation \* - \* entre \_Match et Equipe. Clé primaire composée des clés étrangères référençant les clés primaires de \_Match et d'Équipe. Un trigger a été créé pour ne pas permettre que plus de deux équipes soient associés à un match.

1. **Script SQL de création du schéma de la base de données**

Le script est aussi disponible sur http://pastebin.com/4JCmD4Wz

CREATE TABLE \_Match

(

matchID SERIAL,

date\_match DATE,

heure\_debut TIME,

heure\_fin TIME CONSTRAINT chk\_heure\_fin CHECK (heure\_debut IS NOT NULL AND date\_match IS NOT NULL),

CONSTRAINT pk\_matchid PRIMARY KEY (matchID)

);

CREATE TABLE Equipe

(

nomequipe VARCHAR,

numJoueurs SMALLINT,

CONSTRAINT pk\_nom PRIMARY KEY (nomequipe)

);

CREATE TABLE Joueur

(

numero SMALLINT,

nomequipe VARCHAR,

nom VARCHAR,

prenom VARCHAR,

datenaissance DATE,

CONSTRAINT pk\_joueur PRIMARY KEY (numero,nomequipe),

CONSTRAINT fk\_nomequipe FOREIGN KEY (nomequipe) REFERENCES Equipe (nomequipe)

);

CREATE TABLE Gardien

(

numero SMALLINT,

nomequipe VARCHAR,

CONSTRAINT pk\_numero PRIMARY KEY (numero,nomequipe),

CONSTRAINT fk\_Gardien FOREIGN KEY (numero,nomequipe) REFERENCES Joueur (numero,nomequipe)

);

CREATE TABLE But

(

butID SERIAL,

heure TIME,

matchID INTEGER,

numero SMALLINT,

nomequipe VARCHAR,

CONSTRAINT pk\_but PRIMARY KEY (butID),

CONSTRAINT fk\_but\_matchID FOREIGN KEY (matchID) REFERENCES \_Match(matchID),

CONSTRAINT fk\_joueur FOREIGN KEY (numero, nomequipe) REFERENCES Joueur (numero, nomequipe)

);

CREATE TABLE assiste

(

butID INTEGER,

numero\_joueur SMALLINT,

nomequipe VARCHAR,

CONSTRAINT pk\_assiste PRIMARY KEY (butID,numero\_joueur,nomequipe),

CONSTRAINT fk\_but FOREIGN KEY (butID) REFERENCES But (butID),

CONSTRAINT fk\_joueur FOREIGN KEY (numero\_joueur,nomequipe) REFERENCES Joueur (numero, nomequipe)

);

CREATE TABLE Est\_Dans\_Le\_Buts

(

heure\_debut TIME,

heure\_fin TIME,

nb\_buts\_encaisses SMALLINT,

numero\_joueur SMALLINT,

nomequipe VARCHAR,

matchID INTEGER,

CONSTRAINT pk\_Est\_Dans\_Le\_Buts PRIMARY KEY (matchID,numero\_joueur,nomequipe),

CONSTRAINT fk\_est\_dans\_le\_buts\_matchID FOREIGN KEY (matchID) REFERENCES \_Match (matchID),

CONSTRAINT fk\_joueur FOREIGN KEY (numero\_joueur,nomequipe) REFERENCES Gardien (numero,nomequipe)

);

CREATE TABLE Joue

(

matchID INTEGER,

score SMALLINT,

classement SMALLINT,

nomequipe VARCHAR,

CONSTRAINT pk\_joue PRIMARY KEY (matchID, nomequipe),

CONSTRAINT fk\_joue\_matchID FOREIGN KEY (matchID) REFERENCES Match (matchID),

CONSTRAINT fk\_equipe FOREIGN KEY (nomequipe) REFERENCES Equipe(nomequipe)

);

1. **Script pour insérer de données**

Disponible aussi sur <http://pastebin.com/zRNbtTdj>

SET datestyle = "ISO, DMY";

-- Equipe

INSERT INTO Equipe VALUES ('Squirrels', 10);

INSERT INTO Equipe VALUES ('Eagles', 10);

INSERT INTO Equipe VALUES ('Gladiators', 10);

INSERT INTO Equipe VALUES ('Procrastinators', 10);

INSERT INTO Equipe VALUES ('Monkeys', 10);

-- \_Match

INSERT INTO \_Match (matchID,date\_match,heure\_debut,heure\_fin) VALUES (1,'02-02-2001', '21:00:00', '22:30:00');

INSERT INTO \_Match (matchID,date\_match,heure\_debut,heure\_fin) VALUES (2,'06-03-2002', '20:00:00', '21:30:00');

INSERT INTO \_Match (matchID,date\_match,heure\_debut,heure\_fin) VALUES (3,'05-04-2003', '21:00:00', '22:30:00');

INSERT INTO \_Match (matchID,date\_match,heure\_debut,heure\_fin) VALUES (4,'13-05-2004', '17:00:00', '18:30:00');

INSERT INTO \_Match (matchID,date\_match,heure\_debut,heure\_fin) VALUES (5,'12-06-2005', '21:00:00', '22:30:00');

-- Joueur

INSERT INTO Joueur VALUES (1, 'Eagles', 'Manouvrier', 'Maude', '23-08-1987');

INSERT INTO Joueur VALUES (1, 'Gladiators', 'Maude ', 'Martinez', '20-08-1984');

INSERT INTO Joueur VALUES (3, 'Eagles', 'Ildra', 'Jimenez', '12-08-1985');

INSERT INTO Joueur VALUES (2, 'Squirrels', 'Paco', 'Garcia', '02-08-1983');

INSERT INTO Joueur VALUES (3, 'Monkeys', 'Jean', 'Le Fleur', '12-08-1980');

INSERT INTO Joueur VALUES (2, 'Procrastinators', 'Lynda', 'Martinez', '13-08-1980');

INSERT INTO Joueur VALUES (5, 'Gladiators', 'Clarence', 'Fernandez', '17-08-1982');

INSERT INTO Joueur VALUES (6, 'Gladiators', 'Emilie', 'Colbert', '24-08-1983');

INSERT INTO Joueur VALUES (1, 'Procrastinators', 'Donald', 'Trump', '22-08-1996');

INSERT INTO Joueur VALUES (9, 'Eagles', 'Barack', 'Obama', '22-08-1996');

INSERT INTO Joueur VALUES (1, 'Squirrels', 'May', 'Zoera', '18-08-1983');

INSERT INTO Joueur VALUES (1, 'Monkeys', 'Hyrule', 'Link', '19-02-1988');

-- Gardien

INSERT INTO Gardien VALUES (1, 'Eagles');

INSERT INTO Gardien VALUES (1, 'Procrastinators');

INSERT INTO Gardien VALUES (1, 'Gladiators');

INSERT INTO Gardien VALUES (1, 'Squirrels');

INSERT INTO Gardien VALUES (1, 'Monkeys');

-- But

INSERT INTO But (butID,heure,matchID,numero,nomequipe) VALUES (1,'21:00:01', 1, 1, 'Eagles');

INSERT INTO But (butID,heure,matchID,numero,nomequipe) VALUES (2,'21:15:01', 2, 3, 'Eagles');

INSERT INTO But (butID,heure,matchID,numero,nomequipe) VALUES (3,'21:15:01', 3, 1, 'Procrastinators');

INSERT INTO But (butID,heure,matchID,numero,nomequipe) VALUES (4,'17:23:01', 4, 6, 'Gladiators');

INSERT INTO But (butID,heure,matchID,numero,nomequipe) VALUES (5,'21:15:01', 5, 3, 'Monkeys');

-- Joue

INSERT INTO Joue (score, classement, matchID, nomequipe) VALUES (4,1,1,'Eagles');

INSERT INTO Joue (score, classement, matchID, nomequipe) VALUES (3,2,1,'Squirrels');

-- Est\_dans\_Le\_Buts

INSERT INTO Est\_Dans\_Le\_Buts VALUES ('21:00:00', '22:30:00', 3, 1, 'Eagles', 1);

INSERT INTO Est\_Dans\_Le\_Buts VALUES ('22:00:00', '23:00:00', 4, 1, 'Gladiators', 1);

INSERT INTO Est\_Dans\_Le\_Buts VALUES ('22:00:00', '23:00:00', 4, 1, 'Squirrels', 1);

INSERT INTO Est\_Dans\_Le\_Buts VALUES ('22:00:00', '23:00:00', 4, 1, 'Procrastinators', 1);

-- Assiste

INSERT INTO assiste VALUES (1, 3,'Eagles');

1. **Requêtes prédéfinies**

Requête 1

**En SQL**

SELECT e.nomequipe FROM equipe AS e

INNER JOIN Joue ON (e.nomequipe = joue.nomequipe)

INNER JOIN \_Match on (\_match.matchID = joue.matchID)

WHERE \_match.date\_match = '2001-02-02'

**En calcul relationnel**

π nomequipe(equipe ⨝  joue ⨝   σdate\_match = '2001-02-02'(\_Match))

**En algèbre relationnelle**

**{t.nomequipe | equipe(t) ∧ ∃j,Joue(j) ∧ ∃m,\_Match(m) ∧ t.nomequipe = j.nomequipe ∧ j.matchID = m.matchID ∧ m.date\_match = '2001-02-02'}**

Requête 2

**En SQL**

SELECT nom, prenom FROM Joueur

EXCEPT

SELECT nom, prenom FROM Joueur

INNER JOIN Gardien ON (joueur.numero = gardien.numero AND joueur.nomequipe = gardien.nomequipe)

**En calcul relationnel**

π nom, prenom  (Joueur) - π nom, prenom (Gardien)

**En algèbre relationnelle**

{t.nom, t.prenom | Joeur(t) ∧ ∃g, Gardien(g) ∧ t.nomequipe != g.nomequipe ∧ t.numero != g.numero}

Requête 3

**En SQL**

SELECT e.nomequipe, SUM(score) FROM equipe AS e, joue AS j

WHERE e.nomequipe = j.nomequipe

GROUP BY e.nomequipe

Seulement en SQL parce que il y a une agrégation.

1. **Déclencheurs**

Un trigger a été demandé car c'est un travail individuel. Néanmoins, j'ai créé trois. J'ai expliqué les trois triggers et donné des exemples d'utilisation pour le troisième – celui étant le déclencheur demandé par le travail. Le code est aussi disponible sur http://pastebin.com/tbpNhsTZ

Premier Trigger

Le trigger *un\_match\_deux\_equipes\_coherence* déclenche dès qu'il y a un essai d'insertion dans la relation Joue. Il lance la fonction *un\_match\_deux\_equipes\_coherence()*. Cette fonction met dans une variable le nombre de matchs joués – y inclus le nuplet qu'on va ajouter - où le matchID est égal au matchID du nouvel nuplet à ajouter. Si le résultat est plus grand que 2, alors ça veut dire que on va associer plus de 2 équipes à un seul match. Le trigger va alors empêcher cette insertion avec un message d'erreur.

CREATE OR REPLACE FUNCTION un\_match\_deux\_equipes\_coherence()

RETURNS TRIGGER AS

$BODY$

DECLARE

counter INT;

BEGIN

SELECT COUNT(\*) INTO counter

FROM Joue

WHERE matchID = NEW.matchID;

IF counter > 2 THEN

RAISE EXCEPTION 'No more than 2 teams per match allowed';

ABORT TRANSACTION;

END IF;

RETURN NEW;

END;

$BODY$ LANGUAGE 'plpgsql';

CREATE TRIGGER un\_match\_deux\_equipes\_coherence

AFTER INSERT ON Joue

FOR EACH ROW

EXECUTE PROCEDURE un\_match\_deux\_equipes\_coherence();

Deuxième Trigger

Le trigger *un\_match\_un\_equipe\_trois\_gardiens\_coherence* déclenche dès qu'il y a un essai d'insertion dans la relation Est\_Dans\_Le\_Buts. Il lance la fonction *un\_match\_un\_equipe\_trois\_gardiens\_coherence* *()*. Cette fonction met dans une variable le nombre de nuplets dans la relation– y inclus le nuplet qu'on va ajouter - où le matchID est égal au matchID du nouvel nuplet à ajouter et le nom d'équipe est égal au nom d'équipe à ajouter. Si le résultat est plus grand que 3, alors ça veut dire que on va associer plus de 3 gardiens d'un équipe à un seul match. Le trigger va alors empêcher cette insertion avec un message d'erreur.

CREATE OR REPLACE FUNCTION un\_match\_un\_equipe\_trois\_gardiens\_coherence()

RETURNS trigger AS

$BODY$

DECLARE

counter int;

BEGIN

SELECT COUNT(\*) INTO counter

FROM Est\_Dans\_Le\_Buts

WHERE matchID = NEW.matchID

AND nomequipe = NEW.nomequipe;

IF counter > 3 THEN

RAISE EXCEPTION 'No more than 3 gardiens of the same team per match allowed';

ABORT TRANSACTION;

END IF;

RETURN NEW;

END;

$BODY$ LANGUAGE 'plpgsql';

CREATE TRIGGER un\_match\_un\_equipe\_trois\_gardiens\_coherence

AFTER INSERT ON Est\_Dans\_Le\_Buts

FOR EACH ROW

EXECUTE PROCEDURE un\_match\_un\_equipe\_trois\_gardiens\_coherence();

Troisième Trigger

Le trigger *calcul\_score\_equipe* déclenche dès qu'il y a un essai d'insertion dans la relation But. Il lance la fonction *calcul\_score\_equipe()*. Cette fonction va mettre à jour la relation Joue en ajoutant 1 à la colonne Score de l'équipe qui a marqué dont le match et l'équipe correspondent respectivement aux attributs matchID et nomequipe du nouveau nuplet dans But.

CREATE OR REPLACE FUNCTION calcul\_score\_equipe()

RETURNS TRIGGER AS

$BODY$

BEGIN

UPDATE Joue SET score = score + 1

WHERE matchID = NEW.matchID

AND joue.nomequipe = NEW.nomequipe;

RETURN NEW;

END;

$BODY$ LANGUAGE 'plpgsql';

CREATE TRIGGER calcul\_score\_equipe

AFTER INSERT ON But

FOR EACH ROW

EXECUTE PROCEDURE calcul\_score\_equipe();

**Exemples**

SELECT \* FROM Joue

INSERT INTO But (ButID, heure,matchID,numero,nomequipe) VALUES (100, '21:00:01', 1, 1, 'Eagles');

-- Le nuplet avec Eagles dans la relation Joue devrait avoir un but de plus.

SELECT \* FROM JOUE

INSERT INTO But (ButID, heure,matchID,numero,nomequipe) VALUES (101, '21:00:01', 1, 2, 'Squirrels');

-- Le nuplet avec Squirrels dans la relation Joue devrait avoir un but de plus.

SELECT \* FROM JOUE

1. **Conclusion**

Étant un travail individuel, j'ai fait tous les tâches en utilisant comme référence les feuilles du cours ainsi que plusieurs sites web. Je n'ai calculé le temps que j'ai passé dans le projet, mais c'était moins de 10 heures.

Pour assurer la cohérence de la base de données, il faudrait encore ajouter plusieurs contraintes. Si par exemple un nuplet de \_Match est éliminé de la Base de Données (BD), alors il faudrait une contrainte sur la clé étrangère référençant la colonne matchID de Match de la relation Joue du type *ON DELETE {CASCADE | SET NULL … }* et similairement pour d'autres relations.

Il faudrait aussi ajouter contrainte pour éviter que l'heure de fin d'un Match soit inférieure à l'heure de début de ce match dans la relation Match, ce qu'on pourrait faire avec un CHECK. Même chose pour la relation Gardien.

On pourrait aussi améliorer notre trigger *calcul\_score\_equipe* pour éviter une situation ou la somme de la colonne Score de la relation Joue pour un \_Match soit différente que la somme de buts pour ce \_Match dans la relation But.

Le travail a été effectué sur un Mac avec le logiciel pgAdmin 4 et donc c'est possible que il y ait des différences fonctionnelles entre ceci et le logiciel utilisé en cours.

Bien que la BD à un minimum de fonctionnalité, on pourrait l'améliorer avec des contraintes additionnelles pour assurer la cohérence de données et l'intégrité de données.