

Licence ISC parcours MIAGE

Rapport de projet :

Coupe du monde de rugby

Médéric Hurier

mederic.hurier@etudiant.univ-nancy2.fr

Omar Eddasser

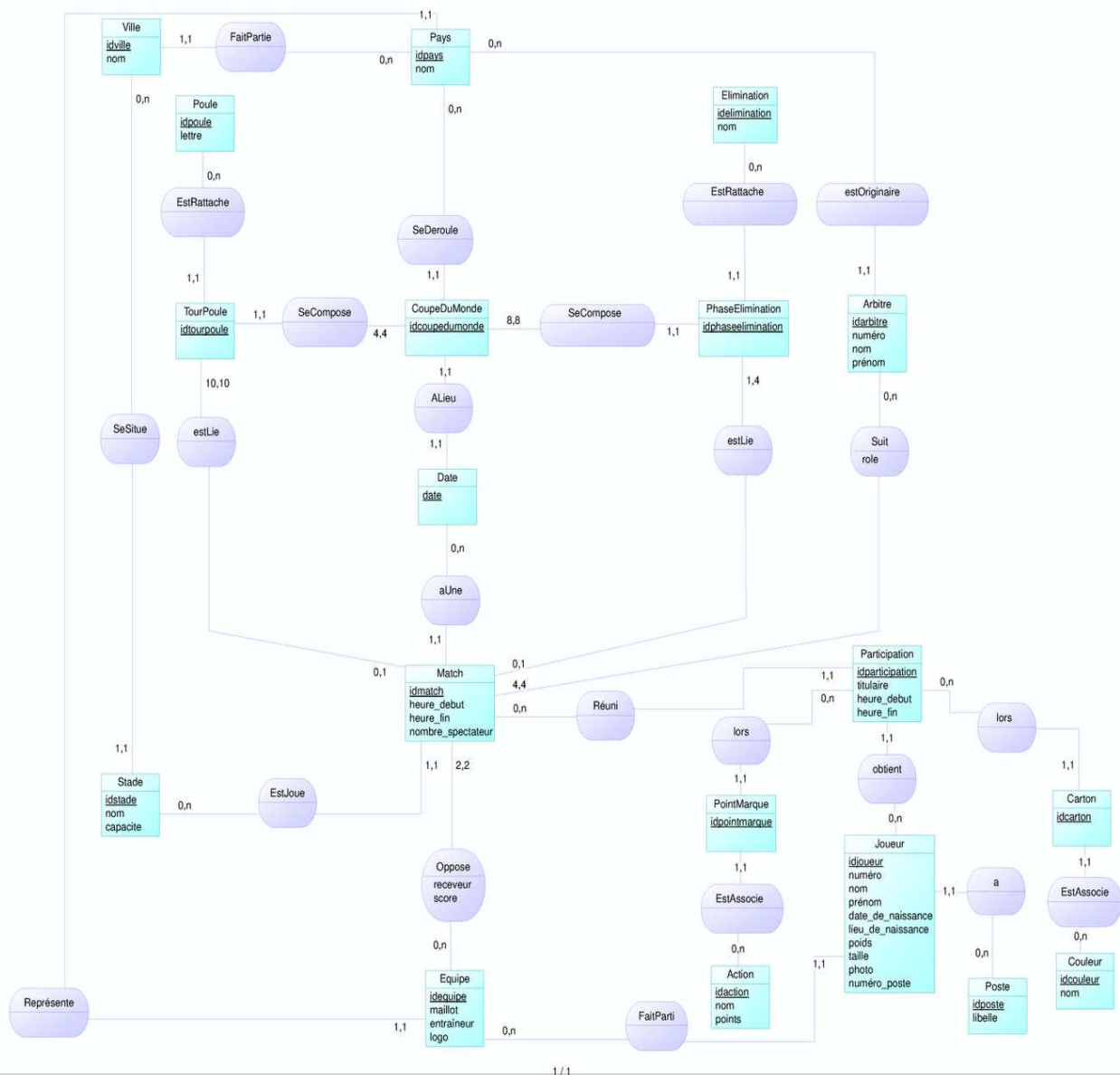
omar.eddasser@etudiant.univ-nancy2.fr



Table des matières

I.	Schéma Entité – Association	2
II.	Dictionnaire de données	4
1.	Description des entités.....	4
2.	Description des attributs.....	4
III.	Règles de gestions	5
1.	Gestion des phases.....	5
2.	Gestion des matchs	6
3.	Gestion des joueurs.....	6
IV.	Modèle physique de données	6
V.	Relations en 3e forme normale	8
VI.	Script de génération SQL	9
VII.	Difficultés	17

I. Schéma Entité – Association



Note : Ce schéma est disponible dans une version plus grande cf. « **modele_ea.png** »

Ce schéma Entité/Association peut être **parcouru de deux façons** distinctes selon le type de requête que l'on souhaite effectuer. Le but est de **limiter la redondance** au maximum, **sans pour autant complexifier les requêtes communes**.

La première navigation part de l'entité coupe du monde jusqu'aux joueurs, à la manière d'un **arbre**. Une coupe du monde est composée de deux phases : **la phase préparatoire** composée de poules, et **la phase éliminatoire** composée d'épreuves (demi-finale, finale ...). Chacune de ces phases regroupe plusieurs matches par poule ou par épreuve. Les matches opposent 2 équipes, qui sont composés de plusieurs joueurs ayant

participé aux matchs. Enfin, chaque participation est associée aux points et aux cartons du joueur.

Cette navigation permet de récupérer successivement :

- Les informations concernant le tournoi : poules, épreuves éliminatoires
- Les matchs liés à ces phases
- Les équipes et les joueurs qui ont participé à ces matchs
- Les points marqués et les cartons pris par les joueurs

Ainsi, on facilite la recherche vers les éléments détaillés.

La deuxième navigation part des entités de plus bas niveau, tel que les joueurs ou les équipes, pour retrouver les informations associées:

- Poste occupé par le joueur
- Matchs joués
- Score d'une équipe pour un match, grâce à l'association "Oppose"
- Tous les points marqués et les cartons pris sur un match ou depuis le début de la coupe

Cette double lecture est **centrée autour de l'entité Match**. On peut facilement partir d'un match pour retrouver toutes les informations citées ci-dessus.

Au niveau de l'écriture, **les redondances sont minimales**. Les informations dupliquées sont:

- Le score d'une équipe, qui peut être déduit des nombres de points marqués par les joueurs ayant participé à un match
- Les associations Match-Équipe et Match-Joueur, qui permettent de consulter rapidement les équipes qui se sont affrontées, et le détail des actions pour chaque joueur.

De plus, les libellés et les informations communes ont été extraits dans des entités séparées. Par exemple, les lettres de poule qui sont communes à toutes les coupes du monde se trouvent dans l'entité Poule.

Il en résulte un schéma composé de nombreuses entités, mais **simple à implémenter**. Il n'existe pas d'associations ternaires, ou de duplicata non justifiés.

II. Dictionnaire de données

1. Description des entités

Entité / Association	Description
Coupe du monde	Une coupe du monde qui a lieu à une année et dans un pays
Poule	Intitulé de phase préparatoire (A, B, C, D)
Tour de poule	Regroupe 5 équipes dans une poule de coupe du monde
Élimination	Intitulé de phase éliminatoire (demi-finale, finale ...)
Phase d'élimination	Tournoi d'élimination entre les équipes retenues
Match	Confrontation entre deux équipes
Stade	Lieu public où se déroulent les matchs
Arbitre	Personne qui surveille le déroulement d'un match
Suit	Attribution des arbitres par match
Équipe	Ensemble de joueur pour un pays
Oppose	Équipes rencontrées lors d'un match
Joueur	Professionnel de rugby
Poste	Place tenu dans l'équipe (arrière, ailier ...)
Participation	Enregistrement d'un joueur pour un match
Action	Mouvement qui rapporte des points (essai, transformation)
Point marqué	Action d'un joueur lors d'un match
Carton	Faute d'un joueur lors d'un match

2. Description des attributs

C : Caractère, S : Chaîne de caractère, E : Entier, B : Booléen, D : Date, F : Fichier

Entité	Attribut	Type	Description
Poule	Lettre	C	Lettre pour identifier la poule (A, B, C ...)
Élimination	Nom	S	Nom de l'épreuve (demi-finale, finale ...)
Match	Id	E	Identifiant du match
	Heure début	D	Heure de début du match
	Heure fin	D	Heure de fin de match
	Nombre spectateur	E	Nombre de spectateurs ayant assisté au match
Stade	Id	E	Identifiant de stade
	Nom	S	Nom du stade (Stade de France)
	Capacité	E	Capacité totale en nombre de spectateur

Arbitre	Numéro	E	Identifiant de l'arbitre
	Nom	S	Nom de l'arbitre
	Prénom	S	Prénom de l'arbitre
Suit	Rôle	S	Rôle tenu par l'arbitre lors de la rencontre
Équipe	Code	E	Identifiant de l'équipe
	Maillot	S	Couleur du maillot
	Entraîneur	S	Nom et prénom de l'entraîneur
	Logo	F	Logo de l'équipe
Oppose	Receveur	B	Vrai si l'équipe reçoit
	Score	E	Nombre de point marqué par une équipe
Joueur	Numéro	E	Identifiant du joueur
	Nom	S	Nom du joueur
	Prénom	S	Prénom du joueur
	Date de naissance	D	Date de naissance du joueur
	Lieu de naissance	S	Lieu de naissance du joueur
	Poids	E	Poids en kilogrammes du joueur
	Taille	E	Taille en centimètres du joueur
	Photo	F	Photo d'identité du joueur
	Numéro de poste	E	Numéro du maillot
Poste	Libellé	S	Poste tenu par un joueur (arrière, ailier)
Participation	Titulaire	B	Vrai si le joueur été présent au début du match
	Heure début	D	Heure d'entrée du joueur sur le terrain
	Heure fin	D	Heure de départ du joueur sur le terrain
Action	Nom	S	Nom de l'action (essai, transformation)
	Points	E	Nombre de points que rapporte l'action
Couleur	Nom	S	Code couleur en hexadécimale
Pays	Nom	S	Nom du pays
Ville	Nom	S	Nom de la ville

III. Règles de gestions

1. Gestion des phases

- Une coupe du monde peut être identifiée par un pays et une année
- Une coupe du monde se déroule en deux étapes:
 - **la phase préparatoire** avec les matchs de poules. Un tour de poule est l'association d'une poule à une coupe.

- **la phase éliminatoire** avec les éliminations des quarts, des demis et de la finale.
Une phase d'élimination est l'association d'une coupe à une élimination.
- Les poules, identifiées par une lettre unique, sont toujours les mêmes pour les coupes du monde (A, B, C, D)
- Les éliminations, identifiées par un nom unique, sont toujours les mêmes pour les coupes du monde (quarts, demis, finales)
- Pour une coupe, il y a 1 finale, 2 demi-finales, 4 quarts de finales et 1 petite finale
- Les matchs qui ont lieu pendant la coupe sont liés à une poule ou à une élimination

2. Gestion des matchs

- Un match oppose deux équipes et l'une d'entre elle est receveuse.
- Le score d'un match peut être retrouvé par :
 - jointures successives, grâce aux participations des joueurs et aux points marqués
 - somme des scores (pour un match, score de l'équipe qui reçoit + score de l'équipe reçu).
- Un match a lieu dans un stade, et est arbitré par 4 arbitres différents
- On connaît la nationalité de l'arbitre par son pays. **Il ne peut pas arbitrer un match où joue l'équipe de son pays**
- Pour chaque match, il y a 1 arbitre officiel, 2 juges de touche et 1 arbitre vidéo

3. Gestion des joueurs

- Un joueur joue pour une seule équipe. Un numéro unique permet de l'identifier parmi tous les joueurs
- Un joueur a un poste. Ces postes sont communs pour toutes les équipes (défenseur, ailier ...)
- Une équipe est liée à un seul pays, et est identifié par un code unique.
- Un joueur participe à plusieurs matchs, et il peut entrer et sortir à n'importe quel moment dans la partie. S'il est présent au début, il est titulaire.
- Pour une participation à un match, un joueur marque des points et reçoit des cartons
- Les points marqués sont identifiés par un nom unique (essai, transformation). Ils rapportent un nombre de point fixe
- Un carton est d'une couleur unique (rouge, jaune)

IV. Modèle physique de données

Ce diagramme représente l'implémentation du modèle dans un serveur de gestion de base de données MySQL :

- Les attributs sont typés (INT, VARCHAR, BLOB ...)
- Les clés primaires sont identifiées par un icône de clé à gauche de l'attribut
- Les attributs pouvant être nul sont représentés avec un carré vide.
- Les attributs ne pouvant pas être nul sont représentés par un carré plein
- Les index de clé étrangère sont en rouge, les attributs non clés sont en bleu

V. Relations en 3e forme normale

En établissant les dépendances fonctionnelles on trouve un schéma de relation déjà en 3NF

CoupeDuMonde : 3NF

idCoupeDuMonde -> annee

idCoupeDuMonde -> idpays

PhaseElimination : 3NF

idPhaseElimination -> idCoupeDuMonde, idelimination

Elimination : 3NF

idElimination -> nom

TourPoule : 3NF

idTourPoule -> idCoupeDuMonde, idPoule

Poule : 3NF

idPoule -> lettre

Match : 3NF

idMatch -> heure_debut, heure_fin, nombre_spectateur, date

idMatch -> idTourPoule, idElimination, idStade

Stade : 3NF

idStade -> nom, capacite

idStade -> idVille

Arbitre : 3NF

idArbitre -> numéro, nom, prénom

idArbitre -> idPays

Suit : 3NF

idSuit -> role

idSuit -> idarbitre, idmatch

Equipe : 3NF

idEquipe -> maillot, entraîneur, logo

idEquipe -> idPays

Oppose : 3NF

idOppose -> receveur, score

idOppose -> idEquipe, idMatch

Participation : 3NF

idParticipation -> titulaire, heure_debut, heure_fin

idParticipation -> idJoueur, idMatch

Joueur : 3NF

idJoueur -> numéro, nom, prénom, date_de_naissance, lieu_de_naissance, poids, taille, photo, numéro_poste

idJoueur -> idEquipe, idPoste

Poste : 3NF

idPoste -> libelle

Carton : 3NF

idCarton -> idParticipation, idCouleur

Couleur : 3NF

idCouleur -> nom

PointMarque : 3NF

idPointMarque -> idParticipation, idAction

Action : 3NF

idAction -> nom, points

Pays : 3NF

idPays -> nom

idPays -> idEquipe

Ville : 3NF

idVille -> nom

idVille -> idpays

VI. Script de génération SQL

Le script suivant permet de générer le schéma et les index en MySQL en version 5 et avec le moteur InnoDB :

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
```

```
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,  
FOREIGN_KEY_CHECKS=0;
```

```
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';
```

```
DROP SCHEMA IF EXISTS `coupe_du_monde` ;
```

```
CREATE SCHEMA IF NOT EXISTS `coupe_du_monde` DEFAULT CHARACTER SET utf8
```

```
COLLATE utf8_general_ci ;
```

```
USE `coupe_du_monde` ;
```

```
-- -----
```

```
-- Table `coupe_du_monde`.`Pays`
```

```

-----
DROP TABLE IF EXISTS `coupe_du_monde`.`Pays` ;

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Pays` (
  `idPays` INT NOT NULL ,
  `nom` VARCHAR(100) NOT NULL ,
  PRIMARY KEY (`idPays`, `nom`) )
ENGINE = InnoDB;

-----
-- Table `coupe_du_monde`.`CoupeDuMonde`
-----
DROP TABLE IF EXISTS `coupe_du_monde`.`CoupeDuMonde` ;

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`CoupeDuMonde` (
  `idCoupeDuMonde` INT NOT NULL ,
  `annee` YEAR NOT NULL ,
  `pays` INT NOT NULL ,
  PRIMARY KEY (`idCoupeDuMonde`) ,
  CONSTRAINT `pays_CoupeDuMonde_fk`
  FOREIGN KEY (`pays`)
  REFERENCES `coupe_du_monde`.`Pays` (`idPays`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `coupe_du_monde`.`Élimination`
-----
DROP TABLE IF EXISTS `coupe_du_monde`.`Élimination` ;

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Élimination` (
  `idPhaseElimination` INT NOT NULL ,
  `nom` VARCHAR(100) NOT NULL ,
  PRIMARY KEY (`idPhaseElimination`) )
ENGINE = InnoDB;

-----
-- Table `coupe_du_monde`.`PhaseElimination`
-----
DROP TABLE IF EXISTS `coupe_du_monde`.`PhaseElimination` ;

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`PhaseElimination` (
  `idPhaseElimination` INT NOT NULL AUTO_INCREMENT ,
  `coupe` INT NOT NULL ,
  `epreuve` INT NOT NULL ,
  PRIMARY KEY (`idPhaseElimination`) ,
  CONSTRAINT `coupe_PhaseElimination_fk`
  FOREIGN KEY (`coupe`)
  REFERENCES `coupe_du_monde`.`CoupeDuMonde` (`idCoupeDuMonde`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,

```

```

CONSTRAINT `epreuve_PhaseElimination_fk`
  FOREIGN KEY (`epreuve`)
    REFERENCES `coupe_du_monde`.`Élimination` (`idPhaseElimination`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `coupe_du_monde`.`Poule`
-----
DROP TABLE IF EXISTS `coupe_du_monde`.`Poule`;

```

```

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Poule` (
  `idPoule` INT NOT NULL,
  `lettre` CHAR NOT NULL,
  PRIMARY KEY (`idPoule`))
ENGINE = InnoDB;

```

```

-----
-- Table `coupe_du_monde`.`TourPoule`
-----
DROP TABLE IF EXISTS `coupe_du_monde`.`TourPoule`;

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`TourPoule` (
  `idTourPoule` INT NOT NULL AUTO_INCREMENT,
  `coupe` INT NOT NULL,
  `poule` INT NOT NULL,
  PRIMARY KEY (`idTourPoule`),
  CONSTRAINT `poule_TourPoule_fk`
    FOREIGN KEY (`poule`)
      REFERENCES `coupe_du_monde`.`Poule` (`idPoule`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `coupe_TourPoule_fk`
    FOREIGN KEY (`coupe`)
      REFERENCES `coupe_du_monde`.`CoupeDuMonde` (`idCoupeDuMonde`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `coupe_du_monde`.`Ville`
-----
DROP TABLE IF EXISTS `coupe_du_monde`.`Ville`;

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Ville` (
  `idVille` INT NOT NULL,
  `nom` VARCHAR(255) NOT NULL,
  `pays` INT NOT NULL,
  PRIMARY KEY (`idVille`),
  CONSTRAINT `nom_Ville_fk`
    FOREIGN KEY (`pays`)

```

```

REFERENCES `coupe_du_monde`.`Pays` (`idPays` )
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `coupe_du_monde`.`Stade`
-----

```

```

DROP TABLE IF EXISTS `coupe_du_monde`.`Stade` ;

```

```

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Stade` (
  `idStade` INT NOT NULL AUTO_INCREMENT ,
  `nom` VARCHAR(255) NOT NULL ,
  `capacite` INT NOT NULL ,
  `ville` INT NOT NULL ,
  PRIMARY KEY (`idStade`) ,
  CONSTRAINT `ville_Stade_fk`
    FOREIGN KEY (`ville` )
      REFERENCES `coupe_du_monde`.`Ville` (`idVille` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `coupe_du_monde`.`Match`
-----

```

```

DROP TABLE IF EXISTS `coupe_du_monde`.`Match` ;

```

```

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Match` (
  `idMatch` INT NOT NULL AUTO_INCREMENT ,
  `heure_debut` DATETIME NOT NULL ,
  `heure_fin` DATETIME NULL ,
  `nombre_spectateur` INT NOT NULL ,
  `epreuve` INT NULL ,
  `poule` INT NULL ,
  `stade` INT NOT NULL ,
  PRIMARY KEY (`idMatch`) ,
  CONSTRAINT `epreuve_Match_fk`
    FOREIGN KEY (`epreuve` )
      REFERENCES `coupe_du_monde`.`PhaseElimination` (`idPhaseElimination` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `poule_Match_fk`
    FOREIGN KEY (`poule` )
      REFERENCES `coupe_du_monde`.`TourPoule` (`idTourPoule` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `stade_Match_fk`
    FOREIGN KEY (`stade` )
      REFERENCES `coupe_du_monde`.`Stade` (`idStade` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `coupe_du_monde`.`Arbitre`
-----
DROP TABLE IF EXISTS `coupe_du_monde`.`Arbitre` ;

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Arbitre` (
  `idArbitre` INT NOT NULL ,
  `numero` INT NOT NULL ,
  `nom` VARCHAR(100) NOT NULL ,
  `prenom` VARCHAR(100) NOT NULL ,
  `pays` INT NOT NULL ,
  PRIMARY KEY (`idArbitre`) ,
  CONSTRAINT `pays_Arbitre_fk`
  FOREIGN KEY (`pays` )
  REFERENCES `coupe_du_monde`.`Pays` (`idPays` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `coupe_du_monde`.`Suit`
-----
DROP TABLE IF EXISTS `coupe_du_monde`.`Suit` ;

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Suit` (
  `idSuit` INT NOT NULL AUTO_INCREMENT ,
  `role` VARCHAR(100) NOT NULL ,
  `match` INT NOT NULL ,
  `arbitre` INT NOT NULL ,
  PRIMARY KEY (`idSuit`) ,
  CONSTRAINT `arbitre_Suit_fk`
  FOREIGN KEY (`arbitre` )
  REFERENCES `coupe_du_monde`.`Arbitre` (`idArbitre` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
  CONSTRAINT `match_Suit_fk`
  FOREIGN KEY (`match` )
  REFERENCES `coupe_du_monde`.`Match` (`idMatch` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `coupe_du_monde`.`Équipe`
-----
DROP TABLE IF EXISTS `coupe_du_monde`.`Équipe` ;

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Équipe` (
  `idEquipe` INT NOT NULL ,
  `code` INT NOT NULL ,
  `maillot` VARCHAR(100) NOT NULL ,

```

```

`entraîneur` VARCHAR(255) NOT NULL ,
`logo` BLOB NOT NULL ,
PRIMARY KEY (`idEquipe`) )
ENGINE = InnoDB;

```

```

-----
-- Table `coupe_du_monde`.`Poste`
-----

```

```

DROP TABLE IF EXISTS `coupe_du_monde`.`Poste` ;

```

```

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Poste` (
  `idPoste` INT NOT NULL ,
  `libelle` VARCHAR(100) NOT NULL ,
  PRIMARY KEY (`idPoste`) )
ENGINE = InnoDB;

```

```

-----
-- Table `coupe_du_monde`.`Joueur`
-----

```

```

DROP TABLE IF EXISTS `coupe_du_monde`.`Joueur` ;

```

```

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Joueur` (
  `idJoueur` INT NOT NULL ,
  `numero` INT NOT NULL ,
  `nom` VARCHAR(100) NOT NULL ,
  `prenom` VARCHAR(100) NOT NULL ,
  `date_de_naissance` DATE NOT NULL ,
  `lieu_de_naissance` VARCHAR(100) NOT NULL ,
  `poids` INT NOT NULL ,
  `taille` INT NOT NULL ,
  `photo` BLOB NOT NULL ,
  `numero_poste` INT NOT NULL ,
  `equipe` INT NOT NULL ,
  `poste` INT NOT NULL ,
  PRIMARY KEY (`idJoueur`) ,
  CONSTRAINT `poste_Joueur_fk`
    FOREIGN KEY (`poste`)
      REFERENCES `coupe_du_monde`.`Poste` (`idPoste`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `equipe_Joueur_fk`
    FOREIGN KEY (`equipe`)
      REFERENCES `coupe_du_monde`.`Équipe` (`idEquipe`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `coupe_du_monde`.`Couleur`
-----

```

```

DROP TABLE IF EXISTS `coupe_du_monde`.`Couleur` ;

```

```
CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Couleur` (
  `idCouleur` INT NOT NULL ,
  `couleur` VARCHAR(50) NOT NULL ,
  PRIMARY KEY (`idCouleur`) )
ENGINE = InnoDB;
```

```
-- -----
-- Table `coupe_du_monde`.`Participation`
-- -----
```

```
DROP TABLE IF EXISTS `coupe_du_monde`.`Participation` ;
```

```
CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Participation` (
  `idParticipation` INT NOT NULL AUTO_INCREMENT ,
  `titulaire` TINYINT(1) NOT NULL ,
  `heure_debut` DATETIME NOT NULL ,
  `heure_fin` DATETIME NULL ,
  `joueur` INT NOT NULL ,
  `match` INT NOT NULL ,
  PRIMARY KEY (`idParticipation`) ,
  CONSTRAINT `joueur_Participation_fk`
    FOREIGN KEY (`joueur` )
    REFERENCES `coupe_du_monde`.`Joueur` (`idJoueur` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `match_Participation_fk`
    FOREIGN KEY (`match` )
    REFERENCES `coupe_du_monde`.`Match` (`idMatch` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```

```
-- -----
-- Table `coupe_du_monde`.`Carton`
-- -----
```

```
DROP TABLE IF EXISTS `coupe_du_monde`.`Carton` ;
```

```
CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Carton` (
  `idCarton` INT NOT NULL AUTO_INCREMENT ,
  `couleur` INT NOT NULL ,
  `participation` INT NOT NULL ,
  PRIMARY KEY (`idCarton`) ,
  CONSTRAINT `couleur_Carton_fk`
    FOREIGN KEY (`couleur` )
    REFERENCES `coupe_du_monde`.`Couleur` (`idCouleur` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `participation_Carton_fk`
    FOREIGN KEY (`participation` )
    REFERENCES `coupe_du_monde`.`Participation` (`idParticipation` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
```



```

-----
-- Table `coupe_du_monde`.`Action`
-----
DROP TABLE IF EXISTS `coupe_du_monde`.`Action` ;

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Action` (
  `idAction` INT NOT NULL ,
  `nom` VARCHAR(100) NOT NULL ,
  `points` INT NOT NULL ,
  PRIMARY KEY (`idAction`) )
ENGINE = InnoDB;

-----
-- Table `coupe_du_monde`.`PointMarque`
-----
DROP TABLE IF EXISTS `coupe_du_monde`.`PointMarque` ;

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`PointMarque` (
  `idPointMarque` INT NOT NULL AUTO_INCREMENT ,
  `action` INT NOT NULL ,
  `participation` INT NOT NULL ,
  PRIMARY KEY (`idPointMarque`) ,
  CONSTRAINT `action_PointMarque_fk`
    FOREIGN KEY (`action`)
      REFERENCES `coupe_du_monde`.`Action` (`idAction`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `participation_PointMarque_fk`
    FOREIGN KEY (`participation`)
      REFERENCES `coupe_du_monde`.`Participation` (`idParticipation`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

-----
-- Table `coupe_du_monde`.`Oppose`
-----
DROP TABLE IF EXISTS `coupe_du_monde`.`Oppose` ;

CREATE TABLE IF NOT EXISTS `coupe_du_monde`.`Oppose` (
  `idOppose` INT NOT NULL AUTO_INCREMENT ,
  `receveur` TINYINT(1) NOT NULL ,
  `score` INT NULL ,
  `equipe` INT NOT NULL ,
  `match` INT NOT NULL ,
  PRIMARY KEY (`idOppose`) ,
  CONSTRAINT `equipe_Oppose_fk`
    FOREIGN KEY (`equipe`)
      REFERENCES `coupe_du_monde`.`Équipe` (`idEquipe`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `match_Oppose_fk`

```

```
FOREIGN KEY (`match` )  
REFERENCES `coupe_du_monde`.`Match` (`idMatch` )  
ON DELETE NO ACTION  
ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

VII. Difficultés

La plus grande partie de notre réflexion a porté sur la schématisation du problème. Extraire le schéma Entité / Association n'a pas été chose facile, et nous avons hésité entre plusieurs solutions, chacune répondant plus ou moins bien à l'enjeu de l'exercice.

Par exemple, nous pouvions faciliter et optimiser les requêtes de recherche en reliant les équipes directement aux poules. Cependant, cela aurait compliqué les accès en écriture et les mises à jour.

Au final, nous avons opté pour la solution la plus robuste. Les informations pourront toujours être regroupées dans des vues.

Concernant la technique, nous n'avons pas pu trouver un logiciel efficace de schématisation fonctionnant à la fois sur Windows et Linux. Les logiciels de modélisation UML semblent bien plus fournis et efficace dans ce domaine, mais l'utilisation de SyBase en TP nous a permis de palier à ce soucis.

L'avantage de ces outils est qu'ils permettent de générer directement les schémas relationnels. A cause de la dimension logique de la modélisation, le résultat généré pour le schéma EA doit pourtant être interprété manuellement. Ce problème ne s'est pas posé en MySQL.

En conclusion, l'étape conceptualisation est rendue difficile par le choix des modèles et des outils. Une méthodologie plus stricte permettrait sans doute de se focaliser davantage sur la réponse au problème.