

---

# COMPTE RENDU BASE DE DONNEE COUPE DU MONDE 2018

---

30 JUIN 2023

## **I. Introduction**

Dans le cadre de la Coupe du Monde de la FIFA 2018, il était impératif de mettre en place une base de données bien conçue pour gérer efficacement toutes les données liées à l'événement. Ce compte rendu détaille le processus de conception et de modélisation de cette base de données que nous avons simulée en classe, ainsi que les défis rencontrés et les compétences acquises tout au long du projet.

## **II. Contexte et Objectifs du Projet**

La base de données de la Coupe du Monde 2018 avait pour mission de centraliser toutes les informations pertinentes sur les équipes, les joueurs, les matchs, les stades, et les statistiques de jeu. L'objectif principal était de fournir aux utilisateurs une plateforme où ils pourraient accéder facilement à ces données et générer des rapports pour analyser les performances des équipes et des joueurs.

## **III. Conception de la Base de Données**

La conception de la base de données a été réalisée en suivant une approche relationnelle. Les principales entités identifiées étaient les équipes, les joueurs, les matchs, les stades et les statistiques. Chaque entité était associée à des attributs spécifiques décrivant ses caractéristiques.

### **Compétences Acquises :**

- Compréhension des concepts de modélisation de données avec UML.
- Capacité à identifier les classes et leurs attributs à partir des besoins du projet.

### **Difficultés Rencontrées :**

- Détermination de la structure optimale de la base de données pour répondre à toutes les exigences du projet.
- Détermination des associations et des multiplicités entre les classes.
- Gestion des dépendances et des relations entre les différentes entités.

## **IV. Modélisation de la Base de Données**

Le diagramme de classes UML a servi de base pour la modélisation de la base de données relationnelle. Chaque classe a été transformée en une table, et les

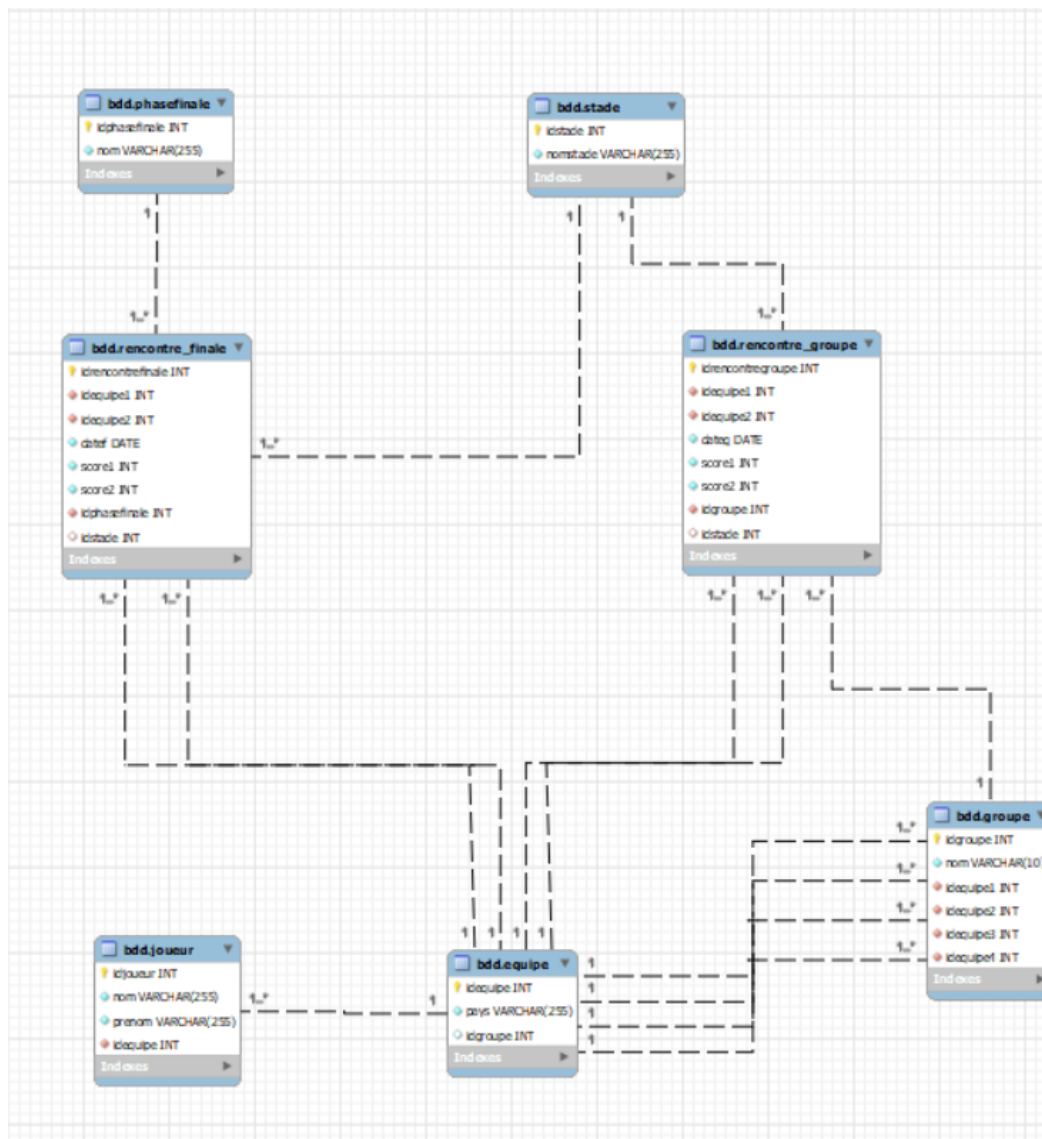
associations entre les classes ont été représentées par des clés étrangères dans les tables correspondantes. Les cardinalités ont été spécifiées pour définir les relations entre les entités.

### Compétences Acquisies :

- Maîtrise de la transformation d'un diagramme de classes en schéma relationnel.
- Capacité à définir les relations et les contraintes d'intégrité référentielle entre les tables.

### Difficultés Rencontrées :

- Gestion des relations many-to-many et des tables de jointure nécessaires pour les représenter dans la base de données.



## V. Requêtes Complexes

Des requêtes SQL complexes ont été élaborées pour extraire des informations spécifiques de la base de données. Cela incluait des requêtes pour récupérer les détails des matchs joués dans un stade particulier, les statistiques d'un joueur spécifique, ou les performances globales d'une équipe pendant le tournoi.

### Compétences Acquises :

- Capacité à formuler des requêtes SQL complexes pour interroger une base de données.
- Compréhension des concepts avancés de SQL tels que les jointures, les sous-requêtes et les agrégats.

### Difficultés Rencontrées :

- Optimisation des requêtes pour assurer des performances maximales de la base de données, en particulier pour les requêtes complexes sur de grandes quantités de données.

```
/* Avoir le nom des pays dans un groupe */
SELECT e.pays FROM equipe e , groupe g WHERE g.idgroupe = e.idgroupe AND g.nom = "GROUPE A";

SELECT e.pays FROM equipe e , groupe g
WHERE g.nom = "GROUPE A"
AND (e.idequipe = g.idequipe1 OR e.idequipe = g.idequipe2 OR e.idequipe = g.idequipe3 OR e.idequipe = g.idequipe4 );

/*Avoir les finaliste*/
SELECT e.pays FROM equipe e , rencontre_finale rf , phasefinale pf
WHERE pf.idphasefinale = rf.idphasefinale AND pf.nom = "La finale"
AND (e.idequipe = rf.idequipe1 OR e.idequipe = rf.idequipe2) ;

/*Les equipes lors du premier match*/
SELECT e.pays , rg.dateg FROM equipe e , rencontre_groupe rg
WHERE (e.idequipe = rg.idequipe1 OR e.idequipe = rg.idequipe2)
ORDER BY dateg LIMIT 2 ;

SELECT e1.pays , e2.pays FROM equipe e1 , equipe e2, rencontre_groupe rg
WHERE e1.idequipe = rg.idequipe1 AND e2.idequipe = rg.idequipe2 LIMIT 1 ;

SELECT e.pays , DATE_FORMAT(rg.dateg, '%d %m %y') FROM equipe e , rencontre_groupe rg
WHERE (e.idequipe = rg.idequipe1 OR e.idequipe = rg.idequipe2)
ORDER BY dateg LIMIT 2;

/* Les noms de joueurs d'une équipe */
SELECT j.nom , j.prenom FROM equipe e , joueur j
WHERE e.idequipe = j.idequipe AND e.pays = "BRESIL"
ORDER BY j.nom ASC;
```

```

/* Les jours où l'équipe x a joué */
SELECT rg.dateg FROM equipe e , rencontre_groupe rg
WHERE (e.idequipe = rg.idequipe1 OR e.idequipe = rg.idequipe2) AND e.pays = "FRANCE"
UNION
SELECT rf.datef FROM equipe e , rencontre_finale rf
WHERE (e.idequipe = rf.idequipe1 OR e.idequipe = rf.idequipe2) AND e.pays = "FRANCE" ;

SELECT rg.dateg AS "Les jours où la France a joué " FROM equipe e , rencontre_groupe rg
WHERE (e.idequipe = rg.idequipe1 OR e.idequipe = rg.idequipe2) AND e.pays = "FRANCE"
UNION
SELECT rf.datef FROM equipe e , rencontre_finale rf
WHERE (e.idequipe = rf.idequipe1 OR e.idequipe = rf.idequipe2) AND e.pays = "FRANCE" ;

/* Le nombre total de buts marqué dans la compétition */

SELECT SUM(rf.score1+rf.score2+rg.score1+rg.score2) AS
"Nombre de buts marqué dans la compétition" FROM
rencontre_finale rf, rencontre_groupe rg WHERE
( rf.idequipe1 = rg.idequipe1 OR rf.idequipe2 = rg.idequipe2 ) ;

/* Pays vainqueur du tournoi */
SELECT e.pays
FROM rencontre_finale rf, equipe e
WHERE e.idequipe = rf.idequipe1 AND rf.score1 > rf.score2
AND datef = (SELECT MAX(datef) FROM rencontre_finale)
UNION
SELECT e.pays
FROM rencontre_finale rf, equipe e
WHERE e.idequipe = rf.idequipe2 AND rf.score1 < rf.score2
AND datef = (SELECT MAX(datef) FROM rencontre_finale);

```

## VI. Conclusion

La conception et la modélisation de la base de données de la Coupe du Monde 2018 ont été des étapes cruciales pour garantir le bon fonctionnement de l'application et la disponibilité des données. Ce projet a permis d'acquérir de précieuses compétences en conception de bases de données relationnelles et en interrogation de données à l'aide de requêtes SQL complexes.

## VII. Annexes

### A. Schéma de la Base de Données

```

CREATE TABLE equipe (
    idequipe INT NOT NULL AUTO_INCREMENT,
    pays VARCHAR(255) NOT NULL,
    idgroupe INT,
    PRIMARY KEY (idequipe)
);

CREATE TABLE groupe (
    idgroupe INT NOT NULL AUTO_INCREMENT,
    nom VARCHAR (10) NOT NULL,
    idequipe1 INT NOT NULL,
    idequipe2 INT NOT NULL,
    idequipe3 INT NOT NULL,
    idequipe4 INT NOT NULL,
    PRIMARY KEY (idgroupe),
    CONSTRAINT groupe_equipe1_fk FOREIGN KEY (idequipe1) REFERENCES equipe(idequipe) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT groupe_equipe2_fk FOREIGN KEY (idequipe2) REFERENCES equipe(idequipe) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT groupe_equipe3_fk FOREIGN KEY (idequipe3) REFERENCES equipe(idequipe) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT groupe_equipe4_fk FOREIGN KEY (idequipe4) REFERENCES equipe(idequipe) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE joueur (
    idjoueur INT NOT NULL AUTO_INCREMENT,
    nom VARCHAR(255) NOT NULL,
    prenom VARCHAR(255) NOT NULL,
    idequipe INT NOT NULL,
    PRIMARY KEY (idjoueur),
    CONSTRAINT equipe_fk FOREIGN KEY (idequipe) REFERENCES equipe(idequipe) ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE stade (
    idstade INT NOT NULL AUTO_INCREMENT,
    nomstade VARCHAR (255) NOT NULL,
    PRIMARY KEY (idstade)
);

CREATE TABLE rencontre_groupe (
    idrencontregroupe INT NOT NULL AUTO_INCREMENT,
    idequipe1 INT NOT NULL,
    idequipe2 INT NOT NULL,
    dateg DATE NOT NULL,

```