

**Projet de bases de données relationnelles - Modélisation**

---

Novembre/décembre 2002  
Reponsable : Magdalena Puceva

Stéphane Décoppet  
Simon Barbey  
Reza Etemad-Sajadi  
Sandro Saitta

# Basketball



# Sommaire

<b>1. Abréviations et domaines</b>	2
<b>2. Diagramme EA</b>	3
<b>3. Définitions des TE</b>	4
ChampionShip	4
InterNation_ChampionShip	4
League_ChampionShip	4
Team	5
Club	5
National_team	5
Match	6
InterNation_match	6
League_match	6
Player	7
Sponsor	8
<b>4. Définitions des TA</b>	9
is_member_of	9
is_part_of	9
participates	9
plays	10
sponsORIZes	11
<b>5. Contraintes d'intégrité</b>	12

**Annexe :** le diagramme EA en mode texte.

## 1. Abréviations et domaines

EA	=	Entité-Association
TE	=	Type d'Entité
TA	=	Type d'Attribut
CI	=	Contraintes d'Intégrité
ID	=	IDentificateur

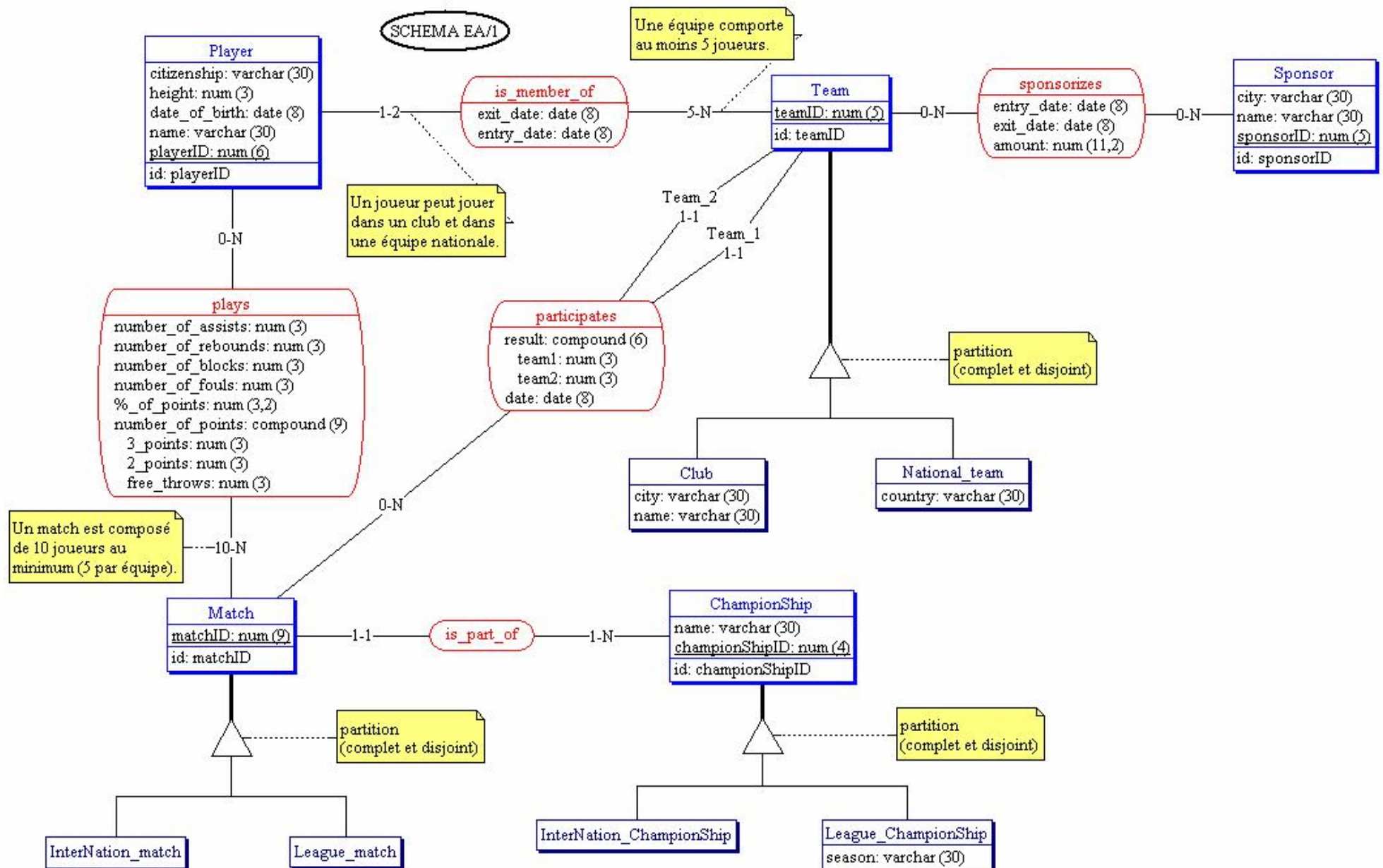
**Domaine** NumPos : [0 : ∞[

*Exemples :*

NumPos (3) signifie un type NumPos de longueur 3.

NumPos (5, 4) signifie une type NumPos de longueur 5 avec 4 chiffres après la virgule.

## 2. Diagramme EA



### 3. Définitions des TE



**Nom : Championship**

*Sur-type* : ---

*Définition* : généralisation des types de championnats possibles

*Identifiant(s)* : championshipID

*CI propre(s)* : ---

*Attribut(s)* :

- *Nom* : championshipID

*Définition* : identificateur de l'entité Championship

*Cardinalités* : 1:1

*Domaine* : NumPos (4)

- *Nom* : name

*Définition* : nom du championnat

*Cardinalités* : 1:1

*Domaine* : varchar (30)



**Nom : InterNation\_Championship**

*Sur-type* : Championship

*Définition* : la première spécialisation de l'entité Championship, tout type de championnats attribué aux équipes nationales

*Identifiant(s)* : Championship.championshipID

*CI propre(s)* : ---

*Attribut(s)* : ---



**Nom : League\_Championship**

*Sur-type* : Championship

*Définition* : la deuxième spécialisation de l'entité Championship, tout type de championnats attribué aux clubs

*Identifiant(s)* : Championship.championshipID

*CI propre(s)* : ---

*Attribut(s)* :

- *Nom* : season

*Définition* : sauvegarde de la saison durant laquelle le match s'est disputé

*Cardinalités* : 1:1

*Domaine* : varchar (30)



**Nom : Team**

*Sur-type* : ---

*Définition* : tout équipe de basketball

*Identifiant(s)* : teamID

*CI propre(s)* : ---

*Attribut(s)* :

- *Nom* : teamID

*Définition* : identificateur de l'entité Team

*Cardinalités* : 1:1

*Domaine* : NumPos (5)



**Nom : Club**

*Sur-type* : Team

*Définition* : tout club de basketball

*Identifiant(s)* : Team.teamID

*CI propre(s)* : ---

*Attribut(s)* :

- *Nom* : city

*Définition* : la ville où se trouve le club

*Cardinalités* : 1:1

*Domaine* : varchar (30)

- *Nom* : name

*Définition* : nom du club

*Cardinalités* : 1:1

*Domaine* : varchar (30)



**Nom : National\_team**

*Sur-type* : Team

*Définition* : toute équipe nationale de basketball

*Identifiant(s)* : ---

*CI propre(s)* : ---

*Attribut(s)* :

- *Nom* : country

*Définition* : le pays de l'équipe nationale

*Cardinalités* : 1:1

*Domaine* : varchar (30)



*Nom* : **Match**

*Sur-type* : ---

*Définition* : généralisation des types de matchs possibles

*Identifiant(s)* : matchID

*CI propre(s)* : ---

*Attribut(s)* :

- *Nom* : matchID

*Définition* : identificateur de l'entité Match

*Cardinalités* : 1:1

*Domaine* : NumPos (9)



*Nom* : **InterNation\_match**

*Sur-type* : Match

*Définition* : la première spécialisation de l'entité Match, tout type de matchs attribué aux équipes nationales

*Identifiant(s)* : Match.matchID

*CI propre(s)* : ---

*Attribut(s)* : ---



*Nom* : **League\_match**

*Sur-type* : Match

*Définition* : la deuxième spécialisation de l'entité Match, tout type de matchs attribué aux clubs

*Identifiant(s)* : Match.matchID

*CI propre(s)* : ---

*Attribut(s)* : ---



**Nom : Player**

*Sur-type* : ---

*Définition* : tout joueur de basket

*Identifiant(s)* : playerID

*CI propre(s)* : ---

*Attribut(s)* :

- *Nom* : playerID

*Définition* : identificateur de l'entité Player

*Cardinalités* : 1:1

*Domaine* : NumPos (6)

- *Nom* : name

*Définition* : nom du joueur

*Cardinalités* : 1:1

*Domaine* : varchar (30)

- *Nom* : date\_of\_birth

*Définition* : date de naissance du joueur

*Cardinalités* : 1:1

*Domaine* : date (8)

- *Nom* : height

*Définition* : taille du joueur en centimètres

*Cardinalités* : 1:1

*Domaine* : NumPos (3)

- *Nom* : country

*Définition* : pays d'origine du joueur

*Cardinalités* : 1:1

*Domaine* : varchar (30)

 **Nom : Sponsor**  
*Sur-type* : ---  
*Définition* : tout sponsor ayant sponsorisé une équipe à un moment donné  
*Identifiant(s)* : sponsorID  
*CI propre(s)* : ---  
*Attribut(s)* :

- *Nom* : name

*Définition* : le nom du sponsor

*Cardinalités* : 1:1

*Domaine* : varchar (30)

- *Nom* : city

*Définition* : la ville où réside le sponsor

*Cardinalités* : 1:1

*Domaine* : varchar (30)

- *Nom* : sponsorID

*Définition* : identifie un sponsor de manière unique

*Cardinalités* : 1:1

*Domaine* : NumPos (5)



## 4. Définitions des TA



**Nom : is\_member\_of**

*Définition* : lie un joueur à l'équipe à laquelle il appartient

*TE participants* : Player, Team

*Cardinalité(s)* Player (1:2), Team (5:N)

*Identifiant(s)* : Player.playerID et Team.teamID

*CI propre(s)* : pas de chevauchement sur les dates pour un joueur et une équipe donnée

*Attribut(s)* :

- *Nom* : entry\_date

*Définition* : date d'entrée du joueur dans l'équipe

*Cardinalités* : 1:1

*Domaine* : date (8)

- *Nom* : exit\_date

*Définition* : date de sortie du joueur de l'équipe

*Cardinalités* : 1:1

*Domaine* : date (8)



**Nom : is\_part\_of**

*Définition* : lie un match au championnat auquel il appartient

*TE participants* : Match, Championship

*Cardinalité(s)* : Match (1:1), Championship (1:N)

*Identifiant(s)* : Match.matchID

*CI propre(s)* : ---

*Attribut(s)* : ---



**Nom : participates**

*Définition* : 2 équipes participent à un match qui fait partie d'un championnat

*TE participants* : Team avec 2 roles: Team\_1, Team\_2.

Match avec 1 role.

*Cardinalité(s)* : Team\_1 (1:1), Team\_2 (1:1), Match (0:N)

*Identifiant(s)* : Team.teamID

*CI propre(s)* : Une équipe ne peut pas jouer contre elle-même

*Attribut(s)* :

- *Nom* : date

*Définition* : la date du match

*Cardinalités* : 1:1

*Domaine* : date (8)



**Nom : plays**

*Définition* : lie un joueur aux matchs auxquels il a participé

*TE participants* : Player, Match

*Cardinalité(s)* : Player (0:N), Match (10:N)

*Identifiant(s)* : Player.playerID et Match.matchID

*CI propre(s)* : le pourcentage de points ne peut pas être de 100% si le joueur n'a pas marqué de points. Le pourcentage de points ne peut pas être de 0% si le joueur a marqué au moins 1 point.

*Attribut(s)* :

- *Nom* : number\_of\_assists

*Définition* : nombre d'assist que le joueur a fait dans le match

*Cardinalités* : 1:1

*Domaine* : NumPos (3)

- *Nom* : number\_of\_rebounds

*Définition* : nombre de rebonds que le joueur a fait dans le match

*Cardinalités* : 1:1

*Domaine* : NumPos (3)

- *Nom* : number\_of\_blocks

*Définition* : nombre de blocks que le joueur a fait dans le match

*Cardinalités* : 1:1

*Domaine* : NumPos (3)

- *Nom* : number\_of\_fouls

*Définition* : nombre de fautes que le joueur a fait dans le match

*Cardinalités* : 1:1

*Domaine* : NumPos (3)

- *Nom* : %\_of\_points

*Définition* : pourcentage de points que le joueur a fait dans le match

*Cardinalités* : 1:1

*Domaine* : NumPos (5, 4)

- *Nom* : number\_of\_points

*Définition* : attribut complexe regroupant les différents types de points possibles

*Cardinalités* : 1:1

*Attributs composants* :

- *Nom* : 3\_points

*Définition* : nombre de 3 points que le joueur a fait dans le match

*Cardinalités* : 1:1

*Domaine* : NumPos (3)

- *Nom* : 2\_points

*Définition* : nombre de 2 points que le joueur a fait dans le match

*Cardinalités* : 1:1

*Domaine* : NumPos (3)

- *Nom* : free\_throws  
*Définition* : nombre de lancé-franc que le joueur a fait dans le match  
*Cardinalités* : 1:1  
*Domaine* : NumPos (3)



*Nom* : **sponsorizes**

*Définition* : lie un sponsor à une équipe

*TE participants* : Sponsor, Team

*Cardinalité(s)* : Sponsor (0:N), Team (0:N)

*Identifiant(s)* : Sponsor.sponsorID et Team.teamID

*CI propre(s)* : la date de sortie est plus grande que la date d'entrée pour une équipe et un sponsor, il ne peut pas y avoir de chevauchement sur les dates.

*Attribut(s)* :

- *Nom* : entry\_date  
*Définition* : date de début de sponsorisation  
*Cardinalités* : 1:1  
*Domaine* : date (8)
- *Nom* : exit\_date  
*Définition* : date de fin de sponsorisation  
*Cardinalités* : 1:1  
*Domaine* : NumPos (8)
- *Nom* : amount  
*Définition* : montant de la sponsorisation  
*Cardinalités* : 1:1  
*Domaine* : NumPos (11,2)

## 5. Contraintes d'intégrité

Voici les différentes CI de notre schéma EA.

- Chaque joueur [Player] peut changer de club [Club].
- Un joueur [Player] peut jouer dans une équipe nationale [National\_team] si et seulement s'il est de la nationalité correspondante.
- Les équipes [Team] peuvent changer de sponsors [Sponsor].
- Un club [Club] (respectivement une équipe nationale [National\_team]) ne peut pas participer à un championnat d'équipes nationales [InterNation\_ChampionShip] (respectivement interleague [League\_ChampionShip]).
- Une équipe [Team] ne peut pas jouer contre elle-même.
- Un joueur [Player] faisant partie d'une équipe nationale [National\_team] et d'un club [Club] ne peut jouer pour les deux à la même date.

## Projet de bases de données relationnelles – Traduction relationnelle

---

Janvier 2003  
Responsable : Magdalena Puceva

Stéphane Décoppet  
Simon Barbey  
Reza Etemad-Sajadi  
Sandro Saitta

# Basketball

## 2<sup>ème</sup> partie : Traduction relationnelle



# Sommaire

1. Introduction .....	2
2. Schéma conceptuel (corrigé) .....	2
3. Tables et schéma relationnelles .....	5
4. Requêtes SQL .....	10
Requête n°1 .....	10
Requête n°2 .....	10
Requête n°3 .....	11
Requête n°4 .....	11
Requête n°5 .....	12
Requête n°6 .....	13
Requête n°7 .....	14
5. Scripts Oracle .....	17

## 1. Introduction

Le but de cette partie du projet est d'effectuer la traduction relationnelle de notre schéma conceptuel. Pour cela nous avons d'abord corrigé notre schéma EA (chap. 2). Ensuite, nous en avons déduits les tables relationnelles (chap. 3) avec l'aide du cours "Base de données relationnelles". Pour finir, nous avons écrit et testé sur nos tables les requêtes SQL demandées (chap. 4).

## 2. Schéma conceptuel (corrigé)

Les modifications suivantes ont été apportées au schéma conceptuel original :

- Ajout de l'attribut `label` dans l'association `InterNation_match` qui spécifie le niveau du match (demi-finale ou finale).
- Renommage de certains attributs de `plays` et de `participates` pour des raisons pratiques.
- Renommage de l'attribut `citizenship` de `Player` en `country`. Nous avons remarqué que le pays d'origine du joueur était plus intéressant que sa citoyenneté.
- Changement du type de l'attribut `percent_points` de `num(3, 2)` en `num(3)`. En effet la valeur recherchée correspond à un pourcentage entre 0 et 100%.
- Suppression de l'attribut composé `result` de l'association `participates` qui était redondant.

Voici donc ci-dessous (page 3) la nouvelle version du schéma conceptuel revue et corrigé.



### 3. Tables relationnelles

En partant du schéma conceptuel (chap. 3) et en se référant au cours "Base de données relationnelles", nous en avons déduit les tables relationnelles ci-dessous ainsi que le schéma relationnel qui suit :

```
CREATE TABLE player (  
  playerID          INTEGER          NOT NULL,  
  country           VARCHAR2(30)     NOT NULL,  
  height            INTEGER          NOT NULL,  
  date_of_birth     VARCHAR2(8)      NOT NULL  
                                CHECK(LENGTH(date_of_birth) = 8),  
  name              VARCHAR2(30)     NOT NULL,  
  PRIMARY KEY (playerID)  
);  
  
CREATE TABLE team (  
  teamID            INTEGER          NOT NULL,  
  PRIMARY KEY (teamID)  
);  
  
CREATE TABLE club (  
  teamID            INTEGER          NOT NULL,  
  city              VARCHAR2(30)     NOT NULL,  
  name              VARCHAR2(30)     NOT NULL,  
  PRIMARY KEY (teamID),  
  FOREIGN KEY (teamID) REFERENCES team (teamID) ON DELETE  
  CASCADE  
);  
  
CREATE TABLE national_team (  
  teamID            INTEGER          NOT NULL,  
  country           VARCHAR2(30)     NOT NULL,  
  PRIMARY KEY (teamID),  
  FOREIGN KEY (teamID) REFERENCES team (teamID) ON DELETE  
  CASCADE  
);  
  
CREATE TABLE ChampionShip (  
  championShipID    INTEGER          NOT NULL,  
  name              VARCHAR2(30)     NOT NULL,  
  PRIMARY KEY (championShipID)  
);  
  
CREATE TABLE internation_championship (  
  championShipID    INTEGER          NOT NULL,  
  PRIMARY KEY (championShipID),  
  FOREIGN KEY (championShipID) REFERENCES championShip  
  (championShipID) ON DELETE CASCADE  
);
```



```

CREATE TABLE league_championship (
  championShipID      INTEGER          NOT NULL,
  season              VARCHAR2(30)     NOT NULL,
  PRIMARY KEY (championShipID),
  FOREIGN KEY (championShipID) REFERENCES championShip
  (championShipID) ON DELETE CASCADE
);

CREATE TABLE Match (
  matchID             INTEGER          NOT NULL,
  championShipID      INTEGER          NOT NULL,
  PRIMARY KEY (matchID),
  FOREIGN KEY (championShipID) REFERENCES championShip
  (championShipID) ON DELETE CASCADE
);

CREATE TABLE internation_match (
  matchID             INTEGER          NOT NULL,
  label              VARCHAR2(30)     NOT NULL,
  PRIMARY KEY (matchID),
  FOREIGN KEY (matchID) REFERENCES match (matchID) ON DELETE
  CASCADE
);

CREATE TABLE league_match (
  matchID             INTEGER          NOT NULL,
  PRIMARY KEY (matchID),
  FOREIGN KEY (matchID) REFERENCES match (matchID) ON DELETE
  CASCADE
);

CREATE TABLE sponsor (
  sponsorID          INTEGER          NOT NULL,
  city               VARCHAR2(30)     NOT NULL,
  name               VARCHAR2(30)     NOT NULL,
  PRIMARY KEY (sponsorID)
);

CREATE TABLE is_member_of (
  playerID           INTEGER          NOT NULL,
  teamID             INTEGER          NOT NULL,
  entry_date         VARCHAR2(8)     NOT NULL
                          CHECK(LENGTH(entry_date) = 8),
  exit_date          VARCHAR2(8)     NOT NULL
                          CHECK(LENGTH(exit_date) = 8),
  PRIMARY KEY (teamID, playerID),
  FOREIGN KEY (teamID) REFERENCES team (teamID) ON DELETE
  CASCADE,
  FOREIGN KEY (playerID) REFERENCES player (playerID),
  CHECK(entry_date <= exit_date)
);

```

```

);

CREATE TABLE participates (
matchID                INTEGER                NOT NULL,
team1ID                INTEGER                NOT NULL,
team2ID                INTEGER                NOT NULL,
participation_date     VARCHAR2(8)           NOT NULL
                                CHECK(LENGTH(participation_date) = 8),
PRIMARY KEY (team1ID, team2ID, matchID),
FOREIGN KEY (matchID) REFERENCES match (matchID) ON DELETE
CASCADE,
FOREIGN KEY (team1ID) REFERENCES team (teamID) ON DELETE
CASCADE,
FOREIGN KEY (team2ID) REFERENCES team (teamID) ON DELETE
CASCADE,
CHECK(team1ID <> team2ID)
);

CREATE TABLE plays (
matchID                INTEGER                NOT NULL,
playerID              INTEGER                NOT NULL,
nb_assists             INTEGER                NOT NULL
                                CONSTRAINT ass CHECK(nb_assists >= 0),
nb_rebounds            INTEGER                NOT NULL
                                CONSTRAINT reb CHECK(nb_rebounds >= 0),
nb_blocks              INTEGER                NOT NULL
                                CONSTRAINT blo CHECK(nb_blocks >= 0),
nb_fouls               INTEGER                NOT NULL
                                CONSTRAINT fou CHECK(nb_fouls >= 0),
percent_points         INTEGER                NOT NULL
                                CONSTRAINT per CHECK(percent_points
BETWEEN 0 AND 100),
nb_3_points            INTEGER                NOT NULL
                                CONSTRAINT po3 CHECK(nb_3_points >= 0),
nb_2_points            INTEGER                NOT NULL
                                CONSTRAINT po2 CHECK(nb_2_points >= 0),
nb_free_throws         INTEGER                NOT NULL
                                CONSTRAINT po1 CHECK(nb_free_throws >= 0),
PRIMARY KEY (matchID, playerID),
FOREIGN KEY (matchID) REFERENCES match (matchID) ON DELETE
CASCADE,
FOREIGN KEY (playerID) REFERENCES player (playerID) ON DELETE
CASCADE,
-- if the player has no points at all, the percent of points
-- for this player must be 0%
CONSTRAINT ban CHECK(NOT((
    (nb_3_points * 3) +
    (nb_2_points * 2) +
    nb_free_throws) = 0)
    AND NOT(percent_points = 0)),
-- if the player has at least 1 points, the percent of points

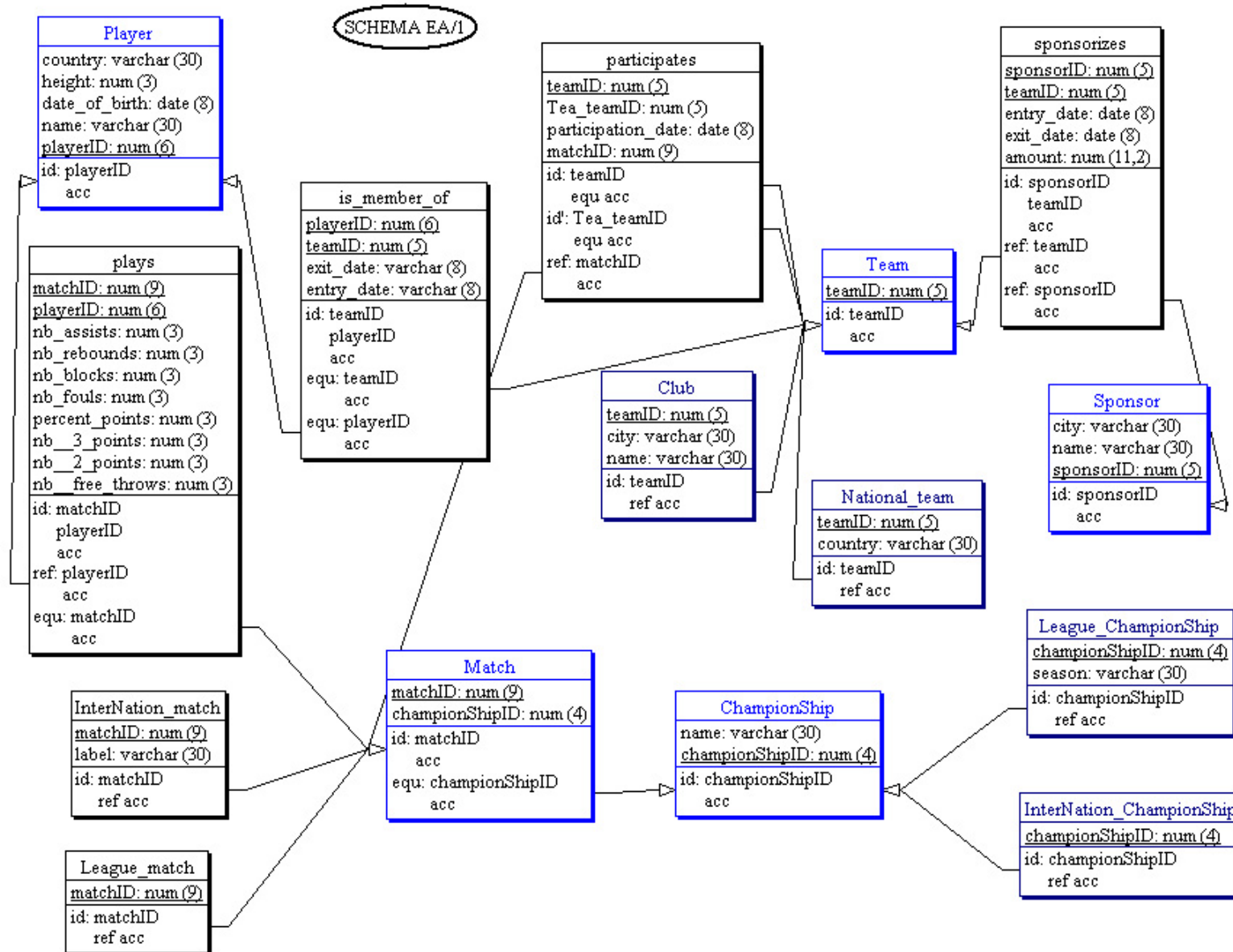
```

```

-- for this player must be different of 0%
CONSTRAINT bin CHECK((((nb_3_points * 3) +
                        (nb_2_points * 2) +
                        nb_free_throws) > 0)
AND (percent_points > 0))
);

CREATE TABLE sponsorizes (
sponsorID          INTEGER          NOT NULL,
teamID             INTEGER          NOT NULL,
entry_date         VARCHAR2(8)      NOT NULL
                        CHECK(LENGTH(entry_date) = 8),
exit_date          VARCHAR2(8)      NOT NULL
                        CHECK(LENGTH(exit_date) = 8),
amount             NUMBER(11,2)     NOT NULL
                        CHECK(amount >= 0),
PRIMARY KEY (sponsorID, teamID),
FOREIGN KEY (sponsorID) REFERENCES sponsor (sponsorID) ON
DELETE CASCADE,
FOREIGN KEY (teamID) REFERENCES team (teamID) ON DELETE
CASCADE,
CHECK(entry_date <= exit_date)
);

```



## 4. Requêtes SQL

Les 7 requêtes SQL (ainsi que les vues correspondantes) que nous avons effectué sont données ci-dessous :

**Requête n°1 :** List the first 10 players who have the greatest total number of points for their national teams ever ?

```
CREATE OR REPLACE VIEW player_greatest_points (name,
nb_points) AS
    SELECT player.name,
           SUM(plays.nb_3_points * 3 + plays.nb_2_points * 2 +
               plays.nb_free_throws) AS nb_points
    FROM player, is_member_of, national_team, plays
    WHERE is_member_of.playerID = player.playerID
    AND    plays.playerID = player.playerID
    AND    is_member_of.teamID = national_team.teamID
    AND    player.country = national_team.country
    GROUP BY player.name
    ORDER BY nb_points DESC;
```

```
SELECT name, nb_points
FROM player_greatest_points
WHERE ROWNUM < 11;
```

**Requête n°2 :** List the first 3 players who have the greatest free throws percentage in a final of the European Championship 2002 ?

-- nombre de free throws que le joueur à fait dans une finale de l'european championship 2002

```
CREATE OR REPLACE VIEW playerthrowsEC AS
    SELECT player.playerID,
           SUM(plays.nb_free_throws) AS tot_player_throws
    FROM player,
         plays,
         championship,
         internation_match,
         match
    WHERE player.playerID = plays.playerID
    AND    plays.matchID = match.matchID
    AND    match.championshipID = championship.championshipID
    AND    match.matchID = internation_match.matchID
    AND    championship.name = 'EuropeanChampionShip'
    AND    internation_match.label = 'finale'
    GROUP BY player.playerID;
```

-- nombre de free throws que l'équipe à fait dans une finale de l'european championship 2002

```
CREATE OR REPLACE VIEW natithrowsEC AS
    SELECT national_team.teamID,
```

```

        SUM(tot_player_throws) AS tot_team_throws
FROM playerthrowsEC, is_member_of, national_team
WHERE is_member_of.playerID = playerthrowsEC.playerID
AND    is_member_of.teamID = national_team.teamID
GROUP BY national_team.teamID;

-- pourcentage de free throws pour une finale de l'european
championship
CREATE OR REPLACE VIEW playerpercentthrows AS
    SELECT player.name,
           ROUND((playerthrowsEC.tot_player_throws /
natithrowsEC.tot_team_throws * 100), 2) AS percent_throws
    FROM playerthrowsEC,
         natithrowsEC,
         is_member_of,
         player
    WHERE is_member_of.playerID = playerthrowsEC.playerID
    AND    is_member_of.teamID = natithrowsEC.teamID
    AND player.playerID = playerthrowsEC.playerID
    ORDER BY percent_throws DESC;

SELECT * FROM playerpercentthrows
WHERE ROWNUM < 4;

```

**Requête n°3 : Which club has the highest average height ?**

```

CREATE OR REPLACE VIEW club_average_height (club,
average_height) AS
    SELECT club.name, AVG(player.height) AS average_height
    FROM player, is_member_of, club
    WHERE is_member_of.playerID = player.playerID
    AND    is_member_of.teamID = club.teamID
    GROUP BY club.name
    ORDER BY average_height DESC;

SELECT club, average_height
FROM club_average_height
WHERE ROWNUM < 2;

```

**Requête n°4 : Which sponsor has sponsored the most number of national teams that won the World Championship ?**

```

CREATE OR REPLACE VIEW ViewNationScore AS
SELECT interNation_match.matchID, National_team.teamID,
SUM(plays.nb_3_points * 3 + plays.nb_2_points * 2 +
plays.nb_free_throws) AS Score
FROM plays, interNation_match, Player, National_team,
is_member_of
WHERE interNation_match.matchID = plays.matchID
AND    National_team.teamID = is_member_of.teamID
AND    Player.playerID = plays.playerID

```

```

AND    is_member_of.playerID = player.playerID
GROUP BY interNation_match.matchID, national_team.teamID;

CREATE OR REPLACE VIEW ViewMatchWinnerWC (matchID, teamID) AS
SELECT team1.matchID, team1.teamID
FROM ViewNationScore team1,
     ViewNationScore team2
WHERE team1.Score > team2.Score
AND    team1.matchID = team2.matchID;

CREATE OR REPLACE VIEW ViewSponsorWinnerWC (Name, Num) AS
SELECT sponsor.name, COUNT(*) AS Num
FROM sponsorizes,
     Match,
     National_team,
     internation_match,
     ViewMatchWinnerWC,
     Championship,
     sponsor
WHERE internation_match.label = 'finale'
AND    Championship.name = 'WorldChampionShip'
AND    Match.matchID = ViewMatchWinnerWC.matchID
AND    sponsorizes.teamID = ViewMatchWinnerWC.teamID
AND    National_team.teamID = ViewMatchWinnerWC.teamID
AND    internation_match.matchID = Match.matchID
AND    sponsorizes.sponsorID = sponsor.sponsorID
GROUP BY sponsor.name
ORDER BY Num DESC;

SELECT Name, Num
FROM ViewSponsorWinnerWC
WHERE Num = (SELECT MAX(Num) FROM ViewSponsorWinnerWC);

```

**Requête n°5 :** For each club, list the players who have the greatest percentage of 3 points in the current season ?

```

-- nombre de 3 points que le joueur à fait dans la saison
CREATE OR REPLACE VIEW player3season AS
    SELECT player.playerID, SUM(plays.nb_3_points) AS
tot_nb_3_points
    FROM player, plays, match, league_championship
    WHERE player.playerID = plays.playerID
    AND    plays.matchID = match.matchID
    AND    match.championshipID =
league_championship.championshipID
    AND    league_championship.season = '2003'
    GROUP BY player.playerID;

-- nombre de 3 points que l'équipe à fait dans la saison
CREATE OR REPLACE VIEW club3season AS

```

```

        SELECT club.teamID, SUM(tot_nb_3_points) AS
team_nb_3_points
        FROM player3season, is_member_of, club
        WHERE is_member_of.playerID = player3season.playerID
        AND is_member_of.teamID = club.teamID
        GROUP BY club.teamID;

-- nombre de 3 points du meilleur joueur par equipe
CREATE OR REPLACE VIEW maxclub3season AS
        SELECT club3season.teamID,
MAX(player3season.tot_nb_3_points) AS max_3_points
        FROM player3season, club3season, is_member_of
        WHERE is_member_of.playerID = player3season.playerID
        AND is_member_of.teamID = club3season.teamID
        GROUP BY club3season.teamID;

-- nom de l'équipe avec ces meilleurs joueurs aux 3 points
pour la saison courante
SELECT club.name, player.name,
ROUND((player3season.tot_nb_3_points /
club3season.team_nb_3_points * 100), 2) AS
max_percent_3_points
FROM maxclub3season, is_member_of, player3season, player,
club3season, club
WHERE player3season.playerID = is_member_of.playerID
AND is_member_of.teamID = maxclub3season.teamID
AND player3season.tot_nb_3_points =
maxclub3season.max_3_points
AND player.playerID = player3season.playerID
AND maxclub3season.teamID = club3season.teamID
AND maxclub3season.teamID = club.teamID
ORDER BY club.name, player.name;

```

**Requête n°6 :** For a particular club, which player has the most number of assists per game ?

```

CREATE OR REPLACE VIEW club_best_assists (club, match, player,
best_number_of_assists) AS
        SELECT club.name, league_match.matchID, player.name,
plays.nb_assists
        FROM league_match, plays, is_member_of, club, player
        WHERE plays.matchID = league_match.matchID
        AND is_member_of.teamID = club.teamID
        AND is_member_of.playerID = player.playerID
        AND plays.playerID = player.playerID
        AND club.name = 'ParisBC'
ORDER BY league_match.matchID ASC;

CREATE OR REPLACE VIEW best_assists_match AS
        SELECT match, MAX(best_number_of_assists) AS max_assists
        FROM club_best_assists
        GROUP BY match;

```



```

SELECT club, best_assists_match.match, player, max_assists
FROM best_assists_match, club_best_assists
WHERE max_assists = best_number_of_assists
AND best_assists_match.match = club_best_assists.match
ORDER BY best_assists_match.match;

```

**Requête n°7 :** List all clubs who won the Euroleague more than 3 times ?

```

CREATE OR REPLACE VIEW score_team1 (championship, match, club,
score) AS
SELECT league_championship.championshipID,
league_match.matchID, club.name, SUM(plays.nb_3_points * 3 +
plays.nb_2_points * 2 + plays.nb_free_throws) AS nb_points
FROM plays, league_match, participates, club, player,
is_member_of, league_championship, match
WHERE club.teamID = participates.team1ID
AND participates.matchID = league_match.matchID
AND league_match.matchID = plays.matchID
AND league_championship.championshipID =
match.championshipID
AND player.playerID = plays.playerID
AND player.playerID = is_member_of.playerID
AND is_member_of.teamID = club.teamID
AND match.matchID = league_match.matchID
GROUP BY league_championship.championshipID,
league_match.matchID, club.name
ORDER BY league_match.matchID ASC;

```

```

CREATE OR REPLACE VIEW score_team2 (championship, match, club,
score) AS
SELECT league_championship.championshipID,
league_match.matchID, club.name, SUM(plays.nb_3_points * 3 +
plays.nb_2_points * 2 + plays.nb_free_throws) AS nb_points
FROM plays, league_match, participates, club, player,
is_member_of, league_championship, match
WHERE club.teamID = participates.team2ID
AND participates.matchID = league_match.matchID
AND league_match.matchID = plays.matchID
AND league_championship.championshipID =
match.championshipID
AND player.playerID = plays.playerID
AND player.playerID = is_member_of.playerID
AND is_member_of.teamID = club.teamID
AND match.matchID = league_match.matchID
GROUP BY league_championship.championshipID,
league_match.matchID, club.name
ORDER BY league_match.matchID ASC;

```

```

CREATE OR REPLACE VIEW win_team1 (team, match, championship)
AS

```

```

SELECT score_team1.club, score_team1.match,
score_team1.championship
FROM score_team1, score_team2
WHERE score_team1.score > score_team2.score
AND score_team1.match = score_team2.match;

CREATE OR REPLACE VIEW win_team2 (team, match, championship)
AS
SELECT score_team2.club, score_team2.match,
score_team2.championship
FROM score_team1, score_team2
WHERE score_team2.score > score_team1.score
AND score_team1.match = score_team2.match;

CREATE OR REPLACE VIEW win_team (team, match, championship) AS
SELECT * FROM win_team1
UNION
SELECT * FROM win_team2;

CREATE OR REPLACE VIEW win_team_ch (team, championship,
nb_win) AS
SELECT win_team.team, win_team.championship,
COUNT(win_team.championship)
FROM win_team
GROUP BY win_team.team, win_team.championship;

CREATE OR REPLACE VIEW champ_nb_win (championship, nb_max) AS
SELECT win_team_ch.championship, MAX(win_team_ch.nb_win)
FROM win_team_ch
GROUP BY win_team_ch.championship
ORDER BY win_team_ch.championship ASC;

CREATE OR REPLACE VIEW win_ch_team (team, championship) AS
SELECT DISTINCT win_team_ch.team, win_team_ch.championship
FROM win_team_ch, champ_nb_win
WHERE win_team_ch.nb_win = champ_nb_win.nb_max
ORDER BY win_team_ch.team ASC;

CREATE OR REPLACE VIEW nb_ch_win (team, nb_champ_win) AS
SELECT win_ch_team.team, COUNT(win_ch_team.team)
FROM win_ch_team
GROUP BY win_ch_team.team;

SELECT nb_ch_win.team
FROM nb_ch_win
WHERE nb_ch_win.nb_champ_win > 2;

```

## 5. Scripts Oracle

Les scripts Oracle nécessaires se trouvent dans le dossier bdr20 de la disquette en annexe.  
Les noms des fichiers sont les suivants :

`create.sql` : création des tables et contraintes  
`drop.sql` : destruction des tables  
`insert.sql` : chargement des tables  
`queries.sql` : exécution des requêtes

Notre username et notre password sont bdr20.