

# 2024-24303 - Data Bases - Universitat Pompeu Fabra

## 1st Lab Challenge

Jan Aguiló Plana, Pol-Antoni Bonet Malgà, Ramon Casas i Luque - Team P101-6

October 2024

This document includes the required sections as indicated in the assignment instructions of the first challenge of the subject. First we explain the assumptions we worked with to deploy our database, second we provide the conceptual and the relational model diagrams, and finally we show the script used to deploy our database in a SQL server. A full resolution version of both diagrams is available as an attached file in the submission, whereas the .sql file with the script is also provided.

## 1. Assumptions

Here is the list of assumptions of our implemented model:

- There are no 2 **stadiums** with the same **arena\_name**.
- Each **franchise** has only one stadium and each **stadium** belongs to only one **franchise**.
- Each **assistant coach** can only have one boss, i.e. one assigned **head coach**.
- We connected the **head coach** with the **national team** and with the **franchise** in two different relations 1 to 1 with optionality assuming that the **head coach** can train just a single **national team**, just a single **franchise**, one of each at the same time or none of them.
- The **mascot** represents the staff person who works as a mascot.
- Both the entity **Ticket** and the **Zone** have booleans indicating whether they are **VIP**. We are aware of it and we assumed these are independent: someone can have a **VIP ticket** without having a seat in a **VIP zone**.
- Since we connected the staff entity with the **franchise**, for the **mascot** we don't have any direct connection to the **franchise**. This setup still allows the **franchise** to have a **mascot** or not, as indicated in the assignment.
- Note that while a **player** can only play for a **national team**, as illustrated in the conceptual model, the relationship is many to many since a **player** can be selected for more than one year, i.e. it can belong to several rosters. Hence, this does not mean a player can play for multiple countries, but rather that it can be selected in different years. Regarding this, note also that nationality and the country a player plays for are not necessarily the same, as it accounts for exceptions that may occur in a real life setting.
- When implementing the database in SQL, for the **gender** we assume there are only male and female and we represent them as an ENUM variable.
- Instead of **age** we put **birthdate** so that the age can be queried easily and updates automatically.
- The city in the **franchise** and the city in the **stadium** may be different. That's why we put an attribute in each of the two entities.

## 2. Conceptual Model (E-R) Diagram

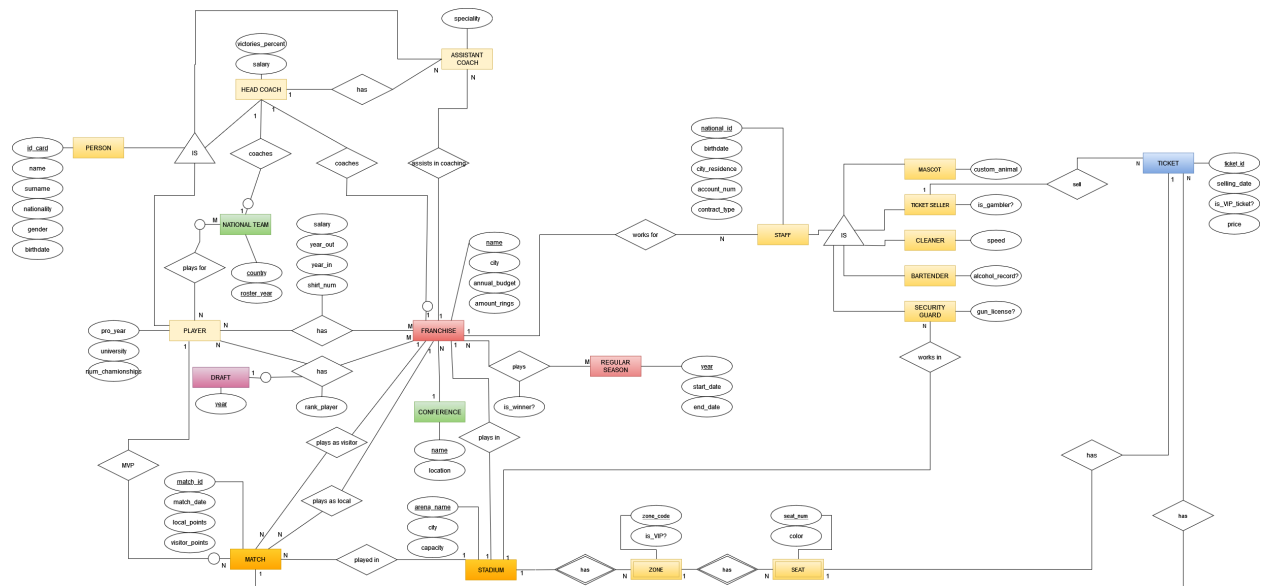


Figure 1: ER\_Diagram

### 3. Relational Model Diagram

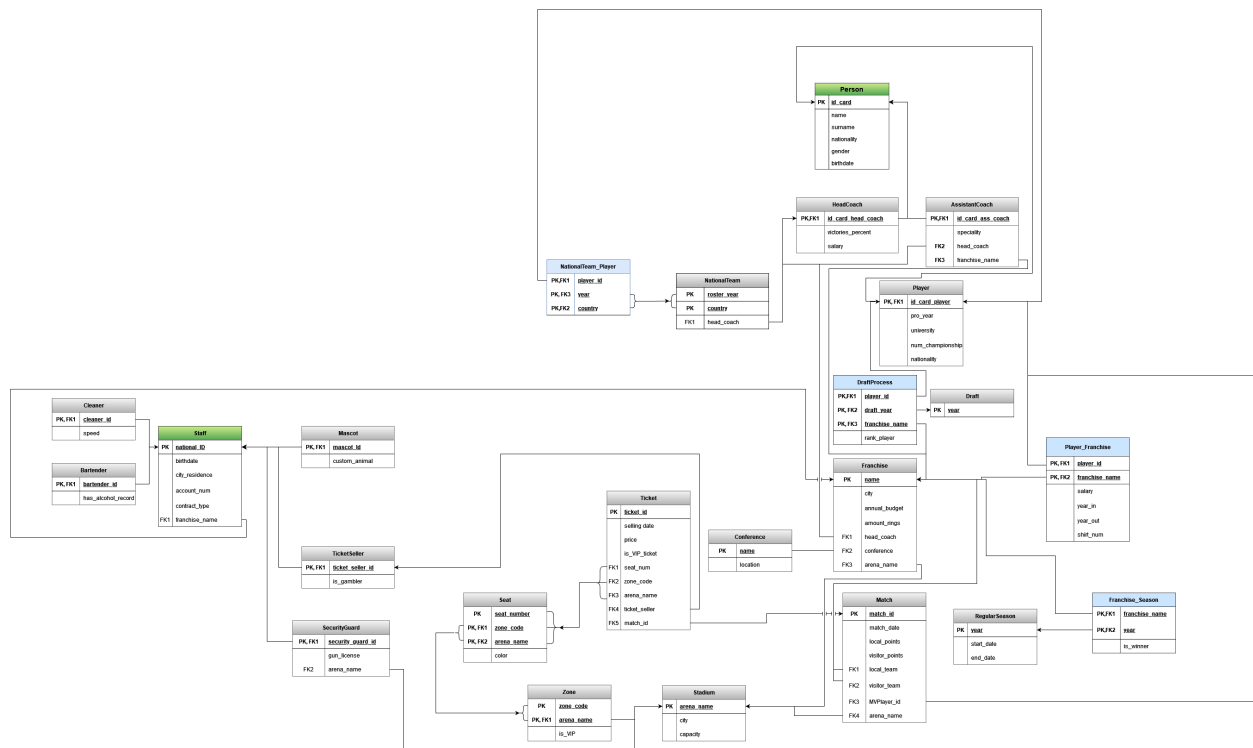


Figure 2: RM\_Diagram

## 4. SQL code

As indicated in the assignment, we deployed the database in the server:

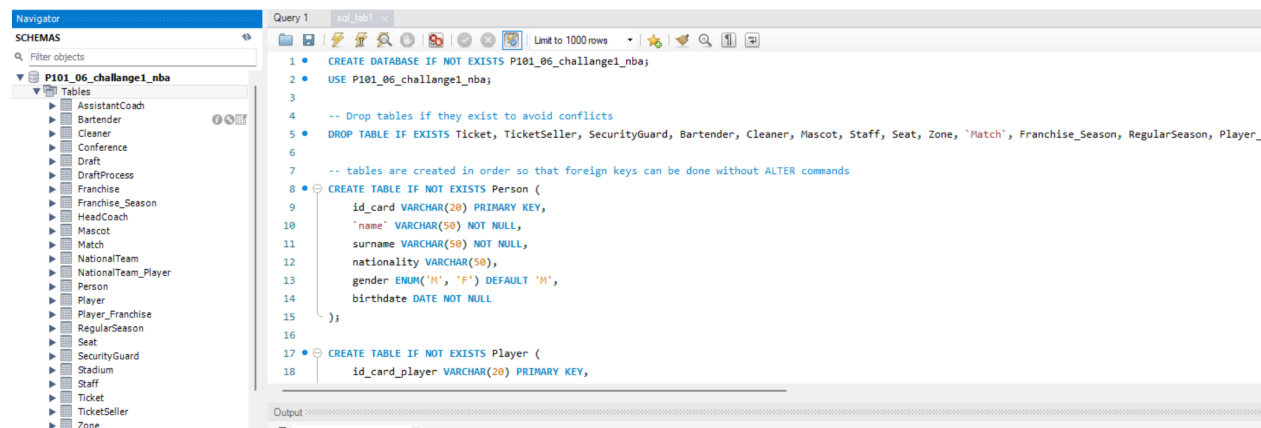


Figure 3: SQL\_NBA\_Database

We used the following code, which can be executed several times and always delivers the same result:

```
CREATE DATABASE IF NOT EXISTS P101_06_challenge1_nba;
USE P101_06_challenge1_nba;
```

```
-- Drop tables if they exist to avoid conflicts
```

```
DROP TABLE IF EXISTS Ticket, TicketSeller, SecurityGuard, Bartender, Cleaner, Mascot, Staff, Seat, Zone,
`Match`, Franchise_Season, RegularSeason, Player_Franchise, DraftProcess, Draft, NationalTeam_Player,
NationalTeam, AssistantCoach, Franchise, Stadium, Conference, HeadCoach, Player, Person;
```

```
-- tables are created in order so that foreign keys can be done without ALTER commands
```

```
CREATE TABLE IF NOT EXISTS Person (
    id_card VARCHAR(20) PRIMARY KEY,
    `name` VARCHAR(50) NOT NULL,
    surname VARCHAR(50) NOT NULL,
    nationality VARCHAR(50),
    gender ENUM('M', 'F') DEFAULT 'M',
    birthdate DATE NOT NULL
);
```

```
CREATE TABLE IF NOT EXISTS Player (
    id_card_player VARCHAR(20) PRIMARY KEY,
    pro_year INT UNSIGNED,
    university VARCHAR(100),
    num_championships INT UNSIGNED DEFAULT 0,
    FOREIGN KEY (id_card_player) REFERENCES Person(id_card)
);
```

```
CREATE TABLE IF NOT EXISTS HeadCoach (
    id_card_head_coach VARCHAR(20) PRIMARY KEY,
    victories_percent DECIMAL(5, 2) DEFAULT 0.00,
    salary DECIMAL(12, 2) NOT NULL,
    FOREIGN KEY (id_card_head_coach) REFERENCES Person(id_card)
);
```

```

CREATE TABLE IF NOT EXISTS Conference (
    `name` VARCHAR(50) PRIMARY KEY,
    location VARCHAR(50)
);

CREATE TABLE IF NOT EXISTS Stadium (
    arena_name VARCHAR(50) PRIMARY KEY,
    city VARCHAR(100),
    capacity INT UNSIGNED
);

CREATE TABLE IF NOT EXISTS Franchise (
    `name` VARCHAR(50) PRIMARY KEY,
    city VARCHAR(100),
    annual_budget DECIMAL(15, 2),
    amount_rings INT UNSIGNED DEFAULT 0,
    head_coach VARCHAR(20),
    conference VARCHAR(50),
    arena_name VARCHAR(50),
    FOREIGN KEY (head_coach) REFERENCES HeadCoach(id_card_head_coach),
    FOREIGN KEY (conference) REFERENCES Conference(`name`),
    FOREIGN KEY (arena_name) REFERENCES Stadium(arena_name)
);

CREATE TABLE IF NOT EXISTS AssistantCoach (
    id_card_ass_coach VARCHAR(20) PRIMARY KEY,
    speciality VARCHAR(50),
    franchise_name VARCHAR(50),
    FOREIGN KEY (id_card_ass_coach) REFERENCES Person(id_card),
    FOREIGN KEY (franchise_name) REFERENCES Franchise(`name`)
);

CREATE TABLE IF NOT EXISTS NationalTeam (
    roster_year INT UNSIGNED,
    country VARCHAR(50),
    head_coach VARCHAR(20),
    PRIMARY KEY (roster_year, country), -- Composite key!
    FOREIGN KEY (head_coach) REFERENCES HeadCoach(id_card_head_coach)
);

CREATE TABLE IF NOT EXISTS NationalTeam_Player (
    player_id VARCHAR(20),
    `year` INT UNSIGNED,
    country VARCHAR(50),
    PRIMARY KEY (player_id, `year`, country),
    FOREIGN KEY (player_id) REFERENCES Player(id_card_player),
    FOREIGN KEY (`year`, country) REFERENCES NationalTeam(roster_year, country)
);

CREATE TABLE IF NOT EXISTS Draft (
    `year` INT UNSIGNED PRIMARY KEY
);

```

```

CREATE TABLE IF NOT EXISTS DraftProcess (
    player_id VARCHAR(20),
    draft_year INT UNSIGNED,
    franchise_name VARCHAR(50),
    rank_player INT UNSIGNED,
    PRIMARY KEY (player_id, draft_year, franchise_name),
    FOREIGN KEY (player_id) REFERENCES Player(id_card_player),
    FOREIGN KEY (draft_year) REFERENCES Draft(`year`),
    FOREIGN KEY (franchise_name) REFERENCES Franchise(`name`)
);

CREATE TABLE IF NOT EXISTS Player_Franchise (
    player_id VARCHAR(20),
    franchise_name VARCHAR(50),
    salary DECIMAL(12, 2),
    year_in INT UNSIGNED,
    year_out INT UNSIGNED,
    shirt_num INT UNSIGNED,
    PRIMARY KEY(player_id, franchise_name),
    FOREIGN KEY (player_id) REFERENCES Player(id_card_player),
    FOREIGN KEY (franchise_name) REFERENCES Franchise(`name`)
);

CREATE TABLE IF NOT EXISTS RegularSeason (
    `year` INT UNSIGNED PRIMARY KEY,
    start_date DATE,
    end_date DATE
);

CREATE TABLE IF NOT EXISTS Franchise_Season (
    franchise_name VARCHAR(50),
    `year` INT UNSIGNED,
    is_winner BOOLEAN DEFAULT FALSE,
    PRIMARY KEY (franchise_name, `year`),
    FOREIGN KEY (franchise_name) REFERENCES Franchise(`name`),
    FOREIGN KEY (`year`) REFERENCES RegularSeason(`year`)
);

CREATE TABLE IF NOT EXISTS `Match` (
    match_id INT PRIMARY KEY AUTO_INCREMENT,
    match_date DATE,
    local_points INT UNSIGNED,
    visitor_points INT UNSIGNED,
    local_team VARCHAR(50),
    visitor_team VARCHAR(50),
    MVPlayer_id VARCHAR(20),
    arena_name VARCHAR(50),
    FOREIGN KEY (local_team) REFERENCES Franchise(`name`),
    FOREIGN KEY (visitor_team) REFERENCES Franchise(`name`),
    FOREIGN KEY (MVPlayer_id) REFERENCES Player(id_card_player),
    FOREIGN KEY (arena_name) REFERENCES Stadium(arena_name)
);

CREATE TABLE IF NOT EXISTS Zone (

```

```

zone_code VARCHAR(20), -- might include letters, as in the picture of the assignment
arena_name VARCHAR(50),
is_vip BOOLEAN DEFAULT FALSE,
PRIMARY KEY (zone_code, arena_name),
FOREIGN KEY (arena_name) REFERENCES Stadium(arena_name)
);

CREATE TABLE IF NOT EXISTS Seat (
    seat_number INT UNSIGNED,
    zone_code VARCHAR(20),
    arena_name VARCHAR(50),
    color VARCHAR(50),
    PRIMARY KEY (seat_number, zone_code, arena_name),
    FOREIGN KEY (zone_code, arena_name) REFERENCES Zone(zone_code, arena_name)
);

CREATE TABLE IF NOT EXISTS Staff (
    national_id VARCHAR(20) PRIMARY KEY,
    birthdate DATE,
    city_residence VARCHAR(100),
    account_num VARCHAR(50),
    contract_type ENUM('Full-Time', 'Part-Time') DEFAULT 'Full-Time',
    franchise_name VARCHAR(50),
    FOREIGN KEY (franchise_name) REFERENCES Franchise(`name`)
);

CREATE TABLE IF NOT EXISTS Mascot (
    mascot_id VARCHAR(20) PRIMARY KEY,
    custom_animal VARCHAR(50),
    FOREIGN KEY (mascot_id) REFERENCES Staff(national_id)
);

CREATE TABLE IF NOT EXISTS Cleaner (
    cleaner_id VARCHAR(20) PRIMARY KEY,
    speed DECIMAL(5, 2),
    FOREIGN KEY (cleaner_id) REFERENCES Staff(national_id)
);

CREATE TABLE IF NOT EXISTS Bartender (
    bartender_id VARCHAR(20) PRIMARY KEY,
    has_alcohol_record BOOLEAN, -- don't assume FALSE by default
    FOREIGN KEY (bartender_id) REFERENCES Staff(national_id)
);

CREATE TABLE IF NOT EXISTS SecurityGuard (
    security_guard_id VARCHAR(20) PRIMARY KEY,
    gun_license BOOLEAN, -- don't assume TRUE by default
    arena_name VARCHAR(50),
    FOREIGN KEY (security_guard_id) REFERENCES Staff(national_id),
    FOREIGN KEY (arena_name) REFERENCES Stadium(arena_name)
);

CREATE TABLE IF NOT EXISTS TicketSeller (
    ticket_seller_id VARCHAR(20) PRIMARY KEY,

```

```

    is_gambler BOOLEAN, -- don't assume FALSE by default
    FOREIGN KEY (ticket_seller_id) REFERENCES Staff(national_id)
);

CREATE TABLE IF NOT EXISTS Ticket (
    ticket_id INT PRIMARY KEY AUTO_INCREMENT,
    selling_date DATE,
    price DECIMAL(7,2),
    is_vip_ticket BOOLEAN DEFAULT FALSE,
    seat_number INT UNSIGNED,
    zone_code VARCHAR(20),
    arena_name VARCHAR(50),
    ticket_seller VARCHAR(20),
    match_id INT,
    FOREIGN KEY (seat_number, zone_code, arena_name)
    REFERENCES Seat(seat_number, zone_code, arena_name),
    FOREIGN KEY (ticket_seller) REFERENCES TicketSeller(ticket_seller_id),
    FOREIGN KEY (match_id) REFERENCES `Match`(match_id)
);

```