

NBA Database - Project2

Database Account: yh3290

Web Front-End App URL: http://34.139.184.252:8111/

Team Members: Jerry Liu (jl6007), Yong Hao (yh3290)

Description

1. Create composite types.

Add **Full_Name** as a new composite type in *Player* table to replace the old **First_Name** and **Last_Name**. We use the composite type to represent the name attribute, so users can either treat first and last name as a whole or separately.

2. Add arrays.

Add **Career_Stats** as an array type in *Player* table to replace the old **Career_PTS**, **Career_AST** and **Career_REB**. The array type provides extendibility to player's career stats, which means we can easily populate the *Player* table with more stats attributes like *Block* or *Steal* in the future.

3. Add triggers.

Add trigger functions **if_roll_back()**, **if_roll_back_game()** and triggers **check_player_del**, **check_plays_del**. The **check_player_del** trigger enforces the ≥ 1 participation constraint of *Team* by asserting that every team has at least one player. If one tries to delete all players in a certain team, the command will stop before the last selected player's record.

Unlike Team-Player, we have *Plays* to show the relationship between *Player* and *Game*. Therefore, we add a trigger on *Plays*, which helps assert that there is at least one player playing in a game to enforce the ≥ 1 participation constraint of the *Game* table.

Schema Modification Commands

Please note that we have created a new table called *New_Player* to replace the old *Player* table, to show the composite types and arrays features.

```
CREATE TYPE Full_Name AS (  
  First_Name VARCHAR(30),  
  Last_Name VARCHAR(30)  
);
```

```
CREATE TABLE New_Player (  
  Player_ID INTEGER,  
  Name Full_Name,  
  Date_Birth DATE,  
  Height INTEGER,  
  Weight INTEGER,  
  Jersey INTEGER,  
  Position CHARACTER(30),  
  Team_ID INTEGER,  
  CAREER_STATS DECIMAL [],  
  ALL_STAR_NUM INTEGER,  
  PRIMARY KEY (Player_ID),  
  FOREIGN KEY (Team_ID) REFERENCES Team  
);
```

Triggers:

```
CREATE TRIGGER check_player_del  
BEFORE DELETE ON new_player  
FOR EACH ROW  
EXECUTE FUNCTION if_roll_back();  
  
CREATE OR REPLACE FUNCTION if_roll_back() RETURNS TRIGGER AS $example_table$  
  BEGIN  
    IF (SELECT COUNT(*) FROM new_player WHERE team_id = old.team_id) <= 1  
  THEN  
    RETURN NULL;  
    ELSE  
    RETURN OLD;  
    END IF;  
  END;  
$example_table$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER check_plays_del
BEFORE DELETE ON plays
FOR EACH ROW
EXECUTE FUNCTION if_roll_back_game();

CREATE OR REPLACE FUNCTION if_roll_back_game() RETURNS TRIGGER AS
$example_table_game$
BEGIN
    IF (SELECT COUNT(*) FROM plays WHERE game_id = old.game_id) <= 1 THEN
        RETURN NULL;
    ELSE
        RETURN OLD;
    END IF;
END;
$example_table_game$ LANGUAGE plpgsql;
```

Queries

1.

```
SELECT COUNT(*) FROM New_Player  
WHERE (name).First_Name = 'John';
```

Sometimes, people may want to know how many players in the NBA have the same first name with them. The above query gives such results precisely. Here, it will return the number of players whose first name is John.

2.

```
SELECT * FROM New_Player  
WHERE Career_Stats[1] >= 20 AND Career_Stats[2] >= 5 AND Career_Stats[3] >= 5;
```

A player who has a 20(PTS)+5(AST)+5(REB) stats is considered all-around. Here, the query returns all all-around players in the history of NBA. In the future, we may add Block and Steal to better formulate our radar chart algorithm, and we may make the "all-around" standard as 20+5+5+1(BLK)+1(STL).

3.

```
DELETE FROM New_Player  
WHERE team_id = 1610612738;
```

This query tries to delete all players of the team Celtics. Since the trigger we set ensures at least one player for each team, there is still going to be one Celtics player left in the returned *New_Player* table.

(Since we have run this query during testing, there is only one Celtics player record left in *New_Player* now. Therefore, for the second (and future) runs, the query will return "DELETE 0", meaning that no record has been deleted. This corresponds to our implementation. To test the deletion with more than one records, use other teams, e.g., team_id = 1610612737 or team_id = 1610612739)