

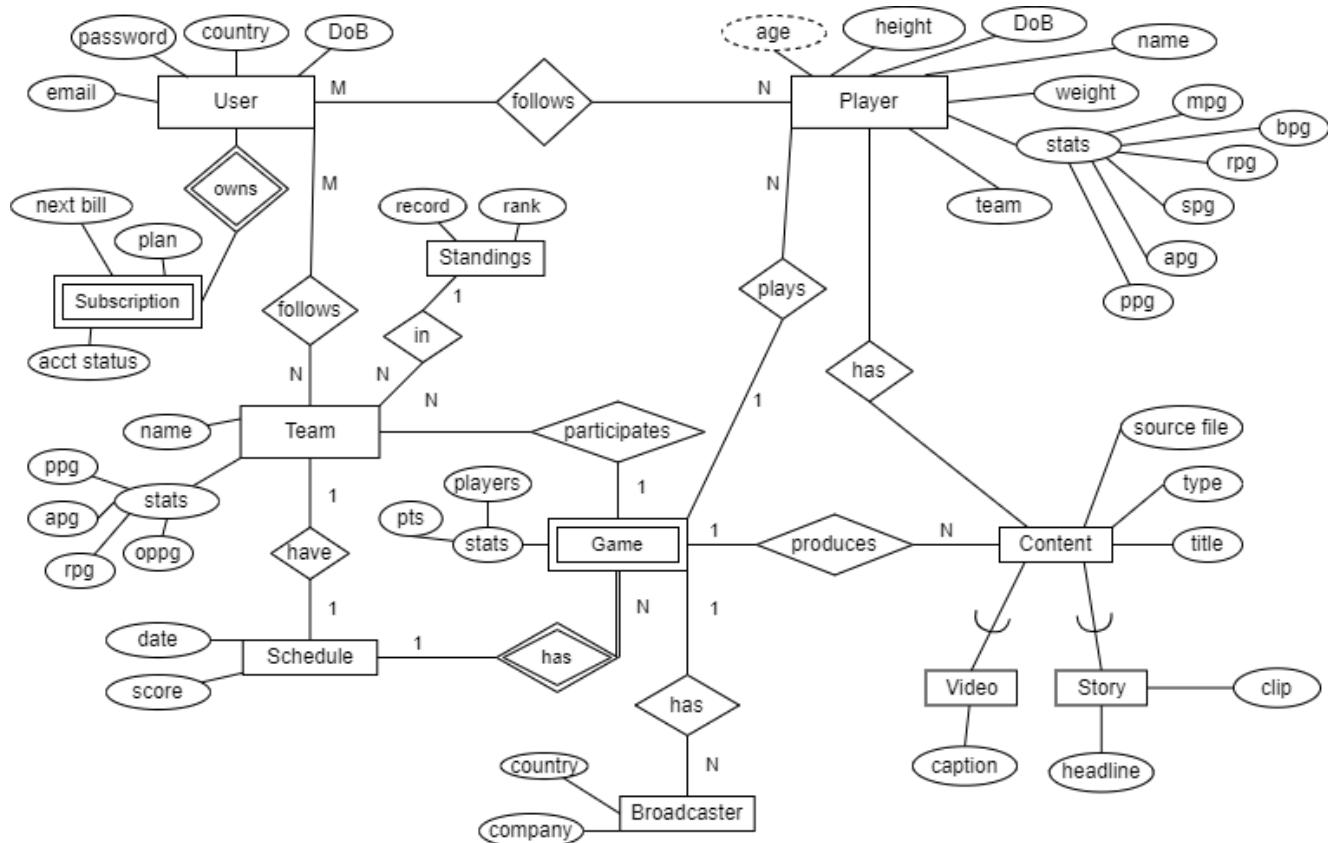
NBA League Pass Application: Project Phase III

Ethan Jean, Josh Kim, Justin Yeh

1. Problem Statement

- What: Our database application is an NBA league Pass: a bootstrap mobile app for watching live NBA games, following league news, and tracking league statistics. Ultimately, it is an all-in-one app that grants access to anything a fan would need to follow the NBA.
- Why: This application can be used to live stream NBA games, check game schedules, keep up with the latest NBA news, and track league standings, alongside players' statistics.

2. Conceptual Database Design



3. Logical Database Design:

USER (PK:user_id, email, DoB, password, country, FK:subscription_id)

USER		
Attribute	Type	Description
user_id PK	integer	unique identifier for each user
first_name	string	First name of the user
last_name	string	Last name of the user
email	string	email address of the user
DoB	string	date of birth of the user (MM/DD/YYYY)
password	string	password of the user
country	string	country of the user

SUBSCRIPTION (PK:subscription_id, FK: user_id, plan, account_status, next_bill)

SUBSCRIPTION		
Attribute	Type	Description
subscription_id PK	integer	unique identifier for each subscription
user_id FK	integer	references USER.user_id
plan	string	subscription plan
account_status	string	status of the user's account
next_bill	string	date of the next bill (MM/DD/YYYY)

TEAM (team_id, name, ppg, rpg, apg, oppg, news)

TEAM		
Attribute	Type	Description
team_id PK	integer	unique identifier for each team
name	string	name of the team
ppg	integer	points per game of the team
rpg	integer	rebounds per game of the tea
apg	integer	assists per game of the team
oppg	integer	opposing points per game of the team

SCHEDULE (schedule_id, team_id, date)

SCHEDULE		
Attribute	Type	Description
schedule_id PK	integer	unique identifier for each schedule
team_id FK	integer	references TEAM.team_id
date	string	date of the scheduled game (MM/DD/YYYY)

GAME (game_id, team1_id, team2_id, score_team1, score_team2, broadcaster_id)

GAME		
Attribute	Type	Description
game_id PK	integer	unique identifier for each game
team1_id FK	integer	references TEAM.team_id
team2_id FK	integer	references TEAM.team_id
score	integer	score of the game
broadcaster_id FK	integer	references BROADCASTER.broadcast_id

PERFORMANCE (player_id, game_id, minutes, pts, reb, ast, stl, blk)

PERFORMANCE		
Attribute	Type	Description
player_id PK & FK	integer	references PLAYER.player_id
game_id FK	integer	references GAME.game_id
minutes	integer	minutes played
pts	integer	points obtained
reb	integer	rebounds obtained
ast	integer	assists obtained
stl	integer	steals obtained

blk	integer	blocks obtained
-----	---------	-----------------

PLAYER (player_id, name, team_id, number, position, age, DoB, height, weight, mpg, ppg, rpg, apg, bpg, spg)

PLAYER		
Attribute	Type	Description
player_id PK	integer	unique identifier for each player
name	string	name of the player
team_id FK	integer	references TEAM.team_id
age	integer	player's age
DoB	string	player's date of birth (MM-DD-YYYY)
height	string	player's height (ft'inch)
weight	integer	player's weight (lb)
mpg	integer	minutes per game of the player
ppg	integer	points per game of the player
rpg	integer	rebounds per game of the player
apg	integer	assists per game of the player
bpg	integer	blocks per game of the player
spg	integer	steals per game of the player

BROADCASTER (PK: broadcaster_id, brand, country, FK: game_id)

BROADCASTER		
Attribute	Type	Description
broadcaster_id PK	integer	unique identifier for each broadcaster
brand	string	brand name of the broadcaster
country	string	the country where the broadcaster is located
game_id FK	integer	references GAME.game_id

STANDINGS (PK: type, PK: rank, record, FK: team_id)

STANDINGS		
Attribute	Type	Description
type PK	string	type of the standings (conference, league, division)
rank PK	integer	the rank of the team in the standings
record	string	the W-L record of the team
team_id FK	integer	the unique identifier for the team associated with these standings

CONTENT (PK: content_id, PK: source_file_hash, title, type)

CONTENT

Attribute	Type	Description
content_id PK	integer	unique identifier for each content
source_file_hash PK	integer	the file path or URL for the content file
title	string	title of the content
type	string	type of the content (video, image, audio, etc.)

STORY (PK: content_id, PK: source_file_hash, headlines, FK: game_id)

STORY		
Attribute	Type	Description
content_id PK	integer	Unique identifier for each record in the table
source_file_hash PK	string	The file path or URL for the video file
headlines	string	Headline of the story
game_id FK	integer	References GAME.game_id

VIDEO (PK: content_id, PK: source_file_hash, FK: player_id)

VIDEO		
Attribute	Type	Description
content_id PK	integer	unique identifier for each content
source_file_hash PK	string	the file path or URL for the video file
title	string	title of the video
type	string	type of the video (game highlights, player interview, etc.)
player_id FK	integer	the ID of the player that the video belongs to

Own (PK:USER, PK:SUBSCRIPTION)

Attribute	Type	Description
user_id PK	integer	unique identifier for each user
subscription_id PK	integer	unique identifier for each subscription

Follow_P (PK: USER, PK: PLAYER)

Attribute	Type	Description
user_id PK	integer	unique identifier for each user
player_id PK	integer	unique identifier for each player

Follow_T (PK:USER, PK: TEAM)

Attribute	Type	Description
user_id PK	integer	unique identifier for each user
team_id PK	integer	unique identifier for each team

Store (PK: GAME, PK: PERFORMANCE)

Attribute	Type	Description
game_id PK	integer	unique identifier for each game
player_id PK	integer	unique identifier for each player

Play (PK: PLAYER, PK: GAME)

Attribute	Type	Description
game_id PK	integer	unique identifier for each game
player_id PK	integer	unique identifier for each player

Participate (PK: TEAM, PK: GAME)

Attribute	Type	Description
game_id PK	integer	unique identifier for each game
team_id PK	integer	unique identifier for each team

Have (PK: TEAM, PK: SCHEDULE)

Attribute	Type	Description
schedule_id PK	integer	unique identifier for each schedule
team_id PK	integer	unique identifier for each team

Has_1 (PK: SCHEDULE, PK: GAME)

Attribute	Type	Description
schedule_id PK	integer	unique identifier for each schedule
game_id PK	integer	unique identifier for each game

Has_2 (PK: GAME, PK: BROADCASTER)

Attribute	Type	Description
broadcaster_id PK	integer	unique identifier for each broadcaster
game_id PK	integer	unique identifier for each game

Has_3 (PK: PLAYER, PK: CONTENT)

Attribute	Type	Description
content_id PK	integer	unique identifier for each broadcaster
source_file_hash PK	integer	the file path or URL for the content file
player_id PK	integer	unique identifier for each player

Produce (PK: GAME, PK: CONTENT)

Attribute	Type	Description
content_id PK	integer	unique identifier for each broadcaster
source_file_hash PK	integer	the file path or URL for the content file
game_id PK	integer	unique identifier for each game

4. Application Program Design

Function 1: Cancel_sub

//This function is to cancel a subscription for a user. It accesses “USER” table, “SUBSCRIPTION”, and “OWN” relationship

Input: user_id

Steps:

- (1) Check the user_id in the “OWN” relationship
- (2) Access the corresponding subscription_id for the user in the “SUBSCRIPTION” table

- (3) In the “SUBSCRIPTION” table, modify plan, account_status, and next_bill attributes to null
- (4) Display notification to user that plan is canceled

Function 2: Follow_team

//This function is to add specific teams to a user's following list. It accesses “Follow_T” and “TEAM” tables.

Input: team_id, user_id

Steps:

- (1) Check the user_id in “Follow_T”
- (2) Access corresponding team_id to user_id
- (3) If team_id is NULL, insert user-inputted team_id
- (4) If team_id exists, check if user-inputted team_id matches, if so, output “Team is already followed”. If team_id is not the same as newly user-inputted team id, modify the existing attribute to include the initial team_id value with the additional team_id's.
- (5) Display notification “You are now following this team”.

Function 3: Watch_team

//This function allows users to watch videos associated with a team. It accesses “TEAM”, “PLAYER”, and “VIDEO” tables.

Input: team_id

Steps:

- (1) Join “TEAM” and “PLAYER” tables on matching team_id in order to get all players in the user inputted team
- (2) Join this table and “VIDEO” on matching player_id to get all videos of players in the team
- (3) Display videos in a feed to user

Function 4: Best_stat

//This function allows users to check the best performing player in any individual player statistic. It accesses the “PLAYER” table.

Input: Any Individual Statistic \$Stat in Domain D:{mpg, ppg, rpg, apg, bpg, spg}

Steps:

- (1) Access the “PLAYER” table and use aggregate function MAX on \$Stat
- (2) Display the player name and his \$Stat

Function 5: Check_caster

//This function is to check the broadcasters for a certain game from a video. It accesses “BROADCASTER”, “STORY”, and “VIDEO” tables.

Input: content_id

Steps:

- (1) Access “VIDEO” table and join with “STORY” on content_id
- (2) Join “BROADCASTER” with the previously joined table on game_id
- (3) Display broadcaster_id, brand, country, and future game_id's

Function 6: Best_player

//This function displays the best offensive and defensive player in a game. It accesses “PERFORMANCE” table and “PLAYER” table

Input: game_id

Steps:

- (1) Access tuples in "PERFORMANCE" table with matching user-inputted game_id
- (2) For each tuple, SUM its pts, ast, reb and store this as offensive output to the tuple
- (3) For each tuple, SUM its stl, blk, reb and store this as defensive output to the tuple
- (4) Use aggregate function MAX to find maximum of offensive and defensive output and player_id of the two tuples
- (5) JOIN with "PLAYER" table on matching player_id
- (6) Display player names of whether they are offensive or defensive player of the game

Function 7: Build_feed

//This function builds a personalized news feed for players and teams an user follows. It accesses "GAME", "STORY", "VIDEO", "FOLLOW_P", and "FOLLOW_T" tables.

Input: user_id

Steps:

- (1) Select user_id for "FOLLOW_P" and "FOLLOW_T"
- (2) Join "FOLLOW_P" with VIDEO by player_id
- (3) Join "GAME" with "STORY" by game_id
- (4) Join "FOLLOW_T" with the table from (3) by team_id
- (5) Select title, type, player_id for table from (2) and headlines and game_id for table from (4)
- (6) Display personalized stories with headlines and game for all followed teams
- (7) Display personalized highlights with video title, type, and players associated for all followed players

Insert

This function inserts a new user into the "USERS" table.

Steps:

1. First, it selects the maximum value of the "user_id" column from the "USERS" table and stores it in the "@last_userid" variable using the "MAX()" function and "INTO" keyword.
2. Then, it sets the values for the "@new_user_fname", "@new_user_lname", "@new_user_email", "@new_user_DOB", and "@new_user_passwd" variables.
3. After that, it inserts a new row into the "USERS" table with the values from the variables. The "@last_userid + 1" expression generates a new user_id that is one greater than the maximum user_id in the table.

```
SELECT MAX(user_id) INTO @last_userid FROM USERS;
SET @new_user_fname = 'John';
SET @new_user_lname = 'Doe';
SET @new_user_email = 'johndoe@example.com';
SET @new_user_DOB = '1990-01-01';
SET @new_user_passwd = 'youcantseeme';
INSERT INTO USERS (user_id, first_name, last_name, email, DOB, password)
VALUES
(@last_userid + 1, @new_user_fname, @new_user_lname, @new_user_email,
```

```
@new_user_DOB, @new_user_passwd);
```

Delete

This function deletes all records from the "SUBSCRIPTIONS" table where the value of "user_id" column is equal to 3.

```
DELETE FROM SUBSCRIPTIONS WHERE user_id = 3;
```

Modify

This function updates the "plan" and "account_status" columns in the "SUBSCRIPTIONS" table for the row where the "user_id" is 2. Specifically, it will set the "plan" column to 'Premium' and the "account_status" column to 'Active'.

```
-- Intent to modify a subscription plan for a user
UPDATE SUBSCRIPTIONS
SET plan = 'Premium', account_status = 'Active'
WHERE user_id = 2;
```

One SQL query that is on a single table.

```
SELECT * FROM SUBSCRIPTIONS;
```

Query that requires joining multiple tables

This query selects distinct player IDs and names of players who scored at least 10 points, grabbed at least 5 rebounds, and dished out at least 5 assists in games that were broadcasted on ESPN. The query retrieves data from three tables, namely the PERFORMANCE table, GAMES table, and PLAYERS table.

- (1) Join the PERFORMANCE, GAMES, and PLAYERS tables.
- (2) Filter the games that were broadcasted on ESPN and select only the players who scored at least 10 points, grabbed at least 5 rebounds, and dished out at least 5 assists in those games.

```
-- Find the 10-5-5 players that had played in games broadcasted by ESPN

SELECT DISTINCT pf.player_id, p.name
FROM PERFORMANCE pf, GAMES g, PLAYERS p
WHERE (pf.game_id = g.game_id AND
      pf.pts >= 10 AND
```

```
pf.reb >= 5 AND
pf.ast >= 5 AND
pf.player_id = p.player_id
AND g.broadcast_id = (SELECT broadcast_id
                      FROM BROADCASTERS
                      WHERE brand = 'ESPN' );
```

Retrieves the name of each player along with their average points, rebounds, and assists across all games they have played

```
-- Find all players recent performances

SELECT
p.name,
ROUND(AVG(pf.pts), 1) AS avg_pts,
ROUND(AVG(pf.reb), 1) AS avg_reb,
ROUND(AVG(pf.ast), 1) AS avg_ast
FROM PERFORMANCES pf, PLAYERS p
WHERE pf.player_id = p.player_id
GROUP BY pf.player_id;
```

Query that uses one or more aggregation functions

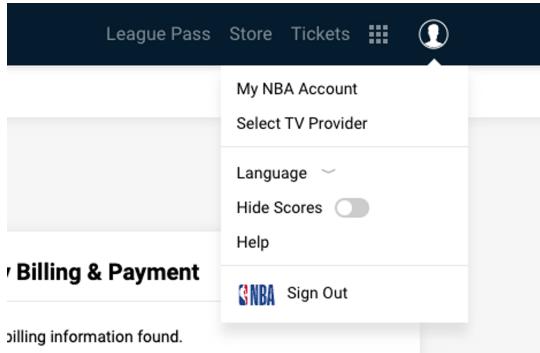
Selects the name and points per game of the player with the highest ppg score in the PLAYERS table.

```
SELECT name, ppg
FROM PLAYERS
WHERE ppg = (
    SELECT MAX(ppg)
    FROM PLAYERS
);
```

5. User Interface Design

Function 1: Cancel_sub

Click on My NBA Account and select 'Cancel Subscription' upon which a confirmation will pop up. Confirm that you wish to cancel the subscription.



Function 2: Follow_team

Out of a list of all NBA teams, click the button on the right to follow the team you desire.



FOLLOW YOUR FAVORITE TEAMS

Get news, game updates, highlights and more info on your favorite teams

-  Atlanta Hawks Follow
-  Boston Celtics Follow
-  Brooklyn Nets Follow
-  Charlotte Hornets Follow
-  Chicago Bulls Follow
-  Cleveland Cavaliers Follow
-  Dallas Mavericks Follow

Function 3: Watch_team

Under the 'Watch' section in the app, scroll through the Live and Upcoming streams to watch a broadcast of a specific team.

LIVE & UPCOMING STREAMS



Apr 17 - 20:00
Warriors Pregame Show



Apr 17 - 23:30
Kings Postgame Show

Function 4: Best_stat

In the Players section of the app, locate the league leaders for each type of statistic by changing the 'Stat Category' option and clicking 'Get Stats'.

Players ▾ Official Leaders ▾

SEASON

2022-23

SEASON TYPE

Regular Season

PER MODE

Per Game

STAT CATEGORY

PTS

Player Bio

SCOPE

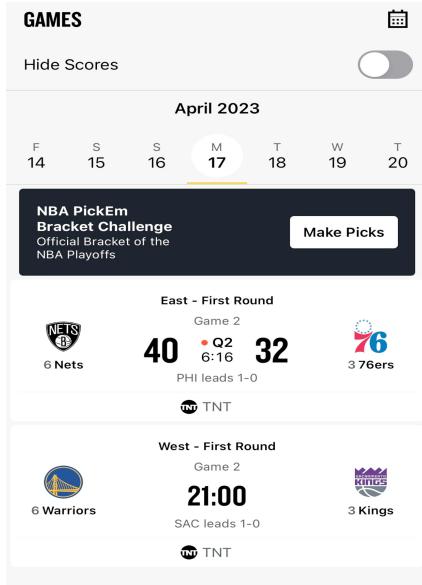
All Players

Reset Filters **Get Stats**

[Advanced Filters ▾](#)

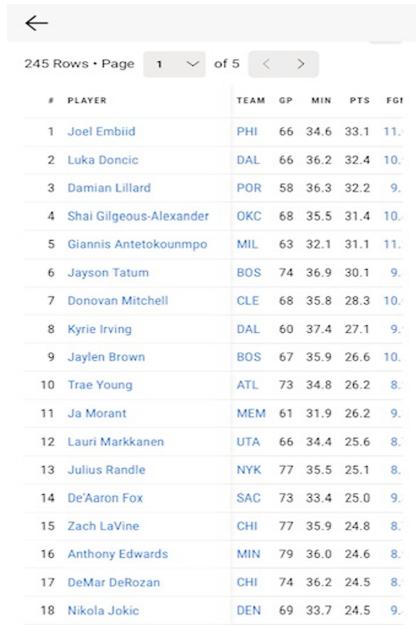
Function 5: Check_caster

Under the Games section of the app, either click the game itself or look at the icon below each game to see the broadcaster.



Function 6: Best_player

Under the Players tab, select the option to view ‘Best Player’ and click ‘Get Stats’ similar to what we did above, to access these values.



#	PLAYER	TEAM	GP	MIN	PTS	FG%
1	Joel Embiid	PHI	66	34.6	33.1	11.
2	Luka Doncic	DAL	66	36.2	32.4	10.
3	Damian Lillard	POR	58	36.3	32.2	9.
4	Shai Gilgeous-Alexander	OKC	68	35.5	31.4	10.
5	Giannis Antetokounmpo	MIL	63	32.1	31.1	11.
6	Jayson Tatum	BOS	74	36.9	30.1	9.
7	Donovan Mitchell	CLE	68	35.8	28.3	10.
8	Kyrie Irving	DAL	60	37.4	27.1	9.
9	Jaylen Brown	BOS	67	35.9	26.6	10.
10	Trae Young	ATL	73	34.8	26.2	8.
11	Ja Morant	MEM	61	31.9	26.2	9.
12	Lauri Markkanen	UTA	66	34.4	25.6	8.
13	Julius Randle	NYK	77	35.5	25.1	8.
14	De'Aaron Fox	SAC	73	33.4	25.0	9.
15	Zach LaVine	CHI	77	35.9	24.8	8.
16	Anthony Edwards	MIN	79	36.0	24.6	8.
17	DeMar DeRozan	CHI	74	36.2	24.5	8.
18	Nikola Jokic	DEN	69	33.7	24.5	9.

Function 7: Build_feed

In the ‘Home’ tab of the app, scroll through to see relevant games happening soon as well as content that is happening around the league. Personalized content will also populate the feed with videos and articles related to your favorited players and teams.

