

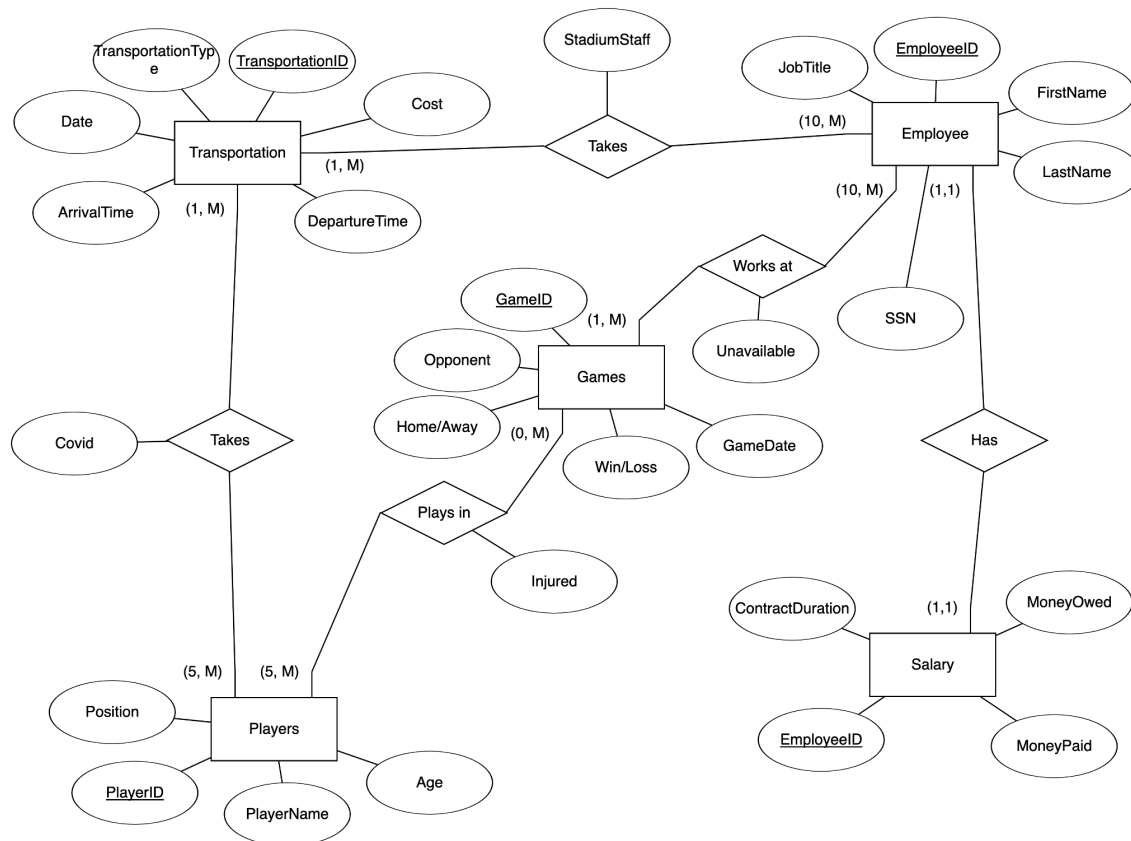
## Description:

It is an NBA team database to store player, employee, salary, and games played and won in one database to organize everything. This database application has nine entities with five relationships, only Employee to Salary is one to one, the rest of the relationships are many to many relationships.

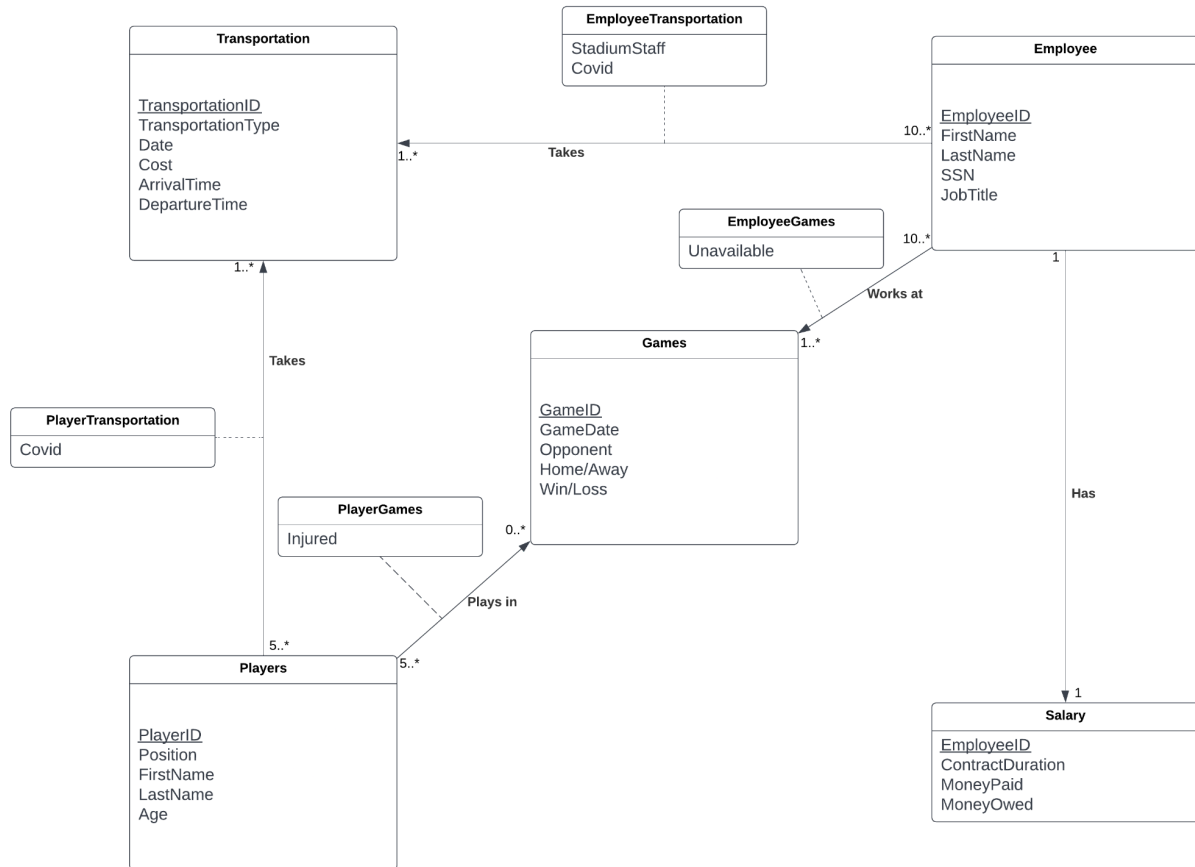
## Constraints:

- Every game needs at least 5 players to play per game.
- At least 5 players have to take transportation.
- At least 10 employees have to be at a game, this is mostly the coaching staff.
- At least 10 employees have to take transportation with players, this is for away games.
- Stadium staff, like a cashier or ticket vendor, cannot use transportation as they only work at the stadium.
- All employees have one salary.
- If someone has covid, they cannot take transportation.

## ERD:



UML:



Relational Schema:

Transportation(TransportationID, TransportationType, GameDate, Cost, ArrivalTime, DepartureTime)

Player(PlayerID, Position, PlayerName, Age)

Game(GameID, Opponent, HomeAway, WinLoss, GameDate)

Employee(EmployeeID, JobTitle, FirstName, LastName, SSN)

EmployeeTransportation(EmployeeID, TransportationID, StadiumStaff, Covid)

FK: EmployeeID references Employee

FK: TransportationID references Transportation

PlayerTransportation(PlayerID, TransportationID, Covid)

FK: PlayerID references Player

FK: TransportationID references Transportation

PlayerGame(PlayerID, GameID, Injured)

FK: PlayerID references Player

FK: GameID references Game

EmployeeGame(EmployeeID, GameID, Unavailable)

FK: EmployeeID references Employee

FK: GameID references Game

Salary(EmployeeID, ContractDuration, MoneyPaid, MoneyOwed)

FK: EmployeeID references Employee

Normalization:(All examples are using rough/shortened names for attributes and relations)  
As seen through the schema the data the database is in third normal form

First Normal Form: Every Attribute is a single valued attribute. I.e there are no attributes with values like gamesPlayed = BucksvsWarriors, BucksvsWizards

Second Normal Form: There is no partial dependency meaning that no non-key is dependent on another set.

Third Normal Form: There is no transitive dependency. This means for our data set to access other attributes there is only one way rather than being able to connect attributes through a transitive property i.e if Player  $\rightarrow$  Games and Games  $\rightarrow$  location then Player  $\rightarrow$  location is not a relation. Because this is a transitive relationship.

SQL code:

Use master;

Create database BasketballGame;

Use BasketballGame;

Drop table Player;

Create table Player (  
PlayerID INT NOT NULL,  
Position CHAR(2) NOT NULL,  
PlayerName CHAR(30) NOT NULL,  
Age NUMERIC(8) NOT NULL,  
PRIMARY KEY (PlayerID));

Insert into Player VALUES  
(1, 'PG', 'Kyrie Irving', 30),  
(2, 'SG', 'Bradley Beal', 28),  
(3, 'SF', 'Lebron James', 37),  
(4, 'PF', 'Kevin Durant', 33),  
(5, 'C', 'Nikola Jokic', 27);

Drop table Employee;

Create table Employee (  
EmployeeID INT NOT NULL,  
JobTitle CHAR(30) NOT NULL,  
FirstName CHAR(30) NOT NULL,  
LastName CHAR(30) NOT NULL,  
SSN CHAR(30) NOT NULL,  
PRIMARY KEY (EmployeeID));

Insert into Employee VALUES  
(1, 'Referee', 'Dave', 'Paul', '111-11-1111'),  
(2, 'Head Coach', 'Steve', 'Carrell', '222-22-2222'),  
(3, 'Assistant Coach', 'James', 'James', '333-33-3333'),  
(4, 'Water Boy', 'Logan', 'Fruit', '444-44-4444'),  
(5, 'Cashier', 'Nick', 'John', '555-55-5555'),

```
(6, 'Player', 'Kyrie', 'Irving', '666-66-6666'),  
(7, 'Player', 'Bradley', 'Beal', '777-77-7777'),  
(8, 'Player', 'Lebron', 'James', '888-88-8888'),  
(9, 'Player', 'Kevin', 'Durant', '999-99-9999'),  
(10, 'Player', 'Nikola', 'Jokic', '000-00-0000');
```

Drop table Game;

```
Create table Game (  
GameID INT NOT NULL,  
Opponent VARCHAR(50) NOT NULL,  
HomeAway CHAR(30) NOT NULL,  
WinLoss CHAR(30) NOT NULL,  
GameDate DATE NOT NULL,  
PRIMARY KEY (GameID));
```

```
Insert into Game VALUES  
(1, 'Kings', 'Home', 'Win', '1-11-22'),  
(2, 'Lakers', 'Away', 'Win', '2-22-22'),  
(3, 'Nets', 'Away', 'Win', '3-30-22'),  
(4, 'Nuggets', 'Home', 'Loss', '4-14-22'),  
(5, 'Bucks', 'Away', 'Loss', '5-15-22');
```

Drop table Transportation;

```
Create table Transportation (  
TransportationID INT NOT NULL,  
TransportationType VARCHAR(30) NOT NULL,  
GameDate DATE NOT NULL,  
Cost NUMERIC(8,2) NOT NULL,  
ArrivalTime CHAR(10) NOT NULL,  
DepartureTime CHAR(10) NOT NULL,  
PRIMARY KEY (TransportationID));
```

```
Insert into Transportation VALUES  
(1, 'Bus', '5-15-22', 339.99, '3:00pm', '6:00am'),  
(2, 'Car', '4-30-22', 450.99, '2:00am', '7:00pm'),  
(3, 'Plane', '2-22-22', 980.99, '7:00am', '3:00pm'),  
(4, 'Car', '9-25-22', 740.99, '9:00am', '5:00pm'),  
(5, 'Plane', '3-14-22', 570.99, '4:00pm', '2:00am');
```

Drop table EmployeeTransportation;

Create table EmployeeTransportation (  
EmployeeID INT NOT NULL,  
TransportationID INT NOT NULL,  
StadiumStaff BIT NOT NULL,  
Covid BIT NOT NULL,  
PRIMARY KEY (EmployeeID, TransportationID),  
FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),  
FOREIGN KEY (TransportationID) REFERENCES Transportation(TransportationID));

Insert into EmployeeTransportation VALUES

(1, 1, 0, 1),  
(2, 2, 0, 0),  
(3, 3, 0, 0),  
(4, 4, 0, 1),  
(5, 5, 1, 0);

Drop table PlayerTransportation;

Create table PlayerTransportation (  
PlayerID INT NOT NULL,  
TransportationID INT NOT NULL,  
Covid BIT NOT NULL,  
PRIMARY KEY (PlayerID, TransportationID),  
FOREIGN KEY (PlayerID) REFERENCES Player(PlayerID),  
FOREIGN KEY (TransportationID) REFERENCES Transportation(TransportationID));

Insert into PlayerTransportation VALUES

(1, 1, 0),  
(2, 2, 1),  
(3, 3, 0),  
(4, 4, 0),  
(5, 5, 1);

Drop table PlayerGame;

Create table PlayerGame (  
PlayerID INT NOT NULL,

```
GameID INT NOT NULL,  
Injured BIT NOT NULL,  
PRIMARY KEY (PlayerID, GameID),  
FOREIGN KEY (PlayerID) REFERENCES Player(PlayerID),  
FOREIGN KEY (GameID) REFERENCES Game(GameID));
```

Insert into PlayerGame VALUES

```
(1, 1, 1),  
(2, 2, 0),  
(3, 3, 0),  
(4, 4, 0),  
(5, 5, 1);
```

Drop table EmployeeGame;

```
Create table EmployeeGame (  
EmployeeID INT NOT NULL,  
GameID INT NOT NULL,  
Unavailable BIT NOT NULL,  
PRIMARY KEY (EmployeeID, GameID),  
FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),  
FOREIGN KEY (GameID) REFERENCES Game(GameID));
```

Insert into EmployeeGame VALUES

```
(1, 1, 0),  
(2, 2, 0),  
(3, 3, 0),  
(4, 4, 1),  
(5, 5, 1);
```

Drop table Salary;

```
Create table Salary (  
EmployeeID INT NOT NULL,  
ContractDuration CHAR(30) NOT NULL,  
MoneyPaid CHAR(30) NOT NULL,  
MoneyOwed CHAR(30) NOT NULL,  
PRIMARY KEY (EmployeeID),  
FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID));
```



Insert into Salary VALUES

(6, 'Year 2 of 5', '\$12,000,000', '\$50,000,000'),  
(7, 'Year 3 of 4', '\$48,000,000', '\$60,000,000'),  
(8, 'Year 1 of 2', '\$35,000,000', '\$70,000,000'),  
(9, 'Year 3 of 5', '\$60,000,000', '\$80,000,000'),  
(10, 'Year 4 of 4', '\$90,000,000', '\$90,000,000'),  
(1, 'Year 1 of 1', '\$130,000', '\$130,000'),  
(2, 'Year 2 of 3', '\$7,000,000', '\$10,000,000'),  
(3, 'Year 1 of 2', '\$750,000', '\$2,000,000'),  
(4, 'Year 1 of 1', '\$50,000', '\$50,000'),  
(5, 'Year 1 of 1', '\$55,000', '\$55,000');

Queries:

1)

```
Select GameDate, ArrivalTime, DepartureTime, TransportationType
From Transportation
Where TransportationType = 'Plane';
```

2)

```
Select Opponent, GameDate
From Game
Where GameDate = '4-14-22';
```

3)

```
Select FirstName, LastName, JobTitle, ContractDuration
From Employee E
Join Salary S
On E.EmployeeID = S.EmployeeID
Where ContractDuration = 'Year 1 of 1';
```

4)

```
Select E.FirstName, E.LastName, ET.Covid, P.PlayerName, PT.Covid
From Employee E
Join EmployeeTransportation ET
On E.EmployeeID = ET.EmployeeID
Join Transportation T
On ET.TransportationID = T.TransportationID
Join PlayerTransportation PT
On T.TransportationID = PT.TransportationID
Join Player P
On PT.PlayerID = P.PlayerID
Where ET.Covid = 1 OR PT.Covid = 1;
```

## Different Uses

There are many different ways this database application can be used:

- By members of the sports organization for information:
  - A player could use this application to check Covid clearance or what their salary was or currently is
  - A staff member, such as a coach, could see past games locations and who on the coaching staff was there and what role they played

By team leaders for analytics:

- Trainers could track players injuries over time and see if certain locations or teams are leading to injuries.
- Coaches can track how many games and what games certain players played in, very helpful for formatting the team for the best outcome say they play a team twice.

## Pros

Learning real world applications with databases

Grasp on the database design process from start to finish

How to troubleshoot database problems

## Cons

Tedious to type in data

Only one of us had access to the DBMS at a time

## Conclusion

Overall this project educated us on ways to apply skills taught in class to potential real world applications. While challenging at times, especially working with everyone's busy schedules, the team work aspect was a great opportunity to practice working in a team. This project also shed light on the design process of a database application. We saw first hand how it was assembled from idea to final product.