NBA Team Management - Database Specification

GROUP7 Zhongyu Shi, Xiaoyu Wang, Guodong Shi

Database Purpose:

The purpose of the database is to maintain data, generate information, as the foundation of team management. It will be used by team management staff only.

Business Problem Addressed:

- Allow team management team staff, coach team staff to generate descriptive reports
- Provide insight through the analysis of physical data, training and game player played, helping coach team to customize and optimize training plan for players to enhance their performance (e.g., through the analysis Kevin is a PF, good at offense yet short at defense, then his training plan will be focus on strength and agility, etc.,)
- Provide insight through the analysis of game player played, team profit and cost, sponsor, helping management team to pursue the most profitable operating portfolio (e.g., there are three SG in the team A while we only need two, through the analysis, we find two of them are sponsored and can bring money to team, so we decide to trade the rest one, etc.,)
- Provide insight through the analysis of physical data and game player played, helping management team to observe the current situation and development of a certain player. (e.g. from a series of data related to a certain player's behavior in games, like his 3pt%, total rebound, etc., and if comparing to the changes of his physical data, we can draw a brief conclusion of his performance and future growth.)

Business Rules:

- Each team may have one or more managers
- Each team may have one or more coaches
- Each team may have one or more players
- Each team may have one or more managers
- Each coach may have one or more trainings
- Each player may have one or more trainings
- Each player may have one or more physical data
- Each sponsor may have one or more sponsored teams
- Each sponsor may have one or more sponsored players
- Each game may have one or more teams
- Each game may have one or more players
- Since coach trains players, so it's identifying relationship between coach and training.
- Since player completes training, so it's identifying relationship between player and training.

- Since PlayerID is FK but not the PK in PhysicalData, so it's non-identifying relationship between player and physical data.
- Since game and team have Many-to-Many relationship, so it's identifying relationship between game and team.
- Since player and team have Many-to-Many relationship, so it's identifying relationship between game and player.
- Since sponsor and team have Many-to-Many relationship, so it's identifying relationship between sponsor and team.
- Since sponsor and player have Many-to-Many relationship, so it's identifying relationship between sponsor and player.

Design Requirements (Credit to Professor Simon Wang):

- Use Crow's Foot Notation
- Specify the primary keys
- Draw relationship between different entities
- Correct in identifying/ non-identifying relationship representation
- All entities are connected, and no unnecessary relationship
- Correct cardinality

Design Decisions

Entity Name	Why Entity Included	How Entity is Related to Other Entities
Team	Team entity is an important one for it contains essential data related to many different entities, and when we try to do some data analysis, we need to know the information like city, region and stadium of a certain team.	As one of the core entity in our database, the PK(TeamID) of Team entity, is related to 5 entities: Player, Game, Manager, Coach and Sponsor entity. As there are 2 kinds of Many-to-Many relationships with these entities, we are able to find out corresponding associative entities: Game Team Played and Sponsor To Team.
Player	Player entity contains basic data of an NBA player, like player name, college, salary, Debut, and other attributes. When combining Player entity with other entities, we can gain many useful information which will then assist our profound analysis.	The PK(PlayerID) of Player entity, is related to 5 entities: Team, Game, Sponsor, Physical Data and Training entity. As there are 2 kinds of Many-to-Many relationships with these entities, we are able to find out corresponding associative entities: Game Player Played and Sponsor To Player.

Manager	Manager entity contains general data of a management team, like the team they managed, position, and other attributes.	The PK(ManagerID) of Manager entity is associated to Team entity. When analyzing the current situation and development of a certain team, we can do it from Manager view.
Coach	Coach entity contains simple information of a coach team, like the specification they focused, position, as well as the team they trained.	The PK(CoachID) of Coach entity is related to Team entity and Training entity. When considering whether the coach is well-fitted with a certain team, or whether the coach has great impact on a player's growth, we can come to a conclusion from those relationships.
Sponsor	Sponsor entity contains basic data of a sponsor, like sponsor's company, the method they offered, total amount, sponsor history(time) and other attributes.	Sponsor entity and Team entity, Player entity as well, both have Many-to-Many relationships with each other. So we find out corresponding associative entities: Sponsor To Team and Sponsor To Player.
Game	One of the primary purpose of our NBA database is to collect information about game factors related to teams and players. Game entity contains essential data like date, season, stadium, the home team and visiting team played in a game, and what's more, what scores they gained in the game. Those data are valuable for we can analyze multiple aspects of a team or a player.	The PK(GameID) of Game entity, is related to Team entity and Player entity. Since both of them have Many-to-Many relationships with Game entity, we are able to find out the corresponding associative entities: Game Team Played and Game Player Played. From those relationships, insight may be gained about these factors in assessing a certain team or player.

GameTeam Played	we draw it as an associative entity due to the Many-to-Many relationship between Game and Team entity, the attributes there include general fields to valuate a team performed in a game, like the team steal, team block, 3-points made, attempt and percentage, total rebound and off-rebound, free throw made, attempt and percentage, total scores a team gained and so forth.	There is a Many-to-Many relationship between Game entity and Team entity, therefore, GameTeamPlayed is an associative entity.
GamePlayer Played	we draw it as an associative entity due to the Many-to-Many relationship between Game and Player entity, the attributes there include general fields to valuate a player performed in a game, like the player steal, block, 3-points made, attempt and percentage, free throw made, attempt and percentage, rebound and off-rebound, individual score and so forth.	There is a Many-to-Many relationship between Game entity and Player entity, therefore, GamePlayerPlayed is an associative entity.
SponsorTo Team	we draw it as an associative entity due to the Many-to-Many relationship between Sponsor and Team entity, the attributes there include the amount of money they invested to the team, the sponsor history(time) to the team, and the method they offered to the team.	There is a Many-to-Many relationship between Sponsor entity and Team entity, therefore, SponsorToTeam is an associative entity.
SponsorTo Player	we draw it as an associative entity due to the Many-to-Many relationship between Sponsor and Player entity, the attributes there include the amount of money they invested to the player, the sponsor history(time), and the method they offered to the team	There is a Many-to-Many relationship between Sponsor entity and Player entity, therefore, GamePlayerPlayed is an associative entity.

Training	Training entity contains basic data of Training, like training items, amount of training, Training data, training start time, training end time. When combining Training entity with Coach entity and Player entity, we can gain many useful information to evaluate the coach or player.	The PK(TrainingID) of Training entity is related to Player entity and Coach entity. When considering whether the coach is well-fitted with a certain team, or whether the coach has great impact on a player's growth, we can come to a conclusion from those relationships.
PhysicalData	PhysicalDate entity contains basic data of player's physical data, like height, weight, agility, jump, speed, strength, stamina and so forth. When combining PhysicalData entity with Player entity, it is easy for us to know the physical data about a player.	The PK(PhysicalDataID) of PhysicalData entity is related to Player entity. There is a one-one relationship between PhysicalData entity and Player entity, one player only has one physical data, one physical data is only for one player.