Group 82
Project Step 2 Draft

Team Members: William O'Brien, Deptmer Ashley
Project Title: Youth Basketball League Schedule Management System

a) Fixes based on Feedback from Step 1:

The below outlines only the feedback that suggested changes be made in Step 1. This does not include the full verbatim reviews, as that adds too much text and filler. The individual suggested changes are instead pulled out:

From the TAs:

- "Some general numbers and entity information is stated, but it is not entirely clear what problem the DB is solving. Why does the sports league need a DB? The point of this overview is to paint a picture for your audience about what gave rise to your DB and how your DB actually solves the problem."

Ed Discussion Review 1:

- I feel that a little more information is required in terms of the context of the situation. For instance, I was wondering whether this tournament was solely for one age bracket/level of play or if there are multiple brackets for different ages/skill levels.
- The M:M relationship between Teams and Games is a bit odd as each Game can really only have two teams. I understand the requirement for a M:M relationship, but I would instead picture it as an intersection table between two teams (if that makes sense).
- The 1:M relationship between Teams and Coaches is also a bit odd to me as I would imagine it to be a 1:1 relationship if there are only five teams. A workaround to this would be to have multiple brackets (whether that be by age or skill level) as I find it more reasonable for a coach to have teams in different brackets.
- Something small I noticed that was probably a typo is in the Games entity, there is an attribute called "Gamescol" which I assume was accidentally generated when making the diagram.

Ed Discussion Review 2:

- No suggested feedback

Ed Discussion Review 3:

- I think that your overview is a good start but is lacking some finer details. I think that the overview should be more clear on the problem that the project will solve. For example, this project could help the league with record keeping, saving lots of time. Additionally, more specific facts could help sell the project even further and allows readers to grasp the scale of the project.

Ed Discussion Review 4

- The overview states what the web app will be, but doesn't exactly state the problem (e.g. "Basketball leagues need a tool that's more reliable than paper to keep track of games")
- The overview lists facts about the web app itself, but not about the problem that it's solving. Yes, there are 6 entities so far, and they each represent a single, clear idea.

- Yes, most of the data is present in the outline, although I suggest renaming “Teams_has_Games” to "TeamGames", since it should represent an entity instead of a fact.
- There is mostly consistency, although the FKs in the Overview should have their full name, e.g. "coachID" instead of "coach". The "Teams_has_Games" table is also missing in the Outline. There was a "Gamescol" column in the ERD, and it's not clear what that represents or why it's capitalized

Actions taken based on feedback in Step 1:
The first action we took was providing more detail in the overview section of the submission. The TAs made a good point that we did not provide details as to what problem our database is solving. We added additional information regarding the need for scheduling and roster creation in a basketball league. This was helpful feedback that required a change.

The Ed discussion Review 1 provided two suggestions. The first was changing the M:M relationship for teams and games. We chose not to change this in our submission because we feel that we chose the correct relationship for this database. As a game has multiple teams and a team plays in multiple games, an M:M relationship felt like the correct decision. Using this relationship, the database can easily be queried to display the games played for an individual team or player, thus showing the necessary schedule a family may need. The second piece of feedback in Review 1 was to change the coach to player relationship. We currently have the relationship set up so that a coach can be a coach for multiple teams. This was a deliberate decision as, based on personal experience, creating teams is easy but finding sufficient coaches is not easy. This was a coach can be used to coach for multiple teams if required. For example, if only 3 coaches sign up for a league with 5 teams, then two coaches will need to coach 2 teams. Lastly, the first review correctly pointed out that the Games entity in the ERD diagram had an additional unnecessary column called gamesCol. This was removed from the diagram.

Ed Discussion Review 2 did not have any feedback that required us to make a decision.

For Ed Discussion Review 3, the only action that was recommended was adding more detail to the overview. This was also suggested by the TAs and the changes made regarding this feedback were discussed earlier.

For Ed Discussion Review 4, there was limited feedback that required decisions for making a change. The basis of the review was on how attributes and entities are labeled. The review mentioned that the intersection table for teams and games should be renamed to TeamGames. This suggestion was accepted, and a change was made. The other notable change was to add the IDs for entities in the overview. This detail was added to the overview in parentheses.

b) Fixes based on Feedback from Step 2:

Ed Discussion Review 1:

- Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?
    - In the DB outline, under Gyms I'm seeing regulationSize: bool, while in the schema it is a TINYINT.
    - In the DB outline, under Games I'm seeing something for "team". But nothing in the schema shows that.
- Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
    - I'm not seeing consistency. In the schema, I can see PK and other attributes are lowercase and camelcase. But I can see this changes for FKs which are capitalized and separated by an underbar. Over in the database outline, those same FKs are instead back to the original naming scheme of lowercase and camelcase. I'm guessing the schema FKs names are the default given to you by mySQL, so I would suggest changing them to be more inline with their names in the outline.
- Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?
    - Yes, it is easy enough to read. Nice straight lines.
- Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?
    - Yes, I can see the two FKs under TeamGames.
- Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?
    - I'm not seeing any normalization issues from what I can tell.
- Is the SQL file syntactically correct?
    - Yes, the file runs correctly. However, if you want to run this in mySQL without issue, I would suggest changing "Create or Replace Table" to just "Create Table" and instead include "DROP TABLE IF EXISTS TABLENAME" at the start of the file.
- In the SQL, are the data types appropriate considering the description of the attribute in the database outline?
    - Yes, the data types seem appropriate.
- In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?
    - FKs are correctly defined and used relative to the schema.
- In the SQL, are relationship tables present when compared to the ERD/Schema?
    - Yes, they are inline when compared to the ERD.
- In the SQL, is all example data shown in the PDF INSERTED?
    - Yes, all the data was inserted from the PDF.
- Is the SQL well-structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?
    - Yes, the SQL is very neat and well commented. Does not seem exported at all.

Ed Discussion Review 2:

A basketball league database is a pretty interesting idea! Here's my review of the draft and SQL file.

The model in the file follows the same outline provided in the PDF, the tables (Coaches, Teams, Players, Gyms, Games, and TeamGames) are defined just as they are mentioned in the Schema. Though the TeamGames isn't mentioned in the ERD (I'm not certain it is standard to mention intersection tables in ER diagrams though).

Speaking of, the Schema and ERD are quite easy to read. None of the relationship lines are crossed, good job!

The naming conventions used are consistent in both the Draft and the SQL file, with singular attributes and plural entities. The Schema has a strange way of mentioning the foreign keys (EX: Coaches_coachID), I would just name the foreign key identical to the primary key it is referencing.

Your intersection Table (TeamGames) is structured properly and utilizes two foreign keys (teamID and gameID) as stated in the instructions.

The example data is clear, additionaly, I thought it was quite clever using the SELECT coachID from Coaches where name = "blah" to fill the foreign key.

I think there is a small Normalization issue (I am bad with normalizing so take this with a grain of salt), If I were to delete a Coach or a Team, there would be some leftover data in other tables. I would set some ON DELETE SET NULL on the coachID for the teams (if said team can exist without a coach). Maybe a CASCADE ON DELETE if the whole team gets deleted for the coach.

Overall, great job! Have a good week!

Ed Discussion Review 3:

Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?

The schema mostly follows the provided outline. All entities are represented in the SQL code. “teams” is in the DB outline for “Games” but not in the ERD.

Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?

There is not a lot of consistency in the naming of attributes, specifically when it comes to capitalization. Additionally camel case is used in the outline while the ERD uses underscores.

Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?

The schema is easy to read. The relationship lines in the ER diagram are clear and not crossed.

Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?

The TeamGames table is properly formed to handle the many-to-many relationship between Teams and Games using teamID and gameID as foreign keys

Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?

No normalization issues from what I can see. The schema appears to be in 3NF.

Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)
I was able to run the SQL file. It is structured well.
In the SQL, are the data types appropriate considering the description of the attribute in the database outline?
I think you're using the optimal data types on everything!
In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?
Primary and foreign keys are correctly defined. You could improve upon the code by adding CASCADE operations for ON DELETE and ON UPDATE as they are not specified in the current schema. AKA Add ON DELETE SET NULL for coachID in Teams if a team can exist without a coach, and ON DELETE CASCADE if deleting a team should also delete related records.
ex code for above - FOREIGN KEY (coachID) REFERENCES Coaches(coachID) ON DELETE CASCADE ON UPDATE CASCADE
In the SQL, are relationship tables present when compared to the ERD/Schema?
Yes
In the SQL, is all example data shown in the PDF INSERTED?
Yes

Ed Discussion Review 4:
Does the schema present a physical model that follows the database outline and the ER logical diagram exactly?
Yes it does.
Is there consistency in a) naming between overview, outline, ER and schema entity/attributes b) entities plural, attributes singular c) use of capitalization for naming?
It is consistent. 2 small notes:

- In the outline under Games *team* is listed instead of *teamID*
- Intersection table *TeamGames* can be called *Team**s**Games* to be consistent.

Is the schema easy to read (e.g. diagram is clear and readable with relationship lines not crossed)?
Yes, it is very clear.
Are intersection tables properly formed (e.g. two FKs and facilitate a M:N relationship)?
Yes, with 2 FK for teamID and gameID.
Does the sample data suggest any non-normalized issues, e.g. partial dependencies or transitive dependencies?
No issues I could find.
Is the SQL file syntactically correct? This can be easily verified by using PhPMyAdmin and your CS 340 database (do not forget to take backup of your own database before you do this!)
Yes. I was able to run the commands with no issues.
In the SQL, are the data types appropriate considering the description of the attribute in the database outline?
Yes they are correct.

In the SQL, are the primary and foreign keys correctly defined when compared to the Schema? Are appropriate CASCADE operations declared?
They are set up correctly but no Cascade definition is available.
In the SQL, are relationship tables present when compared to the ERD/Schema?
Yes. *TeamGames*.
In the SQL, is all example data shown in the PDF INSERTED?
0 discrepancies between SQL and the PDF.
Is the SQL well-structured and commented (e.g. hand authored) or not (e.g. exported from MySQL)?
Very comprehensive SQL comments(!)
Overall I think your project is very well executed. I enjoyed reviewing it!
Alon

Actions taken based on feedback in Step 2:
Most feedback was great and not many changes were implemented. Our plan is consistent and thought-out, so we are executing what we know to be good. Some feedback includes information about future plans which we will take into consideration when we are at that point.

c) Project Outline and Database Outline - Updated Version:

OVERVIEW:
Our database will be a scheduling tool for a youth basketball league. Specifically, a basketball league, which has 6 total teams (teamID). Each team consists of 7 players (playerID). The league will have 4 coaches (coachID). The database will record teams, players in the league, games (gamesID) played between teams, and coaches. A coach can coach for multiple teams. Each player will be recorded in the database and assigned to a team. Each team will have a coach, team name, and a color. The games will be played between two teams and there will be a record of where the game is played (gymID). Each team will play in 6 total games. There will be 4 gyms available in the database where games can be scheduled.

The problem we are aiming to solve is the difficulty of accessing information regarding games and times. By creating this database, players, coaches, and spectators will be able to view any information that they need regarding the league, games, when games are played, players on teams, rules of the league, etc. For example, a league manager may print out a schedule of games for a single team. Or a coach may sort for games they are coaching in. Last example, a gym may create a schedule for all times games will be played there.

The database we plan to implement will have 5 different entities: team, player, coach, game, and gym. Each entity is spelled out in words below:

- **Teams**: Records the roster and basic information about a team in the league.
  - teamID: int, auto_increment, unique, not NULL, PK
  - name: varchar, not NULL, unique
  - color: varchar, not NULL, unique
  - coachID:, not Null, a 1:M relationship where coach ID is a FK inside a team. Each team must have 1 coach, and each coach can coach on multiple teams.

- **Players**: Records personal information about a player and what team they play for.
  - playerID: int, auto_increment, unique, not NULL, PK
  - name: varchar, not NULL
  - email: varchar, not NULL
  - birthday: date, not NULL
  - teamID: a 1:M relationship where teamID is a FK inside a player

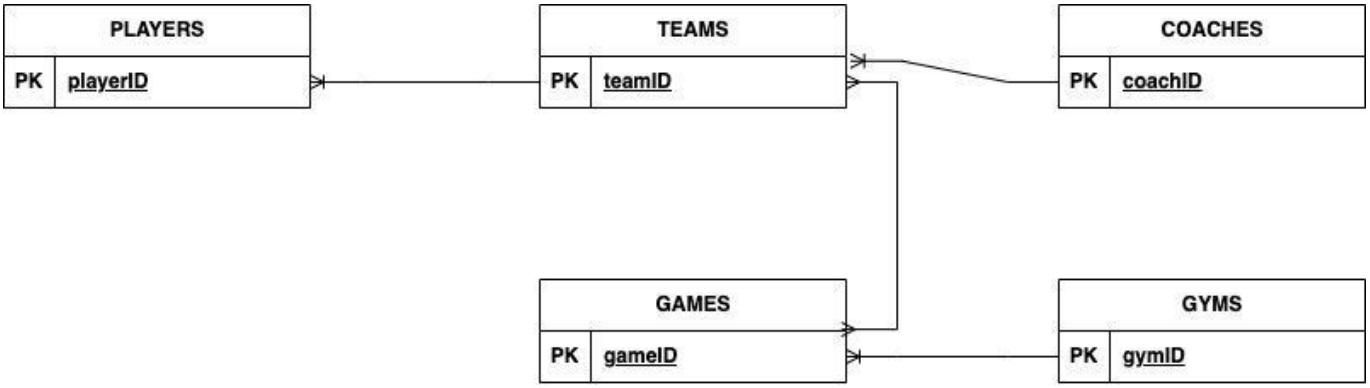
- **Coaches**: Records personal information about a coach and what team(s) they coach for.
  - coachID: int, auto_increment, unique, not NULL, PK
  - name: varchar, not NULL
  - email: varchar, not NULL
  - trainingComplete: tinyint, not NULL. 0 means false, 1 means true

- **Gyms:** A record of the gym name and if it is a regulation size court
  - gymID: int, auto_increment, unique, not NULL, PK
  - name: varchar, not NULL, unique
  - regulationSize: bool, not NULL. 0 means false, 1 means true

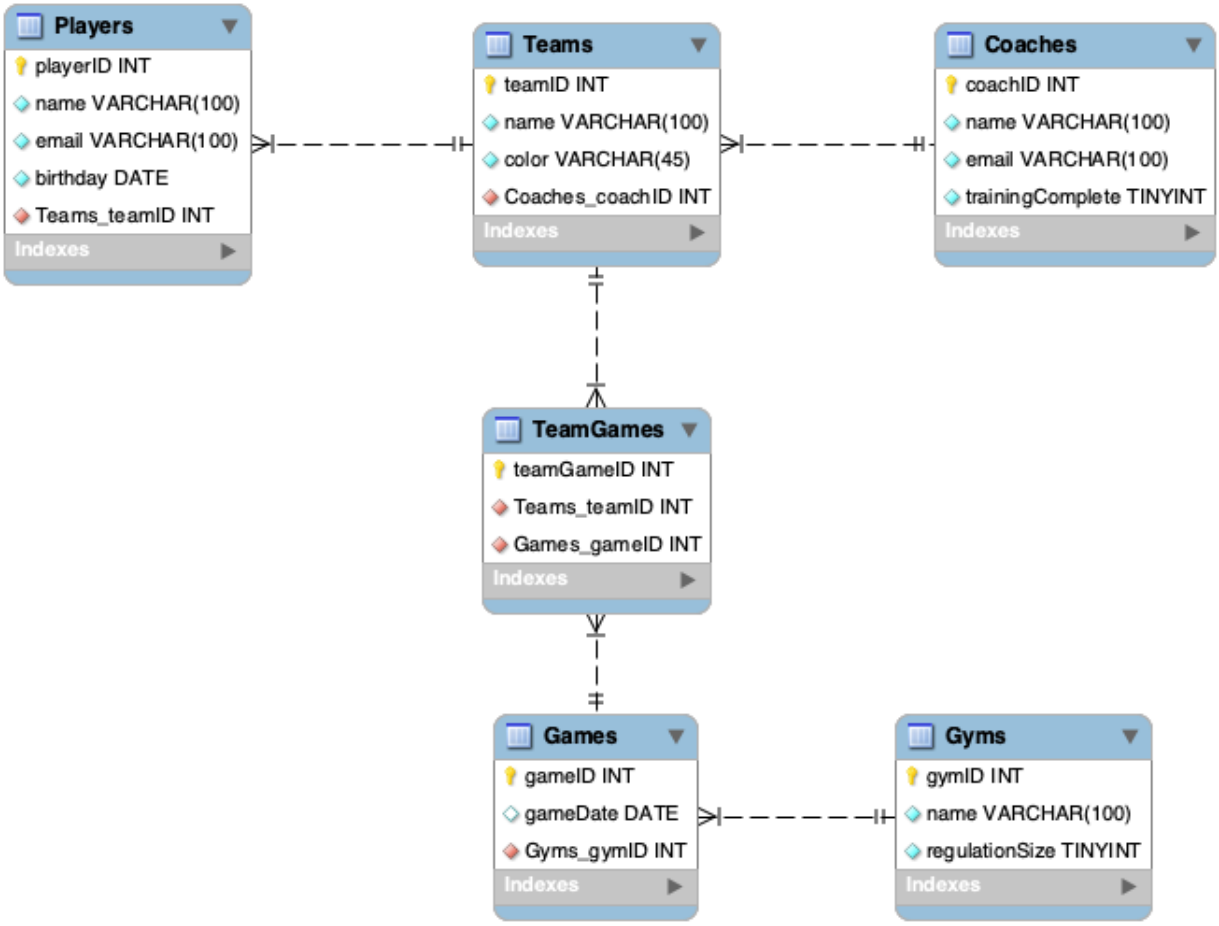
- **Games**: Records the information of a game including the gameID, date, and gym the game is played at.
  - gameID: int, auto_increment, unique, not NULL, PK
  - gameDate: date, not NULL
  - team: a N:M many-to-many relationship, where each game can have multiple teams and each team can play in multiple games.
  - gymID: a 1:M relationship where gymID is a FK in the Games table. Each game can have only 1 gym but each gym can have multiple games played there.

- **TeamGames**: This is the intersection table for Games and Teams. It records a game and the team that plays in the game. This satisfies the many to many relationship between teams and games.
  - teamGameID: int, auto_increment, unique, not NULL, PK
  - teamID: a 1:M relationship where teamID is a FK inside the TeamGames table.
  - gameID: a 1:M relationship where gameID is a FK inside the TeamGames table.

d) Entity-Relationship Diagram:



e) Schema:


Players
playerID INT
name VARCHAR(100)
email VARCHAR(100)
birthday DATE
Teams_teamID INT
Indexes
Teams
teamID INT
name VARCHAR(100)
color VARCHAR(45)
Coaches_coachID INT
Indexes
Coaches
coachID INT
name VARCHAR(100)
email VARCHAR(100)
trainingComplete TINYINT
Indexes
TeamGames
teamGameID INT
Teams_teamID INT
Games_gameID INT
Indexes
Games
gameID INT
gameDate DATE
Gyms_gymID INT
Indexes
Gyms
gymID INT
name VARCHAR(100)
regulationSize TINYINT
Indexes

f) Example Data:

The below tables outline sample data that is inserted into the table using the SQL file submitted with the zip file. While the overview section calls out larger amounts of data for the database to include, the example data only includes a small subset to demonstrate functionality.

COACH DATA			
coachID (PK)	**name**	**email**	**trainingComplete**
1	Jessie Harrison	jessieharrison@team.com	1
2	Kelly Garner	kellygarner@hotmail.com	1
3	Kaden Reid	kadenreid@gmail.com	1

TEAM DATA			
teamID (PK)	**name**	**color**	**coachID (FK)**
1	Tigers	Orange	1
2	Panthers	Blue	2
3	Eagles	Green	3
4	Bulldogs	Red	1

PLAYER DATA				
playerID (PK)	**name**	**email**	**birthday**	**teamID (FK)**
1	Max Jenkins	maxjenkins@aol.com	2002-02-04	1
2	Jade Phillips	jadephillips@team.com	2002-06-10	1
3	Sana Phelps	sanaphelps@gmail.com	2001-11-18	1
4	Jerry Davis	jerrydavis@hotmail.com	2003-01-24	1
5	Jan Park	janpark@gmail.com	2001-12-14	2
6	Tim Landry	timlandry@yahoo.com	2002-08-01	2
7	Cordell Roberts	cordellroberts@gmail.com	2003-02-01	2
8	Nana Kelley	nanakelley@yahoo.com	2002-04-22	3

GYM DATA		
gymID (PK)	**name**	**regulationSize**
1	Southport Fieldhouse	1
2	Kalama Gym	0
3	Allen Center	1

GAMES DATA		
gameID (PK)	**gameDate**	**gymID (FK)**
1	2024-10-01	1
2	2024-10-03	2
3	2024-10-05	3
4	2024-10-08	1
5	2024-10-10	2
6	2024-10-12	3

TEAMGAMES DATA		
teamGameID	**teamID (FK)**	**gameID (FK)**
1	1	1
2	2	1
3	3	2
4	4	2
5	1	3
6	3	3
7	2	4
8	4	4
9	1	5
10	4	5

11	2	6
12	3	6