Entities:

1. **User**:
    - **Assumptions**: Users can create accounts and save podcasts. Each user has unique Username and Password.
    - **Reason for being an entity**: Users need to manage their personal information, such as login details and saved podcasts.
    - **Cardinality**: A user can save many podcasts (0..*), but each podcast is only saved by one user (1..1).
2. **Podcast**:
    - **Assumptions**: A podcast is defined by ID, the associated User, and its name. Each podcast can reference multiple books, people, and companies.
    - **Reason for being an entity**: Podcasts need to reference other entities like books, people, and companies.
    - **Cardinality**: A podcast must belong to exactly one user (1..1). It can reference many PodcastBooks, PodcastPeople, and PodcastCompanies (0..* for all).
3. **PodcastBooks:**
    - **Assumptions:** This entity links podcasts with the books they reference.
    - **Reason for being an entity:** Since multiple podcasts can reference the same book and a book can be mentioned in multiple podcasts, the PodcastBooks entity will allow us to track which podcasts mention which books.
    - **Cardinality:** 1 for podcasts and 0..* for books, meaning a PodcastBook can reference only one podcast but a PodcastBook can refer to many Books.
4. **PodcastPeople:**
    - **Assumptions:** This entity links podcasts with the people they reference.
    - **Reason for being an entity:** Since multiple podcasts can reference the same book and a book can be mentioned in multiple podcasts, the PodcastBooks entity will allow us to track which podcasts mention which books.
    - **Cardinality:** 1 for podcasts and 0..* for people, meaning a PodcastPeople can reference only one podcast but a PodcastPeople can refer to many People.
5. **PodcastCompanies:**
    - **Assumptions:** This entity links podcasts with the companies they reference.
    - **Reason for being an entity:** Since multiple podcasts can reference the same book and a book can be mentioned in multiple podcasts, the PodcastBooks entity will allow us to track which podcasts mention which books.
    - **Cardinality:** 1 for podcasts and 0..* for companies, meaning a PodcastCompany can reference only one Podcast but a PodcastCompany can refer to many Companies.
6. **Books**:
    - **Assumptions**: Books mentioned in podcasts are stored with information such as the name, Wikipedia link, author, average rating, and book description.
    - **Reason for being an entity**: Since books may be referenced by multiple podcasts, it is better to make it an entity rather than an attribute of a podcast.

    - **Cardinality**: Books can have 1 relation with a PodcastBooks..
7. **People**:
    - **Assumptions**: People mentioned in podcasts are stored with their name, Wikipedia link, net worth, profession, and a description of who the person is.
    - **Reason for being an entity**: People can be referenced by multiple podcasts so it is better to make them an entity rather than an attribute.
    - **Cardinality**: People can have 1 relation with a PodcastPeople.
8. **Companies**:
    - **Assumptions**: Companies mentioned in podcasts are stored with details like name, Wikipedia link, state, revenue, industry, and company description.
    - **Reason for being an entity**: Companies can also be referenced by multiple podcasts so it is better to make it an entity rather than an attribute as well.
    - **Cardinality**: Companies can have 1 relation with a PodcastCompanies.

Relationships:

- **Saved:**
    - **Assumptions**: A user can save multiple podcasts, but a podcast is associated with only one user.
    - **Cardinality**: 0..* on the podcast side, 0..1 on the user side (as users can save podcasts but do not have to).
- **BookReference/PersonReference/CompanyReference**:
    - **Assumptions**: A podcast can reference multiple PodcastsBooks, PodcastPeople, and PodcastCompanies. But each of these can only reference one Podcast
    - **Cardinality**: Their relationships are one-to-many since we want a single podcast to be able to reference any amount of books, people, or companies. We do this via the PodcastBooks, PodcastPeople, and PodcastCompanies entities to ensure the BCNF normalization where BookID and PodcastID can identify any book (same with company and people entities).

Normalization:

- We used BCNF Normalization. Initially, when we made our first iteration of the UML we had the PodcastsBook & Books (and the corresponding tables for Companies and People) as the same table. The issue with this is PodcastID is a partial dependency as it would not uniquely determine BookID. For example, a podcast could reference multiple books, but the podcast ID wouldn't uniquely identify a particular book. In our case, PodcastID was not a superkey so the database wasn't normal. To fix this we created the PodcastBook, PodcastCompanies, and PodcastPeople to break up the many-to-many relations. These tables ensure the combination of PodcastID + BookID (or CompanyID, PersonID) forms a key that allows each entity to have its attributes without dependency on PodcastID alone.

Relational Schema:

User(UserID: INT [PK], Username: VARCHAR(255), Password: VARCHAR(255))

Podcast(PodcastID: INT [PK], UserID: INT [FK to User.UserID], PodcastName: VARCHAR(255), Link: VARCHAR(255))

PodcastBooks(PodcastID: INT [FK to Podcast.PodcastID], BookID: INT [FK to Books.BookID])

Books(BookID: INT [PK], BookName: VARCHAR(255), WikipediaLink: VARCHAR(255), Author: VARCHAR(255), AvgRating: DECIMAL(3,2), BookDescription: VARCHAR(255))

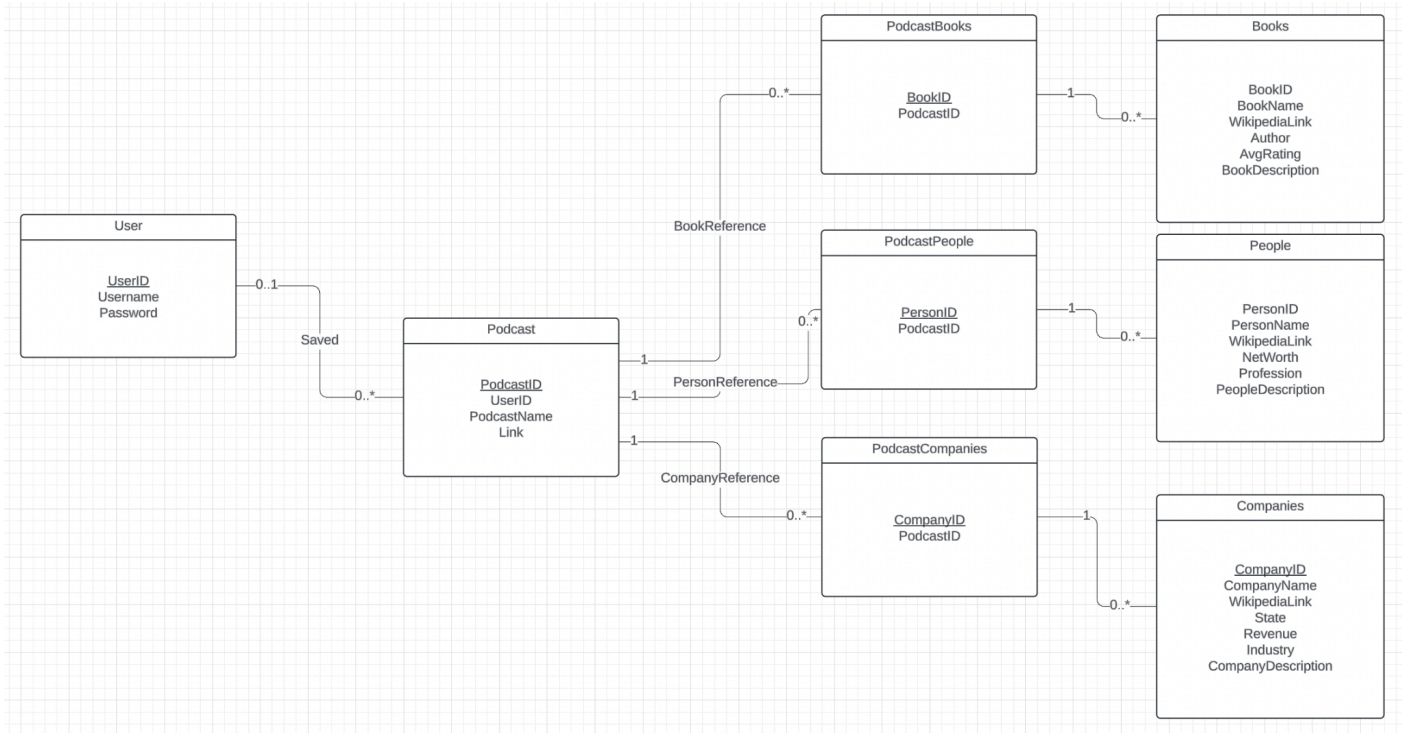
PodcastPeople(PodcastID: INT [FK to Podcast.PodcastID], PersonID: INT [FK to People.PersonID])

People(PersonID: INT [PK], PersonName: VARCHAR(255), WikipediaLink: VARCHAR(255), NetWorth: DECIMAL(15,2), Profession: VARCHAR(255), PeopleDescription: VARCHAR(255))

PodcastCompanies(PodcastID: INT [FK to Podcast.PodcastID], CompanyID: INT [FK to Companies.CompanyID])

Companies(CompanyID: INT [PK], CompanyName: VARCHAR(255), WikipediaLink: VARCHAR(255), State: VARCHAR(255), Revenue: DECIMAL(15,2), Industry: VARCHAR(255), CompanyDescription: VARCHAR(255))

UML Diagram:


User
UserID
Username
Password
0..1
Saved
0..*
Podcast
PodcastID
UserID
PodcastName
Link
1
BookReference
0..*
PodcastBooks
BookID
PodcastID
1
0..*
Books
BookID
BookName
WikipediaLink
Author
AvgRating
BookDescription
1
PersonReference
0..*
PodcastPeople
PersonID
PodcastID
1
0..*
People
PersonID
PersonName
WikipediaLink
NetWorth
Profession
PeopleDescription
1
CompanyReference
0..*
PodcastCompanies
CompanyID
PodcastID
1
0..*
Companies
CompanyID
CompanyName
WikipediaLink
State
Revenue
Industry
CompanyDescription