Project title : Database design of MUSIC management system.

Submitted to :

Md. Samsuddoha

Assistant Professor

Dept. of Computer Science and Engineering

University Of Barishal.

Submitted by :

Arnob Saha

Dept. of Soil and Environmental sciences

University Of Barishal

Serial : 029 (DB-1)

Date of submission: 09-06-2024

Database design

1. Designing (Entity Relationship) ER Diagram

Steps of drawing ERD

1. Identify the Entities required.
2. Identify the Attributes and Primary key for each Entity.
3. Identify the Relationship needed.
4. Identify the Cardinality Ratio and Participation.
5. Draw the Diagram.

1. Identify the Entities Required

The entities required are:

- Users
- Artists
- Albums
- Tracks
- Playlists
- Playlist_Tracks
- User_Reviews
- User_Favorites

2. Identify the Attributes and Primary Key for Each Entity

Users

Attributes: user_id (Primary Key), username, email, password, date_joined

Artists

Attributes: artist_id (Primary Key), name, bio, genre

Albums

Attributes: album_id (Primary Key), title, release_date, artist_id (Foreign Key)

Tracks

Attributes: track_id (Primary Key), title, duration, album_id (Foreign Key), artist_id (Foreign Key)

Playlists

Attributes: playlist_id (Primary Key), name, description, user_id (Foreign Key), created_at

Playlist_Tracks

Attributes: playlist_id (Foreign Key), track_id (Foreign Key), added_at

User_Reviews

Attributes: review_id (Primary Key), user_id (Foreign Key), track_id (Foreign Key), rating, comment, review_date

User_Favorites

Attributes: user_id (Foreign Key), track_id (Foreign Key)

3. Identify the Relationships Needed

a. Users create Playlists

b. Artists create Albums

c. Albums contain Tracks

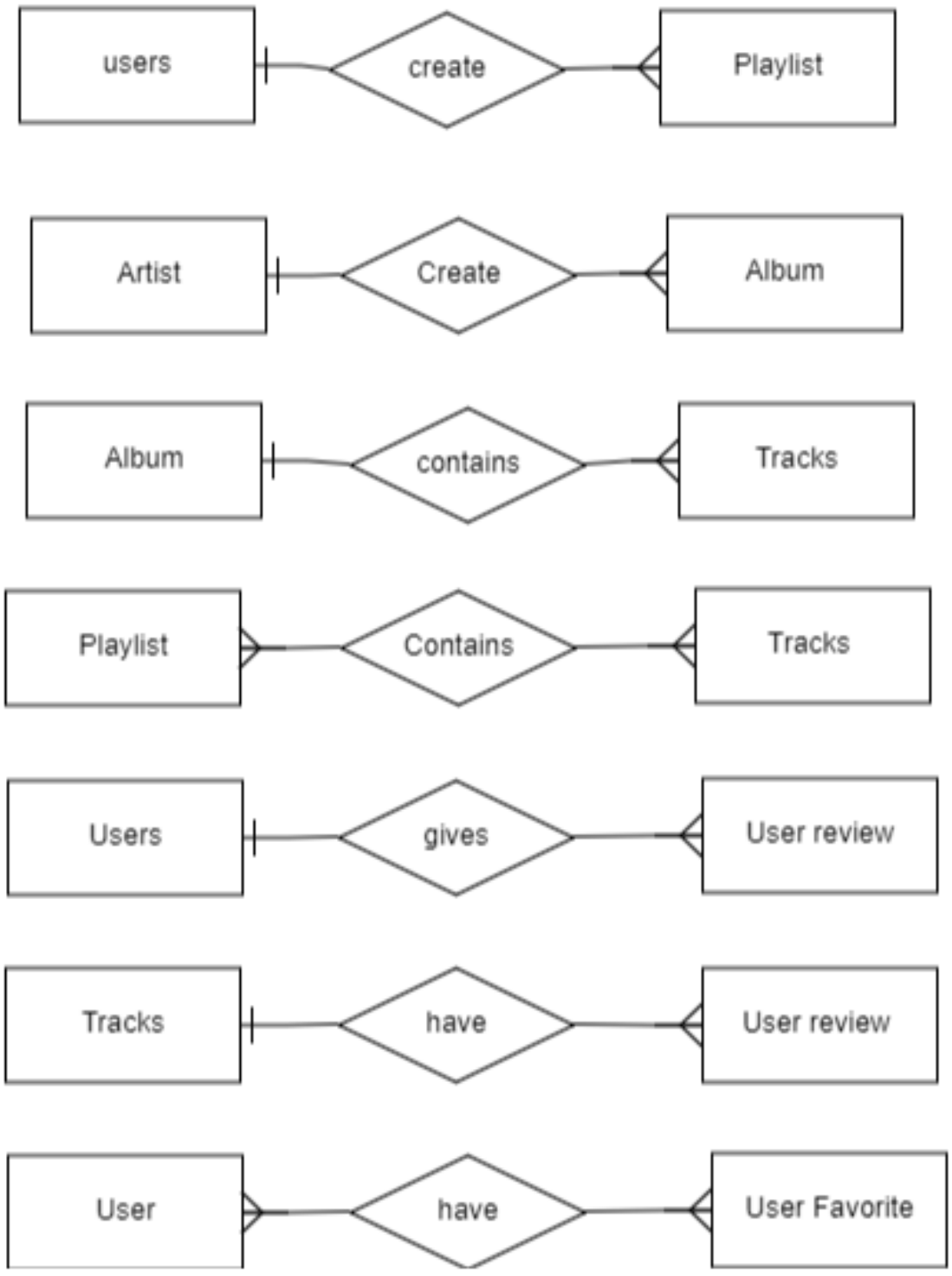
d. Playlists contain Tracks

e. Users review Tracks

f. Users favorite Tracks

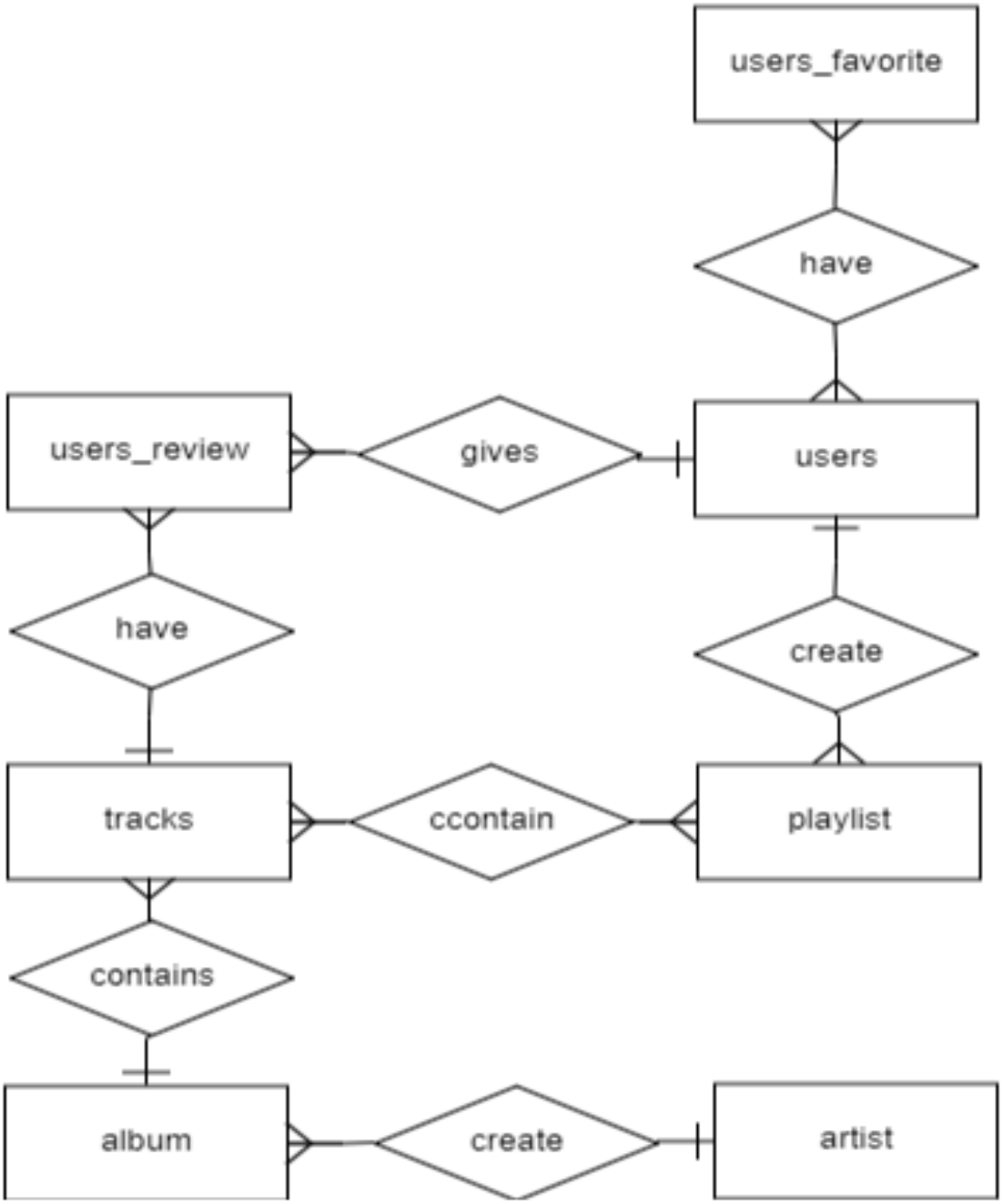
4. Identify the Cardinality Ratio and Participation

- **Users and Playlists**: One-to-Many (One user can create many playlists, each playlist is created by one user)
- **Artists and Albums**: One-to-Many (One artist can create many albums, each album is created by one artist)
- **Albums and Tracks**: One-to-Many (One album can contain many tracks, each track belongs to one album)
- **Playlists and Tracks**: Many-to-Many (One playlist can contain many tracks, and one track can be in many playlists)
- **Users and User_Reviews**: One-to-Many (One user can write many reviews, each review is written by one user)
- **Tracks and User_Reviews**: One-to-Many (One track can have many reviews, each review is for one track)
- **Users and User_Favorites**: Many-to-Many (One user can have many favorite tracks, and one track can be favorited by many users)


users
create
Playlist
Artist
Create
Album
Album
contains
Tracks
Playlist
Contains
Tracks
Users
gives
User review
Tracks
have
User review
User
have
User Favorite

5. Diagram:

Here's the ER diagram:



2.Reduction to DATABASE schema:

Users(user_id (Primary Key), username, email, password, date_joined)

Artists(artist_id (Primary Key), name, bio, genre)

Albums(album_id (Primary Key), title, release_date, artist_id)

Track(track_id (Primary Key), title, duration, album_id (Foreign Key), artist_id)

Playlists(playlist_id (Primary Key), name, description, user_id (Foreign Key), created_at)

Playlist_Tracks:(playlist_id (Foreign Key), track_id (Foreign Key), added_at)

User_Reviews(review_id (Primary Key), user_id (Foreign Key), track_id (Foreign Key), rating, comment, review_date)

User_Favorites(user_id (Foreign Key), track_id (Foreign Key))