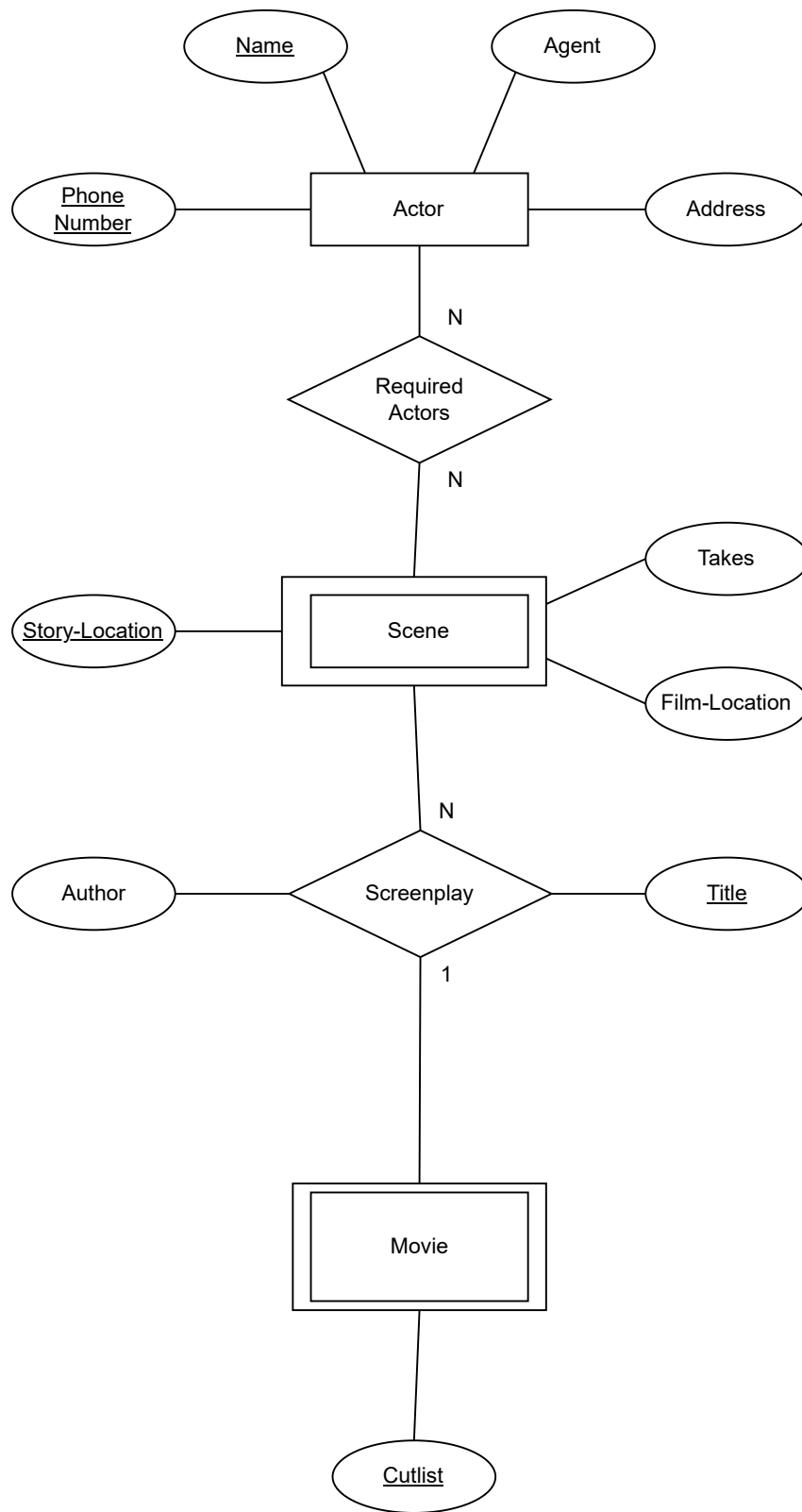


Question 1 ER Model



Question 1 Schema

Assumptions:

- A movie can only exist if there already exists a screenplay
- A scene can only exist if the required actors are present and are filmed at the film location

Actor			
<u>PhoneNumber</u>	<u>Name</u>	Agent	Address

Scene		
<u>Story-Location</u>	Film-Location	Takes

Movie	
<u>Cutlist</u>	<u>Screenplay-Title</u>

Screenplay		
<u>Title</u>	<u>Story-Location</u>	Author

Required Actors			
<u>Story-Location</u>	<u>PhoneNumber</u>	<u>Name</u>	Actors

Foreign Keys:

Table attribute -> table attribute

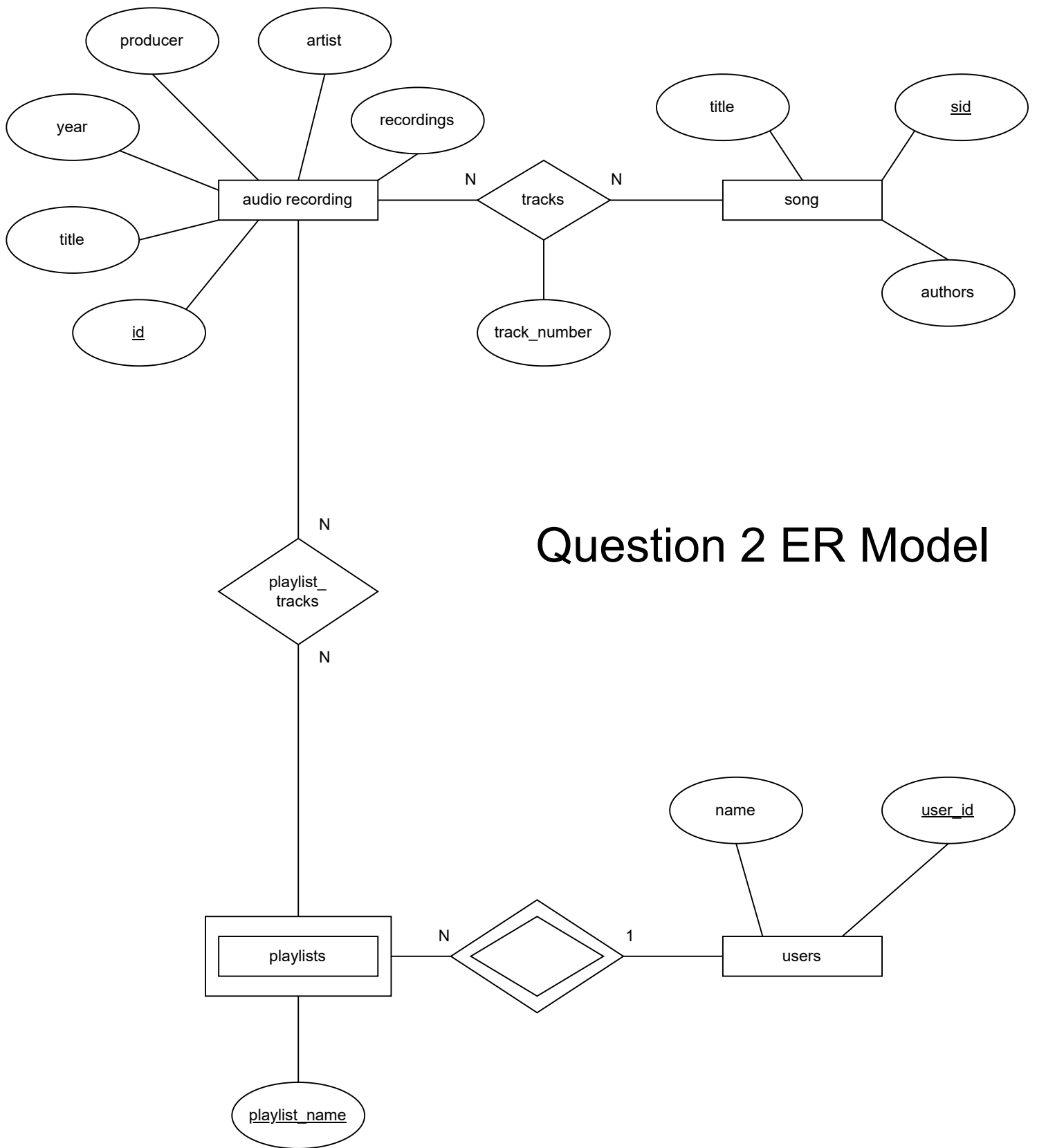
Required Actors PhoneNumber -> Actor PhoneNumber

Required Actors Name -> Actor Name

Required Actors Story-Location -> Screenplay Story-Location

Screenplay Story-Location -> Scene Story-Location

Movie Screenplay-Title -> Screenplay Title



Question 2 ER Model

Question 2 Schema

Assumptions:

- A track is the act of composing a song into an audio recording
- Playlist_Tracks interact with audio recordings so that only audio recordings can be placed on a playlist
- Playlists only exist if there exists a user that creates a playlist
- Recordings are the number of times a song was recorded

Song		
<u>Sid</u>	Title	Authors

Tracks		
<u>Id</u>	<u>Sid</u>	Track_Number

Audio Recording					
<u>Id</u>	Title	Artist	Producer	Year	Recordings

Playlist_Tracks		
<u>Id</u>	<u>Playlist_Name</u>	<u>User_id</u>

Playlist	
<u>User_id</u>	<u>Playlist_Name</u>

User		
<u>User_id</u>	<u>Playlist_Name</u>	Name

Foreign keys:

Table attribute -> table attribute

User Playlist_Name -> Playlist Playlist_Name

Playlist User_id -> User User_id

Playlist_Tracks Playlist_Name -> Playlist Playlist_Name

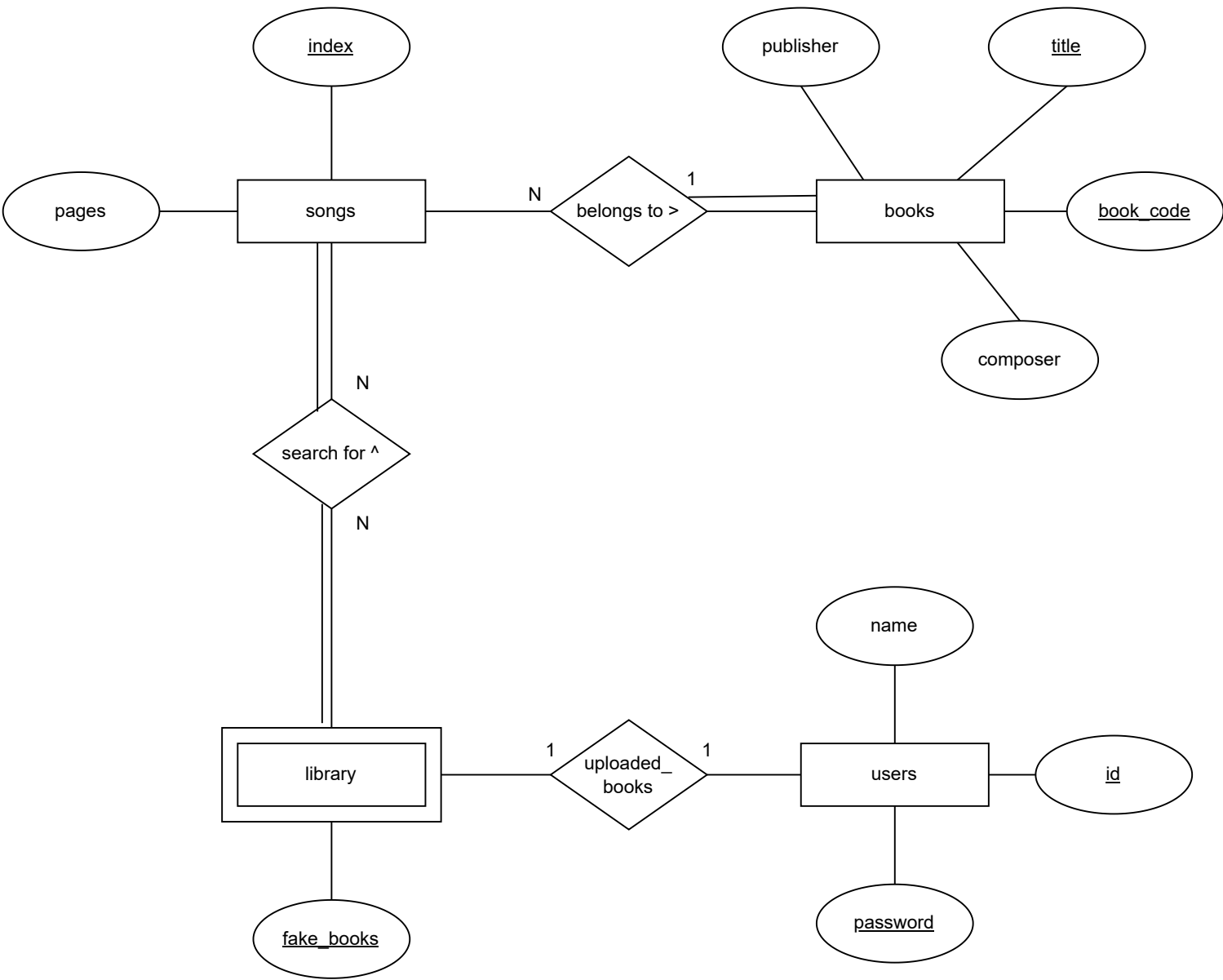
Playlist_Tracks User_id -> User User_id

Playlist_Tracks Id -> Recordings Id

Tracks Id -> Audio Recording Id

Tracks Sid -> Song Sid

Question 3 ER Model



Question 3 Schema

Assumptions:

- The pages attribute takes the offset into account
- The library can only exist if there are books added to the database or books are uploaded by a user
- A fake_books is an instance of user's books that are both accessed and uploaded
- An uploaded book can only be accessed by the user who uploaded it (stored in fake_books)

Users			
<u>Id</u>	<u>Password</u>	Name	<u>Fake_books</u>

Library			
<u>Fake_books</u>	<u>Id</u>	<u>Password</u>	<u>Index</u>

Songs			
<u>Index</u>	Pages	<u>Book_code</u>	<u>Fake_books</u>

Books				
<u>Book_code</u>	<u>Title</u>	Publisher	Composer	<u>Index</u>

Foreign Keys:

Table attribute -> table attribute

Users Fake_books -> Library Fake_books

Library Id -> Users Id

Library Password -> Users Password

Library Index -> Songs Index

Songs Fake_books -> Library Fake_books

Songs Book_code -> Books Book_code

Books Index -> Songs Index

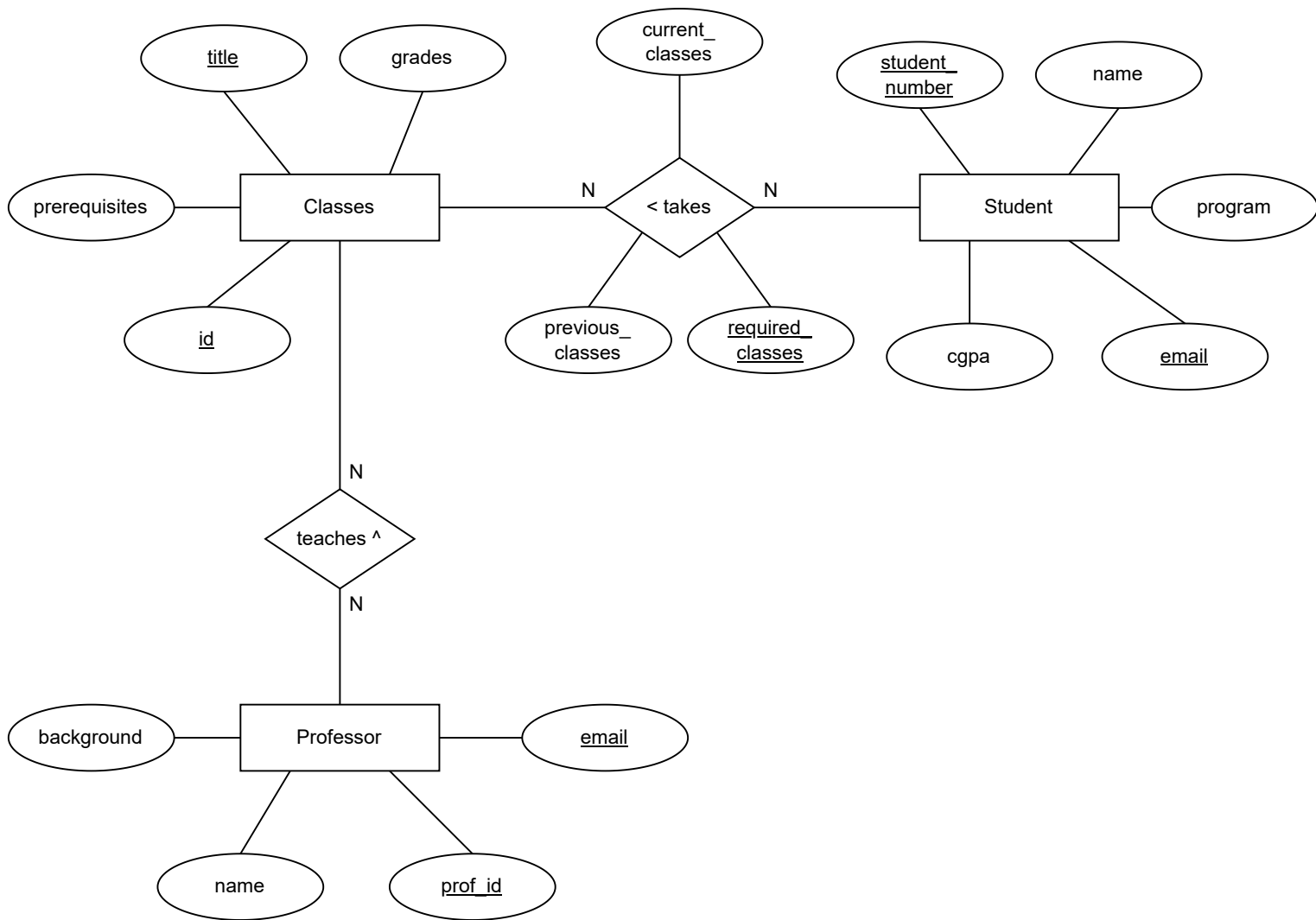
Carleton Class Database Project

Description

This project is motivated by the needs of Professor Nel for COMP3005. A Carleton Class database will monitor classes that Carleton Offers as well as the students registered in those classes. This will aide university students as it will display the courses needed for their degree as well the required prerequisites. This will also aide students while taking courses as the letter grade they receive will be a reflection on how well they understand the material.

The Carleton Class database will also be of use to professors as they are able to see how their students are doing in their class. This will allow for professors to adjust their teaching style or course content to aid the students in succeeding.

Finally, it will aid Carleton University as will keep track of all students and their required classes, their average, a class average for a professor and ultimately determine whether a student can graduate.



Schema

Assumptions:

- Background is the professors teaching i.e. software, math, English, etc.
- Classes can in fact exist if there are no students or professors as it is its own entity
- The previous_classes attribute stores classes that students have taken, including grades
- Any entity with more than one underlined attribute has a compound primary key

Student					
<u>Student_number</u>	<u>Email</u>	Name	Program	CGPA	<u>Required_classes</u>

Takes						
<u>Required_classes</u>	Current_classes	Previous_classes	<u>Title</u>	<u>Id</u>	<u>Student_number</u>	<u>Email</u>

Classes						
<u>Title</u>	<u>Id</u>	Grades	Prerequisites	<u>Required_classes</u>	<u>Prof_id</u>	<u>Email</u>

Professor					
<u>Prof_id</u>	<u>Email</u>	Name	Background	<u>Title</u>	<u>Id</u>

Foreign Keys:

Table attribute -> table attribute

Student Required_classes -> Takes Required_classes

Takes Student_number -> Student Student_number

Takes Email -> Student Email

Takes Title -> Classes Title

Takes Id -> Classes Id

Classes Required_classes -> Takes Required_classes

Classes Prof_id -> Professor Prof_id

Classes Email -> Professor Email

Professor Id -> Classes Id

Professor Title -> Classes Title