University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: October 25th**, 2021**

Group Number: 108

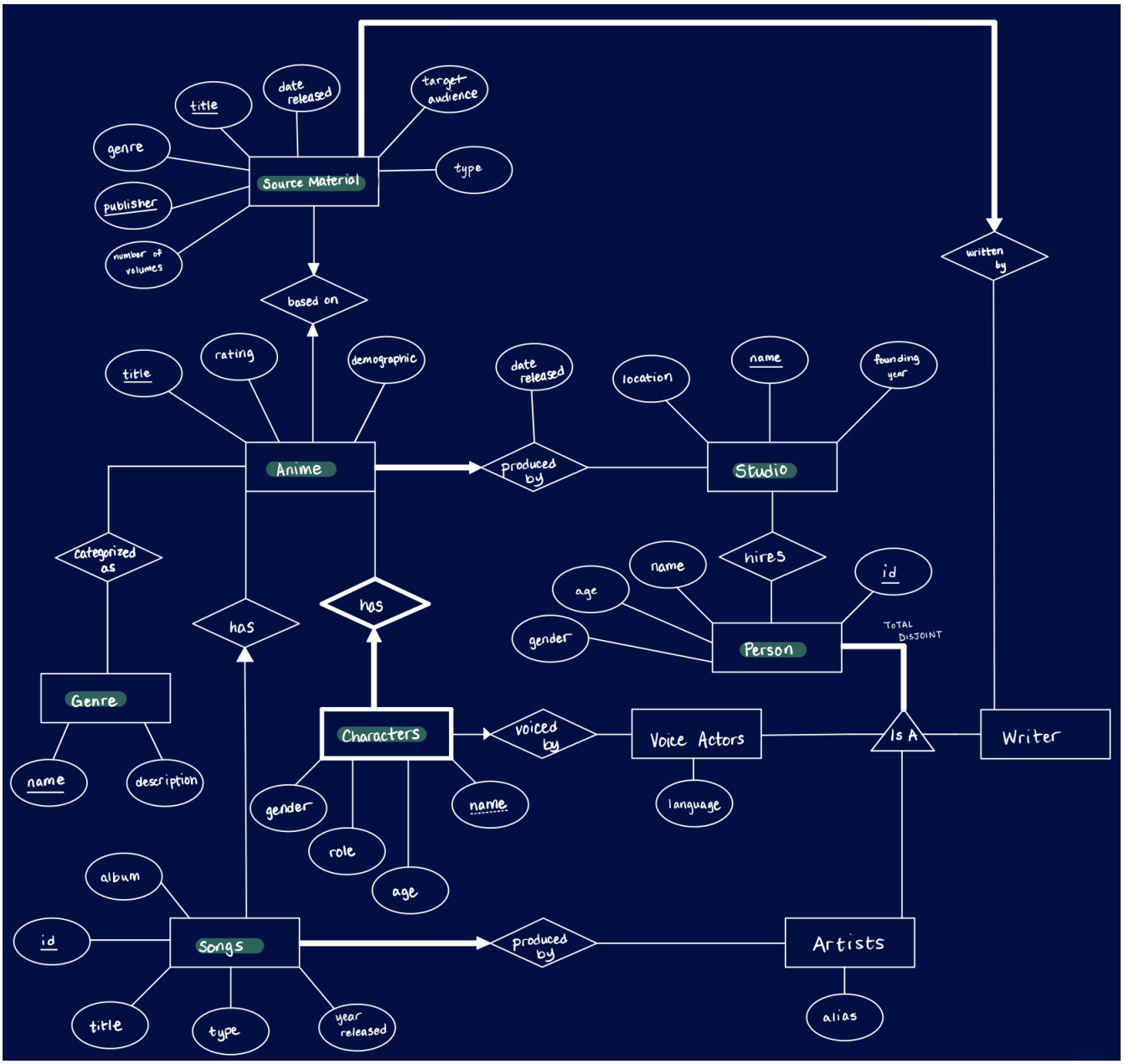
Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Jessa Shi	30612148	u0d2b	jessashi96@gmail.com
Crystal Li	19712603	q7a3b	crystalliyy0226@gmail.com
Ethan Ly	89125033	i0p2b	ethanly@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

CPSC 304: Milestone 2:
Anime Database

Anime Database: ER Diagram



Changes to our ER diagram include...

1. **Changing ISA's to triangles from the original error of diamonds**
2. **Adding in constraints to ISA's because of TA recommendation**
3. **Moving Anime entity's date released attribute over to producedBy relation**
4. **Adding description attribute to Genre entity**
5. **Removing type attribute from Artists attribute**
6. **Removing licensor attribute from Anime entity because it was irrelevant**
7. **Removing ISA relationship between SourceMaterial, Manga, Light Novel and representing the two with a type attribute instead**
8. **Making Character entity a weak entity**
9. **Adding targetAudience attribute to SourceMaterial entity**
10. **Adding yearReleased, album attributes to Song entity**
11. **Removing length attribute from Song entity**
12. **Adding alias attribute to Artist entity**
13. **Adding language attribute to Voice Actor entity**
14. **Replacing birthday attribute to Person entity with age attribute**
15. **Adding id attribute to Person entity and using it as the new key**

Relational Model Schema

> Anime_BasedOn_ProducedBy (animeTitle: string, **sourceMaterialTitle**: string, **sourceMaterialPublisher**: string, **studioName**: string, dateReleased: string, demographic: string, rating: integer) ({sourceMaterialTitle, sourceMaterialPublisher} must be UNIQUE, studioName cannot be NULL)

> Genre(name: string, description: string)

> CategorizedAs(**animeTitle**: string, **genreName**: string)

> SourceMaterial_WrittenBy(sourceTitle: string, publisher: string, **writerID**: string, type: string, dateReleased: string, genre: string, targetAudience: string, numVolumes: integer) (writerID cannot be NULL)

> Studio(studioName: string, location: string, foundingYear: integer)

> Hires(**studioName**: string, **personID**: string)

> Person(personID: string, name: string, gender: string, age: integer, language: string)

> VoiceActor(voiceActorID: string, name: string, gender: string, age: integer, language: string)

> Writer(writerID: string, name: string, gender: string, age: integer)

> Artists(artistID: string, name: string, gender: string, age: integer, alias: string)

> Characters_VoicedBy_Has(characterName: string, **animeTitle**: string, **voiceActorID**: string, gender: string, role: string, age: integer) (animeTitle cannot be NULL)

> Songs_ProducedBy_Has(songID: string, **animeTitle**: string, **artistId**: string, title: string, type: string, album: string, yearReleased: string) (artistID cannot be NULL)

Functional Dependencies

- Anime_BasedOn_ProducedBy: animeTitle → sourceMaterialTitle, sourceMaterialPublisher, studioName, dateReleased, demographic, rating
- SourceMaterial_WrittenBy: sourceTitle, publisher → writerID, type, dateReleased, genre, targetAudience, numVolumes
- SourceMaterial_WrittenBy: genre → targetAudience
- Genre: name → description
- Studio: studioName → location, foundingYear
- Person: personID → name, age, gender
- VoiceActor: voiceActorID → voiceActorLanguage, name, age, gender
- Artist: artistID → artistAlias, name, age, gender
- Characters_VoicedBy_Has: animeTitle, characterName → voiceActorID, gender, role, age
- Songs_ProducedBy_Has: songID → animeTitle, artistID, title, type, album, yearReleased
- Songs_ProducedBy_Has: title, yearReleased → album

Normalization

FD1: SourceMaterials_WrittenBy
sourceTitle, publisher → writerID, type, dateReleased, genre, targetAudience, numVolumes
genre → targetAudience
R(sourceTitle, publisher, writerID, type, dateReleased, genre, targetAudience, numVolumes)
Decompose: genre → targetAudience (violates BCNF)
R1(genre, targetAudience)
R2(title, publisher, writerID, type, dateReleased, genre, numVolumes)

FD2: Songs_ProducedBy_Has
songID → animeTitle, artistID, title, type, album, yearReleased
title, yearReleased → album
R(songID, animeTitle, artistID, title, type, album, yearReleased)
Decompose: title, yearReleased → album (violates BCNF)

R1(title, yearReleased, album)
R2(songID, animeTitle, artistID, title, type, yearReleased)

List of Tables:

- Anime_BasedOn_ProducedBy (animeTitle: string, **sourceMaterialTitle**: string, **sourceMaterialPublisher**: string, **studioName**: string, dateReleased: string, demographic: string, rating: integer) ({sourceMaterialTitle, sourceMaterialPublisher} must be UNIQUE, studioName cannot be NULL)
- Genre(name: string, description: string)
- CategorizedAs(**animeTitle**: string, **genreName**: string)
- SourceMaterialClassification(genre: string, targetAudience: string)
- SourceMaterialInfographics(sourceTitle: string, publisher: string, **writerID**: string, type: string, dateReleased: string, **genre**: string, numVolumes: integer) (writerID cannot be NULL)
- Studio(studioName: string, location: string, foundingYear: integer)
- Hires(**studioName**: string, **personID**: string)
- Person(personID: string, name: string, gender: string, age: integer, language: string)
- VoiceActor(voiceActorID: string, name: string, gender: string, age: integer, language: string)
- Writer(writerID: string, name: string, gender: string, age: integer)
- Artists(artistID: string, name: string, gender: string, age: integer, alias: string)
- Characters_VoicedBy_Has(characterName: string, **animeTitle**: string, **voiceActorID**: string, gender: string, role: string, age: integer) (animeTitle cannot be NULL)
- SongsRecord(title: string, yearReleased: string, album: string)
- SongType(songID: string, **animeTitle**: string, **artistID**: string, **title**: string, type: string, **yearReleased**: string)

SQL DDL

https://github.students.cs.ubc.ca/CPSC304-2021W-T1/project_i0p2b_q7a3b_u0d2b/blob/master/database.sql

Populating Tables

https://github.students.cs.ubc.ca/CPSC304-2021W-T1/project_i0p2b_q7a3b_u0d2b/blob/master/database.sql

Queries

- Insertion: Add name, location, founding date to the studio table
- Deletion: Delete in character table with anime delete-on-cascade
- Update: Update title of song in the song table
- Selection: Select anime with given rating