University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: 10/12/2024

Group Number: 36

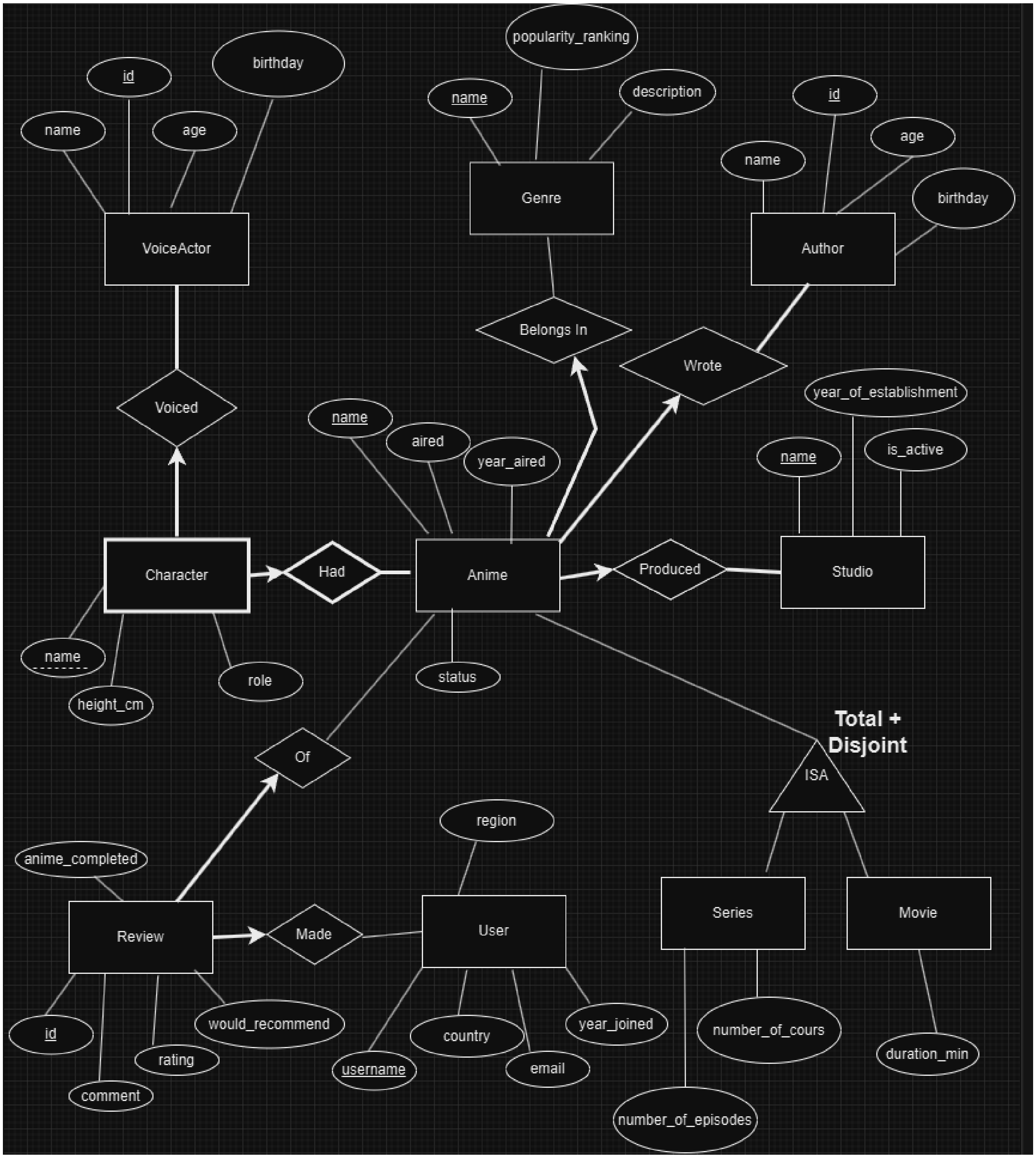
Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Enoch Hsu	56998222	t7j4b	ehsu03@students.cs.ubc.ca
Ryan Tsz Tsun Chan	98578685	q0u8p	ryanttchan123@gmail.com
Bogdan Popereko	31414576	u8l5j	pk762@students.cs.ubc.ca

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

Project Summary:
Our database intends to organize the information about anime so users of the database can easily find any details of interest. Additionally, the database will also store reviews created by users about specific anime.

ER Diagram:



Changelog

- changed attributes to be snake case for ease of viewing
- changed the character and anime relationship to many to one to reflect characters being a weak entity of anime
- changed the genre and anime relationship to one to many to simplify the data
- removed question mark for consistency
- renamed statuses to fit boolean forms for simplicity
- renamed size and duration to a specific unit for ease of data storing
- changed genre to not require an anime for clarity
- change country of origin to country for simplicity
- added attributes for having more FDs
- changed entity names to camel case for sql syntax requirements
- there were no TA suggestions from the previous milestone

Schema:

Author (id: INT, name: VARCHAR(255), age: INT, birthday: DATE)
PKs: id
Not null: name

VoiceActor (id: INT, name: VARCHAR(255), age: INT, birthday: DATE)
PKs: id
Not null: name

Studio (name: VARCHAR(255), year_of_establishment: INT, is_active: BOOLEAN)
PKs: name
Not null: year_of_establishment, is_active

Genre (name: VARCHAR(30), popularity_ranking: INT, description: VARCHAR(255))
PKs: name
Unique: popularity_ranking
Not null: description

Anime (name: VARCHAR(255), author_id: INT, studio: VARCHAR(255), genre: VARCHAR(30), year_aired: INT, status: ENUM('Aired', 'Airing', 'Unaired'))

PKs: name
FKs: author_id -> Author(id), studio -> Studio(name), genre -> Genre(name)
Not null: year_aired, author_id, studio, genre, status

Movie (anime_name: VARCHAR(255), duration_min: INT)
PKs: anime_name
FKs: anime_name -> Anime(name)

Series (anime_name: VARCHAR(255), number_of_episodes: INT, number_of_cours: INT)
PK: anime_name
FKs: anime_name -> Anime(name)

Character (name: VARCHAR(255), anime_name: VARCHAR(255), va_id: INT, height_cm: INT, role: ENUM('Protagonist', 'Antagonist', 'Supporting', 'Other'))
PKs: (anime_name, name)
FKs: va_id -> VoiceActor(id), anime_name -> Anime(name)
Not null: role, anime_name

User (username: VARCHAR(255), country: VARCHAR(50), region: ENUM('Africa', 'Asia', 'Europe', 'North America', 'Oceania', 'South America'), email: VARCHAR(255), year_joined: INT)
PKs: username
Unique: email
Not null: year_joined

Review (id: INT, user: VARCHAR(255), anime: VARCHAR(255), comment: VARCHAR(1000), rating: INT, would_recommend: BOOLEAN, anime_completed: BOOLEAN)
PKs: id
FKs: user -> User(username), anime -> Anime(name)
CKs: {user, anime}
Not null: rating, would_recommend, anime_completed

Functional Dependencies:

VoiceActor

- id -> name, age, birthday
- birthday -> age

Author

- id -> name, age, birthday
- birthday -> age

Studio

- name -> year_of_establishment, is_active

Genre

- name -> popularity_ranking, description

Anime

- name -> author_id, studio, genre, year_aired, status

Movie

- anime_name -> duration_min

Series

- anime_name -> number_of_episodes, number_of_cours
- number_of_episodes -> number_of_cours

Character

- name, anime_name -> va_id, height_cm, role

User

- username -> country, region, email, year_joined
- country -> region

Review

- id -> user, anime, rating, comment, would_recommend, anime_completed
- user, anime -> id

Normalization:

Author (id: INT, name: VARCHAR(255), age: INT, birthday: DATE)
PKs: id
id -> name
id -> age (redundant)
id -> birthday
birthday -> age (violates 3NF, decompose)
↓
Birthday (date: DATE, age: INT)
PKs: date
+
Author (id: INT, name: VARCHAR(255), birthday: DATE)
PKs: id
FKs: birthday -> Birthday(date)

VoiceActor (id: INT, name: VARCHAR(255), age: INT, birthday: DATE)
PKs: id
id -> name
id -> age (redundant)
id -> birthday
birthday -> age (violates 3NF, decompose)
↓
Birthday (date: DATE, age: INT) *Note: same schema as author's birthday so they will be combined*
PKs: date
+
VoiceActor (id: INT, name: VARCHAR(255), birthday: DATE)
PKs: id
FKs: birthday -> Birthday(date)

Studio (name: VARCHAR(255), year_of_establishment: INT, is_active: BOOLEAN)
PKs: name
name -> year_of_establishment
name -> is_active
↓
Studio is in 3NF and BCNF

Genre (name: VARCHAR(30), popularity_ranking: INT, description: VARCHAR)
PKs: name
name -> popularity_ranking
name -> description
↓
Genre is in BCNF and 3NF

Anime (name: VARCHAR(255), author_id: INT, studio: VARCHAR(255), genre: VARCHAR(30), year_aired: INT, status: ENUM('Aired', 'Airing', 'Unaired'))
PKs: name
FKs: author_id, genre, studio
name -> author_id
name -> studio
name -> genre
name -> year_aired
name -> status
↓
Review is in BCNF and 3NF

Movie (anime_name: VARCHAR(255), duration_min: INT)
PKs: anime_name
anime_name -> duration_min
↓
Movie is in BCNF and 3NF

Series (anime_name: VARCHAR(255), number_of_episodes: INT, number_of_cours: INT)
PKs: anime_name
anime_name -> number_of_episodes
anime_name -> number_of_cours (redundant)
number_of_episodes -> number_of_cours (violates 3NF, decompose)
↓
SeriesLength (number_of_episodes: INT, number_of_cours: INT)
PKs: number_of_episodes
+
Series (anime_name: VARCHAR(255), number_of_episodes: INT)

PKs: anime_name
FKs: number_of_episodes -> SeriesLength(number_of_episodes)

Character (name: VARCHAR(255), anime_name: VARCHAR(255), va_id: INT, height_cm: INT, role: ENUM('Protagonist', 'Antagonist', 'Supporting', 'Other')
PKs: (anime_name, name)
name, anime_name -> va_id
name, anime_name -> height
name, anime_name -> role
↓
Character is in BCNF and 3NF

User (username: VARCHAR(255), country: VARCHAR(50), region: ENUM('Africa', 'Asia', 'Europe', 'North America', 'Oceania', 'South America'), email: VARCHAR(255), year_joined: INT)
PKs: username
username -> country
username -> email
username -> year_joined
username -> region (redundant)
country -> region (violates 3NF, decompose)
↓
Country (name: VARCHAR(50), region: ENUM('Africa', 'Asia', 'Europe', 'North America', 'Oceania', 'South America'))
PKs: country
+
User (username: VARCHAR(255), country: VARCHAR(50), email: VARCHAR(255), year_joined: INT)
PKs: username
FKs: country -> Country(name)

Review (id: INT, user: VARCHAR(255), anime: VARCHAR(255), comment: VARCHAR(1000), rating: INT, would_recommend: BOOLEAN, anime_completed: BOOLEAN)
PKs: id
FKs: user, anime
id -> anime

id -> comment
id -> rating
id -> would_recommend
id -> anime_completed
id -> user
user, anime -> id
↓
Review is in BCNF and 3NF

Creating Tables:

```
CREATE TABLE Birthday (
date DATE PRIMARY KEY,
age INT NOT NULL);

CREATE TABLE Author (
id INT PRIMARY KEY,
name VARCHAR(255) NOT NULL,
birthday DATE,
FOREIGN KEY (birthday) REFERENCES Birthday(date));

CREATE TABLE VoiceActor (
id INT PRIMARY KEY,
name VARCHAR(255) NOT NULL,
birthday DATE,
FOREIGN KEY (birthday) REFERENCES Birthday(date));

CREATE TABLE Studio (
name VARCHAR(255) PRIMARY KEY,
year_of_establishment INT NOT NULL,
is_active BOOLEAN NOT NULL);

CREATE TABLE Genre (
name VARCHAR PRIMARY KEY,
popularity_ranking INT,
description VARCHAR(255) NOT NULL);
```

```
CREATE TABLE Anime (
name VARCHAR(255) PRIMARY KEY,
author_id INT NOT NULL,
studio VARCHAR(255) NOT NULL,
genre VARCHAR(30) NOT NULL,
year_aired INT NOT NULL,
status ENUM('Aired', 'Airing', 'Unaired') NOT NULL,
FOREIGN KEY (author_id) REFERENCES Author(id),
FOREIGN KEY (studio) REFERENCES Studio(name),
FOREIGN KEY (genre) REFERENCES Genre(name));

CREATE TABLE Movie (
anime_name VARCHAR(255) PRIMARY KEY,
duration_min INT,
FOREIGN KEY (anime_name) REFERENCES Anime(name));

CREATE TABLE SeriesLength (
number_of_episodes INT PRIMARY KEY,
number_of_cours INT);

CREATE TABLE Series (
anime_name VARCHAR(255) PRIMARY KEY,
number_of_episodes INT,
FOREIGN KEY (number_of_episodes) REFERENCES
SeriesLength(number_of_episodes));

CREATE TABLE Character (
name VARCHAR(255) NOT NULL,
anime_name VARCHAR(255) NOT NULL,
va_id INT,
height_cm INT,
role ENUM('Protagonist', 'Antagonist', 'Supporting', 'Other') NOT NULL,
PRIMARY KEY (name, anime_name),
FOREIGN KEY (va_id) REFERENCES VoiceActor(id));

CREATE TABLE Country (
name VARCHAR(50) PRIMARY KEY,
region ENUM('Africa', 'Asia', 'Europe', 'North America', 'Oceania', 'South America') NOT NULL));

CREATE TABLE User (
username VARCHAR(255) PRIMARY KEY,
country VARCHAR(50),
email VARCHAR(255),
year_joined INT NOT NULL,
FOREIGN KEY (country) REFERENCES Country(name));

CREATE TABLE Review (
id INT PRIMARY KEY,
user VARCHAR(255) NOT NULL,
anime VARCHAR(255) NOT NULL,
comment VARCHAR(1000),
rating INT NOT NULL,
would_recommend BOOLEAN NOT NULL,
anime_completed BOOLEAN NOT NULL,
FOREIGN KEY (user) REFERENCES User(username),
FOREIGN KEY (anime) REFERENCES Anime(name));
```

Populating the tables:

```
INSERT INTO Birthday (date, age) VALUES
('1973-02-09', 51),
('1992-02-26', 32),
('1980-02-20', 44),
('1986-12-28', 37),
('1991-06-26', 33),
('1988-08-29', 36),
('1999-02-24', 25),
('1959-08-23', 65);

INSERT INTO Author (id, name, birthday) VALUES
(0, 'Makoto Shinkai', '1973-02-09'),
(1, 'Gege Akutami', '1992-02-26'),
(2, 'Sui Ishida', '1986-12-28'),
(3, 'Aka Akasaka', '1988-08-29'),
(4, 'Hiroshi Onogi', '1959-08-23');
INSERT INTO Author (id, name) VALUES
(5, 'Hiromu Arakawa'),
(6, 'Haruichi Furudate');

INSERT INTO VoiceActor (id, name, birthday) VALUES
(0, 'Yuichi Nakamura', '1980-02-20'),
(1, 'Natsuki Hanae', '1991-06-26'),
(2, 'Konomi Kohara', '1992-06-28'),
(3, 'Yurie Igoma', '1999-02-24'),
(4, 'Shinichiro Miki', '1968-03-18');

INSERT INTO Studio (name, year_of_establishment, is_active) VALUES
('CoMix Wave Films', 2007, true),
('MAPPA', 2011, true),
('Pierrot', 1979, true),
('A-1 Pictures', 2005, true),
('Doga Kobo', 1973, true),
('Bones', 1998, true),
('Production I.G', 1987, true);

INSERT INTO Genre (name, popularity_ranking, description) VALUES
('Drama', 1, 'Plot-driven stories focused on realistic characters experiencing human struggle.'),
('Adventure', 2, 'Embark on thrilling journeys filled with excitement, danger, and discovery.'),
('Dark Fantasy', 3, 'Fantasy with gloomy, dark tone or a sense of horror and dread.'),
('Comedy', 4, 'Taking viewers to a better mood and serve as an entertainment.'),
('Sports', 5, 'Training for and participating in a sport takes priority, with the goal of furthering one's athletic abilities to win a competition or achieve some social standing.');

INSERT INTO Anime (name, author_id, studio, genre, year_aired, status) VALUES
('Kimi no Na wa.', 0, 'CoMix Wave Films', 'Drama', 2016, 'Aired'),
('Tenki no Ko', 0, 'CoMix Wave Films', 'Drama', 2019, 'Aired'),
```

```sql
('Jujutsu Kaisen Season 1', 1, 'MAPPA', 'Adventure', 2020, 'Aired'),
('Jujutsu Kaisen 0 Movie', 1, 'MAPPA', 'Adventure', 2021, 'Aired'),
('Tokyo Ghoul: RE', 2, 'Pierrot', 'Dark Fantasy', 2018, 'Aired'),
('Kaguya-sama: Love is War Season 1', 3, 'A-1 Pictures', 'Comedy', 2019, 'Aired'),
('Kaguya-sama wa Kokurasetai: First Kiss wa Owaranai', 3, 'A-1 Pictures', 'Comedy',
2019, 'Aired'),
('Oshi No Ko Season 1', 4, 'Doga Kobo', 'Drama', 2023, 'Aired'),
('Fullmetal Alchemist: Brotherhood', 5, 'Bones', 'Dark Fantasy', 2009, 'Aired'),
('Haikyuu!! Movie: Gomisuteba no Kessen', 6, 'Production I.G', 'Sports', 2024, 'Aired')

INSERT INTO Movie (anime_name, duration_min) VALUES
('Kimi no Na wa.', 104),
('Tenki no Ko', 112),
('Kaguya-sama wa Kokurasetai: First Kiss wa Owaranai', 96),
('Jujutsu Kaisen 0 Movie', 85),
('Haikyuu!! Movie: Gomisuteba no Kessen', 104);

INSERT INTO SeriesLength (number_of_episodes, number_of_cours) VALUES
(24, 2),
(12, 1),
(11, 1),
(64, 5),
(13, 1);

INSERT INTO Series (anime_name, number_of_episodes) VALUES
('Jujutsu Kaisen Season 1', 24),
('Tokyo Ghoul: RE',13),
('Kaguya-sama: Love is War Season 1', 12),
('Oshi No Ko Season 1', 11),
('Fullmetal Alchemist: Brotherhood', 64);

INSERT INTO Character (name, anime_name, va_id, height_cm, role) VALUES ('Satoru
Gojo', 'Jujutsu Kaisen Season 1', 0, 190, 'Protagonist'),
('Ken Kaneki', 'Tokyo Ghoul: RE', 1, 170, 'Protagonist'),
('Chika Fujiwara', 'Kaguya-sama: Love is War Season 1', 2, 154, 'Protagonist'),
('Ruby Hoshino', 'Oshi No Ko Season 1', 3, 158, 'Protagonist'),
('Roy Mustang', 'Fullmetal Alchemist: Brotherhood', 4, 180, 'Protagonist');
```

```
INSERT INTO Country (name, region) VALUES
('Hong Kong', 'Asia'),
('Canada', 'North America'),
('China', 'Asia'),
('Australia', 'Oceania'),
('Dubai', 'Asia');

INSERT INTO User (username, country, email, year_joined) VALUES
('ryanttchan', 'Hong Kong', 'ryan@gmail.com', 2020),
('animegod', 'Canada', 'animegod@gmail.com', 2018),
('animelord', 'China', 'animelord@gmail.com', 2019),
('anonymous', 'Australia', 'anonymous@gmail.com', 2023),
('randomguy', 'Middle East', 'random@gmail.com', 2021);

INSERT INTO Review (id, user, anime, comment, rating, would_recommend,
anime_completed) VALUES
(0, 'ryanttchan', 'Jujutsu Kaisen Season 1', 'Anime of the year', 10, TRUE, TRUE),
(1, 'animegod', 'Tokyo Ghoul: RE', 'Worst anime that I've watched', 0, FALSE, FALSE),
(2, 'animelord', 'Kaguya-sama: Love is War Season 1', 'Overrated anime', 7, FALSE,
TRUE),
(3, 'anonymous', 'Oshi No Ko Season 1', 'Worth the hype', 9, TRUE, TRUE),
(4, 'randomguy', 'Fullmetal Alchemist: Brotherhood', 'Best anime series of all time', 10,
TRUE, TRUE);
```