University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 4

Date: 11/29/2024

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 Description:
Our project created a database containing the information about anime so users of the database can easily find any details of interest. Additionally, our project contains a frontend UI which can allow users to interact with the database in specific ways.

Schema Deviations:
We changed the attributes year_of_establishment and is_active for Studio to not be NOT NULL because we wanted to be able to add anime that were made by studios not already in the database.

Queries:
All of the queries are from DatabaseConnectionHandler.java
2.1.1 INSERT - line 109

```
String query = "INSERT INTO Anime SELECT ?,?,?,?,?,? " +
      "WHERE EXISTS (SELECT * FROM Author WHERE id = ?) " +
      "AND EXISTS (SELECT * FROM Genre WHERE name = ?)";
```

2.1.2 UPDATE - line 165

```
String query = "UPDATE anime SET author_id = ?, studio = ?, genre = ?,
year_aired = ?, status = ? " +
      "WHERE name = ?";
```

2.1.3 DELETE - line 193

```
String query = "DELETE FROM Anime WHERE name = ?";
```

2.1.4 Selection - line 316

```
String query = "SELECT * FROM Anime";
if (!clauseList.isEmpty()) {
   query += " WHERE ";
   for (int i = 0; i < operators.size(); i++) {
      query += clauseList.get(0) + " " + operators.get(i) + " ";
   }
   query += clauseList.get(clauseList.size() - 1);
}
```

2.1.5 Projection - line 352

```
String query = "SELECT " + params + " FROM Anime";
```

2.1.6 Join - line 371

```
String query = "SELECT Studio.name AS studio_name, Anime.name\n" +
      "FROM Studio\n" +
      "JOIN Anime ON Studio.name = Anime.studio\n" +
```

```
      "WHERE Studio.name = '" +
      studio + "'";
```

2.1.7 Aggregation with GROUP BY - line 217

Description: Display how many anime of each genre is in the database.

```
String query = "SELECT genre, COUNT(name) FROM Anime GROUP BY genre";
```

2.1.8 Aggregation with HAVING - line 391

Description: Display the name of the studios that have made more than one anime.

```
String query = "SELECT studio, COUNT(*) AS anime_count\n" +
      "FROM Anime\n" +
      "GROUP BY studio\n" +
      "HAVING COUNT(*) >= 2;";
```

2.1.9 Nested aggregation with GROUP BY - line 239

Description: Display how many anime studios made in the last 4 years (counting from the most recent anime in the database).

```
String query = "SELECT studio, COUNT(name) FROM Anime WHERE year_aired >
(SELECT MAX(year_aired) - 5 " +
      "FROM Anime) GROUP BY studio";
```

2.1.10 Division - line 410

Description: Display the name of anime that have been reviewed by every user.

```
String query = "SELECT anime\n" +
      "FROM Review\n" +
      "GROUP BY anime\n" +
      "HAVING COUNT(DISTINCT user) = (SELECT COUNT(*) FROM User);";
```