COMP 6591: Introduction to Knowledge-Base Systems

Assignment 1

Summer 2022, sections CC

June 27, 2022

Contents

1 General Information

Date posted: Monday, June 27th, 2022.

Date due: Friday, July 8th, 2022, by 23:59.[1].

Weight: 6% of the overall grade.

2 Introduction

This assignment targets basic Knowledge Representation and Queries using Prolog. In this assignment you implement a small database system to represent movie information in both relational as well as Prolog database format.

3 Ground rules

You work on a team of maximum 2 students(including yourself). Each team should designate a leader who will submit the assignment electronically. See Submission Notes for the details. ONLY one copy of the assignment is to be submitted by the team leader. Upon submission, you must book an appointment with the marker team and demo the assignment. All members of the team must be present during the demo to receive the credit. Failure to do so may result in zero credit.

This is an assessment exercise. You may not seek any assistance from others while expecting to receive credit **(You must work strictly within your team).** Failure to do so will result in penalties or no credit.

4 Overview

This assignment you create a simple Knowledge-Base in SQLite and Prolog. The knowledge-base consists of some movie information. The following information is suggested to be included in the assignment.

[1] see submission notes

- Movie titles: including the IMDB id, name, description / plot, etc.
- Directors and Writers
- Cast (names and possibly roles)
- Keywords (a few common keywords per movie)
- Genres
- Languages
- image urls

4.1 Dataset Requirements

While the format and size of the data is open, the following non-functional requirements must be addressed. You may use IMDB datasets (1), any public services, or any other movie databases, to gather the above data.

1. The data-set must include at least 10 titles.
2. The data-set must include at least two movies with more than one language.
3. At least 3 cast members per movie must be included.
4. There must exist a movie with no director or writer info.
5. Include at least 5 keywords per movie. Make sure some movies share one or two keywords.
6. Make sure at least two genres are available.
7. Not all movies have images.

Do not include any TV-series.

5 Your Assignment

Your assignment consist of three parts: (i) Knowledge Representation in Relational Model using SQLite, (ii) Knowledge Representation using Prolog, (iii) Knowledge-Base Queries
It is strongly recommended that you use command line interface (CLI) for both parts.

5.1 Knowledge Representation in Relational Model using SQLite

SQLite (3) is a light-weight relational database engine that enables implementing a small local database in a single file. The first step of the assignment is to create a small database in SQLite that stores the above mentioned movie info.

5.1.1 Database Scheme and DDL

Provide the database schema. Include your DDL queries.

5.1.2 Database Queries

Using the database you created above, write the queries corresponding to the 7 data-set requirements in 4.1 to verify that they are met.

5.1.3 CLI Scripts to export the data into CSV

Write two sample shell scripts to export two of the tables into CSV format [2].

5.2 Knowledge Representation using Prolog

In this section, we want to represent the movie information in Prolog. To do so, you may use the technique you in 5.1.3 or you may create the whole knowledge-base from scratch in Prolog.

5.2.1 Database Facts

Represent the movie information as facts in Prolog.
While many of the facts may be presented using simple terms, make sure your database contains some instances of compound terms and complex structures, including Prolog lists. Here are some suggestions: using lists for movie languages (i.e. `title(..., [en fr], ...).`,

[2] To achieve this, you may use SQLite CLI and IO redirection. An example of a sample script to provide a text output in bash is given in the following:

```
echo -e "select * from titles;\n" | sqlite3 "$dbfile" 2>&1
```

using compound terms for casts (to include name and role), etc. You may however choose and apply the complex structures on any pieces of information, as you feel is applicable.

5.2.2 Queries and Rules

This assignment does not specify any specify Prolog rule to be implemented. Instead, you decide what rules are necessary. See section 5.3 for more details.

5.3 Knowledge-Base Queries

This section defines the queries to be implemented in both **relational** and **deductive** models. For each of the following, write the query/ies in prolog, followed by the corresponding sql in relational model.

1. Search for a movie title.

   Prolog: see `atom_concat` for matching substrings in the title.
2. list all movie titles.

   Prolog: return the movies individually, and all in one list.
3. list all movies played by an actor.
4. list all distinct movies played by two given actors.
5. list all distinct movies played by either of two given actors.
6. list all distinct movies played by some actor within a *list* of years.
7. list movie ids, and titles sorted by id.
8. list movie ids, and titles sorted by title.

6 What to Submit

The whole assignment is submitted by the due date under the corresponding assignment box. It has to be completed by ALL members of the team in one submission file.

Submission Notes

Clearly include the names and student IDs of all members of the team in the submission. Indicate the team leader.

IMPORTANT: You are allowed to work on a team of 2 students at most (including yourself). Any teams of 3 or more students will result in 0 marks for all team members. If your work on a team, ONLY one copy of the assignment is to be submitted. You must make sure that you upload the assignment to the correct assignment box on Moodle. No email submissions are accepted. Assignments uploaded to the wrong system, wrong folder, or submitted via email will be discarded and no resubmission will be allowed. Make sure you can access Moodle prior to the submission deadline. The deadline will not be extended.

Naming convention for the uploaded file: Create one zip file, containing all needed files for your assignment using the following naming convention. The zip file should be called a#_studids, where # is the number of the assignment, and studids is the list of student ids of all team members, separated by (_). For example, for the first assignment, student `12345678` would submit a zip file named `a1_12345678.zip`. If you work on a team of two and your IDs are `12345678` and `34567890`, you would submit a zip file named `a1_12345678_34567890.zip`.

Submit your assignment electronically on Moodle based on the instruction given by your instructor as indicated above: `https://moodle.concordia.ca`

Please see course outline for submission rules and format, as well as for the required demo of the assignment. A working copy of the code and a sample output should be submitted for the tasks that require them. A text file with answers to the different tasks should be provided. Put it all in a file layout as explained below, archive it with any archiving and compressing utility, such as WinZip, WinRAR, tar, gzip, bzip2, or others. You must keep a record of your submission confirmation. This is your proof of submission, which you may need should a submission problem arises.

7 Grading Scheme

Relational Database	15 pts
Verification Queries	5 pts
CSV Export	5 pts
KB is Prolog	15 pts
Prolog Queries	30 pts
SQL-Queries	10 pts
Use of Complex Structures	20 pts

Total: 100 pts.

References

1. IMDB Datasets: `https://www.imdb.com/interfaces/`
2. SWI-Prolog: `https://www.swi-prolog.org/Download.html`
3. SQLite: `https://www.sqlite.org/index.html`
4. SQLite DDL: `https://www.sqlitetutorial.net/sqlite-create-table/`
5. SQLite CLI: `https://www.sqlite.org/cli.html`
6. SQLite to CSV: `https://www.sqlitetutorial.net/sqlite-export-csv/`
7. proSQLite: Prolog File Based Databases via an SQLite Interface:

   `https://link.springer.com/content/pdf/10.1007/978-3-642-45284-0.pdf`
8. proSQLite on SWI-Prolog:

   `https://www.swi-prolog.org/pack/list?p=prosqlite`
9. Sort by key in Prolog:

   `https://www.swi-prolog.org/pldoc/man?predicate=keysort/2`