Project 1: Relational Database System:

Design, Implementation and Query Processing of Wikidata

1. Task introduction

This is a group-based mini research project. Each group could include 2-3 students with clear work division for each member. The task is introduced as follows.

(1) Study and understand the given data source and design some useful queries.

(2) Design a relational database (a set of schemas) for the data, so that you can load the data into database and submit queries you designed and the queries listed in the requirements.

(3) Verify the design of your models. You should optimize your design (e.g., by implementing indexes), so that the database can return results of the queries within reasonable time.

(4) Implement a simple graphical user interface to perform your queries, so that I can check the performance of your design.

(5) Prepare a presentation of 5-10 mins to showcase your preliminary design.

(6) Write a report to describe your design and implementation and hand out all source code (MySQL script as well as GUI code).

2. Specifications

(1) Data

Description:

Wikidata is a free and open knowledge base that can be read and edited by both humans and machines. Wikidata acts as central storage for the structured data of its Wikimedia sister projects including Wikipedia, Wikivoyage, Wikisource, and others. Wikidata also provides support to many other sites and services beyond just Wikimedia projects!

The content of Wikidata is available under a free license, exported using standard formats, and can be interlinked to other open data sets on the linked data web.

For this project, you are provided with the JSON dump of Wikidata's latest data. You can download it from http://adapt.seiee.sjtu.edu.cn/~frank/wikidata-latest-all.json.bz2 and extract it with bzip.

Please carefully read https://www.mediawiki.org/wiki/Wikibase/DataModel/JSON and https://www.mediawiki.org/wiki/Wikibase/DataModel/Primer for the dump file formatting as well as the Wikidata basic data model definitions. In general, everything in the data dump is an entity and there are two kinds of them, item and property. In order to make use of the knowledge, we can make a lot of statements from these entities and their relations.

In Wikidata, we do not actually model the items themselves, but statements about them. We do not say that Berlin has a population of 3,5 M, we say that there is this statement about

Berlin's population being 3,5 M as of 2011 according to the German statistical office. A statement may consist of

- one property (in the example, "population")
- one value (3,5 M)
- optionally one or more qualifiers (in this example, "as of 2011" is one of the qualifiers)
- optionally one or more references (the Germans statistical office)

The property, value, and qualifiers together are also called the claim, which together with any source references forms a statement.

You are required to create a relational database to store the entire data dump. You should carefully design those tables as well to fit the need of queries and large data quantity.

Required queries

1) Given a name, return all the entities that match the name.
2) Given an entity, return all preceding categories (instance of and subclass of) it belongs to.
3) Given an entity, return all entities that are co-occurred with this entity in one statement.
4) Given an entity, return all the properties and statements it possesses.
5) Design and implement a basic Q&A (questioning and answering) system, e.g. if I ask what is the population of China, it should return the correct answer.

These must be supported by your database design efficiently.

(2) Model

A database model is a type of data model that determines the logical structure of a database and fundamentally determines in which manner data can be stored, organized, and manipulated. The most popular example of a database model is the relational model, which uses a table-based format. (See Wikipedia).

You need to design a model for the given data using ER diagrams, and convert the model to schemes of a real relational database (MySQL).

(3) Optimization

The initial version usually cannot satisfy the all the demands. In order to make your model fast or efficient for your proposed queries, you need to revise the design of your model and adjust parameters of database to make your database run faster. The potential optimization can include but is not limited to refining table design, building index, denormalization, etc. However, no matter what kind of optimization you use, you should give the persuasive reason that you really need to do it and it indeed has certain effects. In other words, you need to design solid experiments to demonstrate your design.

(4) Experiment Design

Experiments are designed to demonstrate your ideas. An effective experiment should at least contain the following parts:

1) Hardware specifications
2) Dataset
3) Test queries

4) Initialization scripts
5) Experiment procedure
6) Result analysis

For each of your optimization, you should design an experiment to demonstrate the advantages and disadvantages and whether you will take that optimization as part of your design. Say, an index might improve speed of query. However, it also takes disk space to store them. As a result, if one column is not like to be filtered, there is no need to build index on it.