Andrepont Joel
Fontana Noe
Castells Thibault

Homework 2

Big Data
March 15, 2019

Database Assignment

1 Designing an UML Diagram

We choose to model a database concerning the relations between scientific documents, attributes such as references, and a user driven recommendation system. The relations will allow for the searching of documents as they pertain to certain subject topics, university majors, authors, or even the user/admin based recommendations. We provide the schema for such a system in the UML diagram.

Scientific documents can more specifically be considered to be either books, papers, or online class notes. Each of these has one or more authors, a URL that acts as a primary identifier, a year of publication and a title. The URL of each document, which is unique by definition, is sufficient to access any online resource. Books are published by publisher and have a DOI, (Digital Object Identifier). Class notes are summaries of lectures for university courses prepared by either students or professors. Papers in the system are those that are easily accessible on arxiv.org. We do this in order to facilitate the easy and legal access of high quality scientific papers for students.

References to each document are related by the name of the referencee and that which they reference. Thus one can find what an author references and who they are referenced by easily in our system. We have also considered relations that can occur with documents and encompassed topics. For example, a document may deal with several topics such as applied mathematics, statistics, and computer science. Moreover, a topic can have subtopics and/or be a subtopic.

The user voting system will allow for two standards of quantifying recommendations. The first is a user scoring system, a user can assign a score to a document that varies from one to five. The count of recommendations for a document can also be referenced. Recommendations for documents also reference a major, that is to say, a document may be recommended to computer science, history, or mathematics major. Through this implementation we are able to create a relation that allows for system and user recommendations based on a given major. User recommendations are referenced by the name of the user while system based recommendations may be referenced by the user name "admin".

2 Database Queries

2.1 Database schema for SQLPlus

- usr(usr_id, first_name, email)
- topic(topic_id, title)
- has_subtopic(#topic_id, #topic_id) (in the DB : has_subtopic(#child_id, #parent_id))
- has_topic(#topic_id, #url_)
- document(doc_id, url_, dop, title) where dop stands for "date of publication"
- book(#doc_id, doi, publisher)
- paper(#doc_id, journal)
- notes(#doc_id, course)
- author(author_id, name_)
- written_by(#author_id, #doc_id)
- has_reference(#doc_id,#doc_id) (in the DB : has_reference(#doc_id,#reference))
- major(majorName)
- is_taught(#majorName,#topic_id)
- has_interest(#topic_id,#usr_id)
- recommended_docs(grade, #usr_id, #majorName, #doc_id)

2.2 Queries

2.3 5 Basic Queries

List all subtopics of the topic with ID 2:

```
WITH topic_tree AS
(SELECT *
FROM topic t1, has_subtopic hs
WHERE (t1.topic_id = 2)
AND (hs.parent_id = t1.topic_id)
```

AND (hs.parent_id != hs.child_id))
SELECT t1.title
FROM topic_tree tt, topic t1
WHERE t1.topic_id = tt.child_id;

Topic_tree := $\rho_{\text{topic\_id, title, parent\_id, child\_id}}(\sigma_{\text{topic\_id = 2} \wedge \text{has\_subtopic.parent\_id = topic.topic\_id}}(\text{topic} \times \text{has\_subtopic})$
$\Pi_{topic.title}(\sigma_{topic_tree.topic\_id=topic.topic\_id}(topic\_tree \times topic)$

List titles of all recommendations by a user:
SELECT title
FROM document d1
WHERE d1.doc_id =
(SELECT doc_id
FROM usr, recommended_docs rs
WHERE usr.usr_id = rs.usr_id);
$\Pi_{title}(\sigma_{document.doc\_id=\Pi_{doc\_id}(\sigma_{usr.usr\_id=recommended\_docs.usr\_id}(usr \times recommended\_docs))}(document))$

Name and id of authors who have published at least one book between 1980 and 1990 :
SELECT DISTINCT author_id, name
FROM author a, book b, written_by wb, document d
WHERE a.author_id = wb.author_id AND b.doc_id = wb.doc_id AND d.doc_id = b.doc_id AND dop $\geq$ '1980-01-01' AND dop $\leq$ '1990-12-31' ;
$\Pi_{author\_id,name}(\sigma_{author.author\_id=written\_by.author\_id \wedge book.doc\_id=written\_by.doc\_id \wedge document.doc\_id=book.doc\_id}$
$\wedge dop \geq '1980-01-01' \wedge dop \leq '1990-12-31'(author \times book \times written\_by \times document))$

User ID of the users interested in the topic "Statistics" :
SELECT usr.usr_id
FROM usr, has_interest hi
WHERE usr.usr_id = hi.usr_id AND hi.topic_id = 'Statistics';
$\Pi_{usr.usr\_id}(\sigma_{usr.usr\_id=has\_interest.usr\_id \wedge has\_interest.topic\_id='Statistics'}(usr \times has\_interest))$

Document ID of the documents which concern any topic taught in the major "Physics" :
SELECT DISTINCT d.doc_id
FROM document d, has_topic ht, is_taught it
WHERE d.doc_id = ht.doc_id AND ht.topic_id = is.topic_id AND majorName = 'Physics';

$\Pi_{document.doc\_id}(\sigma_{document.doc\_id=has\_topic.doc\_id \wedge has\_topic.topic\_id=is\_taught.topic\_id \wedge majorName='Physics'}(document \times has\_topic \times is\_taught))$

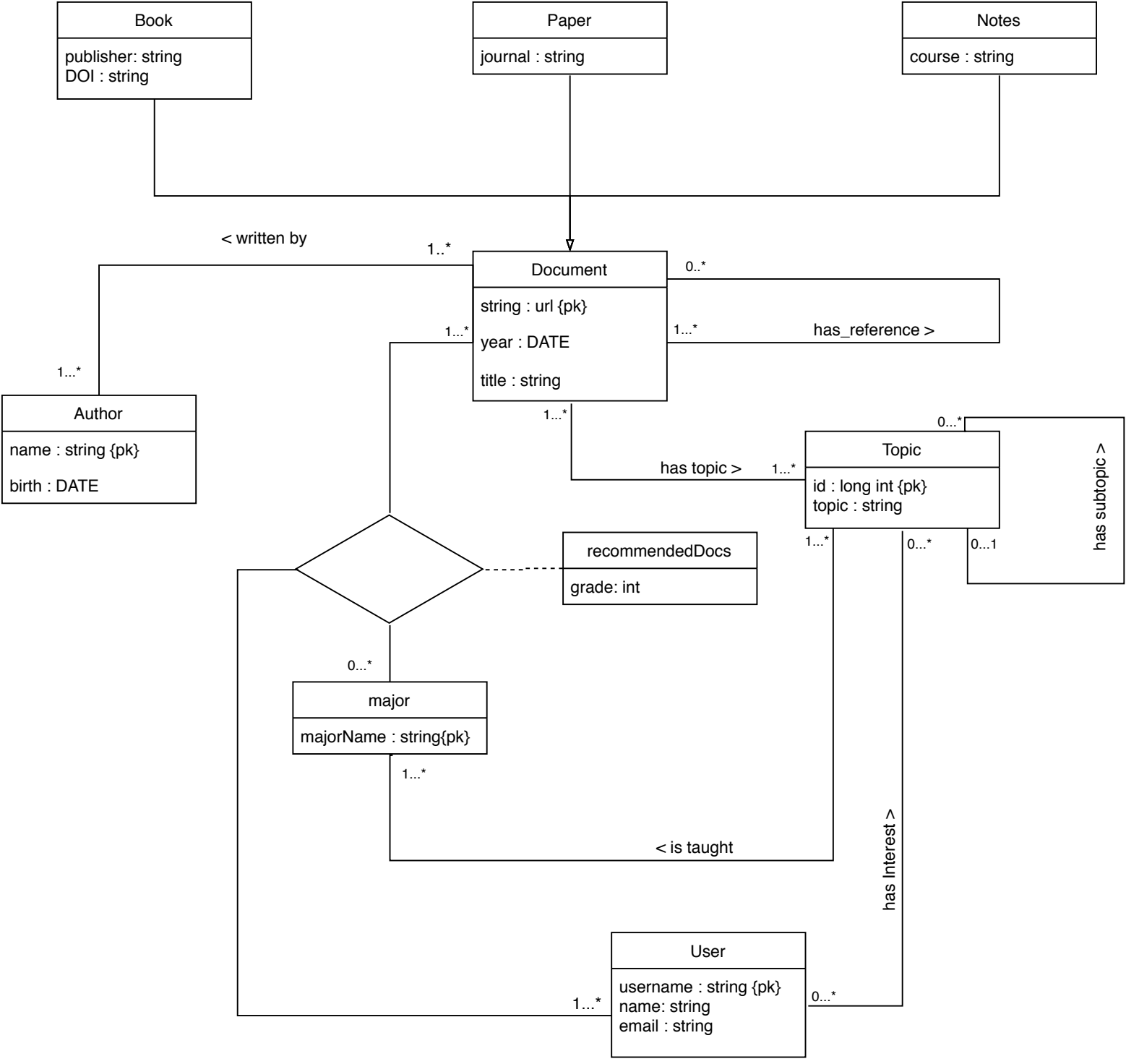
List all authors of all papers:
SELECT a.name_, title
FROM (document d NATURAL JOIN written_by wb), author a
WHERE wb.author_id = a.author_id;
$\Pi_{author.name\_,title}(\sigma_{written\_by.author\_id=author.author\_id}((document \bowtie written\_by) \times author))$

2.4 3 Complex Queries

Name and id of authors that are referenced at least 2 times :
SELECT a.author_id, a.name
FROM author a, written_by wb, has_reference hr
WHERE a.author_id = wb.author_id AND wb.doc_id = hr.doc$_i d$
GROUP BY a.author$_i d, a.name$
HAVING COUNT(hr.reference) $\geq$ 2;

Topics which have at least 3 subtopics :
SELECT DISTINCT topic_id, title
FROM topic
WHERE topic_id IN
(SELECT topic_id
FROM topic t, has_subtopic hs
WHERE t.topic_id = has_subtopic.parent_id
GROUP BY t.topic_id
HAVING COUNT(*) $\geq$ 3);

Document ID of the documents graded 5, for each major :
SELECT rd.majorName, rd.doc_id
FROM recommended_docs rd, major m, document d
WHERE rd.majorName = m.majorName AND rd.doc_id = d.doc_id AND grade = 5
GROUP BY rd.majorName, rd.doc_id;


Book
publisher: string
DOI : string
Paper
journal : string
Notes
course : string
< written by
1..*
Document
string : url {pk}
year : DATE
title : string
0..*
1...*
has_reference >
1...*
Author
name : string {pk}
birth : DATE
1...*
1...*
has topic >
1...*
Topic
id : long int {pk}
topic : string
0...*
has subtopic >
0...1
recommendedDocs
grade: int
1...*
0...*
0...*
major
majorName : string{pk}
1...*
< is taught
has Interest >
User
username : string {pk}
name: string
email : string
1...*
0...*