Database Design and Implementation

CS 645

Course Overview
Instructor

Alexandra Meliou

Teaching assistants

Ardavan Bozorgi
Xi Chen
http://reverie.cs.umass.edu/courses/645

no https!
Paper reading and presentation

Book chapter

Assignment due
This course covers the design and implementation of traditional relational database systems as well as advanced data management systems. The course will treat fundamental principles of databases such as the relational model, conceptual design, and schema refinement. We will also cover core database implementation issues including storage and indexing, query processing and optimization, and transaction management. Additionally, we will cover modern topics and challenges through paper readings and discussions.

Course work will include homework assignments, paper reviews and presentations, a (late) midterm, and a mini, collaborative project.

Prerequisites: an undergraduate-level course on databases or operating systems. 3 credits.

Course Time: Mo We 2:30 pm - 3:45 pm, Hadtruck Lab Add room 124

Instructional team:
Alexandre Meliou
Antawan Bizorgi
Xi Chen
Instructor
Teaching Assistant
Teaching Assistant

Contact: Please use Campuswire for questions to the instructional team

Office hours: TBD

Recommended textbook:
Our recommended textbook is the 3rd Edition of "Database Management Systems" by Ramakrishnan and Gehrke. The textbook is available from Amazon. The lecture notes will be posted online after each class.
https://campuswire.com/p/GB0492DFB

Please sign up and use Campuswire for questions of general interest
We will add you

submissions will be through Gradescope
Database Management Systems
(3rd edition)

http://pages.cs.wisc.edu/~dbbook
Course format

- Mo-Wed, 2:30-3:45pm, Hasbrouck Lab Add 124
- Student-led paper discussions
- Homework assignments
  - 5 individual assignments
  - group mini-project
- Late Midterm
Disclaimer

- The class is actively designed, so there may be changes to the content, structure, and assignment types.

- You are a crucial part of this development.
  - Be vocal about the things you like and the things you don’t like.
  - Feel free to make suggestions.
# Grading

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework assignments</td>
<td>40%</td>
</tr>
<tr>
<td>Paper reviews, presentations, and class</td>
<td>20%</td>
</tr>
<tr>
<td>participation</td>
<td></td>
</tr>
<tr>
<td>Midterm</td>
<td>20%</td>
</tr>
<tr>
<td>Mini-project</td>
<td>20%</td>
</tr>
</tbody>
</table>
Course work

- 5 assignments
  - Practical experience
  - Written problem sets
  - Late policy: 3 grace days, 10% penalty per day after that

- 20 paper presentations
  - role-playing format (more on this in a bit!)

- Reproducibility group project
assignments

- to be done individually
- published on the webpage
- submissions through Gradescope
- first will be released by the end of this week
Students participate in different roles:

- paper author
- academic researcher
- academic historian
- industry practitioner
- peer reviewer
paper reading and discussions

- Students participate in different roles:
  - paper author
  - academic researcher
  - academic historian
  - industry practitioner
  - peer reviewer

A team of 5-6 students prepares a 7-minute presentation highlighting the main contributions and results to deliver in class. Each student will take this role for one paper.
paper reading and discussions

- Students participate in different roles:
  - paper author
  - academic researcher
  - academic historian
  - industry practitioner
  - peer reviewer

Two teams (5-6 students each) prepare 1-2 slides proposing imaginary follow-up work to this paper. Each student will take this role for two papers.
Students participate in different roles:

- paper author
- academic researcher
- academic historian
- industry practitioner
- peer reviewer

Two teams (5-6 students each) prepare 3-4 slides on the context of this paper with respect to prior and subsequent work. Each student will take this role for two papers.
Students participate in different roles:

- paper author
- academic researcher
- academic historian
- industry practitioner
- peer reviewer

Two teams (5-6 students each) prepare 1-2 slides from the perspective of a company or organization considering to use this work in an application or product. Each student will take this role for two papers.
paper reading and discussions

- Students participate in different roles:
  - paper author
  - academic researcher
  - academic historian
  - industry practitioner
  - peer reviewer

Students individually write reviews of the paper, highlighting strengths and weaknesses. Each student will take this role for three papers.
Sign-up opens Feb 9; link will be posted on the website.

You cannot sign up for 2 different roles for the same paper.

You should not erase or move anyone else’s name in the spreadsheet.

You need to sign up in total for:

1 paper as a paper author
2 papers as an academic researcher
2 papers as an academic historian
2 papers as an industry practitioner
3 papers as a scientific reviewer
Learning goals

fundamentals

query languages

relational design

data modeling
Learning goals

- query languages
- relational design
- fundamentals
- data modeling
- storage indexing
- processing optimization
- internals
- transactions
Learning goals

fundamentals
- query languages
- relational design
- data modeling

internals
- storage indexing
- processing optimization
- transactions

expressiveness
- theory
- complexity
- static analysis

fundamentals
- query languages
- relational design
- data modeling

internals
- storage indexing
- processing optimization
- transactions

expressiveness
- theory
- complexity
- static analysis
Learning goals

- fundamentals
  - query languages
  - relational design
  - data modeling
- internals
  - storage indexing
  - processing optimization
  - transactions
- expressiveness
- theory
  - expressiveness
  - static analysis
  - complexity
- advanced
  - security privacy
  - fairness diversity
  - provenance
  - MapReduce Spark
  - relational design
  - data modeling
- security privacy
- fairness diversity
- provenance
- MapReduce Spark
- relational design
- data modeling
- query languages
Why database research is exciting

- One of the broadest areas
- Well integrated theory and systems

A microcosm of CS:

- Languages, operating systems, data structures, theory, algorithms, distributed systems, statistics
What is a DBMS?

large integrated collection of data

- declarative
- efficient querying
- concurrent users
- reliable storage
- access control
what about file systems?

no efficient access
no query language
no specialized buffering
no recovery from failure
no safe concurrent access
Evolution

- Early DBMSs evolved from file systems
- Many small items, many queries and updates
  - e.g., banking, reservations
- Hierarchical / network model
  - users had to think about how data was stored
the relational model

- E. F. Codd, 1970
- data independence
- declarative language
- mathematical foundation
generality & declarativity

Programmers and users do not need to know about storage, indexes, sort orders, concurrent users, etc.

Use logical model, high-level schema

The DBMS determines how to retrieve the data
levels of abstraction

View 1  View 2  View 3

Conceptual Schema

Physical Schema
Example: university DB

- Conceptual schema:
  - Students(sid:integer, name:string)
  - Courses(cid:integer, name:string, semester:string)
  - Professors(fid:integer, name:string)
designing a schema

- Convert to tables and constraints
- Physical design: disk layout, indices

<table>
<thead>
<tr>
<th>sid</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jill</td>
</tr>
<tr>
<td>2</td>
<td>Bo</td>
</tr>
<tr>
<td>3</td>
<td>Maya</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>fid</th>
<th>cid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>621</td>
</tr>
<tr>
<td>1</td>
<td>645</td>
</tr>
<tr>
<td>3</td>
<td>390</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>cid</th>
<th>name</th>
<th>sem.</th>
</tr>
</thead>
<tbody>
<tr>
<td>645</td>
<td>DB</td>
<td>S'18</td>
</tr>
<tr>
<td>621</td>
<td>Soft. Eng.</td>
<td>S'18</td>
</tr>
<tr>
<td>345</td>
<td>DB</td>
<td>F'17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>sid</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Brun</td>
</tr>
<tr>
<td>2</td>
<td>Meliou</td>
</tr>
<tr>
<td>3</td>
<td>Miklau</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>fid</th>
<th>cid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>621</td>
</tr>
<tr>
<td>2</td>
<td>645</td>
</tr>
<tr>
<td>2</td>
<td>345</td>
</tr>
</tbody>
</table>
queries

SELECT C.name
FROM Students S, takes T, Course C
WHERE S.name = "Mary" AND S.sid = T.sid
    AND T.cid = C.cid

find all courses that Mary takes
behind the scenes

SELECT C.name
FROM Students S, takes T, Course C
WHERE S.name = "Mary" AND S.sid = T.sid
    AND T.cid = C.cid
DBMSs and DB research

- Huge industry
  - Large data warehouses
  - Distributed databases
  - Integration
- But: not all data is in a DBMS
  - Scientific data
  - Personal data
  - www
- Data management research has expanded
DB research is broad

- core topics (DB internals, processing, optimization, transactions)
- scientific data
- streaming data
- provenance, security, privacy
- cleaning, matching, integration
- distributed data / querying
- usability, visualization
- crowdsourcing
- ...

...
<table>
<thead>
<tr>
<th>Year</th>
<th>Title</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>Automatic Term Ambiguity Detection</td>
<td>Tyler Baldwin, Yunyao Li, Bogdan Alexe, Ioana Roxana Stanoi. ACL (2) 2013: 804-809</td>
</tr>
<tr>
<td></td>
<td>Constructing consumer profiles from social media data</td>
<td>Mauricio A. Hernández, Kirsten Hildrum, Prateek Jain, Rohit Wagle, Bogdan Alexe, Rajasekar Krishnamurthy, Ioana Roxana Stanoi, Chitra Venkatramani. BigData Conference 2013: 710-716</td>
</tr>
<tr>
<td></td>
<td>Searching for objects driven by context</td>
<td>Bogdan Alexe, Nicolas Heess, Yee Whye Teh, Vittorio Ferrari. NIPS 2012: 890-898</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bogdan Alexe, Mauricio A. Hernández, Kristen Hildrum, Rajasekar Krishnamurthy, Georgia Koutrakia, Meenakshi Nagarajan, Haggal Roitman,</td>
</tr>
</tbody>
</table>
MapMerge: Correlating Independent Schema Mappings

Bogdan Alexe
UC Santa Cruz

Mauricio Hernández
IBM Almaden

Lucian Popa
IBM Almaden

Wang-Chiew Tan
IBM Almaden & UC Santa Cruz

ABSTRACT
One of the main challenges in data integration is to design the mappings between the schemas of two data sources. We present a new method for merging two schema mappings and preserving the links between the two. Our approach is based on a preference model that considers the importance of the links between the two mappings. We evaluate our approach on a real-world data integration scenario.

Preference-aware Integration of Temporal Data

Bogdan Alexe
IBM Almaden
balexexus.ibm.com

Mary Roth
IBM Almaden and UCSC
torkrothus.ibm.com

Wang-Chiew Tan
UCSC
tan@cs.ucsc.edu

ABSTRACT
A complete description of an integration framework is given. The framework is designed to integrate data sources, but rather, it is designed to integrate data sources. Applications based on the framework are presented.

Searching for objects driven by context

Bogdan Alexe
BIWI
ETH Zurich

Nicolas Heess
Gatsby Unit
UCL

Yee Whye Teh
Department of Statistics
University of Oxford

Vittorio Ferrari
School of Informatics
University of Edinburgh

Abstract
The dominant visual search paradigm for object class detection is sliding windows. Although simple and effective, it is also wasteful, unnatural and rigidly hardwired. We propose strategies to search for objects which intelligently explore the space of windows by making sequential observations at locations decided based on previous observations. Our strategies adapt to the class being searched and to the content of a particular test image, exploiting context as the statistical
SELECT a3.fname, a3.lname
FROM Actor a0, Casts c0, Casts c1, Casts c2, Casts c3, Actor a3
WHERE a0.fname = 'Kevin' AND a0.lname = 'Bacon' AND
  c0.pid = a0.id AND c0.mid = c1.mid AND
  c1.pid = c2.pid AND c2.mid = c3.mid AND
  c3.pid = a3.id AND
NOT (a3.fname = 'Kevin' and a3.lname = 'Bacon') AND
  NOT EXISTS (SELECT xc1.pid
               FROM Actor xa0, Casts xc0, Casts xc1
               WHERE xa0.fname = 'Kevin' AND
                 xa0.lname = 'Bacon' AND
                 xa0.id = xc0.pid AND
                 xc0.mid = xc1.mid AND xc1.pid = a3.id)
GROUP BY a3.id, a3.fname, a3.lname;
databases for applications

sloan digital sky survey
Questions?

please give us feedback!