Course resources

- Website:
  http://avid.cs.umass.edu/courses/645

- Moodle:
  https://moodle.umass.edu/
  - Assignment submission (unless otherwise noted)
  - Grades

- Piazza
  - Please use Piazza for questions of general interest to the whole class
  - To discuss personal issues (e.g., grades) please email the instructor and TA.
Recommended textbook

Database Management Systems (3rd edition)

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

- MoWe, 2:30pm–3:45am, Eng. Lab 323
  - Combination of lecture and paper presentations

- Homework deliverables
  - Individual assignments and a group project

- Exams
  - A late midterm (April)
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 address challenges in modern networked information systems, including data mining, provenance, data stream management, and probabilistic databases.

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

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

**Course Time:** Mo We 2:30 pm - 3:45 pm, Engineering Lab 323

**Professor:** Alexandra Meliou  
Contact: ameli [at] cs [dot] umass [dot] edu  
Office Hours: TBA

**Teaching Assistant:** Keen Sung  
Contact: ksun [at] cs [dot] umass [dot] edu  
Office Hours: TBA

**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.
Paper presentations. Click on the icon to see which paper you should read for the corresponding lecture.

<table>
<thead>
<tr>
<th>week</th>
<th>date</th>
<th>day</th>
<th>topic</th>
<th>textbook</th>
<th>slides</th>
<th>deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 1</td>
<td>Jan 23</td>
<td>Mon</td>
<td>Course overview</td>
<td>Ch 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Jan 25</td>
<td>Wed</td>
<td>Project overview and discussion</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 2</td>
<td>Jan 30</td>
<td>Mon</td>
<td>Relational Algebra</td>
<td>Ch 4</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Feb 1</td>
<td>Wed</td>
<td>Relational algebra</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 3</td>
<td>Feb 6</td>
<td>Mon</td>
<td>SQL and Datalog</td>
<td>Ch 5, 24</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>Feb 8</td>
<td>Wed</td>
<td>Views</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Feb 10</td>
<td>Mon</td>
<td>Data Modeling</td>
<td>Ch 3, 24</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Grading

<table>
<thead>
<tr>
<th>Item</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework assignments</td>
<td>25%</td>
</tr>
<tr>
<td>Paper reviews and presentations</td>
<td>15%</td>
</tr>
<tr>
<td>Midterm</td>
<td>25%</td>
</tr>
<tr>
<td>Project</td>
<td>35%</td>
</tr>
</tbody>
</table>
Assignments

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

- 15-18 paper presentations
  - All students need to read the papers
  - Each student presents once in a group of 2-3
  - Each student needs to write a review for 4 papers

- Group project: more next lecture
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
Topics

- **Fundamentals**
  - Query languages
  - Relational design
  - Data modeling

- **Theory**
  - Expressiveness
  - Static analysis
  - Complexity

- **Internals**
  - Storage, indexing
  - Query processing, optimization
  - Transactions

- **Advanced topics**
  - Privacy
  - Probabilistic DB
  - Provenance
  - Parallel DB
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?

- A very large integrated collection of data
- A collection of software designed to store and manage data
  - Declarative interface
  - Efficient querying
  - Concurrent users
  - Reliable storage and recovery
  - Access control
What about file systems?

- Schema is limited
- No efficient way to access
- No specialized buffering
- No query language
- No recovery from failure
- Not safe concurrent access
Evolution

- Early DBMS’s (1960’s) evolved from file systems
- Many small items, many queries and updates
  - Banking
  - Airline reservations
- Hierarchical/network data model
  - Users think about the way data is stored
  - No high level language
The relational model

- E. F. Codd, 1970
  - Data independence
  - Declarative language
  - Mathematical foundation
Generality and 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

- **Views:** how users see the data
- **Conceptual schema** defines logical structure
- **Physical schema** describes files and indices
Example: University database

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

<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>F'12</td>
</tr>
<tr>
<td>621</td>
<td>Soft. Eng.</td>
<td>F'12</td>
</tr>
<tr>
<td>390</td>
<td>DB</td>
<td>S'13</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>fid</th>
<th>sid</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>621</td>
</tr>
<tr>
<td>2</td>
<td>645</td>
</tr>
<tr>
<td>3</td>
<td>390</td>
</tr>
</tbody>
</table>

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

- Find all courses that “Mary” takes

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

- Behind the scenes: the query processor figures out how to answer it efficiently
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

The optimizer chooses an execution plan

\[ \sigma_{\text{name}="\text{Mary"}} \]
\[ \bowtie_{\text{sid}=\text{sid}} \]
\[ \bowtie_{\text{cid}=\text{cid}} \]
\[ \pi_{\text{s.name}} \]
DBMSs and database 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
Trends

- Main memory systems
- Security and privacy
- Uncertain data
- Provenance
- Data cleaning
- Crowdsourcing
- Electronic commerce
- ... 

- Social networking
- Large scale sensing applications
- Bioinformatics
- Streaming data
- Database usability
- Information extraction
- ...
Questions on course organization and content?

Feel free to ask the instructor or the TA

For questions of general interest, please use Piazza, so that other students can benefit from the answers