Computer Science 180: Database Systems

• Arthur M. Keller
• Email: ark@cse.ucsc.edu (best way to reach me)
• Office: Baskin Engineering 153a, (831) 459-1485
• Office Hours: Tuesdays 2:30–3:30pm and by appointment
• Class web page:
  http://www.cse.ucsc.edu/classes/cmps180
• My web page: http://www.cse.ucsc.edu/~ark
• Assignments due most Tuesdays; Project Parts due most Thursdays
• T.A. – T.J. Steed
Textbooks

Required:

  ◆ This book will be available in the bookstore in early October. First chapters handed out today.

Recommended: (one of these)


Available in Library:

• Books in PostgreSQL
Grading

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction Letter and Personal Resume</td>
<td>2%</td>
</tr>
<tr>
<td>Assignments</td>
<td>16%</td>
</tr>
<tr>
<td>Project</td>
<td>32%</td>
</tr>
<tr>
<td>Midterm Exam</td>
<td>15%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>35%</td>
</tr>
</tbody>
</table>

- **Midterm**: Nov. 1 (TH) in class
- **Final**: Dec. 3 (M) 8–11AM in class
- **Introduction Letter and Personal Resume**
  - This assignment includes your name and contact information, your background, relevant courses, and relevant experience. Please tell why you are taking this course and what you expect to get out of it. Tell what makes you unique and interesting. Also include a statement acknowledging that you have read and understand the policies regarding academic dishonesty for this class. This assignment is due **Oct. 2 (T)**.
Schedule

• Today Sep 20 (TH)
  ◆ Intro, Entity-Relationship Model.
  ◆ Read Chapter 1 and Sections 2.1-2.2.

• Next Class: Sep. 25 (T)
  ◆ More Entity-Relationship Model.
  ◆ Read Sections 2.3-2.4.

• Note: Sep. 27 (TH) Class cancelled.
Syllabus

• The background and history of database management systems.
• The fundamentals of using a database management systems.
• Industry standards used for database management systems.
• Theoretical background of the relational model.
• Queries and Updates.
• Transactions and Security.
• Object-oriented, object-relational, semi-structured and XML database systems.
What is a Database Management System?

1. Manages very large amounts of data.
2. Supports efficient access to very large amounts of data.
3. Supports concurrent access to v.l.a.d.
   ◆ Example: bank and its ATM machines.
4. Supports secure, atomic access to v.l.a.d.
   ◆ Contrast two people editing the same UNIX file – last to write “wins” – with the problem if two people deduct money from the same account via ATM machines at the same time – new balance is wrong whichever writes last.
Relational Model

• Based on tables, as:

<table>
<thead>
<tr>
<th>acct#</th>
<th>name</th>
<th>balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>12345</td>
<td>Sally</td>
<td>1000.21</td>
</tr>
<tr>
<td>34567</td>
<td>Sue</td>
<td>285.48</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

• Today used in *most* DBMS's.
The DBMS Marketplace

- Relational DBMS companies – Oracle, Sybase – are among the largest software companies in the world.
- IBM offers its relational DB2 system. With IMS, a nonrelational system, IBM is by some accounts the largest DBMS vendor in the world.
- Microsoft offers SQL-Server, plus Microsoft Access for the cheap DBMS on the desktop, answered by “lite” systems from other competitors.
- Relational companies also challenged by “object-oriented DB” companies.
- But countered with “object-relational” systems, which retain the relational core while allowing type extension as in OO systems.
Three Aspects to Studying DBMS's

1. Modeling and design of databases.
   - Allows exploration of issues before committing to an implementation.

   - SQL = “intergalactic dataspeak.”

3. DBMS implementation.

CS180 = (1) + (2), while (3) is covered in CS277.
Entity/Relationship Model

Diagrams to represent designs.

- **Entity** like object, = “thing.”
- **Entity set** like class = set of “similar” entities/objects.
- **Attribute** = property of entities in an entity set, similar to fields of a struct.
- In diagrams, entity set → rectangle; attribute → oval.
Relationships

- Connect two or more entity sets.
- Represented by diamonds.

![Diagram showing relationships between Students, Taking, and Courses](image-url)
Relationship Set

Think of the “value” of a relationship set as a table.

- One column for each of the connected entity sets.
- One row for each list of entities, one from each set, that are connected by the relationship.

<table>
<thead>
<tr>
<th>Students</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sally</td>
<td>CS180</td>
</tr>
<tr>
<td>Sally</td>
<td>CS111</td>
</tr>
<tr>
<td>Joe</td>
<td>CS180</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Multiway Relationships

Usually binary relationships (connecting two E.S.) suffice.

- However, there are some cases where three or more E.S. must be connected by one relationship.
- Example: relationship among students, courses, TA's. Possibly, this E/R diagram is OK:
• Works in CS180, because each TA is a TA of all students. Connection student-TA is \textit{only} via the course.

• But what if students were divided into sections, each headed by a TA?
  
    
    ◆ Then, a student in CS180 would be related to only one of the TA's for CS180. Which one?

• Need a 3-way relationship to tell.
Students Courses TAs

Students | Courses | TAs
---|---|---
Ann | CS180 | Jan
Sue | CS180 | Pat
Bob | CS180 | Jan
... | ... | ...
Beers-Bars-Drinkers Example

• Our running example for the course.
Multiplicity of Relationships

Many-many  Many-one  One-one

Representation of Many-One
• E/R: arrow pointing to “one.”
  ◆ Rounded arrow = “exactly one.”
Example:
Drinkers Have Favorite Beers
One-One Relationships

Put arrows in both directions.

Design Issue:
Is the rounded arrow justified?

Design Issue:
Here, manufacturer is an E.S.; in earlier diagrams it is an attribute. Which is right?
Attributes on Relationships

- Shorthand for 3-way relationship:
• A true 3-way relationship.
  - Price depends jointly on beer and bar.

• Notice arrow convention for multiway relationships: “all other E.S. determine one of these.”
  - Not sufficiently general to express any possibility.
  - However, if price, say, depended only on the beer, then we could use two 2-way relationships: price-beer and beer-bar.
  - Or better: just make price an attribute of beer.
Converting Multiway to 2-Way

- Baroque in E/R, but necessary in certain “object-oriented” models.
- Create a new connecting E.S. to represent rows of a relationship set.
  - E.g., (Joe's Bar, Bud, $2.50) for the Sells relationship.
- Many-one relationships from the connecting E.S. to the others.
Roles

Sometimes an E.S. participates more than once in a relationship.

• Label edges with *roles* to distinguish.

<table>
<thead>
<tr>
<th>Husband</th>
<th>Wife</th>
</tr>
</thead>
</table>
| $d_1$   | $d_2$
| $d_3$   | $d_4$
| …      | …    |
• Notice *Buddies* is symmetric, Married not.
  ◆ No way to say “symmetric” in E/R.

**Design Question**

Should we replace *husband* and *wife* by one relationship *spouse*?