Logistics

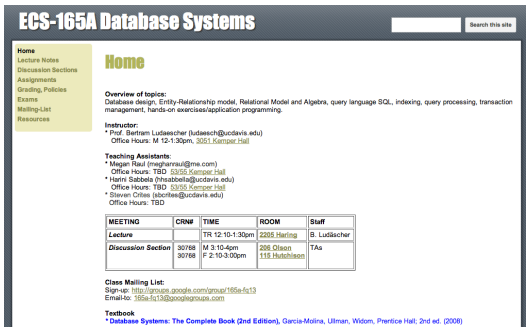
Instructor:

- Prof. Bertram Ludaescher (ludaesch@ucdavis.edu)
  - Office Hours: **M 1-2pm**, 3051 Kemper Hall

Teaching Assistants:

- Meghan Raul (meghanraul@me.com)
  - Office Hours: **F 11am-1pm** 55 Kemper Hall
  - Discussions
    - M 3:10-4pm 206 Olson
    - F 2:10-3pm 115 Hutchison
- Harini Sabbela (hhsabbella@ucdavis.edu)
  - Office Hours: **R 1:50-2:50pm** 55 Kemper Hall
- Steven Crites (sbcrites@ucdavis.edu)
  - Office Hours: TBD 55 Kemper Hall

sites.google.com/site/ecs165afq13



More Logistics

Class page:

- sites.google.com/site/ecs165afq13

Mailing list:

- Sign up: groups.google.com/group/ecs165a-fq13
- Email: ecs165a-fq13@googlegroups.com

Textbook:

- ***Database Systems: The Complete Book (2nd Edition)*** by Garcia-Molina, Ullman, and Widom, Pearson Prentice-Hall, 2008/2009.

Homework Assignment

- Approx. 5 **individual** homework assignments
  - Exception: the last one is typically a **group** project
- Assignment #1:
  - out on Monday (ER Modeling, on paper)
- Other assignments, e.g.
  - relational algebra, SQL, …
- Homework submission:
  - box in Kemper Hall
  - hand-in tool (SQL assignments, using PostgreSQL)
  - also: SmartSite

165A Course Topics

- Conceptual Modeling: ER Diagrams
- Relational Model, Relational Algebra
- SQL (Structured Query Language)
- Integrity Constraints
- Storage structures, Indexing
  - e.g. B+ trees
- Query Processing
  - e.g. simple query rewriting
    - "pushing selections", …
- Transactions & Concurrency Control
  - Basic notions, e.g. serializability
- Additional Topics & Current Trends

165A Course Topics

- Focus is on
  - **Foundations** (relational model, queries, SQL,…)
  - **Practical experience** with SQL
    - We'll use PostgreSQL
      - A "real" (full-featured), scalable DBMS
      - Open source, available @CSIF and @home!
        - » Might also look at MySQL, SQLite, and
        - » Embedded SQL (e.g. with Python)
- **Individual** Assignments (default)
- **Group** Project (at the end)
- If you can't get enough: 165B (more later)

Grading and Policies

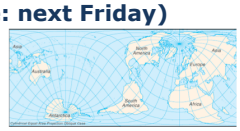
- Grading:
  - Approximately (see web page for details):
    - 40% Homework Assignments
    - 20% Midterm (also individual ;-)
    - 40% Final (and yes: this one too!
- Academic Conduct
  - Be polite
  - Don't cheat
- Ask when in doubt
- Make good use of the mailing-list!

Get your text book

- UC Davis Stores Textbook Rental Program
  - DATABASE SYSTEMS:COMPLETE BOOK
    - Retail Price
      - $175.80 new
      - $131.85 used
    - Rental price
      - $94.93 new
      - $50.98 used

Assignment #1 (Due: next Friday)

Make sure that **your name** is clearly shown on your submission, and that you submit your homework to the right homework box. Grading is based on the **correctness**, **readability**, and **completeness** of your solutions. Unless otherwise noticed, all homework assignments in this class are **individual assignments**!



Problem 1: ER Modeling
You have been assigned to develop a new database for the *Boogle Information Systems Inc.* As a first step you need to design an ER diagram that captures relevant information about the countries of the world.

- A *country* has a number of attributes, including a *name*, (land) *area*, *population*, and *GDP*. A country has also a capital *city*, which itself has a *name*, *population*, and geographical coordinates (*latitude*, *longitude*). With each country there are also a number of spoken *languages* and predominant *religions* associated.
- Countries are located on *continents*, the latter having a *name* and *area*. Any two countries may have a *border* between them of a particular *length*.
- A city can be part of *country*, and can also be situated on an *island*, or at a *lake*, *river*, or *sea*. A lake has a *name*, *area*, maximal *depth*, and *altitude*; a river has a *name*, *length*, and can *flow* in or out of a lake, or into a sea; a sea has a *name*, (maximal) *depth*, and may be connected with another sea. An island has a *name*, *area*, and may be part of one or more countries.

Assignment #1 (cont'd)

Here are the specific tasks you have to address based on the above information:

(a) Design a conceptual database schema (i.e., an Entity-Relationship diagram) for the above scenario based on the given information. Be sure to indicate primary key attributes for each entity type! Also, indicate derived (computed, not stored) attributes, if any.

For each relationship type, clearly specify the semantics of a relationship type using the **(min,max) notation** presented in class. Note that an edge without a (min,max) specification is assumed to have the default specification (0,*). Use the modeling concepts discussed in class. Also, you don't have to specify the data types (domains) for the attributes.

Assignment #1 (cont'd)

(b) Identify (reasonable) constraints that you are **unable** to capture using the standard ER modeling constructs. Do **not** specify trivial domain constraints, e.g., "attribute X is a positive number". Formulate these constraints in plain English. For each constraint, give a brief explanation why you need the constraint and why you cannot express it in the ER diagram, e.g., using cardinalities.

(c) Translate your ER schema into a relational schema: For each table, list all attributes. To specify the tables, use the notation from class, i.e.,

$$\langle table\_name \rangle \; (\langle attribute_1 \rangle, \ldots, \langle attribute_n \rangle)$$

... plus the notation for foreign key constraints (where needed). You do not have to specify any integrity constraints other than foreign key constraints, nor the attribute domains for the tables.

Are there relationship types for which you do not need a table? Explain your answer.

Problem 2: ER Modeling (cont'd)

(a) Which of the attributes in your ER diagram from Problem 1 do change regularly over time (say yearly)? Explain how you would modify your ER diagram to include such time-dependent information. What would that mean for your relational schema?

(b) Pick one time-varying attribute (say GDP) and describe the changes to the ER diagram and to the relational model in detail for that one attribute.