ECS 165B: Database System Implementation

UC Davis, Spring 2011

Acknowledgements: design of course project for this class borrowed from CS 346 @ Stanford's RedBase project, developed by Jennifer Widom, and used with permission. Slides based on earlier ones by Raghu Ramakrishnan, Johannes Gehrke, Jennifer Widom, Bertram Ludaescher, and Michael Gertz.
Welcome to ECS 165B!

Agenda for today’s class:

– Logistics and course overview
– Introduction to the DavisDB project
– Technical material: pages, files, buffers, records (Chapter 13 of textbook)
Course Logistics

Instructor:
Prof. Todd J. Green (green@cs.ucdavis.edu)
Office hours: Tuesdays, 1:00-3:00pm, 3055 Kemper Hall

Teaching assistant:
Armen Khodaverdian (aekhodaverdian@ucdavis.edu)
Office hours: TBD, 055 Kemper Hall

Meeting times:
MWF 9-9:50am, 147 Olson
Discussion section M 1:10-2pm, starting second week of class
More Logistics

Course webpage:

http://www.cs.ucdavis.edu/~green/courses/ecs165b

Class mailing list:

ecs165b-s11@ucdavis.edu

Anyone in the class can post! Don't be shy!

Textbook (same as 165A this year):

What's This Course About?

ECS 165A (last quarter):

   how to use a DBMS

ECS 165B (this quarter):

   how to build a DBMS
What's This Course About?

**Primary focus:** quarter-long implementation project
   - You will build major components of a (simplified) relational database system, **DavisDB**, in C++
   - In teams of 2, delivered in 4 stages

**Secondary focus:** a sampler of further topics in databases
   - XML and semistructured data, data warehousing, ...
   - A taste of database theory

**Meta-focus:** large-scale software engineering (debugging, revision control systems, best coding practices, ...)
How Will This Course Be Graded?

**Basic formula:** project 80% (in 4 parts), closed-book quizzes 20% (2 of them)

No midterm, no final...

...but this will be a difficult, **time-consuming** class!

Code graded for correctness, efficiency, and style

Extra credit for winners of **DavisDB I/O efficiency contest**, as well as the **DavisDB code beauty contest**
Should I Take This Class?

Pre-requisites:
• DBMS fundamentals (ECS 165A)
• C/C++ programming and data structures (ECS 60)
• Ability to work independently and plan ahead
• Significant time to devote to project

What you'll get out of the class:
• Deeper and broader knowledge of DBMS
• Software engineering experience that will pay off once you enter the real world
• Images of the CSIF's soul-crushing mountain scenery posters forever burned into your retinas
Forming Teams

The project will be done in teams of 2.*

Choose your partner carefully! Your grades for the project will be identical. It's up to you to figure out how to share the work and get along. No marriage counseling provided.**

Start thinking about your teams; we’ll form them early next week.

Also: if you are going to drop the class, better to do it before we form teams.

*If you prefer to work alone, you may do so, but you will still be responsible for the same work as the teams, and no special allowance will be made in grading.

**Divorces may be granted on a case-by-case basis.
Some Project Logistics

Team members will coordinate their efforts, and submit their code, via **subversion** (a standard revision control system)

A short (1-2 page), high-level **writeup** will be part of the submitted work

**Standard platform**: the CSIF Linux machines

Automated testing for correctness (~80% of score), manual grading of writeup, design, and code style (~20% of score)

We'll emphasize fundamental skills, such as the proper use of a **debugger**. *(printf won't cut it in this class, just as it doesn't in the real world.)*

More on the logistics when Project Part 1 is assigned
Review: Basic DBMS Architecture

Basic Database Architecture

(Fig. 1.3, p. 20)

File and Access Methods
- Buffer Manager
- Disk Space Manager
- Recovery Manager

Transaction Manager
- Lock Manager
- Concurrency Control

DBMS
- Plan Executor
- Operator Evaluator
- Parser
- Optimizer
- Query Evaluation Engine

Application Front Ends
- SQL Interface
- SQL Commands

Web Forms
- Application Front Ends
- SQL Interface

SQL Commands

DBMS

Query Processing

File and Access Methods

Plan Executor

Operator Evaluator

Parser

Optimizer

Query Evaluation Engine

Transaction Manager

Lock Manager

Concurrency Control

Recovery Manager

Index Files

Data Files

System Catalog
DavisDB Architecture (What's Left Out)

Diagram:
- **DBMS**
  - **Plan Executor**
  - **Operator Evaluator**
  - **Parser**
  - **Optimizer**
  - **File and Access Methods**
  - **Buffer Manager**
  - **Disk Space Manager**

- **Concurrency Control**
  - Transaction Manager
  - Lock Manager

- **SQL Commands**
- **Application Front Ends**
- **SQL Interface**

- **Application Front Ends**: Web Forms

- **Query Execution**
  - (simplified!)
  - Query Evaluation Engine

- **System Catalog**
- **Index Files**
- **Data Files**
Major Components of DavisDB

User

- Command Parser (given)
  - Query Engine (4)
  - System Manager (3)
  - Record Manager (1)

Query Engine (4)

- Indexing (2)
  - Indexing
  - Read/write/scan records
  - Create/delete indexes
  - Get metadata

Indexing (2)

- Get metadata
- Create/delete indexes
- Read/write/scan records
- Indices
- Create files, read/write pages

System Manager (3)

- Startup, DDL, bulk load
- Read/write/scan records
- Data, metadata

Record Manager (1)

- Data, metadata
- Read/write/scan records

Disk Space Manager (given)

- Disk space manager
- Data, metadata

Buffer Manager (given)

- Buffer manager
- Data, metadata

OS File System

- OS file system
- Commands
- Results

OS File System

- OS file system
- Commands
- Results

Disk Space Manager (given)

- Disk space manager
- Data, metadata

Buffer Manager (given)

- Buffer manager
- Data, metadata

OS File System

- OS file system
- Commands
- Results
Important Dates

Project due dates, subject to change:
- Part 1 (record manager): 4/17
- Part 2 (indexing): 5/1
- Part 3 (system manager): 5/15
- Part 4 (query engine): 6/3

Quizzes:
- Quiz #1: 4/27
- Quiz #2: 6/6

Also:
- Warmup homework (serialization and memory management): 4/3
- Mid-quarter course evaluation: 5/4