

# EECS 484: Database Management Systems

Instructor: Kristen LeFevre  
 Email: [klefevre@eecs.umich.edu](mailto:klefevre@eecs.umich.edu)  
 Office Hours: Mondays 12-2, CSE 4705

GSI: Jay Athalye  
 Email: [jathalye@eecs.umich.edu](mailto:jathalye@eecs.umich.edu)  
 Office Hours: Wednesdays 1:30-3:30, CSE 1637

Course Web Page: <http://www.eecs.umich.edu/courses/eecs484>

---

---

---


---

---

---

---

---



## Course Outline

C++ programming intensive!  
 Need to pick up Java  
 Prereq: EECS 281 or equivalent

- GOAL: Basic introduction to database management systems.
- Two perspectives:
  - External (*Database user*)
    - Data models, ER model, relational model, SQL, database design ...
    - Java/JDBC Project: Common platform for building database applications
  - Internal (*Database implementer*)
    - File organizations, access methods, sorting, concurrency control, recovery, ...
    - Minirel Project: Build components of a Relational Database System
- Textbook "Database Management Systems", by Raghu Ramakrishnan & Johannes Gehrke. 3<sup>rd</sup> ed.

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 2

---

---

---


---

---

---

---

---



## Word Locator Project

- Build a database application to search keywords in a collection of files.
- Two phases:
  - Phase 1: Standalone Java application. Individual Assignment. **Available now!**
  - Phase 2: A JDBC/Oracle application. Group Assignment.
- Tutorial this Friday: Intro to Java for C++ programmers

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 3

---

---

---

---

---

---

---

---



## Minirel Project

- Build key components of a DBMS in a series of 2 group assignments
  - Programming intensive, implemented in C++
- Plug these into a skeleton database framework
  - Practice reading and understanding source code
- Simple relational database engine!
- Skilled DB user + DB Implementation + Team and project management skills
- Valuable skills that are highly sought by the industry

1/7/09

EECS 484: Database Management Systems, Kristen LeFevre

4

---

---

---

---

---

---

---

---



## Groups

- Total of three group projects
- You may work with a partner, or choose to work alone
- Same partner for all three projects
  - Unless BOTH partners decide to split up and work alone
- Start looking for partners now!
- Register your group with GSI by January 21

1/7/09

EECS 484: Database Management Systems, Kristen LeFevre

5

---

---

---

---

---

---

---

---



## Discussion Sections

- Not optional!
- Project covered in the discussion section.
- Exams will have project-related questions.

1/7/09

EECS 484: Database Management Systems, Kristen LeFevre

6

---

---

---


---

---

---

---

---



## Course Policies

- Assignment 1 is now posted
  - Java warm up
  - "dating period" for potential project partners
  - Due January 21 @10:30am
- Late policy
  - No late days for individual assignment
  - 3 late days total for group projects
- Academic Honesty: CoE Honor Code for all students. Specifics on course website. Questions? – Ask me first!

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 7

---

---

---


---

---

---

---

---



## Course Policies

- **Disabilities:** I am happy to make reasonable accommodations provided that you take the following steps:
  1. Make sure that you are registered with the Services for Students with Disabilities Office.
  2. Contact me **during the first two weeks of class** to make appropriate arrangements.

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 8

---

---

---


---

---

---

---

---



## Accounts & Resources

- Ctools (Assignments)
  - Login to <http://ctools.umich.edu> using your kerberos and check that you can view the source code and instructions for submitting Assignment 1
- Ctools (Discussion Forum)
  - Send email to GSI if you can't access the forum.
  - This is a shared resource, so use it judiciously.
  - Monitored ~2 days. Best way to get questions answered – Use Discussion Session and Office Hours!

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 9

---

---

---

---

---

---

---

---

## Course Grading

Mid-Term, February 16, 10:30-12:00	20%
Final Exam, April 28, 1:30-3:30	30%
Quizzes: Two 30 minutes quizzes [In class: February 4, March 25]	6%
Java Word Locator Assignment [now posted on the course web page, due Jan. 21]	8%
JDBC Word Locator Assignment [group assignment]	12%
Two Minirel Assignments [group assignment]	24%

**No make up exams or quizzes**

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 10

---

---

---

---

---

---

---

---

---

---

## What Is a DBMS?

- DBMS = Database Management System
- Database: Large, integrated collection of data.
- Models some real-world *enterprise*
  - Entities (e.g., students, courses)
  - Relationships (e.g., Lisa Simpson is taking EECS 484)
- **DBMS**: a software package designed to store and manage databases

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 11

---

---

---

---

---

---


---

---

---

---

## Why Use a DBMS?



- Data independence.
- Efficient access.
- Reduced application development time.
- Uniform data administration.
- Data integrity and security.
- Concurrent access, recovery from crashes.

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 12

---

---

---

---

---

---


---

---

---

---

## Why Study Databases?



- Data is useless without the tools to extract information from the data (queries)
  - "Optimal" pricing of an airline ticket
- Datasets increasing in diversity and volume.
  - Websites, Digital libraries, interactive video, Human Genome project, mobile applications
  - ... need for DBMS exploding
- Databases touch most of CS
  - OS, languages, theory, AI, multimedia, logic, ...

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 13

---

---

---

---

---

---

---

---

## Data Models

- Data model:** a collection of concepts for describing data.
- Schema:** a description of a particular collection of data, using a given data model.
- Relational model:** the most widely-used model today.
  - Data model: Database is a collection of **relations**  
A relation is a table with rows and columns.
  - Every relation has a schema, which describes the columns (also called the fields or attributes).
- Entity-Relationship (ER) model:** A "semantic" data model, i.e. a higher-level more user-intuitive model
  - A (relational) DBMS only understands the relational model  
→ Must translate an ER schema to a relational schema

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 14

---

---

---

---

---

---

---

---

## Example: University Database

- Three tables
  - Students(sid:string, name:string, login:string, age:integer)
  - Courses(cid: string, cname:string, credits:integer)
  - Enrolled(sid:string, cid:string, grade:string)

Students			
sid	name	login	age
13	Lisa	lsimp	40
41	Bart	bart	20

Courses		
cid	cname	Cr.
E-484	EECS484	4
E-584	EECS584	3

Enrolled		
sid	cid	Grade
41	E-484	A-
13	E-584	A+

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 15

---

---

---

---

---

---

---

---

## Levels of Abstraction

```

    graph TD
      ES1[External Schema 1] --> CS[Conceptual Schema]
      ES2[External Schema 2] --> CS
      CS <--> PS[Physical Schema]
      PS --- DB[(Database)]
  
```

- Conceptual schema: defines logical structure (aka logical schema)
- Physical schema: describes the files and indexes used.
- External schema: describe how users see the data, *views*
- Many external schemas, 1 conceptual (logical) schema & 1 physical schema.

- Schemas are defined using Data Definition Language (DDL)
- Data is modified/queried using Data Manipulation Language (DML)

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 16

---

---

---

---

---

---

---

---

## Example: University Database

- Conceptual schema:
  - Students(sid: string, name: string, login: string, age: integer)
  - Courses(cid: string, cname: string, credits: integer)
  - Enrolled(sid: string, cid: string, grade: string)
- Physical schema:
  - Relations stored as unordered files.
  - Index on first column of Students.
- External Schema:
  - View: Course\_info(cid: string, enrollment: integer)
  - View: Class\_rank(sid: string, gpa: real, rank: integer)

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 17

---

---

---

---

---

---

---

---

## Data Independence

- Applications insulated from data format and storage details.
- Logical data independence: Protection from changes in *logical* structure of data.
  - External / Conceptual schemas
- Physical data independence: Protection from changes in *physical* structure of data.
  - Conceptual / Physical schemas

**One of the key benefits of using a DBMS**

**Other key benefits:**

- ✓ declarative query processing
- ✓ transactions

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 18

---

---

---

---

---

---

---

---

## Transactions

```

    graph TD
      B1[Bank Balance : $100] --> R1[Read Balance: $100]
      B1 --> R2[Read Balance: $100]
      R1 --> S1[Sufficient funds?]
      R2 --> S2[Sufficient funds?]
      S1 -- Yes --> P1[Pay $25]
      S2 -- Yes --> P2[Pay $25]
      P1 --> NB1[New balance: $75]
      P2 --> NB2[New balance: $75]
      NB1 --> B2[Bank Balance : $75]
      NB2 --> B2
  
```

- Inconsistency caused by interleaving actions of different user programs
- DBMS provides the illusion of a "single-user" system
  - Key concept: **Transaction**, an atomic sequence of R/W
  - **Concurrency control**, **transaction management**

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 19

---

---

---

---

---

---

---

---

---

---

## Lots of People use DBMS ...

- DBMS vendors
- DB application programmers
  - E.g. smart webmasters
- *Database administrator (DBA)*
  - Designs logical /physical schemas
  - Handles security and authorization
  - Data availability, crash recovery
  - Database tuning as needs evolve

**Must understand how a DBMS works!**

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 20

---

---

---

---

---

---

---

---

---

---

## Summary

- DBMS used to maintain, query large datasets.
- Benefits include recovery from system crashes, concurrent access, quick application development, data integrity and security.
- Levels of abstraction give data independence.
- DBAs hold responsible jobs and are **well-paid!**
- DBMS R&D is one of the most exciting areas in CS.

1/7/09 EECS 484: Database Management Systems, Kristen LeFevre 21

---

---

---

---

---

---

---

---

---

---



## Intro to Java

- Assignment 1 – Word Locator project in Java
- Read “Java for C++ Programmers” handout for Friday’s discussion

1/7/09

EECS 484: Database Management Systems, Kristen LeFevre

22

---

---

---

---

---

---

---

---



## Java - Hello World!

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

1/7/09

EECS 484: Database Management Systems, Kristen LeFevre

23

---

---

---

---

---

---

---

---

## Java Types

### Primitive Types

<b>boolean</b>	bool in C++
<b>char</b>	holds 1 character
<b>byte</b>	8-bit signed int
<b>short</b>	16-bit signed int
<b>int</b>	32-bit signed int
<b>long</b>	64-bit signed int
<b>float</b>	Floating point
<b>double</b>	Double precision floating point

### Reference Types

```
array int [] a;
a = new int [8];
a[0] = 11;

class class Circle {
    private int x, y, radius;

    Circle(int x, int y,
           int radius) {
        this.x = x;
        this.y = y;
        this.radius = radius;
    }
}
```

**Pass by value**

**Arrays and classes are really pointers!**

---

---

---

---

---

---

---

---

```
class Pair {int x, y;}

void f() {
    int n = 1;
    Pair p = new Pair();
    p.x = 2; p.y = 3;
    System.out.println(n); // prints 1
    System.out.println(p.x); // prints 2
    g(n,p);
    System.out.println(n); // still prints 1
    System.out.println(p.x); // prints 100
}

void g(int num, Pair ptr) {
    System.out.println(num); // prints 1
    num = 17; // changes only the local copy
    System.out.println(num); // prints 17
    System.out.println(ptr.x); // prints 2
    ptr.x = 100; // changes x of the caller's Pair!
    ptr = null; // changes only the local ptr
}
```

---

---

---

---

---

---

---

---