What is This Class about?

• Data is everywhere!
• Managing data is critical:
  – scientific discoveries
  – online services (social networks, online retailers)
  – decision making
• Databases are the core technology
• In this class:
  – How do we use a database?
  – How to we build a database?
DATA LANDSCAPE

The image shows a data landscape infographic with categories such as Vertical, Consumer, Business Intelligence, Analytics and Visualization, Ad/Media, Data As A Service, Infrastructure, and Technologies. The infographic is titled "The Big Data Landscape" and dated January 2014. It includes logos of various companies and technologies related to big data and analytics.
COURSE LOGISTICS
TEACHING STAFF

• **Instructor:** Paris Koutris
  – [paris@cs.wisc.edu](mailto:paris@cs.wisc.edu)
  – Office hours:
    • *Monday* 9:30-10:30 am (after class)
    • *Thursday* 1:00-2:00 pm

• **Teaching Assistant:** Udip Pant
  – Office hours: *Wednesday* 3:00-4:00 pm
ABOUT ME

• undergrad in Athens, Greece
• Ph.D. in University of Washington (the other UW)
• at UW-Madison since Fall 2015!

Research Interests
• parallel processing of big data
• data pricing
• uncertainty in data management
COURSE FORMAT

• Lectures M+W 8:00-9:15 am
• Discussions F 8:00-9:15 am

• 4 Projects (in groups of 1-3)
• 2 Homework assignments (individual)

• Midterm Exam
• Final Exam
COMMUNICATION

  - Announcements
  - Lectures
  - Assignments

- **Mailing List**: compsci564-1-f16@lists.wisc.edu

- **Piazza**: Canvas or login to [https://piazza.com/wisc/fall2015/compsci564_fa15/home](https://piazza.com/wisc/fall2015/compsci564_fa15/home)
  - Questions
  - Discussions
TEXTBOOK

• Database Management Systems (3d edition)
  Raghu Ramakrishnan and Johannes Gehrke
**Prerequisites**

- Data structures and algorithm background necessary!
  - *CS 367* is a must

- For the *projects*
  - programming-heavy
  - C++ will be used for the database internals
GRADING

- Projects (4): 5% + 7% + 8% + 13%
- Homework (2): 6% + 6%
- Midterm: 20%
- Final: 35%
PROJECTS

- **Project #1** [5%]
  - Designing a database (ER-model, schema)

- **Project #2** [7%]
  - Querying a database (SQL)

- **Project #3** [8%]
  - Implementing a database (Buffer management)

- **Project #4** [13%]
  - Implementing a database (B+ trees)
HOMEWORKS

• Homework #1 [6%]
  – Schema design
  – Relational Algebra

• Homework #2 [6%]
  – Database Internals
EXAMS

• Midterm Exam
  – *when*: October 26 (8:00-9:15 am)
  – *where*: in class

• Final Exam
  – *when*: December 19 (7:45-9:45 am)
  – *where*: TBD
**Database**

*What is a database?*
- A collection of files storing related data

*What are some examples of databases?*
- payroll database
- Amazon’s product information
- bank account database
DBMS

What is a Database Management System (DBMS)?

• A program written by someone else that allows us to manage efficiently a large database and allows data to persist over long periods of time

What are examples of DBMSs?

• SQL Server, Microsoft Access (Microsoft)
• DB2 (IBM)
• Oracle
• MySQL, PostgreSQL, SQLite
**Example: Online Bookstore**

- What data do we need to store?
- How will we use the data stored?
Example: Online Bookstore

• What **functionality** do we want to support?
  – efficient querying
  – multiple users
  – recovery after crashes
  – security, user authorization
DATA STORAGE

• Data stored for a long period of time (persistent data): the data outlives the application
• Large amounts of data (100s of GB)
• User authorization on which data to access
• Protection from system crashes
 Queries & Updates

• Store and retrieve data in an efficient way
  – Organize data on disk
  – Index data for faster access
• Make efficient use of memory hierarchy
• Safely allow concurrent access to the data
• Allow the data to be updated safely
**Concurrency Control**

- Alice and Bob have the same number for a gift certificate of **$100** at the online bookstore
  - Alice @ her office orders ”Book A” for **$30**
  - Bob @ his office orders ”Book B” for **$60**

- Questions:
  - What is the ending credit?
  - What if second book costs **$80**?
  - What if system crashes?
CRASH RECOVERY

• How do we make sure no data is lost after the system has crashed?
Schema Change

• Say that we need to add a new field to books
  – entails changing file formats
  – need to rewrite virtually all applications
What Can a DBMS Do?

- All the above!!
- Automate a lot of boring operations on data
  - don’t have to program over and over
  - can write complex data manipulations in just a few lines
- Make execution very fast
  - scales up to very large data sets
- Make concurrent access/modification possible
  - many users can use the data at the same time
**KEY CONCEPTS**

- **Data model**: abstraction that describes the data
- **Schema**: describes a specific database using the “language” of the data model
- **Query Language**: high-level language to allow a user to pose queries easily
  - Declarative languages (SQL)
- **Query optimizer/compiler**: code that evaluates the query efficiently
DATA INDEPENDENCE

The application does not change when the underlying data structure or storage changes

- **Physical independence**: can change how data is stored on disk without maintenance to applications
- **Logical independence**: can change schema without affecting applications
## Relational Model

- The data is stored in **tables** (*relations* in the mathematical sense)
- A database is a set of tables

<table>
<thead>
<tr>
<th>name</th>
<th>price</th>
<th>author</th>
<th>hardcover</th>
</tr>
</thead>
<tbody>
<tr>
<td>007456</td>
<td>The Da Vinci Code</td>
<td>Dan Brown</td>
<td>yes</td>
</tr>
<tr>
<td>909405</td>
<td>Ender’s Game</td>
<td>Orson Scott Card</td>
<td>no</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
Querying the Data

• SQL or other declarative languages
• Example: find all books written by Dan Brown

```
SELECT * 
FROM books B
WHERE B.author = "Dan Brown"
```
**Query Processor**

- **Optimizer**: what is the best imperative execution plan for the given query?
- **Evaluation**: execute the plan as efficiently as possible
**PEOPLE**

- **DB application developer**: writes programs that query and modify data
- **DB designer**: establishes schema
- **DB administrator**: loads data, tunes system, keeps whole thing running
- **Data analyst**: data mining, data integration
- **DBMS implementor**: builds the DBMS
COURSE CONTENT

Design & Modeling Data:
• Entity-Relationship model
• Relational model, schema normalization

Querying the Data:
• Relational algebra
• SQL

Database Internals:
• Data storage, file organization, buffer management
• Indexes
• Relational operators, query optimization

Transactions, Big Data
INTERESTED IN MORE?

CS 764
• gory details on how a DBMS works
• transactions/concurrency/internals

CS 774
• newer types of data and how to manage them
• data integration

CS 784
• the theory behind databases