EECS 647: Introduction to Database Systems

Instructor: Luke Huan

Spring 2009
Queries for Today

- What is a database?
- What is a database management system?
- Why take a database course?
- Who will teach?
- How to take the class?
- Preview of class contents
What: Database Systems Then
What: Database Systems Today

Related Searches: database, fundamentals of database systems, software engineering.


Books: See all 17664 items

Buy new: $117.95 $99.08 Used & new from $84.00

Get it by January 23, 2007, if you order in the next 5 hours and 22 minutes.

**What:** Database Systems Today

![Bank of America Online Banking Interface](image)

**John Jones - Personal Accounts**
Monday, January 12, 2004

**I want to...**
- View my account details
- Set up a bill payment
- Pay a bill
- Transfer funds between accounts

**Account**
- Interest Checking - 3858
- Regular Savings - 0490
- Fixed Term CD - 2747
- Fixed Term IRA - 4128

**Announcements**
What: Database Systems Today
So... **What** is a Database?

- A *database* is a very large, integrated collection of data.
- Data is a group of facts that can be recorded and have an implicit meaning.
- Typically a database is used to model a real-world “enterprise” (or a *miniworld*)
  - **Entities** (e.g., *basketball teams, games*)
  - **Relationships** (e.g. *KU’s basketball team won* <you name it> last week)
- Might surprise you how flexible this is
  - **Web search:**
    - Entities: words, documents
  - **P2P filesharing:**
    - Entities: words, filenames, hosts
    - Relationships: word in filename, file available at host
Databases make these folks happy ...

- End users in *many* fields
  - Business, education, science, …
- DB application programmers
  - Build data entry & analysis tools on top of DBMSs
  - Build web services that run off DBMSs
- Database administrators (DBAs)
  - Design logical/physical schemas
  - Handle security and authorization
  - Data availability, crash recovery
  - Database tuning as needs evolve
- DBMS vendors, programmers
  - Oracle, IBM, MS …

...must understand how a DBMS works
What is a Database Management System?

- A **Database Management System (DBMS)** is a collection of programs that enable users to create and maintain databases.
  - store, manage, and access data in a databases.
- Typically this term is used narrowly
  - Relational databases with transactions
    - E.g. Oracle, DB2, SQL Server
  - Mostly because they predate other large repositories
    - Also because of technical richness
  - When we say **DBMS** in this class we will usually follow this convention
    - But keep an open mind about applying the ideas!
**What: Is the WWW a DBMS?**

- Fairly sophisticated search available
  - Crawler indexes pages on the web
  - Keyword-based search for pages
- But, currently
  - Data is mostly unstructured and untyped
  - Search only:
    - Can’t modify the data
    - Can’t get summaries, complex combinations of data
  - Few guarantees provided for freshness of data, consistency across data items, fault tolerance, …
  - Web sites typically have a (relational) DBMS in the background to provide these functions.
What: Is a File System a DBMS?

- Thought Experiment 1:
  - You and your project partner are editing the same file.
  - You both save it at the same time.
  - Whose changes survive?
  - A) Yours
  - B) Partner's
  - C) Both
  - D) Neither
  - E) ???

Q: How do you write programs over a subsystem when it promises you only "???"?

A: Very, very carefully!!
Current Commercial Outlook

- A major part of the software industry:
  - Oracle, IBM, Microsoft
  - also Sybase, Informix (now IBM), Teradata
  - smaller players: java-based dbms, devices, OO, …
- Lots of related industries
  - data warehouse, document management, storage, backup, reporting, business intelligence, ERP, CRM, app integration
- Traditional Relational DBMS products dominant and evolving
  - adapted for extensibility (user-defined types), native XML support.
- Open Source coming on strong
  - MySQL, PostgreSQL, Apache Derby, BerkeleyDB, Ingres, EigenBase
- And of course, the other “database” technologies
  - Search engines, P2P, etc.
What database systems will we cover?

- We will be try to be broad and touch upon
  - Relational **DBMS** (e.g. Oracle, SQL Server, DB2, Postgres)
  - “Semi-structured” **DB systems** (e.g. XML repositories like Xindice)
  - Data mining: transfer data to knowledge!

- Starting point
  - We assume you have used web search engines
  - We assume you don’t know relational databases
    - Yet they pioneered many of the key ideas
  - So focus will be on relational DBMSs
    - With frequent side-notes on search engines, XML issues
Why take this class?

A. Database systems are at the core of CS
B. They are incredibly important to society
C. The topic is intellectually rich
D. It isn’t that much work
E. Looks good on your resume

Let’s spend a little time on these
Who?

- Instructors
  - Prof. Luke Huan, EECS
  - jhuan@eeecs.ku.edu

- Prof. Office Hours:
  - Eaton Hall 2034, M, W 4:15-5:15pm
How? Workload

- Projects with a “real world” focus:
  - Build a web-based application w/PostgreSQL and your favorite internet programming language (PHP, JAVA, PERL, just naming a few)
  - Self-initiated projects with similar scope
- Homework assignments and quizzes
- Exams – 2 Midterms & 1 Final
- Projects to be done in groups of 2
  - Pick your partner ASAP
- The course is “front-loaded”
  - most of the hard work is in the first half
How? Administrative

  - Available at KU Student Stores.
- Class website - people.eecs.ku.edu/~jhuan/EECS647_S09
  - read it regularly
  - Please include eecs 647 in your mail to the instructor (for quick response)
- Grading, hand-in policies, etc. is on syllabus

* The textbook will be available at the KU bookstore today or tomorrow
Academic Misconduct

- We take academic misconduct very seriously.
- We encourage student to work together for doing homework and projects, but each student should write down their own solution.
  - You are absolutely encouraged to discuss with your classmates about your homework assignments, programming exercises, and final projects.
  - You are responsible for all your works.
- For homework assignments and projects resulted from discussion, please always acknowledge other people’s contribution by including a sentence at the beginning of the hand-in saying “I discussed the homework (project) with XXX (include more names if necessary)”. There is absolutely NO penalty for doing so.
Agenda for the rest of today

- Today: a preview of the class contents:
  - ER model
  - Logical database design
  - File system
  - Transaction
  - Data mining
  - Database in the future
  - Relational model
  - SQL
  - Query processing
  - XML

- Next Monday
  - the Entity-Relationship model

- Today’s lecture is from Chapter 1 in R & G
- Read Chapter 2 for next class.
Agenda …

- Design a Database
  - ER model
  - Relational model
  - Database logical design
  - Query with SQL
Data Models Describe Data

- A **data model** is a collection of concepts for describing data.

- Different levels of data models
  - Conceptual (semantic) data model uses data (facts) to describe a physical world
  - Representational model describes data in database management systems
  - Physical data model describe how data is stored in files in computers.
An ER Model for a mini-World

- **Entity**: Real-world object distinguishable from other objects. An entity is described (in DB) using a set of attributes.

- **Relationship**: Association among two or more entities. E.g., John works in Pharmacy department.
Relational Model Describes Data

- Relational model is the most popular representational data model that are used by current commercial DBMS.
- Relational model describes data using tables

<table>
<thead>
<tr>
<th>SSN</th>
<th>Name</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456789</td>
<td>John Smith</td>
<td>30000</td>
</tr>
<tr>
<td>333444555</td>
<td>Franklin Wong</td>
<td>40000</td>
</tr>
<tr>
<td>453453453</td>
<td>Joyce English</td>
<td>25000</td>
</tr>
</tbody>
</table>

- Cardinality = 3, degree = 3, all rows distinct
- Do all columns in a relation instance have to be distinct?
Logical Database Design

**Employees**
- **ssn**
- **name**
- **Salary**

**Departments**
- **number**
- **name**
- **manager**

**Works_For**

<table>
<thead>
<tr>
<th>SSN</th>
<th>Name</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456789</td>
<td>John Smith</td>
<td>30000</td>
</tr>
<tr>
<td>333444555</td>
<td>Franklin Wong</td>
<td>40000</td>
</tr>
<tr>
<td>453453453</td>
<td>Joyce English</td>
<td>25000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DNumber</th>
<th>DName</th>
<th>Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>research</td>
<td>123456789</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SSN</th>
<th>Dnum</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456789</td>
<td>5</td>
</tr>
<tr>
<td>333444555</td>
<td>5</td>
</tr>
<tr>
<td>453453453</td>
<td>5</td>
</tr>
</tbody>
</table>
### Logical Database Design (Cont.)

<table>
<thead>
<tr>
<th>SSN</th>
<th>Name</th>
<th>Salary</th>
<th>DNum</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456789</td>
<td>John Smith</td>
<td>30000</td>
<td>5</td>
</tr>
<tr>
<td>333444555</td>
<td>Franklin Wong</td>
<td>40000</td>
<td>5</td>
</tr>
<tr>
<td>453453453</td>
<td>Joyce English</td>
<td>25000</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DNumber</th>
<th>DName</th>
<th>Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>research</td>
<td>123456789</td>
</tr>
</tbody>
</table>

**Diagram:**
- Employees
  - `ssn`
  - `name`
  - `Salary`
- Works_For
- Departments
  - `number`
  - `manager`
Logical Database Design (Cont.)

```
<table>
<thead>
<tr>
<th>SSN</th>
<th>Name</th>
<th>Salary</th>
<th>DNumber</th>
<th>DName</th>
<th>Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456789</td>
<td>John Smith</td>
<td>30000</td>
<td>5</td>
<td>research</td>
<td>123456789</td>
</tr>
<tr>
<td>333444555</td>
<td>Franklin Wong</td>
<td>40000</td>
<td>5</td>
<td>research</td>
<td>123456789</td>
</tr>
<tr>
<td>453453453</td>
<td>Joyce English</td>
<td>25000</td>
<td>5</td>
<td>research</td>
<td>123456789</td>
</tr>
</tbody>
</table>
```
**SQL: Query Information from a Database**

- **Query 1:** Retrieve the salary of Franklin Wong

  **Q1:**
  
  ```sql
  SELECT SALARY
  FROM EMPLOYEE
  WHERE NAME='Franklin Wong'
  ```

  **Answer:** 40000

---

<table>
<thead>
<tr>
<th>SSN</th>
<th>Name</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456789</td>
<td>John Smith</td>
<td>30000</td>
</tr>
<tr>
<td>333444555</td>
<td>Franklin Wong</td>
<td>40000</td>
</tr>
<tr>
<td>453453453</td>
<td>Joyce English</td>
<td>25000</td>
</tr>
</tbody>
</table>
Agenda …

- Design a Database Management System
  - File systems
    - Disk storage
    - Indexing
  - Query processing
  - Transactions
    - Concurrency
    - Recovery
Components of a Disk

- The platters spin (say, 90rps).
- The arm assembly is moved in or out to position a head on a desired track. Tracks under heads make a cylinder (imaginary!).

![Diagram of disk components](image-url)
Indexing is the Key to Speed up Disk Operations

- “Find all employees whose salary > $25000”
  - If data is in sorted file, do binary search to find first such employee, then scan to find others.
  - Cost of binary search can be quite high.
- Simple idea: Create an ‘index’ file.

Can do binary search on (smaller) index file!
Concurrent execution of user programs

- Why?
  - Utilize CPU while waiting for disk I/O
    - (database programs make heavy use of disk)
  - Avoid short programs waiting behind long ones
    - e.g. ATM withdrawal while bank manager sums balance across all accounts
Key concept: Transaction

- an atomic sequence of database actions (reads/writes)
- takes DB from one consistent state to another
Example

- Here, **consistency** is based on our knowledge of banking “semantics”
- In general, up to writer of transaction to ensure transaction preserves consistency
- DBMS provides (limited) automatic enforcement, via **integrity constraints**
  - e.g., balances must be $\geq 0$

Transaction
```
checking: $200
savings: $1000
```

```
transfer $100 from Saving to checking
```

```
checking: $300
savings: $900
```
Agenda …

- Advanced topics
  - XML
  - Data mining
    - Association
    - Classification
    - Clustering
  - Database in the future
Acknowledgement

- We thank Dr. Jun Yang at the Duke University to provide his class slides of database management systems.
- Some slides used in this class are downloaded from UC Berkeley Introduction to Database Systems class at http://cs186.declarativity.net/