§0 Introduction

Qin Zhang
Self introduction: my research interests

- **Algorithms for Big Data:**
  - streaming/sketching algorithms;
  - algorithms on distributed data;
  - data structures;
  - database algorithms;

- **Complexity:**
  - communication complexity.

I publish on top database conferences, e.g.,

- **The Communication Complexity of Distributed Set-Joins**
  2015 ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS’15)

- **Robust Set Reconciliation**
  2014 ACM SIGMOD International Conference on Management of Data (SIGMOD’14)
Write down answers on the piece of paper.

What is database?
Write down answers on the piece of paper.

What is database?

What are the typical topics of a database course?
Write down answers on the piece of paper.

What is database?

What are the typical topics of a database course?

What are the key aspects to build an efficient database?
Build a database

How to represent data?
(depending on the data AND the desired functionalities)
The big picture

Build a database

How to represent data?
(depends on the data AND the desired functionalities)

How to operate on data to answer queries?
The big picture

Build a database

How to represent data?
(dependents on the data AND the desired functionalities)

How to operate on data to answer queries?

How to speed up the operations?
The big picture

Build a database

How to represent data?
( depends on the data AND the desired functionalities )

How to operate on data to answer queries?

How to speed up the operations?

How to make the database reliable yet efficient?
The big picture

Build a database

How to represent data?
(depends on the data AND the desired functionalities)

How to operate on data to answer queries?

How to speed up the operations?

How to make the database reliable yet efficient?

Make a car

Layout design / Modeling

Design and implement physical functionalities

Tune the performance

Consider various environmental stresses.
How to represent the data?

<table>
<thead>
<tr>
<th>Title</th>
<th>Year</th>
<th>Length</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Wars</td>
<td>1977</td>
<td>124</td>
<td>color</td>
</tr>
<tr>
<td>Mighty Ducks</td>
<td>1991</td>
<td>104</td>
<td>color</td>
</tr>
<tr>
<td>Wayne’s World</td>
<td>1992</td>
<td>95</td>
<td>color</td>
</tr>
</tbody>
</table>

OR?

.Movie
  - Sci-Fi
    - Star Wars, 1977
  - Cartoon
    - Mighty Ducks, 1991
  - Comedy
    - Wayne’s World, 1992
How to represent the data?

<table>
<thead>
<tr>
<th>Title</th>
<th>Year</th>
<th>Length</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Wars</td>
<td>1977</td>
<td>124</td>
<td>color</td>
</tr>
<tr>
<td>Mighty Ducks</td>
<td>1991</td>
<td>104</td>
<td>color</td>
</tr>
<tr>
<td>Wayne’s World</td>
<td>1992</td>
<td>95</td>
<td>color</td>
</tr>
</tbody>
</table>

OR?

OR?
How to represent data? (cont.)

Please vote:
(1) Table
(2) Tree
(3) Graph
How to operate on data?

Given the data, say, a set of tables, how to answer queries?

For table representation, queries may depend data in all tables.

**Product**

<table>
<thead>
<tr>
<th>PName</th>
<th>Price</th>
<th>Category</th>
<th>Manufacturer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gizmo</td>
<td>19.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>Powergizmo</td>
<td>29.99</td>
<td>Gadgets</td>
<td>GizmoWorks</td>
</tr>
<tr>
<td>SingleTouch</td>
<td>149.99</td>
<td>Photography</td>
<td>Canon</td>
</tr>
<tr>
<td>MultiTouch</td>
<td>203.99</td>
<td>Household</td>
<td>Hitachi</td>
</tr>
</tbody>
</table>

**Company**

<table>
<thead>
<tr>
<th>cName</th>
<th>StockPrice</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>GizmoWorks</td>
<td>25</td>
<td>USA</td>
</tr>
<tr>
<td>Canon</td>
<td>65</td>
<td>Japan</td>
</tr>
<tr>
<td>Hitachi</td>
<td>15</td>
<td>Japan</td>
</tr>
</tbody>
</table>

**Q: Find all products under price 200 manufactured in Japan?**
How to operate on data? (cont.)

Product(PName, Price, Category, Manufacturer)
Company(CName, StockPrice, Country)

**SQL**

```
SELECT x.PName, x.Price
FROM Product x, Company y
WHERE x.Manufacturer=y.CName
    AND y.Country='Japan'
    AND x.Price ≤ 200
```

**Implementation**

How to physically implement operators in SQL, e.g., SELECT?
How to speed up the operation?

Relational operations can sometimes be computed much faster if we have precomputed a suitable data structure on the data. This is called **Indexing**.
How to speed up the operation?

Relational operations can sometimes be computed much faster if we have precomputed a suitable data structure on the data. This is called **Indexing**.

Most notably, two kinds of index structures are essential to database performance:

1. **B-trees**.
2. **External hash tables**.

For example, hash tables may speed up relational operations that involve finding all occurrences in a relation of a particular word.
How to make a good operation plan?

How to optimize the orders of the operations?

$$R(A, B, C, D), S(E, F, G)$$

Find all pairs $$(x, y), x \in R, y \in S$$ such that

(1) $$x.D = y.E$$, 
(2) $$x.A = 5$$ and 
(3) $$y.G = 9$$

$$\sigma_{A=5 \land G=9}(R \bigtriangleup_{D=E} S) = \sigma_{A=5}(R) \bigtriangleup_{D=E} \sigma_{G=9}(S)$$

**Q:** Use the LHS or RHS?
How to make the database reliable yet efficient?

What if the database system fails at some point?
How to make the database reliable yet efficient?

What if the database system fails at some point?

In large systems we need to allow many transactions (queries and updates) to run in parallel

How to perform concurrency control?
How to make the database reliable yet efficient?

What if the database system fails at some point?

In large systems we need to allow many transactions (queries and updates) to run in parallel.

How to perform concurrency control?

We will talk about how transactions are implemented using locking mechanisms.
How to make the database reliable yet efficient?

What if the database system fails at some point?

In large systems we need to allow many transactions (queries and updates) to run in parallel

How to perform concurrency control?

We will talk about how transactions are implemented using locking mechanisms.

This knowledge is useful in database implementation, e.g., it makes it possible in some cases to avoid (or reduce) rollbacks of transactions, and generally make transactions wait less for each other.
The focus of this course

Concept (B561)

Logic
(express the query)

Algorithm
(solve the query)

Implementation
(B662 Database System and Internal Design)

System
(implementation)
The focus of this course

**Concept (B561)**

**Logic**
(express the query)

Data Representation, Relational Algebra, SQL (Datalog), etc.

**Algorithm**
(solve the query)

Indexing, Query Optimization, Concurrency Control, etc.

---

**Implementation**
(B662 Database System and Internal Design)

**System**
(implementation)
The focus of this course

Concept (B561)

Logic
(express the query)
Data Representation, Relational Algebra, SQL (Datalog), etc.

Algorithm
(solve the query)
Indexing, Query Optimization, Concurrency Control, etc.

Implementation
(B662 Database System and Internal Design)

System
(implementation)

And you need math!!
What’s more in this course?
Advanced topics

- **Beyond (will cover)**
  "SQL, Relational Algebra, Data Models, Storage, Views and Indexing, Query Processing, Query Optimization, Transaction Recovery, Concurrency Control"
Advanced topics

- **Beyond (will cover)**
  "SQL, Relational Algebra, Data Models, Storage, Views and Indexing, Query Processing, Query Optimization, Transaction Recovery, Concurrency Control"

- **I will give you a taste of**
  1. **Data Privacy**
  2. **Some NoSQL Models**, e.g.
      (a) Streaming
      (b) MapReduce and ActiveDHT
  3. **Optimal Join Algorithms**
Advanced topics

- **Beyond (will cover)**
  "SQL, Relational Algebra, Data Models, Storage, Views and Indexing, Query Processing, Query Optimization, Transaction Recovery, Concurrency Control"

- **I will give you a taste of**
  1. **Data Privacy**
  2. **Some NoSQL Models**, e.g.
      (a) Streaming
      (b) MapReduce and ActiveDHT
  3. **Optimal Join Algorithms**

Cannot find in textbooks ...
Advanced topics

- **Beyond (will cover)**
  "SQL, Relational Algebra, Data Models, Storage, Views and Indexing, Query Processing, Query Optimization, Transaction Recovery, Concurrency Control"

- **I will give you a taste of**
  1. **Data Privacy**
  2. **Some NoSQL Models**, e.g.
     (a) Streaming
     (b) MapReduce and ActiveDHT
  3. **Optimal Join Algorithms**

Cannot find in textbooks ...

Take a look at top DB conferences: SIGMOD/PODS!
Welcome to B561! But you should know:

If you want a **challenging** DB course with a **character**, emphasizing the theoretical foundation and a little bit research oriented, this course would help.

If you want a “standard” undergraduate DB course following a textbook, or an “industry oriented” DB course teaching you how to write e.g. *an online shopping system*, this is NOT the right course for you.
Course plan

Two interwoven threads:

0 : (1) Introduction
Basic 1 : (4) SQL, Relational Algebra/Calculus, Datalog
Basic 2 : (2) View, Index, Constraints
Basic 3 : (3) Data Models
Advanced 1 : (3) Streaming Model, MapReduce and ActiveDHT
Basic 4 : (4) Query Optimization
Advanced 2 : (1) Optimal Join Algorithms
Basic 5 : (3) Transactions
Advanced 3 : (2) Data Privacy

Basic: must master
Advanced: try your best
Resources

- Main reference book (we will go beyond this)
  - Database Systems: The Complete Book
    by Hector Garcia-Molina, Jeff Ullman
    and Jennifer Widom, 2nd Edition

- Other reference books (undergrad textbooks ... )
  - Database Management Systems
    by Ramakrishnan and Guhrke, 3rd Edition
  - Database System Concepts
    by UllSilberschatz, Korth and Sudarshan,
    6th Edition
Other reference books (cont.):

- **Readings in Database Systems** “Red book”
  Hellerstein and Stonebraker, eds., 4th Edition
  (Will be one of our readings)

- **Foundations of Databases: The Logical Level**
  “Alice book”
  by Abiteboul, Hull, Vianu

- **Concurrency Control and Recovery in Database Systems**
  by Bernstein, Hadzilacos, Goodman

  \(^a\)http://research.microsoft.com/en-us/people/philbe/cccontrol.aspx
Other reference books (cont.):

- **Algorithms and Data Structures for External Memory**
  by Vitter
  

- **Data Streams: Algorithms and Applications**
  by S. Muthukrishnan
  
  [http://www.cs.rutgers.edu/~muthu/stream-1-1.ps](http://www.cs.rutgers.edu/~muthu/stream-1-1.ps)
Other reference books (cont.):

- **Algorithms and Data Structures for External Memory** by Vitter
  \(^a\)http://www.ittc.ku.edu/~jsv/Papers/Vit.I0_book.pdf

- **Data Streams: Algorithms and Applications** by S. Muthukrishnan
  \(^a\)http://www.cs.rutgers.edu/ muthu/stream-1-1.ps

These are surely not enough, and sometimes dated. Want to learn more? Read original papers!
Instructors

- Instructor: Qin Zhang
  Email: qzhangcs@indiana.edu
  Office hours: Tue. 3-4pm @ Lindley 430A

- Associate Instructors:
  - Ali Varamesh
  - Yuan Xie
  - Chao Tao

  Office hours: posted on course website
Grading

Assignments 20% : 6 written assignments (each 2-4%).
Solutions should be typeset in LaTeX (highly recommended) or Word.

Project 20% : Write a report on one topic.
Details will be posted at the beginning of the 3rd week.

Exams 60% : Mid-term (30%) and Final (30%).
Grading

Assignments 20% : 6 written assignments (each 2-4%).
Solutions should be typeset in LaTeX (highly recommended) or Word.

Project 20% : Write a report on one topic.
Details will be posted at the beginning of the 3rd week.

Exams 60% : Mid-term (30%) and Final (30%).

Final score will be a weighted average (according to XX%).
Final grade will generally follow the normal distribution
LaTeX: Highly recommended tools for assignments/reports

1. Read wiki articles:
   http://en.wikipedia.org/wiki/LaTeX

2. Find a good LaTeX editor.

3. Learn how to use it, e.g., read “A Not So Short Introduction to LaTeX 2e” (Google it)
Participants are expected to have a background in algorithms and data structures. For example, have taken

1. C241 Discrete Structures for Computer Science
2. C343 Data Structures
3. B403 Introduction to Algorithm Design and Analysis

or equivalent courses, and know some basics of databases.
Participants are expected to have a background in algorithms and data structures. For example, have taken
1. C241 Discrete Structures for Computer Science
2. C343 Data Structures
3. B403 Introduction to Algorithm Design and Analysis
or equivalent courses, and know some basics of databases.

I haven’t taken B403 “Introduction to Algorithm Design and Analysis” or equivalent courses. Can I take the course? Or, will this course fit me?

Generally speaking, this is an advanced course. It will be very difficult if you do not have enough math/algorithm background.
The goal of this course

Open / change your views of the world (of databases)
The goal of this course

Open / change your views of the world (of databases)

Seriously, it is not just SQL programming.

Read “The relational model is dead, SQL is dead, and I don’t feel so good myself”
Announcements

1. We will have a background survey this Thursday (Aug. 27).
   - To help you decide whether to take this course.
   - To let the instructors know about your background.

0 It will NOT be counted in the final grade.
0 I will be out-of-town, and AI’s will take care of this.
Announcements

1. We will have a background survey this Thursday (Aug. 27).
   – To help you decide whether to take this course.
   – To let the instructors know about your background.
   o It will NOT be counted in the final grade.
   o I will be out-of-town, and AI’s will take care of this.

2. Due to the course arrangements, I only have 30 mins between the two sessions, and need to walk from WH to GL (10 mins walk) AND have lunch.
   So if you have questions, ask me
   – before the first session, or
   – after the second session, or
   – in my office hours, or
   – walk with me from WH to GL with sandwiches
Thank you!

Questions?
Like to work on a summer project on building a library for streaming algorithms? Talk to me.

Look for interns on data infrastructure engineer? Talk to Dr. Minaxi Gupta (minaxi@cs.indiana.edu) @ https://www.edmodo.com/