§0 Introduction

Qin Zhang
Self introduction: my research interests

• **Algorithms for Big Data:**
  streaming/sketching algorithms;
  algorithms on distributed data;
  I/O-efficient algorithms;
  data structures;

• **Complexity:**
  communication complexity.

I am a theoretician, but also work on databases (and occasionally on data mining)
What is databases?
What is databases?

What should DB concept course include?
What is databases?

What should DB concept course include?

I ask because:

(previous feedbacks on “what did you like least”)

1. the course “did not follow books and design like undergraduate database course”

2. should be “more industry orientation”

3. “the course content had too much research orientation and algorithms”

4. “no coding project”

5. “no point in putting effort in 20-line long SQL query which you will never write in practical software”
What is databases?

What should DB concept course include?

I ask because:

(previous feedbacks on “what did you like least”)

1. the course “did not follow books and design like undergraduate database course”

I hope you are graduate students :)  

2. should be “more industry orientation”

3. “the course content had too much research orientation and algorithms”

4. “no coding project”

5. “no point in putting effort in 20-line long SQL query which you will never write in practical software”
What is databases?

What should DB concept course include?

I ask because:

(previous feedbacks on “what did you like least”)

1. the course “did not follow books and design like undergraduate database course”

2. should be “more industry orientation”

3. “the course content had too much research orientation and algorithms”

4. “no coding project”

5. “no point in putting effort in 20-line long SQL query which you will never write in practical software”

I hope you are graduate students :) This is not a course for professional training
What is databases?

What should DB **concept** course include?

I ask because:

(previous feedbacks on “what did you like least”)

1. the course “did not follow books and design like undergraduate database course”
2. should be “more industry orientation”
3. “the course content had too much research orientation and algorithms”
4. “no coding project”
5. “no point in putting effort in 20-line long SQL query which you will never write in practical software”

I hope you are graduate students :) This is not a course for professional training 

– DB cannot live without algorithms.
– IUB is a research oriented univ.
What is databases?

What should DB concept course include?

I ask because:

(previous feedbacks on “what did you like least”)

1. the course “did not follow books and design like undergraduate database course”

2. should be “more industry orientation”

3. “the course content had too much research orientation and algorithms”

4. “no coding project”

5. “no point in putting effort in 20-line long SQL query which you will never write in practical software”
What is databases?

What should DB concept course include?

I ask because:

(previous feedbacks on “what did you like least”)

1. the course “did not follow books and design like undergraduate database course”

2. should be “more industry orientation”

3. “the course content had too much research orientation and algorithms”

4. “no coding project”

5. “no point in putting effort in 20-line long SQL query which you will never write in practical software”

I hope you are graduate students :)

This is not a course for professional training

– DB cannot live without algorithms.

– IUB is a research oriented univ.

Please consider B662

Are you sure?
What does a typical undergrad database course cover?
How to represent the data in the computer?

<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 in the computer?

<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?**

```
Movie
  Sci-Fi
    Star Wars, 1977
  Cartoon
    Mighty Ducks, 1991
  Comedy
    Wayne’s World, 1992
```
How to operate on data?

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

**Difficulty:** Queries may depend crucially on the 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

<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>

### SQL

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

### Relational Algebra

\[
\pi_{\text{PName}, \text{Price}} 
\left( 
\sigma_{\text{Price} \leq 200 \land \text{Country} = 'Japan'} \left( \text{Product} \bowtie_{\text{Manufacturer} = \text{CName}} \text{Company} \right) \right)
\]
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 value.
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) \(x \cdot D = y \cdot E\), (2) \(x \cdot A = 5\) and (3) \(y \cdot G = 9\)

\[ \sigma_{A=5 \land G=9}(R \ominus D=E S) = \sigma_{A=5}(R) \ominus D=E \sigma_{G=9}(S) \]

Q: Use the LHS or RHS?
How to deal with transactions?

Transactions with the ideal ACID properties resolve the semantic problems that arise when many concurrent users access and change the same database.

- Atomicity (= recovery)
- Consistency
- Isolation (= concurrency control)
- Durability
How to deal with transactions?

Transactions with the ideal ACID properties resolve the semantic problems that arise when many concurrent users access and change the same database.

- Atomicity (＝recovery)
- Consistency
- Isolation (＝concurrency control)
- Durability

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

This knowledge is useful in database programming, 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.
Summary

Database =

Logic
(express the query)

Algorithm
(solve the query)

System
(implementation)
Summary

Database =

Logic
(express the query)

Algorithm
(solve the query)

System
(implementation)

Concept (our focus)

Implementation
(see B662 Database System and Internal Design)
Summary

Database =

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

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

System
(implementation)

Concept (our focus)

Implementation
(see B662 Database System and Internal Design)
Summary

Database =

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

Algorithm
(solve the query)

System
(implementation)

And you need **math!!**

Implementation
(see B662 Database System and Internal Design)

Concept (our focus)

Indexing, Query Optimization, Concurrency Control, etc.
What’s more in this course?
Advanced topics

Beyond
”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) The Data Stream Model
   (b) MapReduce and ActiveDHT
3. **Optimal Join Algorithms**
Advanced topics

Beyond
”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) The Data Stream Model
   (b) MapReduce and ActiveDHT
3. Optimal Join Algorithms

Are these related to databases? cannot find in the book
Advanced topics

Beyond
”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) The Data Stream Model
   (b) MapReduce and ActiveDHT
3. Optimal Join Algorithms

Are these related to databases? cannot find in the book

Take a look at leading DB conferences:
SIGMOD/PODS/VLDB
Other important topics in databases

More but probably will not cover much

1. **Tree-based data models** e.g., XML
2. **Graph-based data models** e.g., RDF
3. **Spatial** databases
4. **Parallel and Distributed** databases
   partly covered in MapReduce
5. **Social Networks**
6. **Uncertainty** in databases
   etc.
Welcome to this course, but ... it is your decision:

If you want a **challenging** DB course **with a character** (emphasize the theoretical foundation but not system; a little bit research oriented), this course should work.

If you want a “standard” undergraduate DB course following a textbook, or an “industry oriented” DB course teaching you how to write practical SQL queries, this is **NOT** the right course for you.
Two interleaved 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: at least get the idea
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

Other reference books (cont.):

- **Algorithms and Data Structures for External Memory**
  by Vitter
  
  \[a\text{http://www.ittc.ku.edu/~jsv/Papers/Vit.IO_book.pdf}\]

- **Data Streams: Algorithms and Applications**
  by S. Muthukrishnan
  
  \[a\text{http://www.cs.rutgers.edu/ muthu/stream-1-1.ps}\]
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)

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

- Instructor: Qin Zhang
  Email: qzhangcs@indiana.edu
  Office hours: ?? Lindley 430A

- Associate Instructors:
  -
  -
  -
  -

  Office hours: posted on course website
Assignments 20% : Four written assignments (each 5%).

Solutions should be typeset in LaTeX (highly recommended) or Word.

Quiz 20% : Randomly, 3-5 times, in class
(XX% of each quiz will be given on the quiz paper, summing up to 20%)

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

Assignments 20% : Four written assignments (each 5%).

Solutions should be typeset in LaTeX (highly recommended) or Word.

Quiz 20% : Randomly, 3-5 times, in class
(XX% of each quiz will be given on the quiz paper, summing up to 20%)

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

Final score will be a weighted average (according to XX%).
Final grade will follow the Guassian 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.
Any other questions?
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”
Thank you!

Questions?

A few introductory slides are based on Prof. Rasmus Pagh’s slides
http://www.itu.dk/people/pagh/ADBT06/