DATABASES
in a few slides
DATABASE

a collection of logically related records

UOD: Universe of Discourse
(all the facts in the db)
RELATIONAL DB

• all information that comprises the universe of discourse is represented as relations.

• relations can be precisely defined mathematically.

• can also be defined using tables

• relational database = a collection of tables representing a UOD
RDBMS

• Relational database management system
  • software that enables users to define, create, and maintain a database and provide controlled access.

• RDBMS ranking
  • Oracle
  • MySQL (Oracle) - mariaDB
  • Microsoft SQL server
  • IBM DB2 (10%)
RDBMS RANKING

1. Oracle      1467
2. MySQL      1296
3. Microsoft SQL server  1226
4. PostgreSQL   228
5. DB2         188
Why The Older-Than-Dirt Postgres Database Is Hot With Hipsters And Oldsters Alike

Mainly, it just works.
Boring has never looked so cool. The decades-old relational-database management system Postgres, once the forgotten older sibling to MySQL, has been on a tear the last few years. Postgres has Oracle to thank for some of its newfound sexiness, as Oracle has fumbled MySQL's community outreach at crucial moments.

But far more of Postgres' renaissance stems from the fact that it boringly, reliably works. It turns out "boring" is a critical feature in a database.
PostgreSQL Hits 9.3, New Levels Of Popularity With The Cool Kids

PostgreSQL has long been a great database, yet strangely ignored. Now it's rising in popularity. Why?
SQL

• S-Q-L or sequel (mySQL = my-S-Q-L or my-sequel)
  • ANSI standard = S-Q-L

• SQL = Structured Query Language for RDBMS
  • orig. IBM 1970s SEQUEL (Structure English Query Language)
  • the ‘standard’ language for RDBMS
All ways of describing stuff

relational models

SQL
8 WEEKS
All ways of describing stuff

relational models

SQL
NOSQL
<table>
<thead>
<tr>
<th>Rank</th>
<th>Last Month</th>
<th>DBMS</th>
<th>Database Model</th>
<th>Score</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>1.</td>
<td>Oracle</td>
<td>Relational DBMS</td>
<td>1439.16</td>
<td>-20.63</td>
</tr>
<tr>
<td>2.</td>
<td>2.</td>
<td>MySQL</td>
<td>Relational DBMS</td>
<td>1277.51</td>
<td>+8.93</td>
</tr>
<tr>
<td>3.</td>
<td>3.</td>
<td>Microsoft SQL Server</td>
<td>Relational DBMS</td>
<td>1198.61</td>
<td>-1.44</td>
</tr>
<tr>
<td>4.</td>
<td>4.</td>
<td>PostgreSQL</td>
<td>Relational DBMS</td>
<td>254.49</td>
<td>+0.48</td>
</tr>
<tr>
<td>5.</td>
<td>5.</td>
<td>MongoDB</td>
<td>Document store</td>
<td>250.90</td>
<td>+4.38</td>
</tr>
<tr>
<td>6.</td>
<td>6.</td>
<td>DB2</td>
<td>Relational DBMS</td>
<td>200.13</td>
<td>-10.12</td>
</tr>
<tr>
<td>7.</td>
<td>7.</td>
<td>Microsoft Access</td>
<td>Relational DBMS</td>
<td>139.14</td>
<td>-0.76</td>
</tr>
<tr>
<td>8.</td>
<td></td>
<td>Cassandra</td>
<td>Wide column store</td>
<td>98.75</td>
<td>+4.69</td>
</tr>
<tr>
<td>9.</td>
<td></td>
<td>SQLAlchemy</td>
<td>Relational DBMS</td>
<td>96.20</td>
<td>+1.49</td>
</tr>
<tr>
<td>10.</td>
<td></td>
<td>Redis</td>
<td>Key-value store</td>
<td>94.24</td>
<td>+6.36</td>
</tr>
</tbody>
</table>
ALONG SIDE DB STUFF
learn about developing web apps.
OLD SKOOL: PHP
## Results

<table>
<thead>
<tr>
<th>Website</th>
<th>Visits</th>
<th>Programming Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>facebook.com</td>
<td>870,000,000</td>
<td>PHP</td>
</tr>
<tr>
<td>youtube.com</td>
<td>790,000,000</td>
<td>Python</td>
</tr>
<tr>
<td>yahoo.com</td>
<td>590,000,000</td>
<td>PHP*</td>
</tr>
<tr>
<td>live.com</td>
<td>540,000,000</td>
<td>ASP.NET</td>
</tr>
<tr>
<td>wikipedia.org</td>
<td>460,000,000</td>
<td>PHP</td>
</tr>
<tr>
<td>msn.com</td>
<td>450,000,000</td>
<td>ASP.NET</td>
</tr>
<tr>
<td>blogspot.com</td>
<td>370,000,000</td>
<td>Python*</td>
</tr>
<tr>
<td>baidu.com</td>
<td>310,000,000</td>
<td>PHP</td>
</tr>
<tr>
<td>microsoft.com</td>
<td>280,000,000</td>
<td>ASP.NET</td>
</tr>
<tr>
<td>qq.com</td>
<td>250,000,000</td>
<td>PHP</td>
</tr>
<tr>
<td>bing.com</td>
<td>230,000,000</td>
<td>ASP.NET</td>
</tr>
<tr>
<td>ask.com</td>
<td>190,000,000</td>
<td>SSI</td>
</tr>
<tr>
<td>taobao.com</td>
<td>170,000,000</td>
<td>PHP</td>
</tr>
<tr>
<td>twitter.com</td>
<td>160,000,000</td>
<td>Ruby on Rails</td>
</tr>
<tr>
<td>New</td>
<td>160,000,000</td>
<td>PHP</td>
</tr>
<tr>
<td>Websites</td>
<td>Popularity (unique visitors)</td>
<td>Front-end (Client-side)</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Google.com</td>
<td>1,100,000,000</td>
<td>JavaScript</td>
</tr>
<tr>
<td>YouTube.com</td>
<td>1,000,000,000</td>
<td>Flash, JavaScript</td>
</tr>
<tr>
<td>Facebook.com</td>
<td>900,000,000</td>
<td>JavaScript</td>
</tr>
<tr>
<td>Yahoo</td>
<td>750,000,000</td>
<td>JavaScript</td>
</tr>
<tr>
<td>Amazon.com</td>
<td>500,000,000</td>
<td>JavaScript</td>
</tr>
<tr>
<td>Wikipedia.org</td>
<td>475,000,000</td>
<td>JavaScript</td>
</tr>
<tr>
<td>Twitter.com</td>
<td>290,000,000</td>
<td>JavaScript</td>
</tr>
<tr>
<td>Bing</td>
<td>285,000,000</td>
<td>JavaScript</td>
</tr>
</tbody>
</table>
# Back-end (Server-side) Table in Most Popular Websites

<table>
<thead>
<tr>
<th>Websites</th>
<th>ASP.NET</th>
<th>C</th>
<th>C++</th>
<th>D</th>
<th>Erlang</th>
<th>Go</th>
<th>Hack</th>
<th>Java</th>
<th>JavaScript</th>
<th>Perl</th>
<th>PHP</th>
<th>Python</th>
<th>Ruby on Rails</th>
<th>Scala</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google.com</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>YouTube.com</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Facebook.com</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Yahoo</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Amazon.com</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Wikipedia.org</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Twitter.com</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Bing</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>eBay.com</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>MSN.com</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Microsoft</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
COOL KIDS USE

- server side Javascript: node.js
- Python
GOAL

- Learning skills that will help you get a cool job
- Learning skills that you can use on the job
FIRST 8 WEEKS

• a solid working knowledge of how to design relational database systems for web applications

• learn the basics of web app development

• combine the 2: data driven web apps
FRAMWORK

• supports development of dynamic websites, web applications and web resources.

• alleviates overhead of doing it all yourself.

• microframework - core of the framework is simple. (read “easy to get started”)
FRAMEWORKS

- support Model-view-controller pattern.
- means data model is separate from UI
- this modularity enables code reuse and reduces complexity.
- a lot of options for frameworks - Ruby on Rails, Django, Symfony
FRAMEWORK HAZARDS

- full use of a framework would hide a lot of the database heavy lifting.
- would keep the database a mystery.
- you design it and it magically works.
- we are taking the middle road
MODEL VIEW CONTROLLER

responsive
Ron Zacharski

Home

Current Courses

**cs220**
Computer Programming
This honors course uses a particular language, Java, to help you develop rudimentary programming skills.

**cs350**
Databases & web apps
This course provides an introduction to databases and their application, particularly in data driven web applications.
server

url request to server

respond w/ webpage & assets

user clicks on link

respond w/ webpage & assets
url request to server
respond w/ webpage & assets
user clicks on link
respond w/ just the requested data
START W/ OLD SCHOOL

quick transition to responsive (angular)
THAT’S WHAT WE ARE GOING TO DO
WHAT WE ARE NOT DOING
THIS COURSE

• Applications of Databases
• ‘Real DB courses’
  • CPSC410 Database Principles and Design
  • UT CS347: Data Management
Topics related to the engineering and design of database systems, including: data models; database and schema design; schema normalization and integrity constraints; query processing; query optimization and cost estimation; transactions; recovery; concurrency control; isolation and consistency; distributed, parallel, and heterogeneous databases; adaptive databases; trigger systems; key-value stores; object-relational mappings; streaming databases; DB as a service. Lecture and readings from original research papers. 6.830 includes semester-long project and paper.
A programmer’s guide to DB
How to develop data driven web apps.
DESIGNED FOR THOSE WITH

- no database or web development experience
- some experience but have holes in knowledge
NOT

• skills on web design

• skills on usability

• database experts
EXAMPLE

• me
• asymmetric multiprocessing

• some apps 14,000% faster than legacy db.
## 10000 Series Specifications

<table>
<thead>
<tr>
<th>Hardware</th>
<th>10050</th>
<th>10100</th>
<th>10200</th>
<th>10400</th>
<th>10600</th>
<th>10800</th>
</tr>
</thead>
<tbody>
<tr>
<td>Racks (Standard 19&quot;)</td>
<td>1 (50% populated)</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>Total SPUs</td>
<td>56</td>
<td>112</td>
<td>224</td>
<td>448</td>
<td>672</td>
<td>896</td>
</tr>
<tr>
<td>(Query Processing Nodes)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Data (max.)</td>
<td>6.25 TB</td>
<td>12.5 TB</td>
<td>25 TB</td>
<td>50 TB</td>
<td>75 TB</td>
<td>100 TB</td>
</tr>
</tbody>
</table>

**Physical Cabinet Specifications (per rack)**

- **Height**: 80.5” / 2050 mm
- **Width**: 24” / 610 mm
- **Depth**: 38” / 970 mm
- **Weight**: 1,100 lbs / 520 kg (max.)
- **Operating Temperature**: 50F to 95F / 10C to 35C
- **Cooling Requirements**: 12,000 BTU/hour
- **Power Requirements**: Survivable A/B dual power feed, 30A fusing
- **Safety**: UL/CSA/EN60950
- **Emissions**: FCC Part 15, ICES-003, AUS/NZ C-Tick, VCCI and EN55022 Class A; European Immunity: EN55024
NETEZZA
• building citation graphs
  • 1 million papers - 25 million citations
    • mySQL dual processor desktop - 40 hrs
    • Netezza .38 hrs. (288M citations - 50 hrs)
NETEZZA ENGINEERS

- expert at designing db
- expert at writing efficient queries
SUMMARY

• not expert

• sufficient expertise to create web app. startup
ACCELERATED CLASS

• covering 15 weeks of material in 8 weeks.

• not a survey or overview class

• guarantee you that you will know this stuff backwards and forwards.

• focus on software development skills

• this is the last time the course will be 3 credits.
WHAT DO WE DO IN THE LAST 7 WEEKS
All ways of describing stuff

relational models

SQL
All ways of describing stuff

noSQL

relational models

SQL

mongoDB

node.js

redis
DIGRESSION
GUT-FEELING

not cognitively hard.
not complex algorithms or formulas
puzzles
compare to 405
fun
GOAL

• useful

• useful for job interviews.
back to

RELATIONAL DATABASES
TEAM DELIVERABLES

• sent to ron.zacharski@gmail.com by tonight

• section number, team number and name of team

• team picture with names of people in correct order

• individual deliverable: 3x5 card w/ real name, avatar name, & team number