Relational Databases and SQL

CS102
Winter 2019
Relational Database Management Systems

- Around for more than 40 years
- $50+ billion industry
- No sign of slowing down

Why so successful?

- Simple data model
- High-level expressive query language
- Reliable systems
- Scalable systems

Even today’s “NoSQL” systems are starting to look more and more like RDBMSs
Popular RDBMSs

- **Commercial proprietary systems**
  - Oracle
  - Microsoft SQL Server
  - IBM DB2
  - Others ...

- **Open-source systems**
  - MySQL
  - SQLite
  - PostgreSQL
  - Others ...
Basic Concepts

- Relation (table)
- Attribute (column)
- Tuple (row)
- Types and domains

Cities

<table>
<thead>
<tr>
<th>city</th>
<th>country</th>
<th>latitude</th>
<th>longitude</th>
<th>temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aalborg</td>
<td>Denmark</td>
<td>57.03</td>
<td>9.92</td>
<td>7.52</td>
</tr>
<tr>
<td>Aberdeen</td>
<td>United Kingdom</td>
<td>57.17</td>
<td>-2.08</td>
<td>8.1</td>
</tr>
<tr>
<td>Abisko</td>
<td>Sweden</td>
<td>63.35</td>
<td>18.83</td>
<td>0.2</td>
</tr>
<tr>
<td>Adana</td>
<td>Turkey</td>
<td>36.99</td>
<td>35.32</td>
<td>18.67</td>
</tr>
<tr>
<td>Albacete</td>
<td>Spain</td>
<td>39.0</td>
<td>-1.87</td>
<td>12.62</td>
</tr>
<tr>
<td>Algeciras</td>
<td>Spain</td>
<td>36.13</td>
<td>-5.47</td>
<td>17.38</td>
</tr>
<tr>
<td>Amiens</td>
<td>France</td>
<td>49.9</td>
<td>2.3</td>
<td>10.17</td>
</tr>
<tr>
<td>Amsterdam</td>
<td>Netherlands</td>
<td>52.35</td>
<td>4.92</td>
<td>8.93</td>
</tr>
<tr>
<td>Ancona</td>
<td>Italy</td>
<td>43.6</td>
<td>13.5</td>
<td>13.52</td>
</tr>
<tr>
<td>Andorra</td>
<td>Andorra</td>
<td>42.5</td>
<td>1.52</td>
<td>9.6</td>
</tr>
<tr>
<td>Angers</td>
<td>France</td>
<td>47.48</td>
<td>-0.53</td>
<td>10.98</td>
</tr>
<tr>
<td>Ankara</td>
<td>Turkey</td>
<td>39.93</td>
<td>32.86</td>
<td>9.86</td>
</tr>
<tr>
<td>Antalya</td>
<td>Turkey</td>
<td>36.89</td>
<td>30.7</td>
<td>11.88</td>
</tr>
<tr>
<td>Arad</td>
<td>Romania</td>
<td>46.17</td>
<td>21.32</td>
<td>9.32</td>
</tr>
<tr>
<td>Athens</td>
<td>Greece</td>
<td>37.98</td>
<td>23.73</td>
<td>17.41</td>
</tr>
<tr>
<td>Augsburg</td>
<td>Germany</td>
<td>48.35</td>
<td>10.99</td>
<td>4.54</td>
</tr>
</tbody>
</table>
Differences Between Table and Spreadsheet

- Name is significant
- Order is not significant - can change on re-open
- Regular structure, more “row-oriented”
- Data elements always values, not formulas

<table>
<thead>
<tr>
<th>city</th>
<th>country</th>
<th>latitude</th>
<th>longitude</th>
<th>temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aalborg</td>
<td>Denmark</td>
<td>57.03</td>
<td>9.92</td>
<td>7.52</td>
</tr>
<tr>
<td>Aberdeen</td>
<td>United Kingdom</td>
<td>57.17</td>
<td>-2.08</td>
<td>8.1</td>
</tr>
<tr>
<td>Abisko</td>
<td>Sweden</td>
<td>63.35</td>
<td>18.83</td>
<td>0.2</td>
</tr>
<tr>
<td>Adana</td>
<td>Turkey</td>
<td>36.99</td>
<td>35.32</td>
<td>18.67</td>
</tr>
<tr>
<td>Albacete</td>
<td>Spain</td>
<td>39.0</td>
<td>-1.87</td>
<td>12.62</td>
</tr>
<tr>
<td>Algeciras</td>
<td>Spain</td>
<td>36.13</td>
<td>-5.47</td>
<td>17.38</td>
</tr>
<tr>
<td>Amiens</td>
<td>France</td>
<td>49.9</td>
<td>2.3</td>
<td>10.17</td>
</tr>
<tr>
<td>Amsterdam</td>
<td>Netherlands</td>
<td>52.35</td>
<td>4.92</td>
<td>8.93</td>
</tr>
</tbody>
</table>
Creating and Loading Data

System-dependent, but can nearly always start with CSV file or similar
Querying

Query executed over one or more tables, returns table as its result
Querying

Query executed over one or more tables, returns table as its result

- Find all cities with temperature between 15 and 25, return city and country
- Find average city temperature for each country
- Find all cities in countries that are in the EU but don’t have coastline, return city and country
- Find all pairs of cities that are close together, i.e., longitude and latitude are less than 0.5 apart
- Find the westernmost city
The SQL Language

- Also more than 40 years old
  One of oldest languages still in use (others?)
- Supported by all RDBMSs, standardized across products
  More or less ...
- Interactive or embedded in programs
- Also can be used to modify the database
Jupyter Notebooks
(formerly iPython notebooks)

- Modeled after “laboratory notebooks”
- In one notebook can combine text boxes (“markdown”) with boxes containing executable code in a wide variety of languages
- Can run/re-run boxes (cells) individually, or run/re-run entire notebook

Rapid adoption in many sectors
Jupyter Notebooks

- Can download to your computer but no one-button download yet
- By default students will use notebooks in the cloud, courtesy Instabase, Google Cloud, and Amazon Web Services
- Either way, notebooks run in a web browser
Jupyter Notebooks

- Can download to your computer but no button download yet

- By default students will use notebooks in the cloud, courtesy Instabase, Google Cloud, and Amazon Web Services

- Either way, notebooks run in a web browser

- Anaconda
- Python 2.7
Jupyter Notebook Hints

To execute a cell, click inside the box then type `shift-enter` (or `shift-return`)

If nothing happens, some cell is probably still executing

Try Kernel > Interrupt or Kernel > Restart
Agenda: Basic SQL

(Creating and populating tables)

1. Basic SELECT statement
2. Ordering
3. Joins
4. Basic aggregation
5. Limit clause

For help while working with SQL:
- Tutorials and help pages (website)
- Web search

Advanced SQL (Thursday)
- Duplicates
- Table variables
- Subqueries of all types
- Advanced aggregation
- Data modification