SQL - The Language of Databases

- Developed by IBM in the 1970s
- Create and process database data
- SQL programming is a critical skill !!!
Facebook and Databases

- Relational databases are accessed in much the same way across the board: SQL. Learning how SQL works is crucial to getting anything done in databases, and any GUI is largely a wrapper around the SQL statements one uses to make those actions happen.
- Knowing a little about database design (layout, B-trees, file storage, normalization) is good, mostly for helping you understand good queries.
- We run the LAMP stack here, so we primarily use MySQL databases across the site.
- I hope this helps a little. Another good motivation may be found in the requirements for most engineering positions here on http://www.facebook.com ;)

Thanks!
Nick from Facebook
(a few years ago)

Relational Query Languages

- A major strength of the relational model:
  - supports simple, powerful querying of data
- Ad-hoc queries
- High-level (declarative) languages
  - Queries can be written intuitively
  - DBMS is responsible for efficient evaluation.
SQL DDL and DML

- SQL statements can be divided into two categories:
  - **Data definition language (DDL)** statements
    - Used for creating and modifying tables, views, and other structures
    - CREATE, DROP, ALTER
  - **Data manipulation language (DML)** statements.
    - Used for queries and data modification
    - INSERT, DELETE, UPDATE, SELECT

The SQL SELECT Statement

- Basic SQL Query:
  ```sql
  SELECT [DISTINCT] column_name(s) | *
  FROM table_name(s)
  [WHERE conditions]
  ```
Selecting All Columns: The Asterisk (*) Keyword

```
SELECT *
FROM Students;
```

<table>
<thead>
<tr>
<th>Student Number</th>
<th>Student LastName</th>
<th>Student FirstName</th>
<th>Email</th>
<th>PhoneNumber</th>
<th>MajDeptName</th>
</tr>
</thead>
<tbody>
<tr>
<td>190</td>
<td>Smith</td>
<td>John</td>
<td><a href="mailto:jsmith@usna.edu">jsmith@usna.edu</a></td>
<td>410-431-3456</td>
<td>ComSci</td>
</tr>
<tr>
<td>673</td>
<td>Doe</td>
<td>Jane</td>
<td><a href="mailto:jdoe@usna.edu">jdoe@usna.edu</a></td>
<td></td>
<td>ComSci</td>
</tr>
<tr>
<td>312</td>
<td>Doe</td>
<td>Jane</td>
<td><a href="mailto:jdoe2@usna.edu">jdoe2@usna.edu</a></td>
<td>443-451-7865</td>
<td>Math</td>
</tr>
</tbody>
</table>

Specific Columns and Rows from One Table

```
SELECT StudentNumber, StudentLastName, StudentFirstName
FROM Students
WHERE MajDeptName = 'ComSci';
```

<table>
<thead>
<tr>
<th>Student Number</th>
<th>Student LastName</th>
<th>Student FirstName</th>
</tr>
</thead>
<tbody>
<tr>
<td>190</td>
<td>Smith</td>
<td>John</td>
</tr>
<tr>
<td>673</td>
<td>Doe</td>
<td>Jane</td>
</tr>
</tbody>
</table>
The DISTINCT Keyword

```
SELECT StudentLastName
FROM Students;
```

<table>
<thead>
<tr>
<th>StudentLastName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
</tr>
<tr>
<td>Doe</td>
</tr>
<tr>
<td>Doe</td>
</tr>
</tbody>
</table>

```
SELECT DISTINCT StudentLastName
FROM Students;
```

<table>
<thead>
<tr>
<th>StudentLastName</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
</tr>
<tr>
<td>Doe</td>
</tr>
</tbody>
</table>

Class Exercise

- Department(DeptName, ChairName, WebAddress, DivName)

- Find the name of the Chair of the ‘Math’ Department
WHERE Clause Options

- AND, OR
- IN, NOT IN, BETWEEN
- LIKE
  Wild cards:
    - SQL-92 Standard (SQL Server, Oracle, etc.):
      - _ = Exactly one character
      - % = Any set of characters (zero or more)
    - MS Access
      - ? = Exactly one character
      - * = Any set of characters (zero or more)
- Example:
  Students(SNb, SName, Email, Major)
  Find alpha and name of SCS or SIT students with SNb starting with ‘12’

Sorting the Results

SELECT [DISTINCT] column_name(s) | *
FROM table_name(s)
[WHERE conditions]
[ORDER BY column_name(s) [ASC/DESC]]

Example:
Students(SNb, SName, Email, Major)

SELECT SNb, SName
FROM Students
ORDER BY SName ASC, SNb DESC
Summary (partial)

- SELECT [DISTINCT] column_name(s)
  FROM table_name
  WHERE conditions
  ORDER BY column_name(s) [ASC/DESC]