B345 Internet Science and Technology

Week 10 lecture 2

Today's Lecture Learning Objective

- Understand the issues related to connecting a database to the web.
    - Specifically the 3-tier web application.

Data, Metadata and Programs

- Reminder from last lecture:
    - Different types of "data" being passed at different levels.
    - Not clear what is "data" and what is not.

Basic 3-tier Web Application

- Why the 3-tiers?
- Alternatives:
  - No web front end?
  - No DBMS?
    - Data in web page content (text, images, sound, etc)
    - Data manipulation in server-side scripts
    - Data manipulation in client-side scripts

<< See Diagram A >>

Why DBMS?

- What a database management system can manage for your web app:
    - Efficient storage and retrieval
    - Data integrity
    - Multiple access control
    - Transaction control
    - Crash recovery
- B218 and B334

Database Connectivity

- Technologies:
    - Proprietary APIs
    - ODBC
    - JDBC
    - ADO.NET
    - OLE DB

ODBC

- API for consistent access to different DBMS.
- ODBC drivers
- ODBC interface has
    - Standard way to connect to DBMS
    - Function calls to send SQL queries.
    - Standard data types

<< See Diagram B and C >>

Beyond Basic Connectivity

- .NET and ADO.NET
    - Towards distributed computing
- JDBC
    - Language and Java technology integration
- Do we need it?
    - Most of current problems in web app still is poor DB design and DBMS problems, rather than connectivity issues.

Deciding on the Connectivity Technology

- Which platform?
- What support in chosen DBMS?
- Proprietary or standard connectivity API?
  - For integration with other apps.
  - For easier development within and between projects.

Deciding on the DBMS

- Responsibilities of the DBMS
    - See earlier in the lecture.
- Not the responsibility of the DBMS:
    - Interaction with the users.
    - Presentation of the results or responses.
    - Done by client and server-side scripts.

Easing the job of DBMS

- Scripts can:
    - Preprocess data (eg. validation).
    - Collate requests and queries.
    - Cache result.
- App development: deciding between
    - What the DBMS *can* do.
    - What the DBMS *should* do.

Moving from small to large DBMS

- Small:
  - MS-Access
  - MySQL
- Large:
  - MS SQL Server
  - Oracle 9i
  - IBM DB2

Moving from small to large DBMS

- Gains in
  - Scalability, Performance, Manageability, Integration, Security, etc.

- Features
  - Difference in standard features closing fast.
  - Advanced features:
    - » XML formats
    - » Development tools
    - » Web Services