Week 9

Database Connectivity, Web Technologies & the Cloud
Topics

- Database connectivity
  - Native SQL
  - ODBC (Open Database Connectivity)
  - DAO (Data Access Objects)
  - RDO (Remote Data Objects)
  - ADO (ActiveX Data Objects)
  - JDBC (Java Database Connector)
- Web Services
  - XML (eXtensible Markup Language)
  - DTD (Document Type Definition)
- Cloud Services
Database Connectivity

- How application programs (database middleware) connect and communicate with data sources
- Data repository (data source) – the data management application
  - Used to store data generated by an application program
- ODBC, OLE-DB, ADO.NET - the backbone of MS Universal Data Access (UDA) architecture
Native SQL Connectivity

• Connection interface provided by database vendors
  – Unique to each vendor

• Interfaces optimized for particular vendor’s DBMS

• Example: Oracle RDBMS
  – Must install and configure Oracle’s SQL*Net interface in client computer
ODBC (Open Database Connectivity)

• Open Database Connectivity (ODBC)
  – Microsoft’s implementation of a superset of SQL Access Group Call Level Interface (CLI)
  – Widely supported database connectivity interface
  – Any Windows application can access relational data sources
  – Uses SQL via standard application programming interface (API)
Basic ODBC

• Basic ODBC architecture has three main components:
  – High-level ODBC API through which application programs access ODBC functionality
  – Driver manager that is in charge of managing all database connections
  – ODBC driver that communicates directly to DBMS
Basic ODBC Connection

1. From Excel, select Get External Data, From Other Sources and From Microsoft Query options to retrieve data from an Oracle RDBMS.
2. Select the Gradora ODBC data source.
3. Enter the authentication parameters. ODBC uses the connection parameters to connect to the data source.
4. Select the table and columns to use in the query.
5. Select filtering options to restrict the rows returned.
6. Select sorting options to order the rows.
7. Select Return Data to Microsoft Office Excel.
8. Excel uses the ODBC API to pass the SQL request down to the database. Oracle executes the request and generates a result set. Excel issues calls to the ODBC API to retrieve the result set and populate the spreadsheet.
DAO (Data Access Objects)

- Data Access Objects (DAO)
  - Object-oriented API used by MS Access, MS FoxPro, and dBase databases from Visual Basic programs
  - Provided an optimized interface that exposed functionality of Jet data engine to programmers
  - DAO interface can also be used to access other relational style data sources
RDO

- Remote Data Objects (RDO)
  - Higher-level object-oriented application interface used to access remote database servers
    - Uses lower-level DAO and ODBC for direct access to databases
    - Optimized to deal with server-based databases, such as MS SQL Server, Oracle, and DB2
  
- Implemented as shared code dynamically linked to Windows via dynamic-link libraries
ODBC, DOA & RDO Access

Introduction to Relational Databases

Client Applications
- MS Word
- MS Access
- MS Excel

Remote Data Objects
Data Access Objects
Jet Engine supports MS Access databases and other SQL-aware data sources.

ODBC API
ODBC Driver Manager
ODBC Database Driver
- Oracle Driver
- MS SQL Driver
- ODBC Driver

Oracle
MS SQL
Access

Database vendors provide ODBC database drivers so Windows applications can access their respective databases.
OLE-DB

- Object Linking and Embedding - Database middleware that adds object-oriented functionality for access to data
- Series of COM objects provides low-level database connectivity for applications
- OLE-DB did not provide support for scripting languages
- ActiveX Data Objects (ADO) provides high-level application-oriented interface to interact with OLE-DB, DAO, and RDO
- ADO provides unified interface to access data from any programming language that uses the underlying OLE-DB objects
ADO.NET

- Data access component of Microsoft’s .NET application development framework
- DataSet is disconnected memory-resident representation of database
- DataSet is internally stored in XML format
  - Data in DataSet made persistent as XML documents
- Specific objects manipulate data in data source
  - Connection - Command
  - DataReader - DataAdapter
  - DataSet - DataTable
ADO.NET
Java Database Connectivity (JDBC)

- Java is an object-oriented programming language
  - Runs on top of Web browser software
- Advantages of JDBC:
  - Company can leverage existing technology and personnel training
  - Allows direct access to database server or access via database middleware
  - Provides a way to connect to databases through an ODBC driver
Java Database Connectivity (JDBC)
Web Services
Internet Database Connectivity

- Permit rapid response by bringing new services and products to market quickly
- Increase customer satisfaction through creation of Web-based support services
- Allow anywhere, anytime data access using mobile smart devices via the Internet
- Yield fast and effective information dissemination through universal access
Web Platform Advantages

- Hardware and Software independence
- Platform independence and portability
- Reduced training and support costs
- Single platform development
- Location independence
- Rapid development using open source standard development tools
- Free client application (web browser)
Middleware (Server Side Extension)

- Web server heart of system
- Dynamic Web pages required
- Server-side extension: a program that interacts directly with the Web server
- Middleware must be well integrated
Web Server Interfaces

• Two well-defined Web server interfaces:
  – Common Gateway Interface (CGI)
  – Application Programming Interface (API)

• Disadvantage of CGI scripts:
  – Loading external script decreases system performance
  – Language and method used to create script also decrease performance

• API is more efficient than CGI
  – API is treated as part of Web server program
The Web Browser

• Client Software interprets HTML code received from Web server
• Presents different page components in standard way
• Web is a stateless system: Web server does not know the status of any clients
Client-Side Extensions

• Add functionality to Web browser
• Plug-in - an external application automatically invoked by the browser when needed
• Java and JavaScript - embedded in Web page and activated by an event
• ActiveX and VBScript - embedded in Web page and activated by event
  – Primarily used by Windows applications
Web Application Servers

• Common functions
  – Connect to and query a database from Web page
  – Create dynamic Web search pages
  – Enforce referential integrity

• Some features
  – Security and user authentication
  – Access to multiple services
Web Database Development

• Process of interfacing databases with the Web browser
• Common programming languages
  – ColdFusion
  – PHP
  – vbscript
Web Database Development

```html
<html>
<head>
<title>Rob & Coronel - ColdFusion Examples</title>
</head>
<body bgcolor="LIGHTBLUE">
  <center>
    <cfquery name="venlist" datasource="RobCor">
      SELECT * FROM VENDOR ORDER BY VEN_CODE
    </cfquery>
  </center>
  <center>
    <cfoutput>
      Your query returned #venlist.RecordCount# records
    </cfoutput>
  </center>
  <cfoutput query="venlist">
    VENDOR CODE: #VEN_CODE#
    VENDOR NAME: #VEN_NAME#
    CONTACT PERSON: #VEN_CONTACT_NAME#
    ADDRESS: #VEN_ADDRESS#
    CITY: #VEN_CITY#
    STATE: #VEN_STATE#
    ZIP: #VEN_ZIP#
    PHONE: #VEN_PH#
    FAX: #VEN_FAX#
    E-MAIL: #VEN_EMAIL#
    CUSTOMER ID: #VEN_CUS_ID#
    SUPPORT ID: #VEN_SUPPORT_ID#
    SUPPORT PHONE: #VEN_SUPPORT_PH#
    VENDOR WEB PAGE: #VEN_WEB_PAGE#
  </cfoutput>
</html>
```
What is XML?

• XML stands for eXtensible Markup Language
• Data is formatted in a way that is easy for humans and machines to read
• XML tags are not predefined - you must define your own tags
• XML uses a Document Type Definition (DTD) or an XML Schema to describe the data
• XML with a DTD or XML Schema is designed to be self-descriptive
How XML complements HTML

• XML separates Data from HTML
• Data can be exchanged between incompatible systems
• With XML, plain text files can be used to share data
• XML can also be used to create new languages, such as the Wireless Markup Language (WML)
When To Use XML

- Use when you want to:
  - Communicate between different programs and systems
  - Store complex configuration data
  - Store or move data in databases
  - Display output by different clients, e.g. desktops and PDAs. (Use XSLT to do this)
  - I use it with web applications that pull data feeds from online stores such as Amazon

- Do not use when:
  - You need a relational database model, XML is less efficient than RDBs
XML Files

- Stored as text files
- Can be created with any text editor
- Hierarchical data storage format
- Must contain at least one (root) element
- File names end in .xml extension by default, but vary with use
XML example

<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
  <to> Tom </to>
  <from> Jane </from>
  <heading> Reminder </heading>
  <body> Meeting </body>
</note>

• The first line is the XML declaration and defines the XML version and character encoding used in the document
• The second line is the root element
• The next 4 lines describes four child elements
• The last line ends the element
Another Example

```xml
<?xml version="1.0"?>

<CustomerList>
    <customer id="1">
        <fname> Jen </fname>
        <lname> Lo </lname>
        <title> CEO </title>
        <company> Blue Moon Enterprises </company>
    </customer>
    <customer id="2">
        <fname> Austin </fname>
        <lname> Powers </lname>
        <title> Gold Member </title>
        <company> MiniEvil </company>
    </customer>
</CustomerList>

New tags and the corresponding document structure are “invented” as needed
Tags And Elements

Consider the following XML snippet:

```xml
<customer id="2">
  <fname> Austin </fname>
  <lname> Powers </lname>
  <title> Gold Member </title>
  <company> MiniEvil </company>
</customer>
```

- `<fname>` is a **start tag**
- `</fname>` is an **end tag**
- `<fname>Austin</fname>` is an **element**
- id (in customer tag) is an **attribute**
XML Rules

- XML documents must have a root element
- All other elements are within this root element
- All elements can have sub elements (child elements) that are properly nested within their parent element:

  ```xml
  <root>
    <child>
      <subchild>....</subchild>
    </child>
  </root>
  ```

- All XML elements must have a closing tag
- XML tags are case sensitive
- All XML elements must be properly nested

  Incorrect–<b><i>This is bold and italic</i></b>

  Correct–<b><i>This is bold and italic</i></b>
XML Element Rules

- XML elements can have attributes in name/value pairs

- The attribute value must always be quoted
  
  ```xml
  <?xml version="1.0" encoding="ISO-8859-1"?>
  <note date="6/1/2004">
    <to>Tom</to>
  </note>
  ```

- Comment syntax
  
  ```xml
  <!-- this is a comment -->
  ```

- XML elements can have relationships
  
  ```xml
  <title>My First XML</title>
  <chapter>Introduction to XML</chapter>
  <para>What is HTML</para>
  ```
XML Element Naming Rules

• Names can contain letters, numbers, and other characters
• Names must not start with a number or punctuation character
• Names must not start with the letters xml (or XML or Xml ..)
• Names cannot contain spaces
• Non-English letters like 艸 are perfectly legal in XML element names, but watch out for problems if the software vendor doesn't support them
• The ‘:’ should not be used in element names because it is reserved for namespaces
Well-Formed/Valid Documents

- Well-Formed document:
  - Follows XML syntax rules
  - Every start tag must have an end tag
  - If a document is not well-formed, then it is NOT an XML document

- Valid document:
  - Must be well-formed
  - Must be validated against a Document Type Definition (DTD) that specifies what legal elements and attributes can and/or must exist in the document

- A Schema is an alternative to a DTD
Document Type Definitions (DTD)

- Document Type Definition (DTD)
  - File with .dtd extension that describes elements
  - Provides composition of database’s logical model
  - Defines the syntax rules or valid tags for each type of XML document
- Companies engaging in e-commerce transaction must develop and share DTDs
- DTD referenced from inside XML document
• A DTD can be declared inline in your XML document or as an external reference
• If a DTD is included in your XML document, it should be wrapped in a DOCTYPE definition

Inline

```xml
<!DOCTYPE root-element [element-declarations]>
```

External Reference

```xml
<!DOCTYPE note SYSTEM "Outside.dtd">
```
Example Inline DTD

```
1: <?xml version="1.0"?>
2: <!DOCTYPE note [ 
3: <!ELEMENT note     (to,from,heading,body)> 
4: <!ELEMENT to       (#PCDATA)> 
5: <!ELEMENT from     (#PCDATA)> 
6: <!ELEMENT heading  (#PCDATA)> 
7: <!ELEMENT body     (#PCDATA)> 
8: ]> 
9: <note> 
10: <to> Tom </to> 
11: <from> Jane </from> 
12: <heading>Reminder</heading> 
13: <body>Don't forget me this weekend!</body> 
14: </note>
```
Example External Reference DTD

Note.dtd

```xml
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>
  <to> Tom </to>
  <from> Jane </from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```
XML Presentation

• XML separates data structure from presentation and processing

• Extensible Style Language (XSL) displays XML data
  – Defines the rules by which XML data are formatted and displayed
  – Two parts:
    • Extensible Style Language Transformations (XSLT)
    • XSL style sheets
XML Namespaces

• Namespaces can be specified inside the XML file or the DTD file
• The purpose of XML Namespaces is to distinguish between duplicate elements and attribute names

For example:

```xml
<vehicles>
  <sedans><price>$</price></sedans>
  <trucks><price>$</price></trucks>
</vehicles>
```

• Both “sedans” and “trucks” have the same “price” element, so the parser doesn’t know which one is which
Namespace Example

• `<xml:namespace ns="http://books.org/" prefix="B"/>
  • `<xml:namespace ns="http://ecommerc.org/" prefix="E"/>

  • `<E:Order>
    • `<E:SoldTo>
      • `<E:LastName>Layman</E:LastName>
      • `<E:FirstName>Andrew</E:FirstName>
    • </E:SoldTo>
  • </E:Order>

  • `<E:Item Price="5.95">
    • `<B:Book>
      • `<B:Title>The Call of the Wild</B:Title>
      • `<B:Author>London, Jack</B:Author>
    • </B:Book>
  • </E:Item>`
XML Parsers

- To parse is “to classify a word or analyze a sentence in terms of grammar”
- A Parser is a program that reads the source code of a language and recognizes each element of it
- An XML parser reads the XML source, breaks it down into its elements, and allows another program to act on these elements
Processing XML

To process an XML document, you have several options:

1. Write a program to parse and process (do it yourself)
2. Write a program using some libraries (DOM or SAX)
3. Write processing rules using XSLT
XSL and XSLT

- Process XML data using rules
- The eXtensible Stylesheet Language (XSL) is used to format XML documents for display (e.g. in a browser)
- The eXtensible Stylesheet Language Transformation (XSLT) allows converting XML documents into any other format including XML, HTML, etc,
XSL and XSLT Transformations

The process can render different Web pages for different purposes, such as one page for a Web browser and another for a mobile device.
XML Applications

- B2B exchanges
- Legacy systems integration
- Web page development
- Database support
- Database meta-dictionaries
- XML databases
- XML services
Cloud Computing Services
Cloud Computing Services

• Cloud computing
  – “A computing model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computer resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.”
  – Potential to become a “game changer”
Cloud Services Characteristics

- Ubiquitous - access via Internet technologies
- Shared infrastructure
- Lower costs and variable pricing
- Flexible and scalable services
- Dynamic provisioning
- Service orientation
- Managed operations
Cloud Implementation Types

- **Public cloud** - a service provider makes resources, available to the general public over the Internet, which may be free or offered on a pay-per-usage model.
- **Private cloud** - the cloud (the pool of resource) is only accessible by a single organization providing that organization with greater control and privacy.
- **Hybrid cloud** - an integrated cloud service utilizing both private and public clouds to perform distinct functions within the same organization.
- **Community cloud** - the cloud infrastructure is shared by several organizations and supports a specific community that has shared concerns (e.g., mission, security requirements, policy, and compliance considerations).
Cloud Services

• Software as a Service (SaaS)
  – Microsoft Office Live
  – Google Docs
  – Online email providers

• Platform as a Service (PaaS)
  – Amazon web services
  – Amazon relational data services or simple DB
  – MS Azure of MS SQL services

• Infrastructure as a Service (IaaS)
  – Web hosting
  – On-line storage
# Advantages and Disadvantages

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low initial cost</td>
<td>Security, privacy and compliance</td>
</tr>
<tr>
<td>Scalability</td>
<td>Loss of ownership and control</td>
</tr>
<tr>
<td>Reliability</td>
<td>Difficult data migration</td>
</tr>
<tr>
<td>Support for mobile computing</td>
<td>complex SLAs Long term availability</td>
</tr>
<tr>
<td>Ubiquitous access</td>
<td>Difficult integration with onsite IT</td>
</tr>
<tr>
<td>Easy provisioning</td>
<td></td>
</tr>
</tbody>
</table>
Cloud SQL Management Service

• Provides relational data management to companies of any size
• Avoids high cost of personnel & maintenance
• Leverages the Internet to provide:
  – Hosted data management
  – Standard protocols
  – A common programming interface
• Great choice for businesses with limited information technology resources
Summary

• Database connectivity methods
  – Native SQL – provided by vendor
  – Microsoft Universal Accesss
    • ODBC (Open Database Connectivity)
    • DAO (Data Access Objects)
    • RDO (Remote Data Objects)
    • ADO (ActiveX Data Objects)
  – JDBC (Java Database Connector)

• Web Services
  • XML (eXtensible Markup Language)
  • DTD (Document Type Definition)

• Cloud Services
Assignment 3 (Week 9 - 100 points)

- Write a paper in which you:
  - Compare the structure of a relational database optimized for online transactions with a data warehouse optimized for processing and summarizing large amounts of data
  - Compare the database requirements for operational data and for decision support data
- Describe three (3) examples in which:
  - Databases could be used to support decision making in a large organizational environment
  - Data warehouses and data mining could be used to support data processing and trend analysis in large organizational environment
- Use at least three (3) quality resources in this assignment.