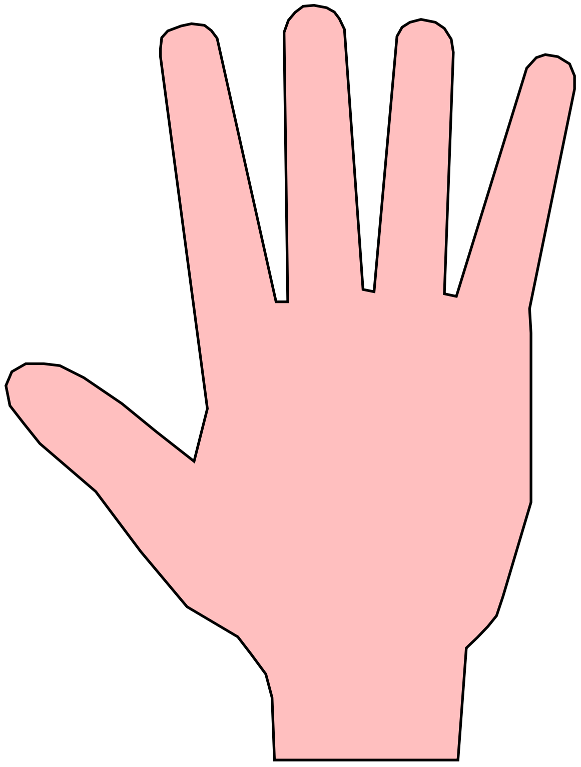
Part 1:
SQLTags Concepts

Introduction- Presenter

- Presentation Goals To Share:
  - Experience,
  - Insights, and
  - Practices
- Get Feedback
- Answer Questions

Introduction- Audience

- Who are You?
- Experience Level?
  - Oracle/JDBC (DBA)
  - Java/JSP (Developers)
  - HTML/Graphics? (UI)



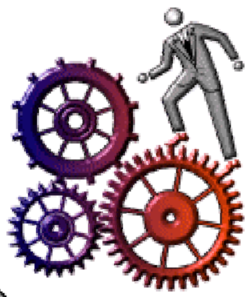
Overview

- SQLTags is a free, open source, object-relational mapping toolkit and “development framework” that provides Java developers with a new and innovative way to access and manipulate data stored within a relational database.
- SQLTags is targeted, primarily, at the Java Server Pages (JSP) environment and can be easily adapted to either the, so called, “Model 1” or “Model 2” JSP Architectures.
- At the SQLTags core is a generator that builds a Java class and a JSP tag for each table within a specified database schema, and a set of run-time support classes and JSP tags that facilitate database development within Java.
- The generator packages everything up into a single "Java Archive" or "jar" file for easy deployment into your web server environment(s).
- This “jar” file can be used within any JSP page to provide easy access to the underlying tables.

Introduction- Skill Sets

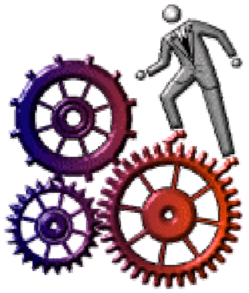
- Web Applications Development Requires the Following Skill Sets:
  - HTML/Graphic Designers/User Interface
  - Database Administrators/Database Designers
  - Developers/Programmers
- Each “Skill Set” Requires Dedication; thus
- Each “Skill Set” Requires Different People

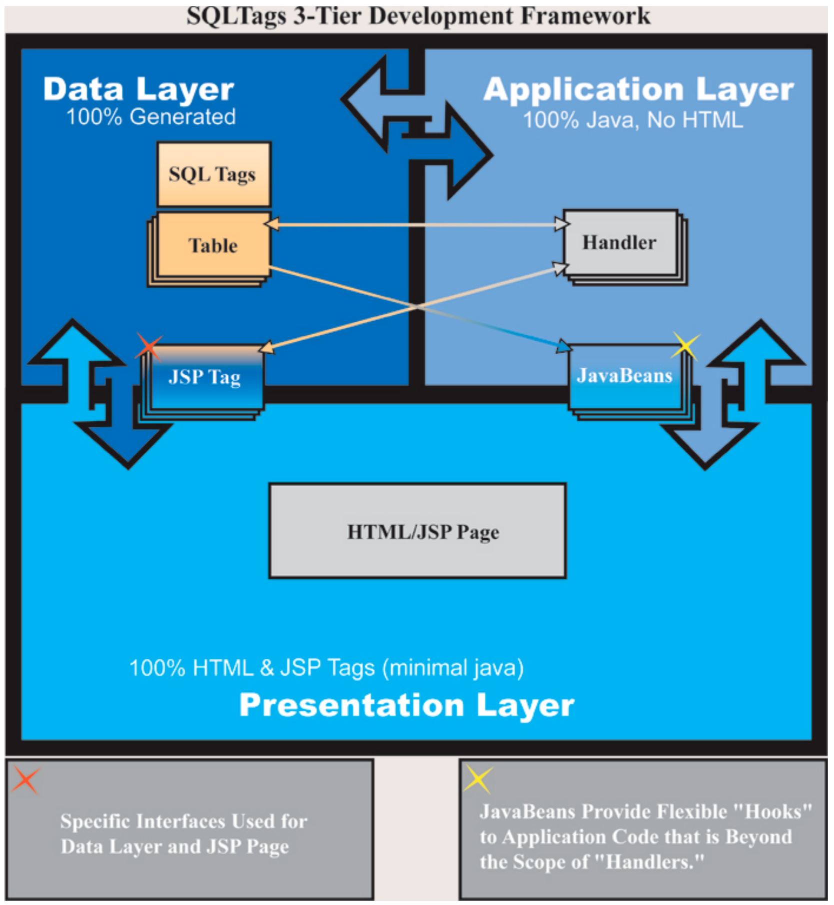
Introduction- Skill Sets



- HTML/Graphic Designers
- Database Administrators/ Database Designers
- Programmers

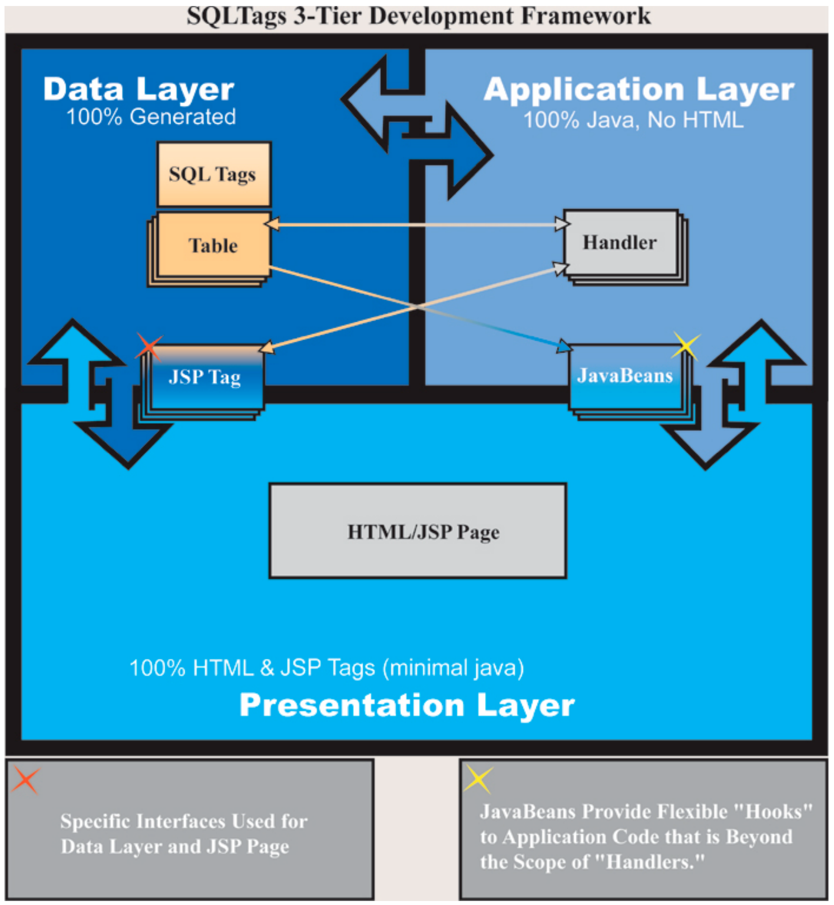
Introduction- Teamwork



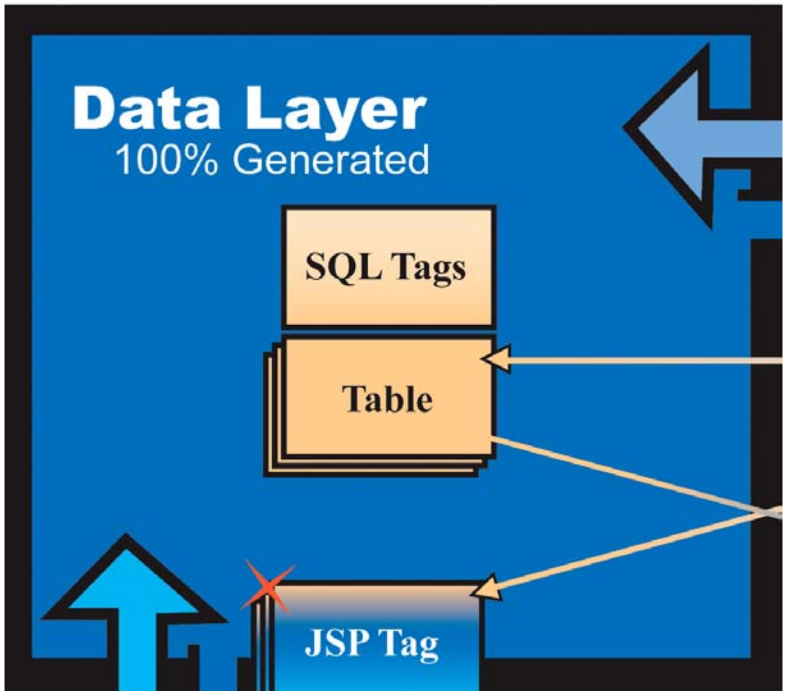

SQLTags 3-Tier Development Framework
Data Layer
100% Generated
Application Layer
100% Java, No HTML
SQL Tags
Table
Handler
JSP Tag
JavaBeans
HTML/JSP Page
100% HTML & JSP Tags (minimal java)
Presentation Layer
Specific Interfaces Used for Data Layer and JSP Page
JavaBeans Provide Flexible "Hooks" to Application Code that is Beyond the Scope of "Handlers."

SQLTags Framework

- Data Layer
  - SQLTags Classes
  - 100% Generated
- Application Layer
  - SQLTagsHandler
  - JavaBeans
- Presentation Layer
  - HTML
  - SQLTags JSP TAGS

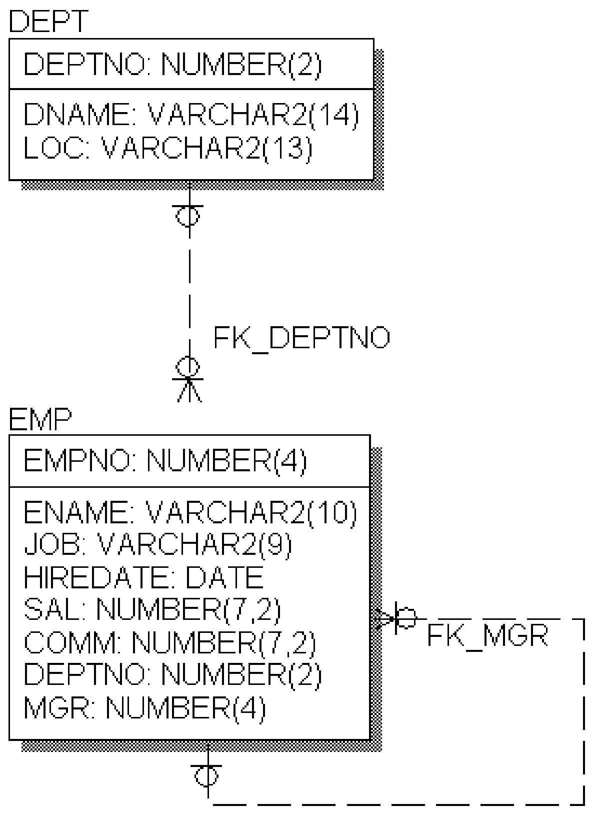

SQLTags 3-Tier Development Framework
Data Layer
100% Generated
Application Layer
100% Java, No HTML
SQL Tags
Table
Handler
JSP Tag
JavaBeans
HTML/JSP Page
100% HTML & JSP Tags (minimal java)
Presentation Layer
Specific Interfaces Used for Data Layer and JSP Page
JavaBeans Provide Flexible "Hooks" to Application Code that is Beyond the Scope of "Handlers."

Data Layer



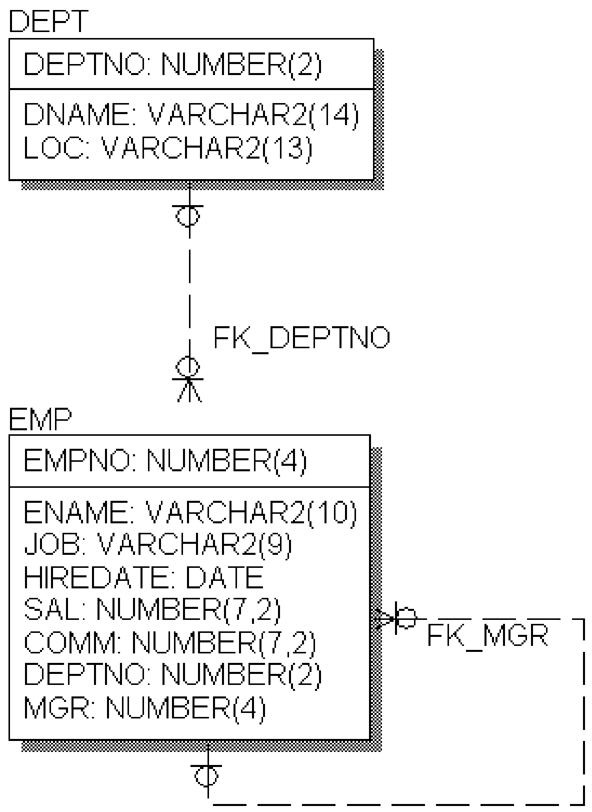
- DBAs maintain Database Schema and Generated Classes
- Based on Entity-Relationship Diagram ("ERD")
- Database Tables, Indexes, and Other Objects
- Generated Java Classes

Data Layer



- ERD Describes
  - Database
    - Tables
      - DEPT, EMP
    - Foreign Keys
      - FK_DEPTNO
      - FK_MGR
    - Indexes and Other …
  - Java Classes
    - DEPT.class, DEPT_TAG.class
    - EMP.class, EMP_TAG.class

Data Layer



- Java Classes
  - Follows “JavaBean” conventions
  - Available to JSP Page
    - as JavaBeans
    - and as TAGS
  - Provides Accessors and Mutators for:
    - Columns
    - Foreign Keys

Data Layer

DEPT.java

```
// Accessors
String getDEPTNO()
String getDNAME()
String getLOC()

// Mutators
void setDEPTNO(String s)
void setDNAME(String s)
void setLOC(String s)

// FK PARENT Accessors
// FK CHILDREN Accessors
EMP getFK_DEPTNO_CHILDREN()
…
boolean insert()
boolean update()
boolean delete()
boolean select()
boolean fetch()
```

- DEPT Class Example
  - Accessors & Mutators
    - Columns
    - Foreign Keys, too!
  - DML Operations
    - Insert
    - Update
    - Delete
  - More …

Data Layer

EMP.java

```
// Accessors
String getEMPNO()
String getENAME()
String getJOB()
String getHIREDATE()
…
// Mutators
void setEMPNO(String s)
void setENAME(String s)
void setJOB(String s)
Void setHIREDATE(String s)
…
// FK PARENT Accessors
DEPT getFK_DEPTNO_PARENT()
EMP  getFK_MGR_PARENT()
// FK CHILDREN Accessors
EMP  getFK_MGR_CHILDREN()
…
```

- EMP Class Example
  - Accessors & Mutators
    - Columns
    - Foreign Keys, too!
  - DML Operations
    - Insert
    - Update
    - Delete
  - More …

Data Layer

Example.jsp

```
<html>
…
<sqltags:dept
    id="d"
    operation="Insert"
    DEPTNO="10"
    DNAME="ACCOUNTING"
    LOC="NEW YORK"
    />
…
</html>
```

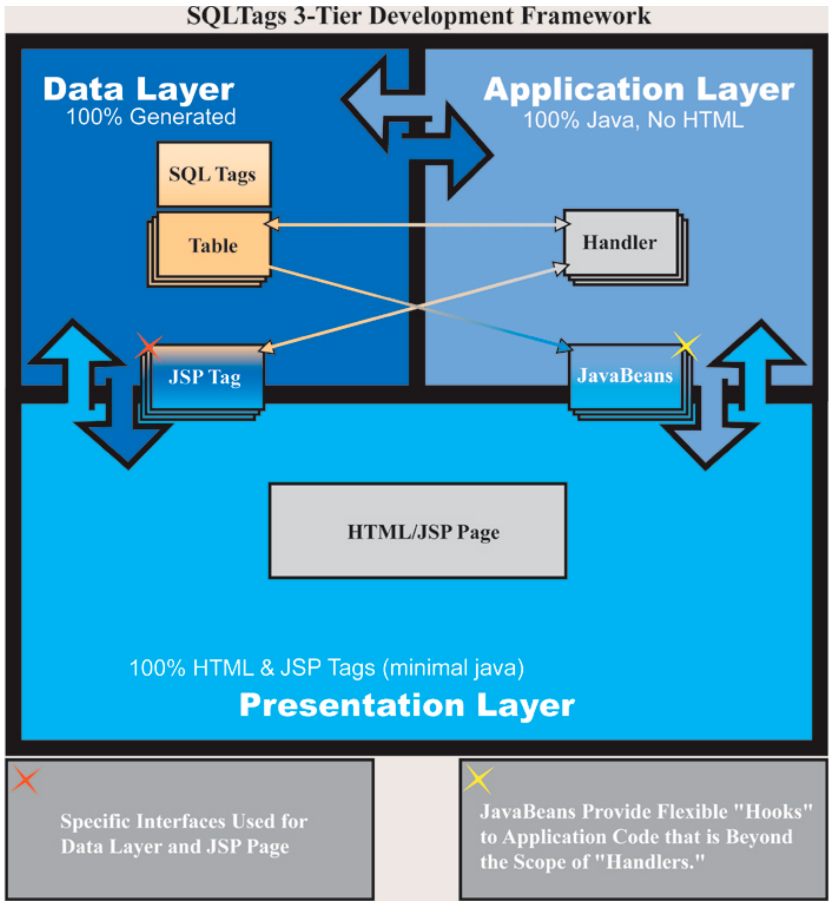
Example2.jsp

```
<html>
…
<sqltags:dept
    id="d2"
    properties="true"
    operation="Insert"
    />
…
</html>
```

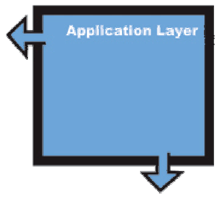
- SQLTags TAGS
- Each Table Becomes a JSP TAG
- Attributes for Each Column
- Integration with JSP "Request" object
- Class is Exposed via Scrptlet Variable
- Much, Much, More …

SQLTags Framework

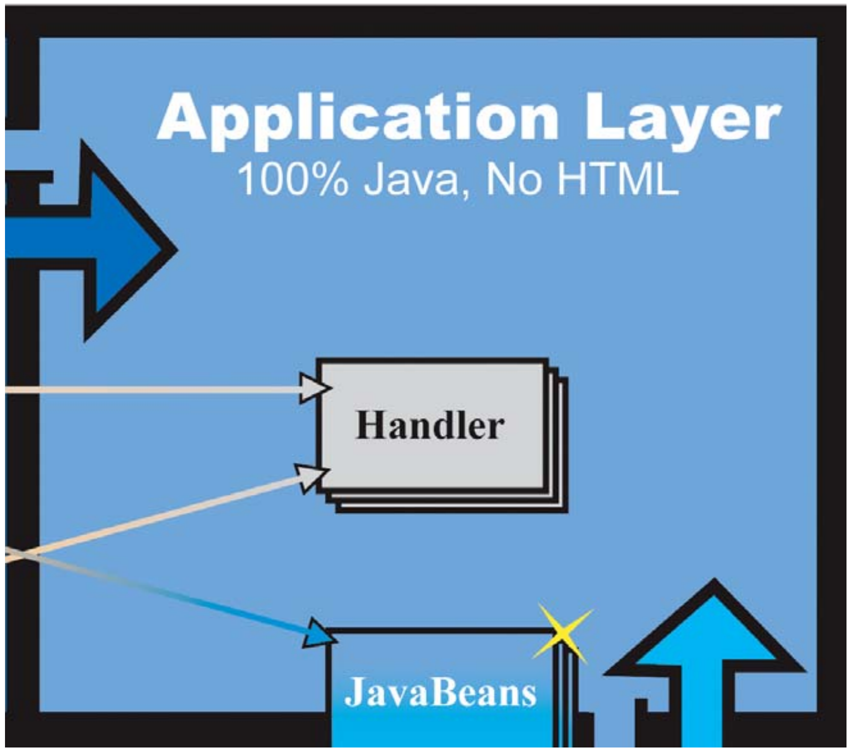
- Data Layer
  - SQLTags Classes
  - 100% Generated
- Application Layer
  - SQLTagsHandler
  - JavaBeans
  - JSP Pages
- Presentation Layer
  - HTML Pages
  - SQLTags JSP TAGS


SQLTags 3-Tier Development Framework
Data Layer
100% Generated
Application Layer
100% Java, No HTML
SQL Tags
Table
Handler
JSP Tag
JavaBeans
HTML/JSP Page
100% HTML & JSP Tags (minimal java)
Presentation Layer
Specific Interfaces Used for Data Layer and JSP Page
JavaBeans Provide Flexible "Hooks" to Application Code that is Beyond the Scope of "Handlers."

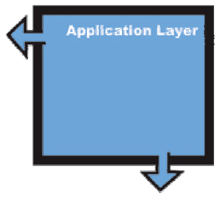
Application Layer



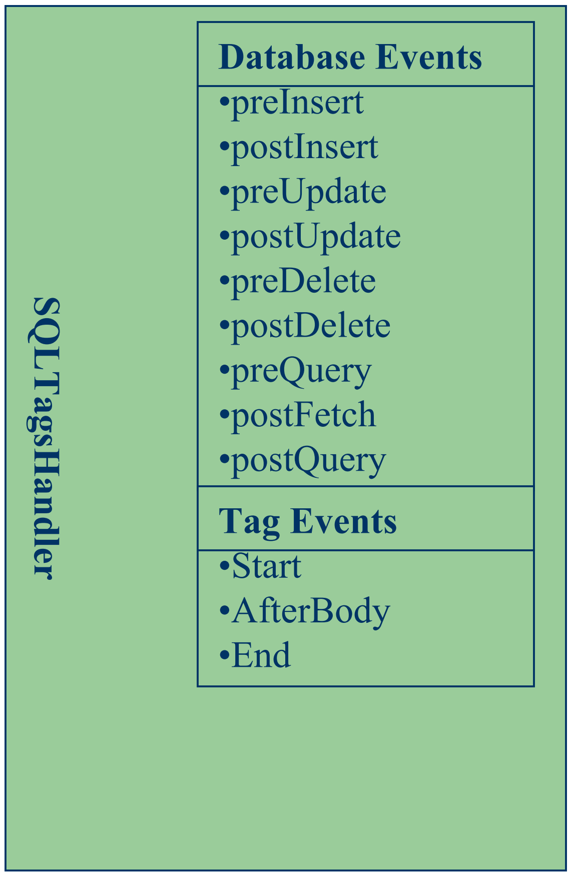
- Java Developers maintain
- SQLTagsHandlers “handle”
  - Database Events
- JavaBeans
  - Generalized Functions
- Other JSP Tags
  - More Generalized Functions abstracted into JSP tags
- Servlets/Struts Components



Application Layer

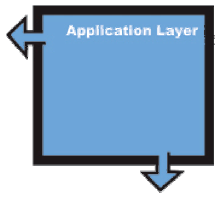


- SQLTagsHandler
  - Programmers Build Java Classes that "Extend" SQLTagsHandler
  - Defines specific "Events" where Application Specific code is called
  - Provides a Structure for **Integration with** Generated Code and JSP
  - Provides a Structure for **Separation from** Generated Code and JSP



Database Events
•preInsert
•postInsert
•preUpdate
•postUpdate
•preDelete
•postDelete
•preQuery
•postFetch
•postQuery
Tag Events
•Start
•AfterBody
•End

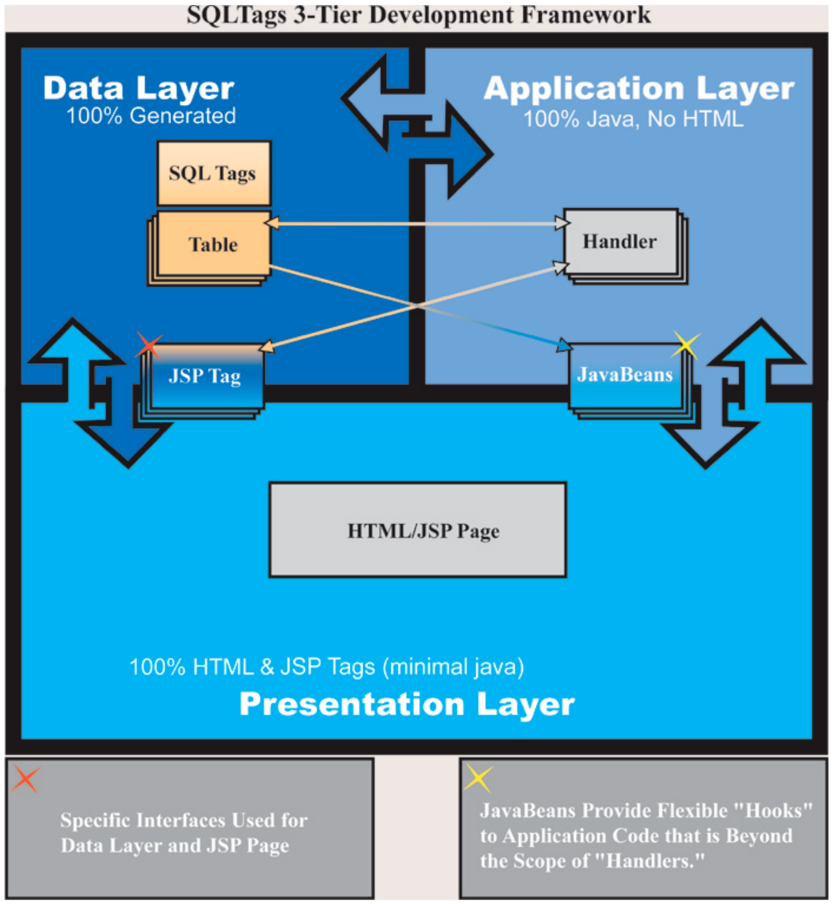
Application Layer



- JavaBeans
  - Nothing within the SQLTags Framework Prohibits Development or use of Additional JavaBeans.
  - In fact, the generated “Table” classes are JavaBeans.
- JSP Tags
  - Nothing within the SQLTags Framework Prohibits Development or use of Additional JSP Tags.

SQLTags Framework

- Data Layer
  - SQLTags Classes
  - 100% Generated
- Application Layer
  - SQLTagsHandler
  - JavaBeans
- Presentation Layer
  - HTML
  - SQLTags JSP TAGS


SQLTags 3-Tier Development Framework
Data Layer
100% Generated
Application Layer
100% Java, No HTML
SQL Tags
Table
Handler
JSP Tag
JavaBeans
HTML/JSP Page
100% HTML & JSP Tags (minimal java)
Presentation Layer
Specific Interfaces Used for Data Layer and JSP Page
JavaBeans Provide Flexible "Hooks" to Application Code that is Beyond the Scope of "Handlers."

Presentation Layer

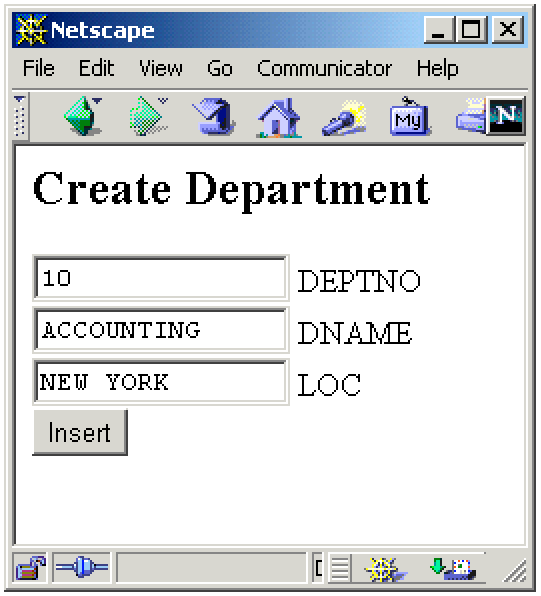
- JSP Pages are Maintained by HTML Authors using:
  - HTML, DHTML, XML
  - JavaScript
  - SQLTags TAGS
  - Other JSP TAGS
  - JavaBeans
  - Very, Very Limited Embedded Java (Scriptlets)

Presentation Layer

- SQLTags Built-in Features
  - properties="true"
  - paging="true"
  - caching="true"
  - "handler reference"
  - scriptlet variables
  - SELECT formatting
  - BIND formatting
- Looping
  - based on "where"
- Data Manipulation:
  - Insert, Update, Delete
- Automatic Join on Foreign Keys to:
  - Parent
  - Children

Presentation Layer

```
<h2>Create Department</h2>
<FORM>
       <INPUT NAME="DEPTNO" SIZE="15" VALUE="10"> DEPTNO
   <BR><INPUT NAME="DNAME"  SIZE="15" VALUE="ACCOUNTING"> DNAME
   <BR><INPUT NAME="LOC"    SIZE="15" VALUE="NEW YORK"> LOC
   <BR><INPUT NAME="fop" TYPE="submit" VALUE="Insert">
</FORM>
```



Create Department

10	DEPTNO
ACCOUNTING	DNAME
NEW YORK	LOC

Insert

Since the “properties” attribute is “true” and the field names in the Form match the column names from the DEPT table the values will automatically be assigned.

Processed By

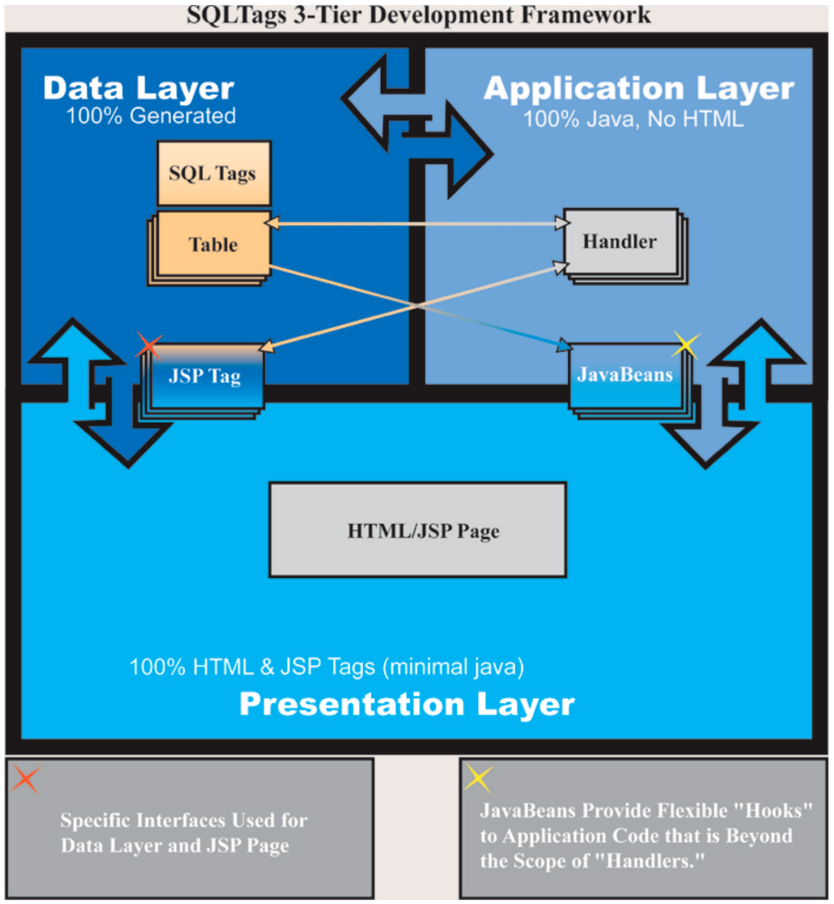
```
<sqltags:dept
    id="d"
    buttonName="fop"
    properties="true"
    />
```

Presentation Layer

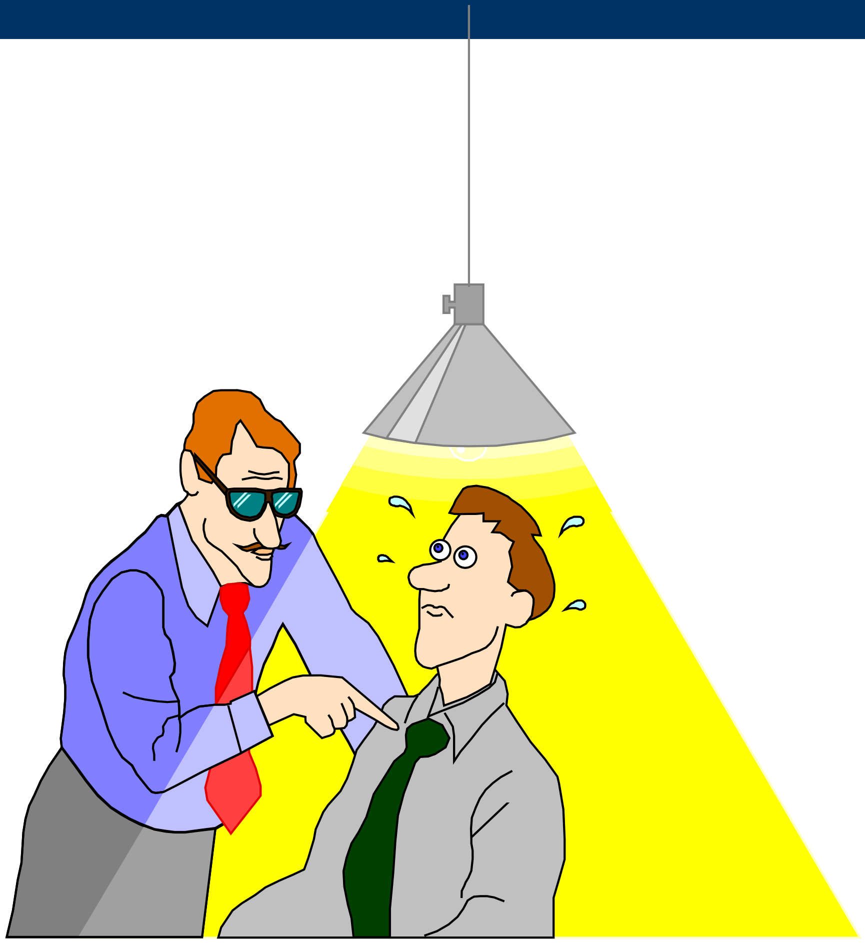
- Support Tags
  - **Where** - defines complex where clause
  - **OrderBy** - defines order by clause on "join"
  - **Fetch** - Fetch Rows
  - **First** - First on page
  - **Last** - Last on page
- Additional Tags
  - **Connect** - Connect to database using pool
  - **Cursor** - generic cursor tag for complex queries
  - **Previous** - previous page Anchor
  - **Next** - next page Anchor

SQLTags Framework

- Data Layer
  - SQLTags Classes
  - 100% Generated
- Application Layer
  - SQLTagsHandler
  - JavaBeans
- Presentation Layer
  - HTML
  - SQLTags JSP TAGS


SQLTags 3-Tier Development Framework
Data Layer
100% Generated
Application Layer
100% Java, No HTML
SQL Tags
Table
Handler
JSP Tag
JavaBeans
HTML/JSP Page
100% HTML & JSP Tags (minimal java)
Presentation Layer
Specific Interfaces Used for Data Layer and JSP Page
JavaBeans Provide Flexible "Hooks" to Application Code that is Beyond the Scope of "Handlers."

Questions?



- SQLTags …
  - Data, Application, Presentation
- Java/JSP …
- User Interface …

Part 2:
Using SQLTags

Part 2: Agenda

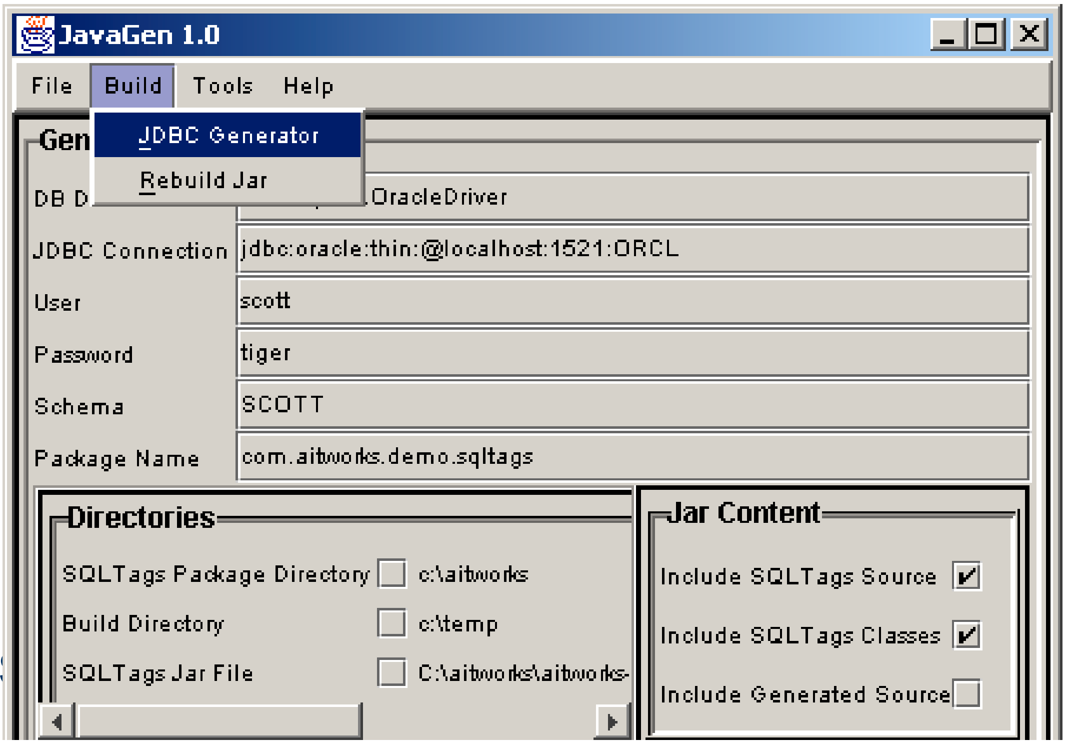
- Overview
- Generator
- JavaServer Pages: Essential Components
- Querying the Database
- Data Manipulation
- Helpful Features
- Built-in Tags

Part 2:Overview

- Run Generator
  - Define package, identify Schema
  - Create Jar file
- Setup Deployment Descriptor (web.xml)
  - Define connection pool properties
  - DataSource
- Create JSP Pages
  - Reference packages,
  - Jar file
- Query Database
  - Simple Table
  - Master-Detail
  - Foreign Key Lookup
  - Cursor
- Modify Data
  - Operation
  - Arrays
  - Using Properties=true

SWING Generator

- Database Driver-- requires JDBC MetaData
- JDBC URL
- User, Password
- Schema
- Package
- Directories
  - SQLTags
  - Build
- Jar File & Content



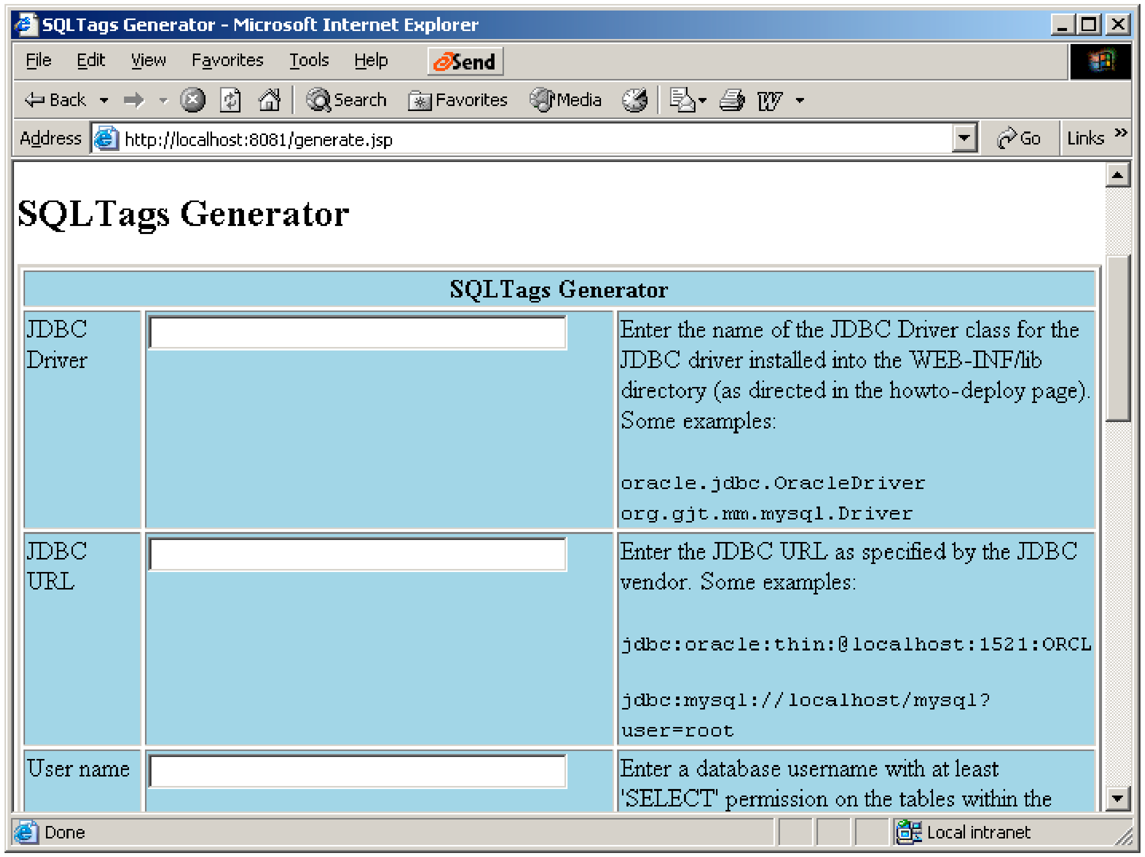
DB D OracleDriver
JDBC Connection jdbc:oracle:thin:@localhost:1521:ORCL
User scott
Password tiger
Schema SCOTT
Package Name com.aitworks.demo.sqltags

Directories
SQLTags Package Directory c:\aitworks
Build Directory c:\temp
SQLTags Jar File C:\aitworks\aitworks-

Jar Content
Include SQLTags Source ✓
Include SQLTags Classes ✓
Include Generated Source

HTTP Generator

- Database Driver-- requires JDBC MetaData
- JDBC URL
- User, Password
- Schema
- Package
- Jar File
- Options


SQLTags Generator - Microsoft Internet Explorer
File Edit View Favorites Tools Help Send
Back Search Favorites Media
Address http://localhost:8081/generate.jsp
Go Links
Done
Local intranet

SQLTags Generator

SQLTags Generator		
JDBC Driver		Enter the name of the JDBC Driver class for the JDBC driver installed into the WEB-INF/lib directory (as directed in the howto-deploy page). Some examples: `oracle.jdbc.OracleDriver` `org.gjt.mm.mysql.Driver`
JDBC URL		Enter the JDBC URL as specified by the JDBC vendor. Some examples: `jdbc:oracle:thin:@localhost:1521:ORCL` `jdbc:mysql://localhost/mysql?user=root`
User name		Enter a database username with at least 'SELECT' permission on the tables within the

Generator Notes

- **JDBC Drivers**
  - for Oracle, MySQL included in the "Tar" file
  - More on the SQLTags CD-ROM
  - Links on SQLTags web site
- **Package Name**
  - location for generated table and tag classes
  - com.aitworks.sqltags.demo.sqltags (examples)
- **SQLTags Output ".jar" File**
  - generates to /WEB-INF/tmp
- **Contents**
  - SQLTags Classes for single Jar
  - Source code, too

JSP Components

- Deployment -- JDBC Connection Pool
- Taglib-- Generated Jar

```
<%@ taglib uri="demoTags.jar” prefix=”demo” %>
```

- Connection-- wraps all Tags, ref Properties

```
<sqltags:connection id="connect">
   . . .
</sqltags:connection>
```

Deployment Descriptor

- The J2EE Deployment Descriptor for a standard Web Application is the file: WEB-INF/web.xml
- The Deployment Descriptor contains installation specific parameters for each web application.
- SQLTags “Context Parameters” example:

```
<context-param>
   <param-name>SQLTags.userName</param-name>
   <param-value>scott</param-value>
</context-param>
```

Deployment Descriptor

SQLTags.bindStrings

Should almost always be set to false. Default is true. Controls the way in which variables are bound to SQL statements for inserts and updates and in the where clauses for selects, updates, and deletes. if bindStrings = true all bindings are done using the setString (PreparedStatement). If bindStrings = false Numerical and Data types are bound using native Java Number and Date types. bindStrings = false is recommended. Default is true for backward compatibility.

SQLTags.useCM

Valid values: true or false; default is true. Controls use of either the internal <SQLTags:> JDBC Connection Pool or an external DataSource maintained by the Application Server. True indicates the internal Connection Pool is used referencing the JDBC context params (below), false indicates an external DataSource will be used instead and the related context params (below) will be ignored.

SQLTags.dataSource

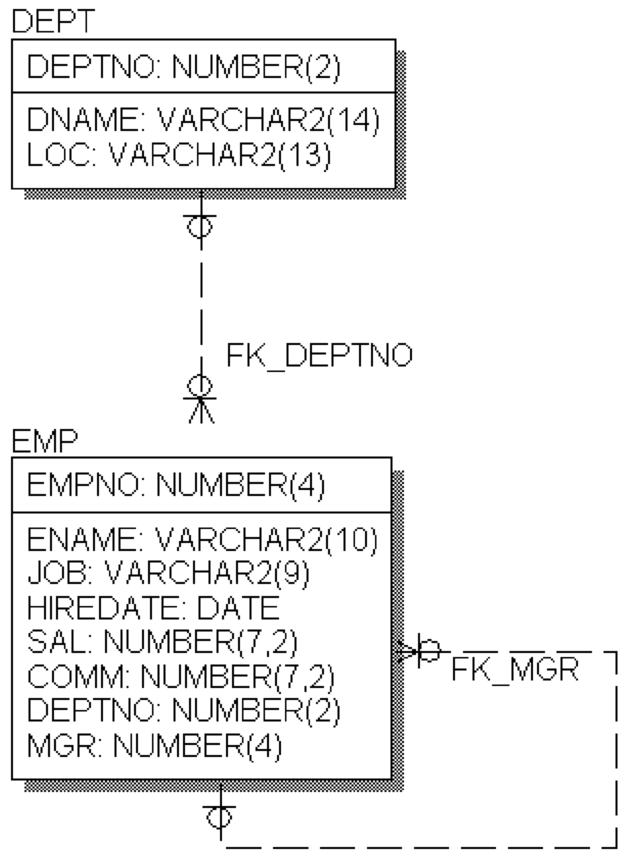
Defines the default DataSource to be used when useCM (above) is false and no dataSource attribute is included on the connections when useCM is false. Default value is `jdbc/SQLTagsDS`.

The following Context Params are only needed if SQLTags.useCM is set to true.

SQLTags.databaseDriver	Database driver class as provided by JDBC vendor.
SQLTags.connectionUrl	The JDBC connection URL as defined by the JDBC Vendor.
SQLTags.maxPoolSize	Maximum number of JDBC connections allowed in the pool.
SQLTags.poolSize	Normal/starting size of JDBC connection pool.
SQLTags.userName	Database username used to connect to the database.
SQLTags.password	Password for SQLTags.userName.

Example Data Model

- The "Classic"
  - EMP - DEPT Example
- Critical Components
  - Table Names
  - Column Names
- Equally as important
  - Foreign Key Names
  - FK_DEPTNO
  - FK_MGR



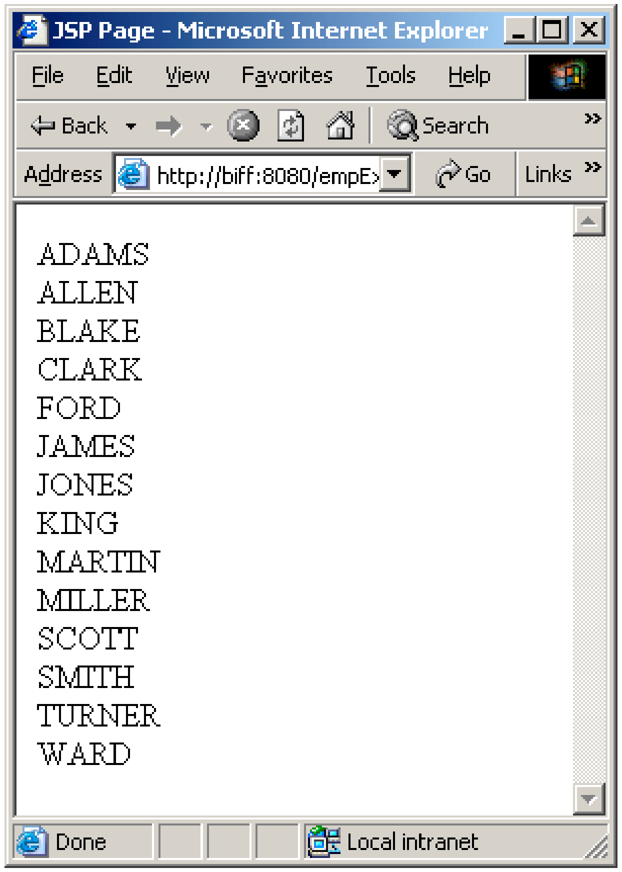
Querying the Database

- SQLTags can query the database in several ways:
  - Single Table Query
  - Master-Detail Nesting
  - Parent Lookup
  - Free-form, generic Cursor tag
- Complex queries are supported with nested <where> and <fetch> tags

Single Table Queries

- Simple Example

```
<%@ taglib uri="demoTags.jar" prefix="sqltags" %>
<html>
<head><title>JSP Page</title></head>
<body>
<sqltags:connection id="connect">

    <sqltags:emp id="e" where="order by ENAME">
        <%= e.getENAME() %><BR>
    </sqltags:emp>

</sqltags:connection>
</body>
</html>
```

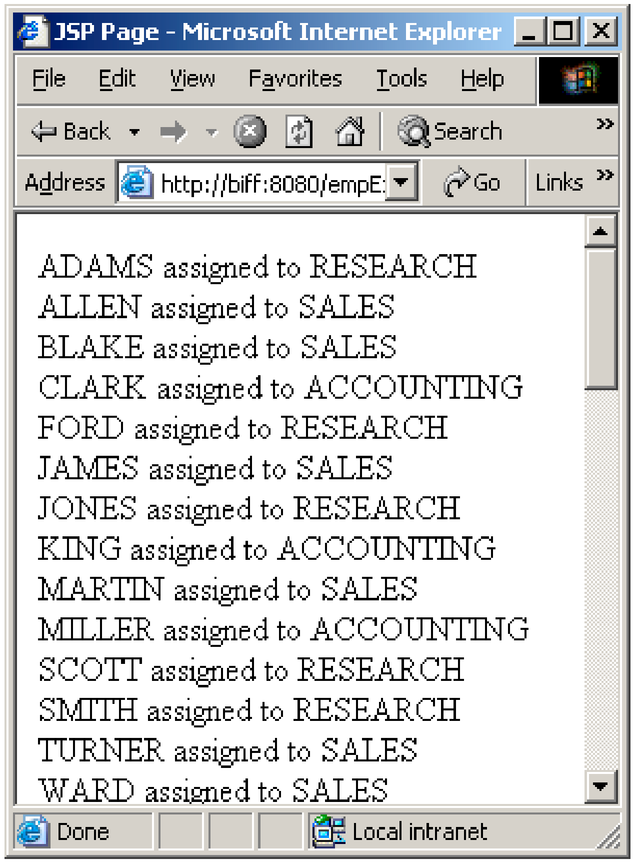

JSP Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search
Address http://biff:8080/empE
Go Links
Done
Local intranet

```
ADAMS
ALLEN
BLAKE
CLARK
FORD
JAMES
JONES
KING
MARTIN
MILLER
SCOTT
SMITH
TURNER
WARD
```

Parent FK Queries

- Parent Lookup Example

```
<%@ taglib uri="demoTags.jar" prefix="sqltags" %>
<html>
<head><title>JSP Page</title></head>
<body>
<sqltags:connection id="connect">

      <sqltags:emp id="e" where="order by ename">
         <%= e.getENAME() %> assigned to
         <%= e.getFK_DEPTNO_PARENT().getDNAME() %>
         <BR>
      </sqltags:emp>

</sqltags:connection>
</body>
</html>
```


JSP Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search
Address http://biff:8080/empE: Go Links
Done
Local intranet

```
ADAMS assigned to RESEARCH
ALLEN assigned to SALES
BLAKE assigned to SALES
CLARK assigned to ACCOUNTING
FORD assigned to RESEARCH
JAMES assigned to SALES
JONES assigned to RESEARCH
KING assigned to ACCOUNTING
MARTIN assigned to SALES
MILLER assigned to ACCOUNTING
SCOTT assigned to RESEARCH
SMITH assigned to RESEARCH
TURNER assigned to SALES
WARD assigned to SALES
```

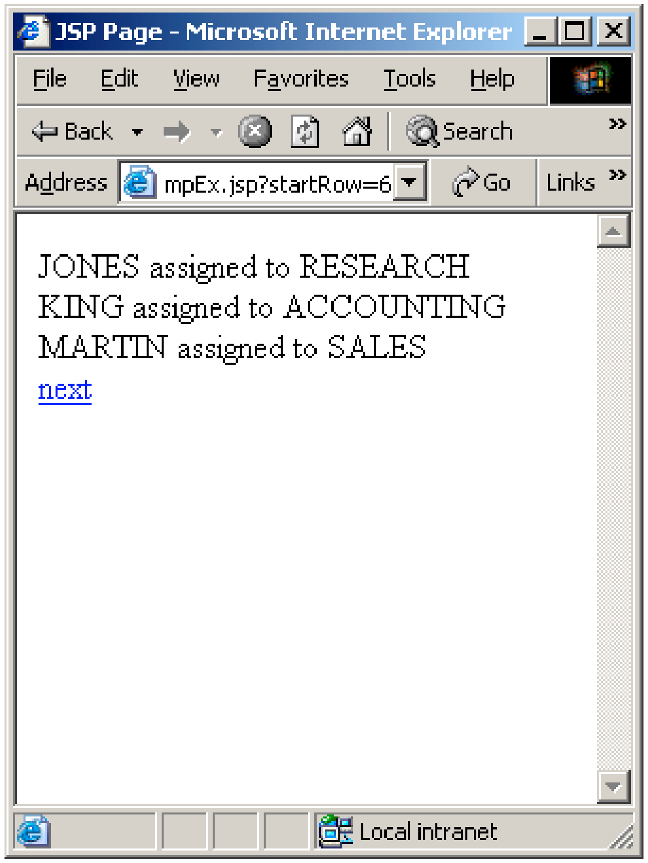
Paging Support

- Paging Support is Built-in Feature
  - Set paging Attribute to “true”
  - Set displaySize Attribute to desired page size
  - Add a <Last> tag to identify last row or query
  - Use <previous> and <next> tags to build links to next and previous pages
- Example …

Paging Support

- Paging Example

```
<%-- edited for space --%>
<html>
<head><title>JSP Page</title></head>
<body>
<sqltags:connection id="connect">

      <sqltags:emp paging="true" displaySize="3"
             id="e" where="order by ename">
         <%= e.getENAME() %> assigned to
         <%= e.getFK_DEPTNO_PARENT().getDNAME() %>
         <BR>
         <sqltags:last name="e">
            <sqltags:next href="empEx.jsp"
                  parentName="e">next
            </sqltags:next>
         </sqltags:last>
      </sqltags:emp>

</sqltags:connection>
<%-- edited for space --%>
```

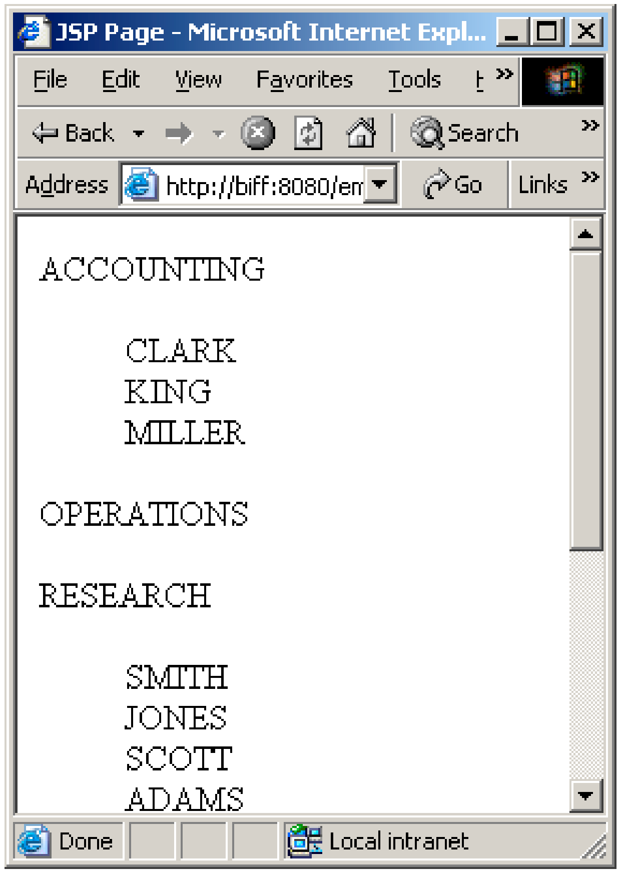

JSP Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search
Address mpEx.jsp?startRow=6 Go Links
Local intranet

JONES assigned to RESEARCH
KING assigned to ACCOUNTING
MARTIN assigned to SALES
next

Master-Detail Queries

- Master-Detail Example

```
<%-- edited for space --%>
<sqltags:dept id="d" where="order by DNAME">
   <%= d.getDNAME() %>

   <BLOCKQUOTE>
   <sqltags:emp id="e" foreignKey="FK_DEPTNO"
              parentName="d">
     <%= e.getENAME() %><BR>
   </sqltags:emp>
   </BLOCKQUOTE>

</sqltags:dept>
<%-- edited for space --%>
```


JSP Page - Microsoft Internet Expl...
File Edit View Favorites Tools
Back Search
Address http://biff:8080/em Go Links
Done Local intranet

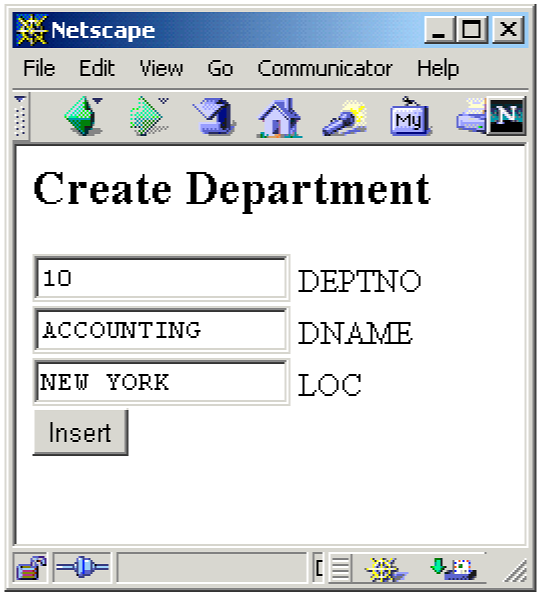
```
ACCOUNTING

      CLARK
      KING
      MILLER

OPERATIONS

RESEARCH

      SMITH
      JONES
      SCOTT
      ADAMS
```

Data Manipulation

- DML can be Specified as Follows:
  - Operation Attribute
    - <sqltags:emp id="e" operation="insert" />
    - Executes specified operation
  - ButtonName Attribute
    - <sqltags:emp id="e" buttonName="button" />
    - Executes operation contained in button Parameter
  - Direct Call via Snippet Variable
    - <% e.insert(); e.update(); e.delete(); %>

Request Properties

```
<h2>Create Department</h2>
<FORM>
       <INPUT NAME="DEPTNO" SIZE="15" VALUE="10"> DEPTNO
    <BR><INPUT NAME="DNAME"  SIZE="15" VALUE="ACCOUNTING"> DNAME
    <BR><INPUT NAME="LOC"    SIZE="15" VALUE="NEW YORK"> LOC
    <BR><INPUT NAME="fop" TYPE="submit" VALUE="Insert">
</FORM>
```



Create Department

10	DEPTNO
ACCOUNTING	DNAME
NEW YORK	LOC

Since the “properties” attribute is “true” and the **field names in the Form match the column names** from the DEPT table the values will automatically be assigned.

Processed By

```
<sqltags:dept
    id=”d”
    buttonName=”fop”
    properties=”true”
    />
```

Array Processing

- SQLTags supports multi-row Update, Inserts, and Delete
- Rows of Inputs are named as follows:
  - columnName[arrayIndex]
    - ENAME[1]="Steve"; ENAME[2]="Joe"; …
    - `<input name="ename[1]" value="Steve">`
- Operation Names Match arrayIndex
  - operation[arrayIndex]
    - operation[1]="insert"; operation[2]="delete";
- Each Operation Controls one Row

Handler Classes

- SQLTags Provides “Database Events” for User-Defined Validation and Business Logic Support
  - preInsert, postInsert
  - preUpdate, postUpdate
  - preDelete, postDelete
  - preQuery, postFetch, postQuery
- If a “pre” event Fails, operation is halted
- SQLTags Provides “Tag Events”
  - start, afterBody, end

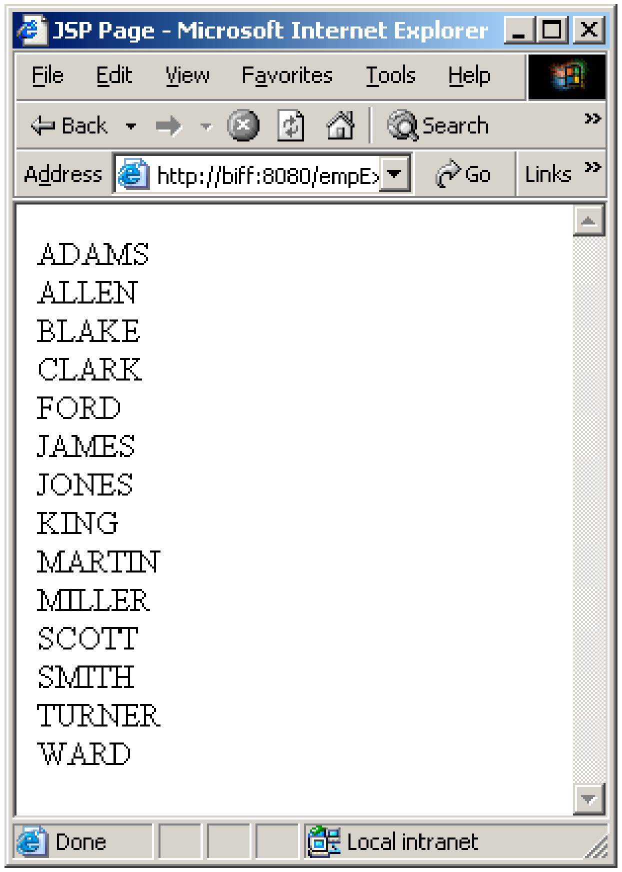
Support Tags

- Paging Support
  - <first>, <last>, <next>, <previous>
- Complex Queries
  - <where>, <orderBy>, <fetch>, <cursor>, <statement>
- Connection, Initialization
  - <connection>, <initialize>
- Authorization
  - <authorize>

Cursor Tag

- Cursor Example

```
<%@ taglib uri="demoTags.jar" prefix="sqltags" %>
<html>
<head><title>JSP Page</title></head>
<body>
<sqltags:connection id="connect">

  <sqltags:cursor id="cur"
           sql="select * from EMP order by ENAME">
        <sqltags:fetch name="cur">
          <%= cur.getString("ENAME") %><BR>
        </sqltags:fetch>
  </sqltags:cursor>

</sqltags:connection>
</body>
</html>
```

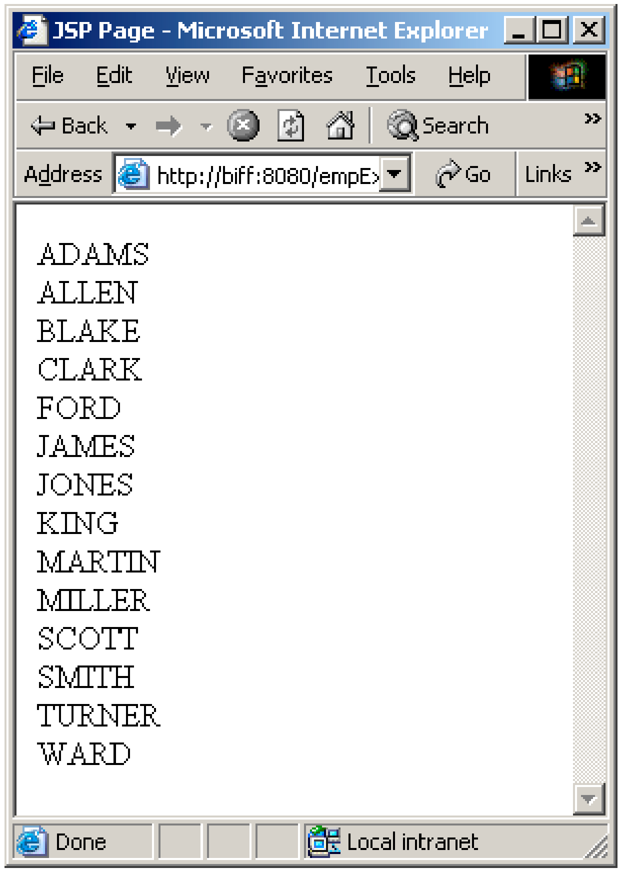

JSP Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search
Address http://biff:8080/empE Go Links
Done
Local intranet

```
ADAMS
ALLEN
BLAKE
CLARK
FORD
JAMES
JONES
KING
MARTIN
MILLER
SCOTT
SMITH
TURNER
WARD
```

Statement,Where,Fetch

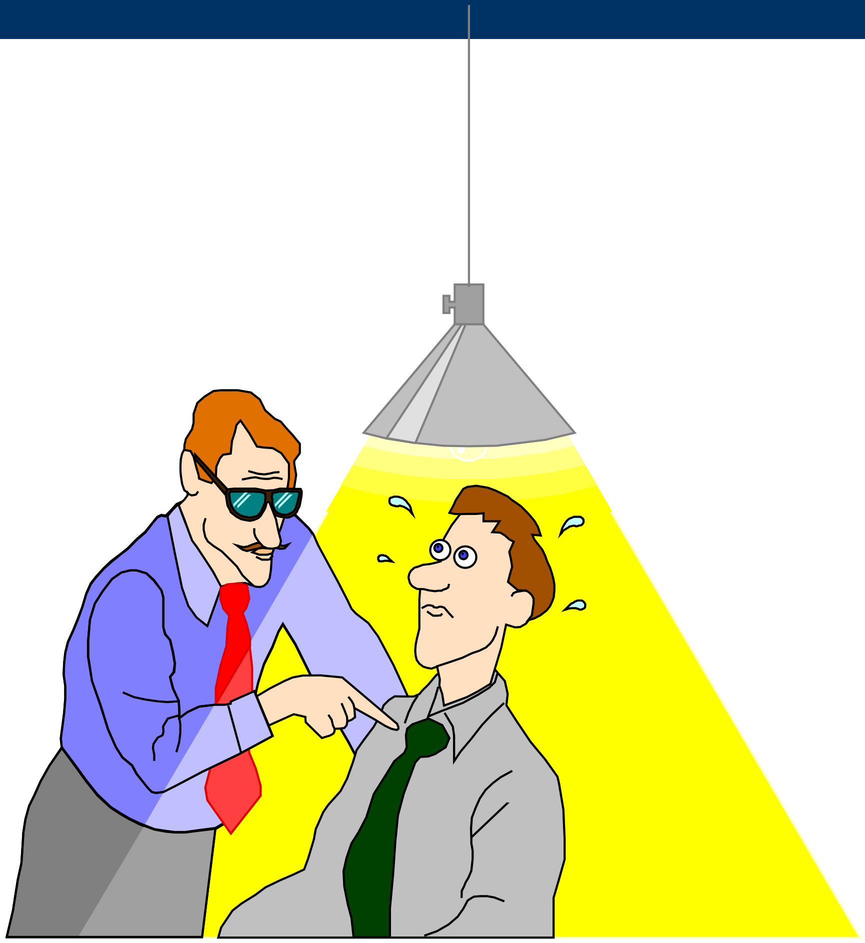
- Statement, Where, Fetch

```
<%-- edited for space --%>
<sqltags:connection id="connect">

  <sqltags:cursor id="cur">
        <sqltags:statement name="cur">
         select * from EMP
         order by ENAME
        </sqltags:statement>
        <sqltags:fetch name="cur">
          <%= cur.getString("ENAME") %><BR>
        </sqltags:fetch>
  </sqltags:cursor>

</sqltags:connection>
<%-- edited for space --%>
```


JSP Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search
Address http://biff:8080/empEx Go Links
Done Local intranet

```
ADAMS
ALLEN
BLAKE
CLARK
FORD
JAMES
JONES
KING
MARTIN
MILLER
SCOTT
SMITH
TURNER
WARD
```

Questions?



- Database/JDBC …
- Java/JSP …
- User Interface …

- Generator
- JSP Development