Applicazioni web three-tier

- Three-tier distributed applications
  - User interface
  - Business logic
  - Database access
- Il Web container è impiegato per realizzare il middle tier
- Un esempio da Deitel “Java How To Program” ch.26
  - SurveyServlet
  - Survey.html
  - MySQL database
CREATE DATABASE IF NOT EXISTS animalsurvey;
USE animalsurvey;
DROP TABLE IF EXISTS surveyresults;
CREATE TABLE surveyresults (  id INT NOT NULL ,  surveyoption varchar (20) NOT NULL ,  votes INT NOT NULL ,  PRIMARY KEY (id)) TYPE=INNODB ;  
insert into surveyresults (id,surveyoption,votes) values (1, 'Dog', 0);
insert into surveyresults (id,surveyoption,votes) values (2, 'Cat', 0);
insert into surveyresults (id,surveyoption,votes) values (3, 'Bird', 0);
insert into surveyresults (id,surveyoption,votes) values (4, 'Snake', 0);
insert into surveyresults (id,surveyoption,votes) values (5, 'None', 0);
A Web-based application that uses JDBC

// SurveyServlet.java
// A Web-based survey that uses JDBC from a servlet
package it.unina.at;

import java.io.*;
import java.text.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class SurveyServlet extends HttpServlet {
    private Connection connection;
    private PreparedStatement updateVotes, totalVotes, results;

    // set up database connection and prepare SQL statements
    public void init( ServletConfig config )
        throws ServletException
    {
        // attempt database connection and create PreparedStatements
        try {
            Class.forName( "com.mysql.jdbc.Driver" );
            connection = DriverManager.getConnection( "jdbc:mysql://localhost:3306/animalsurvey" );
        } catch (

A Web-based application that uses JDBC (2)

```java
// PreparedStatement to add one to vote total for a specific animal
updateVotes = connection.prepareStatement(
    "UPDATE surveyresults SET votes = votes + 1 " +
    "WHERE id = ?"
); 

// PreparedStatement to sum the votes
totalVotes = connection.prepareStatement(
    "SELECT sum( votes ) FROM surveyresults"
); 

// PreparedStatement to obtain surveyoption table's data
results = connection.prepareStatement(
    "SELECT surveyoption, votes, id " +
    "FROM surveyresults ORDER BY id"
); 
```

}
A Web-based application that uses JDBC (3)

```java
// for any exception throw an UnavailableException to
// indicate that the servlet is not currently available
try {
    // set up response to client
    response.setContentType("text/html");
    PrintWriter out = response.getWriter;
    DecimalFormat twoDigits = new DecimalFormat("0.00");
}
```
A Web-based application that uses JDBC (4)

66 // start XHTML document
67 out.println( "<?xml version = "1.0"?>" );
68
69 out.println( "<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">" );
70
71 out.println( "<html xmlns = "http://www.w3.org/1999/xhtml">" );
72
73 // head section of document
74 out.println( "<head>" );
75
76 // read current survey response
77 int value =
78     Integer.parseInt( request.getParameter( "animal" ) );
79 80 81
A Web-based application that uses JDBC (5)

```java
// attempt to process a vote and display current results
try {
    // update total for current survey response
    updateVotes.setInt(1, value);
    updateVotes.executeUpdate();

    // get total of all survey responses
    ResultSet totalRS = totalVotes.executeQuery();
    totalRS.next();
    int total = totalRS.getInt(1);

    // get results
    ResultSet resultsRS = results.executeQuery();
    out.println("<title>Thank you!</title>");
    out.println("</head>");
    out.println("</head>");
```
A Web-based application that uses JDBC (6)

```java
out.println("<body>" );
out.println("<p>Thank you for participating." );
out.println("<br /></p><pre>

// process results
int votes;

while ( resultsRS.next() ) {
    out.print( resultsRS.getString(1) );
    out.print(": ");
    votes = resultsRS.getInt(2);
    out.print( twoDigits.format((double) votes / total * 100) );
    out.print("% responses: ");
    out.println( votes );
}

resultsRS.close();

out.print("Total responses: ");
out.println( total );

// end XHTML document
out.println("</pre></body></html>" );
out.close();
```
A Web-based application that uses JDBC (7)

126 // if database exception occurs, return error page
127     catch ( SQLException sqlException ) {
128         sqlException.printStackTrace();
129         out.println( "<title>Error</title>" );
130         out.println( "</head>" );
131         out.println( "<body><p>Database error occurred. " );
132         out.println( "Try again later.</p></body></html>" );
133         out.close();
134     }
135 } // end of doPost method
136
137 } // end of doPost method
public void destroy() {
    // attempt to close statements and database connection
    try {
        updateVotes.close();
        totalVotes.close();
        results.close();
        connection.close();
    } catch (SQLException sqlException) {
        sqlException.printStackTrace();
    }

    // handle database exceptions by returning error to client
    // end of destroy method
<?xml version = "1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title>Survey</title></head>
<body>
<form method = "post" action = "/AT_servlets/animalsurvey">
  <p>What is your favorite pet?</p>
  <p>
    <input type = "radio" name = "animal" value = "1" />Dog</p>
    <input type = "radio" name = "animal" value = "2" />Cat
    <input type = "radio" name = "animal" value = "3" />Bird
    <input type = "radio" name = "animal" value = "4" />Snake
    <input type = "radio" name = "animal" value = "5" checked = "checked" />None
  </p>
  <p><input type = "submit" value = "Submit" /></p>
</form></body>
</html>
Esempi

Image of two web pages showing a survey and the results.

Survey Question:
What is your favorite pet?
- Dog
- Cat
- Bird
- Snake
- None

Submit button.

Survey Results:
Thank you for participating.
Results:

Dog: 57.89% responses: 22
Cat: 23.68% responses: 9
Bird: 10.53% responses: 4
Snake: 5.26% responses: 2
None: 2.63% responses: 1
Total responses: 38