E-Commerce Database and Servlet Introduction

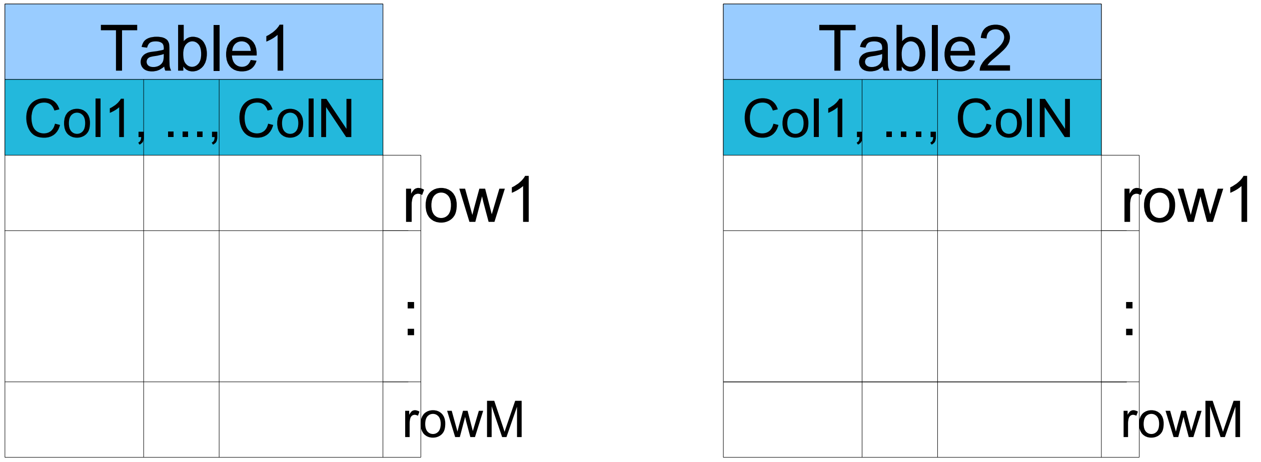
Jim Wilenius

UPPSALA UNIVERSITET

Informationsteknologi

Database Basics

- What is a Database?
  - Stores information (as one or more files)
  - Organized logically in Tables
  - Columns and Rows

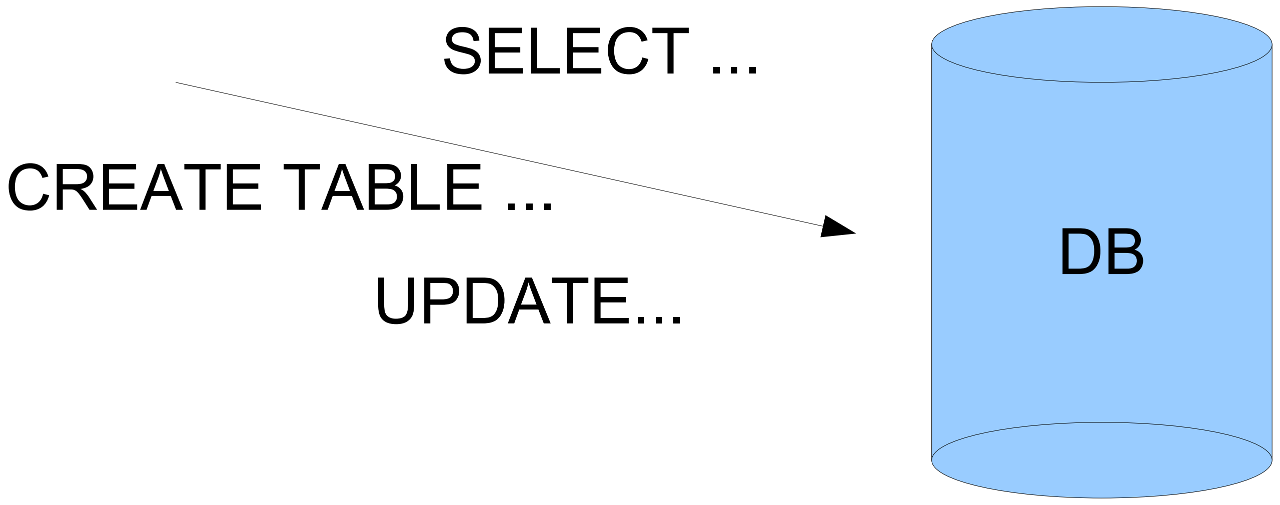


UPPSALA
UNIVERSITET

Informationsteknologi

Database Basics

- Actions
  - Create Tables
  - Ask Questions (query) with criteria
  - Change information



UPPSALA UNIVERSITET

Informationsteknologi

Database Basics

- Example table:
  - Person {name, age, pNo}

Arne	young	123456
Jim	infant	234567
Kim	infant	345678

  - Query:

```
SELECT name, pNo FROM Person WHERE age = 'infant'
```

returns the record set:

    Jim 234567
    Kim 345678

UPPSALA UNIVERSITET

Informationsteknologi

Database Basics

- SQL – structured query language
  - ANSI Standard for accessing Databases
  - DML – Data Manipulation Language
    - SELECT
    - UPDATE
    - DELETE
    - INSERT INTO
  - DDL – Data Definition Language
    - CREATE TABLE
    - DROP TABLE
    - ALTER TABLE
    - CREATE INDEX
    - DROP INDEX

UPPSALA UNIVERSITET

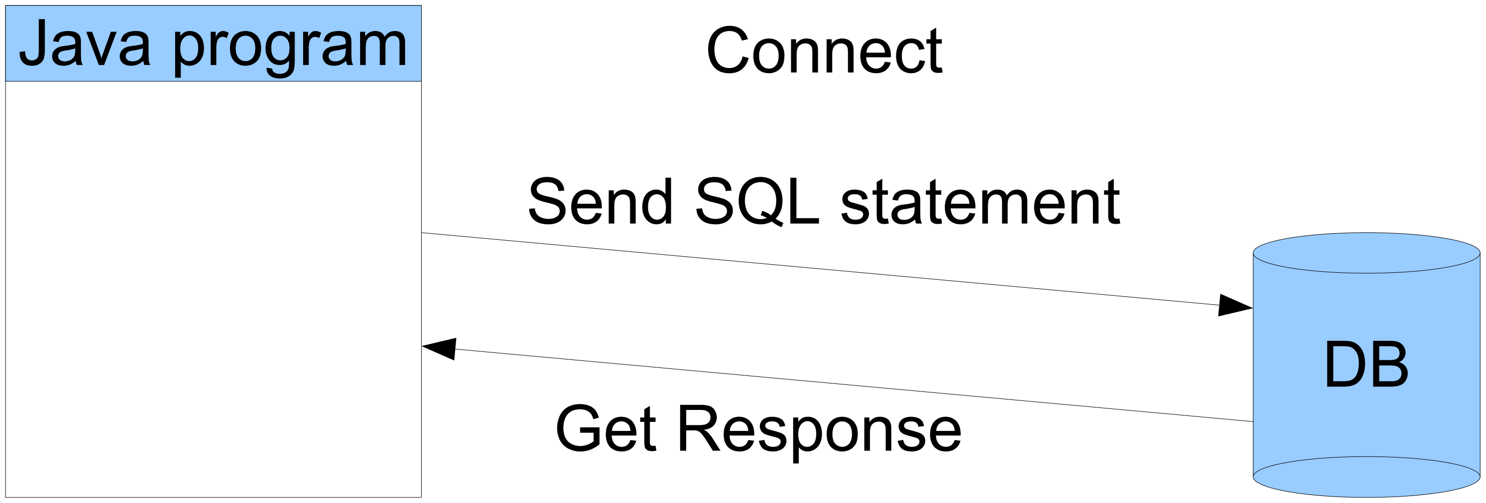
Informationsteknologi

Database Basics

- You now know the basic ideas...
  - What a DB is
  - How data is organized
  - That you can manipulate or query data
- How do you actually USE it in a practical application?

Database and Java

- JAVA – JDBC (java database connectivity)
  - Load the database driver
  - Open a connection to the database
  - Use connection to create/execute statements
  - Get a result-set from the db (when applicable).



UPPSALA UNIVERSITET

Informationsteknologi

Database and Java

- Load driver
  - Class.forName("org.hsqldb.jdbcDriver");
  - Throws ClassNotFoundException
- Create Connection – to myDB
  - ***con*** = DriverManager.getConnection(
    "jdbc:hsqldb:file:***myDB***",
    "sa", "");
  - Throws SQLException

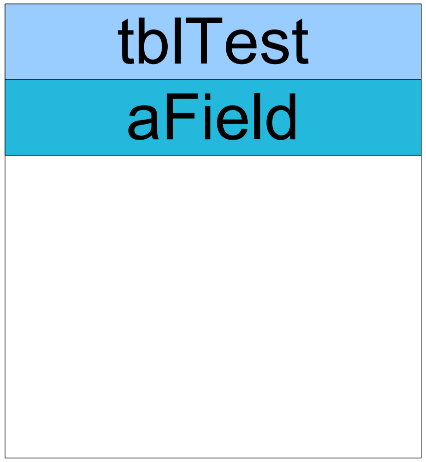
UPPSALA UNIVERSITET

Informationsteknologi

Database and Java

- Use the connection ***con*** to create a table

```
Statement stmt = con.createStatement();
String e = "CREATE TABLE tblTest ( aField VARCHAR)"
stmt.executeUpdate(e);
stmt.close();
```



  - This will create the initial db file with one table
  - Information can be added to the database

UPPSALA UNIVERSITET

Informationsteknologi

Database and Java

- Use the connection ***con*** to add information

```
Statement stmt = con.createStatement();
String e = “INSERT INTO tblTest (aField) VALUES
('test')”
stmt.executeUpdate(e);
stmt.close();
```

- Do it again: with 'hej'

- The table will have 2 rows

tblTest
aField
test
hej

UPPSALA UNIVERSITET

Informationsteknologi

Database and Java

- Use the connection ***con*** to add information

```
PreparedStatement p = con.prepareStatement(
    "INSERT INTO tblTable ( aField ) VALUES( ? )" );
p.setString(1, “saippuakauppias”);
p.executeUpdate();
p.close();
```

tblTest
aField
test
hej
saippuakauppias

- The table, from before, will now have 3 rows
- Practical when setting binary data p.setBytes(N, myByteArray)

UPPSALA UNIVERSITET

Informationsteknologi

Database and Java

- Use the connection ***con*** to get information

```
Statement s = con.createStatement(
    "SELECT * FROM tblTable" );
ResultSet r = s.executeQuery();
... use r ...
s.close();
```

tblTest
aField
test
hej
saippuakauppias

- The ResultSet will contain the three rows.
- r.next() - go to next row
- r.getXXXX() - methods to access columns
- Read the javadoc it is very helpful!

UPPSALA UNIVERSITET

Informationsteknologi

Database and Java

- Close everything nicely
  - Execute a statement with “SHUTDOWN”
  - Close the ***con*** object con.close();

UPPSALA UNIVERSITET

Informationsteknologi

Database and Java

- You now have the basic knowledge of
  - How to connect to a DB in java
  - How to create a statement and execute it
  - How to close the DB connection
- Use the provided links and google
  - More details about SQL syntax
  - Other types of statements (javadoc)
  - Field Types
    - VARCHAR
    - INTEGER
    - VARBINARY
    - ... (see HSQLDB documentation, Data Types)

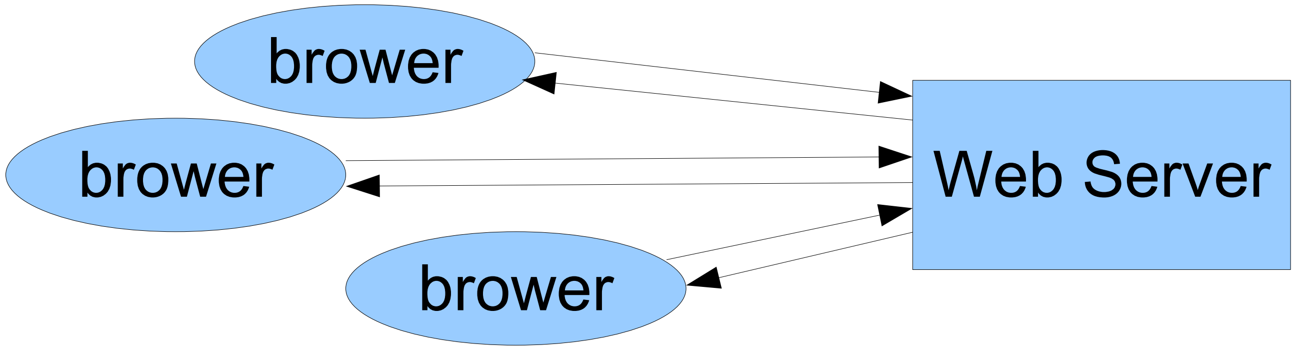
UPPSALA UNIVERSITET

Informationsteknologi

Web Servers

- The software that serves the web pages you see when you browse the Internet

- Many clients (web browsers)
- One server (web server)
  - Responds to clients requests for HTML pages



Web Servers

- Static Server
  - Serves only static HTML pages (somefile.html)
  - Not very powerful / useful
- Tomcat (and others)
  - Static HTML
  - Dynamic pages
    - Created when the request is made from the brower
    - JSP / Servlets

UPPSALA UNIVERSITET

Informationsteknologi

Web Server – dynamic page

- JSP
  - Dynamic contents in HTML file
  - Files are name .jsp instead of .html
    - Example: example.jsp

```
<HTML>
<BODY>
The time is <%= new java.util.Date() %>
</BODY>
</HTML>
```

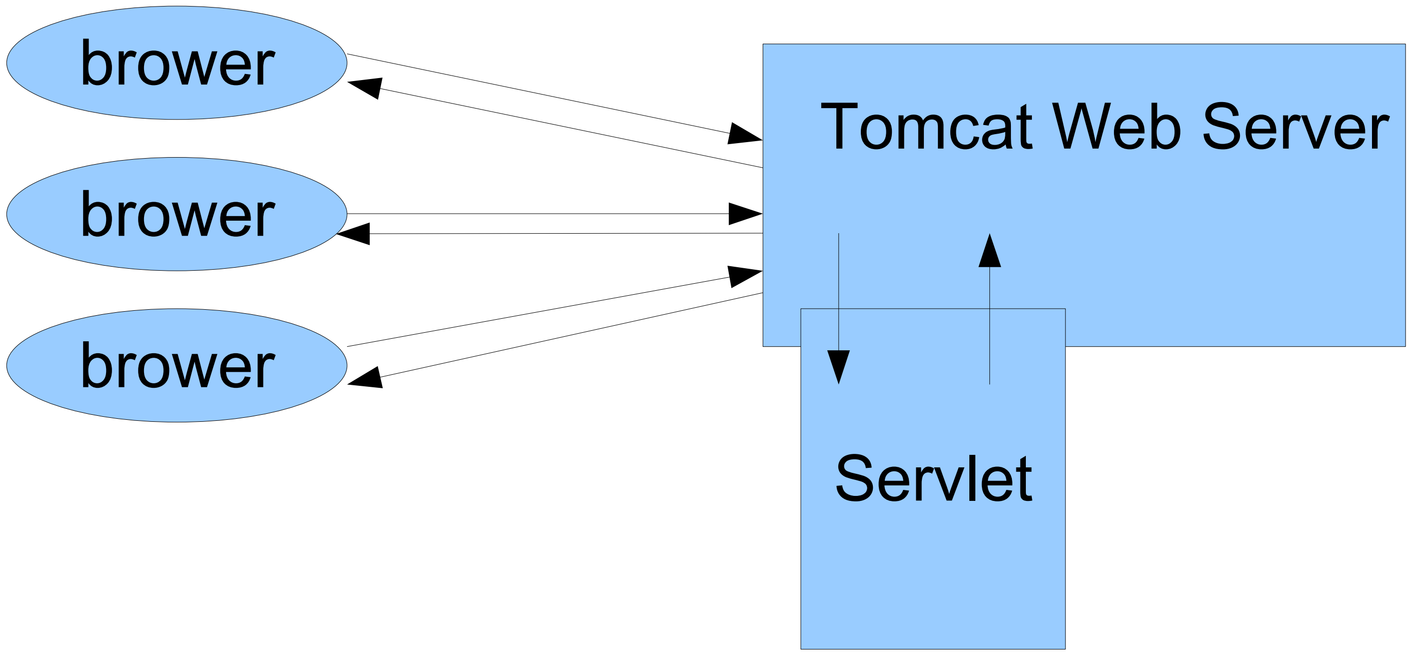
  - Evaluated each time the page is requested

UPPSALA UNIVERSITET

Informationsteknologi

Web Server – Servlets

- Servlets – java programs
  - Called by the Tomcat server



UPPSALA UNIVERSITET

Informationsteknologi

Web Server – Servlets

- Setting up a Servlet in tomcat
  - See the lab instructions for directory structure
- *web.xml* – how Tomcat knows what to do

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app>

   <servlet>
      <servlet-name>aTestServlet</servlet-name>
      <servlet-class>MyServletClass</servlet-class>
   </servlet>

   <servlet-mapping>
      <servlet-name>aTestServlet</servlet-name>
      <url-pattern>/myservlet</url-pattern>
   </servlet-mapping>

</web-app>
```

Servlets

- What is a Servlet really...
  - Normal java program (a class)
  - No `public static void` main(...) required
  - *default constructor* MUST exist
  - Extends the HTTPServlet class
  - Override methods to customize Servlet
    - doGet(...)
    - doPost(...)
    - Init(...)
    - destroy(...)

UPPSALA UNIVERSITET

Informationsteknologi

Servlets - access

- Ways of accessing the Servlet
  - An URL http://server.com/myservlet
  - With parameters – append *?paramName=value&...*

```
<html> <body>
 <!-- user inputs parameter in a form -->
 <form name="input" method="get"
       action="myservlet">
 <input type="text" name="Param2">
 <input type="submit" value="Submit">
 </form>

 <!-- fixed parameter -->
 <a href="myservlet?param1=v1&param2=v2">
  fixedParameter
 </a>

</body> </html>
```

UPPSALA UNIVERSITET

Informationsteknologi

Servlets - customizing

- *Override methods*
  - **doGet**(...) $\longrightarrow$ - called on each request of the page
  - doPost(...) $\nearrow$ which one, depends on the request
  - Init(...) - called once when Servlet is loaded
  - destroy(...) - called once when Servlet is stopped
- *doGet(...)* and *doPost(...)*
  - *doGet(...) is the default*
  - 2 parameters
  - *HttpServletRequest*
  - *HttpServletResponse*

UPPSALA UNIVERSITET

Informationsteknologi

Servlets

- *HttpServletRequest* object
  - Contains request information
  - Form/fixed parameters
  - Use the .getParameter(“paramName”) when using forms or fixed parameters.
  - Has many useful methods, read the api documentation for useful information.

UPPSALA UNIVERSITET

Informationsteknologi

Servlets

- *HttpServletResponse* res
  - res.getWriter() returns a PrintWriter
    - Use to output your resulting HTML text.
    - Example:
      PrintWriter out = res.getWriter();
      out.println(“<html><body> static servlet page </body></html>”);
  - Contains all HTML headers and error codes see the servlet-api documentation

UPPSALA UNIVERSITET

Informationsteknologi

Servlets

- init() and init(ServletConfig s)
  - Override this method to do Load-time initialization. (eg. create connection to DB)
  - Initialize state variables
  - ServletConfig has parameter values from the web.xml for a specific servlet.

UPPSALA UNIVERSITET

Informationsteknologi

Servlets

- Example: init(ServletConfig c)
  in the web.xml add to the <servlet> tag

```
    ...
    <servlet>
      <servlet-name>aTestServlet</servlet-name>
      <servlet-class>MyServletClass</servlet-class>
      <init-param>
        <param-name>testParam</param-name>
        <param-value>theValue</param-value>
    </init-param>
    </servlet>
    ...
```

- Access from ServletConfig c
  - c.getParameter(“testParam”)

UPPSALA UNIVERSITET

Informationsteknologi

Servlets

- destroy()
  - Called when the servlet is stopped. (close the DB gracefully)