12 – SQL Basics

LBSCI 700 | Spring 2019

Queens College, CUNY

12-sql.pdf

SQL: Intro

SQL Query and Data Retrieval

Exercises

Digital Libraries and Dynamic Websites

Interactivity of Systems

SQL Intro

SQL Intro

- What is SQL?
  --Not a programming language – a Query Language
  --Non-procedural language

- Why SQL?
  --de facto standard for RDB
  --Relatively easy to learn (vocabulary of less than 100 words)

SQL Intro

Structured Query Language

- allows access to a database
- ANSI standard computer language
- creates DB objects
- executes queries against a database
- retrieves data from a database
- inserts new records in a database
- deletes records from a database
- updates records in a database

ANSI

American National Standards Institute

- Cross platform compliance (Access, Oracle, PostgreSQL, MS SQL Server…)
- Different versions / Same base keyword functions
- Own proprietary extensions

Anatomy of a Database

- Table
- Column
- Record
- All in one Tablespace

SQL Statements & Data Retrieval

SQL Statements

allows tables to be created or deleted. Indexes (keys) can be defined, specify links between tables, and impose constraints between database tables.

Common data definition language (DDL) statements for SQL:

- **CREATE TABLE** - creates a new database table
- **ALTER TABLE** - alters (changes) a database table
- **DROP TABLE** - deletes a database table
- **CREATE INDEX** - creates an index (search key)
- **DROP INDEX** - deletes an index

Data & SQL

SQL statement allows addition of data in rows as determined by column…the INSERT INTO command.

Formula Statement:

```
INSERT INTO table_name (column1, column2,...)
VALUES (value1, value2,....)
```

Common DML for SQL

Allows for retrieval, update, insertion, and deletion of records.

Common Data Manipulation Language (DML) statements for SQL:

SELECT- extracts data from a database table

UPDATE - updates data in a database table

DELETE - deletes data from a database table

Simple SQL Query Statement

Table name = BOOK

title	author_ln	author_fn	publisher	date
A Land Remembered	Smith	Patrick	Pineapple Press	1984
Kalahari Typing School for Men	Smith	Alexander McCall	Pantheon	2002
The Great Serum Race	Miller	Debbie S.	Bantam	2006

SELECT TITLE FROM BOOK;

title
A Land Remembered
Kalahari Typing School for Men
The Great Serum Race

Simple SQL Query Statement

Table name = BOOK

title	author_ln	author_fn	publisher	date
A Land Remembered	Smith	Patrick	Pineapple Press	1984
Kalahari Typing School for Men	Smith	Alexander McCall	Pantheon	2002
The Great Serum Race	Miller	Debbie S.	Bantam	2006

DELETE AUTHOR_FN FROM BOOK ;

title	author_ln	publisher	date
A Land Remembered	Smith	Pineapple Press	1984
Kalahari Typing School for Men	Smith	Pantheon	2002
The Great Serum Race	Miller	Bantam	2006

SQL Notes

1. Not case sensitive
2. Separate multiple column requests with a *comma* (select Title, Date from Books)
3. Some versions use a *semicolon* after the statement line
4. Use a *semicolon* to separate multiple commands issued to the DB in one session.
5. Use * for selecting all columns in a table

Select Distinct SQL Statement

Returns only distinct values by adding a DISTINCT to the SELECT statement.

Syntax example

SELECT DISTINCT column_name(s) FROM table_name

Distinct

Table name = BOOK

Title	Author LN	Author FN	Branch	Date
A Land Remembered	Smith	Patrick	Main	1984
Kalahari Typing School for Men	Smith	Alexander McCall	Airport	2002
The Great Serum Race	Miller	Debbie S.	Mall	2006
A Land Remembered	Smith	Patrick	Airport	1984
The Great Serum Race	Miller	Debbie S.	Main	2006

SELECT DISTINCT Title FROM Books;

Title
A Land Remembered
Kalahari Typing School for Men
The Great Serum Race

Conditional limiters: “Where” clause

WHERE clause, the following operators can be used:

Operator	Description
=	Equal
<> or !=	Not equal
>	Greater than
<	Less than
>=	Greater than or equal
<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern
IN	If you know the exact value you want to return for at least one of the columns

Note: In some versions of SQL the <> operator may be written as !=

Exercises

Exercise #1

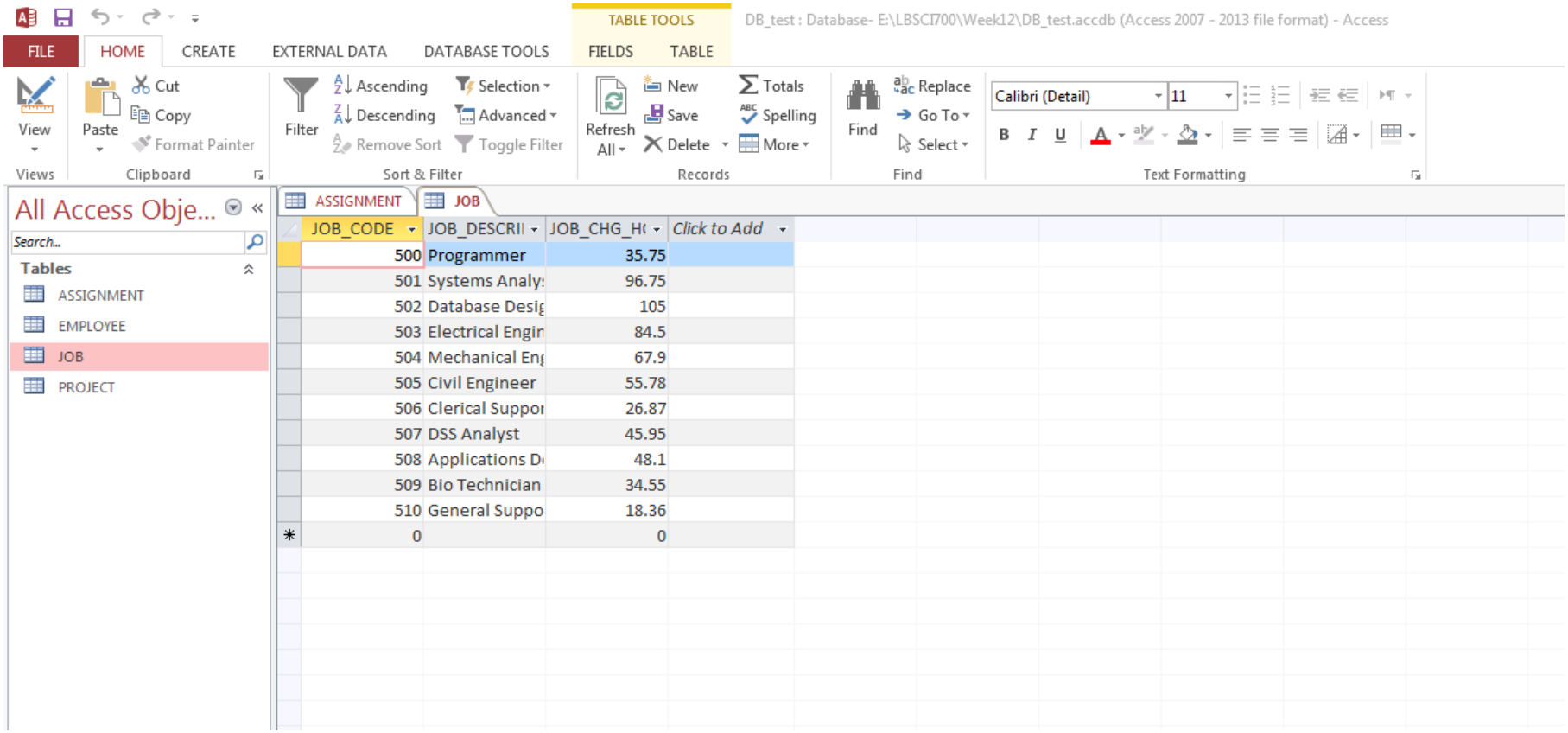
Provide the resulting output for the following SQL statement for the Books table shown below.

SELECT * FROM Books WHERE Author LN='Smith'

Title	Author LN	Author FN	Branch	Date
A Land Remembered	Smith	Patrick	Main	1984
Kalahari Typing School for Men	Smith	Alexander McCall	Airport	2002
The Great Serum Race	Miller	Debbie S.	Mall	2006
A Land Remembered	Smith	Patrick	Airport	1984
The Great Serum Race	Miller	Debbie S.	Main	2006

MS Access

What it looks like in Open Office / Access



Exercise #2

Write two simple SQL queries

1. Write a simple create table SQL statement
2. Write a simple insert record SQL statement
3. Find SQL statement in Access
4. Explain all the above three to person next to you.
5. Demonstrate one statement through Access or selected DB.

Digital Libraries and Dynamic Websites

Generalized Digital Library Development

- What do you need to do?

  --1. Get stuff (collection building & maintenance)

  --2. Get stuff into your digital library (contribute)

  --3. Get stuff out of your digital library (storage and retrieval)

Generalized Digital Library Development

- What stuff do you need?

  --Hardware?

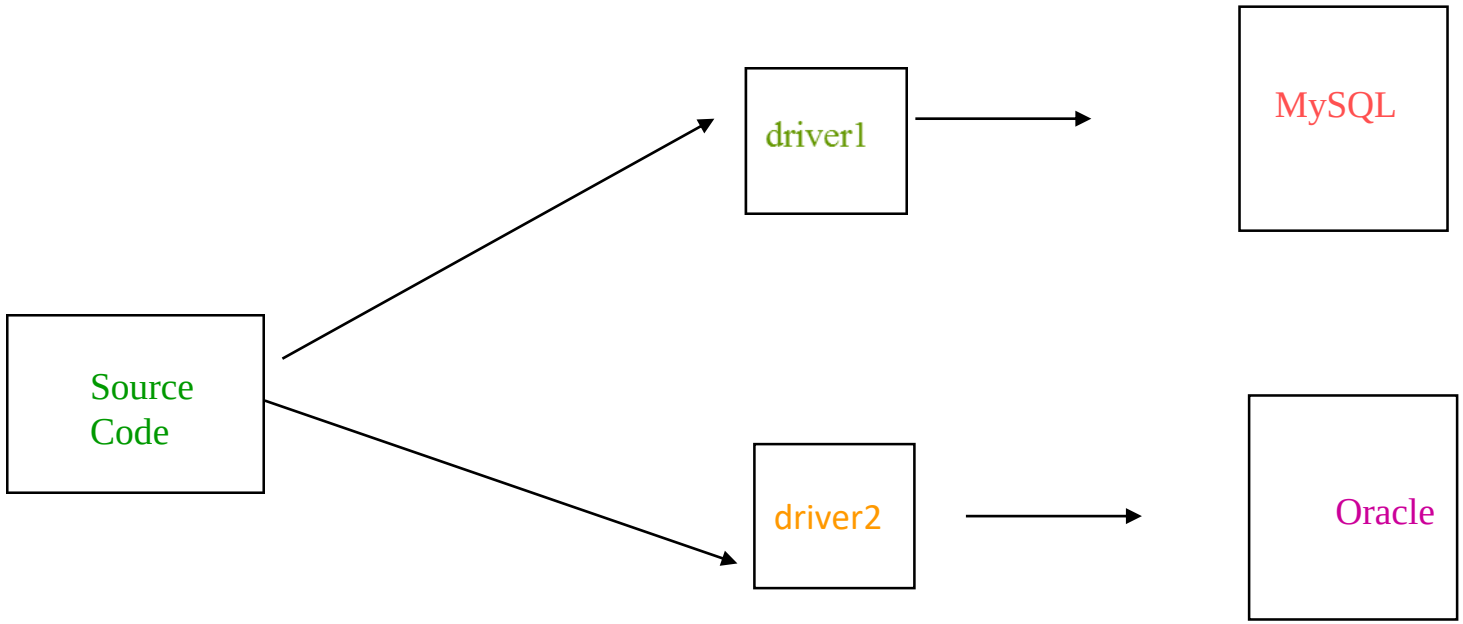
  --Software?

Interactivity of Systems

Database Connectivity: Database Drivers

- Provides compatibility for communication
- Hides DB system specific features from clients
- Operational View of Client-DB Connection

Operational View of Client-DB Connection



Server-side Programs

- Java's Servlets

  --create applications that take advantage of the server machine's resources

- PHP

  --with PHP, we can connect to and manipulate databases

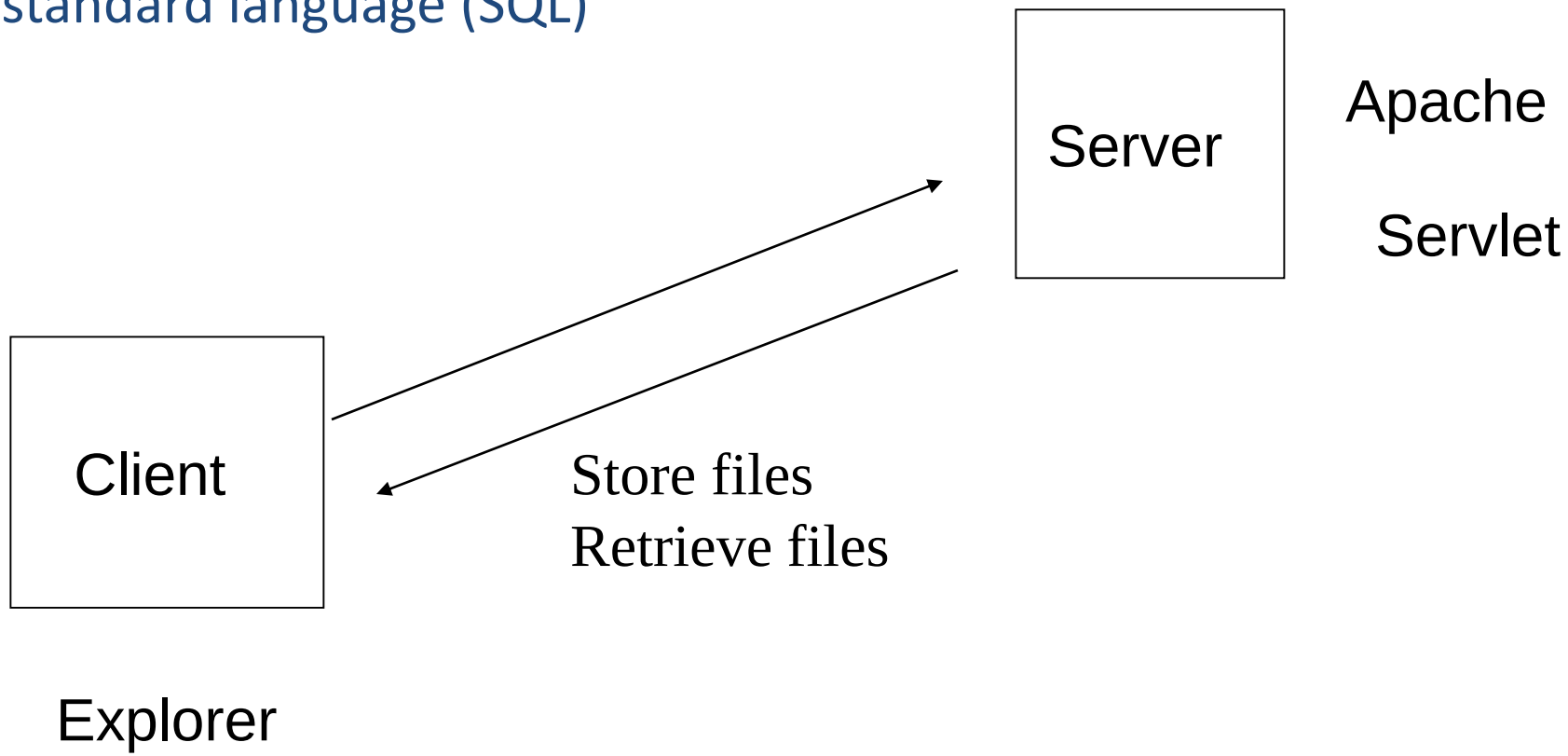
  https://www.w3schools.com/php/php_mysql_intro.asp

  https://www.w3schools.com/php/php_mysql_connect.asp

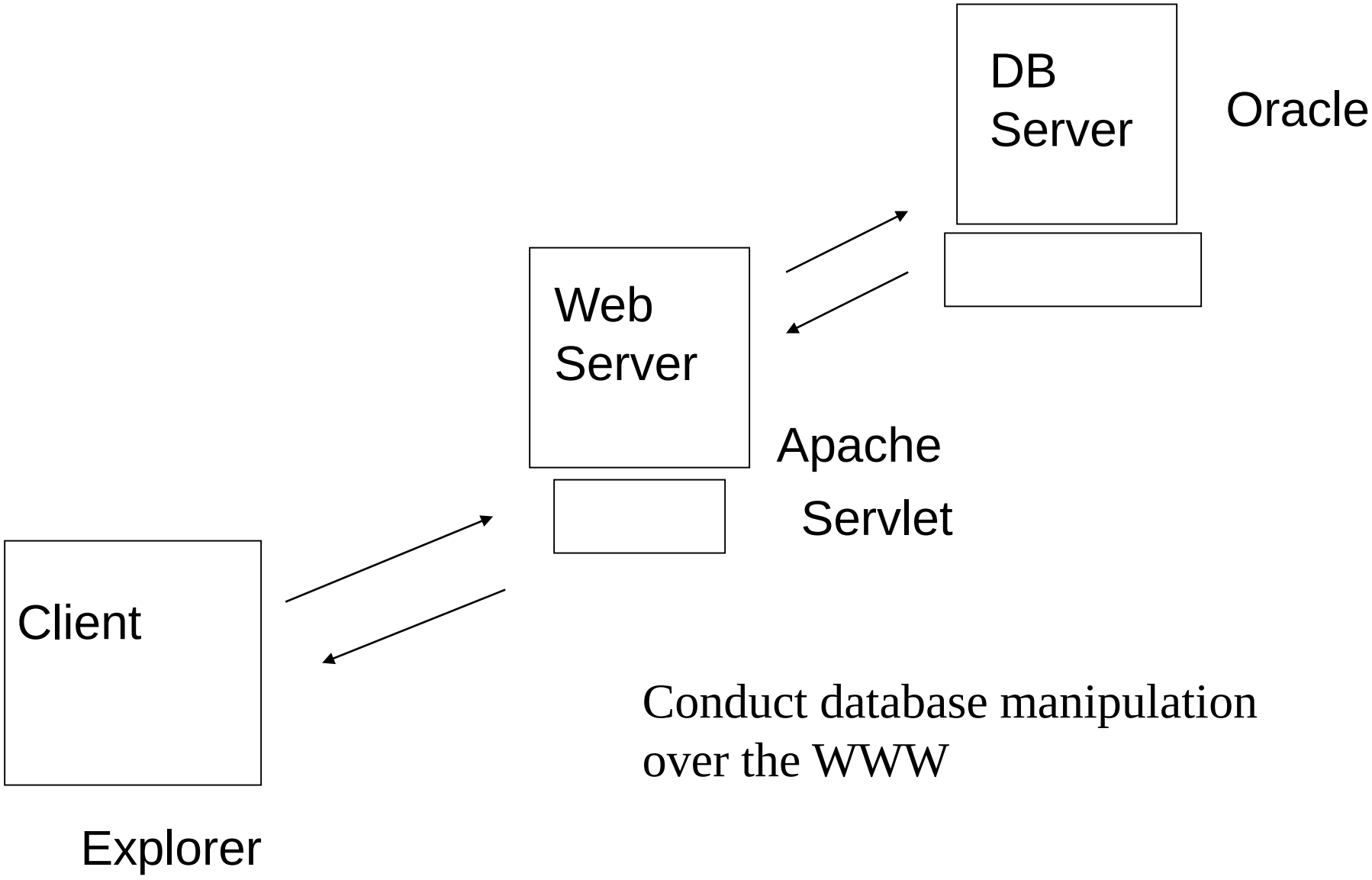
… …

Server-Side Programs

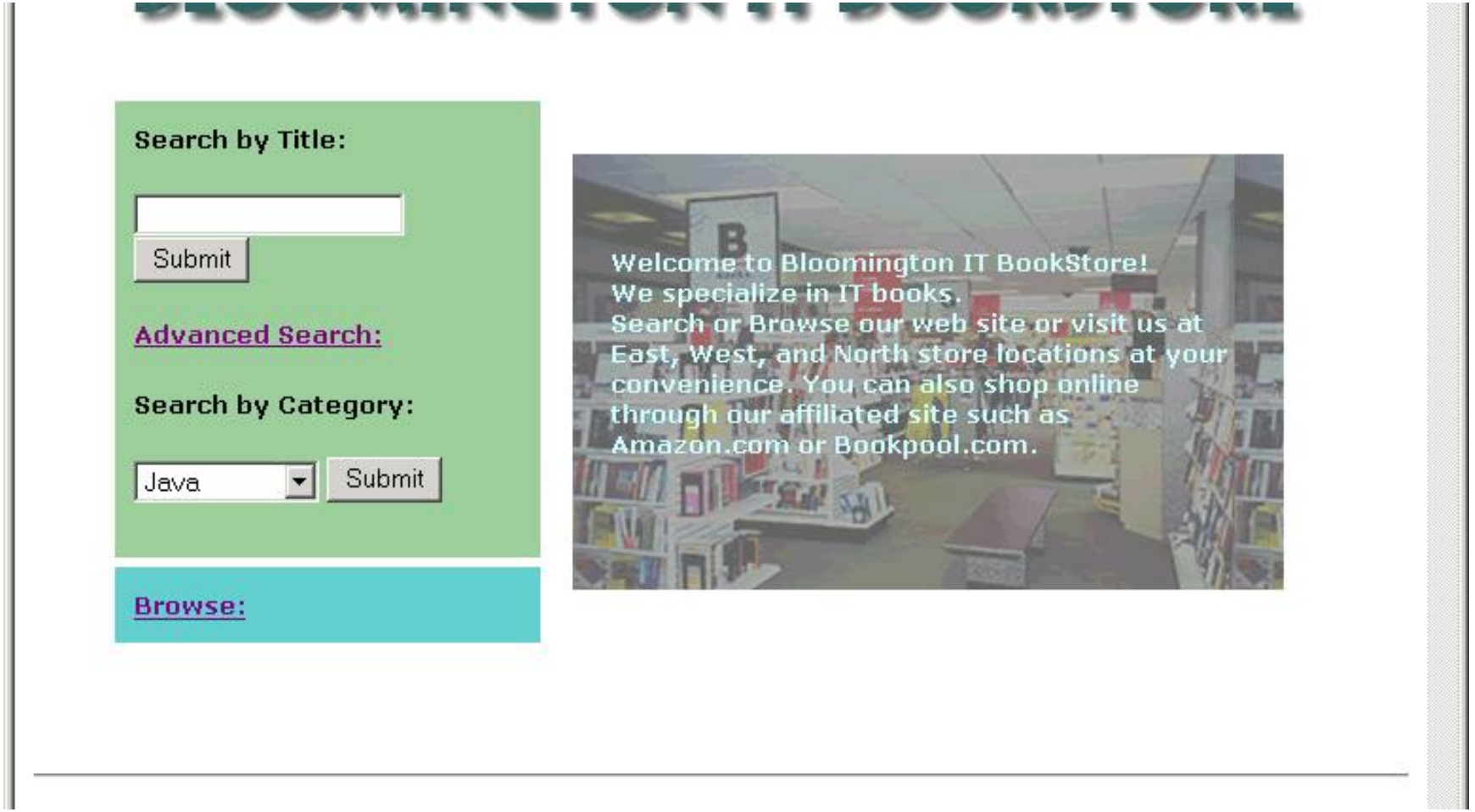
Provides clients with the independence to query using a standard language (SQL)



Server-Side Programs



HTML Front-End



HTML Front-End Code Fragment

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html><head><title>BookStore</title><meta http-equiv="Content-Type" content="text/html;charset=iso-8859-1"></head>
<body>
<p align="center"><img src="title.gif" width="645" height="53"> </p>
<center>

 <table width="650" border="0" cellspacing="5" cellpadding="10">
  <tr>
   <td width="200" bgcolor="#99CC99">
    <p><font size="2" face="Verdana, Arial, Helvetica, sans-serif"><strong>Search by Title:</strong></font>
      <form method="post" action="http://xxxxxx/servlet/TitleServlet">
       <input name="title" value="">
       <input name="submit" type="submit" value="Submit">
     </form>
    <form name="form1" method="post" action="">
     <strong><font size="2" face="Verdana, Arial, Helvetica, sans-serif">
       <a href="search.html">Advanced Search:</a> </font></strong>
    </form>
     <p> <font size="2" face="Verdana, Arial, Helvetica, sans-serif"><strong>Search
      by Category:</strong></font>
     <form method="post" action="http://xxxxxx/servlet/CategoryServlet">
     <select name="category">
      <option value="java" SELECTED>Java</option>
      <option value="perl">Perl</option>
      <option value="javascript">JavaScript</option>
      <option value="web_server">Web Server</option>
      <option value="html">HTML</option>
     </select>
     <input type="submit" name="Submit" value="Submit">
    </form>
```

Servlet Code Fragment: TitleServlet

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
import java.sql.*;

public class TitleServlet extends HttpServlet {

/** * Initialize global variables */
public void init(ServletConfig config) throws ServletException { super.init(config); }

/** * Process the HTTP Post request */
/**
  * Process the HTTP Post request
  */
 public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
   String title = request.getParameter("title");
   response.setContentType("text/html");
   PrintWriter out = new PrintWriter (response.getOutputStream());
   out.println("<html>");
   out.println("<head><title>Title Search</title></head>");
   out.println("<body><table border=1 cellpadding = 5 cellspacing = 0>");
   try {
     DriverManager.registerDriver(new oracle.jdbc.driver.OracleDriver());
     Connection conn = DriverManager.getConnection ("jdbc:oracle:thin:@xxxxxx:1521:oed1","username","password");
     Statement stmt = conn.createStatement();
     ResultSet rs = stmt.executeQuery("SELECT pict, year, title, author FROM book where title like '%" + title + "%'");
     ResultSetMetaData rsmd = rs.getMetaData();
    int nCols = rsmd.getColumnCount();
   int i;
   for (i=0; i<nCols; i++)
    out.println("<th>"+rsmd.getColumnName(i+1)+"</th>");
   out.println("</tr>");
   while (rs.next()) …
```

Recap

SQL: Intro

SQL Query and Data Retrieval

Exercises

Digital Libraries and Dynamic Websites

Interactivity of Systems

Last Things

Before Final Week (May 20)

- Group Presentation - May 13
  --Group Presentation Slides - May 12

- Personal Site - May 5, May 13
- Usability Study - May 5, May 13

- Database (group) – May 6

*Class on May 5 - Lab

*Final Review – May 13 ~ May 20

ToDo

- Database (group)
  - -- MS Access
  - -- MySQL Workbench
    https://www.mysql.com/products/workbench/
  - -- phpMyAdmin (freemysqlhosting.net)

- Keep working on your site and usability study
- Look for emails

* Next Monday, May 5, Asynchronous Online