

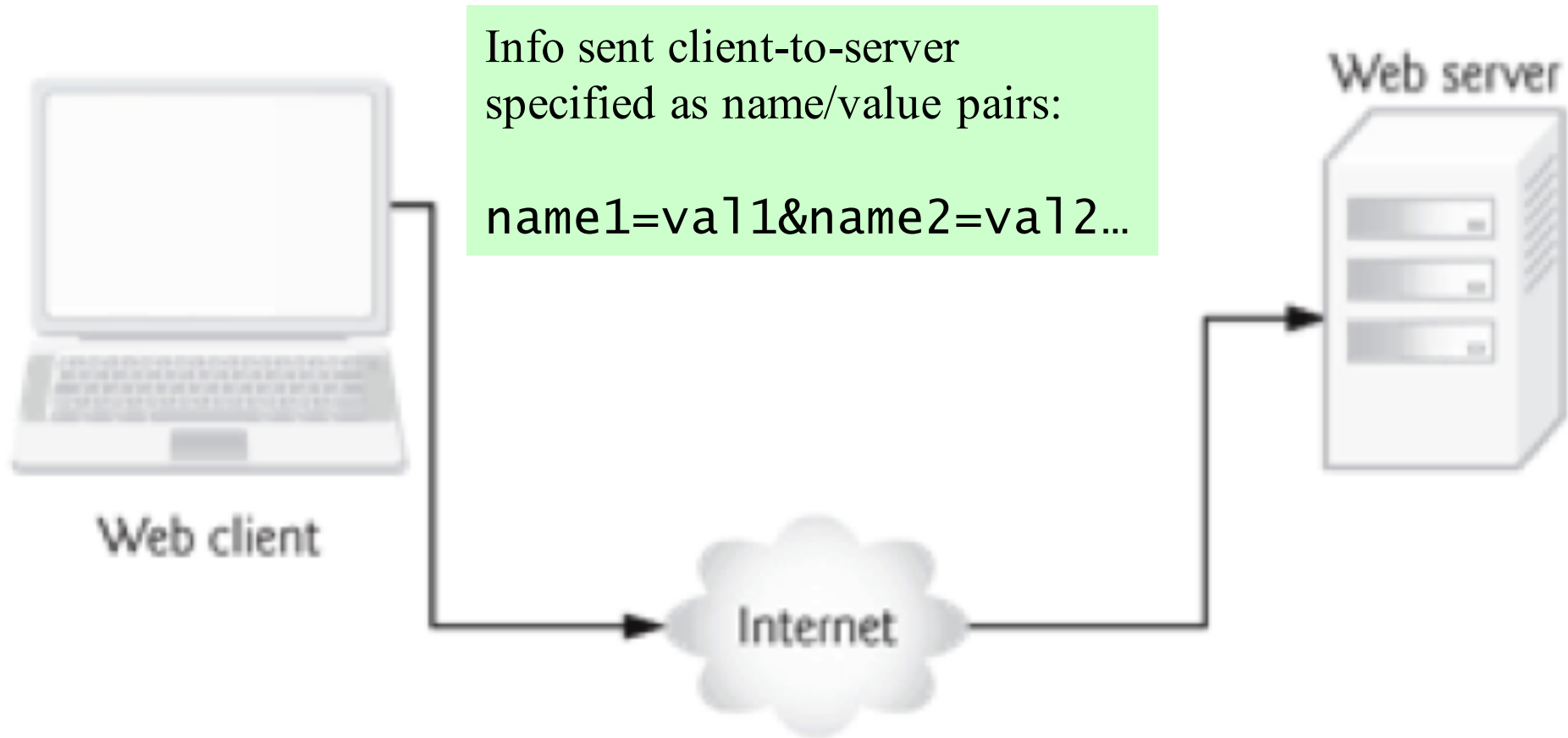
COMP 484

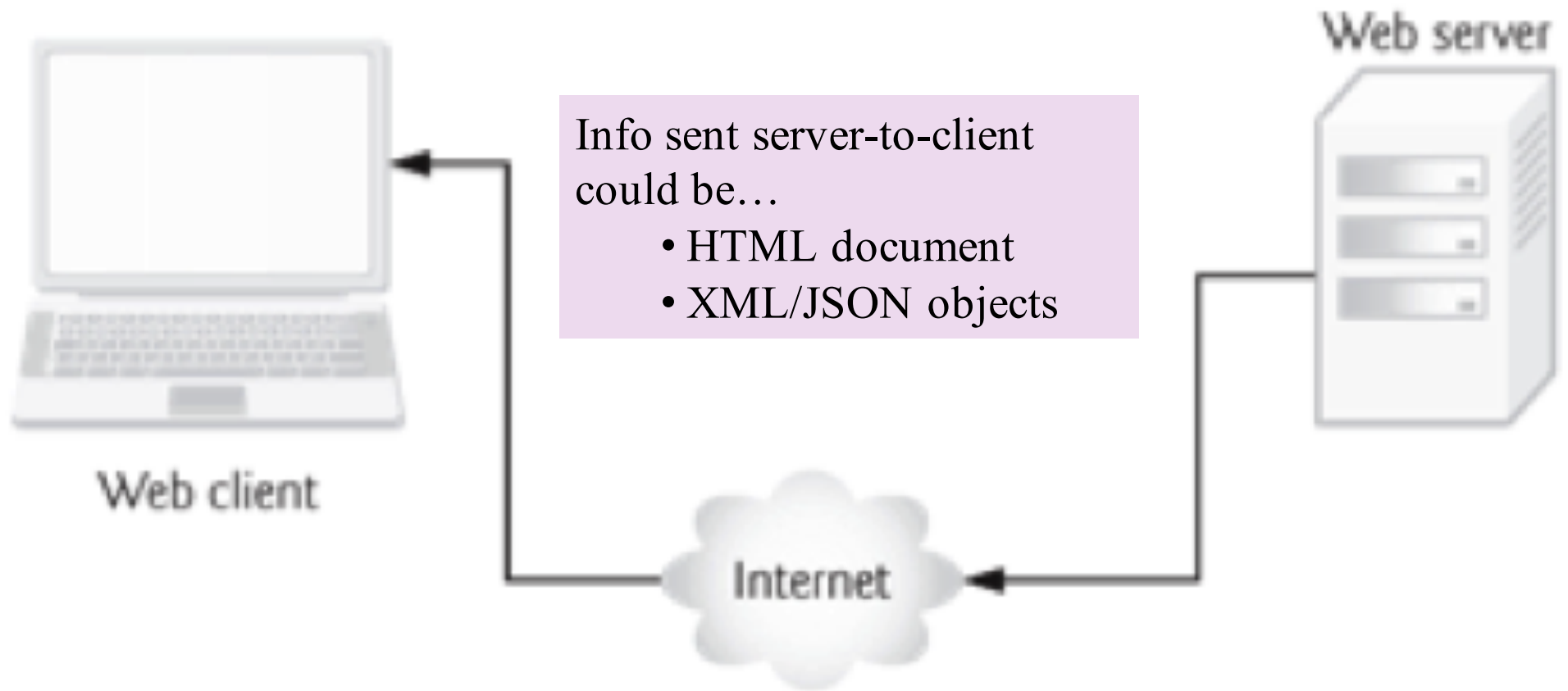
Web Engineering I

Instructor: ~kaplan (akaplan@csun.edu)

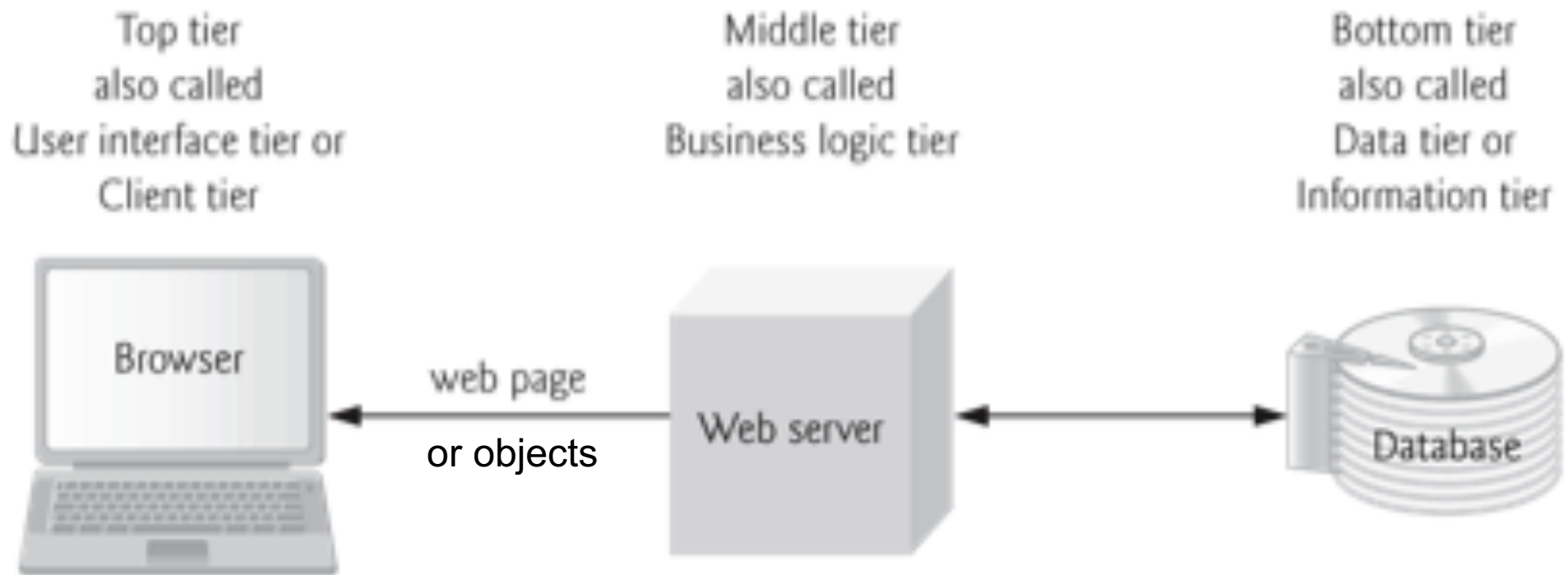
Lecture #3: 7/20/2020

Relational Databases & SQL



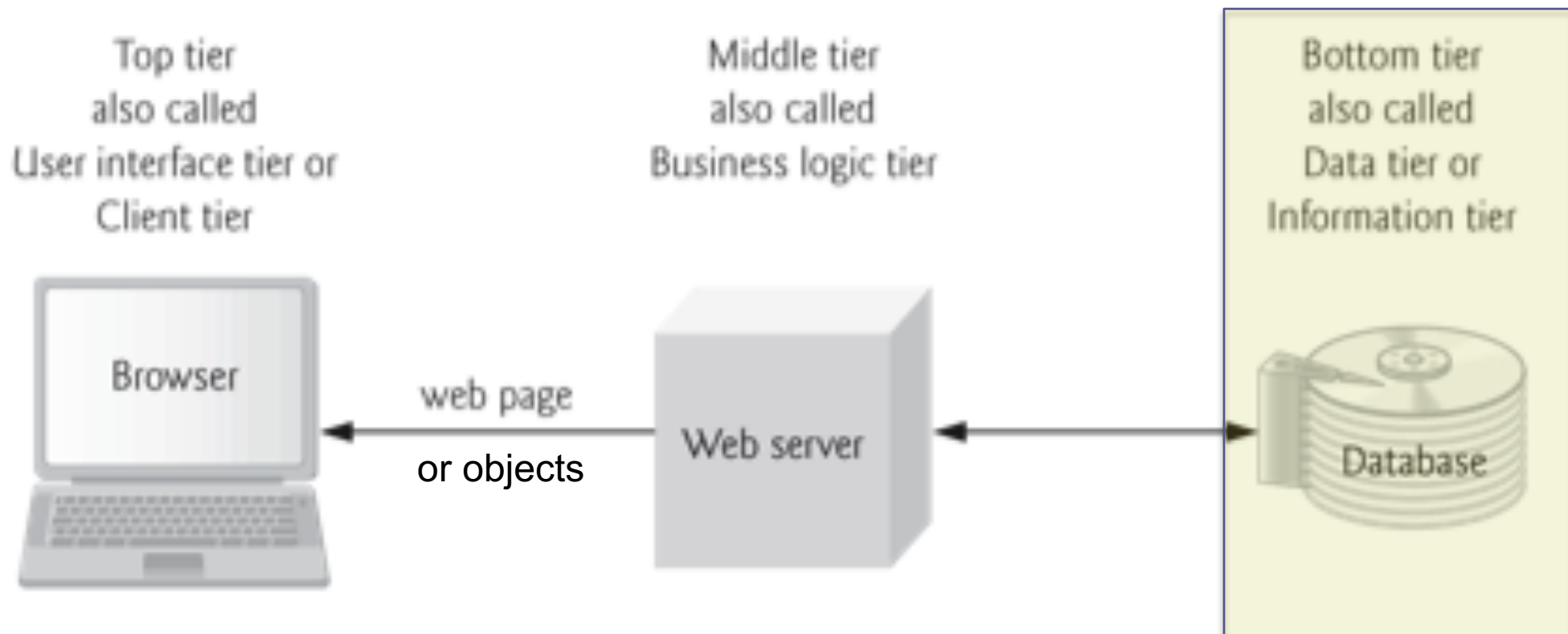


Multitier Application Architecture



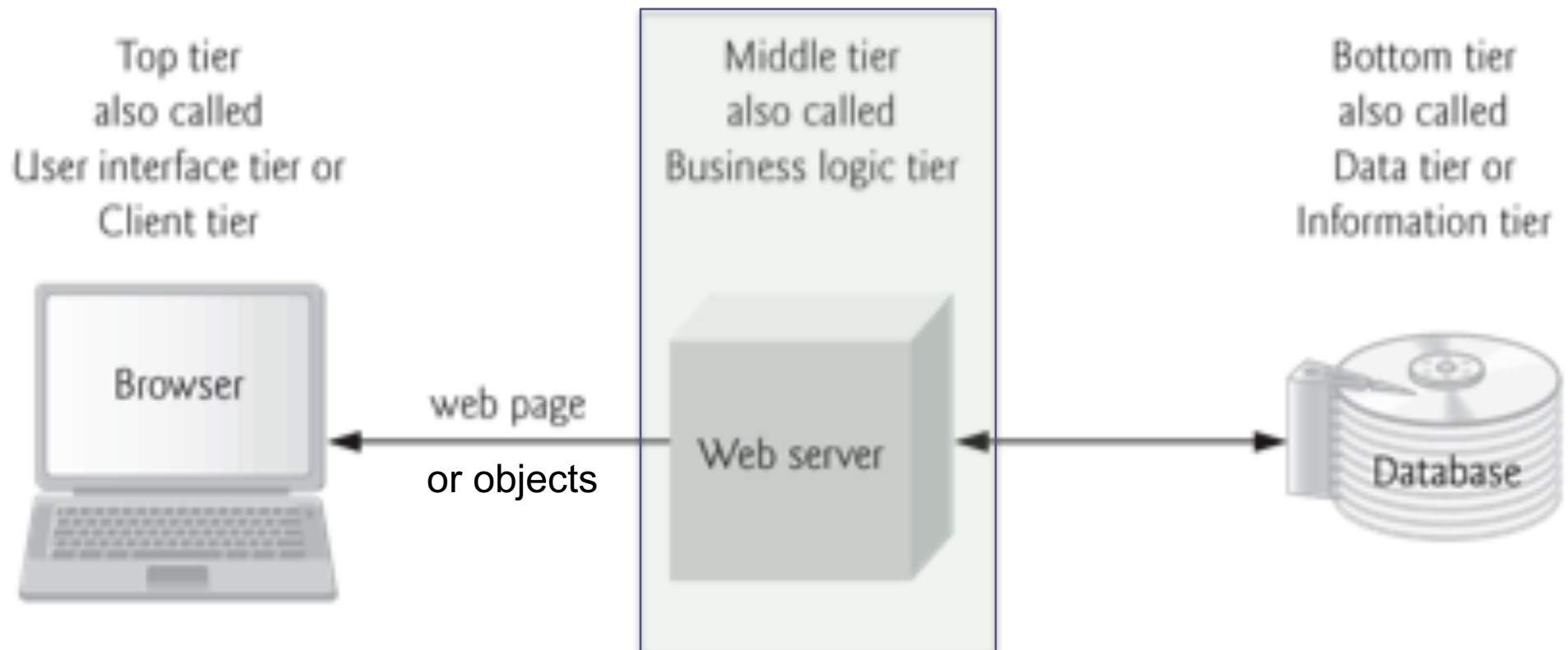
- Different functionality in separate tiers
- Tiers of web-based app typically on separate computers
 - Can be on same computer

Multitier Application Architecture



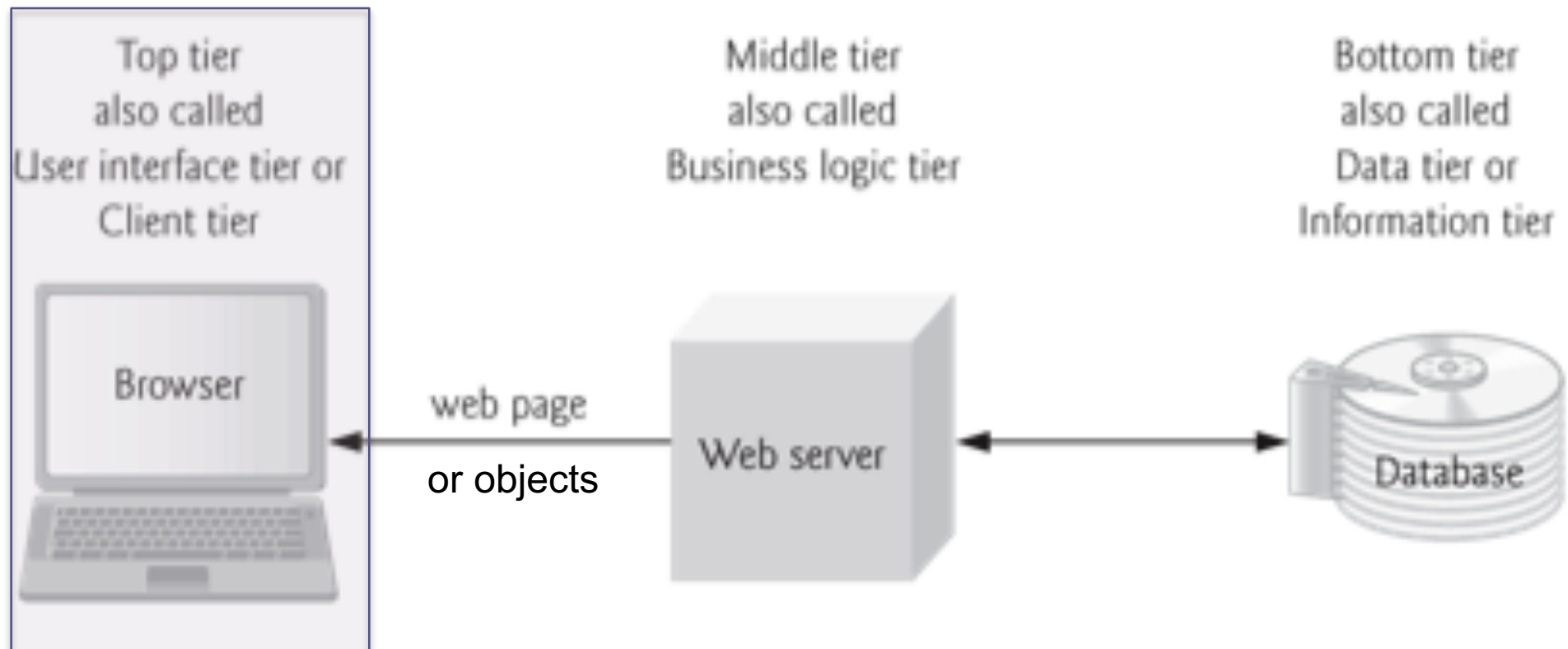
- Bottom Tier (Data/Information Tier)
 - Maintains application's data
 - E.g. MySQL relational database management system
 - Others: MS SQL Server, Oracle DBMS

Multitier Application Architecture



- Middle Tier (Business Logic Tier)
 - Enforces business rules
 - Updates/queries database based on user inputs
 - Presents data to users

Multitier Application Architecture

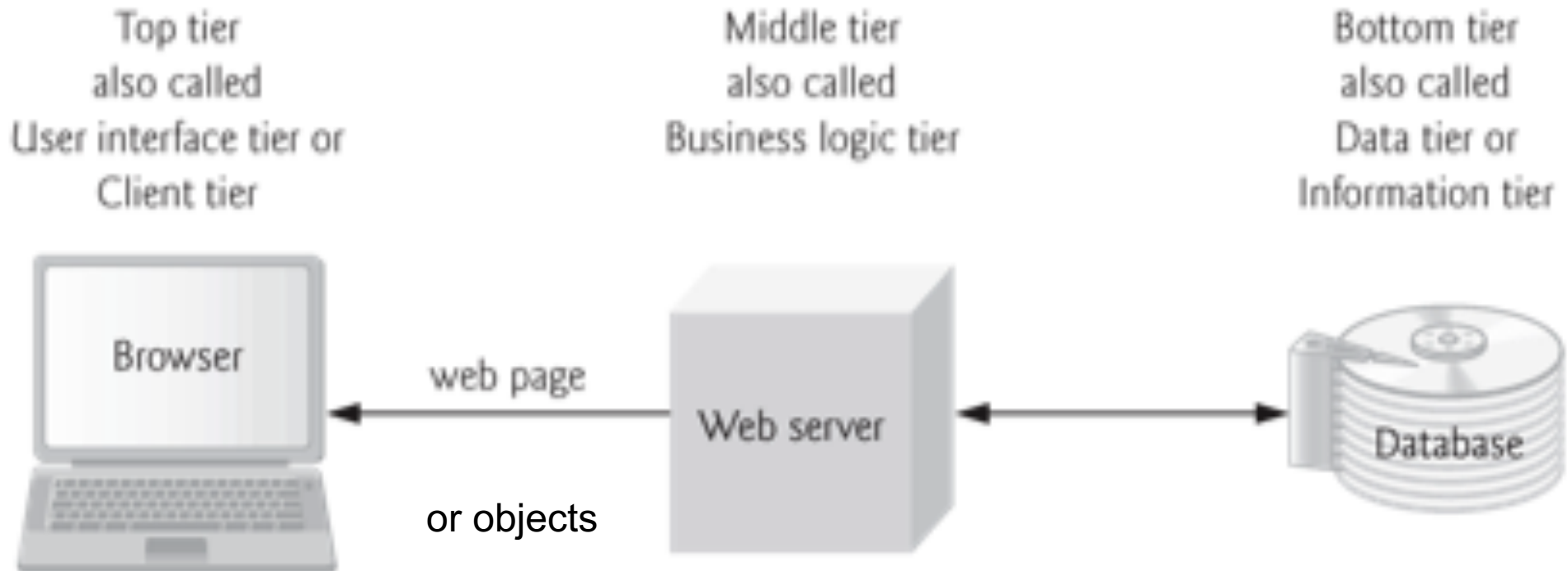


- **Top Tier (Client/UI Tier)**
 - Interacts with middle tier to make requests and retrieve data
 - Displays retrieved data to end user
 - Never interacts directly with bottom tier

Databases

- A database is an organized collection of data
- A *database management system (DBMS)* provides mechanisms for storing, organizing, retrieving and modifying data for many users
- Today's most popular database management systems are *relational databases*
- SQL is the international standard language used almost universally with relational database systems to perform queries and manipulate data
- Programs connect to, and interact with, relational databases systems via an interface

Why Host Web Server & Database on Separate Machines?



- **Security**
 - Database should only be touched by web server (never accessed directly by internet clients), can be hosted on private subnet
- **Scalability**
 - Web servers (stateless) – can scale easily behind load balancer
 - Databases – hard to scale horizontally (consistency)
- **Performance**
 - Web server and database need not compete for resources on same machine

Relational Databases

- A relational database stores data in tables
 - Tables are composed of rows, and rows are composed of columns in which values are stored
 - Each column of a table represents a different attribute in a row of data
 - A primary key is a set of one or more columns which provide a value unique to each row
 - Cannot be replicated in other rows

| | Number | Name | Department | Salary | Location |
|-------|-------------|----------|------------|--------|-------------|
| | 23603 | Jones | 413 | 1100 | New Jersey |
| | 24568 | Kerwin | 413 | 2000 | New Jersey |
| Row { | 34589 | Larson | 642 | 1800 | Los Angeles |
| | 35761 | Myers | 611 | 1400 | Orlando |
| | 47132 | Neumann | 413 | 9000 | New Jersey |
| | 78321 | Stephens | 611 | 8500 | Orlando |
| | Primary key | | Column | | |

SQL - 4 types of operations

- **SELECT**: read data from table(s)
- **INSERT**: create new data row
- **UPDATE**: change existing data row(s)
- **DELETE**: remove existing data row(s)

SQL: A Basic SELECT Query

- The basic form of a SELECT query is
`SELECT * FROM tableName`
in which the asterisk (*) *wildcard character* indicates that all columns from the *tableName* should be retrieved
- To retrieve specific columns, replace the * with a comma-separated list of column names

```
SELECT * FROM Authors
```

| AuthorID | FirstName | LastName |
|----------|-----------|----------|
| 1 | Paul | Deitel |
| 2 | Harvey | Deitel |
| 3 | Abbey | Deitel |
| 4 | Michael | Morgano |
| 5 | Eric | Kern |

```
SELECT AuthorID, LastName FROM Authors
```

| AuthorID | LastName |
|----------|----------|
| 1 | Deitel |
| 2 | Deitel |
| 3 | Deitel |
| 4 | Morgano |
| 5 | Kern |

Table: Authors

| AuthorID | FirstName | LastName |
|----------|-----------|----------|
| 1 | Paul | Deitel |
| 2 | Harvey | Deitel |
| 3 | Abbey | Deitel |
| 4 | Michael | Morgano |
| 5 | Eric | Kern |

Table: AuthorISBN

| AuthorID | ISBN | AuthorID | ISBN |
|----------|------------|----------|------------|
| 1 | 0132152134 | 2 | 0132575663 |
| 2 | 0132152134 | 1 | 0132662134 |
| 1 | 0132151421 | 2 | 0132662134 |
| 2 | 0132151421 | 1 | 0132404134 |
| 1 | 0132575663 | 2 | 0132404134 |
| 1 | 013705842X | 1 | 0132121134 |
| 2 | 013705842X | 2 | 0132121134 |
| 3 | 013705842X | 3 | 0132121134 |
| 4 | 013705842X | 4 | 0132121134 |
| 5 | 013705842X | | |

Relational Database Overview: A *books* database

- The combined columns of the AuthorISBN table represent the table's *primary key*—
 - thus, each row in this table must be a *unique* combination of an AuthorID and an ISBN
- A foreign key is a column in a table that matches the primary-key column in another table
 - The foreign key helps maintain the **Rule of Referential Integrity**: Every foreign-key value must appear as another table's primary-key value

Relational Database Overview: A *books* database (cont)

Table: Titles

| ISBN | Title | EditionNumber | Copyright |
|------------|---|---------------|-----------|
| 0132152134 | Visual Basic 2010 How to Program | 5 | 2011 |
| 0132151421 | Visual C# 2010 How to Program | 4 | 2011 |
| 0132575663 | Java How to Program | 9 | 2012 |
| 0132662361 | C++ How to Program | 8 | 2012 |
| 0132404168 | C How to Program | 6 | 2010 |
| 013705842X | iPhone for Programmers: An App-Driven Approach | 1 | 2010 |
| 0132121360 | Android for Programmers: An App-Driven Approach | 1 | 2012 |

SQL : WHERE Clause

- SQL uses the optional **WHERE** clause in a query to specify the selection criteria for the query. The basic form of a query with selection criteria is

```
SELECT columnName1, columnName2, ... FROM  
      tableName WHERE criteria
```


| ISBN | Title | EditionNumber | Copyright |
|------------|---|---------------|-----------|
| 0132152134 | Visual Basic 2010 How to Program | 5 | 2011 |
| 0132151421 | Visual C# 2010 How to Program | 4 | 2011 |
| 0132575663 | Java How to Program | 9 | 2012 |
| 0132662361 | C++ How to Program | 8 | 2012 |
| 0132404168 | C How to Program | 6 | 2010 |
| 013705842X | iPhone for Programmers: An App-Driven Approach | 1 | 2010 |
| 0132121360 | Android for Programmers: An App-Driven Approach | 1 | 2012 |

SELECT Title, EditionNumber, Copyright
FROM Titles WHERE Copyright > 2010

| Title | EditionNumber | Copyright |
|---|---------------|-----------|
| Visual Basic 2010 How to Program | 5 | 2011 |
| Visual C# 2010 How to Program | 4 | 2011 |
| Java How to Program | 9 | 2012 |
| C++ How to Program | 8 | 2012 |
| Android for Programmers: An App-Driven Approach | 1 | 2012 |

SQL : WHERE Clause (cont.)

Pattern Matching: Zero or More Characters

- The WHERE clause can contain operators <, >, <=, >=, =, <> and LIKE.
- Operator LIKE is used for string pattern matching with wildcard characters percent (%) and underscore (_)
 - A percent character (%) in a pattern indicates that a string matching the pattern can have zero or more characters at the percent character's location in the pattern
 - An underscore (_) in the pattern string indicates a single character at that position in the pattern

Table: Authors

| AuthorID | FirstName | LastName |
|----------|-----------|----------|
| 1 | Paul | Deitel |
| 2 | Harvey | Deitel |
| 3 | Abbey | Deitel |
| 4 | Michael | Morgano |
| 5 | Eric | Kern |

```

SELECT AuthorID, FirstName, LastName
FROM Authors
WHERE LastName LIKE 'D%'

```

| AuthorID | FirstName | LastName |
|----------|-----------|----------|
| 1 | Paul | Deitel |
| 2 | Harvey | Deitel |
| 3 | Abbey | Deitel |

Table: Authors

| AuthorID | FirstName | LastName |
|----------|-----------|----------|
| 1 | Paul | Deitel |
| 2 | Harvey | Deitel |
| 3 | Abbey | Deitel |
| 4 | Michael | Morgano |
| 5 | Eric | Kern |

```
SELECT AuthorID, FirstName, LastName  
FROM Authors  
WHERE LastName LIKE '_o%'
```

| AuthorID | FirstName | LastName |
|----------|-----------|----------|
| 4 | Michael | Morgano |

SQL: ORDER BY Clause

- The result of a query can be sorted in ascending or descending order using the optional ORDER BY clause. The simplest form of an ORDER BY clause is

```
SELECT columnName1, columnName2, ... FROM  
       tableName ORDER BY column ASC
```

```
SELECT columnName1, columnName2, ... FROM  
       tableName ORDER BY column DESC
```

where ASC specifies ascending order, DESC specifies descending order and column specifies the column on which the sort is based

- (The default sorting order is ascending, so ASC is optional)
- Multiple columns can be used for ordering purposes with an ORDER BY clause of the form

```
ORDER BY column1 sortingOrder, column2 sortingOrder, ...
```
- The WHERE and ORDER BY clauses can be combined in one query. If used, ORDER BY must be the last clause in the query

```
SELECT AuthorId, FirstName, LastName  
FROM Authors ORDER BY LastName ASC
```

| AuthorID | FirstName | LastName |
|----------|-----------|----------|
| 1 | Paul | Deitel |
| 2 | Harvey | Deitel |
| 3 | Abbey | Deitel |
| 5 | Eric | Kern |
| 4 | Michael | Morgano |

```
SELECT AuthorId, FirstName, LastName  
FROM Authors ORDER BY LastName DESC
```

| AuthorID | FirstName | LastName |
|----------|-----------|----------|
| 4 | Michael | Morgano |
| 5 | Eric | Kern |
| 1 | Paul | Deitel |
| 2 | Harvey | Deitel |
| 3 | Abbey | Deitel |

```
SELECT AuthorId, FirstName, LastName  
FROM Authors ORDER BY LastName, FirstName
```

| AuthorID | FirstName | LastName |
|----------|-----------|----------|
| 3 | Abbey | Deitel |
| 2 | Harvey | Deitel |
| 1 | Paul | Deitel |
| 5 | Eric | Kern |
| 4 | Michael | Morgano |

Table: Titles

24

| ISBN | Title | EditionNumber | Copyright |
|------------|---|---------------|-----------|
| 0132152134 | Visual Basic 2010 How to Program | 5 | 2011 |
| 0132151421 | Visual C# 2010 How to Program | 4 | 2011 |
| 0132575663 | Java How to Program | 9 | 2012 |
| 0132662361 | C++ How to Program | 8 | 2012 |
| 0132404168 | C How to Program | 6 | 2010 |
| 013705842X | iPhone for Programmers: An App-Driven Approach | 1 | 2010 |
| 0132121360 | Android for Programmers: An App-Driven Approach | 1 | 2012 |

SELECT * FROM Titles WHERE Title LIKE
'%How to Program' ORDER BY Title ASC

| ISBN | Title | EditionNumber | Copyright |
|------------|----------------------------------|---------------|-----------|
| 0132404168 | C How to Program | 6 | 2010 |
| 0132662361 | C++ How to Program | 8 | 2012 |
| 0132575663 | Java How to Program | 9 | 2012 |
| 0132152134 | Visual Basic 2010 How to Program | 5 | 2011 |
| 0132151421 | Visual C# 2010 How to Program | 4 | 2011 |

Minor textbook
error corrected
in dark-red

Merging Data from Multiple Tables:

INNER JOIN

- An **INNER JOIN** operator merges rows from two tables by matching values in columns that are common to the tables. The basic form for the **INNER JOIN** operator is:

```
SELECT columnName1, columnName2, ...  
FROM table1  
INNER JOIN table2  
    ON table1.columnName = table2.columnName
```

- The **ON clause** of the **INNER JOIN** specifies the columns from each table that are compared to determine which rows are merged
- The following query produces a list of authors accompanied by the ISBNs for books written by each author:

```
SELECT FirstName, LastName, ISBN  
FROM Authors  
INNER JOIN AuthorISBN  
    ON Authors.AuthorID = AuthorISBN.AuthorID  
ORDER BY LastName, FirstName
```

```
SELECT FirstName, LastName, ISBN
FROM Authors
INNER JOIN AuthorISBN
    ON Authors.AuthorID = AuthorISBN.AuthorID
ORDER BY LastName, FirstName
```

can also be written in implicit join notation, as follows...

```
SELECT FirstName, LastName, ISBN
FROM Authors, AuthorISBN
WHERE Authors.AuthorID = AuthorISBN.AuthorID
ORDER BY LastName, FirstName
```

```

SELECT FirstName, LastName, ISBN
FROM Authors
INNER JOIN AuthorISBN
ON Authors.AuthorID = AuthorISBN.AuthorID
ORDER BY LastName, FirstName

```

| FirstName | LastName | ISBN | FirstName | LastName | ISBN |
|-----------|----------|------------|-----------|----------|------------|
| Abbey | Deitel | 013705842X | Paul | Deitel | 0132151421 |
| Abbey | Deitel | 0132121360 | Paul | Deitel | 0132575663 |
| Harvey | Deitel | 0132152134 | Paul | Deitel | 0132662361 |
| Harvey | Deitel | 0132151421 | Paul | Deitel | 0132404168 |
| Harvey | Deitel | 0132575663 | Paul | Deitel | 013705842X |
| Harvey | Deitel | 0132662361 | Paul | Deitel | 0132121360 |
| Harvey | Deitel | 0132404168 | Eric | Kern | 013705842X |
| Harvey | Deitel | 013705842X | Michael | Morgano | 013705842X |
| Harvey | Deitel | 0132121360 | Michael | Morgano | 0132121360 |
| Paul | Deitel | 0132152134 | | | |

SQL - 4 types of operations

- **SELECT**: read data from table(s)
- **INSERT**: create new data row
- **UPDATE**: change existing data row(s)
- **DELETE**: remove existing data row(s)

SQL : INSERT Statement

- An INSERT statement inserts a new row into a table. The basic form of this statement is

```
INSERT INTO tableName ( columnName1,  
                           columnName2, ...,  
                           columnNameN )  
VALUES ( value1, value2, ..., valueN )
```

where *tableName* is the table in which to insert the row

- Followed by a comma-separated list of column names in parentheses
- Then, SQL keyword VALUES and a comma-separated list of values in parentheses
- **Each value is mapped to corresponding column in inserted row**

```
INSERT INTO Authors (
  FirstName, LastName )
VALUES ( 'Sue', 'Red' )
```

| AuthorID | FirstName | LastName |
|----------|-----------|----------|
| 1 | Paul | Deitel |
| 2 | Harvey | Deitel |
| 3 | Abbey | Deitel |
| 4 | Michael | Morgano |
| 5 | Eric | Kern |

- The statement indicates that values are provided for the FirstName and LastName columns.
 - The corresponding values are 'Sue' and 'Red'
- AuthorID not specified in INSERT because it is an autoincremented column

SQL - 4 types of operations

- SELECT: read data from table(s)
- INSERT: create new data row
- **UPDATE:** change existing data row(s)
- DELETE: remove existing data row(s)

SQL : UPDATE Statement

- An UPDATE statement modifies data in a table. Its basic form is

```
UPDATE tableName
    SET columnName1 = value1,
        columnName2 = value2, ...,
        columnNameN = valueN
    WHERE criteria
```

where *tableName* is the table in which to update data

The *tableName* is followed by keyword SET and a comma-separated list of column name/value pairs in the format *columnName* = *value*

The optional WHERE clause *criteria* determines which rows to update
(without it, all rows of the table are updated)

| AuthorID | FirstName | LastName |
|----------|-----------|----------|
| 1 | Paul | Deitel |
| 2 | Harvey | Deitel |
| 3 | Abbey | Deitel |
| 4 | Michael | Morgano |
| 5 | Eric | Kern |
| 6 | Sue | Red |

```
UPDATE Authors
SET LastName = 'Black'
WHERE FirstName = 'Sue' AND LastName = 'Red'
```

| AuthorID | FirstName | LastName |
|----------|-----------|----------|
| 1 | Paul | Deitel |
| 2 | Harvey | Deitel |
| 3 | Abbey | Deitel |
| 4 | Michael | Morgano |
| 5 | Eric | Kern |
| 6 | Sue | Black |

SQL - 4 types of operations

- SELECT: read data from table(s)
- INSERT: create new data row
- UPDATE: change existing data row(s)
- **DELETE**: remove existing data row(s)

DELETE Statement

- A SQL DELETE statement removes rows from a table. Its basic form is

DELETE FROM *tableName* WHERE
criteria

where *tableName* is the table from which to delete a row (or rows)

The optional WHERE *criteria* determines which rows to delete. If this clause is omitted, all the table's rows are deleted

| AuthorID | FirstName | LastName |
|----------|-----------|----------|
| 1 | Paul | Deitel |
| 2 | Harvey | Deitel |
| 3 | Abbey | Deitel |
| 4 | Michael | Morgano |
| 5 | Eric | Kern |
| 6 | Sue | Black |

DELETE FROM Authors
WHERE FirstName = 'Sue' AND LastName = 'Black'

| AuthorID | FirstName | LastName |
|----------|-----------|----------|
| 1 | Paul | Deitel |
| 2 | Harvey | Deitel |
| 3 | Abbey | Deitel |
| 4 | Michael | Morgano |
| 5 | Eric | Kern |

MySQL

- MySQL (pronounced “my sequel”) is a robust and scalable relational database management system (RDBMS) that was created by the Swedish consulting firm TcX in 1994
- MySQL, now owned by Oracle, is a multiuser, multithreaded RDBMS server that uses SQL to interact with and manipulate data
- Multithreading capabilities enable MySQL database to perform multiple tasks concurrently, allowing the server to process client requests efficiently
- Implementations of MySQL are available for Windows, Mac OS X, Linux and UNIX

