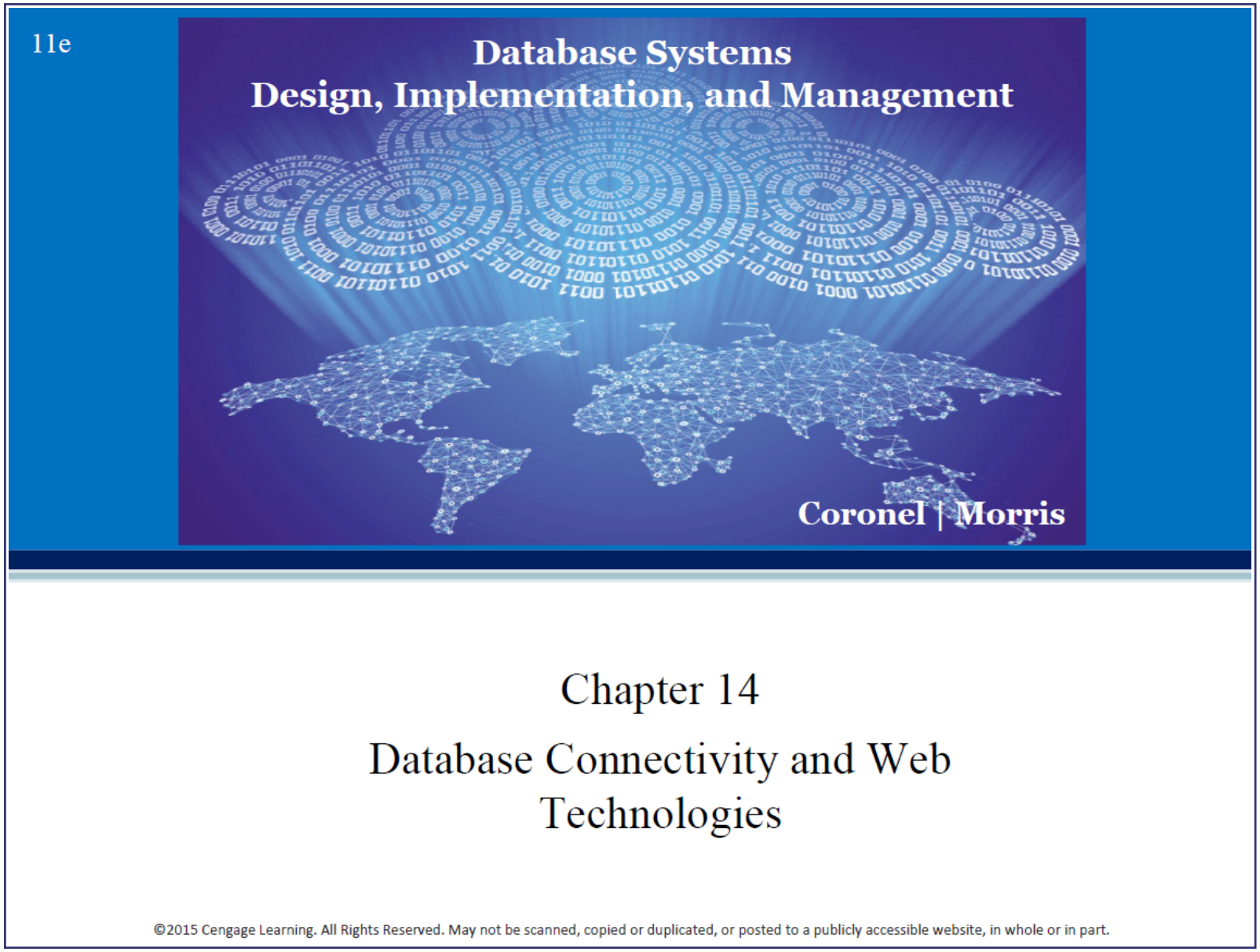
Database connectivity

Ch.14


11e
Database Systems
Design, Implementation, and Management
Coronel | Morris
Chapter 14
Database Connectivity and Web Technologies
©2015 Cengage Learning. All Rights Reserved. May not be scanned, copied or duplicated, or posted to a publicly accessible website, in whole or in part.

Learning Objectives

- In this chapter, students will learn:
  - About various database connectivity technologies
  - How Web-to-database middleware is used to integrate databases with the Internet
  - About Web browser plug-ins and extensions
  - What services are provided by Web application servers
  - What Extensible Markup Language (XML) is and why it is important for Web database development
  - About cloud computing and how it enables the database-as-a-service model

Database Connectivity

- **Database middleware**: Provides an interface between the application program and the database
- Data repository/source - Data management application (eg. Oracle RDBMS) that is used to store data generated by an application program

3

Various connectivity options

- Native SQL (provided by vendors)
- M'soft: ODBC, DAO+JET, RDO
- M'soft: OLE-DB
- M'soft: ADO.NET
- JDBC (from Sun)

'UDA'

Microsoft's Universal Data Access (UDA): Collection of technologies used to access any type of data source and manage the data through a common interface

ODBC, OLE-DB and ADO.NET form the backbone of the MS UDA architecture

Native SQL Connectivity

- Connection interface provided by database vendors, which is unique to each vendor
- Interfaces are optimized for particular vendor's DBMS
- Maintenance is a burden for the programmer

ODBC, DAO+Jet, RDO

- **Open Database Connectivity (ODBC)**:Microsoft's implementation of a superset of SQL Access Group **Call Level Interface (CLI)** standard for database access
  - Widely supported database connectivity interface
  - Allows Windows application to access relational data sources by using SQL via standard **application programming interface (API)**
  - Too much of a 'low level' API, so need something more

7

JET: 'Joint Engine Technology'.

ODBC, DAO+Jet, RDO

- **Data Access Objects (DAO)**: Object-oriented API used to access MS Access, MS FoxPro, and dBase databases from Visual Basic programs
  - Provides an optimized interface that expose functionality of **Jet** data engine to programmers
  - DAO interface can be used to access other relational style data sources as well

8

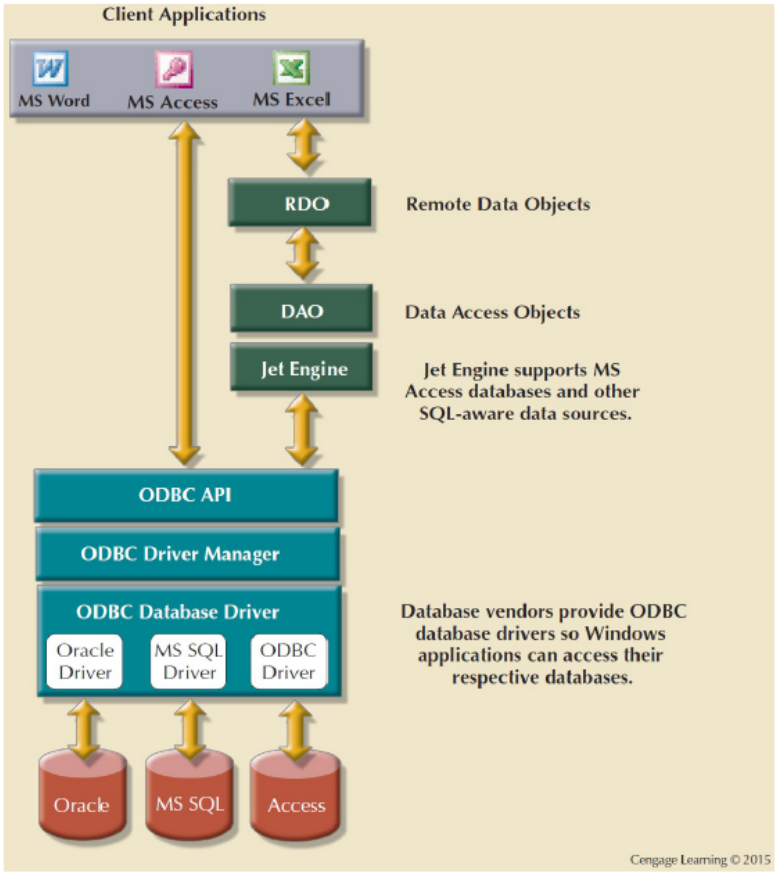
ODBC, DAO+Jet, RDO

- **Remote Data Objects (RDO)**
  - Higher-level object-oriented application interface used to access remote database servers
- **Dynamic-link libraries (DLLs)**
  - Implements ODBC, DAO, and RDO as shared code that is dynamically linked to the Windows operating environment

Components of ODBC Architecture

- High-level ODBC API through which application programs access ODBC functionality
- Driver manager that is in charge of managing all database connections
- ODBC driver that communicates directly to DBMS

Figure 14.2 - Using ODBC, DAO, and RDO to access databases



11

Object Linking and Embedding for Database (OLE-DB)

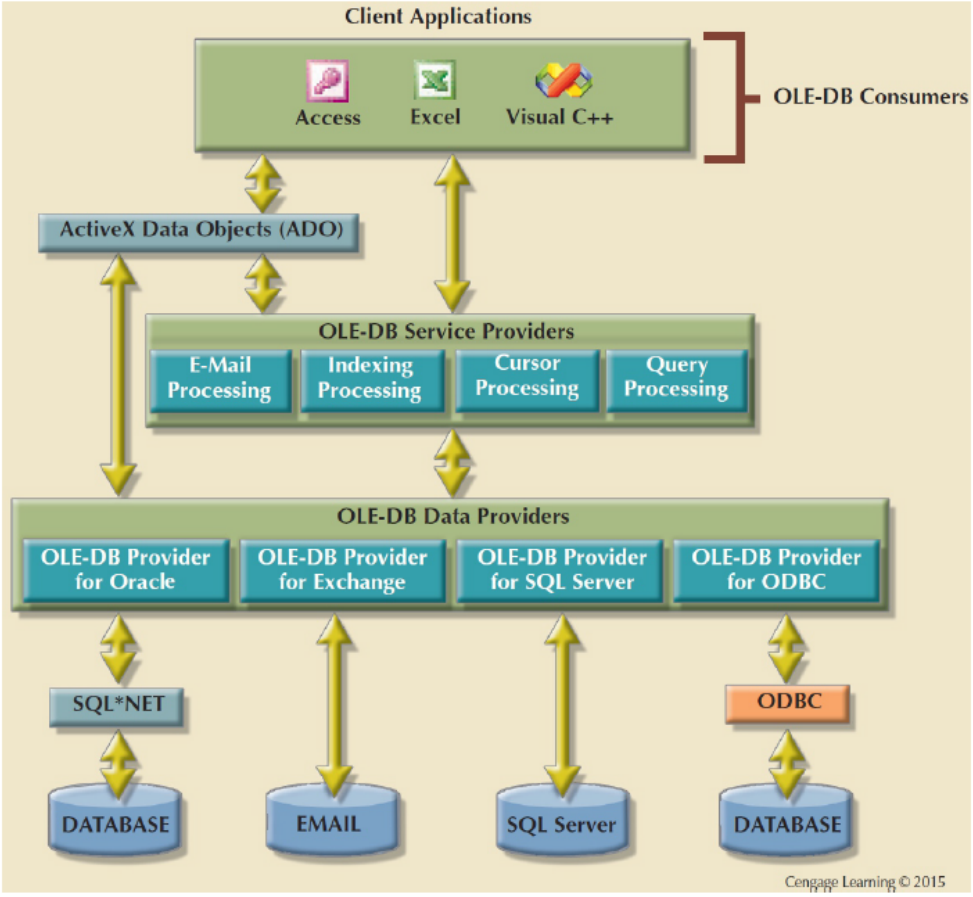
- Database middleware that adds object-oriented functionality for access to data
- Series of COM objects provides low-level database connectivity for applications
- Types of objects based on functionality
  - Consumers (request data)
  - Providers (produce data – from data sources)

COM was modeled after OMG's CORBA... Here is more...

Object Linking and Embedding for Database (OLE-DB)

- Does not provide support for scripting languages
- **ActiveX Data Objects (ADO)**: Provides:
  - High-level application-oriented interface to interact with OLE-DB, DAO, and RDO
  - Unified interface to access data from any programming language that uses the underlying OLE-DB objects

Figure 14.5 - OLE-DB Architecture



15

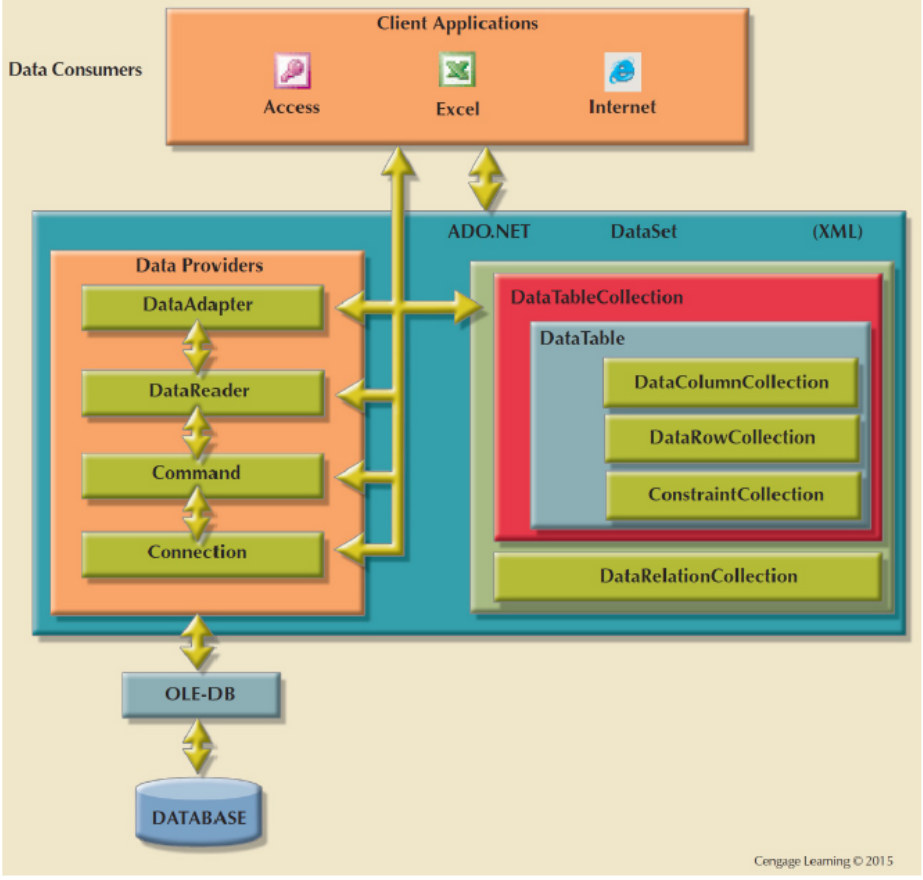
ADO.NET

- Data access component of Microsoft's .NET application development framework (improves on the ADO + O LE-DB functionality)
- **Microsoft's .NET framework**
  - Component-based platform for developing distributed, heterogeneous, interoperable applications
  - Manipulates any type of data using any combination of network, operating system, and programming language

ADO.NET

- Features critical for the development of distributed applications
  - **DataSet**: Disconnected memory-resident representation of database
  - **XML support**
    - DataSet is internally stored in XML format
    - Data in DataSet could be made persistent as XML documents

Figure 14.6 - ADO.NET Framework



18

Java Database Connectivity (JDBC)

- **Java**: Object-oriented programming language that runs on top of web browser software, on smartphones, on the desktop and on servers
- JDBC: Application programming interface that allows a Java program to interact with a wide range of data sources – a simple URL is all is needed to connect to a db

Advantages of JDBC

- Company can leverage existing technology and personnel training
- Allows direct access to database server or access via database middleware
- Allows programmers to use their SQL skills to manipulate the data in the company's databases
- Provides a way to connect to databases through an ODBC driver

JDBC usage example

Here is some sample code that uses the JDBC API:

```
public class HW2 {
  private Connection conn=null;
  private final String connection;
  private final String username;
  private final String password;
  private Statement stmt = null;

  HW2(String username,String password,String connection) {
    this.username = username;
    this.password = password;
    this.connection = connection;
  }

  void getDBConnection() {
    try {
      DriverManager.registerDriver(new oracle.jdbc.OracleDriver());
    } catch (SQLException ex) {
      System.out.println("Please install Oracle Driver.");
      return;
    }
    try {
      conn = DriverManager.getConnection(connection, username, password);
    } catch (SQLException e) {
      System.out.println(e);
      return;
    }
    if (conn != null) {
      System.out.println("Connection Succeeded.");
    } else {
      System.out.println("Connection failed.");
    }
  }

  public static void main(String[] args) {
    HW2 obj = new HW2("<Username>", "<Password>", "jdbc:oracle:thin:@localhost:1522:orcl1");
    obj.getDBConnection();
    System.out.println("Test Exit.");
  }
}
```


After we create a connection, we can create a statement, execute a query, get back a result set and iterate through it:

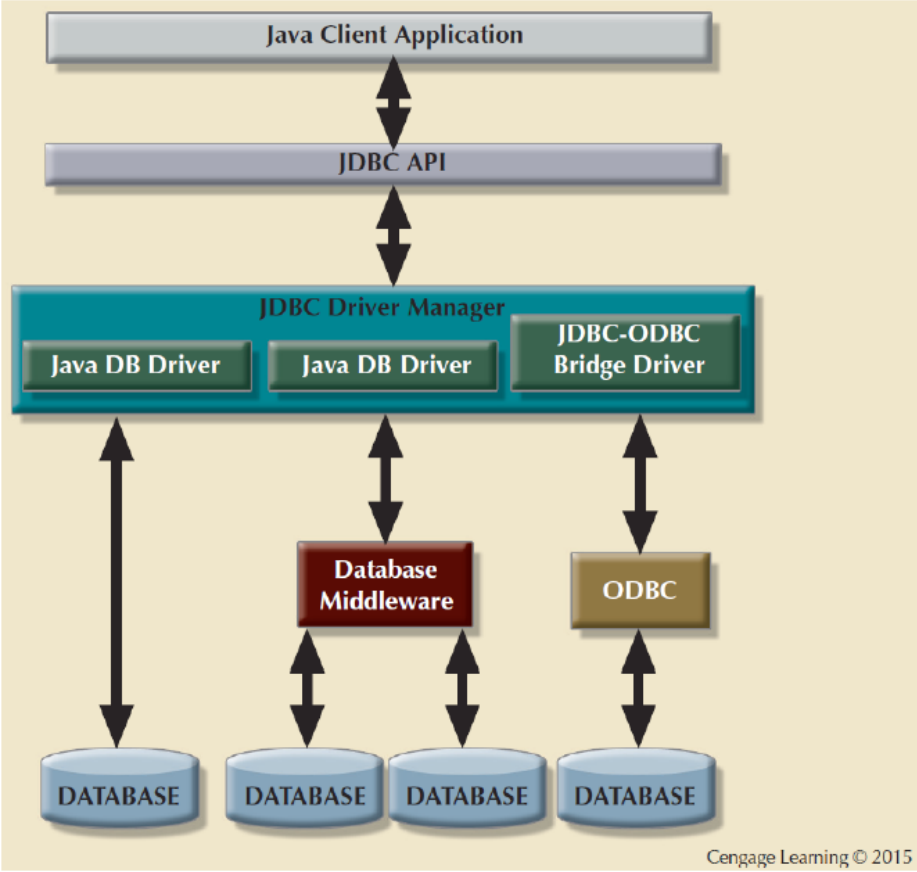
```
// create a connection via DriverManager.getConnection()
// ...

Statement myStmt = myConn.createStatement();

ResultSet myRslt= myStmt.executeQuery("....");
```


Here is a complete example.

Figure 14.7 - JDBC Architecture



23

Database Internet Connectivity

- Allows new innovative services that:
  - Permit rapid response by bringing new services and products to market quickly
  - Increase customer satisfaction through creation of web-based support services
  - Allow anywhere, anytime data access using mobile smart devices via the Internet
  - Yield fast and effective information dissemination through universal access

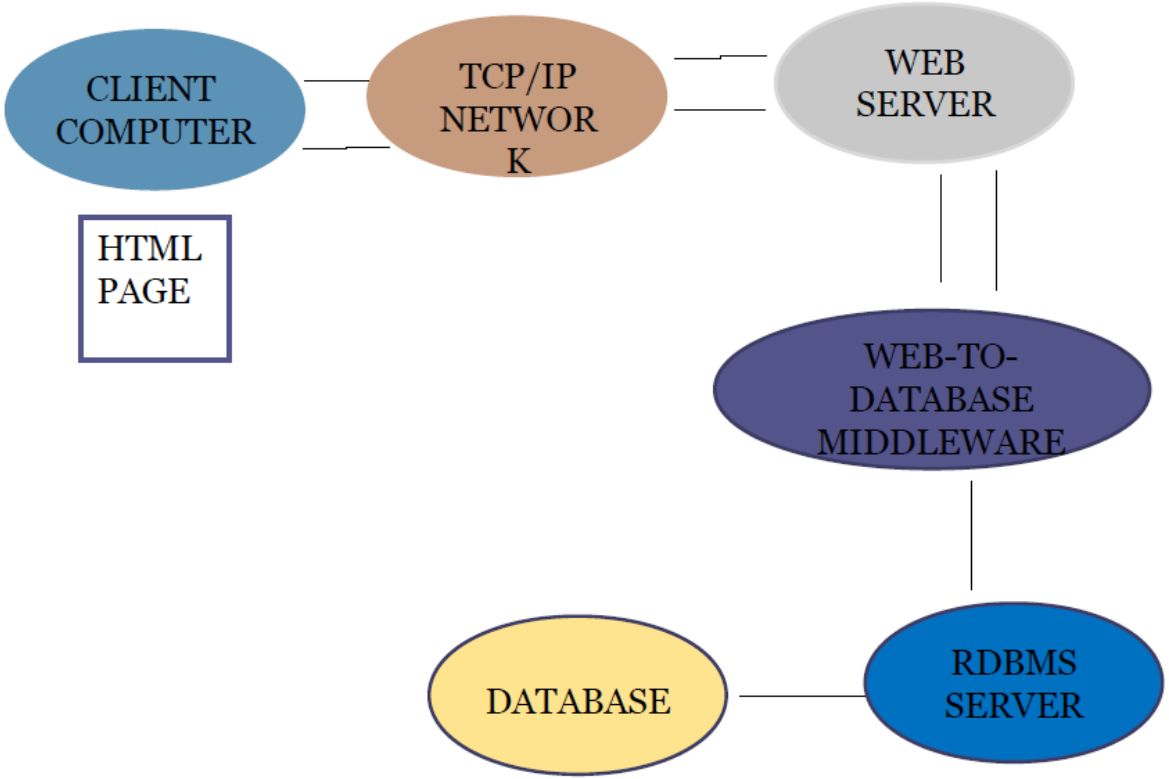
Why? One word: "e-Commerce" :)

Web-to-Database Middleware

- Web server is the main hub through which Internet services are accessed
- **Server-side extension**: Program that interacts directly with the web server
  - Also known as **web-to-database middleware**
  - Provides its services to the web server in a way that is totally transparent to the client browser

27

Web-to-Database Middleware



28

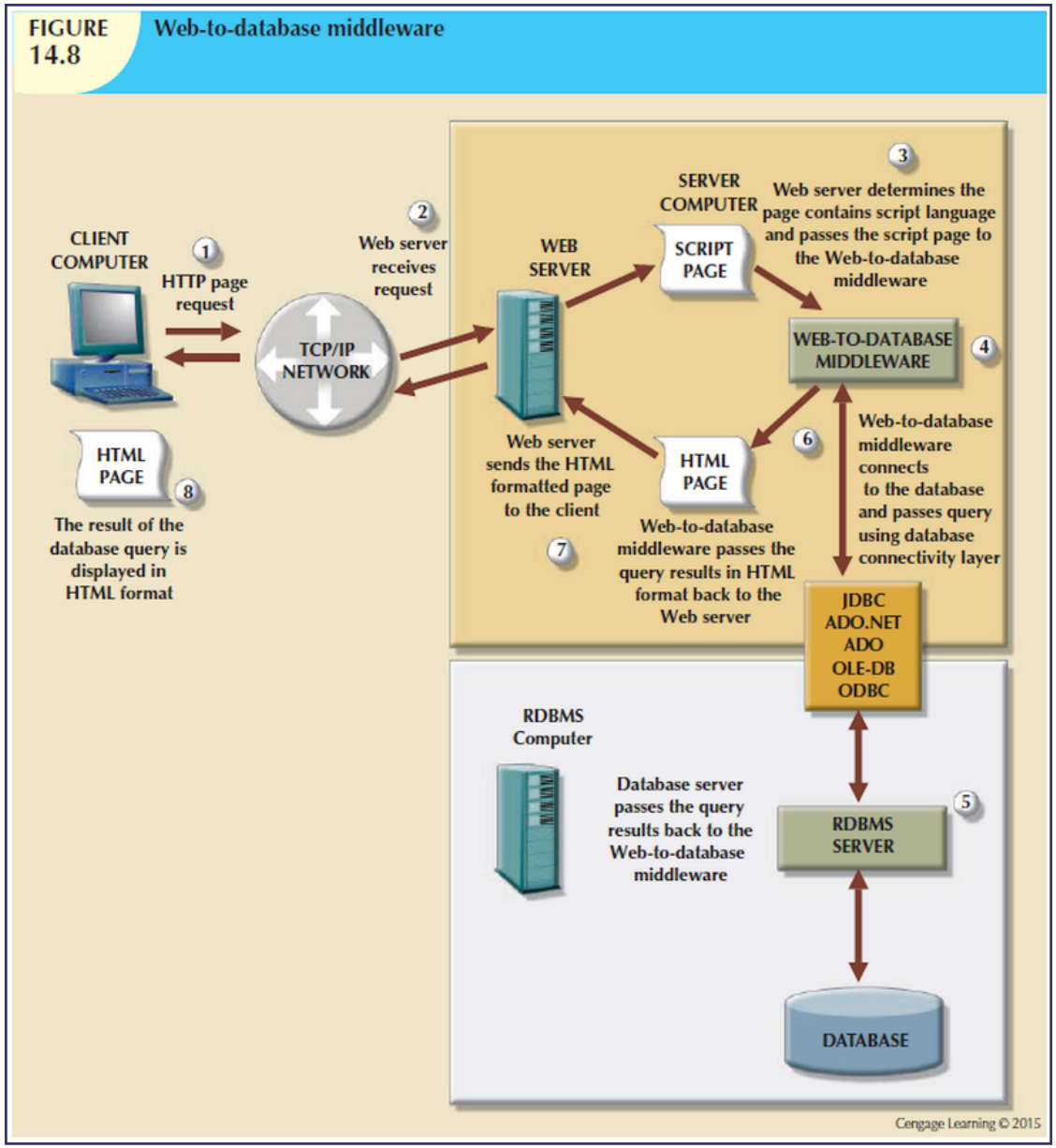
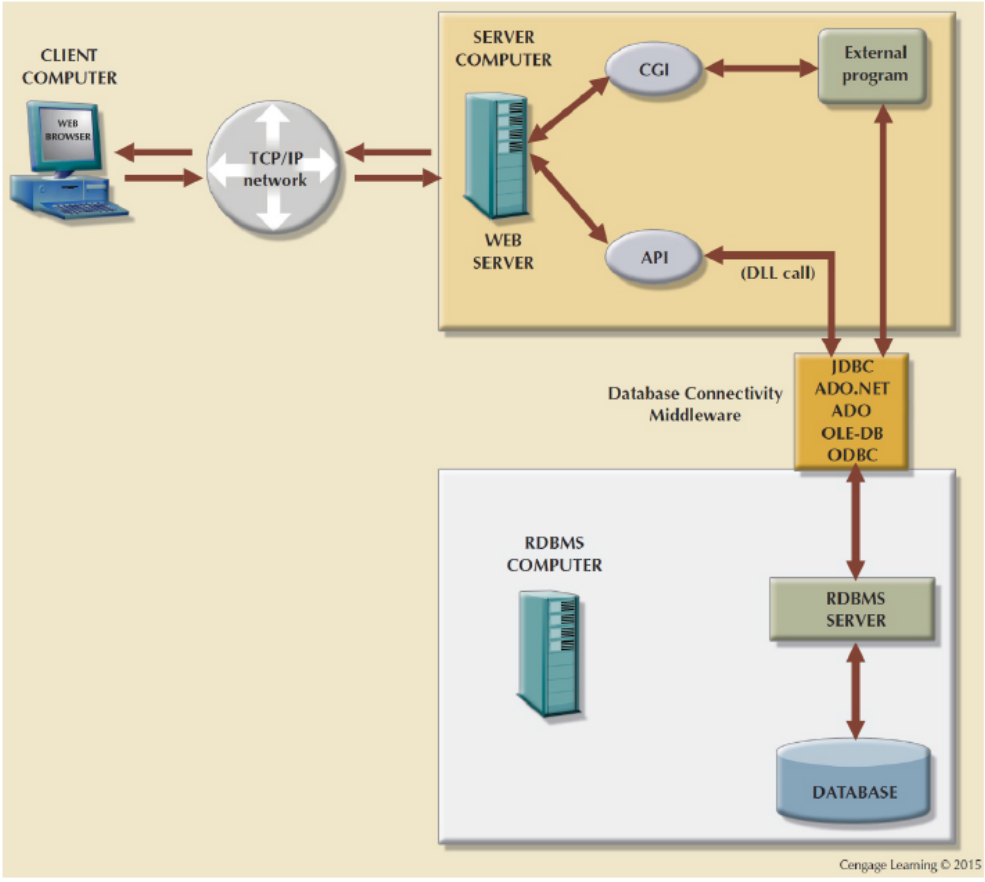


FIGURE 14.8
Web-to-database middleware
CLIENT COMPUTER
1
HTTP page request
TCP/IP NETWORK
2
Web server receives request
WEB SERVER
SERVER COMPUTER
SCRIPT PAGE
3
Web server determines the page contains script language and passes the script page to the Web-to-database middleware
WEB-TO-DATABASE MIDDLEWARE
4
Web-to-database middleware connects to the database and passes query using database connectivity layer
JDBC
ADO.NET
ADO
OLE-DB
ODBC
RDBMS Computer
RDBMS SERVER
5
DATABASE
Database server passes the query results back to the Web-to-database middleware
6
HTML PAGE
Web-to-database middleware passes the query results in HTML format back to the Web server
7
Web server sends the HTML formatted page to the client
HTML PAGE
8
The result of the database query is displayed in HTML format
Cengage Learning © 2015

Figure 14.9 - Web Server CGI and API Interfaces



29

Here is Perl 'CGI' script to interface w/ a DB.

Here are notes on using CORBA to interface a webserver with a DB. These can also be used: JSP, ASP.

In addition to CGI scripts and APIs, server side includes (SSIs) can also be used for DB connectivity.

As an FYI note, Java provides very many ways to connect a client to a DB via a webserver: applets, sockets, servlets, RMI, CORBA, agents..

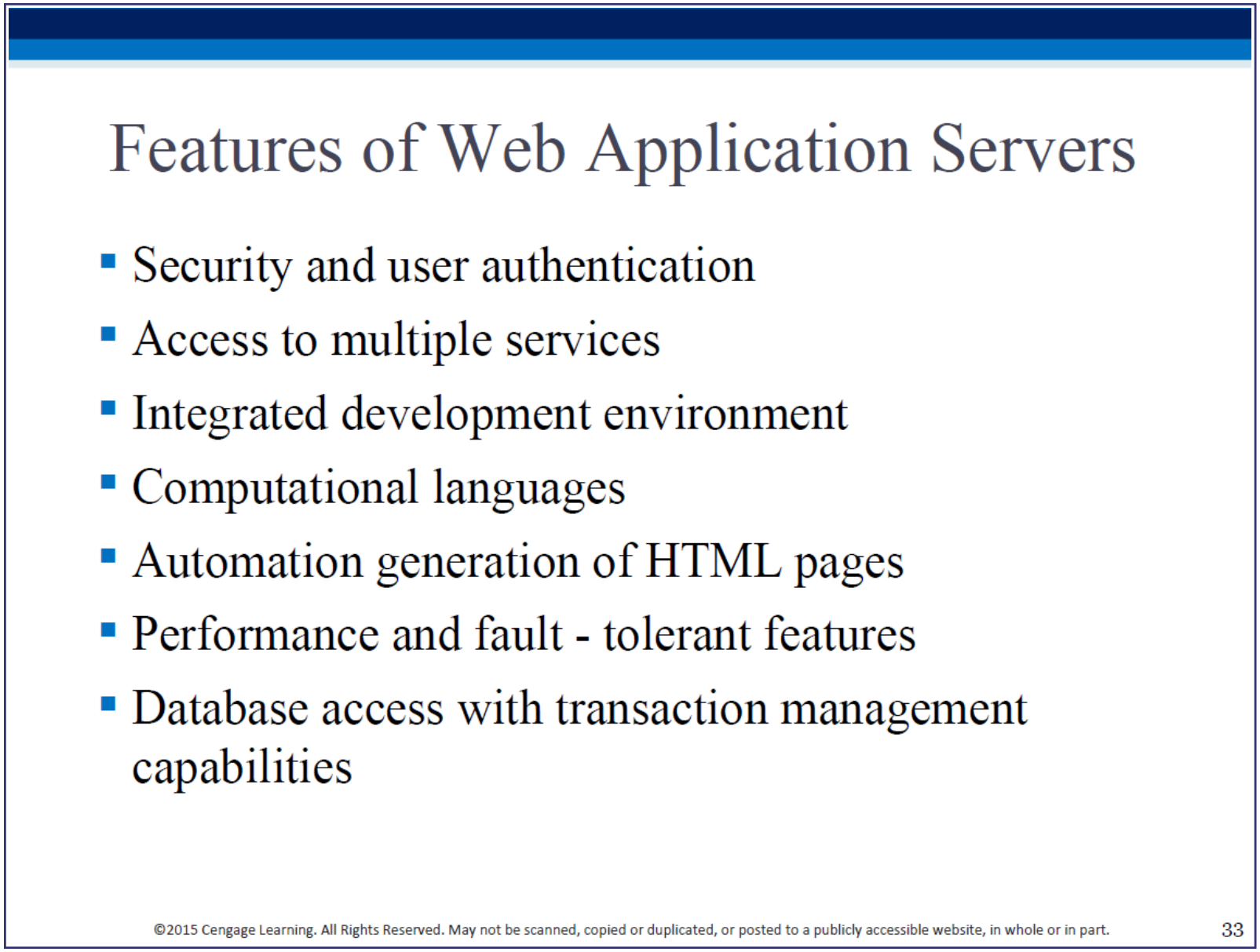
Client-Side Extensions

- Add functionality to Web browser
- Types
  - **Plug-in**: External application automatically invoked by the browser when needed
  - **Java and JavaScript**: Embedded in web page
    - Downloaded with the Web page and activated by an event
  - **ActiveX and VBScript**: Embedded in web page
    - Downloaded with page and activated by event
    - Oriented to Windows applications

Web Application Servers

- Middleware application that expands the functionality of web servers by linking them to a wide range of services
- Uses
  - Connect to and query database from web page
  - Create dynamic web search pages
  - Enforce referential integrity

A 'web application server' is a specialized server that interfaces with web services such as databases, search engines. The client (eg browser) can query these data sources and have results generated dynamically.



Examples of web application servers: WebLogic, ColdFusion/JRun, WebSphere Application Server, WebObjects, IIS, WildFly (JBoss), Tomcat, Jetty..

Each web application server (WAS) offers its own programming environment. Eg. CFML can be used to consume web services and present results for the end user (likewise for DB result sets).