	SSN COLLEGE OF ENGINEERING ***Department of Computer Science & Engineering***	***Faculty:*** ***P.Mirunalini, Asso. Prof.*** ***N.Sujaudeen, Asst. Prof.*** ***B. Senthil Kumar, Asst. Prof.***
	UCS1412 – Database Lab ***Assignment – 10***	***Assigned: 10-Mar-20*** ***Due: 2 Lab Hours***
Title: Database Application Programming using JDBC		

DB Application Programming

The JDBC API is the industry standard for database-independent connectivity between the Java programming language and a wide range of databases. The JDBC API provides a call-level API for SQL-based database access. JDBC technology allows you to use the Java programming language to exploit "Write Once, Run Anywhere" capabilities for applications that require access to enterprise data.

The JDBC API makes it possible to do three things:

- Establish a connection with a database or access any tabular data source
- Send SQL statements
- Process the results

JDBC Drive Types:

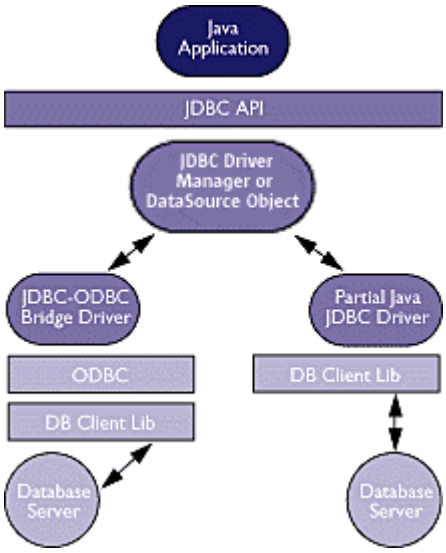
JDBC technology drivers fit into one of four categories:

Left side, Type 1: JDBC-ODBC Bridge plus ODBC Driver

This combination provides JDBC access via ODBC drivers. ODBC binary code -- and in many cases, database client code -- must be loaded on each client machine that uses a JDBC-ODBC Bridge. Sun provides a JDBC-ODBC Bridge driver, which is appropriate for experimental use and for situations in which no other driver is available.

Right side, Type 2: A native API partly Java technology-enabled driver

This type of driver converts JDBC calls into calls on the client API for Oracle, Sybase, Informix, DB2, or other DBMS. Note that, like the bridge driver, this style of driver requires that some binary code be loaded on each client machine.

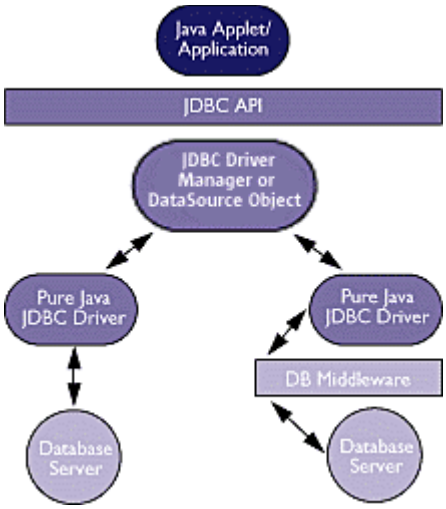


Right side, Type 3: Pure Java Driver for Database Middleware

This style of driver translates JDBC calls into the middleware vendor's protocol, which is then translated to a DBMS protocol by a middleware server. The middleware provides connectivity to many different databases.

Left side, Type 4: Direct-to-Database Pure Java Driver

This style of driver converts JDBC calls into the network protocol used directly by DBMSs, allowing a direct call from the client machine to the DBMS server and providing a practical solution for intranet access.



Requirements

- Software: The Java 2 Platform (either the Java 2 SDK, Standard Edition, or the Java 2 SDK, Enterprise Edition), an SQL database, and a JDBC technology-based driver for that database.
- Hardware: Same as for the Java 2 Platform.

Reference:

http://www.oracle.com/technetwork/java/javase/jdbc/index.html

Problem Specification:

Front-end: NetBeans IDE 6.x (Java)

Back-end: Oracle10g

Schema:

Emp_Payroll (*eid, ename, dob, sex, designation, basic, da, hra, pf, mc, gross, tot_deduc, net_pay*)

Design an interface for the above schema and perform the following operations through the application:

1. Insert (*eid, ename, dob, sex, designation, basic)*

Calculate da, hra, pf ,mc, gross, total deductions and net pay as described below and update the same in the database for the inserted employee.

2. Update (*ename, dob, sex, designation, basic)*
3. Delete
4. Search the record (using primary key)
5. Exit

To calculate the net pay of an employee, develop a PL/SQL procedure/function that accepts only the eid and basic and calculates as per the following:

Dearness Allowance [DA] = 60%

House Rent Allowance [HRA]=11%

Provident Fund [PF] = 4%

Mediclaim [MC] = 3%

Gross = Basic + DA + HRA

Total Deduction = PF + MC

Net Pay = Gross – Total Deduction

Call the procedure/function from the application by passing appropriate parameter(s) and update the corresponding record.

What you have to submit:

1. Schema Diagram with constraints
2. Interface Design, DB Connectivity and Database Updates