

A Simple Example

```
import java.sql.*;
import java.io.*;
class simple {
    public static void main (String args [])
        throws SQLException, IOException {

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
        } catch (ClassNotFoundException e) {
            System.out.println ("Could not load the driver");
        }

        String user, pass;
        Connection conn = DriverManager.getConnection
            ("jdbc:oracle:thin:@tinman.cs.gsu.edu:1521:tinman",
            "raj","raj123");

        Statement stmt = conn.createStatement ();

        ResultSet rset = stmt.executeQuery
            ("select eno,ename,zip,hdate from employees");
        while (rset.next ()) {
            System.out.println(rset.getString(1) + " " +
                               rset.getString(2) + " " +
                               rset.getString(3) + " " +
                               rset.getString(4));
        }
        stmt.close();
        conn.close();
    }
}
```

Developing JDBC Applications

The basic steps involved in developing JDBC applications are:

1. Import the JDBC classes (`java.sql.*`).
2. Load the JDBC drivers.
3. Connect to the database.
4. Interact with the database using JDBC.
5. Disconnect from database.

Using the Statement object

```
void add_catalog(Connection conn)
    throws SQLException, IOException {

    Statement stmt = conn.createStatement();

    String cnum    = readEntry("Course Number: ");
    String ctitle  = readEntry("Course Title : ");
    String query = "insert into catalog values (" +
        "'" + cnum + "'," + ctitle + "')";
    try {
        int nrow = stmt.executeUpdate(query);
    } catch (SQLException e) {
        System.out.println("Error Adding Catalog Entry");
        while (e != null) {
            System.out.println("Message:"+e.getMessage());
            e = e.getNextException();
        }
        return;
    }
    stmt.close();
    System.out.println("Added Catalog Entry");
}
```

```

void create_table(Connection conn)
    throws SQLException, IOException {
    String query = "create table deleted_scores (" +
        "sid      varchar2(5) not null," +
        "term     varchar2(10) not null," +
        "lineno   number(4) not null," +
        "compname varchar2(15) not null," +
        "points   number(4) check(points >= 0))";

    Statement stmt = conn.createStatement ();
    try {
        stmt.executeUpdate(query);
    } catch (SQLException e) {
        System.out.println("Could not create table");
        while (e != null) {
            System.out.println("Message:"+e.getMessage());
            e = e.getNextException();
        }
        return;
    }
    System.out.println("Table created");
    stmt.close();
}

```

```

void drop_student(Connection conn,
                  String term_in, String ls)
    throws SQLException, IOException {

String id = readEntry("Student ID to drop: ");
String query0 = "insert into deleted_scores " +
    "select * from scores where sid = '" + id +
    "' and term = '" + term_in + "' and lineno = " + ls;
String query1 = "delete scores where sid = '" + id +
    "' and term = '" + term_in + "' and lineno = " + ls;
String query2 = "delete enrolls where sid = '" + id +
    "' and term = '" + term_in + "' and lineno = " + ls;

conn.setAutoCommit(false);
Statement stmt = conn.createStatement ();
int nrows;
try {
    nrows = stmt.executeUpdate(query0);
    nrows = stmt.executeUpdate(query1);
    nrows = stmt.executeUpdate(query2);
} catch (SQLException e) {
    System.out.println("Could not drop student");
    while (e != null) {
        System.out.println("Message: "+e.getMessage());
        e = e.getNextException();
    }
    conn.rollback();
    return;
}
System.out.println("Dropped student");
conn.commit();
conn.setAutoCommit(true);
stmt.close();
}

```

Using the PreparedStatement object

```
void add_students(Connection conn)
    throws SQLException, IOException {

    String id, ln, fn, mi;
    PreparedStatement stmt = conn.prepareStatement(
        "insert into students values (?, ?, ?, ?)" );
    do {
        id = readEntry("ID (0 to stop): ");
        if (id.equals("0"))
            break;
        ln = readEntry("Last Name      : ");
        fn = readEntry("First Name     : ");
        mi = readEntry("Middle Initial: ");
        try {
            stmt.setString(1,id);
            stmt.setString(2,fn);
            stmt.setString(3,ln);
            stmt.setString(4,mi);
            stmt.executeUpdate();
        } catch (SQLException e) {
            System.out.println("Error adding student");
        }
    } while (true);
    stmt.close();
}
```

Using the CallableStatement object

```
import java.sql.*;
import java.io.*;
class call3 {
    public static void main (String args [])
        throws SQLException, IOException {
        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
        } catch (ClassNotFoundException e) {
            System.out.println ("Could not load the driver");
        }
        String user, pass;
        user = readEntry("userid : ");
        pass = readEntry("password: ");
        Connection conn = DriverManager.getConnection
            ("jdbc:oracle:oci7:"+user+"/"+pass);

        int enum = readNumber("Enter the employee number: ");
        int cnum = readNumber("Enter the customer number: ");
        int onum = readNumber("Enter the order      number: ");
        CallableStatement stmt = conn.prepareCall
            ("{call process_orders.add_order(?,?,?,?)}");
        stmt.setInt(1,onum);
        stmt.setInt(2,cnum);
        stmt.setInt(3,enum);
        stmt.setNull(4,Types.DATE);
        conn.setAutoCommit(false);
    }
}
```

```

try {
    stmt.executeUpdate();
} catch (SQLException e) {
    System.out.println("Could not add order");
    conn.rollback();
    return;
}
stmt = conn.prepareStatement(
    "{call process_orders.add_order_details(?,?,?)}");
do {
    int pnum = readNumber(
        "Enter the part number (0 to stop): ");
    if (pnum == 0)
        break;
    int qty = readNumber("Enter the quantity : ");
    stmt.setInt(1,onum);
    stmt.setInt(2,pnum);
    stmt.setInt(3,qty);
    try {
        stmt.executeUpdate();
    } catch (SQLException e) {
        System.out.println("Could not add odetail");
    }
} while (true);
conn.commit();
conn.setAutoCommit(true);
stmt.close();
conn.close();
}

```



```

import java.sql.*;
import java.io.*;
class call1 {
    public static void main (String args [])
        throws SQLException, IOException {
    try {
        Class.forName ("oracle.jdbc.driver.OracleDriver");
    } catch (ClassNotFoundException e) {
        System.out.println ("Could not load the driver");
    }
    String user, pass;
    user = readEntry("userid : ");
    pass = readEntry("password: ");
    Connection conn = DriverManager.getConnection
        ("jdbc:oracle:oci7:"+user+"/"+pass);

    String cnum = readEntry(
        "Enter the customer number to find city: ");
    CallableStatement stmt =
        conn.prepareCall (" {? = call get_city (?) }");
    stmt.setString(2,cnum);
    stmt.registerOutParameter(1,Types.VARCHAR);
    stmt.execute();
    String city = stmt.getString(1);
    if (stmt.isNull())
        System.out.println("Customer's city = Null");
    else
        System.out.println("Customer's city = "+
            stmt.getString(1));

    stmt.close();
    conn.close();
}
}

```

Executing SQL Queries

```
void print_report(Connection conn,
                  String term_in, String ls)
    throws SQLException, IOException {

    String query0 = "select a, b, c, d from courses " +
        "where term = '" + term_in + "'" + " and lineno = " + ls;

    String query1 = "select compname, maxpoints, weight " +
        "from components where term = '" + term_in +
        "'" + " and lineno = " + ls;

    String query2 = "select E.sid, S.lname, S.fname " +
        "from enrolls E, students S " +
        "where S.sid = E.sid and " +
        "E.term = '" + term_in + "'" + " and " +
        "E.lineno = " + ls + " order by lname, fname";

    String query3 = "select points " + "from scores " +
        "where term = '" + term_in + "'" + " and " +
        "lineno = '" + ls + "'" + " and " +
        "sid = ? and " + // substitute ? by sid
        "compname = ?"; // substitute ? by compname
```

```
double total;
int scaleA, scaleB, scaleC, scaleD;
// Read the grade cut off points from courses
Statement stmt = conn.createStatement ();
ResultSet rset0;
try {
    rset0 = stmt.executeQuery(query0);
} catch (SQLException e) {
    System.out.println("Problem reading scales");
    while (e != null) {
        System.out.println("Message:"+e.getMessage());
        e = e.getNextException();
    }
    return;
}
rset0.next();
scaleA = rset0.getInt(1);
scaleB = rset0.getInt(2);
scaleC = rset0.getInt(3);
scaleD = rset0.getInt(4);
```

```

System.out.print("SID  LNAME          FNAME          ");
// Read the component information into arrays
ResultSet rset1;
try {
    rset1 = stmt.executeQuery(query1);
} catch (SQLException e) {
    System.out.println("Problem reading components");
    while (e != null) {
        System.out.println("Message:"+e.getMessage());
        e = e.getNextException();
    }
    return;
}
String comp_names[] = new String[20];
double comp_maxpoints[] = new double[20];
double comp_weight[] = new double[20];
int ncomps=0;
while (rset1.next()) {
    System.out.print(rset1.getString(1)+" ");
    comp_names[ncomps] = rset1.getString(1);
    comp_maxpoints[ncomps] = rset1.getDouble(2);
    comp_weight[ncomps] = rset1.getDouble(3);
    ncomps++;
}

```

```
System.out.println("AVG    GRADE");
// Read the students enrolled in the class
// For each student and for each grade component,
// read the score, print and keep total
ResultSet rset2;
try {
    rset2 = stmt.executeQuery(query2);
} catch (SQLException e) {
    System.out.println("Problem reading students");
    while (e != null) {
        System.out.println("Message:"+e.getMessage());
        e = e.getNextException();
    }
    return;
}
```

```
PreparedStatement stmt2 =
    conn.prepareStatement(query3);
while (rset2.next()) {
    total = 0.0;
    System.out.print(rset2.getString(1)+" ");
    System.out.print(rset2.getString(2));
    for (int k=0;
        k < (12-rset2.getString(2).length()); k++)
        System.out.print(" ");
    System.out.print(rset2.getString(3));
    for (int k=0;
        k < (12-rset2.getString(3).length()); k++)
        System.out.print(" ");
```

```

for (int i=0; i < ncomps; i++) {
    stmt2.setString(1,rset2.getString(1));
    stmt2.setString(2,comp_names[i]);
    ResultSet rset3;
    try {
        rset3 = stmt2.executeQuery();
    } catch (SQLException e) {
        System.out.println("Problem reading scores");
        while (e != null) {
            System.out.println("Message:"+e.getMessage());
            e = e.getNextException();
        }
        return;
    }
    try {
        rset3.next();
    } catch (SQLException e) {
        System.out.println("No entry for " +
            rset2.getString(3) + " in " + comp_names[i]);
        while (e != null) {
            System.out.println("Message:"+e.getMessage());
            e = e.getNextException();
        }
        continue;
    }
}

```

```

    total = total +
        ((rset3.getDouble(1)/comp_maxpoints[i])*
         comp_weight[i]);
    System.out.print(rset3.getString(1));
}

// Print the total and grade.
Double tot = new Double(total);
for (int k2=0;
     k2 < (6-tot.toString().length()); k2++)
    System.out.print(" ");
System.out.print(total + "      ");
if (total >= scaleA) System.out.println("A");
else if (total >= scaleB) System.out.println("B");
else if (total >= scaleC) System.out.println("C");
else if (total >= scaleD) System.out.println("D");
else System.out.println("F");
}
stmt.close();
}

```


ResultSet metadata – SQL Interpreter

```
import java.sql.*;
import java.io.*;
class meta3 {
public static void main (String args [])
    throws SQLException, IOException {

    try {
        Class.forName ("oracle.jdbc.driver.OracleDriver");
    } catch (ClassNotFoundException e) {
        System.out.println ("Could not load the driver");
    }

    String user, pass;
    user = readEntry("userid : ");
    pass = readEntry("password: ");
    Connection conn = DriverManager.getConnection(
        "jdbc:oracle:oci7:"+user+"/"+pass);

    System.out.println("Welcome to the SQL Interpreter\n");
    System.out.print("SQL> ");
```

```
Statement stmt = conn.createStatement ();
do {
    String query = readQuery();
    if (query .equals("exit"))
        break;
    ResultSet rset;
    try {
        rset = stmt.executeQuery(query);
    } catch (SQLException e) {
        System.out.println("Not well formed query");
        continue;
    }
    ResultSetMetaData rsetmd = rset.getMetaData();
    int nCols;
    nCols = rsetmd.getColumnCount();
    for (int i = 1; i <= nCols; i++) {
        System.out.print(rsetmd.getColumnName(i));
        int colSize = rsetmd.getColumnDisplaySize(i);
        for (int k=0;
            k < colSize-rsetmd.getColumnName(i).length();
            k++)
            System.out.print(" ");
    }
    System.out.println("");
}
```

```

while (rset.next ()) {
    for (int i = 1; i <= nCols; i++) {
        String val = rset.getString(i);
        if (rset.isNull())
            System.out.print("null");
        else
            System.out.print(rset.getString(i));
        int colSize;
        if (rset.isNull()) colSize = 4;
        else colSize = rsetmd.getColumnDisplaySize(i);
        if (rset.isNull()) {
            for (int k=0; k < colSize-4; k++)
                System.out.print(" ");
        }
        else {
            for (int k=0;
                k < colSize-rset.getString(i).length();
                k++)
                System.out.print(" ");
        }
    }
    System.out.println("");
}
} while (true);
stmt.close();
conn.close();
System.out.println("Thank you for using the SQL"+
    " Interpreter\n");
}

```

```
//readQuery function
static String readQuery() {
    try {
        StringBuffer buffer = new StringBuffer();
        System.out.flush();
        int c = System.in.read();
        while(c != ';' && c != -1) {
            if (c != '\n')
                buffer.append((char)c);
            else {
                buffer.append(" ");
                System.out.print("SQL> ");
                System.out.flush();
            }
            c = System.in.read();
        }
        return buffer.toString().trim();
    } catch (IOException e) {
        return "";
    }
}
}
```

Oracle REF_CURSOR type

```
create or replace package refcursor_jdbc as
  type refcurtype is ref cursor;
  function get_courses (term_in varchar2)
    return refcurtype;
end refcursor_jdbc;
/
show errors
create or replace package body refcursor_jdbc as
  function get_courses (term_in varchar2)
    return refcurtype as
    rc refcurtype;
begin
  open rc for
    select lineno, courses.cno, ctitle
    from   courses, catalog
    where  courses.cno = catalog.cno and
          courses.term = term_in;
  return rc;
end;
end refcursor_jdbc;
/
show errors
```

```

import java.sql.*;
import java.io.*;
import oracle.jdbc.driver.*;

class refcur {
    public static void main (String args [])
        throws SQLException, ClassNotFoundException {

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
        } catch (ClassNotFoundException e) {
            System.out.println ("Could not load the driver");
        }

        String user, pass;
        user = readEntry("userid : ");
        pass = readEntry("password: ");
        Connection conn = DriverManager.getConnection
            ("jdbc:oracle:oci7:"+user+"/"+pass);

        String term_in = readEntry("Enter Term: ");

        CallableStatement cstmt = conn.prepareCall
            ("{ ? = call refcursor_jdbc.get_courses (?)}");

        cstmt.registerOutParameter (1, OracleTypes.CURSOR);
        cstmt.setString (2, term_in);
        cstmt.execute ();
        ResultSet rset = (ResultSet)cstmt.getObject(1);

```

```
System.out.println("Courses offered during " +  
                    term_in + " are:");  
while (rset.next ()) {  
    System.out.print(rset.getString(1) + " ");  
    System.out.print(rset.getString(2) + " ");  
    System.out.println(rset.getString(3));  
}  
cstmt.close();  
conn.close();  
}  
}
```

Processing multiple ResultSets

```
import oracle.jdbc.driver.*;
import java.sql.*;
import java.io.*;

class call2 {
    public static void main (String args [])
        throws SQLException, IOException {

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
        } catch (ClassNotFoundException e) {
            System.out.println ("Could not load the driver");
        }

        String user, pass;
        user = readEntry("userid : ");
        pass = readEntry("password: ");
        Connection conn = DriverManager.getConnection
            ("jdbc:oracle:oci7:"+user+"/"+pass);

        String cnum = readEntry("Enter customer number: ");
        CallableStatement cstmt;
        ResultSet rset;
```



```
cstmt = conn.prepareStatement
("begin " +
 "open ? for select count(*) from orders " +
 "where cno = ?;" +
 "open ? for select ono,received from orders " +
 "where cno = ?;" +
 "open ? for select orders.ono,parts.pno,pname,qty " +
 "from orders,odetails,parts where " +
 "orders.ono = odetails.ono and " +
 "odetails.pno = parts.pno " +
 "and cno = ?;" + "end;");

cstmt.setString(2,cnum);
cstmt.setString(4,cnum);
cstmt.setString(6,cnum);
cstmt.registerOutParameter (1, OracleTypes.CURSOR);
cstmt.registerOutParameter (3, OracleTypes.CURSOR);
cstmt.registerOutParameter (5, OracleTypes.CURSOR);
cstmt.execute ();
```

```

rset = ((OracleCallableStatement)cstmt).getCursor(1);
while (rset.next ()) {
    System.out.println ("Customer has " +
                        rset.getString(1) + " orders");
}
System.out.println("The orders are:");
rset = ((OracleCallableStatement)cstmt).getCursor(3);
while (rset.next ()) {
    System.out.print("Order Number " +
                    rset.getString(1));
    System.out.println(":Received on " +
                      rset.getDate(2).toString());
}
System.out.println("The order details are:");
System.out.println("ONO    PNO    PNAME    QUANTITY");
rset = ((OracleCallableStatement)cstmt).getCursor(5);
ResultSetMetaData rsetmd = rset.getMetaData();
while (rset.next ()) {
    System.out.print(rset.getString(1));
    System.out.print(" " + rset.getString(2));
    System.out.print(" " + rset.getString(3));
    for (int k=0;
         k<(rsetmd.getColumnDisplaySize(3)-
            rset.getString(3).length()); k++)
        System.out.print(" ");
    System.out.println(" " + rset.getString(4));
}
cstmt.close();
conn.close();
}
}

```

Database Metadata

```
import java.sql.*;
import java.io.*;

class meta2 {
    public static void main (String args [])
        throws SQLException, IOException {

        try {
            Class.forName ("oracle.jdbc.driver.OracleDriver");
        } catch (ClassNotFoundException e) {
            System.out.println ("Could not load the driver");
        }

        String user, pass;
        user = readEntry("userid : ");
        pass = readEntry("password: ");
        Connection conn = DriverManager.getConnection(
            "jdbc:oracle:oci7:"+user+"/"+pass);

        DatabaseMetaData dmd = conn.getMetaData();

        System.out.println("Database Product Name = " +
            dmd.getDatabaseProductName());
        System.out.println("JDBC Driver Name = " +
            dmd.getDriverName());
    }
}
```

```
System.out.println("Tables starting with C " +
                    "in schema BOOK are:");
ResultSet rset = dmd.getTables(null,"BOOK","C%",null);
while (rset.next()) {
    System.out.println(rset.getString(3));
    // print table name
}
int n = dmd.getMaxColumnsInTable();
System.out.println("Maximum number of columns " +
                    "allowed in a table = " + n);

conn.close();
}
}
```