# CS 701-Transaction Processing Using Microsoft Access

## Purpose

The purpose of this micro seminar is to discuss the process of developing transaction processing using Microsoft Access.

## Problem

The XYZ corporation needs a simple database system in which to track department, employee, and dependent history. To support the corporation's business objectives, the user needs to be able to perform the following transactions:

- Add a department
- Delete a department
- Modify a department
- Add an employee
- Delete an employee
- Modify an employee
- Add a dependent
- Delete a dependent
- Modify a dependent

## Database Schema

### Department

| | | | |
|---|---|---|---|
| <u>DNUMBER</u> | DNAME | DLOCATION | DMGRSSN |

### Employee

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| <u>SSN</u> | FNAME | MI | LNAME | BDATE | DNO | SALARY | SEX |

### Dependent

| | | | | | | |
|---|---|---|---|---|---|---|
| <u>ESSN</u> | SSN | FNAME | MI | LNAME | BDATE | BDATE |

## Create Database and tables

Look at the CS 605 tutorial for step by step pictorial description of the process.

## Create Transactions

There are times when you do not want one statement to take effect unless another one completes. The way to be sure that either both actions occur or neither action occurs is to use a transaction. A transaction is a set of one or more statements that are executed together as a unit, so either all of the statements are executed, or none of the statements is executed.

**<u>Step 1:</u>**

To make an ODBC connection to MS Access using a programming language (like java), on a 64-bit OS, these are the things you need to do (If you are using 32-bit OS, then ignore step 3 and 4):

1.) Go to Control Panel
2.) Go to Administrative Tools
3.) Right click on Data Sources and go to its properties
4.) In properties change the following and click 'OK'
    Target field to:
        %SystemRoot%\SysWOW64\odbcad32.exe
    From:
        %SystemRoot%\System32\odbcad32.exe
    Start In to:
        %SystemRoot%\SysWOW64
    From:
        %SystemRoot%\System32
5.) Having done the above, go to Data Sources, and in that go to System DSN.
6.) In System DSN click on Add and select the driver and click 'Finish'.
        Driver *.mdb (for Access 2003), *.accdb (**for** Access 2007).
7.) Having done this, select the file you want by going to 'Select' and choosing the file you want to use in your program and give the Data Source Name(DSN) the same as name the name of the file.

Note: The DSN need not be the same as the name of the file. But whatever you give the DSN, make sure you use that in your program to get Connection.

**<u>Step 2:</u>**

Now the next step is to use a java program and embed SQL queries and execute them as a transaction. The steps we need to follow are: load the driver, open connection to database, execute the queries, commit the transaction, and close the connection.

1. Code for Loading driver:

To establish a connection with the DBMS, first you must load the driver.

```
try {
      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
      System.out.println("driver loaded");
} //end of try
catch(java.lang.ClassNotFoundException e) {
      System.err.print("ClassNotFoundException: ");
      System.err.println(e.getMessage());
}//end of catch
```

2. Code for opening connection:

When a connection is created, it is in auto-commit mode. This means that each individual SQL statement is treated as a transaction and is automatically committed right after it is executed. The way to allow two or more statements to be grouped into a transaction is to disable auto-commit mode. Once auto-commit mode is disabled, no SQL statements are committed until you call the method `commit` explicitly. This is demonstrated in the following lines of code, where `conn` is an active connection:

```
                   This is the same name you specified in step one(7),DNS

try {
      conn = DriverManager.getConnection("jdbc:odbc:proj");
      System.out.println("Connected to the database");
      conn.setAutoCommit(false); ←—— This removes the auto-commit mode
}//end of try
catch (SQLException se){
      System.out.println("Not connected to database");
      System.out.println(se);
}//end of catch
```

3. Query part:

In general, executeQuery method is used for SELECT queries. The executeQuery method returns a ResultSet object. For Insert, Delete and Update queries, executeUpdate method should be used.

➢ For “Select”:

The ResultSet interface provides methods for retrieving and manipulating the results of executed queries. The executeQuery method returns a ResultSet object. A ResultSet object maintains a cursor, which points to its current row of data. When a `ResultSet` object is first created, the cursor is positioned before the first row. To move the cursor, you can use the following:

- next() — moves the cursor forward one row. Returns true if the cursor is now positioned on a row and false if the cursor is positioned after the last row.

The `ResultSet` interface declares getter methods (`getBoolean`, `getLong`, and so on) for retrieving column values from the current row. Your application can retrieve values using either the index number of the column or the name of the column. Columns are numbered from 1. Column names used as input to getter methods are case insensitive. When a getter method is called with a column name and several columns have the same name, the value of the first matching column will be returned.

The get*XXX* method of the appropriate type retrieves the value in each column. For example, the first column in each row of 'rs' is DName, which stores a value of SQL type TEXT. The method for retrieving the value is getString. The second column in each row stores a value of SQL type INT, and the method for retrieving values of that type is getInt. Each time the method next is invoked, the next row becomes the current row, and the loop continues until there is no more row in rs.

```
try{
      Statement stmt = conn.createStatement();
      ResultSet rs = stmt.executeQuery("SELECT * FROM Department");

       //to print the results of query
      while( rs.next() ){
            System.out.print(rs.getString("DName")) ;
            System.out.print("\t");
            System.out.print(rs.getInt("Dnumber")) ;
            System.out.print("\t");
            System.out.println( rs.getString("MgrSSN")) ;
      }//end of while

      rs.close() ;
      stmt.close() ;
 }//end of try
 catch (Exception excep){
            System.out.println("Failed to execute query\n"+excep);

 }//catch
```

- <u>For "Insert","Delete" and "Update":</u>

For insert, delete and update operations, the `createStatement` method has two arguments. The first argument is one of three constants added to the `ResultSet` API to indicate the type of a `ResultSet` object: `TYPE_FORWARD_ONLY`, `TYPE_SCROLL_INSENSITIVE`, and `TYPE_SCROLL_SENSITIVE`. The second argument is one of two `ResultSet` constants for specifying whether a result set is read-only or updatable: `CONCUR_READ_ONLY` and `CONCUR_UPDATABLE`. The point to remember here is that if you specify a type, you must also specify whether it is read-only or updatable. Also, you must specify the type first, and because both parameters are of type `int`, the compiler will not complain if you switch the order. Specifying the constant `TYPE_FORWARD_ONLY` creates a nonscrollable result set, that is, one in which the cursor moves only forward. If you do not specify any constants for the type and updatability of a `ResultSet` object, you will automatically get one that is `TYPE_FORWARD_ONLY` and `CONCUR_READ_ONLY`.

1. `TYPE_FORWARD_ONLY` — The result set is not scrollable; its cursor moves forward only, from before the first row to after the last row. The rows contained in the result set depend on how the underlying database materializes the results. That is, it contains the rows that satisfy the query at either the time the query is executed or as the rows are retrieved.

2. `TYPE_SCROLL_INSENSITIVE` — The result set is scrollable; its cursor can move both forward and backward relative to the current position, and it can move to an absolute position.

3. `TYPE_SCROLL_SENSITIVE` — The result set is scrollable; its cursor can move both forward and backward relative to the current position, and it can move to an absolute position.

```
try   {
Statement stmt =conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
                                            ResultSet.CONCUR_UPDATABLE);
    stmt.executeUpdate(query);
    System.out.println("Executed the query successfully");
    stmt.close() ;
}//end of try
catch (Exception excep){
      System.out.println("Failed to execute query"+excep);
 }//end of catch
```

4. <u>Commit to database:</u>

```
try {
   conn.commit();
   System.out.println("Changes successfully committed");
}//end of try
catch (Exception e){
   System.out.println("Failed to commit changes: n" + e);
   System.exit(0);
}//end of catch
```

5. <u>Close connection:</u>

```
try   {
    conn.close();
    System.out.println("Connection  successfully  closed ");
}//end of try
catch (Exception excep){
    System.out.println("Unable to close connection: n" + excep);
    System.exit(0);
}//end of catch
```

**Example:**

A sample code is given. In this example all the transactions considered are only for the department table. Please review the code incorporated to recognize the units of work described and see how they can be put together in a program.

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

public class Database
{
	static boolean exit = false;
	static Connection conn = null;

	public static void main(String[] args)
	{
		String choice;
		int N=0;

		while(!exit)
		{
			displayConsole();
			choice = captureInput();
			try
			{
				N = Integer.parseInt(choice);
			}//end of try
			catch(Exception e)
			{
				System.err.println("Enter valid option"+e);
				displayConsole();
			}//end of catch
			switchCase(N);
		}//end of while
	}// end of main


	public static void displayConsole()
	{
		boolean exit = false;

	System.out.println("***********************************************
*********");
		System.out.println("Welcome to Database tutorial ");

System.out.println("*****************************************************
***");
		System.out.println("1. Insert record to database");
		System.out.println("2. Delete record from database");
		System.out.println("3. Update record in database");
		System.out.println("4. Retrieve record in database");
```

```
            System.out.println("5. Exit");
            System.out.println("Enter your choice:");

        }// end of displayConsole

        public static String captureInput()
        {
              String choice="";
              try
              {
                    BufferedReader in = new BufferedReader(new
                                            InputStreamReader(System.in));
                choice = in.readLine();
            }//try
            catch(IOException e)
            {
              System.out.println("Enter valid numeric choice");
            }//end of catch
          return choice;
        }//end of captureInput

        public static void switchCase(int N)
        {
            switch (N)
            {
             case 1:
             {
               String DName;
               String Dnumber;
               String MgrSSN;
               boolean quit=false;
               while(!quit)
               {
                      System.out.println("Enter Department Name:");
                    DName = captureInput();
                    System.out.println("Enter Department Number:");
                    Dnumber = captureInput();
                    System.out.println("Enter Department Manager SSN");
                    MgrSSN = captureInput();

                    String query="INSERT INTO Department VALUES('" +
                                    DName + "',"+Dnumber+","+MgrSSN+")";
                    loadDriver();
                    openConnection();
                    try
                    {
                      Statement stmt =
conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR
_UPDATABLE);
                      stmt.executeUpdate(query);
                      System.out.println("Executed the query
                                                    successfully");
                      ResultSet rs = stmt.executeQuery("SELECT * FROM
                                                       Department");

                      while( rs.next() )
                      {
```

```
                     System.out.print(rs.getString("DName")) ;
                     System.out.print("\t\t");

                     System.out.print(rs.getString("Dnumber"));
                     System.out.print("\t");
                     System.out.println( rs.getString("MgrSSN")) ;
                   }//end of while

                    rs.close() ;
                    stmt.close() ;
                  }//end of try
                  catch (Exception excep){
                  System.out.println("A record with this primary key
value already exists or table to which you are trying to make changes
is opened\n" + excep);
                    System.out.println("Failed to execute query");
                  }//end of catch
                   commitDb();
                   closeConnection();
                  System.out.println("Do u want to go back to main
menu,press y to do so");
                   String ch=captureInput();
                   if(ch.equals("Y") || ch.equals("y"))
                   {
                        quit=true;
                   }//end of if
             }//end of while
             break;
           }//end of case 1

           case 2:{
               String DName=null;
               String Dnumber=null;
               String MgrSSN=null;
               String ch;
               boolean quit=false;

               while(!quit)
               {
                     boolean done=false;
                     loadDriver();
                   openConnection();
                   try  {
                     Statement stmt =
conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR
_UPDATABLE);
                     ResultSet rs = stmt.executeQuery("SELECT * FROM
Department");
                     while( rs.next() ){

System.out.println(rs.getString("DName")+"\t"+rs.getString("Dnumber")+"
\t"+rs.getInt("MgrSSN")) ;
                     }//end of while
                     while(!done)
                     {
                           rs.first();
                           System.out.println("Enter a valid department
number of the record you want to delete");
                           ch = captureInput();
                           while( rs.next() ){
                                 if(rs.getString("Dnumber").equals(ch))
                                 {
                                       done=true;
                                       DName=rs.getString("DName");
                                       Dnumber=ch;
                                       MgrSSN=rs.getString("MgrSSN");
                                 }//end of if
                           }//end of rs while
                     }//end of done while

                     String query="DELETE FROM Department WHERE
DName='"+DName+"' AND DNumber='"+Dnumber+"' AND MgrSSN='"+MgrSSN+"'";
                     //System.out.println(query);
                     stmt.executeUpdate(query);
                     System.out.println("Executed the query
successfully");
                     rs = stmt.executeQuery("SELECT * FROM
Department");

                     while( rs.next() ){
                           System.out.print(rs.getString("DName")) ;
                           System.out.print("\t");
                           System.out.print(rs.getString("Dnumber")) ;
                           System.out.print("\t");
                           System.out.println( rs.getString("MgrSSN"))
;
                     }//end of while
                     rs.close() ;
                     stmt.close() ;
                   }//end of try
                   catch (Exception excep){
                     System.out.println("The table to which you are
trying to make changes is open,please close\n" + excep);
                     System.out.println("Failed to execute query");
                   }//catch
                   commitDb();
                   closeConnection();
                   System.out.println("Do u want to go back to main
menu,press y to do so");
                   String op=captureInput();
                   if(op.equals("Y") || op.equals("y")){
                     quit=true;
                   }//end of if
               }//end of while
               break;
           }//end of case 2

           case 3:{
               String DName=null;
               String Dnumber=null;
               String MgrSSN=null;
               String ch;
               String query;
               boolean quit=false;

               while(!quit)
               {
                     boolean done=false;
                     loadDriver();
                   openConnection();
                   try  {
                     Statement stmt =
conn.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR
_UPDATABLE);
                     ResultSet rs = stmt.executeQuery("SELECT * FROM
Department");
                     while( rs.next() ){

System.out.println(rs.getString("DName")+"\t"+rs.getString("Dnumber")+"
\t"+rs.getInt("MgrSSN")) ;
                     }//end of while
                     while(!done)
                     {
                           rs.first();
                           System.out.println("Enter a valid department
number of the record you want to update:");
                           ch = captureInput();
                           while( rs.next() ){
                                 if(rs.getString("Dnumber").equals(ch))
                                 {
                                       done=true;
                                       DName=rs.getString("DName");
                                       MgrSSN=rs.getString("MgrSSN");
                                 }//end of if
                           }//end of rs while
                     }//end of done while
                     System.out.println("Enter a new department number
you want to update the record to:");
                     ch = captureInput();
                     Dnumber=ch;
                     query="UPDATE Department SET DNumber='"+Dnumber+"'
WHERE DName='"+DName+"' AND MgrSSN='"+MgrSSN+"'";
                     //System.out.println(query);
                     stmt.executeUpdate(query);
                     System.out.println("DO you want to update the
DName of the record,press y to update");
                     ch = captureInput();
                     if(ch.equals("y") || ch.equals("Y"))
                     {
                           System.out.println("Enter a new department
name you want to update the record to:");
                           ch = captureInput();
                           DName=ch;
                     }//end of if
                     System.out.println("DO you want to update the
MgrSSN of the record,press y to update");
                     ch = captureInput();
                     if(ch.equals("y") || ch.equals("Y"))
                     {
                           System.out.println("Enter a new department
name you want to update the record to:");
                           ch = captureInput();
                           MgrSSN=ch;
                     }//end of if
                     query="UPDATE Department SET DName='"+DName+"'
,MgrSSN='"+MgrSSN+"' WHERE DNumber='"+Dnumber+"'";
                     //System.out.println(query);
                     stmt.executeUpdate(query);
                     System.out.println("Executed the query
successfully");
                     rs = stmt.executeQuery("SELECT * FROM
Department");
                     while( rs.next() ){
                           System.out.print(rs.getString("DName")) ;
                           System.out.print("\t");
                           System.out.print(rs.getString("Dnumber")) ;
                           System.out.print("\t");
                           System.out.println( rs.getString("MgrSSN"))
;
                     }//end of while

                     rs.close() ;
                     stmt.close() ;
                   }//end of  try
                   catch (Exception excep){
                     System.out.println("A record with this primary key
value already exists or table to which you are trying to make changes
is opened\n" + excep);
                              System.out.println("Failed to execute
query");
                   }//catch
                   commitDb();
                   closeConnection();
                   System.out.println("Do u want to go back to main
menu,press y to do so");
                   String op=captureInput();
                   if(op.equals("Y") || op.equals("y")){
                     quit=true;
                   }//end of if
               }//end of while
               break;
           }//end of case 3

           case 4:
           {
               loadDriver();
                   openConnection();
                   try{
                        Statement stmt = conn.createStatement();
                        ResultSet rs = stmt.executeQuery("SELECT * FROM
Department");

                        while( rs.next() ){
                              System.out.print(rs.getString("DName")) ;
                          System.out.print("\t");
                          System.out.print(rs.getString("Dnumber")) ;
```

```
                              System.out.print("\t");
                              System.out.println( rs.getString("MgrSSN")) ;
                            }//end of while

                            rs.close() ;
                            stmt.close() ;
                       }//end of try
                       catch (Exception excep){
                            System.out.println("Failed to execute
query\n"+excep);

                       }//catch
                       commitDb();
                       closeConnection();
                       System.out.println("Do u want to go back to main
menu,press any key to do so");
                       String op=captureInput();
                       break;
               }//end of case 4

               case 5:
               {
                   exit=true;
                   System.out.println("Have a nice day..!!");
                   break;
                }//end of case 5
               }//end of switch
          }//end of switchCase

          public static void loadDriver(){
                try {
                      Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                      System.out.println("driver loaded");
                } //end of try
                      catch(java.lang.ClassNotFoundException e) {
                            System.err.print("ClassNotFoundException: ");
                      System.err.println(e.getMessage());
                }//end of catch
         }// end of loadDriver

          public static void openConnection(){
                try {
                      conn = DriverManager.getConnection("jdbc:odbc:proj");

                      System.out.println("Connected to the database");
                      conn.setAutoCommit(false);
                }//end of try
                      catch (SQLException se){
                            System.out.println("Not connected to
database");
                      System.out.println(se);
                }//end of catch

            }//end of openConnection

          public static void commitDb(){
                try {
                      conn.commit();
                      System.out.println("Changes successfully committed");
                }//end of try
                      catch (Exception e){
                      System.out.println("Failed to commit changes: n" +
    e);
                      System.exit(0);
                }//end of catch
        }//end of commitDb


          public static void closeConnection(){
                try   {
                      conn.close();
                      System.out.println("Connection  successfully  closed
    ");
                }//end of try
                      catch (Exception excep){
                      System.out.println("Unable to close connection: n" +
    excep);
                      System.exit(0);
                }//end of catch
          }//end of closeConnection


     }//end of Database
```