```java
package DBLayer;

import ModelLayer.*;

import java.sql.*;
import java.util.ArrayList;


public class DBEmployee implements IFDBEmployee
{

   private Connection con;

   public DBEmployee()
   {
         //**Opretter en ny instance af DBmfEmployee
         con = DbConnection.getInstance().getDBcon();
   }

   public ArrayList<Employee> getAllEmployees(boolean
   retrieveAssociation)
   {
         return miscWhere("", retrieveAssociation);
   }

   public Employee findEmployeeByFname(String fname, boolean
   retrieveAssociation)
   {
         String wClause = "fname like '%" + fname + "%'";
         System.out.println("SearchEmployee" + wClause);
         return singleWhere(wClause, retrieveAssociation);
   }

   public Employee findEmployeeByLname(String lname, boolean
   retrieveAssociation)
   {
         String wClause = "lname like '%" + lname + "%'";
         System.out.println("SearchEmployee" + wClause);
         return singleWhere(wClause, retrieveAssociation);
   }

   public Employee findEmployeeByID(int employeeID, boolean
   retrieveAssociation)
   {
         String wClause = "employeeID = " + employeeID;
         return singleWhere(wClause, retrieveAssociation);
   }

   public int insertEmployee (Employee emp) throws Exception
   {
   int rc = -1;
   String qry = "INSERT INTO mfEmployee(fname, lname, custID,
   address, zipcode, city, phoneno, email, type) VALUES('"+
               emp.getEmployeeID() + "', '" +
               emp.getFname() + "', '" +
               emp.getLname() + "', '" +
               emp.getAddress() + "', '" +
               emp.getZipCode() + "', '" +
               emp.getCity() + "', '" +
               emp.getPhoneNo() + "', '" +
               emp.getEmail() + "')";

         System.out.println("insert : " + qry);
         try
```

```
65          {
66                Statement stmt = con.createStatement();
67            stmt.setQueryTimeout(5);
68              rc = stmt.executeUpdate(qry);
69            stmt.close();
70          }
71          catch(SQLException ex)
72          {
73                System.out.println("Employee er ikke oprettet");
74                throw new Exception("Employee er ikke tilføjet");
75          }
76          return (rc);
77
78    }
79
80    public int updateEmployee(Employee emp)
81    {
82          Employee empObj = emp;
83          int rc= -1;
84
85          String qry = "UPDATE mfEmployee SET" +
86                      "employeeID ='" + empObj.getEmployeeID() + "', " +
87                      "fname = '" + empObj.getFname()+"', " +
88                      "lname ='" + empObj.getLname() + "', " +
89                      "address ='" + empObj.getAddress() + "', " +
90                      "zipCode ='" + empObj.getZipCode() + "', " +
91                      "city ='" + empObj.getCity() + "', " +
92                      "phoneNo ='" + empObj.getPhoneNo() + "', " +
93                      "email ='" + empObj.getEmail() + "', ";
94                      System.out.println("Update query: " + qry);
95                      try
96                      {
97                            Statement stmt = con.createStatement();
98                            stmt.setQueryTimeout(5);
99                            rc = stmt.executeUpdate(qry);
100
101                           stmt.close();
102                     }
103                     catch(Exception ex)
104                     {
105                           System.out.println("Update exception in mfEmployee
db:
106                           " + ex);
107                     }
108                     return(rc);
109
110   }
111
112   public int deleteEmployee(int employeeID)
113   {
114         int rc =-1;
115
116                     String qry = "DELETE FROM mfEmployee WHERE employeeID =
117                     '" + employeeID + "'";
118                     System.out.println(qry);
119                     try
120                     {
121                           Statement stmt = con.createStatement();
122                           stmt.setQueryTimeout(5);
123                           rc = stmt.executeUpdate(qry);
124                           stmt.close();
125                     }
126                      catch(Exception ex)
127                      {
```

```
                                    System.out.println("Delete exception in
mfEmployee
                                    db: "+ex);
                        }
                      return(rc);
   }

   private Employee buildEmployee(ResultSet results)
   {
         Employee empObj = new Employee();
         try
         {
               empObj.setEmployeeID(results.getInt("employeeID"));
               empObj.setFname(results.getString("fname"));
               empObj.setLname(results.getString("lname"));
               empObj.setAddress(results.getString("address"));
               empObj.setZipCode(results.getInt("zipcode"));
               empObj.setCity(results.getString("city"));
               empObj.setPhoneNo(results.getInt("phoneNo"));
               empObj.setEmail(results.getString("email"));


         }
         catch(Exception e)
         {
               System.out.println("error in building employee obj");
         }
         return empObj;
   }

   private String buildQuery(String wClause)
   {
         String query = "SELECT employeeID, fname, lname, address,
         zipcode, city, phoneNo, email FROM mfEmployee";
         if (wClause.length()>0)
               query = query + " WHERE " + wClause;
         return query;
   }

   private ArrayList<Employee> miscWhere(String wClause,
   boolean retrieveAssociation)
   {
         ResultSet results;
         ArrayList<Employee> employees = new ArrayList<Employee>();

         String qry = buildQuery(wClause);

         try
         {
                Statement stmt = con.createStatement();
            stmt.setQueryTimeout(5);
            results = stmt.executeQuery(qry);


         while(results.next())
         {
               Employee empObj = new Employee();
               empObj = buildEmployee(results);
               employees.add(empObj);
         }
         stmt.close();
         return employees;
   }
         catch(Exception e)
```

```
191         {
192               System.out.println(e.getMessage());
193               return null;
194         }
195   }
196
197   private Employee singleWhere(String wClause, boolean
198   retrieveAssociation)
199   {
200         ResultSet results;
201         Employee empObj = new Employee();
202         String qry = buildQuery(wClause);
203         System.out.println(qry);
204         try
205         {
206               Statement stmt = con.createStatement();
207               stmt.setQueryTimeout(5);
208               results = stmt.executeQuery(qry);
209               if(results.next())
210               {
211                     empObj = buildEmployee(results);
212               }
213               stmt.close();
214         }
215         catch(Exception e)
216         {
217               System.out.println("Query exception: " + e);
218         }
219         return empObj;
220   }
221
222 }
```