

MySQLOperatoerDAO.java

```

1 package webapplication.datalayer;
2
3 import java.sql.ResultSet;
4
5
6
7
8
9
10
11
12
13
14 public class MySQLOperatoerDAO implements OperatoerDAO {
15     SQLMapper map = new SQLMapper();
16
17     public MySQLOperatoerDAO(){
18         try { new Connector(); }
19         catch (InstantiationException e) { e.printStackTrace(); }
20         catch (IllegalAccessException e) { e.printStackTrace(); }
21         catch (ClassNotFoundException e) { e.printStackTrace(); }
22         catch (SQLException e) { e.printStackTrace(); }
23     }
24
25
26     @Override
27     public OperatoerDTO getOperatoer(int oprId) throws DALException {
28
29         String[] values = new String[]{Integer.toString(oprId)};
30
31         String statement = map.getStatement("opr_SELECT");
32         statement = map.insertValuesIntoString(statement, values);
33         System.out.println("Query: "+statement);
34         ResultSet rs = Connector.doQuery(statement);
35
36         OperatoerDTO temp;
37
38         try {
39             //Get the operatoer
40             if (!rs.first()) throw new DALException("Operatoeren " + oprId + " findes
ikke");
41             temp = new OperatoerDTO (rs.getInt("opr_id"), rs.getString("opr_navn"),
rs.getString("ini"), rs.getString("password"), rs.getBoolean("opr_active"));
42
43             /*ROLES START*/
44             rs = Connector.doQuery(map.getStatement("ro_SELECT_ALL"));
45
46             List<RolePair> roles = new ArrayList<RolePair>();
47             while (rs.next()){
48                 roles.add( new RolePair(rs.getInt("opr_id"), rs.getString("role")));
49             }
50
51             for(RolePair role: roles){
52                 if(role.getOprId() == temp.getOprId()){
53                     temp.addRole(role.getRole());
54                 }
55             }
56             /*END OF ROLES*/
57
58             return temp;
59         }
60         catch (SQLException e) {throw new DALException(e); }
61     }
62
63
64     @Override
65     public List<OperatoerDTO> getOperatoerList() throws DALException {
66         List<OperatoerDTO> list = new ArrayList<OperatoerDTO>();
67
68
69

```

```

70     OperatoerDTO temp;
71     try
72     {
73         ResultSet rs = Connector.doQuery(map.getStatement("opr_SELECT_ALL"));
74
75         while (rs.next())
76         {
77             temp = new OperatoerDTO(rs.getInt("opr_id"), rs.getString("opr_navn"),
rs.getString("ini"), rs.getString("password"), rs.getBoolean("opr_active"));
78             list.add(temp);
79         }
80
81         rs = Connector.doQuery(map.getStatement("ro_SELECT_ALL"));
82
83         /*WORKS*/
84         List<RolePair> roles = new ArrayList<RolePair>();
85         while (rs.next()){
86             roles.add( new RolePair(rs.getInt("opr_id"), rs.getString("role")));
87         }
88         /*END OF WORKS*/
89
90         for(OperatoerDTO opr : list){
91             for(RolePair role: roles){
92                 if(role.getOprId() == opr.getOprId())
93                     opr.addRole(role.getRole());
94             }
95         }
96
97     }
98     catch (SQLException e) { throw new DAException(e); }
99     return list;
100 }
101
102 @Override
103 public void createOperatoer(OperatoerDTO opr) throws DAException {
104
105     /*INSERT OPERATOER INTO DATABASE*/
106     String statement = map.getStatement("opr_INSERT");
107     String[] values = new String[]{Integer.toString(opr.getOprId()), opr.getOprNavn(),
opr.getIni(), opr.getPassword(), String.valueOf(opr.getOprActive())};
108     statement = map.insertValuesIntoString(statement, values);
109     System.out.println(statement);
110     Connector.doUpdate(statement);
111
112     /*INSERT THE ROLES OF THE OPERATOER INTO THE DATABASE TABLE 'role'*/
113     for(String role : opr.getRoles()){
114         statement = map.getStatement("ro_INSERT");
115         values = new String[]{Integer.toString(opr.getOprId()), role};
116         statement = map.insertValuesIntoString(statement, values);
117         System.out.println(statement);
118         Connector.doUpdate(statement);
119     }
120
121 }
122
123 @Override
124 public void updateOperatoer(OperatoerDTO opr) throws DAException {
125     //Update operatoer table
126     String statement = map.getStatement("opr_UPDATE");
127     String[] values = new String[]{opr.getOprNavn(), opr.getIni(), opr.getPassword(),
String.valueOf(opr.getOprActive()), Integer.toString(opr.getOprId())}; //Only difference
from INSERT is operatoer id is at the end

```

MySQLOperatoerDAO.java

```

128     statement = map.insertValuesIntoString(statement, values);
129     System.out.println(statement);
130     Connector.doUpdate(statement);
131
132     statement = map.getStatement("ro_DELETE");
133     values = new String[]{Integer.toString(opr.getOprId())};
134     statement = map.insertValuesIntoString(statement, values);
135     System.out.println(statement);
136     Connector.doUpdate(statement);
137
138     for(String role : opr.getRoles()){
139         statement = map.getStatement("ro_INSERT");
140         values = new String[]{Integer.toString(opr.getOprId()), role};
141         statement = map.insertValuesIntoString(statement, values);
142         System.out.println(statement);
143         Connector.doUpdate(statement);
144     }
145
146 }
147
148 private class RolePair{
149     private int oprId;
150     private String role;
151
152     RolePair(int oprId, String role){
153         this.oprId = oprId;
154         this.role = role;
155     }
156
157     public int getOprId (){return oprId;}
158     public String getRole(){return role;}
159 }
160
161 }
162

```