

MySQLRaavareBatchDAO.java

```

1 package webapplication.datalayer;
2
3 import java.sql.ResultSet;
4
5
6
7
8
9
10
11
12
13
14
15 /**
16  * @author Danny
17  *
18  */
19 public class MySQLRaavareBatchDAO implements RaavareBatchDAO {
20     SQLMapper map = new SQLMapper();
21
22     public MySQLRaavareBatchDAO(){
23         try { new Connector(); }
24         catch (InstantiationException e) { e.printStackTrace(); }
25         catch (IllegalAccessException e) { e.printStackTrace(); }
26         catch (ClassNotFoundException e) { e.printStackTrace(); }
27         catch (SQLException e) { e.printStackTrace(); }
28     }
29
30     @Override
31     public RaavareBatchDTO getRaavareBatch(int rbId) throws DALException {
32         String statement = map.getStatement("rab_SELECT");
33         String[] values = new String[]{Integer.toString(rbId)};
34         statement = map.insertValuesIntoString(statement, values);
35         System.out.println("Query: "+statement);
36         ResultSet rs = Connector.doQuery(statement);
37
38         try {
39             if (!rs.first()) throw new DALException("Raavarebatch " + rbId + " findes
ikke");
40             return new RaavareBatchDTO (rs.getInt("rb_id"), rs.getInt("raavare_id"),
rs.getDouble("maengde"));
41         }
42         catch (SQLException e) {throw new DALException(e); }
43     }
44
45     @Override
46     public List<RaavareBatchDTO> getRaavareBatchList() throws DALException {
47         List<RaavareBatchDTO> list = new ArrayList<RaavareBatchDTO>();
48         ResultSet rs = Connector.doQuery(map.getStatement("rab_SELECT_ALL")); //Without
value inserted. Returns all rows
49         try
50         {
51             while (rs.next())
52             {
53                 list.add(new RaavareBatchDTO (rs.getInt("rb_id"), rs.getInt("raavare_id"),
rs.getDouble("maengde")));
54             }
55         }
56         catch (SQLException e) { throw new DALException(e); }
57         return list;
58     }
59
60     @Override
61     public List<RaavareBatchDTO> getRaavareBatchList(int raavareId) throws DALException {
62         List<RaavareBatchDTO> list = new ArrayList<RaavareBatchDTO>();
63         String statement = map.getStatement("rab_SELECT_ALL_raavare_id");
64         statement = map.insertValuesIntoString(statement, new
String[]{Integer.toString(raavareId)});
65         ResultSet rs = Connector.doQuery(statement);
66         try

```

MySQLRaavareBatchDAO.java

```

67     {
68         while (rs.next())
69         {
70             list.add(new RaavareBatchDTO (rs.getInt("rb_id"), rs.getInt("raavare_id"),
rs.getDouble("maengde")));
71         }
72     }
73     catch (SQLException e) { throw new DALException(e); }
74     return list;
75 }
76
77 @Override
78 public void createRaavareBatch(RaavareBatchDTO raavarebatch) throws DALException {
79     String statement = map.getStatement("rab_INSERT");
80     String[] values = new String[]{
81         Integer.toString(raavarebatch.getRbId()),
82         Integer.toString(raavarebatch.getRaavareId()),
83         Double.toString(raavarebatch.getMaengde())
84     };
85     statement = map.insertValuesIntoString(statement, values);
86     System.out.println(statement);
87     Connector.doUpdate(statement);
88
89 }
90
91 @Override
92 public void updateRaavareBatch(RaavareBatchDTO raavarebatch) throws DALException {
93     String statement = map.getStatement("rab_UPDATE");
94     String[] values = new String[]{
95         Integer.toString(raavarebatch.getRaavareId()),
96         Double.toString(raavarebatch.getMaengde()),
97         Integer.toString(raavarebatch.getRbId())
98     };
99     statement = map.insertValuesIntoString(statement, values);
100    System.out.println(statement);
101    Connector.doUpdate(statement);
102
103 }
104
105 }
106

```