

# MySQLViewDAO.java

```

1 package webapplication.datalayer;
2
3 import java.sql.ResultSet;
4
5
6
7
8
9
10
11
12
13
14
15
16 public class MySQLViewDAO implements ViewDAO {
17     SQLMapper map = new SQLMapper();
18
19     public MySQLViewDAO(){
20         try { new Connector(); }
21         catch (InstantiationException e) { e.printStackTrace(); }
22         catch (IllegalAccessException e) { e.printStackTrace(); }
23         catch (ClassNotFoundException e) { e.printStackTrace(); }
24         catch (SQLException e) { e.printStackTrace(); }
25     }
26
27     @Override
28     public List<ViewRaavareNavneDTO> getRaavareNavneList() throws DALException {
29         List<ViewRaavareNavneDTO> list = new ArrayList<ViewRaavareNavneDTO>();
30         ResultSet rs = Connector.doQuery(map.getStatement("view_SELECTALL_raavare_navne"));
31         try
32         {
33             while (rs.next())
34             {
35                 list.add(new ViewRaavareNavneDTO(rs.getInt("pb_id"),
36 rs.getInt("recept_id"), rs.getInt("rb_id"), rs.getInt("raavare_id"),
37 rs.getString("raavare_navn")));
38             }
39         }
40         catch (SQLException e) { throw new DALException(e); }
41         return list;
42     }
43
44     @Override
45     public List<ViewRaavareNavneDTO> getRaavareNavneListPbId(int pbId) throws DALException
46     {
47         List<ViewRaavareNavneDTO> list = new ArrayList<ViewRaavareNavneDTO>();
48         String statement = map.getStatement("view_SELECT_raavare_navne_pb");
49         statement = map.insertValuesIntoString(statement, new
50 String[]{Integer.toString(pbId)});
51
52         ResultSet rs = Connector.doQuery(statement);
53         try
54         {
55             while (rs.next())
56             {
57                 list.add(new ViewRaavareNavneDTO(rs.getInt("pb_id"),
58 rs.getInt("recept_id"), rs.getInt("rb_id"), rs.getInt("raavare_id"),
59 rs.getString("raavare_navn")));
60             }
61         }
62         catch (SQLException e) { throw new DALException(e); }
63         return list;
64     }
65
66     @Override
67     public List<ViewRaavareNavneDTO> getRaavareNavneListRbId(int rbId) throws DALException
68     {
69         List<ViewRaavareNavneDTO> list = new ArrayList<ViewRaavareNavneDTO>();
70         String statement = map.getStatement("view_SELECT_raavare_navne_rb");
71         statement = map.insertValuesIntoString(statement, new
72 String[]{Integer.toString(rbId)});
73         ResultSet rs = Connector.doQuery(statement);
74     }
75 }

```

```

66         try
67         {
68             while (rs.next())
69             {
70                 list.add(new ViewRaavareNavneDTO(rs.getInt("pb_id"),
71 rs.getInt("recept_id"), rs.getInt("rb_id"), rs.getInt("raavare_id"),
72 rs.getString("raavare_navn")));
73             }
74         }
75         catch (SQLException e) { throw new DALEException(e); }
76         return list;
77     }
78     @Override
79     public List<ViewUdskriftDTO> getUdskriftList() throws DALEException {
80         List<ViewUdskriftDTO> list = new ArrayList<ViewUdskriftDTO>();
81         String statement = map.getStatement("view_SELECTALL_udskrift");
82         ResultSet rs = Connector.doQuery(statement);
83
84         try
85         {
86             while (rs.next())
87             {
88                 ProduktBatchKompDTO pbkomp = new ProduktBatchKompDTO (rs.getInt("pb_id"),
89 rs.getInt("rb_id"), rs.getDouble("tara"), rs.getDouble("netto"), rs.getInt("opr_id"));
90                 list.add(new ViewUdskriftDTO(pbkomp, rs.getString("opr_navn"),
91 rs.getInt("raavare_id"), rs.getString("raavare_navn"), rs.getInt("receptId"),
92 rs.getInt("status")));
93             }
94         }
95         catch (SQLException e) { throw new DALEException(e); }
96         return list;
97     }
98     @Override
99     public List<ViewUdskriftDTO> getUdskriftList(int pbId) throws DALEException {
100         List<ViewUdskriftDTO> list = new ArrayList<ViewUdskriftDTO>();
101         String statement = map.getStatement("view_SELECT_udskrift");
102         statement = map.insertValuesIntoString(statement, new
103 String[]{Integer.toString(pbId)});
104         ResultSet rs = Connector.doQuery(statement);
105
106         try
107         {
108             while (rs.next())
109             {
110                 ProduktBatchKompDTO pbkomp = new ProduktBatchKompDTO (rs.getInt("pb_id"),
111 rs.getInt("rb_id"), rs.getDouble("tara"), rs.getDouble("netto"), rs.getInt("opr_id"));
112                 list.add(new ViewUdskriftDTO(pbkomp, rs.getString("opr_navn"),
113 rs.getInt("raavare_id"), rs.getString("raavare_navn"), rs.getInt("recept_id"),
114 rs.getInt("status")));
115             }
116         }
117         catch (SQLException e) { throw new DALEException(e); }
118         return list;
119     }
120 }

```