

# SQLMapper.java

```

1 package webapplication.sqlconnector;
2
3 import java.util.ArrayList;
4
5
6 public class SQLMapper {
7     private List<StringPair> valuePairs = new ArrayList<StringPair>();
8
9     public SQLMapper(){
10         String allStatements =
11             "opr_SELECT                                = SELECT * FROM operatoer WHERE opr_id = ?;\n" +
12             "\r\n" +
13             "pb_SELECT                                = SELECT * FROM produktbatch WHERE pb_id =\n" +
14             "?;\r\n" +
15             "pb_komponent_SELECT                        = SELECT * FROM produktbatchkomponent WHERE\n" +
16             "pb_id = ? AND rb_id = ?;\r\n" +
17             "rec_SELECT                                = SELECT * FROM recept WHERE recept_id = ?;\n" +
18             "\r\n" +
19             "rec_komponent_SELECT                    = SELECT * FROM receptkomponent WHERE\n" +
20             "recept_id = ? AND raavare_id = ?;\r\n" +
21             "rab_SELECT                                = SELECT * FROM raavarebatch WHERE rb_id =\n" +
22             "?;\r\n" +
23             "ra_SELECT                                = SELECT * FROM raavare WHERE raavare_id =\n" +
24             "?;\r\n" +
25             "\r\n" +
26             "opr_SELECT_ALL                            = SELECT * FROM operatoer;\r\n" +
27             "pb_SELECT_ALL                            = SELECT * FROM produktbatch;\r\n" +
28             "pb_komponent_SELECT_ALL                = SELECT * FROM produktbatchkomponent;\r\n" +
29             "\r\n" +
30             "pb_komponent_SELECT_ALL_rec_id = SELECT * FROM produktbatchkomponent WHERE\n" +
31             "rb_id = ?;\r\n" +
32             "rec_SELECT_ALL                            = SELECT * FROM recept;\r\n" +
33             "rec_komponent_SELECT_ALL                = SELECT * FROM receptkomponent;\r\n" +
34             "rec_komponent_SELECT_ALL_rec_id= SELECT * FROM receptkomponent WHERE\n" +
35             "recept_id = ?;\r\n" +
36             "rab_SELECT_ALL                            = SELECT * FROM raavarebatch;\r\n" +
37             "rab_SELECT_ALL_raavare_id                = SELECT * FROM raavarebatch WHERE\n" +
38             "raavare_id = ?;\r\n" +
39             "ra_SELECT_ALL                            = SELECT * FROM raavare;\r\n" +
40             "ro_SELECT_ALL                            = SELECT DISTINCT * FROM role;\r\n" +
41             "\r\n" +
42             "opr_INSERT                                = INSERT INTO operatoer(opr_id, opr_navn,\n" +
43             "ini, password, opr_active) VALUES (?, '?', '?', '?', ?);\r\n" +
44             "pb_INSERT                                = INSERT INTO produktbatch(pb_id, status,\n" +
45             "recept_id) VALUES (?, '?', ?);\r\n" +
46             "pb_komponent_INSERT                    = INSERT INTO produktbatchkomponent(pb_id,\n" +
47             "rb_id, tara, netto, opr_id) VALUES (?, ?, ?, ?, ?);\r\n" +
48             "rec_INSERT                                = INSERT INTO recept(recept_id,\n" +
49             "recept_navn) VALUES (?, '?');\r\n" +
50             "rec_komponent_INSERT                    = INSERT INTO receptkomponent(recept_id,\n" +
51             "raavare_id, nom_netto, tolerance) VALUES (?, ?, ?, ?);\r\n" +
52             "rab_INSERT                                = INSERT INTO raavarebatch(rb_id,\n" +
53             "raavare_id, maengde) VALUES (?, ?, ?);\r\n" +
54             "ra_INSERT                                = INSERT INTO raavare(raavare_id,\n" +
55             "raavare_navn, leverandoer) VALUES (?, '?', '?');\r\n" +
56             "ro_INSERT                                = INSERT INTO role(opr_id, role) VALUES (?,\n" +
57             "'?');\r\n" +
58             "\r\n" +
59             "opr_UPDATE                                = UPDATE operatoer SET opr_navn = '?', ini\n" +
60             "= '?', password = '?', opr_active = ? WHERE opr_id = ?;\r\n" +
61             "pb_UPDATE                                = UPDATE produktbatch SET status = ?,\n" +
62             "recept_id = ? WHERE pb_id = ?;\r\n" +
63             "pb_komponent_UPDATE                    = UPDATE produktbatchkomponent SET tara =

```

# SQLMapper.java

```

?, netto = ?, opr_id = ? WHERE pb_id = ? AND rb_id = ?;\r\n" +
43         "rec_UPDATE                                = UPDATE recept SET recept_navn = '?' WHERE
recept_id = ?;\r\n" +
44         "rec_komponent_UPDATE                        = UPDATE receptkomponent SET nom_netto = ?,
tolerance = ? WHERE recept_id = ? AND raavare_id = ?;\r\n" +
45         "rab_UPDATE                                = UPDATE raavarebatch SET raavare_id = ?,
maengde = ? WHERE rb_id = ?;\r\n" +
46         "ra_UPDATE                                = UPDATE raavare SET raavare_navn = '?',
leverandoer = '?' WHERE raavare_id = ?;\r\n" +
47
48         "ro_DELETE                                = DELETE FROM role WHERE opr_id = ?;\r\n" +
49         "ro_CONCAT_ALL                            = SELECT role.opr_id, GROUP_CONCAT(DISTINCT
role) AS 'roles'" +
50
51         "ro_CONCAT                                = SELECT role.opr_id, GROUP_CONCAT(DISTINCT
role) AS 'roles'" +
52
53         " FROM role WHERE opr_id = ?" +
54         " GROUP BY opr_id;\r\n" +
55
56         "view_create_mad                            = CREATE VIEW mad AS SELECT rk.recept_id,
rk.raavare_id, re.recept_navn, ra.raavare_navn, ra.leverandoer FROM receptkomponent rk
NATURAL JOIN recept re NATURAL JOIN raavare ra;\r\n" +
57
58         "view_create_vejning                        = CREATE VIEW vejning AS SELECT opr.opr_id,
opr.opr_navn, pbk.tara, pbk.netto, ra.raavare_id, ra.raavare_navn, rab.maengde FROM
operatoer opr NATURAL JOIN produktbatchkomponent pbk NATURAL JOIN raavare ra NATURAL JOIN
raavarebatch rab;\r\n" +
59
60         "view_SELECT_mad                            = SELECT * FROM mad WHERE recept_id = ? AND
raavare_id = ?;\r\n" +
61         "view_SELECT_vejning                        = SELECT * FROM vejning WHERE opr_id = ?;
\r\n" +
62         "view_SELECT_raavare_navne_pb                = SELECT * FROM raavare_navne WHERE pb_id =
?;\r\n" +
63         "view_SELECT_raavare_navne_rb                = SELECT * FROM raavare_navne WHERE rb_id =
?;\r\n" +
64         "view_SELECT_udskrift                        = SELECT * FROM udskrift WHERE pb_id = ?;
\r\n" +
65
66         "view_drop_mad                            = DROP VIEW mad;\r\n" +
67         "view_drop_vejning                        = DROP VIEW vejning; \r\n" +
68
69         "view_SELECTALL_mad                        = SELECT * FROM mad;\r\n" +
70         "view_SELECTALL_vejning                    = SELECT * FROM vejning;\r\n" +
71         "view_SELECTALL_raavare_navne                = SELECT * FROM raavare_navne;\r\n" +
72         "view_SELECTALL_udskrift                    = SELECT * FROM udskrift;\r\n" +
73         "opr_specific_name                            = SELECT opr_navn FROM operatoer WHERE
opr_id = ?;\r\n"
74
75         ;
76         String[] statementsArray = allStatements.split("\r\n");
77         String key = "";
78         String value = "";
79
80         for(String statement : statementsArray){
81             if(statement.indexOf(" ") > 0){
82                 key = statement.substring(0, statement.indexOf("=")-1);
83                 key = key.replaceAll(" ", "");
84                 key = key.replaceAll("\t", "");
85             }
86             if(statement.indexOf("=") > 0){
87                 value = statement.substring(statement.indexOf("=")+2);

```

# SQLMapper.java

```

87         }
88         valuePairs.add(new StringPair(key, value));
89     }
90
91     /*for(StringPair pair : valuePairs){
92         System.out.println(pair.getKey() + ":" + pair.getValue());
93     }*/
94
95 }
96
97 public String getStatement(String str){
98     for(StringPair pair : valuePairs){
99         if(pair.getKey().equals(str)){
100             return pair.getValue();
101         }
102     }
103     return "";
104
105     /*
106     Properties props = new Properties();
107     try {
108         File file = new File("SQL.txt");
109         FileInputStream in = new FileInputStream(file);
110         props.load(in);
111         String res = props.getProperty(str);
112         in.close();
113         return res;
114     } catch (FileNotFoundException e){
115         throw new IllegalStateException("SQL file does not exist");
116     } catch (IOException e) {
117         throw new IllegalStateException("Unable to load properties");
118     }*/
119 }
120
121 public String insertValuesIntoString(String statement, String[] values){
122     for(int i=0; i<values.length; i++){
123         statement = statement.replaceFirst("[?]", values[i]);
124     }
125     //System.out.println(statement);
126     return statement;
127 }
128
129 private class StringPair{
130     String key;
131     String value;
132
133     StringPair(String key, String value){
134         this.key = key;
135         this.value = value;
136     }
137
138     String getKey(){ return key; }
139
140     String getValue(){ return value; }
141 }
142
143 }
144

```