

ItemDAO.java

```

1 package model.dao;
2
3 import java.sql.ResultSet;
4
10
11 public class ItemDAO extends DAO<Item> {
12     private static Map<String, String[]> COLUMNS;
13     private static List<String[]> searchColumns;
14
15     public ItemDAO() throws Exception {
16         super("ITEM");
17         COLUMNS = new HashMap<String, String[]>();
18         searchColumns = new ArrayList<String[]>();
19
20         COLUMNS.put("qty", new String[]{"QTY", "num"});
21         COLUMNS.put("name", new String[]{"NAME", "str"});
22         COLUMNS.put("price", new String[]{"PRICE", "num"});
23         COLUMNS.put("id", new String[]{"NUMBER", "str"});
24         COLUMNS.put("num", new String[]{"NUMBER", "str"});
25         COLUMNS.put("cid", new String[]{"CATID", "num"});
26         COLUMNS.put("unit", new String[]{"UNIT", "str"});
27
28         searchColumns.add(COLUMNS.get("name"));
29         searchColumns.add(COLUMNS.get("num"));
30     }
31
32     @Override
33     public String convertToColumnName(String by) {
34         String result = COLUMNS.get("cid")[0];
35
36         if (COLUMNS.containsKey(by) ||
37 COLUMNS.containsKey(by.toLowerCase()))
38         {
39             result = COLUMNS.get(by)[0];
40         }
41         else if (by.equals(COLUMNS.get("name")[0]))
42         {
43             result = COLUMNS.get("name")[0];
44         }
45         else if (by.toLowerCase().equals("price") ||
46 by.equals(COLUMNS.get("price")[0]))
47         {
48             result = COLUMNS.get("price")[0];
49         }
50         else if (by.toLowerCase().equals("quantity") ||

```

ItemDAO.java

```
by.equals(COLUMNS.get("qty")[0]))
49     {
50         result = COLUMNS.get("qty")[0];
51     }
52     else if (by.toLowerCase().equals("number") ||
by.toLowerCase().equals("num") || by.equals(COLUMNS.get("id")[0]))
53     {
54         result = COLUMNS.get("id")[0];
55     }
56     else if (by.toLowerCase().equals(COLUMNS.get("cid")[0]))
57     {
58         result = COLUMNS.get("cid")[0];
59     }
60     else if (by.toLowerCase().equals(COLUMNS.get("unit")[0]))
61     {
62         result = COLUMNS.get("unit")[0];
63     }
64     return result;
65 }
66
67 @Override
68 public List<String[]> getSearchColumns() {
69     return searchColumns;
70 }
71
72 @Override
73 public Item createBean(ResultSet r) throws Exception {
74     Item item = new Item();
75     item.setName(getName(r));
76     item.setQuantity(getQuantity(r));
77     item.setPrice(getPrice(r));
78     item.setNumber(getNumber(r));
79     item.setCatID(getCatID(r));
80     item.setUnit(getUnit(r));
81     return item;
82 }
83
84 @Override
85 public Map<String, String[]> getColumns() {
86     return COLUMNS;
87 }
88
89 public String getId(ResultSet r) throws Exception {
90     return r.getString(COLUMNS.get("id")[0]);
```

ItemDAO.java

```
91     }
92
93     public String getName(ResultSet r) throws Exception {
94         return r.getString(COLUMNS.get("name")[0]);
95     }
96
97     public double getPrice(ResultSet r) throws Exception {
98         return r.getDouble(COLUMNS.get("price")[0]);
99     }
100
101     public int getQuantity(ResultSet r) throws Exception {
102         return r.getInt(COLUMNS.get("qty")[0]);
103     }
104
105     public String getNumber(ResultSet r) throws Exception {
106         return r.getString(COLUMNS.get("num")[0]);
107     }
108
109     public int getCatID(ResultSet r) throws Exception {
110         return r.getInt(COLUMNS.get("cid")[0]);
111     }
112
113     public String getUnit(ResultSet r) throws Exception {
114         return r.getString(COLUMNS.get("unit")[0]);
115     }
116
117     public static void main(String[] args) throws Exception {
118         ItemDAO itemDao = new ItemDAO();
119         System.out.println(itemDao.getAll().toString());
120     }
121 }
122
```