

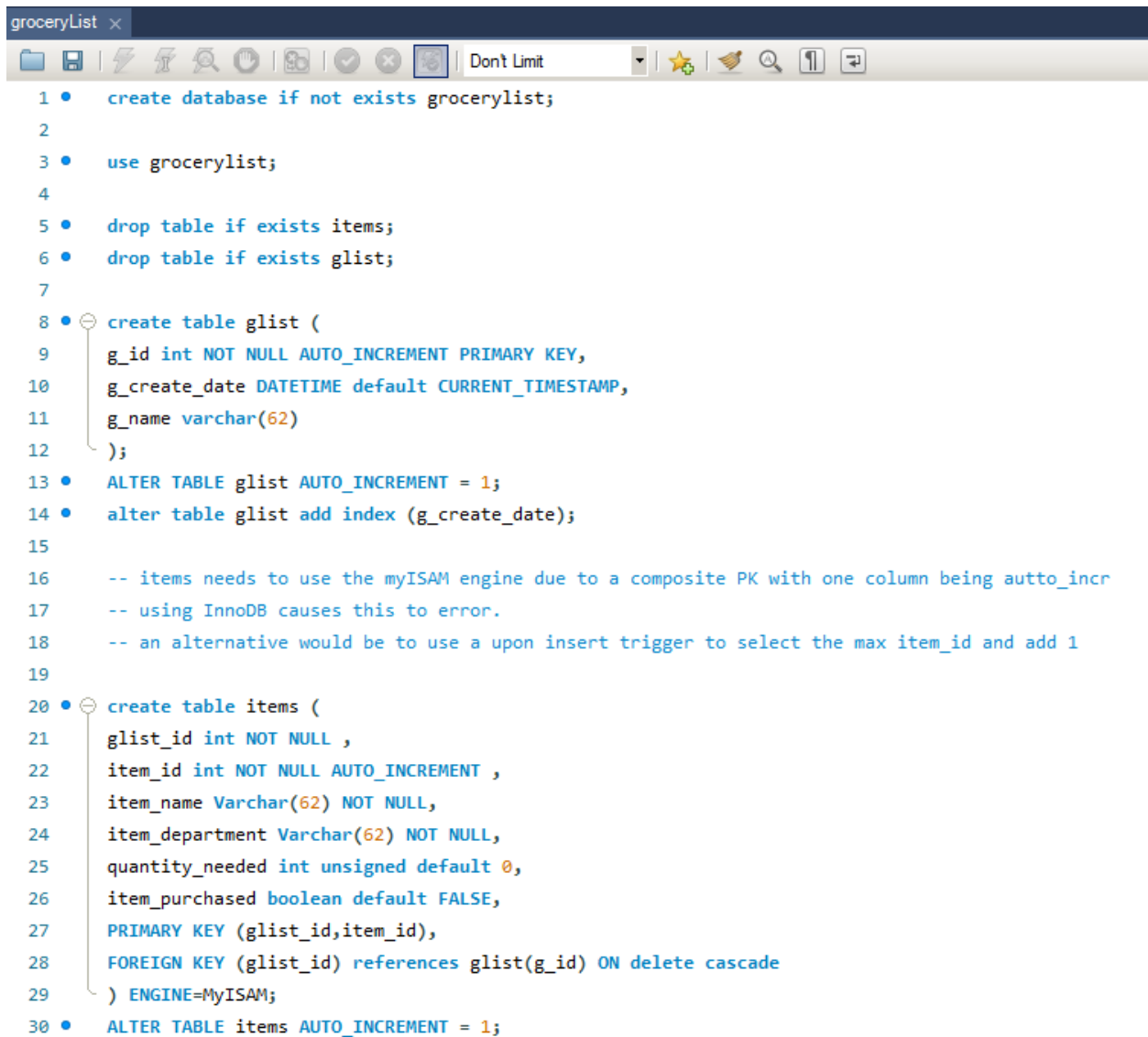
Relational Databases with MySQL Week 5 Coding Assignment

Student Note: Only a single table , groceryList, was implement to complete the assignment

The items table and code is included as a learning exercise to be done later.

Screenshots of Code:

SQL commands



```
1 • create database if not exists grocerylist;
2
3 • use grocerylist;
4
5 • drop table if exists items;
6 • drop table if exists glist;
7
8 • create table glist (
9     g_id int NOT NULL AUTO_INCREMENT PRIMARY KEY,
10    g_create_date DATETIME default CURRENT_TIMESTAMP,
11    g_name varchar(62)
12 );
13 • ALTER TABLE glist AUTO_INCREMENT = 1;
14 • alter table glist add index (g_create_date);
15
16 -- items needs to use the myISAM engine due to a composite PK with one column being auto_incr
17 -- using InnoDB causes this to error.
18 -- an alternative would be to use a upon insert trigger to select the max item_id and add 1
19
20 • create table items (
21     glist_id int NOT NULL ,
22     item_id int NOT NULL AUTO_INCREMENT ,
23     item_name Varchar(62) NOT NULL,
24     item_department Varchar(62) NOT NULL,
25     quantity_needed int unsigned default 0,
26     item_purchased boolean default FALSE,
27     PRIMARY KEY (glist_id,item_id),
28     FOREIGN KEY (glist_id) references glist(g_id) ON delete cascade
29 ) ENGINE=MyISAM;
30 • ALTER TABLE items AUTO_INCREMENT = 1;
```

DBConnection

```
DBConnection.java X groceryListApp.java groceryList.java groceryItem.java GLDao.java Menus.java
1 package groceryListDao;
2
3 import java.sql.Connection;
4
5
6
7 public class DBConnection {
8     public final static String connectionString = "jdbc:mysql://localhost:3306/grocerylist";
9     private static Connection conn;
10    private static DBConnection instance;
11
12    private DBConnection (Connection conn) {
13        this.conn = conn;
14    }
15
16    public static Connection getConnection() {
17        if ( instance == null) {
18            try {
19                conn = DriverManager.getConnection(connectionString,"root","root");
20                instance = new DBConnection(conn);
21                System.out.println("Connected to groceryList");
22            }
23            catch (SQLException e) {
24                System.out.println("Error connecting to DB groceryList");
25                e.printStackTrace();
26            }
27        }
28        return DBConnection.conn;
29    }
30 }
31 }
```

groceryListApp

```
DBConnection.java groceryListApp.java X groceryList.java groceryItem.java GLDao.java Menus.java
1 package codingAssignment;
2
3 public class groceryListApp {
4
5     public static void main(String[] args) {
6
7         // groceryList gl = new groceryList();
8         Menus m = new Menus();
9         m.startMenu();
10    }
11 }
```

Menus (part1)

```
DBConnection.java  groceryListApp.java  groceryList.java  groceryItem.java  GLDao.java  Menus.java X
1  package codingAssignment;
2
3  import java.sql.SQLException;
4
5
6
7
8
9
10 public class Menus {
11     private List<String> glOptions = Arrays.asList(
12         "Display Grocery Lists",
13         "Choose a Grocery List",
14         "Create a Grocery List",
15         "Change Grocery List Name",
16         "Delete a Grocery List" );
17     // private GLDao glDao = new GLDao();
18     private groceryList gl;
19     Scanner kb;
20     String selection;
21     int current_gID;
22
23     public Menus() {
24         gl = new groceryList();
25         current_gID = 0;
26         selection = "";
27         kb = new Scanner(System.in);
28     }
29
30     public void startMenu () {
31         try {
32             do {
33                 displayGroceryListsMenu();
34                 selection = kb.nextLine();
35                 if (selection.equals("1")) {
36                     displayGroceryLists();
37                 } else if (selection.equals("2")) {
38                     current_gID = selectGroceryList();
39                     //groceryListItemsMenu(); -- to be implemented later as a learning exercise
40                     // -- submenus and filling items into each grocery list
41                 } else if (selection.equals("3")) {
42                     createGroceryList();
43                 } else if (selection.equals("4")) {
44                     changeGroceryList();
45                 } else if (selection.equals("5")) {
46                     deleteGroceryList();
47                 } else if (selection.equals("-1")) { break; }
48                 else {
49                     System.out.println("Please Enter a valid selection ..... -1 to exit");
50                 }
51
52                 System.out.println("Press enter to continue .....");
53                 kb.nextLine();
54             } while (!selection.equals("-1"));
55
56         } catch (SQLException e) {
57             e.printStackTrace();
58         }
59     }
60 }
```

Menus (part2)

```
DBConnection.java  groceryListApp.java  groceryList.java  groceryItem.java  GLDao.java  *Menus.java X

61 private int selectGroceryList() throws SQLException {
62     int gSelect = 0;
63     boolean validSelection = false;
64     do {
65         gSelect = 0;
66         System.out.println("Choose a Grocery List -----");
67         int glsize = displayGroceryLists();
68         if (glsize > 0) {
69             gSelect = Integer.parseInt(kb.nextLine()); //Really should surround this with NumberFormat Exception
70             if (gSelect > 0 && gSelect <= glsize) validSelection = true;
71         } else validSelection = true;
72     } while (!validSelection);
73     return gSelect;
74 }
75
76 private int displayGroceryLists() throws SQLException {
77     int glsize = gl.displayGroceryLists();
78     return glsize;
79 }
80
81 private void displayGroceryListsMenu() {
82     System.out.println("--- Grocery Lists ---");
83     for (int i = 0; i < glOptions.size(); i++) {
84         System.out.println(i+1 + ". " + glOptions.get(i));
85     }
86 }
87
88 private void createGroceryList() throws SQLException {
89     System.out.println("Enter a Grocery List Name ---");
90     String gName = kb.nextLine();
91     try {
92         gl.addGroceryList(gName);
93         System.out.println("added grocery List : " + gName);
94     } catch (SQLException e) {
95         throw(e);
96     }
97 }
98
99 private void changeGroceryList() throws SQLException {
100     displayGroceryLists();
101     System.out.println("Enter a Grocery List ID to change the name of ---");
102     int gID = Integer.parseInt(kb.nextLine());
103     System.out.println("Enter a New Name -----");
104     String gName = kb.nextLine();
105     gl.changeGroceryList(gID,gName);
106 }
107
108 private void deleteGroceryList() throws SQLException {
109     displayGroceryLists();
110     System.out.println("Enter a Grocery List ID to delete ---");
111     int gID = kb.nextInt();
112     gl.deleteGroceryList(gID);
113 }
114 }
```

groceryList

```
DBConnection.java  groceryListApp.java  *groceryList.java  X  groceryItem.java  GLDao.java  *Menus.java

1 package codingAssignment;
2
3 import java.sql.SQLException;
4 import java.util.Map;
5 import groceryListDao.GLDao;
6 import groceryListDao.GLDao.glcolumns;
7
8 public class groceryList {
9
10     private int gID;
11     private Map<Integer,groceryListDao.GLDao.glcolumns> glists;
12
13     private final static GLDao gld = new GLDao();
14
15     //
16     // -- to be used later as a learning exercise
17     // private Map<Integer,groceryItem> items;
18     // private int itemID;
19     //
20     public groceryList() {
21         gID = 0;
22     }
23
24     private int getgID() {
25         return gID;
26     }
27
28     private void setgID(int gID) {
29         this.gID = gID;
30     }
31
32     public int displayGroceryLists() throws SQLException {
33         int glsize = 0;
34         Map<Integer,glcolumns> glMap = gld.getGroceryLists();
35         for (Map.Entry<Integer, glcolumns> gl : glMap.entrySet()) {
36             System.out.println(gl.getKey() + " : " + gl.getValue().toString() + "\n");
37             glsize++;
38         }
39         return glsize;
40     }
41
42     public void addGroceryList(String gname) throws SQLException {
43         //-- for later TreeMap<Integer,groceryListDao.GLDao.glcolumns> glists = new TreeMap<Integer,groceryListDao.GLDao.glcolumns>();
44         // -- for later INSERT INTO THE GROCERY LIST AND RETRIEVE THE DB Values of id and date/time
45         gld.newGList(gname);
46         return;
47     }
48
49     public void changeGroceryList(int glID, String gName) throws SQLException {
50         gld.changeGLName(glID,gName);
51     }
52
53
54     public void deleteGroceryList(int glID) throws SQLException {
55         gld.deleteGL(glID);
56     }
57 }
```

GLDao (part 1)

```
DBConnection.java  groceryListApp.java  *groceryList.java  groceryItem.java  GLDao.java  *Menus.java

1 package groceryListDao;
2
3 import java.sql.Connection;
4 import java.sql.Date;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.time.LocalDate;
9 import java.util.Map;
10 import java.util.TreeMap;
11
12 public class GLDao {
13
14     private Connection conn;
15     private final String GET_GLIST_QUERY = "SELECT * FROM glist";
16     private LocalDate gDate;
17     private String INSERT_GLIST = "INSERT INTO glist ( g_name ) VALUES (?)";
18     private String DELETE_GLIST = "DELETE FROM glist where g_id = ?";
19     private String UPDATE_GLIST = "UPDATE glist SET g_name = ? WHERE g_id = ?";
20     private final String QUERY_FOR_NEWEST_GID_STRING = "SELECT MAX(g_id) FROM glist";
21
22
23     public class glcolumns {
24         LocalDate ldate;
25         String gname;
26
27         public String toString() {
28             String str = ldate.toString() + " " + gname;
29             return str;
30         }
31     }
32
33     public GLDao () {
34         conn = DBConnection.getConnection();
35     }
36
37     public Map<Integer,glcolumns> getGroceryLists() throws SQLException {
38         ResultSet rs = conn.prepareStatement(GET_GLIST_QUERY).executeQuery();
39         Map<Integer,glcolumns> glMap = new TreeMap<>();
40         while (rs.next()) {
41             glcolumns glc = new glcolumns();
42             glc.ldate = rs.getDate(2).toLocalDate();
43             glc.gname = rs.getString(3);
44             glMap.put ( ((Integer)rs.getInt(1)) , glc);
45         }
46         return glMap;
47     }
48
49     public int displayGroceryLists() throws SQLException {
50         int glsize = 0;
51         Map<Integer,glcolumns> glMap = getGroceryLists();
52         for (Map.Entry<Integer, glcolumns> gl : glMap.entrySet()) {
53             System.out.println(gl.getKey() + " : " + gl.getValue().toString() + "\n");
54             glsize++;
55         }
56         return glsize;
57     }
58 }
```

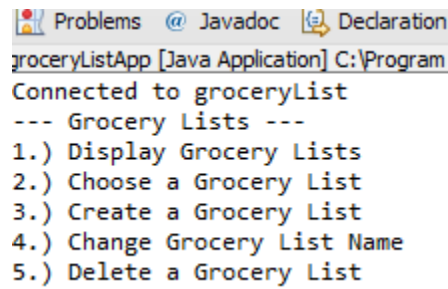
GLDao (part 2)

```
DBConnection.java  groceryListApp.java  groceryList.java  groceryItem.java  GLDao.java  X  Menus.java
58
59 public void newList(String gname) throws SQLException {
60     PreparedStatement ps = conn.prepareStatement(INSERT_GLIST);
61     ps.setString(1, gname);
62     ps.executeUpdate();
63
64 }
65
66 public void deleteGL(int glID) throws SQLException {
67     PreparedStatement ps = conn.prepareStatement(DELETE_GLIST);
68     ps.setInt(1, glID);
69     ps.executeUpdate();
70 }
71
72 public void changeGLName(int glID, String gName) throws SQLException {
73     PreparedStatement ps = conn.prepareStatement(UPDATE_GLIST);
74     ps.setString(1, gName);
75     ps.setInt(2, glID);
76     ps.executeUpdate();
77 }
78 |
79 }
80
```

groceryItem //Stub not currently implemented – kept for practice later

```
DBConnection.java  groceryListApp.java  groceryList.java  *groceryItem.java  GLDao.java  Menus.java
1  package codingAssignment;
2
3  public class groceryItem {
4
5      private int glist_id;
6      private String name;
7      private String department;
8      private int quantity;
9      private boolean purchased;
10
11     public groceryItem(int glist_id, String name, String department, int quantity, boolean purchased) {
12         this.setGlist_id(glist_id);
13         this.setName(name);
14         this.setDepartment(department);
15         this.setQuantity(quantity);
16         this.setPurchased(purchased);
17     }
18
19     public int getGlist_id() {
20         return glist_id;
21     }
22     public void setGlist_id(int glist_id) {
23         this.glist_id = glist_id;
24     }
25
26     public String getName() {
27         return name;
28     }
29     public void setName(String name) {
30         this.name = name;
31     }
32
33     public String getDepartment() {
34         return department;
35     }
36     public void setDepartment(String department) {
37         this.department = department;
38     }
39
40     public int getQuantity() {
41         return quantity;
42     }
43     public void setQuantity(int quantity) {
44         this.quantity = quantity;
45     }
46
47     public boolean isPurchased() {
48         return purchased;
49     }
50     public void setPurchased(boolean purchased) {
51         this.purchased = purchased;
52     }
53
54 }
```

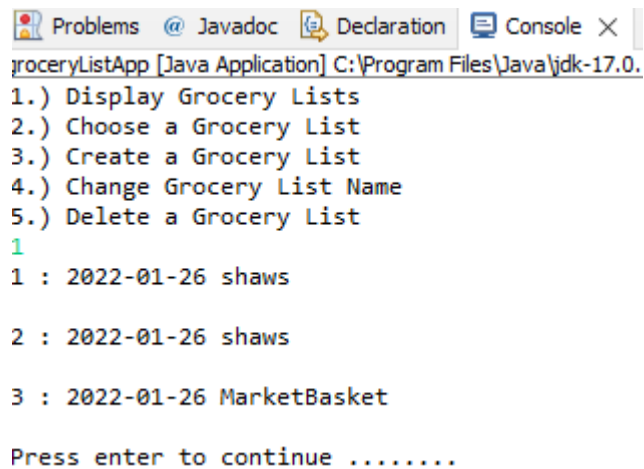

Screenshots of Running Application:



The screenshot shows an IDE window with tabs for 'Problems', 'Javadoc', and 'Declaration'. The console output for 'groceryListApp [Java Application] C:\Program' displays the following menu:

```
Connected to groceryList
--- Grocery Lists ---
1.) Display Grocery Lists
2.) Choose a Grocery List
3.) Create a Grocery List
4.) Change Grocery List Name
5.) Delete a Grocery List
```

Display



The screenshot shows the same IDE window with an additional 'Console' tab. The console output for 'groceryListApp [Java Application] C:\Program Files\Java\jdk-17.0.' shows the application's response to the 'Display' option:

```
1.) Display Grocery Lists
2.) Choose a Grocery List
3.) Create a Grocery List
4.) Change Grocery List Name
5.) Delete a Grocery List
1
1 : 2022-01-26 shaws

2 : 2022-01-26 shaws

3 : 2022-01-26 MarketBasket

Press enter to continue .....
```

Create

```
Problems @ Javadoc Declaration Console X
groceryListApp [Java Application] C:\Program Files\Java\jdk-17.

--- Grocery Lists ---
1.) Display Grocery Lists
2.) Choose a Grocery List
3.) Create a Grocery List
4.) Change Grocery List Name
5.) Delete a Grocery List
3
Enter a Grocery List Name ---
Walmart
added grocery List : Walmart
Press enter to continue .....

--- Grocery Lists ---
1.) Display Grocery Lists
2.) Choose a Grocery List
3.) Create a Grocery List
4.) Change Grocery List Name
5.) Delete a Grocery List
1
1 : 2022-01-26 shaws

2 : 2022-01-26 shaws

3 : 2022-01-26 MarketBasket

7 : 2022-01-27 Walmart

Press enter to continue .....
```

Update a grocery list name

```
Problems @ Javadoc Declaration Console X
groceryListApp [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\j
--- Grocery Lists ---
1.) Display Grocery Lists
2.) Choose a Grocery List
3.) Create a Grocery List
4.) Change Grocery List Name
5.) Delete a Grocery List
4
1 : 2022-01-26 shaws
2 : 2022-01-26 shaws
3 : 2022-01-26 MarketBasket
7 : 2022-01-27 Walmart
Enter a Grocery List ID to change the name of ---
7
Enter a New Name -----
Stop and Shop
Press enter to continue .....

--- Grocery Lists ---
1.) Display Grocery Lists
2.) Choose a Grocery List
3.) Create a Grocery List
4.) Change Grocery List Name
5.) Delete a Grocery List
1
1 : 2022-01-26 shaws
2 : 2022-01-26 shaws
3 : 2022-01-26 MarketBasket
7 : 2022-01-27 Stop and Shop
Press enter to continue .....
```

Delete a Grocery List

```
--- Grocery Lists ---
1.) Display Grocery Lists
2.) Choose a Grocery List
3.) Create a Grocery List
4.) Change Grocery List Name
5.) Delete a Grocery List
5
1 : 2022-01-26 shaws

2 : 2022-01-26 shaws

3 : 2022-01-26 MarketBasket

7 : 2022-01-27 Stop and Shop

Enter a Grocery List ID to delete ---
7
Press enter to continue .....
```

```
--- Grocery Lists ---
1.) Display Grocery Lists
2.) Choose a Grocery List
3.) Create a Grocery List
4.) Change Grocery List Name
5.) Delete a Grocery List
1
1 : 2022-01-26 shaws

2 : 2022-01-26 shaws

3 : 2022-01-26 MarketBasket

Press enter to continue .....
```

URL to GitHub Repository:

<https://github.com/david2joh/sqlWeek4.git>