



CS623

GROUP PROJECT

MySQL

By Dac Vu Ho: <https://github.com/dacvuho241>
Wildenslo Osias: <https://github.com/Wildenslo55>

Threefold Solution



Schemas

- Schemas:

- Product*(#*prod*, *pname*, *price*)
- Depot*(#*dep*, *addr*, *volume*)
- Stock*(#*prod*, #*dep*, *quantity*)

- Instances:

Product

#prod	pname	price
p1	tape	2.5
p2	tv	250
p3	vr	80

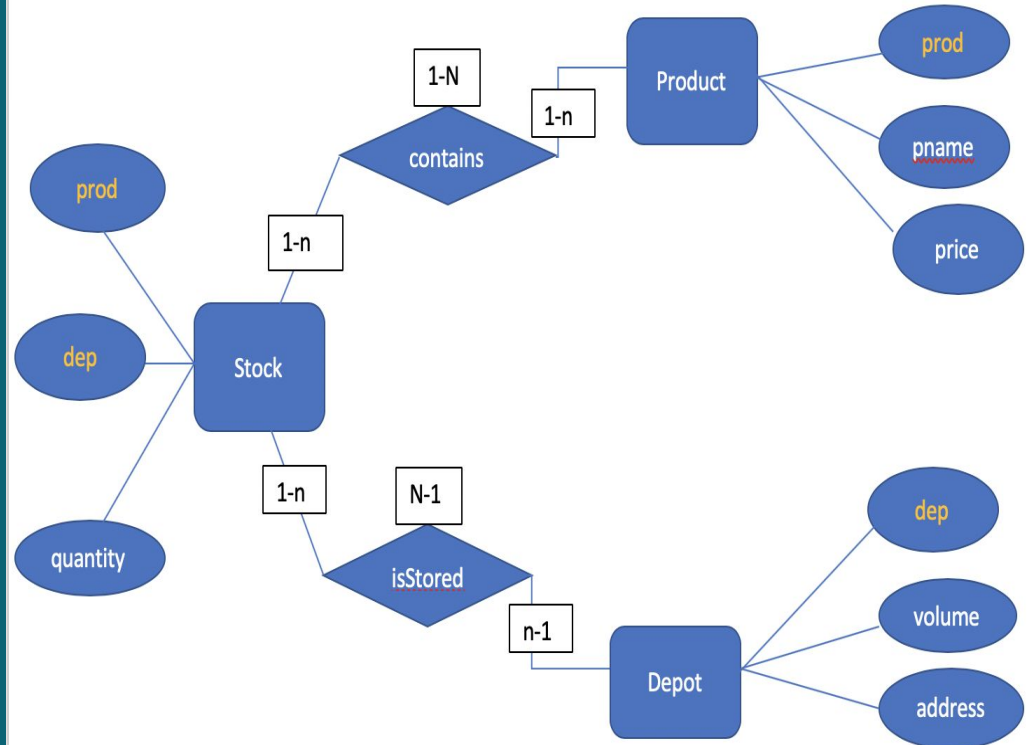
Depot

#dep	addr	volume
d1	New York	9000
d2	Syracuse	6000
d4	New York	2000

Stock

#prod	#dep	quantity
p1	d1	1000
p1	d2	-100
p1	d4	1200
p3	d1	3000
p3	d4	2000
p2	d4	1500
p2	d1	-400
p2	d2	2000

ERD



Java Program (Part I)

```
1 package Team4;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.SQLException;
5 import java.sql.Statement;
6 /**
7  * @author Dac Vu Ho
8  */
9 public class GroupProject {
10
11     public static void main(String args[]) throws SQLException, ClassNotFoundException {
12
13         // Load the MySQL driver
14         Class.forName("com.mysql.cj.jdbc.Driver");
15
16         // Connect to the database
17         Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/groupproject");
18
19         // For atomicity
20         conn.setAutoCommit(false);
21
22         // For isolation
23         conn.setTransactionIsolation(Connection.TRANSACTION_SERIALIZABLE);
24
25         Statement stmt1 = null;
```

```
26     try {
27         // create statement object
28         stmt1 = conn.createStatement();
29
30         // Either the 2 following inserts are executed, or none of them are. This is atomicity.
31         //The depot d1 changes its name to dd1 in Depot
32         stmt1.executeUpdate("UPDATE `groupproject`.`depot` SET `dep` = 'dd1' WHERE (`dep` = 'd1')");
33         //The depot d1 changes its name to dd1 in Stock
34         stmt1.executeUpdate("UPDATE `groupproject`.`stock` SET `dep` = 'dd1' WHERE (`dep` = 'd1')");
35     } catch (SQLException e) {
36         System.out.println("catch Exception");
37         // For atomicity
38         conn.rollback();
39         stmt1.close();
40         conn.close();
41         return;
42     } // main
43     conn.commit();
44     stmt1.close();
45     conn.close();
46 }
```

Java Program (PartII)

Loading Driver and Connecting to MySQL Database

```
// Load the MySQL driver
Class.forName("com.mysql.cj.jdbc.Driver");

// Connect to the database
Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/groupproject",
```

Java Program (Part III)

Implementing ACID Properties

Atomicity:

```
conn.setAutoCommit(false);
```

```
conn.rollback();  
stmt1.close();  
conn.close();
```

Consistency:

```
stmt1.executeUpdate("UPDATE `groupproject`.`depot` SET `dep` = 'dd1' WHERE (`dep` = 'd1')");  
//The depot d1 changes its name to dd1 in Stock  
stmt1.executeUpdate("UPDATE `groupproject`.`stock` SET `dep` = 'dd1' WHERE (`dep` = 'd1')");
```

Isolation:

```
conn.setTransactionIsolation(Connection.TRANSACTION_SERIALIZABLE);
```

Durability:

```
try {  
} catch (SQLException e) {  
}  
conn.commit();  
stmt1.close();  
conn.close();
```

Transaction Execution (Part I)

The screenshot displays a database management tool interface with three main panes. The left pane shows the 'groupproject' schema with tables 'depot' and 'stock'. The middle pane shows a query 'SELECT * FROM groupproject.depot'. The right pane shows the 'Result Grid' with a table of depot data. A green arrow points from the 'depot' table in the left pane to the 'Result Grid' in the right pane.

Left Pane (Schemas):

- groupproject
 - Tables
 - depot
 - stock
 - Columns
 - Indexes
 - Foreign Keys
 - Triggers
 - Views
 - Stored Procedures
 - Functions
- sakila
- student
- sys
- world

Middle Pane (Query):

```
SELECT * FROM groupproject.depot
```

Right Pane (Result Grid):

dep	addr	volume
d2	Syracuse	6000
d4	New York	2000
dd1	New York	9000
NULL	NULL	NULL

Transaction Execution (PartII)

The screenshot displays two instances of the SQL Server Enterprise Manager interface, both showing the 'groupproject' database. The left instance shows the 'stock' table with the following data:

prod	dep	quantity
p1	d1	1000
p1	d2	-100
p1	d4	1200
p2	d1	-400
p2	d2	2000
p2	d4	1500
p3	d1	3000
p3	d4	2000
NULL	NULL	NULL

The right instance shows the 'depot' table with the following data:

prod	dep	quantity
p1	d2	-100
p1	d4	1200
p1	dd1	1000
p2	d2	2000
p2	d4	1500
p2	dd1	-400
p3	d4	2000
p3	dd1	3000
NULL	NULL	NULL

Arrows indicate data flow from the 'stock' table in the left instance to the 'depot' table in the right instance. Specifically, the 'prod' and 'dep' columns in the 'depot' table correspond to the 'prod' and 'dep' columns in the 'stock' table. The 'quantity' column in the 'depot' table is the result of a transaction operation on the 'stock' table.

Thank You! Merci!

