

## Middleware for Heterogeneous and Distributed Information Systems – Exercise Sheet 2

Wednesday, November 5, 2008 – 10:00 to 11:30 – Room 48-379

### SQLJ Program Preparation

SQLJ is a standard for embedding SQL in the Java programming language. Unlike JDBC, which is a call-level interface (CLI), SQLJ is a language extension. The compilation of SQLJ source code involves a series of steps<sup>1</sup>.

1. Describe the steps required to prepare an SQLJ program for execution. For each step, name the required utility program and the input and output files.
2. Comment on the following statement: “SQLJ combines the advantages of embedded SQL with portability and vendor-independence”

### JDBC and SQLJ

```
#sql [ctx] iter = {  
    SELECT LASTNAME, FIRSTNAME  
    FROM STUDENT  
    WHERE MAJOR = 'Computer Science'  
    ORDER BY FIRSTNAME, LASTNAME };  
  
String lastname = null;  
String firstname = null;  
  
while (true) {  
    #sql { FETCH :iter INTO :lastname , :firstname };  
    if (iter.endFetch())  
        break;  
    System.out.println(firstname + " " + lastname);  
}
```

**Figure 1: Sample SQLJ program**

1. Rewrite the program shown in Figure 1 and make use of the JDBC call-level interface instead of SQLJ.

---

<sup>1</sup> See IBM Redbook DB2 for z/OS and OS/390: Ready for Java chapters 9, 10, and 11 available at <http://www.redbooks.ibm.com/abstracts/sg246435.html>

2. At what point in time will the following errors be encountered in either version of the program.
  - a. Syntax error in the SQL statement (e.g. SELCET instead of SELECT)
  - b. Misspelled table name
  - c. Data type mismatch in the FETCH statement

### **X/Open DTP**

The X/Open Distributed Transaction Processing (DTP) model allows application programs to share resources provided by multiple resource managers and to coordinate their work in global transactions using the two phase commit protocol.

1. The X/Open DTP model distinguishes three functional components involved in distributed transaction processing, namely the Application Program (AP), the Transaction Manager (TM), and Resource Managers (RMs). Explain the role of each of these components for distributed transaction processing.
2. The functional components interact by means of standardized interfaces. Name these interfaces and explain their functionality.
3. The X/Open DTP model has been generalized to support transactions in *distributed environments*. That is, multiple TM domains may be involved in a global transaction. What challenges arise in distributed environments as opposed to local environments? What additional component was introduced to meet these challenges? Which of the interfaces required modifications and how do these modifications look like?
4. Use a sequence diagram to sketch the interaction between the functional components during transaction processing in a distributed environment. Assume that one local RM and one remote RM are involved and the transaction completes successfully.