Hochschule Fulda
University of Applied Sciences

FACHBEREICH
ANGEWANDTE
INFORMATIK

Prof. Dr. Prof. h.c. Peter Peinl

Verteilte Datenbanken
Distributed Databases

Assignment 1

The subject of this assignment is to add a few entities (objects, rows) per student to the flight database. Please use your favorite software development environment to implement a Java program that accesses our Oracle installation (oralv9a) as you did last week. To complete this assignment you have to extend the small program that you made operational in the preceding week.

The flight database is available for update using the userid PROJB01 and password ASSIGN1.

In order not to impede each other you are asked to observe the following rules :

- Set the autocommit mode to "false" immediately after connecting to the database.
- Set the isolation level of the transaction to "serializable" immediately after connecting to the database.
- Catch all SQLExceptions and do a rollback before leaving the exception.
- Only commit your transaction after all updates and inserts have been successful.

The task consists of several steps that have to be implemented in the following order:

1. You are to add yourself as a passenger to the database.

2. You are to store two airports that are not yet contained in the database. The data (names etc.) should be real not fictitious. You may look up the data in the Internet.

3. You are to store one airline that are not yet contained in the database. The data (names etc.) should be real not fictitious. You may look up the data in the Internet.

4. Add flights to and from both "your" airports operated by "your" airline to Frankfurt (FRA). Hint: You need to insert more than one row to specify all information about a flight.

5. Make a booking from "your" first airport via Frankfurt to "your" second airport. The day of the booking is today plus 30 days (in SQL : SYSDATE + 30).

6. Make a booking from "your" first airport via Frankfurt to New York (JFK). The day of the booking is today plus 40 days

7. Write a query that searches for all flights (nonstop or one stop and transfer) from an arbitrary airport X to an arbitrary airport Y and print the flights/legs.

In order to succeed we will coordinate names of new airports and airlines at the beginning of the lecture!