Databases
HFT Stuttgart

Finalizing Your Database Project

Please chose a relational database system to implement your database with.

Create your database with the required relational tables and insert some sample data. Include referential integrity constraints and discuss physical design options. Implement interesting transactions and triggers as explained in the sample document below.

Extend your project with additional functionality, accessing it from a Graphical User Interface. You can use Java and JDBC or PHP or Hibernate or whatever you prefer. The complexity of the GUI is not relevant for passing the project.

Each project team must turn in

- a folder with a printout of the project documentation
- and the documentation as a PDF document in an email.

Structure your documentation as shown in the document in the following pages here.

Activity	Deadline/Date
Email your documentation as a single PDF file to dorothee.koch@hft-stuttgart.de and to viktor.kampo@hft-stuttgart.de	**June 21, 2010**
Deliver a hard copy on paper of your complete documentation to Mr. Kampo or Prof. Koch	**June 21, 2010**
Three project teams selected by me will demonstrate their projects in a live demo.	**July 1, 2010**

Remember:
Passing of the project is the prerequisite for participating in the exam.
The project is graded only on a pass/fail basis.
Use the following sample structure for your project documents.

A Database for the
Management of Weather Forecasts

Project for the class Databases
in the Summer Semester 2010

The following persons have contributed to this project:

< signature person 1>	< signature person 2>
< printed name person 1>	< printed name person 2>

1 Data Model for the Weather Forecast Database

1.1 Explanation of the Data and the Application

<Overview: your explanation of the situation you want to store data about; which data is relevant, who will use it for which purposes>

1.2 The Data Model

<your ER or UML model>

2 Relational Design

2.1 Table Schemas

<list of the created tables with their schemas>

2.2 Normalization

<brief explanation of the normalization status and what decisions were made for this>

2.3 Integrity Constraints

<Discussion of necessary integrity constraints. Which decisions were made>

3 System Requirements

<brief description of the DBMS (which version) you have used under which operating system, and any other necessary software or hardware parameters>

4 Use Cases / Transactions / Triggers

4.1 Use Cases

<a description of the functionality of how your database will be used. This includes a description of use cases with diagrams.>

4.2 Transactions

<explanation of the queries / transactions that belong to the use cases. One query should show something involving your integrity constraints. You should also document your transaction design here and explain which isolation level should be used and why. You must have at least one interesting transaction for each group member, and a minimum of three queries involving joins (five, if your group has more than two members).>

4.3 Triggers

<explain the triggers that you have implemented. There should be at least one trigger per group member>

5 Physical Design

<Make assumptions about the quantity of data and the types of data access for the future use of your database, and suggest a physical design for your system. Explain your design decisions.>

6 List of References

<list of references other than the class notes that you have used>

7 Appendix

7.1 Database Tables with Data

<the printout of your tables with contents>

7.2 Query Examples and Transactions

<the printout of your queries / transactions with results>

7.3 Description of the Application and User Interface

<the code of your application. Please use inline comments!>

7.4 ???

<any other interesting things you did that you would like to document>