4003-485 Database Concepts: P3 Handout

(Revised May 31, 2009)

1. Objectives

- To experience several phases of a database application development cycle.
- To model a data domain using E/R diagrams.
- To implement the data domain in terms of relational database tables.
- To use database techniques to improve your database design and implementation.
- To develop a database application in Java that exercises your database tables.

Note: this project handout only provides broad requirements to permit each team with sufficient flexibility. Because the project handout is not fully prescriptive, each team must consult with me on an ongoing basis to ensure that their work is appropriate and deserving of a satisfactory grade! Feel free to make appropriate decisions, but clearly document each decision and its rationale, which must be included in the final project report.

2. A Database Application

Select an "interesting" data domain and application, and then design and develop a database application. To provide some idea of what is expected, here are a few initial suggestions for data domains, but feel free to to develop your own data domain and application:

- ***Movies World:*** Model movies, actors, directors, genres, theaters, playing times, and reviews, and develop relational tables based on this domain. Use the many sources available on the web to populate your database. The database application can support queries such as finding movies by reviews, or by actors, or by given genre, or playing in Rochester, and so on. Examples of new data and changes to data include the making and release of new movies, reviews/ratings made by critics and moviegoer, and so on.
- ***Persistent Games Data***: Model the kind of data that you think that needs to be stored in an MMO game, and build an application around how this data is managed. Of course, you don't have to build the game itself, but you need to hypothesize what the game might need and what data needs to be stored and accessed. Refer to the following paper by Walker White et al., especially Section 2.3:
  http://www.cs.cornell.edu/~wmwhite/papers/2007-SIGMOD-Record-Games.pdf
  Of course, you are welcome to search for other references, or make use of your own knowledge of games!
- ***US Census:*** Using the US Census data (lots of data available at their website), develop an application that can present interesting information about US states, population, and more; the data lends itself to several interesting entities and relationships. The application could provide functionality to present the data in useful ways, e.g., how are the number of telephones related to the income in a zip code, and also provide functionality to enter and modify data.
- ***Campus Rentals:*** Model rental apartments or campus dorms, their features (bedrooms, fixtures, and so on), transportation availability, neighborhood crime rates, rental rates, noise level, geek level, and so on. Queries would be include how to find suitable apartments based on how much a tenant wants to pay or where they want to live, and so on. New apartments could be added and rented apartments could be modified as needed.
- ***Roll-your-own:*** The team comes up with a similar domain and application.

2.1 Specifics

- Select an application that requires a database.
- Model a data domain and define application functionality, which must include all CRUD applications. (CRUD = Create, Retrieve, Update and Delete.)
- Design and implement the database schema, with integrity constraints. In particular, you must make use of primary and foreign keys, integrity constraints, triggers, and stored procedures, and indexes.
- Populate the tables with reasonable amounts of data (a handful of rows is not enough: some tables must have more than 100s-1000s of rows!).
- Write the code needed to access the database system from a Java application: use JDBC or JDO or Hibernate. If you use any other Java package (or any other programming language), check with me for appropriateness.

4003-485 Database Concepts: P3 Handout

(Revised May 31, 2009)

3. Schedule and Deliverables

Due Date	Event	High-level Description
See myCourses calendar for each submission deadline. (No late submissions, but early submissions are always welcome!)	***Team formation***	• In class discussion.
	Preproposal	• A one-page plain text outline of your idea and application, with some description of the data and proposed application functionality.
	Phase 1 Submission	• Initial proposal (with initial E/R diagrams) describing your proposed data domain and application.. Use shareware/freeware software to generate E-R diagrams, or you scan in a hand-drawn diagram! • Describe data domain and submit initial ER diagram with ~8 entity sets and ~4 relationship sets. When converted to tables, the database must have around 8 meaningful tables. • Summarize the functionality that your database application will provide and what the final submission will include
	Phase 2 Submission	• A skeletal version of the basic database application and database that implements Phase 1 proposed functionality. Include basic documentation, code, SQL files/scripts to populate database, etc.
	Phase 3 Submission (including presentation, P3 exam, and final submission	• A complete version of your database system and application o Must include integrity constraints, primary and foreign keys, triggers, stored procedures, and indexes. o Tables with reasonable amounts of data (a handful of rows is not enough: some tables must have more than 100s-1000s of rows!). • Presentation • Final submission must include: o A softcopy of your presentation. o Working (and documented) source code of your database and application o The README file, which should describe the layout of your submission, and specifically the contributions made by each team member to the code.

4. Submission Information

For each of the three major submissions, place the files to be submitted (e.g., `*.java`, `*.sql`, `*.txt`, `*.doc`, `READMEs`, and `Makefiles`) in directories named `phase1`, `phase2`, and `phase3` respectively in your CS group account on a CS department SUN machine. Each submission should be complete and include all relevant files (even those submitted in earlier project submissions). Make each submission self-contained.

On or before the deadline for each phase, submit each phase electronically using:

```
submit -v rkr-grd dbc-phase0 phase0
submit -v rkr-grd dbc-phase1 phase1
submit -v rkr-grd dbc-phase2 phase2
submit -v rkr-grd dbc-phase3 phase3
```

If you implement the project on your own computer, upload all files to your CS UNIX group account before submission. Documentation must also include instructions on how to install your DBMS and data.

4003-485 Database Concepts: P3 Handout

(Revised May 31, 2009)

5. Project Evaluation

P3 scoring will be done as follows:

Step 1: Team Score

Phase 0	10%	
Phase 1	20%	
Phase 2	25%	
Phase 3	25%	
Paper (including documentation)	20%	
Overall team score	**100%**	
Scaled team score	**60%**	(60% of overall team score)

Step 2: Individual Score

Individualized team score	60% ± δ	Adjusted up or down from scaled team score based on (a) peer evaluation feedback, and (b) individual P3-related performance during demo and quarter (as observed by instructor).
Weekly status updates	5%	Individual submissions. See peer evaluation week survey.
Presentation/demo	15%	Individual score (composite of group and individual).
P3 questions during final exam	20%	If a student's performance on P3-related questions is significantly better or worse than her or his team's, individual grade will be further adjusted up or down.
Final P3 score	**100%**	See P3 component weight in the Course Syllabus handout.

Follow the Academic Honesty policies mentioned in the Syllabus handout. As usual, check with the instructor if you do not understand something. Ignorance of the rules is never an excuse!