

## CMSI 486 Introduction to Database Systems

Fall 2013 -- 3.0 units

Tuesday, Thursday, 10:50 a.m. - 12:05 p.m. in Pereira 140

Final exam **11:00 a.m. on Tuesday**, December 10, 2013

### Course Overview

---

**Dr. Stephanie E. August** *Associate Professor*

*Office Hours:* Tuesday, 5:00-6:00 p.m.

Wednesday, 1:00-4:00 p.m. *and by appointment.*

*These times will occasionally be rescheduled as needed to accommodate an NSF commitment*

*Office:* Doolan 201b

*Students are welcome to drop by any time and check whether instructor is available*

*Phone:* (310) 338-5973

*Internet:* saugust@lmu.edu

*Prof. August plans to respond within one business day.*

*Saturday, Sunday, and holidays are not business days.*

*Website:* <http://myweb.lmu.edu/saugust>

---

*Course Website:* MYLMU Connect [CMSI-486-01 Intro Database Systems 201330](#)

---

### Course Description

This course introduces the essential principles that guide the design, implementation, and management of systems capable of managing large amounts of data efficiently. Topics include database system structure, semantic data modeling, relational databases, formal query languages, integrity and security, physical design of databases, indexing and hashing. Query processing and optimization, transaction processing, concurrency, and crash recovery are introduced. The course also offers students hands-on experience with a DBMS and database design tool. New directions in the field are surveyed.

### Specific Learning Outcomes:

Upon completion of this course, it is anticipated that the student will be able to:

- Use an entity-relationship diagram to design a small relational database
- Document a relational database
- Normalize an existing database
- Create and populate database tables
- Write basic database queries using SQL
- Map database schema to class diagrams
- Estimate the time required to process basic database queries
- Identify factors that determine the efficiency and effectiveness of a database

### Prerequisites

CMSI 386 Programming Languages; CMSI 387 Operating Systems; mastery of a high-level programming language such as Java, C++, or another language that allows for user-defined data types or structures; some knowledge of computer systems and elementary logic

**Required**

- . Willingness to participate in class discussions, workshops, and classroom activities, and to complete readings prior to lectures. Classroom activities will include both individual and group assignments.
- . One 15-minute meeting with the instructor during the semester, either during office hours or by appointment.

**Expected Work**

Readings prior to lectures, and participation in class discussions and workshops.

Written and oral homework assignments and programming assignments to reinforce lectures and readings.

Term project involving the description and definition of a DBMS for a business enterprise.

- . The course project is due on the last day of class.
- . The course project includes multiple deliverables outlined in the project handout.
- . The deliverables are treated as homework assignments.
- . They are reviewed and returned, with the expectation that each subsequent deliverable will incorporate the previous, and now revised, deliverables.
- . If you complete all class assignments well and on time, the majority of the work for the project will be completed before the project is due.
- . The final project submission is the basis for the full project grade, which means that working on the project throughout the semester can lead to a perfect score on the project.

**Project Notebooks** (*this will change iff online submission proves to be effective for review*)

- . A 3-ring binder has proved to be the most effective way to (a) review consistency among components of the project, e.g., requirements, ERD, and DDL, and (b) to keep track of comments made on the various iterations of the project for each student.
- . I expect each student to store in a 3-ring binder, neatly organized, all documents related to the term project. The project binder must be complete, up-to-date, and available on each project due date. At the end of the term, students turn in the notebook for final grading. Students can pick up their graded project notebooks at the beginning of the following term. I will recycle any projects remaining after three weeks into the subsequent term.

MyLMU|Connect will be used to distribute course updates, assignments, and documents. Students are expected to check the MyLMU|Connect course twice a week and make certain they receive email addressed to their LionMail email account.

Completion of all book problems related to the assigned readings is recommended

**Work Load Expectations**

Students are expected to spend an average minimum of six hours per week on class-related learning activities, in line with LMU's Credit Hour Policy (see [http://www.lmu.edu/Assets/LMU+Credit+Hour+Policy\\_Final.pdf](http://www.lmu.edu/Assets/LMU+Credit+Hour+Policy_Final.pdf)). Programming assignments might require additional time for some students to complete.

### Slip Days

You have three (3) *Slip Days*.

- . Use them to extend a due date, 1 slip day for a 1-day extension
- . Use them one at a time or all at once or in any combination
- . They follow you around when you pair up – you are counted individually!
  - E.g. *A* has 2, *B* has 0. Project is late by 1 day. *A* uses 1, *B* is 1 day late
- . Late is 1/3 off per day

### Exams

One brief quiz on basic course concepts, one midterm, one comprehensive final exam.

### Text and Required Materials

Silberschatz, Abraham, Korth, Henry F. and Sudarshan, S. *Database System Concepts*. 6th ed. Mc Graw-Hill, 2011. ISBN 978-0-07-352332-3 Any edition or similar textbook will do, but it will be the student's responsibility to identify relevant portions of the text.

#### Required technology:

- Access to a computer running MySQL Workbench.
- Access to course information on *MyLMU Connect*. Students are expected to inspect the course MyLMU|Connect frequently for announcements, updates, assignments, and documents, and to use MyLMU|Connect to communicate among class members.
- *LionShare* or similar file sharing software to make student files available to the instructor and other students.

#### Useful technology:

- Laptop for in-class activities. Not required; these group activities simply require *someone* in each group to bring a laptop to class.

### Additional References

Churcher, Clare. *Beginning database design: from novice to professional*. Apress, Berkeley, CA, 2007. ISBN978-1590597699

Connolly, Thomas, Begg, Carolyn E., and Begg, Carolyn. *Database Systems: A practical approach to design, implementation and management*. 4th ed. Addison-Wesley, Harlow, England, 2004. ISBN 978-0321210258

Date, C. J. *An Introduction to Database Systems*. 8th edition. Addison-Wesley Publishing Company, Reading MA, 2003.

Date, C.J and Darwin, Hugh. *A Guide to the SQL Standard: A user's guide to the standard relational language SQL*. 4th ed. Addison-Wesley, Reading MA, 1997.

Elmasri, Ramez and Navathe, Shamkant B. *Fundamentals of Database Systems*. 5th ed. Addison-Wesley, Reading MA, 2006.

Shah, Nilesh. *Database Systems Using Oracle: A Simplified Guide to SQL and PL/SQL*. 2<sup>nd</sup> ed. Prentice Hall, 2004.

Garcia-Molina, Hector; Ullman, Jeffrey D; and Widom, Jennifer D. *Database Systems: The Complete Book*. Prentice Hall, 2001.

Supplementary materials as posted on MyLMU|Connect or handed out in class.

### Tentative Nature of the Syllabus

If necessary, this syllabus and its contents are subject to revision; students are responsible for any changes or modifications distributed in class or posted on LMU's course management system MYLMU Connect.

### Grading

Your final grade will be weighted as follows:

Activity	Course Points	Percentage of Total Grade
Participation	45	15%
Assignments	60	20%
Project	60	20%
Quiz	15	5%
Midterm	60	20%
Final	60	20%

Participation rewards good behavior. Think EPA:

- . Effort: Office hours, doing every single reading, writing, programming, presentation assignment
- . Participation: Actively participating in all classroom activities, raising hand in class, asking questions on the discussion board
- . Altruism: Helping other students in lab sessions, answering questions on the discussion board

All assignments, project materials, and oral presentations will be graded on content (50%) and presentation (50%). Assignments and other materials are to be uploaded to MyLMU|Connect or left on the instructor's desk, depending upon the assignment, before class on the due date.

The course project includes multiple deliverables outlined in the project handout. The deliverables are treated as homework assignments. They are reviewed and returned, with the expectation that each subsequent deliverable will incorporate the previous, and now revised, deliverables. The final project submission is the basis for the full project grade, which means that working on the project throughout the semester can lead to a perfect score on the project.

Students will participate in several group activities in the classroom over the course of the semester. Students will be graded both on the group product and on their individual contribution to the group effort. Students who miss these activities will receive no credit for the group product. Make sure your name is on each group report.

A request for a grade of Incomplete will be considered only when the student requesting the incomplete has completed 80% of the coursework, and has at least a B average in the coursework completed.

Assignments are due at the beginning of class on the due date and should be uploaded to MyLMU|Connect before class starts. Papers not submitted by 10 minutes after class has started are considered late. Late assignments are penalized as noted in *Slip Days* above.

Refer to the **Teaching Philosophy and Course Policies** handout for additional information.

Project	Deliverables due on Thursdays
Project Reviews	In class; date approximate
Homework	Assignments due Thursdays
Workshops	In class as needed

Week	Date	Topic	Readings in SKS	Assignments/Activities
<b>Database Management</b>				
1	27-Aug 29-Aug	Course Introduction Evolution of database management systems	ch.1	<i>Syllabus quiz</i>
<b>The Relational Model and Relational Operations</b>				
2	3-Sep 5-Sep	The Relational Model Basic Operations: select, project, rename, cartesian product	ch. 2	<b>HW #01</b>
3	10-Sep 12-Sep	Complex Operations: join, set operations, division <b>Quiz</b>		Project Deliverable #1
<b>Semantic Data Modeling</b>				
4	17-Sep 19-Sep	Entities, attributes, relationships Cardinality, optionality Supertypes, subtypes; weak sets and associative entities	ch. 7	<b>HW #02</b>
5	24-Sep 26-Sep	Mapping ERDs to Tables Reverse Engineering		Modeling Workshop Table Mapping Workshop Reverse Engineering Workshop Project Deliverable #2
<b>Commercial Relational Query Language: SQL</b>				
6	1-Oct 3-Oct	Data Manipulation Language Data Definition Language	ch. 3, 4, and possibly parts of 27	<b>HW #03</b> SQL Workshop
7	8-Oct 10-Oct	Data Control Language Application Programming Interfaces		Project Deliverable #3
<b>Relational Database Integrity and Design</b>				
8	15-Oct 17-Oct	Functional Dependencies Relational Database Design: Normalization Implementing Integrity Constraints	ch. 8	<b>HW #04</b> Project Design Review #1 Functional Dependency Workshop
9	22-Oct 24-Oct	Maintaining Database Security <b>Midterm</b>		Normalization Workshop Project Deliverable #4
<b>Physical Database Design</b>				
10	29-Oct 31-Oct	File Structure and Storage	ch. 10	<b>HW #05</b>
11	1-Nov 5-Nov 7-Nov	<i>Last day to withdraw or apply for Credit/No Credit grading</i> B+ Trees Indexing and Hashing	ch. 11	B+ Workshop Project Deliverable #5
<b>Query Processing</b>				
12	12-Nov 14-Nov	Query Processing	ch. 12	Project Review #2
13	19-Nov 21-Nov	Query Optimization	ch. 13	Project Deliverable #6

Project	Deliverables due on Thursdays
Project Reviews	In class; date approximate
Homework	Assignments due Thursdays
Workshops	In class as needed

14	26-Nov	Query Optimization	HW #06
	28-Nov	<i>Thanksgiving</i>	
<b>Transaction Management and Performance</b>			
15	3-Dec	Recovery, Transaction Management, Concurrency	
	5-Dec	DBMS Architectures, Performance Factors, Future Trends	selections from ch. 14, 15, 16
16	10-Dec	<b>Final, 11 a.m.</b>	Project Deliverable #6

Note: Dates and topics might change as the term progresses. Students are responsible for any changes or modifications announced in class.

## Teaching Philosophy and Course Policies

Dr. Stephanie E. August

### Philosophy

The instructor guides, the student explores. My job is to show the students what to learn and how to master it. The student's job is to explore the paths laid out before them. I assume that students are taking the class because the subject interests them and expect the students to take the initiative in learning the material. Everything we learn will be useful later, so it is important to be exposed to many different ideas.

I expect all students to contribute to the learning experience and not simply wait to be taught. In a graduate course, students come to class prepared to discuss the topic at hand, relate it to their previous experience, and project how it will be used in the future. Although undergraduates might think they have little direct experience with the topic, they are encouraged to find evidence of it in their life experiences and current news articles. All students are advised to transfer knowledge from other disciplines wherever possible.

Students are always welcome to discuss course material with the instructor, but they are also expected to take responsibility for mastering the course material and to seek out additional references for amplification and clarification of course concepts. The course description provides several additional references to use as a starting point.

### Expected Work

Course-specific expectations are noted in the Course Description. In addition, I have the following general expectations.

Students are expected to take responsibility for mastering course material, rather than expecting to be provided with all the answers. If an assignment is unclear, the student should discuss the assignment with the instructor, but is also expected to seek out published resources related to the assignment. Brief clarification can be accomplished in class; extended discussion of the assignment, especially when it does not pertain to the entire class, will be handled during office hours, by phone or email, or by appointment.

*Students are responsible for all the material in the assigned readings, whether or not it is covered in class, and for all material presented in class, whether or not it is in the assigned readings.* Students are expected to complete the assigned reading prior to lecture and to participate in class discussions.

Students are expected to take responsibility for keeping track of deliverables and due dates throughout the semester. Students are expected to turn in materials according to the schedule distributed by the instructor at the beginning of the term, unless the instructor explicitly issues an updated course schedule; students should not expect the instructor to remind them of due dates. Late assignments are not accepted, except as noted in the Course Description or announced in class.

Course information will be published on MyLMU|Connect, which I will often refer to by its former name, *Blackboard*. Students are expected to:

- inspect the course MyLMU|Connect site frequently for announcements, updates, assignments, documents
- use the course MyLMU|Connect site to communicate among class members via discussion boards and email
- make certain they receive email from the email address listed for them on MyLMU|Connect

**Email Communication**

At times I will communicate with the entire class using campus email systems, or with individual students using MyLMU|Connect or the student's Lion mail account on PROWL. **It is essential** that you regularly check your lion.lmu.edu email address or forward your lion account email to your preferred email address.

My goal is to respond to your email within one business day. Saturday, Sunday, and holidays are not business days. When you have questions about an imminent assignment, post them to the course discussion board where another student has the opportunity to answer the questions. There is a possibility that I will respond in the middle of the night on a weekend, but this not something on which to count.

**Participation and Expectations for Classroom Behavior**

Students are expected to attend every class meeting. This develops the faculty/student relationship and builds a sense of community among the students. Roll will not be taken on a daily basis, but graded group activities will occur in class throughout the semester, and a grade of 0 will be entered for those activities that a student misses.

Students are expected to be active participants throughout the entire class and to contribute to the quality of the discussion. Please note that the frequency with which a student speaks in class is not a key criterion for effective class participation. The classroom should be considered a laboratory in which students can test their ability to convince their peers that they have approached complex problems correctly and that their approach will achieve the desired results.

Criteria that we use to measure effective class participation include:

1. Is the comment clear and relevant to the current discussion?
2. Does the student support the comment well using case facts and tools developed in the class?
3. Does the student explore all the implications and importance of the comment?
4. Is the comment insightful? Does it broaden the discussion and clarify the issues?
5. Are comments complete and concise (does the comment cover the point as well as possible in a few words as possible)?

An average comment satisfies 1 and some of 2. A good comment satisfies 1-3. An excellent comment satisfies 1-5. Class participation represents a major component of your grade in this course (15-30%).

Before asking "Will this be on the test?" or "When is the assignment due?" refer to the section on *Expected Work* above and contemplate whether the question meets these criteria for effective class participation. If not, see me during office hours, drop by my office, phone in the question, or send email. It is fair game to ask before or after class "When can I come by your office to go over this material?" but conversations requiring more than 30 seconds before or after class are likely to make one of us late for the next class or commitment and are unlikely to receive their due attention.

Students are not allowed to say 'I don't know' in this class when asked a question. A student is not required to know, but is expected to think. So if I ask you a question and you don't know the answer, you are responsible to think of an answer, to guess, to speculate, to wonder aloud.

Questions referring to material that was covered when the student asking the question was absent will be answered (or not) in class at the instructor's discretion.

While participation should be the norm, students are cautioned against over participating. If you have made several contributions already, let someone else have a turn, and avoid calling out answers to questions directed toward other students.



Disruptive behavior which is persistent or significantly interferes with classroom activities may be subject to disciplinary action. A student may be referred to the Office of Student Judicial Affairs if their behavior constitutes a violation of the conduct code.

### Grading

Your final grade will be weighted as noted in the Course Description handout.

Graduate-level final grades are assigned as follows:

A-/A	Superior	Student has mastered the material and presents it in a professional manner.
B-/B/B+	Satisfactory	Student has a good grasp of the material and presents it clearly.
C+/C/C-	Poor	Student did some work, but does not have a strong grasp of fundamental concepts.
F	Fail	Student failed to learn fundamental concepts.

Graduate students must earn at least a B in each 500-level course for it to count toward their degree. 500-level courses in which the student receives a B- or lower will need to be repeated for credit.

Undergraduate-level final grades are assigned as follows:

90- 100%	A-/A	Student has mastered the material
80- 89%	B-/B/B+	Student has a good grasp of the material.
70- 79%	C-/C/C+	Student has a basic understanding of the material.
50- 69%	D	Student did some work, but failed to learn fundamental concepts.
0- 49%	F	Student did an insufficient amount of work to satisfy course requirements.

This breakdown represents the floor of each grade range. For example, a student scoring 90% will receive *at least* a grade of A-.

Unless otherwise stated for a specific assignment or deliverable, half of the grade will reflect *content*, and half of the grade will reflect *presentation*.

An incomplete will be granted only when the student requesting the incomplete has completed 80% of the coursework, and has at least a B average in the coursework completed.

### Work Load Expectations

Graduate students are expected to spend an average minimum of six to eight hours per week on class-related learning activities. Undergraduate students are expected to spend an average minimum of six hours per week on class-related learning activities. Programming assignments might require additional time for some students to complete. Refer to LMU's Credit Hour Policy, (see [http://www.lmu.edu/Assets/LMU+Credit+Hour+Policy\\_Final.pdf](http://www.lmu.edu/Assets/LMU+Credit+Hour+Policy_Final.pdf)).

### Assignments

Submitted assignments reflect your attitude toward learning. Written assignments may be handwritten or typed. All are expected to be neat and legible. *Homework and papers with multiple spelling and/or grammatical errors or not representative of college-level work will be returned ungraded.* Find yourself a good spelling and grammar checker and/or a trusted human editor, if you have difficulty with English spelling and/or grammar. The Academic Resource Center in Daum Hall offers such a service, both on an appointment and a drop-in basis (see <http://academics.lmu.edu/arc/> or contact the center by phone 310.338.2847 or email [arc@lmu.edu](mailto:arc@lmu.edu)).

Student's name, the course name, the assignment number and due date must appear in the upper right corner of the first page of the assignment. Answer questions with complete sentences; include the question in the answer and show your reasoning.

Staple or otherwise fasten together pages of assignments submitted on paper or put the pages in a folder in which they are securely fastened. Points will be deducted for loose pages.

**Exams**

*Once a quiz or exam has begun, it will be assumed that anyone leaving the room is finished with the exam. Have adequate supplies (paper, pens, pencils, tissues) on hand, and take care of personal needs before coming to class. No makeup quizzes or exams will be given, except as noted below.*

If your job, team sports, or other commitments require you to travel during the term, special arrangements should be made prior to a missed class for submitting assignments, receiving assignments and handouts, or rescheduling quizzes or exams. If you miss a quiz or exam without making prior arrangements, I will enter a grade of 0 for the quiz or exam, and no makeups will be allowed.

**Electronic Devices**

Turn off and put out of sight all electronic devices other than laptops or tablet computers used for taking notes in class. Reading emails, completing homework assignments, or searching the internet for anything that will not augment the classroom experience for the entire class (students and instructor) should be avoided altogether. The distractions they cause disrupt class and usurp precious class time. A repeat offender may lose credit for the day's work.

**Academic Honesty and Integrity**

Students find it helpful to discuss approaches to assignments and projects with their classmates. However, unless an assignment has explicitly been declared a team assignment, each student is expected to complete and write up the assignment or project component on his or her own. When an approach has been discussed in a group, each student should clearly note on the copy of the work that she or he submits the nature of the collaboration and the name of each collaborator. Researching a problem on the Internet is considered to be collaboration and should be noted on the assignment. Relevant URLs should be noted on the assignment. Cheating on assignments by failing to note collaboration or not writing up the assignment on an individual basis, cheating on examinations, plagiarism, falsification of data, and related violations of LMU standards of honesty and integrity are not tolerated. Students who commit such offenses will receive a failing grade for the assignment or exam and/or a failing grade for the course, as well further disciplinary action.

Students are expected to understand what plagiarism is and avoid all forms of it. The website <http://libguides.lmu.edu/plagiarism> explains the plagiarism and discusses techniques for avoiding it. The page includes links to several informative tutorials.

Academic dishonesty will be treated as an extremely serious matter, with serious consequences that can range from receiving no credit for assignments/tests to expulsion. It is never permissible to turn in any work that has been copied from another student or copied from a source (including the Internet) without properly acknowledging the source. It is your responsibility to make sure that your work meets the standard of academic honesty set forth in the "LMU Honor Code and Process" in the LMU Bulletin (see [http://www.lmu.edu/about/services/registrar/Bulletin/Bulletins\\_in\\_PDF\\_Format.htm](http://www.lmu.edu/about/services/registrar/Bulletin/Bulletins_in_PDF_Format.htm)).

**Americans with Disabilities Act**

Students with special needs as addressed by the Americans with Disabilities Act who need reasonable modifications, special assistance, or accommodations in this course should promptly direct their request

to the Disability Support Services Office. Any student who currently has a documented disability (physical, learning, or psychological) needing academic accommodations should contact the Disability Services Office (Daum Hall Room 224, 310-338-4535) as early in the semester as possible. All discussions will remain confidential. Please visit <http://www.lmu.edu/dss> for additional information.

**Tentative Nature of the Syllabus**

If necessary, this syllabus and its contents are subject to revision; students are responsible for any changes or modifications distributed in class or posted on LMU's course management system MyLMU|Connect.

**Survival**

Eat before you are hungry, drink before you are thirsty, and sleep before you are tired, and you will have energy left to celebrate the completion of the course and enjoy the semester break. Eating, drinking, and sleeping are to be done outside the labs and classrooms. Covered beverages are acceptable, unless otherwise prohibited by the facility, providing that the student leaves the classroom at least as neat as it was then they arrived. Repeat offenders will find it difficult to complete the course, since they will be asked to leave the classroom and will lose their lab privileges.

To report an emergency or suspicious activity, contact the LMU Department of Public Safety by phone (x222 or 310-338-2893) or at the nearest emergency call box. In the event of an evacuation, follow the evacuation signage throughout the building to the designated safe refuge area where you will receive further instruction from Public Safety or a Building Captain. For more safety information and preparedness tips, visit <http://www.lmu.edu/emergency>.

# CMSI 486 DATABASE SYSTEMS

Dr. Stephanie E. August

## Enterprise Model Project

### Enterprise Model Project

#### Introduction

Over the course of the term, you will be designing and implementing a database as described in appendix A. This project is broken into a number of deliverables as listed in table 1. The due date for each deliverable is listed in appendix A.

Table 1. Project deliverables.

Deliverable	Assignment
1	Describe the enterprise
2	Define the environment
3	Develop logical and conceptual models of your enterprise
4	Define the database and formulate queries
5	Consider database integrity and security
6	Implement the database
7	Lessons learned
final	Completed notebook and floppy containing design, tables, queries, traces

You are expected to use a 3-ring notebook with dividers to organize the material for this project. There should be one divider for each of the sections listed in table 2 below. A detailed description of each section follows.

There will be eleven sections in all.

Pages are to be numbered consecutively within each section. For example, the first page of the enterprise description (section IV) should be numbered IV-1, the second page of the enterprise description should be numbered IV-2, and the  $n$ th page of the description should be IV- $n$ . If the design section (section VI) spans 10 pages they should be numbered VI-1 through VI-10.

Label all tables and figures. Table labels appear before the table, centered. Figure labels appear after the figure, centered. Observe the examples in this document and follow a similar format.

As each interim assignment is returned, you are expected to incorporate into your next version any corrections, modifications, or additions which I have noted on your project. The previous assignment, with corrections/modifications/additions indicated, should be included in the "history" section, and submitted each step of the way, along with the revised and augmented work. Your final report should contain one section for each deliverable, as well as the title page, table of contents, copy of the assignment with supplements, and historical section. The last should contain the previous submissions that have since been revised, as well as any notes you have made along the way to help you complete the project. These would include, but not be limited to, design tradeoffs and bibliographical references.

Table 2. Project notebook sections.

Part of Deliverable #	Belongs in Section #	Section Contents
3	I.	title page [see sample in appendix C]
3	II.	Table of Contents [three levels of headings with page numbers]
3	III.	<copy of the assignment, any supplementary material provided during the semester, as well as all evaluation sheets>
1	IV.	Description of the Enterprise
2	V.	Definition of the Environment V.1. Input and report forms. A list of input and report forms, with itemized data items. V.2. Assumptions. List of assumptions for your enterprise database. V.3. User-oriented data dictionary. V.4. Cross-reference table.
3	VI.	Enterprise Database Design VI.1. Logical model of the enterprise. VI.1.1. List of entities and attributes. VI.1.2. List of relationships and attributes. VI.1.3. Entity-Relationship diagram of the enterprise. VI.2. Conceptual model of the enterprise. VI.3. Table dictionary. VI.4. Attribute dictionary.
4	VII.	Database and Query Definition VII.1. Database Definition. SQL DDL for your database objects. VII.2. Database Queries. English version of 10+ database queries, and the SQL DML for each database query VII.3. Review sign-off sheet . VII.4. Design Limitations. Discussion of the limitations of your design.
5	VIII.	Database Integrity and Security VIII.1 Functional Dependencies. A list of the functional dependencies that hold on your database. VIII.2. Adjustments for Normalization. An explanation of the changes needed to normalize your database. VIII.3. Integrity and Security. A list (in English) of the integrity and security constraints which are to hold on your database.
6	IX.	Implementation Notes IX.1. Indices. A list of the indices used by your database, with a justification for each. IX.2. Data. The data used to populate your database. IX.3. Query Trace. A trace of the execution of each of your queries. IX.4. Implementation Assessment. An assessment of how smoothly your implementation went.
7	X.	Lessons Learned
2	XI.	Database Development History

As each interim assignment is returned, you are expected to incorporate into your next version any corrections, modifications, or additions which I have noted on your project. The previous assignment, with corrections/modifications/additions indicated, should be included in the "history" section, and submitted each step of the way, along with the revised and augmented work. Your final report should contain one section for each deliverable, as well as the title page, table of contents, copy of the assignment with supplements, and historical section. The last

## DATABASE SYSTEMS: Enterprise Model Project

should contain the previous submissions that have since been revised, as well as any notes you have made along the way to help you complete the project. These would include, but not be limited to, design tradeoffs and bibliographical references.

Each deliverable, except for deliverable 1, should be accompanied by a title page and a current table of contents. The table of contents should include at least the top three levels of headings and relevant page numbers.

The remainder of this document issues a caveat to developers, then describes the assignments associated with each deliverable. Appendix A provides a project problem description. Appendix B includes a template that can be used to develop your project notebook. Appendix C includes a sample title page. A copy of the review signoff sheet required in section VII can be found in Appendix D.

You are expected to be familiar with the required contents, format, and due date of each section of your project notebook.

### Notes to the Wise

Plan ahead, work all during term. You can't do this project in a single night. You can't even do an entire deliverable in a single night for the most part!!!

Remember the five *Ps*: *Proper planning prevents poor performance.*

Doing a good job on the project requires spending 8 to 10 hr/week on the course.

The deliverables in this project make sense and guide you through the development process if you follow them in order. The steps will appear redundant otherwise. For best results, complete each deliverable in sequence, and revise with each deliverable.

### **Deliverable 1: Describe the enterprise.**

**Turn in: Section IV. Description of the Enterprise:** Textual description of your enterprise.

Write a textual description of the enterprise (500 words). Describe the purpose of the enterprise, the people involved, and the information processing performed. Include descriptions of the records that need to be maintained, the entities to which they relate, and the relationships that exist among the entities. At this point, the goal is to understand the enterprise that you will be modeling. This description will evolve, and, perhaps, become more detailed, over the course of the project. *Do not include any "techtalk" in your description!* That is, avoid database-specific terms such as entity, relationship, or cardinality. Gear the description to a non-technical manager. Use the description in Appendix A (if provided) as a starting point, tailoring it to your specific enterprise model and adding details and questions/queries as needed.

**Deliverable 2: Define the environment.**

**Turn in:** Title Page

Table of Contents

Copy of Assignment

Section IV. Description of the Enterprise (revised as needed)

**New:** Section V.1. Input and report forms: List of input and report forms, with itemized data items.

Section V.2. Assumptions: List of assumptions for your enterprise database.

Section V.3 User-oriented data dictionary.

Section V.4 Cross-reference table.

Section XI, containing any material from S.I-IV that was revised.

In this step you will take a closer look at the environment in which your database will exist. Assume that you are having or have had a series of meetings and interviews with users of the proposed system, to determine their data needs and preferences. In real life, you will be working with the end users from day one, to ensure that the final product will meet their needs, as well as to gain their support for and confidence in the project.

First, consider how information will be put into the system, and retrieved from it. If you were actually working with an enterprise to design and implement an information processing system, you would have to identify and write out the format for each input document and database report, and for each input and output screen for every routine transaction to be performed against the database. For example, suppose we were developing a DBMS for a software consulting firm. One form we might use would be the consultant application form shown in figure 1. A screen listing potential consultants for a job appears in figure 2. In a large enterprise, the data analysis or business systems analysis staff would most likely perform this deliverable.

*Without actually designing the layout of the input/output screens and database reports, carefully think through daily operations. Draw up a list of the input form and report forms, and itemize the data values utilized on each.* For example, corresponding to figure 1 we would have the list shown in figure 3.

Next, **write out a list of assumptions for your environment** that you would have picked up during your customer meetings. You will add to this list as the project progresses. Some assumptions from the consulting enterprise are shown in figure 4. This example is by no means exhaustive.

Now, to help refine your system, you develop two documents:

- A *user-oriented data dictionary*, consisting of an alphabetical list of every data item referenced in any document, report, or routine transaction and an informal definition for each item. The dictionary should be created in table form, with two columns: Datum, and Informal Definition. An example of the dictionary can be found in table 3.

Users will reference this document independently from accessing the database itself to better understand the enterprise semantics and write appropriate queries and database programs.

SoftWare Consultants, Inc. Application Form			
Date of Application _____			
Name _____			
Address _____			
Telephone _____	Email _____	Fax _____	
Social Security Number _____			
Date of Birth* _____		Sex* _____	
Skills -- For each of the following, indicate your experience level:			
	None	Some	Extensive
C	_____	_____	_____
C++	_____	_____	_____
Windows	_____	_____	_____
Object-Oriented Programming	_____	_____	_____
Relational DBMS	_____	_____	_____
CLIPS	_____	_____	_____
Lisp	_____	_____	_____
Date Available to Work: _____			
Restrictions on Work Dates: _____			
To be filled in by interviewer:			
Date Hired _____		Reason not hired _____	
* SoftWare Consultants, Inc. does not discriminate on the basis of age or sex of applicant.			

Figure 1. Input form: Consultant application.

Potential Consultants			
1. The following consultant is eligible for the job.			
2. To see other eligible consultants, press enter until the message "No others are eligible" is displayed.			
Employee ID _____	Social Security Number _____		
Name _____			
Address _____			
Telephone _____	Email _____	Fax _____	
Rating _____			
Last Date Worked _____		Sex* _____	
Skill Levels:			
C _____	CLIPS _____	Linux _____	
C++ _____	Lisp _____	Windows _____	
Object-Oriented Programming _____		Relational DBMS _____	

Figure 2. Output Screen: Potential consultants for a job.

- A *cross-reference table* showing which items appear on the various documents, reports, or transactions you have already identified. Table 4 shows how the data would map to the documents, reports, and transactions of the consulting enterprise.

This document allows us to trace requirements to data items, forms, reports, and database transactions. If we make a change to any of these artifacts, this document will enable us to understand the impact the change will have on other artifacts.



## DATABASE SYSTEMS: Enterprise Model Project

SoftWare Consultants, Inc. -- Application Form

- Date of Application
- Name
- Address
- Telephone
- Email
- Fax
- Social Security Number
- Date of Birth
- Sex
- Skill experience level (None/Some/Extensive)
  - C
  - C++
  - Windows
  - Object-Oriented Programming
  - Relational DBMS
  - CLIPS
  - Lisp
- Date Available to Work
- Restrictions on Work Dates
- Date Hired
- Reason not hired

Figure 3. List corresponding to the consultant application input form of figure 1.

1. Prospective consultants fill out an application form, and are interviewed by both a regional consulting manager and a current consultant to ascertain interests of consultants and confirm their skill levels.
2. Consultants are assigned to only one job at a time.
3. Both skill level and requirements of the job determine a consultant's wage.
4. Clients are charged a daily rate and are billed weekly.
5. Assignments can be from one day to several months duration.
6. A consultant might work at the customer site, an office of SoftWare Consultants, Inc., or at the consultant's home, depending upon the requirements of the job.

Figure 4 Some assumptions for the SoftWare Consultants, Inc., enterprise.

Table 3. User-oriented data dictionary.

Datum <sup>1</sup>	Information Definition
consultantName	The name of a consultant in the form <last>, <first and rest>. Example: Doe, Jane Marie Smith
consultantAddress	The address of a consultant. Includes street number, street name, unit number, city, state, postal code if relevant, and country, if outside the U.S. Example A: 123 N. Main Street, Suite 14 Central, CA 99999      Example B: O'Donnell Road Doolan, County Clare, Ireland
consultantPhone	The telephone number of a consultant. Includes area code (if US) or country code and city code (if not US) and phone number.
potentialConsultantName	The name of a prospective consultant. Refer to <i>conName</i> for format.
potentialConsultantAddress	The address of a prospective consultant. Refer to <i>conAddr</i> for format.
potentialConsultantPhone	The telephone number of a prospective consultant. Refer to <i>conPhone</i> for format.
software_skill	Software skill possessed by a person or required by a job. Examples: C#, LISP, CLIPS, Java, Linux, Windows, relational DBMS
software_skill_level	Level of expertise of a skill either possessed by a consultant or required by a project. Allowed values: <i>None, some, extensive</i> .
...	...

Table 4. Mapping of data to forms and transactions.

Datum	Form or screen								
	Prospective Consultant Application	Prospective Client Application	Client Information	Consultant	:				
consultantName				X					
consultantAddress				X					
consultantPhone				X					
potentialConsultantName	X								
potentialConsultantAddress	X								
potentialConsultantPhone	X								
software_skill	X			X					
software_skill_level	X			X					
...									...

<sup>1</sup> Avoid using all capitals for your attribute names; it is easier to read names with mixed capitalization or lowercase with underscores ( \_ ) between words.

**Deliverable 3: Develop logical and conceptual models of your enterprise.**

**Turn in:** Title Page

Table of Contents (revised as needed)

Copy of Assignment

Section IV. Description of the Enterprise (revised as needed)

Section V. Definition of the Environment (revised as needed)

**New:** Section VI.1. Logical model of the enterprise.

VI.1.1. List of entities and attributes.

VI.1.2. List of relationships and attributes.

VI.1.3. Entity-relationship diagram of the enterprise.

Section VI.2. Conceptual model of the enterprise.

Section VI.3. Table dictionary.

Section VI.4. Attribute dictionary.

Section XI, containing any material from S.I-V that was revised.

Now you are going to develop both the logical and conceptual models of your enterprise. First, you develop a logical or semantic model of your enterprise. This is a three step process:

- *Make a list of all entities and their associated attributes.*  
This may take several attempts, and different designers will arrive at different solutions. In identifying entities, you will examine the data dictionary you developed in Section V.3. Think about the enterprise, and enumerate the persons, places, events, objects, or concepts that you need to keep information about. The original data dictionary may have some items that you need not store in the database. They can be dropped from the list of attributes.
- *Make a list of relationships to be represented and any descriptive attributes for them.*  
At this point you may decide not to store some of the items from the original data dictionary. For example, information such as *payroll totals* might only be needed for a periodic payroll report. It might make more sense to calculate it when needed, than to explicitly store it in the database. Document any changes you make, and retain this information in the history section.

For the SoftWare Consultants, Inc., example we might have the entity sets:

**CONSULTANT**<sup>2</sup>: consultantName, consultantPhone, consultantAddress,  
consultantID, SS#, DoB

**SKILLS**: software\_skill

and the relationship set:

**CONSULTANT\_SKILLS**: consultantID, software\_skill,  
software\_skill\_level

---

<sup>2</sup> Entity and relationship names are capitalized here to make them stand out; use mixed case or lower case with underscores in the actual database.

## DATABASE SYSTEMS: Enterprise Model Project

- Draw an E-R diagram to represent the enterprise.  
Be sure to identify relationship cardinalities, and any weak entity sets. Use generalization and aggregation as necessary to express relationships.

This document provides a picture of the entire enterprise model. People working with the database will use the ERD to navigate a database the way we use a road map to navigate through our cities and states.

Pointers:

- The textual description of the enterprise you are modeling should reflect the semantics of your entity-relationship diagram *accurately*, including entity sets, relationship sets, cardinalities, and attributes.
- Each entity set should represent a single concept -- don't confuse *order* and *product*, for instance, or *order* and *customer*.
- Explicitly represent relationships between/among entity sets in the E-R diagram.
- Each attribute should have a unique name!
- Indicate the primary key (PK) for each entity set.
- Indicate which attributes are candidate keys (CK) in each attribute. Remember that the PK is always a CK!
- Do not include foreign keys in the ERD. They are implicit in the relationships represented in the diagram. If you include FKs in spite of this request, you must explicitly identify each foreign key attribute as a foreign key in each relation and indicate its source or parent table.
- Indicate each discriminator of a weak set, where such a discriminating attribute exists.

Next, *produce a conceptual model of your enterprise*, by reducing the E-R diagram to tables in the relational model, as we did in class. You can do this using either the

*table-name-R(attribute-1, ..., attribute-n)*

CK: *attribute-1, attribute-i*

FK: *attribute-j* references *table-name-S.attribute-k*

notation, or the

<i>table-name-R</i>		
<i>attribute-1</i>	...	<i>attribute-n</i>
PK		CK
CK		FK
		references <i>table-name-S.attribute-k</i>

notation. Be sure to indicate in each table which attributes participate in the primary key, which attributes are parts of candidate keys, and which attributes are foreign keys. Remember to indicate the source of each foreign key.

The final steps of this deliverable are to create a *revised* data dictionary in two parts. The first part, the "table dictionary", will consist of a *three-column table* listing *each table* to be included in the database, the *attributes* that are in the table, and an *informal definition of the table*. This will provide an easy reference guide to your database. The second part of the data dictionary, the "attribute dictionary", will consist of a revised version of the user-oriented data dictionary described in section V.3 above. *To that document, add a column specifying the table in which each attribute is used*. If an attribute has been renamed and used as a foreign key in a table, add it to the attribute dictionary list, specifying the table in which the renamed version is used. In the definition portion of the entry, indicate the attribute which the foreign key attribute references.

**Deliverable 4: Define the database and formulate queries.**

**Turn in: Title Page**

**Table of Contents** (revised as needed)

**Copy of Assignment**

**Section IV. Description of the Enterprise** (revised as needed)

**Section V. Definition of the Environment** (revised as needed)

**Section VI. Enterprise Database Design** (revised as needed)

**New: Section VII.1. Database Definition:** SQL DDL for your database objects.

**Section VII.2. English version of 10+ database queries, and the SQL DML for each database query.**

**Section VII.3. Review sign-off sheet.**

**Section VII.4. Design Limitations: Discussion of the limitations of your design.**

**Section XI, containing any material from S.I-VI that was revised.**

Write the SQL DDL statements to create all tables needed to implement the design completed in Deliverable 3. Include relevant integrity constraints for foreign keys, that is, for each foreign key declared, specify in the DDL the action that should be taken if the related valued in the parent table is deleted or updated. Use ALTER TABLE statements to include integrity constraints rather than declaring them in the DDL for each table. Include your DDL in one or more .sql files, with an accompanying database build file that will load the SQL statements.

*Compose English language queries that are needed to process at least ten (10) nonroutine requests for information from the database just created. These ten queries should be OLAP-oriented, rather than OLTP-oriented. For each, write the request in English, followed by the corresponding SQL command. At least 5 of these should be "difficult" queries, that is, queries involving multiple tables and/or complex operations on a single table.*

*Team with another student to review designs, schemes. A Review Sign-off Sheet is found in appendix D. Have the other student sign off on the design using this sheet, and include evidence of the sign-off in the notebook. The comment section should have something more meaningful than "Looks OK to me." You will receive two grades for this sheet. First, you be graded on the quality of the review and comments that you provide for *some other* student. Second, you will*

be graded on whether or not a completed sheet (with your reviewer's comments) is included in your notebook.

*Identify the limitations of your design.* Include a discussion of the kinds of information which are difficult to extract from your database, due to its content and structure and/or due to the limitations of SQL. Propose modifications to your enterprise model or the query language which would make it easier to extract the information.

**Deliverable 5: Consider database integrity and security.**

**Turn in:** Title Page

Table of Contents (revised as needed)

Copy of Assignment

**Section IV. Description of the Enterprise** (revised as needed)

**Section V. Definition of the Environment** (revised as needed)

**Section VI. Enterprise Database Design** (revised as needed)

**Section VII. Database and Query Definition** (*revised to incorporate the DDL to support the constraint listed in Section VIII.3*, and other revisions as needed)

**New:** **Section VIII.1. Functional Dependencies:** A list of the functional dependencies that hold on your database.

**Section VIII.2. Adjustments for Normalization:** An explanation of the changes needed to normalize your database.

**Section VIII.3. Integrity and Security:** A list (in English) of the integrity and security constraints which are to hold on your database.

**Section XI, containing any material from S.I-VII that was revised.**

At this point you need to concern yourself with integrity and security issues related to your database. First, let us consider normalization. This will be a two-step process:

- *Identify all of the functional dependencies that hold on the database.*
- *Normalize each relation identified in the preceding deliverable. Be sure that every attribute listed in the first step above appears in at least one table. Then decide whether the table should be implemented in the highest normal form. If not, explain why.*

*Modify your environment definition and cross-reference table from deliverable 2 to reflect any changes resulting from the normalization process.*

*Revise your table and attribute dictionaries from section VI.3 and VI.4 as needed.*

*Revise your entity-relationship diagram as needed.*

*List the integrity and security constraints that should hold on your database. Consider how the use of foreign keys, assertions, triggers, and grant statements support integrity and security in your database. Your writeup should clearly identify classes of users and the privileges accorded to each. Justify each constraint employed. Identify and define any views required. Any views created need to be added to both parts of the data dictionary. Write the SQL statements needed to support these constraints, wherever possible. If SQL does not support integrity and security*

constraints which you would like to maintain on your DB, explain where it falls short, and how you can compensate for the shortcoming(s). (You can suggest application programs or extensions to SQL to shore up the constraint support mechanism.) Refer to the textbook for ideas.

**Deliverable 6: Implement the database.**

**Turn in:** Title Page

Table of Contents (revised as needed)

Copy of Assignment

Section IV. Description of the Enterprise (revised as needed)

Section V. Definition of the Environment (revised as needed)

Section VI. Enterprise Database Design (revised as needed)

Section VII. Database and Query Definition (*revised to include the SQL DDL for index declarations listed in Section XI.1*, and other revisions as needed)

Section VIII. Database Integrity and Security

*New:* Section IX.1. **Indices:** A list of the indices used by your database, with a justification for each.

Section IX.2. **Data:** The data used to populate your database.

Section IX.3. **Query Trace:** A trace of the execution of each of your queries.

Section IX.4. **Implementation Assessment:** An assessment of how smoothly your implementation went.

**Section XI, containing any material from S.I-VIII that was revised.**

*Implement your database using Oracle or PostgreSQL.* Make any needed adjustments to your DDL files. Indicate where in your database an index needs to be added, and explain why it should be added. Add the DDL to generate each index, as needed.

*Populate your database.* Create a file containing the *insert* statements needed to perform your initial data load.

*Implement, and test, and save each of the queries* developed in Deliverable 4. If possible, test your implementation on more than one DBMS, and describe the differences you encountered, if any. If you use ANSI standard SQL, your code should run on any ANSI-compliant DBMS.

Create a trace of the creation and load of your database and the test of each query, using the `spool` facility of your DBMS or a similar mechanism.

**Deliverable 7: Lessons learned.**

**Turn in:** Section XI. Lessons Learned.

Sections I-IX (revised as needed)

**Section XI, containing any material from S.I-IX that was revised.**

*Describe what you learned during the course of this project.* This discussion should be about 500 words in length. Your lessons learned section should at a minimum address the following questions: What would you do differently next time? What database resources should you have used, or do you wish had been available? How many hours did you put in? How useful was the

## DATABASE SYSTEMS: Enterprise Model Project

project in helping you learn the course material. What would facilitated completing the project? How do you like ORACLE or the DBMS engine you used on the project?

### **Final Deliverable: Deliver completed project**

**Turn in: Code and documentation** Individual medium (CD, link to files on an accessible file server, folder with files uploaded to MyLMU Connect) containing a README file and all source code, build files, data files, query files, and listing files showing a trace of database build and load, as well as query processing.

**Completed notebook** Includes complete project report, lessons learned, development history, SQL, English queries and SQL implementation, query trace, and an inventory of all files includes on the medium and the location of the code and documentation if it is not included on a medium submitted in the notebook itself.

**Demo** One-on-one demo with instructor loading BUILD.SQL. files and data files, then running the query script and comparing the results to those included in the project report.

### **Note:**

Some of the ideas for this project were taken from:

Ricardo, Catherine M. *Database Systems: Principles, design, and implementation*.  
New York, Macmillan, 1990. 0-02-399665-X



# CMSI 486 INTRODUCTION TO DATABASE SYSTEMS

Dr. Stephanie E. August

## Enterprise Model Project

### Appendix A. Project Details.

#### A.1. Project Domain

Over the course of the term, you will be designing and implementing a database for an enterprise whose nature will be discussed separately. Appendix A provides basic problem statement for various enterprise databases. You will need to supply a problem statement, following discussions with the instructor, refine the requirements for the project and develop the scope of your model. Due dates for each project deliverable follow in section A.3.

#### A.2. Prototypical Problem Statement

C.J. Date once said that only three database models actually existed: A university database, a part-supplier database, and a rental (think library or video rental enterprise) database. If a database programmer could model these three enterprises, s/he could model anything. This section gives a brief overview of each enterprise. *Caveat emptor* -- these descriptions are only the beginning! As you develop your data model, you will ask and answer questions that delve much more deeply into the structure and operation of the enterprise in order to design a database that will provide a stable design with good performance over a long period of time.

##### A.2.1 University Database

Your friend Dahta Bass has earned a vast fortune building intelligent homes and has decided to open a new online university to educate the next generation of students who will write interactive novels, build cognitive companions, design and construct biomechanical limbs, and develop ecologically sound farms, as well as study philosophy, art, and ancient literature. She heard that you were learning about database management systems, and has asked you to help her establish a reliable database to support all aspects of the day-to-day operations of the university. You have agreed to help her refine her requirements and implement a database management for her new endeavor. Dahta has told you that the university will evaluate and track applications for both graduate and undergraduate programs covering a wide variety of majors and also enrolls non-degree students. The university offers many classes. Class size varies from a size of one for an independent study course to 350 for the largest lecture course. Students will take several courses at once in most cases. Most courses are taught by a single faculty member, but some courses are team taught by two or more faculty members. A faculty member can belong to one department or have a joint appointment to more than one department. Each department belongs to a particular college or school within the university and is headed by one faculty member who is the chair of the department. Each college or school has a dean to whom all department heads within the college or school report.

The university will track each faculty member's publications, which can be conference or journal articles, book chapters, or books. Faculty often coauthor the publications with other faculty, students, or other outside collaborators. In the performing and fine arts, sculptures, choreographed dances, and the like are equivalent to publications.

Each faculty member has either a full-time or part-time assignment. Full-time faculty hold rank as assistant, associate, or full professor. Each faculty member advises one or more students.

Ms. Bass needs to record and retain information about the students, faculty, and staff at the university, and keep track of enrollments and grades. Tuition and other fees have not been established. The cost per unit may be the same across campus, or may vary by program.

The university will need to generate transcripts listing a student's majors, minors, course enrollments, grades, transfer credits, and full name, as well as any special notes recorded by the registrar. They will also need to produce reports of student demographic data, schedules of classes, lists of students on a dean's list, and a list of faculty organized by degree, rank, or department. Since Dahta anticipates having each faculty member teach several courses, she will want to record which faculty member is teaching each course

Clearly, Ms. Bass needs help identifying what information needs to be included in the final system. You asked her to clarify her needs by supplying you with a list of questions the database should be able to answer. She supplied you with the following:

*What classes does Indiana Jones teach?*

*What is the capacity for CMSI 522?*

*How many students from Gig Harbor, Washington, graduate with honors in 2009?*

*Which faculty members live in Tarzana?*

*List the certifications held by each staff member.*

*List the transcript for Olivia Sanchez.*

*How many students are currently enrolled in courses that are team taught?*

*What is the average SAT score of entering freshmen?*

*What is the average GRE score of students admitted to the MS in Digital Humanities program over the last three years?*

*Which courses are most popular?*

Over the next few weeks you work with Dahta to carefully scope out the extent of the information to be included in the delivered system, then proceed to design and implement it.

### **A.2.2 Video/DVD Rental Enterprise Database**

Your friend Seymour Muni has decided to expand his business empire and open a video/DVD rental enterprise. He heard that you were learning about database management systems, and has asked you to help her establish a reliable database to support day-to-day business operations. You have agreed to help her refine her requirements and implement a database management for her new enterprise. Seymour has told you that the enterprise rents movies in various formats to

customers and keeps an inventory of items on hand. The store keeps multiple copies of popular movies on hand, and might have the same movie available in various formats. He would like the customers to search her inventory by movie title and genre, and is considering extending the search capability to include actors, directors, producers, and release date. He is also toying with the idea of providing biographical information about actors and producers and directors, but isn't sure whether her budget will allow this feature. Seymour is planning to sell older movies that are no longer worth keeping on the shelves.

Mr. Muni needs to record and retain information about her customers, and keep track of rentals and sales. Customers will be allowed to rent multiple videos at once. Rental prices have not been established. They might be the same for all items, or he might vary the price by day of the week, type of movie rented, rating, or number of movies rented at a time.

The enterprise will need to generate periodic reports listing sales per reporting period. Since Seymour anticipates having multiple employees working at the store, he will want to record which employee completed each customer transaction.

Clearly, Mr. Muni needs help identifying what information needs to be included in the final system. You asked her to clarify her needs by supplying you with a list of questions the database should be able to answer. He supplied you with the following:

*Does the store carry Bambi?*

*How many copies of Matrix Revisited are currently available for rental?*

*Who rented both Sneakers and The Net in 2002?*

*Which customers live in Gig Harbor?*

*List all customers who have children in the 10-14 year age range.*

*What is the average age of the clerks who handled the rental transactions in August 2003?*

*What is the name of customer C129?*

*List the G movies currently available that were released between 1999 and 2001.*

*How many rentals were returned late in December?*

*Which customers have rented both versions of Father of the Bride?*

*Which customers have generated the most revenue for the store?*

*Which neighborhoods have generated the most revenue for the store?*

Over the next few weeks you work with Seymour to carefully scope out the extent of the information to be included in the delivered system, then proceed to design and implement it.

### A.2.2 Part-Supplier Database

Your friend Wei Tu Fun is going to automate the family business. You have been enlisted to establish a reliable database to support day-to-day business operations. You have agreed to help refine the requirements and implement a database management for the new enterprise. Wei has told you that the enterprise sells goods to customers and keeps an inventory of items on hand. Multiple suppliers supply goods to the enterprise. Each item in the store can be ordered from multiple suppliers, and each supplier can supply multiple items, although not all items are ordered from every supplier and each supplier does not supply all items. Items in your store have both regular and sale prices. Each purchase from a supplier is for only a single item. The warehouses contain seasonal items. The company sells about 700 items throughout the year, but at any given time stocks only about 400.

Clearly, Wei needs help identifying what information needs to be included in the final system. You asked Wei to clarify needs by supplying a list of questions the database should be able to answer. You received the following:

*Which supplier supplied the good purchased by Mr. Jones?*

*Which warehouses have teak lawn chairs in stock?*

*How many soccer balls are in the Vashon Island warehouse?*

*What is the date of Mr. Cobain's most recent order?*

*List the items on order number 6297.*

*Were any CDs obtained from Amoeba Records in Berkeley or the Virgin Megastore in Seattle?*

*Has order 2391 been filled?*

*Which items are only stocked in summer?*

*Which parts are in the aquarium kit?*

*How many kits include the Whoosh 926 aquarium filter?*

Over the next few weeks you work with Wei to carefully scope out the extent of the information to be included in the delivered system, then proceed to design and implement it.

### A.3. Project Due Dates

This project is broken into a number of deliverables as listed in table A-1. The table displays the points allocated for each deliverable as well as its due date.

Table A-1. Project deliverables and due dates.

Deliverable	Assignment	Date Due	Week
#1	Describe the enterprise	12 SEP 2013	3
#2	Define the environment	26 SEP 2013	5
#3	Develop logical and conceptual models of your enterprise	10 OCT 2013	7
#4	Define the database and formulate queries	24 OCT 2013	9
#5	Consider database integrity and security	7 NOV 2013	11
#6	Implement the database	21 NOV 2013	13
#7	Lessons learned	5 DEC 2013	15
final	Completed notebook, upload or medium containing design, tables, queries, traces - everything needed to build, run, and test the project	5 DEC 2013	15

Note: The *Lessons learned* section *is not optional*. Any project without this piece will be considered late until it is submitted.

# **VESLL Corporation**

## **Database Project**

This project designs and implements a database to track projects and billing records for the VESLL Corporation.

**CMSI 486 Enterprise Project Fall 2013**

Seymour Muni

12 September 2013

**PROJECT TITLE: Table of Contents**

**II. Table of Contents**

III. Copy of Assignment .....	III-1
IV. Description of the Enterprise .....	IV-1
V. Definition of the Environment .....	V-1
VI. Enterprise Database Design .....	VI-1
VII. Database and Query Definition.....	VII-1
VIII. Database Integrity and Security .....	VIII-1
IX. Implementation Notes .....	IX-1
X. Lessons Learned.....	X-1
XI. Database Development History .....	XI-1

**PROJECT TITLE: Copy of Assignment**

**III. Copy of Assignment**



**PROJECT TITLE: Description of the Enterprise**

**IV. Description of the Enterprise**

**PROJECT TITLE: Definition of the Environment**

**V. Definition of the Environment**

**PROJECT TITLE: Enterprise Database Design**

**VI. Enterprise Database Design**

**PROJECT TITLE: Database and Query Definition**

**VII. Database and Query Definition**

**PROJECT TITLE: Database Integrity and Security**

**VIII. Database Integrity and Security**

**PROJECT TITLE: Implementation Notes**

**IX. Implementation Notes**

**PROJECT TITLE: Lessons Learned**

**X. Lessons Learned**

**PROJECT TITLE: Database Development History**

**XI. Database Development History**





# **VESLL Corporation**

## **Database Project**

This project designs and implements a database to track projects and billing records for VESLL.

**CMSI 486 Enterprise Project Fall 2013**

Wei Tu Fun

26 September 2013

### **VII.3. Review Sign-off Sheet.**

I have reviewed the design and code for this database. My comments are as follows:

Reviewed by: \_\_\_\_\_<Reviewer's Signature>\_\_\_\_\_

<Reviewer's Name>

Date