RECEIVED
FEB - 8 2001
LIBERAL STUDIES

LSC Use Only
Number: ______
Submission Date: ______
Action-Date: ______

UWUCC USE Only
Number: 01-12e ~~00-55e~~
Submission Date:
Action-Date: App ~~4/17/01~~
Senate App 12/4/01

CURRICULUM PROPOSAL COVER SHEET
University-Wide Undergraduate Curriculum Committee

I. **CONTACT**

Contact Person Charles Shubra Phone 357-7917

Department Computer Science

II. **PROPOSAL TYPE (Check All Appropriate Lines)**

____ **COURSE** Introduction to DBMS
Suggested 20 character title

____ New Course* ______
Course Number and Full Title

X Course Revision COSC 441 Introduction to Database Management Systems
Course Number and Full Title

____ Liberal Studies Approval+ for new or existing course ______
Course Number and Full Title

____ Course Deletion ______
Course Number and Full Title

X Nr and/or Title Change COSC 441 Introduction to Database Management Systems
Old Number and/or Full Old Title

COSC 341 Introduction to Database Management Systems
New Number and/or Full New Title

____ Course or Catalog Description Change ______
Course Number and Full Title

Rev.

RECEIVED
NOV 19 2001
LIBERAL STUDIES

____ **PROGRAM:** ____ Major ____ Minor ____ Track

____ New Program* ______
Program Name

____ Program Revision* ______
Program Name

____ Program Deletion* ______
Program Name

____ Title Change ______
Old Program Name

______
New Program Name

III. **Approvals** (signatures and date)

Tia Watts
Department Curriculum Committee

G. L. Buterbaugh
Department Chair

02/08/01
College Curriculum Committee

College Dean

+Director of Liberal Studies (where applicable)

*Provost (where applicable)

Part II. Description of Curriculum Change

1. New Syllabus of Record

See Attachment A the new syllabus of record.

2. Summary of the proposed revision.

The principal revision is to include subject matter presented in the COSC 315 course in the COSC 441 course. Adjustments in the description and syllabus are contained in Attachment A, the new syllabus of record.

3. Justification for the revision

This is a component in a series of changes necessitated by the ascent of data base technology to primary importance as the technology for the organization, storage and retrieval of large collections of information. Prior to this ascent integrated file systems constructed using indexed sequential access methods served this purpose and that technology was the subject of the COSC 315 course. The faculty in consultation with our Corporate Advisory Board formulated a change in the curriculum. The COSC 315 course will become inactive. Some of the subject matter will migrate to the COSC 220 course other material will migrate to the revised COSC 441 which will be renumbered as COSC 341. The movement of the existing COSC 441 course from the senior year to the sophomore/junior year necessitates the number change. The material, which is being moved deals with hardware device characteristics, the role of the operating system (specifically the file system) in data handling and the details of the indexed sequential access methods. Movement of the COSC 441 course to earlier in the student's coursework is required because of the need for computer science interns to work with data base technology. All of the interns had to work with databases but only about 50% of the interns had taken the course prior to the internship experience. Further, knowledge of databases was recommended for the COSC 319 Software Engineering Concepts and the COSC 320 Software Engineering Practice courses.

4. Old Syllabus of Record

See Attachment B the old syllabus of record.

5. Letters of Support

N/A

I. Catalog Description

COSC 341 Introduction to Database Management Systems

3 credits
3 lecture hours
0 lab hours
(3c-01-3sh)

Prerequisites: COSC 220 (or equivalent) and 310 or permission of instructor

Study of data base concepts. Detailed study of information concepts and the realization of those concepts using the relational data model. Practical experience gained designing and constructing data models and using SQL to interface to both multi-user DBMS packages and to desktop DBMS packages.

II. Course Objectives

1. To apply knowledge of internal data structures and extend this knowledge to external data structures necessary for file management.
2. To examine how data storage technology (hardware and access methods) works.
3. To extend integrated file systems to the data base concept of data storage and retrieval.
4. To define the terminology embodied in data base systems and that used with specific software packages.
5. To develop an understanding of the relational data model theory.
6. To develop an understanding of standard SQL in terms of relational data model theory using relational algebra or relational calculus.
7. To study standard SQL and compare the standard with the SQL implementation available with existing DBMS software packages.
8. To study the theory behind the relational approach to DBMS and the impact of this theory on data model design and validation (Normalization process).
9. To gain practical programming experience with a multi-user relational DBMS package such as Oracle or SQL Server.
10. To gain practical programming experience with a microcomputer based data base package such as Microsoft Access, Fox Pro, Paradox.
11. To investigate client/server technologies as they pertain to databases e.g. ODBC.

III. Detailed Course Outline

1. Introduction to Database Concepts and Architecture. (3 hrs.)
Characteristics, advantages and implications of the database approach to information systems as contrasted with traditional integrated file systems. DBMS architecture including employing super data structures to implement relationships among records. The database system environment including data models, schemas, database languages and interfaces.

2. Record Storage and Primary File Organizations. (3 hrs.)
Characteristics of secondary storage devices. Blocking and buffering as techniques to lessen the impact of physical I/O. The role of the operating system in secondary storage organization and access.

3. Trees and Indexed Access Methods. (3 hrs.)
A quick review of trees. Introduction of the B+Tree as the basis of indexed sequential access methods. Insertion, deletion and retrieval using the B+Tree. Indexes on multiple keys.

4. Information Science Concepts and Data Modeling Using Entity Relationship Diagrams.(4 hrs.)
Information analysis to identify query keys, candidate keys, entities, attributes, relationships and integrity constraints. ER modeling as a means of representing information concepts. Extended entity relationship modeling as it relates to specialization, generalization and inheritance.

5. Exam 1 (1 hr.)

6. The Relational Data Model and Relational Algebra. (6 hrs.)
Relational model concepts. Definition and use of the 8 relational algebra operations to query a relational database. Defining a relational schema using an ER diagram.

7. SQL The Relational Database Standard. (4 hrs.)
Use of SQL to define a relational data model. SQL as an implementation of the relational algebra operations. Basic and complex queries in SQL. Insert, delete and update statements in SQL. Defining and using Views in SQL.

8. Exam 2 (1 hr.)

9. Practical Experience with a DBMS Product that Implements the Relational Approach and SQL. (7 hrs.)
In-depth introduction to an existing DBMS package that implements the relational approach and SQL such as ORACLE, MS SQL Server or VaxRdbSql. Students acting as DBA's will create a data model using the SQL language. Again, using the SQL language students will Insert, Update and Delete data values from the database. Students will then implement complex queries to retrieve data from the database.

10. Nonnalization as a Process for Verification of Data Model Design (3 hrs.)
Definition of functional dependency, full functional dependency, transitive dependency and multi-valued dependency. Definition of the normal forms from unnormalized through 4th normal form and how to apply the normalization process to recognize normal

forms. How to move a data model to a higher normal form and the issues of denormalization as it applies to retrieval performance.

11. SQL Interaction with Third Generation Programming Languages. (4 hrs)
Methods for extending data base functionality to third generation programming languages such as Cobol. Description of precompliers, subroutine calls, module level language and dynamic SQL. Students will implement a program written in a third generation programming language, which provides insertion, deletion, updating and query capabilities to a database.

12. Desk Top DataBase Packages. (3 hrs)
Presentation of a desktop database package as an implementation of a relational DBMS such as MS Access, Fox Pro or Paradox. Students will construct a data model and database using the package. Students will then implement numerous queries using the IDE of the package.

IV. Evaluation Methods

The final grade is determined as follows:

Exam 1 and 2	200 points
Final	100 points
Homework	100 points
SQL Online Project	50 points
SQL Embedded Project	50 points
Micro DBMS Project	50 points
	550 points

The lines for the course grades are as follows:

Total Possible Points

100 - 90% - A
89 - 80% - B
79 - 70% - C
69 - 60% - D
59 - 0% - F

Suggested Assignments and Projects:

Homework 1: Given Application Statements Design Entity-Relationship Diagram (1 week)
Students are supplied with a 5-6 paragraph description of an application and they then identify components of a data model (entity, attributes, relationships and constraints) as present in the description. Students then represent the data model using an ER diagram. Discussions develop concerning what is an entity vs an attribute. Attributes of relationships are also encountered and how these are to be implemented depending on the cardinality of the relationships. Extended ER diagram components such as super class and subclass also are discovered.

Homework2: Given the Cobol Source Code Describing the File Environment for an Integrated File Application Design an Entity-Relationship Diagram (1 week)
The record descriptions (in the form of Cobol source code) of an integrated file application are provided to the students. They then identify components of a data model as present in this source code. Students then represent the data model using an ER diagram. Naming conventions in the Cobol source mislead students and they have to rely on the functional dependencies to determine good design. Discussions develop concerning why tables are embedded within records and how many to many relationships are implemented in integrated file systems.

Homework3: Given a Set of Queries Write the Relational Algebra Needed to Provide Answers (1 week)
Students use the 8 relational algebra operations to answer a series of complex queries. Students realize that many alternative solutions are possible.

Homework4: For the Same Set of Queries Write Standard SQL Queries (1 week)
Students learn how the 8 relational algebra operations are present in standard SQL while learning alternative methods of answering queries.

Project1a: Using a Non-Desktop DBMS Implement the Data Model (1 week)
Using a mainframe or client/server database construct the domains, tables and views necessary to implement a data model. Syntax of the DBMS is compared to that described in the SQL Standard. The data model will contain attributes, various data types, domains, constraints and other clauses. Students then show the meta-data present in the data model.

Project1b: Using a Non-Desktop DBMS Insert Data Values into the Tables (1 week)
Various methods of placing values into the base tables are discussed including import utilities, SQL Insert statements, stored procedures and Writing a custom program. Discussions of data types and problems with integrity constraints occur. The importance of "Commit" and "Rollback" are emphasized. Select statements are used to demonstrate the correct contents of the tables.

Project1c: Using a Non-Desktop DBMS Implement SQL Query Solutions (1 week)
Students enter the SQL Select statements from Homework 4 and determine if the answers are correct.

Project2: Using SQL with Third Generation Languages (2 weeks)
Students write an interactive menu driven program, which uses SQL to interact with their database. The SQL mechanisms needed such as cursors are discussed and implemented. The mechanisms used to allow a 3gl programming language to interact with a DBMS are also discussed. These include embedded SQL using a precompiler, a call level interface or API, dynamic SQL and module level language. Students are required to answer several queries some preplanned some parameterized. One query requires the use of a cursor. The program also requires Update, Delete and Insert operations with logic to deal with DBMS objections to violations or integrity constraints.

Homework5: Apply the Normalization Process to a Data Model (1 week)
Given a data model, apply the normalization process to validate the design.

Project3: Implement the Data Base Using a Desk Top DBMS (1 week)
Students utilize a GUI interface to define a data model. Data values are then imported from fixed width text files into the base tables. Numerous discussions result concerning data type mismatches. Students then use the GUI to implement queries. The differences between the DBMS and the relational data model and the SQL standard are discussed.

V. Required textbooks, supplemental books and readings

1. Codd, E. F., "Relational Database: A Practical Foundation for Productivity", Communications of the ACM, Vol. 25, No. 2, February, 1982, pp. 109- 117.

2. Elmasri, R., Navathe, S., Fundamentals of Database Systems, Third Edition, Addison-Wesley, 2000.

3. Kent, William, "A Simple Guide to Five Normal Forms in Relational Database Theory", Communications of the ACM, Vol. 26, No. 1, February 1983, pp. 120-124.

VI. Special resource requirements

None.

VII. Bibliography

1. Castano S., Fugini M., Martella G., and Samarati P., *Database Security*, New York, N.Y.: ACM Press/reading, Mass.: Addison-Wesley (1995).
2. Celko, J., *SQL for Smarties: Advanced SQL Programming*, San Francisco, Calif.: Morgan Kaufmann (1995).
3. Codd, E.F., "Domains, Keys, and Referential Integrity in Relational Databases," *InfoDB* 3, No. 1 (Spring 1988).
4. Date, C.J., *Relational Database Writings 1991-1994.* Reading, Mass.: Addison-Wesley (1995).
5. Date, C.J. and Darwen H., *A Guide to the SQL Standard (4th edition)*, Reading Mass.: Addison-Wesley (1997).
6. Date, C.J., "Three-Valued Logic and the Real World," in C. J. Date and Hugh Darwen, *Relational Database Writings 1989-1991.* Reading, Mass, Addison-Wesley (1992).
7. Gulutzan P., Pelzer T., *SQL-99 Complete*, Lawrence, Kan.: R&D Books Miller Freeman, Inc. (1999).
8. Lempel A., "Cryptology in Transition," *ACM Comp. Surv. 11, No. 4*, Special Issue on Cryptology (December 1979).
9. Loney K., Theriault M., *Oracle 8i DBA Handbook*, Berkeley, Calif.: Osborne/McGraw-Hill (2000).
10. Loomis M.E., *Object Databases The Essentials*, Reading Mass. Addison-Wesley (1995).
11. Melton, J. and Simon A.R., *Understanding the New SQL: A Complete Guide*, San Mateo, Calif.: Morgan Kaufmann (1993).
12. U.S. Department of Commerce/National bureau of Standards, *Data Encryption Standard*, Federal Information Processing Standards Publication 46 (January 1977).

Indiana University of Pennsylvania
Computer Science Department

Dr. Shubra

COSC 441 - Data Base Management

Introduction

This three credit upper level course is being offered to provide interested motivated students with a fundamental knowledge of, and a practical experience with, the database concept. The course is a logical extension of the concepts taught in COSC 315 Large File Organization and Access, COSC 310 Data Structures and COSC 220 Applied Computer Programming.

The course texts are:

Required:

Elmasri, R., Narathe, S., Fundamentals of Database Systems, Third Edition, Addison-Wesley, 2000.

Codd, E. F., "Relational Database: A Practical Foundation for Productivity", Communications of the ACM, Vol. 25, No. 2, February, 1982, pp. 109- 117.

Kent, William, "A Simple Guide to Five Normal Forms in Relational Database Theory", Communications of the ACM, Vol. 26, No. 1, February, 1983, pp. 120-124.

Objectives:

1. Extend integrated file systems to the data base concept of data storage and retrieval.
2. Define the terminology embodied in data base systems and that used with specific software packages.
3. Develop an understanding of the three fundamental approaches to a data base model. Namely, relational, hierarchical, and network.
4. Study end-user query languages available with existing DBMS software packages and categorize these languages using relational algebra or relational calculus theory.
5. Study the theory behind the relational approach to DBMS and the impact of this theory on data model design.
6. Gain practical programming experience with the relational DBMS package available on the IUP Dec Vex System, i.e., VAX/RDB/SQL.
7. Gain practical programming experience with a microcomputer based data base package such as Microsoft Access, DBASE III PLUS, Paradox, Oracle, etc.
8. Investigate client/server technologies as they pertain to databases e.g. ODBC.

Tentative Course Schedule where:

E = Elmasri's Text
Codd = E. F. Codd paper
Kent = William Kent paper

Week	Topic	Reading
1 - 2	Course Introduction, DBMS Concepts, ER Model	E1, E2, E3
3 - 4	Relational Data Model Relational Algebra	Codd, E7
5 - 6	Exam 1 SQL Standard	E1-3,Codd,E7 E8
7 - 8	VAX/RDB/SQL SQL Project (online)	Handouts
9	Exam 2 Embedded SQL Embedded SQL Project	E8, Vax/Rdb/SQL E10.5
10	Normalization & Data Base Design	E14, E15 Kent
11-12	Microcomputer DBMS Microsoft Access Project	E10.7-E10.9
13	Emerging Trends	E4, E10.1-E10.6, E17, E19, E20

Grading:

Exam 1 and 2	200 points
Final	100 points
Homework	50 points
SQL Online Project	50 points
SQL Embedded Project	50 points
Micro DBMS Project	50 points
	500 points

At worst, I will draw the line for the course grades as follows:

Total Possible Points

100 - 90%	-	A
89 - 80%	-	B
79 - 70%	-	C
69 - 60%	-	D
59 - 0%	-	F

Because of the need to demonstrate software packages and to have hands-on exercises, the class will not always be held in the originally scheduled room. Room changes will be announced in class.