Original
LBST → 04/21/10

Course Revision –Computer Science Curriculum

10-26

LSC Use Only No: LSC Action-Date:	**UWUCC USE Only** No. UWUCC Action-Date: Senate Action Date: AP-10/12/10 App-2/22/11

Curriculum Proposal Cover Sheet - University-Wide Undergraduate Curriculum Committee

Contact Person: **David T. Smith**	Email Address: **dtsmith@iup.edu**
Proposing Department/Unit: **Computer Science**	Phone: **7-4478**

Check all appropriate lines and complete information as requested. Use a separate cover sheet for each course proposal and for each program proposal.

1. Course Proposals (check all that apply)

_____New Course _____Course Prefix Change _____Course Deletion

_X_Course Revision _____Course Number and/or Title Change _____Catalog Description Change

Current Course prefix, number and full title	***Proposed course prefix, number and full title, if changing***
COSC 341 Intro to Database Management Systems	

2. Additional Course Designations: check if appropriate

_____ This course is also proposed as a Liberal Studies Course.

_____ This course is also proposed as an Honors College Course.

_______ Other: (e.g., Women's Studies, Pan-African)

3. Program Proposals ______Catalog Description Change _____Program Revision

_____New Degree Program ______Program Title Change _____Other

_____New Minor Program ______New Track

Bachelor of Science- Computer Science/ all tracks **Bachelor of Arts- Computer Science** ***Current program name***	***Proposed program name, if changing***

4. Approvals		**Date**
Dept Curriculum Committee Chair	Waleed Farag	3/24/2010
Department Chair		3/25/10
Coll. Curriculum Committee Chair	Anne Kondo	4/20/10
College Dean		4/20/10
Director of Liberal Studies *		
Director of Honors College *		
Provost *		
Additional signatures as appropriate:		
(include title)		
UWUCC Co-Chairs	Gail S Sechrist	10/13/10

* where applicable

Course Revision –Computer Science Curriculum

Part II. Description of Curriculum Change

1. New Syllabus of Record

I. Course Description

COSC 341 Introduction to Database Management Systems **3c-0l-3cr**

Prerequisites: COSC 110 and COSC 210

Provides fundamental knowledge of, and practical experience with, database concepts. Includes 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 Outcomes

Upon successful completion of the course, the student will be able to:

1. Differentiate database systems from file systems by enumerating the features provided by database systems and describe each in both function and benefit.
2. Define the terminology, features, classifications, and characteristics embodied in database systems.
3. Analyze an information storage problem and derive an information model expressed in the form of an entity relation diagram and other optional analysis forms, such as a data dictionary.
4. Demonstrate an understanding of the relational data model.
5. Transform an information model into a relational database schema and to use a data definition language and/or utilities to implement the schema using a DBMS.
6. Formulate, using relational algebra, solutions to a broad range of query problems.
7. Formulate, using SQL, solutions to a broad range of query and data update problems.
8. Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.
9. Use an SQL interface of a multi-user relational DBMS package to create, secure, populate, maintain, and query a database.
10. Use a desktop database package to create, populate, maintain, and query a database.
11. Demonstrate a rudimentary understanding of programmatic interfaces to a database and be able to use the basic functions of one such interface.

III. Detailed Course Outline

Academic Hours

1. Introduction to Database. (1½ hrs.)
   Characteristics, advantages and implications of the database approach to information systems as contrasted with traditional integrated file systems. DBMS architecture. History of database systems. Roles involved with database systems.

2. Database Concepts and Architecture. (1½ hrs.)
   The database system environment including data models, schemas, database languages and interfaces. Three-schema architecture and data independence. Classifications of database systems.

3. 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.

4. The Relational Data Model. (4 hrs.)
   Relational model concepts. Referential integrity, entity integrity, and other constraints. Defining a relational schema from an ER diagram.

5. Exam 1. (1 hrs.)

6. The Relational Algebra (and brief introduction to Relational Calculus). (6 hrs.)
   Definition and use of relational algebra operations to query a relational database.

7. SQL. (7 hrs.)
   Use of SQL to define a relational data model. Basic and complex queries in SQL. Insert, delete and update statements in SQL. Defining and using Views in SQL. Implement security with Grant/Revoke.

8. Exam 2. (1 hrs.)

9. Practical Experience with a Relational DBMS providing SQL. (6 hrs.)
   In-depth introduction to an existing DBMS package that implements the relational approach using SQL, such as ORACLE or MS SQL Server. Students acting as DBA's will create a data model using the SQL language. Using SQL students will Insert, Update and Delete data values from the database. Students will then implement complex queries to retrieve data from the database.

10. Normalization as a Process for Verification of Data Model Design. (4 hrs.)
    Definition of functional dependency, full functional dependency, transitive dependency and multi-valued dependency. Definition of the normal forms from un-normalized 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 de-normalization as it applies to retrieval performance.

11. SQL Interaction with Programming Interfaces. (3 hrs.)
    Methods for extending database functionality to programming languages such as COBOL or Java. Description of pre-compliers, dynamic SQL, ODBC, and JDBC. Students will gain introductory experience with a programmatic interface to insert, update, delete and query data in 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. Students will construct a data model and database using the package. Students will then implement queries using the IDE of the package.

Total (42 hrs.)

Final (2 hrs.)

IV. Evaluation Method

Grade for the course will be determined as follows:

Exams	
Exam 1	100 points
Exam 2 (or 4 Quizzes)	100 points
Final	100 points
Assignment/Project Option	
Data Modeling	100 points
DBMS Implementation/Queries	100 points
-OR-	
Case Study Option	
ER Diagram	50 points
DB Schema	50 points
Load	20 points
Base Queries	30 points
Program Interface/or alternative	50 points
Homework Assignments	
Various (10 – 25 points each)	100 points
Total	600 points

Grade is assigned by and applying the following scale:

540 and above	(90-100%)	A
480 – 539	(80-89%)	B
420 – 479	(70-79%)	C
360 – 419	(60-69%)	D
359 and below	(< 60%)	F

Attendance Policy

Attendance is crucial to success in this course. To encourage class attendance, the following policy will be used: Attendance will be taken at every class. For each unexcused absence, starting with the fourth, 2% will be deducted from the overall class grade. Generally, excused absences involve illness with a doctor's excuse, verifiable family emergencies, or conflicting university activity.

Suggested Assignments and Projects:

Homework 1: Given Application Statements Formulate Entity-Relationship Diagram (1 week)
Students are supplied with a 5-6 paragraph description of an application and they analyze and indentify components of an information 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.

Homework 2: Transform an Entity Relation Diagram into a Relational Schema (1 week)
Given an Entity Relation Diagram, the students transform the diagram into a relational schema. Discussions develop on various design decision that need to be made to overcome deficiencies in the target model .

Homework 3: Given a Set of Query Problems Formulate Solutions using Relational Algebra (1 week)
Students use the 8 relational algebra operations to answer a series of complex queries. Students realize that many alternative solutions are possible.

Homework 3: Given a Set of Query Problems Formulate Solutions using SQL (1 week)
Students learn how the 8 relational algebra operations are present in standard SQL while learning alternative methods of answering queries.

Homework 4: Apply the Normalization Process to a Data Model (1 week)
Given a data model, apply the normalization process to validate the design.

Project 1: Using an Enterprise level DBMS Implement a Relational Schema (1 week)
Using an enterprise level DBMS construct the domains, tables and views necessary to implement a relational schema. 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.

Project 2: Using an Enterprise level DBMS Implement 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.

Project 3: Using an Enterprise level DBMS Implement SQL Query Solutions (1 week)
Students enter the SQL Select statements from Homework 4 and determine if the answers are correct.

Project 4: Programmatic Interfaces (2 weeks)
Given a completed working application that is using a façade simulating a database, replace the façade with code to connect the application to a database. Students gain experience with design decisions needed to overcome the impedance mismatch between procedural languages and a relational database. This task may be accomplished using embedded SQL, standards based API, stored procedures, or combination thereof.

Project 5: Implement the Database 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

Required Textbook:

1. Elmasri, R., Navathe, S., *Fundamentals of Database Systems*, Fifth Edition, Addison-Wesley, 2006.

Required Reading of Seminal Papers:

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

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

Access to an Enterprise DBMS. Educational program offered by Oracle Corporation is currently in place. Hardware on which Oracle is installed is in place, operational, and accessible by an Oracle client over the internet. The Oracle client is available to download by students free of charge.

VII. Bibliography

1. Castano S., Fugini M., Martella G., and Samarati P., *Database Security*, New York, NY: ACM Press/reading, Mass.: Addison-Wesley (1995).
2. Cattell R., Barry D., Eastman J., Jordan D., Russell C. et. al., *The Object Data Standard ODMG 3.0*, San Diego, CA: Morgan Kaufmann (2000).
3. Celko, J., *SQL for Smarties: Advanced SQL Programming*, San Francisco, CA: Morgan Kaufmann (1995).
4. Date, C.J., *Relational Database Writings 1991-1994*. Reading, MA: Addison-Wesley (1995).
5. Date, C.J. and Darwen H., *A Guide to the SQL Standard*, fourth edition), Reading MA: 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, MA: Addison-Wesley (1992).
7. Garcia-Molina H., Ullman J., and Widom J., *Database Systems The Complete Book*, Upper Saddle River, NJ: Prentice Hall (2002).
8. Gulutzan P., Pelzer T., *SQL-99 Complete*, R&D Books Miller Freeman, Inc. (1999).
9. Loney K., Theriault M., *Oracle 10g DBA Handbook*, Oracle Press (2005).
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, CA: Morgan Kaufmann (1993).
12. Microsoft, *ODBC 2.0 Programmer's Reference and SDK Guide*, Microsoft Press (1994).
13. "Data Encryption Standard", *Federal Information Processing Standards Publication 46-3*, Washington: National bureau of Standards (1999).

2. Summary of Proposed Revisions

Course objectives and detailed course outline have been updated to reflect the removal of content related to internal data structures and file management.

The course objectives have further been enhanced to provide clarification and use measures targeting higher levels in Bloom's taxonomy.

Additional minor changes have been made to content, suggested assignments and projects, and bibliography to keep current and in-line with professor's experience over the last four semesters. The reflections of two professors have been incorporated.

Security features have been introduced at various points in the course.

Oracle support using the in-place education support program offered by Oracle has been added to the Special resource requirements section.

A correction has been made to the pre-requisites.

3. Justification for Revision

Given the adoption of a 50 minute time period, there is insufficient time to cover all content. The content removed has not been covered in the last four semesters. While the removed content has value, it is not needed in meeting the current Computer Science program objectives. There are no subsequent courses that are dependent on the removed content. Portions of the removed content may be considered for inclusion in a new course positioned as an upper level elective.

Security has become a growing concern in software industry and security within databases is one point of enforcement. Adding discussions of security at appropriate points in the course will strengthen the value of this course as well as the overall Computer Science program. Furthermore, the addition of security will strengthen our position as an NSA Center of Academic Excellence.

Inclusion of text in the special resource requirements section clarifies that students will have access to the required resources needed for the course.

A previous course revision removed the pre-requisite COSC 310 without replacing it with COSC 210, the pre-requisite of COSC 310. While the removal of COSC 310 was correct, it was intended to be replaced with COSC 210. This omission was an error. COSC 210 is now been added. The "or permission of instructor" was also in error as this referred to COSC 310.

4. The Old Syllabus of Record

I. Catalog Description

COSC 341 Introduction to Database Management Systems 3c-0l-3sh

Prerequisites: COSC 110 or permission of instructor

Study of database 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 database concept of data storage and retrieval.

4. To define the terminology embodied in database 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 database 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. Normalization 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 database 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 Database 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).

Part II. Letters of Support

Not Applicable. This is an internal change to the Computer Science program. Affected programs are all tracks in the Computer Science major, the Computer Science minor, and the Information Assurance Minor.

Indiana University of Pennsylvania
MIS and Decision Sciences Department
203 Eberly College of Business & Information Technology

Date: February 2, 2011

To: Dr. Terry Fries, Chair
Computer Science Department Curriculum Committee

From: Kustim Wibowo, Chair kw
MIS and Decision Sciences Department

RE: COSC 341 Introduction to Database Management Systems Revision

The MIS and Decision Sciences Department is pleased to support the proposed revisions to COSC 341 Introduction to Database Management Systems. This course revision will have no effect on our student population since the MIS program requires the course IFMG 390 Database Theory and Practice for the MIS majors.