Course Change Proposal
Form A

Academic Group (College): Engineering and Computer Science
Academic Organization (Department): Computer Science
Type of Course Proposal: New __ Change x Deletion __

Date: 1/26/15
Submitted by: Ying Jin
Department Chair: Cui Zhang
Semester Effective: Fall __ Spring __, 2015_

For Catalog Copy: Yes _X_ No __
CCE (Extension): Yes _X_ No __

This course replaces experimental course Subject Area (prefix) and Catalog Nbr (course number):

If changing an existing course, should new version be considered a repeat of the original version? If so, the same Course ID will be maintained. If not, a new Course ID will be assigned. Note: In PeopleSoft terminology, the Course ID is the unique system identifier, not the Catalog Nbr.

Yes _X_ No __

Change from:
Subject Area (prefix) & Catalog Nbr (course no.): Title: Database Management and File Organization Units: 3
CSC 134

Change to:
Subject Area (prefix) & Catalog Nbr (course no.): Title: Database Management Systems Units: 3
CSC 134

JUSTIFICATION:
The names of the three database courses (CSC 134, 174, 176) have been changed along with this proposal. "File Organization" has been removed from the title of this course to better reflect the course materials. This course covers the file organization for physical layer of database management systems which is part of the database management system. The course content has been updated to current practice. The names and catalog descriptions have been updated to reflect these changes. The new course description describes the course content in details with updated terminologies.

SQL, as the standard language to programming any relational databases, has become more important, therefore more SQL materials are covered, including schema definition, constraints, and queries compared to "introduction to SQL" in the old catalog description.

To have sufficient time for SQL, "information interchange and XML" and "database processing on the Web" have been removed. The database programming part of "database processing on the web" has been moved to CSC 174 and CSC 176. The materials removed from this course do not belong to the core courses.

"elements of commercial database management systems" has been removed, because it is largely redundant, given the rest of the course description (the content of it has been spread and covered by the rest of the course description).

ER to relational model mapping and relational algebra have always been important in the past and currently, so they are being added in the catalog description.

"File systems, storage structures and access methods" have been replaced with "access methods such as indexing and hash structure", which is more specific on the topics covered, and is more appropriate for a database course.

NEW COURSE DESCRIPTION: (Not to exceed 80 words and language should conform to catalog copy. See Guidelines for Catalog Course Description http://www.csus.edu/umual/AcadAff/ESC0060.htm)
Entity-Relationship (ER) model; relational model; relational database design by ER-to-relational mapping; design of applications using database technology; SQL: schema definition, constraints, and queries; relational algebra; data normalization; access methods such as indexing and hash structures; introduction to transaction processing.
Prerequisite: CSC 130. 3 units.

Note:
Prerequisite: CSC 130
Enforced at Registration: Yes _X_ No __
Corequisite:
Enforced at Registration: Yes No __
FOR NEW COURSE PROPOSALS OR SUBSTANTIVE CHANGES ONLY:

Description of the Expected Learning Outcomes: Describe outcomes using the following format: “Students will be able to: 1), 2), etc.”

Students will be able to:
1) Analyze a database design in Normal Forms
2) Model an application in an Entity-Relationship diagram
3) Map an Entity-Relational model design to a design in relational model
4) Create a relational database using SQL
5) Write database queries in SQL
6) Explain how database files are organized and accessed

**Attach a list of the required/recommended course readings and activities [Note: it is understood that these are updated and modified as needed by the instructor(s).] This attachment should be forwarded only to your Dean’s office, not Academic Affairs.

Assessment Strategies: A description of the assessment strategies (e.g., portfolios, examinations, performances, pre and post tests, conferences with students, student papers) which will be used by the instructor to determine the extent to which students have achieved the learning outcomes noted above:

1) Examinations
2) Homework with programming assignments
3) Quizzes

For whom is this course being developed?

Majors in the Dept_x_Majors of other Depts__Minors in the Dept_x_General Education__Other__

Is this course required in a degree program (major, minor, graduate degree, certificate)? Yes_x_No__

If yes, identify program(s): Computer Science, B.S.

Does the proposed change or addition cause a significant increase in the use of College or University resources (lab room, computer facilities, faculty, etc.)? Yes__No_x__

If yes, attach a description of resources needed and verify that resources are available.

Indicate which department or programs will be affected by the proposed course (if any): Computer Science

The Department Chair's signature below indicates that affected programs have been sent a copy of this proposal form.

Accessibility: Following course approval, and prior to the start of the semester in which the new or revised course will be taught for the first time, an accessibility checklist [available at http://www.csus.edu/accessibility/checklist.html] shall be completed and submitted to the appropriate Dean’s office. An accessible syllabus shall also be made available online, preferably prior to the start of that semester’s open registration period.

Approvals: If proposed change, new course or deletion is approved, sign and date below. If not approval, forward without signing to the next reviewing authority, and attach an explanatory memorandum to the original copy.

Signatures:

Department Chair: ____________________ Date: 2/6/2015

College Dean or Associate Dean: ________________ 2/6/15

University Committee:

Assoc. Dean for Undergraduate Studies
OR Dean of Graduate Studies:

Distribution: Academic Affairs (original), Department Chair and College Dean. Dean’s office to send original after approval to Academic Affairs, at mail zip 6016.
CSC 134: Database Management Systems

Credits and contact hours: 3 semester units

Course coordinator: Ying Jin


Supplemental Material: None

Catalog Description: Entity-Relationship (ER) model; relational model; relational database design by ER-to-relational mapping; design of applications using database technology; SQL: schema definition, constraints, and queries; relational algebra; data normalization; access methods such as indexing and hash structures; introduction to transaction processing.

Prerequisite: CSC 130. 3 units.

Course Type: Required

Course Outcomes: Students completing this course will be able to
1. Understand and use relational databases;
2. Demonstrate competence in using SQL.

Student Outcomes: This course contributes to the following student outcomes. Graduates should be able to
a. Apply fundamental knowledge of mathematics, algorithmic principles, computer theory, and principles of computing systems in the modeling and design of computer-based systems that demonstrate an understanding of tradeoffs involved in design choices;
  d. Use current skills, techniques, and tools necessary for computing practice.

Brief List of Topics:
1. Introduction to database system concepts and architecture;
2. Entity-Relationship (ER) model;
3. Relational model;
4. Relational database design by ER-to-relational Mapping;
5. Relational algebra;
6. SQL: schema definition, constraints, and queries
7. Functional dependencies and normalization for relational databases;
8. Storage, file structures, and hashing;
9. Indexing structure for files;
10. Introduction to query processing and transaction processing.
Additional Information

Expanded List of Topics:
1. Introduction to database system concepts and architecture: data models, database languages, database management system architecture (1 hours)
2. Conceptual modeling using Entity-Relationship (ER) model: presentation of entities, attributes, and relationships; constructing ER diagrams from narrative specifications (4 hours)
3. Implementation model: relational model and relational database constraints (3 hours)
4. Mapping from conceptual model to implementation model: Relational database design by ER-to-relational Mapping (3.5 hours)
5. Relational algebra: selection, projection, union, intersection, difference, Cartesian product, join, equal join, nature join (6.5 hours)
6. SQL: schema definition, constraints, and queries. Data definition language of SQL. Insert, delete, and update statements. Query specification. Introduction to views. MySQL as a case study for using commercial relational database system (12.5 hours)
7. Normalization for relational databases. Functional dependencies, closures, minimal cover, 1st, 2nd, 3rd normal form, and BCNF. (6.5 hours)
8. Storage, file structures, and hashing (external hashing, dynamic hashing); (3.5 hours)
9. Indexing structure for files: single level index (primary index, clustering index, secondary index), multilevel index, B tree indexing, B+ tree indexing; (3 hours)
10. Introduction to query processing and transaction processing. (1.5 hour)

Prerequisites by Topic:
**Thorough understanding of:**
- Choice of data structures, per application
- Hashing techniques

**Basic Understanding of:**
- Algorithm analysis and computational complexity
- Sets and basic operations on sets

**Exposure to:**
- Basic use of mass storage such as disk

Outcomes:
**Thorough understanding of:**
- Relational model
- Database design:
  - conceptual modeling using ER
  - mapping from ER to implementation model
  - database schema design
  - schema-based constraints
- Query specification using SQL and relational algebra
- Index and hashing for database files
- Functional dependencies and normal forms (1NF, 2NF, 3NF, and BCNF)
Basic Understanding of:
  • SQL view definition
  • Semantic integrity constraints (application-based constraints)

Exposure to:
  • Basic transaction processing concepts

Document status: latest updates 1/10/15 (Ying Jin)
COURSE DESCRIPTION

Dept., Number  CSC 134  Course Title: Database Management and File Organization
Semester hours  3  Course Coordinator: Ying Jin
URL (if any): http://gaia.ecs.csus.edu/~jiny/

Catalog Description
File systems, storage structures, and access methods; data modeling; Entity-Relationship analysis and data normalization; design of applications using database technology; elements of commercial database management systems; introduction to transaction processing; introduction to SQL; information interchange and XML; database processing on the Web. Prerequisite: At least a C-grade in CSC 130 and full CSC, CPE, or MATH/CSC major status.

Textbook

References

Course Goals
1. Introduction to I/O functionality within computer architecture.
2. Study parameters and operational behavior of mass storage, mainly disk.
3. An introduction to relational database model and fundamentals of the SQL query language.
4. Information interchange, XML and Web/Internet information flows.

Prerequisites by Topic
Thorough understanding of:
- Choice of data structures per application.
- Trace and debugging skills used in typical programming language environments.

Basic understanding of:
- Data structures.
- Probability and statistics.
- Algorithm analysis and computational complexity.
- Interpreted/compiled languages and basics of UNIX/Windows file systems.

Major Topics Covered in the Course
1. System component architecture; I/O architectures & I/O operations: controller, bus, device, drivers; physical tape & disk - organization: sector, block, track, cylinder, volume; memory/storage hierarchy (2.25 hours).
2. Disk directories, files, records; physical record formats; record blocking; buffered I/O intro; serial and random record processing; disk scheduling algorithms intro (2.25 hours).
3. File Access Methods (AM) and performance formulas of following organizations: heap AM and sequential AM; indexes – fixed and dynamic AMs; dense, sparse, and secondary indexes; clustering techniques; single and double buffered I/O algorithms; hash AM – static
and dynamic hashes; fast sorting of large files; choosing the appropriate AM for a given application; more file system facilities: concurrency, access rights, record blocking; storage allocation overview; ufs (aka UNIX file system) & ntfs (WINNT+ OSs); hardware parallelism: RAID functionality (CSc 139 has different RAID viewpoint) (11.5 hours).

4. Database models File Management Systems (FMS) comparison with Database Management Systems (DBMS); DBMS models and DB AMs; the transaction paradigm; introduction to transaction processing (tp); ACID properties; commit and rollback functionality (2.5 hours).

5. Entity analysis; ERD models; EERD design intro; constructing ERDs from specifications; EERD extensions of ERD (2.5 hours).

6. Relational DBMS (RDBMS) environments; RDBMS model overview; Oracle – SQL*PLUS functional usage, PL/SQL intro; tables and views; SQL (11.5 hours).

7. UML for DB analysis; DB design; data normalization; UML diagrams for DB analysis and design; ERD to relational schema mapping; data normalization based on key and data-field dependencies (2.5 hours).

8. Database processing on the Web; XML, data exchange; Web/Internet workflows (2.5 hours).

9. In-class quiz, bi-weekly, throughout the semester (2 hours).

Laboratory Projects

1. Programming Assignment 1 – Build data structures for a hierarchical records database.

2. The database consists of a small number of (simulated) disk blocks. Do initial load of records into the database (2 weeks).

3. Programming Assignment 2 – Using the database structure developed in Assignment 1, apply database record transactions to that database. This amounts to implementing a simple query processor. Insert/update and retrieve record operations must model common hierarchical record processing rules. A simple linear index must be traversed when doing record access (2.5 weeks).

4. Written Analysis Assignment 3 – Calculate the size of a dynamic index for a dynamic indexed disk file. Size parameters include bucket size, initial fill factor and overhead for data buckets. The index structure’s nodes have structure depending on node size, descriptor format, free space allocation and so on (1 week).

5. Programming/Lab Assignment 4 (student assistance is needed) – Do introductory SQL processing and interaction with corresponding relational database (1.5 weeks).

6. Programming/Lab Assignment 5 – More advanced SQL query processing. Multi-table queries; view processing; loading tables from externally-supplied data (2 weeks).

7. Written Assignment 6 – Constructing an ERD from an English specification (1 week).

8. Lab Assignment 7 – Performing an ERD to relational schema mapping; using UML-based tools to synthesize a relational schema with a database application (1 week).

9. Written Assignment 8 – Workflow analysis for a typical Web-based n-tier application having a database backend (1.5 weeks).

Estimated Curriculum Category Content (Semester hours)

<table>
<thead>
<tr>
<th>Area</th>
<th>Core</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>Data Structures</td>
<td></td>
<td>0.5</td>
</tr>
</tbody>
</table>
Software Design | Prog. Languages
-----------------|------------------
Comp. Arch.      | 0.5              

Oral and Written Communications
Use of tools such as ERD and UML involves written analysis and design (prose) descriptions and documentation for design of process flows as well as database structures. In addition, programming-assignment source code documentation is critiqued (and counted as part of the assignment grade for each programming assignment) throughout the semester.

Social and Ethical Issues
No significant component.

Theoretical Content
1. Size, structure and performance complexity analysis (4 weeks).
2. ERD representations (1 week).
3. UML representations for databases (1 week).
4. Relational database schemas and relational model constraints (2 weeks).
5. Introductory queuing and workflow modeling (1 week).

Problem Analysis
In the first part of the course, students analyze the structure (space and representation techniques) of various common file organizations. This analysis is done along with cost/performance analysis of various file access methods used in (file) record processing. Analysis includes estimates of space and time and size. Later in the course, UML-based analysis as well as ERD and EERD analysis (based on entity/attribute/relationship representations) is used to model database content. The database-content analysis is coupled with introductory query and transaction estimates and sizing. Finally, at the end of the course a case study of Web-centric applications and workflow is done. Elementary service queuing and traffic distributions and loads are covered.

Solution Design
Physical database design activities in the course start with class examples and a written exercise involving selection (optimal if possible) of file organizations and access methods for a database with a pre-specified query load. A database normalization exercise is done by students to experience the practical consequences of the normalization process.