CMP3101 Database Systems

Period per Week			Contact Hour per Semester	Weighted Total Mark	Weighted Exam Mark	Weighted Continuous Assessment Mark	Credit Units
LH	PH	TH	CH	WTM	WEM	WCM	CU
45	30	00	60	100	60	40	4

Rationale

This Course is designed to address the needs of the student undertaking computer engineering in database systems.

Objectives

- To introduce the student to the fundamental concepts of data models
- To expose the student to the methods and techniques appropriate for a given problem
- To introduce the student to the selection and implementation of appropriate database solutions that reflect all suitable constraints, including scalability and usability.

Course Content

1. History and Overview of Database Systems

- Indicate some reasons for studying database systems
- Highlight some people that influenced or contributed to the area of database systems
- Indicate some important topic areas such as information systems, database systems and, data modeling
- Contrast the meanings between data, information, and knowledge

- Describe a database system and its components
- Mention the use of database query languages
- Describe the meaning and purpose of a data model
- Explore some additional resources associated with database systems
- Explain the purpose and role of database systems in computer engineering

2. ***Fundamentals of Database Systems***
   - Components of database systems; problem of the accuracy of information
   - Database management system (DBMS) functions: the different possibilities and the role they play in a database system
   - Database architectures: the possibilities, the concept of, the importance of and the reality of data independence
   - Use of a database query language
3. ***Data Modeling***
   - Data modeling: the role of this, the benefits it brings and the common approaches; conceptual data model, physical data model, and representational data model
   - Basic concepts: to include key, foreign key, record, relation
   - Conceptual models: possibilities, entity-relationship model and UML; strengths and weaknesses; notational issues
   - Object-oriented model: the main concepts and object identity, type constructors, encapsulation, inheritance, polymorphism, and versioning; basic approaches
   - Relational data model: basic terminology, basic approaches, strengths and weaknesses
4. ***Relational Databases***
   - Concepts of conceptual schema and relational schema: uses, comparisons, mapping conceptual schema to relational schema
   - Entity integrity constraint and referential integrity constraint; definitions, uses
   - Relational algebra and relational calculus
   - Relational algebra operations from mathematical set theory (union, intersection, difference, and Cartesian product) contrasted with the relational algebra operations developed specifically for relational databases (select, product, join, and division).
5. ***Database Query Languages***
   - Overview of database languages
   - Structured query language (SQL): fundamental concepts including data definition, query formulation, update sub-language, constraints, integrity
   - Query processing strategies; query optimization
   - Query by example and 4th-generation environments
   - Embedding non-procedural queries in a procedural language
   - Introduction to object query language
6. ***Relational Database Design***
   - Database design
   - The concept of functional dependency

- Normal forms: first, second, third and Boyce-Codd normal forms; motivation for each of these, applicability; mechanisms for producing these normal forms
- Multi-valued dependency: fourth normal form; join dependency; fifth normal form
- Representation theory

7. Transaction Processing

- Transactions: the purpose and the nature of transactions; creating a transaction using SQL; characteristics of efficient transaction execution; concept of commit
- Failure and recovery: the differing possibilities, their strengths and weaknesses
- Concurrency control: the special problems introduced by concurrency; the solution to these problems; isolation levels and their effects

8. Distributed Databases

- Advantages offered by the introduction of distribution: the problems introduced
- Distributed data storage: techniques used for data fragmentation, replication, and allocation during the distributed database design process
- Distributed query processing: strategies for executing distributed queries
- Distributed transaction model
- Concurrency control: the different approaches including those based on the distinguished copy techniques and the voting method
- Homogeneous and heterogeneous solutions
- The client-server approach

9. Physical Database Design

- Storage requirements for a range of data including characters, numbers, strings, text, sound, video and file structure
- Characteristics of storage to support a range of databases including use of CDs, memory in machines of different kinds; nature of the storage systems involved and the factors influencing choice
- Records and record types; fixed length and variable length; storage organization
- Files of different kinds and file structures: sequential files, indexed files, hashed files; signature files; files with dense index
- B-trees; definition, use in particular in implementing dynamic multi-level indices
- Data compression: reasons for using this, compression algorithms, strengths and weaknesses of each approach; software support
- Database efficiency and tuning: performance measures, monitoring performance

Learning Outcomes

On completion of this course the student will be able to:

- Identify some contributors to database systems and relate their achievements to the knowledge area, explain how knowledge differs from information and data, and describe how computer engineering uses or benefits from database systems and information management.

- Explain the characteristics that distinguish the database approach from the traditional approach of programming with data files, cite the basic goals, functions, models, components, applications, and social impact of data management systems, describe the components of a database system and give examples of their use and use a query language to elicit information from a database.
- Categorize data models based on the types of concepts that they provide to describe the database structure and compare and contrast the basic principles of the relational data model and those of the object-oriented model as they apply to computer engineering applications.
- Prepare a relational schema from a conceptual model developed using the entity-relationship model, explain and demonstrate the concepts of entity integrity constraint and referential integrity constraint and demonstrate the successful formulation of queries in the relational algebra and queries in the tuple relational calculus.
- Create a relational database schema in SQL that incorporates key, entity integrity, and referential integrity constraints; demonstrate data definition in SQL and retrieving information from a database using the SQL *SELECT* statement, evaluate a set of query processing strategies and select the optimal strategy, and demonstrate the ability to embed object-oriented queries into an appropriate stand-alone programming language
- Determine the functional dependency between two or more attributes that are a subset of a relation, demonstrate the ability to transform a relation into a prescribed normal form, explain the impact of normalization on the efficiency of database operations and query optimization and describe a multi-valued dependency and the type of constraints it specifies.
- Explain the purpose of rollback and the way logging assures that proper rollback takes place; outline the special problems arising from the use of concurrency and the effect of effect of different isolation levels on the concurrency control mechanisms, and demonstrate the ability to apply suitable ideas on failure and recovery to an application taken from computer engineering.
- Evaluate simple strategies for executing a distributed query to select the strategy that minimizes the amount of data transfer, Explain how to use the two-phase commit protocol to deal with committing a transaction that accesses databases stored on multiple nodes, and compare and contrast the different approaches to distributed concurrency control.
- Cite examples of the application of primary, secondary, and clustering indexes, explain the theory and application of internal and external hashing techniques and its use in facilitating dynamic file expansion, describe the relationships among hashing, compression, and efficient database searches and explain how physical database design affects database transaction efficiency.

Recommended and Reference Books

[1] Hector Garcia-Molina, Jeff Ullman, and Jennifer Widom, 2008. *Database Systems: The Complete Book* (DS:CB), 2nd Edition , Prentice Hall, ISBN-10: 0130319953, ISBN-13: 9780130319951

[2] Jeffrey D. Ullman, Jennifer Widom, 2007. *A First Course in Database Systems*. Prentice Hall, ISBN 013600637X

[3] Rebecca Riordan, 2005. *Designing Effective Database Systems*. Addison-Wesley ISBN 0321290933.

[4] Thomas M. Connolly, and Carolyn E. Begg, 2004. *Database Systems: A Practical Approach to Design, Implementation*. Addison-Wesley, ISBN 0321294017