University of Macau
Faculty of Science and Technology
Department of Computer and Information Science
SFTW370 Database Systems I
Syllabus
2nd Semester 2013/2014
Part A – Course Outline

Compulsory course in Computer Science

Course description:
(2-2) 4 credits. This course provides an overview of database architecture, database system and management design, which covers both the theoretical and practical aspects of the relational model. Topics include an in-depth study of relational database model, fundamentals of database design techniques, structured query language (SQL), schema refinement and normal forms, storage and various indexing techniques. The course also includes a study of a commercially available database system and database application programming.

Course type:
Theoretical with substantial laboratory/practice content

Prerequisites:
- SFTW111 – Algorithms and Data Structures I

Textbook(s) and other required material:
- A. Silberschatz, H. F. Korth and S. Sudarshan, *Database System Concepts*, Sixth edition, McGraw Hill, 2011.

References:
- R. Elmasri and Shamkant B. Navathe, *Fundamentals of Database Systems*, Sixth edition, Addison-Wesley, 2011.
- C.J. Date, *An Introduction to Database Systems***,** Eighth edition, Addison-Wesley, 2004.

Major prerequisites by topic:
- Computers and programming
- Data structures and algorithms.

Course objectives:
- Learn theoretical principles of database design with ER model and relational data model. [a, c, e, k, l]
- Learn structured query languages (SQL) and relational algebra for data manipulation. [a, c, e, k]
- Learn schema refinement techniques and normal form theory for good database design. [a, c, e, k]
- Learn data storage and indexing techniques to understand the fundamentals of database systems. [a, e, k]
- Design and develop a database application with a case study problem. [a, c, e, k, l]

Topics covered:
- **Conceptual database model (3 hrs) –** overview of database design, entity-relationship (ER) model, entities, attributes, and entity sets, relationships and relationship sets, additional features of the ER model, conceptual database design with the ER model.
- **Relational data model (3 hrs) –** integrity constraints over relations, enforcing integrity constraints, querying relational data, logical database design: ER to relational, views, destroying/altering tables and views.
- **Relational queries (8 hrs) –** relational algebra and its expressive power, basic SQL query, union, intersect, and except, nested queries, aggregate queries, null values and outer joins.
- **Schema refinement and normal forms (4 hrs) –** introduction to schema refinement, functional dependencies, normal forms, properties of decompositions, schema refinement in database design, normalization.
- **Data storage (5.5 hrs) –** disk space management, RAID and memory hierarchy, buffer management, files and indexes, page formats and record formats, comparison of three file organizations: heap files, sorted files and hashed files, properties of indexes: clustered versus unclustered, B+ tree versus hashing, indexes using composite search keys.

- **Indexing (4.5 hrs) –** Indexed Sequential Access Method (ISAM), B+ trees: a dynamic index structure, operations of search, insert and delete with a B+ tree, bulk-loading a B+ tree, the order concept of B+ tree, static hashing, extendible hashing, linear hashing.

Class/laboratory schedule:

Timetabled work in hours per week			No of teaching weeks	Total hours	Total credits	No/Duration of exam papers
Lecture	Tutorial	Practice				
2	1	1	14	56	4	1 / 2 hours

Student study effort required:

Class contact:	
Lecture	28 hours
Tutorial & Laboratory	28 hours
Mid-term exam	2 hours
Other study effort	
Self-study	15 hours
Project / Case study	20 hours
Total student study effort	93 hours

Student assessment:
Final assessment will be determined on the basis of:

In-class Exercise	5%	Lab Practice	5%
Homework	10%	Group Projects	20%
Midterm exam	25%	Final exam	35%

Course assessment:
The assessment of course objectives will be determined on the basis of:
- Homework, exercise, practice, project and exams
- Course evaluation

Course outline:

Weeks	Topic	Course work
1-2	**Introduction to Database Design** Observing the steps in designing a database, how to use the ER model to create an initial design. The main concepts of the ER model will be discussed, which is used in conceptual database design phase. In which, entities, attributes, entity sets, relationships and relationship sets, and some additional features of the ER model will be learned.	Group project Assignment 1
2-3	**The Relational Model** Using SQL to create and modify relations, study the integrity constraints over relations, and observe how to enforce the integrity constraints. Learn logical database design in more detail: how to translate an ER model into relational model. Views are also introduced, and destroying/altering tables and views will be checked as well.	
4-5	**Relational Algebra** Understand the importance of relational algebra, learn the basic algebra operators and how are they combined to write complex queries, in which selection, projection, join, set and division operations will be explored in detail.	Assignment 2
6-8	**SQL** Understand what is included in the SQL language, study various SQL query, such as SELECT, UNION, INTERSECT, and EXCEPT.	Assignment 3

Weeks	Topic	Course work
	Experience how does SQL build to retrieve data, including the nested queries and aggregate operators. Learn also the complex integrity constraints in SQL, and understand the usefulness of triggers.	
9-10	**Schema Refinement and Normal Forms** Introduce functional dependencies and reasoning about FDs, learn normal forms and properties of decompositions, study the normalization and schema refinement in database design.	Assignment 4
10	**Midterm**	
11	**Storage and Index** Obtain the idea that how does a DBMS store and access persistent data, why is I/O cost so important for database operations. Have a look at how does a DBMS organize files of data records on disk to minimize I/O costs. Then will learn the index and why need to use index. Two types of indexes will be introduced: tree-based and hash-based, and their important properties will be described as well.	Assignment 5
12	**Storing Data: Disks and Files** Observe the physical characteristics of disks and how do they affect the design of database systems. Know what are RAID storage systems, and what are their advantages. Learn also how does a DBMS keep track of space on disks, how does a DBMS access and modify data on disks, and what is the significance of pages as a unit of storage and transfer. Describe how does a DBMS create and maintain files of records, how are records arranged on pages, and how are pages organized within a file.	
13	**Tree-Structured Indexing** Learn ISAM index and B+ tree index handle search, insert and delete, and showing why are they good for range selections. Study key compression, bulk-loading and the order concept.	Assignment 6
14	**Hash-based Indexing** Various hashing techniques will be introduced, explain why are they especially good for equality searches but useless for range selections. Study extendible hashing and linear hashing, to see how do they handle search, insert, and delete. Also talk about the similarities and differences between extendible and linear hashing.	
14	**Final Review**	Project presentation and demonstration

Contribution of course to meet the professional component:
This course prepares students with fundamental knowledge to database design and management.

Relationship to CS program objectives and outcomes:
This course primarily contributes to the Computer Science program outcomes that develop student abilities to:
(a) an ability to apply knowledge of mathematics, science, and engineering.
(c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
(e) an ability to identify, formulate, and solve engineering problems.
(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

The course secondarily contributes to the Computer Science program outcomes that develop student abilities to:
(l) an ability to use the computer/IT tools relevant to the discipline along with an understanding of their processes and limitations.

Relationship to CS program criteria:

Criterion	DS	PF	AL	AR	OS	NC	PL	HC	GV	IS	IM	SP	SE	CN
Scale: 1 (highest) to 4 (lowest)	4	4	3				2				1		2	

Discrete Structures (DS), Programming Fundamentals (PF), Algorithms and Complexity (AL), Architecture and Organization (AR), Operating Systems (OS), Net-Centric Computing (NC), Programming Languages (PL), Human-Computer Interaction (HC), Graphics and Visual Computing (GV), Intelligent Systems (IS), Information Management (IM), Social and Professional Issues (SP), Software Engineering (SE), Computational Science (CN).

Course content distribution:

Percentage content for			
Mathematics	Science and engineering subjects	Complementary electives	**Total**
0%	100%	0%	100%

Persons who prepared this description:
Dr. Sam Chao

Part B – General Course Information and Policies

2nd Semester 2013/2014

Instructor: Dr. Sam Chao
Office: R108
Office hour: Wed 10:00 am – 12:00 pm, Thu 9:30 am – 11:30 am, Fri 10:00 am – 12:00 pm, or by appointment
Phone: 8397 8051
Email: lidiasc@umac.mo

Time/Venue: Tue 9:30 – 11:30, JG06 (Lecture)
Wed 14:00 – 16:00, NG03; Sat 8:30 – 10:30, NG03 (Tutorial/Laboratory)

Grading distribution:

Percentage Grade	Final Grade	Percentage Grade	Final Grade
100 - 93	A	92 - 88	A–
87 - 83	B+	82 - 78	B
77 - 73	B–	72 - 68	C+
67 - 63	C	62 - 58	C–
57 - 53	D+	52 - 50	D
below 50	F		

Comment:
The objectives of the lectures are to explain and to supplement the text material. Students are responsible for the assigned material whether or not it is covered in the lecture. Students who wish to succeed in this course should read the assignments prior to the lecture and should work all homework and lab assignments, as well as the group projects. You are encouraged to look at other sources (other texts, etc.) to complement the lectures and text.

Homework policy:
The completion and correction of homework is a powerful learning experience; therefore:
- There will be approximately 4-6 homework assignments.
- Homework is due one week after assignment unless otherwise noted, no late homework is accepted.
- All assignments must be word-processed in A4 size, and all the diagrams must be developed by a drawing tool such as Visio. No handwritten work will be accepted or counted for grading.
- There will be occasional in-class assignments during the class.
- Lab exercise is another major hands-on assignment for the course.
- The course grade will be based on the average of the HW grades.

Group projects:
The project provides students with meaningful experience to design and implement the components of an RDBMS and an RDBMS application in the course:
- The application domain will be discussed further in class.
- The projects are in groups of 2-3, no more than 3.
- The projects will be presented and demonstrated to the instructor or TAs.

Exam:
One 2-hour mid-term exam and one 3-hour final exam will be held during the semester. Both the mid-term and final exams are closed book examinations.

Note:
- Laboratory session is important part of this course and attendance is strongly recommended.
- Check course web pages for announcement, homework and lectures. Report any mistake on your grades within one week after posting.
- No make-up exams are given except for CLEAR medical proof.
- Cheating is absolutely prohibited by the university.

Appendix:

Rubric for Program Outcomes

Rubric for (a)	5 (Excellent)	3 (Average)	1 (Poor)
Understand the theoretic background	Students understand theoretic background and the limitations of the respective applications.	Students have some confusion on some background or do not understand theoretic background completely.	Students do not understand the background or do not study at all.
Use a correct model and formulation correctly	Students choose a model correctly and properly apply correct techniques.	Students choose a wrong model sometime, use a wrong formula, or a different technique.	Students use a wrong model and wrong formula, or do not know how to model.

Rubric for (c)	5 (Excellent)	3 (Average)	1 (Poor)
Design capability and design constraints	Student understands very clearly what needs to be designed and the realistic design constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.	Student understands what needs to be designed and the design constraints, but may not fully understand the limitations of the design constraints.	Student does not understand what needs to be designed and the design constraints.
Process to meet desired needs	Student understands very clearly the process of the design.	Student understands what the needs of the process design, but may not fully understand the limitations of the design constraints.	Student does not understand the process.

Rubric for (e)	5 (Excellent)	3 (Average)	1 (Poor)
Modeling, problem formulation and problem solving	Students choose and properly apply the correct techniques.	Students model correctly but cannot select proper technique or model incorrectly but solve correctly accordingly.	Students at loss as to how to solve a problem.

Rubric for (k)	5 (Excellent)	3 (Average)	1 (Poor)
Use modern principles, skills, and tools in engineering practice	Student applies the principles, skills and tools to correctly model and analyze engineering problems, and understands the limitations.	Student applies the principles, skills and tools to analyze and implement engineering problems.	Student does not apply principles and tools correctly and/or does not correctly interpret the results.

Rubric for (l)	5 (Excellent)	3 (Average)	1 (Poor)
Use modern computer/IT tools relevant to the discipline	Student uses computer/IT tools relevant to the engineering discipline, and understands their limitations.	Student uses computer /IT tools relevant to the engineering discipline.	Student does not use computer/IT tools relevantly, and does not understand their limitations.