COURSE DESCRIPTION

Department and Course Number	CS 122	**Course Coordinator**	Huiping Guo
Course Title	Using Relational Databases and SQL	**Total Credits**	3

Current Catalog Description:

An introduction to relational databases and the SQL query language. Database modeling as collection of objects and their relationships; Entity-relationship model. SQL as a query language. Grouping and other advanced queries.

Textbook:

Randolph, Gary and Griffin, Jeffrey., *SQL Essentials*, Franklin Beedle and Associates, 2004.

References:

Fehily, Chris., SQL: *Visual Quick Start Guide*, Peachpit Press, 2002.

Patrick, J, John., *SQL Fundamentals*, *2nd Edition*, Prentice Hall, 2002.

Wilton, Paul and Colby, John., *Beginning SQL*, Wrox, 2005.

Bagui, Sikha and Earp, Richard., *Learning SQL: A Step-by-Step Guide using Access*, Addison Wesley, 2003.

Course Goals:

At the end of the course, students are able to
- Set up and use at least one mainstream database management system.
- Use the SQL query language to express compound search conditions, combine and process data from multiple columns or tables, and format the results into user-friendly reports.
- Design and implement a database schema in 3rd Normal Form and improve an existing database schema by normalization.

These course goals contribute to the success of **Student Learning Outcomes 1.d, 5, and 6**.

Prerequisites by Topic:

Computer literacy.

Major Topics Covered in the Course:

- Introduction to relational database systems and SQL
- Selections

- Joins
- Aggregations
- Functions and set operations
- Subqueries
- Views and temporary tables
- Table creation and updates
- Introduction to database design
- Normalization
- SQL query performance issues

Laboratory Projects (specify number of weeks on each):

Each week the students complete a 1.5-hour lab project on a selected topic:

- Week 1: Familiarize with the schema of given database, and retrieve information using simple selection queries.

- Week 2: Construct selection queries with more complex predicates using various operators.

- Week 3: Combine information from two tables using join queries.

- Week 4: Design more complex join queries such as self joins or joins that involve three or more tables.

- Week 5: Practice with aggregation queries, and the combination of aggregations and joins.

- Week 6: More on aggregation queries, as well as string and date functions.

- Week 7: Retrieve data using subqueries and set operators.

- Week 8: Create relations with given schema, and populate the relations using insert and update statements.

- Week 9: Given a database schema in 1st Normal Form, normalize it to 3rd Normal Form and implement it in a database management system.

- Week 10: Implement the same query using different techniques and study the performance implications of the different implementations.

Estimate Curriculum Category Content (Quarter Hours)

Area	Core	Advanced	Area	Core	Advanced
Algorithms	0.25		Data Structures	0.25	
Software Design	0.5		Prog. Languages	2.0	
Comp. Arch.					

Oral and Written Communications:

Written documentation of software built in labs and homework assignments.

Social and Ethical Issues:

No significant component.

Theoretical Content:

- Introduction to relational theory (1 week)
- Data modeling and normalization (2 weeks)

Problem Analysis:

In the first part of the course, students are given database schemas, and are required to identify the relations and attributes to be used in query construction. In the latter part of the course, students are given database schemas that are in 1st Normal Form, and are required to identify the deficiencies in the databases designs, and the steps to be taken to eliminate such deficiencies using normalization techniques.

Solution Design:

Solution design in this course mostly involves choosing appropriate SQL statements to perform certain database tasks, such as creating database relations, update data in the database, process and retrieve information under various search conditions. We also introduce to students to basic database design technique, namely, starting with a single relation in 1st Normal Form and improving it using normalization technique. More advanced database design paradigms are discussed in a more advanced course.