COURSE OUTLINE: CSD123 - DATABASES I

Prepared: Rodney Martin
Approved: Martha Irwin, Dean, Business and Information Technology

Course Code: Title	CSD123: DATABASES I
Program Number: Name	2095: COMPUTER PROGRAMMING
Department:	COMPUTER STUDIES
Academic Year:	2024-2025
Course Description:	Databases are employed for data storage and retrieval in most software systems. Learners in this course are introduced to relational and non-relational (NoSQL) databases and their typical uses. By interacting with real databases, students gain an understanding of the importance of normalization and the advantages and disadvantages of the relational and non-relational models. Students use SQL to manipulate and query relational databases and gain hands-on experience with the use of non-relational databases. As part of this course, students use popular database systems, which may include MySQL, SQLite, MongoDB, etc.
Total Credits:	4
Hours/Week:	4
Total Hours:	56
Prerequisites:	CSD110
Corequisites:	There are no co-requisites for this course.
This course is a pre-requisite for:	CSD214, CSD216, CSD226, CSD227, CSD235
Vocational Learning Outcomes (VLO's) addressed in this course: **Please refer to program web page for a complete listing of program outcomes where applicable.**	**2095 - COMPUTER PROGRAMMING** VLO 3 Implement and maintain secure computing environments. VLO 9 Support the analysis and definition of software system specifications based on functional and non-functional requirements. VLO 11 Apply one or more programming paradigms such as, object-oriented, structured or functional programming, and design principles, as well as documented requirements, to the software development process. VLO 12 Model, design, implement, and maintain basic data storage solutions.
Essential Employability Skills (EES) addressed in this course:	EES 1 Communicate clearly, concisely and correctly in the written, spoken, and visual form that fulfills the purpose and meets the needs of the audience. EES 2 Respond to written, spoken, or visual messages in a manner that ensures effective communication. EES 3 Execute mathematical operations accurately. EES 4 Apply a systematic approach to solve problems. EES 5 Use a variety of thinking skills to anticipate and solve problems.

	EES 9 Interact with others in groups or teams that contribute to effective working relationships and the achievement of goals. EES 10 Manage the use of time and other resources to complete projects. EES 11 Take responsibility for ones own actions, decisions, and consequences.
Course Evaluation:	Passing Grade: 50%, D A minimum program GPA of 2.0 or higher where program specific standards exist is required for graduation.
Other Course Evaluation & Assessment Requirements:	Students are expected to be present to write all tests in class, unless otherwise specified. If a student is unable to write a test due to illness or a legitimate emergency, that student must contact the professor prior to class and provide reasoning. Should the student fail to contact the professor, the student shall receive a grade of zero on the test. If a student is not present 10 minutes after the test begins, the student will be considered absent and will not be given the privilege of writing the test. Students exhibiting academic dishonesty during a test will receive an automatic zero. Please refer to the College Academic Dishonesty Policy for further information. In order to qualify to write a missed test, the student shall have: a.) attended at least 75% of the classes to-date. b.) provide the professor an acceptable explanation for his/her absence. c.) be granted permission by the professor. NOTE: The missed test that has met the above criteria will be an end-of-semester test. Labs / assignments are due on the due date indicated by the professor. Notice by the professor will be written on the labs / assignments and verbally announced in advance, during class. Labs and assignments that are deemed late will have a 10% reduction per academic day to a maximum of 5 academic days at 50% (excluding weekends and holidays). Example: 1 day late - 10% reduction, 2 days late, 20%, up to 50%. After 5 academic days, no late assignments and labs will be accepted. If you are going to miss a lab / assignment deadline due to circumstances beyond your control and seek an extension of time beyond the due date, you must contact your professor in advance of the deadline with a legitimate reason that is acceptable. It is the responsibility of the student who has missed a class to contact the professor immediately to obtain the lab / assignment. Students are responsible for doing their own work. Labs / assignments that are handed in and are deemed identical or near identical in content may constitute academic dishonesty and result in a zero grade. Students are expected to be present to write in-classroom quizzes. There are no make-up options for missed in-class quizzes. Students have the right to learn in an environment that is distraction-free, therefore, everyone is expected to arrive on-time in class. Should lectures become distracted due to students walking in late, the professor may deny entry until the 1st break period, which can be up to 50 minutes after class starts or until that component of the lecture is complete. Grade Definition Grade Point Equivalent A+ 90 - 100% 4.00

A 80 - 89%
B 70 - 79% 3.00
C 60 - 69% 2.00
D 50 - 59% 1.00
F (Fail) 49% and below 0.00

CR (Credit) Credit for diploma requirements has been awarded.
S Satisfactory achievement in field /clinical placement or non-graded subject area.
U Unsatisfactory achievement in field/clinical placement or non-graded subject area.
X A temporary grade limited to situations with extenuating circumstances giving a student additional time to complete the requirements for a course.
NR Grade not reported to Registrar`s office.
W Student has withdrawn from the course without academic penalty.

Books and Required Resources:

Only reference documentation and free education resources are used in this course

Course Outcomes and Learning Objectives:

Course Outcome 1	**Learning Objectives for Course Outcome 1**
1. Explain the problems that database systems aim to solve	1.1 Describe the disadvantages of spreadsheet/file processing 1.2 Explain the modification problems that can arise in spreadsheets/files 1.3 Explain why data redundancy can cause modification problems 1.4 Define data integrity, referential integrity, domain integrity, and entity integrity 1.5 Describe how databases help to enforce data integrity and reduce data redundancy 1.6 Describe the components of a database management system
Course Outcome 2	**Learning Objectives for Course Outcome 2**
2. Describe the components and structure of a well normalized relational database	2.1 Define the terms schema, table, column, row, data type, and key, and describe how they relate to each other 2.2 Identify the primary key of a table and explain its significance 2.3 Explain the purpose of a foreign key 2.4 Explain in informal terms what it means for a relational database to be normalized 2.5 Identify the data type of a table column 2.6 Define 1-to-1, 1-to-many, and many-to-many relationships, and identify such relationships in the tables of a database 2.7 Define and identify recursive relationships 2.8 Define and identify supertype/subtype relationships
Course Outcome 3	**Learning Objectives for Course Outcome 3**
3. Retrieve and manipulate data in a database using basic SQL queries	3.1 Retrieve specific data from a table using SELECT statements 3.2 Filter and sort data in a query result using WHERE and ORDER BY clauses 3.3 Limit the number of rows in a query result using LIMIT 3.4 Count rows in a query result using the COUNT function 3.5 Use calculated columns in queries

	3.6 Prevent duplicate rows in query results using the DISTINCT keyword 3.7 Relabel columns in query results using aliasing
Course Outcome 4	**Learning Objectives for Course Outcome 4**
4. Manipulate SQL data using DML queries	4.1 Insert data into a table 4.2 Update existing data in a table 4.3 Delete data from a table 4.4 Copy data from one table to another 4.5 Create copies of table schemas 4.6 Create and use temporary tables 4.7 Create and use database views
Course Outcome 5	**Learning Objectives for Course Outcome 5**
5. Retrieve data from multiple tables using joins	5.1 Describe the nature of a cross join 5.2 Describe the nature of an inner join 5.3 Explain how foreign keys can be used in inner joins to combine related data from separate tables 5.4 Write SQL queries involving inner joins 5.5 Define left outer joins and explain how they differ from inner joins 5.6 Explain when to employ left outer joins 5.7 Write SQL queries involving left outer joins 5.8 Write queries involving multiple joins 5.9 Relabel a table involved in a query using aliasing 5.10 Write queries joining data from tables in a recursive relationship
Course Outcome 6	**Learning Objectives for Course Outcome 6**
6. Integrate software applications with databases	6.1 Retrieve and manipulate information in a database using a database connector library in an application 6.2 Mitigate SQL injection attacks using prepared statements 6.3 Securely store passwords in a database 6.4 Describe the syntax, capabilities, and limitations of stored procedures, user defined functions, and triggers 6.5 Define what is meant by an `atomic` database operation 6.6 Describe the purpose of transactions 6.7 Use transactions to ensure atomicity in a sequence of SQL statements 6.8 Explain what a deadlock is, how it can arise in situations where transactions are used, and how to mitigate the problem
Course Outcome 7	**Learning Objectives for Course Outcome 7**
7. Analyze data using grouping and aggregate functions	7.1 Describe the nature of various common aggregate functions 7.2 Use aggregate functions in conjunction with groups 7.3 Define what a functional dependency is and describe how the concept relates to grouping 7.4 Filter grouped query results using a HAVING clause 7.5 Distinguish between WHERE and HAVING clauses, and describe the limitations of each
Course Outcome 8	**Learning Objectives for Course Outcome 8**

8. Use subqueries to create advanced queries	8.1 Explain how subqueries can be used to dynamically compute values or tables used in an outer query 8.2 Write queries that use a subquery in place of a scalar value 8.3 Write queries that use a subquery in place of a list 8.4 Write queries that use a subquery in place of a table 8.5 Write queries that involve joining a subquery to a table 8.6 Describe what a corelated subquery is and explain why it should be avoided if possible
Course Outcome 9	**Learning Objectives for Course Outcome 9**
9. Describe NoSQL databases and when they are appropriate	9.1 Describe the different types of NoSQL databases 9.2 Discuss the advantages and disadvantages of NoSQL vs relational databases 9.3 Use a document store database

Evaluation Process and Grading System:

Evaluation Type	Evaluation Weight
Formative assessments & activities	10%
Lab assignments	40%
Tests	50%

Date: June 16, 2024

Addendum: Please refer to the course outline addendum on the Learning Management System for further information.