CPRG 307 - Database Programming and Testing

Course Description:
Database Programming and Tuning is a continuation of Database Design and Programming. Learners will focus on learning database specific structured languages and embedding SQL into these languages. Learners will demonstrate efficient code creation using an industry identified database vendor language. Topics include basic programming structures, storing code inside the database, triggers, writing efficient code, advanced SQL concepts, and code testing.

3 credits

Time Guidelines:
The standard instructional time for this course is 90 hours.

Effective Term:
Fall 2017/2018

Prerequisite(s):

- CPRG 250 - Database Design and Programming
- CPRG 251 - Object-Oriented Programming Essentials

Course Assessment:
Assignments, exams and in-class work are to be completed to the satisfaction of the instructor to meet requirements for successful completion of this course.

Students will be evaluated on the basis of:

Professionalism	15%
Labs	5%
Quizzes	25%
Assignments	10%
Midterm Exam	20%
Final Exam (Comprehensive)	25%
Total:	100%

It is mandatory that a student achieve a passing grade in the technical component (out of 85) to pass the course.

Other Course Information:
Attendance Policy for all IT Courses

Students are expected to attend and to be on time for all classes. Attendance is taken at the beginning of every class. Please notify your instructor if you must be absent from a class, or contact your instructor as soon as possible after missing a class. You will be required to justify your absence (an unexcused absence is any absence for which no explanation or justification is given - such as a doctor's note, accident report, letter attesting to dire family or personal circumstances, etc). Classes will not be re-taught; any handouts or notes that you miss by not attending class should be obtained from classmates. You will be responsible for the material and concepts taught during the missed class, along with any assignments.

It is important that you let your instructor know if you are going to be absent. Email is the preferred method but your instructor will confirm this with you at the start of the course.

Unexcused Absence Consequences:
Students who miss an evaluation because of an unexcused absence will receive a zero mark in that evaluation. Students must be proactive and advise instructors of any extenuating circumstances. Unexcused absence hours that exceed 30% of total course hours, will result in an 'F' grade for the course. The following three step process is in effect:

Three Step Process

Unexcused Absences	**Warning**	**Meet with**	**Consequences**
10% of total course hours	1st	Instructor	Discussion of consequences should unexcused absences continue
20% of total course hours	2nd	Academic Chair	Deduction of one letter grade for final course mark
30% of total course hours		Academic Chair	An 'F' grade for course

Learner Engagement
In order to be successful the learner is expected to be engaged in learning activities for a total of 9 to 12 learning hours a course per week which includes both in-class and out-of-class time.

Core Abilities
Core Abilities are broad outcomes or skills that every learner of ICT programs are expected to achieve. These skills go beyond the context of a specific course or program and are the skills employers tell us they expect employees to have. ICT has identified five Core Abilities that are important in every area of learning which will be assessed through established criteria.

- Communicate Effectively
- Develop Self
- Work with Others
- Act Responsibly
- Think Critically

The development of core abilities leads to professionalism. Core abilities and professionalism will be assessed throughout this course using established criteria and the professionalism rubric.

SAIT Policies and Procedures:
For information on the SAIT Grading Scale, please visit policy AC 3.1.1 Grading Progression Procedure: http://www.sait.ca/Documents/About SAIT/Administration/Policies and Procedures/AC.3.1.1 Grading and Progression Procedure.pdf

For information on SAIT Academic Policies, please visit: www.sait.ca/about-sait/administration/policies-and-procedures/academic-student

Required Course Publication(s):
Casteel, J. (2013). *Oracle 11g: PL/SQL Programming*. Cengage Learning. ISBN: 9781133947363.

Course Material(s):
Oracle Express Software
Manufacturer: Oracle
Source: www.oracle.com

Course Learning Outcome(s):
1. Create database objects that assist developers during the software development process.

    Objectives:

        1.1 Explain the purpose of database views.

        1.2 Use the appropriate commands to create and modify views.

        1.3 Explain why sequences are useful to the database programmer.

        1.4 Use the appropriate commands to create and modify sequences.

        1.5 Use the sequences in DML commands.

        1.6 Explain why synonyms are useful in the management of a development environment.

        1.7 Show data dictionary information on database objects.

2. Incorporate programming concepts in software development solutions.

    Objectives:

        2.1 Create application development documentation.

        2.2 Document the problem path and solution.

        2.3 Create efficient, modular and reusable code.

        2.4 Produce in-line documentation for all source code.

3. Create simple programs using basic programming structures.

    Objectives:

        3.1 Describe the structure of a programming block.

        3.2 Create a programming block.

        3.3 Explain data declarations for scalar variables.

        3.4 Create data declarations for scalar variables.

        3.5 Use the built-in database function to output results to the screen.

        3.6 Explain the use of SQL commands in a programming block.

        3.7 Use SQL commands in a programming block.

        3.8 Modify code to correct syntax errors.

        3.9 Create variables that dynamically retrieve the data type from a database column.

        3.10 Use composite data types.

3.11 Explain variable scope.

4. Create programs using various structures that control the flow of a program.

Objectives:

4.1 Explain how various forms of the IF statement control program flow.

4.2 Create various looping structures to control program flow.

4.3 Explain the purpose of cursors in a program.

4.4 Create cursors to control the retrieval of data.

4.5 Create cursor FOR loops.

4.6 Identify the difference between basic cursor loops and cursor FOR loops.

4.7 Use the CASE structure to replace complex IF statements.

5. Utilize exception handling to process runtime errors.

Objectives:

5.1 Describe the database vendor’s error-handling environment within the programming block.

5.2 Demonstrate the integration of exception handlers into a program.

5.3 Use predefined exceptions provided by the database vendor.

5.4 Use user-defined exceptions.

5.5 Apply error codes and messages retrieved from the database vendor’s built-in functions inside of the exception block.

5.6 Employ user-defined error codes and messages.

5.7 Use embedded programming blocks to identify specific causes of exception.

6. Create stored procedures.

Objectives:

6.1 Explain the difference between an anonymous block and a named program unit.

6.2 Explain the structure of the CREATE PROCEDURE command.

6.3 Explain the use of parameters in a procedure.

6.4 Use parameters in a procedure to make code reusable.

6.5 Identify syntax errors in a procedure.

6.6 Modify procedure code to correct syntax errors.

6.7 Execute a procedure.

6.8 Execute a procedure from another programming block.

6.9 Show a stored procedure’s structure.

6.10 Explain the use of subprograms.

6.11 Create a subprogram.

6.12 Explain the flow of exceptions when using multiple programming blocks.

6.13 Use the data dictionary to view information about stored procedures in the database.

6.14 Use the DROP PROCEDURE command to remove stored procedures from the database.

7. Create stored functions.

Objectives:

7.1 Explain the difference between a function and a procedure.

7.2 Explain the structure of the CREATE FUNCTION command.

7.3 Use parameters in a function to make code reusable.

7.4 Execute a function.

7.5 Identify syntax errors in a function.

7.6 Modify function code to correct syntax errors.

7.7 Explain the use of the RETURN command in functions.

7.8 Use the RETURN command in a function.

7.9 Explain the difference between actual and formal parameters.

7.10 Describe function restrictions and requirements.

7.11 Use the data dictionary to view information about functions in the database.

7.12 Use the DROP FUNCTION command to remove stored functions from the database.

7.13 Identify the advantages of using packages.

8. Determine which database triggers are required to enforce business rules that cannot be enforced using constraints.

Objectives:

8.1 Describe the types and structure of triggers.

8.2 Create different types of triggers.

8.3 Use different types of triggers to solve programming problems.

8.4 Use trigger predicates in the logging of database activity.

8.5 Explain the difference between a DML trigger and a system trigger.

8.6 Explain the uses of database triggers.

8.7 Identify mutating and constraining tables within triggers.

8.8 Use the ALTER TRIGGER command to modify an existing trigger in the database.

8.9 Use the DROP TRIGGER command to remove triggers from the database.

8.10 Query the data dictionary to view information about triggers in the database.

9. Build an integrated business solution.

Objectives:

9.1 Explain Enterprise Resource Planning (ERP) and enterprise system concepts.

9.2 Call procedures, functions and utilities from a GUI application.

9.3 Explain business process concepts.

9.4 Apply business process concepts using a GUI application to a programming problem.

10. Devise tests to ensure coded solutions work as expected.

Objectives:

10.1 Describe why software testing is required.

10.2 Describe the difference between static and dynamic testing.

10.3 Explain various testing methods for software.

10.4 Create a test plan for a software solution.

© 2015, Southern Alberta Institute of Technology (SAIT). All Rights Reserved.
This document and materials herein are protected by applicable intellectual property laws. Unauthorized reproduction and distribution of this publication in whole or part is prohibited.