

Database Design and Management with PL/SQL (Oracle)

6662 36 weeks

Table of Contents

Acknowledgments.....	2
Course Description.....	3
Task Essentials Table.....	3
Curriculum Framework.....	9
Introducing PL/SQL Programming Concepts.....	9
Using PL/SQL Program Data	11
Using SQL in PL/SQL.....	14
Constructing PL/SQL Program Structures to Control Execution Flow	15
Using Cursors.....	18
Using Exception Handling.....	21
Examining PL/SQL Composite Data Types	24
Creating and Managing Procedures	25
Creating and Managing Functions	28
Designing Packages	31
Creating and Removing Triggers.....	36
Using Advanced Data Types	41
Understanding Procedural Dependencies	42
Using the PL/SQL Compiler.....	44
Preparing for Industry Certification.....	45
Developing Employability Skills	47
SOL Correlation by Task	51
Cyber Security and Cyber Forensics Infusion Units.....	58
Entrepreneurship Infusion Units	59

Microsoft Imagine Academy Resources 59
Appendix: Credentials, Course Sequences, and Career Cluster Information 60

Acknowledgments

The components of this instructional framework were developed by the following curriculum development panelists:

Jill Anderson, Instructor, Matoaca High School Center for Information Technology,
Chesterfield County Public Schools
Frank Baber, Director, Oracle Cloud Solution Hubs Developer Innovation Team, Oracle,
Reston
Gail Drake, Instructor, Battlefield High School, Prince William County Public Schools
Denise Hobbs, Regional Director, North America, Oracle, Albuquerque, New Mexico
Lisa Palombo, Senior Curriculum Developer, Oracle, St. George, Utah
Brian Thomas, Instructor, James Wood High School, Frederick County Public Schools

Correlations to the Virginia Standards of Learning were reviewed and updated by:

Leslie R. Bowers, English Teacher (ret.), Newport News Public Schools
Vickie L. Inge, Mathematics Committee Member, Virginia Mathematics and Science
Coalition
Anne F. Markwith, New Teacher Mentor (Science), Gloucester County Public Schools
Michael Nagy, Social Studies Department Chair, Rustburg High School, Campbell
County Public Schools

The framework was edited and produced by the CTE Resource Center:

Nathan K. Pope, Writer/Editor
Kevin P. Reilly, Administrative Coordinator

Virginia Department of Education Staff

Judith P. Sams, Specialist, Business and Information Technology and Related Clusters
Dr. Tricia S. Jacobs, CTE Coordinator of Curriculum and Instruction
Dr. David S. Eshelman, Director, Workforce Development and Initiatives
George R. Willcox, Director, Operations and Accountability

Office of Career, Technical, and Adult Education
Virginia Department of Education

Course Description

Suggested Grade Level: 10 or 11 or 12

Prerequisites: 6660

Web-based technologies used throughout industry, including interactive websites, accounting programs, research tools, search engines, e-learning environments, email managers, and numerous other applications, depend upon relational databases. PL/SQL, an extension of the SQL programming language, provides additional database functionality through variables, conditional statements, iterative controls, and packaging. Students enhance their relational database design and algorithm design skills by learning to write PL/SQL code that includes anonymous blocks, sub programs, procedures, functions, control structures, packages, and triggers, all within a browser-based programming environment. The cooperative education method is available for this course. Students combine classroom instruction and supervised on-the-job training in an approved position with continuing supervision throughout the year.

Recommended prerequisite(s): Advanced Database Design and Management (Oracle) 6661

Task Essentials Table

- Tasks/competencies designated by plus icons (⊕) in the left-hand column(s) are essential
- Tasks/competencies designated by empty-circle icons (○) are optional
- Tasks/competencies designated by minus icons (⊖) are omitted
- Tasks marked with an asterisk (*) are sensitive.

Task Number	6662	Tasks/Competencies
Introducing PL/SQL Programming Concepts		
39	⊕	Explain PL/SQL.
40	⊕	Differentiate between SQL and PL/SQL.
41	⊕	Explain the need for PL/SQL.
42	⊕	Describe the structure of a PL/SQL block.
43	⊕	Identify the different types of PL/SQL blocks.
Using PL/SQL Program Data		
44	⊕	Use variables in PL/SQL.

45	+	Describe valid and invalid identifiers in PL/SQL.	
46	+	Describe reserved words, delimiters, literals, and comments in PL/SQL.	
47	+	List data types used in PL/SQL.	
48	+	Identify the benefits of anchoring data types with the %TYPE attribute.	
49	+	Use built-in SQL functions in PL/SQL.	
50	+	Differentiate between implicit and explicit conversions.	
51	+	Demonstrate how functions can be used to explicitly convert data types.	
Using SQL in PL/SQL			
52	+	Evaluate the SQL statements that can be directly included in a PL/SQL executable block.	
53	+	Retrieve data in PL/SQL.	
54	+	Manipulate data with DML statements in PL/SQL.	
55	+	Use SQL cursor attributes.	
56	+	Use transaction control statements in PL/SQL.	
Constructing PL/SQL Program Structures to Control Execution Flow			
57	+	Identify the uses and types of conditional control structures.	
58	+	Construct and use an IF statement.	
59	+	Construct and use an IF-THEN-ELSE statement.	
60	+	Use basic loops with EXIT conditions.	
61	+	Use basic loops with EXIT WHEN conditions.	
62	+	Use WHILE loops.	
63	+	Use FOR loops.	
64	+	Use nested loops.	
Using Cursors			
65	+	Distinguish between an implicit and an explicit cursor.	

66	+	Declare and control explicit cursors.	
67	+	Use cursor simple LOOP to FETCH data.	
68	+	Use cursor FOR LOOP to fetch data.	
69	+	Declare and use cursors with parameters.	
70	+	Lock rows using the FOR UPDATE clause.	
71	+	Reference the current row with the WHERE CURRENT clause.	
72	+	Use multiple cursors.	
Using Exception Handling			
73	+	Describe the function of exceptions.	
74	+	Describe the function of an exception handler.	
75	+	Handle exceptions in PL/SQL programs.	
76	+	Trap predefined Oracle server exceptions.	
77	+	Trap non-predefined Oracle server errors.	
78	+	Trap user-defined errors.	
79	+	Describe scope of exceptions.	
Examining PL/SQL Composite Data Types			
80	+	Describe user-defined PL/SQL records.	
81	+	Create a user-defined PL/SQL record.	
82	+	Use PL/SQL INDEX BY Table and INDEX BY Table of Records.	
Creating and Managing Procedures			
83	+	Identify the characteristics and benefits of a stored procedure.	
84	+	Invoke a stored procedure.	
85	+	List the steps for creating a procedure.	
86	+	Create a procedure with parameters.	

87	+	Invoke a procedure that has parameters.	
88	+	List the types of parameter modes.	
89	+	Describe the DEFAULT option for parameters.	
90	+	Describe the method for propagating exceptions.	
91	+	Remove a procedure.	
92	+	Identify how to view and manage procedures.	
Creating and Managing Functions			
93	+	Define <i>stored function</i> .	
94	+	Create a function.	
95	+	List the procedures for invoking a function.	
96	+	List the advantages of user-defined functions in SQL statements.	
97	+	List circumstances in which user-defined functions can be called from within a SQL statement.	
98	+	Describe the restrictions on calling functions from SQL statements.	
99	+	Remove a function.	
100	+	Identify how to view stored objects in the data dictionary.	
101	+	Identify differences between invoker and definer rights.	
Designing Packages			
102	+	Identify a package specification and body.	
103	+	Create packages (e.g., related variables, cursors, constants, exceptions, procedures, and functions).	
104	+	Invoke a package construct.	
105	+	Designate package constructs as public or private.	
106	+	Drop packages.	
107	+	Identify benefits of packages.	

108	+	Create packages that use the overloading feature.
109	+	Identify restrictions on using packaged functions in SQL statements.
110	+	Invoke packaged functions from SQL.
111	+	Identify persistent states in package variables and cursors.
112	+	Describe the use and application of some Oracle server-supplied packages (e.g., DBMS_SQL, DBMS_OUTPUT, and UTL_FILE).
113	+	Use EXECUTE IMMEDIATE.
114	+	Describe the benefits of EXECUTE IMMEDIATE over DBMS_SQL for native dynamic SQL.
Creating and Removing Triggers		
115	+	Describe database triggers and types and the uses of each.
116	+	Create a DML trigger.
117	+	List the DML trigger components.
118	+	Create a statement-level trigger.
119	+	Describe the trigger-firing sequence options.
120	+	Use conditional predicates in a DML trigger.
121	+	Create a row-level trigger.
122	+	Use the OLD and NEW qualifiers in a database trigger.
123	+	Create an INSTEAD OF trigger.
124	+	Describe events that cause database triggers to fire.
125	+	Create a trigger for a DDL statement.
126	+	Create a trigger for a system event.
127	+	Describe the functionality of the CALL statement.
128	+	Describe the effect of a mutating table.
129	+	View trigger information in the dictionary views.

130	+	Alter a trigger status.	
131	+	Remove a trigger.	
Using Advanced Data Types			
132	+	Describe the function of LOB (large object) data types.	
133	+	Migrate from LONG to LOB.	
134	+	Manage BFILES.	
Understanding Procedural Dependencies			
135	+	Describe the implications of procedural dependencies.	
136	+	Describe dependent objects and referenced objects.	
137	+	View dependency information in the dictionary views.	
138	+	Use the UTLDTREE script.	
139	+	Use the IDEPTREE and DEPTREE views.	
140	+	List the procedures for minimizing dependency failures.	
Using the PL/SQL Compiler			
141	+	Describe PL/SQL initialization parameters.	
142	+	Use PL/SQL initialization parameters.	
143	+	Identify compiler warning messages.	
144	+	Describe conditional compilation.	
145	+	Describe the hiding of source code.	
Preparing for Industry Certification			
146	+	Describe the process and requirements for obtaining industry certifications related to the Database Design and Management with PL/SQL (Oracle) course.	
147	+	Identify testing skills and strategies for a certification examination.	

148	⊕	Demonstrate ability to successfully complete selected practice examinations (e.g., practice questions similar to those on certification exams).
149	○	Complete an industry certification examination representative of skills learned in this course (e.g., MOS, MTA, IC3).
Developing Employability Skills		
150	⊕	Investigate job opportunities, using the Internet and other sources.
151	⊕	Investigate careers in the information technology field.
152	⊕	Research certification and educational opportunities.
153	⊕	Compose or update a printed résumé.
154	⊕	Prepare an electronic résumé.
155	⊕	Create a letter of application.
156	⊕	Complete an electronic application form.
157	⊕	Create an interview follow-up letter.
158	⊕	Identify the steps to follow in resigning from a position.
159	⊕	Develop a portfolio containing representative samples of student's work (e.g., program design, technical documentation and design, source code, and output).
160	⊕	Deliver an oral presentation of portfolio.
161	⊕	Identify potential employment barriers for nontraditional groups and ways to overcome barriers.

Legend: ⊕ Essential ○ Non-essential ⊖ Omitted

Curriculum Framework

Introducing PL/SQL Programming Concepts

Task Number 39

Explain PL/SQL.

Definition

Explanation should include

- defining the acronym *PL/SQL*
- identifying the contexts in which the programming language is used
- identifying the generation of PL/SQL
- comparing procedural programming languages to nonprocedural programming languages.

Task Number 40

Differentiate between SQL and PL/SQL.

Definition

Differentiation should include

- comparing the generation of the languages
- comparing the language types—procedural or nonprocedural.

Task Number 41

Explain the need for PL/SQL.

Definition

Explanation should include

- the limitations of PL/SQL
- the capabilities of procedural logic
- the benefits of modularized program development.

Task Number 42

Describe the structure of a PL/SQL block.

Definition

Description should include

- the declarative section
- the executable section
- the exception-handling section.

Task Number 43

Identify the different types of PL/SQL blocks.

Definition

Identification should include

- an anonymous block
- a subprogram in the form of a procedure
- a subprogram in the form of a function.

Using PL/SQL Program Data

Task Number 44

Use variables in PL/SQL.

Definition

Use should include

- defining temporary storage of data with a variable
- manipulating data
- reusing a variable.

Task Number 45

Describe valid and invalid identifiers in PL/SQL.

Definition

Description should include

- the PL/SQL objects
- the properties of an identifier.

Task Number 46

Describe reserved words, delimiters, literals, and comments in PL/SQL.

Definition

Description should include

- the process of locating a complete list of PL/SQL reserved words
- the list of common PL/SQL reserved words
- simple and compound delimiters
- classifications of literals.

Task Number 47

List data types used in PL/SQL.

Definition

List should include

- defining the data types and describing their necessity
- listing the categories of data types
- classifying a data type into a data-type category
- describing the characteristics of the data types.

Task Number 48

Identify the benefits of anchoring data types with the %TYPE attribute.

Definition

Identification should include

- advantages of defining a variable with the same data type as the data populating the variable from a table

- the way PL/SQL code is intertwined with table design and the definition of variables in the PL/SQL code.

Task Number 49

Use built-in SQL functions in PL/SQL.

Definition

Use should include using single-row and multi-row SQL functions.

Related Standards of Learning

Mathematics

COM.7

The student will select and call library functions to process data, as appropriate.

Task Number 50

Differentiate between implicit and explicit conversions.

Definition

Differentiation should include

- contrasting implicit and explicit conversions of data types
- describing when implicit conversions of data types take place
- listing drawbacks of implicit data type conversions.

Task Number 51

Demonstrate how functions can be used to explicitly convert data types.

Definition

Demonstration should include using the following functions:

- TO_CHAR
- TO_NUMBER

- TO_DATE

Using SQL in PL/SQL

Task Number 52

Evaluate the SQL statements that can be directly included in a PL/SQL executable block.

Definition

Evaluation should include determining whether

- the SELECT statement can be implemented into a PL/SQL block
- Data Manipulation Language (DML) statements including INSERT, UPDATE, and DELETE can be used to make changes to data
- Data Definition Language (DDL) and Data Control Language (DCL) statements can be included in PL/SQL
- Transaction Control Language statements, such as COMMIT, ROLLBACK, and SAVEPOINT, can be used to make permanent changes to the database or to discard.

Task Number 53

Retrieve data in PL/SQL.

Definition

Retrieving data requires

- implementing the SELECT statement into a PL/SQL block
 - constructing the INTO clause to hold the values returned by the SQL SELECT statement
 - selecting data columns from a table and placing the data into variables.
-

Task Number 54

Manipulate data with DML statements in PL/SQL.

Definition

Manipulating data includes

- implementing a PL/SQL block that will INSERT new rows into a table
- implementing a PL/SQL block that will UPDATE existing rows in a table
- implementing a PL/SQL block that will DELETE rows from a table
- reviewing the MERGE statement.

Task Number 55

Use SQL cursor attributes.

Definition

Using a SQL cursor attribute includes

- defining an implicit cursor and differentiating it from an explicit cursor
 - implementing cursor attributes for implicit cursors: SQL%FOUND, SQL%NOTFOUND, and SQL%ROWCOUNT in a PL/SQL block.
-

Task Number 56

Use transaction control statements in PL/SQL.

Definition

Using transaction control statements includes

- defining a transaction
 - paralleling a transaction to the SQL required to complete the transaction
 - implementing transaction control statements, such as COMMIT, ROLLBACK, and SAVEPOINT, to make changes to the database permanent or to discard those changes.
-
-

Constructing PL/SQL Program Structures to Control Execution Flow

Task Number 57

Identify the uses and types of conditional control structures.

Definition

Identification should include

- describing the use of conditional control structures
- listing the types of conditional control structures.

Task Number 58

Construct and use an IF statement.

Definition

Construction and use of an IF statement should include

- considering a program that will have alternative courses of action
- writing a program in PL/SQL that contains alternative courses of action within a block, based on a condition
- executing the PL/SQL program that contains alternative courses of action.

Task Number 59

Construct and use an IF-THEN-ELSE statement.

Definition

Construction and use an IF-THEN-ELSE statement should include

- describing the function of THEN in a program that considers alternative courses of action
- describing the function of ELSE in a program that considers alternative courses of action
- describing a program that will have alternative courses of action and multiple tests on a condition or conditions
- writing a program in PL/SQL that contains alternative courses of action within a block, based on a condition, adds additional statements based on additional conditions, and has a last alternative

- executing the PL/SQL program that contains alternative courses of action and multiple conditions.

Task Number 60

Use basic loops with EXIT conditions.

Definition

Use should include

- determining the need for LOOP statements
- recognizing different types of LOOP statements
- determining the need to have an EXIT from a LOOP
- writing a program in PL/SQL that contains a flow, runs once, and stops.

Task Number 61

Use basic loops with EXIT WHEN conditions.

Definition

Using should include

- defining when to use an EXIT and when to use an EXIT WHEN statement in a LOOP
- writing a program in PL/SQL that contains a flow and runs until a conditional termination is reached.

Task Number 62

Use WHILE loops.

Definition

Use should include

- describing when the WHILE loop is needed in PL/SQL
- defining the WHILE loop construct and the condition for exit
- reviewing the syntax on the WHILE loop
- writing a program in PL/SQL that contains an iterative loop and runs until a condition is met.

Task Number 63

Use FOR loops.

Definition

Use should include

- describing when the FOR LOOP is needed in PL/SQL
- defining the FOR LOOP construct and the condition for exit
- reviewing the syntax on the FOR LOOP
- writing a program in PL/SQL that sets a control statement and runs an iterative loop until the control statement is met.

Task Number 64

Use nested loops.

Definition

Use should include

- identifying which loops can be placed within one another
- describing loop labels
- writing a program in PL/SQL that has at least two loops, one inside of the other.

Using Cursors

Task Number 65

Distinguish between an implicit and an explicit cursor.

Definition

Distinction should include

- defining *cursor*, with regards to PL/SQL
- describing where the cursor exists
- describing when an implicit cursor is created

- describing when one would use an explicit cursor.

Task Number 66

Declare and control explicit cursors.

Definition

Declaring and controlling explicit cursors should include writing a program in PL/SQL that

- declares a cursor by
 - - naming the cursor
 - selecting the data columns
 - selecting the files
 - selecting the rows
- POPULATES the cursor
- OPENS the cursor
- FETCHS data from the cursor
- CLOSES the cursor.

Task Number 67

Use cursor simple LOOP to FETCH data.

Definition

Use should include writing a program in PL/SQL that uses a declared cursor, and, after OPENING the cursor

- creates a simple loop using LOOP...END LOOP
- manipulates or reads data FETCHed within the loop into the cursor.

Task Number 68

Use cursor FOR LOOP to fetch data.

Definition

Use should include

- defining the difference between a cursor simple loop and a cursor FOR LOOP
- reviewing the syntax of the cursor FOR LOOP

- writing a program in PL/SQL that
 - declares and populates a cursor to be used in the cursor FOR LOOP
 - creates an iterative process by defining a FOR LOOP that reads a record into the declared cursor
 - reads while the record contains data.

Task Number 69

Declare and use cursors with parameters.

Definition

Declaration and use should include

- defining the use of a parameter in a cursor in PL/SQL
- correlating or translating business needs into the use parameters in cursors.

Declaring a cursor with parameters should include writing a program in PL/SQL that declares a cursor by

- naming the cursor
- defining the variable that can be passed
- selecting the data columns
- selecting the files
- selecting the rows.

Using a cursor with parameters should include

- using a simple loop example:
 - OPEN the cursor, providing a value to the variable to be passed.
 - Create a LOOP.
 - FETCH the cursor.
 - Manipulate, read the data.
 - Close the LOOP.
 - Close the cursor.
- using a FOR LOOP example
- using the parameter in a FOR LOOP program.

Task Number 70

Lock rows using the FOR UPDATE clause.

Definition

Locking rows should include

- defining the FOR UPDATE clause
- writing a program in PL/SQL that includes
 - a cursor
 - a user requirement to prevent another user from opening the records while the UPDATE is executed.

Task Number 71

Reference the current row with the WHERE CURRENT clause.

Definition

Reference should include

- describing where in a PL/SQL program code the FOR UPDATE clause is allowed and where the WHERE CURRENT OF clause is allowed
- writing a program in PL/SQL that
 - performs an UPDATE
 - uses the FOR UPDATE clause in the query to lock the rows
 - uses the WHERE CURRENT OF to update the row.

Task Number 72

Use multiple cursors.

Definition

Use should include

- describing when one would use multiple cursors
- writing a program in PL/SQL that
 - declares a cursor
 - uses the cursor in either a simple loop or a FOR loop
 - manipulates and reads the data.

Using Exception Handling

Task Number 73

Describe the function of exceptions.

Definition

Description should include

- defining *exception*
- listing possible causes of exceptions
- identifying terminology that relates to exceptions.

Task Number 74

Describe the function of an exception handler.

Definition

Description should include

- defining *exception handler*
- listing possible uses of an exception handler
- identifying the importance of an exception handler in PL/SQL.

Task Number 75

Handle exceptions in PL/SQL programs.

Definition

Handling should include

- listing the types of exceptions that can occur in a PL/SQL program
- describing the way an error can be trapped in a PL/SQL program
- writing a program that
 - includes an EXCEPTION section
 - traps the error(s)
 - reads the error(s) and provides new instructions to the program that allows the program to resume.

Task Number 76

Trap predefined Oracle server exceptions.

Definition

Trapping should include

- describing the way that Oracle produces errors that are implicitly raised
- determining which exceptions are predefined Oracle server errors
- writing a program that
 - includes an EXCEPTION section
 - traps a predefined Oracle server error
 - reads the error and provides new instructions to the program that allows the program to resume.

Task Number 77

Trap non-predefined Oracle server errors.

Definition

Trapping non-predefined Oracle server errors should include

- describing non-predefined errors
- reading the error when raised
- writing a program that
 - declares an Oracle error
 - includes an EXCEPTION section
 - traps the Oracle error declared
 - reads the error and gives new instructions to the program that allows the program to resume.

Task Number 78

Trap user-defined errors.

Definition

Trapping user-defined errors should include

- describing situations in which a developer determines a condition is abnormal and needs to be trapped

- analyzing the procedure to explicitly raise an exception
- writing a program that
 - declares an Oracle error
 - raises an error defined in the declaration section
 - includes an EXCEPTION section
 - traps the error raised in the EXCEPTION section
 - reads the error and provides new instructions to the program that allow the program to resume.

Task Number 79

Describe scope of exceptions.

Definition

Description should include

- defining *scope of exceptions*
- recognizing each exception-scope with the exception in nested blocks
- describing the propagation of exceptions in nested blocks.

Examining PL/SQL Composite Data Types

Task Number 80

Describe user-defined PL/SQL records.

Definition

Description should include:

- structure of a user-defined RECORD
- uses for a user-defined RECORD.

Task Number 81

Create a user-defined PL/SQL record.

Definition

Creating a user-defined PL/SQL record should include

- selecting the data
- declaring field data type (e.g., %TYPE, %ROWTYPE, RECORDs)
- referring to the record in the code execution section.

Task Number 82

Use PL/SQL INDEX BY Table and INDEX BY Table of Records.

Definition

Use should include

- using the primary key to access and find data
- demonstrating the use of built-in Oracle functions to reference elements in the table:
 - EXISTS
 - COUNT
 - FIRST/LAST
 - PRIOR/NEXT
 - DELETE
 - TRIM

Creating and Managing Procedures

Task Number 83

Identify the characteristics and benefits of a stored procedure.

Definition

Identification should include

- anonymous blocks

- definition of *subprograms*
- benefits of subprograms
- definition of *stored procedure*.

Task Number 84

Invoke a stored procedure.

Definition

Procedure might include invoking a stored procedure in PL/SQL from

- an anonymous block
- another procedure
- a calling application.

Task Number 85

List the steps for creating a procedure.

Definition

List should include

- defining the create procedure process
- defining the process of writing the PL/SQL procedure block
- reviewing examples of PL/SQL procedures.

Task Number 86

Create a procedure with parameters.

Definition

Creating a procedure should include

- identifying the principles of using procedures with parameters
- programming the CREATE or REPLACE PROCEDURE code, including the definition of the parameters
- programming the PL/SQL procedure block using the parameters defined in the procedure definition.

Task Number 87

Invoke a procedure that has parameters.

Definition

Invoking a procedure that has parameters should include

- creating an anonymous block in PL/SQL with a direct call to the stored procedure
- providing the procedure name and the parameter values
- invoking the procedure created from another procedure
- calling the new procedure in the correct order.

Task Number 88

List the types of parameter modes.

Definition

List should include

- the formal type of parameter
- the actual type of parameter
- IN parameters
- OUT parameters.

Task Number 89

Describe the DEFAULT option for parameters.

Definition

Description should include identifying the IN parameter mode and when it should be used.

Task Number 90

Describe the method for propagating exceptions.

Definition

Description should include

- explaining how exceptions are handled in a block and what happens when a block does not contain a method to handle an exception
- identifying the EXCEPTION section
- determining the exception raised
- determining exception handling
- explaining the OTHERS handler.

Task Number 91

Remove a procedure.

Definition

Removal should include implementing the DROP PROCEDURE command.

Task Number 92

Identify how to view and manage procedures.

Definition

Identification should include

- a review of the three classes of tables in the data dictionary:
 - USER tables
 - ALL tables
 - DBA tables
- the USER_OBJECTS table that define the type and name of an object
- the USER_SOURCECODE table that contains source code for subprograms
- the ALL_SOURCE table that contains source code for all the subprograms that one has privileges to invoke.

Creating and Managing Functions

Task Number 93

Define *stored function*.

Definition

Definition should include

- identifying the definition of subprograms
- describing the parameters of a function
- describing the value returned by a function.

Task Number 94

Create a function.

Definition

Creation should include

- programming the CREATE FUNCTION command with only an IN parameter mode and the RETURN clause
- programming the PL/SQL function/subprogram block
- referring to examples of PL/SQL functions.

Task Number 95

List the procedures for invoking a function.

Definition

List should include

- invoking a function as part of PL/SQL expressions by using a local variable in the autonomous block to hold the function's returned value
- invoking a function as a parameter to another subprogram and passing functions between the subprograms
- invoking a function as an expression in a SQL statement.

Task Number 96

List the advantages of user-defined functions in SQL statements.

Definition

List should include

- the advantages of functions in the WHERE clause
- the advantages of functions directly in the SELECT clause
- an explanation of the way functions can manipulate data
- description of code reduction with the implementation of functions.

Task Number 97

List circumstances in which user-defined functions can be called from within a SQL statement.

Definition

List should include

- the SELECT statement and a description of the way functions are used in this clause
- conditional expressions in SQL statements, such as WHERE and HAVING
- functions in ORDER BY and GROUP BY
- functions used in the VALUES clause in an INSERT statement
- the SET clause in the UPDATE statement.

Task Number 98

Describe the restrictions on calling functions from SQL statements.

Definition

Description should include

- the IN parameter
- the RETURN from a function, and the SQL size limits on the returned value
- an identification of datatypes that are allowed in PL/SQL and not allowed in SQL.

Task Number 99

Remove a function.

Definition

Removal should include implementing the DROP FUNCTION command.

Task Number 100

Identify how to view stored objects in the data dictionary.

Definition

Identification should include

- the three classes of tables in the data dictionary:
 - USER tables
 - ALL tables
 - DBA tables
- the USER_OBJECTS table that defines an object_type as a function
- the USER_SOURCECODE table that contains source code for subprograms
- the ALL_SOURCE table that contains source code for all the subprograms that one has privileges to invoke.

Task Number 101

Identify differences between invoker and definer rights.

Definition

Identification should include

- defining *definer* in relation to a PL/SQL program
- explaining the way a program executes when definer's rights are used:
 - The definer needs privileges on the database objects referenced by the subprogram.
 - The invoker needs EXECUTE privileges on the subprogram.
- defining *invoker* in relation to a PL/SQL program
- explaining the way a program executes when invoker's rights are used:
 - The invoker needs EXECUTE privileges on the subprogram and needs privileges on the database objects referenced by the subprogram.
 - The definer does not need any privileges.
- reviewing examples in each scenario.

Designing Packages

Task Number 102

Identify a package specification and body.

Definition

Identification should include

- the contents of a package in PL/SQL
- the purpose and constructs of a package specification
- the contents and location of a package body.

Task Number 103

Create packages (e.g., related variables, cursors, constants, exceptions, procedures, and functions).

Definition

Creation of packages should include

- writing a program for the package specification:
 - programming the CREATE or REPLACE PACKAGE code naming the package
 - defining the variables and constructs to be used in the package
- writing a program for the package body:
 - programming the CREATE or REPLACE BODY code naming the package body to include the subprograms that were defined in the package specification within a BEGIN – END block.

Task Number 104

Invoke a package construct.

Definition

Invoking a package construct should include

- using the dot-prefix naming conventions
- using the method to invoke subprograms in anonymous blocks
- calling a package and its subprogram(s) and passing the parameters.

Task Number 105

Designate package constructs as public or private.

Definition

Designation should include

- identifying the differences between public and private components
- describing the visibility of package components based on where they are declared
- declaring the public component (for public components) in the package specification
- declaring the private component (for private components) in the package body.

Task Number 106

Drop packages.

Definition

Dropping packages should include

- dropping entire packages by implementing the DROP PACKAGE command
- dropping just the package body by implementing the DROP PACKAGE BODY command.

Task Number 107

Identify benefits of packages.

Definition

Identification should include

- encapsulation
- modularity
- the value of hiding code (only the declarations in the package specification are visible; private components are hidden)
- the execution of packages and processing speed.

Task Number 108

Create packages that use the overloading feature.

Definition

Creation should include

- naming two or more subprograms with the same name
- requiring these subprograms to accept similar sets of parameters with different data types.

Task Number 109

Identify restrictions on using packaged functions in SQL statements.

Definition

Identification should include describing functions eligible to be called, and all other solutions are restricted.

Task Number 110

Invoke packaged functions from SQL.

Definition

Invoking the packaged function should be performed by calling the function and passing any required parameters.

Task Number 111

Identify persistent states in package variables and cursors.

Definition

Identification should include

- package variables—identifying when the package state is initialized, determining the duration that the package state is persistent
- package cursors—identifying when the package state is initialized, describing whether the cursor is opened or closed; if open, describe the number of rows that were fetched by the cursor

Task Number 112

Describe the use and application of some Oracle server-supplied packages (e.g., DBMS_SQL, DBMS_OUTPUT, and UTL_FILE).

Definition

Description should include

- the debugging process
- some of the functions within the standard Oracle packages
- how to work with the operating system from within PL/SQL
- exceptions within the Oracle server-supplied packages.

Task Number 113

Use EXECUTE IMMEDIATE.

Definition

Use should include

- defining where EXECUTE IMMEDIATE is allowed to be used in PL/SQL
- describing how EXECUTE IMMEDIATE is allowed to be used in PL/SQL
- including EXECUTE IMMEDIATE within a procedure or function to demonstrate syntax and code use.

Task Number 114

Describe the benefits of EXECUTE IMMEDIATE over DBMS_SQL for native dynamic SQL.

Definition

Description should include

- native dynamic SQL
- EXECUTE IMMEDIATE in native dynamic SQL
- DBMS_SQL
- programming requirements for native dynamic SQL and DBMS_SQL package
- execution time for native dynamic SQL and DBMS_SQL package.

Creating and Removing Triggers

Task Number 115

Describe database triggers and types and the uses of each.

Definition

Description should include

- where database triggers are stored
- which database triggers run against
- how database triggers execute
- a comparison of application triggers and database triggers
- the events that can cause a database trigger to fire.

Task Number 116

Create a DML trigger.

Definition

Creation should include the following:

- Name—Defining the trigger name
- Timing—Determining when the trigger will execute; before, after, or instead of the triggering DML statement
- Event—Determining what DML statement the trigger will fire against
- Object name—Determining table or view associated with the trigger
- Iterations—Determining how many times the trigger will fire (e.g., once per loop)
- Trigger body—Writing a PL/SQL block to include the actions of the trigger

Task Number 117

List the DML trigger components.

Definition

List should include

- timing
- event
- object name
- trigger body.

Task Number 118

Create a statement-level trigger.

Definition

Creation should include determining when the trigger should fire:

- BEFORE—Before the triggering DML event on a table
 - AFTER—After the triggering DML event on a table
 - INSTEAD OF—Instead of the triggering DML event on a view
-

Task Number 119

Describe the trigger-firing sequence options.

Definition

Description should include

- INSERT
- UPDATE (column)
- DELETE.

Add logic to the operations above to determine when the trigger is fired.

Task Number 120

Use conditional predicates in a DML trigger.

Definition

Use should include

- demonstration of the IF and ELSIF statements in PL/SQL
- determination of when conditional predicates might be implemented

- database trigger with conditional predicates, including the statements INSERTING, DELETING, and UPDATING.

Task Number 121

Create a row-level trigger.

Definition

Creation should include

- differentiating a row trigger from a statement trigger
- implementing the FOR EACH ROW clause in a trigger statement.

Task Number 122

Use the OLD and NEW qualifiers in a database trigger.

Definition

Use should include writing a row trigger that references OLD.column_name and NEW.column_name to work with the pre- and post-updated values.

Task Number 123

Create an INSTEAD OF trigger.

Definition

Creation should include

- differentiating between a view and a complex view
- determining when data in a view can be updated
- writing a trigger that will update the tables that are used to create a view INSTEAD OF the present view.

Task Number 124

Describe events that cause database triggers to fire.

Definition

Description should include

- database start-up and shut-down
- user connection
- errors raised.

Task Number 125

Create a trigger for a DDL statement.

Definition

Creation should include the following:

- Describing the ON DATABASE clause
- Describing the ON SCHEMA clause
- Reviewing triggers for other DDL statements
- Writing the trigger:
 - Trigger name—Defining the trigger name
 - Timing—Determining when the trigger will execute; before, after, or instead of
 - Event—Determining the database event and ON DATABASE or ON SCHEMA
 - Trigger body—Writing a PL/SQL block to include the actions of the trigger

Task Number 126

Create a trigger for a system event.

Definition

Creation should include

- reviewing system events (e.g., LOGON, LOGOFF, SERVERERROR)
- writing the trigger:
 - Trigger name—Defining the trigger name
 - Timing—Determining when the trigger will execute; before, after, or instead of
 - Event—Determining the system event and the ON DATABASE or ON SCHEMA
 - Trigger body—Writing a PL/SQL block to include the actions of the trigger

Task Number 127

Describe the functionality of the CALL statement.

Definition

Description should include

- reviewing a PL/SQL trigger that CALLS another procedure/sub-program
- syntax of the CALL statement.

Task Number 128

Describe the effect of a mutating table.

Definition

Description should include the following:

- A mutating table cannot be read or changed by another process.
- A mutating table can only be called by specific triggers.

Task Number 129

View trigger information in the dictionary views.

Definition

The following data dictionary views contain information about triggers:

- USER_TRIGGERS
- ALL_TRIGGERS
- DBA_TRIGGERS

Columns indicate the following:

- The new column, BASE_OBJECT_TYPE, specifies whether the trigger is based on DATABASE, SCHEMA, table, or view.
- The old column, TABLE_NAME, is null if the base object is not table or view.
- The column ACTION_TYPE specifies whether the trigger is a call-type trigger or a PL/SQL trigger.
- The column TRIGGER_TYPE includes two additional values: BEFORE EVENT and AFTER EVENT, applicable only to system events.
- The column TRIGGERING_EVENT includes all system and DML events.

Task Number 130

Alter a trigger status.

Definition

Alteration should include

- writing an SQL statement to DISABLE and REENABLE a trigger
- implementing DISABLE and ENABLE for all triggers for a table
- acting to RECOMPILE a trigger.

Task Number 131

Remove a trigger.

Definition

Removal should include writing a program using the DROP TRIGGER statement.

Using Advanced Data Types

Task Number 132

Describe the function of LOB (large object) data types.

Definition

Description should include the need for retaining data that is stored in files that are picture, photo, documents, programs, or video. Large object file types include

- CLOB (character large object)
- BLOB (binary large object)
- BFILE (binary file)
- NCLOB (national character large object).

Task Number 133

Migrate from LONG to LOB.

Definition

Migration should include

- identifying the specifications of the LONG data type
- identifying the specifications of the LOB data type
- identifying the SQL explicit convert functions
- using the ALTER table command MODIFY LONG column to a LOB column.

Task Number 134

Manage BFILES.

Definition

Management should include

- identifying where BFILES are stored
- identifying media storage options for a BFILE
- determining how the file can be used, read, modified, and searched
- determining when to use and when not to use the BFILE data type
- creating a DIRECTORY object in Oracle to control access to BFILES.

Understanding Procedural Dependencies

Task Number 135

Describe the implications of procedural dependencies.

Definition

Description should include the following:

- PL/SQL objects reference other PL/SQL objects.
- It is possible to affect the dependent object and the accuracy of the program by altering the definition of a reference object.

Task Number 136

Describe dependent objects and referenced objects.

Definition

Description should include listing all objects and determining if each is

- dependent on another object
- referenced by another object
- both of the above.

Task Number 137

View dependency information in the dictionary views.

Definition

Viewing dependency information should include

- identifying USER_DEPENDENCIES
- identifying ALL_DEPENDENCIES
- writing a select statement to see the dependency information in the dictionary views.

Task Number 138

Use the UTLDTREE script.

Definition

Procedure should include running the Oracle script utldtree.sql.

Task Number 139

Use the IDEPTREE and DEPTREE views.

Definition

Procedure should include selecting all columns from each view to see available data and working with the data as necessary.

Task Number 140

List the procedures for minimizing dependency failures.

Definition

List should include

- %ROWTYPE
- variables declared with %TYPE
- SELECT *
- column list with INSERT statements.

Using the PL/SQL Compiler

Task Number 141

Describe PL/SQL initialization parameters.

Definition

Description should include

- how PLSQL_CODE_TYPE can improve execution speed
- how PLSQL_OPTIMIZE_LEVEL can improve execution speed.

Task Number 142

Use PL/SQL initialization parameters.

Definition

Use USER_PLSQL_OBJECT_SETTINGS to see how a PL/SQL program was compiled.

Task Number 143

Identify compiler warning messages.

Definition

Identification should include

- explaining the similarities and differences between a warning and an error
- comparing the warning levels that can be set by the PLSQL_WARNINGS parameter
- calling the DBMS_WARNING server-supplied package from within a PL/SQL program dependent on another object and referenced by another object.

Task Number 144

Describe conditional compilation.

Definition

Description should include

- benefits of a conditional compilation
- conditional compilation's role in a PL/SQL program, referring to selection, inquiry, and error directives.

Task Number 145

Describe the hiding of source code.

Definition

Description should include

- benefits of obfuscated PL/SQL source code
- concepts of the DBMS_DDL.CREATE_WRAPPED server-supplied procedure
- use of the Wrap utility to obfuscate PL/SQL source code.

Preparing for Industry Certification

Task Number 146

Describe the process and requirements for obtaining industry certifications related to the Database Design and Management with PL/SQL (Oracle) course.

Definition

The description should include a list of industry certifications related to the Database Design and Management with PL/SQL (Oracle) course and the process/requirements for obtaining the certifications from

- official websites of the testing organization/vendor
- practice materials and tests based on information from the testing organization/vendor
- information from certified instructors or industry-certified professionals
- information in the "Introduction/Course Description" section of this document.

FBLA Competitive Events and Activities Areas

Job Interview

Task Number 147

Identify testing skills and strategies for a certification examination.

Definition

Identification should be undertaken by

- conducting an Internet research project
- reviewing materials from exam and practice-exam publishers
- interviewing certified instructors and/or industry-certified professionals.

FBLA Competitive Events and Activities Areas

Job Interview

Task Number 148

Demonstrate ability to successfully complete selected practice examinations (e.g., practice questions similar to those on certification exams).

Definition

Demonstration should include successfully completing practice examinations for selected certifications related to the course obtained from vendor sites and/or materials from publishers. The level of performance on a practice examination serves as a gauge of the applicant's readiness for formal industry testing.

FBLA Competitive Events and Activities Areas

Job Interview

Task Number 149

Complete an industry certification examination representative of skills learned in this course (e.g., MOS, MTA, IC3).

Definition

Completion will be achieved when the student applicant earns an examination score deemed "passing" by the testing organization. Qualifying examinations are those currently approved at the state level as representative of Database Design and Development with PL/SQL (Oracle) skills.

Students should be encouraged to attain industry certification as evidence of their database design and management skill level and general employability.

Developing Employability Skills

Task Number 150

Investigate job opportunities, using the Internet and other sources.

Definition

Investigation should match individual abilities, aptitudes, and job expectations with

- careers available in the database design field
- requirements to enter each career
- industry employment trends
- career-advancement opportunities.

Task Number 151

Investigate careers in the information technology field.

Definition

Identification should include

- job descriptions
- education requirements
- projected earnings
- industry outlook
- other career trends.

Task Number 152

Research certification and educational opportunities.

Definition

Research should include

- IC3 Certification
- Microsoft Certified Application Specialist (MCAS)
- Microsoft Office Specialist (MOS)
- Microsoft Technology Associate (MTA)
- Oracle Certification Program Examinations
- Virginia Workplace Readiness Assessment and Digital Literacy Certification Examination
- Virginia Workplace Readiness Assessment and IC3 Certification.

Task Number 153

Compose or update a printed résumé.

Definition

Résumé should include periodic updates of

- educational background
- work history
- honors and awards
- membership in club and/or community activities, leadership positions held, and community service.

Task Number 154

Prepare an electronic résumé.

Definition

Electronic or online résumé should include complete information and be posted and updated only at reliable career sites. Other considerations include the following:

- Posting at pay versus free sites
- The way file types are affected by Internet posting (HTML versus Word)
- The way file sizes affect email transmission and reception
- Security and confidentiality issues

Task Number 155

Create a letter of application.

Definition

The letter of application should include

- appropriate business-letter format
- three or four short paragraphs emphasizing the salient points of the résumé, identifying what the applicant will bring to the job
- indication of personal knowledge of the company
- addresses of appropriate company contacts
- personal contact information.

Task Number 156

Complete an electronic application form.

Definition

The electronic or online application form should include complete, accurate, and effectively organized information. It should adhere to additional criteria specifically related to electronic transmittal of such information (e.g., attention to security concerns, inclusion of keywords to enhance interest in the application, use of scanner-friendly format).

Task Number 157

Create an interview follow-up letter.

Definition

The brief follow-up letter in business-letter format should include

- acknowledgment of the interview on a specific date
- appreciation for the interviewer's time
- reminder of qualifications
- reiteration of continued interest in the job
- request for further action, such as a second interview or meeting.

Task Number 158

Identify the steps to follow in resigning from a position.

Definition

Identification should include

- submitting an oral or written resignation from the job
- providing ample time for a replacement to be found—usually two weeks
- offering to train the replacement.

Task Number 159

Develop a portfolio containing representative samples of student's work (e.g., program design, technical documentation and design, source code, and output).

Definition

Portfolio should include a résumé and a combination of electronic and non-electronic documents representative of qualifications. Items included should reflect knowledge, skills, and ability.

Task Number 160

Deliver an oral presentation of portfolio.

Definition

Informative and persuasive presentations should be made by

- gathering and organizing evidence to support a position
- presenting evidence clearly and convincingly
- using grammatically correct language, including vocabulary appropriate to the topic, audience, and purpose.

Oral presentation should include references to the main features of the résumé and describe features of other examples of qualifications. Personal background information should focus on professional knowledge, skills, and abilities.

Task Number 161

Identify potential employment barriers for nontraditional groups and ways to overcome barriers.

Definition

Identification of employment barriers should include any form of discrimination (e.g., gender, ethnicity, age, religion) in hiring or promoting. Ways to overcome the barriers include

- scholarships
- job training programs
- mentorships
- minority assistance programs.

SOL Correlation by Task

39	Explain PL/SQL.	English: 10.5, 11.5, 12.5
40	Differentiate between SQL and PL/SQL.	English: 10.5, 11.5, 12.5
41	Explain the need for PL/SQL.	English: 10.5, 11.5, 12.5
42	Describe the structure of a PL/SQL block.	English: 10.5, 11.5, 12.5
43	Identify the different types of PL/SQL blocks.	English: 10.5, 11.5, 12.5

44	Use variables in PL/SQL.	English: 10.3, 11.3, 12.3 Mathematics: COM.15
45	Describe valid and invalid identifiers in PL/SQL.	English: 10.5, 11.5, 12.5 Mathematics: COM.15
46	Describe reserved words, delimiters, literals, and comments in PL/SQL.	English: 10.5, 10.6, 11.5, 11.6, 12.5, 12.6 Mathematics: COM.15, COM.16
47	List data types used in PL/SQL.	English: 10.3, 10.5, 11.3, 11.5, 12.3, 12.5 Mathematics: COM.15, COM.16
48	Identify the benefits of anchoring data types with the %TYPE attribute.	English: 10.3, 10.5, 11.3, 11.5, 12.3, 12.5
49	Use built-in SQL functions in PL/SQL.	Mathematics: COM.7
50	Differentiate between implicit and explicit conversions.	English: 10.5, 10.6, 11.5, 11.6, 12.5, 12.6 Mathematics: COM.16
51	Demonstrate how functions can be used to explicitly convert data types.	Mathematics: COM.16
52	Evaluate the SQL statements that can be directly included in a PL/SQL executable block.	English: 10.3, 10.5, 11.3, 11.5, 12.3, 12.5
53	Retrieve data in PL/SQL.	
54	Manipulate data with DML statements in PL/SQL.	Mathematics: COM.14
55	Use SQL cursor attributes.	
56	Use transaction control statements in PL/SQL.	
57	Identify the uses and types of conditional control structures.	English: 10.5, 10.6, 11.5, 11.6, 12.5, 12.6 Mathematics: COM.8
58	Construct and use an IF statement.	English: 10.6, 10.7, 11.6, 11.7, 12.6, 12.7 Mathematics: COM.3, COM.8
59	Construct and use an IF-THEN-ELSE statement.	English: 10.5, 11.5, 12.5 Mathematics: COM.8
60	Use basic loops with EXIT conditions.	English: 10.5, 11.5, 12.5 Mathematics: COM.13

61	Use basic loops with EXIT WHEN conditions.	English: 10.3, 10.6, 10.7, 11.3, 11.6, 11.7, 12.3, 12.6, 12.7 Mathematics: COM.13
62	Use WHILE loops.	English: 10.3, 10.5, 10.6, 10.7, 11.3, 11.5, 11.6, 11.7, 12.3, 12.5, 12.6, 12.7 Mathematics: COM.13
63	Use FOR loops.	English: 10.3, 10.5, 10.6, 10.7, 11.3, 11.5, 11.6, 11.7, 12.3, 12.5, 12.6, 12.7 Mathematics: COM.13
64	Use nested loops.	English: 10.3, 10.6, 10.7, 11.3, 11.6, 11.7, 12.3, 12.6, 12.7 Mathematics: COM.13
65	Distinguish between an implicit and an explicit cursor.	English: 10.3, 10.5, 11.3, 11.5, 12.3, 12.5
66	Declare and control explicit cursors.	Mathematics: COM.3
67	Use cursor simple LOOP to FETCH data.	Mathematics: COM.3, COM.13
68	Use cursor FOR LOOP to fetch data.	English: 10.3, 10.5, 10.6, 10.7, 11.3, 11.5, 11.6, 11.7, 12.3, 12.5, 12.6, 12.7 Mathematics: COM.13
69	Declare and use cursors with parameters.	English: 10.3, 10.5, 11.3, 11.5, 12.3, 12.5 Mathematics: COM.16
70	Lock rows using the FOR UPDATE clause.	English: 10.3, 10.6, 10.7, 11.3, 11.6, 11.7, 12.3, 12.6, 12.7
71	Reference the current row with the WHERE CURRENT clause.	English: 10.5, 10.6, 10.7, 10.8, 11.5, 11.6, 11.7, 11.8, 12.5, 12.6, 12.7, 12.8
72	Use multiple cursors.	English: 10.5, 10.6, 10.7, 11.5, 11.6, 11.7, 12.5, 12.6, 12.7 Mathematics: COM.9

73	Describe the function of exceptions.	English: 10.3, 10.5, 10.6, 10.7, 11.3, 11.5, 11.6, 11.7, 12.3, 12.5, 12.6, 12.7
74	Describe the function of an exception handler.	English: 10.5, 10.6, 10.7, 11.5, 11.6, 11.7, 12.5, 12.6, 12.7
75	Handle exceptions in PL/SQL programs.	English: 10.5, 10.6, 10.7, 11.5, 11.6, 11.7, 12.5, 12.6, 12.7 Mathematics: COM.14, COM.18
76	Trap predefined Oracle server exceptions.	English: 10.5, 10.6, 10.7, 11.5, 11.6, 11.7, 12.5, 12.6, 12.7 Mathematics: COM.14, COM.18
77	Trap non-predefined Oracle server errors.	English: 10.5, 10.6, 10.7, 11.5, 11.6, 11.7, 12.5, 12.6, 12.7 Mathematics: COM.14, COM.18
78	Trap user-defined errors.	English: 10.5, 10.6, 10.7, 11.5, 11.6, 11.7, 12.5, 12.6, 12.7 Mathematics: COM.14, COM.18
79	Describe scope of exceptions.	English: 10.3, 10.5, 11.3, 11.5, 12.3, 12.5
80	Describe user-defined PL/SQL records.	English: 10.5, 11.5, 12.5
81	Create a user-defined PL/SQL record.	English: 10.1, 10.5, 11.1, 11.5, 12.1, 12.5
82	Use PL/SQL INDEX BY Table and INDEX BY Table of Records.	
83	Identify the characteristics and benefits of a stored procedure.	English: 10.3, 10.5, 11.3, 11.5, 12.3, 12.5 Mathematics: COM.5, COM.16
84	Invoke a stored procedure.	Mathematics: COM.9, COM.14

85	List the steps for creating a procedure.	English: 10.3, 10.5, 11.3, 11.5, 12.3, 12.5 Mathematics: COM.3, COM.4
86	Create a procedure with parameters.	English: 10.5, 11.5, 12.5 Mathematics: COM.14, COM.16
87	Invoke a procedure that has parameters.	Mathematics: COM.14
88	List the types of parameter modes.	English: 10.6, 10.7, 11.6, 11.7, 12.6, 12.7
89	Describe the DEFAULT option for parameters.	English: 10.5, 11.5, 12.5
90	Describe the method for propagating exceptions.	English: 10.5, 11.5, 12.5 Mathematics: COM.14, COM.18
91	Remove a procedure.	Mathematics: COM.14
92	Identify how to view and manage procedures.	English: 10.5, 11.5, 12.5
93	Define <i>stored function</i> .	English: 10.3, 10.5, 11.3, 11.5, 12.3, 12.5 Mathematics: COM.9
94	Create a function.	English: 10.8, 11.8, 12.8 Mathematics: COM.14
95	List the procedures for invoking a function.	English: 10.6, 10.7, 11.6, 11.7, 12.6, 12.7 Mathematics: COM.16
96	List the advantages of user-defined functions in SQL statements.	English: 10.5, 10.6, 10.7, 11.5, 11.6, 11.7, 12.5, 12.6, 12.7
97	List circumstances in which user-defined functions can be called from within a SQL statement.	English: 10.6, 10.7, 11.6, 11.7, 12.6, 12.7
98	Describe the restrictions on calling functions from SQL statements.	English: 10.5, 11.5, 12.5
99	Remove a function.	Mathematics: COM.14
100	Identify how to view stored objects in the data dictionary.	English: 10.5, 11.5, 12.5
101	Identify differences between invoker and definer rights.	English: 10.3, 10.5, 11.3, 11.5, 12.3, 12.5
102	Identify a package specification and body.	English: 10.5, 11.5, 12.5
103	Create packages (e.g., related variables, cursors, constants, exceptions, procedures, and functions).	English: 10.6, 10.7, 11.6, 11.7, 12.6, 12.7

		Mathematics: COM.3, COM.4, COM.5, COM.14
104	Invoke a package construct.	Mathematics: COM.14
105	Designate package constructs as public or private.	English: 10.5, 11.5, 12.5 Mathematics: COM.3
106	Drop packages.	Mathematics: COM.3
107	Identify benefits of packages.	English: 10.5, 11.5, 12.5
108	Create packages that use the overloading feature.	English: 10.5, 11.5, 12.5 Mathematics: COM.14
109	Identify restrictions on using packaged functions in SQL statements.	English: 10.5, 11.5, 12.5 Mathematics: COM.3, COM.9
110	Invoke packaged functions from SQL.	
111	Identify persistent states in package variables and cursors.	English: 10.5, 11.5, 12.5
112	Describe the use and application of some Oracle server-supplied packages (e.g., DBMS_SQL, DBMS_OUTPUT, and UTL_FILE).	English: 10.5, 11.5, 12.5 Mathematics: COM.9
113	Use EXECUTE IMMEDIATE.	English: 10.3, 10.5, 11.3, 11.5, 12.3, 12.5
114	Describe the benefits of EXECUTE IMMEDIATE over DBMS_SQL for native dynamic SQL.	English: 10.5, 11.5, 12.5
115	Describe database triggers and types and the uses of each.	English: 10.5, 11.5, 12.5
116	Create a DML trigger.	Mathematics: COM.14
117	List the DML trigger components.	English: 10.6, 10.7, 11.6, 11.7, 12.6, 12.7 Mathematics: COM.14
118	Create a statement-level trigger.	
119	Describe the trigger-firing sequence options.	English: 10.5, 11.5, 12.5
120	Use conditional predicates in a DML trigger.	English: 10.5, 11.5, 12.5
121	Create a row-level trigger.	English: 10.5, 11.5, 12.5
122	Use the OLD and NEW qualifiers in a database trigger.	English: 10.6, 11.6, 12.6
123	Create an INSTEAD OF trigger.	English: 10.5, 10.6, 11.5, 11.6, 12.5, 12.6 Mathematics: COM.8
124	Describe events that cause database triggers to fire.	English: 10.5, 11.5, 12.5
125	Create a trigger for a DDL statement.	English: 10.5, 10.6, 11.5, 11.6, 12.5, 12.6

		Mathematics: COM.14
126	Create a trigger for a system event.	English: 10.5, 10.6, 11.5, 11.6, 12.5, 12.6 Mathematics: COM.14
127	Describe the functionality of the CALL statement.	English: 10.5, 11.5, 12.5
128	Describe the effect of a mutating table.	English: 10.5, 11.5, 12.5 Mathematics: COM.18
129	View trigger information in the dictionary views.	English: 10.8, 11.8, 12.8
130	Alter a trigger status.	English: 10.6, 11.6, 12.6 Mathematics: COM.3
131	Remove a trigger.	Mathematics: COM.14
132	Describe the function of LOB (large object) data types.	English: 10.5, 11.5, 12.5 Mathematics: COM.15
133	Migrate from LONG to LOB.	English: 10.5, 11.5, 12.5
134	Manage BFILES.	English: 10.5, 11.5, 12.5
135	Describe the implications of procedural dependencies.	English: 10.5, 11.5, 12.5
136	Describe dependent objects and referenced objects.	English: 10.5, 11.5, 12.5
137	View dependency information in the dictionary views.	English: 10.5, 10.6, 11.5, 11.6, 12.5, 12.6
138	Use the UTLDTREE script.	
139	Use the IDEPTREE and DEPTREE views.	
140	List the procedures for minimizing dependency failures.	English: 10.6, 11.6, 12.6 Mathematics: COM.4, COM.18
141	Describe PL/SQL initialization parameters.	English: 10.5, 11.5, 12.5
142	Use PL/SQL initialization parameters.	
143	Identify compiler warning messages.	English: 10.5, 11.5, 12.5
144	Describe conditional compilation.	English: 10.5, 11.5, 12.5
145	Describe the hiding of source code.	English: 10.5, 11.5, 12.5
146	Describe the process and requirements for obtaining industry certifications related to the Database Design and Management with PL/SQL (Oracle) course.	English: 10.5, 10.8, 11.5, 11.8, 12.5, 12.8
147	Identify testing skills and strategies for a certification examination.	English: 10.1, 10.2, 10.8, 11.1, 11.2, 11.8, 12.1, 12.2, 12.8
148	Demonstrate ability to successfully complete selected practice examinations (e.g., practice questions similar to those on certification exams).	

149	Complete an industry certification examination representative of skills learned in this course (e.g., MOS, MTA, IC3).	
150	Investigate job opportunities, using the Internet and other sources.	English: 10.2, 10.5, 10.8, 11.2, 11.5, 11.8, 12.2, 12.5, 12.8
151	Investigate careers in the information technology field.	English: 10.5, 11.5, 12.5 History and Social Science: GOVT.1
152	Research certification and educational opportunities.	English: 10.5, 10.8, 11.5, 11.8, 12.5, 12.8
153	Compose or update a printed résumé.	English: 10.6, 10.7, 11.6, 11.7, 12.6, 12.7
154	Prepare an electronic résumé.	English: 10.6, 10.7, 11.6, 11.7, 12.6, 12.7
155	Create a letter of application.	English: 10.6, 10.7, 11.6, 11.7, 12.6, 12.7
156	Complete an electronic application form.	English: 10.5, 10.6, 10.7, 11.5, 11.6, 11.7, 12.5, 12.6, 12.7
157	Create an interview follow-up letter.	English: 10.6, 10.7, 11.6, 11.7, 12.6, 12.7
158	Identify the steps to follow in resigning from a position.	English: 10.1, 10.5, 11.1, 11.5, 12.1, 12.5
159	Develop a portfolio containing representative samples of student's work (e.g., program design, technical documentation and design, source code, and output).	English: 10.1, 10.6, 10.7, 11.1, 11.6, 11.7, 12.1, 12.6, 12.7
160	Deliver an oral presentation of portfolio.	English: 10.1, 10.7, 11.1, 11.7, 12.1, 12.7
161	Identify potential employment barriers for nontraditional groups and ways to overcome barriers.	English: 10.5, 10.8, 11.5, 11.8, 12.5, 12.8 History and Social Science: GOVT.1, GOVT.9, VUS.13, VUS.14

Cyber Security and Cyber Forensics Infusion Units

Cyber Security and Cyber Forensics Infusion Units (CYBR) were designed to be infused with designated CTE courses to help students in those programs achieve additional, focused, validated tasks/competencies in personal and professional cyber security skills. These units are not

mandatory, and, as such, the tasks/competencies are marked as "optional," to be taught at the instructor's discretion.

Entrepreneurship Infusion Units

Entrepreneurship Infusion Units may be used to help students achieve additional, focused competencies and enhance the validated tasks/competencies related to identifying and starting a new business venture. Because the unit is a complement to certain designated courses and is not mandatory, all tasks/competencies are marked “optional.”

Microsoft Imagine Academy Resources

Microsoft Imagine Academy (MSIA) offers classroom resources and materials and instructional techniques that will help enhance instruction and learning for this course. Using the school's membership ID and product key for the Microsoft Imagine Academy, all resources are available through the [MSIA Member Dashboard on the Microsoft site](#).

- To access the curriculum resources, select the Classroom Tile from the member site.
- To access downloadable curriculum resources including the MOAC e-Book, Lesson Plans, and Study Guides select Curriculum Overview - Curriculum Downloads.
- To access Online Learning videos and tutorials select Online Learning Directory tile.
- For more information visit: [How to Get Started with Microsoft Imagine Academy Program](#).

Appendix: Credentials, Course Sequences, and Career Cluster Information

Industry Credentials: Only apply to 36-week courses

- Certified Internet Web (CIW) Advanced HTML 5 and CSS 3 Specialist Examination
- Certified Internet Web (CIW) Data Analyst Examination
- Certified Internet Web (CIW) Database Design Specialist Examination
- Certified Internet Web (CIW) E-Commerce Services Specialist Examination
- Certified Internet Web (CIW) Internet Business Associate Examination
- Certified Internet Web (CIW) JavaScript Specialist Examination
- Certified Internet Web (CIW) Network Technology Associate Examination
- Certified Internet Web (CIW) Site Development Associate Examination
- Certified Internet Web (CIW) Social Media Strategist Examination
- Certified Internet Web (CIW) User Interface Designer Examination
- Certified Internet Web (CIW) Web Design Specialist Examination
- Certified Internet Web (CIW) Web Security Specialist Examination
- Cloud Essentials Certification Examination
- College and Work Readiness Assessment (CWRA+)
- IC3 Digital Literacy Certification Examination
- Microsoft 365 Fundamentals Examination
- Microsoft Certified Azure Fundamentals Examination
- Microsoft Dynamics 365 Fundamentals Examination
- Microsoft Office Specialist (MOS) Examinations
- Microsoft Technology Associate (MTA) Examinations
- National Career Readiness Certificate Assessment
- Oracle Certified Associate Examinations
- Oracle Programming with PL/SQL Examination
- Workplace Readiness Skills for the Commonwealth Examination

Concentration sequences: *A combination of this course and those below, equivalent to two 36-week courses, is a concentration sequence. Students wishing to complete a specialization may take additional courses based on their career pathways. A program completer is a student who has met the requirements for a CTE concentration sequence and all other requirements for high school graduation or an approved alternative education program.*

- Database Design and Management (Oracle) (6660/36 weeks)
- International Baccalaureate Information Technology in a Global Society (IB6613/36 weeks)
- Java Programming (Oracle) (6661/36 weeks)

Career Cluster: Information Technology	
Pathway	Occupations
Information Support and Services	Computer Support Specialist Database Administrator Database Analyst Information Systems Analyst Instructional Coordinator Network Systems and Data Communication Analyst
Network Systems	Computer and Information Systems Administrator Computer Software Engineer Computer Support Specialist Telecommunications Specialist

Career Cluster: Science, Technology, Engineering and Mathematics	
Pathway	Occupations
Engineering and Technology	Computer Hardware Engineer Computer Programmer Computer Software Engineer Network and Computer Systems Administrator Network Systems and Data Communication Analyst Production, Planning, Expediting Clerk Project Manager Stockroom, Warehouse, or Storage Yard Stock Clerk Technical Writer Telecommunications Specialist Transportation Manager