

# Database Programming

(10053)

## Rationale Statement:

The study of formalized database design. This course will focus on relational model design and the use of SQL. Students will use a modern relational database to implement designs and learn the basics of data management.

Provides hands-on experience with procedural extensions to the SQL. Topics include data control languages, control structures, exception handling, stored procedures, triggers, cursors, and cursor processing.

## Course Description:

Grade Level: 11-12

## Course Topics:

Principles of database fundamentals

Structured query language (SQL)

Handle exceptions and create reusable commands

## Core Technical Standards & Examples

| Indicator #1: Understand the principles of database fundamentals as an integral part of an information system. |  |
|--|--|
| Bloom's Taxonomy Level   | Standard and Examples  |
| Remembering  | <b>DP1.1 Demonstrate knowledge of database modeling</b><br>Examples: <ul style="list-style-type: none"><li>• Demonstrate knowledge of entities, fields, primary keys, foreign keys, and relationships</li><li>• Use modeling techniques such as entity relationship diagrams (ERDs)</li><li>• Demonstrate knowledge of data base normalization to the 3<sup>rd</sup> normal form</li></ul>   |
| Applying   | <b>DP1.2 Demonstrate knowledge of database rules as they relate to business processes</b><br>Examples: <ul style="list-style-type: none"><li>• Use relationship cardinalities such as “one-to-one”, “one-to-many”, and “many-to-many”</li><li>• Use field properties such as data type, length, and masks to ensure data is usable in business processes</li><li>• Demonstrate knowledge to ensure top performance of the database by using indexes, searches, and sorts</li></ul> |

**Indicator #2: Understand the principles of structured query language (SQL) to add, edit, delete, and extract information from a database.**

| Bloom's Taxonomy Level | Standard and Examples  |
|------------------------|--|
| <p><b>Applying</b></p> | <p><b>DP2.1 Demonstrate understanding of basic procedural languages and SQL (PL/SQL) structures and how it relates to existing data</b><br/> <b>Examples:</b></p> <ul style="list-style-type: none"> <li>• Demonstrate mastery of the PL/SQL environment to use existing toolsets and user interfaces</li> <li>• Demonstrate knowledge of the types of PL/SQL blocks</li> <li>• Generate output from PL/SQL blocks that are already written and validated</li> <li>• Identify the Different Types of Identifiers in a PL/SQL subprogram</li> <li>• Use the declarative section to define identifiers in a PL/SQL subprogram</li> <li>• Declare and use variables in a PL/SQL subprogram</li> <li>• Store Data in variables in a PL/SQL subprogram</li> </ul> |
| <p><b>Applying</b></p> | <p><b>DP2.2 Use SQL as a procedural language to interact with databases to answer business questions.</b><br/> <b>Examples:</b></p> <ul style="list-style-type: none"> <li>• Demonstrate knowledge of PL/SQL syntax guidelines</li> <li>• Write executable statements in PL/SQL</li> <li>• Use literals in PL/SQL</li> <li>• Customize identifier assignments with SQL functions</li> <li>• Use nested blocks as statements</li> <li>• Use operators in PL/SQL</li> </ul>  |
| <p><b>Applying</b></p> | <p><b>DP2.3 Use control structures, composite data types, and explicit cursors in PL/SQL to improve productivity on a database management system.</b><br/> <b>Examples:</b></p> <ul style="list-style-type: none"> <li>• Use IF and CASE statements for conditional processing</li> <li>• Use loops for iterative control of queries and statements</li> <li>• Use INDEX BY tables to hold multiple values of same data types</li> <li>• Increase the Flexibility of Cursors By Using Parameters</li> </ul>  |

**Indicator #3: Understand how to handle exceptions and create reusable commands as part of a database management system.**

| <b>Bloom's Taxonomy Level</b> | <b>Standard and Examples</b>   |
|-------------------------------|--|
| <b>Applying</b>               | <b>DP3.1 Demonstrate knowledge of handling exceptions to automatically validate values of stored data</b><br>Examples: <ul style="list-style-type: none"><li>• Use predefined exceptions to catch data entry errors in the database records</li><li>• Create exception traps for non-predefined errors</li><li>• Use the RAISE_APPLICATION_ERROR procedure to report errors to applications</li></ul>                            |
| <b>Applying</b>               | <b>DP3.2 Demonstrate knowledge of procedures and functions to mandate processing as part of the database management system</b><br>Examples: <ul style="list-style-type: none"><li>• Demonstrate knowledge of the differences between anonymous blocks and subprograms</li><li>• Create and invoke procedures</li><li>• Creating and invoking functions</li><li>• Pass parameter to a function so value can be returned</li></ul> |