# Database Design and Management (Oracle)

**6660 36 weeks**

## Table of Contents

- Acknowledgments ................................................................................................................................. 2
- Course Description ................................................................................................................................. 3
- Task Essentials Table ............................................................................................................................. 3
- Curriculum Framework ........................................................................................................................... 10
- Exploring Database Technologies .......................................................................................................... 10
- Identifying Business Requirements ...................................................................................................... 15
- Examining Entity-Relationship Basics ................................................................................................. 18
- Applying Design Concepts to Database Models .................................................................................... 21
- Transitioning from Design Concepts to Database Management .......................................................... 27
- Writing Structured Query Language (SQL) Statements ......................................................................... 30
- Restricting and Sorting Data Using SQL ............................................................................................... 33
- Performing Single-Row Functions ........................................................................................................ 39
- Using JOIN Tables ................................................................................................................................. 44
- Aggregating Data Using GROUP Functions ........................................................................................ 45
- Applying Advanced Data Selection Techniques .................................................................................... 48
- Applying Data Manipulation Language ............................................................................................... 52
- Modifying and Managing Tables ......................................................................................................... 57
- Defining Database Constraints ............................................................................................................. 62
- Creating and Managing Views .............................................................................................................. 64
- Creating Additional Database Objects ................................................................................................. 67
- Maintaining Database Security and System Security ......................................................................... 72
- Making Database Transactions ............................................................................................................ 75
- Preparing for Industry Certification ..................................................................................................... 77
- SOL Correlation by Task ....................................................................................................................... 80
- Teacher Resources ............................................................................................................................... 84
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

This course includes database design and Structured Query Language (SQL) programming. Students study database fundamentals, including database development, modeling, design, and normalization. In addition, students are introduced to database programming with SQL. Students gain the skills and knowledge needed to use features of database software and programming to manage and control access to data. Students will prepare for the first of two certification exams.

*Recommended prerequisite: Information Technology Fundamentals 6670*

### Task Essentials Table

- Tasks/competencies designated by plus icons (⊕) in the left-hand column(s) are essential
- Tasks/competencies designated by empty-circle icons (〇) are optional
- Tasks/competencies designated by minus icons (☒) are omitted
- Tasks marked with an asterisk (*) are sensitive.

<table>
<thead>
<tr>
<th>Task Number</th>
<th>6660</th>
<th>Tasks/Competencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploring Database Technologies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>⊕</td>
<td>Research the history of databases.</td>
</tr>
<tr>
<td>40</td>
<td>⊕</td>
<td>List the major types of databases.</td>
</tr>
<tr>
<td>41</td>
<td>⊕</td>
<td>Distinguish between a conceptual and a physical database model.</td>
</tr>
<tr>
<td>42</td>
<td>⊕</td>
<td>Compare the structure of relational and non-relational database structures.</td>
</tr>
<tr>
<td>43</td>
<td>⊕</td>
<td>Identify the characteristics of a relational database.</td>
</tr>
<tr>
<td>44</td>
<td>⊕</td>
<td>Examine the database development life cycle.</td>
</tr>
<tr>
<td>45</td>
<td>⊕</td>
<td>Research the future direction of database technologies.</td>
</tr>
</tbody>
</table>

Identifying Business Requirements
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>46</td>
<td></td>
<td>Describe the process of modeling business requirements.</td>
</tr>
<tr>
<td>47</td>
<td></td>
<td>Apply business concepts to the database model.</td>
</tr>
<tr>
<td>48</td>
<td></td>
<td>Define entities among elements of significance.</td>
</tr>
<tr>
<td>49</td>
<td></td>
<td>Define attributes of each entity.</td>
</tr>
<tr>
<td>50</td>
<td></td>
<td>Select unique identifiers (UIDs).</td>
</tr>
<tr>
<td>51</td>
<td></td>
<td>Define types of unique identifiers.</td>
</tr>
<tr>
<td>52</td>
<td></td>
<td>Define business rules.</td>
</tr>
<tr>
<td></td>
<td><strong>Examining Entity-Relationship Basics</strong></td>
<td></td>
</tr>
<tr>
<td>53</td>
<td></td>
<td>Analyze entities for relationships that exist among them.</td>
</tr>
<tr>
<td>54</td>
<td></td>
<td>Distinguish among relationship types.</td>
</tr>
<tr>
<td>55</td>
<td></td>
<td>Describe relationship transferability.</td>
</tr>
<tr>
<td>56</td>
<td></td>
<td>Name relationships.</td>
</tr>
<tr>
<td>57</td>
<td></td>
<td>Explain relationship optionality.</td>
</tr>
<tr>
<td>58</td>
<td></td>
<td>Explain relationship degree/cardinality.</td>
</tr>
<tr>
<td></td>
<td><strong>Applying Design Concepts to Database Models</strong></td>
<td></td>
</tr>
<tr>
<td>59</td>
<td></td>
<td>Identify elements of the graphic representation of a database model.</td>
</tr>
<tr>
<td>60</td>
<td></td>
<td>Define drawing conventions for readability.</td>
</tr>
<tr>
<td>61</td>
<td></td>
<td>Illustrate business rules in an entity-relationship model.</td>
</tr>
<tr>
<td>62</td>
<td></td>
<td>Define the normalization process.</td>
</tr>
<tr>
<td>63</td>
<td></td>
<td>Perform the normalization process.</td>
</tr>
<tr>
<td>64</td>
<td></td>
<td>Resolve many-to-many relationships.</td>
</tr>
<tr>
<td>65</td>
<td></td>
<td>Model hierarchical data.</td>
</tr>
<tr>
<td>66</td>
<td></td>
<td>Model recursive relationships.</td>
</tr>
<tr>
<td>67</td>
<td></td>
<td>Model exclusive relationships.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>68</td>
<td>Define relational-database terminology.</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>Define a fact and dimension.</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>Verbalize a diagram's relationship notation.</td>
<td></td>
</tr>
<tr>
<td>Transitioning from Design Concepts to Database Management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>Summarize the database-design process.</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>Convert a conceptual design to a physical database model.</td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>Map simple entities, attributes, and primary keys.</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>Identify data constraints.</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>Map relationships to foreign keys.</td>
<td></td>
</tr>
<tr>
<td>Writing Structured Query Language (SQL) Statements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>Describe SQL.</td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>Distinguish among categories of SQL statements.</td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>Demonstrate the syntax for select statements (projection).</td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>Demonstrate methods for selecting columns and arithmetic expressions (selection).</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>Incorporate column alias and literals in a SELECT statement.</td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>Describe operator precedence.</td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>Describe methods for displaying a table.</td>
<td></td>
</tr>
<tr>
<td>Restricting and Sorting Data Using SQL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>Restrict data, using the WHERE clause.</td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>Define comparison operators (e.g., =, &gt;, &lt;, &gt;=, &lt;=, &lt;&gt;, !=)</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>Restrict data, using the BETWEEN ... AND and IN clauses.</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>Restrict data, using wildcards and patterns within the LIKE condition.</td>
<td></td>
</tr>
<tr>
<td>87</td>
<td>Demonstrate the use of the ESCAPE character with wildcard characters.</td>
<td></td>
</tr>
<tr>
<td>Page</td>
<td>Topic</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>Restrict (or specify) data containing nulls, using the IS (NOT) NULL clause.</td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>Demonstrate the use of two or more conditional statements using logical operators (i.e., AND, OR, NOT).</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>Sort data by using ResultSet with the ORDER BY clause.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Performing Single-Row Functions</strong></td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>Explain the concept of functions.</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>Distinguish between the two categories of functions in SQL.</td>
<td></td>
</tr>
<tr>
<td>93</td>
<td>Demonstrate restricting data using ANY and ALL.</td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>Define single-row functions.</td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>Describe the types/categories of single-row functions.</td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>Demonstrate the use of character, number, and date functions in SELECT statements.</td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>Describe the use of conversion functions.</td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>Demonstrate the use of null character handling.</td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>Demonstrate the use of conditional expressions.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Using JOIN Tables</strong></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>Describe the concept of joining data from two or more tables.</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>Demonstrate the use of ANSI joins.</td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>Demonstrate the use of Oracle proprietary joins.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Aggregating Data Using GROUP Functions</strong></td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>Define group/aggregate functions.</td>
<td></td>
</tr>
<tr>
<td>104</td>
<td>Describe how to write a query containing group functions.</td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>Construct SQL code by applying a GROUP BY clause that uses column, alias, and number.</td>
<td></td>
</tr>
<tr>
<td>106</td>
<td>Construct code by applying a HAVING clause.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>107</td>
<td>♡</td>
<td>Demonstrate additional functionality of the GROUP BY clause.</td>
</tr>
<tr>
<td><strong>Applying Advanced Data Selection Techniques</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>♡</td>
<td>Describe the types of problems that subqueries can solve.</td>
</tr>
<tr>
<td>109</td>
<td>♡</td>
<td>Define subqueries.</td>
</tr>
<tr>
<td>110</td>
<td>♡</td>
<td>Construct a single-row subquery.</td>
</tr>
<tr>
<td>111</td>
<td>♡</td>
<td>Construct a multi-row subquery.</td>
</tr>
<tr>
<td>112</td>
<td>♡</td>
<td>Construct a correlated subquery.</td>
</tr>
<tr>
<td>113</td>
<td>♡</td>
<td>Write a multiple-column subquery.</td>
</tr>
<tr>
<td>114</td>
<td>♡</td>
<td>Explain the behavior of subqueries when null values are retrieved.</td>
</tr>
<tr>
<td><strong>Applying Data Manipulation Language</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>♡</td>
<td>Describe data manipulation language (DML).</td>
</tr>
<tr>
<td>116</td>
<td>♡</td>
<td>Describe integrity constraints.</td>
</tr>
<tr>
<td>117</td>
<td>♡</td>
<td>Insert rows into a table.</td>
</tr>
<tr>
<td>118</td>
<td>♡</td>
<td>Update data within a table.</td>
</tr>
<tr>
<td>119</td>
<td>♡</td>
<td>Delete rows in a table.</td>
</tr>
<tr>
<td>120</td>
<td>♡</td>
<td>Construct a MERGE statement.</td>
</tr>
<tr>
<td>121</td>
<td>♡</td>
<td>Construct a multi-table insert statement.</td>
</tr>
<tr>
<td><strong>Modifying and Managing Tables</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>122</td>
<td>♡</td>
<td>Describe data definition language (DDL).</td>
</tr>
<tr>
<td>123</td>
<td>♡</td>
<td>Create a table.</td>
</tr>
<tr>
<td>124</td>
<td>♡</td>
<td>Create a table, using a subquery.</td>
</tr>
<tr>
<td>125</td>
<td>♡</td>
<td>Describe the data dictionary.</td>
</tr>
<tr>
<td>126</td>
<td>♡</td>
<td>Define common data types.</td>
</tr>
<tr>
<td>127</td>
<td>♡</td>
<td>Describe date/time data types.</td>
</tr>
<tr>
<td>Page</td>
<td>Exercise</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>Write code to alter table definitions.</td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>Write SQL code to manipulate column definitions, using DROP, RENAME, and TRUNCATE commands.</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>Differentiate among TRUNCATE, DROP, and DELETE.</td>
<td></td>
</tr>
<tr>
<td>131</td>
<td>Describe FLASHBACK.</td>
<td></td>
</tr>
</tbody>
</table>

**Defining Database Constraints**

<table>
<thead>
<tr>
<th>Page</th>
<th>Exercise</th>
</tr>
</thead>
<tbody>
<tr>
<td>132</td>
<td>Describe the necessity for database constraints.</td>
</tr>
<tr>
<td>133</td>
<td>Distinguish between column-level and table-level constraints.</td>
</tr>
<tr>
<td>134</td>
<td>List the types of constraints and their applications.</td>
</tr>
<tr>
<td>135</td>
<td>Write a table- and column-level constraint.</td>
</tr>
<tr>
<td>136</td>
<td>View constraints with table definition and the data dictionary.</td>
</tr>
</tbody>
</table>

**Creating and Managing Views**

<table>
<thead>
<tr>
<th>Page</th>
<th>Exercise</th>
</tr>
</thead>
<tbody>
<tr>
<td>137</td>
<td>Describe a view.</td>
</tr>
<tr>
<td>138</td>
<td>Create a view.</td>
</tr>
<tr>
<td>139</td>
<td>Write SQL code to retrieve data through a view.</td>
</tr>
<tr>
<td>140</td>
<td>Write SQL code to alter the definition of a view.</td>
</tr>
<tr>
<td>141</td>
<td>Manipulate tables, using the INSERT, UPDATE, and DELETE commands through a view.</td>
</tr>
<tr>
<td>142</td>
<td>Write SQL code to remove a view.</td>
</tr>
<tr>
<td>143</td>
<td>Create an inline view.</td>
</tr>
<tr>
<td>144</td>
<td>Describe the procedures for performing a top-n analysis.</td>
</tr>
</tbody>
</table>

**Creating Additional Database Objects**

<table>
<thead>
<tr>
<th>Page</th>
<th>Exercise</th>
</tr>
</thead>
<tbody>
<tr>
<td>145</td>
<td>Describe database objects and their uses.</td>
</tr>
<tr>
<td>146</td>
<td>Define a sequence object.</td>
</tr>
<tr>
<td>147</td>
<td>Create a sequence.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>148</td>
<td>Use a sequence.</td>
</tr>
<tr>
<td>149</td>
<td>Modify a sequence.</td>
</tr>
<tr>
<td>150</td>
<td>Describe the index.</td>
</tr>
<tr>
<td>151</td>
<td>Describe issues that affect the decision to create an index.</td>
</tr>
<tr>
<td>152</td>
<td>Create an index.</td>
</tr>
<tr>
<td>153</td>
<td>Create private and public synonyms.</td>
</tr>
<tr>
<td>154</td>
<td>Describe the type of information available in the data dictionary.</td>
</tr>
<tr>
<td>155</td>
<td>Create SQL code to retrieve information from the data dictionary.</td>
</tr>
</tbody>
</table>

**Maintaining Database Security and System Security**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>156</td>
<td>Define object privileges.</td>
<td></td>
</tr>
<tr>
<td>157</td>
<td>Construct an object privilege.</td>
<td></td>
</tr>
<tr>
<td>158</td>
<td>Describe roles.</td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>Define system privileges.</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>Describe system privileges that can be granted to a user.</td>
<td></td>
</tr>
<tr>
<td>161</td>
<td>Define a database link.</td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>Create a new database user.</td>
<td></td>
</tr>
<tr>
<td>163</td>
<td>Drop a database user.</td>
<td></td>
</tr>
</tbody>
</table>

**Making Database Transactions**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>164</td>
<td>Define Transaction Control Language (TCL).</td>
<td></td>
</tr>
<tr>
<td>165</td>
<td>Describe the importance of transaction control to businesses.</td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>Demonstrate the use of SAVEPOINT, ROLLBACK, and COMMIT.</td>
<td></td>
</tr>
</tbody>
</table>

**Preparing for Industry Certification**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>167</td>
<td>Describe the process and requirements for obtaining industry certifications related to the Database Design and Management (Oracle) course.</td>
<td></td>
</tr>
</tbody>
</table>
Identify testing skills and strategies for a certification examination.

Demonstrate ability to successfully complete selected practice examinations.

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

Legend: ☑ Essential ☐ Non-essential ☐ Omitted

Curriculum Framework

Exploring Database Technologies

Task Number 39

Research the history of databases.

Definition

Research should compile facts about the origin of modern databases, such as

- the first mention of a relational database in an article published by Edgar F. “Ted” Codd in 1970, which led to IBM’s System R
- information about System R’s evolution into DB2
- Honeywell’s first commercially available database in 1976
- developments through the 1980s and 1990s
- Oracle database products and other major companies, such as Sybase and Microsoft
- the influence of the Internet and commercial online businesses.

NBEA Achievement Standards for Information Technology

Analyze how developments in information technology affect the supply/demand characteristics of the job market.

Evaluate how information technology transforms business processes and relationships.

Examine how information technology changes the breadth and level of worker responsibilities.
Explain the nature and interrelationships of bytes, fields, records, and databases.

Identify and evaluate how information technology developments changes the way humans do their work.

Illustrate how information technology changes organization structures.

---

**Task Number 40**

**List the major types of databases.**

**Definition**

Listing should include the examination of several vendor products from companies such as Microsoft and IBM.

**Microsoft Imagine Academy Resources**

[2.022] Microsoft Digital Literacy: Productivity Programs—Lesson 6
Introduction to Database Programs

Access Lesson Plan: Understanding Microsoft Access 2010

[5.033] Access 2010: Project 1
Graduation Information (project)

**NBEA Achievement Standards for Information Technology**

Identify the appropriate type of database for a particular situation (e.g., flat, relational).

---

**Task Number 41**

**Distinguish between a conceptual and a physical database model.**

**Definition**
Distinction should be based on understanding that conversion from the conceptual to the physical requires taking a blueprint (conceptual) and translating ideas, objectives, and terminology into a practical, usable database structure (physical).

A summary of this process that further analyzes the transition between conceptual and physical should include the following elements:

- **Conceptual**: Strategize and analyze—conceive a plan by talking with database users and stakeholders or management.
- **Design**: create a data model using paper or a modeling program, such as Oracle SQL Developer.
- **Physical**: Build and document—build the database through writing the program (the physical) and demonstrating with modeling software. Make any necessary adjustments to the design for technology considerations, physical surroundings, or other processes that influence the implementation of the database.
- **Produce**: once the program runs smoothly, go into full production and complete the conversion of the old to the new.

**NBEA Achievement Standards for Information Technology**

*Develop design specifications for record types, output, and data stores.*

*Identify and select logical and physical structures appropriate for specific applications.*

---

**Task Number 42**

**Compare the structure of relational and non-relational database structures.**

**Definition**

Comparison should include a description of the two main database types:

- **Relational**, in which data is arranged according to a schema that allows data to be displayed as tables with rows and columns. Data integrity is of particular concern in relational databases, and relational database management systems (RDBMS) use a number of constraints to ensure that the data contained in the tables is reliable and accurate.
- **Non-relational**, or NoSQL (not only SQL), which can be schema agnostic, allowing unstructured and semi-structured data to be stored and manipulated. NoSQL databases fall into one or more of the following categories:
  - Key-Value stores
  - Wide-Column stores
Task Number 43

Identify the characteristics of a relational database.

Definition

Identification of database characteristics should include relational database characteristics, such as:

- the relationship between tables along primary/foreign (unique) key lines
- the potential of using the SQL programming language to manipulate data
- the inclusion of objects (e.g., views, sequences, synonyms, constraints) vs. a flat-file database (where all data is in one table with no relationship to other data in other tables).

Microsoft Imagine Academy Resources

[2.022] Microsoft Digital Literacy: Productivity Programs—Lesson 6
[5.033] Access 2010: Project 1

NBEA Achievement Standards for Information Technology

Identify the appropriate type of database for a particular situation (e.g., flat, relational).

Task Number 44

Examine the database development life cycle.

Definition

Examination should include the steps used when developing a database:

1. Read documentation concerning business rules.
2. Interview key figures in the client company.
3. Develop the conceptual model independent of technological or financial concerns.
4. Receive customer or client approval.
5. Convert the entity-relationship diagram (ERD) to a data-design model. (This brings the conceptual design down to a more physical level in preparation for the actual writing of the code.)
6. Write the code.
7. Test the code.
8. Implement the database.

NBEA Achievement Standards for Information Technology

Identify and explain the steps in the systems development life cycle.

Task Number 45

Research the future direction of database technologies.

Definition

Research should result in

- in-memory database
- data protection
- JSON data representation
- database virtualization or containerization
- linking big data and transactional data
- analytics on transactional data-based databases and coexistence of cloud and on-premises databases.

NBEA Achievement Standards for Information Technology

Analyze how developments in information technology affect the supply/demand characteristics of the job market.

Evaluate how information technology transforms business processes and relationships.

Examine how information technology changes the breadth and level of worker responsibilities.

Identify and evaluate how information technology developments changes the way humans do their work.

Illustrate how information technology changes organization structures.
Identifying Business Requirements

Task Number 46
Describe the process of modeling business requirements.

Definition
Description should include personal interviews with stakeholders, job site investigation, and survey of documents and management requirements and projections.

NBEA Achievement Standards for Information Technology
Define system requirements using structured systems analysis tools.

Task Number 47
Apply business concepts to the database model.

Definition
Application may be demonstrated by converting a written or verbal case study into an entity-relationship diagram (ERD).

Microsoft Imagine Academy Resources

Access Lesson Plan: Understanding Microsoft Access 2010

[5.033] Access 2010: Project 1
Graduation Information (project)

NBEA Achievement Standards for Information Technology
Identify and apply appropriate application development tools.
Incorporate appropriate human interface design principles.

---

**Task Number 48**

**Define entities among elements of significance.**

**Definition**

Defining entities is an iterative process that requires

- selecting elements of significance to the business tracked by data
- naming the data tracked, generally using a noun (i.e., person, place, thing, event).

An entity will have occurrences, known as instances.

**Microsoft Imagine Academy Resources**

[Access Lesson Plan: Create Database Tables Using Access]
[Summer Jobs (project)]

**NBEA Achievement Standards for Information Technology**

Plan and develop record specifications.

---

**Task Number 49**

**Define attributes of each entity.**

**Definition**

Defining attributes requires selecting adjectives that describe the entity. An attribute has a single value and is a specific piece of information that helps

- describe an entity
- quantify an entity
- qualify an entity
- classify an entity
Task Number 50

Select unique identifiers (UIDs).

Definition

Selection should be based on

- identifying attributes, relationships, or both
- determining whether a natural UID exists
- using a sequence to artificially generate a number if a natural UID does not exist
- validating the attribute by asking

1.  
   o Is the attribute unique?
   o Is the attribute mandatory?

A UID may be one attribute or more than one attribute combined.

Microsoft Imagine Academy Resources

[2.022] Microsoft Digital Literacy: Productivity Programs—Lesson 6
Introduction to Database Programs
Access Lesson Plan: Create Database Tables Using Access
Summer Jobs (project)
Plan and develop record specifications.

**Task Number 51**

**Define types of unique identifiers.**

**Definition**

Definition of types of unique identifiers should include a discussion of artificial, simple, composite, primary, and secondary unique identifiers (UIDs).

**Task Number 52**

**Define business rules.**

**Definition**

Definition of business rules should include the analysis of an interview with a stakeholder to determine the rules that govern the way a company operates. Entities, attributes, and relationships should be obtained from these rules and graphically represented.

**Examining Entity-Relationship Basics**

**Task Number 53**

**Analyze entities for relationships that exist among them.**
Definition

Analysis should be conducted by compiling a matrix diagram. Compilation should be made in a graph that contains all entities listed along the x- and y-axis; plotted coordinates represent all possible relationships (the relationship matrix).

Steps are as follows:

1. Determine the existence of a relationship between the entity in the x-axis and the entity in the y-axis. (Sentences are always read x to y.)
2. Name the relationship, using an appropriate verb.

Task Number 54

Distinguish among relationship types.

Definition

Distinction should include

- a one-to-one relationship
- a one-to-many relationship
- a many-to-many relationship.

Microsoft Imagine Academy Resources

[2.022] Microsoft Digital Literacy: Productivity Programs—Lesson 6
Introduction to Database Programs
Access Lesson Plan: Modify Database Tables
Summer Job Openings (project)

Task Number 55

Describe relationship transferability.

Definition

Description should include
• difference between transferable and nontransferable relationships
• illustration of nontransferable relationships on ERDs
• examples of relationship transferability.

Task Number 56

Name relationships.

Definition

Naming relationships should include selecting an appropriate verb that typically ends in a preposition and conveys how the two entities are related, and placing the verb on the diagram.

Microsoft Imagine Academy Resources

[2.022] Microsoft Digital Literacy: Productivity Programs—Lesson 6
Introduction to Database Programs
Access Lesson Plan: Modify Database Tables
Summer Job Openings (project)

Task Number 57

Explain relationship optionality.

Definition

Explanation should be based on identification of optionality terminology (e.g., must or may, may or could).

Task Number 58

Explain relationship degree/cardinality.

Definition
Explanation of degree/cardinality can be shown by substituting the words *one or many* or *one and only one* in the relationship name.

### Applying Design Concepts to Database Models

#### Task Number 59

**Identify elements of the graphic representation of a database model.**

**Definition**

Identification should include a database model, or entity-relationship diagram, that contains entities, attributes, and identifiers.

**NBEA Achievement Standards for Information Technology**

Define system requirements using structured systems analysis tools.

#### Task Number 60

**Define drawing conventions for readability.**

**Definition**

Definition should include an explanation of ERD conventions, such as:

- “Crows fly south and east.”
- High-volume entities belong on the upper left.
- Use white space to clarify.
- Break up large diagrams into smaller modules
Task Number 61

Illustrate business rules in an entity-relationship model.

Definition

Illustration requires interpreting written business rules and accurately portraying them on an entity-relationship diagram and includes

- creating relationship lines between entities that have a documented business relationship
- including the relationship verb, ordinality, and cardinality.

Microsoft Imagine Academy Resources

Access Lesson Plan: Modify Database Tables
Summer Job Openings (project)

Task Number 62

Define the normalization process.

Definition

Definition should include the necessary steps to follow when revising a database design to ensure data integrity.

Microsoft Imagine Academy Resources

[2.022] Microsoft Digital Literacy: Productivity Programs—Lesson 6
Introduction to Database Programs

NBEA Achievement Standards for Information Technology

Normalize a database schema.

Task Number 63
Perform the normalization process.

Definition

Performance should include

- locating attributes within a graphical representation to ensure that there is no redundancy or misplaced attributes in the diagram. (There are many normal forms, but the first three, used in a systematic fashion, perform checks on the data.)
- comparing normal forms by using the following questions associated with each:
  - First normal form: Will there be one and only one value for this attribute for each instance?
  - Second normal form: Does the attribute depend upon the entire unique identifier for its identification?
  - Third normal form: Are there any attributes that have attributes of their own? Are there any repeating groups?

Microsoft Imagine Academy Resources

[2.022] Microsoft Digital Literacy: Productivity Programs—Lesson 6
Introduction to Database Programs

NBEA Achievement Standards for Information Technology

Normalize a database schema.

Task Number 64

Resolve many-to-many relationships.

Definition

A many-to-many relationship can be modeled; however, it cannot be implemented in a relational database because it violates data integrity. A many-to-many relationship is reduced to two one-to-many relationships with an intersecting entity. Create a new entity that is positioned between the existing two entities.

Resolution is accomplished by following these steps:

1. Determine the name of an intersecting entity (name using a combination of the two many-to-many names if necessary).
2. Define the first one-to-many relationship and put the many on the intersecting entity.
3. Add the other relationship (which should be the original relationship between the two entities without the cardinality). Define the second one-to-many relationship, again with the many on the intersecting entity.

Microsoft Imagine Academy Resources

[2.022] Microsoft Digital Literacy: Productivity Programs—Lesson 6
Introduction to Database Programs
Access Lesson Plan: Modify Database Tables
Summer Job Openings (project)

Task Number 65

Model hierarchical data.

Definition

Modeling should be performed by using a tree model that illustrates

- explicitly—an entity for each level of the hierarchy (very detailed)
- generically—an entity with recursive relationships providing a low degree of specificity (offers flexibility).

NBEA Achievement Standards for Information Technology

Identify and select logical and physical structures appropriate for specific applications.

Modify record structures.

Plan and develop a database schema.

Plan and develop record specifications.

Use application development tools associated with a database system to create solutions for organization problems.

Use database application development tools to create information systems to solve organization problems.
Task Number 66

Model recursive relationships.

Definition

Modeling should demonstrate the application of a recursive relationship, when one entity has an attribute that relates to a second attribute in the same entity.

NBEA Achievement Standards for Information Technology

Identify and select logical and physical structures appropriate for specific applications.

Modify record structures.

Plan and develop a database schema.

Plan and develop record specifications.

Use application development tools associated with a database system to create solutions for organization problems.

Use database application development tools to create information systems to solve organization problems.

Task Number 67

Model exclusive relationships.

Definition

Modeling exclusive relationships can be accomplished through the arc-drawing convention that enforces mutual exclusivity and exhaustiveness.

NBEA Achievement Standards for Information Technology

Identify and select logical and physical structures appropriate for specific applications.

Modify record structures.

Plan and develop a database schema.
Plan and develop record specifications.

Use application development tools associated with a database system to create solutions for organization problems.

Use database application development tools to create information systems to solve organization problems.

Task Number 69
Define relational-database terminology.

Definition

Definition should include the following relational-database terms:

- Table
- Row
- Column
- Primary key
- Unique key
- Foreign key
- Data integrity

Definition may also include the following data-modeling terms:

- Entity
- Attribute
- Unique identifier
- Relationships
- Optionality
- Cardinality

Microsoft Imagine Academy Resources

[2.022] Microsoft Digital Literacy: Productivity Programs—Lesson 6
Introduction to Database Programs
Access Lesson Plan: Understanding Microsoft Access 2010
[5.033] Access 2010: Project 1
Graduation Information (project)
Task Number 70

**Define a fact and dimension.**

**Definition**

Definition should include

- how a fact, also known as a measure, can support numeric operations such as simple arithmetic, statistical calculations, and simultaneous equations
- how dimensions provide filtering, grouping, and labeling for individual, non-overlapping data elements
- how they relate to multidimensional terminology terms (e.g., members, hierarchies, sparsity, data cube, star, constellation schemas).

Task Number 71

**Verbalize a diagram's relationship notation.**

**Definition**

Verbalization is accomplished by reading aloud using the following sentence: “Each and every [entity name] [may/must] [relationship verbs] [one or many/one and only one] [entity name].”

Transitioning from Design Concepts to Database Management

Task Number 72

**Summarize the database-design process.**

**Definition**
Summary should include a translation of rules and objectives into an entity-relationship diagram, which must then be translated into a data-design plan for implementing the design, with ample consideration given to real-world technology and financial/budgetary concerns.

Task Number 73

Convert a conceptual design to a physical database model.

Definition

Conversion from a conceptual to a physical database model requires translating ideas, objectives, and terminology from a blueprint (conceptual) into practical, usable database structure (physical).

NBEA Achievement Standards for Information Technology

Develop design specifications for record types, output, and data stores.

Identify and select logical and physical structures appropriate for specific applications.

Task Number 74

Map simple entities, attributes, and primary keys.

Definition

Mapping is completed by drawing a matrix with all entities along the top and sides. The intersection of each block in the table is checked to discover whether any relationship exists between those entities.

NBEA Achievement Standards for Information Technology

Identify and select logical and physical structures appropriate for specific applications.

Task Number 75

Identify data constraints.

Definition
Identification of constraints should include

- UIDs
- primary key/foreign key relationships
- domains/data specifics
- check constraints
- data types required for data that will occupy a particular column.

Microsoft Imagine Academy Resources

Creating Complex Queries to Extract and Process Specific Information
Access Lesson Plan: Modify Database Tables
Summer Job Openings (project)

NBEA Achievement Standards for Information Technology

Identify and select logical and physical structures appropriate for specific applications.

Task Number 76

Map relationships to foreign keys.

Definition

Mapping relationships to foreign keys should include the rule that states that the many relationship pulls the foreign key into its entity. If there is not many relationships, then a determination as to the location of the foreign key must be made by revisiting the business rules.

Microsoft Imagine Academy Resources

Creating Complex Queries to Extract and Process Specific Information
Access Lesson Plan: Modify Database Tables
Summer Job Openings (project)

NBEA Achievement Standards for Information Technology

Identify and select logical and physical structures appropriate for specific applications.
Writing Structured Query Language (SQL) Statements

Task Number 77

Describe SQL.

Definition

Description should include

- SQL is a programming language for getting information to and from a database.
- SQL is both American National Standards (ANSI) and International Organization for Standardization (ISO) compliant.
- database companies, such as Oracle, often add proprietary commands to SQL to extend its functions.

Microsoft Imagine Academy Resources

Creating Complex Queries to Extract and Process Specific Information

Task Number 78

Distinguish among categories of SQL statements.

Definition

Distinction should include the following:

- Data definition language (DDL) defines the structure and relationship of the tables.
- Data manipulation language (DML) defines the manipulation of the data in the database.
- Data control language (DCL) controls access to the data.
Task Number 79

Demonstrate the syntax for select statements (projection).

Definition

Demonstration should include the proficient use of a working syntax, based on the instructor’s objectives and guidelines, and might include some of the following:

- Minimum required clauses of SELECT and FROM
- SELECT statement to project all columns using “*”
- SELECT statement to project specific columns
- SELECT statement including the WHERE clause and one comparison operator to create a selection
- SELECT statement using the keyword DISTINCT to retrieve a list of unique values

NBEA Achievement Standards for Information Technology

Enter data and edit fields and records.

Task Number 80

Demonstrate methods for selecting columns and arithmetic expressions (selection).

Definition

Demonstration should include SELECT statement including the WHERE clause and one comparison operator to create a selection.

Microsoft Imagine Academy Resources

Access Lesson Plan: Query a Database
Task Number 81

Incorporate column alias and literals in a SELECT statement.

Definition

Incorporation should include distinguishing between single- and double-quoted text and how each will be treated by the database, and should include

- the three ways to define column aliases and how the column_name will be presented in the ResultSet:
  - Plain text directly after the column name:
    - SELECT column_name alias, column_name2, …
  - Plain text with double quotes directly after the column name:
    - SELECT column_name “Alias”, column_name2, …
  - Using the “as” operator with double-quoted text directly after column_name:
    - SELECT column_name as “Alias”, column_name2, …
- Outputting string literals using the single quote in SELECT clause and describing how it will presented in the ResultSet:
  - SELECT column_name, ‘String Literal’, column_name2, …

Task Number 82

Describe operator precedence.

Definition

Description should include comparing operator precedence to standard mathematical rules of precedence (because they are identical) and defining operator precedence as that which can be executed or established by the use of parentheses in strategic locations. PEMDAS is an acronym representing operator precedence: parentheses, exponents, multiplication, division, addition, and subtraction.

Task Number 83
Describe methods for displaying a table.

Definition

Description should include any of the following methods:

- Displaying the structure of the table (not including ANY data) requires the use of the DESCRIBE command. The DESCRIBE command belongs to a set of commands executed not by the Oracle Database, but by the SQL*PLUS environment.
- Use SQL Developer’s Object View to see table structure by connecting to a database and then selecting a table from the list.
- Use Toad for Oracle to see an object’s structure by hovering your cursor over the object and pressing F4.
- Use Application Express to display the definition of a table by selecting
  - object browser
  - browse table
  - table tab.

Microsoft Imagine Academy Resources

  Filtering Data

  Access Lesson Plan: Modify Database Tables

  Summer Job Openings (project)

Restricting and Sorting Data Using SQL

Task Number 84

Restrict data, using the WHERE clause.

Definition
Demonstration should include using the WHERE clause to restrict the selection of all data in a table as opposed to only the data useful for answering a question. The WHERE clause should be used to narrow, or filter, results for precision and accuracy.

Microsoft Imagine Academy Resources

Filtering Data
Access Lesson Plan: Modify Database Tables
Summer Job Openings (project)

NBEA Achievement Standards for Information Technology

Sort, prioritize, and retrieve data from databases.

Task Number 85

Define comparison operators (e.g., =, >, <, >=, <=, <> , !=)

Definition

Definition should include identifying operators used to compare one expression to a value or expression, including the following symbols and their functions:

- = equal to
- > greater than
- >= greater than or equal to
- < less than
- <= less than or equal to
- <> not equal to (or != or ^=)

Microsoft Imagine Academy Resources

Filtering Data
Access Lesson Plan: Modify Database Tables
Summer Job Openings (project)
Task Number 86

Restrict data, using the BETWEEN ... AND and IN clauses.

Definition

Demonstration should use the clauses in the following ways:

- BETWEEN ... AND—to show an upper and lower limitation for results returned from the query
- IN—to search for a match to a set of values and return only the matches

Microsoft Imagine Academy Resources

 Filtering Data
 Access Lesson Plan: Modify Database Tables
 Summer Job Openings (project)

NBEA Achievement Standards for Information Technology

Sort, prioritize, and retrieve data from databases.

Task Number 87

Restrict data, using wildcards and patterns within the LIKE condition.

Definition

Restriction with the LIKE condition should include the wildcard operator “%”, which represents one or many placeholders for any letter, and the “_” symbol, which represents one placeholder for any letter.

Microsoft Imagine Academy Resources
Task Number 88

Demonstrate the use of the ESCAPE character with wildcard characters.

Definition

Demonstration should include using the ESCAPE option backward slash (“\”) to indicate that the “_” or “%” is part of the name, not a wildcard value.

Microsoft Imagine Academy Resources

NBEA Achievement Standards for Information Technology

Sort, prioritize, and retrieve data from databases.

Task Number 89
Restrict (or specify) data containing nulls, using the IS (NOT) NULL clause.

Definition

Restriction should include the use of

- IS NULL to test for unavailable, unassigned, or unknown data
- IS NOT NULL, which tests for data that is present in the database.

Microsoft Imagine Academy Resources

Retrieving Unknown Values
Access Lesson Plan: Modify Database Tables
Summer Job Openings (project)

NBEA Achievement Standards for Information Technology

Sort, prioritize, and retrieve data from databases.

Task Number 90

Demonstrate the use of two or more conditional statements using logical operators (i.e., AND, OR, NOT).

Definition

Demonstration should include logical operators:

- AND, which seeks both sides to be TRUE
- OR, which seeks one side to be TRUE
- NOT, which seeks the reverse result

Note that the AND operator attaches itself to the nearest statement, and parentheses may be needed to force the order of operations.

Microsoft Imagine Academy Resources
NBEA Achievement Standards for Information Technology

Sort, prioritize, and retrieve data from databases.

---

**Task Number 91**

**Sort data by using ResultSet with the ORDER BY clause.**

**Definition**

Demonstration of the ORDER BY clause should include the following sorting or querying strategies performed on a given set of data:

- **Ascending**—will cause the result set to be ordered from smallest to largest
- **Descending**—is typically the default order, from largest to smallest
- **Multiple columns**—will produce sorting among results of a particular type
- **Column alias**—is an alternate name assigned to a column to prevent query conflicts

Note: ORDER BY is always the last clause, because it rearranges the final ResultSet after all selection criteria is executed.

**Microsoft Imagine Academy Resources**


 Formatting Result Sets


 Access Lesson Plan: Modify Database Tables


 Summer Job Openings (project)

NBEA Achievement Standards for Information Technology

Sort, prioritize, and retrieve data from databases.
Performing Single-Row Functions

Task Number 92

Explain the concept of functions.

Definition

Explanation should include the following:

- A function takes an input, executes a process, and produces an output.
- A function is an implementation of the input-process-output (IPO) model of computing.
- The IPO model receives inputs from a user or other source, performs some computations on the inputs, and returns the results of the computations.

Task Number 054

Distinguish between the two categories of functions in SQL.

Definition

Distinction should include

- single-row functions, which return a result (e.g., character, number, date, conversion) for a selected row
- group functions, which select multiple rows of data but return only one result (i.e., avg, count, max, min, stddev, sum, variance).

Task Number 93

Demonstrate restricting data using ANY and ALL.

Definition
Demonstration should include the following comparison conditions:

= ANY
> ANY
< ANY
> ALL
< ALL

Microsoft Imagine Academy Resources

2008—E-Learning Module 1
Writing Basic Subqueries
Access Lesson Plan: Modify Database Tables
Summer Job Openings (project)

Task Number 94

Define single-row functions.

Definition

Definition should include the following:

- Single-row functions operate on one row.
- Single-row functions return one result per row.
- Single-row functions can be used in the SELECT, WHERE, and ORDER BY clauses.

NBEA Achievement Standards for Information Technology

Extract useful information using search queries.

Task Number 95

Describe the types/categories of single-row functions.

Definition

Description should include
• character functions
• numeric functions
• date functions
• conversion functions
• null functions
• conditional functions
• other functions.

Task Number 96

Demonstrate the use of character, number, and date functions in SELECT statements.

Definition

Demonstration of character functions should include the following:

- UPPER
- LOWER
- INITCAP
- CONCAT
- SUBSTR
- LENGTH
- INSTR
- LPAD
- TRIM
- CONCAT
- LENGTH

Demonstration of numeric functions (performing operations on numeric data) should include the following:

- ROUND
- TRUNC
- MOD

Demonstration of date functions (performing operations on date data) should include the following:

- SYSDATE
- +
- -
- /
• *  
• MONTHS_BETWEEN  
• ADD_MONTHS  
• NEXT_DAY  
• LAST_DAY  
• ROUND  
• TRUNC

Microsoft Imagine Academy Resources

Using the Select Statement  
Access Lesson Plan: Creating Forms  
Summer Job Openings (project)

NBEA Achievement Standards for Information Technology

Extract useful information using search queries.

---

**Task Number 97**

**Describe the use of conversion functions.**

**Definition**

Description of conversion functions (converting data from one type to another) should include actual queries showing the conversion of

• character data to numbers  
• characters to dates  
• numbers to characters  
• dates to characters.

Conversions that will not result in a loss of data or precision may be completed implicitly, or without specific code performing the conversion. Explicit conversions must be hard-coded to be executable by the processor if the loss of data or precision is possible. Specific functions include

• TO_NUMBER  
• TO_DATE  
• TO_CHAR.
Conversion functions of dates also include format models that lay out a template for date and number conversions.

Conversions functions also include the NVL function that converts nulls to actual values, which can then be manipulated with numeric functions.

Description should also illustrate complex transformation of data by including queries where functions are nested within each other.

**Task Number 98**

**Demonstrate the use of null character handling.**

**Definition**

Demonstration should include queries that use the NVL, NVL2, NVLLIF, and COALESCE commands.

**Microsoft Imagine Academy Resources**


**Task Number 99**

**Demonstrate the use of conditional expressions.**

**Definition**

Demonstration should include writing queries that incorporate CASE or DECODE functions and allow the program to make decisions based on specific criteria.

**Microsoft Imagine Academy Resources**

Using JOIN Tables

Task Number 100

Describe the concept of joining data from two or more tables.

Definition

Description should include using JOINing to combine selected columns and rows of data from two or more tables based on a related column between the defined tables.

Microsoft Imagine Academy Resources

[3.162] SQL Server 2008 Release 2: Joining Data from Multiple Tables in Microsoft SQL Server 2008—E-Learning Module 1
  Querying Multiple Tables by Using Joins
  Access Lesson Plan: Modify Database Tables
  Summer Job Openings (project)

NBEA Achievement Standards for Information Technology

Extract useful information using search queries.

Task Number 101

Demonstrate the use of ANSI joins.

Definition

Demonstration should include using

- cross join
• natural join
• join ... using
• join ... on
• right outer join
• left outer join
• full outer join
• self-join
• non-equijoin.

NBEA Achievement Standards for Information Technology

Extract useful information using search queries.

---

**Task Number 102**

**Demonstrate the use of Oracle proprietary joins.**

**Definition**

Demonstration should include using

- Cartesian join
- equijoin, including self-join and outer join
- non-equijoin
- outer join.

---

**Aggregating Data Using GROUP Functions**

---

**Task Number 103**

**Define group/aggregate functions.**
Definition

Definition should include that group functions are commands that operate on sets of rows to give one result per group and include

- AVG
- COUNT
- MAX
- MIN
- STDDEV
- SUM
- VARIANCE.

Microsoft Imagine Academy Resources

  Summarizing Grouped Data

  Summer Job Openings (project)

Task Number 104

Describe how to write a query containing group functions.

Definition

Description should include the following guidelines for writing a query containing group functions:

- DISTINCT makes the function include unique values only.
- SELECT COUNT (DISTINCT StudentName)
- Results are implicitly ordered in ASC (ascending) when using a GROUP BY clause. DESC (descending) order can be enforced later in an ORDER BY clause.

Microsoft Imagine Academy Resources

  Summarizing Grouped Data

  Summer Job Openings (project)
Task Number 105

Construct SQL code by applying a GROUP BY clause that uses column, alias, and number.

Definition

Construction should include applying a GROUP BY clause in a query that uses a group function in the SELECT statement.

- Any column defined in the SELECT clause where a group function is used, that is NOT a group function, must be present in the GROUP BY clause.
- The GROUP BY clause will produce sub-totals on the field defined to group by.

Microsoft Imagine Academy Resources

Summarizing Grouped Data
Summer Job Openings (project)

Task Number 106

Construct code by applying a HAVING clause.

Definition

Construction should include the following:

- A WHERE clause may still be used to limit the results used to calculate the group function.
- The HAVING clause restricts the groups included in the GROUP BY.

Microsoft Imagine Academy Resources

Summarizing Grouped Data
Summer Job Openings (project)
Task Number 107

Demonstrate additional functionality of the GROUP BY clause.

Definition

Demonstration should include an explanation and example of the extension of the GROUP BY clause to add

- GROUP BY ROLLUP
- GROUP BY CUBE
- BY GROUPING sets.

Microsoft Imagine Academy Resources

  Summarizing Grouped Data
  Summer Job Openings (project)

Applying Advanced Data Selection Techniques

Task Number 108

Describe the types of problems that subqueries can solve.

Definition

Description should include problem scenarios, such as how a selection can be made from unknown criteria based on known facts.

Microsoft Imagine Academy Resources
Task Number 109

Define subqueries.

Definition

Definition should include the fact that there are three types of subquery, a query placed inside of an outer query to discover some unknown criteria necessary for a search:

- Single-row
- Multiple-row
- Multiple-column

Microsoft Imagine Academy Resources

Task Number 072

Construct a single-row subquery.

Definition

Construction should include the following syntax:

```
SELECT column, column
FROM  [table name]
WHERE column >= (  
```
SELECT column with compatible data type in the WHERE clause
FROM [table]
WHERE condition )

Note: Subqueries are complex queries that have a SELECT statement within a SELECT statement. The purpose of a subquery is to find the answer based on unknown criteria.

Microsoft Imagine Academy Resources

Writing Basic Subqueries
Access Lesson Plan: Modify Database Tables
Summer Job Openings (project)

---

Task Number 073

Construct a multi-row subquery.

Definition

Construction should include subqueries that return more than one value, using the following multiple-row comparison operators and guidelines for their use:

- **IN**—used when the outer query WHERE clause is designed to select only those rows that are equal to one of the list of values returned from the inner query
- **ANY**—used when the outer query WHERE clause is designed to select the rows that are equal to, less than, or greater than at least one value in the subquery result set
- **ALL**—used when the outer query WHERE clause is designed to select the rows that are equal to, less than, or greater than all values in the subquery result set
- **IN, ANY, ALL, and NOT**—used in combination with standard comparison operators (+, >, <, >=, <=)

Microsoft Imagine Academy Resources

Writing Basic Subqueries
Access Lesson Plan: Modify Database Tables
Summer Job Openings (project)
Task Number 074

Construct a correlated subquery.

Definition

Constructing a correlated subquery requires that the inner query execute once for each result returned in the parent query. A correlated subquery will have a JOIN from inner to outer query.

Microsoft Imagine Academy Resources

Writing Basic Subqueries
Access Lesson Plan: Modify Database Tables
Summer Job Openings (project)

Task Number 075

Write a multiple-column subquery.

Definition

Writing a multiple-column subquery should include a subquery written in the FROM clause. Syntax clause should include the following commands:

SELECT column1, column2, column3,
FROM table
WHERE (column1, column2, column3) IN
SELECT column1, column2, column3
FROM tablename
WHERE condition;

Related Standards of Learning

Microsoft Imagine Academy Resources

Task Number 076

Explain the behavior of subqueries when null values are retrieved.

Definition

Explanation should include the reason why no rows may be returned when the subquery encounters null values. Evaluate the results of nulls on multi-row subqueries, using

- IN
- ANY
- ALL.

Microsoft Imagine Academy Resources


Applying Data Manipulation Language

Task Number 077

Describe data manipulation language (DML).
Definition

Description should include the concept that DML only affects the data in a table, while the structure remains intact. Description should also include

- SELECT
- INSERT
- UPDATE
- DELETE
- MERGE.

Task Number 078

Describe integrity constraints.

Definition

Description should include

- a rationale for hard coding constraints on a table
- the difference between table-level and column-level constraints
- the importance of giving constraints a meaningful name.

Microsoft Imagine Academy Resources

Creating Complex Queries to Extract and Process Specific Information

Task Number 079

Insert rows into a table.

Definition

Insertion of rows should include

- understanding implicit and explicit INSERT syntax
- using the INSERT clause to add new data to a table (e.g., INSERT INTO [table name] [optional column list here]; VALUES [data, data, data, data]).

Microsoft Imagine Academy Resources
NBEA Achievement Standards for Information Technology

Enter data and edit fields and records.

---

Task Number 080

Update data within a table.

Definition

Updating data should include using the UPDATE command in the following general form:

```
UPDATE [table name]
SET column name = value
```

Microsoft Imagine Academy Resources

[NBEA Achievement Standards for Information Technology](#)

Task Number 081

---

NBEA Achievement Standards for Information Technology

Enter data and edit fields and records.
Delete rows in a table.

Definition

Deletion should include using the following syntax:

DELETE FROM [table name]
WHERE clause

In addition, omitting the WHERE clause will delete all rows in a table. The INSERT, UPDATE, and DELETE commands are part of the DML. This group of commands is not automatically committed to the database and stays in the buffer until a COMMIT occurs. A COMMIT can occur by

- typing a DML (or autocommit command)
- typing the keyword COMMIT
- gracefully exiting the database by typing the word EXIT
- selecting auto commit in Application Express.

If an unwanted delete is made, the keyword ROLLBACK can be typed and the buffer will be cleared of any commands it contains.

Microsoft Imagine Academy Resources

Merging Data in Tables
Access Lesson Plan: Modify Database Tables
Summer Job Openings (project)

NBEA Achievement Standards for Information Technology

Enter data and edit fields and records.

Task Number 082

Construct a MERGE statement.

Definition
Construction should include using the following format:

MERGE INTO table-name USING table/view/subquery
ON  what join condition
WHEN MATCHED THEN UPDATE SET
Col1 = Col2 value
WHEN NOT MATCHED THEN
INSERT (column_list)
VALUES (column_values)

NBEA Achievement Standards for Information Technology

Enter data and edit fields and records.

Task Number 083

Construct a multi-table insert statement.

Definition

Construction should include using the following format:

- **Unconditional statement**

  INSERT ALL
  INTO [table name] VALUES [data, data, data, data]
  INTO [table name] VALUES [data, data, data, data]

- **Conditional statement**

  INSERT ALL
  WHEN [column name] IN [value, value, value] THEN
  INTO [table name] VALUES [data, data, data, data]
  WHEN [column name] IN [value, value, value] THEN
  INTO [table name] VALUES [data, data, data, data]

NBEA Achievement Standards for Information Technology

Enter data and edit fields and records.
Modifying and Managing Tables

Task Number 084

Describe data definition language (DDL).

Definition

Description should include the following operations that change how data is defined in the database:

- CREATE
- DROP
- ALTER
- TRUNCATE
- RENAME

Task Number 085

Create a table.

Definition

Creation should include following these steps:

- Type the keyword CREATE TABLE [tablename]. Note that the table name cannot be more than 30 characters, start with anything but a letter; the keyword cannot be an SQL keyword and cannot contain any special characters except for the “$” and “_”.
- Choose the column names and the datatype and length necessary for each column (see the syntax example below).

```
CREATE TABLE employees(
    employee_id NUMBER(6),
    last_name VARCHAR2(25),
    email VARCHAR2(25),
    salary NUMBER(8,2),
    commission_pct NUMBER(2,2),
```
hire_date DATE,
department_id NUMBER(4),

- Add other constraints as demanded by the table instance charts to ensure data integrity. Note that examples of constrains are NOT NULL, PRIMARY KEY, UNIQUE KEY, FOREIGN KEY, and CHECK CONSTRAINT. Constraints may be added in the column definitions (called column-level constraints) or after the columns are defined (called table-level). NOT NULL constraints are a type of check constraint and can only be defined at column level. Constraints are needed in order to ensure data integrity.

NBEA Achievement Standards for Information Technology

Enter data and edit fields and records.

Task Number 086

Create a table, using a subquery.

Definition

The table should be created by using the following syntax:
CREATE TABLE tablename
[(column, column, …)]
AS
(subquery);

NBEA Achievement Standards for Information Technology

Enter data and edit fields and records.

Task Number 087

Describe the data dictionary.

Definition

Description should include the term metadata or the fact the Oracle Data Dictionary (DD) contains information about its structure, data about users, all objects (including tables, columns, and constraints), use statistics, and privileges.
Task Number 088

Define common data types.

Definition

Definition should include the concept that data types know two things about themselves: the type of information that a variable can hold, and how large it can get. This enforces business rules by constraining data in a way that completes effort checking before it gets to a higher level of checking. Description of data types should center around the types most often used, such as

- VARCHAR2()
- NUMBER()
- DATE
- CHAR
- LONG
- BLOB
- CLOB.

NBEA Achievement Standards for Information Technology

Enter data and edit fields and records.

Task Number 089

Describe date/time data types.

Definition

Description should include the following:

- DATE—stores point-in-time data in a table, and stores century, year, month, day, hours, minutes, and seconds
- knowing the default date format—DD-Mon-YYYY
- TIMESTAMP—stores values that are precise to within fractions of a second
- TIMESTAMP WITH TIME ZONE—stores time zone information in instances when coordination across geographic areas is important
- TIMESTAMP WITH LOCAL TIME ZONE—for use when the time zone is less important or when the time zone of the client system must be displayed

NBEA Achievement Standards for Information Technology
Task Number 090

Write code to alter table definitions.

Definition

Writing code should include using the alteration of a previously created table and using syntax such as the following:

```
ALTER TABLE employees
ADD CONSTRAINT emp_manager_fk
FOREIGN KEY(manager_id)
REFERENCES employees(employee_id);
```

The ALTER TABLE command is a DDL command, which means that it is automatically committed to the database when issued. Guidelines for altering the structure of a table is as follows:

- Add, drop, enable, or disable a constraint without the ability to modify its structure.
- Add a NOT NULL constraint to an existing column by using the MODIFY clause of the ALTER TABLE statement.
- Define a NOT NULL column only if the table is empty or if the column has a value for every row.

NBEA Achievement Standards for Information Technology

Enter data and edit fields and records.

Task Number 091

Write SQL code to manipulate column definitions, using DROP, RENAME, and TRUNCATE commands.

Definition

Writing code should include using the following DDL commands:
• DROP—deletes the entire table including the structure and data
• TRUNCATE—deletes the data content entered into the table, but leaves the structure of the table intact
• RENAME—changes the name of a table, view, sequence, or synonym

NBEA Achievement Standards for Information Technology

Enter data and edit fields and records.

---

**Task Number 092**

**Differentiate among TRUNCATE, DROP, and DELETE.**

**Definition**

Differentiation of the commands should include the following:

- **DROP**
  - removes a table from the database
  - removes all of the tables’ rows, indexes, and privileges
  - cannot be rolled back
- **TRUNCATE**
  - removes all rows from a table
  - cannot be rolled back
  - is faster and does not use as much undo space as a DELETE
- **DELETE**
  - removes rows from a table
  - uses a WHERE clause to remove specified rows
  - removes all rows, unless a WHERE condition is specified
  - is made permanent or undoes changes by using COMMIT or ROLLBACK

---

**Task Number 093**

**Describe FLASHBACK.**

**Definition**

Description of the FLASHBACK statement should include the following:
• The statement restores a table to an earlier state in the event of human or application error.
• The statement is limited by the amount of undo data in the system.
• The statement cannot be rolled back.

Defining Database Constraints

Task Number 094

Describe the necessity for database constraints.

Definition

Description should include following:

• Recollection of data-modeling principles provides data integrity by constraining what users may input into a certain column.
• Constraints enforce rules on data in a table whenever a row is inserted, updated, or deleted from a table.
• Constraints can prevent the deletion of a table if there are dependencies from another table.

Task Number 095

Distinguish between column-level and table-level constraints.

Definition

Distinction should include differences in the location of the code. A column-level constraint is the column definition, whereas a table-level constraint is coded after columns are defined.

Task Number 096

List the types of constraints and their applications.

Definition
List should include the following types of constraints:

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK

Constraints can be found in the data dictionary.

**Task Number 097**

**Write a table- and column-level constraint.**

**Definition**

Constraints should be created at the time of table creation; syntax should follow these examples:

- Column-level constraint (demonstrates PRIMARY KEY and NOT NULL at the column level)

  CREATE TABLE Employee
  (EmpID NUMBER(4) CONSTRAINT Employee_EmpID_PK PRIMARY KEY
  EmpLN VARCHAR2(40) CONSTRAINT Employee_EmpLN_NN NOT NULL)

- Table-level constraint (demonstrates PRIMARY KEY at the table level, with NOT NULL at the column level)

  CREATE TABLE Employee
  (EmpID NUMBER(4) EmpLN VARCHAR2(40) CONSTRAINT Employee_EmpLN_NN
  NOT NULL
  CONSTRAINT Employee_EmpID_PK PRIMARY KEY(EmpID))

**Related Standards of Learning**

**Mathematics**

COM.3
The student will write program specifications that define the constraints of a given problem.

---

**Task Number 098**
View constraints with table definition and the data dictionary.

Definition

Viewing constraints requires creating the constraint for the user table and should include the following:

```
SELECT * or (column name, column name)
FROM user_constraints
WHERE table_name = [table name];
```

Constraints not owned by the user may also be viewed by selecting from all constraints.

Creating and Managing Views

Task Number 099

Describe a view.

Definition

Description of a view should include a list of other database objects (e.g., table, index, synonym, sequence) and the following:

- Views can be described as virtual tables that contain groups of data from various tables and are used to save time or limit user access to sensitive data.
- A view is stored in the database as a piece of code and does not take up a large amount of memory as other database objects do.
- Each time the user asks to see a view, the code that created the view is executed and displayed. Consequently, the view contains the most up-to-date information as defined by the creation criteria.

Task Number 100

Create a view.

Definition
Creation of a view consists of using the keywords CREATE VIEW and the full syntax (Syntax in the square brackets is optional):

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW view
[(alias[, alias]...)]
AS subquery
[WITH CHECK OPTION [CONSTRAINT constraint]]
[WITH READ ONLY [CONSTRAINT constraint]];
```

---

**Task Number 101**

**Write SQL code to retrieve data through a view.**

**Definition**

Writing code to select data through a view should be identical to other SELECT statements that retrieve data, with the exception that a view name should be substituted for a table name.

**Task Number 102**

**Write SQL code to alter the definition of a view.**

**Definition**

Altering a definition of a view should include the following syntax:

```
CREATE OR REPLACE VIEW [name]
(column, column, column)
AS SELECT column, column, column
FROM  [table_name]
WHERE  condition to create a view
```

**Task Number 103**

**Manipulate tables, using the INSERT, UPDATE, and DELETE commands through a view.**

**Definition**

Manipulation of DML commands through a view is executed using the following rules:
1. A row may not be removed if the view contains
   - group functions
   - a GROUP BY clause
   - the DISTINCT keyword
   - the pseudocolumn ROWNUM keyword.

2. Data may not be modified in a view if it contains
   - group functions
   - a GROUP BY clause
   - the DISTINCT keyword
   - the pseudocolumn ROWNUM keyword
   - columns defined by expressions.

3. Data may not be added through a view if the view includes
   - group functions
   - a GROUP BY clause
   - the DISTINCT keyword
   - the pseudocolumn ROWNUM keyword
   - columns defined by expressions
   - NOT NULL columns in the base tables that are not selected by the view.

Task Number 104

Write SQL code to remove a view.

Definition

Writing code should include the DROP VIEW command, the name of the view, and reflect the following syntax:

DROP VIEW nameOfview;

Note that a view created with the CREATE OR REPLACE option will automatically drop the old view of the same name and replace it with the new code. A view can be completely eradicated with the code above.

Related Standards of Learning

Mathematics

COM.1
The student will design and apply computer programs to solve practical problems in mathematics arising from business and applications in mathematics.

**COM.3**
The student will write program specifications that define the constraints of a given problem.

---

**Task Number 105**

**Create an inline view.**

**Definition**

Creation consists of a SELECT statement within a FROM clause of another SELECT statement, such as in this example:

```
SELECT a.<column 1>, a.<column 2>, b.<column 1>
FROM <table 1>a, (SELECT <column 1 or expression>, <column n or expression> FROM <table 2> GROUP BY column 1) b
WHERE a.<column 3> =b.<column 1>
```

---

**Task Number 106**

**Describe the procedures for performing a top-n analysis.**

**Definition**

Description should include the concept that a top-n analysis displays only the top or bottom \( n \) records (with \( n \) representing an integer) from ordered sets of data (e.g., top five earners, last 10 employees, 25 best-selling products), and use the following syntax:

```
SELECT [column_list], ROWNUM [ROWNUM_ALIAS]
FROM (SELECT [column_list] FROM table ORDER BY Top-N column)
WHERE ROWNUM <=n;
```

---

**Creating Additional Database Objects**

---

**Task Number 107**
Describe database objects and their uses.

Definition

Description should include a review of the list of database objects, including

- TABLE (holds data)
- VIEW (represents a subset of data)
- SEQUENCE (generates primary key values)
- INDEX (improves the performance of some queries)
- SYNONYM (represents alternative name for an object that does not replace the old name).

Task Number 108

Define a sequence object.

Definition

Definition should include the concept that a sequence object

- is a separate database object that supplies a number to another database object, such as to a table
- describes when the sequence object is used
- describes how the sequence object is used.

Task Number 109

Create a sequence.

Definition

Creation of a sequence includes the syntax as follows, with \( n \) representing an integer:

```
CREATE SEQUENCE name
INCREMENT BY n
START WITH n
MAXVALUE n
CYCLE
CACHE n;
```
Task Number 110

Use a sequence.

Definition

Using a sequence requires the following syntax:

```
INSERT INTO table name (column, column)
VALUES (value, NEXTVAL, data, data)
```

Task Number 111

Modify a sequence.

Definition

Modification should include using the ALTER SEQUENCE statement to change the increment, minimum and maximum values, cached numbers, and behavior of an existing sequence. The following syntax is required:

```
ALTER SEQUENCE <sequencename>
[INCREMENT BY n]
[START WITH s]
[MAXVALUE x / NOMAXVALUE]
[MINVALUE m / NOMINVALUE]
[CYCLE / NOCYCLE]
[CACHE c / NOCACHE] [ORDER / NOORDER];
```

This statement affects only future sequence numbers used.

Task Number 112

Describe the index.

Definition
Description should include that an index

- is a schema object used to speed up the retrieval of rows by using a pointer
- requires the following syntax:

```sql
CREATE INDEX name
ON [table name] (column)
```

- should be maintained at regularly scheduled periods to ensure efficient operation.

---

**Task Number 113**

**Describe issues that affect the decision to create an index.**

**Definition**

Description should include the following questions and issues that can affect the runtime speed of the query:

- Does the column have a wide range of values?
- Does the column contain a large number of nulls?
- Are columns used frequently in a WHERE?
- Is the table large and are most queries designed to retrieve 2 percent to 4 percent of rows?

When the following are true, an index should be created:

- When the column is used frequently in a WHERE clause or a join condition
- When the column contains a wide range of values
- When the column contains a large number of null values
- When two or more columns are frequently used together in a WHERE clause or a join condition
- When the table is large and most queries are expected to retrieve less than 2 percent to 4 percent of the rows

When the following are not true, an index should not be created:

- When the table is small
- When the columns are not often used as a condition in the query
- When most queries are expected to retrieve more than 2 percent to 4 percent of the rows.
- When the table is updated frequently
Task Number 114

Create an index.

Definition

Creation should include

- focusing on the following syntax:

CREATE INDEX index1 ON TABLE (col1)

- maintaining indexes at regularly scheduled periods to ensure efficient operation.

NBEA Achievement Standards for Information Technology

Describe search strategies and use them to solve common information problems.

Task Number 115

Create private and public synonyms.

Definition

Creation of an index should include using the following syntax:

CREATE PUBLIC SYNONYM name FOR object name

Note: As a point of reference, aliases in a select clause only pertain to that query. Synonyms are permanent. A private synonym must be distinct from all other objects owned by the same user.

Task Number 116

Describe the type of information available in the data dictionary.
Definition

Description should include the following:

- Users and their privileges
- Tables, columns and their data types, integrity constraints, indexes
- Statistics about tables and indexes used by the optimizer
- Privileges granted on database objects
- Storage structures of the database

Task Number 117

Create SQL code to retrieve information from the data dictionary.

Definition

Creation should include the following syntax:

- SELECT* FROM dictionary (to access the data dictionary)
- SELECT* FROM user_tables (to see the names and characteristics of tables owned by the user)
- SELECT* FROM user_catalog (to view all objects owned by the user)

Maintaining Database Security and System Security

Task Number 118

Define object privileges.

Definition
Definition should include the following:

- Object privileges are managed with the Data Control Language (DCL), which authorizes users to access and manipulate data.
- The two statements for object privileges are:
  - GRANT—authorizes user(s) to perform an operation or set of operations on an object
  - REVOKE—removes access to perform an operation on an object

---

**Task Number 119**

**Construct an object privilege.**

**Definition**

Construction should be in the following format:

```
GRANT UPDATE (column names)
ON table_name
TO user_name;
```

---

**Task Number 120**

**Describe roles.**

**Definition**

Description should include the following:

- Roles are named groups of related system and object privileges
- Roles can be granted to users
- Roles can be granted to other roles

---

**Task Number 121**

**Define system privileges.**
Definition

Definition of system privileges should include

- CREATE USER
- DROP USER
- DROP ANY TABLE
- BACK UP ANY TABLE
- SELECT ANY TABLE
- CREATE ANY TABLE.

Task Number 122

Describe system privileges that can be granted to a user.

Definition

Description should identify who should be granted which privileges and what happens if those privileges are revoked in the case of with GRANT option.

Task Number 123

Define a database link.

Definition

Definition should include the concept that a link is a pointer that establishes a one-directional communication path from one database server to another.

Task Number 124

Create a new database user.

Definition

Creation should include the following syntax:
CREATE USER username IDENTIFIED BY password;
Task Number 125

Drop a database user.

Definition

Dropping a database user should include the following syntax:
DROP USER username

Making Database Transactions

Task Number 126

Define Transaction Control Language (TCL).

Definition

Definition should include the following:

- TRANSACTION—a logical unit of statements that must be executed together
- ROLLBACK—will clear the buffer without saving changes (implicit for all DDL or DCL statements)
- SAVEPOINT—creates a chronological marker that can be named in a rollback
- COMMIT—ends the transaction and makes changes in the buffer permanent
- Read consistency—guarantees a consistent view of the data at all times

Task Number 127

Describe the importance of transaction control to businesses.

Definition
Description should include the following:

- Allows rule-based implementation of business processes and policies
- Allows pattern-based implementation of business processes or policies
- Provides integrity behind the data on which many business decisions are made
- May produce internal accountability

**Related Standards of Learning**

**English**

**10.5**
The student will read, interpret, analyze, and evaluate nonfiction texts.

a. Analyze text features and organizational patterns to evaluate the meaning of texts.
b. Recognize an author’s intended audience and purpose for writing.
c. Skim materials to develop an overview and locate information.
d. Compare and contrast informational texts for intent and content.
e. Interpret and use data and information in maps, charts, graphs, timelines, tables, and diagrams.
f. Draw conclusions and make inferences on explicit and implied information using textual support as evidence.
g. Analyze and synthesize information in order to solve problems, answer questions, and generate new knowledge.
h. Analyze ideas within and between selections providing textual evidence.
i. Summarize, paraphrase, and synthesize ideas, while maintaining meaning and a logical sequence of events, within and between texts.
j. Use reading strategies throughout the reading process to monitor comprehension.

**11.5**
The student will read, interpret, analyze, and evaluate a variety of nonfiction texts including employment documents and technical writing.

a. Apply information from texts to clarify understanding of concepts.
b. Read and correctly interpret an application for employment, workplace documents, or an application for college admission.
c. Analyze technical writing for clarity.
d. Paraphrase and synthesize ideas within and between texts.
e. Draw conclusions and make inferences on explicit and implied information using textual support.
f. Analyze multiple texts addressing the same topic to determine how authors reach similar or different conclusions.
g. Analyze false premises, claims, counterclaims, and other evidence in persuasive writing.
h. Recognize and analyze use of ambiguity, contradiction, paradox, irony, sarcasm, overstatement, and understatement in text.

i. Generate and respond logically to literal, inferential, evaluative, synthesizing, and critical thinking questions about the text(s).

Task Number 128

Demonstrate the use of SAVEPOINT, ROLLBACK, and COMMIT.

Definition

Demonstration should include a process flow that demonstrates the following, in order (with autocommit turned off):

1. Execute a transaction against the data (insert or update).
2. Execute a transaction against the data (insert or update).
3. Create SAVE POINTA.
4. Execute a transaction against the data (insert or update).
5. Execute a transaction against the data (insert or update).
6. Create SAVE POINTB.
7. Execute a transaction against the data (insert or update).
8. ROLLBACK to SAVEPOINTA.
9. Execute a transaction against the data (insert or update).
10. Execute a transaction against the data (insert or update).
11. COMMIT.
12. Run a query and validate the updates made to the data.

Preparing for Industry Certification

Task Number 129
Describe the process and requirements for obtaining industry certifications related to the Database Design and Management (Oracle) course.

Definition

Description should include a list of industry certifications related to the Database Design and Management (Oracle) course and the process and requirements for obtaining the certifications from

- official websites of the testing organization/vendor
- practice materials and tests 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

NBEA Achievement Standards for Information Technology

Obtain database management industry certification(s) needed for a chosen career path.

Task Number 130

Identify testing skills and strategies for a certification examination.

Definition

Identification of testing skills and strategies 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 131

Demonstrate ability to successfully complete selected practice examinations.

Definition

Demonstration should include selected practice examinations for 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 132

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 Management (Oracle) skills.

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

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Subject Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>Research the history of databases.</td>
<td>English: 10.8, 11.8&lt;br&gt;History and Social Science: VUS.12, VUS.13, VUS.14, WG.17, WHII.12, WHII.14</td>
</tr>
<tr>
<td>40</td>
<td>List the major types of databases.</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>Distinguish between a conceptual and a physical database model.</td>
<td>Mathematics: COM.10</td>
</tr>
<tr>
<td>42</td>
<td>Compare the structure of relational and non-relational database structures.</td>
<td>English: 10.5, 11.5</td>
</tr>
<tr>
<td>43</td>
<td>Identify the characteristics of a relational database.</td>
<td>English: 10.5, 11.5</td>
</tr>
<tr>
<td>44</td>
<td>Examine the database development life cycle.</td>
<td>English: 10.5, 11.5&lt;br&gt;Mathematics: COM.1, COM.4, COM.6, COM.10, COM.13, COM.16</td>
</tr>
<tr>
<td>45</td>
<td>Research the future direction of database technologies.</td>
<td>English: 10.8, 11.8</td>
</tr>
<tr>
<td>46</td>
<td>Describe the process of modeling business requirements.</td>
<td>English: 10.1, 10.5, 11.1, 11.5</td>
</tr>
<tr>
<td>47</td>
<td>Apply business concepts to the database model.</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>Define entities among elements of significance.</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>Define attributes of each entity.</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>Select unique identifiers (UIDs).</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>Define types of unique identifiers.</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>Define business rules.</td>
<td>English: 10.1, 10.5, 11.1, 11.5</td>
</tr>
<tr>
<td>53</td>
<td>Analyze entities for relationships that exist among them.</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>Distinguish among relationship types.</td>
<td>Mathematics: A.7, A.8, AFDA.1</td>
</tr>
<tr>
<td>55</td>
<td>Describe relationship transferability.</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>Name relationships.</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>Explain relationship optionality.</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>Explain relationship degree/cardinality.</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>Identify elements of the graphic representation of a database model.</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>Define drawing conventions for readability.</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>Illustrate business rules in an entity-relationship model.</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>Define the normalization process.</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>Perform the normalization process.</td>
<td>English: 10.5, 11.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>Resolve many-to-many relationships.</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>Model hierarchical data.</td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>Model recursive relationships.</td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>Model exclusive relationships.</td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>Define relational-database terminology.</td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>Define a fact and dimension.</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>Verbalize a diagram's relationship notation.</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>Summarize the database-design process.</td>
<td>English: 10.5, 11.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mathematics: A.1, COM.1, COM.4, COM.16</td>
</tr>
<tr>
<td>72</td>
<td>Convert a conceptual design to a physical database model.</td>
<td>Mathematics: A.1, COM.10</td>
</tr>
<tr>
<td>73</td>
<td>Map simple entities, attributes, and primary keys.</td>
<td>Mathematics: A.7, COM.14</td>
</tr>
<tr>
<td>74</td>
<td>Identify data constraints.</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>Map relationships to foreign keys.</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>Describe SQL.</td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>Distinguish among categories of SQL statements.</td>
<td>English: 10.3, 10.5, 11.3, 11.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mathematics: COM.14</td>
</tr>
<tr>
<td>78</td>
<td>Demonstrate the syntax for select statements (projection).</td>
<td>Mathematics: COM.14</td>
</tr>
<tr>
<td>79</td>
<td>Demonstrate methods for selecting columns and arithmetic expressions (selection).</td>
<td>Mathematics: A.7, COM.14</td>
</tr>
<tr>
<td>80</td>
<td>Incorporate column alias and literals in a SELECT statement.</td>
<td>Mathematics: COM.14</td>
</tr>
<tr>
<td>81</td>
<td>Describe operator precedence.</td>
<td>English: 10.5, 11.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Mathematics: COM.6</td>
</tr>
<tr>
<td>82</td>
<td>Describe methods for displaying a table.</td>
<td>Mathematics: COM.9, COM.11</td>
</tr>
<tr>
<td>83</td>
<td>Restrict data, using the WHERE clause.</td>
<td>Mathematics: COM.7, COM.13</td>
</tr>
<tr>
<td>84</td>
<td>Define comparison operators (e.g., =, &gt;, &lt;, &gt;=, &lt;=, &lt;&gt;, !=)</td>
<td>Mathematics: A.5, COM.7, COM.13</td>
</tr>
<tr>
<td>85</td>
<td>Restrict data, using the BETWEEN ... AND and IN clauses.</td>
<td>Mathematics: COM.7, COM.13</td>
</tr>
<tr>
<td>86</td>
<td>Restrict data, using wildcards and patterns within the LIKE condition.</td>
<td>Mathematics: COM.7, COM.13</td>
</tr>
<tr>
<td>87</td>
<td>Demonstrate the use of the ESCAPE character with wildcard characters.</td>
<td>Mathematics: COM.7, COM.13</td>
</tr>
<tr>
<td>88</td>
<td>Restrict (or specify) data containing nulls, using the IS (NOT) NULL clause.</td>
<td>Mathematics: COM.7, COM.13</td>
</tr>
<tr>
<td>89</td>
<td>Demonstrate the use of two or more conditional statements using logical operators (i.e., AND, OR, NOT).</td>
<td>Mathematics: COM.8, COM.13</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>90</td>
<td>Sort data by using ResultSet with the ORDER BY clause.</td>
<td>Mathematics: COM.7, COM.9</td>
</tr>
<tr>
<td>91</td>
<td>Explain the concept of functions.</td>
<td>Mathematics: A.7, AFDA.1</td>
</tr>
<tr>
<td>92</td>
<td>Distinguish between the two categories of functions in SQL.</td>
<td></td>
</tr>
<tr>
<td>93</td>
<td>Demonstrate restricting data using ANY and ALL.</td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>Define single-row functions.</td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>Describe the types/categories of single-row functions.</td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>Demonstrate the use of character, number, and date functions in SELECT statements.</td>
<td>Mathematics: COM.7, COM.8</td>
</tr>
<tr>
<td>97</td>
<td>Describe the use of conversion functions.</td>
<td>English: 10.5, 11.5</td>
</tr>
<tr>
<td>98</td>
<td>Demonstrate the use of null character handling.</td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>Demonstrate the use of conditional expressions.</td>
<td>Mathematics: COM.8</td>
</tr>
<tr>
<td>100</td>
<td>Describe the concept of joining data from two or more tables.</td>
<td>English: 10.5, 11.5</td>
</tr>
<tr>
<td>101</td>
<td>Demonstrate the use of ANSI joins.</td>
<td>Mathematics: COM.3, COM.7, COM.8</td>
</tr>
<tr>
<td>102</td>
<td>Demonstrate the use of Oracle proprietary joins.</td>
<td>Mathematics: COM.3, COM.7, COM.8</td>
</tr>
<tr>
<td>103</td>
<td>Define group/aggregate functions.</td>
<td>English: 10.2, 10.5, 11.2, 11.5</td>
</tr>
<tr>
<td>104</td>
<td>Describe how to write a query containing group functions.</td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>Construct SQL code by applying a GROUP BY clause that uses column, alias, and number.</td>
<td></td>
</tr>
<tr>
<td>106</td>
<td>Construct code by applying a HAVING clause.</td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>Demonstrate additional functionality of the GROUP BY clause.</td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>Describe the types of problems that subqueries can solve.</td>
<td>English: 10.5, 11.5</td>
</tr>
<tr>
<td>109</td>
<td>Define subqueries.</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>Construct a single-row subquery.</td>
<td>Mathematics: COM.9</td>
</tr>
<tr>
<td>111</td>
<td>Construct a multi-row subquery.</td>
<td>Mathematics: COM.9</td>
</tr>
<tr>
<td>112</td>
<td>Construct a correlated subquery.</td>
<td>Mathematics: COM.9</td>
</tr>
<tr>
<td>113</td>
<td>Write a multiple-column subquery.</td>
<td>Mathematics: COM.9</td>
</tr>
<tr>
<td>114</td>
<td>Explain the behavior of subqueries when null values are retrieved.</td>
<td>English: 10.5, 11.5</td>
</tr>
<tr>
<td>115</td>
<td>Describe data manipulation language (DML).</td>
<td>English: 10.5, 11.5</td>
</tr>
<tr>
<td>116</td>
<td>Describe integrity constraints.</td>
<td></td>
</tr>
<tr>
<td>117</td>
<td>Insert rows into a table.</td>
<td>Mathematics: COM.14</td>
</tr>
<tr>
<td>118</td>
<td>Update data within a table.</td>
<td>Mathematics: COM.14</td>
</tr>
<tr>
<td>119</td>
<td>Delete rows in a table.</td>
<td>Mathematics: COM.14</td>
</tr>
<tr>
<td>120</td>
<td>Construct a MERGE statement.</td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>Construct a multi-table insert statement.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>122</td>
<td>Describe data definition language (DDL).</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>Create a table.</td>
<td></td>
</tr>
<tr>
<td>124</td>
<td>Create a table, using a subquery.</td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>Describe the data dictionary.</td>
<td></td>
</tr>
<tr>
<td>126</td>
<td>Define common data types.</td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>Describe date/time data types.</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>Write code to alter table definitions.</td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>Write SQL code to manipulate column definitions, using DROP, RENAME, and TRUNCATE commands.</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>Differentiate among TRUNCATE, DROP, and DELETE.</td>
<td></td>
</tr>
<tr>
<td>131</td>
<td>Describe FLASHBACK.</td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>Describe the necessity for database constraints.</td>
<td></td>
</tr>
<tr>
<td>133</td>
<td>Distinguish between column-level and table-level constraints.</td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>List the types of constraints and their applications.</td>
<td></td>
</tr>
<tr>
<td>135</td>
<td>Write a table- and column-level constraint.</td>
<td></td>
</tr>
<tr>
<td>136</td>
<td>View constraints with table definition and the data dictionary.</td>
<td></td>
</tr>
<tr>
<td>137</td>
<td>Describe a view.</td>
<td></td>
</tr>
<tr>
<td>138</td>
<td>Create a view.</td>
<td></td>
</tr>
<tr>
<td>139</td>
<td>Write SQL code to retrieve data through a view.</td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>Write SQL code to alter the definition of a view.</td>
<td></td>
</tr>
<tr>
<td>141</td>
<td>Manipulate tables, using the INSERT, UPDATE, and DELETE commands through a view.</td>
<td></td>
</tr>
<tr>
<td>142</td>
<td>Write SQL code to remove a view.</td>
<td></td>
</tr>
<tr>
<td>143</td>
<td>Create an inline view.</td>
<td></td>
</tr>
<tr>
<td>144</td>
<td>Describe the procedures for performing a top-n analysis.</td>
<td></td>
</tr>
<tr>
<td>145</td>
<td>Describe database objects and their uses.</td>
<td></td>
</tr>
<tr>
<td>146</td>
<td>Define a sequence object.</td>
<td></td>
</tr>
<tr>
<td>147</td>
<td>Create a sequence.</td>
<td></td>
</tr>
<tr>
<td>148</td>
<td>Use a sequence.</td>
<td></td>
</tr>
<tr>
<td>149</td>
<td>Modify a sequence.</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>Describe the index.</td>
<td></td>
</tr>
<tr>
<td>151</td>
<td>Describe issues that affect the decision to create an index.</td>
<td></td>
</tr>
<tr>
<td>152</td>
<td>Create an index.</td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>Create private and public synonyms.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>154</td>
<td>Describe the type of information available in the data dictionary.</td>
<td></td>
</tr>
<tr>
<td>155</td>
<td>Create SQL code to retrieve information from the data dictionary.</td>
<td></td>
</tr>
<tr>
<td>156</td>
<td>Define object privileges.</td>
<td></td>
</tr>
<tr>
<td>157</td>
<td>Construct an object privilege.</td>
<td></td>
</tr>
<tr>
<td>158</td>
<td>Describe roles.</td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>Define system privileges.</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>Describe system privileges that can be granted to a user.</td>
<td></td>
</tr>
<tr>
<td>161</td>
<td>Define a database link.</td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>Create a new database user.</td>
<td></td>
</tr>
<tr>
<td>163</td>
<td>Drop a database user.</td>
<td></td>
</tr>
<tr>
<td>164</td>
<td>Define Transaction Control Language (TCL).</td>
<td></td>
</tr>
<tr>
<td>165</td>
<td>Describe the importance of transaction control to businesses.</td>
<td>English: 10.5, 11.5</td>
</tr>
<tr>
<td>166</td>
<td>Demonstrate the use of SAVEPOINT, ROLLBACK, and COMMIT.</td>
<td></td>
</tr>
<tr>
<td>167</td>
<td>Describe the process and requirements for obtaining industry certifications related to the Database Design and Management (Oracle) course.</td>
<td>English: 10.5, 10.8, 11.5, 11.8</td>
</tr>
<tr>
<td>168</td>
<td>Identify testing skills and strategies for a certification examination.</td>
<td></td>
</tr>
<tr>
<td>169</td>
<td>Demonstrate ability to successfully complete selected practice examinations.</td>
<td></td>
</tr>
<tr>
<td>170</td>
<td>Complete an industry certification examination representative of skills learned in this course (e.g., MOS, MTA, IC3).</td>
<td></td>
</tr>
</tbody>
</table>

### Teacher Resources

#### Instructional Scenarios

**Global Locations Table**

**Duty/Concept Area(s):**

Defining Database Constraints  
Creating Additional Database Objects  
Maintaining Database Security and System Security  
Making Database Transactions

**Scenario:**

---

84
Global Fast Foods has been successful this past year and has opened several new stores. They need to add a table to their database to store information about each of their store’s locations. They want to store the following information: Id, name, date_opened, address, city, ZIP Code (extended format), phone (including area code), email, fax number, web address, manager_id, number_of_employees.

Requirements:

- Write the CREATE TABLE statement to create the GLOBAL_LOCATIONS table.
- For each column, supply an appropriate data type and length.
- Set the date_opened to a default value of SYSDATE.
- Set the web address to have a default value of www.globalfastfood.com.
- Add a table-level constraint that creates a composite unique key between the phone and email.
- Describe the table.
- Query the data dictionary to display the constraint names, types, and delete_rule.

Big Question:

What coding is required to create a table with default values and constraints?

Focused Questions:

1. Which data types are appropriate for the required field?
2. How is a default value coded?
3. What is the proper syntax to add a table- or column-level constraint?
4. How do you query the data dictionary to view information regarding constraints?

SOL Correlation:

Mathematics
COM 3, COM.16

Project-Based Assessment:

- Properly code the CREATE TABLE statement in SQL.
- Use a database application such as Access to create the table.

Football Team Statistics

Duty/Concept Area(s):

Applying Concepts to Database Models
Transitioning from Design Concept to Database
Modifying and Managing Tables
Restricting and Sorting Data Using SQL
Scenario:
Your high school football team would like you to create a database to maintain records on player statistics for all team members. They would also like complete records of every game. Statistics need to be kept for each player on both offense and defense. Offensive statistics include passing attempts, complete passes, passing yards, interceptions, rushing yards, receiving yards, receptions, touchdowns, fumbles, field goals attempted, and field goals made. Defensive statistics include tackles, interceptions, and sacks.

Big Questions:
- What is the best way to create a database to accomplish the task?
- What software should be used?

Focused Questions:
- How do you retrieve specific information from the database?
- How many tables will the database need?
- Will you need to create a data model?
- How will you enter information into the database?
- Do your NOT NULL restraints work?
- Do your foreign key relationships work?
- Do your primary key constraints work?

SOL Correlation:

Mathematics
A.1, COM.13, COM.15, COM.16

Project-Based Assessment:
- A script for creating each table
- A script for each table to insert information
- Any additional code used to complete the project, such as adding foreign keys

Resources:
- NFL.com for sample statistics
- MySQL for possible database software.

Creating Views & Database Security

Duty/Concept Area(s):
Creating and Managing Views
Maintaining Database Security and System Security
Scenario:

As the company’s database administrator (DBA) you are asked by management to provide access to data from the database. Since most employees do not need access to the entire database or entire tables, you are asked to provide a solution.

You have supervisors who need to look up information about the employees in their department but not in others. It is also not necessary for these supervisors to have access to every column of information about these employees. You need to give permission to certain users to query data but make no changes or updates. Since confidentiality and security are key concerns, you must find a way to provide access only to the required information.

You have also been given the task to create new users for the database and assign roles and permissions to current users.

Requirements

1. Create a simple tool for a manager to see the names, salary, department, and hire dates for their assigned employees without giving them access to the base tables.
2. Create new users with usernames and passwords.
3. Use GRANT and REVOKE commands to give privileges to database users.
4. Assign privileges and rights to the new users created above. Be specific and assign rights to certain tables and restrict some users to only certain columns in a table.
   - Possible privilege and rights commands
   - SELECT, INSERT, UPDATE, DELETE, INDEX, ALTER, ALL
5. Use the proper SQL commands to remove privileges and rights from certain users.

Big Question:

How can you provide access to selected data in the database to employees who need it while maintaining security and restricting access to parts of the data tables?

Focused Questions:

- How can you create a snapshot or window to data that only shows certain columns and rows from the base tables?
- How would you permanently delete the snapshot or tool you created in the previous questions when they are no longer needed, while preserving the integrity of the data in the base tables?

SOL Correlation:

*English*

10.5, 11.5

*Mathematics*

COM.1, COM.3, COM.13
Project-Based Assessment:

- Students may work individually or in groups of two or three to accomplish the tasks in this project.
- Students may create an electronic presentation of their solutions.

Note: This scenario can be used with many of the data tables provided in the Application Express environment through the Oracle Academy. Students can also use other data tables they have or create their own tables and populate them with data to use with this scenario.

Resources:

Online:
Oracle Academy data tables available in Oracle Application Express online development environment.

Textbooks:


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.”
Appendix: Credentials, Course Sequences, and Career Cluster Information

Industry Credentials: Only apply to 36-week courses

- 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 Database Design and Programming with SQL Examination
- 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.

- Computer Information Systems (6612/36 weeks)
- Computer Information Systems (6614/18 weeks)
- Computer Information Systems, Advanced (6613/36 weeks)
- Computer Information Systems, Advanced (6615/18 weeks)
- Computer Network Software Operations (6650/36 weeks)
- Computer Network Software Operations, Advanced (6651/36 weeks)
- Cybersecurity Software Operations (6304/36 weeks)
- Database Design and Management with PL/SQL (Oracle) (6662/36 weeks)
- Digital Applications (6611/36 weeks)
- Digital Applications (6617/18 weeks)
- Information Technology Fundamentals (6670/36 weeks)
- International Baccalaureate Information Technology in a Global Society (IB6613/36 weeks)
- Java Programming (Oracle) (6661/36 weeks)
- Office Administration (6621/36 weeks)
- Office Administration (6622/18 weeks)
- Office Specialist I--Preparation (6740/36 weeks)
- Office Specialist II--Preparation (6741/36 weeks)
- Office Specialist III--Preparation (6742/36 weeks)
- Programming (6640/36 weeks)
- Programming, Advanced (6641/36 weeks)
### Career Cluster: Information Technology

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Occupations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information Support and</td>
<td>Computer Support Specialist</td>
</tr>
<tr>
<td>Services</td>
<td>Computer Systems Engineer, Architect</td>
</tr>
<tr>
<td></td>
<td>Database Administrator</td>
</tr>
<tr>
<td></td>
<td>Database Analyst</td>
</tr>
<tr>
<td></td>
<td>Network Systems and Data Communication Analyst</td>
</tr>
<tr>
<td></td>
<td>Software Test Engineer</td>
</tr>
<tr>
<td>Network Systems</td>
<td>Computer and Information Systems Administrator</td>
</tr>
<tr>
<td></td>
<td>Computer Support Specialist</td>
</tr>
<tr>
<td></td>
<td>Computer Systems Engineer, Architect</td>
</tr>
<tr>
<td></td>
<td>Database Analyst</td>
</tr>
<tr>
<td></td>
<td>Network Systems and Data Communication Analyst</td>
</tr>
<tr>
<td></td>
<td>Software Test Engineer</td>
</tr>
</tbody>
</table>

### Career Cluster: Science, Technology, Engineering and Mathematics

<table>
<thead>
<tr>
<th>Pathway</th>
<th>Occupations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engineering and Technology</td>
<td>Computer Hardware Engineer</td>
</tr>
<tr>
<td></td>
<td>Computer Programmer</td>
</tr>
<tr>
<td></td>
<td>Computer Software Engineer</td>
</tr>
<tr>
<td></td>
<td>Network and Computer Systems Administrator</td>
</tr>
<tr>
<td></td>
<td>Network Systems and Data Communication Analyst</td>
</tr>
<tr>
<td></td>
<td>Production, Planning, Expediting Clerk</td>
</tr>
<tr>
<td></td>
<td>Project Manager</td>
</tr>
<tr>
<td></td>
<td>Stockroom, Warehouse, or Storage Yard Stock Clerk</td>
</tr>
<tr>
<td></td>
<td>Technical Writer</td>
</tr>
<tr>
<td></td>
<td>Telecommunications Specialist</td>
</tr>
<tr>
<td></td>
<td>Transportation Manager</td>
</tr>
</tbody>
</table>