This is the second portion of the Database Design and Programming with SQL course. In this portion, students implement their database design by creating a physical database using SQL, the industry-standard database programming language. Upon completion of this course, students have the opportunity to sit for the first of two exams required to earn the Oracle Certified Associate. The course outline below includes section titles and descriptions, lesson titles, and detailed objectives for each lesson.

Section 1 Using Character, Number, and Date Functions
SQL has many functions used to manipulate data. This section continues the discussion of functions and focuses on those that are the most common. Functions are a very powerful feature of SQL and can be used to do the following: perform calculations on data, modify individual data items, manipulate output for groups of rows, format dates and numbers for display, and convert column data types. This section also introduces substitution variables, and how these are used in Oracle Application Express.

Lesson 1: Case and Character Manipulation
• Select and apply single-row functions that perform case conversion and/or character manipulation
• Select and apply character case-manipulation functions LOWER, UPPER, and INITCAP in a SQL query
• Select and apply character-manipulation functions CONCAT, SUBSTR, LENGTH, INSTR, LPAD, RPAD, TRIM, and REPLACE in a SQL query
• Write flexible queries using substitution variables

Lesson 2: Number Functions
• Select and apply the single-row number functions ROUND, TRUNC, and MOD in a SQL query
• Distinguish between the results obtained when TRUNC is applied to a numeric value and ROUND is applied to a numeric value
• State the implications for business when applying TRUNC and ROUND to numeric values

Lesson 3: Date Functions
• Select and apply the single-row functions MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY, LAST_DAY, ROUND, and TRUNC that operate on date data
• Explain how date functions transform Oracle dates into date data or a numeric value
• Demonstrate proper use of the arithmetic operators with dates
• Demonstrate the use of SYSDATE and date functions
• State the implications for world businesses to be able to easily manipulate data stored in date format

Section 2 Using Single Row Functions
The single-row functions described in this section focus on several important topics including creating queries that require the use of numeric, character, and date functions; using a set of general functions that work with any data type and pertain to using null value; converting column data types; and applying the CASE and DECODE methods to implement conditional logic.

Lesson 1: Conversion Functions
• Provide an example of an explicit data-type conversion and an implicit data-type conversion
• Explain why it is important, from a business perspective, for a language to have built-in data-conversion capabilities
• Construct a SQL query that correctly applies TO_CHAR, TO_NUMBER, and TO_DATE single-row functions to produce a desired result
• Apply the appropriate date and/or character format model to produce a desired output
- Explain and apply the use of YYYY and RRRR to return the correct year as stored in the database

Lesson 2: NULL Functions
- Demonstrate and explain the evaluation of a nested function
- List at least four general functions that work with any data type and relate to handling null values
- Explain the use of the COALESCE and the NVL functions
- Explain the use of general functions to deal with null values in data
- Construct and execute a SQL query that correctly applies NVL, NVL2, NULLIF, and COALESCE single-row functions

Lesson 3: Conditional Expressions
- Compare and contrast the DECODE and CASE functions
- Construct and execute a SQL query that correctly uses the DECODE and CASE functions
- Construct and execute two methods for implementing IF-THEN-ELSE conditional logic

Section 3 Executing Database Joins
The SQL statements taught to this point in the course are limited to querying and returning information from one table at a time. Separating data into individual tables and being able to associate the tables with one another is the heart of relational database design. This section introduces the ANSI/ISO SQL 99 join, the syntax that allows separate tables to be combined in one query.

Lesson 1: Cross Joins and Natural Joins
- Construct and execute a natural join using ANSI-99 SQL join syntax
- Create a cross join using ANSI-99 SQL join syntax
- Define the relationship between a cross join and a Cartesian product
- Define the relationship between a natural join and an equijoin
- Explain why it is important to have a standard for SQL as defined by ANSI
- Describe a business need for combining information from multiple data sources

Lesson 2: Join Clauses
- Construct and execute a join with the ANSI-99 USING and ON clauses
- Construct and execute an ANSI-99 query that joins three tables

Lesson 3: Inner Versus Outer Joins
- Compare and contrast an inner and an outer join
- Construct and execute a query to use a left outer join
- Construct and execute a query to use a right outer join
- Construct and execute a query to use a full outer join

Lesson 4: Self Joins and Hierarchical Queries
- Construct and execute a SELECT statement to join a table to itself using a self-join
- Interpret the concept of a hierarchical query
- Create a tree-structured report
- Format hierarchical data
- Exclude branches from a tree structure

Section 4 Working With Group Functions
While single-row functions determine how data is displayed, group functions go one step further by allowing logical operations on sets of rows to give one result per group. Such operations include: finding the average, maximum, minimum, sum, statistical values, counting the number of rows, finding DISTINCT non-duplicate values, and using NVL to force group functions to include NULL values.

Lesson 1: Review of Joins
• Determine the correct join syntax to use given a scenario requiring the join of data from two or more tables

Lesson 2: Group Functions
• Define and give an example of the seven group functions: SUM, AVG, COUNT, MIN, MAX, STDDEV, VARIANCE
• Construct and execute a SQL query using group functions
• Construct and execute group functions that operate only with numeric data types

Lesson 3: COUNT, DISTINCT, NVL
• Construct and execute a SQL query using the COUNT group function
• Use DISTINCT and the NVL function with group functions

Section 5 Using Complex SQL with Aggregated Data
Section 5 focuses on more complex SQL statements. The GROUP BY and HAVING clauses enable users to apply functions to and restrict data returned from aggregate data. GROUP BY ROLLUP and GROUP BY CUBE and GROUPING SETS are extensions to the standard GROUP BY clause, which add subtotals and grand totals to queries returning aggregated data. The GROUPING function, which indicates if a row value is a result of a ROLLUP or CUBE operation on the base column is also introduced in this Section. The SET operators, UNION, UNION ALL, MINUS, and INTERSECT are used to ensure the rows are returned as required in SET Operator queries.

Lesson 1: Using GROUP BY and HAVING clauses
• Construct and execute a SQL query using GROUP BY
• Construct and execute a SQL query using GROUP BY … HAVING
• Construct and execute a GROUP BY on more than one column
• Nest group functions

Lesson 2: Using ROLLUP and CUBE Operations, and GROUPING SETS
• Use ROLLUP to produce subtotal values
• Use CUBE to produce cross-tabulation values
• Use GROUPING SETS to produce a single result set
• Use the GROUPING function to identify the extra row values created by either a ROLLUP or CUBE operation

Lesson 3: Using SET Operators
• Define and explain the purpose of SET operators
• Use a SET operator to combine multiple queries into a single query
• Use SET operator to control the order of rows returned

Section 6 Creating Subqueries
Subqueries are queries written within queries. Students learn that being able to combine two queries into one can be very useful when you need to select rows from a table with a condition that depends on the data in the table itself. This section also covers the more advanced features of the SELECT statement, specifically, single-row subqueries and multiple-row subqueries the ability to utilize a subquery or a "query within a query" is the reason that the word "structured" appears in SQL or Structured Query Language. This section covers the difference between pair-wise and non-pair-wise multi-column subqueries, non-correlated subqueries as well as correlated subqueries. It shows how queries can use the EXIST and NOT EXIST clauses to test for a result from a subquery. It introduces the WITH-clause, which allows the definition of named subqueries to be used in a complex SELECT statement.

Lesson 1: Fundamentals of Subqueries
• Define and explain the purpose of subqueries for retrieving data
• Construct and execute a single-row subquery in the WHERE clause
- Distinguish between single-row and multiple-row subqueries
- Distinguish between pair-wise and non-pair-wise subqueries

Lesson 2: Single Row Subqueries
- Construct and execute a single-row subquery in the WHERE clause or HAVING clause
- Construct and execute a SELECT statement using more than one subquery
- Construct and execute a SELECT statement using a group function in the subquery

Lesson 3: Multiple-row Subqueries
- Use the comparison operators IN, ANY, and ALL correctly in multiple-row subqueries
- Construct and execute a multiple-row subquery in the WHERE clause or HAVING clause
- Describe what happens if a multiple-row subquery returns a null value
- Understand when multiple-row subqueries should be used and when it is safe to use a single-row subquery
- Create a subquery using the EXIST and NOT EXISTS operators to test for returned rows from the subquery

Lesson 4: Correlated Subqueries
- Identify when correlated subqueries are needed
- Construct and execute correlated subqueries
- Construct and execute named subqueries using the WITH clause

Section 7 Constructing DML Statements
Data Manipulation Language (DML) is a core part of SQL. A DML statement is used to add, update, delete, or merge data in the database. A collection of DML statements forms a logical unit of work called a "transaction."

Lesson 1: Insert Statements
- Give examples of why it is important to be able to alter the data in a database
- Construct and execute INSERT statements which insert a single row using a VALUES clause
- Construct and execute INSERT statements that use special values, null values, and date values
- Construct and execute INSERT statements that copy rows from one table to another using a subquery

Lesson 2: Updating Column Values and Deleting Rows
- Construct and execute an UPDATE statement
- Construct and execute a DELETE statement
- Construct and execute a query that uses a subquery to update and delete data from a table
- Construct and execute a query that uses a correlated subquery to update and delete from a table
- Explain how foreign-key and primary-key integrity constraints affect UPDATE and DELETE statements

Lesson 3: DEFAULT Values, MERGE, and Multi-Table Inserts
- Understand when to specify a DEFAULT value
- Construct and execute a MERGE statement
- Construct and execute DML statements using subqueries
- Construct and execute multi-table inserts

Section 8 Working With DDL Statements
The Data Definition Language (DDL), used to create, alter, remove, and drop tables, is covered in this section. DDL statements are used to change the name, remove rows, and add comments to a table. Students will learn about the main database type objects supported by Oracle, they will review a table structure, and they will examine how schema objects are stored and used by the database. Data types include character
data (CHAR AND VARCHAR2), numbers, dates, binary data (BLOB) that is appropriate for storing images, and large character data (CLOB) for storing large character data files such as a resume. Additional data types include LONG, RAW, LONG RAW, and ROWID. This section describes several date and time enhancements for TIMESTAMP. Time data can be stored in fractional_seconds_precision, relative to the local time zone and in interval year-to-month formats. The concept of EXTERNAL TABLES is also introduced. Students are also shown how to perform FLASHBACK table and how to issue FLASHBACK QUERY statements. Since table data can change over time students will use the FLASHBACK QUERY to see the different values before and after updates.

Lesson 1: Creating Tables
- List and categorize the main database objects
- Review a table structure
- Describe how database schema objects are used by the Oracle database
- List and provide an example of each of the number, character, and date data types
- Create a table using the appropriate data type for each column
- Explain the use of external tables
- Query the Data Dictionary to obtain the names and other attributes of database objects

Lesson 2: Using Data Types
- Create a table using TIMESTAMP and TIMESTAMP WITH TIME ZONE column data types
- Create a table using INTERVAL YEAR TO MONTH and INTERVAL DAY TO SECOND column data types
- Give examples of organizations and personal situations where it is important to know to which time zone a date-time value refers

Lesson 3: Modifying a Table
- Explain why it is important to be able to modify a table
- Explain and provide an example for each of the DDL statements ALTER, DROP, RENAME, and TRUNCATE, and the effect each has on tables and columns
- Construct a query and execute the ALTER TABLE commands ADD, MODIFY, and DROP
- Explain and perform FLASHBACK QUERY on a table
- Explain and perform FLASHBACK table operations
- Track the changes to data over a period of time
- Explain the rationale for using TRUNCATE versus DELETE for tables
- Add a comment to a table using the COMMENT ON TABLE command
- Name the changes that can and cannot be made to modify a column
- Explain when and why the SET UNUSED statement is advantageous

Section 9 Ensuring Quality Query Results
This section will highlight methods used to validate effective query results. Students will learn to use validation methods when creating or correcting queries based on expected data results.

Lesson 1: Ensuring Quality Query Results
- Create a query to produce specified data
- Modify a query to produce specified data

Section 10 Creating and Managing Constraints
Constraints enforce rules at the table level to prevent invalid data from entering a table and/or to prevent deletion if there are dependencies with other tables. The five data integrity constraints discussed in this section include: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, and CHECK. Other than the NULL constraint, all other constraints can be defined at either the column level or the table level. All constraints can be created when table columns are defined or added after creation of a table. Students will learn which constraints are created as part of creating a table and why these constraints are created automatically by the database. Database objects created by a user become part of the user's schema and are
stored in the data dictionary. A user can query the data dictionary to view various database objects in their schema.

Lesson 1: Defining NOT NULL and UNIQUE Constraints
- Define the term "constraint" as it relates to data integrity
- State when it is possible to define a constraint at the column level and when it is possible at the table level
- State why it is important to give meaningful names to constraints
- State which data integrity rules are enforced by NOT NULL and UNIQUE constraints
- Write a CREATE TABLE statement which includes NOT NULL and UNIQUE constraints at the table and column levels
- Explain how constraints are created at the time of table creation

Lesson 2: PRIMARY KEY, FOREIGN KEY, and CHECK Constraints
- Define and give an example of a PRIMARY KEY, FOREIGN KEY, and CHECK constraint
- Explain the purpose of defining PRIMARY KEY, FOREIGN KEY, and CHECK constraints
- Demonstrate the creation of constraints at the column level and table level in a CREATE TABLE statement
- Evaluate a business problem requiring the addition of a PRIMARY KEY and FOREIGN KEY constraint and write the code to execute the change
- Query the data dictionary for USER_CONSTRAINTS and interpret the information returned

Lesson 3: Managing Constraints
- List four different functions that the ALTER statement can perform on constraints
- Write ALTER TABLE statements to add, drop, disable, and enable constraints
- Name a business function that would require a DBA to drop, enable and/or disable a constraint, or use the CASCADE syntax
- Query the data dictionary for USER_CONSTRAINTS and interpret the information returned

Section 11 Creating and Managing Views
Views are used to create virtual tables from existing base tables or from other views. Although a view contains no data of its own, it can be used to restrict data access and query information from multiple tables without the user knowing how to write complicated SQL statements. Views also provide groups of users access to data according to their particular function. Simple views are based on one table, contain no group functions, and allow DML operations. Complex Views often do NOT allow DML operations, specifically if they contain a GROUP BY clause, group functions, or the DISTINCT or ROWNUM keywords.

Lesson 1: Creating Views
- List three uses for views from the standpoint of a database administrator
- Explain, from a business perspective, why it is important to be able to create and use logical subsets of data derived from one or more tables
- Create a view with and without column aliases in the subquery using a single base table
- Create a complex view that contains group functions to display values from two tables
- Retrieve data from a view

Lesson 2: DML Operations and Views
- Write and execute a query that performs DML operations on a simple view
- Name the conditions that restrict your ability to modify a view using DML operations
- Write and execute a query using the WITH CHECK OPTION clause
- Explain the use of WITH CHECK OPTION as it applies to integrity constraints and data validation
- Apply the WITH READ ONLY option to a view to restrict DML operations

Lesson 3: Managing Views
• Create and execute a SQL statement that removes a view
• Create and execute a query to create an inline view
• Create and execute a top-n-analysis query

Section 12 Working With Sequences
Along with tables and views, three other objects play an important role in the management of a database, sequences, indexes, and synonyms. Sequences generate unique values, and are used to create integers such as those required for primary keys. Indexes provide direct and fast access to rows in a table. Indexes reduce the amount of disk I/O (input - output) by using an indexed path to locate data quickly. Indexes can be created on the actual base column value, or it can be on the result of apply a function to the column value. Synonyms provide an alternative name for a table, view, sequence, procedure, or other object and are created to shorten lengthy object names and to eliminate the need to qualify an object name with the schema. Synonyms can be both private to one user or it can be public to allow accessibility to all users of the database.

Lesson 1: Working with Sequences
• List at least three useful characteristics of a sequence
• Write and execute a SQL statement that creates a sequence
• Query the data dictionary using USER_SEQUENCES to confirm a sequence definition
• Apply the rules for using NEXTVAL to generate sequential unique numbers in a table
• List the advantages and disadvantages of caching sequence values
• Name three reasons why gaps can occur in a sequence

Lesson 2: Indexes and Synonyms
• Define an index and its use as a schema object
• Define ROWID and its use in locating information in a database
• Name the conditions that cause an index to be created automatically
• Create and execute a CREATE INDEX and DROP INDEX statement
• Create and execute a function-based index
• Create a private and public synonym

Section 13 Fundamentals of Database Security
Controlling user access to ensure database security can be classified into two categories: system security and data security. System security covers access and use of the database at the system level, such as creating usernames and passwords, allocating disk space to users, and granting the system privileges that users can perform. Students will be introduced to database roles and database privileges and they will learn the differences between these and when to choose the appropriate option. Data security is controlled at the object level and each type of object has a particular set of privileges that can be granted to it. This section explains basic system privileges and distinguishes between system and user system privileges. It also discusses basic object privileges. This section also introduces students to REGULAR EXPRESSIONS, which is a mechanism widely used in the computing industry for performing simple and complex pattern matching and replacing of characters in pieces of text. Students will learn how to include regular expressions in check constraint declarations to check for more complex conditions that would otherwise be impossible if the check constraint had been written using only SQL.

Lesson 1: Controlling User Access
• Compare the difference between object privileges and system privileges
• Construct the two commands required to enable a user to have access to a database
• Construct and execute a GRANT… ON … TO statement to assign privileges to objects in their schema to other users and/or PUBLIC
• Query the data dictionary to confirm privileges granted

Lesson 2: Creating and Revoking Object Privileges
• Explain what a ROLE is and what its advantages are.
• Construct a statement to create a ROLE and GRANT privileges to it
• Construct a GRANT ON TO WITH GRANT OPTION statement to assign privileges to objects in their schema to other users and/or PUBLIC
• Construct and execute a statement to REVOKE object privileges from other users and/or from PUBLIC
• Distinguish between privileges and roles
• Explain the purpose of a database link

Lesson 3: Regular Expressions
• Describe regular expressions
• Use regular expressions to search, match, and replace strings in SQL statements
• Construct and execute regular expressions and check constraints

Section 14 Understanding Database Transactions
This section provides an opportunity for students to gain an understanding of the various methods for altering database transactions and the reasons a business would want to control transaction processing.

Lesson 1: Database Transactions
• Define the terms COMMIT, ROLLBACK, and SAVEPOINT as they relate to data transactions
• List three advantages of the COMMIT, ROLLBACK, and SAVEPOINT statements
• Explain why it is important, from a business perspective, to be able to control the flow of transaction processing

Section 15 Oracle Proprietary Join Syntax
This section introduces the students to the Oracle Proprietary Join syntax. It is important for them to be familiar with this syntax, as this is still used widely in industries worldwide.

Lesson 1: Cartesian Product and the Join Operations
• Name the Oracle proprietary joins and their ANSI/ISO SQL: 1999 counterparts
• Describe the purpose of join conditions
• Construct and execute a SELECT statement that results in a Cartesian product
• Construct and execute SELECT statements to access data from more than one table using an equijoin
• Construct and execute SELECT statements that add search conditions using the AND operator
• Apply the rule for using column aliases in a join statement

Lesson 2: Nonequijoins
• Construct and execute a SELECT statement to access data from more than one table using a nonequijoin

Lesson 3: Outer Joins
• Create and execute a SELECT statement to access data from more than one table using an outer join

Section 16 Amazing Books Final Project
The Amazing Books Final Project provides an opportunity for students to apply the concepts they have learned in both the Database Design and Database Programming with SQL courses to create a database application using the Oracle Application Express tool. This tool has many features and functions that can be used to create a sophisticated application. Students are guided through the process of creating a database for a small business by creating tables, layouts, pages, list of values, and forms. Then students input data into the tables and create additional forms and reports.

Lesson 1: Testing
• Develop and apply a strategy for testing that a database functions as designed

Lesson 2: Final Project Database Creation
• Apply SQL concepts to create a functional database appropriate for a small business

Lesson 3: Final Exam Review
• Review the key points about case and character manipulation
• Review number, date, conversion, and general functions
• Review conditional expressions
• Review Cartesian product and join operations
• Review nonequijoins, outer joins, self joins, cross joins, natural joins, and join clauses
• Review group functions, group by syntax and having clauses
• Review single-row and multiple-row subqueries
• Review pair-wise and non-pair-wise subqueries
• Review correlated subqueries
• Review DML statements, insert, update, delete, merge and multi-table inserts
• Review DDL statements, FLASHBACK TABLE, DROP and FLASHBACK QUERY

Section 17 Ensuring Quality Query Results – Advanced Techniques
This section allows students to test the knowledge they have gained in the last two semesters by applying their knowledge to create queries to produced specified output or by testing and modifying queries to produce the desired data output using more advanced SQL capabilities.

Lesson 1: Ensuring Quality Query Results – Advanced Techniques
• Create an advanced query to produce specified data
• Modify an advanced query to produce the specified data

Database Programming Appendix A: Finding a Job
This section contains a number of self-study lessons. These lessons are designed to help students through the process of searching for a job. This process includes examining the current job market and identifying job trends, identifying employers whose goals mesh with yours, researching employers of interest, writing resumes and cover letters, initiating contact through networking, applying to advertised positions, and securing interviews. The section concludes with a discussion about growing and maintaining a business network.

Lesson 1: The Changing Nature of the Job Market
• Identify factors that contribute to the changing nature of the job market
• Identify five jobs or job areas that are on the decline and explain why
• Identify five jobs or job areas that are growing and explain why
• Articulate the changing nature of work and its associated educational requirements
• Evaluate career choices in light of the changing nature of work

Lesson 2: Searching for a Job
• List five techniques for searching for a job
• Identify the two most effective techniques for searching for a job
• Search job-listing databases on the Internet
• Begin to develop a personal network

Lesson 3: Written Communication
• Describe the purpose of and write a cover letter for a sample job opportunity
• Describe the purpose of and write a follow-up letter for a sample job opportunity
• Describe the purpose of a letter of recommendation and list the steps to obtain one

Lesson 4: Interviewing
• Differentiate between a traditional interview and a behavioral interview
• Demonstrate appropriate behavior and dress for an interview
• Prepare for and participate in a mock interview for a sample job opportunity
• Demonstrate the ability to ask appropriate questions during a mock interview
• Evaluate your personal skills as an interviewer and interviewee

Lesson 5: Networking
• Define the term networking
• Describe how to grow and maintain your network
• List three benefits of networking
• Describe the purpose of online networking websites and list three ways social networks can break your job search