Database Design and Programming with SQL – Course Objectives

Overview

This course engages students to analyze complex business scenarios and create a data model—a conceptual representation of an organization's information. Participants implement their database design by creating a physical database using SQL. Basic SQL syntax and the rules for constructing valid SQL statements are reviewed. This course culminates with a project that challenges students to design, implement, and demonstrate a database solution for a business or organization.

Available Curriculum Languages:

- English, Simplified Chinese, Brazilian Portuguese, Spanish, Indonesian

Duration

- Recommended total course time: 180 hours*
- Professional education credit hours for educators who complete Oracle Academy training: 60

  Course time includes instruction, self-study/homework, practices, projects, and assessment

Target Audiences

Educators

- College/university faculty who teach computer programming, information communications technology (ICT), or a related subject
- Secondary school teachers who teach computer programming, ICT, or a related subject

Students

- Students who wish to learn the techniques and tools to design, guild and extract information from a database
- Students who possess basic mathematical, logical, and analytical problem-solving skills
- Novice programmers, as well as those at advanced levels, to learning the SQL Programming language to an advanced level

Prerequisites

Required

- Ease with using a computer
- General knowledge of databases and query activity

Suggested

- None

Suggested Next Courses

- Database Programming with PL/SQL

Lesson-by-Lesson Topics and Objectives

Database Design

Section1 – Introduction
- 1-1 Introduction to the Oracle Academy
  - Give examples of jobs, salaries, and opportunities that are possible by participating in the Academy.
  - Explain how your participation in the Academy can help you take advantage of these opportunities.

- 1-2 Data vs. Information
  - Distinguish between data and information, and provide examples of each
  - Describe and give an example of how data becomes information

- 1-3 History of the Database
  - Describe the evolution of the database and give an example of its role in the business world
  - Name important historical contributions in database development and design
  - List and explain the three major steps in the database development process

- 1-4 Major Transformations in Computing
  - List the major transformations in computing that have occurred since the 1970's.
  - Define and give an example of these terms: hardware, operating system, software
  - Identify examples of businesses that use database software and explain how it is essential to their success
  - Explain the overall mission of the Oracle Corporation

Section 2 – Entities and Attributes
- 2-1 Conceptual and Physical Models
  - Explain the importance of clearly communicating and accurately capturing information requirements
  - Distinguish between a conceptual model and its physical implementation
  - List five reasons for building a conceptual data model
  - Give examples of conceptual models and physical models

- 2-2 Entities, Instances, Attributes, and Identifiers
  - Define and give an example of an entity
  - Distinguish between an entity and an instance of an entity
  - Name and describe attributes for a given entity
  - Distinguish between an attribute and its value
  - Distinguish between mandatory and optional attributes, and between volatile and nonvolatile attributes
  - Select and justify a unique identifier (UID) for an entity

- 2-3 Entity Relationship Modeling and ERDs
  - Define the meaning of "implementation-free" as it relates to data models and database design implementation
  - List the four goals of entity relationship modeling
  - Identify an entity relationship diagram (ERD)

Section 3 – Relationship Basics
- 3-1 Identifying Relationships
  - Interpret and describe relationship optionality
  - Interpret and describe relationship cardinality
  - Relate (connect or join) entities by applying the rules of cardinality and optionality

- 3-2 ER Diagramming Conventions
  - Construct ER diagram components that represent entities, attributes and relationships according to diagramming conventions

- 3-3 Speaking ERDish and Drawing Relationships
  - State relationships between entities in precise words (ERDish)
  - Draw and label relationships correctly on an ERD

- 3-4 Matrix Diagrams
  - Identify relationships using a matrix diagram
  - Draw an ERD from a matrix diagram

Section 4 – Super/Sub Types and Business Rules
- 4-1 Supertypes and Subtypes
  - Define and give an example of a subtype
  - Define and give an example of a supertype
  - State the rules relating to entities and subtypes, and give examples of each
  - Apply the rules of supertype and subtype by evaluating the accuracy of ER diagrams that represent them
  - Apply the rules of supertype and subtype and include them in a diagram when appropriate

- 4-2 Documenting Business Rules
  - Define and compose a structural business rule
  - Define and compose a procedural business rule
  - Recognize that some business rules will require programming
  - Diagram business rules when they can be represented in an ER model

Section 5 – Relationship Fundamentals
- 5-1 Relationship Transferability
  - Describe and give an example of relationship transferability
  - Understand the difference between transferable and nontransferable relationships
  - Illustrate nontransferable relationships on ERDs

- 5-2 Relationship Types
  - Recognize and give examples of a one-to-one relationship
  - Recognize and give examples of a one-to-many relationship
  - Recognize and give examples of a many-to-many relationship
  - Recognize redundant relationships and remove them from the ERD

- 5-3 Resolving Many-to-Many Relationships
  - Identify attributes which belong to many-to-many relationships
  - Demonstrate the steps to resolve a many-to-many relationship using an intersection entity
  - Identify the UID of an intersection entity and represent it in the entity relationship diagram

- 5-4 Understanding CRUD Requirements
  - Create ER models that reflect all business rules gathered during the interview process
  - Identify the create, retrieve, update, and delete (CRUD) requirements of the business
  - Validate your ER model by performing a CRUD analysis

Section 6 – UIDs and Normalization
- 6-1 Artificial, Composite, and Secondary UIDs
  - Define the different types of unique identifiers (UIDs)
  - Define a candidate UID and explain why an entity can sometimes have more than one candidate UID
  - Analyze business rules and choose the most suitable primary UID from the candidates
  - Recognize and discuss the issues of identification in the real world

- 6-2 Normalization and First Normal Form
  - Define the purpose of normalization in database models
  - Define the rule of First Normal Form in the normalization process
  - Determine if an entity conforms to the rule of First Normal Form
  - Convert an entity to First Normal Form if needed

- 6-3 Second Normal Form
  - Define the rule of Second Normal Form in the normalization process
  - Examine a non-normalized entity and determine which rule, or rules of normalization are being violated
  - Apply the rule of Second Normal Form to resolve a violation in the model

- 6-4 Third Normal Form
  - Identify transitive dependencies in a data model
  - Define the rule of Third Normal Form in the normalization process
  - Examine a non-normalized entity and determine which rule, or rules of normalization are being violated
  - Apply the rule of Third Normal Form to resolve a violation in the model

Section 7 – Arcs, Hierarchies, and Recursive Modeling
- 7-1 Arcs
  - Define the term "constraint" as it applies to data modeling
  - Identify an exclusive OR relationship in a business scenario
  - Diagram an arc constraint to represent an exclusive OR relationship
  - Distinguish between the use of an arc and a subtype in the data model

- 7-2 Hierarchies and Recursive Relationships
  - Define and give an example of a hierarchical relationship
  - Identify the UIDs in a hierarchical model
  - Define and give an example of a recursive relationship
  - Represent a recursive relationship in an ERD given a scenario
  - Construct a model using both recursion and hierarchies to express the same conceptual meaning

Section 8 – Changes and Historical Modeling
- 8-1 Modeling Historical Data
  - Identify the need to track data that changes over time
  - Construct ERD models that incorporate elements of "data over time"
  - Identify the UID of an entity that stores historical data; explain and justify the choice of UID

- 8-2 Modeling Change: Time
  - Distinguish between using date as an attribute and DAY as an entity in a data model, depending on business requirements
  - Solve the problem of keeping characteristics of a date by constructing a model that uses DAY as an entity
  - Identify at least three time-related constraints that can result from a time-sensitive model
  - Define and give an example of conditional non-transferability in a time-constrained model

- 8-3 Modeling Change: Price
  - Solve the business requirement of tracking changes in price or values by constructing a model that uses a historical entity
  - Describe the meaning of journaling/logging
  - Identify the business need for journaling/logging and construct a model that addresses this requirement

- 8-4 Drawing Conventions for Readability
  - Apply the Oracle drawing conventions to a data model diagram
  - Identify high volume entities in a data model diagram and explain their significance to the business
  - Redraw a given data model diagram to increase clarity and readability
  - Recognize the usefulness of dividing a complex ERD into a number of functional sub-diagrams

Section 9 – Mapping
- 9-1 Introduction to Relational Database Concepts
  - Define a primary key
  - Define a foreign key
  - Define a column-integrity rule
  - Identify row, column, primary key, unique key, and foreign key elements given a diagram of a table containing these elements
  - Identify violations of data-integrity rules

- 9-2 Basic Mapping: The Transformation Process
  - Distinguish between a conceptual model and a physical model
  - Apply terminology mapping between the two models
  - Understand and apply the Oracle naming conventions for tables and columns used in physical models
  - Transform an entity into a table diagram

- 9-3 Relationship Mapping
  - Apply the rule of relationship mapping to correctly transform 1:M and barred relationships
  - Apply the rule of relationship mapping to correctly transform M:M relationships
  - Transform 1:1 relationships
  - Apply the rule of relationship mapping to correctly transform relationships in an arc

- 9-4 Subtype Mapping
    - State and apply the table, column, identifiers, relationship, and integrity constraint rules for mapping:
        - supertype implementations
        - subtype implementations
        - supertype and subtype arc implementations

Section 10 – Creating Database Projects

- 10-1 System Development Life Cycle
    - List and describe the different stages of the system development life cycle (SDLC)
    - Identify the role of data modeling in the system development life cycle
    - Relate the project tasks to the different stages of the system development life cycle

- 10-2 Project Overview and Getting Started
    - Identify a data-modeling project to solve a business information need
    - Demonstrate essential skills in solving business problems using technology, and professionally presenting these solutions
    - Demonstrate effective team-building skills

- 10-3 Presentation Project Management
    - Track responsibility and progress on a project by using a project-management document
    - Show evidence of progress on the final presentation project by filling in the final project tracking grid

- 10-4 Final Presentation Components
    - Organize the final oral presentation content in logical outline order
    - Identify appropriate content for each part of the presentation
    - Assign presentation roles to each group member

Section 11 – Presenting Database Projects

- 11-1 Creating Tables for the Final Presentation
    - Create sample tables from the final project presentation ERD
    - Insert sample data to the created tables

- 11-2 Preparing Written Documentation
    - Compose well-organized written documentation to accompany a presentation
    - Create a rough draft of, review, and finalize the written documentation
    -
- 11-3 Preparing Visual Materials
    - Construct entity relationship diagrams that demonstrate industry conventions
    - Prepare tables and visuals that support their database documentation
    - Demonstrate and provide examples of table formatting for database documentation
    - Demonstrate and refine oral presentation skills during rehearsal and refinement

- 11-4 Final Presentations
    - Demonstrate the use of a central message and supporting arguments for a final presentation
    - Demonstrate logical analysis of the business rules, operations, and processes in a way that is clear and easy for the client to understand
    - Demonstrate the selection of appropriate business attire for a final presentation

Database Programming with SQL

Section 1 – Introduction
- 1-1 Oracle Application Express
  - Distinguish between application software and system software and give an example of each
  - Log-in to the Oracle Application Express practice environment
  - Execute a simple query to retrieve information from the Database
  - Apply the rules of SQL to display all columns and a subset of columns specified by criteria

- 1-2 Relational Database Technology
  - Define and give an example of a relational database
  - Identify table-key terms, including row, column, field, primary key, and foreign key
  - Relate the importance of databases to everyday life

- 1-3 Anatomy of a SQL Statement
  - Match projection and selection with their correct capabilities
  - Create a basic SELECT statement
  - Use the correct syntax to display all rows in a table
  - Use the correct syntax to select specific columns in a table, modify the way data is displayed, and perform calculations using arithmetic expressions and operators

Section 2 – SELECT and WHERE
- 2-1 Columns, Characters, and Rows
  - Apply the concatenation operator to link columns to other columns, arithmetic expressions, or constant values to create a character expression
  - Use column aliases to rename columns in the query result
  - Enter literal values of type character, number, or date into a SELECT statement
  - Define and use DISTINCT to eliminate duplicate rows
  - Edit, execute, and save SQL statements in Oracle Application Express

- 2-2 Limit Rows Selected
  - Apply SQL syntax to restrict the rows returned from a query
  - Demonstrate application of the WHERE clause syntax
  - Explain why it is important, from a business perspective, to be able to easily limit data retrieved from a table
  - Construct and produce output using a SQL query containing character strings and date values

- 2-3 Comparison Operators
  - Apply the proper comparison operator to return a desired result
  - Demonstrate proper use of BETWEEN, IN, and LIKE conditions to return a desired result
  - Distinguish between zero and NULL, the latter of which is unavailable, unassigned, unknown, or inapplicable
  - Explain the use of comparison conditions and NULL

Section 3 – WHERE, ORDER BY, and Intro to Functions
- 3-1 Logical Comparisons and Precedence Rules
  - Evaluate logical comparisons to restrict the rows returned based on two or more conditions
  - Apply the rules of precedence to determine the order in which expressions are evaluated and calculated

- 3-2 Sorting Rows
  - Construct a query to sort a result set in ascending or descending order
  - State the order in which expressions are evaluated and calculated based on the rules of precedence
  - Construct a query to order a result set using a column alias
  - Construct a query to order a result set for single or multiple columns

- 3-3 Introduction to Functions
  - Identify appropriate applications of single-row functions in query statements
  - Classify a function as a single-row or multi-row function
  - Differentiate between single-row functions and multi-row functions and the results returned by each

Section 4 – Single Row Functions Part I
- 4-1 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

- 4-2 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

- 4-3 Date Functions
  - 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 5 – Single Row Functions Part II
- 5-1 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 YY and RR to return the correct year as stored in the database

- 5-2 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

- 5-3 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 6 – JOINs Part I
- 6-1 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
  - Explain the importance of having a standard for SQL as defined by ANSI
  - Describe a business need for combining information from multiple data sources

- 6-2 Join Clauses
  - Construct and execute a join with the ANSI-99 USING Clause
  - Construct and execute a join with the ANSI-99 ON Clause
  - Construct and execute an ANSI-99 query that joins three tables

- 6-3 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

- 6-4 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 the tree structure

Section 7 – JOINs Part II
- 7-1 Oracle Equijoin and Cartesian Product
  - Name the Oracle proprietary joins and their ANSI/ISO SQL: 99 counterparts
  - 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 table aliases in a join statement

- 7-2 Oracle Nonequijoins and Outer Joins
  - Construct and execute a SELECT statement to access data from more than one table using a nonequijoin
  - Create and execute a SELECT statement to access data from more than one table using an Oracle outer join

Section 8 – Group Functions Part I
- 8-1 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

- 8-2 COUNT, DISTINCT, NVL
  - Construct and execute a SQL query using the COUNT group function
  - Use DISTINCT and the NVL function with group functions

Section 9 – Group Functions Part II
- 9-1 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

- 9-2 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

- 9-3 Using Set Operators
  - Define and explain the purpose of Set Operators
  - Use a set operator to combine multiple queries into a single query
  - Control the order of rows returned using set operators

Section 10 – Subqueries
- 10-1 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

- 10-2 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

- 10-3 Multiple-Row Subqueries
  - Correctly use the comparison operators IN, ANY, and ALL 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
  - Distinguish between pair-wise and non-pair-wise subqueries

- 10-4 Correlated Subqueries
  - Identify when correlated subqueries are needed.
  - Construct and execute correlated subqueries.
  - Create a query using the EXISTS and NOT EXISTS operators to test for returned rows from the subquery
  - Construct and execute named subqueries using the WITH clause.

Section 11 – Ensuring Quality Queries Part I
- 11-1 Ensuring Quality Query Results
  - Create a query to produce specified data
  - Modify a query to produce specified data

Section 12 – DML
- 12-1 INSERT Statements
  - Explain the importance of being 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

- 12-2 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
  - Explain the purpose of the FOR UPDATE Clause in a SELECT statement

- 12-3 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 13 – DDL
- 13-1 Creating Tables
  - List and categorize the main database objects
  - Review a table structure
  - Describe how schema objects are used by the Oracle database
  - 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

- 13-2 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
  - List and provide an example of each of the number, date, and character data types

- 13-3 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 14 – Constraints
- 14-1 Intro to Constraints; 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

- 14-2 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

- 14-3 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 15 – Views
- 15-1 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

- 15-2 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

- 15-3 Managing Views
  - Create and execute a SQL statement that removes a view
  - Create and execute a query using an inline view
  - Create and execute a top-n-analysis query

Section 16 – Sequences and Synonyms
- 16-1 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

- 16-2 Indexes and Synonyms
  - Define an index and its use as a schema object
  - 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 private and public synonyms

Section 17 – Privileges and Regular Expressions
- 17-1 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

- 17-2 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 on objects in your 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

- 17-3 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 18 – TCL
- 18-1 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 19 – Final Project and Exam Review
- 19-1 Testing
  - Develop and apply a strategy for testing that a database functions as designed

- 19-2 Final Project Database Creation
  - Apply SQL concepts to create a functional database appropriate for a small business

- 19-3 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 non-equijoins, 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 CREATE, ALTER, RENAME, TRUNCATE, FLASHBACK TABLE, DROP, and FLASHBACK QUERY
  - Review DCL statements CREATE and REVOKE object privileges

Section 20 – Ensuring Quality Queries Part II
- 20-1 Ensuring Quality Query Results - Advanced Techniques
  - Create an advanced query to produce specified data
  - Modify an advanced query to produce specified data

To search and register for events scheduled in your area, visit the Academy events calendar.