# ORACLE DATABASE 12C: INTRODUCTION TO SQL

Course Code: 7625

This learning path offers an introduction to the Oracle Database 12c database technology, including how to write queries.

This course offers an introduction to the Oracle Database 12c database technology. It covers the concepts of relational databases and the powerful SQL programming language. It also demonstrates essential SQL skills for writing queries against single and multiple tables, manipulating data in tables, and creating database objects.

## What You'll Learn

- Oracle Database 12c  Install and Upgrade: Database Architecture
- Oracle Database 12c  Install and Upgrade: Installation
- Oracle Database 12c  Install and Upgrade: Upgrading
- Oracle Database 12c - Introduction to SQL: Relational Database and SQL Developer
- Oracle Database 12c - Introduction to SQL: SQL*Plus and SELECT Statement
- Oracle Database 12c - Introduction to SQL: Restrict, Sort, and Customize Output
- Oracle Database 12c - Introduction to SQL: Data Conversion and Aggregating Data

## Who Needs to Attend

This course is designed for Database Administrators, support engineers, technical administrators, and Data Warehouse Administrator who wish to learn how to install Oracle Database 12c and upgrade from previous versions.

# ORACLE DATABASE 12C: INTRODUCTION TO SQL

Course Code: 7625

| ON-DEMAND | $945 CAD |
| --- | --- |

## On-Demand Outline

**Oracle Database 12c - Introduction to SQL: Relational Database and SQL Developer**

- Describe the features of Oracle Database 12c
- Describe the salient features of Oracle Cloud 12c
- Describe relational and object relational database management systems
- Define the concepts of a relational database
- Describe the data model
- Describe the Entity Relationship Model
- Use the primary and foreign keys to relate multiple tables
- Recognize the relational database terminology
- Use SQL to query a database and its development environments
- Describe sample schemas in Oracle Database
- List the key features of Oracle SQL Developer
- Create a database connection within SQL Developer
- Browse through the database objects and structures within SQL Developer
- Create a new schema object using SQL Developer
- Use SQL Worksheet to execute SQL and PL/SQL statements and format SQL code
- Work with SQL statements and scripts using SQL Developer
- Execute saved script files using various methods in SQL Developer
- Use snippets in SQL Developer
- Use the Recycle Bin in SQL Developer to recover or purge dropped objects
- Debug procedures and functions
- Display built-in database reports in SQL Developer
- Create a user-defined report in SQL Developer and organize them in folders
- Configure links to external tools and search engines within SQL Developer
- Customize the SQL Developer interface and environment

- Use the Data Modeler in SQL Developer
- Connect to the database and browse the tables in the HR schema using SQL Developer

**Oracle Database 12c - Introduction to SQL: SQL*Plus and SELECT Statement**

- Describe the features of SQL and SQL*Plus
- Describe the SQL*Plus editing commands
- Use the LIST, n, APPEND and CHANGE SQL*Plus editing commands
- Describe the SQL*Plus file commands
- Use the SAVE and START SQL*Plus file commands
- Use the SERVEROUTPUT and SPOOL SQL*Plus commands
- Use the AUTOTRACE SQL*Plus command
- Describe and use the basic SELECT statement
- Recognize the rules and guidelines for constructing valid SQL statements in SQL Developer and SQL*Plus
- Describe and use the arithmetic expressions in SQL
- Apply the standard order of precedence in SQL arithmetic expressions
- Identify how null values are defined in SQL and how they affect arithmetic expression outcomes
- Define column aliases in the SELECT list
- Use the concatenation operator to create a character expression in the SQL statement output
- Use literal character strings in a SELECT statement
- Use the alternative quote (q) operator to select a quotation mark delimiter
- Eliminate duplicate rows from SQL statement output using the DISTINCT qualifier
- Display the structure of a table using the SQL*Plus DESCRIBE command
- Select data from different tables, describe tables, perform arithmetic calculations, and specify column names using aliases

**Oracle Database 12c - Introduction to SQL: Restrict, Sort, and Customize Output**

- Describe the limiting of rows selected in a query and how to use the WHERE clause in a SQL statement
- Use character strings and dates in a WHERE clause in a SQL statement
- Use the comparison operators in a WHERE clause in a SQL statement
- Use the BETWEEN and IN operators in a WHERE clause in a SQL statement
- Use the LIKE operator and combine wildcard characters in a WHERE clause in a SQL statement
- Use the NULL and logical operator conditions in a WHERE clause in a SQL statement
- Use the AND, OR and NOT operators in a WHERE clause in a SQL statement
- Define the rules of precedence in a WHERE clause in a SQL statement
- Use the ORDER BY clause to sort the data returned by a SQL statement
- Use the ROW_LIMITING_CLAUSE to limit the number of rows returned by a SQL statement
- Describe the use of substitution variables in a SQL statement and use a

single-ampersand substitution variable to provide a value in a SQL statement
- Use substitution variables for column names, expressions and text in a SQL statement
- Reuse a substitution variable by making use a double-ampersand substitution variable in a SQL statement
- Use the DEFINE and VERIFY commands to create, assign, and verify a substitution variable when using SQL Developer
- Write a SQL statement using the WHERE clause and pass a substitution variable to the SQL statement
- Identify the types of SQL functions in a SQL statement and describe single-row functions
- Use character functions in a SQL statement
- Use the case-conversion functions in a SQL statement
- Use the character-manipulation functions in a SQL statement
- Use the nesting functions in a SQL statement
- Use the numeric ROUND and TRUNC functions in a SQL statement
- Use the MOD function in a SQL statement
- Work with dates and the RR date format in a SQL statement
- Use SYSDATE, CURRENT_DATE and CURRENT_TIMESTAMP functions in a SQL statement
- Use arithmetic operators with dates in a SQL statement
- Describe and use date-manipulation functions MONTHS_BETWEEN, ADD_MONTHS, NEXT_DAY, LAST_DAY in a SQL statement
- Use the numeric ROUND and TRUNC functions with dates
- Write a SQL statement using the CHARACTER, NUMBER, and DATE functions

**Oracle Database 12c - Introduction to SQL: Data Conversion and Aggregating Data**

- Describe the use of implicit and explicit data conversion within SQL statements
- Use the TO_CHAR function to convert a date to a character data type in a SQL statement
- Describe elements of the Date Format Model in SQL
- Use the TO_CHAR Function to convert a number to a character datatype in a SQL statement
- Use the TO_NUMBER and TO_DATE function to convert a character data type to a number or date format in a SQL statement
- Use the TO_CHAR and TO_DATE functions with the RR date format to return data for a previous century in a SQL statement
- Describe general functions when working with data pertaining to null values
- Use the NVL and NVL2 function when working with null values in a SQL statement
- Use the NULLIF and COALESCE functions when working with null values in a SQL statement
- Use conditional expressions to provide the use of the IF-THEN-ELSE logic in a SQL statement
- Use the CASE expression to facilitate a conditional inquiry in a SQL statement
- Use the DECODE function to facilitate conditional inquiry in a SQL statement

- Describe how to use group functions to return one result based on a set of rows in a SQL statement
- Use the AVG and SUM functions to group numeric data in a SQL statement
- Use the MIN and MAX functions to group numeric, character, and date data types in a SQL statement
- Use the COUNT and COUNT DISTINCT functions in a SQL statement
- Work with null values when grouping a set of rows in a SQL statement
- Create groups of data in a SQL statement
- Describe how to create groups of data and work with the GROUP BY clause in a SQL statement
- Use with the GROUP BY clause on multiple columns in a SQL statement
- Describe illegal queries when using the GROUP BY function in a SQL statement
- Use the HAVING clause to restrict grouped rows returned in a SQL statement
- Use nesting of group functions to a depth of two functions in a SQL statement
- Write SQL statements using the TO_CHAR, TO_DATE, DECODE functions and the CASE Expression
- Write SQL statements using the GROUP BY and HAVING functions

**Oracle Database 12c - Introduction to SQL: Working with Joins and Subqueries**

- Describe how to obtain data from multiple tables using JOINS in a SQL statement
- Join tables automatically based on columns with matching data types and names to retrieve data in a SQL statement
- Use the USING clause to specify columns used for an equijoin to retrieve records in a SQL statement
- Use aliases with the USING clause and use prefixes to join multiple tables in a SQL statement
- Use the ON clause when you specify join conditions in the WHERE clause in a SQL statement
- Using a three-way join to join three tables in a SQL statement
- Use the AND clause or the WHERE clause to apply additional conditions to a join in a SQL statement
- Use a self-join to join a table to itself in a SQL statement
- Use a join condition containing a non-equijoin in a SQL statement
- Use an OUTER join to return records with no direct match in a SQL statement
- Describe the differences between an INNER and OUTER join
- Use a left outer join in a SQL statement
- Use a right outer join in a SQL statement
- Use a full outer join in a SQL statement
- Describe the generation of a Cartesian product returned by a SQL statement
- Create cross joins in a SQL statement
- Use a subquery to solve a problem in a SQL statement
- Describe the rules, guidelines, and types of subqueries used in a SQL statement
- Use a single-row subquery to return one row in a SQL statement
- Display data from a main query using a group function in a subquery to return a single row in a SQL statement

- Use a subquery in the HAVING clause of a SQL statement
- Identify common errors that occur when an incorrect number of rows is returned by a single-row subquery in a SQL statement
- Describe multiple-row subqueries in a SQL statement
- Use the IN, ANY, and ALL operators in multiple-row subqueries in a SQL statement
- Use a multiple-column subquery to return more than one column to the outer query in a SQL statement
- Describe NULL values in a subquery and use the NOT IN operator in a SQL statement
- Display data from multiple tables using joins in a SQL statement
- Combine two queries by making use of a subquery in a SQL statement

**Oracle Database 12c - Introduction to SQL: Data Manipulation Language**

- Define the Data Manipulation Language in SQL
- Add rows to a table in the database by using the INSERT SQL statement
- Work with NULL values during the INSERT SQL statement when adding rows to a table
- Work with SPECIAL values during the INSERT SQL statement when adding rows to a table
- Work with specific DATE and TIME values during the INSERT SQL statement when adding rows to a table
- Save commands with substitution variables to a file and execute the commands in the file
- Use the INSERT SQL statement to copy rows from one table to another
- Change data in a table using the UPDATE SQL statement
- Change data into two columns using a subquery to update the two columns using the UPDATE SQL statement
- Change the rows in a table based on a different table using the UPDATE SQL statement
- Remove rows from a table using the DELETE SQL statement
- Remove rows from a table based on a different table using the DELETE SQL statement
- Use the TRUNCATE SQL statement to delete all rows from a table
- Insert and update rows in a table
- Describe database transactions and how to ensure data consistency during DML, DDL and DCL activity on the database
- Describe the COMMIT and ROLLBACK SQL statements
- Describe the control of the logic within transactions
- Use the ROLLBACK SQL command to undo changes to a defined marker
- Describe the conditions in which an automatic COMMIT statement is issued and the impact on the database
- Describe the state of the database before and after a COMMIT SQL command within the database
- Describe the state of the database before a ROLLBACK SQL command within the database

- Describe read consistency and its role in the database
- Describe using the FOR UPDATE clause in a SQL statement
- Process a transaction using explicit transaction control statements

**Oracle Database 12c - Introduction to SQL: Working with SET Operators and DDL**

- Describe the types of SET operators and their associated guidelines
- Recognize how the Oracle Server manages SET operators
- Use UNION and UNION ALL operators returns all rows in a SQL statement
- Use the INTERSECT operator to return all rows that are common to multiple queries in a SQL statement
- Use the MINUS operator to return all distinct rows selected by the first query in a SQL statement
- Describe how the expressions in the SELECT lists of the queries must match in number in a SQL statement
- Use the ORDER BY clause in set operations in a SQL statement
- Write reports requested by the HR department using the SET operators
- Describe the DDL database objects and the associated naming rules
- Use the CREATE TABLE SQL command to create new tables in the database
- Identify the different Oracle database data types used when creating a table
- Define a default value to be assigned to a column when creating a table
- Define a default value to be assigned to a column when creating a table
- Describe the syntax for defining constraints when creating a table
- Describe the use of the NOT NULL and PRIMARY KEY constraints when creating a table
- Describe the use of the UNIQUE constraint when creating a table
- Describe the use of the CHECK constraint
- Describe the use of the CHECK constraint
- Describe the conditions when a constraint is violated
- Create a new table and insert rows by combining the CREATE TABLE and the AS subquery option in a SQL statement
- Change the definition of a table using the ALTER TABLE SQL command
- Change the definition of a table by adding, modifying, and dropping columns in a table using the ALTER TABLE SQL command
- Use the SET UNUSED option to mark columns as unused using the ALTER TABLE SQL command
- Use the ALTER TABLE SQL command to set a table to read only
- Use the DROP TABLE SQL command to delete a table from the database
- Work with tables using DDL

Visit us at www.globalknowledge.com or call us at 1-866-716-6688.