



Oracle Database 19c: Advanced SQL

Course completed by Siddharth Bahekar
Feb 27, 2024 at 07:45PM UTC • 2 hours 3 minutes

Top skills covered

SQL

Oracle Database

A stylized, handwritten signature in black ink that reads "Dar Brodnitz".

Head of Content Strategy, Learning



Certificate ID: 1a1d70d4b2e9bbba1d8578a77b3696ad7a69a192a17013b1ddf238c8318dc937

W10-LinkedIn Advance SQL Certification – Siddharth Bahekar – **0024147718**

I did the basic certification in Oracle 12c and this assignment we were asked to do advanced Oracle 19c. The first question that came to my mind was the version change and similarly the first topic in the certification course was the same. Version 19c has some upgrades such as multitenant architecture, security, better cloud integration and automatic indexing.

Chapter

1

-

Single row functions in SQL are used to manipulate data within individual rows of a table. They include basic operations like changing the case of text or performing mathematical calculations on numerical values. Multi-row functions, on the other hand, operate on multiple rows of data at once, often used for more complex analysis or manipulation tasks. Hierarchical functions help in querying hierarchical data structures like trees or graphs, allowing traversal from parent to child nodes. Null handling functions assist in managing null values in data, providing options for handling them effectively within queries. Analytic functions enable advanced analytical processing, allowing calculations to be performed across multiple rows while maintaining individual row integrity. Grouping in a window involves partitioning data into groups for analysis within specific windows or ranges. SQL macro table functions enable the creation of custom functions that can be used within SQL queries to simplify complex operations or encapsulate reusable logic.

Chapter 2 -

Set operators are powerful tools used to combine the results of multiple queries into a single result set. The main set operators include Union, Union All, Intersect, and Minus. Union combines the results of two queries, removing duplicates. Union All retains all rows, including duplicates. Intersect returns only the rows that are common to both query results, Minus returns rows from the first query that are not present in the second query. Using the Order By clause with set operators allows for sorting the final result set according to specified criteria. These set operators provide many options for manipulating and combining data from multiple queries, enabling users to perform complex data analysis and retrieval tasks efficiently.

Chapter 3 -

Data Manipulation Language (DML) statements are used to manage data in the database. The MERGE statement is a powerful tool that combines INSERT, UPDATE, and DELETE operations into a single statement, making it efficient for handling data. It starts with an overview of DML statements, showing their role in manipulating data. The MERGE statement begins with the INTO clause, specifying the target table where data will be merged. The USING clause defines the source of data, usually from another table or a subquery. Then, based on specified conditions, the statement performs different actions: when a match is found between the source and target data, the WHEN MATCHED...UPDATE clause updates the target rows; if no match is found, the WHEN NOT MATCHED...INSERT clause inserts new rows into the target table. Additionally, the DELETE clause can be used to remove rows from the target table based on certain conditions. Overall, the MERGE statement simplifies complex data synchronization tasks by combining multiple operations into one.

Chapter 4 -

In Oracle SQL, Data Definition Language (DDL) commands are used to manage the structure of the database, defining and modifying its schema. There are different types of DDL commands, including commands for creating, altering, and dropping database objects such as tables, views, indexes, and sequences. DDL commands enable users to define the structure of the database, specifying the data types, constraints, and relationships between objects. Transactions involving DDL commands are important as they ensure data consistency and integrity during schema modifications. Unlike Data Manipulation Language (DML) commands, DDL commands are auto-committed, meaning they cannot be rolled back within a transaction.

Chapter 5 -

In Oracle SQL, Data Manipulation Language (DML) Transaction Processing involves managing changes to data within the database in a controlled and consistent manner. When a COMMIT command is issued, it permanently saves the changes made in the transaction to the database. On the other hand, ROLLBACK undoes all the changes made in the current transaction, reverting the database to its previous state. SAVEPOINTS offer a middle ground by allowing users to set points within a transaction where they can later choose to rollback to if needed, preserving changes made

up to that point while undoing frequent modifications. These commands ensure data integrity and provide safety for managing transactions effectively, allowing users to control when changes are applied and ensuring that the database remains in a consistent state.

Chapter 6 -

In Oracle SQL, Data Definition Language (DDL) commands are used to manage the structure of the database, including creating and manipulating objects within it. Three key aspects of DDL involve truncating tables, managing synonyms, and handling sequences. Truncating tables removes all rows from a table quickly and efficiently, resetting any associated sequences. Synonyms provide an alternative name for database objects, simplifying access and enhancing security by masking the underlying object's details. Sequences generate unique numeric values, typically used for primary key generation, ensuring data integrity and consistency across multiple transactions.

Chapter 7 -

Row-level constraints are rules applied to individual rows within a database table. These constraints ensure data integrity by defining conditions that must be met for each row. For example, a unique constraint can be applied to a single column to ensure that each value in that column is unique across all rows in the table. Additionally, constraints can involve multiple columns, ensuring that combinations of values across these columns are unique or meet other specified conditions. This ensures that the data stored in the table remains consistent and accurate. A unique index creates an index on one or more columns, ensuring that no two rows in the table have the same combination of values in those columns. Similarly, a unique constraint defines a rule that prevents duplicate values from being inserted into specified columns. Both unique indexes and constraints provide a way to enforce data integrity by preventing the insertion of duplicate or non-unique values. A primary key constraint is a rule applied to one or more columns in a table, specifying that the values in those columns must be unique and not null. The primary key uniquely identifies each row in the table and serves as a unique identifier for the records. A primary key constraint is created on one column, but it can also be defined on multiple columns (composite primary key). By enforcing uniqueness and non-nullability, the primary key constraint ensures data integrity. Foreign key constraints establish a relationship between two tables by specifying that the values in a column (or columns) in one table must match the values in a column (or columns) in another table's primary key column(s). This constraint ensures referential integrity, meaning that the values in the referencing table (the foreign key table) must exist in the referenced table (the primary key table). Foreign key constraints are commonly used to enforce relationships between tables in a relational database, ensuring that related data remains consistent and accurate.