

# Teradata Certification

---

## Developer Exam

### Exam Objectives

The Developer Exam covers the features and functionality of the Advanced SQL Engine through release 16.20. The exam objectives describe the content and focus covered on the exam.

#### **Solution Development Process and Considerations – 17%**

1. Identify database and user concepts (for example: ownership, privileges, separation of duties, database user parent/child relationships, etc.)
2. Identify the types and purposes of indexes (for example: primary, secondary, join, and primary time indexes.)
3. Identify the characteristics and usage of roles, profiles, and privileges.
4. Identify the characteristics of persistent / permanent and temporary tables.
5. Identify the purpose and limitations of multi-value compression (MVC).
6. Identify the characteristics and usage of row and column partitioning, constraints, and dynamic join elimination.
7. Identify the purpose and audience of data models.
8. Identify the processes for efficient database design (for example: data modeling, data profiling, data domains, data and join demographics, and access patterns.)
9. Identify the types of normalization rules and how to show data relations and relationships.
10. Given an Explain plan, identify the processing that would be executed.

#### **Transaction Processing and Lock Management – 10%**

1. Given a scenario, identify the types of locking and lock levels used by Teradata.
2. Identify how and when locks are established and the impact of locking conflicts.
3. Identify the implications of using different transaction modes (ANSI and BTET.)
4. Identify the differences between a transaction, request and statement.
5. Given a scenario, identify the impact of session characteristics on request (for example: ANSI vs BTET, min and max sessions, date format, default character set, etc.)

## Basic SQL Concepts – 40%

1. Identify the correct syntax, usage, and characteristics of macros.
2. Identify the correct syntax, usage, and characteristics of stored procedures.
3. Identify the correct syntax, usage, and characteristics of permanent / persistent tables.
4. Identify the correct syntax, usage, and characteristics of permanent / persistent table level attributes.
5. Identify the correct syntax, usage, and characteristics of CREATE TABLE AS.
6. Identify the correct usage and characteristics of column level attributes.
7. Identify the usage and characteristics of basic data types (for example: VARCHAR, CHAR, DATE, TIMESTAMP, LOB, etc.)
8. Identify the usage and characteristics of table level and column level constraints.
9. Identify the types, correct syntax, usage, characteristics, and implications of Referential Integrity (RI).
10. Identify the correct syntax, usage, and characteristics of primary index and NoPI tables.
11. Identify the correct syntax, usage, and characteristics of secondary indexes.
12. Identify the correct syntax, usage, and characteristics of single and multilevel row partitioning.
13. Identify the correct syntax, usage, and characteristics of hybrid column and row partitioning.
14. Identify the correct syntax, usage, and characteristics of join indexes.
15. Identify the correct syntax, usage, and characteristics of views (for example: nested, base (1:1), updateable, etc.)
16. Identify the correct syntax, usage and characteristics of triggers.
17. Identify the correct syntax, usage, and characteristics of volatile temporary tables.
18. Identify the correct syntax, usage, and characteristics of global temporary tables.
19. Identify the correct syntax, usage, and characteristics of derived tables.
20. Identify the correct syntax, usage, and characteristics of the SELECT statement.
21. Identify the correct syntax, usage, and characteristics of the regular expressions in an SQL statement (for example: CASE, TRIM, COALESCE, NULLIF, etc.)
22. Identify the correct syntax, usage, and characteristics of common clauses, functions, operators, and expressions (for example: SAMPLE, RANDOM, TOP, CAST, etc.)
23. Identify the usage and characteristics of table operators.
24. Identify the usage and characteristics of ANSI SQL:2011 Window aggregate functions (for example: GROUP functions including QUALIFY, ROW\_NUMBER, COUNT, MIN, MAX, AVG, RANK.)
25. Identify the usage and characteristics of Teradata functions (for example: Built-In, String, Table, Hash, Date Time/interval, Arithmetic, etc.)
26. Identify the usage and characteristics of Period data type functions (for example: BEGIN, END, PRECEDES, SUCCEEDS, OVERLAPS, etc.)
27. Identify the usage and characteristics of System Calendar functions (for example: day of week, day of month, week of year, year of calendar, month of year, etc.)
28. Identify the characteristics and usage of the System Calendars (for example: Teradata, ISO, and Compatible.)
29. Identify the characteristics, implications, and correct syntax of joins and the impact of join order.
30. Identify the characteristics, implications, and correct syntax of subqueries in different parts (for example: WHEN clauses, CASE statements, FROM clauses, etc.) of the SQL statement.
31. Identify the characteristics, implications, and correct syntax of MERGE including logging errors.

32. Identify the characteristics, implications, and correct syntax of INSERT, UPDATE, and DELETE statements.

### **Data Integration Strategies - 14%**

1. Identify the characteristics, limitations, and usage of the TPT LOAD Operator.
2. Identify the characteristics, limitations, and usage of the TPT EXPORT Operator.
3. Identify the characteristics, limitations, and usage of the TPT UPDATE Operator.
4. Identify the characteristics, limitations, and usage of the TPT STREAM Operator.
5. Identify the characteristics, limitations, and usage of TPT Operators including DDL, ODBC, SQL INSERTER, and SQL SELECTOR, etc.
6. Identify the characteristics, limitations, and usage of BTEQ.
7. Given a scenario, identify the optimal load strategy (for example: mini-batch, TPT LOAD / UPDATE, TPT STREAM, etc.)
8. Identify the applications and use cases for using the Teradata Studio Suite.

### **Access Layer and Data Delivery Strategies – 3%**

1. Identify the considerations when creating or using an access layer.

### **Solution Optimization – 16%**

1. Given a scenario, identify performance tuning processes and options to improve an existing application (for example: the approach, explain plans, additional indexes, data distribution, predicates, statistics, etc.)
2. Given a scenario, identify the process for designing and populating an effective performance testing environment.
3. Given a DBQL implementation, identify how to make use of query performance data in designing and tuning an application (for example: using production data in development, etc.)
4. Given a scenario about a workload that uses high-volume, critical queries, identify performance tuning considerations