Teradata Certification

Exam Objectives

The Data Engineering Exam covers the features and functionality of Vantage 2.2 including the Advanced SQL Engine through release 17.05.

Manage and Optimize Data Solutions – 16%

- Given a scenario, identify the type of secondary or join indexes that should be created to improve performance of a class of queries.
- Identify the characteristics and usage of row and column partitioning, constraints, and dynamic partition elimination.
- Identify the processes for efficient database design (e.g., data profiling, data domains, data and join demographics, and access patterns.)
- Identify the characteristics and impacts of system defaults (e.g., currency, current date, timestamp, for a geographic region, etc.)
- Given an Explain plan, identify the processing that would be executed.
- Given a scenario about the requirements from a data scientist for new data features, identify the steps to integrate the features.
- Identify the considerations when creating or using an access layer.
- Identify the benefits, usage, and limitations of using QueryGrid and other approaches as part of a data access strategy.

Transaction Processing and Lock Management – 8%

- Given a scenario, identify advanced locking types and implications (e.g., READ UNCOMMITTED (access lock), COMMITTED (load isolation), and SERIALIZABLE (read lock).)
- Identify the characteristics of a transaction, request and statement.
- Given a scenario, identify the impact of session characteristics on request (e.g., error recovery implications such as dead-lock situations, ANSI vs BTET, min and max sessions, date format, default character set, etc.)

SQL Concepts– 39%

- Identify the usage and characteristics of macros.
- Identify the usage and characteristics of stored procedures.
- Identify the usage and characteristics of tables (e.g., permanent, error, and QUEUE)
- Identify the usage and characteristics of advanced data types (JSON, Period)
- Identify the usage and characteristics of column level attributes (e.g., FORMAT, DEFAULT, CASESPECIFIC, COMPRESS, etc.)
- Identify the usage and characteristics of basic data types (e.g., VARCHAR, CHAR, DATE, TIMESTAMP, LOB, etc.)
- Identify the usage and characteristics of table level and column level constraints (e.g., Check, Primary Key, References, and Unique)
- Identify the types, usage, characteristics, and implications of referential integrity (RI).
- Identify the usage and characteristics of primary index and NoPI tables.
- Identify the usage and characteristics of secondary indexes.

- Identify the usage and characteristics of join indexes.
- Identify the use cases of, performance, characteristics, and implications including maintenance issues of column or row partitioning.
- Identify usage and characteristics of views (e.g., nested, base (1:1), updateable, etc.)
- Identify the situations when global temporary tables or volatile temporary tables should be used and the limitations of their use.
- Identify usage and characteristics of the advanced regular expression functions (e.g., REGEXP_SUBSTR, REGEXP_INSTR, REGEXP_SIMILAR, REGEXP_REPLACE, REGEXP_SPLIT_TO_TABLE.)
- Identify the usage and characteristics of derived tables.
- Identify the usage and characteristics of the advanced ANSI SQL:2011 Window aggregates including Group window, cumulative window, and moving window.
- Identify the usage and characteristics of JSON functions (e.g., compression.)
- Identify the usage and characteristics of common clauses, operators, and expressions (e.g., SAMPLE, RANDOM, TOP, CAST, etc.)
- Identify the usage and characteristics of Period data type functions (e.g., BEGIN, END, OVERLAPS, LDIFF, RDIFF, UNTIL_CHANGED/IS NOT UNTIL_CHANGED, IMMEDIATELY PRECEDES, and IMMEDIATELY SUCCEEDS, TD_NORMALIZE, etc.)
- Identify the characteristics and implications of using correlated subqueries.
- Identify the usage and characteristics of ANSI SQL:2011 Window aggregate functions (e.g., GROUP functions including QUALIFY, ROW_NUMBER, COUNT, MIN, MAX, AVG, RANK.)
- Identify the implications of nulls with inner and outer joins.
- Identify the usage and characteristics of scalar functions available in Teradata (Teradata and ANSI variants, including embedded services functions)
- Identify the characteristics and implications of a scalar or a nonscalar subquery.
- Identify the function, characteristics and implications of joins and types of join (e.g., inner join, outer joins, cross join.)
- Identify the characteristics and implications of subqueries in different parts (e.g., WHEN clauses, CASE statements, FROM clauses, etc.) of the SQL statement.
- Identify the characteristics and implications of MERGE including logging errors.
- Identify the characteristics and implications of INSERT, UPDATE, and DELETE statements.

Data Integration Strategies – 13%

- Identify the characteristics, limitations, and usage of TPT LOAD, EXPORT, UPDATE (including MLOADX) and STREAM operators.
- Given a TPT job, identify the additional considerations for error handling and job handling that should be used.
- Identify the characteristics, limitations, and usage of TPT DDL, ODBC, SQL INSERTER, and SQL SELECTOR operators.
- Given a scenario, identify the optimal load strategy (e.g., mini-batch, TPT LOAD / UPDATE, TPT STREAM, etc.)
- Identify the benefits, usage, and limitations of using access modules as part of a data access strategy.

Solution Optimization – 13%

- Given a scenario, identify performance tuning processes and options to improve an existing application (e.g., the approach, explain plans (including dynamic explain), additional indexes, data distribution, predicates, statistics, etc.)

- Identify characteristics and implications of advanced tuning options that improve performance (e.g., IPE, PRPD, and Query Rewrite, etc.)
- Given a DBQL implementation, identify how to make use of query performance data in designing and tuning an application (e.g., using production data in development, etc.)
- Identify the design options and characteristics of an optimized application (e.g., elements of master data, flexibility, extensibility, resiliency, portability, simplicity, etc.)
- Given a scenario with a data quality issue, identify the function(s) that resolves the issue.
- Identify use cases for the output from data profiling.

Vantage Integration and Interoperability – 11%

- Identify the steps and considerations to move data from object storage.
- Identify the steps and considerations to move data to object storage.
- Identify the steps and considerations to leverage data in third-party data storage.
- Identify the steps when building and operarationalizing a predictive model pipeline.
- Identify the steps when building and operarationalizing a data pipeline.
- Identify how Vantage supports event-driven architecture.
- Identify the tools and features Vantage uses to integrate with native cloud services.
- Identify the features of Vantage that enables OpenSource integration.