Teradata Certified Professional Program

System Architect Beta Exam

Exam Objectives

The System Architect Beta Exam covers the features and functionality of the Teradata Database through release 16.20. The exam objectives describe the content and focus covered on the exam.

System Architecture Concepts – 14%

1. Identify the purposes of different types of metadata (business, technical, and operational).
2. Identify the characteristics of conceptual modeling.
3. Given a scenario, identify the logical model that should be created.
4. Given a scenario, identify how to use an extended logical data model.
5. Identify the uses and benefits of logical and physical models.
6. Identify strategies for managing system expansion and contraction.
7. Given a scenario, identify strategies to extend traditional application deployment.
8. Given a scenario, identify how to architect a non-production environment (for example: DEV, QA, etc.)
9. Identify the components of an analytic ecosystem.

Information Management and Data Governance – 11%

1. Given a scenario, identify which loading strategy should be used.
2. Given a scenario, identify which Teradata data acquisition tool(s) should be used.
3. Given a scenario, identify how to create data for a non-production environment.
4. Given a scenario, identify the data retention, placement, and archive strategies that should be used.
5. Identify the benefits of effective master data management.

Performance Design – 16%

1. Given a scenario, identify physical design choices for indexes (PI, SI, and NoPI) for optimal performance (response time and resource consumption).
2. Given a scenario, identify physical design choices for join indexes (Single/Multitable, Aggregate, Global, and Sparse) for optimal performance (response time, maintainability, and resource consumption).
3. Given a scenario, identify physical design choices for row partitioning for optimal performance (response time and resource consumption).
4. Given a scenario, identify physical design choices for column partitioning for optimal performance (response time and resource consumption).
5. Given a scenario, identify physical design choices for time series data (Primary Time Index (PTI)) for optimal performance (response time and resource consumption).
6. Given a scenario, identify physical design choices for data types for optimal performance (response time and resource consumption).

7. Given a scenario, identify which statistics should be collected and how often for optimal performance (response time and resource consumption).
8. Given a scenario, identify the compression option(s) that should be used.
9. Given a scenario, identify how to manage tables and their relationship to the MAPS feature.

Architecting for System Performance – 5%

1. Identify the benefits of Teradata Virtual Storage (TVS) and Teradata Intelligent Memory (TIM).
2. Given a scenario about multiple Vantage systems, identify the Unity components and strategy that should be used.

Data Integration – 11%

1. Given a scenario, identify the appropriate data transformation strategy(ies).
2. Given a scenario, identify the appropriate method/level to organize tightly, loosely, and non-coupled data.
3. Given a scenario, identify when and how surrogate keys should be generated, or when natural keys should be used.
4. Given a scenario, identify the strategy that should be used to achieve the correct level of data granularity.
5. Given a scenario including a data modeling method (for example: snowflake, star, normalized), identify the benefits and tradeoffs.
6. Given a scenario, identify the design considerations when using complex data types, such as JSON, XML, DATASET (AVRO and CSV).
7. Given a scenario, identify the design considerations when integrating temporal data.
8. Given a scenario, identify the design considerations when integrating geospatial data.

Data Security – 7%

1. Identify the security considerations for multi-system environments.
2. Identify use cases for secure zones.
3. Given a scenario, identify which user authentication mechanism should be used.
4. Given a scenario, identify database mechanisms for controlling access to data.
5. Given a scenario, identify how roles can be used to manage privileges for groups of users.

Data Protection and System Availability – 11%

1. Given a scenario, identify the backup and recovery strategy that should be used.
2. Given a scenario, identify design considerations in multi-system data replication and loading.
3. Given a scenario, identify design considerations for disaster recovery.
4. Identify features that provide system protection (for example: fallback, global hot spare, and HSN).

Data Residency and Portability – 3%

1. Given a scenario, identify the considerations for connectivity and data latency for multiple platforms including those hosted in multiple countries.

Information Delivery – 10%

1. Identify use cases where QueryGrid is beneficial.
2. Identify use cases for different access layers.
3. Identify use cases where sandboxes (for example: Teradata Data Labs) should be used.
4. Identify use cases where physicalizing of the access layer or a dependent data mart is beneficial.
5. Given a scenario, identify the appropriate business intelligence (BI) architectures.

Workload Management – 12%

1. Identify capabilities, benefits, and tradeoffs of Workload Management Capacity on Demand (WM COD).
2. Identify how to leverage workload management using workloads and prioritization to meet service level goals.
3. Identify how to leverage workload management using filters and throttles to meet service level goals.
4. Identify how to leverage workload management using exceptions to meet service level goals.
5. Given a scenario, identify how to leverage workload management using state matrix to meet service level goals.
6. Identify how query bands are beneficial for workload management.