Teradata Database Version 2 Release 6.1.0 (V2R6.1.0)

Security Target

Version 2.0
February 2007

TRP Number: 541-0006458

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

This section identifies and provides and overview of the Security Target (ST). It also identifies the Target of Evaluation (TOE) and provides an evaluatable claim of Common Criteria (CC) conformance for the TOE.

1.1 SECURITY TARGET IDENTIFICATION AND OVERVIEW

The Security Target is identified as follows:

Teradata Database Version 2 Release 6.1.0 (V2R6.1.0) Security Target
Version 2.0
February 2007

The TOE described herein is a Relational Database Management System that provides a discretionary access control model to protect stored database objects and resources.

This ST describes the security assumptions, threats, objectives, requirements, and an associated rationale for the Teradata Database and its IT environment. The language used in this Security Target is consistent with the *Common Criteria for Information Technology Security Evaluation, Version 2.3*.

Chapter 1 of this ST provides introductory and identifying information for the ST and the TOE. Chapter 2 describes the TOE and provides some guidance on its use. Chapter 3 describes a security environment in terms of assumptions and threats. Chapter 4 identifies the security objectives of the TOE and of the Information Technology (IT) environment. Chapter 5 describes the TOE security functional requirements, the assurance requirements, as well as requirements on the IT environment. Chapter 6 is the TOE Summary Specification, a description of the functions provided by the Teradata Database to satisfy the security functional and assurance requirements. Chapter 7 provides a rationale for claims of conformance to a registered Protection Profile (PP). Chapter 8 provides a rationale, or pointers to rationale, for objectives, requirements, TOE Summary Specification, strength of functions, and Protection Profile claims.

1.2 TOE IDENTIFICATION

The TOE defined in this ST is identified as follows:

Teradata Database Version 2 Release 6.1.0 (V2R6.1.0)

The TOE is a Relational Database Management System (RDBMS) and is referred to as the Teradata Database within this ST.

1.3 COMMON CRITERIA CONFORMANCE

This ST conforms to the following Common Criteria specifications:

Common Criteria for Information Technology Security Evaluation
Part 2: Security functional requirements
August 2005
Version 2.3

Common Criteria for Information Technology Security Evaluation
Part 3: Security assurance requirements
August 2005
Version 2.3

- EAL 4 augmented with ALC_FLR.3

Additionally, the ST is extended to include additional explicit security functional and environmental requirements. In this ST, explicit security functional requirements are indicated with '_(EXP)' following the component name and explicit environmental requirements are indicated with '_(ENV)' following the component name.

This Security Target does not claim conformance to a Protection Profile.

2. TOE DESCRIPTION

The Target of Evaluation (TOE) is the Teradata Database V2R6.1.0 which is configured and managed as described in the installation, generation, and start-up procedures, administrator guidance, and user guidance documentation referenced in section 6.2 of this ST.

The Teradata Database is a relational database management system (RDBMS) that is designed to access, store, and operate on data using Teradata Structured Query Language (Teradata SQL), which is compatible to ANSI SQL with extensions. The database was developed to allow users to view and manage large amounts of data as a collection of related tables. The database executes as a trusted parallel application (TPA) on a symmetric multiprocessing (SMP) or massively parallel processing (MPP) database server.

Note: This evaluation is limited to Teradata Database V2R6.1 running on Windows Server 2003.

2.1 TOE OVERVIEW

The Teradata Database is a relational database management system (RDBMS) that includes security functionality for parallel database environments supporting multiple concurrent users. The security functionality includes:

- user management - including identification and authentication
- password management controls
- discretionary access control model to enforce access controls on database objects and resources (e.g., databases, users, tables, views, stored procedures and macros)
- extensive set of access rights for the enforcement of the principle of least privilege
- security roles for management of access rights
- configurable auditing facility

The Teradata Database functions as a database server in a traditional client/server environment. Access requests are made via the Teradata Tools and Utilities that provide connectivity to the database and submit Teradata SQL statements to the database. For any access to the database through its defined external interfaces, the database ensures that all security enforcement functions are invoked and succeed before any access request is allowed to proceed.

The Teradata Database operates as a trusted parallel application executing as a set of cooperating processes on an underlying host operating system. The host operating system is not part of the TOE but rather part of the supporting IT Environment. The IT Environment provides several supporting security mechanisms to prevent compromise of the TOE security functions including:

- authentication and authorization of administrator access to database control utilities and other utilities used to manage system resources and I/O interfaces
- isolation of the TOE Security Functions (TSF) to prevent tampering with TSF components (e.g., the TOE processes managing the database)
- network perimeter controls to restrict network access to the database server to mitigate malicious attacks against the operating system upon which the TOE operates

2.2 TOE ARCHITECTURE

The Teradata Database is comprised of several software subsystems including the Parallel Database Extension (PDE), Gateway for LAN, Session Controller, Parser and Access Module Processors (AMP). A Session Controller and a Parser subsystem are always configured together in what is called a Parsing Engine (PE) virtual processor.

The PDE subsystem is a software interface layer that operates on top of the host operating system and provides an interface between the other database subsystems and the underlying operating system software. PDE includes a BYNET driver that manages the communication devices that interconnect the hardware nodes on which the server software is resident. It provides a standard interface for inter-process communications across nodes in a multi-node environment. PDE also includes a Console module (CNS) that manages the interface for input and output generated from a Database Window (DBW) on the Console.

The Gateway for LAN subsystem provides the client communications interface to Client applications connected via a network interface. It receives all messages sent from the client to the server. This includes messages containing Teradata SQL statements as well as messages for functions such as connecting and disconnecting sessions, determining the configuration of the server, establishing the security protocols to be used between the client and server, and responding to test messages that determine the health of the server over the LAN. For messages that contain Teradata SQL, the Gateway for LAN checks those messages to ensure that they conform to the specified protocol and forwards them to a Parser subsystem. The Gateway for LAN also receives response messages from the PE subsystems and returns them to the appropriate Client application. The Gateway for LAN also interacts with PDE for memory management and message handling services and for access to underlying operating system services.

A PE virtual processor always includes a Session Controller and a Parser subsystem. The Session Controller processes external requests to establish or terminate a logical connection between the application and the server. It also provides for the recovery of sessions following client or server failures. The Session Controller manages session activities, such as logon, password validation and logoff. The Parser decomposes SQL into relational data management processing steps. It processes external requests containing Teradata SQL by syntactically parsing the statements and generating a set of steps comprising an execution plan for the statements. Other Parser modules then access the generated steps and send them to one or more AMP subsystems for execution. Parser modules also monitor the execution of the steps, handle errors encountered during processing and return the execution results to the Gateway for return to the Client application.

An AMP subsystem physically structures the TOE managed relational data and it processes the steps of an SQL execution plan to access that data. It also manages a set of relational tables containing the description of the user defined data objects. The AMP subsystem provides access to these dictionary tables to Client applications through standard SQL and to other database subsystems as needed and is responsible for the integrity of the relational data structures. The

AMP subsystem reads and writes the relational data structures from/to disk storage by making calls to the PDE subsystem which subsequently calls the underlying host operating system to perform the required physical read and write operations.

Other components exist in the Teradata Database environment and interface to the database, but are excluded from the definition of the TOE. These components include:

- The operating system on which the database executes
- The database server node upon which the database software and underlying operating system operates
- The disk storage subsystem and its associated SCSI or Fibre Channel interface.
- The Console's Database Window (DBW) software.
- The Teradata Tools and Utilities (Client) applications including the Call Level Interface (CLI) software that processes messages sent to, and received from, the database.

2.2.1 TOE Physical Boundaries

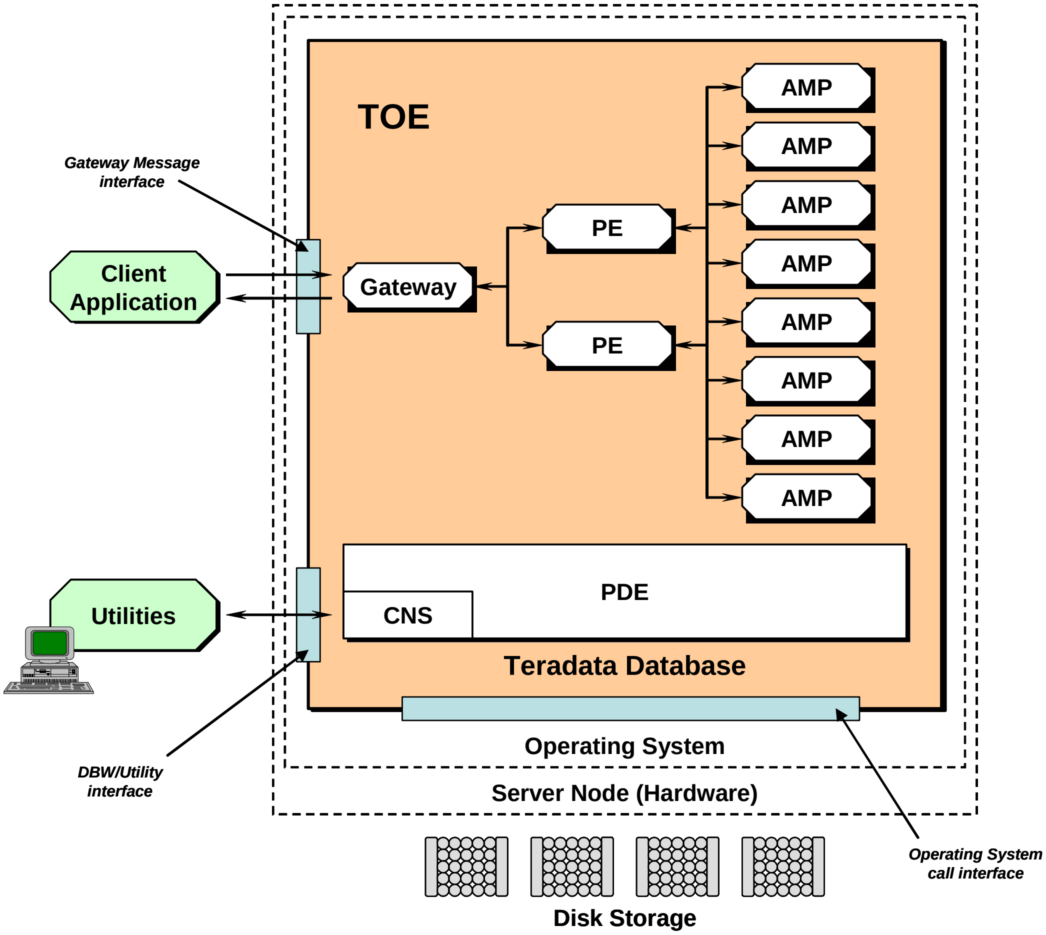
The physical boundaries of the TOE are depicted in the Figure 2-1.

There are two external interfaces to the Teradata Database. The Gateway Message interface receives text messages containing service requests from Client applications and returns text message responses to the applications upon completion of a service request. The DBW/Utility interface provides for Console access to executable processes of the PDE subsystem.

The Teradata Database makes calls to the underlying operating system to access operating system services and to access the associated disk storage subsystem.

Note that the TOE is defined as a software-only TOE. As such, the Server Node (Hardware) and Disk Storage is specifically outside the TOE boundary. (The disk storage resides in a separate disk array cabinet that is packaged completely separately from the Server Node hardware. In some very small environments where the Teradata Database may be running on a standalone server platform, the disk storage may be packaged as part of the server platform.)

Figure 2-1 TOE Physical Boundaries



2.2.2 TOE Logical Boundaries

The logical boundaries of the TOE are defined by the supported security functions. All five subsystems of the TOE contribute to meeting the security functional requirements.

TOE Access - The Teradata Database allows authorized administrative users to restrict access to the database based on user identities.

Identification and Authentication - The Teradata Database provides user identification and authentication through the use of user accounts and the enforcement of password policies. Users must provide a valid username and password before they can access any database objects or resources. Once identified and authenticated, all subsequent actions allowed within that user's session are based on the user's identity, access rights, and active roles.

User Data Protection - The Teradata Database enforces a Discretionary Access Control (DAC) policy for object access based on user identities, object ownership, and active roles. All access to database objects subject to the DAC policy is controlled using access rights. The Teradata Database supports three types of access rights. Implicit rights (ownership rights) are implicitly granted to the immediate owner of a database or database object. Automatic rights are granted automatically by the system to the creator of a database, user, or object, and to a newly created user or database. Explicit rights are granted by any user having the `WITH GRANT OPTION` privilege for that right. The database ensures that the requestor has the appropriate access rights before access to a database object is allowed.

Upon initial installation of the Teradata Database, it has only one user. This user is called user `DBC` and will own all other databases and users in the system. User `DBC` also has access rights on all objects within the database with the exception of `CREATE PROCEDURE` and `EXECUTE PROCEDURE`. Typically, administrative users are created under user `DBC` and are granted access rights for creating and managing all other users, databases, and objects.

Security Audit - The Teradata Database automatically audits all successful and failed user logon attempts in the event log. An authorized administrative user may search and sort logon/logoff records using SQL statements to query a defined system view. Additionally, an authorized administrative user may control the monitoring of access rights checks performed by Teradata Database and may search and sort access log records using SQL statements to query a defined system view.

Security Management - The Teradata Database provides security management functions that enable authorized administrative users to manage the secure operation of the database. These functions include management of users, user security attributes, access rights, security roles, and the audit facilities.

Resource Utilization - The Teradata Database enforces maximum quotas and limits on various resources to ensure that those resources are protected from monopolization by any individual database user. Specifically, an authorized administrator can configure the database to enforce limits on permanent database space allocation, temporary database space usage, and spool database space usage.

Protection of the TSF - The Teradata Database is designed with well-defined interfaces that ensure that all appropriate security checks are made before access is provided to protected database objects and resources. The Teradata Database operates as a set of cooperating processes which are managed by the underlying operating system. These processes operate as a trusted parallel application (TPA) such that no interference is allowed by processes associated with any non-TOE entities. Furthermore, the Teradata Database is designed such that its interfaces do not allow unauthorized users access to database resources.

The Teradata Database is the only application executing on the server and underlying operating system. Other applications, such as web server (e.g., Microsoft Internet Information Services), e-mail server (e.g., Microsoft Exchange Server), domain server (e.g., Microsoft Active Directory), etc. do not run on the Teradata Database server.

2.3 TOE DOCUMENTATION

Teradata provides an extensive set of reference manuals and other documentation to describe the installation, generation, and start-up procedures and to provide administrator and guidance regarding the secure operation of the Teradata Database. Section 6.2 of this ST provides information about the documentation.

3. TOE SECURITY ENVIRONMENT

The security environment for the functions addressed by this ST includes assumptions (A) and threats (T) as described in the following sections.

3.1 ASSUMPTIONS

This section provides a description of assumptions that describe the security aspects of the environment in which the TOE will be used or is intended to be used.

A.DOMAIN_SEPARATION The IT environment will provide a separate domain for the TOE's operation.

A.I_AND_A It is assumed that the IT environment will provide identification and authentication mechanisms for use of utilities under the control of the IT environment.

A.NO_BYPASS The IT environment will ensure the TSF cannot be bypassed in order to gain access to TOE data.

A.NO_EVIL Administrators are non-hostile, appropriately trained, and follow all administrator guidance.

A.NO_GENERAL_PURPOSE There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the database server, other than those services necessary for the operation, administration and support of the database.

A.PHYSICAL It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

A.RESTRICT_OS_ACCESS It is assumed that logon access to the underlying operating system is restricted to authorized Teradata administrators only.

A.ROBUST_ENVIRONMENT It is assumed that the IT environment is at least as robust as the TOE.

A.SECURE_COMMS It is assumed that the IT environment will provide a secure (protected from disclosure, spoofing, and able to detect modification) line of communications between the remote user and the TOE.

A.TIME_STAMPS It is assumed that the IT environment will provide the TOE with the necessary reliable timestamps.

3.2 THREATS

This section provides a description of threats to the assets against which specific protection within the TOE or its environment is required.

T.ACCOUNTABILITY The authorized users of the TOE may not be held accountable for their actions within the TOE.

T.ADMIN_ERROR An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.

T. AUDIT_ COMPROMISE A user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user's action.

T.MASQUERADE A user or process may masquerade as another entity in order to gain unauthorized access to data or TOE resources.

T.POOR_DESIGN Unintentional errors in requirements specification or design of the TOE may occur, leading to flaws that may be exploited by a casually mischievous user or program.

T.POOR_IMPLEMENTATION Unintentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a casually mischievous user or program.

T.POOR_TEST Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being discovered thereby causing potential security vulnerabilities.

T.RESIDUAL_DATA A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.

T.RESOURCE An authenticated database user might consume excessive database resources such that access to database resources by other database users is compromised.

T.SECADMIN The TOE may not be configured with an authorized security administrator user, separate and distinct from other authorized administrative users, to provide for secure administration of the TOE.

T.TSF_COMPROMISE A malicious user or process may cause configuration data to be inappropriately accessed (viewed, modified or deleted).

T.UNAUTHORIZED_ACCESS A user may gain unauthorized access to user data for which they are not authorized according to the TOE security policy.

T.UNIDENTIFIED_ACTIONS Failure of the authorized administrator to identify and act upon unauthorized actions may occur.

3.3 ORGANIZATIONAL SECURITY POLICIES

The security objectives for the TOE and its environment are derived only from the threats and assumptions described above.

4. SECURITY OBJECTIVES

The objectives listed in this section are intended to address all identified assumptions and counter all identified threats. The security objectives for the TOE are prefaced with an ‘O’ and those for the Environment are prefaced with an ‘OE’.

4.1 SECURITY OBJECTIVES FOR THE TOE

O.ADMIN_GUIDANCE
The TOE will provide administrators with the necessary information for secure management.

O.AUDIT_GENERATION
The TOE will provide the capability to detect and create records of security relevant events associated with users.

O.AUDIT_REVIEW
The TOE will contain mechanisms to allow the authorized administrator to view and sort the audit logs.

O.AUDIT_STORAGE
The TOE will contain mechanisms to provide secure storage and management of the audit log.

O.CONFIG_IDENTIFICATION
The configuration of the TOE is fully identified in a manner that will allow implementation errors to be identified, corrected with the TOE being redistributed promptly.

O.DOCUMENTED_DESIGN
The design of the TOE is adequately and accurately documented.

O.I_AND_A
The TOE will contain identification and authentication mechanisms for users to login to the TOE.

O.MANAGE
The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.

O.MEDIATE
The TOE must protect user data in accordance with its security policy.

O.INTERNAL_TOE_DOMAINS
The TSF will maintain internal domains for separation of data and queries belonging to concurrent users.

O.PARTIAL_FUNCTIONAL_TEST The TOE will undergo some security functional testing that demonstrates the TSF satisfies some of its security functional requirements.

O.PARTIAL_SELF_PROTECTION The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces.

O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource within its Scope of Control is not released when the resource is reallocated.

O.RESOURCE The TOE will provide for limiting the consumption of database resources by authorized users of the TOE.

O.SECADMIN The TOE will provide for the creation of an authorized security administrator to isolate administrative actions.

O.TOE_ACCESS The TOE will provide mechanisms that control a user’s logical access to the TOE.

O.VULNERABILITY_ANALYSIS The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any obvious flaws.

4.2 Security Objectives for the Environment

OE.DOMAIN_SEPARATION The IT environment will provide an isolated domain for the execution of the TOE.

OE_I_AND_A The IT environment will contain identification and authentication mechanisms for administrator access to database control utilities and other utilities.

OE.NO_BYPASS The IT environment shall ensure the TOE security mechanisms cannot be bypassed in order to gain access to the TOE resources.

OE.NO_EVIL Sites using the TOE shall ensure that authorized administrators are non-hostile, appropriately trained and follow all administrator guidance.

OE.CONFIG The TOE and the underlying operating system will be installed, configured, managed and maintained in accordance with its guidance documentation and applicable security policies and procedures.

OE.NO_GENERAL_ PURPOSE

There will be no general-purpose computing capabilities (e.g., user applications) available on DMBS servers, other than those services necessary for the operation, administration and support of the database.

OE.PHYSICAL

Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

OE.RESTRICT_OS_ACCESS

The underlying operating system will be configured with only those user accounts required for access by authorized Teradata administrators.

OE.ROBUST_ENVIRONMENT

The IT environment that supports the TOE for enforcement of its security objectives will be of at least the same level of robustness as the TOE.

OE.SECURE_COMMS

The IT environment will provide a secure line of communications between the remote user and the TOE.

OE.TIME_STAMPS

The IT environment will provide reliable time stamps.

OE.TRUST_IT

Each IT entity the TOE relies on for security functions will be installed, configured, managed and maintained in a manner appropriate to the IT entity, and consistent with the security policy of the TOE and the relationship between them.

5. SECURITY REQUIREMENTS

This section identifies the security functional requirements for the TOE and its' IT environment. In addition, this section also presents the security assurance requirements for the TOE. The operations performed on the security functional and assurance requirements contained in this section adhere to the following conventions:

- Iteration: Allows a component to be used more than once with varying operations. In the ST, a number in parenthesis appended to a component indicates iteration.
- Assignment: Allows the specification of an identified parameter. Assignments are indicated using italicized text and are surrounded by brackets (e.g., [*assignment*]).
- Selection: Allows the specification of one or more elements from a list. Selections are indicated using bold italicized text and are surrounded by brackets (e.g., [***selection***]).
- Refinement: Allows the addition of details. Refinements are indicated using bold text for additions to the requirements (e.g., **refinement**).

5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS

The following table provides a summary of the security functional requirements implemented by the TOE.

Table 5-1 TOE Security Functional Requirements

Security Functional Class	Security Functional Component
Security Audit (FAU)	FAU_GEN.1 Audit data generation
	FAU_GEN.2 User identity association
	FAU_SAR.1 Audit review
	FAU_SAR.2 Restricted audit review
	FAU_SAR.3 Selectable audit review
	FAU_SEL.1 Selective audit
	FAU_STG.1 Protected audit trail storage
User Data Protection (FDP)	FDP_ACC.1 Subset access control
	FDP_ACF.1 Security attribute based access control
	FDP_RIP.1 Subset residual information protection
Identification and Authentication (FIA)	FIA_AFL.1 Authentication failure handling
	FIA_ATD.1 User attribute definition
	FIA_SOS.1 Verification of secrets
	FIA_UAU.1(1) Timing of authentication
	FIA_UID.1(1) Timing of identification
	FIA_USB.1 User-subject binding
Security Management (FMT)	FMT_MOF.1 Management of security functions behaviour
	FMT_MSA.1 Management of security attributes

Security Functional Class	Security Functional Component
	FMT_MSA.3 Static attribute initialisation
	FMT_MTD.1 Management of TSF data (1)
	FMT_MTD.1 Management of TSF data (2)
	FMT_MTD.1 Management of TSF data (3)
	FMT_REV.1 Revocation
	FMT_SMF.1 Specification of management functions
	FMT_SMR.1 Security roles
Protection of the TSF (FPT)	FPT_RVM.1(1) Non-bypassability of the TSP
	FPT_SEP_(EXP).1 TSF domain separation
Resource Utilisation (FRU)	FRU_RSA.1 Maximum quotas
TOE Access (FTA)	FTA_TSE.1 TOE session establishment

The following subsections present the security functional requirements for the TOE.

5.1.1 Security Audit (FAU)

5.1.1.1 FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events for the [***not specified***] level of audit; and

c) [*Events specified in Table 5-2*].

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*other audit relevant information, as provided under "Additional Data" in Table 5-2*]

Table 5-2 Auditable Events

Component	Event	Additional Data
FAU_SAR.1	Reading of information from the database audit records	None
FAU_SEL.1	All modifications to the database audit configuration that occur while the database audit collection functions are operating	Modified configuration element
FDP_ACF.1	All requests to perform an operation on a database object covered by the SFP	The assigned user access right

Component	Event	Additional Data
FIA_UAU.1(1)	All use of the user authentication mechanism	None
FIA_UID.1(1)	All use of the user identification mechanism, including the user identity provided	None
FIA_USB.1	Success or failure of binding user security attributes to a database subject (e.g., success and failure to create a database subject)	None
FMT_MOF.1	Modifications in the behaviour of the functions of the TSF	Change of threshold for unsuccessful authentication attempts or actions to be taken in the event of an authentication failure
FMT_MSA.1	All modifications of the values of database security attributes	Modification, deletion or addition of database security attributes
FMT_MTD.1	All modifications to the values of TSF data	None
FMT_REV.1	All attempts to revoke database security attributes	None
FMT_SMR.1	Modifications to the group of users that are part of a role	None
FTA_TSE.1	All attempts at establishment of a user session	None

5.1.1.2 FAU_GEN.2 User identity association

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FIA_UID.1(1) Timing of identification

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.1.1.3 FAU_SAR.1 Audit review

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAR.1.1 The TSF shall provide [*the administrator*] with the capability to read [*all audit information*] from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

5.1.1.4 FAU_SAR.2 Restricted audit review

Hierarchical to: No other components.

Dependencies: FAU_SAR.1 Audit review

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

5.1.1.5 FAU_SAR.3 Selectable audit review

Hierarchical to: No other components.

Dependencies: FAU_SAR.1 Audit review

FAU_SAR.3.1 The TSF shall provide the ability to perform [***searches, sorting***] of audit data based on [*all attributes contained within the audit records*].

5.1.1.6 FAU_SEL.1 Selective audit

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FMT_MTD.1 Management of TSF data

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) [***object identity, user identity, event type***]

b) [*denial of access, frequency of access*].

5.1.1.7 FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.1.1 The TSF shall protect the stored audit records from unauthorized deletion **for actions within the TOE Scope of Control**.

FAU_STG.1.2 The TSF shall be able to [***prevent***] unauthorized modifications to the stored audit records in the audit trail **when attempts to modify audit records occur within the TOE Scope of Control**.

5.1.2 User Data Protection (FDP)

5.1.2.1 FDP_ACC.1 Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the [*Discretionary Access Control policy*] on [database subjects, database objects (*databases, users, tables, views, macros, stored procedures*), *database users, and all permitted operations on database objects by database subjects covered by this policy*].

5.1.2.2 FDP_ACF.1 Security attribute based access control

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1 The TSF shall enforce the [*Discretionary Access Control policy*] to objects based on the following: [*database subject attributes: user identity, active roles; database object attributes: object owner and access rights granted on the object*].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [

a) *If the authorized user associated with the subject is the owner of the object, then the requested access is allowed; or*

b) *If the authorized user associated with the subject has the object access right for the requested access to the object, then the requested access is allowed; or*

c) *If the authorized user associated with the subject is the member of an active role or nested role which has the object access right for the requested access to the object, then the requested access is allowed; or*

d) *If `PUBLIC` has the object access right for the requested access to the object, then the requested access is allowed; or*

e) *Otherwise, the access is denied*].

Note: `PUBLIC` is a special internal user provided by the Teradata Database. Access rights granted to `PUBLIC` are applicable to every valid user of the system and all future users of the system.

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [*no additional rules*].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [*no additional rules*].

5.1.2.3 FDP_RIP.1 Subset residual information protection

Hierarchical to:	No other components.
Dependencies:	No dependencies

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] the following objects: [*database tables*].

5.1.3 Identification and Authentication (FIA)

5.1.3.1 FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

Dependencies: FIA_UAU.1(1) Timing of authentication

FIA_AFL.1.1 The TSF shall detect when [***an administrator configurable positive integer within*** [*1 - 127*]] unsuccessful authentication attempts occur related to [*the last successful authentication for the indicated user identity*].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [*disable the account until unlocked by the administrator or until a configurable number of minutes have elapsed*].

5.1.3.2 FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

Dependencies: No dependencies

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [*database user identifier, authentication data, security roles, profile*].

Note: a profile, if assigned to a database user, may contain user-specific password control attributes.

5.1.3.3 FIA_SOS.1 Verification of secrets

Hierarchical to: No other components.

Dependencies: No dependencies

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [*the following requirements:*

- *Passwords will be restricted to a minimum and maximum number of characters in length,*
- *Passwords will contain a combination of upper and lower case characters,*
- *Passwords will contain at least one numeric character,*
- *Passwords will contain at least one special character,*
- *Passwords will not contain the user's username,*
- *Passwords will be valid for a maximum number of days before expiration,*

- *Previously used passwords may not be re-used for a minimum number of days*].

5.1.3.4 FIA_UAU.1(1) Timing of authentication

Hierarchical to: No other components.

Dependencies: FIA_UID.1(1) Timing of identification

FIA_UAU.1.1(1) The TSF shall allow [*establishment of a virtual circuit for the purpose of transferring authentication information, receipt of error messages upon authentication failure*] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2(1) The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.1.3.5 FIA_UID.1(1) Timing of identification

Hierarchical to: No other components.

Dependencies: No dependencies

FIA_UID.1.1(1) The TSF shall allow [*establishment of a virtual circuit for the purpose of transferring identification information, receipt of error messages upon identification failure*] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2(1) The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.1.3.6 FIA_USB.1 User-subject binding

Hierarchical to: No other components.

Dependencies: FIA_ATD.1 User attribute definition

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [*user identity and active roles*].

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [*subject security attributes are derived from TSF data maintained for each defined user after a successful login with the defined user identity*].

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [*a user can set the active role to any or all roles assigned to them*].

5.1.4 Security Management (FMT)

5.1.4.1 FMT_MOF.1 Management of security functions behavior

Hierarchical to: No other components.

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MOF.1.1 TSF shall restrict the ability to [***disable and enable***] the functions [*relating to the specification of events to be audited*] to [*authorized administrative users*].

5.1.4.2 FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control]
FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MSA.1.1 The TSF shall enforce the [*Discretionary Access Control policy*] to restrict the ability to [***modify, delete,*** [*add, or grant*]] the security attributes [*database object access rights, security roles*] to [*authorized administrative users or authorized users*].

5.1.4.3 FMT_MSA.3 Static attribute initialization

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3.1 The TSF shall enforce the [*Discretionary Access Control policy*] to provide [***restrictive***] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [*no identified roles*] to specify alternative initial values to override the default values when an object or information is created.

5.1.4.4 FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MTD.1.1(1) The TSF shall restrict the ability to [***change_default, modify, delete,*** [*or add*]] the [*user identities and security roles*] to [*authorized administrative users*].

FMT_MTD.1.1(2) The TSF shall restrict the ability to [***change_default, delete,*** [*or add*]] the [*authentication data*] to [*authorized administrative users*].

FMT_MTD.1.1(3) The TSF shall restrict the ability to [***modify***] the [*authentication data*] to [*authorized administrative users and users authorized to modify their own authentication data*].

FMT_MTD.1.1(4) The TSF shall restrict the ability to [***change_default, modify, delete, query***] the [*audit rules and audit records*] to [*authorized administrative users*].

FMT_MTD.1.1(5) The TSF shall restrict the ability to [***change_default, modify, delete***] the [*maximum quotas*] to [*authorized administrative users*].

5.1.4.5 FMT_REV.1 Revocation

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the [***users, objects***] within the TSC to [*authorized users (only for the database objects they own or database objects for which they have been granted database object access rights allowing them to revoke security attributes)*].

FMT_REV.1.2 The TSF shall enforce the rules [*revocation of database object access rights shall affect all subsequent attempts to access the database object*].

5.1.4.6 FMT_SMF.1 Specification of management functions

Hierarchical to: No other components.

Dependencies: No dependencies

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [*beginning and ending the audit function, selection of the audited events, review of audit data, management of database users and authentication data, management of security roles, and management of maximum quotas*].

5.1.4.7 FMT_SMR.1 Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1(1) Timing of identification

FMT_SMR.1.1 The TSF shall maintain the roles [

a) authorized administrative users (DBC, database administrators, security administrator),

b) authorized user,

c) security roles as defined by an authorized administrative user, and

d) users authorized to modify their own authentication data].

Note: there is a difference in terminology between CC Part 2 and the Teradata Database regarding the use of the word "role" in FMT_SMR.1.1. The first usage (e.g., a), and b) above), which is part of the CC Part 2 requirement, refers generally to types of database users that can be created within the TSF. The second usage (e.g., c) above) specifically refers to Teradata Database security roles that can be created and granted to database users.

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

5.1.5 Protection of the TSF (FPT)

5.1.5.1 FPT_RVM.1(1) Non-bypassability of the TSP

Hierarchical to: No other components.

Dependencies: No dependencies

FPT_RVM.1.1(1) The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.1.5.2 FPT_SEP_(EXP).1 TSF domain separation

Hierarchical to: No other components.

Dependencies: No dependencies

FPT_SEP_(EXP).1.1 The TSF shall maintain a security domain that protects it from interference and tampering by untrusted subjects initiating actions through its own TSFI.

FPT_SEP_(EXP).1.2 The TSF shall enforce separation between the security domains of subjects in the TOE Scope of Control.

5.1.6 Resource Utilization (FRU)

5.1.6.1 FRU_RSA.1 Maximum quotas

Hierarchical to: No other components.

Dependencies: No dependencies

FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following resources: [*permanent database space allocation, temporary database space allocation, and spool database space usage for a specified job*] that [***individual user***] can use [***simultaneously***].

5.1.7 TOE Access (FTA)

5.1.7.1 FTA_TSE.1 TOE session establishment

Hierarchical to: No other components.

Dependencies: No dependencies

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on [*user identity and hostid*].

5.2 IT ENVIRONMENT SECURITY FUNCTIONAL REQUIREMENTS

This section identifies the security functional requirements that have been levied onto the IT environment that must be implemented in order for the TOE to enforce its' stated functional claims.

The following table provides a summary of the security functional requirements that must be implemented by the IT environment.

Table 5-3 IT Environment Security Functional Requirements

Security Functional Class	Security Functional Component
Identification and Authentication (FIA)	FIA_UAU.1(2) Timing of authentication
	FIA_UID.1(2) Timing of identification
Protection of the TSF (FPT)	FPT_RVM.1(2) Non-bypassability of the TSP
	FPT_SEP_(ENV).1 TSF Domain separation
	FPT_STM.1 Reliable time stamps

The following subsections present the security functional requirements for the IT environment.

5.2.1 Identification and Authentication (FIA)

5.2.1.1 FIA_UAU.1(2) Timing of authentication

Hierarchical to: No other components.

Dependencies: FIA_UID.1(2) Timing of identification

FIA_UAU.1.1(2) The **IT environment** shall allow [*no actions*] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2(2) The **IT environment** shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.2.1.2 FIA_UID.1(2) Timing of identification

Hierarchical to: No other components.

Dependencies: No dependencies

FIA_UID.1.1(2) The **IT environment** shall allow [*no actions*] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2(2) The **IT environment** shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.2.2 Protection of the TSF (FPT)

5.2.2.1 FPT_RVM.1(2) Non-bypassability of the TSP

Hierarchical to: No other components.

Dependencies: No dependencies

FPT_RVM.1.1(2) The **IT environment** shall ensure that **IT environment security policy** enforcement functions are invoked and succeed before each function within the **IT environment's scope of control** is allowed to proceed.

5.2.2.2 FPT_SEP_(ENV).1 TSF Domain Separation

Hierarchical to: No other components.

Dependencies: No dependencies

FPT_SEP_(ENV).1.1 The IT Environment shall provide hardware that provides virtual memory management and at least two execution rings for executing software.

5.2.2.3 FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

Dependencies: No dependencies

FPT_STM.1.1 The **IT Environment** shall be able to provide reliable time-stamps for its own use and **for use by the TOE**.

5.3 TOE SECURITY ASSURANCE REQUIREMENTS

This section identifies the security assurance requirements that are met by the TOE. These assurance requirements conform to the CC Part 3 requirements for EAL4 augmented with ALC_FLR.3 and are identified in the following table.

Table 5-4 TOE Security Assurance Requirements

Security Assurance Class	Security Assurance Component
Configuration management (ACM)	ACM_AUT.1 Partial CM automation
	ACM_CAP.4: Generation support and acceptance procedures
	ACM_SCP.2: Problem tracking CM coverage
Delivery and operation (ADO)	ADO_DEL.2 Detection of modification
	ADO_IGS.1 Installation, generation, and start-up procedures
Development (ADV)	ADV_FSP.2 Fully defined external interfaces
	ADV_HLD.2: Security enforcing high-level design
	ADV_IMP.1: Subset of the implementation of the TSF
	ADV_LLD.1: Descriptive low-level design
	ADV_RCR.1: Informal correspondence demonstration
	ADV_SPM.1: Informal TOE security policy model
Guidance documents (AGD)	AGD_ADM.1 Administrator guidance
	AGD_USR.1 User guidance
Life cycle support (ALC)	ALC_DVS.1: Identification of security measures
	ALC_FLR.3: Systematic flaw remediation
	ALC_LCD.1: Developer defined life-cycle model
	ALC_TAT.1: Well-defined development tools
Tests (ATE)	ATE_COV.2: Analysis of coverage
	ATE_DPT.1: Testing: high-level design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
Vulnerability assessment (AVA)	AVA_MSU.2: Validation of analysis
	AVA_SOF.1 Strength of TOE security function evaluation
	AVA_VLA.2: Independent vulnerability analysis

The following subsections present the security assurance requirements for the TOE.

5.3.1 Configuration management (ACM)

5.3.1.1 Partial CM automation (ACM_AUT.1)

Dependencies: ACM_CAP.3 Authorization controls

ACM_AUT.1.1D The developer shall use a CM system.

ACM_AUT.1.2D The developer shall provide a CM plan.

ACM_AUT.1.1C The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation.

ACM_AUT.1.2C The CM system shall provide an automated means to support the generation of the TOE.

ACM_AUT.1.3C The CM plan shall describe the automated tools used in the CM system.

ACM_AUT.1.4C The CM plan shall describe how the automated tools are used in the CM system.

ACM_AUT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.1.2 Generation support and acceptance procedures (ACM_CAP.4)

Dependencies: ALC_DVS.1 Identification of security measures

ACM_CAP.4.1D The developer shall provide a reference for the TOE.

ACM_CAP.4.2D The developer shall use a CM system.

ACM_CAP.4.3D The developer shall provide CM documentation.

ACM_CAP.4.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.4.2C The TOE shall be labeled with its reference.

ACM_CAP.4.3C The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.

ACM_CAP.4.4C The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.4.5C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.4.6C The CM documentation shall describe the method used to uniquely identify the configuration items that comprise the TOE.

ACM_CAP.4.7C The CM system shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.4.8C The CM plan shall describe how the CM system is used.

ACM_CAP.4.9C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

ACM_CAP.4.10C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.4.11C The CM system shall provide measures such that only authorized changes are made to the configuration items.

ACM_CAP.4.12C The CM system shall support the generation of the TOE.

ACM_CAP.4.13C The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

ACM_CAP.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.1.3 Problem tracking CM coverage (ACM_SCP.2)

Dependencies: ACM_CAP.3 Authorization controls

ACM_SCP.2.1D The developer shall provide a list of configuration items for the TOE.

ACM_SCP.2.1C The list of configuration items shall include the following: implementation representation; security flaws; and the evaluation evidence required by the assurance components in the ST.

ACM_SCP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2 Delivery and operation (ADO)

5.3.2.1 Detection of modification (ADO_DEL.2)

Dependencies: ACM_CAP.3 Authorization controls

ADO_DEL.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.2.2D The developer shall use the delivery procedures.

ADO_DEL.2.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

ADO_DEL.2.2C The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO_DEL.2.3C The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

ADO_DEL.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2.2 Installation, generation, and start-up procedures (ADO_IGS.1)

Dependencies: AGD_ADM.1 Administrator guidance

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1C The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1.2E The evaluator shall determine that the installation, generation, and start up procedures result in a secure configuration.

5.3.3 Development (ADV)

5.3.3.1 Fully defined external interfaces (ADV_FSP.2)

Dependencies: ADV_RCR.1 Informal correspondence demonstration

ADV_FSP.2.1D The developer shall provide a functional specification.

ADV_FSP.2.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.2.2C The functional specification shall be internally consistent.

ADV_FSP.2.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.

ADV_FSP.2.4C The functional specification shall completely represent the TSF.

ADV_FSP.2.5C The functional specification shall include rationale that the TSF is completely represented.

ADV_FSP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.2 Security enforcing high-level design (ADV_HLD.2)

Dependencies: ADV_FSP.1 Informal functional specification
ADV_RCR.1 Informal correspondence demonstration

ADV_HLD.2.1D The developer shall provide the high-level design of the TSF.

ADV_HLD.2.1C The presentation of the high-level design shall be informal.

ADV_HLD.2.2C The high-level design shall be internally consistent.

ADV_HLD.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.2.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_HLD.2.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

ADV_HLD.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD.2.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.3 Subset of the implementation of the TSF (ADV_IMP.1)

Dependencies: ADV_LLD.1 Descriptive low-level design
ADV_RCR.1 Informal correspondence demonstration

ALC_TAT.1 Well-defined development tools

ADV_IMP.1.1D The developer shall provide the implementation representation for a selected subset of the TSF.

ADV_IMP.1.1C The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.1.2C The implementation representation shall be internally consistent.

ADV_IMP.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_IMP.1.2E The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.4 Descriptive low-level design (ADV_LLD.1)

Dependencies: ADV_HLD.2 Security enforcing high-level design
ADV_RCR.1 Informal correspondence demonstration

ADV_LLD.1.1D The developer shall provide the low-level design of the TSF.

ADV_LLD.1.1C The presentation of the low-level design shall be informal.

ADV_LLD.1.2C The low-level design shall be internally consistent.

ADV_LLD.1.3C The low-level design shall describe the TSF in terms of modules.

ADV_LLD.1.4C The low-level design shall describe the purpose of each module.

ADV_LLD.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV_LLD.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV_LLD.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV_LLD.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV_LLD.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_LLD.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

ADV_LLD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_LLD.1.2E The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.5 Informal correspondence demonstration (ADV_RCR.1)

Dependencies: No dependencies.

ADV_RCR.1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

ADV_RCR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.6 Informal TOE security policy model (ADV_SPM.1)

Dependencies: ADV_FSP.1 Informal functional specification

ADV_SPM.1.1D The developer shall provide a TSP model.

ADV_SPM.1.2D The developer shall demonstrate correspondence between the functional specification and the TSP model.

ADV_SPM.1.1C The TSP model shall be informal.

ADV_SPM.1.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

ADV_SPM.1.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

ADV_SPM.1.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

ADV_SPM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Guidance documents (AGD)

5.3.4.1 Administrator guidance (AGD_ADM.1)

Dependencies: ADV_FSP.1 Informal functional specification

AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C The administrator guidance shall describe each type of security relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

AGD_ADM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.2 User guidance (AGD_USR.1)

Dependencies: ADV_FSP.1 Informal functional specification

AGD_USR.1.1D The developer shall provide user guidance.

AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

AGD_USR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5 Life cycle support (ALC)

5.3.5.1 Identification of security measures (ALC_DVS.1)

Dependencies: No dependencies.

ALC_DVS.1.1D The developer shall produce development security documentation.

ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.1.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

ALC_DVS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1.2E The evaluator shall confirm that the security measures are being applied.

5.3.5.2 Systematic flaw remediation (ALC_FLR.3)

Dependencies: No dependencies.

ALC_FLR.3.1D The developer shall provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.3.2D The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.3.3D The developer shall provide flaw remediation guidance addressed to TOE users.

ALC_FLR.3.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.3.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.3.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.3.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.3.5C The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.3.6C The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC_FLR.3.7C The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.3.8C The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

ALC_FLR.3.9C The flaw remediation procedures shall include a procedure requiring timely responses for the automatic distribution of security flaw reports and the associated corrections to registered users who might be affected by the security flaw.

ALC_FLR.3.10C The flaw remediation guidance shall describe a means by which TOE users may register with the developer, to be eligible to receive security flaw reports and corrections.

ALC_FLR.3.11C The flaw remediation guidance shall identify the specific points of contact for all reports and enquiries about security issues involving the TOE.

ALC_FLR.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5.3 Developer defined life-cycle model (ALC_LCD.1)

Dependencies: No dependencies.

ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2D The developer shall provide life-cycle definition documentation.

ALC_LCD.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

ALC_LCD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5.4 Well-defined development tools (ALC_TAT.1)

Dependencies: ADV_IMP.1 Subset of the implementation of the TSF

ALC_TAT.1.1D The developer shall identify the development tools being used for the TOE.

ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of the development tools.

ALC_TAT.1.1C All development tools used for implementation shall be well-defined.

ALC_TAT.1.2C The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.1.3C The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

ALC_TAT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6 Tests (ATE)

5.3.6.1 Analysis of coverage (ATE_COV.2)

Dependencies: ADV_FSP.1 Informal functional specification
ATE_FUN.1 Functional testing

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

ATE_COV.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6.2 Testing: high-level design (ATE_DPT.1)

Dependencies: ADV_HLD.1 Descriptive high-level design
ATE_FUN.1 Functional testing

ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

ATE_DPT.1.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

ATE_DPT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6.3 Functional testing (ATE_FUN.1)

Dependencies: No dependencies.

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

ATE_FUN.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.6.4 Independent testing - sample (ATE_IND.2)

Dependencies: ADV_FSP.1 Informal functional specification
AGD_ADM.1 Administrator guidance
AGD_USR.1 User guidance
ATE_FUN.1 Functional testing

ATE_IND.2.1D The developer shall provide the TOE for testing.

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer’s functional testing of the TSF.

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.3.7 Vulnerability assessment (AVA)

5.3.7.1 Validation of analysis (AVA_MSU.2)

Dependencies: ADO_IGS.1 Installation, generation, and start-up procedures
ADV_FSP.1 Informal functional specification
AGD_ADM.1 Administrator guidance
AGD_USR.1 User guidance

AVA_MSU.2.1D The developer shall provide guidance documentation.

AVA_MSU.2.2D The developer shall document an analysis of the guidance documentation.

AVA_MSU.2.1C The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.2.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.2.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.2.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.2.5C The analysis documentation shall demonstrate that the guidance documentation is complete.

AVA_MSU.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_MSU.2.2E The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.2.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA_MSU.2.4E The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

5.3.7.2 Strength of TOE security function evaluation (AVA_SOF.1)

Dependencies: ADV_FSP.1 Informal functional specification
ADV_HLD.1 Descriptive high-level design

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the ST.

AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the ST.

AVA_SOF.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2E The evaluator shall confirm that the strength claims are correct.

5.3.7.3 Independent vulnerability analysis (AVA_VLA.2)

Dependencies: ADV_FSP.1 Informal functional specification
ADV_HLD.2 Security enforcing high-level design

ADV_IMP.1 Subset of the implementation of the TSF
ADV_LLD.1 Descriptive low-level design
AGD_ADM.1 Administrator guidance
AGD_USR.1 User guidance

AVA_VLA.2.1D The developer shall perform a vulnerability analysis.

AVA_VLA.2.2D The developer shall provide vulnerability analysis documentation.

AVA_VLA.2.1C The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.

AVA_VLA.2.2C The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.

AVA_VLA.2.3C The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.2.4C The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

AVA_VLA.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.2.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA_VLA.2.3E The evaluator shall perform an independent vulnerability analysis.

AVA_VLA.2.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.

AVA_VLA.2.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a low attack potential.

6. TOE SUMMARY SPECIFICATION

This chapter describes the high-level specification of each TOE Security Function (TSF) that contributes to satisfaction of the SFRs presented in Chapter 5. It also details the Assurance Measures applied to ensure the correct implementation of the SFRs.

6.1 TOE SECURITY FUNCTIONS

Each of the following subsections describes a security function of the Teradata Database. For each security function, details are provided that substantiate how the Teradata Database meets the security function, and ensures that no function can be subverted or bypassed without being traced. At the end of each subsection, the SFRs that are satisfied by the TSF are listed, and when it provides added clarity, short additional details specific to the TSF are provided.

6.1.1 TOE Access

Users are identified using a username. Also, a network interface through which client systems connect to the Teradata Database must have its own unique identifier known as a *hostid*. The database grants logon permission to all users from all *hostid*s by default. However, an authorized administrative user can control which users have access to the database by granting or revoking logons for specific usernames on specific *hostid*s. Therefore, it is possible to deny session establishment based upon user identity and *hostid*.

The TOE Access function satisfies the following security functional requirements:

- FTA_TSE.1 TOE session establishment – The TOE satisfies this requirement by allowing the establishment of a session to be denied based upon the user's identity and *hostid.*

6.1.2 Identification and Authentication

When a user attempts to logon to the Teradata Database, a virtual circuit is established to facilitate the transfer of identification and authentication information between the client and the database and to facilitate the transfer of error messages - such as notification of an expired password or authentication failure - to the client. These are the only TSF-mediated actions performed on behalf of a user prior to the user being identified and authenticated.

The data dictionary maintains a unique set of security attributes for each user including a username and password. Optionally, these security attributes can include a default role and profile.

The Teradata Database provides several configurable controls related to password authentication. The system default password controls are maintained in the data dictionary. Optionally, an authorized administrative user may assign a profile to a user that specifies a different set of password controls. If a user has a profile assigned, then any password controls specified by the profile override the corresponding system default controls. The password management functions provide a Strength of Function level that meets or exceeds *SOF-basic*. The following configurable controls combine to satisfy the requirements regarding passwords:

- The minimum and maximum number of characters required for a valid password
- The requirement that a valid password contain a combination of upper and lower case characters, at least one numeric character, and at least one special character
- The requirement that a valid password may not contain the user's username
- The maximum number of days to elapse before a password expires
- The minimum number of days to elapse before a previously used password may be re-used

Note: Appendix B of the Teradata Database Security Administration reference manual provides guidelines that must be followed "to operate the system at a level of security equivalent to the Common Criteria evaluated configuration." This guidance requires that the default password control policy be configured as follows:

- minimum number of characters required in a valid password string - PasswordMinChar = 8 characters
- maximum number of characters allowed in a valid password string - PasswordMaxChar = 30 characters
- digits required in password - PasswordDigits = 'r'
- alpha and special characters required in password - PasswordSpecChar = 'r'
- username not allowed in password - PasswordSpecChar = 'r'
- number of erroneous sequential logon attempts a user is allowed before the user is locked to further logon attempts – MaxLogonAttempts = 3 attempts
- user lock time duration after the user has exceeded the maximum number of logon attempts - LockedUserExpire = 5 minutes
- time span during which a password is valid - ExpirePassword = 90 days
- time span during which a password may not be reused - PasswordReuse = 270 days

To logon to the Teradata Database, a user provides a username and password. The database verifies that the username and password are valid by comparing them to the corresponding security attributes stored in the data dictionary. If either the username or password is not valid, then the logon will fail and no TSF-mediated actions will be performed on behalf of the user.

If the user exceeds the maximum number of failed logon attempts, then the database will prevent further logon attempts by the user until the applicable password lockout time has elapsed. The guidance provided to the security administrator recommends configuring the password lockout time control such that the user remains locked until the security administrator explicitly unlocks the user.

Upon successful identification and authentication of a user, the Teradata Database establishes a binding between the user identity and the established database session. When a session is established, the session's active role is set to the default role indicated by the user's security attributes if a default role has been granted by an authorized administrative user. During a session, a user may change his or her password in accordance with the password control policy. Also during a session, a user may set the active role to any other role, or ALL roles, that have been granted to the user by an authorized administrative user.

The Identification and Authentication security function satisfies the following security functional requirements:

- FIA_AFL.1 Authentication failure handling – The TOE satisfies this requirement by allowing a security administrator to define the maximum number of failed login attempts allowed before the user account is locked.
- FIA_ATD.1 User attribute definition – The TOE satisfies this requirement by maintaining an association between users, passwords roles, and profiles.
- FIA_SOS.1 Verification of secrets – The TOE satisfies this requirement by allowing a security administrator to define rules required for construction of a valid password.
- FIA_UAU.1(1) Timing of authentication – The TOE satisfies this requirement by ensuring that, after establishment of a virtual circuit, a user is properly authenticated before allowing any other TSF-mediated actions on behalf of that user.
- FIA_UID.1(1) Timing of identification – The TOE satisfies this requirement by ensuring that, after establishment of a virtual circuit, a user is properly identified before allowing any other TSF-mediated actions on behalf of that user.
- FIA_USB.1 User-subject binding – The TOE satisfies this requirement by associating the user's security attributes with the user's session.

6.1.3 User Data Protection

The parser module is directly responsible for discretionary access control (DAC). It is responsible for both generating rows in the system access rights table that give a user the right to access an object and for checking of those access rights on subsequent execution of SQL statements.

All access to database objects subject to the DAC policy is controlled using access rights. Database objects include databases, users, tables, views, macros, and stored procedures. Access rights are maintained through the data dictionary for which access is similarly controlled.

Upon initial installation of the Teradata Database, it has only one user. This user is called user `DBC` and will own all other databases and users in the system. User `DBC` also has access rights on all objects within the database with the exception of `CREATE PROCEDURE` and `EXECUTE PROCEDURE`. Typically, administrative users are created under user `DBC` and are granted access rights for creating and managing all other users, databases, and objects.

The Teradata Database supports three types of access rights. Implicit rights (ownership rights) are implicitly granted to the immediate owner of a database or database object. Automatic rights include all rights on a database, user, or object and are granted automatically by the system to the creator of a database, user, or object, and to a newly created user or database. Explicit rights are granted by any user having the `WITH GRANT OPTION` privilege for that right. The database ensures that the requestor has the appropriate access rights before access to a database object is allowed.

Roles define access rights on database objects for groups of users. An authorized administrative user can assign one or more roles to a user (including a default role). A user who is a member of a role can access all the objects for which the role has access rights. Users can switch from the default role to any other role for which they are a member.

The DAC policy for object access is based on user identities, access rights, and active roles. The following enforceable rules combine to control access to database objects (e.g., databases, users, views, macros, stored procedures, and functions) and the operations that can be performed on these objects:

- Every object created in the database is uniquely identified and the TOE correctly resolves all references to an object.
- The TOE enforces Discretionary Access Control (DAC) on objects based on the following user attributes: (a) identity of the user associated with the session, (b) access rights associated with the user, and (c) access rights associated with any roles active for the user's session.
- The TOE enforces Discretionary Access Control (DAC) on objects based on the following object attributes: (a) the identity of the owner of the object, (b) the object rights granted on the object, and (c) any security policies in force for the object.
- An access right is effective in a user session only if: (a) the access right was granted directly and has not been revoked, (b) the access right was granted indirectly (e.g., implicitly, automatically, or as an '`ALL USERS`' privilege) and has not been revoked, or (c) the access right was granted to the user via membership in a role and has not been revoked from the role, and the role is active in the current session.

The TOE enforces the following rules to determine if access by a subject to a database object is allowed:

- If the user associated with the database session is the owner of the database object, then access to the database object is allowed.
- If the required access right on the database object (or the containing database) has been explicitly or implicitly granted to the user associated with the database session, then access to the database object is allowed.
- If a role is active within the session and the required access right on the database object (or the containing database) has been explicitly granted to the role associated with the database session or to a role nested within the role associated with the database session, then access to the database object is allowed. (A nested role is a role that has been granted to another role. Roles can only be nested one level deep.)
- If the required access right on the database object (or the containing database) has been explicitly granted to `PUBLIC`, then access to the database object is allowed.

The TOE protects against inappropriate reuse of any resource associated with an allocated database object by ensuring that any previous information content associated with that resource is unavailable by ensuring that rows written to disk will overlay all of the storage allocated to the row and by insuring that only rows or the columns extracted from rows are returned as part of a result set. Also, the TOE allows no access to any object once the object has been deleted or dropped.

The User Data Protection security function satisfies the following security functional requirements:

- FDP_ACC.1 Subset access control – The TOE satisfies this requirement by associating access rights with all operations that can be performed on database objects and requiring that a user have the appropriate right in order to perform the corresponding operation.
- FDP_ACF.1 Security attribute based access control – The TOE satisfies this requirement by enforcing a discretionary access control policy based upon user identity and access rights associated with the user, active roles, and database objects.
- FDP_RIP.1 Subset residual information protection – The TOE satisfies this requirement by ensuring that no previous information content associated with a database object is available when the object is re-used.

6.1.4 Security Audit

The parser module is directly responsible for access logging. (The term "access logging" is used as opposed to "audit" since auditing is done based upon accesses to database objects.)

The following steps are required to enable and manage the access logging facility:

- A database initialization program (DIP) script must be run to create the special access log rule macro.
- The database must be reset to initialize the logging software.

By default, only user `DBC` has the rights to control the access logging facility. The rights to control the access logging facility are determined by the right to `EXECUTE` the access log rule macro. User `DBC` may grant this right to an authorized security administrator (or other administrative users).

An authorized administrative user can perform the following actions to control the monitoring of access rights checks performed by Teradata Database:

- Begin logging auditable events, including specifying the rules that determine which accesses are logged
- End logging auditable events
- Query the rules that are used to determine which accesses are logged
- Query access log table records containing auditable events
- Purge aged records from the access log table

System event log records are used to indicate startup and shutdown of the auditing facility.

Rules can be specified to audit events based upon one, all, or any combination of the following items:

- Type of access
- Frequency of access
- Action requested
- Name of the requesting user
- Objects referenced
- Success or failure of the access

The Teradata Database will record the following information into each access log record that is generated, provided that a rule exists to indicate that the information should be recorded: Date and time on which the event was logged; username of the user for whom the log entry was made; session number assigned to the user's session; result code indicating whether the access request was granted or denied; the type of access right for which the check that generated the log entry was performed; the name of the database object for which the log entry was made; the text of the statement that caused the access right check for which the log entry was made.

All audit logs are maintained in protected tables within the data dictionary. Access to the audit log tables and defined system views is restricted to authorized administrative users. An authorized security administrator may search and sort access log records using SQL statements to query a defined system view. As with any SQL query, a result set can be sorted by any of the view's columns, which include date, time, user name, result, object name, etc.

An authorized administrative user may purge aged records from the access log using a defined system view. The view contains logic to ensure that a minimum of 30 days of the most recent logged records are retained.

The Teradata Database automatically audits all successful and failed user logon attempts in the event log. An authorized administrative user may search and sort logon/logoff records using SQL statements to query a defined system view.

Since the audit logs are maintained in protected tables within the data dictionary, the size of the logs is limited only by available permanent space for the user `DBC`. If permanent space for user `DBC` is exhausted, then the database will generate an error log entry and perform a reset.

The Security Audit security function satisfies the following security functional requirements:

- FAU_GEN.1 Audit data generation – The TOE satisfies this requirement by generating the necessary audit records associated with each auditable event and by including the date and time, event type, user identity, result code, and other relevant information in each record.

- FAU_GEN.2 User identity association – The TOE satisfies this requirement by including the associated user identity in each audit record.
- FAU_SAR.1 Audit review – The TOE satisfies this requirement by allowing an authorized security administrator to access audit records using SQL commands.
- FAU_SAR.2 Restricted audit review - The TOE satisfies this requirement by restricting access to audit records to only authorized administrative users.
- FAU_SAR.3 Selectable audit review – The TOE satisfies this requirement by allowing an authorized security administrator to search and sort audit records based upon information contained in the records.
- FAU_SEL.1 Selective audit – The TOE satisfies this requirement by allowing an authorized administrative user to establish rules to determine whether events will be included or excluded from the audit log.
- FAU_STG.1 Protected audit trail storage – The TOE satisfies this requirement by protecting the audit records such that no alterations can be made to the audit records, only an authorized administrative user may have access to the audit records, and only an authorized administrative user may purge aged audit records.

6.1.5 Security Management

The Security Management functions enable authorized users to manage the secure operation of the Teradata Database. Apart from the reserved user ids (`DBC`, `SysAdmin`, `SystemFE`, etc.), the administrator can create new user ids (e.g., `SECADMIN`) with appropriate access rights in order to manage security in a structured manner.

The initial user in the system (`DBC`) has all rights, with the exception of `CREATE PROCEDURE` and `EXECUTE PROCEDURE`. The Teradata Database security features allow for user `DBC` to grant rights associated with security management solely to a security administrator. The designated security administrator typically performs the following duties:

- Establishes and modifies logon rules
- Grants, monitors, and if necessary, revokes access rights
- Defines the users, objects, and SQL functions, if any, to be audited
- Monitors audit logs to detect security incidents and initiate corrective action

Access to database objects is controlled through the use of user ids and associated roles, as explained in the access control security function. Only an authorized administrative user may add, modify, or delete TSF data associated with users and roles. However, other users may be authorized to grant or revoke access rights on database objects based upon ownership rights or other rights which have been explicitly granted to the user.

A user who creates a database or another user becomes the owner and is implicitly granted ownership rights on that space. As the owner of the new space, the user is also automatically granted access rights to anything created in that space. If the new space is a user, the owning user is considered the parent and the newly created user is considered its child. In turn, a child becomes the parent of any new users it creates. A parent may grant itself rights on any objects owned by any of its child users. The immediate owner of an object is the containing user or database. However, the parents in the hierarchy above the containing user or database are also indirect owners of the object. An owner can grant or revoke any right applied to an owned object.

A new user or database is automatically granted all rights on itself, with the exception of the GRANT (WITH GRANT OPTION) and CREATE DATABASE/USER, CREATE PROCEDURE, and EXECUTE PROCEDURE rights. Thus, a newly created user can create tables, views, and macros within its own user space. The automatic right to create objects can be explicitly revoked from a new user by an owner or by the creator of the user. A user can be explicitly granted the right to create databases and other users in its own user space. This right can only be granted by a user who has the GRANT right (WITH GRANT OPTION) and the right to create such users and databases. In the case of stored procedure-related rights, a newly created user gets only the DROP PROCEDURE access right. The rights to create or execute stored procedures are not automatic. These can be explicitly granted to any user by user DBC or by a user having the rights WITH GRANT OPTION. If a user has been granted either the CREATE DATABASE or the CREATE USER right, and subsequently creates a new database or user, Teradata Database automatically grants to that user a series of creator rights on the created space. Similarly, a series of creator rights are automatically granted to the user that creates any database object.

A user may revoke rights on an object from another user, role, or ‘ALL USERS’ only if the user is an owner of the object to which the rights refer, or has the GRANT right (WITH GRANT OPTION), plus all of the rights that are to be revoked on the object. If the object is a view, stored procedure, or macro, the owner of the view or macro must also have the GRANT right (WITH GRANT OPTION), plus all other applicable rights, on the objects referenced by the view, stored procedure, or macro. Implicit ownership rights cannot be revoked. Revoked access rights take effect immediately.

A user must have the CREATE ROLE right to create a role. New users do not implicitly have the CREATE ROLE right. To grant a role, a user must have the WITH ADMIN OPTION privilege on the role. The following users can grant a role to a user or other role:

- User DBC.
- A user who has been granted the specified role WITH ADMIN OPTION. The creator of a role is automatically granted the specified role WITH ADMIN OPTION.
- A user who has an active role to which the specified role was granted WITH ADMIN OPTION. An active role can be a current role or a nested role of a current role.

A grantor does not need to have any rights, including WITH ADMIN OPTION, on the grantee to grant a right to it, whether the grantee is a role or a user.

Rules to configure access logging are managed using the BEGIN LOGGING and END LOGGING SQL statements. By default, only user DBC has the rights to control the access logging facility. The rights to control the access logging facility are determined by the right to

EXECUTE the access log rule macro. User DBC may grant this right to an authorized security administrator.

The Teradata Database enforces maximum quotas and limits on various resources to ensure that those resources are protected from monopolization by any individual database user. Permanent, temporary, and spool space limits are managed by an authorized administrative user through the use of CREATE/MODIFY USER/DATABASE statements.

The Security Management security function satisfies the following security functional requirements:

- FMT_MOF.1 Management of security functions behavior – The TOE satisfies this requirement by restricting the ability to manage the audit function to an authorized administrative user.
- FMT_MSA.1 Management of security attributes – The TOE satisfies this requirement by restricting to only authorized users the ability to modify access rights associated with operations on database objects and granting use of a role.
- FMT_MSA.3 Static attribute initialization – The TOE satisfies this requirement by ensuring that default access rights are granted to database objects and owners whenever an object is created.
- FMT_MTD.1 Management of TSF data – The TOE satisfies this requirement by ensuring that only an authorized administrative user may create or modify TSF data associated with users, roles, the audit function, and maximum quotas.
- FMT_REV.1 Revocation – The TOE satisfies this requirement by ensuring that only authorized users may revoke rights associated with operations on database objects.
- FMT_SMF.1 Specification of management functions – The TOE satisfies this requirement by providing an interface through which an authorized administrative user may manage users, authentication data, security roles, and the audit function, and set maximum quotas.
- FMT_SMR.1 Security roles – The TOE satisfies this requirement by allowing an authorized administrative user to create and maintain roles and to associate users with roles.

6.1.6 Resource Utilization

The Teradata Database enforces maximum quotas and limits on various resources to ensure that those resources are protected from monopolization by any individual database user.

Permanent space is used to store tables, indexes, stored procedures, functions, and permanent journals. Permanent space limits are enforced at the database or user (not table) level. An administrator defines the maximum limit with the PERM parameter of a CREATE/MODIFY USER/DATABASE statement.

Temporary space is used to hold rows of materialized global temporary tables. It is allocated at the database or user level, but not the table level. An authorized administrative user may define the maximum limit with the TEMPORARY parameter of a CREATE/MODIFY USER/DATABASE statement or through a user profile assigned to a user. If a temporary space limit is not explicitly specified for a user or database, then the temporary space limit is inherited from the specification for the immediate owner of the user or database.

Spool space is used to hold the response rows of every query run by a user during a session, to hold intermediate result sets produced during execution of a query, and to hold volatile tables produced during execution of a query. The Teradata Database allocates spool space dynamically only from space that is not being used for permanent or temporary data. An administrator may define the maximum limit with the SPOOL parameter of a CREATE/MODIFY USER/DATABASE statement or through a user profile assigned to a user. If a spool limit is not explicitly specified for a user or database, then the spool limit is inherited from the specification for the immediate owner of the user or database.

The Resource Utilization function satisfies the following security functional requirements:

- FRU_RSA.1 Maximum Quotas – The TOE satisfies this requirement by enforcing maximum space limits to prevent excessive monopolization of resources by any individual database user.

6.1.7 Protection of the TSF

The Teradata Database is designed with well-defined interfaces that ensure that all appropriate security checks are made before access is provided to protected database objects and resources.

The Teradata Database operates as a set of cooperating processes which are managed by the underlying operating system. These processes operate as a trusted parallel application (TPA) such that no interference is allowed by processes associated with any non-TOE entities. Furthermore, the Teradata Database is designed such that its interfaces do not allow unauthorized users access to database resources.

Note that given the defined TOE boundaries, the TOE protection mechanisms could be bypassed through the underlying IT environment and it is assumed that the IT environment provides appropriate protection mechanisms - hence the claim of partial self protection. Requirements are also allocated to the IT Environment because the underlying operating system and hardware are outside the TOE boundary and contribute to the enforcement of domain separation.

The Protection of the TSF security function satisfies the following security functional requirements:

- FPT_RVM.1(1) Non-bypassability of the TSP – The TOE satisfies this requirement by ensuring that all applicable security checks are made by each of its interfaces before allowing access to database objects and resources.
- FPT_SEP_(EXP).1 TSF domain separation – The TOE satisfies this requirement through implementation as a set of protected processes protected from external interference and tampering.

6.2 ASSURANCE MEASURES

The following sections specify the assurance measures of the TOE which are claimed to satisfy the stated assurance requirements. Tables are included that identify the reference documents defining the measures to satisfy the Common Criteria EAL4 augmented with ALC_FLR.3 assurance requirements. All of the references identified in the tables are applicable to the TOE. Some referenced documents may contain further extensions to the referenced TOE version number. For example, V2R6.1.0 indicates Version 2, Major Release 6, Minor Release 1, and Maintenance Level 0. The Maintenance Level identifies modifications were performed on the document for problem isolation. The Rationale column in the tables contains further details to indicate how the measures contribute to the satisfaction of the requirements.

6.2.1 Configuration management (ACM)

Table 6-1 describes the documentation that describes the configuration management assurance measures applied to the TOE.

Table 6-1 Configuration Management Assurance Measures

Requirement	Reference(s)	Rationale
ACM_AUT.1: Partial CM automation	(1) Software Configuration Management (SCM) CMM Practices 541-0001722 (2) ClearCase Labeling and Branching Standards 007-0005448	Ref. (1) is the CM plan, which includes the policies and procedures that oversee the NCR CM system. Ref. (2) provides standards for tying the product's version number to the revision control system.

Requirement	Reference(s)	Rationale
ACM_CAP.4: Configuration items	(1) Software Configuration Management (SCM) CMM Practices 541-0001722	Ref. (1) is the CM plan, which includes the policies and procedures that oversee the NCR CM system. This plan also describes the procedures used to support the generation of the TOE and to accept modified or newly created configuration items as part of the TOE.
	(2) ClearCase Labeling and Branching Standards 007-0005448	Ref. (2) provides standards for tying the product's version number to the revision control system.
	(3) Configuration Item List: • 6.1.0_config_spec.txt • 6.1.0_config_spec_BYNET.txt • 6.1.0_config_spec_PDE.txt • 6.1.0_source.txt • 6.1.0_source_BYNET.txt • 6.1.0_source_PDE.txt	Ref. (3) is the configuration item list that is called for throughout the ACM assurance class, with the first three files being the configuration specifications and the last three files being itemized lists of the source modules.
	(4) Web TRP Quick Reference 541-0002801	Ref. (4) provides a quick reference user's guide to the web-based Technical Review Repository (TRP) system. Additional details are available through the system's on-line Help facility. The TRP system is used for configuration management of design documentation, functional specification, test plans, and other documents pertaining to the development of the TOE.
	(5) Information Engineering Development and Delivery Process 541-0002721	Ref. (5) describes the Information Engineering process used for configuration management of reference manuals that provide user and administration guidance for the TOE.
	(6) DARTS Quick Reference 541-0005154	Ref. (6) describes the Defect and Request Tracking System (DARTS) that is used to maintain information about all reported software flaws in each release of the TOE and status of corrective actions.
ACM_SCP.2: Problem tracking CM coverage	(1) Configuration Item List: • 6.1.0_config_spec.txt • 6.1.0_config_spec_BYNET.txt • 6.1.0_config_spec_PDE.txt • 6.1.0_source.txt • 6.1.0_source_BYNET.txt • 6.1.0_source_PDE.txt	Ref. (1) is the configuration item list that is called for throughout the ACM assurance class, with the first three files being the configuration specifications and the last three files being itemized lists of the source modules.
	(2) Evaluation Evidence Configuration Item List.doc	Ref. (2) provides a list of the evaluation evidence required by the assurance components in the ST.
	(3) Sample DARTS Report: DR Report - V2R6.1.0.xls	Ref. (3) is a report produced by the DARTS system that identifies each Discrepancy Report (DR) written against the TOE. This report demonstrates that information regarding previous and current security flaws and their resolution is maintained.

6.2.2 Delivery and operation (ADO)

Table 6-2 describes the documentation that describes the delivery and operation assurance measures applied to the TOE.

Table 6-2 Delivery and Operation Assurance Measures

Requirement	Reference(s)	Rationale
ADO_DEL.2: Detection of modification	(1) Teradata V2R6.1 Delivery Process 541-0004	Ref. (1) describes the delivery process for Teradata Database V2R6.1.
	(2) Customer Procedures for NCR, W. Columbia Pegasus Logistics Group	Ref. (2) describes the process and procedures used for shipping a system installed with the TOE from staging to a customer.
ADO_IGS.1: Installation, generation, and start-up procedures	(1) Teradata Database Release Summary Release V2R6.1.0 B035-1098-115A	Ref. (1) provides a high-level description of the Teradata Database V2R6.0.x maintenance releases, including Requests for Change (RFCs) and the Discrepancy Reports (DRs) contained in each release that affect documentation.
	(2) Teradata Database Base System Release Definition Release V2R6.1.0 B035-1725-115K	Ref. (2) supports understanding the requirements, dependencies, and support resources for Teradata Database V2R6.1.0.
	(3) Teradata Database for Microsoft Windows Server 2003 (32-bit) Installation Guide Release 6.1 B035-5219-115K	Ref. (3) describes the steps to install the Teradata Database software.
	(4) NCR 540S Node for Microsoft Windows Server 2003 (32-bit) Software Installation Guide Release 6.1 B035-5218-115K	Ref. (4) describes how to install the Microsoft Windows Server 2003 operating system and configure the Teradata Database and associated software on a NCR 540S node.
	(5) Parallel Upgrade Tool (PUT) for Microsoft Windows User Guide Release 3.04.02 B035-5710-115K	Ref. (5) provides instructions for installing/upgrading the PUT software and using PUT to install and configure the Teradata Database.
	(6) Teradata Database Security Administration Release V2R6.1.0 B035-1100-115A	Ref. (6) Appendix B of this document provides important guidelines that must be followed in order to operate the Teradata Database in a secure configuration.

6.2.3 Development (ADV)

Table 6-3 describes the documentation that describes the development assurance measures applied to the TOE.

Table 6-3 Development Assurance Measures

Requirement	Reference(s)	Rationale
ADV_FSP.2: Fully defined external interfaces	(1) Teradata Server EAL4 CC Evaluation Functional Specification 541-0004655 (2) Teradata Server EAL4 CC Evaluation High Level Design 541-0004656	Ref. (1) describes the external interfaces to the security functions in an informal manner. Ref. (2) fully describes all interfaces to the TSF.
ADV_FSP.2: Fully defined external interfaces (cont.)	(3) Teradata Database Database Administration Release V2R6.1 B035-1093-115A	Ref. (3) provides supporting information regarding the creation of account strings used for performance management.
	(4) Teradata Database Security Administration Release V2R6.1 B035-1100-115A	Ref. (4) provides supporting information regarding configuration of password controls, access rights, and management of the audit function.
	(5) Teradata Database Data Dictionary Release V2R6.1 B035-1092-115A	Ref. (5) provides supporting information regarding the structure of data dictionary tables.
	(6) Teradata Database SQL Reference Release V2R6.1 Fundamentals B035-1141-115A Data Definition Statements B035-1144-115A	Ref. (6) explains relational database concepts to the security administrator and other interested users. *Fundamentals* and *Data Definition Statements* also provide references for a security administrator to create users, databases, and tables.
	(7) Teradata Database Utilities Utilities – Volume 1 A-F B035-1102-115A Utilities – Volume 2 G-S B035-1102-115A Utilities – Volume 3 T-Z B035-1102-115A	Ref. (7) provides supporting information regarding database utilities described in the Functional Specification.
	(8) Teradata Database Messages Release V2R6.1 & Teradata Tools and Utilities 08.01.00 B035-1096-115A	Ref. (8) provides supporting information regarding the messages produced by the Gateway.
	(9) Teradata Database Graphical User Interfaces: Database Window and Teradata MultiTool Release V2R6.1 B035-1095-115A	Ref. (9) provides supporting information describing the Database Window (DBW) and screens provided.

Requirement	Reference(s)	Rationale
ADV_FSP.2: Fully defined external interfaces (cont.)	(10) Teradata Database Performance Management Release V2R6.1 B035-1097-115A	Ref. (10) provides supporting information describing the performance monitor utility.
	(11) Teradata Manager User Guide Release 07.01.00 B035-2428-115A	Ref. (11) provides supporting information describing DBA tools used for managing a Teradata Database.
	(12) Teradata Call-Level Interface Version 2 Reference for Network-Attached Systems Release 04.08.01 B035-2418-115A	Ref. (12) describes the library of routines that enable a network-attached application to access data on the Teradata Database, including the error and failure codes returned by the Database. It provides details for the functions introduced in the Functional Specification.
	(13) Teradata FastLoad Reference Release 07.07.00 B035-2411-115A	Ref. (13), (14), and (15) provide supporting information regarding utilities that use the partition types identified in the Functional Specification.
	(14) Teradata MultiLoad Reference Release 07.08.00 B035-2409-115A	
	(15) Teradata FastExport Reference Release 07.08.00 B035-2410-115A	
ADV_HLD.2: Security enforcing high-level design	(1) Teradata Server EAL4 CC Evaluation High Level Design 541-0004656	Ref. (1) fully describes the high level design of the TSF.
	(2) Teradata Server EAL4 CC Evaluation Functional Specification 541-0004655	Ref. (2) describes the major components and subcomponents of the Teradata Database software, and relates those components to the security functions described in the ST.
	(3) Teradata Database Security Administration Release V2R6.1 B035-1100-115A	Ref. (3) provides supporting information describing the discretionary access control policy of the server.
	(4) Teradata Database Graphical User Interfaces: Database Window and Teradata MultiTool Release V2R6.1 B035-1095-115A	Ref. (4) provides supporting information describing commands which can be executed through the supervisor screen of the Database Window (DBW).

Requirement	Reference(s)	Rationale
ADV_HLD.2: Security enforcing high-level design (cont.)	(5) Teradata Database Utilities Utilities – Volume 1 A-F B035-1102-115A Utilities – Volume 2 G-S B035-1102-115A Utilities – Volume 3 T-Z B035-1102-115A	Ref. (5) provides supporting information regarding database utilities that can be started in a command prompt window.
	(6) Teradata Call-Level Interface Version 2 Reference for Channel-Attached Systems Release 06.11.01 B035-2417-115A	Ref. (6) provides supporting information regarding messages transmitted between a channel-attached application and the Gateway subsystem.
	(7) Teradata Call-Level Interface Version 2 Reference for Network-Attached Systems Release 04.08.01 B035-2418-115A	Ref. (7) provides supporting information regarding messages transmitted between a network-attached application and the Gateway subsystem.
	(8) Teradata FastLoad Reference Release 07.07.00 B035-2411-115A	Ref. (8), (9), and (10) provide supporting information regarding utilities that use the partition types identified in the High Level Design.
	(9) Teradata MultiLoad Reference Release 07.08.00 B035-2409-115A	
	(10) Teradata FastExport Reference Release 07.08.00 B035-2410-115A	
	(11) Teradata Archive/Recovery Utility Reference Release 08.01.00 B035-2412-115A	Ref. (11) provides supporting information describing the session's runtime parameter input to the Arcmain client utility.
	(12) Parallel Upgrade Tool (PUT) for Microsoft Windows User Guide Release 3.04.02 B035-5710-115K	Ref. (11) provides supporting information describing the vproc configuration process.
ADV_IMP.1: Subset of the implementation of the TSF	(1) Teradata Database V2R6.1.0.51 Source Code List /vob/gateway/src/gtw/gtwassign.c /vob/dbsv2/src/ses/ses/seslogon.c /vob/dbsv2/src/ses/ses/seslogev.c /vob/dbsv2/src/par/opt/optimize.c /vob/dbsv2/src/par/par/paraccr.c /vob/dbsv2/src/amp/awt/awtmain.c	Ref. (1) is a list of source code modules that provides the implementation representation for a selected subset of the TSF.

Requirement	Reference(s)	Rationale
ADV_LLD.1: Descriptive low-level design	(1) Teradata Server EAL4 CC Evaluation Low Level Design AMP Subsystem 541-0005919	Ref. (1) fully describes the low level design of the AMP subsystem.
	(2) Teradata Server EAL4 CC Evaluation Low Level Design PDE Subsystem 541-0005920	Ref. (2) fully describes the low level design of the PDE subsystem.
	(3) Teradata Server EAL4 CC Evaluation Low Level Design Gateway Subsystem 541-0005921	Ref. (3) fully describes the low level design of the Gateway subsystem.
	(4) Teradata Server EAL4 CC Evaluation Low Level Design Session Control Subsystem 541-0005922	Ref. (4) fully describes the low level design of the Session Control subsystem.
	(5) Teradata Server EAL4 CC Evaluation Low Level Design Parser Subsystem 541-0005923	Ref. (5) fully describes the low level design of the Parser subsystem.
ADV_RCR.1: Informal correspondence demonstration	(1) Teradata Database EAL4 CC Evaluation Representation Correspondence 541-0004678	Ref. (1) provides a mapping of the security functions identified in the Security Target to interfaces and enforcement modules defined in the Functional Specification, High-Level Design, and Low Level Design evidence.
ADV_SPM.1: Informal TOE security policy model	(1) Teradata Server EAL4 CC Evaluation Security Policy Model 541-0006147	Ref. (1) describes the TOE security policy (TSP) model. It demonstrates correspondence between the TSP model and the functional specification.

6.2.4 Guidance documents (AGD)

Table 6-4 describes the documentation that describes the guidance documents assurance measures applied to the TOE.

Table 6-4 Guidance Documents Assurance Measures

Requirement	Reference(s)	Rationale
AGD_ADM.1: Administrator guidance	(1) Teradata Database Database Administration Release V2R6.1 B035-1093-115A	Ref. (1) provides security administrator guidance including the creation, administration, and security of the relational objects. Chapter 3 specifically provides administrator guidance for management of quotas specific to space allocations.

Requirement	Reference(s)	Rationale
AGD_ADM.1: Administrator guidance (cont.)	(2) Teradata Database Security Administration Release V2R6.1 B035-1100-115A	Ref. (2) provides a reference guide for the security administrator in formulating, implementing and auditing a security policy, and explains creation of users, databases, and tables.
	(3) Teradata Database SQL Reference Release V2R6.1 Fundamentals B035-1141-115A Statement and Transaction Processing B035-1142-115A Data Definition Statements B035-1144-115A Data Manipulation Statements B035-1146-115A	Ref. (3) explains relational database concepts to the security administrator and other interested users. Volumes 1 and 4 also provide references for a security administrator to create users, databases, and tables.
	(4) Teradata Database Graphical User Interfaces: Database Window and Teradata MultiTool Release V2R6.1 B035-1095-115A	Ref. (4) introduces and explains the Teradata Database DBW and its commands, which allow system administrators to control the operation of the Teradata Database system.
	(5) Teradata Database Introduction to Teradata Warehouse Release V2R6.1.0 (Teradata Database) Release 8.1 (Teradata Warehouse) B035-1091-115A	Ref. (5) introduces system administration and security
	(6) Teradata Database Database Design Release V2R6.1 B035-1094-115A	Ref. (6) describes database design for a security administrator.
	(7) Teradata Database Utilities Utilities – Volume 1 A-F B035-1102-115A Utilities – Volume 2 G-S B035-1102-115A Utilities – Volume 3 T-Z B035-1102-115A	Ref. (7) consists of three volumes of Teradata Database utility program descriptions. These utilities are used primarily by field engineers, developers, and system administrators
	(8) Teradata Database Data Dictionary Release V2R6.1 B035-1092-115A	Ref. (8) describes the Teradata Database Data Dictionary and contains information about system views that allow administrators to access underlying table information stored in the Teradata Database.

Requirement	Reference(s)	Rationale
AGD_ADM.1: Administrator guidance (cont.)	(9) Teradata Database Performance Management Release V2R6.1 B035-1097-115A (10) Teradata Archive/Recovery Utility Reference Release 08.01.00 B035-2412-115A (11) Teradata Manager User Guide Release 7.01 B035-2428-115A (12) Teradata Index Wizard User Guide Release 1.03.01 B035-2506-115A	Ref. (9) provides guidance to an administrator for tuning the database system performance. Ref. (10) describes a group of products designed for archiving, restoring and recovering databases and tables. Ref. (11) describes an extensive suite of DBA tools for managing a Teradata Database system. Ref. (12) provides guidance to a database administrator to create or identify a workload, perform index analysis for a workload, and verify and apply index recommendations to increase database efficiency and maximize system performance.
AGD_USR.1: User guidance	(1) User Guidance Recap 541-0004679	Ref. (1) was produced to take the place of user guidance. Teradata does not produce separate end user guidance documentation. This document maps requirements to documents/chapters that provide the user guidance.

6.2.5 Life cycle support (ALC)

Table 6-5 describes the documentation that describes the life cycle support assurance measures applied to the TOE.

Table 6-5 Life Cycle Support Assurance Measures

Requirement	Reference(s)	Rationale
ALC_DVS.1: Identification of security measures	(1) Corporate Management Policy Manual Protecting Information within NCR Policy No. 1402 (2) Information Protection Standard Information Protection Baseline Requirements Policy No. 102 (3) Securitas Security Services Facility Entry and Exit Control No. 105	Ref. (1) and (2) describe the NCR corporate policy and standards for the protection of information assets - including those assets that represent the TOE design and implementation. Ref. (3) describes the physical, procedural, personnel, and other security measures that are implemented to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

Requirement	Reference(s)	Rationale
ALC_FLR.3: Systematic flaw remediation	(1) DARTS Quick Reference 541-0005154 (2) Teradata Database Engineering Software DR Process Using DARTS 541-0001057	Ref. (1) and (2) describe the Defect and Request Tracking System (DARTS) and software Discrepancy Report (DR) process that is used to track all reported software flaws in each release of the TOE and to maintain information about the flaw and status of corrective actions.
	(3) Teradata Global Support Center Service Delivery Process Support Center Practices 2.0 950003	Ref. (3) describes the methods and procedures used to communicate with TOE users and to manage the process used for resolution of security flaws.
	(4) Teradata Global Support Center Incident Management Process Support Center Practices 2.0 950000 (5) Teradata Global Support Center Tech Alert Process 950012	Ref. (4) and (5) describe the processes by which TOE users register to receive flaw reports and corrections, the processes through which TOE users report and inquire of suspected security flaws in the TOE, and the processes used to provide timely information regarding flaws, corrections and guidance on corrective actions to TOE users.
	(6) Teradata Support Plan Teradata Customer Services	Ref. (6) is the template for the Support Plan that is provided to each Teradata customer. This plan describes procedures for reporting flaws and the processes used for monitoring and receiving timely responses for the flaw reports and the associated corrections.
ALC_LCD.1: Developer defined life-cycle model	(1) Teradata Research & Development PRP Quick Reference 541-0000017	Ref. (1) describes the Product Realization Process (PRP) used for development and maintenance of the TOE. It includes a life-cycle model that encompasses the procedures, tools and techniques used to develop and maintain the TOE.
	(2) Teradata Research & Development Alternate Development Models 541-0001900	Ref. (2) defines the approved development methodology (life cycle) options for use in development of the TOE.
	(3) Master Integration Plan for Teradata Database Version 2 Release 6.1 541-0004943	Ref. (3) defines the specific life cycle models used for development of the TOE.
ALC_TAT.1: Well-defined development tools	(1) Microsoft Visual Studio Microsoft Corp. Online Reference Manual http://msdn2.microsoft.com/en-us/library/ms269115.aspx	Ref. (1) describes the integrated development environment used for development of the TOE.

6.2.6 Tests (ATE)

Table 6-6 describes the documentation that describes the tests assurance measures applied to the TOE.

Table 6-6 Tests Assurance Measures

Requirement	Reference(s)	Rationale
ATE_COV.2: Analysis of coverage	(1) Teradata Server EAL4 CC Evaluation System Test Overview 541-0004842	Ref. (1) provides an overview of the tests performed by the developer against the functional claims contained in this Security Target.
ATE_DPT.1: Testing: high-level design	(1) Teradata Server EAL4 CC Evaluation System Test Overview 541-0004842	Ref. (1) includes an analysis of the depth of testing of the high -level design.
ATE_FUN.1: Functional testing	(1) Teradata Server EAL4 CC Evaluation System Test Overview 541-0004842 (2) Teradata Server EAL4 CC Evaluation Test Suite 1 541-0004843 (3) Teradata Server EAL4 CC Evaluation Test Suite 2 541-0004844 (4) Teradata Server EAL4 CC Evaluation Test Suite 3 541-0004845 (5) Teradata Server EAL4 CC Evaluation Test Suite 4 541-0004846 (6) Teradata Server EAL4 CC Evaluation Test Suite 5 541-0004847 (7) Teradata Server EAL4 CC Evaluation Test Suite 6 541-0006329	Ref. (1) provides an overview of the tests performed by the developer against the claims contained in this Security Target. Ref. (2), (3), (4), (5), (6), and (7) define the specific procedures, inputs, and outputs used by the developer while executing each test case. These documents also identify the expected and actual results of the security functional testing performed by the developer.

Requirement	Reference(s)	Rationale
ATE_IND.2: Independent testing	(1) Teradata Server EAL4 CC Evaluation System Test Overview 541-0004842 (2) Teradata Server EAL4 CC Evaluation Test Suite 1 541-0004843 (3) Teradata Server EAL4 CC Evaluation Test Suite 2 541-0004844 (4) Teradata Server EAL4 CC Evaluation Test Suite 3 541-0004845 (5) Teradata Server EAL4 CC Evaluation Test Suite 4 541-0004846 (6) Teradata Server EAL4 CC Evaluation Test Suite 5 541-0004847 (7) Teradata Server EAL4 CC Evaluation Test Suite 6 541-0006329	The evaluation team is responsible for reviewing the developer's functional testing. In addition, they are responsible for the documentation of the activities performed during independent testing. As a result, the documents referenced are utilized as inputs to the evaluation team's testing efforts.

6.2.7 Vulnerability assessment (AVA)

Table 6-7 describes the documentation that describes the tests assurance measures applied to the TOE.

Table 6-7 Vulnerability Assessment Assurance Measures

Requirement	Reference(s)	Rationale
AVA_MSU.2: Validation of analysis	(1) Teradata Database EAL4 CC Evaluation Misuse/Guidance Analysis 541-0006423	Ref. (1) describes the analysis performed by the developer to demonstrate that guidance is provided for secure operation in all modes of operation of the TOE.
AVA_SOF.1: Strength of TOE security function evaluation	(1) Teradata Database EAL4 CC Evaluation Strength of Function Analysis 541-0004942	The TOE performs user authentication via a password mechanism. This is the only probabilistic / permutational mechanism provided by the TOE and its implementation meets the requirements for SOF-basic.

Requirement	Reference(s)	Rationale
AVA_VLA.2: Independent vulnerability analysis	(1) Teradata Database EAL4 CC Evaluation Vulnerability Analysis 541-0004834	Ref. (1) describes the activities performed by the developer to identify potential vulnerabilities against the TOE and how they have been mitigated. This includes the identification of vulnerabilities available in the public domain as well as those against the TOE evaluated configuration.

7. PROTECTION PROFILE CLAIMS

This Security Target does not claim conformance to a Protection Profile.

8. RATIONALE

8.1 SECURITY OBJECTIVES RATIONALE

This section shows that all secure usage assumptions and threats are completely covered by the security objectives. In addition, each security objective is demonstrated to counter or address at least one assumption or threat.

Table 8-1 Rationale for TOE Security Objectives

Threats	TOE Security Objectives	Rationale
T.ACCOUNTABILITY	O.AUDIT_GENERATION	O.AUDIT_GENERATION addresses this threat by providing the authorized administrator with the capability of configuring the audit mechanism to record the actions of a specific user, or review the audit trail based on the identity of the user. Additionally, the administrator's ID is recorded when any security relevant change is made to the TOE.
	OE.TIME_STAMPS	OE.TIME_STAMPS plays a role in addressing this threat by requiring the IT Environment to provide a reliable time stamp. The audit mechanism is required to include the current date and time in each audit record. All audit records that include the user ID, will also include the date and time that the event occurred.
	O.TOE_ACCESS	O.TOE_ACCESS addresses this threat by requiring the TOE to identify and authenticate all authorized users prior to allowing any TOE access or any TOE mediated access on behalf of those users.
T.ADMIN_ERROR	O.ADMIN_GUIDANCE	O.ADMIN_GUIDANCE helps to mitigate this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner. Having this guidance helps to reduce the mistakes that an administrator might make that could cause the TOE to be configured in a way that is insecure.
T.AUDIT_COMPROMISE	O.AUDIT_REVIEW	O.AUDIT_REVIEW ensures that the TOE will provide mechanisms to review the audit logs. These requirements will ensure the data is in a suitable manner for the administrator to interpret as well as giving the administrator a way to search and sort within the log to find appropriate data.
	O.AUDIT_STORAGE	O.AUDIT_STORAGE ensures the TOE will provide a secure mechanism for storing and managing the TOE audit log.
	O.MANAGE	O.MANAGE ensures that the TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the audit logs, and restrict these functions and facilities from unauthorized use.

Threats	TOE Security Objectives	Rationale
T.MASQUERADE	O.I_AND_A	O.I_AND_A ensures that the TOE will contain identification and authentication mechanisms for users to login to the TOE.
	O.TOE_ACCESS	O.TOE_ACCESS mitigates this threat by controlling the logical access to the TOE and its resources. By constraining how and when authorized users can access the TOE, and by mandating the type and strength of the authentication mechanism this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this objective provides the administrator the means to control the number of failed login attempts a user can generate before an account is locked out, further reducing the possibility of a user gaining unauthorized access to the TOE.
T.POOR_DESIGN	O.CONFIG_IDENTIFICATION	O.CONFIG_IDENTIFICATION plays a role in countering this threat by requiring the developer to provide control of the changes made to the TOE's design.
	O.DOCUMENTED_DESIGN	O.DOCUMENTED_DESIGN ensures that the design of the TOE is documented, permitting detailed review by evaluators and validators.
	O.VULNERABILITY_ANALYSIS	O.VULNERABILITY_ANALYSIS ensures that the design of the TOE is analyzed for design flaws.
T.POOR.IMPLEMENTATION	O.CONFIG_IDENTIFICATION	O.CONFIG_IDENTIFICATION plays a role in countering this treat by requiring the developer to provide control of the changes made to the TOE's design. Although the previous three objectives help minimize the introduction of errors into the implementation.
	O.PARTIAL_FUNCTIONAL_TEST	O.PARTIAL_FUNCTIONAL_TEST increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification, high-level, and low-level design) will be discovered through testing.
	O.VULNERABILITY_ANALYSIS	O.VULNERABILITY_ANALYSIS helps reduce errors in the implementation that may not be discovered during functional testing. Ambiguous design documentation and the fact that exhaustive testing of the external interfaces is not required may leave bugs in the implementation undiscovered in functional testing.

Threats	TOE Security Objectives	Rationale
T.POOR_TEST	O.DOCUMENTED_DESIGN	O.DOCUMENTED_DESIGN helps to ensure that the TOE's documented design satisfies the security functional requirements. In order to ensure the TOE's design is correctly realized in its implementation, the appropriate level of functional testing of the TOE's security mechanisms must be performed during the evaluation of the TOE.
	O.PARTIAL_FUNCTIONAL_TEST	O.PARTIAL_FUNCTIONAL_TEST increases the likelihood that any errors that do exist in the implementation (with respect to the functional specification, high level, and low-level design) will be discovered through testing.
	O.VULNERABILITY_ANALYSIS	O.VULNERABILITY_ANALYSIS addresses this concern by requiring a vulnerability analysis be performed in conjunction with testing that goes beyond functional testing. This objective provides a measure of confidence that the TOE does not contain security flaws that may not be identified through functional testing. While these testing activities are a necessary activity for successful completion of an evaluation, this testing activity does not address the concern that the TOE continues to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to end users to ensure the TOE's security mechanisms continue to operator correctly once the TOE is fielded.
T.RESIDUAL_DATA	O.RESIDUAL_INFORMATION	O.RESIDUAL_INFORMATION counters this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process.
T.RESOURCE	O.RESOURCE	O.RESOURCE ensures that the TOE provides an authorized administrator with controls to limit the consumption of database resources by an authorized database user.
T.NO_SECADMIN	O.SECADMIN	The TOE has the objective of providing an authorized administrator user for secure administration. This is a separate user from other administrative users that the TOE may provide.

Threats	TOE Security Objectives	Rationale
T.TSF_COMPROMISE	O.RESIDUAL_INFORMATION	O.RESIDUAL_INFORMATION is necessary to mitigate this threat, because even if the security mechanisms do not allow a user to explicitly view TSF data, if TSF data were to inappropriately reside in a resource that was made available to a user, that user would be able to inappropriately view the TSF data.
	O.PARTIAL_SELF_PROTECTION	O.PARTIAL_SELF_PROTECTION ensures the TOE is capable of protecting itself from attack.
	O.MANAGE	O.MANAGE is necessary because an access control policy is specified to control access to TSF data. This objective is used to dictate who is able to view and modify TSF data, as well as the behaviour of TSF functions.
	O.INTERNAL_TOE_DOMAINS	O.INTERNAL_TOE_DOMAINS ensures the TOE will establish separate domains for data belonging to users.
T.UNAUTHORIZED_ACCESS	O.MEDIATE	O.MEDIATE ensures that all accesses to user data are subject to mediation, unless said data has been specifically identifies as public data. The TOE requires successful authentication to the TOE prior to gaining access to any controlled-access content. Lastly, the TSF will ensure that all configured enforcement functions (authentication, access control rules, etc.) must be invoked prior to allowing a user to gain access to TOE or TOE mediated services. The TOE restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc. to the administrator. This feature ensures that no other user can modify the information flow policy to bypass the intended TOE security policy.
T.UNIDENTIFIED_ACTIONS	O.ADMIN_GUIDANCE	The threat of an authorized administrator failing to know about malicious audit events produces the objectives of the authorized administrator having the facilities and knowing how to use them (O.ADMIN_GUIDANCE).
	O.MANAGE	The threat of an authorized administrator failing to know about malicious audit events produces the objectives of the authorized administrator having the capability to use the mechanisms (O.MANAGE) to review audit records.

Table 8-2 Rationale for IT Environmental Objectives

Assumptions	IT Environmental Objectives	Rationale
A.DOMAIN_SEPARATION	OE.DOMAIN_SEPARATION	OE.DOMAIN_SEPARATION ensures the IT environment will provide an isolated domain for the TOE's execution.

Assumptions	IT Environmental Objectives	Rationale
A.I_AND_A	OE.I_AND_A	OE.I_AND_A ensures the IT environment will provide mechanisms for administrators to be authenticated before any database control utilities and other utilities used to manage system resources and I/O interfaces may be used.
A.NO_BYPASS	OE.NO_BYPASS	OE.NO_BYPASS ensures the TOE cannot be bypassed in order to gain unauthorized access of TOE resources.
A.NO_EVIL	OE.NO_EVIL	All authorized administrators are trustworthy individuals, having background investigations commensurate with the level of data being protected, have undergone appropriate admin training, and follow all admin guidance.
A.NO_GENERAL_PURPOSE	OE.NO_GENERAL_PURPOSE	The DBMS server must not include any general-purpose computing or storage capabilities. This will protect the TSF data from malicious processes.
A.PHYSICAL	OE.PHYSICAL	The TOE, the TSF data, and protected user data is assumed to be protected from physical attack (e.g., theft, modification, destruction, or eavesdropping). Physical attack could include unauthorized intruders into the TOE environment, but it does not include physical destructive actions that might be taken by an individual that is authorized to access the TOE environment.
A.RESTRICT_OS_ACCESS	OE.RESTRICT_OS_ACCESS	The underlying operating system running on the DBMS server must include only those user accounts required by authorized Teradata administrators. Restricting access to the operating system protects against tampering by malicious users.
A.ROBUST_ENVIRONMENT	OE.ROBUST_ENVIRONMENT	The TOE shall only be installed in an IT environment that is at least as robust as the TOE. The TOE is basic robustness, therefore, all elements in the environment the TOE depends on for enforcement of its security objectives are also assumed to be basic robustness. These elements could include the operating system, encryption devices, and/or boundary protection devices.
	OE.TRUST_IT	The IT entities in the environment are correctly installed, configured, managed and maintained.
A.SECURE_COMMS	OE.SECURE_COMMS	OE.SECURE_COMMS states that the environment must provide a secure line of communication for transfer of TSF data. This is necessary because access to the TOE may be distributed geographically with users and authorized administrators in different locations. The objective OE.SECURE_COMMS does not necessarily mandate that the communications between the remote user or administrator and the TOE be encrypted.

Assumptions	IT Environmental Objectives	Rationale
A.TIME_STAMPS	OE.TIME_STAMPS	OE.TIME_STAMPS states that the environment will maintain reliable timestamps and those will be used by the TOE to stamp each audit record with a date and time.

8.2 SECURITY REQUIREMENTS RATIONALE

Table 8-3 demonstrates the mapping of Security Requirements to TOE Security Objectives. Rationale for each mapping is included in the table.

Table 8-3 Rationale for TOE Security Requirements

Security Objective	Security Requirements	Rationale
O.ADMIN_GUIDANCE	ADO_DEL.2	ADO_DEL.2 ensures that the administrator is provided documentation that instructs them how to ensure the delivery of the TOE, in whole or in parts, has not been tampered with or corrupted during delivery.
	ADO_IGS.1	ADO_IGS.1 ensures the administrator has the information necessary to install the TOE in the evaluated configuration. Often times a vendor's product contains software that is not part of the TOE and has not been evaluated. The Installation, Generation, and Start-up (IGS) documentation ensures that once the administrator has followed the installation and configuration guidance the result is a TOE in a secure configuration.
	AGD_ADM.1	AGD_ADM.1 mandates the developer provide the administrator with guidance on how to operate the TOE in a secure manner. This includes describing the interfaces the administrator uses in managing the TOE, security parameters that are configurable by the administrator, how to configure the TOE's rule set and the implications of any dependencies of individual rules. The documentation also provides a description of how to setup and review the auditing features of the TOE.
	AGD_USR.1	AGD_USR.1 is intended for non-administrative users, but could be used to provide guidance on security that is common to both administrators and non-administrators.
	AVA_MSU.2	AVA_MSU.2 ensures that the guidance documentation is complete and consistent, and notes all requirements for external security measures.

Security Objective	Security Requirements	Rationale
O.AUDIT_GENERATION	FAU_GEN.1	FAU_GEN.1 defines the set of events that the TOE must be capable of recording. This requirement ensures that the administrator has the ability to audit any security relevant events that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event.
	FAU_GEN.2	FAU_GEN.2 ensures that the audit records associate a user identity with the auditable event. In the case of authorized users, the association is accomplished with the userid.
	FAU_SEL.1	FAU_SEL.1 allows the security administrator to configure which auditable events will be recorded in the audit trail. This provides the administrator with the flexibility in recording only those events that are deemed necessary by site policy, thus reducing the amount of resources consumed by the audit mechanism.
	FIA_USB.1	FIA_USB.1 requires that all subjects that act on behalf of users must have a binding that associates the subjects with a user. This is necessary to be able to associate audit records with user identities.
O.AUDIT_REVIEW	FAU_SAR.1	FAU_SAR.1 requires that only the authorized administrator has the capability to read the audit records which must be presented in a manner suitable for the administrator to interpret them.
	FAU_SAR.2	FAU_SAR.2 prohibits all other users read access of the audit records.
	FAU_SAR.3	FAU_SAR.3 requires the TOE to provide a mechanism for the security administrator to search and sort through the audit records.
O.AUDIT_STORAGE	FAU_STG.1	FAU_STG.1 requires that only the authorized administrator may delete the audit records ensuring that no malicious users may compromise the data stored within the audit records.
	FMT_MTD.1	FMT_MTD.1 allows only the authorized administrator to query the logs and clear the logs.
	FMT_SMF.1	FMT_SMF.1 lists the mechanisms available to the administrator for managing the audit records.
O.CONFIG_IDENTIFICATION	ACM_CAP.4	ACM_CAP.4 addresses this objective by requiring that there be a unique reference for the TOE, and that the TOE is labelled with that reference. It also requires that there be a CM system in place, and that the configuration items that comprise the TOE are uniquely identified. This provides a clear identification of the composition of the TOE.

Security Objective	Security Requirements	Rationale
O.DOCUMENTED_DESIGN	ADV_FSP.2	ADV_FSP.2 requires that the interfaces to the TOE be documented and specified.
	ADV_HLD.2	ADV_HLD.2 requires the high level design of the TOE be documented and specified and that said design be shown to correspond to the interfaces.
	ADV_RCR.1	ADV_RCR.1 requires that there be a correspondence between adjacent layers of the design decomposition.
	ALC_FLR.3	ALC_FLR.3 addresses this objective by requiring that there be a mechanism in place for identifying flaws subsequent to fielding, and for distributing those flaws to entities operating the system.
O.I_AND_A	FIA_AFL.1	FIA_AFL.1 requires the TOE to provide a mechanism for the security administrator to restrict the number of unsuccessful authentication attempts allowed before a user account is locked.
	FIA_ATD.1	FIA_ATD.1 requires the TOE to maintain user identities and passwords belonging to individual users.
	FIA_SOS.1	FIA_SOS.1 requires the TOE to enforce rules requiring the construction of strong passwords and to prevent brute-force password attacks.

	FIA_UAU.1(1)
	FIA_UID.1(1)
		FIA_UAU.1(1) and FIA_UID.1(1) require the TOE to successfully identify and authenticate a user before establishing a session on behalf of the user.

	FIA_USB.1	FIA_USB.1 requires that all subjects that act on behalf of users must have a binding that associates the subjects with a user uniquely.
O.INTERNAL_TOE_DOMAINS	FPT_SEP_(EXP).1	FPT_SEP_(EXP).1 requires the TOE to maintain a separate domain for its own execution separate from other processes.
O.MANAGE	FMT_MOF.1	FMT_MOF.1 requires that the ability to use particular TOE capabilities be restricted to the administrator.
	FMT_MSA.1	FMT_MSA.1 requires that the ability to perform operations on security attributes be restricted to particular roles.
	FMT_MTD.1	FMT_MTD.1 requires that the ability to manipulate TOE content is restricted to administrators.
	FMT_REV.1	FMT_REV.1 restricts the ability to revoke attributes to authorized users.
	FMT_SMF.1	FMT_SMF.1 identifies the management functions that are available to the authorized administrator.
	FMT_SMR.1	FMT_SMR.1 defines the specific security roles to be supported.

Security Objective	Security Requirements	Rationale
O.MEDIATE		The FDP requirements were chosen to define the policies, the subjects, objects, and operations for how and when mediation takes place in the TOE.
	FDP_ACC.1	FDP_ACC.1 defines the Access Control policy that will be enforced on a list of subjects acting on the behalf of users attempting to gain access to a list of named objects. All the operation between subject and object covered are defined by the TOE's policy.
	FDP_ACF.1	FDP_ACF.1 defines the security attribute used to provide access control to objects based on the TOE's access control policy.
O.PARTIAL_FUNCTIONAL_TEST	ATE_COV.2	ATE_COV.2 requires that there be a correspondence between the tests in the test documentation and the TSF as described in the functional specification.
	ATE_FUN.1	ATE_FUN.1 requires that the developer provide test documentation for the TOE, including test plans, test procedure descriptions, expected test results, and actual test results. These need to identify the functions tested, the tests performed, and test scenarios. There require that the developer run those tests, and show that the expected results were achieved.
	ATE_IND.2	ATE_IND.2 requires that the evaluators test a subset of the TSF to confirm correct operation, on an equivalent set of resources to those used by the developer for testing. These sets should include a subset of the developer run tests.
O.PARTIAL_SELF_PROTECTION	FPT_SEP_(EXP).1	FPT_SEP_(EXP).1 is used to ensure the TSF provides a domain that protects itself from untrusted users. If the TSF cannot protect itself it cannot be relied upon to enforce its security policies.
O.RESIDUAL_INFORMATION	FDP_RIP.1	FDP_RIP.1 is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data.
O.RESOURCE	FRU_RSA.1	FRU_RSA.1 requires that the TOE provide controls to limit the consumption of database resources by authorized database users.
O.SECADMIN	FMT_SMR.1	The TOE will establish, at least, an authorized security administrator user. The authorized security administrator will be given rights to perform certain tasks that other users will not be able to perform. These rights include, but are not limited to, access to audit information and security functions. (FMT_SMR.1)

Security Objective	Security Requirements	Rationale
O.TOE_ACCESS	FIA_ATD.1	FIA_ATD.1 defines the attributes of users, including a userid that is used by the TOE to determine a user's identity and enforce what type of access the user has to the TOE.
	FIA_USB.1	FIA_USB.1 ensures that all subjects that act on behalf of users will have a binding that associates the subjects with a user uniquely.
	FPT_RVM.1(1)	FPT_RVM.1(1) ensures that all applicable security checks are made by each of the TOE interfaces before allowing access to database objects and resources.
	FTA_TSE.1	FTA_TSE.1 allows the TOE to restrict access to the TOE based on certain criteria.
	AVA_SOF.1	AVA_SOF.1 requirement is applied to the password mechanism used by the local administrator. For this TOE, the strength of function specified is basic. This requirement ensures the developer has performed an analysis of the password mechanism to ensure the probability of guessing a local administrator's password would require a high-attack potential, as defined in Annex B of the CEM. This analysis takes into account the password spaces, as well as any feature of the password mechanism that plays a role in limiting the number of failed authentication attempts within a given time period.
O.VULNERABILITY_ANALYSIS	AVA_VLA.21	The AVA_VLA.2 component provides the necessary level of confidence that vulnerabilities do not exist in the TOE that could cause the security policies to be violated. AVA_VLA.1 requires the developer to perform a systematic search for potential vulnerabilities in all the TOE deliverables. For those vulnerabilities that are not eliminated, a rationale must be provided that describes why these vulnerabilities cannot be exploited by a threat agent with a low attack potential, which is in keeping with the desired assurance level of this TOE. As with the functional testing, a key element is this component is that an independent assessment of the completeness of the developer's analysis is made, and more importantly, an independent vulnerability analysis coupled with testing of the TOE is performed. This component provides the confidence that security flaws do not exist in the TOE that could be exploited by a threat agent or moderate (or lower) attack potential to violate the TOE's security policies.

Table 8-4 demonstrates the mapping of Security Requirements to IT Environment Security Objectives. Rationale for each mapping is included in the table.

Table 8-4 Rationale for IT Environment Security Requirements

Security Objective	Security Requirements	Rationale
OE.DOMAIN_SEPARATION	FPT_SEP_(ENV).1	FPT_SEP_(ENV).1 ensures the IT environment will provide the TOE with an isolated domain for its execution.

OE.I_AND_A	FIA_UAU.1(2)
	FIA_UID.1(2)
		FIA_UAU.1(2) and FIA_UID.1(2) requires that the IT environment must identify and authenticate administrators before they are given access to database control utilities and other utilities used to manage system resources and I/O interfaces.

OE.NO_BYPASS	FPT_RVM.1(2)	FPT_RVM.1(2) ensures the TOE cannot be bypassed in order to gain unauthorized access of TOE resources.
OE.NO_EVIL	N/A	This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.
OE.CONFIG	N/A	This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.
OE.NO_GENERAL_PURPOSE	N/A	This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.
OE.PHYSICAL	N/A	This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.
OE.RESTRICT_OS_ACCESS	N/A	This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.
OE.ROBUST_ENVIRONMENT	N/A	This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.
OE.SECURE_COMMS	N/A	This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.
OE.TIME_STAMPS	FPT_STM.1	FPT_STM.1 requires the IT environment to provide reliable time stamps to the TOE to use for audit generation.

Security Objective	Security Requirements	Rationale
OE.TRUST_IT	N/A	This objective does not contain any IT security requirements because it is a non-IT related objective. Thus, the CC does not mandate it map to any requirements.

Table 8-5 demonstrates that all Security Functional Requirements (SFR) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective for which it is intended to satisfy.

Table 8-5 Security Functional Requirements Mapped to Security Objectives

	O.AUDIT_GENERATION	O.AUDIT_REVIEW	O.AUDIT_STORAGE	O.I_AND_A	O.INTERNAL_TOE_DOMAINS	O.MANAGE	O.MEDIATE	O.PARTIAL_SELF_PROTECTION	O.RESIDUAL_INFORMATION	O.RESOURCE	O.SECDAMIN	O.TOE_ACCESS	OE.DOMAIN_SEPARATION	OE.I_AND_A	OE.NO_BYPASS	OE.TIME_STAMPS
FAU_GEN.1	X															
FAU_GEN.2	X															
FAU_SAR.1		X														
FAU_SAR.2		X														
FAU_SAR.3		X														
FAU_SEL.1	X															
FAU_STG.1			X													
FDP_ACC.1							X									
FDP_ACF.1							X									
FDP_RIP.1									X							
FIA_AFL.1				X												
FIA_ATD.1				X								X				
FIA_SOS.1				X												
FIA_UAU.1(1)				X												
FIA_UAU.1(2)														X		
FIA_UID.1(1)				X												
FIA_UID.1(2)														X		
FIA_USB.1	X			X								X				
FMT_MOF.1						X										

	O.AUDIT_GENERATION	O.AUDIT_REVIEW	O.AUDIT_STORAGE	O.I_AND_A	O.INTERNAL_TOE_DOMAINS	O.MANAGE	O.MEDIATE	O.PARTIAL_SELF_PROTECTION	O.RESIDUAL_INFORMATION	O.RESOURCE	O.SECDAMIN	O.TOE_ACCESS	OE.DOMAIN_SEPARATION	OE.I_AND_A	OE.NO_BYPASS	OE.TIME_STAMPS
FMT_MSA.1						X										
FMT_MSA.3						X										
FMT_MTD.1			X			X										
FMT_REV.1						X										
FMT_SMF.1			X			X										
FMT_SMR.1						X					X					
FPT_RVM.1(1)												X				
FPT_RVM.1(2)															X	
FPT_SEP_(EXP).1					X			X								
FPT_SEP_(ENV).1													X			
FPT_STM.1																X
FRU_RSA.1										X						
FTA_TSE.1												X				

8.3 EXPLICIT REQUIREMENTS RATIONALE

Table 8-6 provides the rationale for the inclusion of the explicit functional requirements found in this Security Target.

Table 8-6 Rationale for Explicit Functional Requirements

Explicit Requirement	Rationale
FPT_SEP_(EXP).1	The security functional requirement component FPT_SEP.1 appearing in Part 2 of the Common Criteria does not provide the ability to specify domain separation enforcement functionality for a software only TOE. FPT_SEP_(EXP).1 is explicitly stated to address this issue and to specify a delineation between domain separation functionality provided by a software only TOE (FPT_SEP_EXP.1) and the domain separation functionality relied upon by the underlying operating system and hardware of the IT environment (FPT_SEP_ENV_EXP.1).

Table 8-7 provides the rationale for the inclusion of the explicit environmental requirements found in this Security Target.

Table 8-7 Rationale for Explicit Environmental Requirements

Explicit Requirement	Rationale
FPT_SEP_(ENV).1	The security functional requirement component FPT_SEP.1 appearing in Part 2 of the Common Criteria does not provide the ability to specify domain separation enforcement functionality for a software only TOE. FPT_SEP_(ENV).1 is explicitly stated to address this issue and to specify a delineation between domain separation functionality provided by a software only TOE (FPT_SEP_EXP.1) and the domain separation functionality relied upon by the underlying operating system and hardware of the IT environment (FPT_SEP_ENV_EXP.1).

8.4 TOE SUMMARY SPECIFICATION RATIONALE

Table 8-8 describes the association between the TOE Security Functions and the TOE Security Functional Requirements. This table, in conjunction with rationale provided in Section 6.1, demonstrates that the TOE Security Functional Requirements are satisfied.

Table 8-8 TSF and SFR Mapping

	TOE Access	Identification and Authentication	User Data Protection	Security Audit	Security Management	Resource Utilization	Protection of the TSF
FAU_GEN.1				X			
FAU_GEN.2				X			
FAU_SAR.1				X			
FAU_SAR.2				X			
FAU_SAR.3				X			
FAU_SEL.1				X			
FAU_STG.1				X			
FDP_ACC.1			X				
FDP_ACF.1			X				
FDP_RIP.1			X				
FIA_AFL.1		X					
FIA_ATD.1		X					
FIA_SOS.1		X					
FIA_UAU.1(1)		X					
FIA_UID.1(1)		X					

	TOE Access	Identification and Authentication	User Data Protection	Security Audit	Security Management	Resource Utilization	Protection of the TSF
FIA_USB.1		X					
FMT_MOF.1					X		
FMT_MSA.1					X		
FMT_MSA.3					X		
FMT_MTD.1					X		
FMT_REV.1					X		
FMT_SMF.1					X		
FMT_SMR.1					X		
FPT_RVM.1(1)							X
FPT_SEP_(EXP).1							X
FRU_RSA.1						X	
FTA_TSE.1	X						

Additional rationale for the TOE Summary Specification is defined in Chapter 6, TOE Security Functions.

8.5 TOE ASSURANCE REQUIREMENTS RATIONALE

The TOE stresses assurance through vendor actions that are within the bounds of current best commercial practice. The TOE provides, primarily via review of vendor-supplied evidence, independent confirmation that these actions have been competently performed.

The general level of assurance for the TOE is:

a) Consistent with current best commercial practice for IT development and provides a product that is competitive against non-evaluated products with respect to functionality, performance, cost, and time-to-market, and

b) Meets current constraints on widespread acceptance, by expressing its claims against EAL4 augmented with ALC_FLR.3 from part 3 of the Common Criteria.

8.6 STRENGTH OF FUNCTION RATIONALE

A strength of function analysis is required for security functions in which a qualification of their security behavior can be made using the results of a quantitative or statistical analysis of the security behavior of these mechanisms and the effort required to overcome them. The behavior of the following security functions provided by the Teradata Database may be realized by probabilistic or permutational mechanisms:

- FIA_SOS.1 Verification of Secrets
- FIA_UAU.1 Timing of authentication

The Teradata Database requires that a password be provided in order to authenticate a user prior to establishing a database session. This mechanism meets or exceeds a rating of *SOF-basic* in that the mechanism provides adequate protection against casual breach of TOE security by attackers possessing a low attack potential.

8.7 DEPENDENCIES RATIONALE

Table 8-9 demonstrates that the dependencies that exist based on the security functional requirements included in this Security Target are satisfied.

Table 8-9 Rationale for Satisfaction of SFR Dependencies

Security Functional Requirement	Dependencies	Rationale
TOE Security Functional Requirements		
FAU_GEN.1	FPT_STM.1	Satisfied by FPT_STM.1
FAU_GEN.2	FAU_GEN.1 FIA_UID.1(1)	Satisfied by FAU_GEN.1 and FIA_UID.1(1)
FAU_SAR.1	FAU_GEN.1	Satisfied by FAU_GEN.1
FAU_SAR.2	FAU_SAR.1	Satisfied by FAU_SAR.1
FAU_SAR.3	FAU_SAR.1	Satisfied by FAU_SAR.1
FAU_SEL.1	FAU_GEN.1 FMT_MTD.1	Satisfied by FAU_GEN.1 and FMT_MTD.1
FAU_STG.1	FAU_GEN.1	Satisfied by FAU_GEN.1
FDP_ACC.1	FDP_ACF.1	Satisfied by FDP_ACF.1
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	Satisfied by FDP_ACC.1 and FMT_MSA.3
FDP_RIP.1	None	N/A
FIA_AFL.1	FIA_UAU.1(1)	Satisfied by FIA_UAU.1(1)
FIA_ATD.1	None	N/A
FIA_SOS.1	None	N/A
FIA_UAU.1(1)	FIA_UID.1	Satisfied by FIA_UID.1(1)
FIA_UID.1(1)	None	N/A
FIA_USB.1	FIA_ATD.1	Satisfied by FIA_ATD.1

Security Functional Requirement	Dependencies	Rationale
FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	Satisfied by FMT_SMF.1 and FMT_SMR.1
FMT_MSA.1	FDP_ACC.1 FMT_SMF.1 FMT_SMR.1	Satisfied by FDP_ACC.1. FMT_SMF.1, and FMT_SMR.1
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	Satisfied by FMT_MSA.1 and FMT_SMR.1
FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	Satisfied by FMT_SMF.1 and FMT_SMR.1
FMT_REV.1	FMT_SMR.1	Satisfied by FMT_SMR.1
FMT_SMF.1	None	N/A
FMT_SMR.1	FIA_UID.1(1)	Satisfied by FIA_UID.1(1)
FPT_RVM.1(1)	None	N/A
FPT_SEP_(EXP).1	None	N/A
FRU_RSA.1	None	N/A
FTA_TSE.1	None	N/A
IT Environment Security Functional Requirements		
FIA_UAU.1(2)	FIA_UID.1(2)	Satisfied by FIA_UID.1(2)
FIA_UID.1(2)	None	N/A
FPT_RVM.1(2)	None	N/A
FPT_SEP_(ENV).1	None	N/A
FPT_STM.1	None	N/A

This Security Target also includes security assurance requirements that have dependencies. Since EAL 4 has been adopted in this Security Target and EAL 4 is defined in the Common Criteria, it is assumed that all of the dependencies within that assurance level have been addressed. The only change to the set of EAL 4 security assurance requirements made in this Security Target is the addition of ALC_FLR.1. ALC_FLR.1 has no dependencies and therefore all of the security assurance requirement dependencies are satisfied.

8.8 PROTECTION PROFILE CLAIMS RATIONALE

This Security Target does not claim conformance to a Protection Profile.

APPENDIX A - ACRONYMS

AMP	Access Module Processor
AWS	Administration Workstation
CLI	Call Level Interface
CNS	Console Subsystem
CPU	Central Processing Unit
DAC	Discretionary Access Control
DBW	Database Window
LAN	Local Area Network
MPP	Massive Parallel Processing
OS	Operating System
PDE	Parallel Database Extensions
PE	Parsing Engine
RDBMS	Relational Database Management System
SMP	Symmetric Multi Processing
SQL	Structured Query Language
TDP	Teradata Director Program
TPA	Trusted Parallel Application
vdisk	Virtual Disk
vproc	Virtual Processor