Microsoft SQL Server™ 2005 SP1 Database Engine Common Criteria Evaluation

Security Target

SQL Server 2005 Team

Author: Roger French

Status: Final
Version: 1.4
Last Saved: 2007-01-23
File Name: MS_SQL_ST_EAL1_1.4

Abstract

This document is the Security Target (ST) for Microsoft SQL Server™ 2005 Common Criteria Certification

Keywords

CC, ST, Common Criteria, SQL, Security Target

Revision History

Date	Version	Author	Edit
2006-10-17	0.1	Roger French	Initial Version
2006-10-19	0.2	Roger French	First consistent set of objectives, threats and SFRs
2006-10-19	0.3	Roger French	Updated rationale
2006-10-20	0.4	Roger French	Consistency check between Rationale and rest of the ST
2006-10-23	0.5	Roger French	Removed threats against assurance
2006-10-23	0.6	Roger French	First version for kick-off meeting
2006-10-26	0.7	Roger French	Reviewed the rationale, First version for evaluation
2006-11-02	0.8	Roger French	Incorporated feedback from meeting with BSI
2006-11-08	0.81	Roger French	Updated certification ID
2006-11-08	1.0	Roger French	First final version after evaluation
2006-12-15	1.1	Roger French	Updates to SF.AC
2006-12-21	1.2	Roger French	Minor editorial updates
2007-01-10	1.3	Roger French	Minor editorial updates
2007-01-23	1.4	Roger French	Updates to SF.AU

This page intentionally left blank

Table of Contents

List of Tables

Page

List of Figures

Page

1 ST Introduction

This chapter presents security target (ST) identification information and an overview of the ST. An ST contains the information technology (IT) security requirements of an identified Target of Evaluation (TOE) and specifies the functional and assurance security measures offered by that TOE to meet stated requirements. An ST principally defines:

a) A security problem expressed as a set of assumptions about the security aspects of the environment, a list of threats that the TOE is intended to counter, and any known rules with which the TOE must comply (chapter 3, TOE Security Environment).

b) A set of security objectives and a set of security requirements to address the security problem (chapters 4 and 5, Security Objectives and IT Security Requirements, respectively).

c) The IT security functions provided by the TOE that meet the set of requirements (chapter 6, TOE Summary Specification).

The structure and content of this ST comply with the requirements specified in the Common Criteria (CC), Part 1, Annex C, and Part 3, chapter 5.

1.1 ST Identification

This chapter provides information needed to identify and control this ST and its Target of Evaluation (TOE).

ST Title:	**Microsoft SQL Server 2005™ SP1 Database Engine Common Criteria Evaluation Security Target**
ST Version:	1.4
Date:	2007-01-23
Author:	Roger French, Microsoft Corporation
Certification-ID:	BSI-DSZ-CC-0416
TOE Identification:	Database Engine of Microsoft SQL Server 2005 Enterprise Edition (English) SP1 and its related guidance documentation.
TOE Version:	9.00.2047.00
TOE Platform:	Windows Server 2003 Enterprise Edition (English) SP1 including MS05-042, MS05-039, MS05-027, A patch that updates the Internet Protocol (IP) Security (IPSec) Policy Agent is available for Windows Server 2003 and Windows XP (KB 907865) as specified in [WIN_ST].
CC Identification:	Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005 (also known as ISO 15408).
Evaluation Assurance Level:	EAL1
PP Conformance:	none
Keywords:	CC, ST, Common Criteria, SQL, Security Target

1.2 ST Overview

The TOE is the database engine of SQL Server 2005 SP1. SQL Server is a Database Management System (DBMS).

The TOE has been developed as the core of the DBMS to store data in a secure way.

The security functionality of the TOE comprises:

- Security Management
- Access Control
- Identification and Authentication
- Security Audit

A summary of the TOE security functions can be found in chapter 2, TOE Description. A more detailed description of the security functions can be found in chapter 6, TOE Summary Specification.

Please note that only the SQL Server 2005 database engine is addressed in this ST. Other related products of the SQL Server 2005 platform, such as Service Broker, provide services that are useful but are not central to the enforcement of security policies. Hence, security evaluation is not directly applicable to those other products.

1.3 CC Conformance

The TOE is [CC_PART2] extended and [CC_PART3] conformant at the level of assurance EAL1.

1.4 Acknowledgement

This ST does not claim compliance to any Protection Profile. However this ST has been developed based on the U.S. Government Protection Profile for Database Management Systems in Basic Robustness Environments, Version 1.1, 07.06.2006 ([PP]) and uses some of the constructs of this PP. Further parts of the assumptions, threats and objectives and the corresponding parts of the rationale have been taken from [PP].

1.5 Conventions

For this Security Target the following conventions are used:

The CC allows several operations to be performed on functional requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in paragraph 148 of Part 1 of the CC. Each of these operations is used in this ST.

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by **bold text**.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections that have been made are denoted by *italicized text*.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments that have been made are denoted by showing the value in square brackets, [Assignment_value].

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing the iteration number in parenthesis following the component identifier, (iteration_number).

The CC paradigm also allows protection profile and security target authors to create their own requirements. Such requirements are termed ‘explicit requirements’ and are permitted if the CC does not offer suitable requirements to meet the authors’ needs. **Explicit requirements** must be identified and are required to use the CC class/family/component model in articulating the requirements. In this ST, explicit requirements will be indicated with the “_EXP” following the component name.

This ST also includes security requirements on the IT environment. Explicit Environmental requirements will be indicated with the “_(ENV)” following the component name.

2 TOE Description

This chapter provides context for the TOE evaluation by identifying the product type and describing the evaluated configuration. The main purpose of this chapter is to bind the TOE in physical and logical terms. The chapter starts with a description of the product type before it introduces the physical scope, the architecture and last but not least the logical scope of the TOE.

2.1 Product Type

The product type of the Target of Evaluation (TOE) described in this ST is a database management system (DBMS) with the capability to limit TOE access to authorized users, enforce Discretionary Access Controls on objects under the control of the database management system based on user and/or role authorizations, and to provide user accountability via audit of users' actions.

A DBMS is a computerized repository that stores information and allows authorized users to retrieve and update that information. A DBMS may be a single-user system, in which only one user may access the DBMS at a given time, or a multi-user system, in which many users may access the DBMS simultaneously.

The TOE which is described in this ST is the database engine and therefore part of SQL Server 2005. It provides a relational database engine providing mechanisms for Access Control, Identification and Authentication and Security Audit.

SQL Server additionally includes the following tools which are not part of the TOE:

- Replication Services: Data replication for distributed or mobile data processing applications and integration with heterogeneous systems, including existing Oracle databases.
- Notification Services: Notification capabilities for the development and deployment of applications that can deliver personalized, timely information updates to a variety of connected and mobile devices.
- Integration Services: Extract, transform, and load capabilities for data warehousing and enterprise-wide data integration
- Analysis Services: Online analytical processing (OLAP) capabilities for the analysis of large and complex datasets.
- Reporting Services: A comprehensive solution for creating, managing, and delivering both traditional, paper-oriented reports and interactive, Web-based reports.
- Management tools: SQL Server includes integrated management tools for database management and tuning as well as tight integration with tools such as Microsoft Operations Manager (MOM) and Microsoft Systems Management Server (SMS). Standard data access protocols drastically reduce the time it takes to integrate data

in SQL Server with existing systems. In addition, native Web service support is built into SQL Server to ensure interoperability with other applications and platforms.

- Development tools: SQL Server offers integrated development tools for the database engine, data extraction, transformation, and loading (ETL), data mining, OLAP, and reporting that are tightly integrated with Microsoft Visual Studio to provide end-to-end application development capabilities. Every major subsystem in SQL Server ships with it's own object model and set of APIs to extend the data system in any direction that is unique to each business.

The TOE itself only comprises the database engine of the SQL Server 2005 platform which provides the security functionality as required by this ST. All the additional tools as listed before interact with the TOE as a standard SQL client. The scope and boundary of the TOE will be described in the next chapter.

2.2 Physical Scope and Boundary of the TOE

The TOE is the database engine of the SQL Server 2005 and its related guidance documentation.

The following figure shows the TOE (including its internal structure) and its immediate environment.

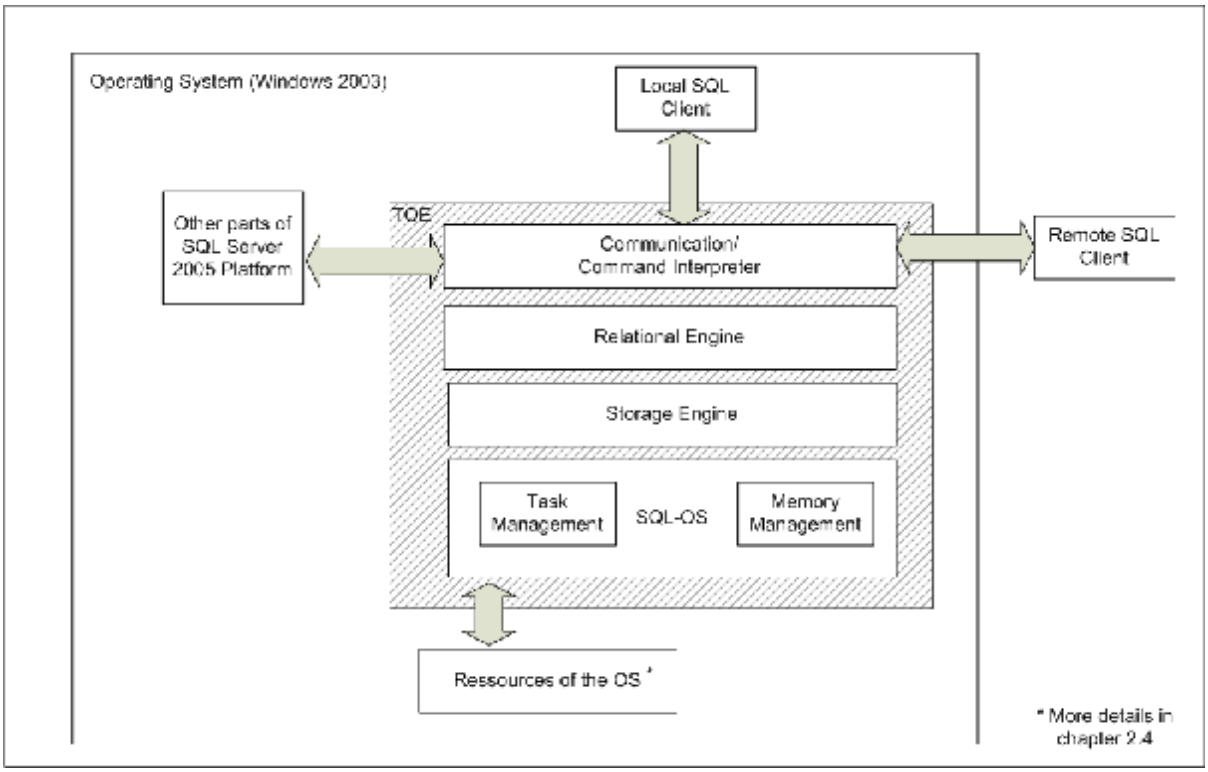

Figure 1: TOE

As seen in Figure 1 the TOE internally comprises the following logical units:

The **Communication** part is the interface for programs accessing the TOE. It is the interface between the TOE and clients performing requests. It processes Tabular Data Stream (TDS) packets to identify the type of packet and translate the packet type into a specific request type.

All responses to user application requests return to the client through this part of the TOE.

The **Relational Engine** is the core of the database engine and is responsible for all security relevant decisions. The relational engine establishes a user context, syntactically checks every Transact SQL (T-SQL) statement, compiles every statement, checks permissions to determine if the statement can be executed by the user associated with the request, optimizes the query request, builds and caches a query plan, and executes the statement.

The **Storage Engine** is a resource provider. When the relational engine attempts to execute a T-SQL statement that accesses an object for the first time, it calls upon the storage engine to retrieve the object, put it into memory and return a pointer to the execution engine. To perform these tasks, the storage engine manages the physical resources for the TOE by using the Windows OS.

The **SQL-OS** is a resource provider for all situations where the TOE uses functionality of the operating system. SQL-OS provides an abstraction layer over common OS functions and was designed to reduce the number of context switches within the TOE. SQL-OS especially contains functionality for Task Management and for Memory Management.

For **Task Management** the TOE provides an OS-like environment for threads, including scheduling, and synchronization —all running in user mode, all (except for I/O) without calling the Windows Operating System.

The **Memory Manager** is responsible for the TOE memory pool. The memory pool is used to supply the TOE with its memory while it is executing. Almost all data structures that use memory in the TOE are allocated in the memory pool. The memory pool also provides resources for transaction logging and data buffers.

The immediate **environment** of the TOE comprises:

The Windows 2003 Server Enterprise Edition Operating System, which hosts the TOE. As the TOE is a software only TOE it lives as a process in the Operating System (OS) and uses the resources of the OS. These resources comprise general functionality (e.g. the memory management and scheduling features of the OS) as well as specific functionality of the OS, which is important for the Security Functions of the TOE (see chapter 5.2 for more details)

Other parts of the SQL Server 2005 Platform, which might be installed together with the TOE. The TOE is the central part of a complete DBMS platform, which realizes all Security Functions as described in this ST. However other parts of the platform may be installed on the same machine if they are needed to support the operation or administration of the TOE. However these other parts will interact with the TOE in the same way, every other client would do.

Clients comprising (local clients and remote clients) are used to interact with the TOE during administration and operation. Services of the Operating System are used to route the communication of remote clients with the TOE.

The TOE relies on functionality of the Windows 2003 Server Operating System and has the following hardware requirements:

- 600-megahertz (MHz) Pentium III-compatible or faster processor; 1-gigahertz (GHz) or faster processor recommended
- 512 megabytes (MB) of RAM or more; 1 gigabyte (GB) or more recommended
- Approximately 350 MB of available hard-disk space for the recommended installation
- Approximately 425 MB of additional available hard-disk space for SQL Server Books Online, SQL Server Mobile Books Online, and sample databases
- CD-ROM or DVD-ROM drive
- Super VGA (1,024x768) or higher-resolution video adapter and monitor
- Microsoft Mouse or compatible pointing device

The following guidance documents and supportive information belong to the TOE:

- SQL Server Books Online, July 2006
- SQL Server Guidance Addendum / Installation / Startup

The website www.microsoft.com/sql/commoncriteria/. contains additional information about the TOE and its evaluated configuration. This website shall be visited before using the TOE.

2.3 Architecture of the TOE

The TOE which is described in this ST comprises one instance of the SQL-Server 2005 database engine but has the possibility to serve several clients simultaneously. All clients which connect to the TOE are within the same enclave as the TOE which means that they are under the same management control and operate under the same security policy constraints.

2.4 Logical Scope and Boundary of the TOE

SQL Server 2005 is able to run multiple instances of the database engine on one machine. After installation one default instance exists. However the administrator is able to add more instances of SQL Server 2005 to the same machine.

The TOE comprises one instance of SQL Server 2005. Within this ST it is referenced either as "the TOE" or as "instance". The machine the instances are running on is referenced as "server" or "DBMS-server".

If more than one instance of SQL Server 2005 is installed on one machine these just represent multiple TOEs as there is no other interface between two instances of the TOE than the standard client interface

In this way two or more instances of the TOE may only communicate through the standard client interface.

The TOE provides the following set of security functionality

- The **Access Control** function of the TOE ensures that only authorized users are able to connect to the TOE and access user data stored in the TOE. It further controls that only authorized administrators are able to manage the TOE.
- The **Security Audit** function of the TOE produces log files about all security relevant events.
- The **Management** function allows authorized administrators to manage the behavior of the security functions of the TOE.
- The **Identification and Authentication**[1] function of the TOE is able to identify and authenticate users based on a Username/Password based mechanism.

The following functions are part of the environment:

- The **Audit Review** and **Audit Storage** functionality has to be provided by the environment and provide the authorized administrators with the capability to review the security relevant events of the TOE.
- The **Access Control Mechanisms** has to be provided by the environment for files stored in the environment
- The environment provides **Identification and Authentication**[1] for users for the cases where this is required by the TOE (The environment AND the TOE provide mechanisms for user authentication. See chapter 6.1.3 for more details).
- The environment has to provide **Time stamps** to be used by the TOE.
- The environment provides a **cryptographic** mechanisms for **hashing** of passwords

All these functions are provided by the underlying Operating System (Windows 2003 Server Enterprise Edition) except Audit Review, for which an additional tool has to be used (e.g. the SQL Server Profiler, which is part of the SQL Server Platform).

Access to the complete functionality of the TOE is possible via a set of SQL-commands (see [TSQL]).

This set of commands is available via:

- Shared Memory

[1] Note that the TOE as well as the environment provides a mechanism for identification and authentication. Chapter 6 will describe this in more detail.

- Named Pipes
- TCP/IP

3 TOE Security Environment

The security environment for the functions addressed by this specification includes threats, security policies, and usage assumptions, as discussed below.

3.1 Assets

The TOE maintains two types of data which represent the assets: User Data and TSF Data.

The primary assets are the User Data which comprises the following:

- The user data stored in or as database objects;
- User-developed queries or procedures that the DBMS maintains for users.

The secondary assets comprise the TSF data that the TOE maintains and uses for its own operation. This kind of data is also called metadata. It especially includes:

- The definitions of user databases and database objects
- Configuration parameters,
- User security attributes,
- Transaction logs,
- Security Audit instructions and records

3.2 Assumptions

The following table lists all the assumptions about the environment of the TOE.

Table 1 - Assumptions

Assumption	Description
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 DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.
A.OS_PP_VALIDATED	The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness and the Operating System provides functionality for • Identification and authentication of users, • Access Control for Files, • Time stamps and • Audit Storage and Audit Review • Hashing of passwords
A.PHYSICAL	It is assumed that appropriate physical security is provided for the server, on which the TOE is installed, considering the value of the stored, processed, and transmitted information.
A.COMM	It is assumed that any communication path from and to the TOE is appropriately secured to avoid eavesdropping and manipulation.

3.3 Threats

The following table lists the threats against the assets, which are protected by the TOE and its environment.

Table 2 - Threats to the TOE

Threat	Description
T. ACCIDENTAL_ADMIN_ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.MASQUERADE	A user or process may claim to be another entity in order to gain unauthorized access to data or TOE resources.
T.TSF_COMPROMISE	A user or process may try to access (i.e. view, modify or delete) configuration data of the TOE. This could allow the user or process to gain knowledge about the configuration of the TOE or could bring the TOE into an insecure configuration in which the security mechanisms for the protection of the assets are not longer working correctly.
T.UNAUTHORIZED_ACCESS	A user may try to gain unauthorized access to user data for which they are not authorized according to the TOE security policy. Within the scope of this threat the user just tries to access assets, he doesn't have permission on, without trying to masquerade another user or circumventing the security mechanism in any other way.

3.4 Organizational Security Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. This chapter identifies the organizational security policies applicable to the TOE.

Table 3 – Organizational Security Policies

Policy	Description
P.ACCOUNTABILITY	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P.ROLES	The TOE shall provide an authorized administrators role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.

4 Security Objectives

The purpose of the security objectives is to detail the planned response to a security problem or threat. Threats can be directed against the TOE or the security environment or both therefore, the CC identifies two categories of security objectives:

- Security objectives for the TOE, and
- Security objectives for the environment.

4.1 Security Objectives for the TOE

This chapter identifies and describes the security objectives of the TOE.

Table 4 - Security Objectives for the TOE

Objective	Description
O.ADMIN_GUIDANCE	The TOE will provide administrators with the necessary information for secure management.
O.ADMIN_ROLE	The TOE will provide authorized administrators roles to isolate administrative actions.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users.
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.I&A	The TOE will provide a mechanism for identification and authentication of users.

4.2 Security Objectives for the Environment

The security objectives for the TOE Environment are defined in the following table.

Table 5 - Security Objectives for the TOE Environment

Objective	Description
OE.NO_EVIL	Sites using the TOE shall ensure that authorized administrators are non-hostile, appropriately trained and follow all administrator guidance.
OE.NO_GENERAL_ PURPOSE	There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.
OE.OS_PP_VALIDATED	The underlying OS has to be validated against an NSA sponsored OS PP of at least Basic Robustness and has to provide functionality for • Identification and authentication of user, • Access Control for Files, • Time stamps and • Audit Storage and Audit Review • Hashing of passwords
OE.PHYSICAL	Physical security shall be provided for the server, on which the TOE will be installed, considering the value of the stored, processed, and transmitted information.
OE.COMM	Any communication path from and to the TOE will be appropriately secured to avoid eavesdropping and manipulation.

All objectives with exception of OE.OS_PP_VALIDATED address the Non-IT environment of the TOE. The objective OE.OS_PP_VALIDATED is related to the non-IT environment as well as to the IT environment, because it contains IT aspects.

5 IT Security Requirements

This chapter defines the IT security requirements that shall be satisfied by the TOE or its environment:

The CC divides TOE security requirements into two categories:

- Security functional requirements (SFRs) (such as, identification and authentication, security management, and user data protection) that the TOE and the supporting evidence need to satisfy to meet the security objectives of the TOE.
- Security assurance requirements (SARs) that provide grounds for confidence that the TOE and its supporting IT environment meet its security objectives (e.g., configuration management, testing, and vulnerability assessment).

These requirements are discussed separately within the following subchapters.

5.1 TOE Security Functional Requirements

The TOE satisfies the SFRs delineated in the following table. The rest of this chapter contains a description of each component and any related dependencies.

Table 6 - TOE Security Functional Requirements

Class FAU: Security Audit	
FAU_GEN.1	Audit data generation
FAU_GEN_EXP.2	User and/or group identity association
FAU_SEL.1	Selective audit
FAU_STG_EXP.4	Administrable Prevention of audit data loss
Class FDP: User Data Protection	
FDP_ACC.1	Subset access control
FDP_ACF.1	Security attribute based access control
Class FIA: Identification and Authentication	
FIA_ATD.1	User attribute definition
FIA_UAU.2	User authentication before any action
FIA_UAU.5	Multiple authentication mechanisms
FIA_UID.2	User identification before any action

Class FMT: Security Management	
FMT_MOF.1	Management of security functions behaviour
FMT_MSA.1	Management of security attributes
FMT_MSA.3	Static attribute initialization
FMT_MTD.1	Management of TSF data
FMT_REV.1(1)	Revocation (user attributes)
FMT_REV.1(2)	Revocation (subject, object attributes)
FMT_SMF.1	Specification of management functions
FMT_SMR.1	Security roles

5.1.1 Class FAU: Security Audit

Audit data generation (FAU_GEN.1)

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 *minimum* level of audit **listed in Table 7**; and

c) [Start-up and shutdown of the DBMS;

d) Use of special permissions (e.g., those often used by authorized administrators[2] to circumvent access control policies)]

[2] Note that in the context of this Security Target the term „Authorized Administrator“ refers either to the „sysadmin“ (sa) or any other user who has the permission to perform the administration activity based on the DAC policy (see also chapter 9.3.1).

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 (if applicable), 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, [none].

Table 7 - Auditable Events

Security Functional Requirement	Auditable Event(s)
FAU_GEN.1	None
FAU_GEN_EXP.2	None
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating.
FDP_ACC.1	None
FDP_ACF.1	Successful requests to perform an operation on an object covered by the SFP.
FIA_ATD.1	None
FMT_MOF.1	None
FMT_MSA.1	None
FMT_MSA.3	None
FMT_MTD.1	None
FMT_REV.1(1)	Unsuccessful revocation of security attributes.
FMT_REV.1(2)	Unsuccessful revocation of security attributes.
FMT_SMF.1	Use of the management functions
FMT_SMR.1	Modifications to the group of users that are part of a role.
FAU_STG_EXP.4	Every modifications to the setting
FIA_UAU.2	Every use of the authentication mechanism.
FIA_UAU.5	The final decision on authentication;
FIA_UID.2	Every use of the authentication mechanism.

User and/or group identity association (FAU_GEN_EXP.2)

FAU_GEN_EXP.2.1 For audit events resulting from actions of identified users and/or identified groups, the TSF shall be able to associate each auditable event with the identity of the user and/or group that caused the event.

Selective audit (FAU_SEL.1)

FAU_SEL.1.1 **Refinement:** The TSF shall **allow only the administrator** to include or exclude auditable events from the set of audited events based on the following attributes:

a) user identity, object identity,

b) [success of auditable security events, failure of auditable security events]

Administrable Prevention of audit data loss (FAU_STG_EXP.4)

FAU_STG_EXP.4.1 The TSF shall take one of the following actions: [

- Overwrite the oldest stored audit records
- Stop the TOE]

As specified by the administrator and [no other action] if the audit trail is full.

5.1.2 Class FDP: User Data Protection

Subset access control (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the [Discretionary Access Control policy] on [all subjects, all DBMS-controlled objects and all operations among them].

Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 The TSF shall enforce the [Discretionary Access Control policy] to objects based on the following:

• [the authorized user identity and/or group membership associated with a subject,

• access operations implemented for DBMS-controlled objects, and

• object identity].

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

• The Discretionary Access Control policy mechanism shall, either by explicit authorized user action or by

	default, provide that database management system controlled objects are protected from unauthorized access according to the following ordered rules: [a) If the requested mode of access is denied to that authorized user deny access b) If the requested mode of access is denied to [any] group of which the authorized user is a member, deny access c) If the requested mode of access is permitted to that authorized user, permit access. d) If the requested mode of access is permitted to any group of which the authorized user is a member, grant access e) Else deny access]
FDP_ACF.1.3	The TSF shall explicitly authorize access of subjects to **DBMS-controlled** objects based on the following additional rules: [• Authorized administrators, the owner of an object and owners of parent objects have access • in case of Ownership-Chaining access is always granted In case a user has been granted access to one or more columns of a table, access to this/these columns is always granted].
FDP_ACF.1.4	The TSF shall explicitly deny access of subjects to objects based on the [no additional explicit denial rules].

5.1.3 Class FIA: Identification and authentication

User attribute definition (FIA_ATD.1)

FIA_ATD.1.1	The TSF shall maintain the following list of security attributes belonging to individual users: • [Database user identifier and/or group memberships; • Security-relevant database roles; and • login-type (SQL-Server login or Windows Account Name) • For SQL-Server login: Hashed password].

User authentication before any action (FIA_UAU.2)

FIA_UAU.2.1	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Multiple authentication mechanisms (FIA_UAU.5)

FIA_UAU.5.1 The TSF shall provide [

- SQL Server Authentication and
- Access to Windows Authentication[3]]

to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [following rules:

- If the login is associated with a Windows user or a Windows group Windows Authentication is used,
- If the login is a SQL Server login the SQL Server authentication is used.

].

User identification before any action (FIA_UID.2)

FIA_UID.2.1 The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.1.4 Class FMT: Security Management

Management of security functions behaviour (FMT_MOF.1)

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

Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 The TSF shall enforce the [Discretionary Access Control policy] to restrict the ability to [*manage*] the security attributes [all] to [authorized administrators].

[3] Windows Authentication is not provided by the TOE but by the environment. For this case the TOE reuses the authentication results of Windows. However, in every case the TOE enforces the policy that each user has to be successfully authenticated before allowed to perform any other action and provides an interface to the operating system to gain the authentication results and to the user to allow the user to start the process of authentication.

Static attribute initialization (FMT_MSA.3)

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 role] to specify alternative initial values to override the default values when an object or information is created.

Management of TSF data (FMT_MTD.1)

FMT_MTD.1.1	The TSF shall restrict the ability to [*include or exclude*] the [auditable events] to [authorized administrators].

Revocation (FMT_REV.1(1))

FMT_REV.1.1(1)	The TSF shall restrict the ability to revoke security attributes associated with *users* within the TSC to [the authorized administrators].
FMT_REV.1.2(1)	The TSF shall enforce the rules [Changes to SQL logins are applied immediately, Changes to logins which are associated with a Windows account may require the user to login to the TOE again before they are applied]

Revocation (FMT_REV.1(2))

FMT_REV.1.1(2)	The TSF shall restrict the ability to revoke security attributes associated with *objects* within the TSC to [the authorized administrators and database users as allowed by the Discretionary Access Control policy].
FMT_REV.1.2(2)	The TSF shall enforce the rules [The changes have to be applied immediately].

Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [

- Add and delete logins
- Add and delete users
- Change role membership for DB scoped roles and Server scoped roles
- Create and destroy database scoped groups
- Create, Start and Stop Audit
- Include and Exclude Auditable events

- Define the mode of authentication
- Define the action to take in case the audit file is full]

Security roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles:[

- Roles as defined in the following tables
- Roles to be defined by authorized administrators].

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

Table 8 – Default Server Roles

Role	Granted Permission(s)
bulkadmin	ADMINISTER BULK OPERATIONS
dbcreator	CREATE DATABASE
diskadmin	ALTER RESOURCES
processadmin	ALTER ANY CONNECTION, ALTER SERVER STATE
securityadmin	ALTER ANY LOGIN
serveradmin	ALTER ANY ENDPOINT, ALTER RESOURCES, ALTER SERVER STATE, ALTER SETTINGS, SHUTDOWN, VIEW SERVER STATE
setupadmin	ALTER ANY LINKED SERVER
sysadmin	CONTROL SERVER (Granted with grant option)

Table 9 - Default Database Roles

Role	Granted Permission(s)	Granted Permission on the Server level	Denied Permission(s)
db_accessadmin	ALTER ANY USER, CREATE SCHEMA CONNECT (Granted with grant option)	VIEW ANY DATABASE	-
db_backupoperator	BACKUP DATABASE, BACKUP LOG, CHECKPOINT	VIEW ANY DATABASE	-
db_datareader	SELECT	VIEW ANY DATABASE	-
db_datawriter	DELETE, INSERT, UPDATE	VIEW ANY DATABASE	-

db_ddladmin	ALTER ANY ASSEMBLY, ALTER ANY ASYMMETRIC KEY, ALTER ANY CERTIFICATE, ALTER ANY CONTRACT, ALTER ANY DATABASE DDL TRIGGER, ALTER ANY DATABASE EVENT NOTIFICATION, ALTER ANY DATASPACE, ALTER ANY FULLTEXT CATALOG, ALTER ANY MESSAGE TYPE, ALTER ANY REMOTE SERVICE BINDING, ALTER ANY ROUTE, ALTER ANY SCHEMA, ALTER ANY SERVICE, ALTER ANY SYMMETRIC KEY, CHECKPOINT, CREATE AGGREGATE, CREATE DEFAULT, CREATE FUNCTION, CREATE PROCEDURE, CREATE QUEUE, CREATE RULE, CREATE SYNONYM, CREATE TABLE, CREATE TYPE, CREATE VIEW, CREATE XML SCHEMA COLLECTION, REFERENCES	VIEW ANY DATABASE	-
db_denydatareader	-	VIEW ANY DATABASE	SELECT
db_denydatawriter	-	-	DELETE, INSERT, UPDATE
db_owner	CONTROL (Granted with grant option)	VIEW ANY DATABASE	-
db_securityadmin	ALTER ANY APPLICATION ROLE, ALTER ANY ROLE, CREATE SCHEMA, VIEW DEFINITION	VIEW ANY DATABASE	-

5.2 Security Requirements for the IT Environment

This section contains the security functional requirements for the IT environment.

The environment of the TOE (the Operating System) has to satisfy the SFRs delineated in the following table. The rest of this chapter contains a description of each component.

Table 10 - TOE Security Functional Requirements for the environment

Class FAU: Security Audit	
FAU_STG.1/ENV	Protected audit trail storage
FAU_SAR.1/ENV	Audit Review
Class FCS: Cryptographic Support	
FCS_COP.1/ENV	Cryptographic Operation
Class FDP: User Data Protection	
FDP_ACC.1/ENV	Subset access control
FDP_ACF.1/ENV	Security attribute based access control
Class FIA: Identification and Authentication	
FIA_UAU.1/ENV	Timing of authentication
FIA_UID.1/ENV	Timing of identification
Class FMT: Security Management	
FMT_MSA.3/ENV	Static attribute initialisation
Class FPT: Protection of the TSF	
FPT_STM.1/ENV	Reliable time stamps

5.2.1 Class FAU: Security Audit

Protected audit trail storage (FAU_STG.1/ENV)

FAU_STG.1.1/ENV The **IT environment** shall protect the stored audit records from unauthorised deletion.

FAU_STG.1.2/ENV The **IT environment** shall be able to *prevent* unauthorised modifications to the stored audit records in the audit trail.

Audit review (FAU_SAR.1)

FAU_SAR.1.1/ENV The **IT environment** shall provide [administrators] with the capability to read [all information] from the audit records.

FAU_SAR.1.2/ENV The **IT environment** shall provide the audit records in a manner suitable for the user to interpret the information.

5.2.2 Class FCS: Cryptographic Support

Cryptographic operation for the IT environment (FCS_COP.1/ENV)

FCS_COP.1.1/ENV The **IT environment** shall perform [hash value calculation] in accordance with a specified cryptographic algorithm [SHA-1] and cryptographic key sizes [not applicable] that meet the following: [FIPS 180-2].

5.2.3 Class FDP: User Data Protection

Subset access control (FDP_ACC.1/ENV)

FDP_ACC.1.1/ENV The **IT environment** shall enforce the [OS discretionary access control policy] on [

subjects – processes acting on behalf of users

objects – NTFS files and/or NTFS directories and registry and Active Directory objects

operations – all operations among subjects and objects covered by OS discretionary access control policy].

Security attribute based access control (FDP_ACF.1/ENV)

FDP_ACF.1.1/ENV The **IT environment** shall enforce the [OS discretionary access control policy] to objects based on the following: [*subject attribute* – security ID of user or group

object attributes – access control list].

FDP_ACF.1.2/ENV The **IT environment** shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [If the operation is explicitly allowed and not explicitly denied by an entry in the access list for the accessing subject, the accessing subject is able to perform the specified operation].

FDP_ACF.1.3/ENV The **IT environment** shall explicitly authorise access of subjects to objects based on the following additional rules: [none].

FDP_ACF.1.4/ENV The **IT environment** shall explicitly deny access of subjects to objects based on the [none].

5.2.4 Class FIA: Identification and authentication

Timing of authentication (FIA_UAU.1/ENV)

FIA_UAU.1.1/ENV	The **IT environment** shall allow [no access to the TOE] on behalf of the user to be performed before the user is authenticated.
FIA_UAU.1.2/ENV	The **IT environment** shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Timing of identification (FIA_UID.1/ENV)

FIA_UID.1.1/ENV	The **IT environment** shall allow [no access to the TOE] on behalf of the user to be performed before the user is identified.
FIA_UID.1.2/ENV	The **IT environment** shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.2.5 Class FMT: Security Management

Static attribute initialisation (FMT_MSA.3/ENV)

FMT_MSA.3.1/ENV	The **IT environment** shall enforce the [OS discretionary access control policy] to provide *restrictive* default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2/ENV	The **IT environment** shall allow the [creator or authorized administrator] to specify alternative initial values to override the default values when an object or information is created.

5.2.6 Class FPT: Protection of the TSF

Reliable time stamps (FPT_STM.1/ENV)

FPT_STM.1.1/ENV	The **IT environment** shall be able to provide reliable time stamps for **the TOE**.

5.3 Security Requirements for the Non-IT Environment

R.EVL The evaluation of the Operating System in the environment has to be performed to at least EAL 1 and against an NSA sponsored OS PP to provide a suitable environment that meets the requirements of the TOE described in this ST.

R.COMM Any communication path from and to the TOE will be appropriately secured to avoid eavesdropping and manipulation. This can be achieved by the use of another IT-product in the environment or by physical protection of the communication path.

R.PHYSICAL It has to be ensured that sufficient physical security is provided for the server, on which the TOE is installed, considering the value of the stored, processed, and transmitted information.

R.ADMIN It has to be ensured that authorized administrators are non-hostile, appropriately trained and follow all administrator guidance. Further no general-purpose computing capabilities (e.g., compilers or user applications) must be available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.

5.4 TOE Security Assurance Requirements

The assurance requirements for the TOE comprise all assurance requirements for EAL 1 as defined in [CC_PART3].

6 TOE Summary Specification

This chapter presents an overview of the security functions implemented by the TOE and the Assurance Measures applied to ensure their correct implementation.

6.1 TOE Security Functions

This chapter presents the security functions performed by the TOE to satisfy the identified SFRs in chapter 5.1.1.

The following table gives an overview of these functions.

Table 11 – Summary of Security Functions

Security Function	Description
Security Management	This Security Function provides the necessary functions to change the behavior of the TSF.
Access Control	This Security Function realizes the Discretionary Access Control Policy for all objects under the control of the TOE.
Identification and Authentication	This Security Function realizes the identification and authentication function of the TOE.
Security Audit	This Security Function realizes the audit functionality for the TOE.

The following paragraphs contain a more detailed description of the security functions.

6.1.1 Security Management (SF.SM)

This Security Function of the TOE allows modifying the TSF data of the TOE and therewith managing the behavior of the TSF.

This comprises the following management functions:

- Add and delete logins on an instance level
- Add and delete users on a database level
- Change role membership for DB scoped roles and Server scoped roles
- Create and destroy database roles
- Create, Start and Stop Security Audit
- Include and exclude Auditable events

- Define the mode of authentication for every login
- Define the action to take in case the audit file is full

All these management functions are available via T-SQL statements directly or realized by Stored Procedures within the TOE which can be called using T-SQL. This Security Function additionally ensures that the management functions are only available for authorized administrators.

The TOE maintains a set of roles on the server level and on the database level as listed in Table 8 – Default Server Roles and Table 9 - Default Database Roles. The TOE maintains a security ID for each login on a server level and each database user. This security ID is used to associate each user with his assigned roles.

6.1.2 Access Control (SF.AC)

The TOE provides a Discretionary Access Control (DAC) mechanism to control the access of users to objects based on the identity of the user requesting access, the membership of this user to roles, the requested operation and the ID of the requested object.

The TOE maintains two kinds of user representations:

1. On an instance level an end user is represented by a login. On this level the Security Function controls the access of logins to objects pertaining to the instance (e.g. to view a database)
2. On a database level an end user is represented by a database user. On this level this Security Function controls the access of database users to objects of the database (e.g. to read or create a table).

Members of the database roles “db_owner” or “db_accessadmin” are able to add users to a database. The TOE maintains an internal security identifier (SID) for every user and role. Each database user can be associated with at most one instance “login”.

Every object controlled by the TOE has an ID, an owner and a name.

Objects in the TOE form a hierarchy and belong to one of three different levels: server, database and schema.

The TOE maintains an Access Control List (ACL) for each object within its scope. These ACLs are stored in a system table which exists in every database for database related ACLs and in a system table in the ‘master’ database for instance level ACLs.

Each entry of an ACL contains a user SID and defines whether a permission is an “Allow” or a “Deny” permission for that SID.

When a new object is created, the creating user is assigned as the owner of the object and has complete control over the object. The ACL for a newly created object is always empty by default.

After creation, grant, deny or revoke permissions on objects can be assigned to users. Changes to the security relevant attributes of objects are immediately applied.

When a user attempts to perform an action to an object under the control of the TOE, the TOE decides whether the action is to be permitted based on the following rules:

1. If the requested mode of access is denied to that authorized user, the TOE will deny access
2. If the requested mode of access is denied to any role of which the authorized user is a member, the TOE will deny access
3. If the requested mode of access is permitted to that authorized user, the TOE will permit access
4. If the requested mode of access is permitted to any role of which the authorized user is a member, the TOE will permit access
5. Else: The TOE will deny access

The TOE permission check for an action on an object includes the permissions of its parent objects. The permissions for the object itself and all its parent objects are accumulated together before the aforementioned rules are evaluated. Note: Some actions require more than one permission.

This means that if a user or a role has been granted a permission to an object this permission is also valid for all child objects. E.g. if a user has been granted a permission to a schema, he automatically has the same permission on all tables within that schema, if the permission has not explicitly been denied. Similarly, if a user has been denied a permission on a schema, he will be denied the same permission to all tables within that schema, regardless of explicit grant permissions.

According to the rules mentioned above in situations where a "deny" and a "grant" statement exist at the same time, the deny statement will take precedence. However the following exception exists: if a user has been granted permission on the column level, the grant statement for those columns will override any deny permission from the hierarchy, e.g. if a user is denied SELECT access to a table, but granted SELECT on a column, the user will be able to SELECT that column only. Note that if a DENY exists for the column, the user will always be denied access.

The rules as described before are always applied when a user requests access to a certain object using a certain operation. There are only two situations where these access control rules are overridden:

1. The system administrator, the owner of an object and owners of parent objects always have access, so for these users the TOE will always allow access to the object
2. In the case of “Ownership Chaining” which is described in chapter 9.1 in more detail the access is allowed.

6.1.3 Identification and Authentication (SF.I&A)

This Security Function requires each user to be successfully authenticated before allowing any other actions on behalf of that user. This is done on an instance level and means that the user has to be associated with a login of the TOE.

The TOE knows two types of logins: Windows accounts and SQL Server logins. The administrator has to specify the type of login for every login he is creating.

The possibility for the TOE to perform its own authentication is necessary because not all users connecting to the TOE are connecting from a Windows environment.

Microsoft Windows account names

These logins are associated with a user account of the Windows Operating System in the environment.

For these logins the TOE requires that the Windows environment passes on the Windows SID(s) of that user to authenticate the user before any other action on behalf of that user is allowed.[4]

For these logins the Windows security identifier (SID) from the Windows account or group is used for identification of that login within the TOE. Any permission is associated with that SID.

Any changes which occur to a Windows account in the environment while a user is connected to the TOE are not applied by the TOE until the user logs off and logs on again.

SQL Server login names

SQL Server logins are not associated with a user of Windows but are maintained by the TOE itself. For every SQL Server login the TOE maintains a login name and a password. The password is not stored in plain text, but hashed using the SHA-1 hash function provided by the Operating System in the environment.

Each SQL Server login name is stored in a system table. SQL Server generates a SID that is used as a security identifier and stores it in this table.

This SID is internally used as a security identifier for the login.

If a user is connecting to the TOE using a SQL Server login he has to provide the username and password. The TOE hashes the password using the hash function provided by the Operating System in the environment, and compares the hash to the value stored for that user. If the values are identical the TOE has successfully authenticated the user.

Any changes that occur to the definition of SQL Server login are immediately applied by the TOE.

[4] Windows authenticates users based on a username and password. After successful authentication of a user Windows associates a list of SID(s) with every user which represent the user and every group the user is a member of. Details can be found in [WIN_ST].

6.1.4 Security Audit (SF.AU)

The TOE produces audit logs for all security relevant actions. These audit logs are stored into files in the environment of the TOE.

The Security Audit of the TOE especially comprises the following events:

- Startup and Shutdown of the TOE
- Start and Shutdown of Security Audit Function
- Every login attempt including the processes for authentication and session establishment
- Every successful request to perform on operation on an object covered by the access control function
- Modifications to the role membership of users
- The use of the Security Function SF.SM

The TOE maintains a set of events which can be additionally audited and provides the administrator with the capability to start a Security Audit process to capture these events.

For each event in the Security Audit logs the following information is stored:

1. Date and Time of the event
2. Identity of the user causing the event (if available)
3. ID of the object
4. Outcome (success or failure) of the event

Furthermore each audit file contains an introduction with the list of events which are audited in the file.

The administrator has the possibility to specify, what should happen in case an audit file is full. The following two scenarios are supported in the evaluated version:

1. Rollover

The administrator specifies a maximum size per trace file and a maximum number of files for the Security Audit. If one audit file is full, the TOE starts the next file until the maximum number of files has been reached. When the maximum number of files has been reached and the last audit file is full, the TOE will start overwriting the oldest audit file.

2. Shutdown

The administrator specifies one trace file with a maximum size and the option to shut down the TOE on any audit error. When the maximum size of the trace file has been reached the TOE will stop operation.

Additionally the fact that the audit log is full and the action taken are audited.

The TOE provides the possibility to create a filter for the audit function. Using this filter mechanism the administrator is able to exclude auditable events from being audited based on the following attributes:

- User identity
- Object identity,
- Success or failure of auditable security events

However to modify the behavior of the Security Audit function by including additional or excluding events from being audited the administrator has to stop the Security Audit process, modify the Security Audit function and start the Security Audit process again.

6.2 Assurance Measures

For the evaluation of the TOE the assurance requirements according to CC EAL1 apply. This chapter identifies the assurance measures that are or will be applied by Microsoft in the course of the evaluation to satisfy the assurance requirements. The corresponding assurance measures are listed in Table 12 below.

Table 12 - Assurance Measures

SAR(s)	Assurance Measure(s)
ACM_CAP.1	Provision of the TOE and this ST
ADO_IGS.1	Provision of installation, generation and startup documentation (as part of administrator guidance documentation)
ADV_FSP.1	Provision of functional specification documentation
ADV_RCR.1	Provision of representation of correspondence documentation
AGD_ADM.1 AGD_USR.1	Provision of user/administrator guidance documentation
ATE_IND.1	Provision of the TOE and this ST

7 Protection Profile (PP) Claims

This Security Target does not claim compliance to any Protection Profile.

8 Rationale

This chapter demonstrates the completeness and consistency of this ST by providing justification for the following:

Traceability The security objectives for the TOE and its environment are explained in terms of threats countered and assumptions met. The SFRs are explained in terms of objectives met by the requirement. The traceability is illustrated through matrices that map the following:

- security objectives to threats encountered
- environmental objectives to assumptions met
- SFRs to objectives met
- Security functions to SFRs met

Assurance Level A justification is provided for selecting an EAL1 level of assurance for this ST.

Dependencies A mapping is provided as evidence that all dependencies are met.

8.1 Rationale for TOE Security Objectives

The following table summarizes the rationale for the security objectives.

Table 13 – Summary of Security Objectives Rationale

Threats, Assumptions, OSP / Security Objectives	O.ADMIN_GUIDANCE	O.ADMIN_ROLE	O.AUDIT_GENERATION	O.MANAGE	O.MEDIATE	O.I&A	OE.NO_EVIL	OE.NO_GENERAL_ PURPOSE	OE.OS_PP_VALIDATED	OE.PHYSICAL	OE.COMM
T.ACCIDENTAL_ADMIN_ERROR	X										
T.MASQUERADE						X					
T.TSF_COMPROMISE				X							
T.UNAUTHORIZED_ACCESS					X	X					
P.ACCOUNTABILITY			X			X					
P.ROLES		X									
A.NO_EVIL							X				
A.NO_GENERAL_PURPOSE								X			
A.OS_PP_VALIDATED									X		
A.PHYSICAL										X	
A.COMM											X

Details are given in the following table.

Table 14 – Rationale for TOE Security Objectives

Threat/Policy	Objectives Addressing the Threat/Policy	Rationale
T.ACCIDENTAL_ADMIN_ERROR An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.	O.ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure management.	O.ADMIN_GUIDANCE counters this threat by ensuring the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner. Having this guidance and considering the assumption A.NO_EVIL removes the threat that an administrator might cause the TOE to be configured insecurely.
T.MASQUERADE A user or process may claim to be another entity in order to gain unauthorized access to data or TOE resources.	O.I&A The TOE will provide a mechanism for identification and authentication of users.	O.I&A counters this threat by providing the means to identify and authenticate the user where the I&A mechanisms of the environment is not used. The correct identity of the user is the basis for any decision of the TOE about an attempt of a user to access data. In this way it is not possible for a user or process to masquerade as another entity and the threat is removed.
T.TSF_COMPROMISE A user or process may try to access (i.e. view, modify or delete) configuration data of the TOE. This could allow the user or process to gain knowledge about the configuration of the TOE or could bring the TOE into an insecure configuration in which the security mechanisms for the protection of the assets are not longer working correctly.	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.MANAGE defines that only authorized administrators shall be able to use the management functionality, provided by the TOE and to counter this threat.
T.UNAUTHORIZED_ACCESS A user may try to gain unauthorized access to user data for which they are not	O.MEDIATE The TOE must protect user data in accordance with its security policy.	O.MEDIATE ensures that all accesses to user data are subject to mediation. The TOE requires successful authentication to the TOE prior to

authorized according to the TOE security policy. Within the scope of this threat the user just tries to access assets, he doesn't have permission on, without trying to masquerade another user or circumventing the security mechanism in any other way.		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. Together with O.I&A this mechanism ensures that no user can gain unauthorized access to data and in this way removes the threat.
	O.I&A The TOE will provide a mechanism for identification and authentication of users.	O.I&A contributes to countering this threat by providing the means to identify and authenticate the user where the I&A mechanism of the environment is not used. The correct identity of the user is the basis for any decision of the TOE about an attempt of a user to access data.
P.ACCOUNTABILITY The authorized users of the TOE shall be held accountable for their actions within the TOE.	O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security relevant events associated with users.	O.AUDIT_GENERATION addresses this policy by providing the authorized administrator with the capability of configuring the audit mechanism to record the actions of a specific user.
	O.I&A The TOE will provide a mechanism for identification and authentication of users.	O.I&A supports this policy by providing the means to identify and authenticate the user where the I&A mechanisms of the environment cannot be used. The identity of the user is stored in the audit logs.
P.ROLES The TOE shall provide an authorized administrator role for secure administration of the TOE. This role shall be	O.ADMIN_ROLE The TOE will provide authorized administrator roles to isolate administrative actions.	The TOE has the objective of providing an authorized administrator role for secure administration. The TOE may provide other roles as well, but only the role of authorized administrator

separate and distinct from other authorized users.		is required (O.ADMIN_ROLE).

8.2 Rationale for the Security Objectives for the Environment

The following table contains the rationale for the IT Environmental Objectives.

Table 15 – Rationale for IT Environmental Objectives

Assumption	**Environmental Objective Addressing the Assumption**	**Rationale**
A.NO_EVIL Administrators are non-hostile, appropriately trained, and follow all administrator guidance.	OE.NO_EVIL Sites using the TOE shall ensure that authorized administrators are non- hostile, are appropriately trained and follow all administrator guidance.	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 There are no general-purpose computing or storage repository capabilities (e.g., compilers or user applications) available on DBMS servers, other than those services necessary for the operation, administration and support of the DBMS.	OE.NO_GENERAL_PURPOSE There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on DMBS servers, other than those services necessary for the operation, administration and support of the DBMS.	The DBMS server must not include any general-purpose commuting or storage capabilities. This will protect the TSF data from malicious processes.
A.OS_PP_VALIDATED The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness and that the Operating System provides functionality for • Identification and authentication of users, • Access Control for Files, • Time stamps and • Audit Storage and Audit Review • Hashing of passwords	OE.OS_PP_VALIDATED The underlying OS has to be validated against an NSA sponsored OS PP of at least Basic Robustness and has to provide functionality for • Identification and authentication of users, • Access Control for Files, • Time stamps and • Audit Storage and Audit Review • Hashing of passwords	The underlying OS must be validated to at least basic robustness to ensure it provides an appropriate level of protection for the DBMS. The OS must provide: • Identification and authentication of users, • Access Control for Files, • Time stamps and • Audit Storage and Audit Review • Hashing of passwords

A.PHYSICAL It is assumed that appropriate physical security is provided for the server, on which the TOE is installed, considering the value of the stored, processed, and transmitted information.	OE.PHYSICAL Physical security shall be provided for the server, on which the TOE will be installed, considering the value of the stored, processed, and transmitted information.	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.COMM It is assumed that any communication path from and to the TOE is appropriately secured to avoid eavesdropping and manipulation.	OE.COMM Any communication path from and to the TOE will be appropriately secured to avoid eavesdropping and manipulation.	A.COMM is completely and directly addressed by OE.COMM. OE.COMM and A.COMM both address the requirement that any communication path to and from the TOE has to be appropriately secured.

8.3 Rationale for the TOE and environmental Security Requirements

The following table contains the rationale for the TOE Security Requirements.

Table 16 – Rationale for TOE Security Requirements

Objective	**Requirements Addressing the Objective**	**Rationale**
O.ADMIN_GUIDANCE The TOE will provide administrators with the necessary information for secure management.	ADO_IGS.1	ADO_IGS.1 ensures the administrator has the information necessary to install the TOE in the evaluated 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 it could be used to provide guidance on security that is common to both administrators and non-administrators (e.g., password management guidelines).
O.ADMIN_ROLE The TOE will provide authorized administrators roles to isolate administrative actions.	FMT_SMR.1	The TOE will establish, at least, an authorized administrator role. The authorized administrator will be given privileges to perform certain tasks that other users will not be able to perform. These privileges include, but are not limited to, access to audit

		information and security functions.
O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security relevant events associated with users.	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_EXP.2	FAU_GEN_EXP.2 ensures that the audit records associate a user and/or group identity with the auditable event.
	FAU_SEL.1	FAU_SEL.1 allows the 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.
	FAU_STG_EXP.4	FAU_STG_EXP.4 allows the administrator to define what should happen in the case where the audit file is full. This provides the administrator with the possibility to decide about possible audit data loss or stopping of services based on the information stored in the database.
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.	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_MSA.3	FMT_MSA.3 requires that default values used for security attributes are restrictive.
	FMT_MTD.1	FMT_MTD.1 requires that the ability to manipulate TOE content is restricted to administrators.
	FMT_REV.1(1) FMT_REV.1(2)	FMT_REV.1 restricts the ability to revoke attributes to the administrator
	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.
O.MEDIATE The TOE must protect user data in accordance with its security policy.	FDP_ACC.1	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 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 subjects and objects 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.I&A The TOE will provide a mechanism for identification and authentication of users.	FIA_ATD.1	FIA_ATD.1 defines the user attributes, necessary for authentication.
	FIA_UAU.2	FIA_UAU.2 realizes the authentication part of O.I&A as it requires that each user has to get successfully authenticated before allowing any other TSF-mediated action on behalf of that user.

	FIA_UID.2	FIA_UID.2 realizes the identification part of O.I&A as it requires that each user has to get successfully identified before allowing any other TSF-mediated action on behalf of that user.
	FIA_UAU.5	FIA_UAU.5 specifies that the TOE uses two methods to ensure that every user has to be successfully authenticated. On the one hand the TOE is able to reuse the authentication results from the environment and on the other hand the TOE provides a password based authentication mechanism.

The following table includes the rationale for the IT and Non-IT Environment Requirements.

Table 17 – Rationale for Environment Requirements

Environmental Objective	**Requirements Addressing the Objective**	**Rationale**
OE.NO_EVIL Sites using the TOE shall ensure that authorized administrators are non-hostile, are appropriately trained and follow all administrator guidance.	R.ADMIN	R.ADMIN defines that it has to be ensured that the administrators of the TOE are non-hostile, are appropriately trained and follow all administrator guidance.
OE.NO_GENERAL_ PURPOSE There will be no general-purpose computing capabilities (e.g., compilers or user applications) available on DMBS servers, other than those services necessary for the operation, administration and support of the DBMS.	R.ADMIN	R.ADMIN ensures that the administrators will not install any general purpose computing capabilities on the DBMS server other than necessary for the operation, administration and support of the DBMS.
OE.OS_PP_VALIDATED The underlying OS has been validated against an NSA sponsored OS PP of at least Basic Robustness and has	R.EVL FCS_COP.1/ENV FDP_ACC.1/ENV FDP_ACF.1/ENV FIA_UAU.1/ENV	R.EVL ensures that the underlying Operating System is evaluated against and NSA sponsored PP. FCS_COP.1/ENV defines the hash functionality, which is

to provide functionality for • Identification and authentication of users, • Access Control for Files, • Time stamps and • Audit Storage and Audit Review • Hashing of passwords	FIA_UID.1/ENV FPT_STM.1/ENV FAU_STG.1/ENV FAU_SAR.1/ENV FMT_MSA.3/ENV	used for hashing of passwords. Further FDP_ACC.1/ENV and FPD_ACF.1/ENV describe that the environment has to provide and access control mechanism for the files of the TOE. FMT_MSA.3/ENV, which defines the policy for the default values for this access control mechanism has been used due to a dependency from FDP_ACF.1/ENV. FIA_UAU.1/ENV and FIA_UID.1/ENV ensure that each user has been successfully identified and authenticated before the TOE can be used (for the case that the user has a local account) and FPT_STM.1/ENV ensures that the environment provides the necessary time stamps for the audit functionality of the TOE. Finally FAU_STG.1.1/ENV and FAU_SAR.1 ensure that the environment provides protected audit storage for the audit logs of the TOE and provides the administrator with the functionality to review the audit logs.
OE.PHYSICAL Physical security shall be provided for the server, on which the TOE will be installed, considering the value of the stored, processed, and transmitted information.	R.PHYSICAL	R.PHYSICAL ensures that the environment provides the necessary physical protection for the assets of the TOE.
OE.COMM Any communication path from and to the TOE will be appropriately secured to avoid eavesdropping and manipulation.	R.COMM	R.COMM defines that any communication path to and from the TOE will be secured either by the use of another IT product of by physical protection.

8.3.1 Mutual support and internal consistency of security requirements

From the details given in this rationale it becomes evident that the functional requirements form an integrated whole and, taken together, are suited to meet all security objectives. Requirements from [CC_PART2] and extended requirements are used to fulfill the security objectives.

The core TOE functionality is represented by the requirements for Access Control (FDP_ACC.1, FDP_ACF.1), Security Audit (FAU_GEN.1, FAU_GEN_EXP.2, FAU_SEL.1 FAU_STG_EXP.4), Identification and Authentication (FIA_ATD.1, FIA_UAU.2, FIA_UAU.5, FIA_UID.2), and Security Management (FMT_MOF.1, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_REV.1(1), FMT_REV.1(2), FMT_SMF.1, FMT_SMR.1.

The ST does not contain any SFR with requirements which conflict with other SFRs.

Together with the SARs out of [CC_PART3] the SFRs are suitable to counter the threats against the TOE as shown in the rationale in Table 16.

Therefore it becomes clear that the SFRs in this ST mutually support each other and form a consistent whole.

8.4 Rationale for Assurance Requirements

The table in chapter 6.2 shows how all assurance requirements were satisfied and that there is at least one assurance measure defined in the TOE Summary Specification to meet each of the security assurance requirements.

The "entry level" of EAL 1 has been chosen to gain an initial assurance that all required functionalities are implemented by the TOE.

8.5 Rationale for satisfying all Dependencies

The following table contains the rationale for satisfying all dependencies of the Security Functional Requirements.

Table 18 – Functional Requirements Dependencies for the TOE

Requirement	Dependency	Satisfied
FAU_GEN.1	FPT_STM.1	This requirement is satisfied by the IT environment because the DBMS is a software only TOE.
FAU_GEN_EXP.2	FAU_GEN.1 FIA_UID.1	Satisfied (FIA_UID.2 is hierarchical to FIA_UID.1) The dependency to FIA_UID.1 is either fulfilled by the TOE (for SQL logins) or by the environment (For windows logins).
FAU_SEL.1	FAU_GEN.1 FMT_MTD.1	Satisfied
FAU_STG_EXP.4	FAU_STG.1	The dependency to FAU_STG.1 is satisfied by the environment. The TOE as a DBMS has to rely on the Operating System to protect the files.
FDP_ACC.1	FDP_ACF.1	Satisfied.
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	Satisfied
FIA_ATD.1	None	N/A
FIA_UAU.2	FIA_UID.1	Satisfied (FIA_UID.2 is hierarchical to FIA_UID.1) The dependency to FIA_UID.1 is either fulfilled by the TOE (for SQL logins) or by the environment (for windows

		logins).
FIA_UAU.5	None	N/A
FIA_UID.2	None	N/A
FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	Satisfied.
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1] FMT_SMF.1 FMT_SMR.1	Dependency satisfied by the combination of FDP_ACC.1, FMT_SMF.1 and FMT_SMR.1
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	Satisfied.
FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	Satisfied.
FMT_REV.1(1)	FMT_SMR.1	Satisfied.
FMT_REV.1(2)	FMT_SMR.1	Satisfied.
FMT_SMF.1	None	N/A
FMT_SMR.1	FIA_UID.1	Satisfied (FIA_UID.2 is hierarchical to FIA_UID.1) The dependency to FIA_UID.1 is either fulfilled by the TOE (for SQL logins) or by the environment (For windows logins).

Table 19 – Functional Requirements Dependencies for the IT environment

Requirement	Dependency	Satisfied
FAU_STG.1/ENV	FAU_GEN.1	Satisfied by the TOE
FAU_SAR.1/ENV	FAU_GEN.1	Satisfied by the TOE
FCS_COP.1/ENV	[FDP_ITC.1 or	The dependencies do not need to be addressed as

	FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	FCS_COP.1/ENV is used to define a hash algorithm rather than an algorithm for encryption.
FDP_ACC.1/ENV	FDP_ACF.1	Fulfilled by the use of FDP_ACF.1/ENV
FDP_ACF.1/ENV	FDP_ACC.1 FMT_MSA.3	Fulfilled by the use of FDP_ACC.1/ENV and FMT_MSA.3/ENV.
FIA_UAU.1/ENV	FIA_UID.1	Fulfilled
FIA_UID.1/ENV	-	-
FMT_MSA.3/ENV	FMT_MSA.3/ENV has been introduced as a dependency from FDP_ACF.1/ENV (see also the rationale in Table 17). As only the first level of dependencies is considered during this analysis, the dependencies resulting from FMT_MSA.3/ENV are not relevant.	-
FPT_STM.1/ENV	-	-

The set of assurance requirements is defined to be EAL 1 without any augmentation and thus all dependencies of the assurance requirements are automatically met.

8.6 Rationale for Explicit Requirements

Table 20 presents the rationale for the inclusion of the explicit functional and assurance requirements.

Table 20 – Rationale for Explicit Requirements

Explicit Requirement	Identifier	Rationale
FAU_GEN_EXP.2	User and/or group identity association	This requirement was needed to replace FAU_GEN.2 to specify that the TOE only needs to record the identity of the user/group if an event has been caused by a user. However this SFR has been developed

		based on the definition of FAU_GEN.2 and has the same family behaviour.
FAU_STG_EXP.4	Administrable Prevention of audit data loss	It has been necessary to develop this explicit Security Functional Requirement because part II of [CC] does not contain any SFR which allows specifying a set of allowed actions which can be taken in the case where the audit is full. For the TOE described in this ST it was necessary to provide authorized administrators with the possibility to specify what should happen if the audit log is full. However there should only be one action to be taken in this case. However this SFR has been developed based on the definition of FAU_STG.4 and has the same family behaviour except that it is not hierarchical to any other SFR. .

8.7 TOE Summary Specification Rationale

The following table summarizes which SFR is addressed by which Security Function:

Table 21 - Assignment of SFRs to Security Functions

Requirement/Security Function	SF.SM	SF.AC	SF.I&A	SF.AU
FAU_GEN.1				X
FAU_GEN_EXP.2				X
FAU_SEL.1	X			X
FAU_STG_EXP.4.1	X			X
FDP_ACC.1		X		
FDP_ACF.1		X		
FIA_ATD.1	X		X	
FIA_UAU.2			X	
FIA_UAU.5			X	
FIA_UID.2			X	
FMT_MOF.1	X			
FMT_MSA.1	X			
FMT_MSA.3		X		
FMT_MTD.1	X			
FMT_REV.1(1)	X		X	
FMT_REV.1(2)		X		
FMT_SMF.1	X			
FMT_SMR.1	X			

The following paragraphs give the more detailed justification for this rationale.

Table 22 – Rationale for TOE Summary Specification

Requirement	Fulfilled by Security Function	Rationale
FAU_GEN.1	SF.AU	This SFR is addressed completely by SF.AU as this function realizes the Security Audit mechanism of the TOE which logs all events required by FAU_GEN.1 and stores them into files in the environment.

FAU_GEN_EXP.2	SF.AU	This SFR is addressed by SF.AU as this function describes that the TOE stores the following information for every logged event: 1. Date and Time of the event 2. Identity of the user causing the event (if available) 3. ID of the object 4. Outcome (success or failure) of the event
FAU_SEL.1	SF.AU, SF.SM	This SFR is addressed by SF.AU as this Security Function allows in principle to include or exclude auditable events from being audited. However the administration is done using the Security Function SF.SM and SF.SM additionally ensures that only authorized administrators are allowed to use this management functionality.
FAU_STG_EXP.4	SF.AU, SF.SM	SF.AU allows the administrator to specify, what should happen in case the audit file are full. SF.AU is in these cases able to stop the TOE or to overwrite the old audit logs. SF.SM allows the admin to specify this action.
FDP_ACC.1	SF.AC	This SFR is completely addressed by SF.AC as this Security Function describes the Discretionary Access Control Mechanism as realized by the TOE which realizes Access Control based on the identity of the user and of the object.
FDP_ACF.1	SF.AC	This SFR is completely addressed by SF.AC as this Security Function describes the Discretionary Access Control Mechanism as realized by the TOE which invokes the same set of ordered rules as required by FDP_ACF.1
FIA_ATD.1	SF.SM SF.I&A	SF.I&A describes that the TOE maintains a security ID for each login on an instance level and each user on a database level and is able to associate these principals with their assigned roles

		in this way. SF.SM describes, which roles are known by the TOE.
FIA_UAU.2	SF.I&A	SF.I&A specifies that each user has to be successfully identified and authenticated before the TOE allows any other action on behalf of that user. It therefore completely realizes this SFR.
FIA_UAU.5	SF.I&A	SF.I&A describes that depending on the kind of the login the TOE is either reusing authentication results of the environment to authenticate a user or uses a Username/Password based mechanism to identify/authenticate a user. This completely realizes FIA_UAU.5
FIA_UID.2	SF.I&A	SF.I&A specifies that each user has to get successfully identified and authenticated before the TOE allows any other action on behalf of that user. It therefore completely realizes this SFR.
FMT_MOF.1	SF.SM	SF.SM provides the management function to start and stop the Security Audit and restricts the ability to use these functions to authorized administrators.
FMT_MSA.1	SF.SM	SF.SM provides the management function to manage all the security attributes and restricts the ability to use these functions to authorized administrators.
FMT_MSA.3	SF.AC	SF.AC specifies that if a new object is created only the owner(s) and the system administrator have access to this object. . Furthermore only users with a permission to parent objects (e.g. the schema or the database) have the same permission on the new object. In this way SF.AC realizes the policy of restrictive default values as required by FMT_MSA.3
FMT_MTD.1	SF.SM	SF.SM provides the management function to include or exclude events from being audited and restricts the ability to use these functions to authorized administrators.
FMT_REV.1(1)	SF.SM, SF.I&A	SF.SM provides the management functions to revoke security attributes

		associated with users and restricts the ability to use these functions to authorized administrators. SF.I&A specifies that changes to a SQL Server login are immediately applied while changes of a Windows Account name require a log of and log on of that user before they are applied.
FMT_REV.1(2)	SF.AC	SF.AC provides the functionality to revoke security attributes associated with objects and ensures that the revocation of attributes of these objects follows the DAC and all changes are applied immediately.
FMT_SMF.1	SF.SM	SF.SM provides all management functions required by FMT_SMF.1 and therefore completely realizes this SFR.
FMT_SMR.1	SF.SM	SF.SM maintains the role as required by FMT_SMR.1 and therefore completely realizes this SFR.

9 Appendix

9.1 Concept of Ownership Chains

Database Objects within the TOE are not always only passive objects. Some objects refer to other objects. This is especially true for Stored Procedures and Views. When multiple database objects access each other sequentially, the sequence is known as a chain. Although such chains do not independently exist, when the TOE traverses the links in a chain, the TOE evaluates access permissions on the constituent objects differently than it would if it were accessing the objects separately. These differences have important implications for managing security.

Ownership chaining enables managing access to multiple objects, such as multiple tables, by setting permissions on one object, such as a view. Ownership chaining also offers a slight performance advantage in scenarios that allow for skipping permission checks.

9.1.1 How Permissions Are Checked in a Chain

When an object is accessed through a chain, the TOE first compares the owner of the object to the owner of the calling object. This is the previous link in the chain. If both objects have the same owner, permissions on the referenced object are not evaluated. In the context of the Discretionary Access Control Mechanism this is not a circumvention of access control as the owner of an object always has complete control over his objects. So if one user is the owner of both objects, the calling object and the called object, the owner also would have direct access to both objects.

9.1.2 Example of Ownership Chaining

In the following illustration, the July2003 view is owned by Mary. She has granted to Alex permissions on the view. He has no other permissions on database objects in this instance. What happens when Alex selects the view?

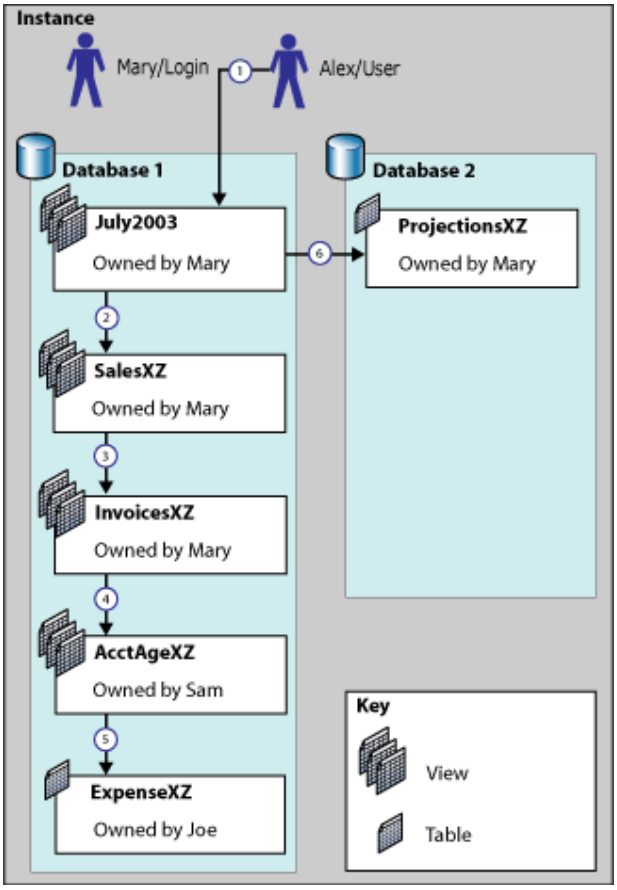

Figure 2: Concept of Ownership Chaining

Alex executes SELECT * on the July2003 view. The TOE checks permissions on the view and confirms that Alex has permission to select on it.

The July 2003 view requires information from the SalesXZ view. The TOE checks the ownership of the SalesXZ view. Because this view has the same owner (Mary) as the view that calls it, permissions on SalesXZ are not checked. The required information is returned.

The SalesXZ view requires information from the InvoicesXZ view. The TOE checks the ownership of the InvoicesXZ view. Because this view has the same owner as the previous object, permissions on InvoicesXZ are not checked. The required information is returned. To this point, all items in the sequence have had one owner (Mary). This is known as an unbroken ownership chain.

The InvoicesXZ view requires information from the AcctAgeXZ view. The TOE checks the ownership of the AcctAgeXZ view. Because the owner of this view is different from the owner of the previous object (Sam, not Mary), full information about permissions on this view is retrieved. If the AcctAgeXZ view has permissions that allow access by Alex, information will be returned.

The AcctAgeXZ view requires information from the ExpenseXZ table. The TOE checks the ownership of the ExpenseXZ table. Because the owner of this table is different from the

owner of the previous object (Joe, not Sam), full information about permissions on this table is retrieved. If the ExpenseXZ table has permissions that allow access by Alex, information is returned.

When the July2003 view tries to retrieve information from the ProjectionsXZ table, the TOE first checks to see whether cross-database chaining is enabled between Database 1 and Database 2. If cross-database chaining is enabled, the TOE will check the ownership of the ProjectionsXZ table. Because this table has the same owner as the calling view (Mary), permissions on this table are not checked. The requested information is returned.

9.2 References

The following documentation was used to prepare this ST:

[CC_PART1] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated August 2005, version 2.3, CCIMB-2005-08-001

[CC_PART2] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional requirements, dated August 2005, version 2.3, CCIMB-2005-08-002

[CC_PART3] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance requirements, dated August 5005, version 2.3, CCIMB-2005-08-003

[CEM] Common Evaluation Methodology for Information Technology Security – Evaluation Methodology, dated August 2005, version 2.3, CCIMB-2005-08-004

[PP] U.S. Government Protection Profile for Database Management Systems in Basic Robustness Environments, Version 1.1, 07.06.2006

[CIM] Consistency Instruction Manual for Development of US Government Protection Profiles for Use in Basic Robustness Environments, Version 3.0 (CIM)

[TSQL] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/acdata/ac_oview_4pcx.asp

[WIN_ST] Microsoft Windows 2003/XP Security Target, Version 1.0. 28.09.2005, Microsoft Corporation

[WIN_VR] National Information Assurance Partnership, Common Criteria Evaluation and Validation Scheme Validation Report Microsoft Windows 2003 Server and XP Workstation Report Number: CCEVS-VR-05-0131 Dated: November 6, 2005 Version: 1.1

[WIN_PP] Controlled Access Protection Profile, Version 1.d, NSA, October, 8th, 1999

9.3 Glossary and Abbreviations

9.3.1 Glossary

The following abbreviations are used in this Security Target:

Abbreviation	Definition
Authorized Administrators	This term refers to a group of users which comprise the “sysadmin” (sa) and any user who is allowed to perform a management operation because the permission has been granted to him within the DAC either by assigning him to a role with administrator permissions or by granting him the possibility to perform an administrative operation explicitly.
DAC	Discretionary Access Control is a mechanism to limit the access of users to objects based on the ID of the user, the ID of the object and a set of access control rules.
DBMS	A DBMS is a computerized repository that stores information and allows authorized users to retrieve and update that information.
Object	An object within the TOE contains data and can be accessed by subjects. However in the TOE an object is not necessarily only a passive entity as some objects refer to other objects.
OC	Ownership Chaining. Explained in chapter 9.1 in more detail.
SQL	The Structured Query Language is a language which can be used to create, modify and retrieve data from a DBMS.
SQL Server	SQL Server is a product of Microsoft to which the TOE belongs.
TDS	Tabular Data Stream is a data format which is used for communication with the TOE.
T-SQL	Extension of the SQL language in order to support control flow, variables, user authentication and various other functions. See also http://msdn.microsoft.com/library/default.asp?url=/library/en-us/acdata/ac_oview_4pcx.asp
Named Pipe	Method for inter process communication

9.3.2 Abbreviations

The following abbreviations are used in this Security Target:

Abbreviation	Definition
ACL	Access Control List
CC	Common Criteria
CCEVS	Common Criteria Evaluation and Validation Scheme
CEM	Common Evaluation Methodology
CIM	Consistency Instruction Manual
DAC	Discretionary Access Control
DBMS	Database Management System
EAL	Evaluation Assurance Level
ETL	Extract, Transform, Load
IT	Information Technology
MOM	Microsoft Operations Manager
MS	Microsoft
NIAP	National Information Assurance Partnership
NSA	National Security Agency
OC	Ownership Chaining
ODS	Open Data Services
OLAP	Online analytical processing
OS	Operating System
OSP	Organizational Security Policy
PP	Protection Profile
sa	System administrator
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SID	Security ID
SMS	System Management Server
SOF	Strength of Function
SQL	Structured Query Language
ST	Security Target
TDS	Tabular Data Stream
TOE	Target of Evaluation

Abbreviation	Definition
TSC	TSF Scope of Control
TSF	TOE Security Functions
T-SQL	Transact SQL