CRUNCHY CERTIFIED POSTGRESQL 12

SECURITY TARGET VERSION 1.8

March 16, 2021

Prepared By:

Crunchy Data Solutions, Inc.

162 Seven Farms Drive, Suite 220
Charleston, SC 29492
www.crunchydata.com

Table of Content

Table of Figures

Table of Tables

Table of Appendixes

1. SECURITY TARGET INTRODUCTION

1.1. Security Target Reference

ST Title:	Crunchy Certified PostgreSQL 12 Security Target
ST Version:	Version 1.8
ST Date:	March 16, 2021
ST Author:	Crunchy Data Solutions, Inc.

1.2. TOE Reference

TOE Identification:	Crunchy Certified PostgreSQL 12
TOE Vendor:	Crunchy Data Solutions, Inc.
TOE Build:	12.5

1.3. TOE Terminology

Terms not explicitly otherwise defined in this Security Target shall have the meaning set forth in Appendix A-1: PostgreSQL Terminology and Appendix A-2: DBMS PP Terminology, each as attached to this Security Target.

Acronyms not explicitly otherwise defined in this Security Target shall have the meaning set forth in Appendix B-1: PostgreSQL Acronyms and Appendix B-2: CC Acronyms, each as attached to this Security Target.

1.4 TOE Overview

Crunchy Certified PostgreSQL 12 (also referred to as PostgreSQL) is an open source relational database management system. The TOE includes PostgreSQL and tools for clients, developers and administrators.

The TOE provides the following security functionality: Security Auditing, User Data Protection, Identification and Authentication (I&A), Security Management, Protection of the TSF and TOE Access.

The TOE is being evaluated at assurance level EAL2 augmented by ALC_FLR.2

The TOE is claiming conformance to the *Protection Profile for Database Management Systems (Base Package), Version 2.12, March 23, 2017.* This PP is referred to in this ST as the DBMS PP.

1.4.1 TOE Type

The TOE Type is a database management system (DBMS).

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

1.4.2 Operating System Platform

The TOE server will be evaluated running on the following operating system platforms:

- Red Hat Enterprise Linux Version 7.8 (RHEL 7)
- Red Hat Enterprise Linux Version 8.2 (RHEL 8)

The Client Connectors (JDBC and libpq) included as TOE components must be installed on each client system. The Administrative and Development Platform includes the standard PostgreSQL libpq Client Connector to support the psql CLI.

Any client system can be used as the Administrator Workstation, thus there is no need for an additional administrator workstation unless it is operationally desired.

1.5. TOE Description

Crunchy Certified PostgreSQL 12 is a software-only TOE. The TOE is made up of the following software components:

1. PostgreSQL 12
2. Client Connectors identified in Section 1.5.2
3. PostgreSQL Audit Extension
4. PostGIS Spatial Extensions

Each of the listed components is described in the sections below. A rationale is also provided for those components not in the TOE.

The TOE is installed using a RPM. The RPM provided contains the TOE's components and is installed by the Linux system administrator using RPM client utilities. The TOE has a separate set of RPM packages for each component, including the following:

- PostgreSQL Server RPM
- PostgreSQL JDBC Driver RPM
- PostgreSQL Audit RPM
- PostGIS RPM

The TOE is delivered by website download through a yum repository made available by Crunchy Data. Prior to downloading the TOE RPM from the yum repository, subscribers are required to authenticate using a unique login name and password combination.

1.5.1 PostgreSQL

PostgreSQL is a computerized repository that stores information and allows authorized users to retrieve and update that information. PostgreSQL may be operated as 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.

PostgreSQL has the capability to limit DBMS access to authorized users, enforce DAC on objects under the control of the DBMS (based on user and optional group authorizations), and provide user accountability via the audit of user actions.

PostgreSQL is comprised of the DBMS server application that performs the following functions:

- Controls users' accesses to user data and DBMS data;
- Interacts with, and possibly supplements portions of, the underlying operating system to retrieve and present the data that are under the DBMS's management;
- Indexes data values to their physical locations for quick retrievals based on a value or range of values;
- Executes pre-written programs (i.e., utilities) to perform common tasks like database backup, recovery, loading, and copying;
- Supports mechanisms that enable concurrent database access (e.g., locks);
- Assists recovery of user data and DBMS data (e.g., transaction log);
- Tracks operations that users perform;
- Implements a data model with which the DBMS data structures and organization can be conceptualized (e.g., hierarchical, object-oriented, relational data models) and DBMS objects defined; and
- Implements high-level language(s) or interfaces that allow authorized users to define database constructs; access and modify user or DBMS data; present user or DBMS data; and perform operations on those data.

PostgreSQL includes the following subcomponents:

- **Server Utilities** are a collection of command line utilities for managing the database. These utilities are only useful when run on the host system where the database server resides.
- **Database Utilities** allow for the creation and removal of databases, database user accounts and retrieving information about the installed version. These command line utilities can be run from a terminal emulation program on any host, independent of where the database server resides.
- **Authentication Support.** PostgreSQL provides support for multiple authentication mechanisms. Please see Section 1.5.10.3 Identification and Authentication for more information.
- **psql:** CLI to PostgreSQL

1.5.2 Client Connectors

Client Connectors are standardized programming interfaces allowing a software developer to connect a customer-specific application to PostgreSQL.

The TOE includes Client Connectors for the following enterprise programming environments:

- **Java Database Connectivity (JDBC)** is a Java database connectivity technology (Java Standard

Edition platform) from Oracle Corporation. This technology is an Application Programming Interface (API) for the Java programming language that defines how a client may access a database. It provides methods for querying and updating data in a database. JDBC is oriented towards DBMS.

- **Libpq, an API for client applications written in C**, is the C application programmer's interface to PostgreSQL. libpq is a set of library functions that allow client programs to pass queries to the PostgreSQL backend server and to receive the results of these queries.

1.5.3 PostgreSQL Audit Extension

PostgreSQL Audit, an open source audit log generator, is included in the TOE. PostgreSQL Audit extends the logging capability supported by PostgreSQL by providing detailed logging classes, the ability to control logging at a per-object level, and including fully-qualified object names for logged statements in independent fields of the log output.

1.5.4 PostGIS Spatial Extensions

PostGIS, an open source Geographic Information Server (GIS), is included in the TOE. PostGIS spatially enables PostgreSQL, allowing it to be used as a backend spatial database for geographic information systems. This is a non-security related component.

1.5.5 Users

The users supported by the TOE are the same as those defined in the DBMS PP. The DBMS PP text is copied below:

"*A DBMS supports two major types of users:*

- *Users who interact with the DBMS to observe and/or modify data objects for which they have authorization to access; and*

- *The authorized administrators who implement and manage the various information-related policies of an organization (e.g., access, integrity, consistency, availability) for the databases that they install, configure, manage, and/or own*."

PostgreSQL supports the first major type of user defined in the DBMS PP through the use of Roles. A Role is used in PostgreSQL to define individual users, groups of users and sets of access privileges.

PostgreSQL supports the second major type of user defined in the DBMS PP, specifically authorized administrators, through:

(i) the Superuser, which is a role maintained by the TOE and

(ii) the Cluster Owner, which is created during the installation of the TOE and is maintained by the operating system.

The Superuser administers the TOE through the TOE's user interfaces and is the focus of the SFR's described in this ST. The Cluster owner has the OS permissions to modify configuration files stored at the OS level and execute command line interfaces.

1.5.6 Data

The data maintained by the TOE is the same as the definition of DBMS data in the DBMS PP as follows:

"*A DBMS stores, and controls access to, two types of data:*

- *The first type is the user data that the DBMS maintains and protects. User data may consist of the following:*

    a) The user data stored in or as database objects;

    b) The definitions of user databases and database objects, commonly known as DBMS metadata; and

    c) The user-developed queries, functions, or procedures that the DBMS maintains for users.

- *The second type is the DBMS data (e.g., configuration parameters, user security attributes, transaction log, audit instructions, and records) that the DBMS maintains and may use to operate the DBMS.*"

1.5.7 Product Guidance

The following product guidance documents are provided with the TOE. The documents are available to download from the TOE Vendor website.

Table 1-1: User Guidance Documents Reference

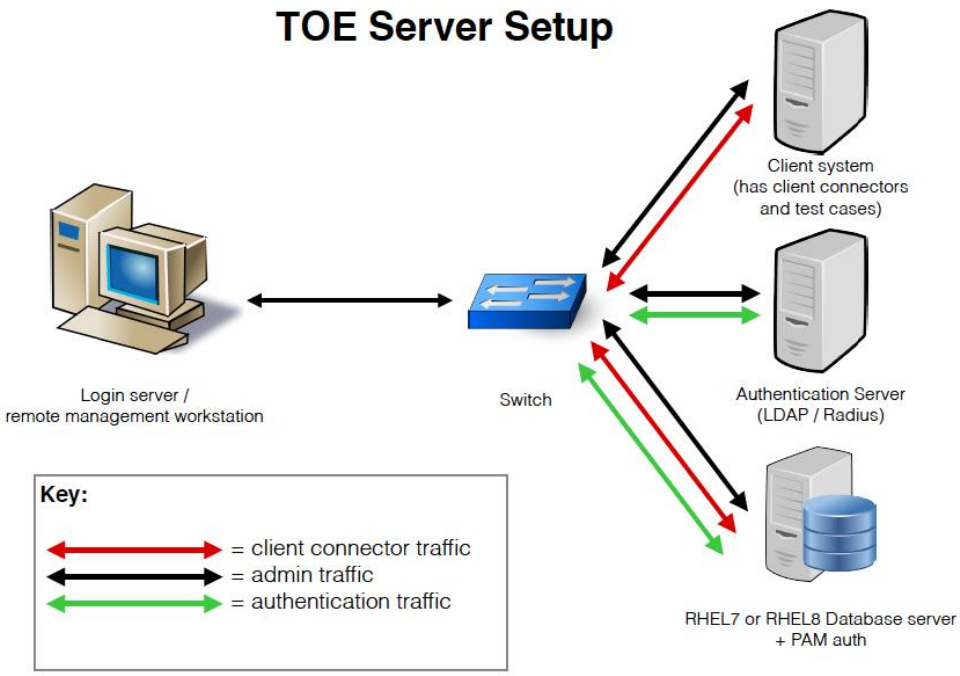
Title	Version	ID
The PostgreSQL Global Development Group; PostgreSQL 12 Documentation, Version 12	12.5	PG
The PostgreSQL JDBC Interface	42.2.18	JDBC
The PostgreSQL Global Development Group; PostgreSQL 12 Documentation, Chapter 31	12.5	libpq
PostgreSQL Audit Extension User Guide	1.4.1	pg_audit
PostGIS 3 Manual	3.0.1	PostGIS
Crunchy Data Secure Installation and Configuration Guide, Version 2.1	2.1	CDUG
Supporting Documentation: Examples of Auditable Events	1.1	SDAU

1.5.8 Sample Configuration of the TOE

Figure 1 depicts a sample configuration of the TOE within its IT environment. Specifically, it shows a sample stand-alone system running the PostgreSQL server on Server 1.

User Applications (outside of the TOE) are shown running on each of Server 2 and Server 3. Each User Application has a Client Connection to the stand-alone PostgreSQL server running on Server 1. The sample stand-alone PostgreSQL server configuration reflected in Figure 1 contains the two Client Connectors supported by the TOE: JDBC, and libpq.

FIGURE 1: TOE SAMPLE CONFIGURATION



1.5.9 Physical Scope of the TOE

Crunchy Certified PostgreSQL 12 is a software-only TOE. The physical scope of the TOE is an RPM.

1.5.9.1 In Scope of Evaluation

The following Crunchy Certified PostgreSQL 12 components are in scope of evaluation:

- PostgreSQL
- Client Connectors
- PostgreSQL Audit Extension
- PostGIS Spatial Extensions

1.5.9.2 Components and Capabilities that are Out of Scope

The TOE relies on the operating system, which is part of the operational environment, to provide cryptographic support for communications, such as SSL encryption. Thus, cryptographic functionality is outside the scope of this evaluation. Specifically, support for secure communication channel and certificate-based authentication was not evaluated.

Crunchy PostgreSQL provides synchronous streaming replication as a way to replicate changes to data on

one database server to the other database servers within a cluster. The evaluated TOE architecture is a stand-alone system running a single PostgreSQL server. Thus, streaming replication functionality is outside the scope of this evaluation.

As described in more detail in Section 1.5.9.3, use of the "Trust", "Ident", "SSL" and "SSPI" authentication methods are outside the scope of this evaluation.

1.5.9.3 Configuration Options that are Out of Scope.

1.5.9.3.1 "Trust" authentication option (not in TOE)

When the trust authentication option is specified, PostgreSQL assumes that anyone who can connect to the server is authorized to access the database with whatever database user name they specify (including Superusers). The use of the PostgreSQL "trust" authentication option is prohibited in the evaluated configuration, since it configures the TOE to not require any authentication functionality.

1.5.9.3.2 "Ident" authentication option (not in TOE)

The "Identification Protocol" is described in RFC 1413. This authentication method is only appropriate for closed networks where each client machine is under tight control and where the database and system administrators operate in close contact. In other words, the system administrators must trust the machine running the ident server. RFC 1413 issues the following warning: "*The Identification Protocol is not intended as an authorization or access control protocol."* Therefore, the use of the "Ident" authentication option is prohibited in the evaluated configuration.

1.5.9.3.3 "SSPI" authentication option (not in TOE)

The "SSPI" authentication is a Windows technology for secure authentication with single sign-on. The TOE server will be not evaluated running the Windows operating system platforms. Therefore, the use of the "SSPI" authentication option is prohibited in the evaluated configuration.

1.5.9.3.4 "SSL" authentication option (not in TOE)

SSL Certificates functionality may be created and used within the TOE, though proper usage requires the SSL Certificate be signed by an external entity. The TOE relies on the operating system, which is part of the operational environment, to provide cryptographic support for communications, such as SSL encryption. Thus, cryptographic functionality is outside the scope of this evaluation. Therefore, the use of the "SSL" authentication option is prohibited in the evaluated configuration.

1.5.9.3.5 GSSAPI Authentication (not in TOE)

GSSAPI is an industry-standard protocol for secure authentication defined in RFC 2743. PostgreSQL supports GSSAPI with Kerberos authentication according to RFC 1964. GSSAPI provides automatic authentication (single sign-on) for systems that support it. The TOE server will be not evaluated running the supported GSSAPI authentication method. Therefore, the use of "GSSAPI" authentication option is prohibited in the evaluated configuration.

1.5.9.3.6 Peer Authentication (not in TOE)

The Peer authentication method works by obtaining the client's operating system user name from the kernel and using it as the allowed database user name (with optional user name mapping). This method is only

supported on local connections. The TOE server will be not evaluated running the supported "Peer" authentication method. Therefore, the use of “Peer” authentication option is prohibited in the evaluated configuration.

1.5.9.3.7 Streaming Replication Configuration (not in TOE)

Crunchy PostgreSQL provides synchronous and asynchronous streaming replication as a way to replicate changes to data on one database server to the other database servers within a cluster. The evaluated TOE architecture is a stand-alone system running a single PostgreSQL server. Therefore, the use of the streaming replication configuration is prohibited in the evaluated configuration.

1.5.9.3.8 Logical Replication Configuration (not in TOE)

Crunchy PostgreSQL provides logical replication as a way to replicate changes to data on one database server to the other database servers within a cluster. The evaluated TOE architecture is a stand-alone system running a single PostgreSQL server. Therefore, the use of the logical replication configuration is prohibited in the evaluated configuration.

1.5.9.4 TOE Operational Environment

1.5.9.4.1 Enclave

The TOE is anticipated to operate within an Enclave and be protected by network segmentation and/or network firewall devices, as dictated by the Enclave design.

The TOE is anticipated to operate within an Enclave that is under the control of a single authority and has a homogeneous security policy, including personnel and physical security, to protect it from other environments. This Enclave could be specific to an organization or a mission and it may contain multiple networks. The Enclave may be logical, such as an operational area network, or be based on physical location and proximity. Any local and external elements that access resources within the Enclave must satisfy the policy of the enclave.

The DBMS is expected to interact with other IT products that reside in the host operating systems, in the IT environment in which the host computer and host operating systems reside, and outside that environment but inside the Enclave. The IT and non-IT mechanisms used for secure exchanges of information between the DBMS and such products are expected to be administratively determined and coordinated. Similarly, the IT and non-IT mechanisms for negotiating or translating the DAC policy involved in such exchanges are expected to be resolved by the organizations involved.

The DBMS may also interact with IT products outside the Enclave such as a CA that is defined as a trusted CA by an IT product within the Enclave.

1.5.9.4.2 TOE Architecture

The TOE can support a network of workstations communicating with several distributed PostgreSQL servers simultaneously. For the purposes of evaluation, the PostgreSQL servers will all be within a single LAN as per Figure 1.

The TOE architecture is an Enclave in which users access the TOE via a LAN. Users in other Enclaves will access the LAN and the host computers and servers on it by way of one or more boundary protection mechanisms (e.g., a firewall) and then through a communications server or router to the LAN. Depending

on the particular Enclave configuration and the DBMS access policy that it supports, all users (both inside and outside the Enclave) may then access an application server, which either connects the TOE user to the Enclave computer on which the TOE operates or manages the complete user/DBMS session.

No operating systems or platforms are included in the TOE.

In addition, the following components in the IT environment are out of scope:

- Authenticator servers, if configured; and
- Terminal emulator

1.5.9.4.3 Functional Dependencies on the IT Environment

The TOE relies on the IT environment for the following security functionality:

- Storage of audit records in operating system files;
- Text Viewer to review audit records;
- Client connectors to initiate authentication from the client application;
- I&A methods that rely upon authentication servers and/or operating system platforms in the IT environment (PAM, LDAP, Radius);
- I&A of the Cluster Owner;
- Maintenance of Cluster Owner's password and security attributes;
- Storage of the TOE configuration files;
- Text Editor to edit the TOE's configuration files stored at the operating system level;
- Reliable timestamps from the operating system;
- Operating system protection of TOE programs and data (audit, configuration files, executables, and database); and
- PostgreSQL relies on the operating system to provide cryptographic support for communications as described in more detail below.

The TOE relies on the IT environment for the following hardware / software / firmware:

Table 1-2: Minimum Hardware / Software

Hardware	**Minimum Requirement**
CPU	1 CPU
RAM	256MB
Hard Disk	700Mb
Software	**Minimum Requirement**
Operating System	Red Hat Enterprise Linux 7.8 Red Hat Enterprise Linux 8.2
Kernel	3.8.x
Architecture	x86_64
GNU Make	3.8
GNU Readline	6.3
Tar	1.28
Gzip	1.2.4

The minimum hardware and software requirements provided in Table 1-2 are provided purely as minimum requirements. Actual system requirements will depend on the details of a user's application. As a starting point, Crunchy provides the following general guidance.:

- **Memory.** If possible, the amount of memory available to PostgreSQL should equal the amount of space the database working set (that part of the database actively being used at any given time) occupies on disk, plus enough space for the maximum number of expected concurrently active connections, ideally with room to grow.

- **Persistent Storage.** The amount of storage required by the TOE is dependent on the user's application and data retention requirements. The type of storage recommended (e.g. SAS hard drives or SSD), and storage configuration (e.g. RAID 1+0 or RAID 5) are also highly dependent on the application.

- **CPU.** x86_64 architecture is recommended. In general, PostgreSQL leverages multiple virtual CPUs very well. However, PostgreSQL does not allow a user to run a single across multiple CPUs.

PostgreSQL relies on Red Hat Enterprise Linux to provide cryptographic support for communications, specifically the OpenSSL library that provides cryptographic protocols. For Red Hat Enterprise Linux 7.8, PostgreSQL relies on the OpenSSL library, Cryptographic Module Validation Program (CMVP) certificate # 3538. For Red Hat Enterprise Linux 8.2, PostgreSQL relies on the OpenSSL library, CMVP certificate #3781.

For additional information regarding system recommendations, please see the Secure Installation and Configuration Guide provided by Crunchy Data.

1.5.10 Logical Scope of the TOE

1.5.10.1 Security Audit

The TOE generates audit records for security relevant events using a standard logging facility that is controlled by a system configuration file. For security relevant events resulting from actions of users or groups, the TOE associates them with the user or group that caused such event.

The TOE provides the capability to select auditable events and determine the information to be included in

the audit record based on settings in system configuration files.

For additional information regarding the TOE's Security Audit functionality, see Section 7.1.1 Security Audit Functions.

1.5.10.2 User Data Protection

The TOE provided DAC controls access to objects on all subjects, all DBMS-controlled objects, and all operations among them. The TOE enforces DAC to objects based on the identity of the subjects or groups to which the subjects and objects belong, with access operations implemented for DBMS-controlled objects and object identity.

The TOE allows authorized administrators to specify how the objects that they control are protected. The TOE provides the capability to grant privileges both on RDBMS objects (such as tables, columns, views, Triggers, Functions, Procedures, Tablespaces and Schemas) and to Roles. The TOE also provides for the inheritance of privileges between Roles. Explicit delegation of privileges on a database object among users is also permitted.

Residual information protection is enforced within the TOE through the implementation of a "write before read" mechanism.

For additional information regarding the TOE's User Data Protection functionality, see Section 7.1.2 User Data Protection.

1.5.10.3 Identification and Authentication

The TOE requires that each user is identified and authenticated prior to allowing any actions on behalf of the user. Further, the TOE requires that users are identified and authenticated by some method before allowing them access to TSF resources.

The available methods (*auth-method: parameter*) for client authentication definition include:

- Password (*password*)
- Pluggable Authentication Modules (*pam*)
- Lightweight Directory Access Protocol (*ldap*)
- RADIUS Authentication (*radius*)

Password authentication is provided wholly within the TOE and available in "scram-sha-256", "md5" and "password" methods.

PAM, LDAP and RADIUS authentication are provided with the support of an external authentication mechanism provided by the IT environment.

Note: The scram-sha-256 implementation is vendor developed to the RFC 7677 specification and is strictly used for password hashing. The scram-sha-256 cryptographic function implementation, or module, has not been FIPS certified. The correctness of the cryptographic module used by the TOE is by Vendor assertion; the correctness and conformance of this cryptographic module to the RFC 7677 standard is not to be part of this evaluation.

Note: The MD5 implementation is vendor developed to the RFC 1321 specification and is strictly used for

password hashing. The MD5 cryptographic function implementation, or module, has not been FIPS certified. The correctness of the cryptographic module used by the TOE is by Vendor assertion; the correctness and conformance of this cryptographic module to the RFC 1321 standard is not to be part of this evaluation.

The TOE associates user security attributes with subjects acting on the behalf of that user through a series of Role security attributes. Only users with sufficient privileges may modify the Role security attributes associated with subjects acting on behalf of users.

For additional information regarding the TOE's Identification and Authentication functionality, see Section 7.1.3 Identification & Authentication.

1.5.10.4 Security Management

The TOE provides security management through the server command line utilities and database command line utilities.

The TOE restricts the ability to perform security management functions to the authorized administrator or users with proper privileges (specifically, the CREATEDB or CREATEROLE Roles). In PostgreSQL, the authorized administrator is referred to as the Superuser.

For additional information regarding the TOE's Security Management functionality, see Section 7.1.4 Security Management.

1.5.10.5 Protection of the TSF

The TOE provides protection of the TSF through support of secure initialization process, self-protection of the TSF from tampering, non-bypassability of the SFR-enforcing functionality, and separation of the security domains.

For additional information regarding the TOE's Protection of TSF functionality, see Section 7.1.5 Protection of the TSF.

1.5.10.6 TOE Access

The TOE is able to restrict the maximum number of concurrent sessions that belong to the same user. The number of multiple concurrent sessions per user is determined by the "connection limit" role security attribute. The "connection limit" is checked during session establishment and is configurable by an authorized administrator.

The TOE provides users with the ability to view their own connection history based on information recorded in an audit log. Upon a session establishment attempt, the TSF stores the date and time of the session establishment attempt of the user and the incremental count of successive unsuccessful session establishment attempts by the user. The TOE allows the user to retrieve the date and time of the previous last successful session establishment, the last unsuccessful attempt to session establishment, and the number of unsuccessful attempts since the previous last successful session establishment.

The TOE can deny session establishment based on user including user identity, time of day, day of the week, group identity, database name, Host IP address, and/or subnet address.

For additional information regarding the TOE's Protection of TSF functionality, see Section 7.1.6 TOE

Access.

2. CONFORMANCE CLAIMS

2.1 Common Criteria Conformance

The TOE is Part 2 extended, Part 3 conformant, and meets the requirements of EAL2 augmented by ALC_FLR.2 Flaw reporting procedures from Common Criteria for Information Technology Security Evaluation, April 2017, Version 3.1 Revision 5.

2.2 Protection Profile Claim

The TOE claims demonstrable conformance to the *Protection Profile for Database Management Systems (Base Package), Version 2.12, March 23, 2017.* This PP is referred to in this ST as the DBMS PP.

Demonstrable conformance is defined in Section D.3 of CC Version 3.1 Revision 4 as follows:

"*Demonstrable conformance is orientated to the PP-author who requires evidence that the ST is a suitable solution to the generic security problem described in the PP.*

Where there is a clear subset-superset type relation between PP and ST in the case of strict conformance, the relation is less clear-cut in the case of demonstrable conformance. STs claiming conformance with the PP must offer a solution to the generic security problem described in the PP, but can do so in any way that is equivalent or more restrictive to that described in the PP."

2.3 Rationale Correctness Claim

All of the rationale for Threats, Assumptions, Policies, Objectives, Security Functional Requirements, and Assurance activities are direct copies from the DBMS PP. Any deficiencies found would need to be addressed to the DBMS PP authors.

3. SECURITY PROBLEM DEFINITION

In this section, the Security Problem Definition (SPD) for a DBMS is described. First, the informal discussion of the SPD is presented followed by a more formal description in terms of the identified threats, policies, and assumptions that will be used to identify the specific security requirements addressed by the DBMS PP.

3.1 Informal Discussion

Given their common usage as repositories of high value data, attackers routinely target DBMS installations for compromise. Vulnerabilities that attackers may take advantage of are:

- Design flaws and programming bugs in the DBMS and the associated programs and systems, creating various security vulnerabilities (e.g. weak or ineffective access controls) which can lead to data loss/corruption, performance degradation etc.

- Unauthorized or unintended activity or misuse by authorized database users, or network/systems managers, or by unauthorized users or hackers (e.g. inappropriate access to sensitive data, metadata or functions within databases, or inappropriate changes to the database programs, structures or security configurations).

- Malware infections causing incidents such as unauthorized access, leakage or disclosure of personal or proprietary data, deletion of or damage to the data or programs, interruption or denial of authorized access to the database, attacks on other systems and the unanticipated failure of database services.

- Data corruption and/or loss caused by the entry of invalid data or commands, mistakes in database or system administration processes, sabotage/criminal damage etc.

3.2 Assets and Threat Agents

The threats given in Section 3.2 refer to various threat agents and assets. The term “threat agent” is defined in CC Part 1. The term “A user or a process acting on behalf of a user” used in this ST, specifies a particular class of entities that can adversely act on assets.

The assets, mentioned in Table 3-1 below are either defined in CC part 1 [REF 1a], or in the glossary given in Appendix A or Appendix B of this document.

The terms “TSF data”, “TSF" and “user data”, are defined in CC Part 1. The terms “executable code within the TSF”, “public objects”, “TOE resources” and “configuration data” are given in the glossary given in Appendix A or Appendix B of this document.

3.3 Threats

The following threats are identified and addressed by the TOE, and should be read in conjunction with the threat rationale, in Section 4.3.

Compliant TOEs will provide security functionality that addresses threats to the TOE and implements policies that are imposed by law or regulation.

Table 3-1: Threats Applicable to the TOE

Threat	Definition
T.ACCESS_TSFDATA	A threat agent may read or modify TSF data using functions of the TOE without the proper authorization.
T.ACCESS_TSFFUNC	A threat agent may use or manage TSF bypassing the protection mechanisms of the TSF.
T.IA_MASQUERADE	A user or a process acting on behalf of a user may masquerade as an authorized entity in order to gain unauthorized access to user data, TSF data, or TOE
T.IA_USER	A threat agent may gain access to user data, TSF data, or TOE resources with the exception of public objects without being identified and authenticated.
T.RESIDUAL_DATA	A user or a process acting on behalf of a user may gain unauthorized access to user or TSF data through reallocation of TOE resources from one user or process
T.TSF_COMPROMISE	A user or a process acting on behalf of a user may cause configuration data to be inappropriately accessed (viewed, modified or deleted), or may compromise executable code within the TSF.
T.UNAUTHORIZED_ACCESS	A threat agent may gain unauthorized access to user data for which they are not authorized according to the TOE security policy.

3.4 Organizational Security Policies

The following organizational security policies are addressed by PP-conformant TOEs:

Table 3-2: Policies Applicable to the TOE

Policy	Definition
P.ACCOUNTABILITY	The authorized users of the TOE shall be held accountable for their actions within the TOE.
P.ROLES	Administrative authority to TSF functionality shall be given to trusted personnel and be as restricted as possible supporting only the administrative duties the person has. This role shall be separate and distinct from other authorized users.
P.USER	Authority shall only be given to users who are trusted to perform the actions correctly.

3.5 Assumptions

This section contains assumptions regarding the IT environment in which the TOE will reside.

Table 3-3: Assumptions Applicable to the TOE Environment

Assumption	Definition
Physical aspects	
A.PHYSICAL	It is assumed that the IT environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the
Personnel aspects	
A.AUTHUSER	Authorized users possess the necessary authorization to access at least some of the information managed by the TOE.
A.MANAGE	The TOE security functionality is managed by one or more competent administrators. The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the guidance documentation.
A.TRAINEDUSER	Users are sufficiently trained and trusted to accomplish some task or group of tasks within a secure IT environment by exercising complete control over their user data.
Procedural aspects	
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.PEER_FUNC_&_MGT	All remote trusted IT systems trusted by the TSF to provide TSF data or services to the TOE, or to support the TSF in the enforcement of security policy decisions are assumed to correctly implement the functionality used by the TSF consistent with the assumptions defined for this functionality and to be properly managed and operate under security policy constraints compatible with those of the TOE.
A.SUPPORT	Any information provided by a trusted entity in the IT environment and used to support the provision of time and date, information used in audit capture, user authentication, and authorization that is used by the TOE is correct and up to date.
Connectivity aspects	
A.CONNECT	All connections to and from remote trusted IT systems and between separate parts of the TSF not protected by the TSF itself are physically or logically protected within the TOE environment to ensure the integrity and confidentiality of the data transmitted and to ensure the authenticity of the communication end points.

4. SECURITY OBJECTIVES

This section identifies the Security Objectives of the TOE and its supporting environment.

These security objectives identify the responsibilities of the TOE and its environment in meeting the security problem definition (SPD).

4.1 TOE Security Objectives

Table 4-1: TOE Security Objectives

Objective Name	Objective Definition
O.ADMIN_ROLE	The TOE will provide a mechanism (e.g. a "role") by which the actions using administrative privileges may be restricted.
O.AUDIT_GENERATION	The TSF must be able to record defined security-relevant events (which usually include security-critical actions of users of the TOE). The information recorded for security-relevant events must contain the time and date the event happened and, if possible, the identification of the user that caused the event, and must be in sufficient detail to help the authorized user detect attempted security violations or potential misconfiguration of the TOE security features that would leave the IT assets open to compromise.
O.DISCRETIONARY_ACCESS	The TSF must control access of subjects and/or users to named resources based on identity of the object, subject, or user. The TSF must allow authorized users to specify for each access mode which users/subjects are allowed to access a specific named object in that access mode.
O.I&A	The TOE ensures that users are authenticated before the TOE processes any actions that require authentication.
O.MANAGE	The TSF must provide all the functions and facilities necessary to support the authorized users that are responsible for the management of TOE security mechanisms, must allow restricting such management actions to dedicated users, and must ensure that only such authorized users are able to access management functionality.
O.MEDIATE	The TOE must protect user data in accordance with its security policy, and must mediate all requests to access such data.
O.RESIDUAL_INFORMATION	The TOE will ensure that any information contained in a protected resource within its Scope of Control is not inappropriately disclosed when the resource is reallocated.
O.TOE_ACCESS	The TOE will provide functionality that controls a user's logical access to user data and to the TSF.

4.2 Operational Environment Security Objectives

Table 4-2: Operational Environment Security Objectives

Objective Name	Definition
OE.ADMIN	Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains.
OE.INFO_PROTECT	Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: • All network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.
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.PHYSICAL	Those responsible for the TOE must ensure that those parts of the TOE critical to enforcement of the security policy are protected from physical attack that might compromise IT security objectives. The protection must be commensurate with the value of the IT assets protected by the TOE.

Table 4-3: Operational Environment IT Security Objectives

Objective Name	Definition
OE.IT_I&A	Any information provided by a trusted entity in the environment and used to support user authentication and authorization used by the TOE is correct and up to date.
OE.IT_REMOTE	If the TOE relies on remote trusted IT systems to support the enforcement of its policy, those systems provide that the functions and any data used by the TOE in making policy decisions, required by the TOE are sufficiently protected from any attack that may cause those functions to provide false results.
OE.IT_TRUSTED_SYSTEM	The remote trusted IT systems implement the protocols and mechanisms required by the TSF to support the enforcement of the security policy. These remote trusted IT systems are managed according to known, accepted, and trusted policies based on the same rules and policies applicable to the TOE, and are physically and logically protected equivalent to the TOE.

4.3 Rationale for TOE Security Objectives

The table below gives the rationale for the TOE security objectives. The Rationale for the TOE Security Objectives tables are directly copied from the DBMS PP.

Table 4-4: Rationale for the TOE Security Objectives

Threat/Policy	TOE Security Objectives Addressing the Threat/Policy	Rationale
P.ACCOUNTIBLITY The authorized users of the TOE shall be held accountable for their actions within the TOE.	O.ADMIN_ROLE The TOE will provide a mechanism (e.g. a "role") by which the actions using administrative privileges may be restricted.	O.ADMIN_ROLE Supports this policy by ensuring that the TOE has an objective to provide authorized administrators with the privileges needed for secure administration.
	O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security relevant events associated with users.	O.AUDIT_GENERATION Supports this policy by ensuring that audit records are generated. Having these records available enables accountability.
	O.I&A The TOE ensures that users are authenticated before the TOE processes any actions that require authentication.	O.I&A Supports this policy by requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action the TOE is defined to provide to authenticated users only.
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to user data and to the TSF.	O.TOE_ACCESS Supports this policy by providing a mechanism for controlling access to authorized users.
P.USER Authority shall only be given to users who are trusted to perform the actions correctly.	O.MANAGE The TSF must provide all the functions and facilities necessary to support the authorized users that are responsible for the management of TOE security mechanisms, must allow restricting such management actions to dedicated users, and must ensure that only such authorized users are able to access management functionality.	O.MANAGE Supports this policy by ensuring that the functions and facilities supporting the authorized administrator role are in place.
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to user data and to the TSF.	O.TOE_ACCESS Supports this policy by providing a mechanism for controlling access to authorized users.
	OE.ADMIN Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of information it contains.	OE.ADMIN Supports this policy by ensuring that the authorized administrator role is understood and used by competent administrators.

Threat/Policy	TOE Security Objectives Addressing the Threat/Policy	Rationale
P.ROLES Administrative authority to TSF functionality shall be given to trusted personnel and be as restricted as possible supporting only the administrative duties the person has. This role shall be separate and distinct from other authorized users.	O.ADMIN_ROLE The TOE will provide a mechanism (e.g. a "role") by which the actions using administrative privileges may be restricted.	O.ADMIN_ROLE 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 is required.
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to user data and to the TSF.	O.TOE_ACCESS Supports this policy by ensuring that an authorized administrator role can be distinguished from other authorized users.
T.ACCESS_TSFDATA A threat agent may read or modify TSF data using functions of the TOE without the proper authorization	O.I&A The TOE ensures that users are authenticated before the TOE processes any actions that require authentication.	O.I&A Supports this policy by requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action the TOE is defined to provide to authenticated users only.
	O.MANAGE The TSF must provide all the functions and facilities necessary to support the authorized users that are responsible for the management of TOE security mechanisms, must allow restricting such management actions to dedicated users, and must ensure that only such authorized users are able to access management functionality.	O.MANAGE Diminishes this threat since it ensures that functions and facilities used to modify TSF data are not available to unauthorized users.
	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource within its Scope of Control is not inappropriately disclosed when the resource is reallocated.	O.RESIDUAL_INFORMATION Diminishes this threat since information contained in protected resources will not be easily available to the threat agent through reallocation attacks.
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to user data and to the TSF.	O.TOE_ACCESS Diminishes this threat since it makes it more unlikely that a threat agent has access to the TOE.

Threat/Policy	TOE Security Objectives Addressing the Threat/Policy	Rationale
T.ACCESS_TSFFUNC A threat agent may use or manage TSF, bypassing the protection mechanisms of the TSF.	O.ADMIN_ROLE The TOE will provide a mechanism (e.g. a "role") by which the actions using administrative privileges may be restricted.	O.ADMIN_ROLE Diminishes this threat by providing isolation of privileged actions.
	O.I&A The TOE ensures that users are authenticated before the TOE processes any actions that require authentication.	O.I&A Diminishes this threat since the TOE requires successful authentication to the TOE prior to gaining access to any controlled-access content. By implementing strong authentication to gain access to these services, an attacker's opportunity to masquerade as another entity in order to gain unauthorized access to data or TOE resources is reduced.
	O.MANAGE The TSF must provide all the functions and facilities necessary to support the authorized users that are responsible for the management of TOE security mechanisms, must allow restricting such management actions to dedicated users, and must ensure that only such authorized users are able to access management functionality.	O.MANAGE Diminishes this threat because an access control policy is specified to control access to TSF data. This objective is used to dictate who is able to view and modify TSF data, as well as the behavior of TSF functions.
	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource within its Scope of Control is not inappropriately disclosed when the resource is reallocated.	O.RESIDUAL_INFORMATION Diminishes this threat by ensuring that TSF data and user data is not persistent when resources are released by one user/process and allocated to another user/process.
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to user data and to the TSF.	O.TOE_ACCESS Diminishes this threat since it makes it more unlikely that a threat agent has access to the TOE.

Threat/Policy	TOE Security Objectives Addressing the Threat/Policy	Rationale
T.IA_MASQUERADE A user or process acting on behalf of a user may masquerade as an authorized entity in order to gain unauthorized access to user data, TSF data, or TOE resources.	O.I&A The TOE ensures that users are authenticated before the TOE processes any actions that require authentication.	O.I&A Diminishes this threat by requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action the TOE has defined to provide to authenticated users only.
	O.MEDIATE The TOE must protect user data in accordance with its security policy, and must mediate all requests to access such data.	O.MEDIATE Diminishes this threat by ensuring that all access to user data are subject to mediation, unless said data has been specifically identified as public data. The TOE requires successful authentication to the TOE prior to gaining access to any controlled-access content. By implementing strong authentication to gain access to these services, an attacker's opportunity to masquerade as another entity in order to gain unauthorized access to data or TOE resources is reduced.
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to user data and to the TSF.	O.TOE_ACCESS Diminishes this threat by controlling the logical access to the TOE and its resources. By constraining how and when authorized users can access the TOE, and by mandating the type and strength of the authentication mechanism this objective helps mitigate the possibility of a user attempting to login and masquerade as an authorized user. In addition, this objective provides the administrator the means to control the number of failed login attempts a user can generate before an account is locked out, further reducing the possibility of a user gaining unauthorized access to the TOE.

Threat/Policy	TOE Security Objectives Addressing the Threat/Policy	Rationale
T.IA_USER A threat agent may gain access to user data, TSF data or TOE resources with the exception of public objects without being identified and authenticated.	O.DISCRETIONARY_ACCESS The TSF must control access of subjects and/or users to named resources based on identity of the object, subject, or user. The TSF must allow authorized users to specify for each access mode which users/subjects are allowed to access a specific named object in that access mode.	O.DISCRETIONARY_ACCESS Diminishes this threat by requiring that data including user data stored with the TOE, have discretionary access control protection.
	O.I&A The TOE ensures that users are authenticated before the TOE processes any actions that require authentication.	O.I&A Diminishes this threat by requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action the TOE is defined to provide to authenticated users only.
	O.MEDIATE The TOE must protect user data in accordance with its security policy, and must mediate all requests to access such data.	O.MEDIATE Diminishes this threat by ensuring that all access to user data are subject to mediation, unless said data has been specifically identified as public data. The TOE requires successful authentication to the TOE prior to gaining access to any controlled-access content. By implementing strong authentication to gain access to these services, an attacker's opportunity to masquerade as another entity in order to gain unauthorized access to data or TOE resources is reduced.
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to user data and to the TSF.	O.TOE_ACCESS Diminishes this threat by controlling logical access to user data, TSF data or TOE resources.
T.RESIDUAL_DATA A user or a process acting on behalf of a user may gain unauthorized access to user or TSF data through reallocation of TOE resources from one user or process to another.	O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource within its Scope of Control is not inappropriately disclosed when the resource is reallocated.	O.RESIDUAL_INFORMATION Diminishes this threat because even if the security mechanisms do not allow a user to view TSF data, if TSF data were to reside inappropriately in a resource that was made available to a user, that user would be able to view the TSF data without authorization.

Threat/Policy	TOE Security Objectives Addressing the Threat/Policy	Rationale
T.TSF_COMPROMISE A user or a process acting on behalf of a user may cause configuration data to be inappropriately accessed (viewed, modified or deleted), or may compromise executable code within the TSF.	O.AUDIT_GENERATION The TOE will provide the capability to detect and create records of security relevant events associated with users.	O.AUDIT_GENERATION Diminishes this threat by providing the authorized administrator with the appropriate audit records supporting the detection of compromise of the TSF.
	O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to user data and to the TSF.	O.TOEACCESS Diminishes this threat since controlled user's logical access to the TOE will reduce the opportunities for an attacker's access to configuration data.

Threat/Policy	TOE Security Objectives Addressing the Threat/Policy	Rationale
T.UNAUTHORIZED_ACCESS A user may gain unauthorized access to user data for which they are not authorized according to the TOE security policy.	O.DISCRETIONARY_ACCESS The TSF must control access of subjects and/or users to named resources based on identity of the object, subject or user. The TSF must allow authorized users to specify for each access mode which users/subjects are allowed to access a specific named object in that access mode.	O.DISCRETIONARY_ACCESS Diminishes this threat by requiring that data including TSF data stored with the TOE, have discretionary access control protection.
	O.MANAGE The TSF must provide all the functions and facilities necessary to support the authorized users that are responsible for the management of TOE security mechanisms, must allow restricting such management actions to dedicated users, and must ensure that only such authorized users are able to access management functionality.	O.MANAGE Diminishes this threat by ensuring that the functions and facilities supporting that authorized users can be held accountable for their actions by authorized administrators are in place.
	O.MEDIATE The TOE must protect user data in accordance with its security policy, and must mediate all requests to access such data.	O.MEDIATE Diminishes this threat because it ensures that all access to user data are subject to mediation, unless said data has been specifically identified as public data. The TOE requires successful authentication to the TOE prior to gaining access to any controlled-access content. By implementing strong authentication to gain access to these services, an attacker's opportunity to conduct a man-in-the-middle and/or password guessing attack successfully is greatly reduced. Lastly, the TSF will ensure that all configured enforcement functions (authentication, access control rules, etc.) must be invoked prior to allowing a user to gain access to TOE or TOE mediated services. The TOE restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc. to the administrator. This feature ensures that no other user can modify the information flow policy to bypass the intended TOE security policy.

4.4 Rationale for Environmental Security Objectives

The table below provides a rationale for the environmental security objectives. The Rationale for Environmental Security Objectives tables are directly copied from the DBMS PP.

Table 4-5: Rationale for Environmental Security Objectives

Assumption	Environmental Objective Addressing the Assumption	Rationale for Specifying the Environmental Security Objective
A.AUTHUSER Authorized users possess the necessary authorization to access at least some of the information managed by the TOE.	OE.INFO_PROTECT Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: • All network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.	OE.INFO_PROTECT Supports the assumption by ensuring that users are authorized to access parts of the data managed by the TOE and is trained to exercise control over their own data. Having trained, authorized users, who are provided with relevant procedures for information protection supports the assumption of co-operation.
	OE.IT_REMOTE If the TOE relies on remote trusted IT systems to support the enforcement of its policy, those systems provide that the functions and any data used by the TOE in making policy decisions, required by the TOE are sufficiently protected from any attack that may cause those functions to provide false results.	OE.IT_REMOTE Supports this assumption by ensuring that remote systems that form part of the IT environment are protected. This gives confidence that the environment is benign.
	OE.IT_TRUSTED_SYSTEM The remote trusted IT systems implement the protocols and mechanisms required by the TSF to support the enforcement of the security policy. These remote trusted IT systems are managed according to known, accepted, and trusted policies based on the same rules and policies applicable to the TOE, and are physically and logically protected equivalent to the TOE.	OE.IT_TRUSTED_SYSTEM Supports this assumption by providing confidence that systems in the TOE IT environment contribute to a benign environment.

Assumption	Environmental Objective Addressing the Assumption	Rationale for Specifying the Environmental Security Objective
A.CONNECT All connections to and from remote trusted IT systems and between separate parts of the TSF not protected by the TSF itself are physically or logically protected within the TOE environment to ensure the integrity and confidentiality of the data transmitted and to ensure the authenticity of the communication end points.	OE.IT_REMOTE If the TOE relies on remote trusted IT systems to support the enforcement of its policy, those systems provide that the functions and any data used by the TOE in making policy decisions, required by the TOE are sufficiently protected from any attack that may cause those functions to provide false results.	OE.IT_REMOTE Supports the assumption by levying a requirement in the environment that connections between trusted systems or physically separated parts of the TOE are sufficiently protected from any attack that may cause those functions to provide false results.
	OE.INFO_PROTECT Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: • All network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authentication databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.	OE.INFO_PROTECT Supports the assumption by requiring that all network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques
	OE.IT_TRUSTED_SYSTEM The remote trusted IT systems implement the protocols and mechanisms required by the TSF to support the enforcement of the security policy. These remote trusted IT systems are managed according to known, accepted and trusted policies based on the same rules and policies applicable to the TOE, and are physically and logically protected equivalent to the TOE.	OE.IT_TRUSTED_SYSTEM Supports the assumption by ensuring that remote trusted IT systems implement the protocols and mechanisms required by the TSF to support the enforcement of the security policy
	OE.PHYSICAL Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.	OE.PHYSICAL Supports the assumption by ensuring that appropriate physical security is provided within the domain.

Assumption	Environmental Objective Addressing the Assumption	Rationale for Specifying the Environmental Security Objective
A.SUPPORT Any information provided by a trusted entity in the IT environment and used to support the provision of time and date, information used in audit capture, user authentication, and authorization that is used by the TOE is correct and up to date.	OE.IT_I&A Any information provided by a trusted entity in the environment and used to support user authentication and authorization used by the TOE is correct and up to date.	OE.IT_I&A Supports the assumption implicitly.
A.MANAGE The TOE security functionality is managed by one or more competent administrators. The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by instructions provided by the guidance documentation.	OE.ADMIN Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of information it contains.	OE.ADMIN Supports the assumption since the authorized administrators are assumed competent in order to help ensure that all the tasks and responsibilities are performed effectively.
	OE.INFO_PROTECT Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: • All network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authentication databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.	OE.INFO_PROTECT Supports the assumption by ensuring that the information protection aspects of the TOE and the system(s) and relevant connectivity that form the platform for the TOE is vital to addressing the security problem, described in this PP. Managing these effectively using defined procedures is reliant on having competent administrators.
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 DBMS servers, other than those services necessary for the operation, administration, and support of the DBMS.	OE.NO_GENERAL_PURPOSE The DBMS server must not include any general-purpose computing or storage capabilities. This will protect the TSF data from malicious processes. The environmental objective is tightly related to the assumption, which when fulfilled will address the assumption.

Assumption	Environmental Objective Addressing the Assumption	Rationale for Specifying the Environmental Security Objective
A.PEER_FUNC_&_MGT All remote trusted IT systems trusted by the TSF to provide TSF data or services to the TOE, or to support the TSF in the enforcement of security policy decisions are assumed to correctly implement the functionality used by the TSF consistent with the assumptions defined for this functionality and to be properly managed and operate under security policy constraints compatible with those of the TOE.	OE.IT_REMOTE If the TOE relies on remote trusted IT systems to support the enforcement of its policy, those systems provide that the functions and any data used by the TOE in making policy decisions, required by the TOE are sufficiently protected from any attack that may cause those functions to provide false results.	OE.IT_REMOTE The assumption that connections between trusted systems or physically separated parts of the TOE is addressed by the objective specifying that such systems are sufficiently protected from any attack that may cause those functions to provide false results.
	OE.IT_TRUSTED_SYSTEM The remote trusted IT systems implement the protocols and mechanisms required by the TSF to support the enforcement of the security policy. These remote trusted IT systems are managed according to known, accepted, and trusted policies based on the same rules and policies applicable to the TOE, and are physically and logically protected equivalent to the TOE.	OE.IT_TRUSTED_SYSTEM The assumption on all remote trusted IT systems to implement correctly the functionality used by the TSF consistent with the assumptions defined for this functionality is supported by physical and logical protections and the application of trusted policies commensurate with those applied to the TOE.

Assumption	Environmental Objective Addressing the Assumption	Rationale for Specifying the Environmental Security Objective
A.PHYSICAL It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.	OE.PHYSICAL Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.	OE.PHYSICAL The TOE, the TSF data, and protected user data is assumed to be protected from physical attack (e.g., theft, modification, destruction, or eavesdropping). Physical attack could include unauthorized intruders into the TOE environment, but it does not include physical destructive actions that might be taken by an individual that is authorized to access the TOE environment.
	OE.INFO_PROTECT Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: • All network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authentication databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.	OE.INFO_PROTECT Supports the assumption by requiring that all network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques.
A.TRAINEDUSER Users are sufficiently trained and trusted to accomplish some task or group of tasks within a secure IT environment by exercising complete control over their user data. .	OE.INFO_PROTECT Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: • All network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authentication databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.	OE.INFO_PROTECT Supports the assumption by ensuring that users are authorized to access parts of the data managed by the TOE and is trained to exercise control over their own data.

Assumption	Environmental Objective Addressing the Assumption	Rationale for Specifying the Environmental Security Objective
P.ACCOUNTABILITY The authorized users of the TOE shall be held accountable for their actions within the TOE.	OE.ADMIN Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of information it contains.	OE.ADMIN Supports the policy that the authorized administrators are assumed competent in order to help ensure that all the tasks and responsibilities are performed effectively.
	OE.INFO_PROTECT Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: • All network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.	OE.INFO_PROTECT Supports the policy by ensuring that the authorized users are trained and have procedures available to support them and that the DAC protections function and are able to provide sufficient information to inform those pursuing accountability.
P.ROLES The TOE shall provide an authorized administrator role for secure administration of the TOE. This role shall be separate and distinct from other authorized users.	OE.ADMIN Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of information it contains.	OE.ADMIN Supports the policy by ensuring that an authorized administrator role for secure administration of the TOE is established.

Assumption	Environmental Objective Addressing the Assumption	Rationale for Specifying the Environmental Security Objective
P.USER Authority shall only be given to users who are trusted to perform the actions correctly.	OE.ADMIN Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of information it contains.	OE.ADMIN Supports the policy by ensuring that the authorized administrators, responsible for giving appropriate authorities to users, are trustworthy.
	OE.INFO_PROTECT Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: • All network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.	OE.INFO_PROTECT Supports the policy by ensuring that users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data and that DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly.
T.IA_MASQUERADE A user or a process acting on behalf of a user may masquerade as an authorized entity in order to gain unauthorized access to user data, TSF data, or TOE resources.	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.NO_GENERAL_PURPOSE The DBMS server must not include any general-purpose computing or storage capabilities. This diminishes the threat of masquerade since only users with DBMS or related functions will be defined in the TOE environment.

Assumption	Environmental Objective Addressing the Assumption	Rationale for Specifying the Environmental Security Objective
T.TSF_COMPROMISE A user or a process acting on behalf of a user may cause configuration data to be inappropriately accessed (viewed, modified or deleted), or may compromise executable code within the TSF.	OE.INFO_PROTECT Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: • All network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.	OE.INFO_PROTECT Diminishes the threat by ensuring that all network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques.
	OE.IT_REMOTE If the TOE relies on remote trusted IT systems to support the enforcement of its policy, those systems provide that the functions and any data used by the TOE in making policy decisions, required by the TOE are sufficiently protected from any attack that may cause those functions to provide false results.	OE.IT_REMOTE Diminishes the threat by ensuring that remote trusted IT systems are sufficiently protected.
	OE.IT_TRUSTED_SYSTEM The remote trusted IT systems implement the protocols and mechanisms required by the TSF to support the enforcement of the security policy. These remote trusted IT systems are managed according to known, accepted and trusted policies based on the same rules and policies applicable to the TOE, and are physically and logically protected equivalent to the TOE.	OE.IT_TRUSTED_SYSTEM Diminishes the threat by ensuring that remote trusted IT systems are managed according to known, accepted and trusted policies based on the same rules and policies applicable to the TOE, and are physically and logically protected equivalent to the TOE.
	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.NO_GENERAL_PURPOSE Diminishes this threat by reducing the opportunities to subvert non TOE related capabilities in the TOE environment.
	OE.PHYSICAL Those responsible for the TOE must ensure that those parts of the TOE critical to enforcement of the security policy are protected from physical attack that might compromise IT security objectives. The protection must be commensurate with the value of the IT assets protected by the TOE.	OE.PHYSICAL Diminishes the threat of a TSF compromise due to exploitation of physical weaknesses or vulnerabilities as a vector in an attack.

Assumption	Environmental Objective Addressing the Assumption	Rationale for Specifying the Environmental Security Objective
T.UNAUTHORIZED_ACCESS A user may gain unauthorized access to user data for which they are not authorized according to the TOE security policy.	OE.INFO_PROTECT Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: • All network and peripheral cabling must be approved for the transmittal of the most sensitive data transmitted over the link. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted using appropriate physical and logical protection techniques. • DAC protections on security-relevant files (such as audit trails and authorization databases) shall always be set up correctly. • Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.	OE.INFO_PROTECT Diminishes the threat by ensuring that the logical and physical threats to network and peripheral cabling are appropriately protected. DAC protections if implemented correctly may support the identification of unauthorized accesses.

5. EXTENDED SECURITY FUNCTIONAL REQUIREMENTS

5.1. Extended Security Functional Requirements for the TOE

FIA_USB_(EXT).2 Enhanced user-subject binding

FIA_USB_(EXT).2 is analogous to FIA_USB.1 except that it adds the possibility to specify rules whereby subject security attributes are also derived from TSF data other than user security attributes.

Component leveling

FIA_USB_(EXT).2 is hierarchical to FIA_USB.1.

Management

See management description specified for FIA_USB.1 in [CC].

Audit

See audit requirement specified for FIA_USB.1 in [CC].

FIA_USB_(EXT).2 Enhanced user-subject binding

Hierarchical to: FIA_USB.1 User-subject binding

Dependencies: FIA_ATD.1 User attribute definition

FIA_USB_(EXT).2 .1

The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [assignment: list of security attributes].

FIA_USB_(EXT).2 .2

The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment: rules for the initial association of attributes].

FIA_USB_(EXT).2 .3

The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [assignment: rules for the changing of attributes].

FIA_USB_(EXT).2 .4

The TSF shall enforce the following rules for the assignment of subject security attributes not derived from user security attributes when a subject is created: [assignment: rules for the initial association of the subject security attributes not derived from user security attributes].

5.2. Rationale for Extended Security Functional Requirements

The table below presents a rationale for the inclusion of the extended functional security requirements found in this PP. Note that there are no extended security assurance requirements (SAR).

Table 5-1: Rationale for Extended Security Functional Requirements

Extended Requirement	Identifier	Rationale
FIA_USB_(EXT).2	Enhanced user-subject binding	A DBMS may derive subject security attributes from other TSF data that are not directly user security attributes. An example is the point-of entry the user has used to establish the connection. An access control policy may also use this subject security attribute within its access control policy, allowing access to critical objects only when the user has connected through specific ports-of-entry.

6. SECURITY REQUIREMENTS

This section defines the functional requirements for the TOE.

Functional requirements in this ST were drawn directly from Part 2 of the CC, or were based on Part 2 of the CC, including the use of extended components. These requirements are relevant to supporting the secure operation of the TOE.

Except for replacing United Kingdom spelling with American spelling, the notation, formatting, and conventions used in this ST are consistent with version 3.1 of the CC. Selected presentation choices are discussed here to aid the ST reader.

The CC allows several operations to be performed on functional requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in clause 8 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** or in the case of deletions, by ~~**crossed out 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 by the PP authors are denoted by *italicized text.*

Selections that have been made by the ST author appear as italicized bold text in square brackets, [***example selection***].

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 by the PP authors are denoted by showing the value in square brackets, [assignment_value].

Assignments that have been made by the ST author appear as bold text in square brackets, [**example assignment**].

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).

6.1. Security Functional Requirements for the TOE

The functional security requirements for the TOE are listed in **Table 6-1** below.

They are taken from either the DBMS PP or are extended components defined in Section 5.

Table 6-1: Functional Components

#	SFR Short Name	SFR Description	From DBMS PP	Extended	Refined
1	FAU_GEN.1	Audit data generation	Yes	No	No
2	FAU_GEN.2	User identity association	Yes	No	No
3	FAU_SEL.1	Selective audit	Yes	No	Yes
4	FDP_ACC.1	Subset access control	Yes	No	No
5	FDP_ACF.1	Security attribute based access control	Yes	No	No
6	FDP_RIP.1	Subset residual information protection	Yes	No	No
7	FIA_ATD.1	User attribute definition	Yes	No	No
8	FIA_UAU.1	Timing of authentication	Yes	No	No
9	FIA_UID.1	Timing of identification	Yes	No	No
10	FIA_USB_(EXT).2	Enhanced user-subject binding	Yes	Yes	No
11	FMT_MOF.1	Management of security functions behavior	Yes	No	No
12	FMT_MSA.1	Management of security attributes	Yes	No	No
13	FMT_MSA.3	Static attribute initialization	Yes	No	No
14	FMT_MTD.1	Management of TSF data	Yes	No	No
15	FMT_REV.1(1)	Revocation	Yes	No	No
16	FMT_REV.1(2)	Revocation	Yes	No	No
17	FMT_SMF.1	Specification of management functions	Yes	No	No
18	FMT_SMR.1	Security roles	Yes	No	No
19	FPT_TRC.1	Internal TSF consistency	Yes	No	No
20	FTA_MCS.1	Basic limitation on multiple concurrent sessions	Yes	No	No
21	FTA_TSE.1	TOE session establishment	Yes	No	No

6.1.1 Security Audit (FAU)

6.1.1.1 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 6-2: Auditable Events**; and

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

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

e) [***no additional events***]].

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, [information specified in column three of **Table 6-2**: Auditable Events, below].

Table 6-2: Auditable Events from DBMS PP

Security Functional Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating	The identity of the authorized administrator that made the change to the audit configuration
FDP_ACC.1	None	None
FDP_ACF.1	Successful requests to perform an operation on an object covered by the SFP	The identity of the subject performing the operation
FDP_RIP.1	None	None
FIA_ATD.1	None	None
FIA_UAU.1	Unsuccessful use of the authentication mechanism	None
FIA_UID.1	Unsuccessful use of the user identification mechanism, including the user identity provided	None
FIA_USB_(EXT).2	Unsuccessful binding of user security attributes to a subject (e.g., creation of a subject)	None
FMT_MOF.1	None	None
FMT_MSA.1	None	None
FMT_MSA.3	None	None
FMT_MTD.1	None	None
FMT_REV.1(1)	Unsuccessful revocation of security attributes	Identity of individual attempting to revoke security attributes
FMT_REV.1(2)	Unsuccessful revocation of security attributes	Identify of individual attempting to revoke security attributes
FMT_SMF.1	Use of the management functions	Identity of the administrator performing these functions
FMT_SMR.1	Modifications to the group of users that are part of a role	Identity of authorized administrator modifying the role definition
FPT_TRC.1	Restoring consistency	None
FTA_MCS.1	Rejection of a new session based on the limitation of multiple concurrent sessions	None
FTA_TSE.1	Denial of a session establishment due to the session establishment mechanism	Identity of the individual attempting to establish a session

6.1.1.2 User identity association (FAU_GEN.2)

FAU_GEN.2.1

For audit events resulting from actions of identified users **and any identified groups**, the TSF shall be able to associate each auditable event with the identity of the [***user and group***] that caused the event.

Application Note:

1) *A user in PostgreSQL is a role with the LOGIN privilege.*

2) *As a user in PostgreSQL is a role with designated privileges, in many cases the user is synonymous with the group. In cases where the role is modified from the user role to execute a statement, this "effective role" will be logged.*

6.1.1.3 Selective audit (FAU_SEL.1)

FAU_SEL.1.1

The TSF shall be able to select the set of events to be audited from the set of all auditable events based on the following attributes:

a) *object identity;*

b) *user identity;*

c) [***group identity***];

d) *event type;*

e) [success of auditable security events;

f) failure of auditable security events;

g) **[severity level; and**

h) **Auditable Events listed in the third column of Table 7-3**].]

6.1.2 User data protection (FDP)

6.1.2.1 Subset access control (FDP_ACC.1)

FDP_ACC.1.1

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

6.1.2.2 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:

a) [**Authorized user identity and/or group membership associated with a subject**;

b) Access operations implemented for DBMS-controlled objects; and

c) Object identity].

FDP_ACF.1.2

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

[The Discretionary Access Control policy mechanism shall, either by explicit authorized user/group 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 permitted to that authorized user, permit access;

c) If the requested mode of access is denied to every group of which the authorized user is a member, deny 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 objects based on the following additional rules: [**subject has authorized administrator role**].

Application Note:

1) The "authorized administrator" is referred to as the "Superuser" in PostgreSQL.

FDP_ACF.1.4

The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [**no additional explicit denial rules**].

6.1.2.3 Subset residual information protection (FDP_RIP.1)

FDP_RIP.1.1

The TSF shall ensure that any previous information content of a resource is made unavailable upon *allocation of the resource* to the following objects: [**database objects listed under FDP_ACC.1**].

6.1.3 Identification and authentication (FIA)

6.1.3.1 User attribute definition (FIA_ATD.1)

FIA_ATD.1.1

The TSF shall maintain the following list of security attributes belonging to individual users:

a) [Database user identifier and any associated group memberships;

b) Security-relevant database roles; and

c) [**Role security attributes listed in the first column of Table 6-3**].

Table 6-3: PostgreSQL Role Security Attributes

Attribute	Default Value	Security Function	Notes
name	none	Identification & Authentication, Access Control	Required field Same as PostgreSQL user identity for roles that have LOGIN attribute value
Superuser	NOSUPERUSER	Access Control	A role with the SUPERUSER attribute can override all access restrictions. Only a Superuser can create a new Superuser. Superuser roles must also have the login permission to be useful, as they still go through user authentication.
createdb	NOCREATEDB	Access Control	A role with the CREATEDB attribute can create new databases
createrole	NOCREATEROLE	Access Control	A role with the CREATEROLE attribute can create, alter and drop other roles. The newly created roles are not restricted to the permissions of the creating user.
inherit	INHERIT	Access Control	A role with the inherit attribute can automatically use whatever database privileges have been granted to all roles it is directly or indirectly a member of. Without INHERIT, membership in another role only grants the ability to SET ROLE to that other role; the privileges of the other role are only available after having done so.
bypassrls	nobypassrls	Access Control	A role with the bypassrls attribute has the ability to bypass the row security system. This attribute can only be set by a Superuser.
login	NOLOGIN	Identification & Authentication	A role with the login attribute is allowed to log in; that is, the role can be given as the initial session authorization name during client connection. A role having the LOGIN attribute value is a user. Roles without this attribute are useful for managing database privileges, including emulating user groups, but they are not considered database users.

Attribute	Default Value	Security Function	Notes
connection limit	-1	Identification & Authentication	Specifies how many concurrent connections the role can make. -1 means no limit. Superuser is not limited by this number.
password	none	Identification & Authentication	A password is only of use for roles having the LOGIN attribute value. This attribute is only used when connections are made using a password authentication scheme.
encrypted	default behavior is determined by the configuration parameter password_encryption	Identification & Authentication	The password is always stored encrypted in the system catalogs. The ENCRYPTED keyword has no effect, but is accepted for backwards compatibility. The method of encryption is determined by the configuration parameter password encryption. If the presented password string is already in MD5-encrypted or SCRAM-encrypted format, then it is stored as-is regardless of password_encryption (since the system cannot decrypt the specified encrypted password string, to encrypt it in a different format). This allows reloading
valid until <date>	none	Identification & Authentication	Date and time after which the role's password is no longer valid. If it is omitted the password will be valid for all time. Validation dates are only applied when using password authentication.
in role	none	Access Control	Specifies one or more existing roles (groups) of which the new role is a member.
replication	NOREPLICATION	Identification & Authentication	Controls whether this role can be used to make a replication connection to the database. Replication roles can read all the raw data in the database similarly to a Superuser. This role has minimal write access to the database, limited to the replication metadata and state information.

6.1.3.2 Timing of authentication (FIA_UAU.1)

FIA_UAU.1.1

The TSF shall allow [**no actions**] on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2

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

6.1.3.3 Timing of identification (FIA_UID.1)

FIA_UID.1.1

The TSF shall allow [**no actions**] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2

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

6.1.3.4 Enhanced user-subject binding (FIA_USB_(EXT).2)

FIA_USB_(EXT).2.1

The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [**Role security attributes listed in column 1 of Table 6-3**].

FIA_USB_(EXT).2.2

The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [**the default values listed in column 2 of Table 6-3 will initially be associated with subjects acting on the behalf of users for each of the corresponding role security attributed listed in column 1 of Table 6-3**].

FIA_USB_(EXT).2.3

The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [**user security attributes associated with subjects acting on behalf of users are changed by means of an ALTER ROLE statement issued by a user that has sufficient privileges to assign the role according to the following rules:**

a) subjects with the authorized administrator role can change any user security attribute settings for any role;

b) subjects with the CREATEROLE privilege can change any user security attribute settings except for Superuser and replication roles.

c) subjects without the authorized administrator role or the CREATEROLE privilege cannot change any user security attribute].

Application Note:

1) *A user in PostgreSQL is a role with the LOGIN privilege.*

2) *The "authorized administrator" is called the "Superuser" in PostgreSQL.*

FIA_USB_(EXT).2.4

The TSF shall enforce the following rules for the assignment of subject security attributes not derived from user security attributes when a subject is created: [**no subject security attributes are assigned other than those derived from user security attributes when a subject is created**].

6.1.4 Security management (FMT)

6.1.4.1 Management of security functions behavior (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].

6.1.4.2 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* [all] the security attributes to [authorized administrators].

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

6.1.4.4 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].

6.1.4.5 Revocation (FMT_REV.1(1))

FMT_REV.1.1(1)

The TSF shall restrict the ability to revoke [**Role security attributes listed in column 1 of Table 6-3**] associated with the *users* under the control of the TSF to [the authorized administrator].

FMT_REV.1.2(1)

The TSF shall enforce the rules [

- **a) User privileges are revoked using the ALTER ROLE command.**
- **b) Users can only revoke the direct privileges that they have granted to another user.**
- **c) Users can only revoke the indirect privileges that they have granted to another user. The revocation takes effect at the next user login**].

6.1.4.6 Revocation (FMT_REV.1(2))

FMT_REV.1.1(2)

The TSF shall restrict the ability to revoke [**object privileges listed in column 2 of Table 7-6**] associated with the *objects* under the control of the TSF to the [authorized administrator] **and database users with sufficient privileges as allowed by the Discretionary Access Control policy**.

FMT_REV.1.2(2)

The TSF shall enforce the rules [

a) **Object privileges are revoked using the REVOKE command.**
b) **Both object privileges and object dependent privileges are revoked using the CASCADE option for the REVOKE command.**
c) **Users can only revoke the direct privileges that they have granted on an object.**
d) **Users can only revoke the indirect privileges that they have granted on an object.**
e) **The revocation takes effect the next time that the object is opened**].

6.1.4.7 Specification of management functions (FMT_SMF.1)

FMT_SMF.1.1

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

a) **Security management functions covered by FMT_MOF.1,**
b) **Management of security attributes covered by FMT_MSA.1,**
c) **Management of TSF data listed in covered by FMT_MTD.1, and**
d) **Revocation of user and object security attributes covered by FMT_REV.1(*)]**

Application notes:

1) *FMT_REV.1(*) references each of FMT_REV.1.1(1), FMT_REV.1.2(1), FMT_REV.1.1(2) and FMT_REV.1.2(2).*

6.1.4.8 Security roles (FMT_SMR.1)

FMT_SMR.1.1

The TSF shall maintain the roles [authorized administrator and **[customized roles created by the authorized administrator that have been assigned the role security attributes CREATEROLE, and CREATEDB as defined in FIA_ATD.1**]].

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

Application notes:

1) *The "authorized administrator" is called the "Superuser" in PostgreSQL.*

6.1.5 Protection of the TOE Security Functions (FPT)

6.1.5.1 Internal TSF consistency (FPT_TRC.1)

FPT_TRC.1.1

The TSF shall ensure that TSF data is consistent when replicated between parts of the TOE.

FPT_TRC.1.2

When parts of the TOE containing replicated TSF data are disconnected, the TSF shall ensure the consistency of the replicated TSF data upon reconnection before processing any requests for [**any function**].

6.1.6 TOE Access (FTA)

6.1.6.1 Basic limitation on multiple concurrent sessions (FTA_MCS.1)

FTA_MCS.1.1

The TSF shall restrict the maximum number of concurrent sessions that belong to the same user.

FTA_MCS.1.2

The TSF shall enforce, by default, a limit of [**-1**] sessions per user.

Application notes:

1) *In PostgreSQL the default value of -1 indicates that, by default, no limit is set. The maximum number of sessions per user can be set in PostgreSQL to a number dictated by policy.*

6.1.6.2 TOE session establishment (FTA_TSE.1)

FTA_TSE.1.1

The TSF shall be able to deny session establishment based on [**attributes that can be set explicitly by authorized administrator(s), including user identity, and [*group identity*, *time of day, day of the week,* [database name, Host IP address, and/or subnet address**]].

6.2 Security Assurance Requirements for the TOE

The Security Assurance Requirements for the TOE are the assurance components of Evaluation Assurance Level 2 (EAL2) augmented by ALC_FLR.2 taken from Part 3 of the Common Criteria. The Security Assurance Requirements were selected according to and in compliance with DBMS PP Section 3.2

"Conformance with Packages". None of the assurance components are refined. The assurance components are listed in Table 6-4.

Table 6-4: Assurance Components

Item	Component	Component Title
1	ADV_ARC.1	Security architecture description
2	ADV_FSP.2	Security-enforcing functional specification
3	ADV_TDS.1	Basic design
4	AGD_OPE.1	Operational user guidance
5	AGD_PRE.1	Preparative procedures
6	ALC_CMC.2	Use of a CM system
7	ALC_CMS.2	Parts of the TOE CM coverage
8	ALC_DEL.1	Delivery procedures
9	ALC_FLR.2	Flaw reporting procedures
10	ASE_CCL.1	Conformance claims
11	ASE_ECD.1	Extended components definition
12	ASE_INT.1	ST introduction
13	ASE_OBJ.2	Security objectives
14	ASE_REQ.2	Derived security requirements
15	ASE_SPD.1	Security problem definition
16	ASE_TSS.1	TOE summary specification
17	ATE_COV.1	Evidence of coverage
18	ATE_FUN.1	Functional testing
19	ATE_IND.2	Independent testing – sample
20	AVA_VAN.2	Vulnerability analysis

Further information on these assurance components can be found in the Common Criteria for Information Technology Security Evaluation (CCITSE) Part 3.

6.3 Rationale for TOE Security Functional Requirements

The following table provides the rationale for the selection of the security functional requirements. It traces

each TOE security objective to the identified security functional requirements

Table 6-5: Rationale for TOE Security Functional Requirements

Objective	Requirements Addressing the Objective	Rationale
O.ADMIN_ROLE The TOE will provide a mechanism (e.g. a "role") by which the actions using administrative privileges may be restricted.	FMT_SMR.1	The TOE will establish, at least, an authorized administrator role. The ST writer may choose to specify more roles. 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. (FMT_SMR.1)
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.2 FAU_SEL.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. This requirement also places a requirement on the level of detail that is recorded on any additional security functional requirements a ST author adds to this PP. FAU_GEN.2 ensures that the audit records associate a user and any associated group identity with the auditable event. In the case of authorized users, the association is accomplished with the user ID. In the case of authorized groups, the association is accomplished with the group ID. 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,
O.DISCRETIONARY_ACCESS The TSF must control access of subjects and/or users to named resources based on identity of the object, subject or user. The TSF must allow authorized users to specify for each access mode which users/subjects are allowed to access a specific named object in that access mode.	FDP_ACC.1 FDP_ACF.1	The TSF must control access to resources based on the identity of users that are allowed to specify which resources they want to access for storing their data. The access control policy must have a defined scope of control [FDP_ACC.1]. The rules for the access control policy are defined [FDP_ACF.1].

Objective	Requirements Addressing the Objective	Rationale
O.I&A The TOE ensures that users are authenticated before the TOE processes any actions that require authentication.	FIA_ATD.1 FIA_UAU.1 FIA_UID.1 FIA_USB_(EXT).2	The TSF must ensure that only authorized users gain access to the TOE and its resources. Users authorized to access the TOE must use an identification and authentication process [FIA_UID.1, FIA_UAU.1]. To ensure that the security attributes used to determine access are defined and available to the support authentication decisions. [FIA_ATD.1]. Proper authorization for subjects acting on behalf of users is also ensured [FIA_USB_(EXT).2]. The appropriate strength of the authentication mechanism is
O.MANAGE The TSF must provide all the functions and facilities necessary to support the authorized users that are responsible for the management of TOE security mechanisms, must allow restricting such management actions to dedicated users, and must ensure that only such authorized users are able to access management functionality.	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	FMT_MOF.1 requires that the ability to use particular TOE capabilities be restricted to the administrator. FMT_MSA.1 requires that the ability to perform operations on security attributes be restricted to particular roles. FMT_MSA.3 requires that default values used for security attributes are restrictive. FMT_MTD.1 requires that the ability to manipulate TOE content is restricted to administrators. FMT_REV.1 restricts the ability to revoke attributes to the administrator. FMT_SMF.1 identifies the management functions that are available to the authorized administrator.
O.MEDIATE The TOE must protect user data in accordance with its security policy, and must mediate all requests to access such data.	FDP_ACC.1 FDP_ACF.1 FPT_TRC.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 operations between subject and object covered are defined by the TOE's policy. FDP_ACF.1 defines the security attribute used to provide access control to objects based on the TOE's access control policy. FPT_TRC.1 ensures replicated TSF data that specifies attributes for access control must be consistent across distributed components of the TOE. The requirement is
O.RESIDUAL_INFORMATION The TOE will ensure that any information contained in a protected resource within its Scope of Control is not inappropriately disclosed when the resource is reallocated.	FDP_RIP.1	FDP_RIP.1 is used to ensure the contents of resources are not available to subjects other than those explicitly granted access to the data.

Objective	Requirements Addressing the Objective	Rationale
O.TOE_ACCESS The TOE will provide mechanisms that control a user's logical access to the TOE.	FDP_ACC.1 FDP_ACF.1 FIA_ATD.1 FTA_MCS.1 FTA_TSE.1	FDP_ACC.1 requires that each identified access control SFP be in place for a subset of the possible operations on a subset of the objects in the TOE. FDP_ACF.1 allows the TSF to enforce access based upon security attributes and named groups of attributes. Furthermore, the TSF may have the ability to explicitly authorize or deny access to an object based upon security attributes. FIA_ATD.1 defines the security attributes for individual users including the user's identifier and any associated group memberships. Security relevant roles and other identity security attributes. FTA_MCS.1 ensures that users may only have a maximum of a specified number of active sessions open at any given time. FTA_TSE.1 allows the TOE to restrict access to the TOE based on certain criteria.

7. TOE SUMMARY SPECIFICATION

7.1 IT Security Functions

This section describes the specific Security Functions of the TOE, including how the TOE meets each SFR listed in Section 6. Table 7-1 maps the functions to the SFRs for the TOE.

Table 7-1: Security Functions Mapped to Security Functional Requirements

Security Function	Sub-Function		SFR
Security Audit	AU-1	Audit Data Generation	FAU_GEN.1.1 FAU_GEN.1.2 FAU_GEN.2.1
	AU-2	Selective Audit	FAU_SEL.1
User Data Protection	DP-1	Subset Access Control	FDP_ACC.1
	DP-2	Discretionary Access Control	FDP_ACF.1.1 FDP_ACF.1.2 FDP_ACF.1.3 FDP_ACF.1.4
	DP-3	Residual Information Protection	FDP_RIP.1
Identification & Authentication	IA-1	User Attribute Definition	FIA_ATD.1.1
	IA-2	Identification and Authentication	FIA_UAU.1.1 FIA_UAU.1.2 FIA_UID.1.1 FIA_UID.1.2
	IA-3	Enhanced user-subject binding	FIA_USB_(EXT).2.1 FIA_USB_(EXT).2.2 FIA_USB_(EXT).2.3 FIA_USB_(EXT).2.4
Security Management	SM-1	Security Roles	FMT_SMR.1
	SM-2	Management of TSF Data and Functions	FMT_MOF.1 FMT_MSA.1 FMT_MTD.1.1 FMT_SMF.1.1
	SM-3	Static Attribute Initialization	FMT_MSA.3.1 FMT_MSA.3.2
	SM-4	Revocation	FMT_REV.1(1) FMT_REV.1.2(1) FMT_REV.1.1(2) FMT_REV.1.2(2)
Protection of the TSF Functions	PT-1	Internal TSF Consistency	FPT_TRC.1
TOE Access Functions	TA-1	Limits on multiple concurrent sessions	FTA_MCS.1.1 FTA_MCS.1.2
	TA-2	TOE Session Establishment	FTA_TSE.1.1

7.1.1 Security Audit Functions

7.1.1.1 Audit Data Generation (AU-1)

FAU_GEN.1.1, FAU_GEN.1.2, FAU_GEN.2.1

log_min_messages

PostgreSQL has the ability to generate an audit record of failed statements through the '`log_min_messages`' setting in `postgresql.conf`. Setting '`log_min_messages`' to ERROR will result in all failed statements being logged. The '`log_min_messages`' setting must be set to ERROR in order to be compliant with the DBMS PP. Any modification of `postgresql.conf` does not take effect until after the server is rebooted.

pg_audit

PostgreSQL Audit Extension allows classes of statements to be logged using the '`pg_audit.log`' setting by using the commands listed in Table 7-2.

Table 7-2: `pg_audit` Logged Statements

pg_audit Command	Logged Statements
READ	SELECT and COPY when the source is a relation or a query.
WRITE	INSERT, UPDATE, DELETE, TRUNCATE, and COPY when the destination is a relation.
FUNCTION	Function calls and DO blocks.
ROLE	Statements related to roles and privileges: GRANT, REVOKE, CREATE/ALTER/DROP ROLE.
DDL	All DDL that is not included in the ROLE class.
MISC	Miscellaneous commands, e.g. DISCARD, FETCH, CHECKPOINT, VACUUM, REINDEX.

Auditable Events

Table 7-3 indicates which Logging Facility (either `log_statement` or `pg_audit`) can be used to log the auditable events required by the DBMS PP by using the Configuration Settings set forth in the third column.

Table 7-3: Schedule of Auditable Events

Auditable Events	Logging Facility	Configuration Settings
Start-up and shutdown of the audit functions	pg_audit	pg_audit is active for the entire duration of the session.
All modifications to the audit configuration that occur while the audit collection functions are operating	pg_audit	pg_audit.log = 'MISC'
Successful requests to perform an operation on an object covered by the SFP	pg_audit	Varies based on the object to be audit logged.
Unsuccessful use of the authentication mechanism	PostgreSQL	log_connections = on
Unsuccessful use of the user identification mechanism, including the user identity provided	PostgreSQL	log_connections = on
Unsuccessful binding of user security attributes to a subject (e.g., creation of a subject)	pg_audit	pg_audit.log = 'ROLE'
Unsuccessful revocation of security attributes	pg_audit	pg_audit.log = 'ROLE'
Unsuccessful revocation of security attributes	pg_audit	pg_audit.log = 'ROLE'
Use of the management functions	pg_audit	pg_audit.log = 'ROLE,MISC'
Modifications to the group of users that are part of a role	pg_audit	pg_audit.log = 'ROLE'
Restoring consistency	PostgreSQL	Always logged no matter what settings are active.
Rejection of a new session based on the limitation of multiple concurrent sessions	PostgreSQL	log_connections = on
Denial of a session establishment due to the session establishment mechanism	PostgreSQL	log_connections = on
Start-up and shutdown of the DBMS	PostgreSQL	Always logged no matter what settings are active.
Use of special permissions (e.g., those often used by authorized administrators to circumvent access control policies)	pg_audit	pg_audit.log = 'ROLE'

Information Recorded in Audit Record

The TOE can be configured to generate an audit record containing:

(a) date and time of the event;

(b) type of event;

(c) subject identity (if applicable);

(d) the outcome (success or failure); and

(e) with respect to the Auditable Events identified in Table 6-2, the corresponding the information required by the Column 3 of Table 6-2.

The information in the audit record is selected by the Cluster Owner using the `log_line_prefix` configuration parameter in the `postgresql.conf` configuration file. The default is an empty string. The `log_line` prefix must be set with the timestamp (%t) and user name (%u) in order to be compliant with the DBMS PP

Table 7-4: PostgreSQL Log Record Prefix Configuration Option

Escape	Effect	Session only
%u	User Name	Yes
%d	Database Name	Yes
%r	Remote Hostname or IP address, and Remote Port	Yes
%h	Remote Host	Yes
%p	Process ID	No
%t	Timestamp without milliseconds	No
%m	Timestamp with milliseconds	No
%i	Command Tag. This is the command which generated the log line.	Yes
%c	Session ID. A unique identifier for each session. It is 2 4-byte hexadecimal numbers (without leading zeros) separated by a dot. The numbers are the Session Start Time and the Process ID, so this can also be used as a space saving way of printing these items.	Yes
%l	Number of the log line for each process, starting at 1	No
%s	Session Start Timestamp	Yes
%v	Virtual transaction ID	No
%x	Transaction ID (0 if none)	Yes
%q	Does not produce any output, but tells non-session processes to stop at this point in the string. Ignored by session processes.	No
%%	Literal %	No

7.1.1.2 Selective Audit (AU-2)

FAU_SEL.1

Selection of Audited Events

As described in more detail in Section 7.1.1.1, the TSF can select the set of events to be audited from the set of all auditable events based on:

(a) object identity by using `pg_audit`;

(b) user and group identity by using through the '`log_statement`' setting in `postgrespl.conf`; and

(c) the Auditable Events listed in the column 1 of Table 7-3 using the Configuration Settings set forth in column 3 of that table.

A user in PostgreSQL is a Role with the LOGIN privilege. As a user in PostgreSQL is a role with designated privileges, in many cases the user is synonymous with the group. In cases where the role is modified from the user role to execute a statement, this "effective role" will be logged.

Filtering on success or failure is done the same way as filtering on event types. To only include successful events, do not include error events in the filtering options. To only include failed events, include error events only.

The authorized administrator can select the set of events to be audited from the set of all auditable events by on by message severity level by using the `log_min_error_statement (enum)` as set forth in Table 7-5.

Table 7-5: Audit Log Message Severity Levels

Value	Description
DEBUG [1-5]	Provides information for use by developers.
INFO	Provides information implicitly requested by the user, e.g., during VACUUM VERBOSE.
NOTICE	Provides information that may be helpful to users, e.g., truncation of long identifiers and the creation of indexes as part of primary keys.
WARNING	Provides warnings to the user, e.g., COMMIT outside a transaction block.
ERROR	Reports an error that caused the current transaction to abort.
LOG	Reports information of interest to administrators, e.g., checkpoint activity.
FATAL	Reports an error that caused the current session to abort.
PANIC	Reports an error that caused all sessions to abort.

7.1.2 User Data Protection Functions

7.1.2.1 Subset Access Control (DP-1)

FDP_ACC.1

Subject security attributes are listed in Table 6-3. Object security attributes are discussed in the following subsections.

7.1.2.1.1 Objects, Their Operations, and Their Privileges

The objects, their operations, and their privileges are listed in Table 7-6 below.

Table 7-6: PostgreSQL Access Control Policy (Objects and Operations)

Object	Operations/ privileges	Description	Default Access on Objects	Privilege Delegation
Table	**Select --Copy To**	Allows (i) Accessors to retrieve rows from a table and (ii) permits the use of "Copy To" command on tables.	Owner only	WITH GRANT OPTION
	Insert --Copy From	Allows (i) Accessors to create new rows in a table and (ii) allows use of "Copy From" command on tables.		
	Update	Allows Accessors to update rows in a table. Most updates will also need SELECT access on the rows to be changed.		
	Delete	Allows Accessors to delete rows from a table. Most deletes will also need SELECT access on the rows to be deleted.		
	References	Allows Accessors to create foreign key constraint on table. Additional permissions are also needed to access the column on each side of the key relationship.		
	Trigger	Allows Accessors to create a trigger on a table.		
	Truncate	Allows Accessors to TRUNCATE the specified table.		
	All privileges	Allows Accessors to be granted all available privileges.		
Column	**Insert**	Allows Accessors to insert rows in a table. If the Accessor does not have update permission on the on the table, the Accessor must specify the specific columns in which values will be inserted and have been granted insert permission on those columns.	Owner only	WITH GRANT OPTION

	Update	Allows Accessors to update rows in a table. If the Accessor does not have update permission on the on the table, the Accessor must specify the specific columns in which values will be updated and have been granted update permission on those columns.		WITH GRANT OPTION
	Select	Allows Accessors to select rows in a table. If the Accessor does not have select permission on the on the table, the Accessor must specify the specific columns in which values will be selected and have been granted selected permission on those columns.	Owner only	WITH GRANT OPTION
	References	Allows Accessors to create foreign key constraint on a specific column. Additional permissions are also needed to access the column on each side of the key relationship.	Owner only	WITH GRANT OPTION
Row	**Insert**	Allows Role to insert a row in a table based on administrator-defined conditions.	Owner only	None
	Update	Allows Role to modify a row in a table based on administrator-defined conditions.		
	Select	Allows Role to select a row from a table based on administrator-defined conditions.		
	Delete	Allows Role to delete a row from a table based on administrator-defined conditions.		
Database	**Create**	Allows Accessors to create new schemas.	Owner only	WITH GRANT OPTION
	Connect	Allows Accessor to connect to the database.	PUBLIC	
	Temporary, Temp	Allow Accessors to created temporary tables while using the database.		
	All privileges	Allows Accessors to be granted all available privileges.		
Foreign Data Wrapper	**Create**	Allows Superuser to create a FDW.	Owner only (user who defines the FDW) .	None
	Create server	Allow a user of the FDW with USAGE privilege on FDW to define a foreign data server.		
	Create user mapping	Allow owner of a FDW or foreign server to define the connection information and user translation for its underlying connection or access. A user can create a User Mapping for its own user name if USAGE privilege on the server has been granted to the user		
Function	**Execute**	Allows Accessors the use of (i) the specified function and (ii) any operators that are implemented on top of the specified function.	EXECUTE (granted to PUBLIC) Executes with privileges of Definer	WITH GRANT OPTION
	All privileges	Allows Accessors to be granted all available privileges.		
Procedure	**Execute**	Allows Accessors the use of (i) the specified function and (ii) any operators that are implemented on top of the specified function.	EXECUTE (granted to PUBLIC) Executes with privileges of Definer	WITH GRANT OPTION
	All privileges	Allows Accessors to be granted all available privileges.		
Tablespace	**Create**	Allows (i) Accessors to create tables and indexes within the Tablespace, and (ii) databases to be created that have the Tablespace as their default Tablespace. See Table 7-9 for breakout of Tablespace privileges for object creation.	Owner only	WITH GRANT OPTION
	All privileges	Allows Accessors to be granted all available privileges.		
Schema	**Create**	Allows Accessor to create new objects within the schema. See **Table 7-7**, for a breakout of schema privileges required for creation/removal of various objects.	Owner only	WITH GRANT OPTION
	Usage	Allows Accessor to access objects contained in the specified schema (assuming that the objects' own privilege requirements are also met).		

	All privileges	Allows Accessors to be granted all available privileges		
Language	**Usage**	Allows Accessors to use of the specified language for the creation of functions in that language.	PUBLIC	WITH GRANT OPTION
	All privileges	Allows Accessors to be granted all available privileges.		
View	**Select**	Allows accessors to retrieve rows from a view	Owner only Executes with privileges of view definer.	WITH GRANT OPTION
	Rule	Allows Accessors to create a rule on the view. This is deprecated syntax, accepted for compatibility only.		
Index	**Create**	Allows table owner or Superuser only to create the index.	Owner Only	None
	Delete	Allows index owner or Superuser only to delete the index.		
Sequence	**Select**	Allows Accessors to retrieve rows from a sequence.	PUBLIC	None
	Usage	Allows Accessors to use the currval and nextval functions.		
	Update	Allows Accessors the use of the nextval and setval functions.		

Table 7-6 Notes:

1) *Superuser bypasses DAC checks and can perform all operations.*

2) *The owner of an object has all privileges by default. The right to drop an object or to alter its definition is not identified by a grantable privilege. It is inherent in the owner, and cannot be granted or revoked. The owner also implicitly has all grant options for the object.*

3) *WITH GRANT OPTION allows the role or user who was granted an object privilege to GRANT the privilege to other roles and users.*

4) *Rights can be granted for specific columns of a table. When a user has access to the whole table, the check for individual column permissions will not also apply.*

5) *Granting permissions on a table does not also give access to sequences used by the table. Permissions on sequences must be set separately.*

7.1.2.1.2 Schema, Database, and Tablespace Creation Privileges

PostgreSQL organizes database information in the following entities: schemas, databases, tables and tablespaces. Access rights and privileges are associated with each of these entities. They are described below.

Schema

Schema objects can be created and manipulated with SQL provided that the user doing so has the required privileges. Schema privileges for object creation or removal within the schema are specified in Table 7-7.

Table 7-7: Schema Privileges for Object Creation/Removal

Object	Operation	Schema Privilege Required	
		CREATE	USAGE
Table	CREATE	X	
	DROP		X
	ALTER TABLE ADD CONSTRAINT FOREIGN KEY		X
	ALTER TABLE DROP CONSTRAINT		X
Index	CREATE	X	X
	DROP		X
Sequence	CREATE	X	
	DROP		X
Trigger	CREATE	X	X
	DROP		X
View	CREATE	X	X
	DROP		X
Rule	CREATE		X
	DROP		X

Database

Database privileges for creation of the database or objects within the database are specified in Table 7-8.

Table 7-8: Database Privileges for Object Creation

Object	Operation	Required Privilege
Temp Table	CREATE	Must have TEMP privilege on the database (this is given to PUBLIC by default)
Database	CREATE DATABASE	CREATEDB
Schema objects (ie. tables, views, etc.)	CREATE	Must have CREATE privilege on the schema in which the object resides.
Public Database Link	CREATE	Must have CREATE PUBLIC DATABASE LINK privilege and CREATE on the database in which the link resides.

Tablespaces

Tablespace privileges for object creation within the Tablespace are specified in Table 7-9.

Table 7-9: Tablespace Privileges for Object Creation

Object	Operation	Required Privilege
Tablespace	CREATE	Database Superuser
Table	CREATE [in the specified tablespace]	CREATE on the target tablespace
Index	CREATE [in the specified tablespace]	CREATE on the target tablespace
Database	CREATE [in the specified tablespace]	CREATE on the target tablespace

7.1.2.2 Discretionary Access Control (DP-2)

FDP_ACF.1.1, FDP_ACF.1.2, FDP_ACF.1.3, FDP_ACF.1.4

PostgreSQL enforces the DAC policy to protect the user data stored in the PostgreSQL database using the security attributes of subjects and objects.

The following ordered rules determine if a user is permitted to perform a requested operation on an object. The process ceases for the requested operation on the object, when either "access is granted" or "access is denied".

The subject must be successfully authenticated prior to execution of the rules below.

1) If the subject has Superuser role attribute. If yes, "access is granted" to the requested object.

2) If the requested operation is object creation, then the following rules apply:

    a) If the subject is assigned the privilege corresponding to the requested operation of the target object as specified in Table 7-6 through Table 7-9, then "access is granted".

    b) If the conditions set forth in (a) above are not satisfied, then "access is denied".

3) If the requested operation is access to an existing object (i.e., an operation as specified in **Table 7-6**), then the following rules apply:

    a) If the subject has USAGE privilege on the TARGET schema, then "access is granted".

    b) If the conditions set forth in (a) above are not satisfied, then "access is denied".

4) If the requested operation is EXECUTE on an existing procedure, function or package, then the following rules apply:

    a) If (i) the SECURITY INVOKER/DEFINER attribute is set to "INVOKER and (ii) the user that called the function is assigned the privilege corresponding to the requested operation on the target object, then "access is granted".

    b) If (i) the SECURITY INVOKER/DEFINER attribute is set to "DEFINER" and (ii) the user that owns the function is assigned the privilege corresponding to the requested operation on the target object, then "access is granted".

    c) If neither of the conditions set forth in (a) or (b) above are satisfied, then "access is denied".

5) If the subject is explicitly assigned the privilege corresponding to the requested operation of the target object, then "access is granted".

6) If the subject is a member of any role (held in the "in roles" attribute) that has been granted the requested privilege on the target object as a result of either (i) the user role having been directly granted membership in the group role plus use of SET ROLE command or (ii) the subject has the INHERIT attribute privilege and is a member of a role that possesses the requested privilege, and

    a) If (i) the target object is a table on which row security has been enabled, (ii) the requested

operation is INSERT or UPDATE in violation of an applicable Row Security Policy, then "access is denied".

b) If the conditions set forth in (a) above are not satisfied, then "access is granted".

NOTE: A role with the INHERIT attribute can automatically use whatever database privileges have been granted to all roles it is directly or indirectly a member of. Without INHERIT, membership in another role only grants the ability to SET ROLE to that other role; the privileges of the other role are only available after having done so.

7) If the conditions set forth in (1) – (6) above does not result in "access is granted", then "access is denied".

7.1.2.3 Residual Information Protection (DP-3)

FDP_RIP.1

Residual information protection is enforced through the implementation of "write before read". Storage for a row is allocated at the time that is in inserted or updated and the new values are written into the allocated space. Data storage and retrieval relies upon indexes and links and there is no way for users to access unallocated disk space.

PostgreSQL does not immediately remove the old version of a row from a table. This approach is necessary to gain the benefits of concurrency control. A row version must not be deleted, while it still may be needed by another transaction. However, eventually, an outdated or deleted row version is no longer of interest to any transaction. The space it occupies must be reclaimed for reuse by new rows, to avoid infinite growth of disk space requirements. PostgreSQL monitors table activity and reclaims space as necessary.

7.1.3 Identification & Authentication Functions

7.1.3.1 User Attribute Definition (IA-1)

FIA_ATD.1.1

PostgreSQL manages database access permissions using the concept of roles. A role can be thought of as either a database user, or a group of database users, depending on how the role is configured. Therefore in PostgreSQL, the concept of roles subsumes the concepts of "users" and "groups".

For additional discussion of the management of user Role security attributes, see Section 7.1.3.2.2 Enhanced User-Subject Binding.

7.1.3.2 Identification and Authentication (IA-2)

FIA_UAU.1.1, FIA_UAU.1.2, FIA_UID.1.1, FIA_UID.1.2

The TOE does not allow access to the TOE until a user has been identified and successfully authenticated by the authentication method set by the authorized administrator.

Client authentication is controlled by the `pg_hba.conf` configuration file. The general format of the `pg_hba.conf` file is a set of records specifying a connection type, a client IP address range (if relevant for the connection type), a database name, a user name, and the authentication method to be used for

connections matching these parameters. The first record with a matching connection type, client address, requested database, and user name is used to perform authentication.

The TOE supports the following authentication methods:

- Password (*password*)
- Pluggable Authentication Modules (*pam*)
- Lightweight Directory Access Protocol (*ldap*)
- RADIUS Authentication (*radius*)

Password authentication is provided wholly within the TOE and available in "scram-sha-256", "md5" and "password" methods.

PAM, LDAP and RADIUS authentication are provided with the support of an external authentication mechanism provided by the IT environment.

The "Trust", "Ident" and "SSPI" authentication methods are prohibited in the evaluated configuration.

Password Authentication

PostgreSQL database passwords are separate from operating system user passwords. The password for each database user is stored in the `pg_authid` system catalog. By default, if no password has been set up, the stored password is null and password authentication always fails for that user.

The "scram-sha-256", "md5" and "password" password authentication methods operate similarly except for the way that the password is sent across the connection. Under the "scram-sha-256" method of password-based authentication, the password is encrypted using scram-sha-256 based encryption. Under the "md5 password" method of password-based authentication, the password is MD5-hashed when sent across the connection. Under the "password" method of password-based authentication, the password is sent across the connection as clear-text.

Note: The scram-sha-256 implementation is vendor developed to the RFC 7677 specification and is strictly used for password hashing. The scram-sha-256 cryptographic function implementation, or module, has not been FIPS certified. The correctness of the cryptographic module used by the TOE is by Vendor assertion; the correctness and conformance of this cryptographic module to the RFC 7677 standard is not to be part of this evaluation.

Note: The MD5 implementation is vendor developed to the RFC 1321 specification and is strictly used for password hashing. The MD5 cryptographic function implementation, or module, has not been FIPS certified. The correctness of the cryptographic module used by the TOE is by Vendor assertion; the correctness and conformance of this cryptographic module to the RFC 1321 standard is not be part of this evaluation.

PAM Authentication

This authentication method operates similarly to password except that it uses a Pluggable Authentication Module (PAM) as the authentication mechanism. PAM is used only to validate user name/password pairs. Therefore the user must already exist in the database before PAM can be used for authentication. A consumer could optionally supply their own service name after the PAM key word in the file `pg_hba.conf`, but this functionality is outside the scope of the evaluation.

LDAP Authentication

Lightweight Directory Access Protocol (LDAP) provides access to a directory server that serves up authentication information. PostgreSQL supports LDAP as defined in RFC 4510. LDAP can only be used to validate user name/password pairs. Therefore a user must already exist in the PostgreSQL database before LDAP can be used for authentication. The server and parameters used are specified after the LDAP key word in the file `pg_hba.conf`.

RADIUS Authentication

This authentication method operates similarly to password except that it uses RADIUS as the password verification method. RADIUS is used only to validate the user name/password pairs. Therefore the user must already exist in the database before RADIUS can be used for authentication.

7.1.3.2.2 Enhanced User-Subject Binding (IA-3)

FIA_USB_(EXT).2.1, FIA_USB_(EXT).2.2, FIA_USB_(EXT).2.3, FIA_USB_(EXT).2.4

PostgreSQL associates Roles with the security attributes listed in the first column of Table 6-3.

By default a user in PostgreSQL is assigned the values for the security attributes that are specified in the second column of Table 6-3.

The Role security attributes associated with subjects acting on behalf of users may be modified by means of an ALTER ROLE statement issued by a user that has sufficient privileges to assign the Role as determined by the following rules:

1) If the subject has Superuser role attribute, then the subject can change any user security attribute settings for any Role.

2) If the subject has CREATEROLE privilege, then the subject can change any user security attribute settings except for Superuser and replication roles.

3) If the subject does not satisfy conditions (1) or (2) above, then the subject cannot change any user security attribute.

Except for security attributes that are specified by default or are modified pursuant to the rules above, PostgreSQL does not provide for a user being assigned any other security attributes.

7.1.4 Security Management Functions

7.1.4.1 Security Roles (SM-1)

FMT_SMR.1.1, FMT_SMR.1.2

The TOE maintains the following roles:

(a) the Superuser; and

(b) customized roles created by the authorized administrator based on security attributes defined in FIA_ATD.1.

Customized roles can be created by within the TOE by either the Superuser or a user with the CREATEROLE privilege by using the CREATE ROLE command.

7.1.4.2 Management of TSF Data and Functions (SM-2)

FMT_MOF.1, FMT_MSA.1, FMT_MTD.1.1, FMT_SMF.1.1

The management and performance of functions relating to:

(a) the specification of events to be audited; and

(b) the determination of the auditable events to be included or excluded

is discussed in Section 7.1.1.1 Audit Data Generation (AU-1) and Section 7.1.1.2 Selective Audit (AU-2).

The management and enforcement of the DAC policy to restrict the ability to manage security attributes is discussed in Section 7.1.3.2.2 Enhanced User-Subject Binding (SM-3)

The performance of the security management functions related to the revocation of user and object security attributes covered by FMT_REV.1(*) is discussed in Section 7.1.4.4 Revocation (SM-4).

7.1.4.3 Static Attribute Initialization (SM-3)

FMT_MSA.3.1, FMT_MSA.3.2

The TSF provides restrictive default values for the security attributes that are used to enforce the DAC policy. The management and enforcement of the DAC policy to provide restrictive default values for security attributes that are to enforce the SFP is discussed in Section 7.1.3.2.2 Enhanced User-Subject Binding (IA-3).

Upon creation of a database object that can contain user data, only the object's owner has access to it. The owner must explicitly grant access to other roles. This includes databases, tables, Tablespaces, and Schema. Functions, procedures, and packages will only execute successfully, if the user has already been granted access to the underlying tables where data is stored. Please see Section 7.1.2.1 Subset Access Controls (DP-1) for details.

7.1.4.4 Revocation (SM-4)

FMT_REV.1.1(1), FMT_REV.1.2(1), FMT_REV.1.1(2), FMT_REV.1.2(2)

In PostgreSQL, the Superuser or a role with the CREATEROLE privilege can execute (i) the ALTER ROLE command used to grant or revoke the role security attributes listed in column 1 of Table 6-3 to User or (ii) the REVOKE SQL command to revoke the "in role" role security attribute. In each case, the command takes effect the next time that the user logs in.

The Superuser, object owner, or a user granted a privilege with the WITH GRANT OPTION clause, can execute the GRANT and REVOKE SQL commands, which are used to grant and revoke, respectively, /object/ privileges to users.

Users granted a privilege with the GRANT OPTION can then use the GRANT option in order to GRANT that privilege to a third user. Privileges that are granted in this fashion (i.e., indirectly) are referred to as dependent privileges.

Object security attributes, or privileges, are controlled by (i) the owner of the object or (ii) a Role that has been granted the privilege by the owner of the object using the WITH GRANT Option clause of the command. Object privileges are revoked using the REVOKE command.

Users can only revoke privileges that they granted, either directly or indirectly.

When object privileges are revoked, the revocation takes effect the next time the role opens the object. If an object privilege is revoked using the CASCADE option, dependent privileges are also revoked. Queries already in progress may be long running and will continue until they are completed unless some other action is taken.

7.1.5 Protection of the TSF Functions

7.1.5.1 Internal TSF Consistency (PT-1)

FPT_TRC.1

The TOE manages two forms of TSF data that may be contained in shared persistent storage. The first form is configuration data and is embodied by the TOE configuration files. The second form relates to RDBMS object privileges and is embodied by the operating system files that make up the system catalog tables.

When the TOE is initially started, the configuration data is read into the main postmaster process. In connection with the TOE start up, some RDBMS object data, including the related privilege-related data, is also read into process-local memory.

When a client connects to the TOE, a new operating system process is spawned from the TOE postmaster process. This new operating system process is referred to as the client backend process. This client backend process is initially an identical copy of the postmaster process, ensuring that TSF data is consistent. From that point forward, the client backend process is completely independent from the postmaster process and other backend processes.

TOE configuration data may be changed by only by authorized administrators in a centralized way by direct modification of the TOE configuration files, followed by an operating system SIGHUP signal which is sent to the postmaster and backend processes. Upon receipt of the SIGHUP, the backend re-reads the TOE

configuration files in order to update its TSF data.

TOE object privilege data may be changed only by authorized administrators in a centralized way by issuing a SQL command which alters RDBMS object privileges. After this change is committed to the TOE, the backend process will read the new TSF data on the next occurrence of access to the affected object.

7.1.6 TOE Access Functions

7.1.6.1 Limits on multiple concurrent sessions (TA-1)

FTA_MCS.1.1, FTA_MCS.1.2

During session establishment, the database name, username and source IP combination are validated against the `pg_hba.conf` file. Once this process succeeds, a connection attempt is made to the database at which time the server determines whether the *maximum number of connections allowed on the server* threshold is met. If the number of session exceeds the threshold limit the connection is refused. If the threshold limit has not been reached the user is authenticated.

The number of multiple concurrent sessions per role is determined by the "`connection limit`" role security attribute. The "connection limit" is checked during session establishment. The default value for the "connection limit" role security attribute is "-1", which in PostgreSQL means that an unlimited number of connections are allowed.

Connection limits per PostgreSQL system-wide can be set within the `postgresql.conf` configuration file by setting the `max_connnections` (integer) parameter. The Superuser, customized roles with the CREATEROLE attribute, and the Cluster Owner can change this parameter. The connection limit is not enforced against Superuser roles.

7.1.6.2 TOE Session Establishment (TA-2)

FTA_TSE.1.1

The TSF can deny session establishment based on user identity, group identity, database name, Host IP address, and/or subnet address contained in the `pg_hba.conf`, and maximum number of connections allowed on server, as designated in the `postgresql.conf` configuration file.

For additional discussion of client authentication based on user identity, group identity, database name, Host IP address, and/or subnet address contained in the `pg_hba.conf` see Section 7.1.3.2 Identification and Authentication (IA-2).

For additional discussion of the maximum number of connections allowed on the server, see Section 7.1.6.1 (TA-1) Limits on multiple concurrent sessions.

In addition to `postgresql.conf`, a PostgreSQL data directory contains a file `postgresql.auto.conf`, which has the same format as `postgresql.conf` but should never be edited manually. This file holds settings provided through the ALTER SYSTEM command. This file is automatically read whenever `postgresql.conf` is, and its settings take effect in the same way. Settings in `postgresql.auto.conf` override those in `postgresql.conf`.

The TSF can deny session establishment based on the time of day and the day of the week as set explicitly

by authorized administrator by use of the VALID UNTIL clause in the CREATE ROLE command. The clause VALID UNTIL 'timestamp' sets a date and time after which the role's password is no longer valid. If this clause is omitted the password will be valid for all time.

With the PAM authentication method, the Linux PAM system can be configured with the `pam_time` module to allow access only during certain times. The `pam_time` module is configured to be used through the `/etc/pam.d` configuration files. The `time.conf` file is used to define which users are allowed to connect during which times. The options for the allowed or disallowed times include the ability to specify the day of the week or all weekdays, as well as the ability to specify the time of day (eg: 0700-1900).

Appendix A-1

TOE TERMINOLOGY

The following terms used in this Security Target shall have the meaning set forth below.

Access Privilege. "Access Privilege" means an object security attribute.

Accessor. "Accessor" means subject accessing a database object.

Authorized User. "Authorized User" means an entity that has been properly identified and authenticated. These users are considered to be legitimate users of the TOE.

Client Connectors. "Client Connectors" means standardized programming interfaces allow a software developer to connect a customer-specific application to PostgreSQL.

Cluster Owner. "Cluster Owner" means a user created during the installation process that is given ownership permissions of the TOE. This user is maintained by the OS and can only access the TSF data stored at the OS level after being authenticated at the OS level.

Database. "Database" means one or more named schemas, which in turn contain tables.

Foreign Data Wrapper. "Foreign Data Wrapper" means a standardized approach in PostgreSQL for handling access to remote objects from SQL databases.

Function. "Fuction" means a predefined block of statements that can be invoked with SQL commands, trigger operators, in view definitions and indexes and return a value.

pg_hba.conf. "pg_hba.conf" means a configuration file in PostgreSQL that is stored in the database cluster's data directory, modifiable only by the Cluster Owner at the operating system level.

postgrespl.conf. "postgrespl.conf" means a human readable operating system level configuration file that can be directly reviewed and modified by a Cluster Owner using an operating system text editor provided with the IT environment.

psql. "psql" means the terminal-based front-end to PostgreSQL enabling PostgreSQL users either type in queries interactively, issue them to PostgreSQL, and see the query results or input them from a file.

Role. "Role" means, in PostgreSQL, either defined individual users, groups of users or sets of access privileges.

RPM Package Manager. "RPM Package Manager" means a collection of software tools that automates the process of installing, upgrading, configuring, and removing software packages for a computer's operating system in a consistent manner.

Schema. "Schema" means a collection of database objects as well as logical structures of data.

Security Definer. "Security Definer" means a function in PostgreSQL that is used to specify that the applicable function is to be executed with the privileges of the role which owns the function.

Security Invoker. "Security Invoker" means a a function in PostgreSQL that is used to specify that that

the applicable function is to be executed with the privileges of the user that calls it.

Superuser. "Superuser" means a PostgreSQL Role that has been assigned the 'superuser' Role attribute and as a consequence bypasses all permission checks in PostgreSQL except the login requirement. Importantly, the Superuse is represents the authorized administrator as defined in the DBMS PP.

Tablespace. "Tablespaces in PostgreSQL allow database administrators to define locations in the file system where the files representing database objects can be stored. Once created, a tablespace can be referred to by name when creating database objects. Tables, indexes, and entire databases can be assigned to particular tablespaces.

Trigger. "Trigger" means an attribute of a Table that is a predefined block of statements executed when a DELETE, INSERT, TRUNCATE, or UPDATE command is executed on the Table.

View. "View" means a selection of rows and columns from a table or set of joined tables.

Write-Ahead Logging. "Write-Ahead Logging" means a method for ensuring data integrity by which changes to data files (where Tables and indexes reside) must be written only after those changes have been logged (i.e., after log records describing the changes have been flushed to permanent storage).

Appendix A-2

DBMS PP TERMINOLOGY

The following terms used in this Security Target shall have the meetings set forth below. The terms and associated meanings set forth in this Appendix A-2 are from the DBMS PP.

Access. "Access" means interaction between an entity and an object that results in the flow or modification of data.

Access Control. "Access Control" means security service that controls the use of resources (hardware and software) and the disclosure and modification of data (stored or communicated).

Accountability. "Accountability" means a property that allows activities in an IT system to be traced to the entity responsible for the activity.

Administrator. "Administrator" means a user who has been specifically granted the authority to manage some portion or the entire TOE and whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TSP.

Assurance. "Assurance" meansa measure of confidence that the security features of an IT system are sufficient to enforce its security policy.

Attack. "Attack" means an intentional act attempting to violate the security policy of an IT system.

Authentication. "Authentication" means a security measure that verifies a claimed identity.

Authentication Data. "Authentication Data" means information used to verify a claimed identity.

Authorization. "Authorization" means permission, granted by an entity authorized to do so, to perform functions and access data.

Authorized Administrator. "Authorized Administrator" means the authorized person in contact with the TOE who is responsible for maintaining its operational capability.

Authorized User. "Authorized User" means an authenticated user who may, in accordance with the TSP, perform an operation.

Availability. "Availability" means timely (according to a defined metric), reliable access to IT resources.

Compromise. "Compromise" means a violation of a security policy.

Confidentiality. "Confidentiality" means a security policy pertaining to the disclosure of data.

Configuration Data. "Configuration Data" means data that is used in configuring the TOE.

Conformant Product. "Conformant Product" means a TOE that satisfied all the functional security requirements in Section 7.1 of the DBMS PP and satisfies all the TOE security assurance requirements in Section 7.2 of the DBMS PP.

Database Management System (DBMS). "Database Management System (DBMS)" means a suite of

programs that typically manage large structured sets of persistent data, offering ad hoc query facilities to many users. They are widely used in business applications.

Discretionary Access Control (DAC). "Discretionary Access Control (DAC)" means a means of restricting access to objects based on the identity of subjects and/or groups to which they belong. Those controls are discretionary in the sense that a subject with certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.

Enclave. Enclave means a collection of entities under the control of a single authority and having a homogeneous security policy. They may be logical, or may be based on physical location and proximity.

Entity. "Entity" means a subject, object, user or another IT device, which interacts with TOE objects, data, or resources.

Executable Code Within the TSF. "Executable Code Within the TSF" the software that makes up the TSF which is in a form that can be run by the computer.

External IT Entity. "External IT Entity" means any trusted Information Technology (IT) product or system, outside of the TOE, which may, in accordance with the TSP, perform an operation.

Identity. "Identity" means a representation (e.g., a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.

Integrity. "Integrity" means a security policy pertaining to the corruption of data and TSF mechanisms.

Named Object. "Named Object" means an object that exhibits all of the following characteristics:

- The object may be used to transfer information between subjects of differing user and/or group identities within the TSF.
- Subjects in the TOE must be able to require a specific instance of the object.
- The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user and/or group identities to require the same instance of the object.

Object. "Object" means an entity within the TSC that contains or receives information and upon which subjects perform operations.

Operating Environment. "Operating Environment" means the total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.

Public Object. "Public Object" means an object for which the TSF unconditionally permits all entities "read" access. Only the TSF or authorized administrators may create, delete, or modify the public objects.

Row Security Policy. "Row Security Policy" means a policy determining access to the table for selecting rows or adding rows when row security is enabled.

Secure State. "Secure State" means a condition in which all TOE security policies are enforced.

Security Attributes. "Security Attributes" means TSF data associated with subjects, objects, and users that are used for the enforcement of the TSP.

Security Level. "Security Level" means the combination of a hierarchical classification and a set of non-hierarchical categories that represent the sensitivity of the information.

Sensitive Information. "Sensitive Information" means information that, as determined by a competent authority, must be protected because its unauthorized disclosure, alteration, loss, or destruction will at least cause perceivable damage to someone or something.

Subject. "Subject" means an entity within the TSC that causes operation to be performed.

Threat. "Threat" means capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.

TOE Resources. "TOE Resources" means anything useable or consumable in the TOE.

Unauthorized User. "Unauthorized User" means a user who may obtain access only to system provided public objects if any exist.

User. "User" means any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

Vulnerability. "Vulnerability" means a weakness that can be exploited to violate the TOE security policy.

Appendix B-1

TOE ACRONYMS

The following acronyms shall have the meanings set forth below.

Acronyms	Definition
API	Application Programming Interface
CLI	Command Line Interface
DBA	Database Administrator
DDL	Data Definition Language
DML	Data Manipulation Language
FDW	Foreign Data Wrapper
GSSAPI	Generic Security Services Application Programming Interface
LDAP	Lightweight Directory Access Protocol
PAM	Pluggable Authentication Modules
RDBMS	Relational DBMS
RPM	RPM Package Manager
SQL	Structured Query Language
SSL	Secure Socket Layer Protocol
SSPI	Security Services Provider Interface
WAL	Write-Ahead Logging

Appendix B-2

CC ACRONYMS

The following acronyms shall have the meanings set forth below. The acronyms and associated meanings set forth in this Appendix B-2 are from the DBMS PP.

Acronyms	Definition
CA	Certificate Authority
CC	Common Criteria
CCIMB	Common Criteria Interpretations Management Board
CM	Configuration Management
COTS	Commercial Off The Shelf
DAC	Discretionary Access Control
DBMS	Database Management System
DBMS PP	Database Management System Protection Profile
EAL	Evaluation Assurance Level
I&A	Identification and Authentication
IT	Information Technology
LAN	Local Area Network
OS	Operating System Profile
OSP	Organizational Security Policy
PP	Protection Profile
SAR	Security Assurance Requirement
SFP	Security Functional Policies
SFR	Security Functional Requirement
SPD	Security Functional Definition
ST	Security Target
TOE	Target of Evaluation
TSE	TSF Scope of Control
TSF	TOE Security Functions
TSFI	TSF Interfaces
TSP	TOE Security Policy