

Oracle® Database Gateway

for Teradata User's Guide

11g Release 1 (11.1)

B31050-01

May 2007

Beta Draft

Oracle Database Gateway for Teradata User's Guide, 11g Release 1 (11.1)

B31050-01

Copyright © 2002, 2007, Oracle. All rights reserved.

Primary Author: Amitai Sela

Contributing Authors: Laurel Hale, Cynthia Kibbe, Kishan Peyetti, Maitreyee Chaliha

Contributors: Orit Curiel, Jacco Draaijer, Vira Goorah

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Alpha and Beta Draft documentation are considered to be in prerelease status. This documentation is intended for demonstration and preliminary use only. We expect that you may encounter some errors, ranging from typographical errors to data inaccuracies. This documentation is subject to change without notice, and it may not be specific to the hardware on which you are using the software. Please be advised that Oracle Corporation does not warrant prerelease documentation and will not be responsible for any loss, costs, or damages incurred due to the use of this documentation.

Contents

Preface	ix
Audience	ix
Documentation Accessibility	ix
Related Documents	x
Conventions	xi
 1 Introduction	
Overview	1-1
Heterogeneous Services Technology	1-2
Oracle Database Gateways	1-2
 2 Teradata Gateway Features and Restrictions	
Using the Pass-Through Feature	2-1
Database Compatibility Issues for Teradata	2-2
Schema Considerations	2-2
Naming Rules	2-2
Rules for Naming Objects	2-2
Case Sensitivity	2-2
Data Types	2-3
Binary Literal Notation	2-3
Data Type Conversion	2-4
Queries	2-4
Row Selection	2-4
Empty Bind Variables	2-4
Locking	2-4
Known Restrictions	2-4
Transactional Integrity	2-5
Transaction Capability	2-5
COMMIT or ROLLBACK in PL/SQL Cursor Loops Closes Open Cursors	2-5
Pass-Through Feature	2-6
Bind Variables for Date Columns	2-6
SQL Syntax	2-6
WHERE CURRENT OF Clause	2-6
CONNECT BY Clause	2-6
ROWID	2-6

EXPLAIN PLAN Statement.....	2-7
Callback Support.....	2-7
SQL*Plus.....	2-7
Database Links.....	2-7
Stored Procedures	2-7
Known Problems	2-8
Encrypted Format Login	2-8
Teradata LONG VARCHAR Data Type	2-8
Schema Names and PL/SQL.....	2-8
Data Dictionary Views and PL/SQL.....	2-9

3 Case Studies

Case Descriptions.....	3-1
Installation Media Contents.....	3-1
Demonstration Files.....	3-2
Demonstration Requirements.....	3-2
Creating Demonstration Tables	3-2
Demonstration Table Definitions.....	3-3
Demonstration Table Contents	3-3
Case 1: Simple Queries	3-4
Case 2: A More Complex Query	3-4
Case 3: Joining Teradata Tables	3-4
Case 4: Write Capabilities.....	3-4
DELETE Statement.....	3-5
UPDATE Statement	3-5
INSERT Statement.....	3-5
Case 5: Data Dictionary Query	3-5
Case 6: The Pass-Through Feature	3-5
UPDATE Statement	3-5
SELECT Statement	3-5

A Data Type Conversion

Data Type Conversion.....	A-1
---------------------------	-----

B Supported SQL Syntax and Functions

Supported SQL Statements	B-1
DELETE	B-1
INSERT	B-1
SELECT	B-1
UPDATE	B-2
Oracle Functions	B-2
Functions Not Supported by Teradata.....	B-2
Functions Supported by Teradata	B-2
Arithmetic Operators	B-2
Comparison Operators.....	B-2
Group Functions	B-3

String Functions	B-3
Other Functions	B-3
C Data Dictionary	
Data Dictionary Support	C-1
Teradata System Catalog Tables	C-1
Accessing the Gateway Data Dictionary	C-1
Direct Queries to Teradata Tables	C-2
Supported Views and Tables	C-2
Data Dictionary Mapping	C-3
Default Column Values	C-4
Gateway Data Dictionary Descriptions	C-4
D Initialization Parameters	
Initialization Parameter File Syntax	D-1
Oracle Database Gateway for Teradata Initialization Parameters	D-2
Initialization Parameter Description	D-3
HS_DB_DOMAIN	D-3
HS_DB_INTERNAL_NAME	D-3
HS_DB_NAME	D-4
HS_DESCRIBE_CACHE_HWM	D-4
HS_LANGUAGE	D-4
Character Sets	D-4
Language	D-5
Territory	D-5
HS_LONG_PIECE_TRANSFER_SIZE	D-5
HS_OPEN_CURSORS	D-5
HS_RPC_FETCH_REBLOCKING	D-6
HS_RPC_FETCH_SIZE	D-6
HS_TIME_ZONE	D-6
HS_TRANSACTION_MODEL	D-7
IFILE	D-7
HS_FDS_CONNECT_INFO	D-8
HS_FDS_DEFAULT_OWNER	D-8
HS_FDS_RECOVERY_ACCOUNT	D-8
HS_FDS_RECOVERY_PWD	D-9
HS_FDS_TRACE_LEVEL	D-9
HS_FDS_TRANSACTION_LOG	D-9
HS_FDS_FETCH_ROWS	D-9

List of Tables

A-1	Data Type Conversions	A-1
C-1	Oracle Data Dictionary View Names and Teradata Equivalents	C-3
C-2	ALL_CATALOG	C-4
C-3	ALL_COL_COMMENTS	C-4
C-4	ALL_CONS_COLUMNS	C-4
C-5	ALL_CONSTRAINTS	C-5
C-6	ALL_IND_COLUMNS	C-5
C-7	ALL_INDEXES	C-5
C-8	ALL_OBJECTS	C-7
C-9	ALL_TAB_COLUMNS	C-7
C-10	ALL_TAB_COMMENTS	C-8
C-11	ALL_TABLES	C-8
C-12	ALL_USERS	C-10
C-13	ALL_VIEWS	C-10
C-14	DBA_CATALOG	C-10
C-15	DBA_COL_COMMENTS	C-10
C-16	DBA_OBJECTS	C-10
C-17	DBA_TAB_COLUMNS	C-11
C-18	DBA_TAB_COMMENTS	C-12
C-19	DBA_TABLES	C-12
C-20	DICT_COLUMNS	C-13
C-21	DICTIONARY	C-13
C-22	DUAL	C-13
C-23	USER_CATALOG	C-14
C-24	USER_COL_COMMENTS	C-14
C-25	USER_CONS_COLUMNS	C-14
C-26	USER_CONSTRAINTS	C-14
C-27	USER_IND_COLUMNS	C-15
C-28	USER_INDEXES	C-15
C-29	USER_OBJECTS	C-16
C-30	USER_TAB_COLUMNS	C-17
C-31	USER_TAB_COMMENTS	C-17
C-32	USER_TABLES	C-18
C-33	USER_USERS	C-19
C-34	USER_VIEWS	C-19

Preface

This manual describes the Oracle Database Gateway for Teradata, which enables Oracle client applications to access Teradata data through Structured Query Language (SQL). The gateway, with the Oracle database, creates the appearance that all data resides on a local Oracle database, even though the data can be widely distributed.

This preface covers the following topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)

Audience

This manual is intended for Oracle database administrators who perform the following tasks:

- Installing and configuring the Oracle Database Gateway for Teradata
- Diagnosing gateway errors
- Using the gateway to access Teradata data

Note: You should understand the fundamentals of Oracle Database Gateways and the UNIX based platforms before using this guide to install or administer the gateway.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documents

For more information, see these Oracle resources:

- *Oracle Database New Features Guide*
- *Oracle Call Interface Programmer's Guide*
- *Oracle Database Administrator's Guide*
- *Oracle Database Advanced Application Developer's Guide*
- *Oracle Database Concepts*
- *Oracle Database Performance Tuning Guide*
- *Oracle Database Error Messages*
- *Oracle Database Globalization Support Guide*
- *Oracle Database Reference*
- *Oracle Database SQL Language Reference*
- *Oracle Database Net Services Administrator's Guide*
- *SQL*Plus User's Guide and Reference*
- *Oracle Database Heterogeneous Connectivity Administrator's Guide*
- *Oracle Database Security Guide*

Many of the examples in this book use the sample schemas, which are installed by default when you select the Basic Installation option with an Oracle Database installation. Refer to *Oracle Database Sample Schemas* for information on how these schemas were created and how you can use them yourself.

Printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://otn.oracle.com/membership/>

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://otn.oracle.com/documentation/>

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)
- [Conventions in Code Examples](#)

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an index-organized table .
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, data types, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executable programs, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names and connect identifiers, user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. <i>Note:</i> Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to start SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Set the QUERY_REWRITE_ENABLED initialization parameter to true. Connect as oe user. The JRepUtil class implements these methods.
<i>lowercase italic monospace (fixed-width) font</i>	Lowercase italic monospace font represents placeholders or variables.	You can specify the <i>parallel_clause</i> . Run <i>old_release</i> .SQL where <i>old_release</i> refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[]	Anything enclosed in brackets is optional.	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	Braces are used for grouping items.	{ENABLE DISABLE}
	A vertical bar represents a choice of two options.	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	Ellipsis points mean repetition in syntax descriptions. In addition, ellipsis points can mean an omission in code examples or text.	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
Other symbols	You must use symbols other than brackets ([]), braces ({ }), vertical bars (), and ellipsis points (...) exactly as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. Because these terms are not case sensitive, you can use them in either UPPERCASE or lowercase.	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;
lowercase	Lowercase typeface indicates user-defined programmatic elements, such as names of tables, columns, or files. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;

Introduction

This chapter introduces the challenge faced by organizations when running several different database systems. It briefly covers Heterogeneous Services, the technology that the Oracle Database Gateway for Teradata is based on.

To get a good understanding of generic gateway technology, Heterogeneous Services, Generic Connectivity, and how Oracle Database Gateways fit in the picture, reading the *Oracle Database Heterogeneous Connectivity Administrator's Guide* first is highly recommended.

This chapter contains the following sections:

- [Overview](#)
- [Heterogeneous Services Technology](#)
- [Oracle Database Gateways](#)

Overview

Heterogeneous data access is a problem that affects a lot of companies. A lot of companies run several different database systems. Each of these systems stores data and has a set of applications that run against it. Consolidation of this data in one database system is often hard - in large part because that many of the applications that run against one database may not have an equivalent that runs against another. Until such time as migration to one consolidated database system is made feasible, it is necessary for the various heterogeneous database systems to interoperate.

Oracle Database Gateways provide the ability to transparently access data residing in a non-Oracle system from an Oracle environment. This transparency eliminates the need for application developers to customize their applications to access data from different non-Oracle systems, thus decreasing development efforts and increasing the mobility of the application. Applications can be developed using a consistent Oracle interface for both Oracle and Teradata.

Gateway technology is composed of two parts: a component that has the generic technology to connect to a non-Oracle system, which is common to all the non-Oracle systems, called Heterogeneous Services, and a component that is specific to the non-Oracle system that the gateway connects to. Heterogeneous Services, in conjunction with the Oracle Database Gateway agent, enables transparent access to non-Oracle systems from an Oracle environment.

Heterogeneous Services Technology

Heterogeneous Services provides the generic technology for connecting to non-Oracle systems. As an integrated component of the database, Heterogeneous Services can exploit features of the database, such as the powerful SQL parsing and distributed optimization capabilities.

Heterogeneous Services extend the Oracle SQL engine to recognize the SQL and procedural capabilities of the remote non-Oracle system and the mappings required to obtain necessary data dictionary information. Heterogeneous Services provides two types of translations: the ability to translate Oracle SQL into the proper dialect of the non-Oracle system as well as data dictionary translations which displays the metadata of the non-Oracle system in the local format. For situations where no translations are available, native SQL can be issued to the non-Oracle system using the pass-through feature of Heterogeneous Services.

Heterogeneous Services also maintains the transaction coordination between Oracle and the remote non-Oracle system, such as providing the two-phase commit protocol to ensure distributed transaction integrity, even for non-Oracle systems that do not natively support two-phase commit.

See Also: *Oracle Database Heterogeneous Connectivity Administrator's Guide* for more information about Heterogeneous Services.

Oracle Database Gateways

The capabilities, SQL mappings, data type conversions, and interface to the remote non-Oracle system are contained in the gateway. The gateway interacts with Heterogeneous Services to provide the transparent connectivity between Oracle and non-Oracle systems.

The gateway must be installed on a machine running either the Teradata database or the Teradata 32-bit client. This machine can be the same machine as the Oracle database or on the same machine as the Teradata database or on a third machine as a standalone. Each configuration has its advantages and disadvantages. The issues to consider when determining where to install the gateway are network traffic, operating system platform availability, hardware resources and storage.

Teradata Gateway Features and Restrictions

After the gateway is installed and configured, you can use the gateway to access Teradata data, pass Teradata commands from applications to the Teradata database, perform distributed queries, and copy data.

This chapter contains the following sections:

- [Using the Pass-Through Feature](#)
- [Database Compatibility Issues for Teradata](#)
- [Known Restrictions](#)
- [Known Problems](#)

Using the Pass-Through Feature

The gateway can pass Teradata commands or statements from the application to the Teradata database using the DBMS_HS_PASSTHROUGH package.

Use the DBMS_HS_PASSTHROUGH package in a PL/SQL block to specify the statement to be passed to the Teradata database, as follows:

```
DECLARE
    num_rows INTEGER;
BEGIN
    num_rows := DBMS_HS_PASSTHROUGH.EXECUTE_IMMEDIATE@TERA('command');
END;
/
```

Where *command* cannot be one of the following:

- BEGIN TRANSACTION
- BT
- COMMIT
- END TRANSACTION
- ET
- ROLLBACK

The DBMS_HS_PASSTHROUGH package supports passing bind values and executing SELECT statements.

See Also: *Oracle Database PL/SQL Packages and Types Reference* and Chapter 3, *Features of Oracle Database Gateways and Generic Connectivity*, of *Oracle Database Heterogeneous Connectivity Administrator's Guide* for more information about the DBMS_HS_PASSTHROUGH package.

Database Compatibility Issues for Teradata

Teradata and Oracle databases function differently in some areas, causing compatibility problems. The following compatibility issues are described in this section:

- [Schema Considerations](#)
- [Naming Rules](#)
- [Data Types](#)
- [Queries](#)
- [Locking](#)

Schema Considerations

The Oracle concept of a schema does not exist in Teradata. An owner included in a query is recognized by Teradata as a database name.

When querying data dictionary tables, the following results are returned:

- ALL_* data dictionary tables, data for every Teradata database is returned. The Teradata database name is returned in the owner column.
- ALL_USERS data dictionary table, the name of every Teradata database is returned.
- USER_* data dictionary tables, data for the default Teradata database is returned. If a default Teradata database is not defined, the DBC Teradata system database is used.

Naming Rules

Naming rule issues include the following:

- [Rules for Naming Objects](#)
- [Case Sensitivity](#)

Rules for Naming Objects

Oracle and Teradata use different database object naming rules. For example, the maximum number of characters allowed for each object name can be different. Also, the use of single and double quotation marks, case sensitivity, and the use of alphanumeric characters can all be different.

See Also: *Oracle Database Reference* and Teradata documentation.

Case Sensitivity

The Oracle database defaults to uppercase unless you surround identifiers with double quote characters. For example, to refer to the Teradata table called emp, enter the name with double quote characters, as follows:

```
SQL> SELECT * FROM "emp"@TERA;
```

However, to refer to the Teradata table called emp owned by Scott from an Oracle application, enter the following:

```
SQL> SELECT * FROM "Scott"."emp"@TERA;
```

If the Teradata table called emp is owned by SCOTT, a table owner name in uppercase letters, you can enter the owner name without double quote characters, as follows:

```
SQL> SELECT * FROM SCOTT."emp"@TERA;
```

or

```
SQL> SELECT * FROM scott."emp"@TERA;
```

Oracle Corporation recommends that you surround all Teradata object names with double quote characters and use the exact letter case for the object names as they appear in the Teradata data dictionary. This convention is not required when referring to the supported Oracle data dictionary tables or views listed in [Appendix C, "Data Dictionary"](#).

If existing applications cannot be changed according to these conventions, create views in Oracle to associate Teradata names to the correct letter case. For example, to refer to the Teradata table emp from an existing Oracle application by using only uppercase names, define the following view:

```
SQL> CREATE VIEW EMP (EMPNO, ENAME, SAL, HIREDATE)
      AS SELECT "empno", "ename", "sal", "hiredate"
      FROM "emp"@TERA;
```

With this view, the application can issue statements such as the following:

```
SQL> SELECT EMPNO, ENAME FROM EMP;
```

Using views is a workaround solution that duplicates data dictionary information originating in the Teradata data dictionary. You must be prepared to update the Oracle view definitions whenever the data definitions for the corresponding tables are changed in the Teradata database.

Data Types

Data type issues include the following:

- [Binary Literal Notation](#)
- [Data Type Conversion](#)

Binary Literal Notation

Oracle SQL uses hexadecimal digits surrounded by single quotes to express literal values being compared or inserted into columns defined as data type RAW.

This notation is not converted to syntax compatible with the Teradata VARBINARY and BINARY data types (a ff surrounded by single quotes followed by hexadecimal digits).

For example, the following statement is not supported:

```
SQL> INSERT INTO BINARY_TAB@TERA VALUES ('ff'xb)
```

Where BINARY_TAB contains a column of data type VARBINARY or BINARY. Use bind variables when inserting into or updating VARBINARY and BINARY data types.

Data Type Conversion

Teradata does not support implicit date conversions. Such conversions must be explicit.

For example, the gateway issues an error for the following `SELECT` statement:

```
SELECT DATE_COL FROM TEST@TERA WHERE DATE_COL = "1-JAN-2001";
```

To avoid problems with implicit conversions, add explicit conversions, as in the following:

```
SELECT DATE_COL FROM TEST@TERA WHERE DATE_COL = TO_DATE("1-JAN-2001")
```

See Also: [Appendix A, "Data Type Conversion"](#) for more information about restrictions on data types.

Queries

Query issues include the following:

- [Row Selection](#)
- [Empty Bind Variables](#)

Row Selection

Teradata evaluates a query condition for all selected rows before returning any of the rows. If there is an error in the evaluation process for one or more rows, no rows are returned even though the remaining rows satisfy the condition.

Oracle evaluates the query condition row-by-row and returns a row when the evaluation is successful. Rows are returned until a row fails the evaluation.

Empty Bind Variables

For `VARCHAR` bind variables, the gateway passes empty bind variables to the Teradata database as a `NULL` value.

Locking

The locking model for an Teradata database differs significantly from the Oracle model. The gateway depends on the underlying Teradata behavior, so Oracle applications that access Teradata through the gateway can be affected by the following possible scenarios:

- Read access might block write access.
- Write access might block read access.
- Statement-level read consistency is not guaranteed.

See Also: Teradata documentation for information about the Teradata locking model.

Known Restrictions

If you encounter incompatibility problems not listed in this section or in "[Known Problems](#)" on page 2-8, please contact Oracle Support Services. The following section describes the known restrictions and includes suggestions for dealing with them when possible:

- [Transactional Integrity](#)
- [Transaction Capability](#)
- [COMMIT or ROLLBACK in PL/SQL Cursor Loops Closes Open Cursors](#)
- [Pass-Through Feature](#)
- [Bind Variables for Date Columns](#)
- [SQL Syntax](#)
- [SQL*Plus](#)
- [Database Links](#)
- [Stored Procedures](#)

Note: If you have any questions or concerns about the restrictions, contact Oracle Support Services.

Transactional Integrity

The gateway cannot guarantee transactional integrity in the following cases:

- When a statement that is processed by the gateway causes an implicit commit in the target database
- When the target database is configured to work in autocommit mode

Note: Oracle corporation strongly recommends the following:

- If you know that executing a particular statement causes an implicit commit in the target database, then ensure that this statement is executed in its own transaction.
 - Do not configure the target database to work in autocommit mode.
-

Transaction Capability

The gateway does not support savepoints. If a distributed update transaction is under way involving the gateway and a user attempts to create a savepoint, the following error occurs:

ORA-02070: database *dblink* does not support savepoint in this context

By default, the gateway is configured as COMMIT_CONFIRM and it is always the commit point site when the Teradata database is updated by the transaction.

COMMIT or ROLLBACK in PL/SQL Cursor Loops Closes Open Cursors

Any COMMIT or ROLLBACK issued in a PL/SQL cursor loop closes all open cursors, which can result in the following error:

ORA-1002: fetch out of sequence

To prevent this error, move the COMMIT or ROLLBACK statement outside the cursor loop.

Pass-Through Feature

Oracle Corporation recommends that you place a DDL statement in its own transaction when executing such a statement with the pass-through feature. An explicit `COMMIT` must be issued after the DDL statement.

If the SQL statements being passed through the gateway result in an implicit commit at the Teradata database, the Oracle transaction manager is unaware of the commit and an Oracle `ROLLBACK` command cannot be used to roll back the transaction.

Bind Variables for Date Columns

Due to limitations in NCR's ODBC driver for Teradata, you cannot compare columns of data type `TIME` or `TIMESTAMP` to a bind variable.

The following SQL statement causes an error message:

```
SQL> select time_column from time_table@tg4tera where time_column = :a;
```

code to be reviewed

The following error is issued:

```
ORA-28500: connection from ORACLE to a non-Oracle system returned this message:
[Transparent gateway for TERA]DRV_Execute: [NCR][ODBC Teradata][Teradata
RDBMS]
Invalid operation on an ANSI Datetime or Interval value. (SQL State: 22005;SQL
Code: -5407)
ORA-02063: preceding 2 lines from TERA
```

SQL Syntax

This section lists restrictions on the following SQL syntax:

- [WHERE CURRENT OF Clause](#)
- [CONNECT BY Clause](#)
- [ROWID](#)
- [EXPLAIN PLAN Statement](#)
- [Callback Support](#)

See Also: [Appendix B, "Supported SQL Syntax and Functions"](#)
for more information about restrictions on SQL syntax.

WHERE CURRENT OF Clause

`UPDATE` and `DELETE` statements with the `WHERE CURRENT OF` clause are not supported by the gateway because they rely on the Oracle `ROWID` implementation. To update or delete a specific row through the gateway, a condition style `WHERE` clause must be used.

CONNECT BY Clause

The gateway does not support the `CONNECT BY` clause in a `SELECT` statement.

ROWID

The Oracle `ROWID` implementation is not supported.

EXPLAIN PLAN Statement

The `EXPLAIN PLAN` statement is not supported.

Callback Support

SQL statements that require the gateway to callback to Oracle database would not be supported.

The following categories of SQL statements will result in a callback:

- Any DML with a sub-select, which refers to a table in Oracle database. For example:

```
INSERT INTO emp@non_oracle SELECT * FROM oracle_emp;
```
- Any `DELETE`, `INSERT`, `UPDATE` or "`SELECT... FOR UPDATE...`" SQL statement containing SQL functions or statements that need to be executed at the originating Oracle database.

These SQL functions include `USER`, `USERENV`, and `SYSDATE`, and the SQL statements are in selects of data from the originating Oracle database. For example:

```
DELETE FROM emp@non_oracle WHERE hiredate > SYSDATE;
```

```
SELECT ename FROM tkhoemp@non_oracle
WHERE hiredate IN (SELECT hiredate FROM tkhoemp)
FOR UPDATE OF empno;
```

- Any SQL statement that involves a table in Oracle database, and a `LONG` or `LOB` column in a remote table. For example:

```
SELECT a.long1, b.empno FROM scott.table@non_oracle a, emp b
WHERE a.id=b.empno;
```

```
SELECT a.long1, b.dummy FROM table_non@non_oracle a, dual b;
```

where `a.long1` is a `LONG` column.

SQL*Plus

You need to use double quotes to wrap around lower case table names.

For example:

```
copy from tkhouser/tkhouser@inst1 insert loc_tkhodept using select * from
"tkhodept"@holink2;
```

Database Links

The gateway is not a shared server process and cannot support shared database links. Each gateway session spawns a separate gateway process and connections cannot be shared.

Stored Procedures

The gateway does not support the procedure feature that allows the execution of stored procedures in a non-Oracle database.

Known Problems

This section describes known problems and includes suggestions for correcting them when possible. If you have any questions or concerns about the problems, contact Oracle Support Services. A current list of problems is available online. Contact your local Oracle Corporation office for information about accessing the list.

The following known problems are described in this section:

- [Encrypted Format Login](#)
- [Teradata LONG VARCHAR Data Type](#)
- [Schema Names and PL/SQL](#)
- [Data Dictionary Views and PL/SQL](#)

Encrypted Format Login

The Oracle database no longer supports the initialization parameter `DBLINK_ENCRYPT_LOGIN`. Up to version 7.3, this parameter's default `TRUE` value prevented the password for the login user ID from being sent over the network (in the clear). Later versions automatically encrypt the password.

If this parameter is set to `TRUE` in the initialization parameter file used by the Oracle9i database, you must change the setting to `FALSE`, the default setting, to allow Oracle9i to communicate with the gateway.

In the current release, Oracle Database 11g, Release 11.1, the `DBLINK_ENCRYPT_LOGIN` initialization parameter is obsolete, so you need not check it.

Teradata LONG VARCHAR Data Type

The following restrictions apply when using `LONG VARCHAR` data types:

- An unsupported SQL function cannot be used in a SQL statement that accesses a column defined as Teradata data type `LONG VARCHAR`.
- You cannot use SQL*Plus to select data from a column defined as Teradata data type `LONG VARCHAR` when the data is greater than 80 characters in length. Oracle Corporation recommends using Pro*C or Oracle Call Interface to access such data in a Teradata database.
- `LONG VARCHAR` data types must be `NULLABLE` for `INSERT` or `UPDATE` to work.
- A table including a `LONG VARCHAR` column must have a unique index defined on the table or the table must have a separate column that serves as a primary key.
- `LONG VARCHAR` data cannot be read through pass-through queries.

The gateway does not support the PL/SQL function `COLUMN_VALUE_LONG` of the `DBMS_SQL` package.

See Also: [Appendix B, "Supported SQL Syntax and Functions"](#)
for more information about restrictions on SQL syntax.

Schema Names and PL/SQL

If you do not prefix a Teradata database object with its schema name in a SQL statement within a PL/SQL block, the following error message occurs:

```
ORA-6550 PLS-201 Identifier table_name must be declared.
```

Change the SQL statement to include the schema name of the object.

Data Dictionary Views and PL/SQL

You cannot refer to data dictionary views in SQL statements that are inside a PL/SQL block.

Case Studies

The following case studies for Teradata demonstrate some of the features of the Oracle Database Gateway. You can verify that the gateway is installed and operating correctly by using the demonstration files included on the distribution media.

The demonstration files are automatically copied to disk when the gateway is installed.

This chapter contains the following sections:

- [Case Descriptions](#)
- [Installation Media Contents](#)
- [Demonstration Files](#)
- [Demonstration Requirements](#)
- [Creating Demonstration Tables](#)
- [Case 1: Simple Queries](#)
- [Case 2: A More Complex Query](#)
- [Case 3: Joining Teradata Tables](#)
- [Case 4: Write Capabilities](#)
- [Case 5: Data Dictionary Query](#)
- [Case 6: The Pass-Through Feature](#)

Case Descriptions

The cases illustrate:

- A simple query (Case 1)
- A more complex query (Case 2)
- Joining Teradata tables (Case 3)
- Write capabilities (Case 4)
- A data dictionary query (Case 5)
- The pass-through feature (Case 6)

Installation Media Contents

The distribution media contains the following:

- Demonstration files.
- One SQL script file that creates the demonstration tables in the Teradata database.
- One SQL script file that drops the demonstration tables from the Teradata database.

Demonstration Files

After a successful gateway installation, use the demonstration files stored in the directory `$ORACLE_HOME/tg4tera/demo` where `$ORACLE_HOME` is the directory under which the gateway is installed. The directory contains the demonstration files, as listed in the following table:

Demonstration Files	Demonstration Files
<code>bldtera.sql</code>	<code>case4c.sql</code>
<code>case1.sql</code>	<code>case5.sql</code>
<code>case2.sql</code>	<code>case6a.sql</code>
<code>case3.sql</code>	<code>case6b.sql</code>
<code>case4a.sql</code>	<code>droptera.sql</code>
<code>case4b.sql</code>	-

Demonstration Requirements

The case studies assume these requirements have been met:

- The gateway demonstration tables are installed in the Teradata database.
- The Oracle database has an account named `SCOTT` with a password of `TIGER`.
- The Oracle database has a database link called `GTWLINK` (set up as public or private to the user `SCOTT`) which connects the gateway to a Teradata database as `SCOTT` with password `TIGER2`.

For example, you can create the database link as follows:

```
SQL> CREATE DATABASE LINK GTWLINK CONNECT TO SCOTT
2 IDENTIFIED BY TIGER2 USING 'GTWSID';
```

- Oracle Net Services is configured correctly and running.

Creating Demonstration Tables

The case studies are based on the `GTW_EMP`, `GTW_DEPT`, and `GTW_SALGRADE` tables. If the demonstration tables have not been created in the Teradata database, use the `bldtera.sql` script to create them.

The script creates the demonstration tables in the Teradata database accordingly:

```
CREATE TABLE GTW_EMP (
EMPNO      SMALLINT NOT NULL
ENAME      VARCHAR(10),
JOB        VARCHAR(9),
MGR        SMALLINT,
HIREDATE   DATETIME,
SAL        NUMERIC(7,2),
COMM       NUMERIC(7,2),
```

```

DEPTNO      SMALLINT)
CREATE TABLE GTW_DEPT (
DEPTNO      SMALLINT NOT NULL,
DNAME       VARCHAR(14),
LOC         VARCHAR(13))
CREATE TABLE GTW_SALGRADE (
GRADE       REAL,
LOSAL       NUMERIC(9,4),
HISAL       NUMERIC(9,4))

```

Demonstration Table Definitions

The table definitions are listed here using information retrieved by the SQL*PLUS DESCRIBE command:

GTW_EMP

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(5)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		NUMBER(5)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(5)

GTW_DEPT

Name	Null?	Type
DEPTNO	NOT NULL	NUMBER(5)
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

GTW_SALGRADE

Name	Null?	Type
GRADE		FLOAT(49)
LOSAL		NUMBER(9,4)
HISAL		NUMBER(9,4)

Demonstration Table Contents

The contents of the Teradata tables are:

GTW_EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30

7902	FORD	ANALYST	7566	03-DEC-81	3000	20
7934	MILLER	CLERK	7782	23-JAN-82	1300	10

GTW_DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

GTW_SALGRADE

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

Case 1: Simple Queries

Case 1 demonstrates the following:

- A simple query.
- A simple query retrieving full date information.

The first query retrieves all the data from `GTW_DEPT` and confirms that the gateway is working correctly. The second query retrieves all the data from `GTW_EMP` including the time portion of the hire date because the default date format was set to `DD-MON-YY HH24:MM:SS` for the session by an `ALTER SESSION` command.

Case 2: A More Complex Query

Case 2 demonstrates the following:

- The functions `SUM(expression)` and `NVL(expr1, expr2)` in the `SELECT` list.
- The `GROUP BY` and `HAVING` clauses.

This query retrieves the departments from `GTW_EMP` whose total monthly expenses are higher than \$10,000.

Case 3: Joining Teradata Tables

Case 3 demonstrates the following:

- Joins between Teradata tables.
- Subselects.

The query retrieves information from three Teradata tables and relates the employees to their department name and salary grade, but only for those employees earning more than the average salary.

Case 4: Write Capabilities

Case 4 is split into three cases and demonstrates the following:

- [DELETE Statement](#)

- [UPDATE Statement](#)
- [INSERT Statement](#)

DELETE Statement

Case 4a demonstrates bind values and subselect. All employees in department 20 and one employee, WARD, in department 30 are deleted.

UPDATE Statement

Case 4b provides an example of a simple UPDATE statement. In this example, employees are given a \$100 a month salary increase.

INSERT Statement

Case 4c is an example of a simple insert statement that does not provide information for all columns.

Case 5: Data Dictionary Query

Case 5 demonstrates data dictionary mapping. It retrieves all the tables and views that exist in the Teradata database that begin with "GTW".

Case 6: The Pass-Through Feature

Case 6 demonstrates the gateway pass-through feature which allows an application to send commands or statements to Teradata.

This case demonstrates:

- A pass-through UPDATE statement using bind variables.
- A pass-through SELECT statement.

UPDATE Statement

Case 6a provides an example of a pass-through UPDATE statement with bind variables. In this example, the salary for EMPNO 7934 is set to 4000.

SELECT Statement

Case 6b provides an example of a pass-through SELECT statement. The data that is returned from the SELECT statement is inserted into a local table at the Oracle database.

Data Type Conversion

This appendix contains the following section:

- [Data Type Conversion](#)

Data Type Conversion

The gateway converts Teradata data types to Oracle data types as follows:

Table A-1 Data Type Conversions

Teradata	Oracle	Comment
BLOB	LONG RAW	
BYTE	RAW	-
BYTEINT	NUMBER(3)	-
CHAR	CHAR	-
CHAR CHARACTER SET UNICO	CHAR	
CLOB	LONG	
DATE	DATE	-
DECIMAL	NUMBER(p[,s])	-
DECIMAL	NUMBER(19)	-
FLOAT	FLOAT(24)	with precision <= single precision float
FLOAT	FLOAT(53)	-
INTEGER	NUMBER(10)	NUMBER range is -2,147,483,647 to 2,147,483,647
INTERVAL YEAR	INTERVAL YEAR TO MONTH	-
INTERVAL MONTH	INTERVAL YEAR TO MONTH	-
INTERVAL YEAR TO MONTH	INTERVAL YEAR TO MONTH	-
INTERVAL DAY	INTERVAL DAY TO SECOND	-
INTERVAL HOUR	INTERVAL DAY TO SECOND	-

Table A-1 (Cont.) Data Type Conversions

Teradata	Oracle	Comment
INTERVAL MINUTE	INTERVAL DAY TO SECOND	-
INTERVAL SECOND	INTERVAL DAY TO SECOND	-
INTERVAL DAY TO HOUR	INTERVAL DAY TO SECOND	-
INTERVAL DAY TO MINUTE	INTERVAL DAY TO SECOND	-
INTERVAL DAY TO SECOND	INTERVAL DAY TO SECOND	-
INTERVAL HOUR TO MINUTE	INTERVAL DAY TO SECOND	-
INTERVAL HOUR TO SECOND	INTERVAL DAY TO SECOND	-
INTERVAL MINUTE TO SECOND	INTERVAL DAY TO SECOND	-
LONG VARCHAR CHARACTER SET UNICO	LONG	-
LONG VARCHAR	LONG	-
SMALLINT	NUMBER (5)	-
TIME	CHAR (15)	-
TIMESTAMP	TIMESTAMP	-
VARBYTE	RAW	with less or equal to 2000 bytes
VARBYTE	LONG RAW	with more than 2000 bytes
VARCHAR	VARCHAR2	If the size is greater than 4000, it's mapped to LONG
VARCHAR () CHARACTER SET UNICO	VARCHAR2	-

Supported SQL Syntax and Functions

This appendix contains the following sections:

- [Supported SQL Statements](#)
- [Oracle Functions](#)

Supported SQL Statements

With a few exceptions, the gateway provides full support for Oracle `DELETE`, `INSERT`, `SELECT`, and `UPDATE` statements.

The gateway does not support Oracle data definition language (DDL) statements. No form of the Oracle `ALTER`, `CREATE`, `DROP`, `GRANT`, or `TRUNCATE` statements can be used. Instead, use the pass-through feature of the gateway if you need to use DDL statements against the Teradata database.

See Also: *Oracle Database SQL Language Reference* for a detailed descriptions of keywords, parameters, and options.

DELETE

The `DELETE` statement is fully supported. However, only Oracle functions supported by Teradata can be used.

See Also: ["Functions Supported by Teradata"](#) on page B-2 for a list of supported functions.

INSERT

The `INSERT` statement is fully supported. However, only Oracle functions supported by Teradata can be used.

See Also: ["Functions Supported by Teradata"](#) on page B-2 for a list of supported functions.

SELECT

The `SELECT` statement is fully supported, with these exceptions:

- `CONNECT BY condition`
- `NOWAIT`
- `START WITH condition`
- `WHERE CURRENT OF`

UPDATE

The UPDATE statement is fully supported. However, only Oracle functions supported by Teradata can be used. Also, you cannot have SQL statements in the subquery that refer to the same table name in the outer query. Subqueries are not supported in the SET clause.

See Also: ["Functions Supported by Teradata"](#) on page B-2 for a list of supported functions.

Oracle Functions

All functions are evaluated by the Teradata database after the gateway has converted them to Teradata SQL.

Functions Not Supported by Teradata

Oracle SQL functions with no equivalent function in Teradata are not supported in DELETE, INSERT, or UPDATE statements, but are evaluated by the Oracle database if the statement is a SELECT statement. That is, the Oracle database performs post-processing of SELECT statements sent to the gateway.

If an unsupported function is used in a DELETE, INSERT, or UPDATE, statement, the following Oracle error occurs:

ORA-02070: database *db_link_name* does not support *function* in this context

Functions Supported by Teradata

The gateway translates the following Oracle database functions in SQL statements to their equivalent Teradata functions:

- [Arithmetic Operators](#)
- [Comparison Operators](#)
- [Group Functions](#)
- [String Functions](#)
- [Other Functions](#)

Arithmetic Operators

Oracle	Teradata
+	+
-	-
*	*
/	/

Comparison Operators

Oracle	Teradata
=	=
>	>

Oracle	Teradata
<	<
>=	>=
<=	<=
<>, !=, ^=	<>, !=
IS NOT NULL	IS NOT NULL
IS NULL	IS NULL

Group Functions

Oracle	Teradata
AVG	AVG
COUNT	COUNT
MAX	MAX
MIN	MIN
SUM	SUM

String Functions

Oracle	Teradata
	CONCAT
INSTR	LOCATE
LENGTH	LENGTH
SUBSTR (second argument cannot be a negative number)	SUBSTRING
UPPER	UCASE

Other Functions

Oracle	Teradata
ABS	ABS
EXP	EXP
LOG(a, b)	LOG10(b) / LOG10(a)
POWER	**
SQRT	SQRT

Data Dictionary

The Oracle Database Gateway for Teradata User's Guide translates a query that refers to an Oracle database data dictionary table into a query that retrieves the data from Teradata system catalog tables. You perform queries on data dictionary tables over the database link in the same way you query data dictionary tables in the Oracle database. The gateway data dictionary is similar to the Oracle database data dictionary in appearance and use.

This appendix contains the following sections:

- [Data Dictionary Support](#)
- [Data Dictionary Mapping](#)
- [Gateway Data Dictionary Descriptions](#)

Data Dictionary Support

The following paragraphs describe the Oracle Database Gateway for Teradata data dictionary support.

Teradata System Catalog Tables

Teradata data dictionary information is stored in the Teradata database as Teradata system catalog tables. The Teradata system catalog tables define the structure of a database. When you change data definitions, Teradata reads and modifies the Teradata system catalog tables to add information about the user tables.

Accessing the Gateway Data Dictionary

Accessing a gateway data dictionary table or view is identical to accessing a data dictionary in an Oracle database. You issue a SQL `SELECT` statement specifying a database link. The Oracle database data dictionary view and column names are used to access the gateway data dictionary in an Oracle database. Synonyms of supported views are also acceptable. For example, the following statement queries the data dictionary table `ALL_CATALOG` to retrieve all table names in the Teradata database:

```
SQL> SELECT * FROM "ALL_CATALOG"@TERA;
```

When a data dictionary access query is issued, the gateway:

1. Uses ODBC API to retrieve information from Teradata system catalog. Refer to ["Data Dictionary Mapping"](#) on page C-3 for details.
2. Sends the translated query to Teradata.

3. Might convert the retrieved Teradata data to give it the appearance of the Oracle database data dictionary table.
4. Passes the data dictionary information from the translated Teradata system catalog table to the Oracle database.

Note: The values returned when querying the gateway data dictionary might not be the same as the ones returned by the Oracle Enterprise Manager DESCRIBE command.

Direct Queries to Teradata Tables

Queries issued directly to individual Teradata system catalog tables are allowed but they return different results because the Teradata system catalog table column names differ from those of the data dictionary view. Also, certain columns in an Teradata system catalog table cannot be used in data dictionary processing.

Supported Views and Tables

The gateway supports the following views and tables:

Supported Views and Tables	Supported Views and Tables
ALL_CATALOG	ALL_COL_COMMENTS
ALL_CONS_COLUMNS	ALL_CONSTRAINTS
ALL_IND_COLUMNS	ALL_INDEXES
ALL_OBJECTS	ALL_TAB_COLUMNS
ALL_TAB_COMMENTS	ALL_TABLES
ALL_USERS	ALL_VIEWS
DBA_CATALOG	DBA_COL_COMMENTS
DBA_OBJECTS	DBA_TABLES
DBA_TAB_COLUMNS	DBA_TAB_COMMENTS
DICT_COLUMNS	DICTIONARY
DUAL	TABLE_PRIVILEGES
USER_CATALOG	USER_COL_COMMENTS
USER_CONS_COLUMNS	USER_CONSTRAINTS
USER_IND_COLUMNS	USER_INDEXES
USER_OBJECTS	USER_TAB_COLUMNS
USER_TAB_COMMENTS	USER_TABLES
USER_USER	USER_VIEWS

No other Oracle database data dictionary tables or views are supported. If you use a view not on the list, you receive the Oracle database error code for no more rows available.

Queries through the gateway of any data dictionary table or view beginning with ALL_ can returns rows from the Teradata database even when access privileges for those Teradata objects have not been granted. When querying an Oracle database with

the Oracle data dictionary, rows are returned only for those objects you are permitted to access.

Data Dictionary Mapping

The tables in this section list Oracle data dictionary view names and ODBC API's used.

Table C-1 Oracle Data Dictionary View Names and Teradata Equivalents

View Name	ODBC API Used
ALL_CATALOG	SQLTables
ALL_COL_COMMENTS	SQLColumns
ALL_CONS_COLUMNS	SQLPrimaryKeys, SQLForeignKeys
ALL_CONSTRAINTS	SQLPrimaryKeys, SQLForeignKeys
ALL_IND_COLUMNS	SQLStatistics
ALL_INDEXES	SQLStatistics
ALL_OBJECTS	SQLTables, SQLProcedures, SQLStatistics
ALL_TAB_COLUMNS	SQLColumns
ALL_TAB_COMMENTS	SQLTables
ALL_TABLES	SQLStatistics
ALL_USERS	SQLTables
ALL_VIEWS	SQLTables
DBA_CATALOG	SQLTables
DBA_COL_COMMENTS	SQLColumns
DBA_OBJECTS	SQLTables, SQLProcedures, SQLStatistics
DBA_TAB_COLUMNS	SQLColumns
DBA_TAB_COMMENTS	SQLTables
DBA_TABLES	SQLStatistics
DICT_COLUMNS	SQLColumns
DICTIONARY	SQLTables
DUAL	<i>(Defined in the Gateway)</i>
USER_CATALOG	SQLTables
USER_COL_COMMENTS	SQLColumns
USER_CONS_COLUMNS	SQLPrimaryKeys, SQLForeignKeys
USER_CONSTRAINTS	SQLPrimaryKeys, SQLForeignKeys
USER_IND_COLUMNS	SQLStatistics
USER_INDEXES	SQLStatistics
USER_OBJECTS	SQLTables, SQLProcedures, SQLStatistics
USER_TAB_COLUMNS	SQLColumns
USER_TAB_COMMENTS	SQLTables
USER_TABLES	SQLStatistics

Table C–1 (Cont.) Oracle Data Dictionary View Names and Teradata Equivalents

View Name	ODBC API Used
USER_USERS	SQLTables
USER_VIEWS	SQLTables

Default Column Values

There is a minor difference between the gateway data dictionary and a typical Oracle database data dictionary. The Oracle database columns that are missing in an Teradata system catalog table are filled with zeros, spaces, null values, not-applicable values (N.A.), or default values, depending on the column type.

Gateway Data Dictionary Descriptions

The gateway data dictionary tables and views provide the following information:

- Name, data type, and width of each column
- The contents of columns with fixed values

They are described here with information retrieved by an SQL*PLUS DESCRIBE command. The values in the Null? column might differ from the Oracle database data dictionary tables and views. Any hardcoded default value is shown to the right of an item, but this is not information returned by DESCRIBE.

Table C–2 ALL_CATALOG

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2 (30)	-
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
TABLE_TYPE	-	VARCHAR2 (11)	"TABLE" or "VIEW"

Table C–3 ALL_COL_COMMENTS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2 (30)	-
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
COLUMN_NAME	NOT NULL	VARCHAR2 (30)	-
COMMENTS	NOT NULL	CHAR (1)	" "

Table C–4 ALL_CONS_COLUMNS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2 (30)	-
CONSTRAINT_NAME	NOT NULL	VARCHAR2 (30)	-
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
COLUMN_NAME	-	VARCHAR2 (8192)	-
POSITION	-	FLOAT (49)	-

Table C-5 ALL_CONSTRAINTS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2 (30)	-
CONSTRAINT_NAME	NOT NULL	VARCHAR2 (30)	-
CONSTRAINT_TYPE	-	VARCHAR2 (1)	"R" or "P"
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
SEARCH_CONDITION	NOT NULL	CHAR (1)	" "
R_OWNER	NOT NULL	VARCHAR2 (30)	" "
R_CONSTRAINT_NAME	NOT NULL	VARCHAR2 (30)	" "
DELETE_RULE	NOT NULL	VARCHAR2 (9)	" "
STATUS	NOT NULL	CHAR (1)	" "
DEFERRABLE	NOT NULL	CHAR (1)	" "
DEFERRED	NOT NULL	CHAR (1)	" "
VALIDATED	NOT NULL	CHAR (1)	" "
GENERATED	NOT NULL	CHAR (1)	" "
BAD	NOT NULL	CHAR (1)	" "
RELY	NOT NULL	CHAR (1)	" "
LAST_CHANGE	-	DATE	NULL

Table C-6 ALL_IND_COLUMNS

Name	Null?	Type	Value
INDEX_OWNER	NOT NULL	VARCHAR2 (30)	-
INDEX_NAME	NOT NULL	VARCHAR2 (30)	-
TABLE_OWNER	NOT NULL	VARCHAR2 (30)	-
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
COLUMN_NAME	-	VARCHAR2 (8192)	-
COLUMN_POSITION	NOT NULL	FLOAT (49)	-
COLUMN_LENGTH	NOT NULL	FLOAT (49)	-
DESCEND	-	VARCHAR2 (4)	"DESC" or "ASC"

Table C-7 ALL_INDEXES

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2 (30)	-
INDEX_NAME	NOT NULL	CHAR (1)	-
INDEX_TYPE	NOT NULL	CHAR (1)	" "
TABLE_OWNER	NOT NULL	VARCHAR2 (30)	-
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
TABLE_TYPE	NOT NULL	CHAR (5)	"TABLE"

Table C-7 (Cont.) ALL_INDEXES

Name	Null?	Type	Value
UNIQUENESS	-	VARCHAR2 (9)	"UNIQUE" or "NONUNIQUE"
COMPRESSION	NOT NULL	CHAR (1)	" "
PREFIX_LENGTH	NOT NULL	NUMBER (10)	0
TABLESPACE_NAME	NOT NULL	CHAR (1)	" "
INI_TRANS	NOT NULL	NUMBER (10)	0
MAX_TRANS	NOT NULL	NUMBER (10)	0
INITIAL_EXTENT	NOT NULL	NUMBER (10)	0
NEXT_EXTENT	NOT NULL	NUMBER (10)	0
MIN_EXTENTS	NOT NULL	NUMBER (10)	0
MAX_EXTENTS	NOT NULL	NUMBER (10)	0
PCT_INCREASE	NOT NULL	NUMBER (10)	0
PCT_THRESHOLD	NOT NULL	NUMBER (10)	0
INCLUDE_COLUMNS	NOT NULL	NUMBER (10)	0
FREELISTS	NOT NULL	NUMBER (10)	0
FREELIST_GROUPS	NOT NULL	NUMBER (10)	0
PCT_FREE	NOT NULL	NUMBER (10)	0
LOGGING	NOT NULL	CHAR (1)	" "
BLEVEL	NOT NULL	NUMBER (10)	0
LEAF_BLOCKS	NOT NULL	NUMBER (10)	0
DISTINCT_KEYS	-	FLOAT (49)	-
AVG_LEAF_BLOCKS_PER_KEY	NOT NULL	NUMBER (10)	0
AVG_DATA_BLOCKS_PER_KEY	NOT NULL	NUMBER (10)	0
CLUSTERING_FACTOR	NOT NULL	NUMBER (10)	0
STATUS	NOT NULL	CHAR (1)	" "
NUM_ROWS	NOT NULL	NUMBER (10)	0
SAMPLE_SIZE	NOT NULL	NUMBER (10)	0
LAST_ANALYZED	-	DATE	NULL
DEGREE	NOT NULL	CHAR (1)	" "
INSTANCES	NOT NULL	CHAR (1)	" "
PARTITIONED	NOT NULL	CHAR (1)	" "
TEMPORARY	NOT NULL	CHAR (1)	" "
GENERATED	NOT NULL	CHAR (1)	" "
SECONDARY	NOT NULL	CHAR (1)	" "
BUFFER_POOL	NOT NULL	CHAR (1)	" "
USER_STATS	NOT NULL	CHAR (1)	" "
DURATION	NOT NULL	CHAR (1)	" "

Table C-7 (Cont.) ALL_INDEXES

Name	Null?	Type	Value
PCT_DIRECT_ACCESS	NOT NULL	NUMBER (10)	0
ITYP_OWNER	NOT NULL	CHAR (1)	" "
ITYP_NAME	NOT NULL	CHAR (1)	" "
PARAMETERS	NOT NULL	CHAR (1)	" "
GLOBAL_STATS	NOT NULL	CHAR (1)	" "
DOMIDX_STATUS	NOT NULL	CHAR (1)	" "
DOMIDX_OPSTATUS	NOT NULL	CHAR (1)	" "
FUNCIDX_STATUS	NOT NULL	CHAR (1)	" "

Table C-8 ALL_OBJECTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2 (30)	-
OBJECT_NAME	-	VARCHAR2 (30)	-
SUBOBJECT_NAME	-	VARCHAR2 (1)	NULL
OBJECT_ID	-	NUMBER	0
DATA_OBJECT_ID	-	NUMBER	0
OBJECT_TYPE	-	VARCHAR2 (18)	"TABLE" or "VIEW" or "INDEX" or "PROCEDURE"
CREATED	-	DATE	NULL
LAST_DDL_TIME	-	DATE	NULL
TIMESTAMP	-	VARCHAR2 (1)	NULL
STATUS	-	VARCHAR2 (1)	NULL
TEMPORARY	-	VARCHAR2 (1)	NULL
GENERATED	-	VARCHAR2 (1)	NULL
SECONDARY	-	VARCHAR2 (1)	NULL

Table C-9 ALL_TAB_COLUMNS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2 (30)	-
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
COLUMN_NAME	NOT NULL	VARCHAR2 (30)	-
DATA_TYPE	-	VARCHAR2 (106)	-
DATA_TYPE_MOD	NOT NULL	CHAR (1)	" "
DATA_TYPE_OWNER	NOT NULL	CHAR (1)	" "
DATA_LENGTH	NOT NULL	FLOAT (49)	-
DATA_PRECISION	-	FLOAT (49)	-

Table C-9 (Cont.) ALL_TAB_COLUMNS

Name	Null?	Type	Value
DATA_SCALE	-	FLOAT (49)	-
NULLABLE	-	VARCHAR2 (1)	"Y" or "N"
COLUMN_ID	NOT NULL	FLOAT (49)	-
DEFAULT_LENGTH	NOT NULL	NUMBER (10)	0
DATA_DEFAULT	NOT NULL	CHAR (1)	" "
NUM_DISTINCT	NOT NULL	NUMBER (10)	0
LOW_VALUE	NOT NULL	NUMBER (10)	0
HIGH_VALUE	NOT NULL	NUMBER (10)	0
DENSITY	NOT NULL	NUMBER (10)	0
NUM_NULLS	NOT NULL	NUMBER (10)	0
NUM_BUCKETS	NOT NULL	NUMBER (10)	0
LAST_ANALYZED	-	DATE	NULL
SAMPLE_SIZE	NOT NULL	NUMBER (10)	0
CHARACTER_SET_NAME	NOT NULL	CHAR (1)	" "
CHAR_COL_DECL_LENGTH	NOT NULL	NUMBER (10)	0
GLOBAL_STATS	NOT NULL	CHAR (1)	" "
USER_STATS	NOT NULL	CHAR (1)	" "
AVG_COL_LEN	NOT NULL	NUMBER (10)	0

Table C-10 ALL_TAB_COMMENTS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2 (30)	-
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
TABLE_TYPE	-	VARCHAR2 (11)	"TABLE" or "VIEW"
COMMENTS	NOT NULL	CHAR (1)	" "

Table C-11 ALL_TABLES

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2 (30)	-
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
TABLESPACE_NAME	NOT NULL	CHAR (1)	" "
CLUSTER_NAME	NOT NULL	CHAR (1)	" "
IOT_NAME	NOT NULL	CHAR (1)	" "
PCT_FREE	NOT NULL	NUMBER (10)	0
PCT_USED	NOT NULL	NUMBER (10)	0
INI_TRANS	NOT NULL	NUMBER (10)	0

Table C-11 (Cont.) ALL_TABLES

Name	Null?	Type	Value
MAX_TRANS	NOT NULL	NUMBER(10)	0
INITIAL_EXTENT	NOT NULL	NUMBER(10)	0
NEXT_EXTENT	NOT NULL	NUMBER(10)	0
MIN_EXTENTS	NOT NULL	NUMBER(10)	0
MAX_EXTENTS	NOT NULL	NUMBER(10)	0
PCT_INCREASE	NOT NULL	NUMBER(10)	0
FREELISTS	NOT NULL	NUMBER(10)	0
FREELIST_GROUPS	NOT NULL	NUMBER(10)	0
LOGGING	NOT NULL	CHAR(1)	" "
BACKED_UP	NOT NULL	CHAR(1)	" "
NUM_ROWS	-	FLOAT(49)	-
BLOCKS	-	FLOAT(49)	-
EMPTY_BLOCKS	NOT NULL	NUMBER(10)	0
AVG_SPACE	NOT NULL	NUMBER(10)	0
CHAIN_CNT	NOT NULL	NUMBER(10)	0
AVG_ROW_LEN	NOT NULL	NUMBER(10)	0
AVG_SPACE_FREELIST_BLOCKS	NOT NULL	NUMBER(10)	0
NUM_FREELIST_BLOCKS	NOT NULL	NUMBER(10)	0
DEGREE	NOT NULL	CHAR(1)	" "
INSTANCES	NOT NULL	CHAR(1)	" "
CACHE	NOT NULL	CHAR(1)	" "
TABLE_LOCK	NOT NULL	CHAR(1)	" "
SAMPLE_SIZE	NOT NULL	NUMBER(10)	0
LAST_ANALYZED	-	DATE	NULL
PARTITIONED	NOT NULL	CHAR(1)	" "
IOT_TYPE	NOT NULL	CHAR(1)	" "
TEMPORARY	NOT NULL	CHAR(1)	" "
SECONDARY	NOT NULL	CHAR(1)	" "
NESTED	NOT NULL	CHAR(1)	" "
BUFFER_POOL	NOT NULL	CHAR(1)	" "
ROW_MOVEMENT	NOT NULL	CHAR(1)	" "
GLOBAL_STATS	NOT NULL	CHAR(1)	" "
USER_STATS	NOT NULL	CHAR(1)	" "
DURATION	NOT NULL	CHAR(1)	" "
SKIP_CORRUPT	NOT NULL	CHAR(1)	" "
MONITORING	NOT NULL	CHAR(1)	" "

Table C-12 ALL_USERS

Name	Null?	Type	Value
USERNAME	NOT NULL	VARCHAR2 (30)	-
USER_ID	-	NUMBER	0
CREATED	-	DATE	NULL

Table C-13 ALL_VIEWS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2 (30)	-
VIEW_NAME	NOT NULL	VARCHAR2 (30)	-
TEXT_LENGTH	NOT NULL	NUMBER (10)	0
TEXT	NOT NULL	CHAR (1)	" "
TYPE_TEXT_LENGTH	NOT NULL	NUMBER (10)	0
TYPE_TEXT	NOT NULL	CHAR (1)	" "
OID_TEXT_LENGTH	NOT NULL	NUMBER (10)	0
OID_TEXT	NOT NULL	CHAR (1)	" "
VIEW_TYPE_OWNER	NOT NULL	CHAR (1)	" "
VIEW_TYPE	NOT NULL	CHAR (1)	" "

Table C-14 DBA_CATALOG

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2 (30)	-
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
TABLE_TYPE	-	VARCHAR2 (11)	"TABLE" or "VIEW"

Table C-15 DBA_COL_COMMENTS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2 (30)	-
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
COLUMN_NAME	NOT NULL	VARCHAR2 (30)	-
COMMENTS	NOT NULL	CHAR (1)	" "

Table C-16 DBA_OBJECTS

Name	Null?	Type	Value
OWNER	-	VARCHAR2 (30)	-
OBJECT_NAME	-	VARCHAR2 (128)	-
SUBOBJECT_NAME	-	VARCHAR2 (1)	NULL
OBJECT_ID	-	NUMBER	0
DATA_OBJECT_ID	-	NUMBER	0

Table C-16 (Cont.) DBA_OBJECTS

Name	Null?	Type	Value
OBJECT_TYPE	-	VARCHAR2 (18)	"TABLE" or "VIEW" or "INDEX" or "PROCEDURE"
CREATED	-	DATE	NULL
LAST_DDL_TIME	-	DATE	NULL
TIMESTAMP	-	VARCHAR2 (1)	NULL
STATUS	-	VARCHAR2 (1)	NULL
TEMPORARY	-	VARCHAR2 (1)	NULL
GENERATED	-	VARCHAR2 (1)	NULL
SECONDARY	-	VARCHAR2 (1)	NULL

Table C-17 DBA_TAB_COLUMNS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2 (30)	-
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
COLUMN_NAME	NOT NULL	VARCHAR2 (30)	-
DATA_TYPE	-	VARCHAR2 (106)	-
DATA_TYPE_MOD	NOT NULL	CHAR (1)	" "
DATA_TYPE_OWNER	NOT NULL	CHAR (1)	" "
DATA_LENGTH	NOT NULL	FLOAT (49)	-
DATA_PRECISION	-	FLOAT (49)	-
DATA_SCALE	-	FLOAT (49)	-
NULLABLE	-	VARCHAR2 (1)	"Y" or "N"
COLUMN_ID	NOT NULL	FLOAT (49)	-
DEFAULT_LENGTH	NOT NULL	NUMBER (10)	0
DATA_DEFAULT	NOT NULL	CHAR (1)	" "
NUM_DISTINCT	NOT NULL	NUMBER (10)	0
LOW_VALUE	NOT NULL	NUMBER (10)	0
HIGH_VALUE	NOT NULL	NUMBER (10)	0
DENSITY	NOT NULL	NUMBER (10)	0
NUM_NULLS	NOT NULL	NUMBER (10)	0
NUM_BUCKETS	NOT NULL	NUMBER (10)	0
LAST_ANALYZED	-	DATE	NULL
SAMPLE_SIZE	NOT NULL	NUMBER (10)	0
CHARACTER_SET_NAME	NOT NULL	CHAR (1)	" "
CHAR_COL_DECL_LENGTH	NOT NULL	NUMBER (10)	0
GLOBAL_STATS	NOT NULL	CHAR (1)	" "

Table C–17 (Cont.) DBA_TAB_COLUMNS

Name	Null?	Type	Value
USER_STATS	NOT NULL	CHAR(1)	" "
AVG_COL_LEN	NOT NULL	NUMBER(10)	0

Table C–18 DBA_TAB_COMMENTS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	-
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
TABLE_TYPE	-	VARCHAR2(11)	"TABLE" or "VIEW"
COMMENTS	NOT NULL	CHAR(1)	" "

Table C–19 DBA_TABLES

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2(30)	-
TABLE_NAME	NOT NULL	VARCHAR2(30)	-
TABLESPACE_NAME	NOT NULL	CHAR(1)	" "
CLUSTER_NAME	NOT NULL	CHAR(1)	" "
IOT_NAME	NOT NULL	CHAR(1)	" "
PCT_FREE	NOT NULL	NUMBER(10)	0
PCT_USED	NOT NULL	NUMBER(10)	0
INI_TRANS	NOT NULL	NUMBER(10)	0
MAX_TRANS	NOT NULL	NUMBER(10)	0
INITIAL_EXTENT	NOT NULL	NUMBER(10)	0
NEXT_EXTENT	NOT NULL	NUMBER(10)	0
MIN_EXTENTS	NOT NULL	NUMBER(10)	0
MAX_EXTENTS	NOT NULL	NUMBER(10)	0
PCT_INCREASE	NOT NULL	NUMBER(10)	0
FREELISTS	NOT NULL	NUMBER(10)	0
FREELIST_GROUPS	NOT NULL	NUMBER(10)	0
LOGGING	NOT NULL	CHAR(1)	" "
BACKED_UP	NOT NULL	CHAR(1)	" "
NUM_ROWS	-	FLOAT(49)	-
BLOCKS	-	FLOAT(49)	-
EMPTY_BLOCKS	NOT NULL	NUMBER(10)	0
AVG_SPACE	NOT NULL	NUMBER(10)	0
CHAIN_CNT	NOT NULL	NUMBER(10)	0
AVG_ROW_LEN	NOT NULL	NUMBER(10)	0

Table C-19 (Cont.) DBA_TABLES

Name	Null?	Type	Value
AVG_SPACE_FREELIST_BLOCKS	NOT NULL	NUMBER (10)	0
NUM_FREELIST_BLOCKS	NOT NULL	NUMBER (10)	0
DEGREE	NOT NULL	CHAR (1)	" "
INSTANCES	NOT NULL	CHAR (1)	" "
CACHE	NOT NULL	CHAR (1)	" "
TABLE_LOCK	NOT NULL	CHAR (1)	" "
SAMPLE_SIZE	NOT NULL	NUMBER (10)	0
LAST_ANALYZED	-	DATE	NULL
PARTITIONED	NOT NULL	CHAR (1)	" "
IOT_TYPE	NOT NULL	CHAR (1)	" "
TEMPORARY	NOT NULL	CHAR (1)	" "
SECONDARY	NOT NULL	CHAR (1)	" "
NESTED	NOT NULL	CHAR (1)	" "
BUFFER_POOL	NOT NULL	CHAR (1)	" "
ROW_MOVEMENT	NOT NULL	CHAR (1)	" "
GLOBAL_STATS	NOT NULL	CHAR (1)	" "
USER_STATS	NOT NULL	CHAR (1)	" "
DURATION	NOT NULL	CHAR (1)	" "
SKIP_CORRUPT	NOT NULL	CHAR (1)	" "
MONITORING	NOT NULL	CHAR (1)	" "

Table C-20 DICT_COLUMNS

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
COLUMN_NAME	NOT NULL	VARCHAR2 (30)	-
COMMENTS	NOT NULL	CHAR (1)	" "

Table C-21 DICTIONARY

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
COMMENTS	NOT NULL	CHAR (1)	" "

Table C-22 DUAL

Name	Null?	Type	Value
DUMMY	NOT NULL	VARCHAR2 (1)	"X"

Table C-23 USER_CATALOG

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
TABLE_TYPE	-	VARCHAR2 (11)	"TABLE" or "VIEW"

Table C-24 USER_COL_COMMENTS

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
COLUMN_NAME	NOT NULL	VARCHAR2 (30)	-
COMMENTS	NOT NULL	CHAR (1)	" "

Table C-25 USER_CONS_COLUMNS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2 (30)	-
CONSTRAINT_NAME	NOT NULL	VARCHAR2 (30)	-
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
COLUMN_NAME	-	VARCHAR2 (8192)	-
POSITION	-	FLOAT (49)	-

Table C-26 USER_CONSTRAINTS

Name	Null?	Type	Value
OWNER	NOT NULL	VARCHAR2 (30)	-
CONSTRAINT_NAME	NOT NULL	VARCHAR2 (30)	-
CONSTRAINT_TYPE	-	VARCHAR2 (1)	"R" or "P"
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
SEARCH_CONDITION	NOT NULL	CHAR (1)	" "
R_OWNER	NOT NULL	VARCHAR2 (30)	-
R_CONSTRAINT_NAME	NOT NULL	VARCHAR2 (30)	-
DELETE_RULE	NOT NULL	VARCHAR2 (9)	" "
STATUS	NOT NULL	CHAR (1)	" "
DEFERRABLE	NOT NULL	CHAR (1)	" "
DEFERRED	NOT NULL	CHAR (1)	" "
VALIDATED	NOT NULL	CHAR (1)	" "
GENERATED	NOT NULL	CHAR (1)	" "
BAD	NOT NULL	CHAR (1)	" "
RELY	NOT NULL	CHAR (1)	" "
LAST_CHANGE	-	DATE	NULL

Table C-27 USER_IND_COLUMNS

Name	Null?	Type	Value
INDEX_NAME	NOT NULL	VARCHAR2 (30)	-
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
COLUMN_NAME	NOT NULL	VARCHAR2 (8192)	-
COLUMN_POSITION	NOT NULL	FLOAT (49)	-
COLUMN_LENGTH	NOT NULL	FLOAT (49)	-
DESCEND	NOT NULL	VARCHAR2 (4)	"DESC" or "ASC"

Table C-28 USER_INDEXES

Name	Null?	Type	Value
INDEX_NAME	NOT NULL	VARCHAR2 (30)	-
INDEX_TYPE	NOT NULL	CHAR (1)	" "
TABLE_OWNER	NOT NULL	VARCHAR2 (30)	-
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
TABLE_TYPE	NOT NULL	CHAR (5)	"TABLE"
UNIQUENESS	-	VARCHAR2 (9)	"UNIQUE" or "NONUNIQUE"
COMPRESSION	NOT NULL	CHAR (1)	" "
PREFIX_LENGTH	NOT NULL	NUMBER (10)	0
TABLESPACE_NAME	NOT NULL	CHAR (1)	" "
INI_TRANS	NOT NULL	NUMBER (10)	0
MAX_TRANS	NOT NULL	NUMBER (10)	0
INITIAL_EXTENT	NOT NULL	NUMBER (10)	0
NEXT_EXTENT	NOT NULL	NUMBER (10)	0
MIN_EXTENTS	NOT NULL	NUMBER (10)	0
MAX_EXTENTS	NOT NULL	NUMBER (10)	0
PCT_INCREASE	NOT NULL	NUMBER (10)	0
PCT_THRESHOLD	NOT NULL	NUMBER (10)	0
INCLUDE_COLUMNS	NOT NULL	NUMBER (10)	0
FREELISTS	NOT NULL	NUMBER (10)	0
FREELIST_GROUPS	NOT NULL	NUMBER (10)	0
PCT_FREE	NOT NULL	NUMBER (10)	0
LOGGING	NOT NULL	CHAR (1)	" "
BLEVEL	NOT NULL	NUMBER (10)	0
LEAF_BLOCKS	NOT NULL	NUMBER (10)	0
DISTINCT_KEYS	-	FLOAT (49)	-
AVG_LEAF_BLOCKS_PER_KEY	NOT NULL	NUMBER (10)	0
AVG_DATA_BLOCKS_PER_KEY	NOT NULL	NUMBER (10)	0

Table C-28 (Cont.) USER_INDEXES

Name	Null?	Type	Value
CLUSTERING_FACTOR	NOT NULL	NUMBER (10)	0
STATUS	NOT NULL	CHAR (1)	" "
NUM_ROWS	NOT NULL	NUMBER (10)	0
SAMPLE_SIZE	NOT NULL	NUMBER (10)	0
LAST_ANALYZED	-	DATE	NULL
DEGREE	NOT NULL	CHAR (1)	" "
INSTANCES	NOT NULL	CHAR (1)	" "
PARTITIONED	NOT NULL	CHAR (1)	" "
TEMPORARY	NOT NULL	CHAR (1)	" "
GENERATED	NOT NULL	CHAR (1)	" "
SECONDARY	NOT NULL	CHAR (1)	" "
BUFFER_POOL	NOT NULL	CHAR (1)	" "
USER_STATS	NOT NULL	CHAR (1)	" "
DURATION	NOT NULL	CHAR (1)	" "
PCT_DIRECT_ACCESS	NOT NULL	NUMBER (10)	0
ITYP_OWNER	NOT NULL	CHAR (1)	" "
ITYP_NAME	NOT NULL	CHAR (1)	" "
PARAMETERS	NOT NULL	CHAR (1)	" "
GLOBAL_STATS	NOT NULL	CHAR (1)	" "
DOMIDX_STATUS	NOT NULL	CHAR (1)	" "
DOMIDX_OPSTATUS	NOT NULL	CHAR (1)	" "
FUNCIDX_STATUS	NOT NULL	CHAR (1)	" "

Table C-29 USER_OBJECTS

Name	Null?	Type	Value
OBJECT_NAME	-	VARCHAR2 (128)	-
SUBOBJECT_NAME	-	VARCHAR2 (1)	NULL
OBJECT_ID	-	NUMBER	0
DATA_OBJECT_ID	-	NUMBER	0
OBJECT_TYPE	-	VARCHAR2 (18)	"TABLE" or "VIEW" or "INDEX" or "PROCEDURE"
CREATED	-	DATE	NULL
LAST_DDL_TIME	-	DATE	NULL
TIMESTAMP	-	VARCHAR2 (1)	NULL
STATUS	-	VARCHAR2 (1)	NULL
TEMPORARY	-	VARCHAR2 (1)	NULL

Table C-29 (Cont.) USER_OBJECTS

Name	Null?	Type	Value
GENERATED	-	VARCHAR2 (1)	NULL
SECONDARY	-	VARCHAR2 (1)	NULL

Table C-30 USER_TAB_COLUMNS

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
COLUMN_NAME	NOT NULL	VARCHAR2 (30)	-
DATA_TYPE	-	VARCHAR2 (106)	-
DATA_TYPE_MOD	NOT NULL	CHAR (1)	" "
DATA_TYPE_OWNER	NOT NULL	CHAR (1)	" "
DATA_LENGTH	NOT NULL	FLOAT (49)	-
DATA_PRECISION	-	FLOAT (49)	-
DATA_SCALE	-	FLOAT (49)	-
NULLABLE	-	VARCHAR2 (1)	"Y" or "N"
COLUMN_ID	NOT NULL	FLOAT (49)	-
DEFAULT_LENGTH	NOT NULL	NUMBER (10)	0
DATA_DEFAULT	NOT NULL	CHAR (1)	" "
NUM_DISTINCT	NOT NULL	NUMBER (10)	0
LOW_VALUE	NOT NULL	NUMBER (10)	0
HIGH_VALUE	NOT NULL	NUMBER (10)	0
DENSITY	NOT NULL	NUMBER (10)	0
NUM_NULLS	NOT NULL	NUMBER (10)	0
NUM_BUCKETS	NOT NULL	NUMBER (10)	0
LAST_ANALYZED	-	DATE	NULL
SAMPLE_SIZE	NOT NULL	NUMBER (10)	0
CHARACTER_SET_NAME	NOT NULL	CHAR (1)	" "
CHAR_COL_DECL_LENGTH	NOT NULL	NUMBER (10)	0
GLOBAL_STATS	NOT NULL	CHAR (1)	" "
USER_STATS	NOT NULL	CHAR (1)	" "
AVG_COL_LEN	NOT NULL	NUMBER (10)	0

Table C-31 USER_TAB_COMMENTS

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
TABLE_TYPE	-	VARCHAR2 (11)	"TABLE" or "VIEW"
COMMENTS	NOT NULL	CHAR (1)	" "

Table C-32 USER_TABLES

Name	Null?	Type	Value
TABLE_NAME	NOT NULL	VARCHAR2 (30)	-
TABLESPACE_NAME	NOT NULL	CHAR (1)	" "
CLUSTER_NAME	NOT NULL	CHAR (1)	" "
IOT_NAME	NOT NULL	CHAR (1)	" "
PCT_FREE	NOT NULL	NUMBER (10)	0
PCT_USED	NOT NULL	NUMBER (10)	0
INI_TRANS	NOT NULL	NUMBER (10)	0
MAX_TRANS	NOT NULL	NUMBER (10)	0
INITIAL_EXTENT	NOT NULL	NUMBER (10)	0
NEXT_EXTENT	NOT NULL	NUMBER (10)	0
MIN_EXTENTS	NOT NULL	NUMBER (10)	0
MAX_EXTENTS	NOT NULL	NUMBER (10)	0
PCT_INCREASE	NOT NULL	NUMBER (10)	0
FREELISTS	NOT NULL	NUMBER (10)	0
FREELIST_GROUPS	NOT NULL	NUMBER (10)	0
LOGGING	NOT NULL	CHAR (1)	" "
BACKED_UP	NOT NULL	CHAR (1)	" "
NUM_ROWS	-	FLOAT (49)	-
BLOCKS	NOT NULL	FLOAT (49)	-
EMPTY_BLOCKS	NOT NULL	NUMBER (10)	0
AVG_SPACE	NOT NULL	NUMBER (10)	0
CHAIN_CNT	NOT NULL	NUMBER (10)	0
AVG_ROW_LEN	NOT NULL	NUMBER (10)	0
AVG_SPACE_FREELIST_BLOCKS	NOT NULL	NUMBER (10)	0
NUM_FREELIST_BLOCKS	NOT NULL	NUMBER (10)	0
DEGREE	NOT NULL	CHAR (1)	" "
INSTANCES	NOT NULL	CHAR (1)	" "
CACHE	NOT NULL	CHAR (1)	" "
TABLE_LOCK	NOT NULL	CHAR (1)	" "
SAMPLE_SIZE	NOT NULL	NUMBER (10)	0
LAST_ANALYZED	-	DATE	NULL
PARTITIONED	NOT NULL	CHAR (1)	" "
IOT_TYPE	NOT NULL	CHAR (1)	" "
TEMPORARY	NOT NULL	CHAR (1)	" "
SECONDARY	NOT NULL	CHAR (1)	" "
NESTED	NOT NULL	CHAR (1)	" "
BUFFER_POOL	NOT NULL	CHAR (1)	" "

Table C-32 (Cont.) USER_TABLES

Name	Null?	Type	Value
ROW_MOVEMENT	NOT NULL	CHAR(1)	" "
GLOBAL_STATS	NOT NULL	CHAR(1)	" "
USER_STATS	NOT NULL	CHAR(1)	" "
DURATION	NOT NULL	CHAR(1)	" "
SKIP_CORRUPT	NOT NULL	CHAR(1)	" "
MONITORING	NOT NULL	CHAR(1)	" "

Table C-33 USER_USERS

Name	Null?	Type	Value
USERNAME	NOT NULL	VARCHAR2(30)	-
USER_ID	-	NUMBER	0
ACCOUNT_STATUS	-	VARCHAR2(4)	"OPEN"
LOCK_DATE	-	DATE	NULL
EXPIRY_DATE	-	DATE	NULL
DEFAULT_TABLESPACE	-	VARCHAR2(1)	NULL
TEMPORARY_TABLESPACE	-	VARCHAR2(1)	NULL
CREATED	-	DATE	NULL
INITIAL_RSRC_CONSUMER_GROUP	-	VARCHAR2(1)	NULL
EXTERNAL_NAME	-	VARCHAR2(1)	NULL

Table C-34 USER_VIEWS

Name	Null?	Type	Value
VIEW_NAME	NOT NULL	VARCHAR2(30)	-
TEXT_LENGTH	NOT NULL	NUMBER(10)	0
TEXT	NOT NULL	CHAR(1)	" "
TYPE_TEXT_LENGTH	NOT NULL	NUMBER(10)	0
TYPE_TEXT	NOT NULL	CHAR(1)	" "
OID_TEXT_LENGTH	NOT NULL	NUMBER(10)	0
OID_TEXT	NOT NULL	CHAR(1)	" "
VIEW_TYPE_OWNER	NOT NULL	CHAR(1)	" "
VIEW_TYPE	NOT NULL	CHAR(1)	" "

Initialization Parameters

The Oracle database initialization parameters in the `init.ora` file are distinct from gateway initialization parameters. Set the gateway parameters in the initialization parameter file using an agent-specific mechanism, or set them in the Oracle data dictionary using the `DBMS_HS` package. The gateway initialization parameter file must be available when the gateway is started.

This appendix contains a list of the gateway initialization parameters that can be set for each gateway and their description. It also describes the initialization parameter file syntax. It includes the following sections:

- [Initialization Parameter File Syntax](#)
- [Oracle Database Gateway for Teradata Initialization Parameters](#)
- [Initialization Parameter Descriptions](#)

Initialization Parameter File Syntax

The syntax for the initialization parameter file is as follows:

1. The file is a sequence of commands.
2. Each command should start on a separate line.
3. End of line is considered a command terminator (unless escaped with a backslash).
4. If there is a syntax error in an initialization parameter file, none of the settings take effect.
5. Set the parameter values as follows:

```
[SET] [PRIVATE] parameter=value
```

Where:

parameter is an initialization parameter name. It is a string of characters starting with a letter and consisting of letters, digits and underscores. Initialization parameter names are case sensitive.

value is the initialization parameter value. It is case-sensitive. An initialization parameter value is either:

- a. A string of characters that does not contain any backslashes, white space or double quotation marks ("")
- b. A quoted string beginning with a double quotation mark and ending with a double quotation mark. The following can be used inside a quoted string:

- * backslash (\) is the escape character
- * \n inserts a new line
- * \t inserts a tab
- * \" inserts a double quotation mark
- * \\ inserts a backslash

A backslash at the end of the line continues the string on the next line. If a backslash precedes any other character then the backslash is ignored.

For example, to enable tracing for an agent, set the `HS_FDS_TRACE_LEVEL` initialization parameter as follows:

```
HS_FDS_TRACE_LEVEL=ON
```

`SET` and `PRIVATE` are optional keywords. You cannot use either as an initialization parameter name. Most parameters are needed only as initialization parameters, so you usually do not need to use the `SET` or `PRIVATE` keywords. If you do not specify either `SET` or `PRIVATE`, the parameter is used only as an initialization parameter for the agent.

`SET` specifies that, in addition to being used as an initialization parameter, the parameter value is set as an environment variable for the agent process. Use `SET` for parameter values that the drivers or non-Oracle system need as environment variables.

`PRIVATE` specifies that the initialization parameter should be private to the agent and should not be uploaded to the Oracle database. Most initialization parameters should not be private. If, however, you are storing sensitive information like a password in the initialization parameter file, then you may not want it uploaded to the server because the initialization parameters and values are not encrypted when uploaded. Making the initialization parameters private prevents the upload from happening and they do not appear in dynamic performance views. Use `PRIVATE` for the initialization parameters only if the parameter value includes sensitive information such as a username or password.

`SET PRIVATE` specifies that the parameter value is set as an environment variable for the agent process and is also private (not transferred to the Oracle database, not appearing in dynamic performance views or graphical user interfaces).

Oracle Database Gateway for Teradata Initialization Parameters

This section lists all the initialization file parameters that can be set for the Oracle Database Gateway for Teradata. They are as follows:

- `HS_DB_DOMAIN`
- `HS_DB_INTERNAL_NAME`
- `HS_DB_NAME`
- `HS_DESCRIBE_CACHE_HWM`
- `HS_LANGUAGE`
- `HS_LONG_PIECE_TRANSFER_SIZE`
- `HS_OPEN_CURSORS`
- `HS_RPC_FETCH_REBLOCKING`
- `HS_RPC_FETCH_SIZE`

- [HS_TIME_ZONE](#)
- [HS_TRANSACTION_MODEL](#)
- [IFILE](#)
- [HS_FDS_CONNECT_INFO](#)
- [HS_FDS_DEFAULT_OWNER](#)
- [HS_FDS_RECOVERY_ACCOUNT](#)
- [HS_FDS_RECOVERY_PWD](#)
- [HS_FDS_TRACE_LEVEL](#)
- [HS_FDS_TRANSACTION_LOG](#)
- [HS_FDS_FETCH_ROWS](#)

Initialization Parameter Description

The following sections describe all the initialization file parameters that can be set for gateways.

HS_DB_DOMAIN

Property	Description
Default value	WORLD
Range of values	1 to 199 characters

Specifies a unique network sub-address for a non-Oracle system. The `HS_DB_DOMAIN` initialization parameter is similar to the `DB_DOMAIN` initialization parameter, described in the *Oracle Database Reference*. The `HS_DB_DOMAIN` initialization parameter is required if you use the Oracle Names server. The `HS_DB_NAME` and `HS_DB_DOMAIN` initialization parameters define the global name of the non-Oracle system.

Note: The `HS_DB_NAME` and `HS_DB_DOMAIN` initialization parameters must combine to form a unique address in a cooperative server environment.

HS_DB_INTERNAL_NAME

Property	Description
Default value	01010101
Range of values	1 to 16 hexadecimal characters

Specifies a unique hexadecimal number identifying the instance to which the Heterogeneous Services agent is connected. This parameter's value is used as part of a transaction ID when global name services are activated. Specifying a nonunique number can cause problems when two-phase commit recovery actions are necessary for a transaction.

HS_DB_NAME

Property	Description
Default value	HO
Range of values	1 to 8 characters

Specifies a unique alphanumeric name for the data store given to the non-Oracle system. This name identifies the non-Oracle system within the cooperative server environment. The `HS_DB_NAME` and `HS_DB_DOMAIN` initialization parameters define the global name of the non-Oracle system.

HS_DESCRIBE_CACHE_HWM

Property	Description
Default value	100
Range of values	1 to 4000

Specifies the maximum number of entries in the describe cache used by Heterogeneous Services. This limit is known as the describe cache high water mark. The cache contains descriptions of the mapped tables that Heterogeneous Services reuses so that it does not have to re-access the non-Oracle data store.

If you are accessing many mapped tables, increase the high water mark to improve performance. Increasing the high water mark improves performance at the cost of memory usage.

HS_LANGUAGE

Property	Description
Default value	System-specific
Range of values	Any valid language name (up to 255 characters)

Provides Heterogeneous Services with character set, language, and territory information of the non-Oracle data source. The value must use the following format:

language[_territory.character_set]

Note: The globalization support initialization parameters affect error messages, the data for the SQL Service, and parameters in distributed external procedures.

Character Sets

Ideally, the character sets of the Oracle database server and the non-Oracle data source are the same. If they are not the same, Heterogeneous Services attempts to translate the character set of the non-Oracle data source to the Oracle database character set, and back again. The translation can degrade performance. In some cases, Heterogeneous Services cannot translate a character from one character set to another.

Note: The specified character set must be a superset of the operating system character set on the platform where the agent is installed.

Language

The language component of the HS_LANGUAGE initialization parameter determines:

- Day and month names of dates
- AD, BC, PM, and AM symbols for date and time
- Default sorting mechanism

Note that Oracle does not determine the language for error messages for the generic Heterogeneous Services messages (ORA-25000 through ORA-28000). These are controlled by the session settings in the Oracle database server.

Note: Use the HS_NLS_DATE_LANGUAGE initialization parameter to set the day and month names, and the AD, BC, PM, and AM symbols for dates and time independently from the language.

Territory

The territory clause specifies the conventions for day and week numbering, default date format, decimal character and group separator, and ISO and local currency symbols. Note that the level of globalization support between the Oracle database server and the non-Oracle data source depends on how the gateway is implemented.

HS_LONG_PIECE_TRANSFER_SIZE

Property	Description
Default value	64 KB
Range of values	Any value up to 2 GB

Sets the size of the piece of LONG data being transferred. A smaller piece size means less memory requirement, but more round-trips to fetch all the data. A larger piece size means fewer round-trips, but more of a memory requirement to store the intermediate pieces internally. Thus, the initialization parameter can be used to tune a system for the best performance, with the best trade-off between round-trips and memory requirements, and network latency.

HS_OPEN_CURSORS

Property	Description
Default value	50
Range of values	1 to the value of Oracle's OPEN_CURSORS initialization parameter

Defines the maximum number of cursors that can be open on one connection to a non-Oracle system instance.

The value never exceeds the number of open cursors in the Oracle database server. Therefore, setting the same value as the `OPEN_CURSORS` initialization parameter in the Oracle database server is recommended.

HS_RPC_FETCH_REBLOCKING

Property	Description
Default value	ON
Range of values	OFF or ON

Controls whether Heterogeneous Services attempts to optimize performance of data transfer between the Oracle database server and the Heterogeneous Services agent connected to the non-Oracle data store.

The following values are possible:

- `OFF` disables reblocking of fetched data so that data is immediately sent from agent to server.
- `ON` enables reblocking, which means that data fetched from the non-Oracle system is buffered in the agent and is not sent to the Oracle database server until the amount of fetched data is equal or higher than the value of `HS_RPC_FETCH_SIZE` initialization parameter. However, any buffered data is returned immediately when a fetch indicates that no more data exists or when the non-Oracle system reports an error.

HS_RPC_FETCH_SIZE

Property	Description
Default value	50000
Range of values	1 to 10000000

Tunes internal data buffering to optimize the data transfer rate between the server and the agent process.

Increasing the value can reduce the number of network round-trips needed to transfer a given amount of data, but also tends to increase data bandwidth and to reduce response time or *latency* as measured between issuing a query and completion of all fetches for the query. Nevertheless, increasing the fetch size can increase latency for the initial fetch results of a query, because the first fetch results are not transmitted until additional data is available.

HS_TIME_ZONE

Property	Description
Default value for '[+ -]hh:mm'	Derived from the <code>NLS_TERRITORY</code> initialization parameter
Range of values for '[+ -]hh:mm'	Any valid datetime format mask

Specifies the default local time zone displacement for the current SQL session. The format mask, [+|-]hh:mm, is specified to indicate the hours and minutes before or after UTC (Coordinated Universal Time—formerly Greenwich Mean Time). For example:

```
HS_TIME_ZONE = [+ | -] hh:mm
```

HS_TRANSACTION_MODEL

Property	Description
Default Value	COMMIT_CONFIRM
Range of Values	COMMIT_CONFIRM, READ_ONLY, SINGLE_SITE

Specifies the type of transaction model that is used when the non-Oracle database is updated by a transaction.

The following values are possible:

- COMMIT_CONFIRM provides read and write access to the non-Oracle database and allows the gateway to be part of a distributed update. To use the commit-confirm model, the following items must be created in the non-Oracle database:
 - Transaction log table. The default table name is HS_TRANSACTION_LOG. A different name can be set using the HS_FDS_TRANSACTION_LOG parameter. The transaction log table must be granted SELECT, DELETE, and INSERT privileges set to public.
 - Recovery account. The account name is assigned with the HS_FDS_RECOVERY_ACCOUNT parameter.
 - Recovery account password. The password is assigned with the HS_FDS_RECOVERY_PWD parameter.
- READ_ONLY provides read access to the non-Oracle database.
- SINGLE_SITE provides read and write access to the non-Oracle database. However, the gateway cannot participate in distributed updates.

IFILE

Property	Description
Default value	None
Range of values	Valid parameter file names

Use the IFILE initialization parameter to embed another initialization file within the current initialization file. The value should be an absolute path and should not contain environment variables. The three levels of nesting limit does not apply.

See Also: *Oracle Database Reference*

HS_FDS_CONNECT_INFO

Property	Description
Default Value	None
Range of Values	Not applicable

HS_FDS_CONNECT_INFO which describes the connection to the non-Oracle system.

The default initialization parameter file already has an entry for this parameter. The syntax for HS_FDS_CONNECT_INFO for the gateway is as follows:

```
HS_FDS_CONNECT_INFO=host_alias:port_number[/database_name]
```

where, *host_alias* is the host alias name or IP address of the machine hosting the Teradata database, *port_number* is the port number of the Teradata database server, and *database_name* is the Teradata database name. The *database_name* variable is optional.

HS_FDS_DEFAULT_OWNER

Property	Description
Default Value	None
Range of Values	Not applicable

The name of the table owner that is used for the non-Oracle database tables if an owner is not specified in the SQL statements.

Note: If this parameter is not specified and the owner is not explicitly specified in the SQL statement, then the user name of the Oracle user or the user name specified when creating the database link is used.

HS_FDS_RECOVERY_ACCOUNT

Property	Description
Default Value	RECOVER
Range of values	Any valid user ID

Specifies the name of the recovery account used for the commit-confirm transaction model. An account with user name and password must be set up at the non-Oracle system. For more information about the commit-confirm model, see the HS_TRANSACTION_MODEL parameter.

The name of the recovery account is case-sensitive.

HS_FDS_RECOVERY_PWD

Property	Description
Default Value	RECOVER
Range of values	Any valid password

Specifies the password of the recovery account used for the commit-confirm transaction model set up at the non-Oracle system. For more information about the commit-confirm model, see the HS_TRANSACTION_MODEL parameter.

The name of the password of the recovery account is case-sensitive.

HS_FDS_TRACE_LEVEL

Property	Description
Default Value	OFF
Range of values	OFF, ON, DEBUG

Specifies whether error tracing is turned on or off for gateway connectivity.

The following values are valid:

- OFF disables the tracing of error messages.
- ON enables the tracing of error messages that occur when you encounter problems. The results are written by default to a gateway log file in LOG directory where the gateway is installed.
- DEBUG enables the tracing of detailed error messages that can be used for debugging.

HS_FDS_TRANSACTION_LOG

Property	Description
Default Value	HS_TRANSACTION_LOG
Range of Values	Any valid table name

Specifies the name of the table created in the non-Oracle system for logging transactions. For more information about the transaction model, see the HS_TRANSACTION_MODEL parameter.

HS_FDS_FETCH_ROWS

Property	Description
Default Value	100
Range of Values	Any integer between 1 and 1000
Syntax	HS_FDS_FETCH_ROWS= <i>num</i>

`HS_FDS_FETCH_ROWS` specifies the fetch array size. This is the number of rows to be fetched from the non-Oracle database and to return to Oracle database at one time. This parameter will be affected by the `HS_RPC_FETCH_SIZE` and `HS_RPC_FETCH_REBLOCKING` parameters.

Index

A

ALTER statement, B-1
Arithmetic operators, B-2

B

BYTE data type, A-1
BYTEINT data type, A-1

C

Case rules, 2-2
Case studies, 3-1
CHAR data type, A-1
character sets
 Heterogeneous Services, D-4
COMMIT
 restrictions, 2-5
Commit point site, 2-5
Comparison operators, B-2
CONNECT BY clause, 2-6
COPY command, 2-7
CREATE statement, B-1
Cursor loops
 restrictions, 2-5

D

Data definition language, B-1
Data dictionary
 views, C-2
Data type
 BYTE, A-1
 BYTEINT, A-1
 CHAR, A-1
 conversion, 2-4
 DATE, A-1
 DECIMAL, A-1
 FLOAT, A-1
 INTEGER, A-1
 LONG, A-2
 LONG VARCHAR, A-2
 RAW, A-1
 SMALLINT, A-2
 TIME, A-2
 TIMESTAMP, A-2

VARBINARY, 2-8
VARBYTE, A-2
VARCHAR, A-2
VARCHAR2, A-2
DATE data type, A-1
DDL statement, 2-6
DECIMAL data type, A-1
DELETE statement, 3-5, B-1, B-2
Demo build SQL script, 3-2
Demonstration files, 3-2
Demonstration tables, 3-2
Demonstration tables build SQL script, 3-2
describe cache high water mark
 definition, D-4
DROP statement, B-1

E

Encrypted format login, 2-8
Error messages
 error tracing, D-9
Errors
 ORA-02070, 2-5

F

fetch array size, with HS_FDS_FETCH_ROWS, D-10
FLOAT data type, A-1

G

Gateway
 case studies, 3-1
 data dictionary tables, C-1
 pass-through feature, 2-1
 supported functions, B-1
 supported SQL syntax, B-1
globalization support
 Heterogeneous Services, D-4
GRANT statement, B-1
Group functions, B-3

H

Heterogeneous Services
 defining maximum number of open cursors, D-5

- optimizing data transfer, D-6
- setting global name, D-4
- specifying cache high water mark, D-4
- tuning internal data buffering, D-6
- tuning LONG data transfer, D-5

Hexadecimal notation, 2-3

HS_DB_NAME initialization parameter, D-4

HS_DESCRIBE_CACHE_HWM initialization parameter, D-4

HS_FDS_CONNECT_INFO, D-8

HS_FDS_DEFAULT_OWNER initialization parameter, D-8

HS_FDS_FETCH_ROWS parameter, D-9

HS_FDS_TRACE_LEVEL initialization parameter, D-9

- enabling agent tracing, D-2

HS_FDS_TRANSACTION_LOG initialization parameter, D-9

HS_LANGUAGE initialization parameter, D-4

HS_LONG_PIECE_TRANSFER_SIZE initialization parameter, D-5

HS_OPEN_CURSORS initialization parameter, D-5

HS_RPC_FETCH_REBLOCKING initialization parameter, D-6

HS_RPC_FETCH_SIZE initialization parameter, D-6

HS_TIME_ZONE initialization parameter, D-6

I

IFILE initialization parameter, D-7

Initialization parameter file

- customizing, D-1

INSERT statement, 3-5, B-1, B-2

INTEGER data type, A-1

K

Known restrictions, 2-4

L

Locking, database, 2-4

LONG data type, A-2

LONG VARCHAR data type, A-2

N

NVL function, 3-4

O

Objects, naming rules, 2-2

ORA-02070, 2-5

P

parameters

- gateway initialization file
- HS_FDS_FETCH_ROWS, D-9

Passing commands to database, 2-6

Pass-Through Feature, 3-5

PL/SQL, 2-8, 2-9

R

RAW data type, A-1

ROLLBACK

- restrictions, 2-5

ROWID, 2-6

S

savepoint support, 2-5

SELECT statement, 3-5, B-1, C-1

SMALLINT data type, A-2

Stored procedures, 2-7

String functions, B-3

SUM function, 3-4

T

TIME data type, A-2

TIMESTAMP data type, A-2

transactional capability, 2-5

transactional integrity, 2-5

TRUNCATE statement, B-1

Two-phase commit, 2-5

U

UPDATE statement, 3-5, B-2

V

VARBINARY data type, 2-8

VARBYTE data type, A-2

VARCHAR data type, A-2

VARCHAR2 data type, A-2

W

WHERE CURRENT OF clause, 2-6