ORACLE DATABASE GATEWAYS

KEY FEATURES

- Integration with non-Oracle relational targets for example DB2 and SQL Server
- Integration with transaction managers for example CICS
- Integration with message queuing systems for example WebSphere MQ
- Integration with legacy targets for example IMS, VSAM and Adabas

Oracle Database Gateways address the needs of disparate data access. In a heterogeneously distributed environment, Gateways make it possible to integrate with any number of non-Oracle systems from an Oracle application. They enable integration with data stores such as DB2, SQL Server and Excel, transaction managers like CICS and message queuing systems like WebSphere MQ.

Introduction

Heterogeneous data access is a problem that affects a lot of companies. Many of them 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 due to the fact 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. The challenge is to quickly, efficiently, and economically deploy data that may exist on many disparate systems through a single application, providing a comprehensive view of the data, regardless of the database or operating system. Oracle offers Oracle Database Gateways for integration in a heterogeneous environment.

Oracle Database Gateways

Oracle Database Gateways provide the ability to transparently integrate with non-Oracle systems 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 non-Oracle systems.

For smooth interoperability between disparate systems, SQL translations, data dictionary translations and data type translations are required, even if the non-Oracle systems are based on SQL standards. Gateways have the ability to translate one system's dialect to another.

Oracle has tailored Gateways to many systems, DB2, Sybase, Informix, SQL Server, IMS, VSAM, Adabas, to name a few. These are specifically coded for the target non-Oracle system. They provide an optimized solution and are also end-to-end certified.

Oracle also offers Database Gateway for ODBC. It is a generic solution that uses an ODBC driver to access any ODBC compliant non-Oracle system. It addresses the needs of data access to many data stores for which Oracle does not have a tailored

solution. Oracle Database Gateway for ODBC makes it possible to integrate with low-end data stores such as MySQL, Foxpro, Access, dBase and non-relational targets like Excel.

Architecture

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 (HS) and a component that is target specific, called an agent. Heterogeneous Services in conjunction with the 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 and is the processing power for Gateways. 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. It 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 that 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.

Agent

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

Key Features

- *Location Transparency* - Users do not need to be aware of the physical location of data, just the names of the tables to be accessed
- *Commit Transparency* - Distributed and non-distributed transactions are committed using the ANSI-standard SQL COMMIT. Oracle automatically detects when a transaction is distributed and uses a two-phase commit protocol to ensure transaction integrity and consistency
- *Two Phase Commit* - Oracle supports two-phase commit with non-Oracle stores. When a public two-phase commit interface is not available, it can be supported when distributed transactions involve Oracle and a single non-Oracle

store

- *Data Type Translations* - Performs automatic data type translations between the Oracle Sever and the non-Oracle system
- *Data Dictionary Translations* - Metadata of the non-Oracle system is translated and displayed in Oracle format
- *Read/Write Access* - Both query and update capabilities are supported for most non-Oracle systems
- *Gateway Mobility* - The Gateway, the non-Oracle system and the Oracle server can all reside on different machines
- *Remote Stored Procedures* - Supports execution of stored procedures defined in non-Oracle system
- *Result Set* - Supports the capability of returning results sets from stored procedures
- *Support for LOBs* - LOB datatype is supported for most non-Oracle systems
- *Date Time datatypes* - Supports Oracle datetime datatypes in SQL and stored procedures
- *Piecewise LONG* - Full support of LONG data type by piecewise handling of the data
- *Pass-Through SQL*- Allows native SQL (including DDL) of target non-Oracle system to be directly transmitted
- *Logon Security* - Enforces access authorizations for remote login
- *Data Encryption* - Available with Oracle's Advanced Security Option
- *NLS and NCHAR Support* - Enables connectivity to non-Oracle systems using multi-byte character sets
- *Improved SQL Generation* - Fine grained capability handling
- *Generated SQL queries cache* - Non-Oracle language queries are generated and cached in Oracle's Shared Pool, making it more efficient for reuse when such queries are frequently used by multiple client sessions
- *Local Cache* - Information about capabilities, SQL translations, and data dictionary translations are stored locally, reducing network traffic
- *Distributed SQL Optimizations* - Improves performance of multi-site JOIN and SET operations