Ensuring Database and Location Transparency in Multiple Heterogeneous Distributed Databases

Shefali Naik(✉)

School of Computer Studies, Ahmedabad University, Ahmedabad, India
shefali.naik@ahduni.edu.in

Abstract. Challenges and issues of distributed database are well known. One of the challenges is obtaining database transparency. In distributed database, the physical database is distributed across heterogeneous database systems. There are several methods of data access from distributed database, still it involves complicated implementation of these methods. Oracle DBMS provides heterogeneous gateway service to connect oracle and non-oracle databases. The data access and distributed transaction execution is made very easy through this gateway. In this paper, the procedure to connect four heterogeneous databases namely PostgreSQL, MySQL, Oracle and MS Access is described. The successful implementation of data access from multiple heterogeneous databases provides very easy and efficient access of data in Oracle using its native language/commands. Oracle user will neither have to bother about architecture nor commands of the remote databases from where data is accessed. Even user need not have to worry about the location of database. Once all the databases are connected through heterogeneous gateway, user will be able to access data and process distributed transactions efficiently from Oracle. The paper covers the whole process.

Keywords: Heterogeneous distributed database · Database transparency Location transparency · Distributed transactions · Heterogeneous gateway

1 Introduction

In Distributed Database [2, 3], the physical database is distributed across multiple sites. One of the architecture of distributed database management system is multiple databases [3, 7]. There are many types of distributed database architectures. One of them is heterogeneous multiple distributed database [1, 3, 7]. In multiple heterogeneous distributed database environment, autonomous multiple databases residing on different locations [3, 7] are connected together to access data of each other. While connecting and accessing data, it should provide transparency at various levels to the users. The different levels of transparency are: Location Transparency, Replication Transparency, Fragmentation Transparency, Database Transparency, Operating System Transparency, Network Transparency, etc. [3–5]. To achieve location and database transparency, Oracle provides heterogeneous gateway service [6]. The meaning and implementation of these two transparencies is explained in the following section.

2 Location and Database Transparency

In distributed database, with location transparency [3], user can access the data without knowledge of location of data. i.e., user doesn't have to specify the location of data in the commands. Similarly, with database transparency [3], user will be able to access data from many databases without knowledge of commands of all the databases. Figure 1 shows the multiple heterogeneous database architecture which connects PostgreSQL, MySQL, MS Access and Oracle database using Oracle's Heterogeneous Gateway Service.

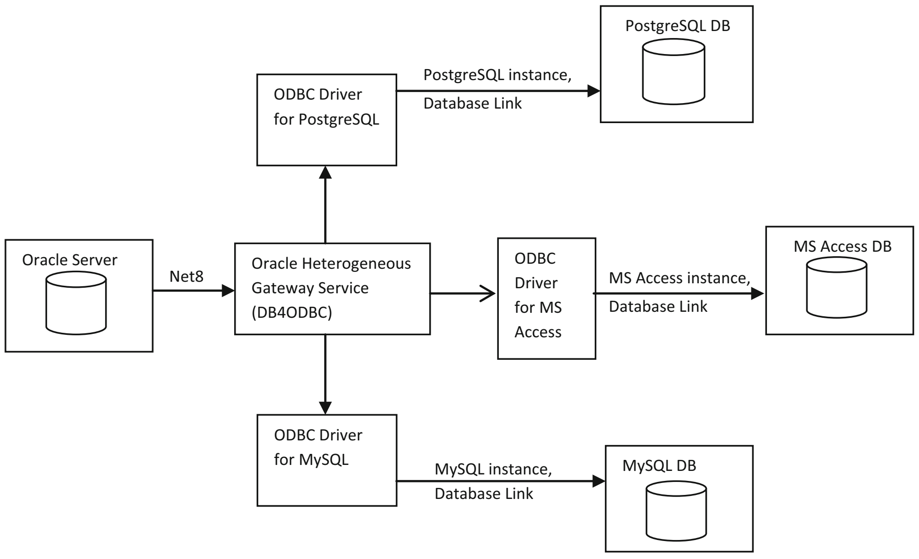
PostgreSQL DB
ODBC Driver for PostgreSQL
PostgreSQL instance, Database Link
Oracle Server
Net8
Oracle Heterogeneous Gateway Service (DB4ODBC)
ODBC Driver for MS Access
MS Access instance, Database Link
MS Access DB
ODBC Driver for MySQL
MySQL instance, Database Link
MySQL DB

Fig. 1. Multiple heterogeneous database architecture

The implementation steps of accessing data from multiple heterogeneous databases [7] are explained in Sect. 3. The process of establishing connection between oracle and non-oracle database require deep knowledge of databases.

3 Implementation

3.1 Steps to Connect Oracle and Non-Oracle Databases

To access data [9] from multiple heterogeneous databases, the respective DBMS software along with its ODBC drivers must be installed. Oracle heterogeneous service is also required which could be installed from oracle technology network. The client library of oracle is also needed. The sequence of flow given below should be followed to connect and access data from non-oracle database to oracle [6].

Step-1. Install Oracle Server, Oracle Client Library and Oracle Heterogeneous Gateway Service from oracle technology network.

Step-2. Install PostgreSQL, MySQL and MS Access databases with respective ODBC drivers.

Step-3. Create users in PostgreSQL and MySQL databases. Grant required privileges to the users. Create tables in specific user's account. Insert data in these tables.

Step-4. Open ODBC data source and create three system data source names for each postgreSQL, MySQL and MS Access databases. Test connections.

Step-5. After successful connections, create three initialization parameter files for each of the system DSN with name "init<DSN>.ora". Save this file in the folder where oracle home is installed. This file will contain name of the database instance, which is system DSN name.

Step-6. Open listener.ora and tnsnames.ora files from the folder where oracle home is installed. Modify both the files with required entries and save.

Step-7. Start listener service from command prompt. Check connection of each of the database instance using "tnsping <DSN>" command.

Step-8. Create database links in oracle client to access data from non-oracle database.

Step-9. Access data from non-oracle database using query language of oracle.

3.2 Flowchart of Oracle and MySQL Connection

The flowchart of whole process of oracle with MySQL connection and data access is given in Fig. 2. The process of connecting different heterogeneous databases could be simplified by writing a program which will automate the steps given in Fig. 2.

4 Achieving Location and Database Transparency

To access the data from heterogeneous databases [9] using the method described in Sects. 2 and 3, there is no need to learn syntax of programming language of all the databases. It could be done only by writing the database link name [9] with table name of a specific database. Thus, database transparency is achieved. The database which is stored on different host, there is no need to specify hostname or address of site anywhere in the commands. Hence, achieving the location transparency. Few examples of SQL commands are given in Fig. 3 and stored procedures are given in Fig. 4.

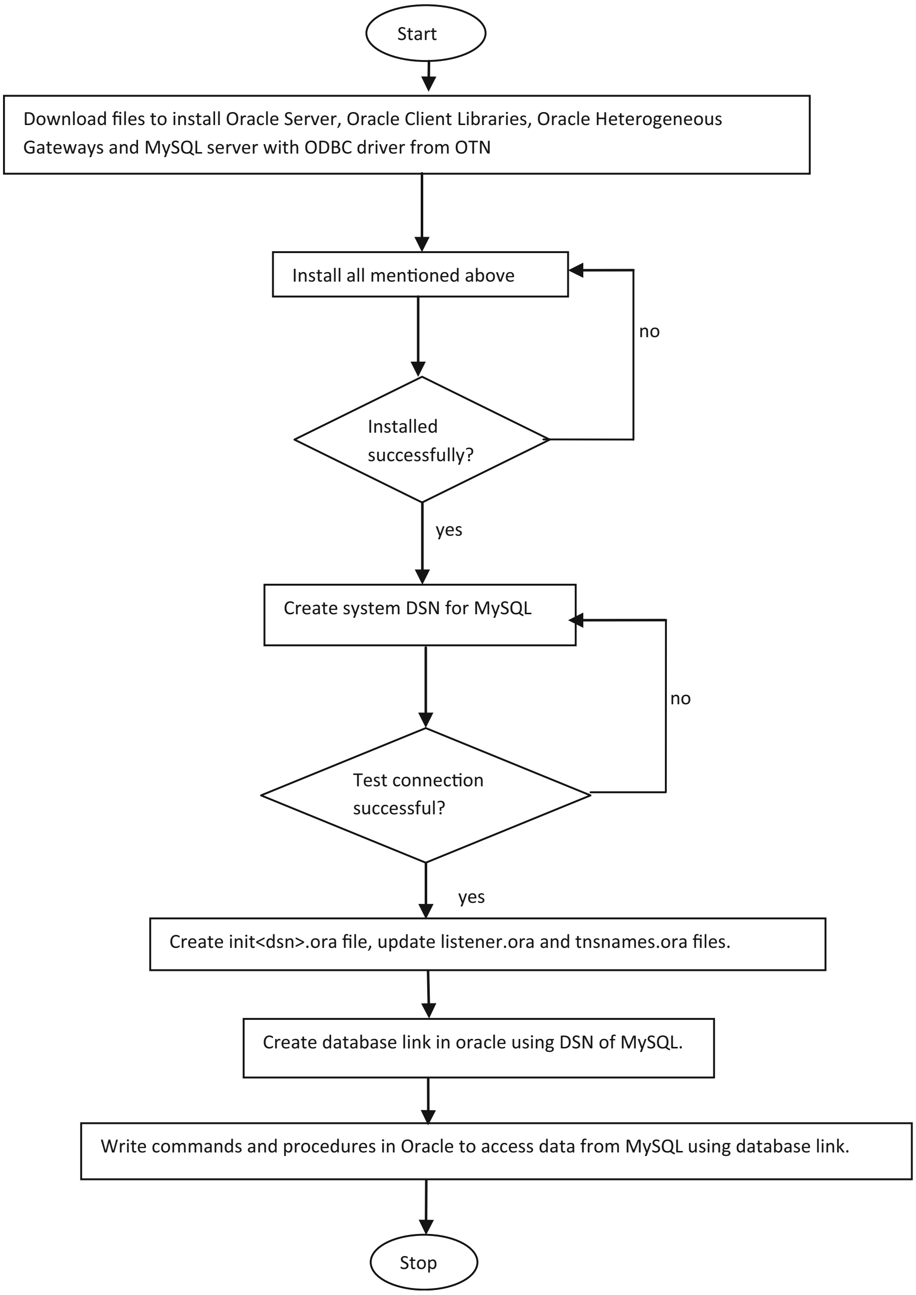
Start
Download files to install Oracle Server, Oracle Client Libraries, Oracle Heterogeneous Gateways and MySQL server with ODBC driver from OTN
Install all mentioned above
Installed successfully?
no
yes
Create system DSN for MySQL
Test connection successful?
no
yes
Create init<dsn>.ora file, update listener.ora and tnsnames.ora files.
Create database link in oracle using DSN of MySQL.
Write commands and procedures in Oracle to access data from MySQL using database link.
Stop

Fig. 2. Flowchart of connection of oracle and non-oracle databases

- Commands written in MySQL

```
mysql> use test;
mysql> create table emp(empno int, empname varchar(30));
mysql> insert into emp values(101,'Shefali');
mysql> create user 'shefali'@'localhost' identified by
'shefali';
mysql> grant all on *.* TO 'sys'@'%'  IDENTIFIED BY
'Shefali';
```

- Commands written in Oracle's SQL Plus

```
SQL>create public database link toms connect to "shefali"
identified by "shefali" using 'MS';

SQL> select * from "emp"@toms;

SQL> select e."eno" from "emp"@toms e;

SQL> select e."eno"||'    '||e."enmae"||'
'||s."sal_date"||'    '||s."salary"
  2  from "emp"@topg e, salary@toac s
  3  where e."eno"=s."empno";
```

Fig. 3. Commands written in oracle client to access data from heterogeneous databases

By creating synonyms [10], the global name could be defined for database link. The global name may be used instead of database link. For ex., "employee" synonym could be created using the following command. Then, "emp"@topg database link could be replaced with "employee".

```
CREATE SYNONYM employee FOR "emp"@topg;
```

```
Create or replace procedure disp as
              cursor c_emp is select * from
"public"."emp"@toms;
              r_emp c_emp%rowtype;
             cursor c_kg is select * from salary@toac;
             r_kg c_kg%rowtype;
begin
            dbms_output.put_line('Data from PostgreSQL…..');
             for r_emp in c_emp loop
                       dbms_output.put_line('Emp Name
==>'||r_emp. "enmae");
            end loop;

            dbms_output.put_line('Data from MS Access…..');
             for r_kg in c_kg loop
                       dbms_output.put_line('Emp No
==>'||r_kg. "empno");
            end loop;
end;
/
```

Fig. 4. Stored procedure written in oracle client to process data of heterogeneous databases

5 Conclusion and Future Work

Data access and process from multiple heterogeneous databases [9] is done very effectively using oracle heterogeneous service. The task is very complicated, which requires in depth knowledge of oracle database administration. Even it is very challenging for database administrators. Therefore, this whole process of creating multiple heterogeneous database environments should be automatic. It may be possible that the whole process could not be automatic, but it could be semi-automatic. The model and implementation of automation of this process could be done in future. It will be very useful if such type of model is developed which ease the process of creation of multiple heterogeneous database environment. Furthermore, work could be done to improve fragmentation, replication and allocation transparencies [3, 4] which will improve performance of distributed concurrent transaction [8] execution in multiple heterogeneous distributed database.

References

1. Ferrier, A., Stangret, C.: Heterogeneity in the distributed database management system SIRIUS-DELTA. In: VLDB (1982)
2. Naik, S.: Concepts of Database Management System. Dorling Kindersley, New Delhi (2014)
3. Özsu, M.T., Valduriez, P.: Principles of Distributed Database Systems. Springer, New York (2011). https://doi.org/10.1007/978-1-4419-8834-8

4. Adiba, M.E., et al.: Issues in distributed data base management systems: a technical overview. In: Proceedings of the Fourth International Conference On Very Large Data Bases, vol. 4. VLDB Endowment (1978)
5. Shefali, N., Samrat, K.: Revisited performance issues in concurrent transaction execution in distributed database management system. Int. J. Curr. Eng. Sci. Res. **2**(4), 23–26 (2015). ISSN (Print): 2393-8374, (Online): 2394-0697
6. Oracle Database Heterogeneous Connectivity User's Guide. https://docs.oracle.com/cd/E11882_01/server.112/e11050.pdf
7. https://www.tutorialspoint.com/distributed_dbms/distributed_dbms_database_environments.htm
8. http://www.techrepublic.com/article/distributed-transactions-span-sql-server-and-oracle/
9. Wang, C.-Y., Spooner, D.L.: Access control in a heterogeneous distributed database management system. In: SRDS (1987)
10. https://docs.oracle.com/cd/B28359_01/server.111/b28310/ds_concepts004.htm#ADMIN12128