ALTIBASE Application Development

Database Link Users' Manual

release 5.3.3

ALTIBASE
PERFORMANCE SOLUTIONS

ALTIBASE Application Development Database Link User’s Manual
Release 5.3.3

Altibase Corporation
10F, Daerung PostTower II, 182-13,
Guro-dong Guro-gu Seoul, 152-847, Korea
Telephone: +82-2-2082-1000 Fax: 82-2-2082-1099
E-mail: support@altibase.com www: http://www.altibase.com

Contents

Preface

About This Manual

This manual contains information to help you understand the concepts of database link and use database link.

Types of Users

This manual is for ALTIBASE users as follows.

- Database Administrators
- Performance Managers
- Database Users
- Application Program Developers
- Technical Assistance Team

This manual assumes that you have the following background :

- A working knowledge of your computer, your operating system, and the utilities that your operating system provides
- Some experience working with relational databases or exposure to database concepts
- Some experience with computer programming
- Some experience with database server administration, operating system administration or network administration

Software Dependencies

This manual assumes that your database server is ALTIBASE server, Version 5.3.1.

How This Manual is Structured

This manual covers the following topics :

- Chapter 1, “Introduction to Database Link”
- This chapter presents an introduction to, and overview of, database link such as what database link is.
- Chapter 2, “Objects, SQL Commands and Data Types for Database Link”
- This chapter covers objects, SQL commands and data types for database link.
- Chapter 3, “Setting up Database Link Environment and Starting up Database Link”
- This chapter discusses how to set up database link environment and how to start up database link.

- Chapter 4, “Working with Database Link”
- This chapter lists all kinds of features that database link provides, and describes how to create and drop database link.
- Appendix A, Property and Data Dictionary
- This appendix lists all kinds of properties and data dictionaries related to database link.

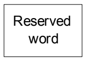
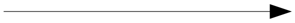
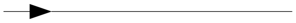
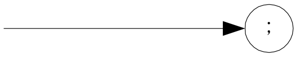
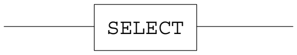
Documentation Conventions

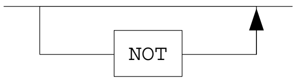
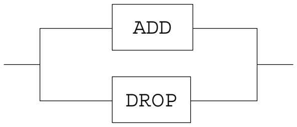
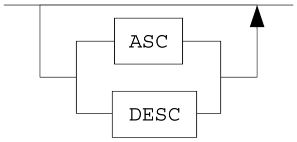
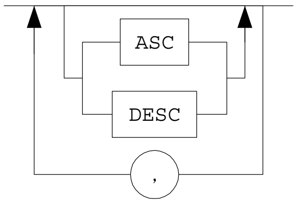
This section offers documentation conventions as follows. They make it easier to gather information from Altibase manuals.

- Command-Line Conventions
- Typographical Conventions

Command-Line Conventions

This section defines and illustrates the format of commands that are available in Altibase products. These commands have their own conventions, which might include alternative forms of a command, required and optional parts of the command, and so forth.

Element	Description
	The command starts. The syntax element which is not a complete command starts with an arrow.
	The command continues in the next line. The syntax element which is not a complete command terminates with this symbol.
	The command continues from the previous line. The syntax element which is not a complete command starts with this symbol.
	End of a statement.
	Mandatory

Element	Description
	Optional
	Mandatory field with optional items Only one field must be provided..
	Optional field with optional item
	Optional Multiple fields are allowed. The comma must be in front of every repetition.

Typographical Convetions

This manual uses the following standard set of conventions to introduce new terms, illustrate screen displays, describe command syntax, and so forth.

Convention	Meaning	Example
[ ]	Displays the optional fields.	VARCHAR [(size)] [[FIXED |] VARIABLE]
{ }	Displays the mandatory fields. Specifies a field that requires selection of more than one item.	{ ENABLE | DISABLE | COMPILE }
|	Argument indicating optional or mandatory fields	{ ENABLE | DISABLE | COMPILE }[ENABLE | DISABLE | COMPILE]

Convention	Meaning	Example
...	Repetition of the previous argument-Specifies the omission of the example codes.	SQL> SELECT ename FROM employee; ENAME ----------------------- SWNO HJNO HSCHOI . . . 20 rows selected.

Other symbols	Other Symbols	EXEC :p1 := 1;
		acc NUMBER(11,2);

Italics	Within text, new terms and emphasized words appear in italics. Within syntax, diagrams, values that you are to specify appear in italics.	ELECT * FROM table_name;
		CONNECT userID/password;

Lower Case Letters	Program elements provided by the user such as table names, column names, file names, etc.	SELECT ename FROM employee;
Upper Case Letters	All elements provided by the system or keywords appear in uppercase letter.	DESC SYSTEM_.SYS_INDICES_;

Related Reading

For additional technical information, consult the following manuals.

- ALTIBASE Administration Starting User's Manual
- ALTIBASE Administration Administrator's Manual
- ALTIBASE Administration Replication User's Manual
- ALTIBASE Application Development SQL User's Manual
- ALTIBASE Application Development ODBC User's Manual
- ALTIBASE Application Development Spatial SQL User's Manual
- ALTIBASE Application Development Application Program Interface User's Manual
- ALTIBASE Tools iSQL User's Manual
- ALTIBASE Message Error Message Reference

On-Line Manuals

Manuals (PDF and HTML) in Korean and English are available at Altibase Technical Center (http://atc.altibase.com/).

Altibase Welcomes Your Comments

Please let us know what you like or dislike about our manuals. To help us with future versions of our manuals, please tell us about any corrections or classifications that you would find useful.

Include the following information :

- The name and version of the manual that you are using
- Any comments that you have about the manual
- Your name, address, and phone number

Write to us at the following electronic mail address : support@altibase.com

When you need an immediate assistance regarding technical issues, please contact Altibase Technical Center.

Thank you. We appreciate your feedback and suggestions.

1 Introduction to Database Link

This chapter describes database link concepts such as what it is, and also lists features that database link provides.

What is Database Link?

Concepts

A database link can be a pointer that defines a one-way communication path from an Altibase server to another database server when you want to access data on a remote database.

These database servers should meet the following conditions.

- Database servers should have mutually logical relationship.
- Database servers should be connected to computer communication network.
- Database servers should be separated locally.

Their logical relationship means that there is a correlation of data between servers, and you should be given the privileges you need to perform your operations and can also centralize data segregated to different databases.

Database servers connected to computer communication network means that they are linked each other via hardware and software. These hardware and software cover from intranet or subnet to WAN or internet. Database servers can be on the same subnet.

Database servers separated locally means that they as data service provider are designed to run independently and cover from short-range hardware platforms to individual sites.

Terminologies

- Local Server

  A database is the subject that creates and uses database link.

  Local server sends queries to remote server for execution.

  Remote server executes them and send their results back to local server. Only ALTIBASE as a local server is available to run database link.

- Remote Server

  A database is the destination where local server sends queries for execution. Remote server executes them sent from a local server such as ALTIBASE, and send their results back to a local server such as ALTIBASE.

  You can also use ALTIBASE or relational databases as a remote server. Relational database should be connected to local server with using ODBC.

- AltiLinker

  AltiLinker is the process that sends/receives data to/from remote server when you use database link. If database link is active, AltiLinker does the same as remote server.

  For example, when remote server is started up, AltiLinker starts running and when server is shut down, AltiLinker stops running.

However, you can manually terminate or restart up AltiLinker with ALTER DATABASE LINKER statement.

- Location Descriptor

  Location descriptor should be used in queries to set database link, and is specified as @ between names of object and link.

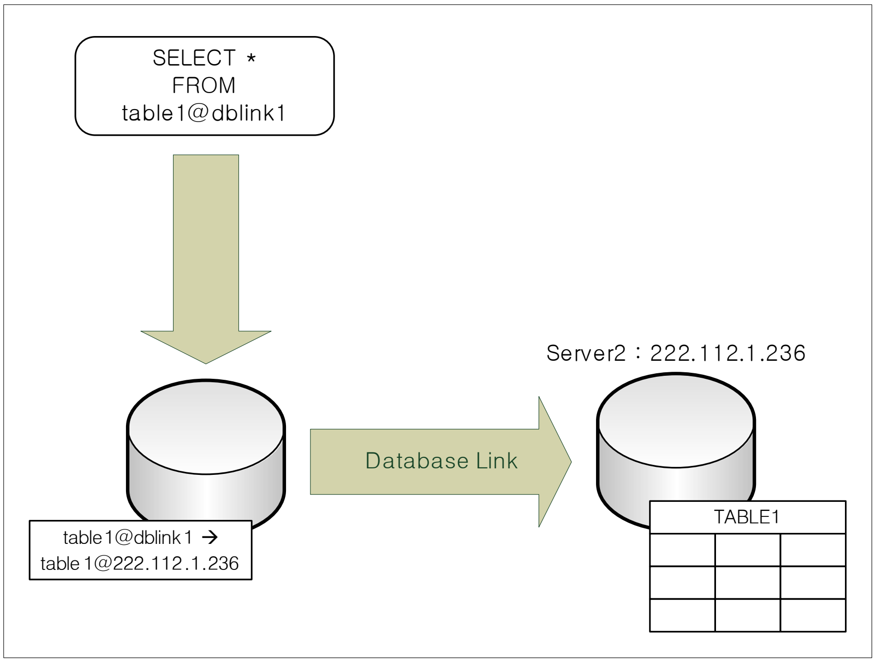
  Location descriptor is used only in FROM clause.

  SELECT * FROM emp@link1;

How to Implement Database Link

A database link is a connection between two physical database servers that allow a client to access them as one logical database.

Figure 1-1 How to Implement Database Link



[Figure 1-1] illustrates how to implement database link in outline.

If you execute queries, local server sends queries to remote server after checking if remote server has their destination table. And then remote server sends their results back to local server.

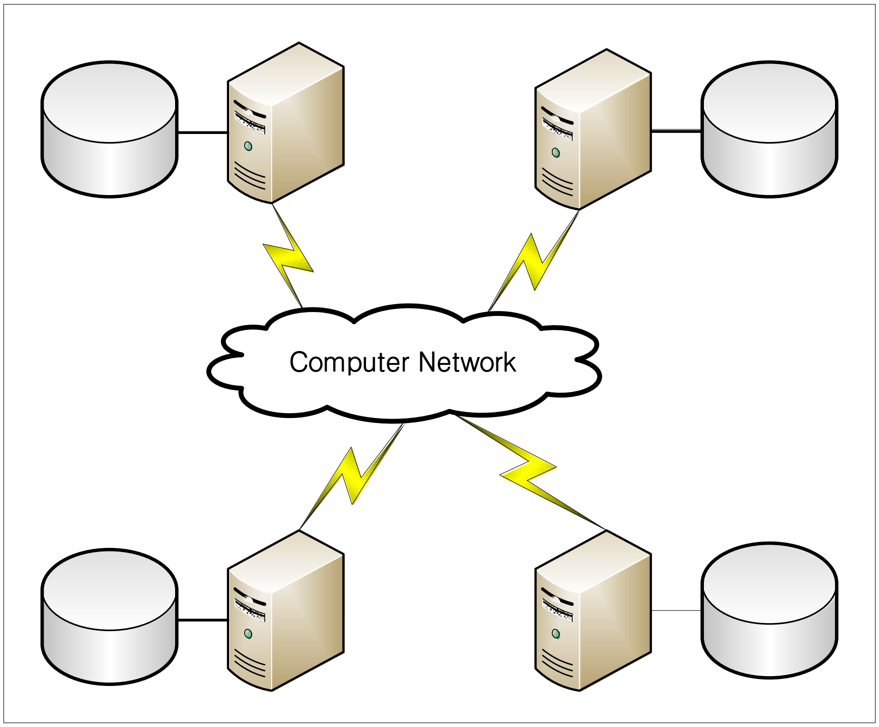
System Environment

Segregated servers should provide independent operating environment to implement database link. In other words, queries should be perfectly suitable for each servers to execute them.

For example, if Site1 stores information about the manufacturing process or a product, local server should be designed to execute transaction without undergoing a process of other systems when transaction related to product information occurs.

The following figure roughly illustrates operating environment of database link. Multiple sites or servers provide independent operating environment and are interconnected with computer communication network. Each of them has its own database. Queries can be sent to other sites and servers for execution through computer communication network.

Figure 1-2 Operating Environment of Database Link



Features of Database Link

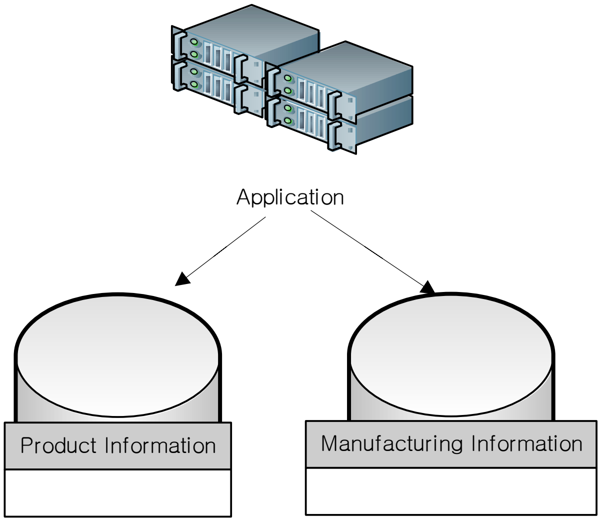
Database link has following features.

- Convenience
- Efficiency
- Scalability
- High Availability

Convenience

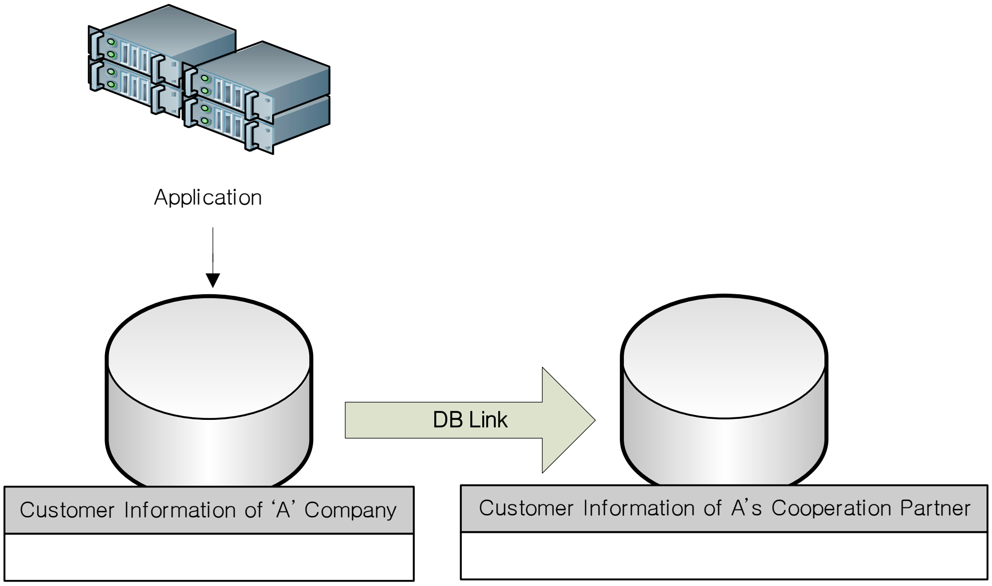
Database link can provide transparency in location of remote servers. In other words, user application does not need to know the location of data used for last result.

Figure 1-3 Application's Access to Each Remote Servers



For example, assume that product information is stored in Site1 and manufacturing information is stored in Site2 respectively as [Figure 1-4]. If you want to know manufacturing information on certain product, user application gets product information in Site1 and manufacturing information in Site2 respectively and then joins data in Site1 and Site2. You can get required information.

Figure 1-4 Application's Access with Database Link

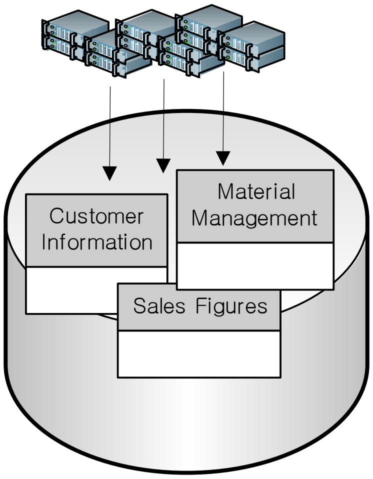

Application
DB Link
Customer Information of 'A' Company
Customer Information of A's Cooperation Partner

However, if executing queries in local server of system using database link, local server displays final results sent by remote server. In other words, even though data are distributed in multiple servers or sites physically, you can use them as if they are stored in one server logically.

Efficiency

If mass data are centralized, QoS can't be guaranteed because multiple transactions can interrupt each other when they occur.

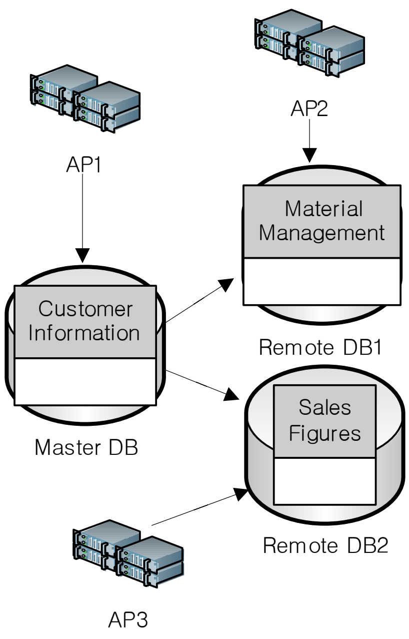
Figure 1-5 The Environment of Single Server



However, if you distribute data that are seldom joined logically to multiple servers, table workloads are also provisioned across them.

You can join data in each table with using database link. In other words, transactions interrupt each other less and less. So you can use system resource such as CPU, memory and IO because transaction can occur in each server independently.

Figure 1-6 Data Distribution Environment



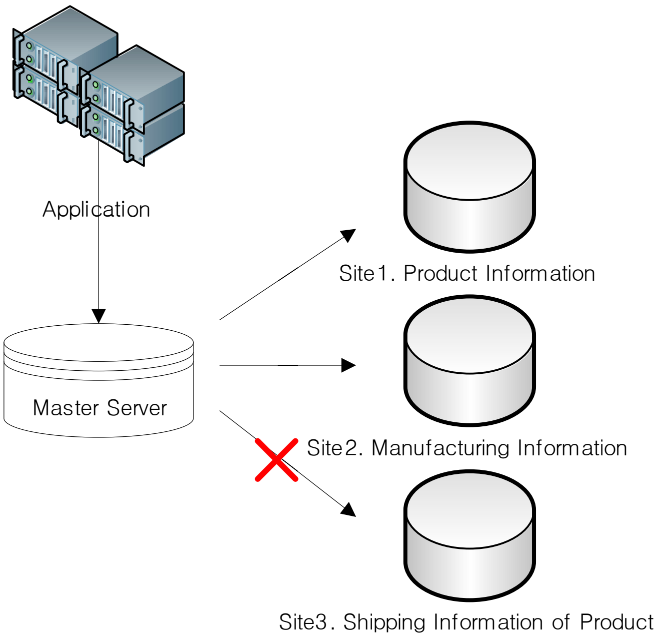
Scalability

Database link has more flexible scalability than centralized system because if using database link, you can install more servers even though running system reaches the limits of its process.

High Availability

Even though failure occurs in some of all systems running database link, the others can continue to perform work. [Figure 1-7] illustrates database link system where Site1 stores product information, Site2 contains manufacturing information and Site3 has shipping information.

Figure 1-7 Available Service even with Failure in some Servers


Application
Master Server
Site1. Product Information
Site2. Manufacturing Information
Site3. Shipping Information of Product

2Objects, SQL Commands and Data Types for Database Link

Objects for Database Link

This section covers what objects are existed in ALTIBASE to support database link.

Schema Objects

The following table displays what shema objects are existed to support database link.

Table 2-1 Schema Objects for Database Link

Schema Object	Supported
Table	O
Index	X
View	O
Stored Procedure	?
Sequence	X
Queue	X
Trigger	X
Synonym	X
Constraint	X

Table

A table is most basic schema object organized in database and stores records physically. Database links are a foundation for inter operability between separated tables.

View

View can join and simplify multiple tables into a single virtual table.

Views can be joined to create new one and made up of base tables using database link. So view is also object using database link because a view is a virtual table defined on base tables.

Stored Procedure

Stored procedure is a database procedure executing several SQL statements at once with input, output and input/output arguements depending on condition specified in body. Database link is used in a stored procedure except ROWTYPE variable declaration and cursor.

SQL Commands for Database Link

Commands	Supported
DDL	X
DCL	X
SELECT	O
DML	X

DDL

Local server can't execute DDL(Data Definition Language) in remote server by using database link.

DCL

Local server can't execute DCL(Data Control Language) in remote server by using database link.

DML

ALTIBASE doesn't support operations such as INSERT, UPDATE and DELETE, but SELECT operation for using database link. A SELECT statement can contain all clauses such as selection, join, subquery, aggregation, set and view. You can use database link with a SELECT query nested inside DDL/DML statement.

Data Types for Database Link

Only standard data types in ODBC are supported because you can access database via ODBC interface when using database link. The following table shows supported or unsupported data types defined by ALTIABSE.

Table 2-2 Data Types for Database Link

ALTIBASE Data Types	ODBC Data Types	Supported
CHAR	SQL_CHAR	O
VARCHAR	SQL_VARCHAR	O
NCHAR	SQL_WCHAR	O
NVARCHAR	SQL_WCHAR	O
BIGINT	SQL_BIGINT	O
DECIMAL	SQL_DECIMAL	O
DOUBLE	SQL_DOUBLE	O
FLOAT	SQL_FLOAT	O
INTEGER	SQL_INTEGER	O
NUMBER	SQL_NUMERIC	O
NUMERIC	SQL_NUMERIC	O
REAL	SQL_REAL	O
SMALLINT	SQL_TINYINT	O
DATE	SQL_DATE	O
BLOB/CLOB		X
BYTE		X
NIBBLE		X
BIT		X
GEOMETRY		X

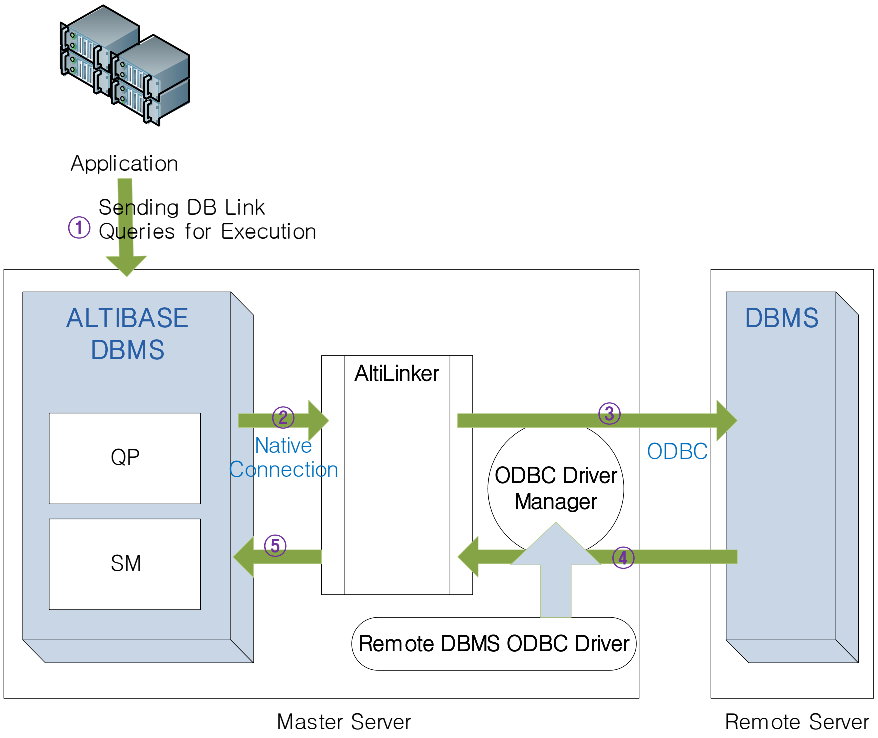
3 Configuration and Operation of Database Link

Database link supported by ALTIBASE uses ODBC to access from local server to remote server. This chapter assumes that you already configure unixODBC as an open source product because ODBC driver manager can be differently installed depending on situation and configuration in the system.

How to Run Database Link

Database Link Procedure

Figure 3-1 Procedure of Running Database Link



1. You should send queries containing locator to local server.

2. Query processor parses these queries and analyzes locator in local server. And then query processor regenerates queries to send to remote server as a result of analyzing locator and plans.

3. If query processor asks storage manager to use a table scan according to the plans, storage manager sends a request of a table scan in remote server to AltiLinker and waits the result. At this time, native connections are built for communication between storage manager and AltiLinker.

4. AltiLinker passes the query from storage manager to remote server for execution. With ODBC, AltiLinker can access and interact with remote server. Therefore, ODBC driver manager should be installed in local server and ODBC driver should be installed in remote server respectively.

5. Remote server executes queries as requested from AltiLinker and then passes the query result via ODBC. Data size of result can be different depending on the format of queries regenerated by query processor.

6. The query result is returned to AltiLinker from remote server and then AltiLinker sends each record of the query result to local server. Local server converts the data from ODBC format into

a format expected by local server and passes them from storage manager to query processor.

7. Query processor returns data after pulling together and filtering data sent from storage manager. Data should be suited for the result of queries executed by you.

Database Link Communication

2.> and 5.> above this section discuss the communication between local server and AltiLinker. You can choose the type of native connections depending on LINKER_LINK_TYPE. If LINKER_LINK_TYPE is set to 0, TCP is used. If LINKER_LINK_TYPE is specified as 1, Unix domain is used. If LINKTER_LINK_TYPE is set to 2, IPC is used.

Only ODBC is supported currently for communication in 3.> and 4.>. For details, see 'Configuration'

Configuration

ODBC Driver Manager Installation

A database link is a connection between local and remote servers that allow a client to access them with ODBC. Altibase ODBC driver should be added to package. To use this, ODBC driver manager should be additionally installed in the system because ALTIBASE doesn't provide ODBC driver manager.

UnixODBC driver manager is supported as an open source product on Unix and Linux. You can find the release at http://www.unixodbc.org. Other ODBC driver managers are Data Direct, Eye Soft and so on. Windows has its own driver.

ODBC provides a common interface for applications to access data from database management systems. Access to database is managed by ODBC driver manager. So the driver manager probably can vary depending on specific database.

ALTIBASE adds library path of ODBC driver manager installed in ALTIBASE to library environment variable of account operated in ALTIBASE and creates odbc.ini in home directory for ALTIBASE account or etc directory where ODBC driver manager is installed.

How to Configure odbc.ini

You should add DSN when creating odbc.ini. and can set several DSNs in odbc.ini. as follows.

```
[ODBC Data Sources]
altibase_odbc = Altibase ODBC Driver

[altibase_odbc]
Driver = /home/altibase/altibase_home/lib/libaltibase_odbc-64bit-ul32.so
ServerType = Altibase
Server = 192.168.3.62
Port = 20300
NLS_USE = US7ASCII
Database = mydb
FetchBufferSize = 64
ReadOnly = no
```

If you create altibase_odbc as DSN, information of destination server with using database link are also specified. At this time, you can choose libaltibase_odbc-64bit-ul32.so or libaltibase_odbc-64bit-ul64.so in $ALTIBASE_HOME/lib/ as a driver if using ALTIBASE as a remote database. Because sizes of SQLLEN and SQLULEN vary depending on types of 64bit ODBC managers.

ALTIBASE version 5.1.5.28 or later supports ALTIABSE 64bit unixODBC driver with both SQLLEN 32bit and 64bit. 64bit server/client packages contain 2 drivers as follows.

Altibase 64 bit unix odbc driver	libaltibase_odbc-64bit-ul64.so(SQLLEN is 64bit)
	libaltibase_odbc-64bit-ul32.so(SQLLEN is 32bit)

You can select a driver depending on ODBC manager as the following table.

Table 3-1 ODBC Manager

64 bit odbc manager	SQLLEN
unix-odbc 64bit(~2.2.12)	32bit
unix-odbc 64bit (~2.2.12) DBUILD_REAL_64_BIT_MODE	64bit
unix-odbc 64bit(2.2.13~)	64bit
unix-odbc 64bit (2.2.13~) BUILD_LEGACY_64_BIT_MODE	32bit
iodbc 64bit	64bit
Window 64bit	64bit

Procedure for Running Database Link

Linker Activation

AltiLinker should be activated to use Altibase database link. To achieve this, you should set the values of database link properties in $ALTIBASE_HOME/conf/altibase.properties.

First, you should start to specify DBLINK_ENABLE as 1 to activate linker and then specify LINKER_PORT_NO as port number used when databases send or receive data. At this time, if LINKER_LINK_TYPE is TCP, you should set LINKER_PORT_NO.

If you start up ALTIBASE after specifying properties, Linker runs with ALTIBASE together. For details about properties related to database link, see properties related to database link section in *Starting User's Manual*.

4 How to Use Database Link

Database link creation, its drop and extra features are covered in this chapter.

CREATE DATABASE LINK

You can create database link with CREATE DATABASE LINK statement for only one remote server.

Syntax

```
CREATE [PUBLIC|PRIVATE] DATABASE LINK dblink_name
WITH ODBC dsn
CONNECT TO user_id IDENTIFIED BY password;
```

Description

To create a database link, you must have the CREATE DATABASE LINK system privilege and user's SYS privilege.

PUBLIC|PRIVATE

You should specify database link to create as PUBLIC or PRIVATE.

You can specify PUBLIC to create a public database link available to all users. If you omit this clause, the database link is private and is available only to you.

dblink_name

This sets the name of database link to be created.

dsn

ODBC driver should be installed in the server because database link enables you to access to remote server with ODBC. And DSN for remote server should be also specified in ODBC environment file.

For details, see 'How to Configure odbc.ini.'

user_id/password

This sets user id and its password required to access data on a database in remote server. However, if odbc.ini has information of user and password, this setting has a higher priority than that specified by this statement. Therefore, ODBC setting is not recommended when you use CREATE DATABASE LINK statement.

However, the user should be granted to access destination objects when connecting to server with using database link. Otherwise, errors occur related to privileges.

Example

<Query1> user1 is specified as both user id and its password for database link in remote database whose DSN is altibase_odbc. At this time, user should create database link whose name is link1 used only for him.

```
iSQL> CREATE PRIVATE DATABASE LINK link1
 WITH ODBC altibase_odbc
 CONNECT TO user1 IDENTIFIED BY user1;
```

<Query2> user1 is specified as both user id and its password for database link in remote database whose DSN is altibase_odbc. At this time, you should create database link whose name is link2 used for all users in the system.

```
iSQL> CREATE PUBLIC DATABASE LINK link2
 WITH ODBC altibase_odbc
 CONNECT TO user1 IDENTIFIED BY user1;
```

DROP DATABASE LINK

You can drop database link with DROP DATABASE LINK statement.

Syntax

```
DROP [PUBLIC|PRIVATE] DATABASE LINK dblink_name;
```

Description

You should be SYS user and have the DROP DATABASE LINK system privilege to drop database link.

dblink_name

This sets the name of database link to be dropped.

Restriction

You can't drop database link while using it. You can drop database link only if you don't execute queries for it. If you drop database link while executing queries, errors occur.

Example

<Query1> user1 should drop database link specified as PRIVATE.

The name of database link is dblink1.

```
iSQL> DROP DATABASE LINK dblink1;
```

<Query2> user1 should drop database link specified as PUBLIC.

The name of database link is dblink1.

```
iSQL> DROP PUBLIC DATABASE LINK dblink1;
```

Extra Features of Database Link

Syntax

```
ALTER DATABASE LINKER START;
ALTER DATABASE LINKER STOP;
ALTER SESSION CLOSE DATABASE LINK;
```

Prerequisite

You can use extra features of database link in sysdba mode with user's SYS privilege.

Description

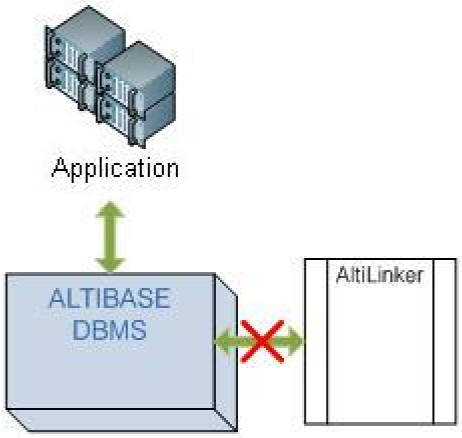
LINKER START

This starts to run AltiLinker. However, there shouldn't be the existing AltiLinker.

LINKER STOP

This stops to run AltiLinker. However, no transaction should exist using database link when terminating database link. Otherwise, you fail to execute this statement.

SESSION CLOSE



You can close only database link with this statement. When you connect to server, session is created there. If you run database in this session, session allocated for you creates other session for database link to connect to AltiLinker. And when session is closed, session for database link is also closed together.

If you access a database link in a session, then the link remains open until you close the session. At this time, you can close a database link session with ALTER SESSION CLOSE DATABASE LINK statement. In other words, you can't end entire session allocated for you with SESSION CLOSE statement but close session for database link.

Example

```
$ isql -u sys -p manager -sysdba
-------------------------------------------------------
 Altibase Client Query utility.
 Copyright 2000, ALTIBASE Corporation or its subsidiaries.
 All Rights Reserved.
-------------------------------------------------------
ISQL_CONNECTION = UNIX, SERVER = 127.0.0.1, PORT_NO = 20300

iSQL(sysdba)> ALTER DATABASE LINKER STOP;
Alter success.
iSQL(sysdba)> ALTER DATABASE LINKER START;
Alter success.
iSQL(sysdba)> ALTER SESSION CLOSE DATABASE LINK;
Alter success.
```

SELECT Statements

ALTIBASE doesn’t support INSERT, UPDATE and DELETE operations currently, but grants only SELECT operation when you use database link. This section discusses operations and features using SELECT statements.

FROM Clause

You should set location descriptor in FROM clause of SELECT statements to use database link. Location descriptor is comprised of @ and link name.

Object name of location descriptor should exist in remote server.

Object can contain user name. You can use table and view as destination objects supporting database link in ALTIBASE.

Example

<Example 1>

```
SELECT * FROM emp@link1;
```

WHERE Clause

You should use WHERE clause when setting condition for object in remote server to use database link. When using WHERE clause, you shouldn’t set location descriptor in WHERE clause. You should give alias to destination object in FROM clause and use alias in WHERE clause.

Example

<Example 1> Local server searches for rows whose value is greater than 100 in I1 on T1 table of remote server that link1 indicates.

```
SELECT * FROM T1@link1 WHERE I1 > 100;
```

<Example 2> Local server searches for names of employees that work in rnd department. They should be saved in both emp1 of local server and emp2 of remote server that link1 indicates.

```
SELECT emp_name
FROM (SELECT emp_no, emp_name FROM emp1
    UNION ALL
    SELECT emp_no, emp_name FROM emp2@link1 ) v1,
  dept
WHERE v1.emp_no = dept.emp_no AND dept.dept_name = 'rnd';
```

Others

You can use options such as selection, join, subquery, aggregation, set and view in SELECT statement. You can also use subselect statements to run database link.

Example

<Example 1> Local server searches for i1 columns in T1 of local server that link1 indicates without repeating same columns.

```
SELECT DISTINCT i1 FROM T1@link1;
```

<Example 2> You can get result containing departments of all employees in local server when joining t_member with t_dept in remote server that link1 indicates. If the value of department ID is 0 or greater, local server estimates the number of employees and their average age in each department.

```
SELECT t1.dept_id, COUNT(*), AVG(age), SUM(age)
FROM t_member@link1 t1, t_dept@link1 t2
WHERE t1.dept_id=t2.dept_id
GROUP BY t1.dept_id
HAVING t1.dept_id>=0;
```

<Example 3> You can get result containing all employees in local server when joining t_member with t_dept in remote server that link1 indicates. If employees are more than 30 years old, local server choose 3 employees. The value of their ID should be biggest in descending numeric order. Last result in local server is expressed in their names, age and the total sum of age of all employees.

```
SELECT t1.name, t1.age,
 (SELECT SUM(age) FROM t_member@link1) sum
FROM t_member@link1 t1,
 (SELECT dept_name, dept_id
 FROM t_dept@link1) t2
WHERE t1.dept_id=t2.dept_id
 AND t1.age<30
 AND 10 > (SELECT count(*) FROM t_dept@link1)
ORDER BY t1.member_id DESC LIMIT 3;
```

<Example 4> Local server searches for name and age in t2 of remote server that link1 indicates, and inputs them in t1 of local server.

```
INSERT INTO t1 SELECT name, age FROM t2@link1;
```

EXEC_REMOTE Hint

Using EXEC_REMOTE hint enables local server to send and execute queries to remote server. If you use EXEC_REMOTE hint, local server can work JOIN in remote server with database link and receive only result set with columns from different tables. And this is also possible for queries if you specify AUTO_REMOTE_EXEC as 1.

Restrictions

You can use EXEC_REMOTE hint if target objects exist in same remote server when executing queries. However, you can't use EXEC_REMOTE hint considering the following.

- Subqueries contain objects of local server.
- Database link names of objects are different.
- Subqueries contain a stored procedure or sequence.
- Host variable is used in subqueries.
- Subqueries reference external column.

Even though subqueries meet one of criterions above, EXEC_REMOTE hint is ignored. If outer and inner queries use EXEC_REMOTE hint, EXEC_REMOTE hint works in outer query and is ignored in inner query. In other words, if outer query doesn't use EXEC_REMOTE hint, EXEC_REMOTE hint can work in inner query.

Example

<Example 1> Local server searches for the name on employee table in remote server that link1 indicates, and then executes entire queries. Only result is returned in local server.

```
SELECT /*+EXEC_REMOTE*/ DISTINCT name
FROM employee@link1;
```

<Example 2> If the sum total of values of I2 in rows whose value is 3 in I1 on T2 table of remote server that link1 indicates are greater than 5, local server searches for rows on T1 table in remote server.

```
SELECT /*+EXEC_REMOTE*/ * FROM T1@link1
WHERE 5 < ( SELECT /*+EXEC_REMOTE*/ SUM(I2)
FROM T2@link1 WHERE I1=3 );
```

<Example 3> This example has same queries as <Example 2> except hint. Subqueries judge truth or false of conditional clause with results returned by remote server that executes entire queries.

And then local server searches for data in T1 by using last results in remote server again.

```
SELECT * FROM T1@link1
WHERE 5 < ( SELECT /*+EXEC_REMOTE*/ SUM(I2)
FROM T2@link1 WHERE I1=3 );
```

EXEC_REMOTE Hint

Appendix A. Property and Data Dictionary

Property related to Database Link

You should modify ALTIBASE property files to have the perfect ones for using database link. Properties related to database link are as follows. For more details, see *Starting User's Manual*.

- AUTO_REMOTE_EXEC
- DBLINK_ENABLE
- DBLINK_QUERY_TIMEOUT
- LINKER_CONNECT_TIMEOUT
- LINKER_LINK_TYPE
- LINKER_PORT_NO
- LINKER_RECEIVE_TIMEOUT
- LINKER_THREAD_COUNT
- LINKER_THREAD_SLEEP_TIME
- MAX_DBLINK_COUNT
- REMOTE_SERVER_CONNECT_TIMEOUT

Data Dictionary related to Database Link

You can check current state of database link with data dictionary and performance view as follows. For more details, see *Administrator's Manual*.

Data Dictionary

- SYS_DATABASE_LINKS_

Performance View

- V$DBLINK_REMOTE_STATEMENT_INFO

Data Dictionary related to Database Link

- V$DBLINK_REMOTE_TRANSACTION_INFO
- V$DBLINK_TRANSACTION_INFO
- V$LINKER_STATUS

Index