



Network Link

Day -1

DB Link - Network

Database Links

To establish connection between one database to another database.

DBLINK	DEFINITIONS
Private dblink	only the owner of a private dblink can use it.
Puplic dblink	all user in database can use it.
Global dblink	defined in an OID an Oracle Name Server anyone on the network can use it



Database Link



database link

DBLINK

database Link is also known as dblink.

Network configure is must.

create database link privilege granted to user who using the dblink.

A database link use to referece a remote database objects.

The remote database will be anoter oracle database,

Accessing a remote object by dblink their local database work as client.



database link

Dblink

```
SQL> show parameter db_name
```

NAME	TYPE	VALUE
-----	-----	-----
db_name	string	sdbt1

```
SQL> show parameter db_name
```

NAME	TYPE	VALUE
-----	-----	-----
db_name	string	saidb

```
SQL> grant create database link to sdbt;
```

```
Grant succeeded.
```



database link

Dblink

```
SQL> conn sdbt  
Enter password:  
Connected.
```

```
SQL> select * from global_name;
```

```
GLOBAL_NAME  
-----
```

```
SDBT1
```

```
SQL> conn steve  
Enter password:  
Connected.
```

```
SQL> select * from global_name;
```

```
GLOBAL_NAME  
-----
```

```
SAIDB
```

```
SQL> select * from session_privs  
2 where privilege like '%DATABASE%';  
PRIVILEGE  
-----
```

```
CREATE DATABASE LINK
```

Dblink

```
SQL> create table test_tab  
2 (id number(5), name varchar2(20));
```

Table created.

```
SQL> insert into test_tab values(1001,'steve  
first line');
```

1 row created.

```
SQL> commit;
```

Commit complete.

database link

Network - Listener

```
SID_LIST_LISTENER =  
  (SID_LIST =  
    (SID_DESC =  
      (SID_NAME = sdbt1)  
      (ORACLE_HOME = /u01/app/oracle/product/12.2.0.1/db_1)  
    )  
  )  
  
LISTENER =  
  (DESCRIPTION_LIST =  
    (DESCRIPTION =  
      (ADDRESS = (PROTOCOL = TCP)(HOST = sdbt.localdomain)(PORT = 1521))  
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))  
    )  
  )
```


database link



Network - Listener

```
SID_LIST_LISTENER =  
  (SID_LIST =  
    (SID_DESC =  
      (SID_NAME = saidb)  
      (ORACLE_HOME = /u01/app/oracle/product/12.2.0.1/db_1)  
    )  
  )  
  
LISTENER =  
  (DESCRIPTION_LIST =  
    (DESCRIPTION =  
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.233)(PORT = 1521))  
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))  
    )  
  )
```

database link



Network - Tnsnames

```
SDBT1 =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = sdbt.localdomain)(PORT = 1521))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = sdbt1)  
    )  
  )
```

```
SAIDB =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.233)(PORT = 1521))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = saidb)  
    )  
  )
```

database link



Network - Tnsnames

```
SAIDB =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.233)(PORT = 1521))  
    (CONNECT_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE_NAME = saidb)  
    )  
  )
```

database link

Dblink – Network

TNSNAMES -saidb

IISTERNER - saidb

`$lsnrctl reload`

`$lsnrctl status`

`$tnsping saidb 2`

database link

Dblink

```
SQL> create database link sai_db_link  
2 connect to steve identified by oracle  
3 using 'saidb';
```

Database link created.

```
SQL> select * from test_tab@sai_db_link;
```

ID	NAME
1001	steve first line

Dblink

```
SQL> create synonym sai_tab for test_tab@sai_db_link;  
Synonym created.
```

```
SQL> insert into sai_tab values (1022,'sdbt through link');  
1 row created.
```

```
SQL> commit;  
Commit complete.
```

```
SQL> select * from test_tab;
```

ID	NAME
1001	steve first line
1022	sdbt through link

database link

Dblink - View

```
SQL> select db_link, username, host  
2 from user_db_links;
```

DB_LINK	USERNAME	HOST
SAI_DB_LINK	STEVE	saidb

Dbadb_links

database link

Dblink - Advantage

no persistent storage, Instead,it points to data stored remotely.

no need for data transformation from binary to text and no data validation.

usually other than Oracle database, is possible with other brand using Heterogeneous Services.

Dblink - Disadvantage

i/o contention increase

network traffic increase

source data captured

Materialized View



Materialized View

Materialized view;

A materialized view stores both the definition of a view and the rows resulting from the execution of the view.

Types
Read only materialized view
Updatable materialized view
Readwrite materialized view



Materialized View

Materialized View – privilege

```
SQL> show parameter db_name
```

NAME	TYPE	VALUE
-----	-----	-----
db_name	string	sdbt1

```
SQL> grant create materialized view to sdbt;
```

```
Grant succeeded.
```

```
SQL> show parameter db_name
```

NAME	TYPE	VALUE
-----	-----	-----
db_name	string	saidb

```
SQL> grant create materialized view to steve;
```

```
Grant succeeded.
```



Materialized View

Materialized View - create

```
SQL> create materialized view tab_mv  
2 refresh fast with primary key  
3 as select * from test_tab@sai_db_link
```

Materialized view created.

```
SQL> create materialized view log on test_tab;  
  
Materialized view log created.
```

```
SQL> select * from tab;
```

TNAME	TABTYPE	CLUSTERID
SAI_TAB	SYNONYM	
TAB_MV	TABLE	
TEST	TABLE	

```
SQL> select * from tab;
```

TNAME	TABTYPE	CLUSTERID
MLOG\$_TEST_TAB	TABLE	
RUPD\$_TEST_TAB	TABLE	
TEST_TAB	TABLE	

Materialized View

Materialized View

```
SQL> desc tab_mv;
Name          Null?   Type
-----
ID            NOT NULL NUMBER(5)
NAME                          VARCHAR2(20)
```

```
SQL> select * from tab_mv;
```

```
      ID NAME
-----
    1001 steve first line
    1022 sdbt through link
```



Materialized View

Materialized View

Materialized view

```
SQL> select * from tab_mv;
```

```
      ID NAME
```

```
-----
```

```
1001 steve first line
```

```
1022 sdbt through link
```

Dblink synonym

```
SQL> select * from sai_tab;
```

```
      ID NAME
```

```
-----
```

```
1001 steve first line
```

```
1003 steve next line
```

```
1004 steve mv line
```

```
1022 sdbt through link
```

```
SQL> insert into test_tab values  
2 (1003,'steve next line');
```

1 row created.

```
SQL> insert into test_tab values  
2 (1004,'steve mv line');
```

1 row created.

```
SQL> commit;
```

Commit complete.



Materialized View

Materialized View - Refresh

```
SQL> exec dbms_mview.refresh('TAB_MV');  
PL/SQL procedure successfully completed.
```

```
SQL> select * from tab_mv;
```

```
  ID NAME  
-----  
 1001 steve first line  
 1022 sdbt through link  
 1003 steve next line  
 1004 steve mv line
```

Refresh Method

Fast

Complete

Force

Refresh Mode

Auto

Manual

Materialized View

Materialized View - information

```
SQL> select mview_name, query, refresh_method  
2 from user_mviews;
```

```
MVIEW_NAME  
-----
```

```
QUERY  
-----
```

```
REFRESH_  
-----
```

```
TAB_MV
```

```
SELECT "TEST_TAB"."ID" "ID", "TEST_TAB"."NAME" "NAME"  
FROM "TEST_TAB"@ "SAI_DB_LIN  
FAST
```




Materialized View

Materialized View – Drop

```
SQL> drop materialized view tab_mv;  
Materialized view dropped.
```

```
SQL> revoke create materialized view from sdbt;  
Revoke succeeded.
```

```
SQL> drop materialized view log on test_tab;  
Materialized view log dropped.
```

```
SQL> revoke create materialized view from steve;  
Revoke succeeded.
```

Materialized View

Materialized View – Advanatage

Persistent storage

updated periodically using refresh.

Fast performance for select queries.

better perfomacne with complex joins

behave like indexes for a Data Warehouses.

Distributed Computing - To take snapshots of remote data.

Materialized view store snapshots of remotely located tables.

Mobile computing.



Thank you