

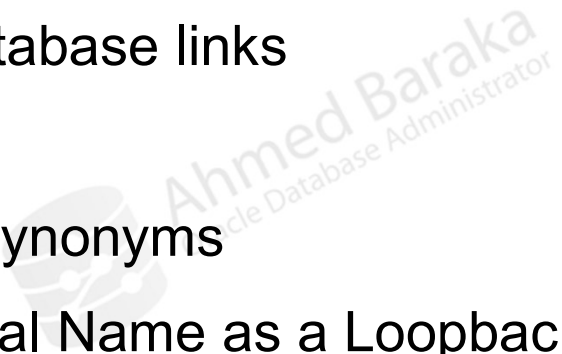
Using Database Links

By Ahmed Baraka

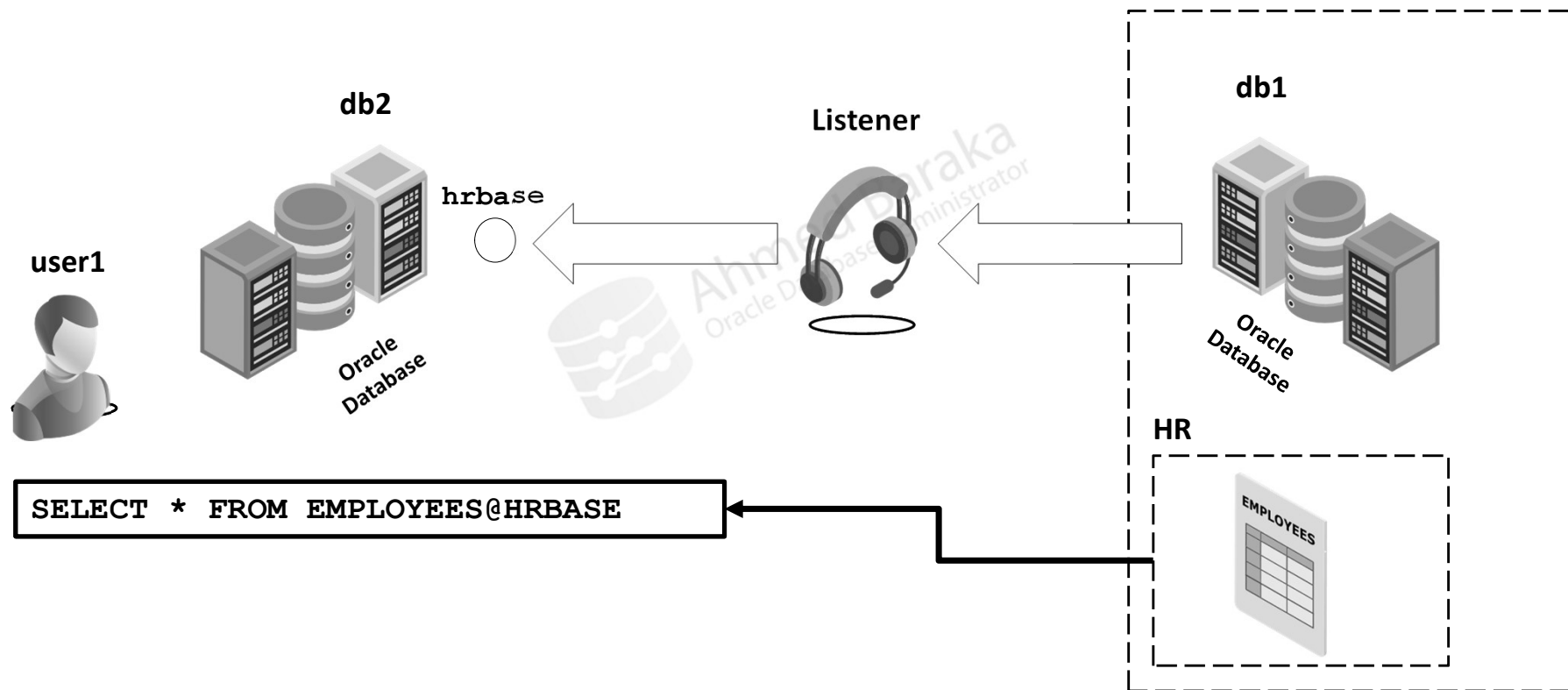
Objectives

In this lecture, you will learn how to perform the following:

- Describe database links, their types, and users
- Understand naming database links
- Create database links
- Use database links in synonyms
- Understand using Global Name as a Loopback Database Link



Database Link



About Database Links

- A database link is a pointer that defines a one-way communication path from an Oracle Database to another database
- It allows a local user to access a link to a remote object without having to be a user on the remote server.
- Database link types:

- **Public:** can be accessed by all database users

```
CREATE PUBLIC DATABASE LINK..
```

- **Private:** can be accessed by only its owner

```
CREATE DATABASE LINK..
```

- **Global:** defined in a directory service

Database Link Users

Command	Description
Connected user	Users connect as themselves. They must have an account on the remote database with the same user name and password as their account on the local database.
Fixed user	Users connect using the user name and password referenced in the link.
Current user	A user connects as a global user. A local user can connect as a global user in the context of a stored procedure, without storing the global user's password in a link definition.

Global Database Names in Database Links

- A database **global name** is formed using the following format:

```
[DB_NAME] . [DB_DOMAIN]
```

- Examples:

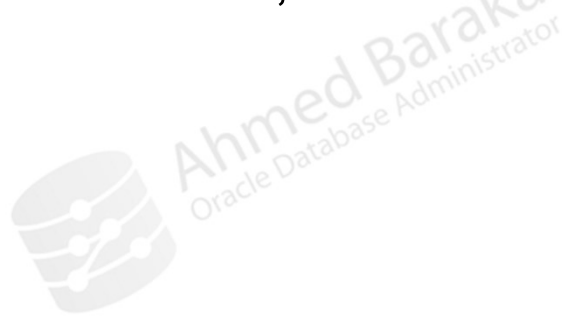
DB_DOMAIN	DB_NAME	Global Database Name
hr	myentity.com	hr.myentity.com
hr	us.myentity.com	hr.us.myentity.com
hr	myentity.org	hr.myentity.org

- To change it: **ALTER DATABASE RENAME GLOBAL_NAME**
- To retrieve the global name of the current database/pdb:

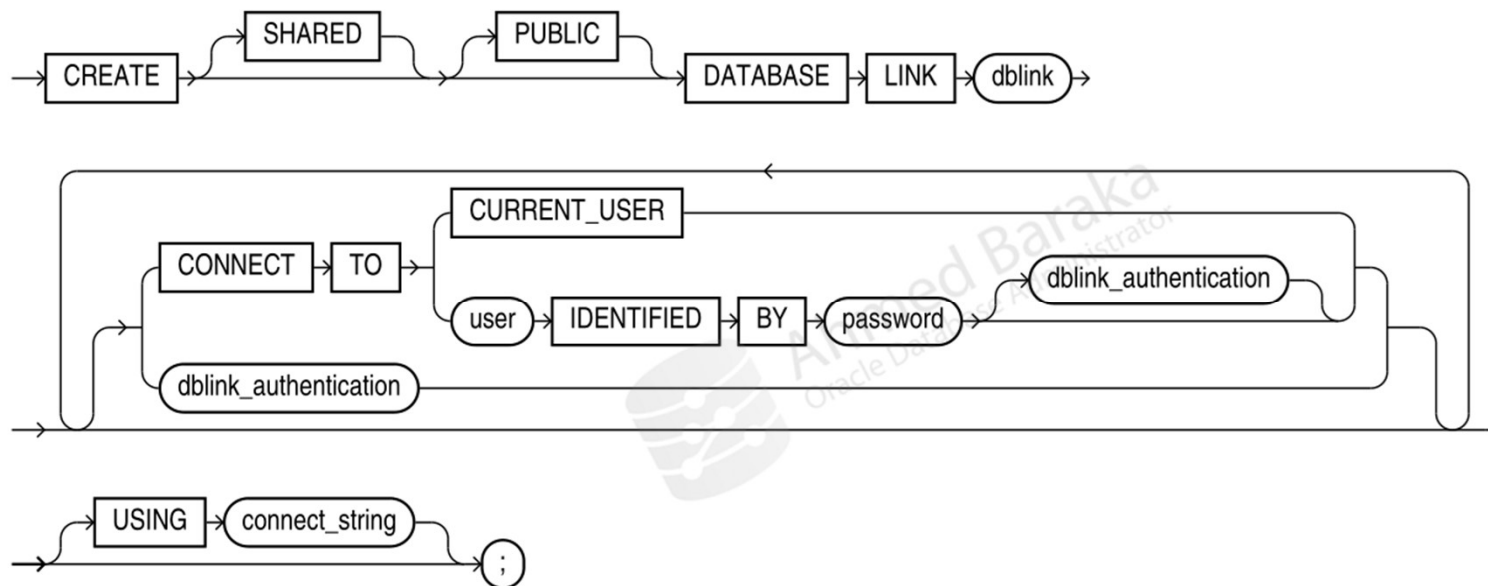
```
SELECT * FROM GLOBAL_NAME;
```

Naming Database Links

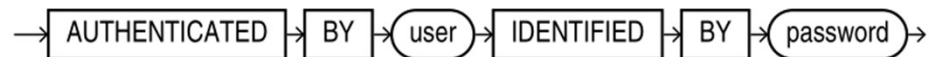
- If **GLOBAL_NAMES** is set to **TRUE**, the database link name must match the Global database name of the remote database (recommended)
- If **GLOBAL_NAMES** is set to **FALSE**, the database link must name can be set to any name.



Creating Database Links



dblink_authentication::=



Creating a Fixed-user Database Link

- Username/password is part of the database link
- Use the following **CREATE DATABASE LINK** syntax:

```
CREATE [PUBLIC] DATABASE LINK <db link name>  
CONNECT TO <user name> IDENTIFIED BY <password>  
USING '<connection string>';
```

- Example:

```
CREATE PUBLIC DATABASE LINK hrlnk  
CONNECT TO hr IDENTIFIED BY hispassword USING 'hr';
```

```
CREATE DATABASE LINK hrlnk  
CONNECT TO hr IDENTIFIED BY hispassword USING '(DESCRIPTION=  
  (ADDRESS=(PROTOCOL=TCP) (HOST=hrdb.myentity.com) (PORT=1521))  
  (CONNECT_DATA=(SERVICE_NAME=oradb.myentity.com)))';
```

Creating a Connected-user Database Link

- No specific username/password is specified in the database link
- The user connects to the remote database as itself
- Use the following **CREATE DATABASE LINK** syntax:

```
CREATE PUBLIC DATABASE LINK <db link name> USING '<connection name>';
```

- Example (when GLOBAL_NAMES is FALSE):

```
CREATE PUBLIC DATABASE LINK hr USING 'hr';
```

Using Database Links in Synonyms

- Accessing a remote object using its fully qualified name:

```
SELECT * FROM EMPLOYEES@HR.MYENTITY.COM
```

- Using synonyms:

```
CREATE SYNONYM EMP FOR EMPLOYEES@HR.MYENTITY.COM ;  
SELECT * FROM EMP;
```

- If the database link is public, the synonym the points to it should also be public:

```
CREATE PUBLIC SYNONYM EMP FOR EMPLOYEES@HR.MYENTITY.COM ;
```

Database Link Restrictions

- The following actions cannot be performed on database links:
 - Grant privileges on remote object
 - Execute **DESCRIBE** operations on remote objects, except the following object types which are supported by **DESCRIBE** command: tables, views, procedures, and functions
 - Analyze remote objects
 - Define or enforce referential integrity
 - Grant roles to users in a remote database
 - Obtain nondefault roles on a remote database.
 - Use a current user link without authentication through SSL, password, or Microsoft Windows native authentication

Using Global Name as a Loopback Database Link

- If in a SQL statement a database link is referenced and its name matches the Global Database name, it is effectively ignored.
- For example, if the current Global database name is **hr.myentity.com**, then the database link portion in the following statement is ignored:

```
SELECT * FROM HR.EMPLOYEES@hr.myentity.com;
```

Obtaining Information About Database Links

View	Description
DBA_DB_LINKS	Retrieves all database links in the database.
DBA_DB_LINK_SOURCES	Identifies all unique source databases that opened database links to the local database.



Documentation

- Oracle Database Administrator's Guide
 - 31.2. Database links
- SQL Language Reference



Summary

In this lecture, you should have learnt how to perform the following:

- Describe database links, their types, and users
- Understand naming database links
- Create database links
- Use database links in synonyms
- Understand using Global Name as a Loopback Database Link

