

Datenbanklinks – Ein Überblick für DBAs

Frank Gerasch
IT-P GmbH
Hannover

Schlüsselworte

Oracle Datenbank, Verteilte Transaktionen

Einleitung

Oracle Database Links sind eine hervorragende Möglichkeit, um Schnittstellen zwischen zwei Datenbanken aufzubauen oder Daten mit Hilfe von z.B. Materialized Views zwischen Oracle Datenbanken zu replizieren. Davon wird in Unternehmen mit verteilten Anwendungen und verteilter Datenhaltung häufig Gebrauch gemacht. In diesem Artikel werde ich einen Überblick über die unterschiedlichen Datenbanklink-Typen geben und die Anwendung von Datenbanklinks erläutern.

Oracle Datenbanklink

Ein Oracle Datenbanklink ist ein Objekt in der Datenbank, welches eine unidirektionale Verbindung zu einer anderen Datenbank ermöglicht. Die Definition des Datenbanklinks enthält mindestens den Namen des Datenbanklinks, des Eigentümers sowie Verbindungsinformationen zur Zieldatenbank. Je nach Typ ist auch der Name und das Passwort des Users abgespeichert, zu dem eine Verbindung hergestellt werden soll.

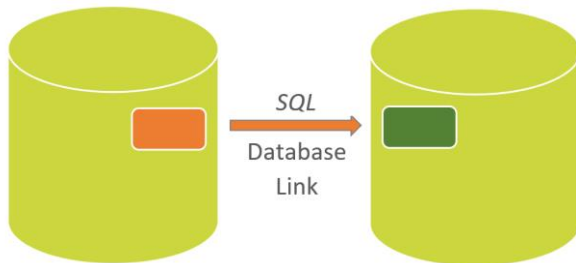


Abb. 1: Konzept Datenbanklink

Der große Vorteil von einem Datenbanklink ist, dass er Benutzern erlaubt auf Objekte anderer User in einer entfernten Datenbank mit den Privilegien des Objekteigentümers zuzugreifen. Mit anderen Worten, ein lokaler Benutzer kann auf eine entfernte Datenbank zugreifen, als wäre er der Eigentümer der Objekte in der entfernten Datenbank.

Zielbeschreibung

Die Verbindungsinformation zur Zieldatenbank wird üblicherweise als TNS-Aliasname angegeben, der mit Hilfe der SQL*Net Konfigurationsdatei tnsnames.ora in einen vollständigen Connect-String aufgelöst wird. Alternativ kann man aber auch den Connect-String selber im „Easy Connect“-Format bei der Erstellung des Datenbanklinks nutzen, um unabhängig von einer tnsnames.ora zu sein.

Beispielsyntax Verwendung Tnsnames-Alias:

```
CREATE DATABASE LINK <DB LINK NAME>  
CONNECT TO <REMOTE USER> IDENTIFIED BY <PASSWORD>  
USING <TNSNAMES-ALIASNAME>;
```

Beispielsyntax Verwendung Easy Connect:

```
CREATE DATABASE LINK <DB LINK NAME>  
CONNECT TO <REMOTE USER> IDENTIFIED BY <PASSWORD>  
USING ' //<SERVER>:<PORT>/<SERVICE_NAME>' ;
```

Public und Private Database Links

Ein Datenbanklink ist üblicherweise vom Typ „Public“ oder „Private“. Es gibt darüber hinaus den Typ „Global“, der einen Directory Server erfordert und für das ganze Netzwerk gilt, jedoch selten genutzt wird.

Ein Datenbanklink vom Typ Public gehört dem User Public, gilt datenbankweit und, wie bei allen Public-Objekten, steht dieser allen Benutzern und PL/SQL Programmen zur Verfügung. Daraus folgt, dass jeder Datenbankbenutzer damit auf die Objekte des definierten Users in der entfernten Datenbank zugreifen kann! Aus Sicherheitsgründen wird daher empfohlen, diesen Typ nur in Ausnahmefällen zu nutzen und lieber Datenbanklinks vom Typ „Private“ zu erstellen.

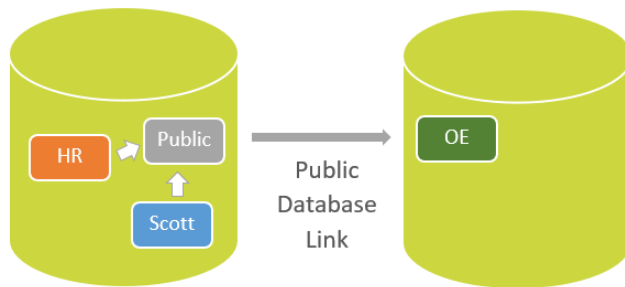


Abb. 2: Public Database Link

Ein Datenbanklink vom Typ „Private“ gehört einem bestimmten User und berechtigt auch nur diesen den Datenbanklink zu nutzen. Sprich, nur der Eigentümer des Datenbanklinks kann auf die Objekte in der entfernten Datenbank zugreifen.

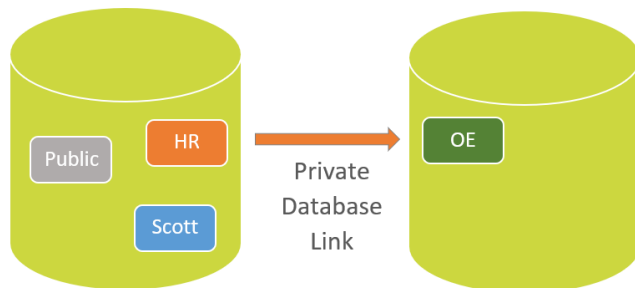


Abb. 2: Private Database Link

Authentifizierungstypen

Bei der Erstellung eines Datenbanklinks wird anhand der Syntax festgelegt, mit welchem Benutzer und welcher Authentifizierungsmethode sich bei der entfernten Datenbank angemeldet wird. Oracle unterscheidet zwischen drei verschiedenen Kategorien.

1. Fixed-User Database Link

Dieser am häufigsten verwendete Datenbanklink beinhaltet einen Usernamen und ein Passwort. Wenn ein solcher Datenbanklink genutzt wird, so wird sich mit dem bei der Erstellung angegebenen Benutzer und Passwort an die entfernte Datenbank verbunden. Da dieser Datenbanklink-Typ auch das Passwort enthält, wirken sich Passwortänderungen in der entfernten Datenbank unmittelbar auf die Funktionsfähigkeit des Datenbanklinks aus. Wenn das Passwort in der entfernten Datenbank geändert wird, so muss der Datenbanklink mit dem neuen Passwort neu erstellt werden. Ein großer Vorteil und sicher auch der Grund für die häufige Verbreitung dieses Typs ist, dass man einem Benutzer mit einfachen Mitteln Zugriff auf sämtliche Objekte eines anderen Benutzers in dessen Kontext erteilen kann.

Beispiel Private Fixed-User Database Link:

```
CREATE DATABASE LINK ORCL2
CONNECT TO OE IDENTIFIED BY secretpassword
USING 'ORCL2';
```

2. Connected-User Database Link

Dies sind Datenbanklinks, die bezüglich des Namens und Passworts einen gleichen User in der entfernten Datenbank erfordern. Bei der Erstellung des Datenbanklinks wird kein User-Name und kein Passwort angegeben. Die Verbindung zur entfernten Datenbank wird mit dem User hergestellt, der den Datenbanklink aufruft. Wenn z.B. der User SYSTEM auf einen Public Datenbanklink zugreift, dann ist der Connected-User SYSTEM und die Verbindung wird zum Schema SYSTEM der entfernten Datenbank hergestellt. Ein Connected-User ist jeder User, der den Datenbanklink aufrufen kann. Passwortänderungen des Nutzers müssen immer auf beiden involvierten Datenbanken identisch und zeitgleich erfolgen, um die Funktionsfähigkeit zu erhalten.

Beispiel Private Connected-User Database Link:

```
CREATE DATABASE LINK ORCL2_OE
USING 'ORCL2';
```

3. Current-User Database Link

Ein solcher Datenbanklink erlaubt es Enterprise-Benutzern (Global Users) auf Objekte einer entfernten Datenbank zuzugreifen, ohne dass Authentifizierungsinformationen beim Aufruf des Datenbanklinks übergeben werden. Bei der Erstellung werden ebenfalls kein User und kein Passwort angegeben. Voraussetzung sind SSL- oder Passwort-authentifizierte Enterprise-Benutzer, die eine entsprechend eingerichtete Infrastruktur (z.B. LDAP Directory) erwarten und die in beiden involvierten Datenbanken angelegt sein müssen. Current-User Datenbanklinks dürfen aufgrund der notwendigen "Enterprise"-Authentifizierung nur mit der Oracle Advanced Security Option verwendet werden.

Beispiel Current-User Database Link:

```
CREATE DATABASE LINK ORCL2_OE CONNECT TO CURRENT_USER USING 'ORCL2';
```

Beispiele der Anwendung von Datenbanklinks

a.) Zugriff auf Daten einer entfernten Datenbank

```
SELECT COL1 FROM [SCHEMA.]REMOTE_TABLE@DB_LINK;
```

b.) Verteilte Remote-Abfrage von Daten

```
SELECT A.COL1
FROM TABLE A, REMOTE_TABLE@DB_LINK B
WHERE A.COL1 = B.COL2;
```

c.) Ausführung entfernter PL/SQL Programme

```
SELECT [SCHEMA.]FUNCTION@DB_LINK@DB_LINK:
EXEC [SCHEMA.]PROCEDURE@DB_LINK({PARAMETERES});
EXEC [SCHEMA.]PACKAGE.PROCEDURE@DB_LINK([PARAMETERS]);
```

d.) Replizierung von Daten via MView

```
CREATE MATERIALIZED VIEW MY_MVIEW ... AS
SELECT * FROM REMOTE_TAB@DB_LINK;
```

Datenbanklink-Name

Der Name eines Datenbanklinks sollte eindeutig sein und darf eine Länge von 128 Bytes nicht überschreiten. Typischerweise ist der Name identisch mit dem Datenbanknamen oder dem Service Namen der Zieldatenbank.

Der Name ist frei wählbar, sofern der Datenbankparameter GLOBAL_NAMES auf FALSE (default) gesetzt ist. Wenn der Parameter auf TRUE gesetzt ist, muss der Name des Datenbanklinks identisch mit dem Global Name der entfernten Datenbank sein.

```
SELECT * FROM GLOBAL_NAME;
```

Datenbanklink-Restriktionen

Folgende Funktionen sind bei der Verwendung von Datenbanklinks nicht möglich:

- Zuweisung von Objekt-Privilegien auf Remote-Objekte
- Ausführung von DESCRIBE-Operationen auf einige Remote-Objekte
- Analyse von Remote-Objekten
- Definition Referentieller Integrität zwischen lokalen und Remote-Objekten
- Zuweisung von Rollen an User einer Remote-Datenbank
- Ausführung von Hash Query Joins, die Shared User Connections nutzen

Datenbanklink-Parameter

Der Parameter OPEN_LINKS gibt die maximale Anzahl gleichzeitig geöffneter Verbindungen zu entfernten Datenbanken in einer Session an. Diese Verbindungen umfassen Datenbankverbindungen sowie externe Prozeduren, von denen jede einen separaten Prozess verwendet.

Beide Verbindungstypen werden geschlossen, wenn die Sitzung beendet wird. Datenbankverbindungen über Datenbanklinks können allerdings auch explizit geschlossen werden,

indem man den Befehl Anweisung ALTER SESSION CLOSE DATABASE LINK in der Session ausführt.

Wenn der Parameter OPEN_LINKS auf 0 gesetzt wird, so wird die Verwendung von Datenbanklinks deaktiviert. Folglich sind keine verteilten Transaktionen mehr möglich.

Der Parameter OPEN_LINKS_PER_INSTANCE ist eine analoge Entsprechung auf der Instanz-Ebene, um z.B. in einzelnen RAC-Instanzen unterschiedliche Konfiguration umzusetzen.

Kontaktadresse:

Frank Gerasch

IT-P GmbH

Seligmannallee 6

D-30173 Hannover

Telefon: +49 (0) 511-616804-270

E-Mail f.gerasch@it-p.de

Internet: www.it-p.de