



Transbase® D

Als Option zu Transbase® lässt sich Transbase® D nahtlos mit den Varianten Transbase® CD und Transbase® Embedded kombinieren. Transbase® D erlaubt die verteilte Ausführung von SQL-Queries auf mehreren Transbase® Datenbanken.

Der Einsatz von Transbase® D bietet im Rahmen des Betriebes der Anwendung folgende Vorteile:

- Datenbanken können in eine lokale Datenbank und in eine zentrale Datenbank aufgeteilt werden
- Datensätze können aus einer Server-Datenbank in eine Client-Datenbank übertragen werden. Lokale Daten können ergänzt oder über Download aktualisiert werden.
- Datensätze können in eine zentrale Server-Datenbank übertragen werden. Veränderungen können über einen Upload übermittelt werden und zentral weiter verarbeitet werden.
- Logischen Datenbanken können in eine reine Read-Only-Datenbank und in eine Read-Write-Datenbank aufgeteilt werden. Queries können so auf diesen beiden Datenbanken prozessiert werden.

Voraussetzungen für den Einsatz

Alle an einer verteilten Query beteiligten Datenbanken müssen auf einer Transbase® Datenbank basieren. Benötigt wird eine Transbase® Version ab 6.1.

Alle Datenbanken müssen die gleichen Authorisierungsdaten, also ein gleiches Login und ein gleiches Passwort aufweisen.

Umsetzung und Realisierung

Eine verteilte Anfrage wird von der empfangenden Datenbank zunächst mit dem lokalen Data Dictionary verarbeitet. Entfernte Tabellen oder Views können wie lokale Tabellen mittels der Syntax:

```
<table_or_view_name>@<dbname>@<hostname>
```

referenziert werden.



Bei einer solchen Referenz baut Transbase® eine Connection zu einer entfernten Datenbank auf und holt zunächst alle Data Dictionary Daten zu dieser Tabelle oder diesem View. Danach wird die verteilte Anfrage so aufgeteilt, dass eine maximale Anfrage dort prozessiert werden kann. Insbesondere unter Einschluss aller lokalen Restriktionen oder unter Einschluss entfernter Sortieroperationen). Der auf die entfernte Datenbank entfallende Teil der Anfrage wird in Form eines Operatorbaums dorthin verschickt, der ohne weitere Übersetzung dort prozessiert werden kann.

Die Abarbeitung der verteilten Query wird durch die empfangende Datenbank gesteuert. Gegebenenfalls wird dazu die lokale Transaktion automatisch auf die entfernten Datenbanken verteilt. Die Connection zu einer verteilten Datenbank bleibt mindestens bis zum Ende der Transaktion bestehen.

Falls Transbase® für parallele Queryverarbeitung konfiguriert ist, werden die entfernten Anfragen asynchron verarbeitet, so dass lokale Verarbeitung und entfernte Verarbeitung parallel stattfinden.

Anwendungsbeispiele

Ein Download könnte so formuliert sein:

```
INSERT INTO t
SELECT *
FROM t@db@host JOIN last_update
WHERE timestamp > last_update.timestamp
```

Ein Upload beispielsweise eine Bestellung, könnte so formuliert werden:

```
INSERT INTO order@db@host
SELECT ? as custno, 15 as qty, partno, partdescription, ...
FROM parts
WHERE partno=?
```

Ein verteilter Join über eine lokale Tabelle parts und eine entfernte Tabelle prices@db@host könnte folgendermaßen aussehen:

```
SELECT price, available_items, parts.*
FROM parts JOIN prices@db@host USING partno
JOIN available@db@host USING partno
WHERE partdesc LIKE 'SCREW%'
```

In diesem Fall wird das LIKE-Prädikat auf der lokalen Tabelle verarbeitet, und die sich daraus ergebende partno-Menge wird mittels Verteilung an die entfernte Datenbank db geschickt, in der die Felder price und available_items ermittelt werden.

Die Transaktion wird dabei automatisch verteilt und – wenn nötig mittels 2-Phasen-Commit abgeschlossen.

Kontakt

Transaction Software GmbH
Willy-Brandt-Allee 2
81829 München

Tel.: +49 89 / 627 09 - 0
Fax: +49 89 / 627 09 - 11

info@transaction.de
www.transaction.de
www.transbase.de