



## Tuning the Mobile Server

DOAG Konferenz  
20. - 22.11.2012

Philipp Loer  
ppl@ordix.de  
www.ordix.de

- Einleitung
- Grundlagen
- Performance-Optimierung
- Fazit

- Datenabgleich zwischen mobile Geräten und einer Oracle-Datenbank
- Clients haben eine eigene Datenbank: sqlite oder Oracle Berkeley DB
- Oracle Mobile Server als Middleware zuständig für die Organisation des Datenabgleichs
- Ziel des Vortrags: Vorstellung von Möglichkeiten des Performancetuning



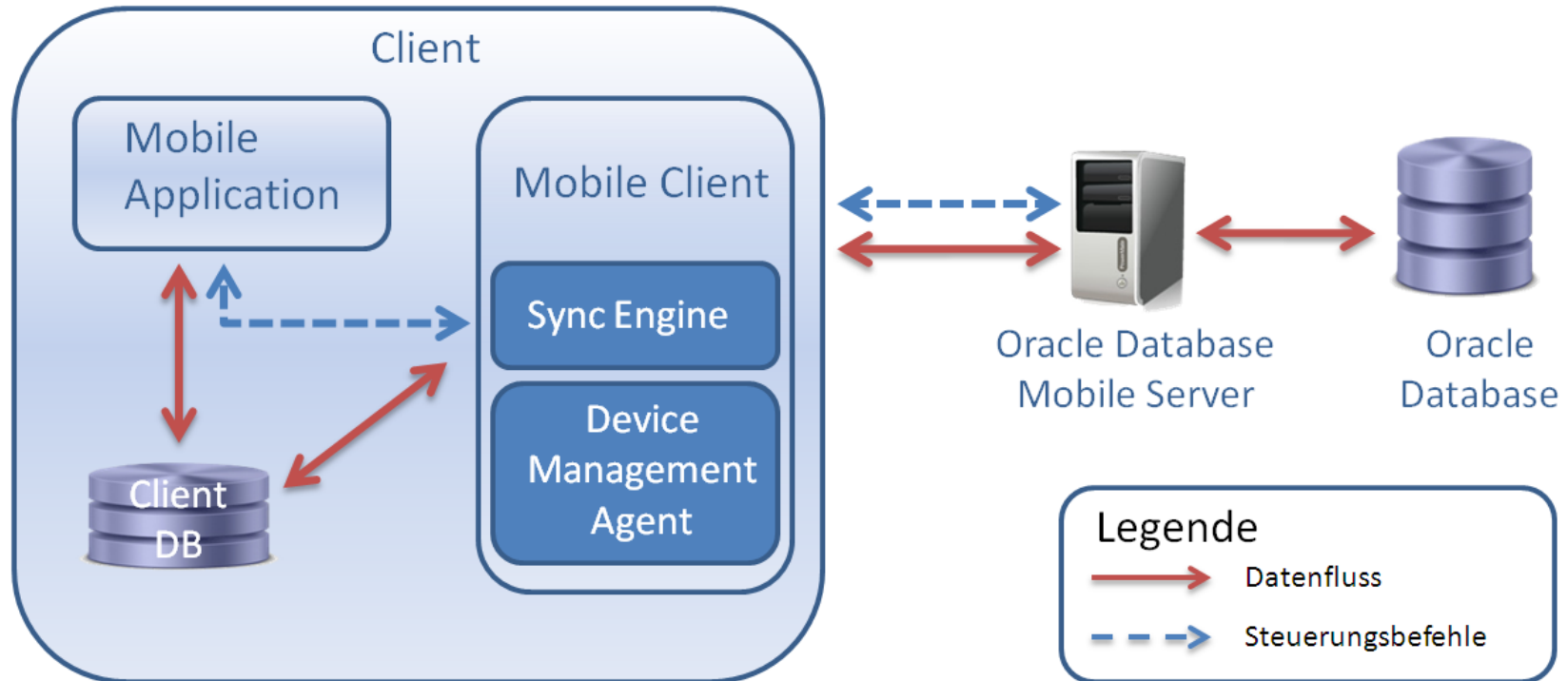
- Datenabgleich zwischen mobile Geräten und einer Oracle-Datenbank:



- Einleitung
- Grundlagen
- Performance-Optimierung
- Fazit

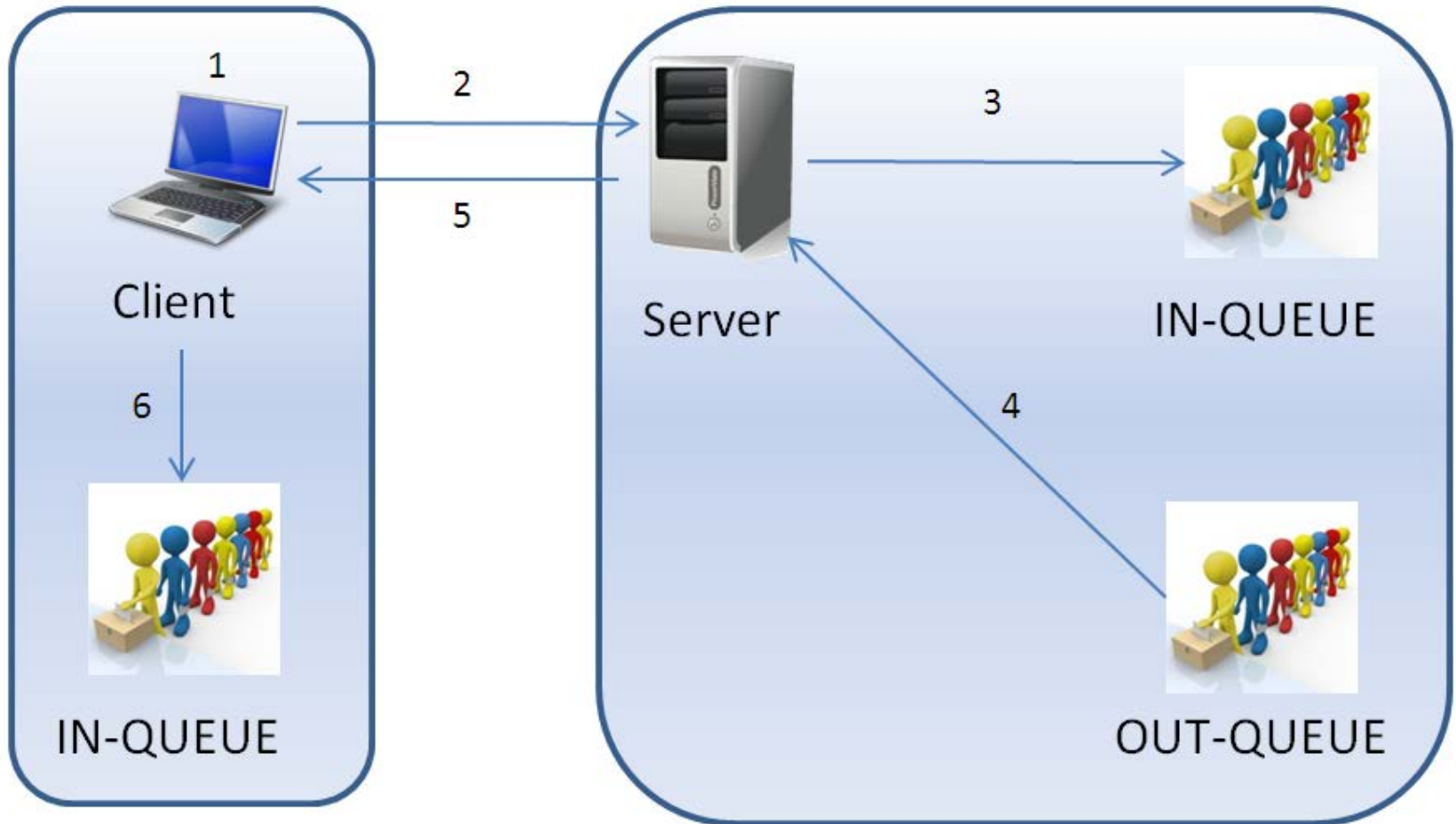


- erste Version 2002 als Oracle Lite 9i
- aktuell: Oracle Database Mobile Server 11g, verfügbar seit März 2011

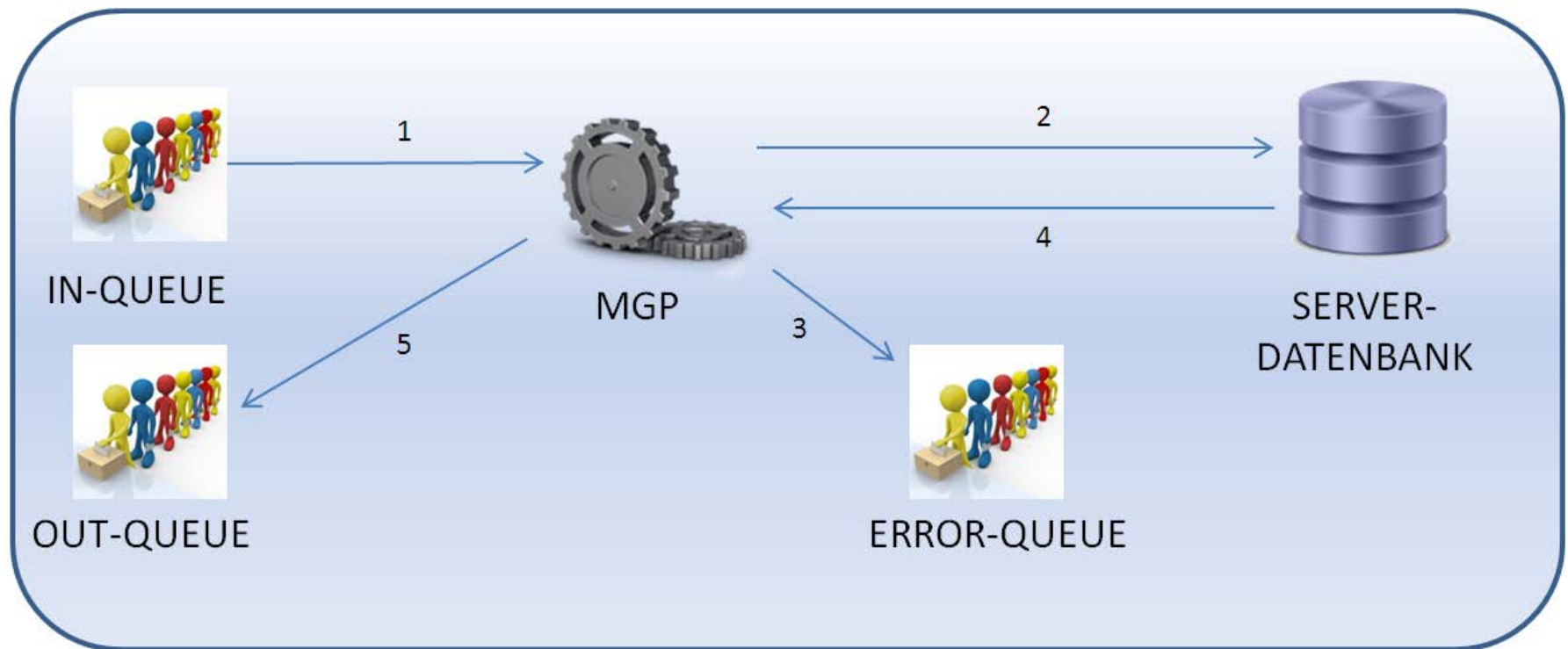


- Eine Publikation enthält
  - Publikationselemente
  - Sequenzen
  - Skripte
- Queues
  - Abarbeitung der Transaktionen und Replikationen nach der Datenübertragung
  - Verhindern die Verlängerung des Datenabgleichs durch Datenbanksperren



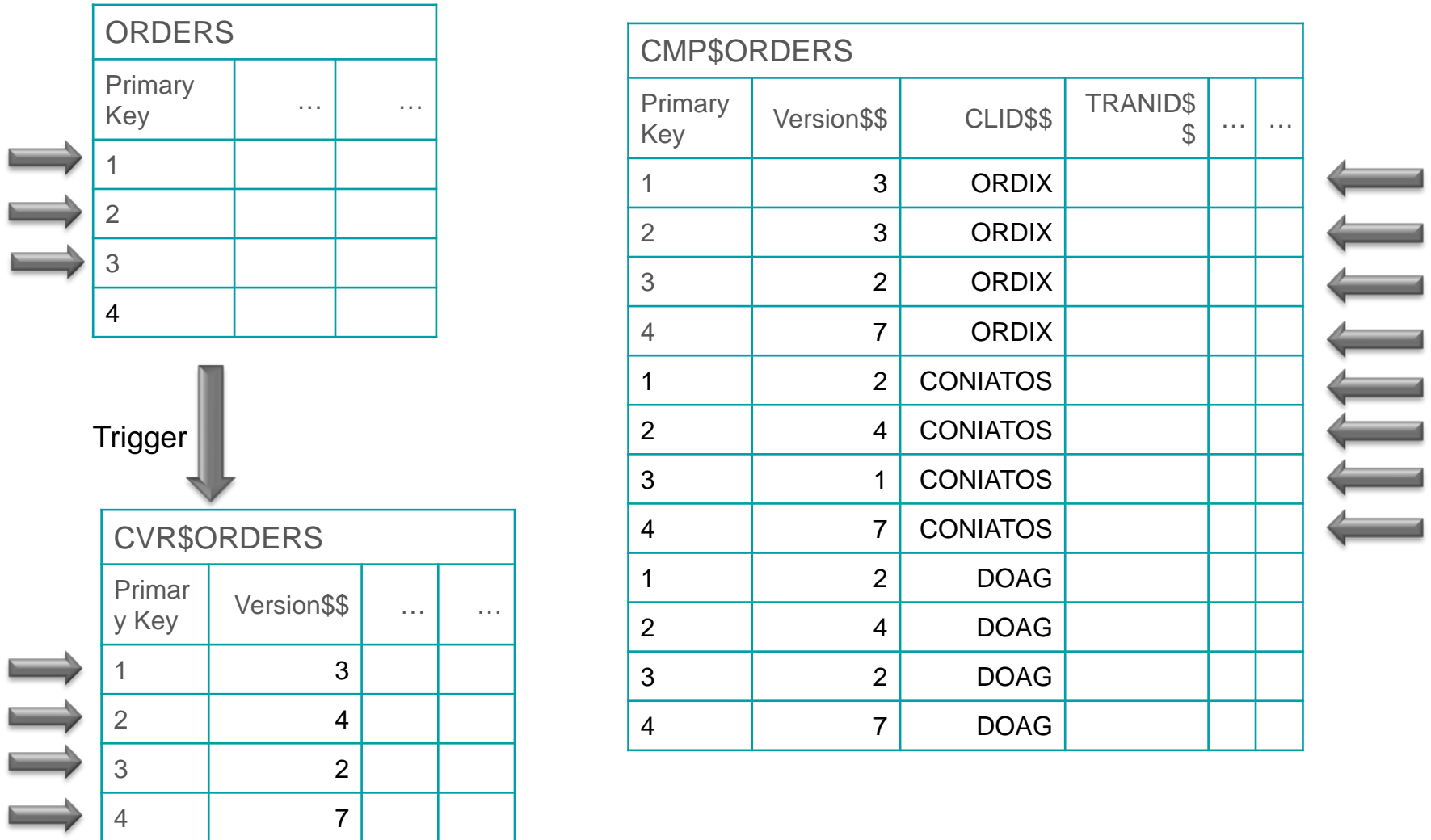






## Logging-Tabellen eines Publikationselementes

<b>Tabelle</b>	<b>Inhalt / Aufgabe</b>	<b>Prefix</b>
Originaltabelle	Masterreplik	-
Versionierungstabelle	Logging der serverseitigen Änderungen	CVR\$
Logging-Tabelle	Logging der übernommenen Änderungen	CLG\$
Mapping-Tabelle	Versionsinfo über die Datenstände der Clients	CMP\$
In-Queue	Von den Clients übergebene, noch nicht verarbeitete Änderungen	CFM\$
Error-Queue	Konfliktbehaftete Transaktionen	CEQ\$



- Einleitung
- Grundlagen
- Performance-Optimierung
- Fazit



### Physikalische Datenorganisation

- Die Mapping-Tabelle enthält die Informationen, welcher Client welche Version eines Datensatzes hat.
- Bei einem Datenabgleich werden immer alle „Log-Informationen“ eines Clients benötigt.



**Die Datensätze der Mapping-Tabelle zu einem bestimmten Client werden auf möglichst wenige Blöcke verteilt.**

Tabelle ORDERS vorher

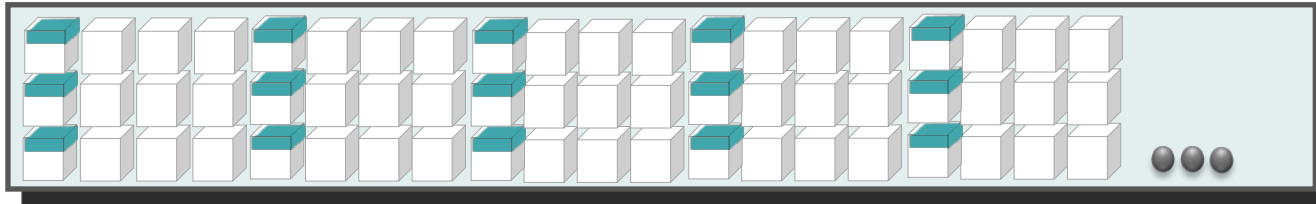
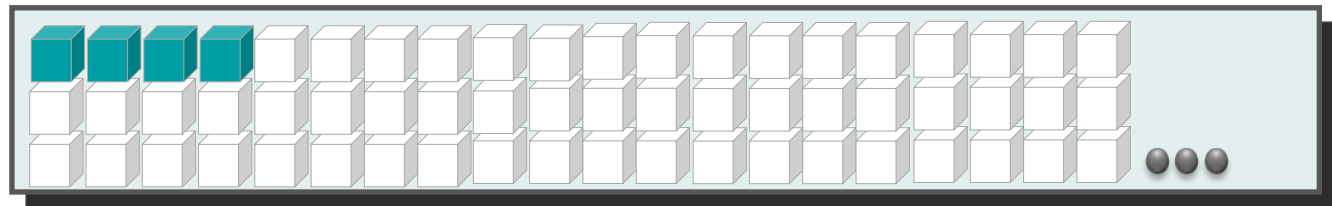


Tabelle ORDERS nachher



Die Datensätze der Mapping-Tabelle zu einem bestimmten Client werden auf möglichst wenige Blöcke verteilt.

## Umsetzung:

- Offline:

```
CREATE TABLE interim_table_name AS SELECT FROM table_name ORDER BY column_name;  
DROP TABLE table_name;  
ALTER TABLE interim_table_name RENAME TO table_name;
```

- Online:

```
CREATE TABLE interim_table_name AS SELECT FROM table_name WHERE 1 = 2;  
exec dbms_redefinition.start_redef_table  
exec dbms_redefinition.copy_table_dependents  
exec dbms_redefinition.finish_redef_table  
exec dbms_stats.gather_table_stats  
DROP TABLE table_name;
```

Elapsed: 00:00:24.45

Id	Operation	Name
0	SELECT STATEMENT	
* 1	HASH JOIN	
2	TABLE ACCESS FULL	CLG\$ORDERS
* 3	TABLE ACCESS FULL	CMP\$ORDERS

### Statistics

```
0 recursive calls
0 db block gets
35902 consistent gets
0 physical reads
0 redo size
282 bytes sent via SQL*Net to client
409 bytes received via SQL*Net from client
1 SQL*Net roundtrips to/from client
0 sorts (memory)
0 sorts (disk)
0 rows processed
```



Elapsed: 00:00:08.01

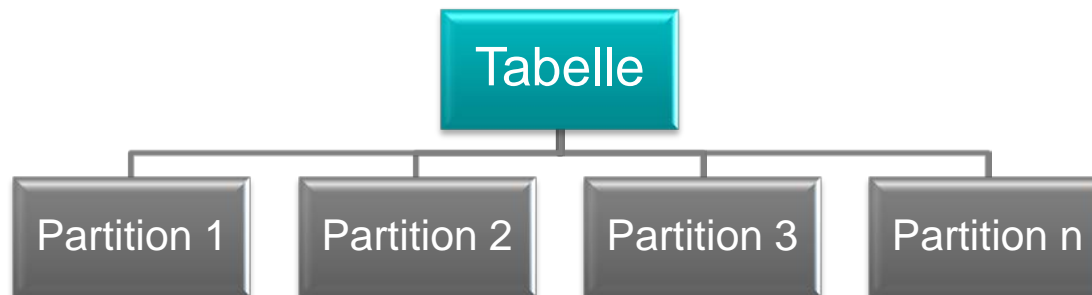
Id	Operation	Name
0	SELECT STATEMENT	
* 1	HASH JOIN	
2	TABLE ACCESS FULL	CLG\$ORDERS
* 3	TABLE ACCESS BY INDEX ROWID	CMP\$ORDERS
* 4	INDEX RANGE SCAN	CMP\$ORDERS_IND1

### statistics

```
1 recursive calls
0 db block gets
1669 consistent gets
0 physical reads
0 redo size
282 bytes sent via SQL*Net to client
409 bytes received via SQL*Net from client
1 SQL*Net roundtrips to/from client
0 sorts (memory)
0 sorts (disk)
0 rows processed
```

## Partitionierung

- horizontale Teilung einer Tabelle
- hier: Hash-Partitionierung nach der Client-ID



Die Datensätze der Mapping-Tabelle zu einem bestimmten Client werden automatisch auf möglichst wenige Blöcke verteilt.

Elapsed: 00:00:02.45

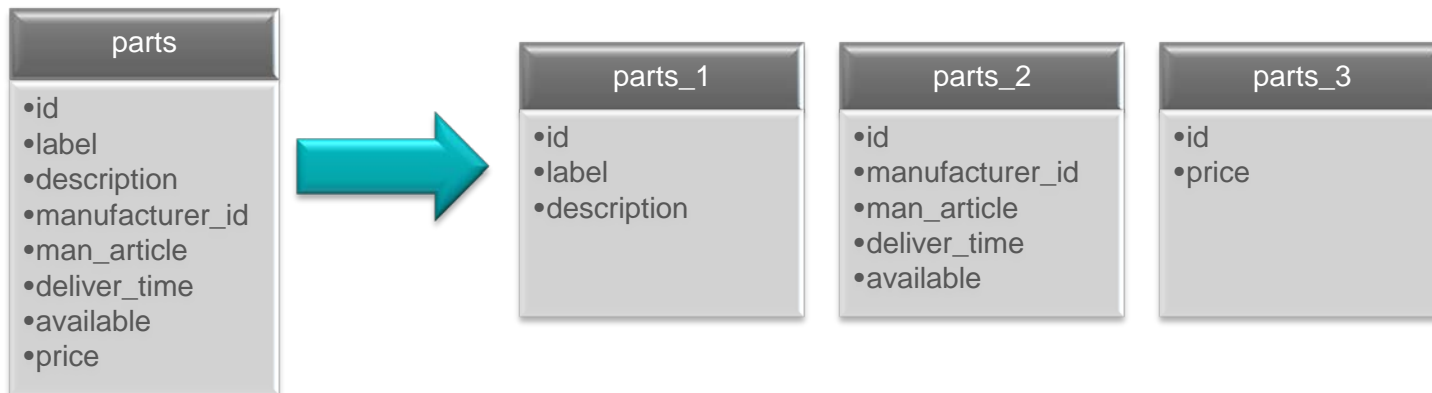
Id	Operation	Name
0	SELECT STATEMENT	
* 1	HASH JOIN	
2	TABLE ACCESS FULL	CLG\$ORDERS
3	PARTITION HASH SINGLE	
* 4	TABLE ACCESS FULL	CMP\$ORDERS

### Statistics

```
0 recursive calls
0 db block gets
291 consistent gets
0 physical reads
0 redo size
282 bytes sent via SQL*Net to client
409 bytes received via SQL*Net from client
1 SQL*Net roundtrips to/from client
0 sorts (memory)
0 sorts (disk)
0 rows processed
```

### Vertikale Partitionierung

- Wird ein Datensatz geändert, so wird immer der ganze Datensatz übertragen.
- Durch eine vertikale Partitionierung kann die zu übertragene Datenmenge reduziert werden.



### Testergebnisse vertikale Partitionierung

	Geänderte Datensätze	Vertikale Partitionierung	Abfrage Mapping-Tabelle	Gesamtdauer
1	0	Nein	72 ms	3,8 s
2	0	Ja	224 ms	3,9 s
3	5000	Nein	109 ms	13,9 s
4	5000	Ja	343 ms	6,0 s



### Shared Maps

- Datenänderung nur auf der Master-Seite erlaubt
- Nachteil: Keine Offline-Transaktionen möglich
- Vorteil: Verminderter „Verwaltungsaufwand“  
Mapping-Tabelle speichert nicht mehr, welcher Client welchen Datenstand hat, sondern nur einen Versionszähler für jeden Datensatz



**Mapping-Tabelle speichert statt einer 1:n-Beziehung  
nur noch 1:1-Beziehungen**

### Testergebnisse Shared Maps

	Geänderte Datensätze	Shared Maps eingesetzt	Abfrage Mapping-Tabelle in ms	Gesamtdauer in s
1	0	nein	985	4,6
2	0	ja	72	3,8
3	5000	nein	1108	13,1
4	5000	ja	109	13,9



## Begrenzung der Datenmenge

- Nicht immer benötigt jeder Client alle Datensätze eines Publikationselementes.
- Durch eine Änderung der Base-Query kann diese angepasst werden
  - Vorherige Base-Query:

```
SELECT * FROM contact_persons ;
```

- Nachher:

```
SELECT cp.*
FROM employees emp , emp_areas ea, areas a, customers c,
contact_persons cp
WHERE emp.id = ea.emp_id
      AND ea.area_id = a.id
      AND a . zipcode = c .zipcode
      AND c.id = cp.customer_id
      AND emp.client_id =:client_id;
```



### Testergebnisse Begrenzung der Datenmenge

	Anzahl übertragener Datensätze	Datenbegrenzung	Abfrage der Mapping-Tabelle	Gesamtdauer	übertragene Datenmenge
1	5000	Ja	120 ms	10,4 s	1,2 MB
2	25000	Ja	312 ms	34,2 s	5,5 MB
3	50000	Ja	374 ms	66,5 s	10,1 MB
4	50000	Nein	369 ms	65,0 s	10,2 MB



## Netzwerkgeschwindigkeit

- Ziel : Ermittlung des Performance-Einflusses der Netzwerkgeschwindigkeit
- Für einen Datenabgleich müssen die geänderten Daten ermittelt, übermittelt und auf der anderen Seite verarbeitet werden.
- Nur für die Übermittlung der Daten ist die Netzwerkgeschwindigkeit von Bedeutung.

### Testergebnisse Netzwerkgeschwindigkeit

	Netzwerkverbindung	Geänderte Datensätze	Übertragungsdauer	übertragene Datenmenge
1	10 Mbit	0	2,4 s	0,2 MB
2	100 Mbit	0	2,0 s	0,2 MB
3	10 Mbit	1.000	13,3 s	0,8 MB
4	100 Mbit	1.000	11,4 s	0,9 MB
5	10 Mbit	5.000	19,8 s	2,0 MB
6	100 Mbit	5.000	15,9 s	1,9 MB

- 100.000 Datensätze in der Tabelle ORDERS

## Weitere Optimierungsmöglichkeiten

- Base-Query Tuning
- Priority Based Replication
- Base-Query Caching
- Index Organized Tables (IOT)
- Partition Mapping
- Mobile.ora Parameter
  - Max Threads
  - Connection Pooling
- Synchronisation Tablespace Layout

- Einleitung
- Grundlagen
- Performance-Optimierung
- Fazit

- Physikalische Datenorganisation
- Horizontale Partitionierung
- Vertikale Partitionierung
- Shared Maps
- Datenmengenbegrenzung
- Weitere Tuningmöglichkeiten





Zentrale Paderborn  
Westernmuer 12 - 16  
33098 Paderborn  
Tel.: 05251 1063-0

Seminarzentrum Wiesbaden  
Kreuzberger Ring 13  
65205 Wiesbaden  
Tel.: 0611 77840-00

Zentrales Fax:  
0180 1 67349 0  
0180 1 ORDIX 0

Weitere Geschäftsstellen  
in Köln, Münster und Neu-Ulm

E-Mail: [info@ordix.de](mailto:info@ordix.de)  
Internet: <http://www.ordix.de>

**Vielen Dank für Ihre Aufmerksamkeit!**