



**ORACLE®**

## **Entwicklung von hochperformanten Anwendungen mit Database Gateways**

Thomas Niewel  
Leitender Systemberater



# Agenda

- Architektur
- Funktionsweise von Database Gateways –  
Unterstützung verschiedenster SQL Dialekte
- Performance Aspekte
- Zugriff auf Fremdsysteme am Beispiel DB2 for  
z/OS (Database Gateway for DRDA)
- Optimierung von Load Prozessen
- Weiteres

# Architektur

## Database Gateways

- Informationen
  - <http://www.oracle.com/technetwork/database/gateways/index.html>
  - Empfehlung: Version  $\geq$  11.2.03



# Oracle Database Gateways

- Oracle Open Systems Gateways
  - Database Gateway for ODBC
  - Database Gateway for Sybase
  - Database Gateway for SQL Server
  - Database Gateway for Teradata
  - Database Gateway for Informix

ORACLE<sup>®</sup> 11<sup>g</sup>  
DATABASE

ORACLE<sup>®</sup>



# Oracle Database Gateways

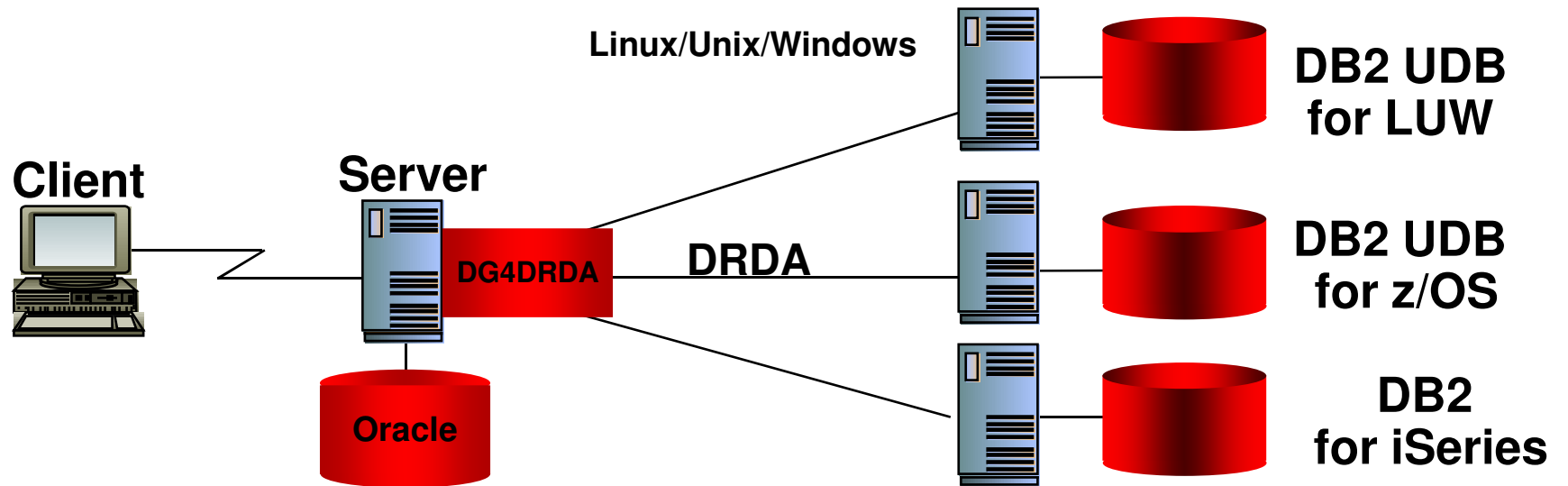
- Oracle Mainframe Gateways
  - Database Gateway for APPC
    - Procedural Gateway
  - Database Gateway for Websphere MQ
    - Procedural Gateway
  - Database Gateway for DRDA
    - “Transparent” Gateway

# Database Gateway for ODBC vs. “Spezielle” Gateways

	DG4ODBC	“Spezielle”
End to end solution	No	Yes
Distributed transactions	No	Yes
Stored procedures	No	Yes
Additional software	Yes	No
Enhanced DD translation	No	Yes

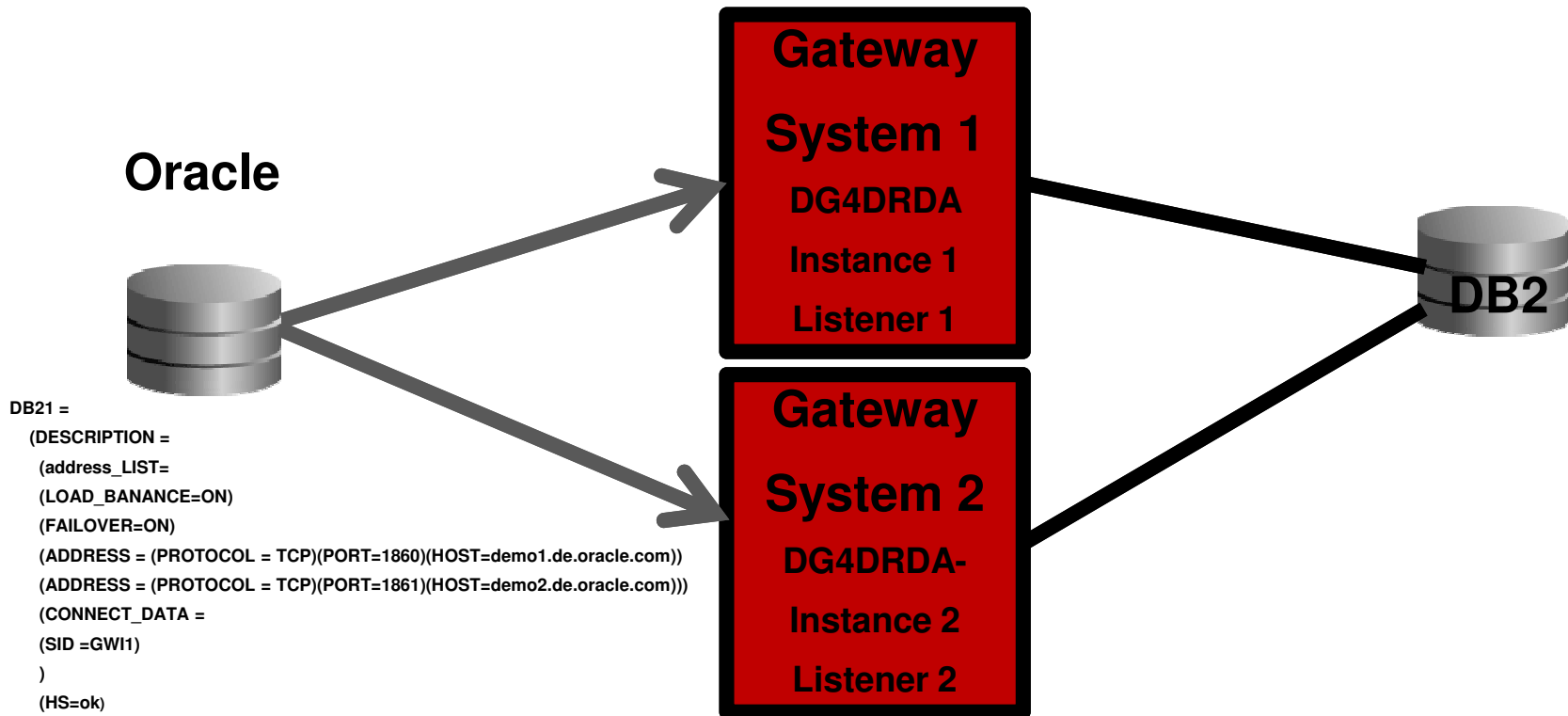
# Architektur

## Database Gateway for DRDA



ORACLE

# Oracle Database Gateways - Hochverfügbarkeit

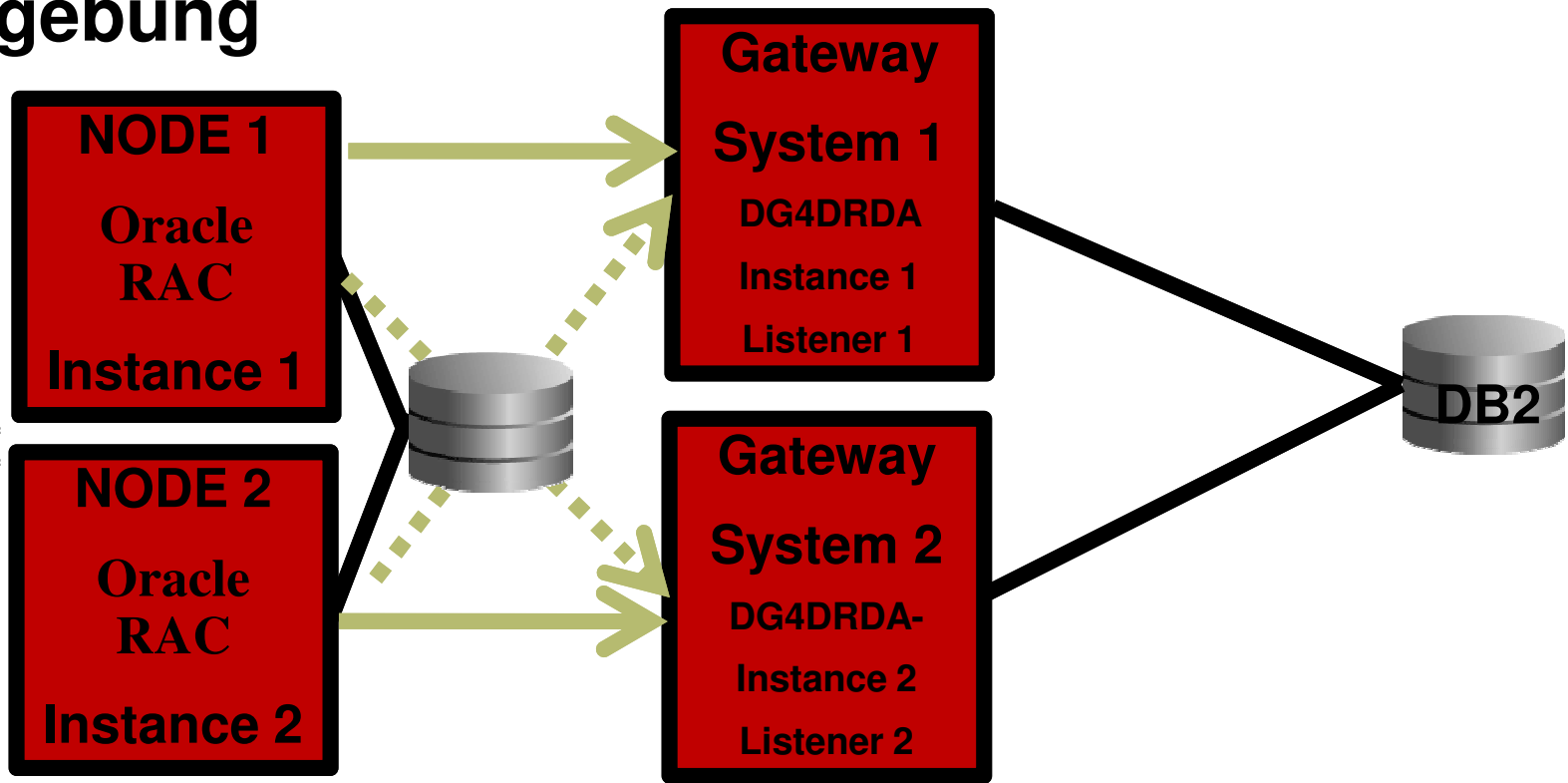


ORACLE



# Oracle Database Gateways in einer RAC Umgebung

```
DB21 =  
(DESCRIPTION =  
  (address_LIST =  
    (LOAD_BALANCE=ON)  
    (FAILOVER=ON)  
    (ADDRESS = (PROTOCOL =  
      (ADDRESS = (PROTOCOL =  
        (CONNECT_DATA =  
          (SID =GW11)  
        )  
      )  
    )  
  )  
(HS=ok
```

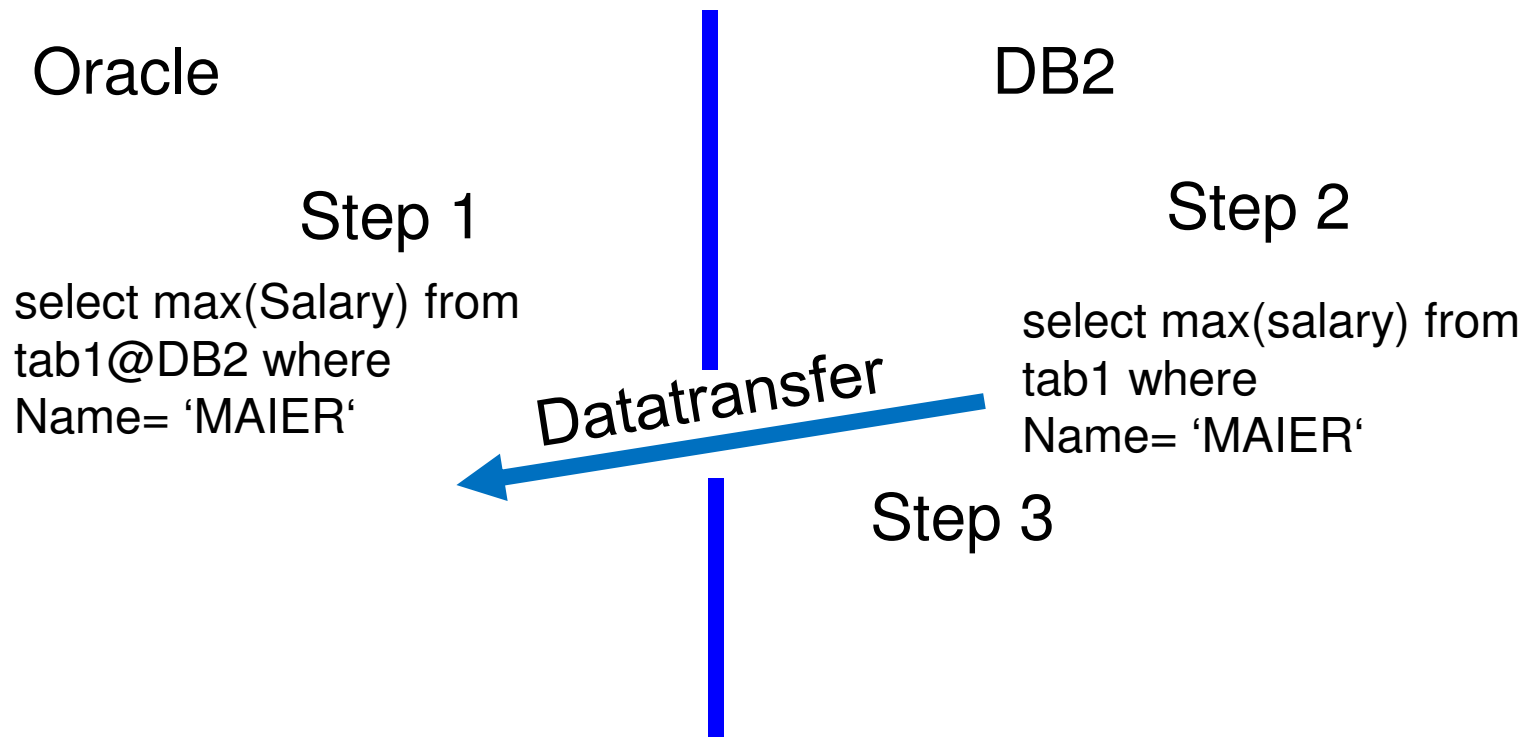




# Agenda

- Architektur
- Funktionsweise von Database Gateways –  
Unterstützung verschiedenster SQL Dialekte
- Performance Aspekte
- Zugriff auf Fremdsysteme am Beispiel DB2 for  
z/OS (Database Gateway for DRDA)
- Optimierung von Load Prozessen
- Weiteres

# Wie arbeiten Oracle Database Gateways? Compatible Functions



# Wie arbeiten Oracle Database Gateways? Translated Functions

Oracle

Step 1

```
select name,age from  
tab1@DB2 where  
nvl(age,18)= 18
```

DB2

Step 2

```
select name,age from  
tab1 where  
value(age,18)= 18
```

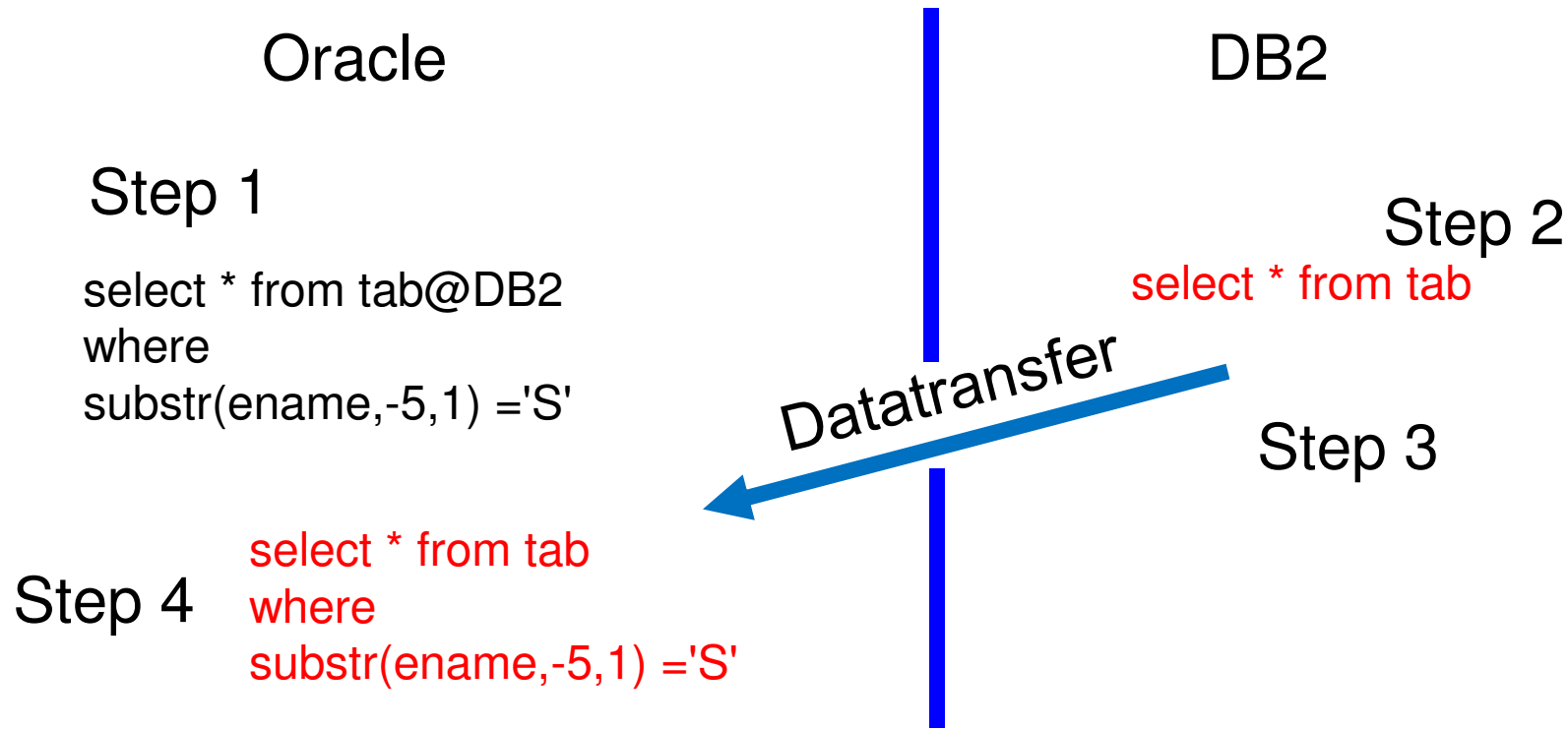
Step 3

Datatransfer



ORACLE

# Wie arbeiten Oracle Database Gateways? Compensated Functions



ORACLE



# Agenda

- Architektur
- Funktionsweise von Database Gateways –  
Unterstützung verschiedenster SQL Dialekte
- Performance Aspekte
- Zugriff auf Fremdsysteme am Beispiel DB2 for  
z/OS (Database Gateway for DRDA)
- Optimierung von Load Prozessen
- Weiteres



## Performance Aspekte

- Compatible Functions
  - Gute Performance
- Translated Functions
  - Gute Performance
- Compensated Functions
  - Performance hängt von dem übertragenen Datenvolumen ab



## Performance Aspekte

- Native Semantics
  - HS\_FDS\_CAPABILITY={*FUNCTION*/{*ON*/*OFF*}}
    - ON=Funktionsunterschiede werden nicht mehr kompensiert
  - HS\_FDS\_CAPABILITY=SUBSTR/OFF (Default)
    - select \* from tniewel.emp1@ds3l where substr(ename,-3,1)='A'
      - Funktion wird auf Oracle Seite ausgeführt
  - HS\_FDS\_CAPABILITY=SUBSTR/ON
    - select \* from tniewel.emp1@ds3l where substr(ename,-3,1)='A'
      - Funktion wird auf DB2 Seite ausgeführt
      - ORA-28500: connection from ORACLE to a non-Oracle system returned this message:[Oracle][ODBC DB2 Wire Protocol driver][UDB DB2 for OS/390 and z/OS]SECOND OR THIRD ARGUMENT OF SUBSTR FUNCTION OUT OF RANGE. {HY000,NativeErr =-138}



# Performance Aspekte

Table 4-1 SQL Compatibility, by Oracle SQL function

Oracle SQL Function	Compatible	Translated	Compensated	Native Semantics Candidate
ABS	-	-	Yes	Yes
ACOS	-	-	Yes	Yes
ADD_MONTHS	-	-	Yes	
ASCII	-	-	Yes	Yes
ASIN	-	-	Yes	Yes
ATAN	-	-	Yes	Yes
ATAN2	-	-	Yes	Yes
AVG	Yes	-	-	-
BITAND	-	-	Yes	Yes
CAST	-	-	Yes	Yes
CEIL	-	CEILING	-	Yes
CHARTOROWID	-	-	Yes	-
CHR	-	-	Yes	Yes
CONCAT	Yes	-	-	-
CONVERT	-	-	Yes	Yes
COS	-	-	Yes	Yes
COSH	-	-	Yes	Yes
COUNT (*)	Yes	-	-	-
COUNT (DISTINCT colname)	Yes	-	-	-
COUNT (ALL colname)	Yes	-	-	COUNTCOL



# Performance Aspekte

- Debugging
  - SQL\_TRACE, tkprof
  - Gateway Trace
  - Explain Plan
    - Explain plan for select ....table@gateway
    - ```
SELECT * FROM Table  
(DBMS_XPLAN.DISPLAY('plan_table', null, 'ALL'));
```



## Compensated Functions - Beispiel

explain plan for

```
select b.ename, b.empno from tniewel.emp@epg1 a,  
scott.emp@epg1 b where a.empno=b.empno and  
soundex(a.ename)='MAIER'
```

/

# Compensated Functions - Beispiel

```
-----  
| Id | Operation          | Name          | Rows | Bytes | Cost | TQ  | IN-OUT | PQ Distrib |  
-----  
0	SELECT STATEMENT		1	40	1			
*  1	HASH JOIN		1	40	1			
*  2	FILTER		1	0	1			
3	REMOTE		1	0	1		E.,OM	SERIAL
4	REMOTE		1	20	1		E.,OM	SERIAL
-----
```

Predicate Information (identified by operation id):

- ```
-----  
1 - access("A2"."EMPNO"="A1"."EMPNO")  
2 - filter(SOUNDEX("A2"."ENAME")='MAIER')
```

Slave SQL Information (identified by operation id):

- ```
-----  
3 - SELECT "EMPNO", "ENAME" FROM "TNIWEL"."EMP"  
4 - SELECT "EMPNO", "ENAME" FROM "SCOTT"."EMP"
```

Note: cpu costing is off



## Compatible Functions - Beispiel

explain plan for

```
select b.ename, b.empno from tniewel.emp@epg1 a,  
scott.emp@epg1 b where a.empno=b.empno and  
a.ename='MAIER'
```

/



# Compatible Functions - Beispiel

```
-----  
| Id | Operation          | Name          | Rows | Bytes | Cost | TQ  | IN-OUT | PQ Distrib |  
-----  
|  0 | SELECT STATEMENT   |               |      |      |      |     |        |            |  
|  1 |  REMOTE            |               |      |      |      | E.,OM | SERIAL |            |  
-----
```

Slave SQL Information (identified by operation id):

```
-----
```

```
1 - SELECT A1."ENAME", A1."EMPNO" FROM "TNIWEL"."EMP" A2, "SCOTT"."EMP" A1 WHERE A2."EMP  
NO"=A1."EMPNO" AND A2."ENAME"='MAIER'
```





## Distributed Join - Beispiel

explain plan for

select a.ename from

```
tniewel.emp a,  
tniewel.emp@epg1 b,  
scott.emp@epg1 c
```

where

```
a.ename=b.ename and b.ename=c.ename;
```

# Distributed Join – nicht optimiert

| Id  | Operation            | Name | Rows | Bytes | Cost | TQ    | IN-OUT | PQ Distrib |
|-----|----------------------|------|------|-------|------|-------|--------|------------|
| 0   | SELECT STATEMENT     |      | 1    | 21    | 3    |       |        |            |
| * 1 | HASH JOIN            |      | 1    | 21    | 3    |       |        |            |
| 2   | MERGE JOIN CARTESIAN |      | 1    | 14    | 2    |       |        |            |
| 3   | REMOTE               |      | 1    | 7     |      | E.,OM | SERIAL |            |
| 4   | BUFFER SORT          |      | 82   | 574   | 2    |       |        |            |
| 5   | TABLE ACCESS FULL    | EMP  | 82   | 574   | 2    |       |        |            |
| 6   | REMOTE               |      | 1    | 7     |      | E.,OM | SERIAL |            |

Predicate Information (identified by operation id):

1 - access("A"."ENAME"="B"."ENAME" AND "B"."ENAME"="C"."ENAME")

Slave SQL Information (identified by operation id):

3 - SELECT "ENAME" FROM "SCOTT"."EMP"

6 - SELECT "ENAME" FROM "TNIWEL"."EMP"

ORACLE





## Distributed Join – optimiert

explain plan for

```
select a.ename from
```

```
tniewel.emp a
```

```
where a.ename = (select b.ename
```

```
from
```

```
tniewel.emp@epg1 b,
```

```
scott.emp@epg1 c
```

```
where
```

```
b.ename=c.ename);
```



# Distributed Join – optimiert

```
-----
```

| Id  | Operation         | Name | Rows | Bytes | Cost | TQ    | IN-OUT | PQ Distrib |
|-----|-------------------|------|------|-------|------|-------|--------|------------|
| 0   | SELECT STATEMENT  |      | 1    | 7     | 2    |       |        |            |
| * 1 | TABLE ACCESS FULL | EMP  | 1    | 7     | 2    |       |        |            |
| 2   | REMOTE            |      |      |       |      | E.,OM | SERIAL |            |

```
-----
```

Predicate Information (identified by operation id):

```
-----
```

1 - filter("A"."ENAME"= (SELECT "A2"."ENAME" FROM "SCOTT"."EMP"@EPG1.DE.ORACLE.COM "A1"))

Slave SQL Information (identified by operation id):

```
-----
```

2 - SELECT A2."ENAME" FROM "TNIIEWEL"."EMP" A2, "SCOTT"."EMP" A1 WHERE A2."ENAME"=A1."ENAM  
E"

Note: cpu costing is off

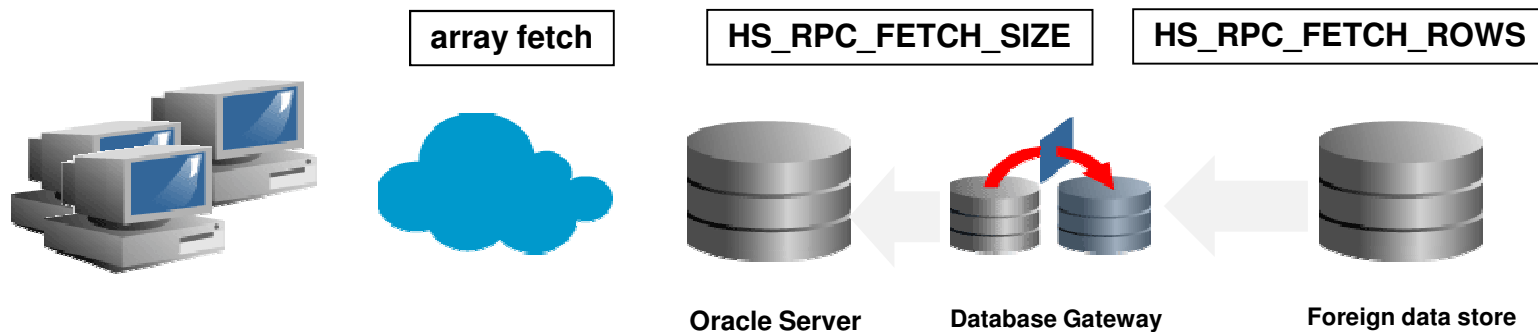
22 Zeilen ausgewöhlt.



# Distributed Joins - Optimierung

- Anlegen von Views
  - DB2: View Merge vs. View Materialization
  - Explain der View
  - Filter

# Optimierung von Bulk Operationen



Empfehlung: Nutzung der default Settings  
ODBC Treiber Abhängigkeiten(DG4ODBC)



# Agenda

- Architektur
- Funktionsweise von Database Gateways –  
Unterstützung verschiedenster SQL Dialekte
- Performance Aspekte
- Zugriff auf Fremdsysteme am Beispiel DB2 for  
z/OS (Database Gateway for DRDA)
- Optimierung von Load Prozessen
- Weiteres



## FDS Katalog Statistiken

- Berücksichtigung der Objekt Statistiken
  - `HS_FDS_SUPPORT_STATISTICS=TRUE`
    - Table Cardinality
    - Key Column Cardinality
- Sind die Objekt Statistiken korrekt ?
- Auswirkungen
  - Join Reihenfolge
  - Transfer der „Remote Tables“
- Empfehlung bei Migrationen von Gateways < 11.2
  - `HS_FDS_SUPPORT_STATISTICS=FALSE`

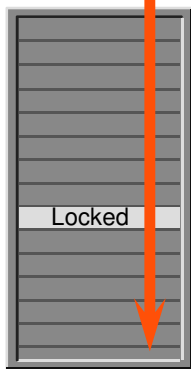


# Locking Modell

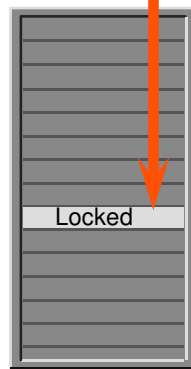
- Beispiel DB2 (Database Gateway for DRDA)
  - HS\_FDS\_TRANSACTION\_ISOLATION=READ\_UNCOMMITTED
  - HS\_FDS\_TRANSACTION\_ISOLATION=READ\_COMMITTED
- Lock Escalation

# DB2 Locking Modell

Uncommitted Read



Read Stability



Cursor Stability



Repeatable Read



***DB2 z/OS Version 10 only supports INSERT and DELETE operations of currently committed.***





# DB2 Locking Modell

Problem: Writers block Readers

- Timeouts
- Abbrüche mit SQLCODE – 911
  
- Lösung
  - Commit Frequently



# Transaction Model

- HS\_TRANSACTION\_MODEL
  - COMMIT\_CONFIRM
  - READ\_ONLY
  - READ\_ONLY\_AUTOCOMMIT
  - SINGLE\_SITE
  - SINGLE\_SITE\_AUTOCOMMIT
    - Kein Rollback möglich



# Zusammenfassung

- Berücksichtigung der Funktionsweise der Nicht-Oracle Datenbanksysteme
  - Nutzung von Funktionen, die nicht „compensated“ sind
  - Durchführung von häufigen Commits (auch bei Read Only Operationen)
  - Predicates (Beispiel DB2)
    - Indexable
      - Index kann genutzt werden
    - Stage 2
      - Predicates werden durch die ressourcenintensive DB2 Komponente RDS(Relational Data System) ausgewertet



# Gateway Passthrough – Beispiel 1

```
DECLARE
CRS binary_integer; RET binary_integer; VAL VARCHAR2(10);AGE Number
BEGIN
    CRS:=DBMS_HS_PASSTHROUGH.OPEN_CURSOR@gtwlink;
    DBMS_HS_PASSTHROUGH.PARSE@gtwlink(CRS,'SELECT NAME, AGE FROM PT_TABLE');
BEGIN
    RET:=0;
    WHILE (TRUE)
    LOOP
        RET:=DBMS_HS_PASSTHROUGH.FETCH_ROW@gtwlink (CRS,FALSE);
        DBMS_HS_PASSTHROUGH.GET_VALUES@gtwlink (CRS,1,VAL);
        DBMS_HS_PASSTHROUGH.GET_VALUES@gtwlink (CRS,2,AGE);
        INSERT INTO PT_TABLE_LOCAL VALUES (VAL);
    END LOOP;
EXCEPTION
WHEN NO_DATA_FOUND THEN
    BEGIN
        DBMS_OUTPUT.PUT_LINE('END OF FETCH');
        DBMS_HS_PASSTHROUGH.CLOSE_CURSOR@gtwlink(CRS) ;
    END;
END;
```



## Gateway Passthrough – Beispiel 2

```
BEGIN
    c := DBMS_HS_PASSTHROUGH.OPEN_CURSOR@DS3L;

    DBMS_HS_PASSTHROUGH.PARSE@DS3L(c, 'set current degree = ''ANY'');

    ret := DBMS_HS_PASSTHROUGH.EXECUTE_NON_QUERY@DS3L(c);

    dbms_output.put_line(ret || ' passthrough output');

    DBMS_HS_PASSTHROUGH.CLOSE_CURSOR@DS3L(c);
END;
```



# Agenda

- Architektur
- Funktionsweise von Database Gateways –  
Unterstützung verschiedenster SQL Dialekte
- Performance Aspekte
- Zugriff auf Fremdsysteme am Beispiel DB2 for  
z/OS (Database Gateway for DRDA)
- Optimierung von Load Prozessen
- Weiteres



# Optimierung von Load Prozessen

- Create Table
- Insert
- Copy
- Materialized Views



# Optimierung von Load Prozessen

- Beispiel

- `CREATE TABLE EMP AS SELECT * FROM SCOTT.EMP@gateway;`
- `INSERT INTO EMP SELECT * FROM SCOTT.EMP@gateway;`
- `COPY FROM SCOTT/TIGER@gateway INSERT EMP  
USING SELECT * FROM SCOTT.EMP@gateway;`





# Optimierung von Load Prozessen

- Beispiel: Transfer von DB2 Daten in die Oracle Welt
  - Materialized Views

```
CREATE MATERIALIZED VIEW empdb2
  REFRESH COMPLETE NEXT SYSDATE + 1
  WITH ROWID
AS
SELECT * FROM scott.emp@gateway
  WHERE deptno=20;
```



# Optimierung von Load Prozessen

- Parallelisierung
  - Oracle Datenbank < 11g
    - Erstellen von DB2-Views zur Verarbeitung von Teilmengen der Daten
  - Ausführen von n Gateway Sessions mit parallelen Zugriff auf die erstellten Views
    - `Insert into localtable Select * from remote_fds_view@gateway`



# Optimierung von Load Prozessen

- Parallelisierung mit der Oracle Datenbank  $\geq 11g$ 
  - Nutzung von `dbms_hs_parallel`
    - `LOAD_TABLE`  
Parallel load der „remote tables“
    - `CREATE_TABLE_TEMPLATE`  
Erstellen „local Table“ auf Basis einer „remote Table“
    - `CREATE_OR_REPLACE_VIEW`  
Erstellen einer „read only View“ zum parallelisierten Lesen der „remote Daten“
    - `DROP_VIEW`  
Löschen von Objekten, die mit `CREATE_OR_REPLACE_VIEW` erstellt wurden



# Optimierung von Load Prozessen

- Beispiel

```
Declare
```

```
remtab varchar2(36);  
dblink varchar2(36);  
oratab varchar2(36);  
rowcnt integer;
```

```
begin
```

```
remtab:='tniewel.emp';  
oratab:='tniewel.emp1';  
dblink:='D71E';  
/* Truncate=False */  
dbms_hs_parallel.load_table(remtab,dblink,oratab,False,4,rowcnt);  
dbms_output.put_line(' # of rows loaded ' ||rowcnt);
```

```
end;
```



## Optimierung von Load Prozessen

Voraussetzung für eine Parallelisierung ist **eine** der folgenden Eigenschaften

- Die „remote Table“ ist „range partitioned“
- Histogramm für eine numerische Spalte ist vorhanden
- Index oder Primärschlüssel auf einer numerischen Spalte ist vorhanden



# Agenda

- Architektur
- Funktionsweise von Database Gateways –  
Unterstützung verschiedenster SQL Dialekte
- Performance Aspekte
- Zugriff auf Fremdsysteme am Beispiel DB2 for  
z/OS (Database Gateway for DRDA)
- Optimierung von Load Prozessen
- Weiteres



# Security

- Passwörter in den initSID.ora Dateien
  - Verschlüsselung durch dg4pwd (10g: tg4pwd) utility
  - Verschlüsseltes Password ist in der Datei „initSID.pwd“ gespeichert
- Zugriff zu „non Oracle“ Tabellen über „Database Links“
  - Userid/Password im Database Link
  - Keine Userid/Password im Database Link
    - Oracle Userid und Password werden genutzt



# Dictionary Views

```
set linesize 132
col username format a15
col machine format a15
col db_link format a20
col program format a15

select a.username,
       c.machine,
       c.program,
       b.db_link,
       c.process
from   v$session a,
       v$hs_session b,
       v$hs_agent c
where  a.sid=b.sid and
       c.agent_id=b.agent_id;
```

| USERNAME | MACHINE                      | PROGRAM                                     | DB_LINK            | PROCESS |
|----------|------------------------------|---------------------------------------------|--------------------|---------|
| TNIEWEL  | sccloud026.de.o<br>racle.com | dg4db2DS3L@sccl<br>oud026.de.orac1<br>e.com | DS3L.DE.ORACLE.COM | 30888   |







# Gateway Trace

- Aktivierung durch Gateway init Parameter
  - HS\_FDS\_TRACE\_LEVEL
    - OFF, ON, DEBUG

HOSGIP returned value of "TRUE" for HS\_FDS\_QUOTE\_IDENTIFIER

SQL text from hgopars, id=1, len=123 ...

```
00: 53454C45 43542041 312E2245 4D504E4F [SELECT A1."EMPNO]
10: 222C2041 312E2245 4E414D45 222C2041 [", A1."ENAME", A]
20: 312E224A 4F42222C 2041312E 224D4752 [1."JOB", A1."MGR]
30: 222C2041 312E2248 49524544 41544522 [", A1."HIREDATE"]
40: 2C204131 2E225341 4C222C20 41312E22 [, A1."SAL", A1."]
50: 434F4D4D 222C2041 312E2244 4550544E [COMM", A1."DEPTN]
60: 4F222046 524F4D20 22544E49 4557454C [O" FROM "TNIEWEL]
70: 222E2245 4D503122 204131 [". "EMP1" A1]
```

Deferred open until first fetch.

ORACLE



## Timeouts - DB2 for z/OS

- **DB2 IDLE THREAD TIMEOUT (IDTHTOIN)**
  - Bewirkt den cancel von Idle Threads im DB2
  - Ist auch für DG4DRDA Gateway Sessions aktiv
  - Sessions werden mit dem Fehler ABEND=04E FOR REASON=00D3003B beendet
  - Auf Oracle Seite muss ein Rollback durchgeführt werden



## Timeouts - DB2 for z/OS

- DG4DRDA init Parameter *HS\_IDLE\_TIMEOUT*
  - Bewirkt den cancel von Oracle Prozessen / DB2 Threads
  - Muss < **IDTHTOIN** sein
  - Sessions werden mit dem Fehler **ORA-28511** beendet
  - Beim Erneuten Ausführen des SQL Befehls wird die Verbindung zum DB2 wieder aufgebaut. Ein Rollback ist nicht notwendig.
- **Timeouts können durch einen Close Database Link vermieden werden**

# Q&A

ORACLE

# Hardware and Software

ORACLE®

# Engineered to Work Together

ORACLE®

---

**ORACLE®**