

Gib mir deine Daten – Datenaustausch zwischen Oracle, PostgreSQL und MariaDB

DOAG-Datenbank-Konferenz
24.05.2023, Düsseldorf
Markus Flechtner

info@ordix.de

www.ordix.de



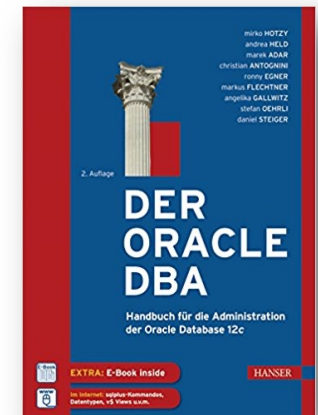
Markus Flechtner

- Diplom-Mathematiker
- Oracle seit 1990
- Entwickler, Oracle-Field-Support, DBA, Consultant
- Principal Consultant & Teamleiter bei der ORDIX AG
- Schwerpunkte
 - Hochverfügbarkeit
 - Migrationen und Upgrades
 - Multitenant
- DOAG-Themenverantwortlicher "OpenSource-Datenbanken"
-  @markusdba
-  @markusdba@mastodon.social
-  markusdba.de|.net
-  mfl@ordix.de

ORDIX AG



DOAG





Oracle ACE
Program

500+ technical experts
helping peers globally

The **Oracle ACE Program** recognizes and
rewards community members for their technical
and community contributions to the Oracle



Oracle ACE
Director



Oracle ACE
Pro



Oracle ACE
Associate



Oracle ACE

ace.oracle.com/nominate

ace.oracle.com

Connect:  aceprogram_ww@oracle.com

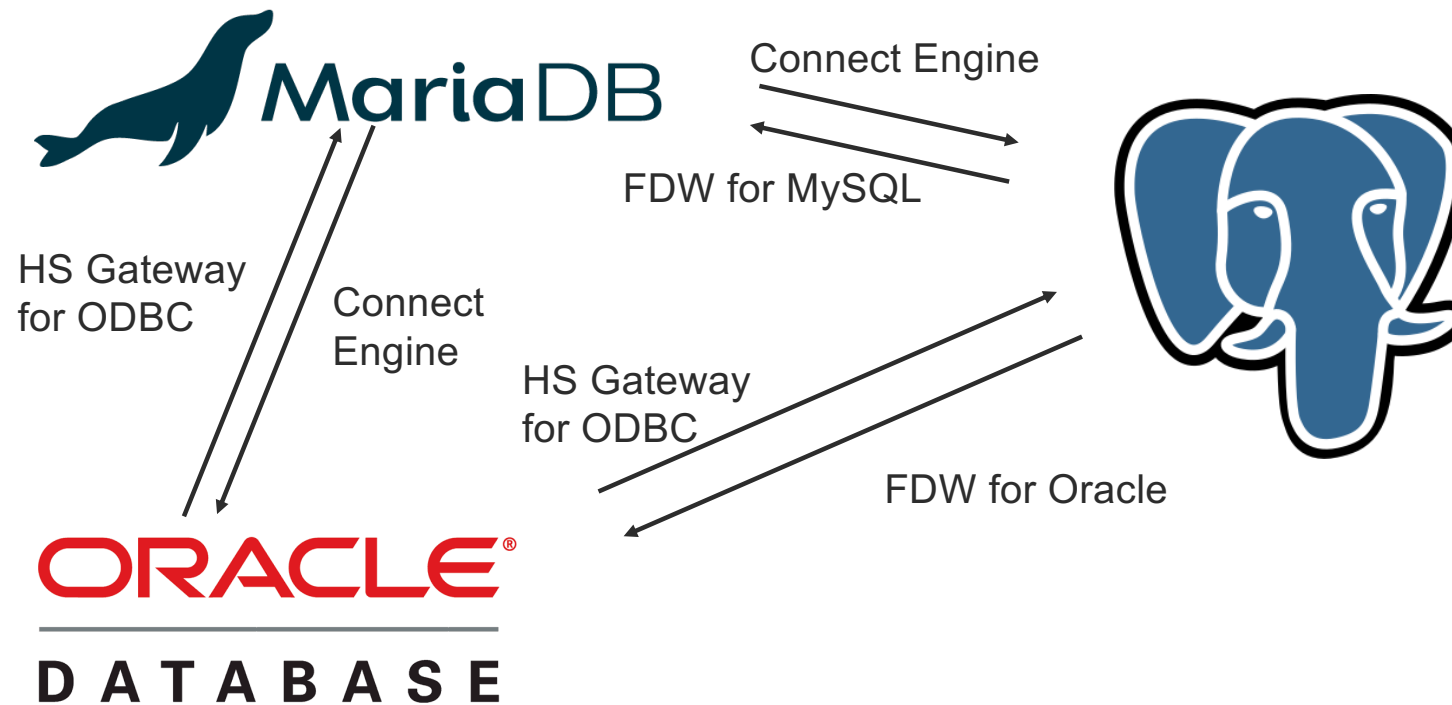
 Facebook.com/OracleACEs

 [@oracleace](https://twitter.com/oracleace)



Worum geht es?

ORDIX AG



Technologien zum Zugriff auf andere Datenbanken

- PostgreSQL
 - Foreign Data Wrapper
- Oracle
 - ODBC Gateway
- MariaDB
 - Connect Engine

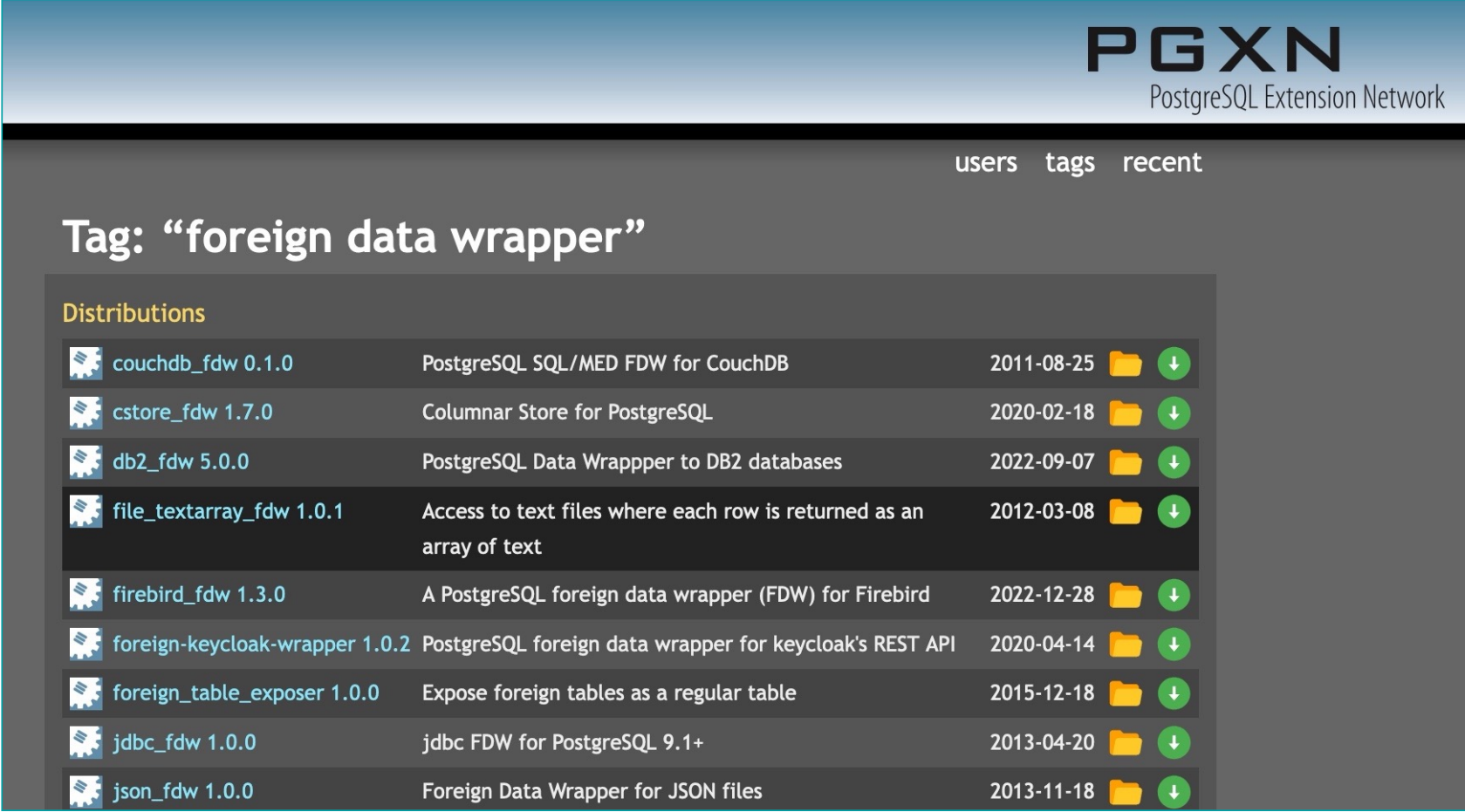
Agenda

ORDIX AG

- PostgreSQL Foreign Data Wrapper
- Oracle Heterogeneous Gateway für ODBC
- MariaDB Connect Engine
- Zusammenfassung & weitere Informationen

Foreign Data Wrapper

- Foreign Data Wrapper (FDW) dienen zum Zugriff von PostgreSQL heraus auf andere Datenquellen
- "Mitgeliefert" (contrib-Package) werden
 - FDW für PostgreSQL (postgres_fdw)
 - FDW für Datei-Zugriffe (file_fdw)
- Weitere Foreign Data Wrapper
 - PostgreSQL Extension Network <https://pgxn.org>
- Übersicht im PostgreSQL-Wiki: https://wiki.postgresql.org/wiki/Foreign_data_wrappers



The screenshot shows the PGXN (PostgreSQL Extension Network) website. At the top right is the PGXN logo with the text 'PostgreSQL Extension Network'. Below the logo are links for 'users', 'tags', and 'recent'. The main heading is 'Tag: "foreign data wrapper"'. Underneath is a section titled 'Distributions' which contains a table of various PostgreSQL extensions. Each row in the table includes a gear icon, the extension name and version, a brief description, the release date, a folder icon, and a download arrow icon.

Extension Name	Description	Date	Download
couchdb_fdw 0.1.0	PostgreSQL SQL/MED FDW for CouchDB	2011-08-25	Download
cstore_fdw 1.7.0	Columnar Store for PostgreSQL	2020-02-18	Download
db2_fdw 5.0.0	PostgreSQL Data Wrappper to DB2 databases	2022-09-07	Download
file_textarray_fdw 1.0.1	Access to text files where each row is returned as an array of text	2012-03-08	Download
firebird_fdw 1.3.0	A PostgreSQL foreign data wrapper (FDW) for Firebird	2022-12-28	Download
foreign-keycloak-wrapper 1.0.2	PostgreSQL foreign data wrapper for keycloak's REST API	2020-04-14	Download
foreign_table_exposer 1.0.0	Expose foreign tables as a regular table	2015-12-18	Download
jdbc_fdw 1.0.0	jdbc FDW for PostgreSQL 9.1+	2013-04-20	Download
json_fdw 1.0.0	Foreign Data Wrapper for JSON files	2013-11-18	Download

Foreign Data Wrapper - Komponenten

Software	Foreign Data Wrapper
Extension	"create extension .."
"Foreign Server" – Verbindungsinformationen für die Datenquelle	"create foreign server .."
User-Mapping (mit welchem Benutzer verbindet sich der PostgreSQL-Benutzer mit der Datenquelle)	"create user mapping .."
Tabellen-Mapping der Tabelle aus dem Fremdsystem	"create foreign table ..."

Foreign Data Wrapper für Oracle

- Webseite: https://github.com/laurenz/oracle_fdw
- Aktuelle Version (Stand Januar 2023): Version 2.5.0 vom 28.10.2022
- Oracle-Client muss auf dem PostgreSQL-Server installiert sein
 - Instant-Client ist ausreichend
- Benötigt LLVM (früher "Low Level Virtual Machine")

Vorbereitungen

- Oracle Client installieren
- LLVM & weitere notwendige Komponenten installieren

```
# dnf -y install postgresql15-devel.x86_64 postgresql15-llvmjit.x86_64  
# dnf -y install gcc.x86_64 gcc-c++.x86_64  
# dnf -y install llvm-toolset-7.x86_64
```

Installation oracle_fdw

- Download der Software

```
# wget https://github.com/laurenz/oracle_fdw/archive/master.zip  
# unzip -uq master.zip
```

- Umgebung setzen

```
# export ORACLE_HOME=/opt/oracle/product/18c/dbhomeXE  
# export LD_LIBRARY_PATH=$ORACLE_HOME/lib
```

- Installation

```
# make  
# make install
```

Vorbereitungen in PostgreSQL (1) – Server-Konfiguration

- Die Oracle-Client-Library muss im Suchpfad sein, wenn der PostgreSQL-Cluster gestartet wird
- Variante 1 (systemctl / override.conf)

```
[Service]
Environment=PGDATA=/pgdata/PG15
Environment=ORACLE_HOME=/opt/oracle/product/18c/dbhomeXE
Environment=LD_LIBRARY_PATH=/opt/oracle/product/18c/dbhomeXE/lib
```

- Variante 2 (start via Shell / pg_ctl)

```
$ export ORACLE_HOME=/opt/oracle/product/18c/dbhomeXE
$ export LD_LIBRARY_PATH=/opt/oracle/product/18c/dbhomeXE/lib

$ pg_ctl start ...
```

Vorbereitungen in PostgreSQL (2) – Erweiterung & Foreign Server

- Erweiterung anlegen

```
postgres# create extension oracle_fdw;  
CREATE EXTENSION
```

- Foreign Server anlegen

```
postgres# create server oraclexe foreign data wrapper oracle_fdw  
2 > options (dbserver '//localhost:1521/XEPDB1');  
CREATE SERVER
```

- Nutzung in PostgreSQL erlauben

```
postgres# grant usage on foreign server oraclexe to public;  
GRANT
```

Vorbereitungen in PostgreSQL (3) – User Mapping & Foreign Table

- User Mapping anlegen

```
postgres# create user mapping for postgres server oraclexe
2 > options (user 'scott', password 'tiger');
CREATE USER MAPPING
```

- Foreign Table anlegen

```
postgres# CREATE FOREIGN TABLE dept (
1 > deptno          integer          NOT NULL,
2 > dname character varying(30),
3 > loc character varying(13),
4 > ) SERVER oraclexe OPTIONS (schema 'SCOTT', table 'DEPT');
CREATE FOREIGN TABLE
postgres# select count(*) from dept;
count
-----
      4
(1 row)
```

GROSSSCHRIFT
ist wichtig

- 1. Versuch

```
postgres# update dept set loc='PADERBORN' where deptno=10;
ERROR:  HV00L: no primary key column specified for foreign Oracle table
DETAIL:  For UPDATE or DELETE, at least one foreign table column must be marked
as primary key column.
HINT:  Set the option "key" on the columns that belong to the primary key.
```

- Korrektur

```
postgres# alter foreign table dept alter "deptno" options (ADD key 'true');
ALTER FOREIGN TABLE
```

- 2. Versuch

```
postgres# update dept set loc='PADERBORN' where deptno=10;
UPDATE 1
```

Einfacher: Schema-Import

```
postgres# IMPORT FOREIGN SCHEMA "SCOTT"  
2 >     FROM SERVER oraclexe  
3 >     INTO public  
4 >     OPTIONS (case 'lower');  
IMPORT FOREIGN SCHEMA  
  
postgres# select count(*) from public.emp;  
count  
-----  
      14  
(1 row)  
  
postgres# update public.emp set sal=5100 where empno=7839;  
UPDATE 1
```

- Die Optionen "LIMIT TO" bzw. "EXCEPT" erlauben die Beschränkung auf bestimmte Tabellen oder den Ausschluss einzelner Tabellen

Foreign Data Wrapper for MySQL/MariaDB

- Verfügbar im PostgreSQL Repository
- Aktuelle Version: 2.9 (Dezember 2022)
- Installation via rpm:

```
root# yum install -y mysql_fdw_15 --nobest --allowdowngrading
```

- Das Package erwartet die MySQL-Binaries, daher muss bei der Verwendung von MariaDB mit den Optionen "--nobest" und "--allowdowngrading" gearbeitet werden

Foreign Data Wrapper for MySQL/MariaDB – Konfiguration

1. Erweiterung anlegen

```
postgres# create extension mysql_fdw;  
CREATE EXTENSION
```

2. Foreign Server anlegen

```
postgres# create server mariadb foreign data wrapper mysql_fdw options  
2> (host 'localhost', port '3306');  
CREATE SERVER
```

3. User Mapping anlegen

```
postgres# create user mapping for public server mariadb options  
2> (username 'demo', password 'geheim');  
CREATE USER MAPPING
```

4. Table Mapping

```
postgres# CREATE FOREIGN TABLE public.city
> (id integer,
> name char(35),
> countrycode char(3),
> district char(20),
> population integer
> )
> SERVER mariadb
> OPTIONS (dbname 'world', table_name 'city');
```

```
postgres# select count(*) from public.city;
 count
-----
  4079
(1 row)
```

Meta-Befehle rund um Foreign Data Wrapper

- Meta-Befehle

<code>\dE[S+] [PATTERN]</code>	list foreign tables
<code>\des[+] [PATTERN]</code>	list foreign servers
<code>\deu[+] [PATTERN]</code>	list user mappings
<code>\dew[+] [PATTERN]</code>	list foreign-data wrappers

- Beispiel

```
postgres# \dE
```

List of relations			
Schema	Name	Type	Owner
-----+-----+-----+-----			
public	bonus	foreign table	postgres
public	dept	foreign table	postgres
public	emp	foreign table	postgres
public	salgrade	foreign table	postgres
(4 rows)			

Views im System Catalog

- pg_foreign_data_wrapper
- pg_foreign_server
- pg_foreign_table
- pg_user_mapping

```
postgres# select * from pg_user_mapping;
 oid | umuser | umserver |          umoptions
-----+-----+-----+-----
 47043 |    10 |   47038 | {user=scott,password=tiger}
(1 row)

postgres# select srvname,srvoptions from pg_foreign_server;
 srvname |          srvoptions
-----+-----
 oraclexe | {dbserver=//localhost:1521/XEPDB1}
(1 row)
```

Agenda

- PostgreSQL Foreign Data Wrapper
- Oracle Heterogeneous Gateway für ODBC
- MariaDB Connect Engine
- Zusammenfassung & weitere Informationen

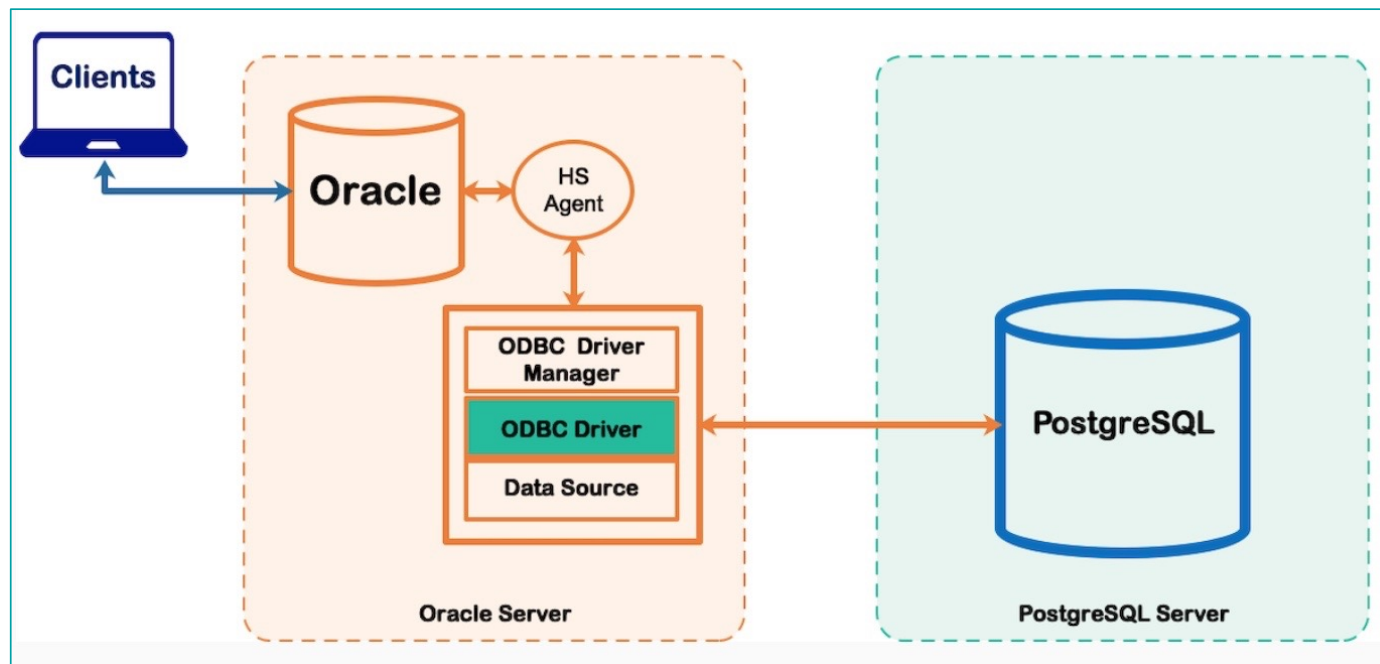
Oracle Heterogeneous Services (HS)

- Unter dem Namen "Oracle Heterogeneous Services (HS)" platziert Oracle die Gateways zu anderen Datenbanken bzw. Systemen
- Beispiele:
 - APPC
 - DRDA (Distributed Relational Database Architecture, DB2)
 - Informix
 - ODBC
 - SQL Server
 - Sybase
 - Teradata
 - WebSphere MQ

- Oracle Database 21c - Licensing Information

Feature / Option / Pack	X E	SE 2	E E	EE - ES	DBCS SE	DBCS EE	DBCS EE-HP	DBCS EE-EP	ExaCS /CC	Notes
Database Gateways	N	Y	Y	Y	Y	Y	Y	Y	Y	Database Gateway for ODBC is included in all offerings. All other Database Gateway products are available for all offerings, but require a separate product license.
Oracle Database Gateway for ODBC	Oracle Database Gateway for ODBC can be installed and used on a machine different from the machine where the Oracle Database is installed and used. It is not necessary to obtain a separate license for the machine running Oracle Gateway for ODBC.									

ODBC-Gateway zu PostgreSQL - Architektur



Quelle: <https://www.enterprisedb.com/postgres-tutorials/accessing-postgresql-databases-using-psqlodbc-oracle>

Installation (1) – Software-Komponenten

1. Zusätzliche Software-Komponenten installieren

```
# ### Oracle Server
# yum install -y unixODBC unixODBC-devel

# ### PostgreSQL-Server
# yum install -y postgresql15-odbc
```

- Informationen zum PostgreSQL-ODBC-Driver: <https://odbc.postgresql.org/>
- Wo liegen die Konfigurationsdateien (ODBC-Konfiguration) auf dem Oracle-Server?

```
$ odbcinst -j
unixODBC 2.3.1
DRIVERS.....: /etc/odbcinst.ini
SYSTEM DATA SOURCES: /etc/odbc.ini
FILE DATA SOURCES..: /etc/ODBCDataSources
USER DATA SOURCES..: /home/oracle/.odbc.ini
SQLULEN Size.....: 8
SQLLEN Size.....: 8
SQLSETPOSIROW Size.: 8
```

Installation (2) – odbcinst.ini

2. ODBC-Konfiguration (/etc/odbcinst.ini)

```
[PostgreSQL]
Description      = ODBC for PostgreSQL
Driver           = /usr/pgsql-15/lib/psqlodbcw.so
Setup           = /usr/lib64/libodbcpsqlS.so
Driver64        = /usr/pgsql-15/lib/psqlodbcw.so
Setup64         = /usr/lib64/libodbcpsqlS.so
FileUsage       = 1
```

Pfad ggf. anpassen

Installation (3) – Datenquelle in odbc.ini definieren

3. ODBC-Datenquelle definieren (/etc/odbc.ini oder /home/oracle/.odbc.ini)

```
[pg]
Database      = scott_db
Description   = Connection to PostgreSQL Cluster PG15
Driver        = PostgreSQL
Servername    = localhost
Port          = 5432
Username      = postgres
Password      = manager
```



Verweis auf odbcinst.ini

- Übersicht über alle Konfigurationsparameter: <https://odbc.postgresql.org/docs/config.html>
- User & Password sind nicht erforderlich → nach erfolgreichem Test entfernen

Installation (4) – Datenquelle testen

4. Testen der ODBC-Datenquelle

```
# isql -v pg
+-----+
| Connected! |
|           |
| sql-statement |
| help [tablename] |
| quit |
|           |
+-----+
SQL> quit
```

- "-v " (verbose" für detailliertere Fehlermeldungen
- Nach dem Test sollten User und Password aus der odbc.ini entfernt werden

Installation (5) – Gateway: init-Datei

5. Init-Datei für das Gateway definieren

```
$ cat $ORACLE_HOME/hs/admin/initpg.ora

HS_FDS_CONNECT_INFO = pg
HS_FDS_TRACE_LEVEL = ON
HS_FDS_SHAREABLE_NAME = /usr/pgsql-15/lib/psqlodbcw.so
HS_LANGUAGE = AMERICAN_AMERICA.WE8ISO8859P15
set ODBCINI=/etc/odbc.ini
```

- Im Verzeichnis \$ORACLE_HOME/hs/admin liegt auch eine Beispieldatei initdg4odbc.ora
- In jedem Fall AMERICAN_AMERICA.WE8ISO8859P15 als Sprache verwenden, damit Fehlermeldungen des unixODBC angezeigt werden! – siehe auch MOS-Note 756186.1

Installation (6) – Gateway-Listener

6. Listener für das Gateway definieren (listener.ora)

```
$ cat $ORACLE_HOME/network/admin/listener.ora

LISTENERPG=
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP) (HOST = dedekind.markusdba.mac) (PORT = 1522))
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1522))
    )
  )

SID_LIST_LISTENERPG=
(SID_LIST=(SID_DESC=(SID_NAME=pg) (ORACLE_HOME=/opt/oracle/product/18c/dbhomeXE)
(PROGRAM=dg4odbc)))
```

- Alternativ zum separaten Listener kann man auch den normalen Listener "LISTENER" verwenden und das Gateway mittels SID_LIST_LISTENER einbinden

Installation (7) – Gateway-Listener starten

7. Listener für das Gateway starten

```
$ lsnrctl start listenerpg

LSNRCTL for Linux: Version 18.0.0.0.0 - Production on 19-JAN-2023 22:25:03
Copyright (c) 1991, 2018, Oracle. All rights reserved.
Starting /opt/oracle/product/18c/dbhomeXE/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 18.0.0.0.0 - Production
System parameter file is
/opt/oracle/product/18c/dbhomeXE/network/admin/listener.ora
Log messages written to
/opt/oracle/diag/tnslsnr/dedekind/listenerpg/alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=dedekind.markusdba.mac) (PORT=1522)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc) (KEY=EXTPROC1522)))
Services Summary...
Service "pg" has 1 instance(s).
  Instance "pg", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
```

Installation (8) – tnsnames.ora-Eintrag definieren & testen

8. Tnsnames.ora-Eintrag definieren und testen

```
$ cat $ORACLE_HOME/network/admin/tnsnames.ora
[...]
pg=
  (DESCRIPTION=
    (ADDRESS=
      (PROTOCOL=TCP) (HOST = dedekind.markusdba.mac) (PORT = 1522))
    (CONNECT_DATA= (SID=pg)) (HS=OK)
  )

$ tnsping pg
TNS Ping Utility for Linux: Version 18.0.0.0.0 - Production on 19-JAN-2023
10:35:19
[...]
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION= (ADDRESS= (PROTOCOL=TCP) (HOST =
dedekind.markusdba.mac) (PORT = 1522)) (CONNECT_DATA= (SID=pg)) (HS=OK))
OK (10 msec)
```

Datenbank-Link anlegen

- Datenbank-Link anlegen und testen

```
SQL> create database link PG connect to "scott" identified by "tiger"
2   using 'PG';
Database link created.

SQL> select * from "scott"."emp"@PG;
select * from "scott"."emp"@PG
          *

ERROR at line 1:
ORA-28500: connection from ORACLE to a non-Oracle system returned this message:
[unixODBC]SCRAM authentication requires libpq version 10 or above
{08001,NativeErr = 101}
ORA-02063: preceding 3 lines from PG
```

[unixODBC]SCRAM authentication requires libpq version 10 or above

- Ursache: unixodbc bringt anscheinend eine alte PostgreSQL-Client-Library libpq.so mit (Centos 7)
- Lösung:

```
# pwd
/usr/lib64

# ls -ltr libpq*
-rwxr-xr-x. 1 root root 197544 Jun 24 2022 libpq.so.5.5
lrwxrwxrwx. 1 root root      12 Jan  8 11:02 libpq.so.5 -> libpq.so.5.5

# rm libpq.so.5

# ln -s /usr/pgsql-15/lib/libpq.so /usr/lib64/libpq.so.5

# ls -al libpq*
lrwxrwxrwx. 1 root root      26 Jan 19 22:21 libpq.so.5 -> /usr/pgsql-15/lib/libpq.so
-rwxr-xr-x. 1 root root 197544 Jun 24 2022 libpq.so.5.5
```

Der Übeltäter

```
SQL> select count(*) from "scott"."emp"@PG;
```

```
  COUNT (*)  
-----  
          14
```

- Achtung:
 - Bei PostgreSQL sind die Objekt-Namen alle in **kleinschrift**
→ daher **Anführungszeichen (' " ')** verwenden
 - → ggf. auf Oracle-Ebene mit **Views** arbeiten, bei denen die PostgreSQL-Namen in **GROSSSCHRIFT** umgewandelt werden

Beispiel für DML

```
SQL> select "salary" from "scott"."emp"@pg where "name"='KING' ;
      salary
-----
      5000

SQL> update "scott"."emp"@pg set "salary"="salary"+100 where "name"='KING' ;
1 row updated.

SQL> commit;
Commit complete.

SQL> select "salary" from "scott"."emp"@pg where "name"='KING' ;
      salary
-----
      5100
```

Achtung bei den Datentypen

SQL> select "rolname","rolcanlogin" from "pg_roles"@pg;

rolname	rolcanlogin
-----	-----
[...]	
pg_checkpoint	0
postgres	1
scott	1
monitor_system_stats	0

0 = false
1 = true

ODBC-Gateway für MySQL/MariaDB (1)

- Gleiches Prinzip wie bei der Verbindung Oracle → PostgreSQL
- Erforderliche Pakete: unixODBC, unixODBC-devel und mariadb-connector-odbc
- /etc/odbc.ini

```
[mariadb]
Database      = world
Description   = Connection to MariaDB
Driver        = MariaDB
Servername     = localhost
Port          = 3306
UID           = fdw_user
PWD           = geheim
```

- Testen mit "isql -v mariadb"

ODBC-Gateway für MySQL/MariaDB (2) – init-Datei für das Gateway

- Init-Datei für das ODBC-Gateway - \$ORACLE_HOME/hs/admin/initmariadb.ora

```
HS_FDS_CONNECT_INFO = pg
HS_FDS_TRACE_LEVEL = ON
HS_FDS_SHAREABLE_NAME = /usr/lib64/libmaodbc.so
HS_LANGUAGE = AMERICAN_AMERICA.WE8ISO8859P15
set ODBCINI=/etc/odbc.ini
```

Aus
/etc/odbcinst.ini

- Eintrag in der listener.ora:

```
$ cat $ORACLE_HOME/network/admin/listener.ora

LISTENERMARIADB=
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 127.0.0.1) (PORT = 1523))
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1523))
    )
  )

SID_LIST_LISTENERMARIADB=
(SID_LIST=(SID_DESC=(SID_NAME=mariadb) (ORACLE_HOME=/opt/oracle/product/23c/dbhomeFree)
(PROGRAM=dg4odbc)))
```

- Listener starten ("lsnrctl start listenermariadb")

ODBC-Gateway für MySQL/MariaDB (4) - tnsnames.ora

- Eintrag in der tnsnames.ora

```
$ cat $ORACLE_HOME/network/admin/tnsnames.ora
[...]
mariadb=
  (DESCRIPTION=
    (ADDRESS=
      (PROTOCOL=TCP) (HOST = 127.0.0.1) (PORT = 1523))
    (CONNECT_DATA= (SID=mariadb)) (HS=OK)
  )
$ tnsping mariadb
TNS Ping Utility for Linux: Version 23.0.0.0.0 - Developer-Release on 15-MAY-2023
14:43:40
Copyright (c) 1997, 2023, Oracle. All rights reserved.
Used parameter files:
/opt/oracle/product/23c/dbhomeFree/network/admin/sqlnet.ora
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION= (ADDRESS= (PROTOCOL=TCP) (HOST =
127.0.0.1) (PORT = 1523)) (CONNECT_DATA= (SID=mariadb)) (HS=OK))
OK (0 msec)
```

- Datenbank-Link anlegen und testen

```
SQL> create database link mariadb connect to "fdw_user" identified by "geheim"
      2 using 'mariadb';
Database link created.

SQL> select count(*) from "country"@ "mariadb";

COUNT (*)
-----
          239
```

- Wie bei PostgreSQL: Namen in Anführungszeichen setzen

Agenda

- PostgreSQL Foreign Data Wrapper
- Oracle Heterogeneous Gateway für ODBC
- MariaDB Connect Engine
- Zusammenfassung & weitere Informationen

MariaDB Connect Engine

- Zugriff auf externe Datenquellen von MariaDB aus
- Gemäß dem MED-Standard ("Management of External Data")
- Datenquellen
 - ODBC
 - JDBC
 - MongoDB
 - CSV, XML
 - ...
- Dokumentation:
 - <https://mariadb.com/kb/en/introduction-to-the-connect-engine/>
- Bei MySQL gibt es nichts Vergleichbares

MariaDB Connect Engine - Installation

- RPM-Paket installieren

```
dnf install -y MariaDB-connect-engine unixODBC
```

- Engine in der DB installieren

```
MariaDB [(none)]> install soname 'ha_connect.so';  
Query OK, 0 rows affected (0.001 sec)
```


MariaDB Connect Engine- vorhandene Engines

```
MariaDB [(none)]> show engines;
```

+-----+-----+-----+ [...]		
Engine	Support	Comment [...]
+-----+-----+-----+ [...]		
CSV	YES	Stores tables as CSV files [...]
MRG_MyISAM	YES	Collection of identical MyISAM tables [...]
MEMORY	YES	Hash based, stored in memory, useful for temporary tables [...]
Aria	YES	Crash-safe tables with MyISAM heritage. Used for internal temporary tables [...]
MyISAM	YES	Non-transactional engine with good performance and small data footprint [...]
CONNECT	YES	Management of External Data (SQL/NOSQL/MED), including Rest query results [...]
SEQUENCE	YES	Generated tables filled with sequential values [...]
InnoDB	DEFAULT	Supports transactions, row-level locking, foreign keys and encryption ... [...]
PERFORMANCE_SCHEMA	YES	Performance Schema [...]
+-----+-----+-----+ [...]		

MariaDB Connect Engine - ODBC-Verbindung konfigurieren

- Analog zu Oracle-ODBC-Gateway
 - `/etc/odbc.ini`
 - Testen mit `isql`

MariaDB Connect Engine - Verbindung zu PostgreSQL

```
MariaDB [(none)]> use world;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [world]> create table emp_pg_remote engine=connect table_type=ODBC
tabname='emp' dbschema='scott' connection='dsn=pg';
Query OK, 0 rows affected (0.048 sec)
```

ODBC-Quelle

```
MariaDB [world]> select * from emp_pg_remote;
```

empno	name	job	manager	hiredate	salary	comm	deptno
7839	KING	PRESIDENT	NULL	1981-11-17	5000.00	NULL	10
7566	JONES	MANAGER	7839	1981-04-02	2975.00	NULL	20
7788	SCOTT	ANALYST	7566	1982-12-09	3000.00	NULL	20
7876	ADAMS	CLERK	7788	1983-01-12	1100.00	NULL	20

[...]

MariaDB Connect Engine - Verbindung zu Oracle

- /etc/odbcinst.ini

```
[Oracle]
Description=Oracle Driver
Driver=/opt/oracle/product/23c/dbhomeFree/lib/libsqora.so.23.1
```

- Instant-Client reicht aus

- /etc/odbc.ini

```
[scott]
Driver = Oracle
UserID = scott
Password = tiger
ServerName = //localhost:1521/XEPDB1
```

- Test mit isql

MariaDB Connect Engine - Verbindung zu Oracle

- Tabelle anlegen

```
MariaDB> create table emp_remote engine=connect table_type=ODBC tabname='EMP'  
dbschema='SCOTT' connection='dsn=scott';  
Query OK, 0 rows affected (3.509 sec)
```

Agenda

- PostgreSQL Foreign Data Wrapper
- Oracle Heterogeneous Gateway für ODBC
- MariaDB Connect Engine
- Zusammenfassung & weitere Informationen

Zusammenfassung

- Datenaustausch zwischen allen drei beteiligten Datenbanken ist möglich
- Installation insgesamt "problemarm"
- Achtung bei der Standard-Schreibweise der Objektnamen:
 - Oracle: GROSS
 - MySQL/MariaDB, PostgreSQL: klein

Weitere Informationen – Oracle ODBC-Gateway

- Oracle Database 21c: Database Heterogeneous Connectivity User's Guide
<https://docs.oracle.com/en/database/oracle/oracle-database/21/heter/index.html>
- Oracle Database 21c: Oracle® Database Gateway for ODBC - User's Guide
<https://docs.oracle.com/en/database/oracle/oracle-database/21/odbcu/index.html>
- Accessing PostgreSQL databases using psqlODBC in Oracle:
<https://www.enterprisedb.com/postgres-tutorials/accessing-postgresql-databases-using-psqlodbc-oracle>
- MOS-Notes
 - How to Resolve Common Errors Encountered while using Database Gateways (DG4IFMX, Dg4MSQL, DG4SYBS), DG4ODBC or Generic Connectivity (Doc ID 234517.1)
 - ORA-28545 If PROGRAM=dg4odbc.exe Is Specified In The Gateway LISTENER.ORA (Doc ID 1323387.1)
 - How to Configure DG4ODBC on 64bit Unix OS (Linux, Solaris, AIX, HP-UX Itanium) to Connect to Non-Oracle Databases Post Install (Doc ID 561033.1)
 - Error Ora-28500 and Sqlstate I or Displays a Square Bracket - [- Issuing Selects From a Unicode Oracle RDBMS With Dg4odbc To Non-Oracle Databases Using the UnixODBC Driver Manager (Doc ID 756186.1)

Weitere Informationen – PostgreSQL Foreign Data Wrappers

- Übersicht im PostgreSQL-Wiki: https://wiki.postgresql.org/wiki/Foreign_data_wrappers
- FDW for Oracle: https://github.com/laurenz/oracle_fdw
- FDW for MySQL/MariaDB: https://github.com/EnterpriseDB/mysql_fdw
- PostgreSQL Accessing MySQL as a Data Source Using mysql_fdw
https://www.percona.com/blog/2018/08/24/postgresql-accessing-mysql-as-a-data-source-using-mysql_fdw/

Weitere Informationen – MariaDB Connect Engine

- CONNECT - MariaDB Knowledge Base
<https://mariadb.com/kb/en/connect/>
- Installing the CONNECT Storage Engine - MariaDB Knowledge Base
<https://mariadb.com › kb › en › installing-the-connect-storage-engine>
- Die MariaDB Connect Engine: Bitte bleiben Sie dran, die Verbindung wird hergestellt.
<https://blog.ordix.de/die-mariadb-connect-engine-bitte-bleiben-sie-dran-die-verbindung-wird-hergestellt>

ORDIX AG
Aktiengesellschaft für Softwareentwicklung,
Schulung, Beratung und Systemintegration

Zentrale Paderborn
Karl-Schurz-Straße 19a
33100 Paderborn
Tel.: 05251 1063-0
Fax: 0180 1 67349 0

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

info@ordix.de
<https://www.ordix.de/>

**Vielen Dank für
Ihre Aufmerksamkeit**

Markus Flechtner
mfl@ordix.de
[@markusdba](#)
@markusdba@mastodon.social
Markusdba.net|.de