ORDIX®

Gib mir deine Daten! –

So sprechen Oracle, PostgreSQL und MariaDB miteinander

Markus Flechtner

DOAG-Konferenz 2026

Nürnberg



Foto von Brecht Corbeel auf Unsplash

info@ordix.de

www.ordix.de

Markus Flechtner

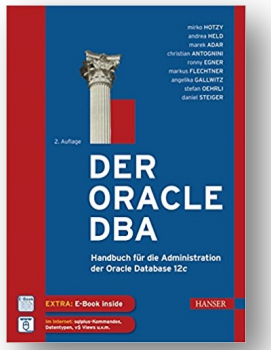
ORDIX®



- Diplom-Mathematiker
- Oracle seit 1990, PostgreSQL seit 2018
- Bisher: Entwickler, Oracle-Field-Support, DBA, Consultant
- Aktuell: Principal Consultant & Abteilungsleiter bei der ORDIX AG
- Schwerpunkte
  - Hochverfügbarkeit
  - Migrationen und Upgrades
  - Multitenant
- @markusdba.bsky.social
- @markusdba@mastodon.social
- markusdba.de|.net mfl@ordix.de

ora2know
The German Oracle Database Community

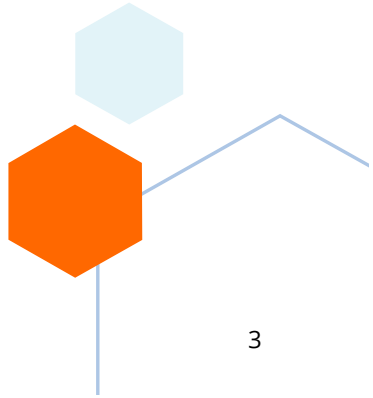
Oracle ACE Pro



Warum müssen Datenbanken Daten austauschen?

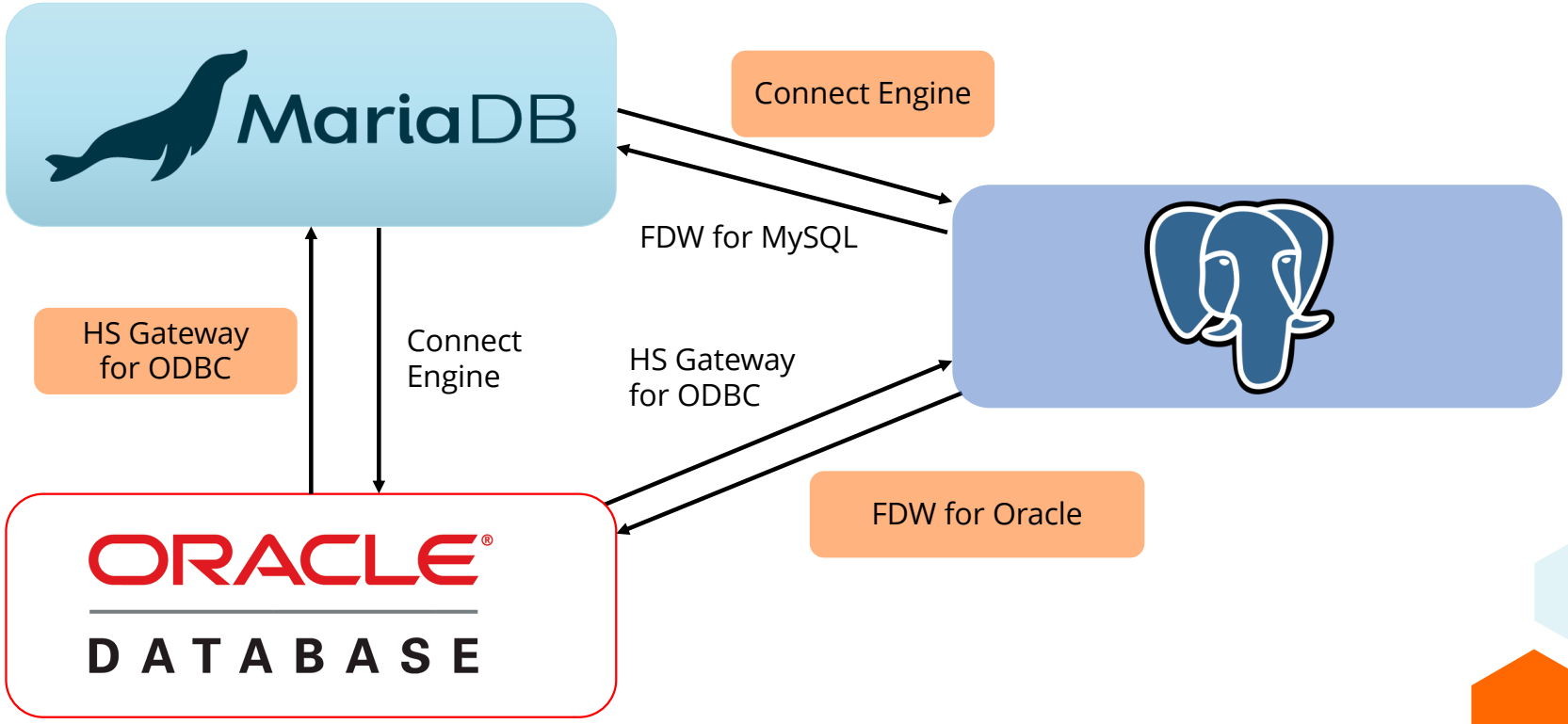
ORDIX®

- **Migration der Daten/Datentransfer/Datenaustausch**
  - Einmaliger Datentransfer
  - Weg von proprietären Datenbanken wie Oracle, hin zu Open Source-Lösungen
  - Ziel: Digitale Souveränität, Lizenzkosten sparen
- **Hybrider Betrieb (mehrere Datenbanken)**
  - Vermeidung redundanter Datenhaltung
  - Datenabgleich
  - Übergreifendes Reporting (Heterogene Umgebungen)



Worum geht es?

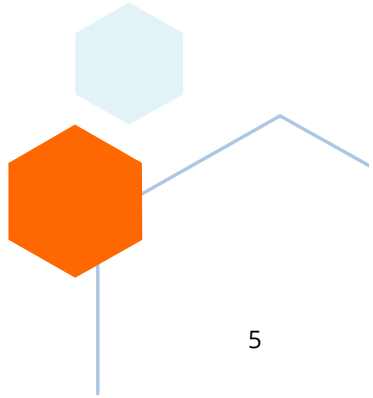
ORDIX®



Technologien für den Zugriff auf andere Datenbanken

ORDIX®

- **PostgreSQL**
  - Foreign Data Wrapper (oracle_fdw bzw. mysql_fdw)
  - Gemäß SQL/MED-Standard („Management of External Data“)
- **Oracle**
  - "Oracle Database Gateway for ODBC", (auch „Heterogeneous Services Gateway for ODBC" oder "dg4odbc" genannt), kurz ODBC-Gateway
- **MariaDB**
  - Connect Engine
  - Angelehnt an den SQL/MED-Standard („Management of External Data“)
  - Nutzt ebenfalls ODBC



Labor-Setup

ORDIX®

ohm	planck	meitner
4 vCPU 16 GB RAM	4 vCPU 8 GB RAM	4 vCPU 8 GB RAM
Oracle Linux 8 Oracle 19c 19.32	Alma Linux 8 PostgreSQL 18.6	Alma Linux 8 MariaDB 12.3

- Folien + ausführliches Installationsbeispiel via https://www.markusdba.de/

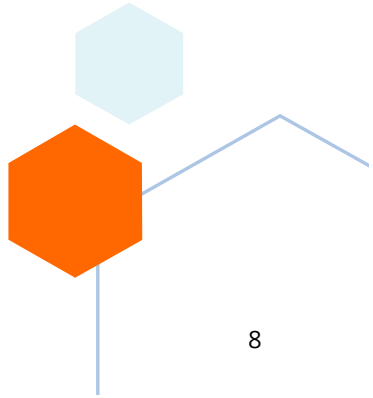
Agenda


PostgreSQL
Foreign Data Wrapper
Oracle
ODBC Gateway
MariaDB
Connect Engine
Vergleich
Zusammenfassung
Weitere Infos

Foreign Data Wrapper

ORDIX®

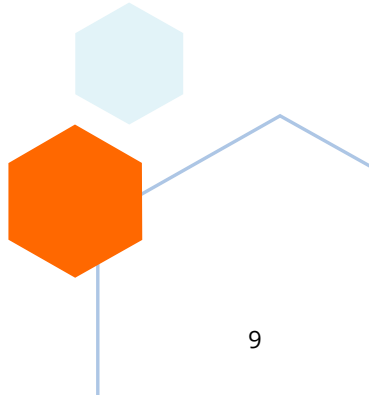
- Foreign Data Wrapper (FDW) dienen zum Zugriff von PostgreSQL heraus auf andere Datenquellen
- "Mitgeliefert" (contrib-Paket) werden
  - FDW für PostgreSQL (postgres_fdw)
  - FDW für Datei-Zugriffe (file_fdw)
- Weitere Quellen
  - PostgreSQL Extension Network https://pgxn.org
  - Projektseiten auf github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



Foreign Data Wrapper – Komponenten / Ablauf der Installation

ORDIX®

Client-Software der Zieldatenbank	z.B. Oracle Instant Client, MariaDB-client-Package etc.
Foreign-Data-Wrapper-Software	**dnf install …** oder **configure;make;make install ..**
Erweiterung (Extension)	**create extension …**
"Foreign Server" - Verbindungsinformationen für die Datenquelle	**create server …**
"User Mapping" - mit welchem Benutzer verbindet sich ein PostgreSQL-Benutzer zur Datenquelle?	**create user mapping …**
Tabellen-Mapping - für jede Tabelle der Datenquelle muss eine **"Foreign Table"** angelegt werden	**create foreign table …** oder **import foreign schema …**

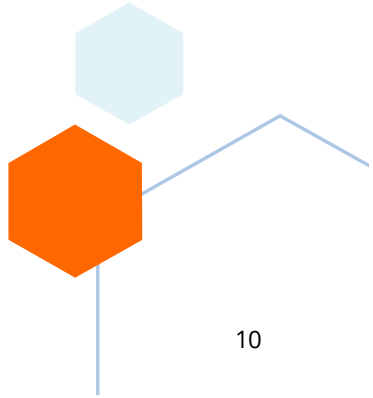


Foreign Data Wrapper für Oracle

ORDIX®

- Webseite: https://github.com/laurenz/oracle_fdw
- Aktuelle Version (Stand September 2026): Version 2.9.0 vom 08.06.2026
- Lizenz: PostgreSQL-Lizenz
- Sehr ausführlich dokumentiert ☺

- Oracle-Client muss auf dem PostgreSQL-Server installiert sein
  - Instant-Client ist ausreichend



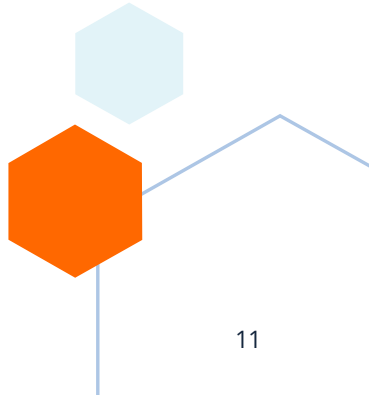
Vorbereitungen

ORDIX®

- notwendige Komponenten installieren

```
# dnf install -y perl-ExtUtils-MakeMaker perl-DBI perl-DBD-Pg \
  perl-tests perl-devel perl-DBI scl-utils redhat-rpm-config \
  gcc gcc-c++ make glibc-devel
```

- Oracle Client installieren
  - Instant Client: "Basic-Paket" und "SDK-Paket"
  - SQL*Plus-Paket kann bei der Fehlersuche helfen



Installation oracle_fdw

ORDIX®

- Download der Software

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

- Umgebung setzen

```
# export ORACLE_HOME=/usr/lib/oracle/19.32/client64/
# export LD_LIBRARY_PATH=$ORACLE_HOME/lib
# export PATH=/usr/pgsql-18/bin:$PATH
```

- Installation

```
# make
# make install
```

Vorbereitungen in PostgreSQL (1) – Server-Konfiguration

ORDIX®

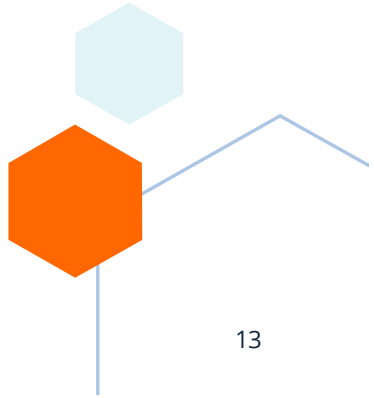
- **Die Oracle-Client-Libraries müssen im Suchpfad sein, wenn der PostgreSQL-Cluster gestartet wird**
- Variante 1 (systemctl / override.conf)

```
[Service]
Environment=ORACLE_HOME=/usr/lib/oracle/19.32/client64/
Environment=LD_LIBRARY_PATH=/usr/lib/oracle/19.32/client64/lib
```

- Variante 2 (Start über die Shell / pg_ctl)

```
$ export ORACLE_HOME=/usr/lib/oracle/19.32/client64/
$ export LD_LIBRARY_PATH=$ORACLE_HOME/lib

$ pg_ctl start ...
```



Vorbereitungen in PostgreSQL (2) – Erweiterung & Foreign Server

ORDIX®

- Erweiterung anlegen (pro Datenbank)

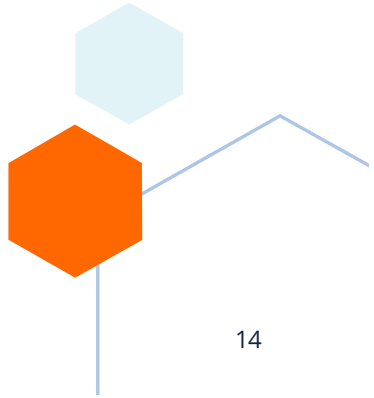
```
oracle_fdw_db=# create extension oracle_fdw;
CREATE EXTENSION
```

- Foreign Server anlegen

```
oracle_fdw_db=# create server ora_demopdb foreign data wrapper oracle_fdw
oracle_fdw_db-# options (dbserver '//ohm.markusdba.internal:1521/DEMOPDB');
CREATE SERVER
```

- Ggf. anderen Benutzern die Nutzung erlauben

```
oracle_fdw_db=# grant usage on foreign server ora_demopdb to public;
GRANT
```



Vorbereitungen in PostgreSQL (3) – User Mapping

ORDIX®

- User Mapping anlegen (pro PostgreSQL-Benutzer)

```
oracle_fdw_db=> create user mapping for ext_scott server ora_demopdb
oracle_fdw_db-> options (user 'scott', password 'tiger');
CREATE USER MAPPING
```

- **ACHTUNG:** Das Password steht im Klartext im Systemkatalog

```
oracle_fdw_db=> \deu+
                    List of user mappings
   Server     | User name |             FDW options
--------------+-----------+-------------------------------------
 ora_demopdb | ext_scott | ("user" 'scott', password 'tiger')
(1 row)
```

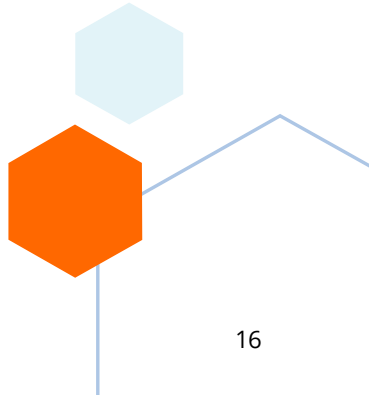
  - Workaround: Secure External Password Store

Table Mapping Oracle → PostgreSQL

ORDIX®

- Für jede Remote-Tabelle in der Oracle-Datenbank muss in PostgreSQL eine "Foreign Table" angelegt werden
  - Zuordnung der Spalten (PostgreSQL-ForeignTable → Oracle Tabelle) erfolgt über die Reihenfolge im CREATE-Befehl
  - Primary-Key-Spalten mit der Option "key=true", markieren sonst kein UPDATE oder DELETE möglich

- Zwei Varianten
  - IMPORT FOREIGN SCHEMA (empfohlen, macht alles "automatisch")
  - CREATE FOREIGN TABLE (aufwändiger)



Schema-Import

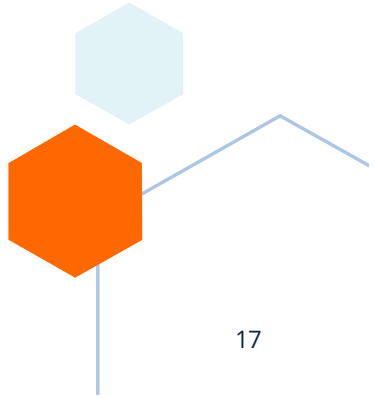
ORDIX®

```
oracle_fdw_db=> import foreign schema "SCOTT" from server ora_demopdb
oracle_fdw_db-> into ext_scott
oracle_fdw_db-> options (case 'lower');
IMPORT FOREIGN SCHEMA

oracle_fdw_db=> select count(*) from ext_scott.emp;
 count
-------
    14
(1 row)

oracle_fdw_db=> update ext_scott.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



CREATE FOREIGN TABLE

ORDIX®

- Foreign Table anlegen

```
oracle_fdw_db=> CREATE FOREIGN TABLE dept (
oracle_fdw_db-> deptno         integer   NOT NULL,
oracle_fdw_db-> dname          character varying(30),
oracle_fdw_db-> loc            character varying(13)
oracle_fdw_db-> ) SERVER ora_demopdb OPTIONS (schema 'SCOTT', table 'DEPT');
CREATE FOREIGN TABLE

oracle_fdw_db=> select count(*) from dept;
 count
-------
     4
(1 row)
```

GROSSSCHRIFT ist wichtig

CREATE FOREIGN TABLE - Primary-Key-Spalten

ORDIX®

- DML auf eine Foreign Table (ohne PK-Informationen)

```
oracle_fdw_db=> 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 (Ergänzung PK-Info)

```
oracle_fdw_db=> alter foreign table dept alter "deptno"
oracle_fdw_db-> options (ADD key 'true');
ALTER FOREIGN TABLE
```

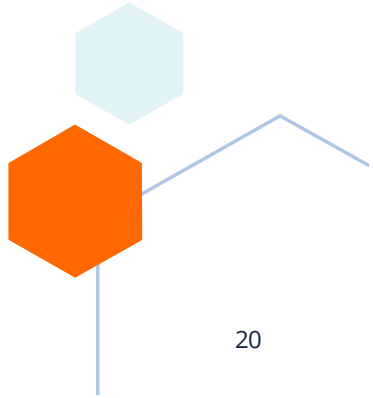
- 2. Versuch

```
oracle_fdw_db=> update dept set loc='PADERBORN' where deptno=10;
UPDATE 1
```

Oracle_fdw - Sonstiges

ORDIX®

- Ausführungspläne & Performance
  - Push-Down von WHERE und ORDER BY ➔ werden auf der Oracle-Seite ausgeführt
  - Joins zwischen **zwei** FDW-Tabellen werden auf der Oracle-Seite ausgeführt
  - Statistiken auf Foreign Tables müssen manuell gesammelt werden
- Default Isolation Level ist "SERIALIZABLE" ➔ führt ggf. zu Fehler "ORA-08177: can't serialize access for this transaction"), kann beim Anlegen des Foreign Servers festgelegt werden
- Datentypen
  - Standard-Datentypen werden unterstützt
  - Spatial-Datentypen nur wenn PostGIS installiert ist
- Weitere Informationen: https://github.com/laurenz/oracle_fdw#4-usage

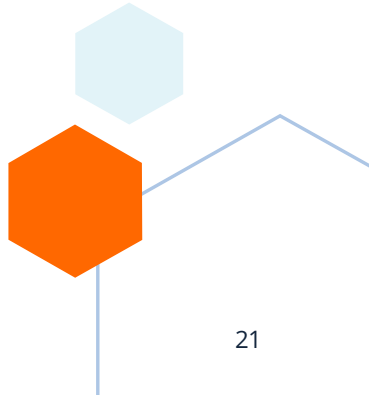


Foreign Data Wrapper for MySQL/MariaDB

ORDIX®

- Entwickelt von Enterprise DB (EDB)
  - Github: https://github.com/EnterpriseDB/mysql_fdw
  - Lizenz: PostgreSQL-Lizenz
  - Aktuelle Version: 2.9.3 (September 2025)
- MariaDB-/MySQL-Client muss vorab installiert sein
- Installation via rpm (PostgreSQL-Repository):
  - Paket abhängig von der PostgreSQL-Version
  - Das Package erwartet die MySQL-Binaries, daher muss bei der Verwendung von MariaDB mit den Optionen "--nobest" und "--allowerasing" gearbeitet werden

```
# dnf install -y mysql_fdw_18 --nobest --allowerasing
```



Foreign Data Wrapper for MySQL/MariaDB – Konfiguration

ORDIX®

1. Erweiterung anlegen

```
mariadb_fdw_db=# create extension mysql_fdw;
CREATE EXTENSION
```

2. Foreign Server anlegen

```
mariadb_fdw_db=# create server mariadb  foreign data wrapper mysql_fdw
mariadb_fdw_db-# options (host 'meitner.markusdba.internal', port '3306');
CREATE SERVER
```

3. User Mapping anlegen

```
mariadb_fdw_db=# CREATE USER MAPPING FOR mariadb_ext SERVER mariadb
mariadb_fdw_db-# OPTIONS (username 'fdw_user', password 'geheim');
CREATE USER MAPPING
```

- Auch hier gilt: mittel "\deu+" kann der Superuser das Passwort auslesen

Foreign Data Wrapper for MySQL/MariaDB – Foreign Table

ORDIX®

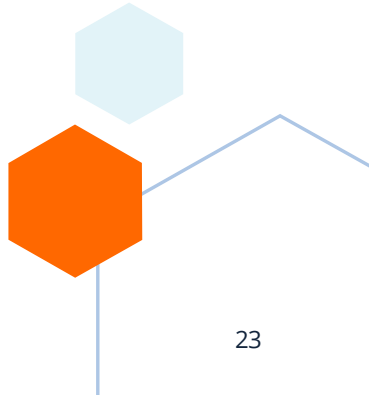
4. Table Mapping

```
mariadb_fdw_db=> IMPORT FOREIGN SCHEMA sakila LIMIT TO (actor, customer)
mariadb_fdw_db-> FROM SERVER mariadb INTO mariadb_ext;
IMPORT FOREIGN SCHEMA
```

- Wenn man mit "CREATE FOREIGN TABLE" arbeitet: "OPTIONS (dbname '<name>', table_name '<table>"),

5. Test

```
mariadb_fdw_db=> SELECT * FROM mariadb_ext.actor LIMIT 1;
 actor_id | first_name | last_name |      last_update
----------+------------+-----------+-----------------------
        1 | PENELOPE   | GUINESS   | 2006-02-15 04:34:33+01
(1 row)
mariadb_fdw_db=> update mariadb_ext.actor set first_name='MIA'
mariadb_fdw_db-> where actor_id=1;
UPDATE 1
```



Meta-Befehle rund um Foreign Data Wrapper

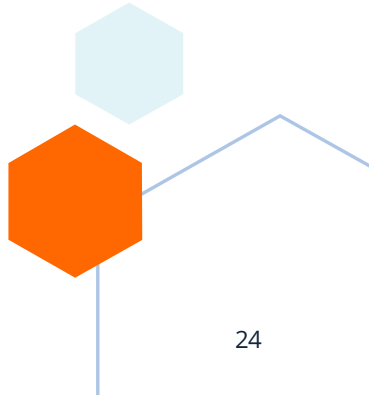
ORDIX®

- Meta-Befehle

```
\dE[S+] [PATTERN]      list foreign tables
\des[+] [PATTERN]      list foreign servers
\det[+] [PATTERN]      list foreign tables
\deu[+] [PATTERN]      list user mappings
\dew[+] [PATTERN]      list foreign-data wrappers
```

- Beispiel

```
mariadb_fdw_db=> \dE
                List of foreign tables
   Schema     |   Name   |     Type       |    Owner
--------------+----------+----------------+-------------
 mariadb_ext | actor    | foreign table | mariadb_ext
 mariadb_ext | customer | foreign table | mariadb_ext
(2 rows)
```

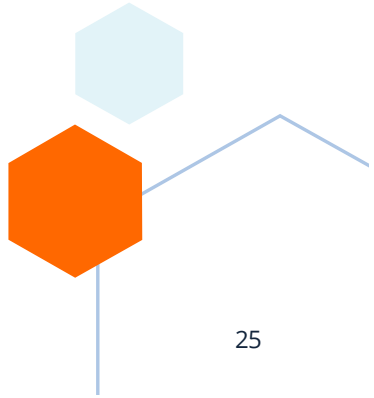


Foreign Data Wrapper im System Catalog

ORDIX®

- pg_foreign_data_wrapper
- pg_foreign_server
- pg_foreign_table
- pg_user_mapping (als PostgreSQL-Superuser: Passworte im Klartext)

```
mariadb_fdw_db=> select srvname,srvoptions from pg_foreign_server;
 srvname |                  srvoptions
---------+-----------------------------------------------
 mariadb | {host=meitner.markusdba.internal,port=3306}
(1 row)
```



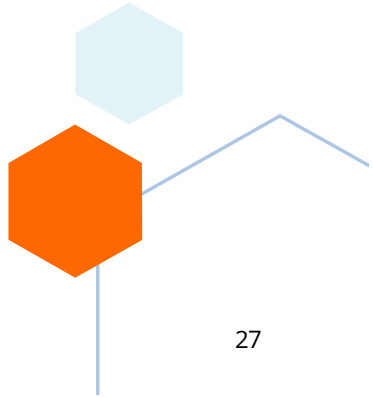
Agenda


PostgreSQL
Foreign Data Wrapper
Oracle
ODBC Gateway
MariaDB
Connect Engine
Vergleich
Zusammenfassung
Weitere Infos

Oracle Heterogeneous Services (HS)

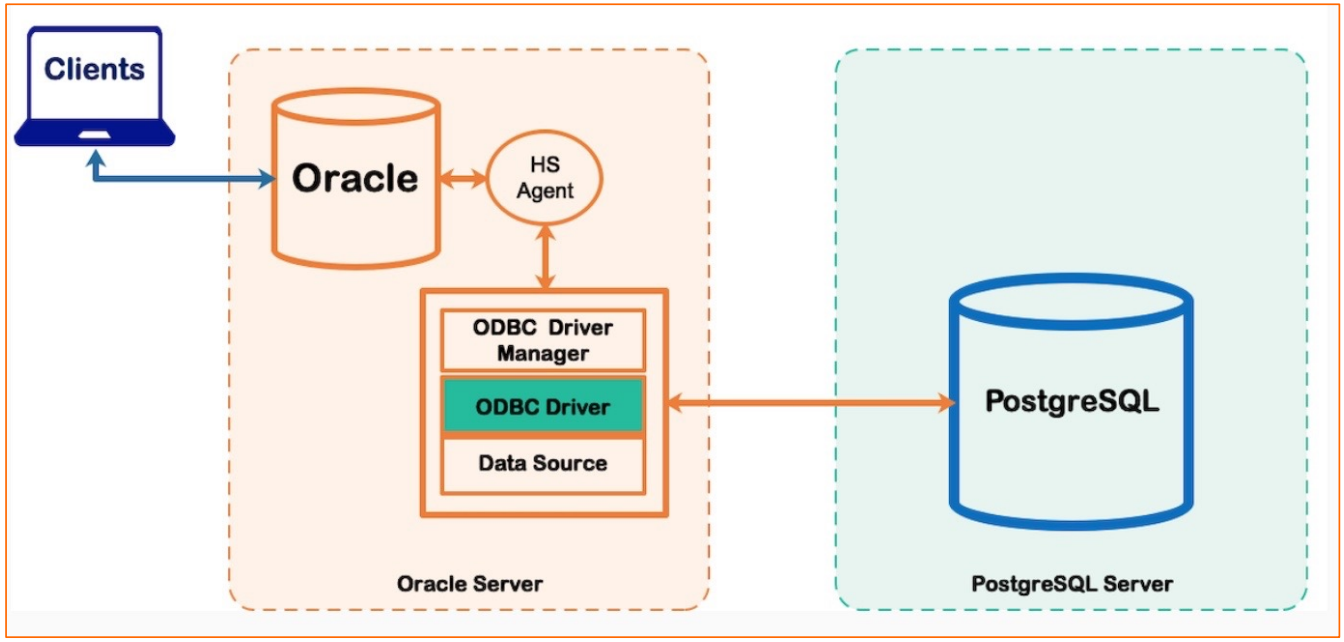
ORDIX®

- Unter dem Namen "Oracle Heterogeneous Services" (HS) platziert Oracle die Gateways zu anderen Datenbanken bzw. Systemen.
- Auswahl:
  - DRDA (Distributed Relational Database Architecture, DB2)
  - Informix
  - SQL Server
  - **ODBC** (in der Datenbank-Lizenz enthalten ☺ (außer FREE-Edition ☹), Quelle: Oracle AI Database Licensing Information, Stand Oktober 2026)



Oracle-ODBC-Gateway zu PostgreSQL - Architektur

ORDIX®



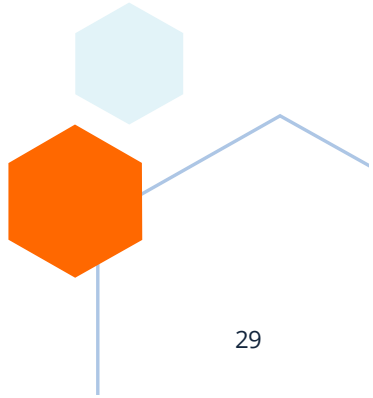
Oracle-SQL
⬇
DB-Link
⬇
Gateway-Listener
⬇
DG4ODBC (HS Agent)
⬇
unixODBC
⬇
PostgreSQL-ODBC-Treiber
⬇
PostgreSQL

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

Ablauf

ORDIX®

1. Software-Komponenten installieren (Client-Software, ODBC-Komponenten)
2. ODBC konfigurieren
3. ODBC-Datenquelle konfigurieren
4. Gateway konfigurieren
5. Gateway-Listener konfigurieren
6. Tnsnames.ora-Eintrag
7. Datenbank-Link anlegen
8. Zugriff auf die "remote-Tabelle" über Datenbank-Link



Installation (1) – Software-Komponenten

ORDIX®

1. Zusätzliche Software-Komponenten auf dem Oracle Datenbank-Server installieren

```
# dnf install -y unixODBC unixODBC-devel postgresql18-odbc
```

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

```
$ odbcinst -j
unixODBC 2.3.7
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

ORDIX®

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

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

Pfad ggf. anpassen

- Meist schon durch unixODBC entsprechend vorbereitet

Installation (3) – Datenquelle in odbc.ini definieren

ORDIX®

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

- Minimalbeispiel

```
[pgscott]
Database        = scott_db
Description     = Connection to PostgreSQL Cluster PG18
Driver          = PostgreSQL
Servername      = planck.markusdba.internal
Port            = 5432
```

Verweis auf odbcinst.ini

- Der PostgreSQL-ODBC-Treiber hat per Default `UseDeclareFetch=0`
  ➔„the driver will retrieve the entire result set“.
  ➔ Konsequenzen für den Speicherbedarf des Gateways beachten
- Empfehlung: UseDeclareFetch = 1.
- Übersicht über alle Konfigurationsparameter: https://odbc.postgresql.org/docs/config.html

Installation (4) – Datenquelle testen

ORDIX®

4. Testen der ODBC-Datenquelle (DSN)

```
[root@ohm oracle]# isql -v pgscott scott tiger
+---------------------------------------+
| Connected!                            |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
+---------------------------------------+
SQL> quit
```

Reihenfolge:
DSN User Password

- "-v " (verbose) für ausführliche Fehlermeldungen
- (nicht vergessen, die pg_hba.conf auf dem PostgreSQL-Server passend zu konfigurieren)

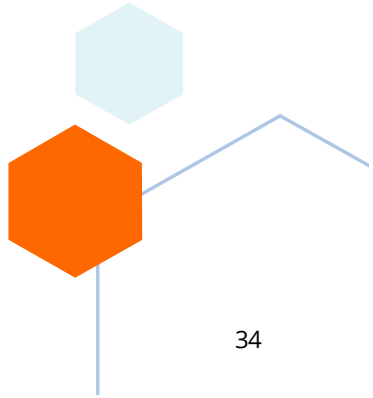
Installation (5) – Gateway: init-Datei

ORDIX®

5. Init-Datei für das Gateway definieren

- Im Verzeichnis $ORACLE_HOME/hs/admin liegt eine Beispieldatei initdg4odbc.ora
- AMERICAN_AMERICA.WE8ISO8859P15 als Sprache verwenden, damit Fehlermeldungen des unixODBC angezeigt werden! – siehe auch MOS-Note 756186.1 / KB741074
- **Wenn mit Read-Only-Oracle-Home gearbeitet wird, dann liegen die init-Dateien für die Gateways in $ORACLE_BASE/homes/<HOME_NAME>/hs/admin**

```
$ cat initpgscott.ora
HS_FDS_CONNECT_INFO = pgscott
HS_FDS_TRACE_LEVEL = ON
HS_FDS_SHAREABLE_NAME=/usr/lib64/libodbc.so
HS_LANGUAGE = AMERICAN_AMERICA.WE8ISO8859P15
set ODBCINI=/etc/odbc.ini
```



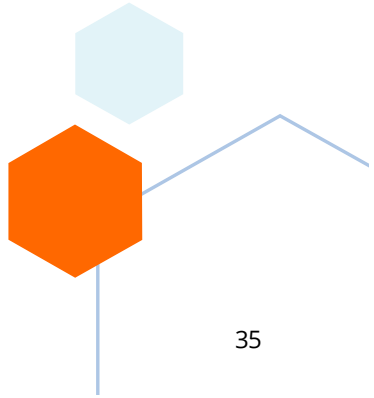
Installation (6) – Gateway-Listener

ORDIX®

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

- Alternativ den normalen Listener "LISTENER" verwenden und Gateway mittels SID_LIST_LISTENER einbinden

```
$ cat $TNS_ADMIN/listener.ora
LISTENERPG=
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = ohm.markusdba.internal)(PORT = 1522))
)
  )
SID_LIST_LISTENERPG=
   (SID_LIST =
     (SID_DESC =
       (SID_NAME=pgscott)
       (ORACLE_HOME=/u00/app/oracle/product/19.32-ee)
       (PROGRAM=dg4odbc)
       (ENVS="LD_LIBRARY_PATH=/usr/lib64:/u00/app/oracle/product/19.32-ee/lib")
     )
   )
```

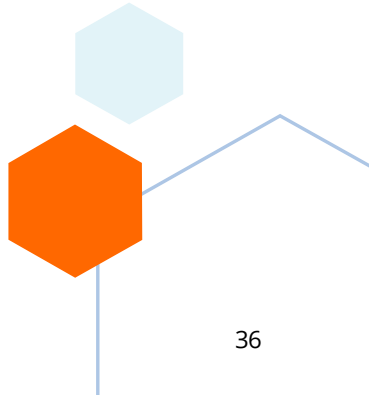


Installation (7) – Gateway-Listener starten

ORDIX®

7. Listener für das Gateway starten

```
[oracle@ohm admin]$ lsnrctl start listenerpg
LSNRCTL for Linux: Version 19.0.0.0.0 - Production on 13-SEP-2026 20:49:43
Copyright (c) 1991, 2026, Oracle.  All rights reserved.
Starting /u00/app/oracle/product/19.32-ee/bin/tnslsnr: please wait...
[…]
Connecting to
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=ohm.markusdba.internal)(PORT=1522)))
STATUS of the LISTENER
------------------------
Alias                     listenerpg
Version                   TNSLSNR for Linux: Version 19.0.0.0.0 - Production
[…]
Services Summary...
Service "pgscott" has 1 instance(s).
  Instance "pgscott", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
```



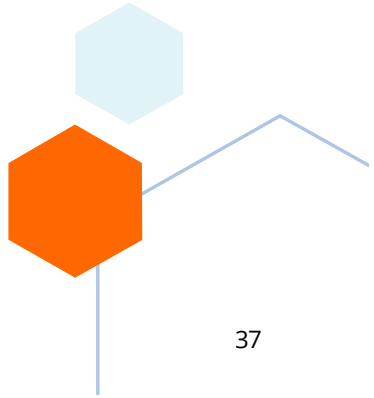
Installation (8) – tnsnames.ora-Eintrag definieren & testen

ORDIX®

8. tnsnames.ora-Eintrag definieren und testen

```
[oracle@ohm ~]$ cat $TNS_ADMIN/tnsnames.ora
[…]
PGSCOTT=
  (DESCRIPTION=
  (ADDRESS=
  (PROTOCOL=TCP)(HOST = ohm.markusdba.internal)(PORT = 1522))
  (CONNECT_DATA= (SID=pgscott)) (HS=OK)
)

[oracle@ohm ~]$ tnsping pgscott
TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 27-OCT-2026 21:23:34
Copyright (c) 1997, 2026, Oracle.  All rights reserved.
Used parameter files:
/u00/app/oracle/network/admin/sqlnet.ora
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION= (ADDRESS= (PROTOCOL=TCP)(HOST =
ohm.markusdba.internal)(PORT = 1522)) (CONNECT_DATA= (SID=pgscott)) (HS=OK))
OK (0 msec)
```



Datenbank-Link anlegen

ORDIX®

9. Datenbank-Link anlegen und testen

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

SQL> select count(*) from "scott"."emp"@pgscott;
  COUNT(*)
----------
        14

SQL> update "scott"."emp"@pgscott set "name"='KONG' where "name"='KING';
1 row updated.
```

- PostgreSQL-Namen sind in Kleinschrift ➔ doppelte Anführungszeichen
- Tipp: auf Oracle-Ebene mit Views arbeiten, mit Namen in Großschrift

Achtung bei den Datentypen

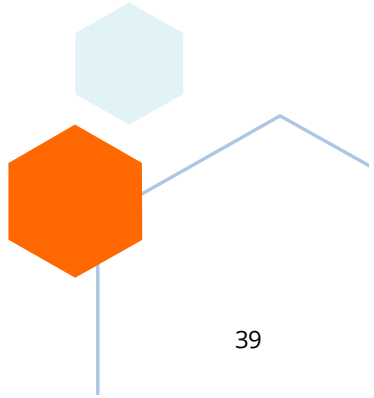
ORDIX®

```
SQL> select "rolname","rolcanlogin" from "pg_roles"@pgscott;

rolname                          rolcanlogin
-------------------------------  ---------------
[…]
pg_checkpoint                    0
postgres                         1
scott                            1
monitor_system_stats             0
```

0 = false

1 = true



ODBC-Gateway für MySQL/MariaDB (1)

ORDIX®

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

```
[mariadb]
Database        = sakila
Description     = Connection to MariaDB
Driver          = MariaDB
Servername      = meitner.markusdba.internal
Port            = 3306
UID             = fdw_user
PWD             = geheim
```

  - Testen mit "isql -v mariadb", nach dem Test UID und PWD aus /etc/odbc.ini löschen

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

ORDIX®

- Init-Datei für das ODBC-Gateway - $ORACLE_HOME/hs/admin/initmariadb.ora bzw. $ORACLE_BASE/homes/<HOME_NAME>/hs/admin/initmariadb.ora

```
# Name of the data source
HS_FDS_CONNECT_INFO=mariadb
# Trace Level (other levels: OFF,ON  or DEBUG)
HS_FDS_TRACE_LEVEL=OFF
# full path name of odbc driver manager or driver
HS_FDS_SHAREABLE_NAME=/usr/lib64/libodbc.so
# set language to American, character set depending on client char set
HS_LANGUAGE = AMERICAN_AMERICA.WE8ISO8859P15
# full path name of the odbc initialization file
set ODBCINI=/etc/odbc.ini
```

Aus /etc/odbc.ini

ODBC-Gateway für MySQL/MariaDB (3) – Gateway-Listener

ORDIX®

- Eintrag in der listener.ora:

```
$ cat listener.ora
LISTENERMARIADB=
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = ohm.markusdba.internal)(PORT = 1523))
  ))
SID_LIST_LISTENERMARIADB=
   (SID_LIST =
     (SID_DESC =
       (SID_NAME=mariadb)
       (ORACLE_HOME=/u00/app/oracle/product/19.32-ee)
       (PROGRAM=dg4odbc)
       (ENVS="LD_LIBRARY_PATH=/usr/lib64:/u00/app/oracle/product/19.32-ee/lib")
     )
   )
```

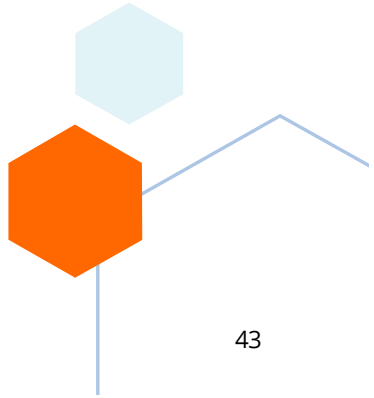
- Listener starten ("lsnrctl start listenermariadb")

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

ORDIX®

- Eintrag in der tnsnames.ora

```
[oracle@ohm admin]$ cat tnsnames.ora
[…]
mariadb=
  (DESCRIPTION=
  (ADDRESS=
  (PROTOCOL=TCP)(HOST = ohm.markusdba.internal)(PORT = 1523))
  (CONNECT_DATA= (SID=mariadb)) (HS=OK)
)
[oracle@ohm admin]$ tnsping mariadb
TNS Ping Utility for Linux: Version 19.0.0.0.0 - Production on 15-SEP-2026 23:01:54
Copyright (c) 1997, 2026, Oracle.  All rights reserved.
Used parameter files:
/u00/app/oracle/network/admin/sqlnet.ora
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION= (ADDRESS= (PROTOCOL=TCP)(HOST =
ohm.markusdba.internal)(PORT = 1523)) (CONNECT_DATA= (SID=mariadb)) (HS=OK))
OK (0 msec)
```



ODBC-Gateway für MySQL/MariaDB (5) – Datenbank-Link & Test

ORDIX®

- Datenbank-Link anlegen und testen

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

SQL> select count(*) from "actor"@mariadb;
COUNT(*)
----------
       200
SQL> update "actor"@mariadb set "first_name"='THYRA' where "actor_id"=200;
1 row updated.
```

- Wie bei PostgreSQL: Namen in Anführungszeichen setzen und ggf. passende Views anlegen

Agenda

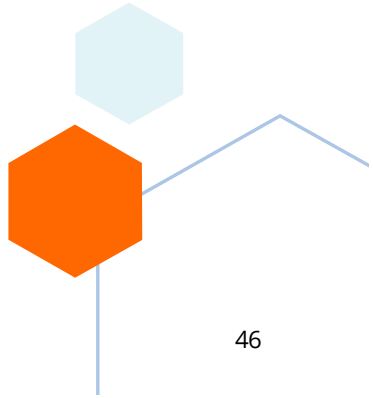

PostgreSQL
Foreign Data Wrapper
Oracle
ODBC Gateway
MariaDB
Connect Engine
Vergleich
Zusammenfassung
Weitere Infos

MariaDB Connect Engine

ORDIX®

- Zugriff auf externe Datenquellen von MariaDB aus
- Teil des MariaDB-Servers
- Datenquellen
  - **ODBC** (nutzen wir für die Verbindung zu Oracle und PostgreSQL)
  - JDBC
  - ...
- Lizenz: GPLv2-Lizenz (GNU General Public License, Version 2)

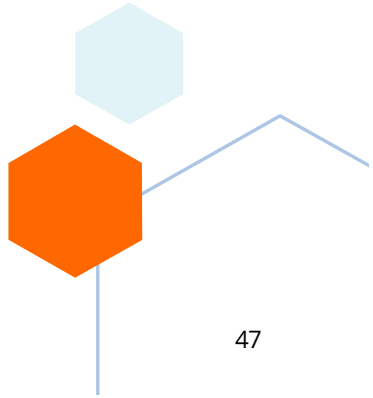
- Bei MySQL gibt es nichts Vergleichbares



Ablauf

ORDIX®

1. Software-Komponenten installieren
2. Connect-Engine in der Datenbank installieren
3. ODBC-Treiber konfigurieren
4. ODBC-Datenquelle konfigurieren
5. Remote Tabelle in MariaDB anlegen ("`CREATE TABLE ... ENGINE=CONNECT ...`")



MariaDB Connect Engine - Installation

ORDIX®

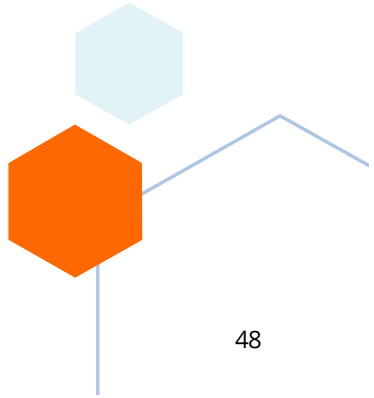
- RPM-Paket installieren

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

- Engine in der Datenbank installieren

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

- Client Software der jeweiligen Datenbank
  - Oracle Instant Client (Basic- + ODBC-Package)
  - PostgreSQL-Client inkl. ODBC-Paket



MariaDB Connect Engine - vorhandene Storage-Engines

ORDIX®

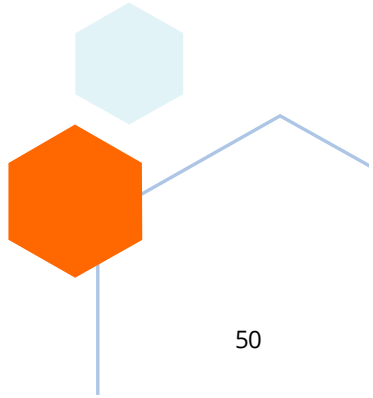
```
MariaDB [(none)]> show engines;
+--------------------+---------+----------------------------------------------------------------------------------------------+
| Engine             | Support | Comment                                                                                      |
+--------------------+---------+----------------------------------------------------------------------------------------------+
| MEMORY             | YES     | Hash based, stored in memory, useful for temporary tables                                    |
| CSV                | YES     | Stores tables as CSV files                                                                   |
| MRG_MyISAM         | YES     | Collection of identical MyISAM tables                                                        |
| Aria               | YES     | Crash-safe tables with MyISAM heritage. Used for internal temporary tables and …             |
| 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                    |
| InnoDB             | DEFAULT | Supports transactions, row-level locking, foreign keys and encryption for tables             |
| SEQUENCE           | YES     | Generated tables filled with sequential values                                               |
| PERFORMANCE_SCHEMA | YES     | Performance Schema                                                                           |
+--------------------+---------+----------------------------------------------------------------------------------------------+
```

(Ausgabe gekürzt)

MariaDB Connect Engine - ODBC vorbereiten

ORDIX®

- Analog zum Oracle-ODBC-Gateway
  - /etc/odbc.ini
  - /etc/odbcinst.ini
  - Testen mit isql
- **LD_LIBRARY_PATH=$ORACLE_HOME/lib** muss im Pfad sein, wenn der MariaDB-Server gestartet wird
  - ➔ Umgebungsvariable oder override.conf (systemd)



MariaDB Connect Engine - Verbindung zu PostgreSQL

ORDIX®

```
MariaDB [sakila]> create table emp_pg_remote  engine=connect table_type=ODBC
                  tabname='emp' dbschema='scott'
                  connection='dsn=postgresql_scottdb';
Query OK, 0 rows affected (0.048 sec)
MariaDB [sakila]> select * from emp_pg_remote;
+-------+--------+-----------+---------+------------+---------+---------+--------+
| empno | name   | job       | manager | hiredate   | salary  | comm    | deptno |
+-------+--------+-----------+---------+------------+---------+---------+--------+
|  7839 | KING   | PRESIDENT |    NULL | 1981-11-17 | 5000.00 |    NULL |     10 |
[…]
MariaDB [sakila]> update emp_pg_remote set name='KONG' where name='KING';
Query OK, 0 rows affected, 1 warning (0.023 sec)
Rows matched: 0  Changed: 0  Warnings: 1
MariaDB [sakila]> select * from emp_pg_remote;
+-------+--------+-----------+---------+------------+---------+---------+--------+
| empno | name   | job       | manager | hiredate   | salary  | comm    | deptno |
+-------+--------+-----------+---------+------------+---------+---------+--------+
|  7839 | KONG   | PRESIDENT |    NULL | 1981-11-17 | 5000.00 |    NULL |     10 |
[…]
```

Rückmeldung falsch, Ergebnis korrekt!

MariaDB Connect Engine - Verbindung zu Oracle

ORDIX®

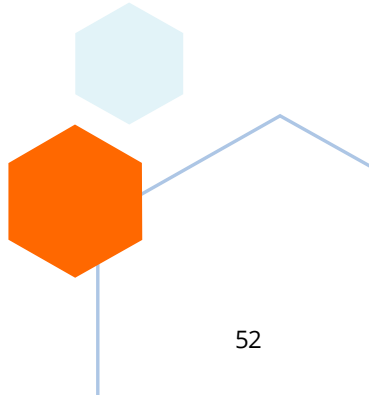
- /etc/odbcinst.ini

```
[Oracle]
Description=Oracle Driver
Driver=/usr/lib/oracle/19.32/client64/lib/libsqora.so.19.1
```

- /etc/odbc.ini

```
[oracle_scott]
Driver = Oracle
UserID = scott
Password = tiger
ServerName = //ohm.markusdba.internal:1521/DEMOPDB
```

- Test mit isql

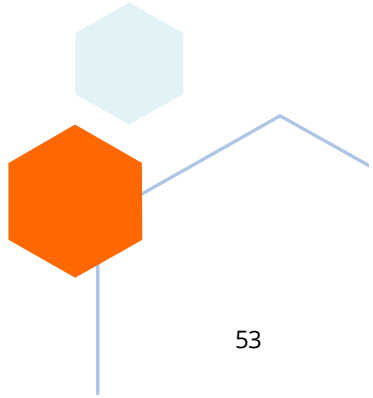


MariaDB Connect Engine - Verbindung zu Oracle

ORDIX®

- "Remote-Tabelle" anlegen

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



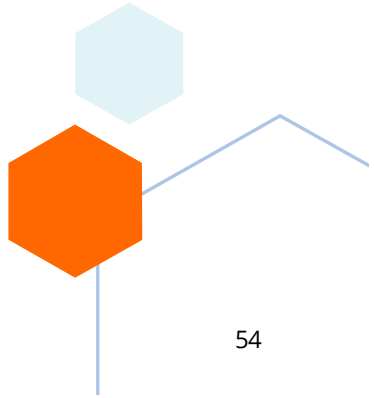
MariaDB Connect Engine - Verbindung zu Oracle - Test

ORDIX®

```
MariaDB [sakila]> select * from emp_remote where ename='KING';
+-------+-------+-----------+------+---------------------+---------+------+--------+
| EMPNO | ENAME | JOB       | MGR  | HIREDATE            | SAL     | COMM | DEPTNO |
+-------+-------+-----------+------+---------------------+---------+------+--------+
|  7839 | KING  | PRESIDENT | NULL | 1981-11-17 00:00:00 | 5000.00 | NULL |     10 |
+-------+-------+-----------+------+---------------------+---------+------+--------+
1 row in set (0.065 sec)

MariaDB [sakila]> update emp_remote set ename='KONG' where ename='KING';
Query OK, 0 rows affected, 1 warning (0.063 sec)
Rows matched: 0  Changed: 0  Warnings: 1

MariaDB [sakila]> select * from emp_remote where empno=7839;
+-------+-------+-----------+------+---------------------+---------+------+--------+
| EMPNO | ENAME | JOB       | MGR  | HIREDATE            | SAL     | COMM | DEPTNO |
+-------+-------+-----------+------+---------------------+---------+------+--------+
|  7839 | KONG  | PRESIDENT | NULL | 1981-11-17 00:00:00 | 5000.00 | NULL |     10 |
+-------+-------+-----------+------+---------------------+---------+------+--------+
1 row in set (0.060 sec)
```




PostgreSQL
Foreign Data Wrapper
Oracle
ODBC Gateway
MariaDB
Connect Engine
Vergleich
Zusammenfassung
Weitere Infos

Agenda

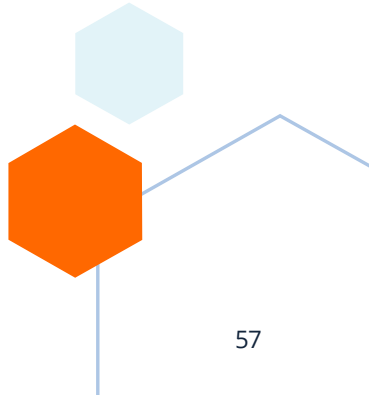
Übersicht - Drei Wege, ähnliche Bausteine

	PostgreSQL FDW	Oracle ODBC-Gateway	MariaDB Connect Engine
Treiber der Gegenseite	Oracle Instant Client bzw. MariaDB-Client	unixODBC + ODBC-Treiber + Client-Software	unixODBC + ODBC-Treiber Client-Software
Verbindung	`create server`	odbc.ini, init-Datei, Listener, tnsnames	DSN in der odbc.ini
Anmeldung	`create user mapping` **(PW!)**	Datenbank-Link	odbc.ini **(Passworte!)**
Tabellen	`import foreign schema` oder `create foreign table`	keine Abbildung, direkt über @link	je Tabelle `create table` ... `engine=connect`
Schreiben	ja, PK-Spalten markieren (Oracle)	ja	ja
Lizenz	Oracle-FDW + MySQL-FDW: PostgreSQL-Lizenz	in der DB-Lizenz enthalten, nicht in der Free-Edition	GPLv2-Lizenz (Teil des MariaDB-Servers)

Komplexität der Konfiguration

ORDIX®

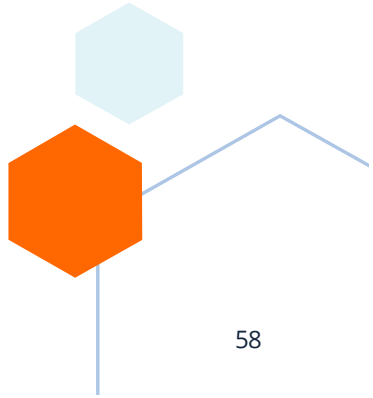
1. **PostgreSQL Foreign Data Wrapper**
    1. Alles innerhalb von PostgreSQL (CREATE SERVER, CREATE USER MAPPING, IMPORT FOREIGN SCHEMA)
2. **MariaDB-Connect-Engine**
    1. ODBC konfigurieren
    2. Alles weitere innerhalb von MariaDB
3. **Oracle ODBC-Gateway**
    1. ODBC konfigurieren
    2. Listener + tnsnames konfigurieren
    3. DB-Link anlegen, ggf. View darüberlegen (Groß-/Kleinschreibung)



Aus leidvoller Erfahrung …

ORDIX®

- Achtung bei der Standard-Schreibweise (Groß-/Kleinschreibung) der Objektnamen
  - Bei Verbindungen zwischen MariaDB und Oracle oder PostgreSQL und Oracle immer mit einer View arbeiten, die die Schreibweise korrigiert
- Datentypen
  - Standard-Datentypen funktionieren, Achtung bei langen Text-Spalten (Limits der einzelnen DBs)
  - Teilweise wird "VARCHAR2" nach "CHAR" gemappt ➔ Leerzeichen am Ende ➔ beim Transfer mit TRIM arbeiten
  - "exotische" Datentypen: Testen, Dokumentation prüfen!
- Strukturänderungen im Fremdsystem
  - ➔ Remote-Objekt (z.B. FOREIGN TABLE in PostgreSQL) neu anlegen



Ein einfacher Performance-Test - Datentransfer

ORDIX®

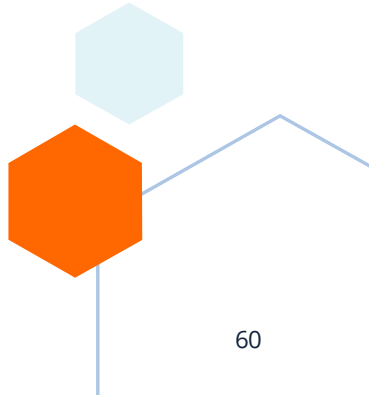
Zugriff von	... nach		select count(*) from remote (Sek.)	insert local select from remote (Sek.)	insert remote select from local (Sek)
Oracle	PostgreSQL	ODBC-GW	0,1	35,7	**ORA-2025**
Oracle	MariaDB	ODBC-GW	0,4	41,8	**ORA-2025**
PostgreSQL	Oracle	FDW	10,9	23,4	704,5
PostgreSQL	MariaDB	FDW	0,3	18,0	5700,9
MariaDB	Oracle	Connect-Engine	8,5	70,0	713,9
MariaDB	PostgreSQL	Connect-Engine	12,5	79,1	964,7

- Alle Systeme mit Default-Einstellungen (außer SGA bzw. Buffer)
- Tabelle mit 2,1 Mio. Datensätzen, Größe (PostgreSQL) 105 MB
- "ORA-2025: All tables in the SQL statement must be at the remote database" – Das Gateway unterstützt den nötigen "Callback-Link" nicht.

Zusammenfassung

ORDIX®

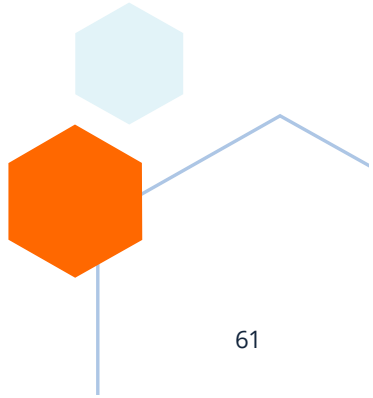
- Alle sechs Richtungen funktionieren; lesend und schreibend
- Migration, Massentransfer: auf der Zielseite ziehen (INSERT … SELECT FROM remote).
  - Schieben war 10- bis 300-mal langsamer, über das Oracle-Gateway gar nicht möglich (ORA-02025)
- Datenabgleich: auf Pushdown achten. WHERE wird weitergegeben; Aggregate liefen im Test nur beim Gateway und bei mysql_fdw remote
- Namen per View angleichen, Datentypen testen, keine gemeinsame Transaktion über das Gateway
- Am einfachsten: PostgreSQL-FDW
  - ODBC-Konfiguration ist aufwändiger, ODBC war im Test (Pull) langsamer
- Gateway: Speicher des dg4odbc-Prozesses beobachten (bis zu 5 GB im Test), Abhilfe: "UseDeclareFetch = 1" in der odbc.ini



Weitere Informationen – Oracle ODBC-Gateway

ORDIX®

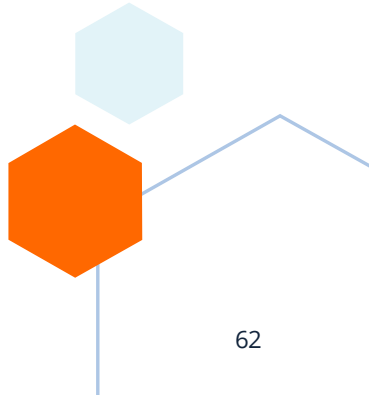
- Oracle AI Database 19c: Oracle® Database Gateway for ODBC - User's Guide
  https://docs.oracle.com/en/database/oracle/oracle-database/19/odbcu/index.html
- MOS-Notes
  - How to Resolve Common Errors Encountered while using Database Gateways (DG4IFMX, Dg4MSQL, DG4SYBS), DG4ODBC or Generic Connectivity (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)
  - DG4ODBC connection Fails With Error ORA-28500 / KB272140



Weitere Informationen – PostgreSQL Foreign Data Wrappers

ORDIX®

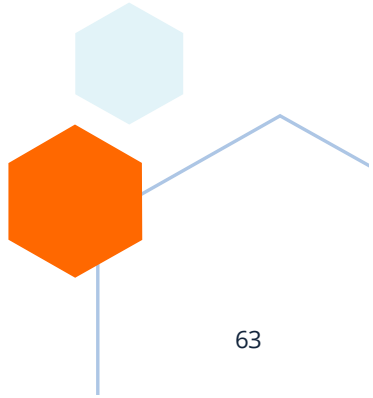
- Übersicht im PostgreSQL-Wiki: https://wiki.postgresql.org/wiki/Foreign_data_wrappers
- FDW for Oracle: https://github.com/laurenz/oracle_fdw
- FDW for MySQL/MariaDB: https://github.com/EnterpriseDB/mysql_fdw
- PostgreSQL Accessing MySQL as a Data Source Using mysql_fdw
  https://www.percona.com/blog/2018/08/24/postgresql-accessing-mysql-as-a-data-source-using-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

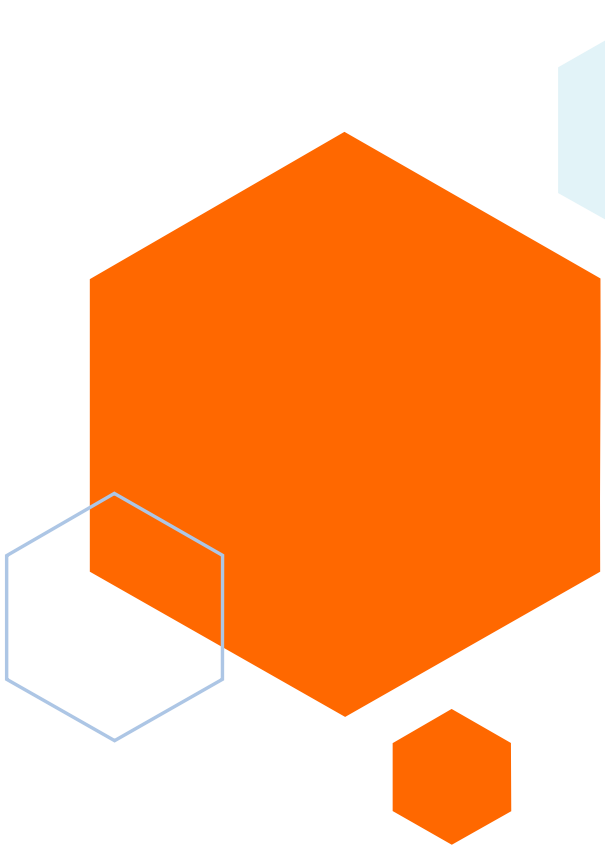


Weitere Informationen – MariaDB Connect Engine

ORDIX®

- Installing the CONNECT Storage Engine - MariaDB Knowledge Base
  https://mariadb.com/docs/server/server-usage/storage-engines/connect/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





Vielen Dank für

Ihre Aufmerksamkeit

Markus Flechtner

@markusdba.bsky.social

@markusdba@mastodon.social

markusdba.de|.net

mfl@ordix.de

ORDIX®

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/