Oracle nach PostgreSQL

Ein Reiseführer aus der Praxis

Roland Stirnimann

Roland Stirnimann – Trivadis AG

- 15 Jahre bei Trivadis
- Oracle HA, Migration
- DevOps
- PostgreSQL
- Datenplattformen

rstirnimann_ch

Agenda

trivadis

- Warm-Up - Ein paar Unterschiede zu Oracle…
- PostgreSQL versus EDB Postgres
- Projekterfahrungen
  - Oracle nach PostgreSQL
  - Oracle nach EDB Postgres Advanced Server
- Fazit

trivadis

Warm-Up - Ein paar Unterschiede zu Oracle...

CONSTRAINT Verhalten (Demo)

trivadis

- Oracle setzt Constraints pro Statement durch, PostgreSQL pro Zeile
- Constraint auf DEFERRABLE setzen in PostgreSQL für gleiches Verhalten wie in Oracle

```
INSERT INTO demo VALUES (1),(2);

UPDATE demo SET n=n+1;
ERROR:  duplicate key value violates unique constraint "demo_pk"
DETAIL:  Key (n)=(2) already exists.
```

Hinweis: INSERT von 2 Zeilen mit einem Statement geht in Oracle nicht.

```
VALUES (1),(2)
```

Quelle: Mathias Zarick, Trivadis Wien

FUNCTION in PL/SQL und PL/pgSQL (Demo)

trivadis

- Unterschiedliches Verhalten bei SELECT INTO
  - Oracle: Restriktiv – nur ein Wert erlaubt
  - PostgreSQL: Toleranter – gibt den ersten zurückgegebenen Wert aus dem SELECT zurück

Oracle

```
CREATE OR REPLACE FUNCTION
get_bal(acc_no IN NUMBER)
RETURN NUMBER
IS
  acc_bal NUMBER(11,2);
BEGIN
SELECT balance
INTO acc_bal
FROM accounts
WHERE account_id = acc_no;
RETURN acc_bal;
END;
/
```

PostgreSQL

```
CREATE OR REPLACE FUNCTION
get_bal(acc_no IN INTEGER)
RETURNS INTEGER
AS $$
DECLARE acc_bal INTEGER;
BEGIN
SELECT balance
INTO acc_bal
FROM accounts
WHERE account_id = acc_no;
RETURN acc_bal;
END;
$$ LANGUAGE plpgsql;
```

Quelle: Mathias Zarick, Trivadis Wien

Statement Error – ROLLBACK (Demo)

trivadis

- Statement Error in **PostgreSQL**
  - Verwendet standardmässig AUTOCOMMIT
  - Rollback zum Anfang der Transaktion oder zum letzten Save Point
  - Transactional DDL Support macht auch DDL Statements rückgängig (z.B. CREATE TABLE)
- Statement Error in **Oracle**
  - Fehlgeschlagenes Statement wird verworfen
  - Transaktion ist noch offen

Quelle: Mathias Zarick, Trivadis Wien

trivadis

Was ist wichtig bei der Migration?

Testen, Testen, Testen...

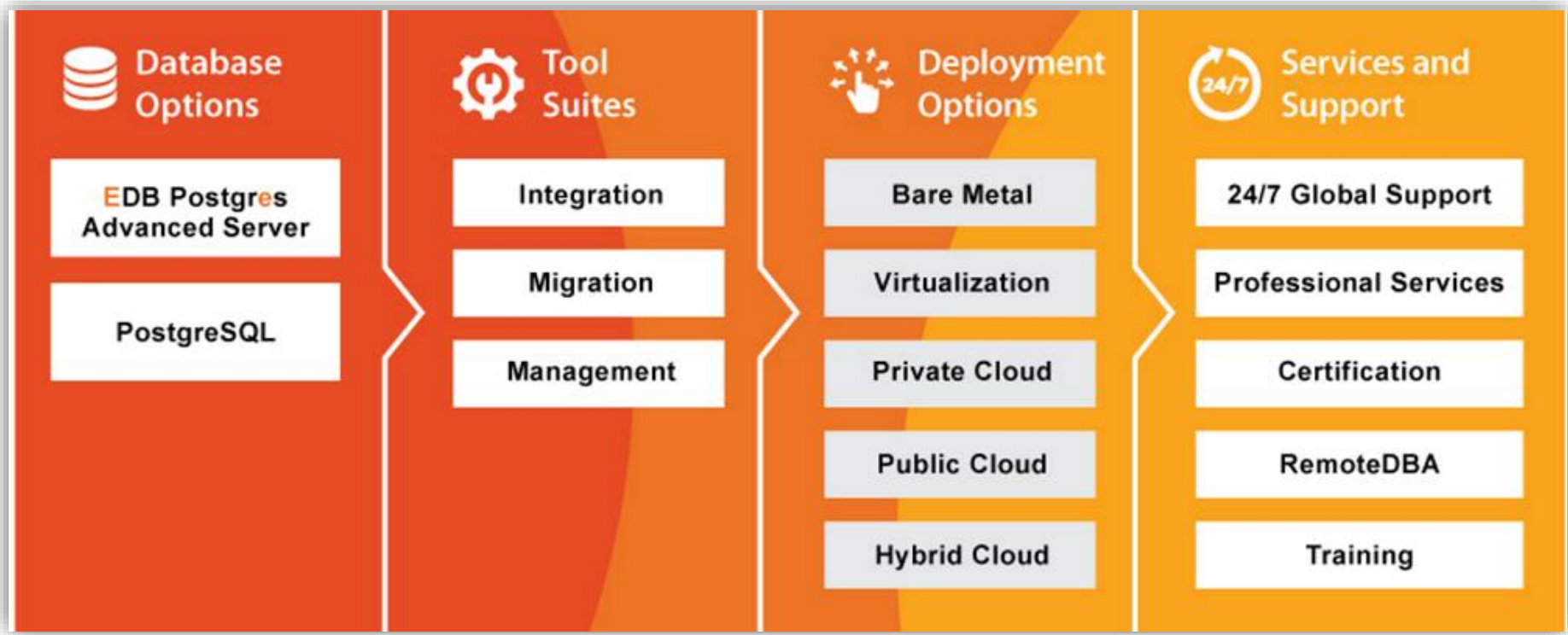
trivadis

PostgreSQL versus EDB Postgres

EDB Postgres Plattform

trivadis

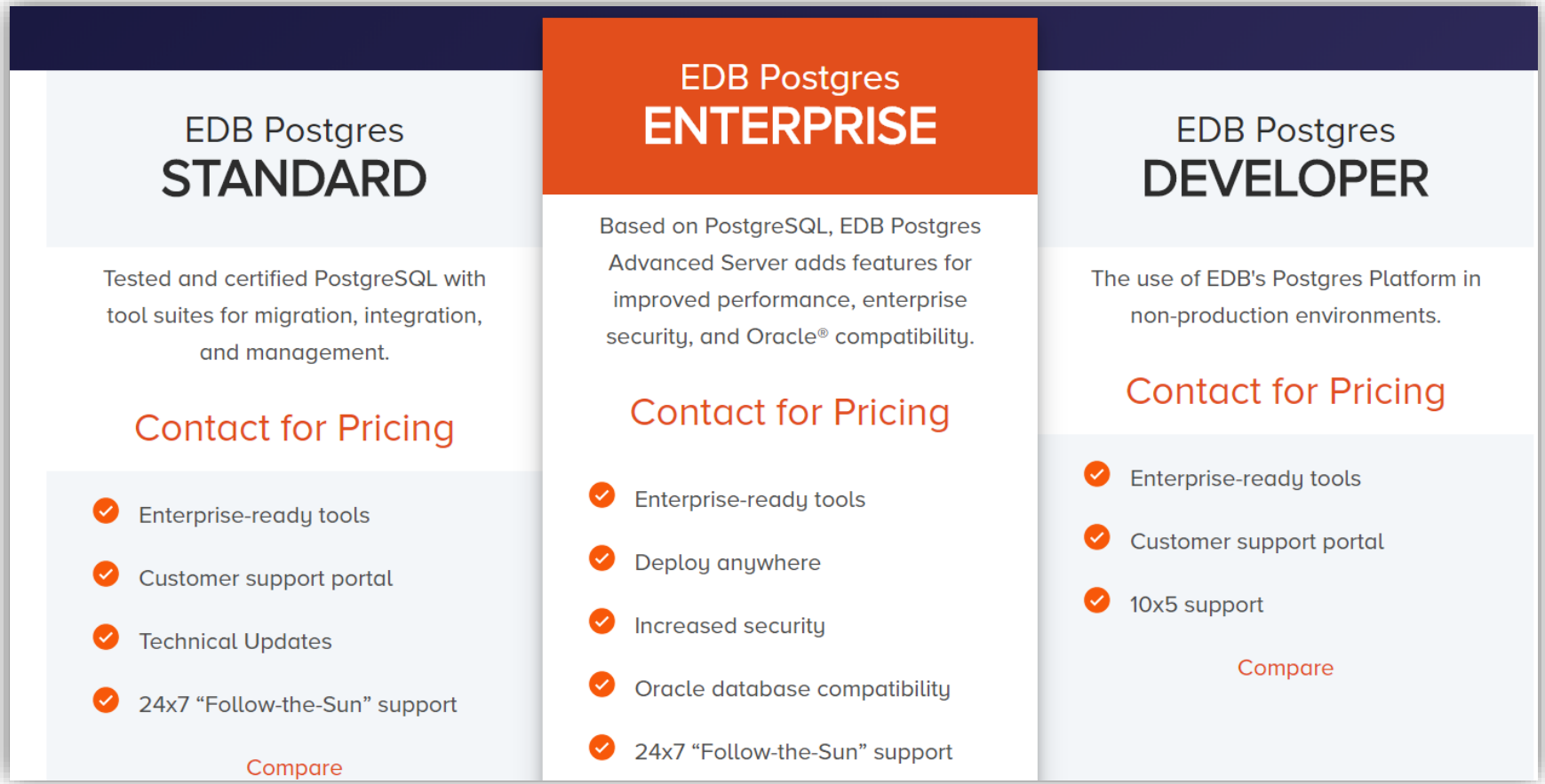
Tools und Support für den Enterprise Einsatz



Quelle: https://www.enterprisedb.com/

EDB Subskriptionen basierend auf Anzahl CPU

trivadis



Quelle: https://www.enterprisedb.com/

EDB Postgres Advanced Server – Im Kern gleich

trivadis

- EDB baut auf PostgreSQL auf (binär kompatibel)
- Zusätzliche Enterprise Features
  - Security: Password Profiles, EDB*Wrap, etc.
  - Performance Features: Hints, erweiterte Analyse, Ressource Manager, etc.
  - Developer: Oracle PL/SQL, hierarchische Query, Synonyms, Oracle DBMS_* Packages, etc.
  - Tools: BART, Failover Manager, PEM, Migration Toolkit, etc.
- Oracle Kompatibilität für «einfachere» Migration
- EDB Dokumentation ergänzt jene von PostgreSQL um die EDB Features
  - https://www.postgresql.org/docs/11/index.html
  - https://www.enterprisedb.com/docs/en/11.0/EPAS_Guide_v11/toc.html
- EDB hat mehrere namhafte PostgreSQL Entwickler in den eigenen Reihen

trivadis

Projekterfahrungen
Projekt 1

Projekt 1 - Energieversorgung
Einschätzung für Migration nach PostgreSQL/EDB

trivadis

- Strategie:
  - Bis 2020 sind 80% der Datenbanken in der AWS Cloud
  - Weg von kommerziellen RDBMS und kein Vendor Lock-in
- Beweggründe:
  - Hardware-Neuanschaffung für Oracle RAC nicht nötig
  - Bestehende Hardware reicht länger wenn Applikationen nach AWS (PostgreSQL) gehen
- Projektziel:
  - Analyse-Tool für Beurteilung der Durchführbarkeit einer Migration in die AWS Cloud der bestehenden 800 Oracle Schemata
  - Ziel-RDBMS ist entweder PostgreSQL (RDS, DBaaS) oder EDB Postgres AS (EC2, IaaS)

Projekt 1 - Anforderungen

trivadis

- Kein direkter Zugriff auf die Datenbanken, Sammeln und Auswerten passiert separat
  - Schritt 1: Daten erheben als CSV
  - Schritt 2: Daten auswerten und darstellen als HTML Report
- Keine Installation von Software, lediglich verteilen von Skripts
- Sammeln von drei verschiedenen :
  - *gather_db_details.sql*: Feature Verwendung, Objekttypen und Schema-Details
  - *gather_db_behavior.sql*: Applikationsverhalten, z.B. zugreifende Programme, Treiber
  - *gather_db_metrics.sql*: Metriken über die Ressourcenverwendung (CPU, I/O)
    - Oracle RAC Unterstützung
    - Essentiell für die Zuordnung zu AWS DB-Instanz-Klassen
      https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Concepts.DBInstanceClass.html

Projekt 1 - Informationen als CSV sammeln

trivadis

- Sammler-Skripts in Oracle ausführen als DBA User
- Option `-M` ist ab 12.2 verfügbar, ansonsten `set colsep` in SQL*Plus verwenden
- In SQL Developer das Resultat als delimited (;) Textdatei exportieren

```
export NLS_DATE_FORMAT='dd.mm.yyyy hh24:mi:ss'

sqlplus -S -M 'CSV on delimiter ;' system@REPO1
@../sql/gather_db_details.sql >gather_db_details.csv

sqlplus -S -M 'CSV on delimiter ;' system@REPO1 \
@../sql/gather_db_behavior.sql >gather_db_behavior.csv

sqlplus -S -M 'CSV on delimiter ;' system@REPO1 \
@../sql/gather_db_metrics.sql >gather_db_metrics.csv
```

Projekt 1 - Informationen auswerten

trivadis

- CSV Dateien dienen als Input für die Report-Erstellung

```
perl -I ../lib mactl.pl \
--input-files "REPO1;../csv/gather_db_details.csv" \
--behavior-input-files "REPO1;../csv/gather_db_behavior.csv" \
--metric-input-files "REPO1;../csv/gather_db_metrics.csv" \
--output-directory ../html --schemas-only
```

Summary

Database	Skipped Count	Bucket #1 Count	Bucket #2 Count	Bucket #3 Count	Showstopper Count	Total Count
REPO1	1	0	2	6	1	10
Total	1	0	2	6	1	10

The classification of each Database Schema is based on the following Bucket definitions:

	#1: Database Schema can be migrated without modifications
	#2: Database Schema can be migrated with moderate amount of modifications
	#3: Database Schema cannot be migrated or with a very high amount of modifications
	Showstopper: Database Schema cannot be migrated because of missing functionalities

Projekt 1 - Report Demo

trivadis

Schema Analysis Report

Name: TIBREPO1905
Authentication Type: PASSWORD
Account Status: OPEN
Segment Size: 1256.13 MB
Object Count: 257

Object Type	Count	PostgreSQL (RDS)			Enterprise DB (EC2)			Showstopper	Explanation
		Bucket #1	Bucket #2	Bucket #3	Bucket #1	Bucket #2	Bucket #3		
Auditing of DDL Statements	0								
Auditing of Objects	0								
Auditing of Privileges	0								
BFILE Columns	0								
BLOB Columns	1								The equaivalent of the BLOB Data Type in PostgreSQL / Enterprise DB is BYTEA, but it is limited to max 1 GB.
CLOB Columns	4								Text Data Types like CHAR, VARCHAR and TEXT support Strings of up to 1 GB in Size.

Projekt 1 - Nächste Schritte

trivadis

- Einfache portierbare Applikationen identifizieren und nach PostgreSQL oder EDB Postgres AS umstellen
- Nach ersten Erfahrungen sich an die härteren Brocken heran wagen
- Mögliche Migrationstools für die Schemata/Datenbank
  - PostgreSQL: ora2pg http://ora2pg.darold.net
  - EDB Postgres AS: Migration Toolkit https://www.enterprisedb.com/products/edb-postgres-platform/edb-migration-tool-kit
- Für Oracle Kompatibilität in PostgreSQL evtl. orafce ins Auge fassen https://github.com/orafce/orafce
- EDB Postgres AS hat Oracle Kompatibilität out-of-the-box

trivadis

Projekterfahrungen
Projekt 2

Projekt 2 - Luftfahrt
Migration nach EDB Postgres Advanced Server

trivadis

- Strategie:
  - Konsolidierung von Herstellern
  - Kosteneinsparungspotenziale analysieren
- Beweggründe:
  - Wartungs- und Supportkosten reduzieren
  - Rundum-Erneuerung der Applikation (Forms)
- Projektziel:
  - Machbarkeit einer Ablösung von Oracle durch EDB Postgres Advanced Server eruieren
  - Migrationsszenarien und Tools bekannt
  - Erforderliche strukturelle Anpassungen bekannt

Projekt 2 - Anforderungen

trivadis

- 2-teiliger Proof-of-Concept durchführen
  - EDB: Database Migration Assessment (DMA) und online EDB Postgres Migration Portal
  - Trivadis: Praktische Durchführung der Migration
- Zeitlich limitierte EDB Postgres Advanced Server Lizenz
- Durchführung mit komplizierter und kritischer Oracle Datenbank
- Server mit Betriebssystem (Centos 7) vom Kunden zur Verfügung gestellt
- EDB Postgres Advanced Server 11
- Zugriff auf Oracle Datenbank via SQLNet

Projekt 2 - EDB Database Migration Assessment

trivadis

- Input war ein Struktur-Export der Datenbank (Export-Skript von EDB)
- PDF Bericht von EDB Professional Services

	Object Counts			
Database/Schema	**Valid**	**Invalid**	**Incompatible (Included in Total)**	**Total**
[redacted]	1743	1486	20	15
Estimated Level of Effort				**15 days**

- Sieht schlimmer aus als es ist, wie wir später beim praktischen Migrationsteil sehen werden!

Projekt 2 - DMA Klassifikation

trivadis

- Klassifikation der Objekte wie Prozeduren, Funktionen, Tabellen, Indizes, etc.

Valid	The object definition can be migrated to EPAS.
Invalid	The object definition may have been valid in Oracle historically, but is currently invalid in Oracle.
Incompatible	The object definition requires some changes or customer workarounds before it can be migrated to EPAS.

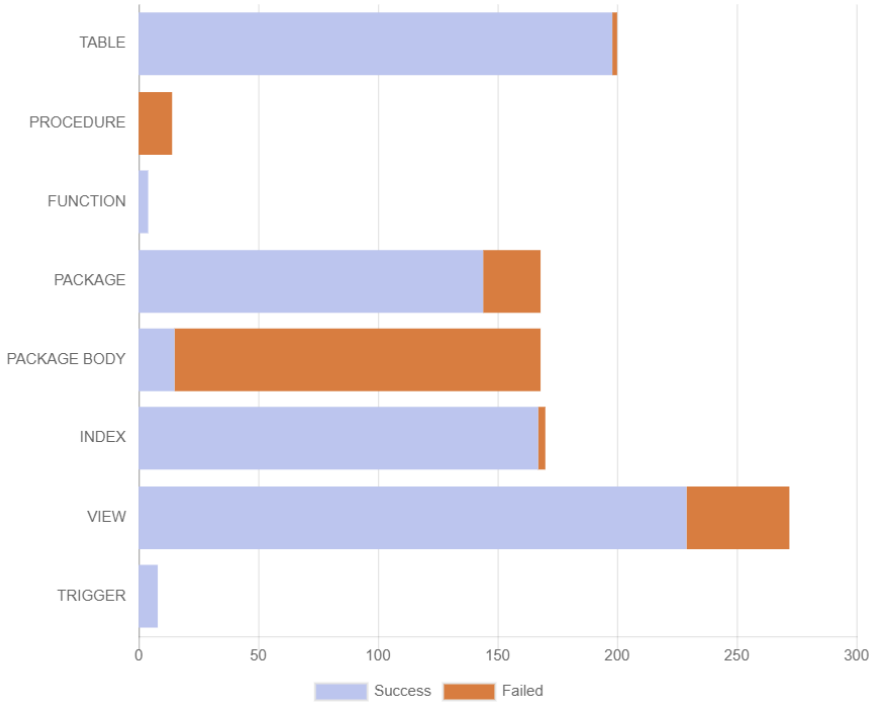
- Der Status **Invalid** tritt häufig auf wegen Abhängigkeiten zu Objekten in anderen Schemata
- **Incompatible** heisst nicht unmöglich
  - Erfordert einen Eingriff im Code
  - Beispiele: Index organized Table muss ersetzt werden oder reservierte Wörter müssen in Anführungszeichen stehen

Projekt 2 - EDB Postgres Migration Portal (1)

trivadis

- Upload des Struktur-Exports inkl. Auswertung (https://www.enterprisedb.com/edb-postgres-migration-portal)

Object Type	Total	✔	✖	🔧	Success Ratio
TABLE	200	198	2	194	99.00%
PROCEDURE	14	0	14	0	0.00%
FUNCTION	4	4	0	0	100.00%
PACKAGE	168	144	24	0	85.71%
PACKAGE BODY	168	15	153	0	8.93%
INDEX	170	167	3	0	98.24%
VIEW	272	229	43	272	84.19%
TRIGGER	8	8	0	0	100.00%
TOTAL	**1004**	**765**	**239**	**466**	**76.20%**



Projekt 2 - EDB Postgres Migration Portal (2)

trivadis

- EDB hat für viele Inkompatibilitäten fertige Lösungen für die Anpassung in Postgres

OBJECT: FUNCTION

Issue
`PIPELINED` functions

Resolution
In Oracle, `PIPELINED` function returns a collection of rows (either a nested table or a varray) that can be queried like a physical database table. In Advanced Server, `RETURN TYPE, RETURN TABLE, and RETURN SETOF RECORDS` functions are used, instead of `PIPELINED` function.

Projekt 2 - EDB Migration Toolkit (1)

trivadis

- EDB MTK ist ein CLI Tool
- Dokumentation: https://www.enterprisedb.com/docs/en/52.0/MTK_Guide_v52.0/toc.html
- Installation via EDB YUM Repository (`yum install edb-migrationtoolkit`)
- Konfigurationsdatei mit Verbindungsdetails zu Postgres und Oracle

```
vi /usr/edb/migrationtoolkit/etc/toolkit.properties

SRC_DB_URL=jdbc:oracle:thin:@192.168.38.186:1521:DB01
SRC_DB_USER=system
SRC_DB_PASSWORD=pw

TARGET_DB_URL=jdbc:edb://localhost:5444/edb
TARGET_DB_USER=enterprisedb
TARGET_DB_PASSWORD=pw

/usr/edb/migrationtoolkit/bin/runMTK.sh –help
```

Projekt 2 - MTK EDB Migration Toolkit (2)

trivadis

- Online oder offline Migration
  - Online exportiert von Oracle und importiert direkt nach EDB Postgres
  - Offline (`-offlineMigration`) exportiert SQL-Dateien für manuellen Import
- Nur Struktur und/oder Daten
  - `-schemaOnly` exportiert/importiert nur die Struktur (DDL)
  - `-dataOnly` exportiert/importiert nur die Daten
  - Ohne explizite Parameterangabe werden Struktur und Daten kopiert
- Verschiedene Optionen für Import von bestimmten Objekt-Typen: `-allTables`, `-allSequences`, `-skipFKConst`, etc.
- Oracle spezifische Optionen: `-allProfiles`, `-allDBLinks`, `-allSynonyms`, etc.
- Weitere Optionen für die Migration, um z.B. Datentypen zu wechseln

Projekt 2 - MTK Vorgehen

trivadis

- Offline Struktur-Export aus Oracle (`select any dictionary` Privileg)

```
runMTK.sh -offlineMigration ~/mig -schemaOnly -logDir ~/mig/logs \
-sourcedbtype oracle -targetSchema schema1 schema1
```

- Anpassen/Korrigieren der DDL Skripts im Export-Verzeichnis ~/mig
- Postgres-Datenbank anlegen mit User, Schema und Tablespace
- Laden der Struktur

```
create user admin password 'xxx';
create database mydb with owner admin;
\c mydb enterprisedb
create user schema1 with login identified by pw;
create tablespace ts_schema1 owner schema1 location '/opt/tbs/schema1';

edb-psql -f mtk_schema1_ddl.sql -o mtk_schema1_ddl.log mydb schema1
```

Projekt 2 - Erkenntnisse (1)

trivadis

- Viele initiale Fehler aufgrund von Abhängigkeiten
  - Weitere Schemata in Migration eingeschlossen
  - Datenbank Link zu anderer Oracle-DB erstellt (Instant Client erforderlich)

```
\c mydb schema1

CREATE DATABASE LINK oradb.world CONNECT TO user1 IDENTIFIED BY 'pw'
USING '//192.168.40.223:1521/ORADB';

select count(*) from table1@oradb.world;
```

- Reihenfolge beim Import ist entscheidend, Schemata und Skript Reihenfolge
- Berechtigungen (GRANT) sind ebenfalls gegenseitig zwischen Schemata zu erteilen

```
grant select on all tables in schema schema1 to schema2,schema3,schema4;
```

Projekt 2 - Erkenntnisse (2)

trivadis

- ROWID Funktionalität in EDB Postgres aktiviert
  Parameter `default_with_rowids=on` in $PGDATA/postgresql.conf
- Diverse Syntax Korrekturen, wo der EDB Parser scheinbar restriktiver ist als Oracle
- Wenige Schlüsselwörter als Spaltennamen mussten in Anführungszeichen gesetzt werden
- BLOB Datentyp einer Tabelle musste nach BYTEA angepasst werden
- Berechtigung auf SYS Package UTL_FILE erteilt und Directory Objekt erstellt
- search_path in DDL-Dateien mit «public» ergänzen für die Sichtbarkeit der Public Synonyms
- Tablespace-Namen setzen in DDL-Dateien damit Objekte korrekt angelegt werden

```
SET search_path=schema1,public;
SET default_tablespace = ts_schema1;
```

Projekt 2 - Empfehlungen

trivadis

- **Die richtigen Leute am Tisch (DBA und Entwickler)!**
  - Paralleles Arbeiten an applikations- und datenbankbezogenen Problemen
  - Agiles, iteratives Herantasten - Migrieren, Verifizieren, Korrigieren, Rollback und von vorne
- Nicht versuchen alle potenziellen Probleme beim ersten Mal lösen zu wollen
- Mit der Struktur beginnen aufgrund der kürzeren Laufzeit und grösserem Problem-Potenzial
- Offline Migration verwenden, um Skripts anpassen zu können vor dem Import
- Datenmigration gut überlegt angehen aufgrund der Laufzeit
  - Grosse Objekte identifizieren und ggf. separat migrieren ohne MTK (Delta-Migration)
  - Grundsätzlich sind bei den Daten eher weniger Migrationsprobleme zu erwarten
  - ABER: Die Laufzeit kann mit dem Standard JDBC Copy viel zu lang sein (Downtime)
  - Alternativen prüfen z.B. via Datenbank Link (Parallelisierung)

Projekt 2 - Nächste Schritte

trivadis

- Portierung der Struktur abschliessen mit allen Abhängigkeiten
- Durchführbarkeit der Datenmigration
  - Mögliche Downtime ergibt die maximale Laufzeit für Go-Live
  - Daran orientiert sich das Migrationskonzept der Daten (DB Link, etc.)
- Konnektivität der Applikation/Schnittstellen mit Postgres JDBC Treiber testen
- Postgres Betriebskonzept ausarbeiten

- Allein mit den 15 Tagen für die Portierung der Struktur ist es nicht getan!

trivadis

Fazit

Fazit

trivadis

- Technisch ist vieles möglich mit entsprechendem Aufwand
- EDB Postgres erleichtert die Migration für Oracle-Features deutlich (PL/SQL)
- Woran orientiert sich die Weiterentwicklung von portierten Applikationen, Oracle oder Postgres?
- PostgreSQL als Alternatives RDBMS im Unternehmen ist sehr sinnvoll
  - EDB bietet auch hier Support
  - Oracle Kompatibilität ist weniger wichtig für neue Projekte ohne Oracle Vergangenheit
- Wissensstand zu PostgreSQL in Unternehmen sehr unterschiedlich (Training)


Eine WELT ermöglichen,
in der intelligente IT
LEBEN und ARBEITEN
völlig selbstverständlich
erleichtert.