Infrastructure at your Service.

Breaking the deadlock, migrating from proprietary databases to PostgreSQL

dbi services

About me

Daniel Westermann

Senior Consultant

Open Infrastructure Technology Leader

+41 79 927 24 46

daniel.westermann@dbi-services.com

Who we are
dbi services

Experts At Your Service

- Over 45 specialists in IT infrastructure
- Certified, experienced, passionate

Based In Switzerland

- 100% self-financed Swiss company
- Over CHF6 mio. turnover

Leading In Infrastructure Services

- More than 120 customers in CH, D, & F
- Over 40 SLAs dbi FlexService contracted

Agenda

Agenda

Proprietary vs Open

Reasons for migrations

(some) Myths

Pitfalls

Tools which assist in migrations

Demo

Proprietary vs Open

Proprietary vs Open

What does proprietary mean?

When you search for "proprietary" on wikipedia the first sentence is:

> "Non-free" redirects here.

And then there is a huge list

> Proprietary church
> Proprietary community
> ...
> Proprietary hardware
> Proprietary software
> ...

Proprietary vs Open
Definitions – Proprietary software

Proprietary software is defined as

- "A software for which the publisher retains intellectual property rights, usually copyright of the source code"
  - sometime patents

Almost always customers do not get the source code

There is a chance that the software vendor listens to his customers and new releases contain features that are really required

Business is generated by licenses, support fees and consulting

Proprietary vs Open
Definitions – Proprietary software

The vendor of the software is suspected to invest money back into the development of the software and to increase quality of the product

Nowadays (at least in the database market) usually you pay once for the license and constantly for the support

Proprietary vs Open
Definitions – Open source software

Open Source Software is a software where you can get the source code free of charge

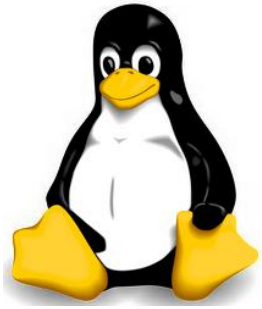
Business is usually generated by providing products or services around an Open Source Sofware

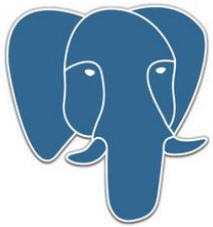
You can directly influence the direction of a Open Source Software by participating in the community

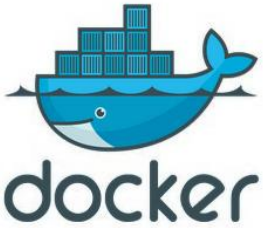
Usually you are allowed to fork Open Source Software and build your own products as long as you keep the Open Source license

Proprietary vs Open
Examples – Open source software

Some examples of successful Open Source products

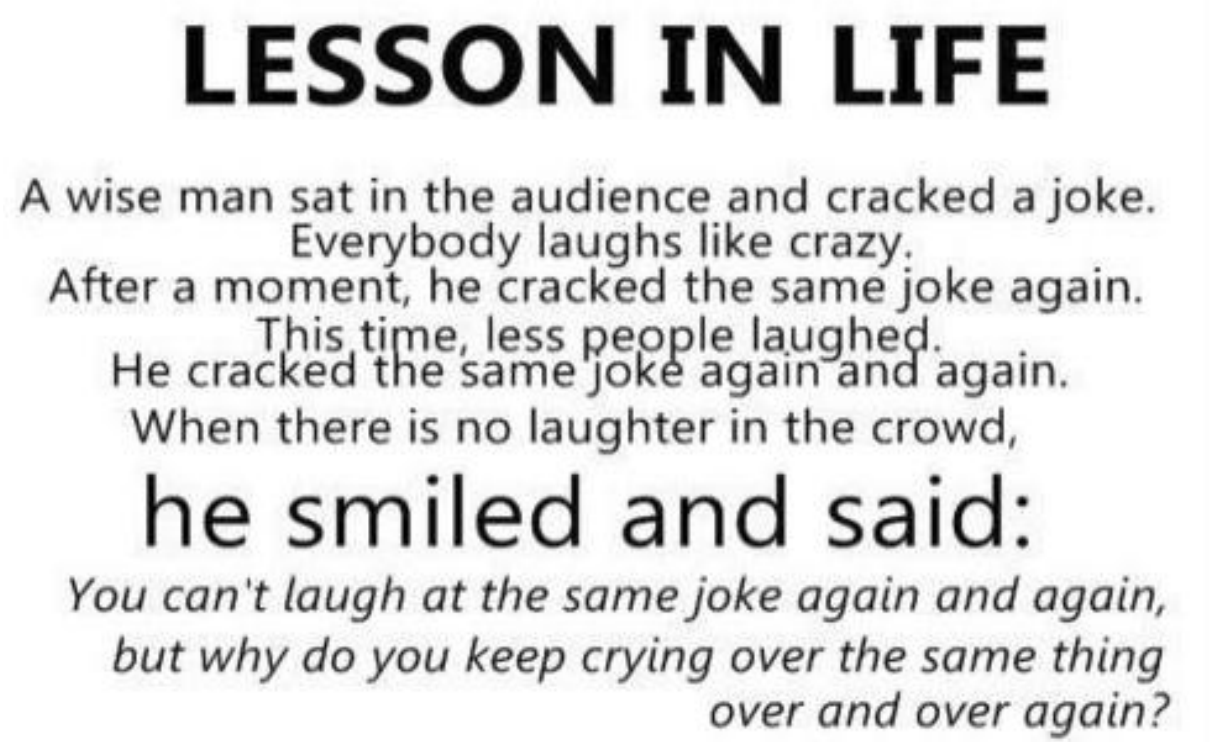






Reasons for migrations

Reasons for migrations
A lesson in life



Reasons for migrations
Some definitions …

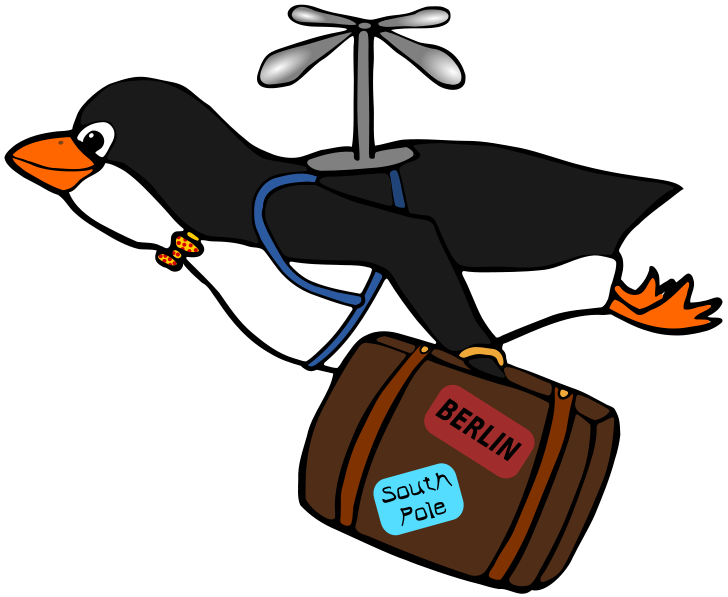
Human migration is the movement by people from one place to another with the intentions of settling temporarily or permanently in the new location

- Looking for a better life
- Making life more attractive
- Adventure
- War
- Mom

Reasons for migrations
Some definitions ...

Animal migration is the relatively long-distance movement of individuals, usually on a seasonal basis

- Escaping from the frozen season
- Food
- Weather



Reasons for migrations
Some definitions …

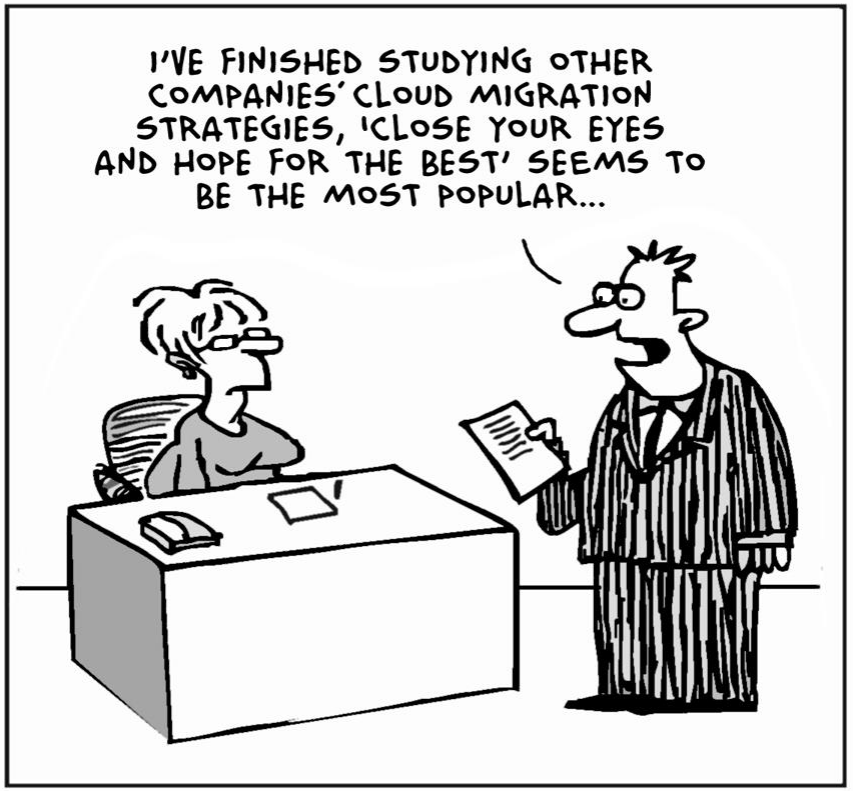
Data migration is the process of transferring data between storage types, formats or computer systems

- Legacy systems out of support
- New software/program versions
- Too much money?

Reasons for migrations
Some definitions …

Cloud migration is the process of transferring local issues into remote issues

- Because everybody does it
- It seems to be cool
- No internal knowhow
- Scalability



Reasons for migrations
Some definitions ...

What do all of these definitions have in common?

It is all about moving to a (hopefully) better place

Reasons for migrations
Some definitions ...

It is the same with migrations to PostgreSQL

It might be a nightmare

It might be hard work

But: if you succeed ...

... your life will be much more attractive

... no more moms

... no more licensing pains

... no more closed IT, you can be open

Reasons for migrations
Why migrating from one rdbms to another?

The most common reasons probably are

- Legacy systems out of support
- Financial pressure (Your boss told you to look for something cheaper)

But this is not the whole story ...

In case of PostgreSQL there is more

- PostgreSQL provides many features that other databases lack
- PostgreSQL easily adapts into your infrastructure landscape (FDWs)
- There is a huge ecosystem around PostgreSQL
- Open Source is the driver for innovation

(some) Myths

(some) Myths
De-mystifying

Myth 1: PostgreSQL does not provide high availability options

PostgreSQL does provide streaming replication by default

A hot standby database is very much the same as Oracle DataGuard

You can add as many standby databases as you want, even cascading

You can use pg-pool II to spread reads across all your hot standby databases

But Oracle provides RAC!

RAC is not HA and usually causes more issues than it resolves

(some) Myths
De-mystifying

Myth 2: PostgreSQL does not provide partitioning

PostgreSQL does provide partitioning

Partitioning is implemented by using table inheritance

What is missing is the SQL syntax to easily create paritions, but this is being worked on: https://commitfest.postgresql.org/10/611/

There are tools which help in partitioning, e.g.

- pg_partman: https://github.com/keithf4/pg_partman
- pg_pathman: https://github.com/postgrespro/pg_pathman

(some) Myths
De-mystifying

Myth 3: There is no integrated backup/restore functionality

PostgreSQL does provide backup/restore functionality

PostgreSQL does provide redo, it is called WAL (write ahead log)

Using pg_backup and WAL you can do PITR

There are tools which help in centralizing backups and provide a catalog as well as easy restore mechanisms

- barman: http://www.pgbarman.org/
- bart: http://www.enterprisedb.com/edb-backup-and-recovery-tool (non-free)
- pgbackrest: https://github.com/pgbackrest/pgbackrest

(some) Myths
De-mystifying

Myth 4: PostgreSQL will not be able to handle the load

(some) Myths
De-mystifying

Myth 5: There is no integrated language for implementing business logic inside the database

PostgreSQL does provide pl/pgsql by default

PostgreSQL does provide much more languages you can use (through extensions) than any of the commercial products

- PL/Java
- PL/Perl
- PL/Python
- PL/R
- PL/Tcl
- PL/Ruby
- PLv8 (Javascript)

(some) Myths
De-mystifying

Myth 6: There are no parallel SQLs in PostgreSQL

PostgreSQL will provide parallelism in the next version (9.6)

- Parallel sequential scans
- Parallel aggregates
- Parallel joins

(some) Myths
De-mystifying

Myth 7: There is no professional support for PostgreSQL

The support you get on the official mailing lists is much better than what you probably experience with the paid support of the commercial products

If you depend on a support contract there are many companies you can choose from

- https://www.postgresql.org/support/professional_support/

Pitfalls

Pitfalls

The general rule of thumb is: The less vendor specific features you use, the easier the migration probably will be (ok, no surprise)

There are some pitfalls you'll need to know before considering a migration to PostgreSQL

Some of the pitfalls may seem like a limitation but in reality it isn't. Do not expect the same implementation of something as you know it from other databases

Pitfalls
Implicit vs explicit conversions

In Oracle you can do this

```
SQL> select to_char(to_date('24.06.2016','dd.mm.yyyy')) mydate
     from dual;
```

What is the result?

```
   MYDATE
---------
24-JUN-16
```

Can I do the same thing in PostgreSQL?

```
postgres=# select to_char(to_date('24.06.2016','dd.mm.yyyy'));
```

What is the result?

```
ERROR:  function to_char(date) does not exist
HINT:  No function matches the given name and argument types. You
might need to add explicit type casts
```

Pitfalls
Implicit vs explicit conversions

The reason?

Oracle does an implicit conversion in the background

PostgreSQL will not do that for you, you have to know what you want to do

```
postgres=# select cast(to_date('24.06.2016','dd.mm.yyyy') as varchar);
  to_date
------------
 2016-06-24
(1 row)
```

Pitfalls
Implicit vs explicit conversions

Another example: In Oracle you can do this

```
SQL> create table t ( a number );

Table created.

SQL> insert into t values ( 20160624 );

1 row created.

SQL> select to_date(a,'yyyymmdd') mydate from t;

MYDATE
---------
24-JUN-16
```

Pitfalls
Implicit vs explicit conversions

The same will not work in PostgreSQL for the same reason

```
postgres=# create table t ( a int );
CREATE TABLE
postgres=# insert into t values ( 20160624 );
INSERT 0 1
postgres=# select to_date(a,'yyyymmdd') from t;


ERROR:  function to_date(integer, unknown) does not exist at character
8
HINT:  No function matches the given name and argument types. You
might need to add explicit type casts.
STATEMENT:  select to_date(a,'yyyymmdd') from t;
ERROR:  function to_date(integer, unknown) does not exist
LINE 1: select to_date(a,'yyyymmdd') from t;
```

Pitfalls
Packages

PostgreSQL does not know the concept of a package

If you want to simulate packages in PostgreSQL you have to use schemas

```
postgres=# create schema my_package_1;
CREATE SCHEMA
postgres=# create function my_package_1.increment(i integer)
postgres=# return integer AS $$
postgres$#          BEGIN
postgres$#                  RETURN i + 1;
postgres$#          END;
postgres$# $$ LANGUAGE plpgsql;
CREATE FUNCTION
```

Pitfalls
Data types and performance

Oracle knows exactly one data type for storing numeric values, which is NUMBER

PostgreSQL knows many more

8.1. Numeric Types

Numeric types consist of two-, four-, and eight-byte integers, four- and eight-byte floating-point numbers, and selectable-precision decimals. Table 8-2 lists the available types.

Table 8-2. Numeric Types

Name	Storage Size	Description	Range
`smallint`	2 bytes	small-range integer	-32768 to +32767
`integer`	4 bytes	typical choice for integer	-2147483648 to +2147483647
`bigint`	8 bytes	large-range integer	-9223372036854775808 to +9223372036854775807
`decimal`	variable	user-specified precision, exact	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
`numeric`	variable	user-specified precision, exact	up to 131072 digits before the decimal point; up to 16383 digits after the decimal point
`real`	4 bytes	variable-precision, inexact	6 decimal digits precision
`double precision`	8 bytes	variable-precision, inexact	15 decimal digits precision
`smallserial`	2 bytes	small autoincrementing integer	1 to 32767
`serial`	4 bytes	autoincrementing integer	1 to 2147483647
`bigserial`	8 bytes	large autoincrementing integer	1 to 9223372036854775807

Pitfalls
Data types and performance

Not every data type is perfect for what you want to do

```
postgres=# create table t1 ( a smallint, b numeric );
CREATE TABLE
postgres=# insert into t1 values ( generate_series ( 1,1000000),
generate_series ( 1,1000000) );
INSERT 0 1000000
postgres=# select sum(a) from t1;
     sum
--------------
 500000500000
Time: 70.650 ms
postgres=# select sum(b) from t1;
     sum
--------------
 500000500000
Time: 142.335 ms
```

Pitfalls
Data types and performance

You need to know what you want to store

Do not use the biggest data type only because it is most convenient

There are differences in performance

INTEGER is not the same as NUMERIC and INTEGER is not the same as BIGINT

Pitfalls
What is NULL and when is it NULL?

In Oracle you can concatenate NULL to a string and get a string

```
SQL> select 'AAA' || null NNNN from dual;

NNN
---
AAA
```

In PostgreSQL you get NULL

```
postgres=# select 'AAA' || null NNNN;
 nnnn
------

(1 row)
```

Pitfalls
What is NULL and when is it NULL?

Is an empty string NULL in Oracle?

```
SET SERVEROUT ON
DECLARE
  lv varchar2(10) := '';
BEGIN
  IF lv IS NULL
  THEN
    dbms_output.put_line('YEAH!!');
  ELSE
    dbms_output.put_line('YIPPEE!!');
  END IF;
END;
/
```

What is the result? YEAH!!

Pitfalls
What is NULL and when is it NULL?

Is an empty string NULL in PostgreSQL

```
SET client_min_messages='NOTICE';
DO $$DECLARE
  lv varchar(10) := '';
BEGIN
  IF lv IS NULL
  THEN
    RAISE NOTICE 'YEAH!!';
  ELSE
    RAISE NOTICE 'YIPPEE!!';
  END IF;
END$$;
```

What is the result? YIPPEE!!

Pitfalls
What is NULL and when is it NULL?

You'll need to check all your code for how it handles NULL

NULL is not handled the same in Oracle and PostgreSQL

In PostgreSQL it is closer to "undefined" and "undefined" is what it really means

Pitfalls
Business logic in the database

Many applications implement business logic in the database

Oracle: PL/SQL, sometimes Java using the integrated JVM

MS SQL Server: T-SQL

DB2: DB2 SQL

Pitfalls
Business logic in the database

If you want to migrate to community PostgreSQL you'll have to

- either re-implement this with a language PostgreSQL supports
- or re-implement in the application itself without using stored functions/procedures/packages
- or do a combination of both

The non-free EDB Postgres Advanced Server provides Oracle compatibility which can lower the costs of migrations

https://www.enterprisedb.com/docs/en/9.5/oracompat/toc.html

Pitfalls
Business logic in the database

In EDB Postgres Advanced Server you can for example do things like these

```
edb=# select * from dbms_utility.get_cpu_time();
 get_cpu_time
--------------
       1.9632
(1 row)

edb=# select count(*) from dba_tables;
 count
-------
   152
(1 row)
```

Pitfalls
Materialized views

There are materialized views in PostgreSQL

Currently there is no "refresh on demand"

Data is not not always current, but sometimes this is not required (historical data)

Materialized views can be used over foreign tables

Tools which assist in migrations

Tools which assist in migrations
ora2pg

http://ora2pg.darold.net/

Tools which assist in migrations
ora2pg

ora2pg moves Oracle and MySQL databases to PostgreSQL

ora2pg is implemented in Perl and requires Perl >= 5.10

ora2pg depends on additional Perl modules

- DBI > 1.614
- DBD:Oracle for migrations from Oracle
- DBD:mysql for migrations from MySQL
- On some distributions Time::HiRes needs to be installed

Tools which assist in migrations
ora2pg

ora2pg can migrate

- Tables
- Sequences
- PL/SQL Packages
- Partitions
- Procedures
- Functions
- Spatial

Tools which assist in migrations
ora2pg - installation

Check that your Perl version matches the requirement

```
[pg@pg ~]$ perl -version | grep version
This is perl 5, version 16, subversion 3 (v5.16.3) built for x86_64-
linux-thread-multi
```

Configure CPAN if not already configured

```
[pg@pg ~]$ cpan
Would you like to configure as much as possible automatically? [yes]
What approach do you want?  (Choose 'local::lib', 'sudo' or 'manual')
 [local::lib] sudo
...
Would you like me to automatically choose some CPAN mirror
sites for you? (This means connecting to the Internet) [yes]
...
Autoconfiguration complete
```

Tools which assist in migrations
ora2pg

Install the Oracle Instant Client for Oracle migrations

(http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html)

```
[pg@pg ~]$ sudo yum localinstall oracle-instantclient12.1-basic-12.1.0.2.0-1.x86_64.rpm
[pg@pg ~]$ sudo yum localinstall oracle-instantclient12.1-sqlplus-12.1.0.2.0-1.x86_64.rpm
[pg@pg ~]$ sudo yum localinstall oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
[pg@pg ~]$ ls /usr/lib/oracle/12.1/client64/
bin  lib
```

Set the environment for installing the DBD::Oracle module

```
[pg@pg ~]$ export ORACLE_HOME=/usr/lib/oracle/12.1/client64/
[pg@pg ~]$ export PATH=$ORACLE_HOME/bin:$PATH
[pg@pg ~]$ export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

Tools which assist in migrations
ora2pg

Install the required modules

```
[pg@pg ~]$ cpan
cpan[1]> get DBD::Oracle
cpan[2]> exit
[pg@pg ~]$ cd .cpan/build/DBD-Oracle*
[pg@pg ~]$ perl Makefile.PL -l
[pg@pg ~]$ make && make test
[pg@pg ~]$ sudo make install
```

Do not try to install DBD::Oracle directly with CPAN, in my case that always fails

Tools which assist in migrations
ora2pg

Install ora2pg

```
[pg@pg ~]$ wget oracle-instantclient12.1-devel-12.1.0.2.0-1.x86_64.rpm
[pg@pg ~]$ tar -axf v17.4.tar.gz
[pg@pg ~]$ cd ora2pg-17.4/
[pg@pg ~]$ perl Makefile.PL
[pg@pg ~]$ make
[pg@pg ~]$ sudo make install
[pg@pg ~]$ ora2pg –version
Ora2Pg v17.4
```

Initialize an ora2pg project

```
[pg@pg ~]$ ora2pg --init_project ora2pg_demo
```

Tools which assist in migrations
ora2pg

The project directory

```
[pg@pg ~]$ cd ora2pg_demo/
[pg@pg ~]$ ls
config  data  export_schema.sh  import_all.sh  reports  schema
sources
```

Define your properties (at least these)

```
[pg@pg ~]$ vi config/ora2pg.conf
ORACLE_HOME     /usr/lib/oracle/12.1/client64/
ORACLE_DSN      dbi:Oracle:host=192.168.22.242;sid=PROD
ORACLE_USER     system
ORACLE_PWD      manager
SCHEMA          SH
```

Tools which assist in migrations
ora2pg

Export the Oracle schema

```
[pg@pg ~]$ pwd
/home/postgres/ora2pg_demo
[pg@pg ~]$ ./export_schema.sh
```

When all is fine this is the output

```
[=========================>] 12/12 tables (100.0%) end of scanning.
[=========================>] 10/10 objects types (100.0%) end of
objects auditing.
Running: ora2pg -p -t TABLE -o table.sql -b ./schema/tables -c
./config/ora2pg.conf
[=========================>] 12/12 tables (100.0%) end of scanning.
[=========================>] 12/12 tables (100.0%) end of table export.
...
```

Tools which assist in migrations
EDB mtk

http://www.enterprisedb.com/products-services-training/products-overview/postgres-plus-solution-pack/migration-toolkit

Tools which assist in migrations
EDB mtk - overview

EDB mtk (migration toolkit) is one of the tools you get from EnterpriseDB when you have at least one subscription

To be honest: It only makes sense if you have a subscription for the professional edition because it provides the Oracle compatibility mode

- Implementation of various Oracle dbms_* packages
- Implementation of PL/SQL
- Implementation for SQLPLUS (if you really want to work with it)
- Implementation of Oracle SQL syntax
- Implementation of Oracle build-in functions (NVL, NVL2, ...)

Tools which assist in migrations
EDB mtk - overview

EDB mtk migrates

Object	Oracle	Sybase	SQL Server	MySQL
Schemas	X	X	X	X
Tables	X	X	X	X
List-Partitioned Tables	X			
Range-Partitioned Table	X			
Constraints	X	X	X	X
Indexes	X	X	X	X
Triggers	X			
Table Data	X	X	X	X
Views	X		X	
Materialized Views	X			
Packages	X			
Procedures	X			
Functions	X			
Sequences	X			
Users/Roles	X			
Profiles	X			
Object Types	X			
Object Type Methods	X			
Database Links	X			

Tools which assist in migrations
EDB mtk - overview

Requirements

- An installation of EDB Postgres Plus Advanced Server
- Java (openjdk)
- Third party JDBC drivers for connecting to either Oracle, MSSQL/Sybase, MySQL or PostgreSQL

Tools which assist in migrations
EDB mtk - setup

Make sure Java is available on your system

```
[pg@pg ~]$ java -version
openjdk version "1.8.0_91"
OpenJDK Runtime Environment (build 1.8.0_91-b14)
OpenJDK 64-Bit Server VM (build 25.91-b14, mixed mode)
```

Download the JDBC driver for Oracle

http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html

Tools which assist in migrations
EDB mtk - setup

Make the JDBC driver available to the JAVA installation

```
[pg@pg ~]$ ls -l /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.91-
0.b14.el7_2.x86_64/jre/lib/ext/
-rw-r--r--. 1 root root 4003338 Apr 20 16:06 cldrdata.jar
-rw-r--r--. 1 root root    9444 Apr 20 16:06 dnsns.jar
-rw-r--r--. 1 root root   48732 Apr 20 16:06 jaccess.jar
-rw-r--r--. 1 root root 1204420 Apr 20 16:06 localedata.jar
-rw-r--r--. 1 root root     617 Apr 20 16:06 meta-index
-rw-r--r--. 1 root root 2037734 Apr 20 16:06 nashorn.jar
-rw-r--r--. 1 root root 3699265 Jun 17 09:27 ojdbc7.jar
-rw-r--r--. 1 root root   30444 Apr 20 16:06 sunec.jar
-rw-r--r--. 1 root root  294210 Apr 20 16:06 sunjce_provider.jar
-rw-r--r--. 1 root root  266627 Apr 20 16:06 sunpkcs11.jar
-rw-r--r--. 1 root root   77886 Apr 20 16:06 zipfs.jar
```

Tools which assist in migrations
EDB mtk - setup

Prepare the toolkit.properties file

```
[pg@pg ~]$ pwd
[PPAS_HOME]/edbmtk/etc
[pg@pg ~]$ cat toolkit.properties
SRC_DB_URL=jdbc:oracle:thin:@192.168.22.242:1521:PROD
SRC_DB_USER=sh
SRC_DB_PASSWORD=sh
TARGET_DB_URL=jdbc:edb://localhost:5433/edb
TARGET_DB_USER=postgres
TARGET_DB_PASSWORD=postgres
[pg@pg ~]$ chmod 600 etc/toolkit.properties
```

If you do not change the permission you'll get

```
MTK-11015: The connection credentials …/etc/toolkit.properties is not
secure and accessible to group/others users. This file contains
plain passwords and should be restricted to Migration Toolkit owner
user only.
```

Tools which assist in migrations
EDB mtk - setup

Do a basic check to verify mtk is running

```
[pg@pg ~]$ pwd
[PPAS_HOME]/edbmtk
[pg@pg ~]$ bin/runMTK.sh -version
Running EnterpriseDB Migration Toolkit (Build 49.0.1) ...
EnterpriseDB Migration Toolkit (Build 49.0.1)
```

Do the migration

```
[pg@pg ~]$ bin/runMTK.sh -dropSchema true SH
Running EnterpriseDB Migration Toolkit (Build 49.0.1) ...
EnterpriseDB Migration Toolkit (Build 49.0.1)
```

Tools which assist in migrations
EDB mtk - setup

This is the result

```
******************** Migration Summary ********************
Tables: 12 out of 12
Constraints: 14 out of 14
Indexes: 13 out of 13
Views: 1 out of 3
Procedures: 1 out of 1
Profiles: 0 out of 1

Total objects: 44
Successful count: 41
Failed count: 3
Invalid count: 0
...
```

Tools which assist in migrations
EDB mtk - setup

This is the result(2)

```
...
List of failed objects
======================
Views
--------------------
1. SH.FWEEK_PSCAT_SALES_MV
2. SH.CAL_MONTH_SALES_MV

Profiles
--------------------
1. ORA_STIG_PROFILE
```

Tools which assist in migrations
EDB mtk - setup

All the logs are generated in the user's home directory by default

```
[pg@pg ~]$ ls .enterprisedb/migration-toolkit/logs/
ls
mtk_SH_20160617095432.log  mtk_SH_20160617095627.log
mtk_SH_20160617100312.log  mtk_-version_20160617095311.log
[pg@pg ~]$ head mtk_SH_20160617100312.log
Running EnterpriseDB Migration Toolkit (Build 49.0.1) ...
Source database connectivity info...
```

Tools which assist in migrations
EDB mtk - setup

What do we have available now?

```
edb=# \dn
  List of schemas
  Name   |  Owner
--------+----------
 public | postgres
 sh     | postgres
(2 rows)
SET
edb=# \d
                  List of relations
 Schema |            Name             | Type  |  Owner
--------+-----------------------------+-------+----------
 sh     | channels                    | table | postgres
 sh     | costs                       | table | postgres
 sh     | costs_costs_1995            | table | postgres
...
```

Tools which assist in migrations
EDB mtk - setup

Many of the Oracle DBA_* view are available

```
edb=# select text from dba_source where name = 'FEEDFAKE';
                    text
-----------------------------------------
 CREATE OR REPLACE PROCEDURE sh.feedfake()
  AUTHID DEFINER IS
 BEGIN
   null;
 END
```

Tools which assist in migrations
EDB mtk - setup

Partitions are migrated automatically

```
edb=# select owner,schema_name,table_name,partitioning_type from
dba_part_tables;
  owner   | schema_name | table_name | partitioning_type
----------+-------------+------------+-------------------
 POSTGRES | SH          | COSTS      | RANGE
 POSTGRES | SH          | SALES      | RANGE

edb=# select table_name,partition_name from dba_tab_partitions;
 table_name | partition_name
------------+----------------
 COSTS      | COSTS_1995
 COSTS      | COSTS_1996
 COSTS      | COSTS_H1_1997
 COSTS      | COSTS_H2_1997
 COSTS      | COSTS_Q1_1998
...
```

Tools which assist in migrations
EDB mtk - setup

What did not work?

```
Getting Profile Resource Definitions...
MTK-13021:Password Profile verify function
ORA12C_STRONG_VERIFY_FUNCTION must be explicitly migrated
Creating Profile: ORA_STIG_PROFILE
MTK-15027: Error creating Profile ORA_STIG_PROFILE
DB-42704: com.edb.u
```

Profiles are supported but you need to create the verify function before

Tools which assist in migrations
EDB mtk - setup

What did not work?

```
MTK-15009: Error Creating Materialized View: CAL_MONTH_SALES_MV
DB-42601: ERROR: syntax error at or near "PREBUILT" at position 51
-- Line 1: CREATE MATERIALIZED VIEW CAL_MONTH_SALES_MV BUILD PREBUILT
--                                                          ^
```

PREBUILT is not supported

https://www.enterprisedb.com/docs/en/9.5/oracompat/Database_Compatibility_for_Oracle_Developers_Guide.1.069.html#pID0E0ERU0HA

Tools which assist in migrations
Database Workbench

http://www.upscene.com/database_workbench/

Tools which assist in migrations
Database Workbench

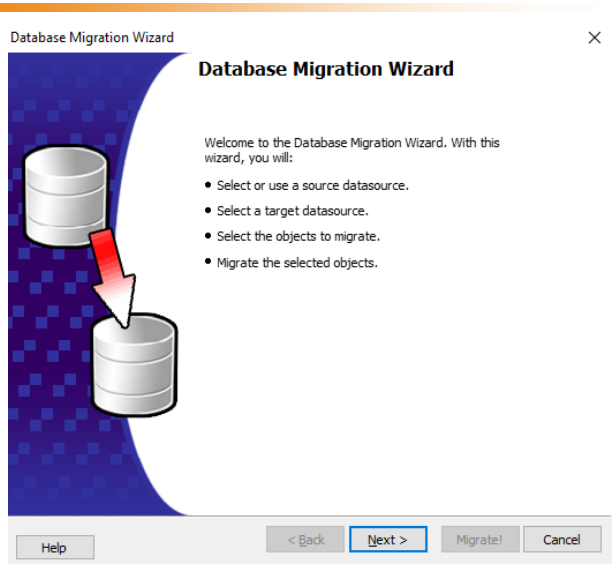
I did not test it but it provides dialogs for migrating from one database to another

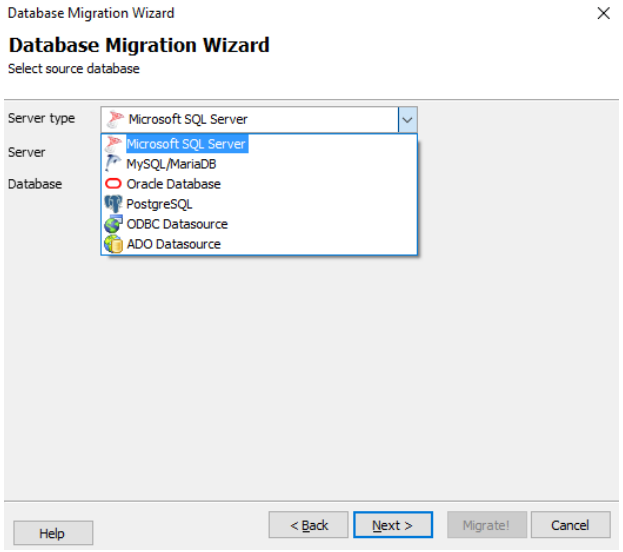
It is a Windows only software (Delphi based) but runs under Wine

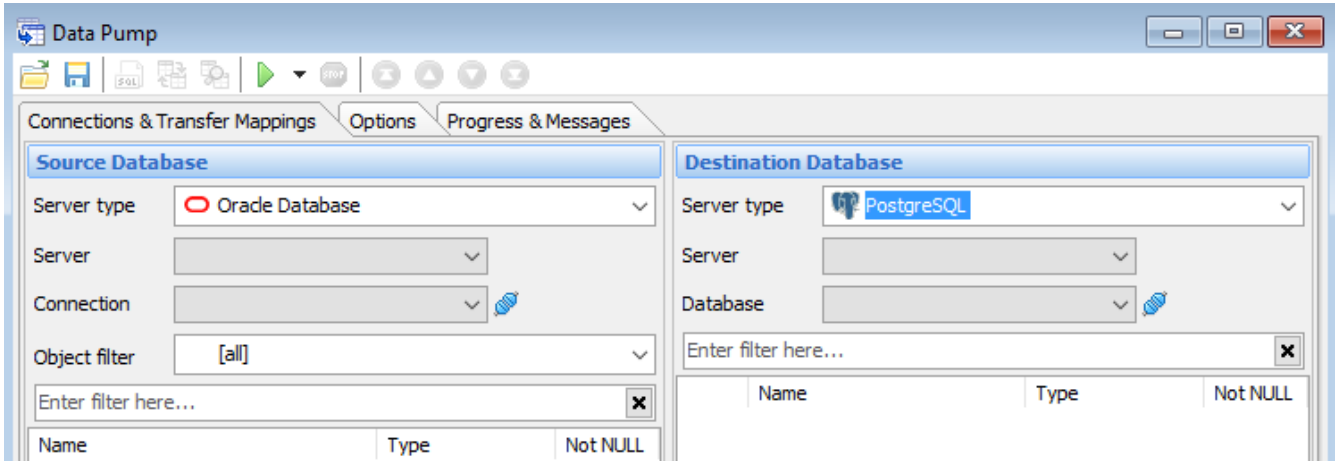
Database workbench is able to connect to

- MySQL
- Oracle
- PostgreSQL (in beta currently)
- Firebird
- MS SQL Server / Sybase / InterBase
- NexusDB

Tools which assist in migrations
Database Workbench







Demo

Demo

Core messages

Core messages

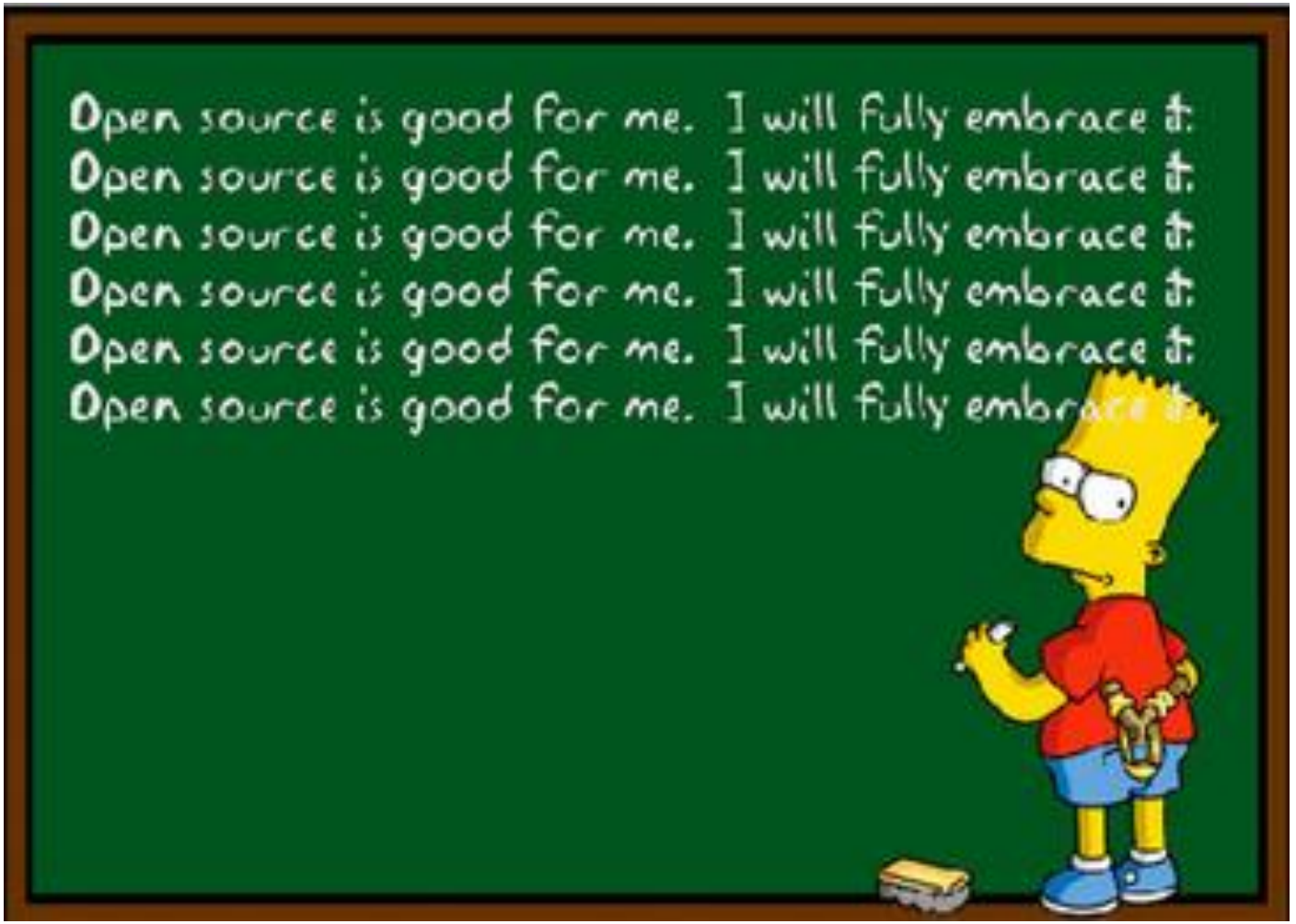
It is not only costs that make migrations to PostgreSQL attractive

The current trend in IT clearly goes into the direction of using Open Source software as the basis for building business

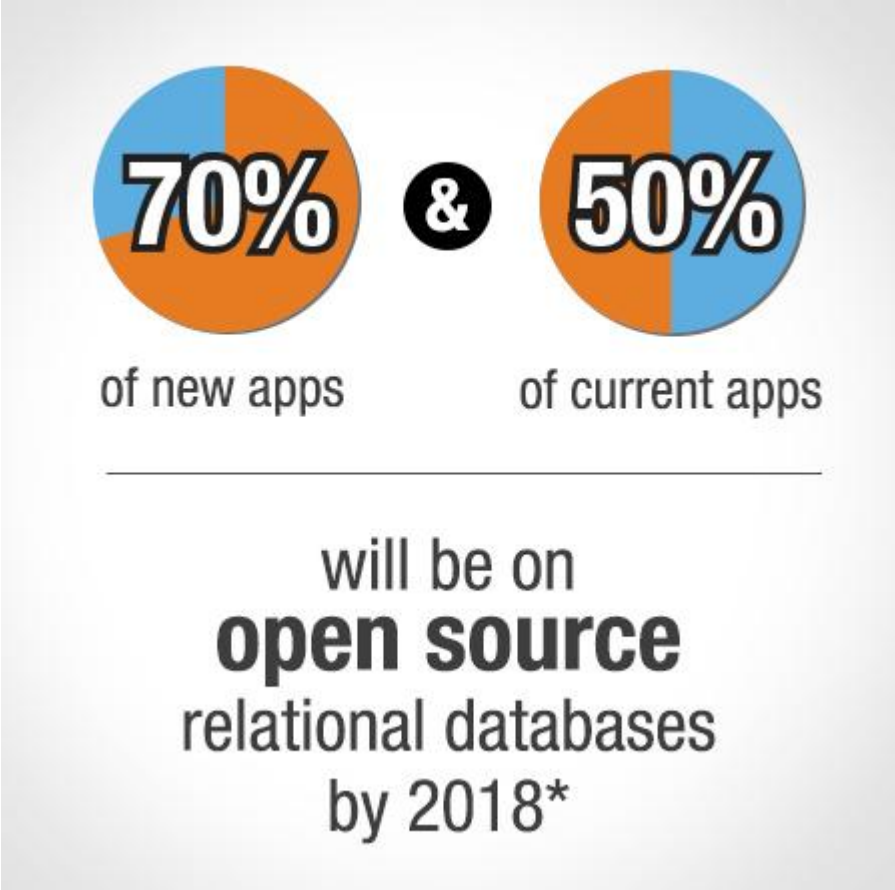
With PostgreSQL there is a more than 20 year old, robust and reliable database system that already supports many big companies

All features you need for enterprise operations are there

Core messages



Core messages



*Gartner, State of Open Source RDBMS, 2015, Donald Feinberg and Merv Adrian, April 21, 2015.

Any questions? Please do ask

Daniel Westermann
Senior Consultant
Open Infrastructure Technology Leader

+41 79 927 24 46
daniel.westermann@dbi-services.com

We look forward to working with you!