

Your Data Deserves the Best

Migration to PostgreSQL

Josef Machytka <josef.machytka@credativ.de>

2025-05-08/09 PostgreSQL Conference Germany 2025

- Founded 1999 in Jülich, Germany
- Close ties to Open-Source Community
- More than 40 Open-Source experts
- Consulting, development, training, support (3rd-level / 24x7)
- Open-Source infrastructure with Linux, Kubernetes and Proxmox
- Open-Source databases with PostgreSQL
- DevSecOps with Ansible, Puppet, Terraform and others
- Since 2025 independent owner-managed company again



- Professional Service Consultant - PostgreSQL specialist at credativ GmbH
 - 30+ years of experience with different databases
 - PostgreSQL (12y), BigQuery (7y), Oracle (15y), MySQL (12y), Elasticsearch (5y), MS SQL (5y)
 - 10+ years of experience with Data Ingestion pipelines, Data Analysis, Data Lake and Data Warehouse
 - 2 years of practical experience with different LLMs / AI including their architecture and principles
 - From Czechia, living now 11 years in Berlin
-
- **LinkedIn:** linkedin.com/in/josef-machytka
 - **ResearchGate:** researchgate.net/profile/Josef-Machytka
 - **Academia.edu:** academia.edu/JosefMachytka
 - **Medium:** medium.com/@josef.machytka
 - **Sessionize:** sessionize.com/josefmachytka

Table of contents

- Boldly Migrate to PostgreSQL
- Meet credativ-pg-migrator
- Informix
- IBM Db2
- Sybase ASE
- MS SQL Server
- MySQL / MariaDB
- SQL Anywhere
- PostgreSQL to PostgreSQL
- Summary



AI images without credits
created by the author
using DeepDreamGenerator

Boldly Migrate to PostgreSQL

Main Reasons for Migration

- Growing licensing payments for proprietary databases
- Lack of support and new features for legacy DBs
- No plans for future development of some legacy DBs
- In some cases lack of administrators, shrinking community
- Legacy DBs often run on old hardware / outdated OS



Typical Migration Issues

- Knowledge is often lost
- “History became legend. Legend became myth.”
- Apps might depend on specific features - ROWIDs
- Constructs specific to older DBs using UNDO logs
- CURSOR WITH HOLD in stored procs, commit in each loop
 - Which is like big transaction with subtransactions
- Differences in SQL syntax (JOINs without ON)
- Some DBs allow variety of hints in SQL statements
- Rewrites in statements / stored procedures necessary



Typical Migration Issues

- No INFORMATION_SCHEMA, only system tables in some DBs
- Over time DBs allowed different character sets
- Some DBs allow easy cross-database calls (tables, funcs)
- Frequent usage of synonyms (for deployment, ETL tasks)
- More freedom with data types (numeric PK, FK for diff types)
- Big differences in partitioning
- Higher max sizes of LOBs in some DBs - 2GB, 4GB, 4TB
- Max lenght of names/identifiers - 128 (255) chars vs 63 bytes
- PostgreSQL shortens identifiers to 63B - no error, just notify



Different Migration Tools Exist

- Different tools are available for migrations
 - Open source, and commercial - with specific use cases
 - Some use CDC (Change Data Capture) & minimal downtime
 - Others require downtime and offline migration
-
- Changes in open source projects can be slow
 - Some of them are not maintained anymore
 - To dynamically address issues we created credativ-pg-migrator
 - We can quickly add new features for specific use cases



Meet credativ Pg-migrator

Meet credativ-pg-migrator

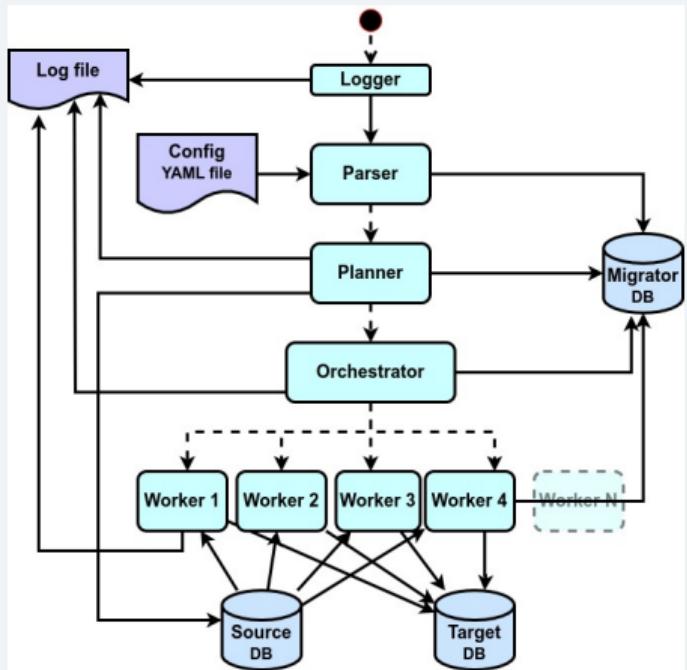


- Inspired by pgloader created by Dimitri Fontaine
- Intended for offline migrations, speed depends on hardware
- Written in Python - JDBC, ODBC or python native DB access
- Other languages have limited support for older DBs
- Written in classes, dynamically pluggable connectors
- Uses well documented and stable libraries
- Pyodbc, JayDeBeApi, cx_Oracle, psycopg2, mysql, ...
- YAML configuration file, text log file



credativ-pg-migrator - Architecture

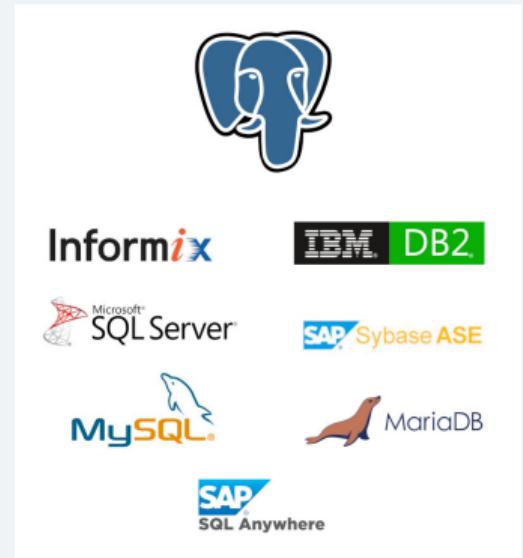
- Modular structure: Parser, Planner, Orchestrator, Workers
- Runs parallel workers, one reader and writer per table
- Speed of migration depends mainly on the hardware
- Creates and fills migration protocol tables
- Protocol tables contain all details about migrated objects
- Outputs detailed INFO and DEBUG messages



credativ-pg-migrator - Features



- Target database is always PostgreSQL
- Supports 7 different source databases
- Informix, Sybase ASE, SQL Anywhere, IBM DB2 LUW
- MS SQL Server, MySQL/MariaDB, (+ PostgreSQL)
- Migrates complete data model from all of them
- Tables, data, indexes, constraints, views
- Allows multiple custom adjustments



- Configurable scope - only schema, only data, both
- Yes/No, Include/exclude - tables, views, funcs, triggers
- Custom defined adjustments:
- Replace Data types (numeric PKs to BIGINT, UDF to standard type)
- Replace Default values (different names of SQL funcs)
- Substitutions of Remote objects references (cross-db refs like db1@tab1)
- Limitatins for Data migrations (time limits: created_at >=, ID in SELECT)



- From Informix can convert Functions, Procedures, Triggers
- Funcs, procs, triggers - yes/no, include/exclude
- Success rate of code conversion 80 to 90%
- Errors mostly due to missing tables, renamed objects/columns
- Some statements may require small manual adjustments
- Conversion of code can be easily added for other DBs too



Future plans for credativ-pg-migrator

- Currently we are adding:
- Migration of data model from Oracle
- Pre-migration analysis of source database
- Analysis of source tables for partitioning/data
- Configurable partitioning for target tables
- Migration of materialized views for relevant DBs
- Conversion of procs, funcs, triggers for supported DBs on demand
- Other source databases on demand

- When code is stable enough, we will open source it



Informix

Informix Dynamic Server (IDS)



- Created by Informix Software in 1980s
- Acquired by IBM in 2001, development now by HCL
- Version 15 released in November 2024
- Concurrency through locking (LOCK MODE ROW/ PAGE)
- UNDO logs, updates in place, show last committed

- credativ-pg-migrator supports full migration from Informix
- Tables, data, indexes, constraints, views
- Conversion of Functions, Procedures, Triggers



Informix – Migration of Data Model

- No schemas, only databases and owners, cross-database references
- Max length of identifiers 128, but also only 18 in old versions
- Names case insensitive, lowercase recommended
- Double quotes do not preserve case, distinguish from keywords
- Data types are very similar, different limits
- BLOB as bytea, CLOB as text - max size 4TB
- BYTE as bytea - max size 2GB
- Complex data types LIST, MULTISET, SET - as ARRAY
- ROW transferred as composite type



- Statements similar to Oracle, JOINs in FROM clause without ON part
- OUTER JOIN -> needs manual rewrite to LEFT OUTER JOIN ON
- SELECT SKIP, SELECT FIRST (LIMIT works in Informix too) -> as LIMIT OFFSET
- Comments with – and /* */ but also with { }
- SET command allows multiple variables: SET (var1, var2) = (val1, val2)
- Majority of built-in SQL functions are the same
- Hierarchical queries with CONNECT BY PRIOR - as recursive CTE

```
SELECT SKIP 5 FIRST 10 * FROM customer;

SELECT c.customer_num, c.lname, o.order_num FROM customer c,
OUTER orders o WHERE c.customer_num = o.customer_num;

SELECT empid, name, mgrid , CONNECT_BY_ISLEAF leaf
FROM employee
START WITH name = 'Smith'
CONNECT BY PRIOR empid = mgrid;
```

- Programming language SPL/4GL, similar to PL/pgSQL
- Functions can return multiple values -> OUT parameters, RECORD or composite type in PG
- Allows "Procedure" returning values, but marks it internally as "Function"
- Global variables in scope of session -> custom config parameters in PG
- Procedures allow CURSOR WITH HOLD -> not allowed in PG stored procedures
- In PostgreSQL cursor or for loop with SELECT FOR UPDATE, if necessary for App logic

```
CREATE PROCEDURE simp_while()
  DEFINE i INT;
  WHILE EXISTS (SELECT fname FROM customer
                 WHERE customer_num > 400)
    DELETE FROM customer WHERE id_2 = 2;
  END WHILE;
  LET i = 1;
  WHILE i < 10
    INSERT INTO tab_2 VALUES (i);
    LET i = i + 1;
  END WHILE;
END PROCEDURE;
```

- Can contain logic and statements and call functions
- Actions: BEFORE, FOR EACH ROW, AFTER or INSTEAD OF FOR EACH ROW
- FOR EACH ROW - after DML is executed but before write into log and table
- Trigger body can contain all these actions
- Actions also on SELECT and INSTEAD OF on view
- On SELECT - can be replaced with RULE ON SELECT in PG

```
CREATE TRIGGER adr_trg UPDATE OF fname, lname ON adr
REFERENCING OLD AS o NEW AS n
    BEFORE
        WHEN ( o.fname != n.fname OR o.lname != n.lname )
            INSERT INTO adrlog (old_value,new_value) VALUES (o.fname,n.fname)
    FOR EACH ROW
        WHEN ( o.fname != n.fname )
            ( INSERT INTO adrlog (old_value,new_value) VALUES (o.fname,n.fname) ),
        WHEN ( o.lname != n.lname )
            ( INSERT INTO adrlog (old_value,new_value) VALUES (o.lname,n.lname) )
    AFTER
        ( INSERT INTO adrlog (old_value,new_value) VALUES (o.fname,n.fname) );
```

IBM DB2

- First released in 1983, based on System R project
- System R was the source of SQL and relational DBs
- Runs on Linux, Unix, Windows (LUW), and mainframes (z/OS)
- LUW and z/OS are different products with specific features
- SELECT 1 FROM SYSIBM.SYSDUMMY1; - common idiom
- credativ-pg-migrator supports DB2 LUW (tested on 11.5)
- Migrates tables, data, indexes, constraints, views

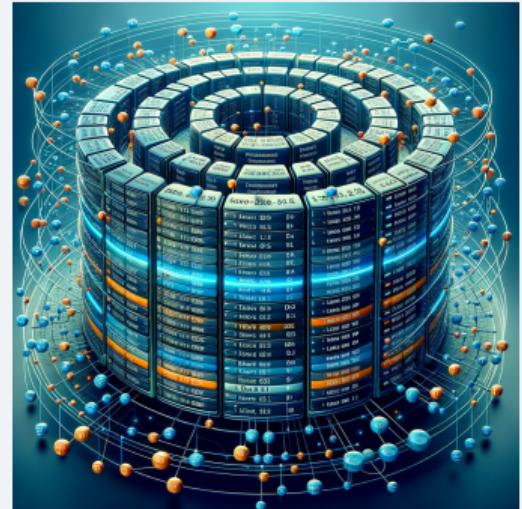


- Names case-sensitive, by default UPPER-CASE, double quotes for preserving case, max length 128 chars
- Data types similar, BLOB & CLOB limit 2GB, VARCHAR to 32KB, JSON (stored as texts), XML, Large objects
- Global variables in scope of schema, but values in scope of session
- Programming language SQL PL, similar to PL/pgSQL
- Implements "basic" SQL triggers and "advanced" SQL PL triggers

```
CREATE PROCEDURE iterator()
LANGUAGE SQL
BEGIN
DECLARE v_dept CHAR(3);
-- ...
DECLARE not_found CONDITION FOR SQLSTATE '02000';
DECLARE c1 CURSOR FOR SELECT deptno, deptname, admrdept FROM department ORDER BY deptno;
DECLARE CONTINUE HANDLER FOR not_found SET at_end = 1;
OPEN c1;
ins_loop:
LOOP
    FETCH c1 INTO v_dept, v_deptname, v_admdept;
    IF at_end = 1 THEN
        LEAVE ins_loop;
    END IF;
    -- ...
END LOOP;
CLOSE c1;
END @
```

IBM DB2 – Advanced Features

- IBM DB2 has many special features
- BLU Acceleration - in-memory columnar data processing
- Adaptive compression, parallel vector processing
- Advanced Data Partitioning - Multidimensional clustering
- High Availability Disaster Recovery - SYNC/NEARS/ASYNC/SUPERAS
- pureScale - clustering for high availability and scalability
- Full migration to PostgreSQL might require re-architecting



Sybase ASE

Sybase Adaptive Server Enterprise (ASE)



- Created 1987 as Sybase, acquired by SAP in 2010
 - No new features lately, last version 16.1 in 2022
 - Uses special "Tabular Data Stream" (TDS) protocol
 - Max length of identifiers 255 chars, but also 30
 - Special internal schemas (dbo), allows cross-database refs
-
- credativ-pg-migrator migrates data model
 - Tables, data, indexes, constraints, views
 - All custom defined replacements
 - Conversion of Funcs/Procs, Triggers can be easily added



- Data types very similar, VARCHAR up to 32KB, TEXT/LONG (N)VARCHAR up to 2GB
- IMAGE up to 2GB, @@textsize (def 32KB) limits TEXT/IMAGE returned data
- Empty string stored as 1 space character, never evaluated as NULL
- Allows User Defined Data Types (UDT) -> PG domain types but problem with FK
- Allows FOREIGN KEY between UDT and base data type -> PG not
- Does not have many SQL features, stored procedures rudimentary, syntax similar to PL/pgSQL
- Triggers rudimentary, "inserted"/"deleted" pseudo-tables, for statements, not row level

```
CREATE TRIGGER tr_audit_employee_update ON employee FOR UPDATE AS
BEGIN
    IF UPDATE(salary)
    BEGIN
        INSERT INTO employee_audit (emp_id, old_salary, new_salary, changed_by, change_time)
        SELECT d.emp_id, d.salary, i.salary, SUSER_SNAME(), GETDATE()
        FROM inserted i JOIN deleted d ON i.emp_id = d.emp_id
        WHERE i.salary <> d.salary -- Log only if salary actually changed
    END
END
```

MS SQL Server

- Created as a fork of Sybase ASE in 1989, later rewritten
- Basic data types very similar, bigger limits
- TEXT, XML, IMAGE, VARBINARY, VARCHAR - max size 2GB
- Max lenght of identifiers 128 chars
- Has schemas in database, allows cross-database references
- Conversion of standard data types is easy
- credativ-pg-migrator converts data model
- Tables, data, indexes, constraints, views



Microsoft SQL Server - Special Data Types

- UNIQUEIDENTIFIER type - 16-byte Globally Unique Identifier (GUID)
- Example: 6F9619FF-8B86-D011-B42D-00C04FC964FF -> UUID in PG
- Default value NEWID() or NEWSEQUENTIALID() -> uuid_generate_v5()
- Column ROWVERSION (binary 8) -> BIGINT + trigger increment in PG(?)
- SQL_VARIANT data type can store INT, BINARY or CHAR -> analysis(?)
- HIERARCHYID -> Itree but analysis needed
- FILESTREAM (BLOB stored as external file) unlimited size
- These special cases need custom analysis



Microsoft SQL Server – Special Features

- CTE implicitly recursive if it references itself
- Query might need adding explicit RECUSIVE in PG
- Query can have OPTION part with multiple hints (remove)
- Local temp tables (#table_name) visible in session
- Globally visible temporary tables (##table_name)
- Global tempdb permanent tables
- Cross-database queries/calls (server.database.schema.object)
- Service Broker configurable asynchronous messaging
- Custom analysis and adjustments necessary



MySQL / MariaDB

- Created in 1995 by MySQL AB
- Acquired by Sun in 2008, by Oracle in 2010
- Oracle acquired Innobase in 2005 (InnoDB engine)
- Forked as MariaDB in 2009 by the original developers
- Over years, paths of both databases diverged
- Specific features depend on storage engine



Main Differences

- Does not have schemas, only databases
- Allows cross-database references on the same server
- Special default/null values -> need custom substitution
- Data types very similar, different limits
- TEXT max 64KB, LONGTEXT/BLOB max 4GB, JSON max 1GB
- Max length of identifiers 64 chars
- credativ-pg-migrator migrates data model
- Tables, data, indexes, constraints, views
- Uses INFORMATION_SCHEMA (engine independent)



SQL Anywhere

- Watcom in 1992, 1995 Sybase, 2010 SAP
- Used mainly in embedded systems/ mobile apps
- Last stable release 17 from 2015
- Two SQL dialects: Watcom SQL and Transact-SQL (T-SQL)
- Uses schemas, cross-database refs with CREATE SERVER
- Max length of identifiers: 128 chars
- credativ-pg-migrator uses ODBC/ JDBC/ sqlanydb
- Migrates tables, data, indexes, constraints, views



- Implements JSON, XML, geospatial data types, BLOBS, long texts - up to 2GG
- Two SQL dialects: Watcom SQL and Transact-SQL (T-SQL) - syntax similar to PL/pgSQL
- Allows temporary funcs/procs, function returns only one value
- Can call external code from libraries (Perl, PHP, Java)
- Global variables in scope of session, user cannot define global vars over all sessions
- Triggers very similar to PostgreSQL, but code in the body of the trigger
- Conversion of code very straightforward, we can add it on demand

```
CREATE FUNCTION fullname( firstname CHAR(30), lastname CHAR(30) )
RETURNS CHAR(61)
BEGIN
    DECLARE name CHAR(61);
    SET name = firstname || ' ' || lastname;
    RETURN (name);
END;
```

PostgreSQL to PostgreSQL

PostgreSQL as Source Database

- Intended for special migration scenarios
- Focused strongly custom replacements and data limitations
- Migrating only a subset of the database
- Like separating living data from archived data
- Combining multiple databases into one as schemas
- Or separating schemas into independent databases
- Separating parts of data model into schemas of DBs
- Soon tool will allow to add or change partitioning
- Makes these actions much easier than manual migrations



Summary

- Migrates data models from 7 different databases:
- Informix, Sybase ASE, SQL Anywhere, IBM DB2 LUW, MS SQL, MySQL/MariaDB
- Replaces of Data types, Default values, Remote references
- Limitations for Data migrations (based on dates, etc.)
- For Informix migrates also triggers, stored procs/funcs

- If you are interested in using the tool, please contact us
- Every database is somehow special - we can add on demand feature you need
- We help you migrate, you will help improve this tool for the community
- When code is stable enough, we will open source it





Questions?