Beyond PostgreSQL

The extensibility of PostgreSQL

Dirk Krautschick
Nordic PG Day, 22 March 2022

trivadis
Part of Accenture

ABOUT ME...

3 HALLO, GRÜEZI, HI!

DIRK KRAUTSCHICK

- since 03/2019 @Trivadis, Germany, Düsseldorf
- PostgreSQL & Oracle, Trainer
- 13 years DBA & Consulting

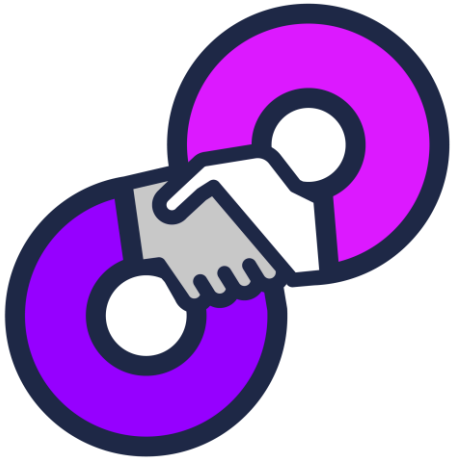
- Married, 2 Junior DBAs
- Mountainbike, swimming, movies, hifi/home cinema

@d33_k4y

ABOUT US...

trivadis
Part of Accenture

5 TRIVADIS & ACCENTURE: #1 FOR DATA & AI



- Together we are 1500 specialists at 34 locations in Switzerland, Germany and Austria with a focus on **Data & Applied Intelligence**.
- Together we support you in the **intelligent end-to-end use of your data.**
- We cover the entire spectrum: from the **development** and **operation** of **data platforms** and **solutions**, to the refinement of data as well as **consulting** and **training**.
- We achieve this through the unique combination of Trivadis' **technological expertise** and Accenture's **strategic know-how** in the field of data.

trivadis
Part of **Accenture**

WARM UP

WARM UP
THE IDEA

- Basic core RDMBS
- Open for more functionality with extensions
- Standardized Interface
- Open and easy implementation

WARM UP
BEING LEIGHTWEIGHT

- Individual set of features
  - Per database
- Only functionality which is really needed
- Benefit of a slight installation

WARM UP
BEING LEIGHTWEIGHT

RDBMS	SOFTWARE	DEFAULT DATABASE
Oracle 19.7	10 GB	6.1 GB
MySQL 8.0	182 MB	173 MB
MariaDB 10.3	122 MB	122 MB
MS SQL Server 2019	1,1 GB	90 MB
PostgreSQL 13.2	29 MB	43 MB

WARM UP
THE POSSIBILITIES

- Special SQL functions, collections or views
- Additional functionalities
- Feature variations
- Further ideas like
  - Alternative storage engines

Exensions insight

EXTENSION INSIGHT
LOCATION

- Located in PostgreSQL software folder
  - Depending on OS distribution
- Subfolder `./share/extension`
- Subfolder `./lib`

```
postgres@localhost /usr/pgsql-14 ]# tree
.
├── bin
├── doc
├── include
├── lib
│   ├── pg_stat_statements.so
│   ├── …
└── share
    ├── contrib
    ├── extension
    │   ├── pg_stat_statements--1.4.sql
    │   ├── pg_stat_statements.control
    │   ├── plpgsql--1.0.sql
    │   ├── plpgsql.control
    │   ├── trivadis_extension--1.0.sql
    │   ├── trivadis_extension.control
…
```

EXTENSION INSIGHT
COMPONENTS

- Control file
  - `trivadis_extension.control`
- Meta information
- Trusted Extensions
  - Since version 13
  - Activation for non superuser

```
# trivadis extension
comment = 'example'
default_version = '1.0’
relocatable = true
trusted = true
```

EXTENSION INSIGHT
COMPONENTS

- SQL Script
  - trivadis_extension--1.0.sql
- Some functionality

```
CREATE FUNCTION trivadis_extension()
RETURNS text
LANGUAGE plpgsql
  AS $$
    DECLARE
      some_text varchar := ' stop war! ';
    BEGIN
    RETURN(some_text);
    END;
$$;
```

EXTENSION INSIGHT
DISCLAIMER

I'M NOT A DEVELOPER ☺

EXTENSION INSIGHT
COMPONENTS

- Extension Building Interface
  - PGXS
- Install postgresql-devel package
- Using a Makefile

```
EXTENSION = trivadis_extension
DATA = trivadis_extension--1.0.sql

PG_CONFIG = pg_config
PGXS := $(shell $(PG_CONFIG) --pgxs)
include $(PGXS)
```

EXTENSION INSIGHT
IMPLEMENTATION

- Implementation in other languages
- e.g. C for better performance
- Same file structure, same control file
- Modules entry in makefile

```
EXTENSION = trivadis_extension
DATA = trivadis_extension--1.0.sql
MODULES = trivadis_extension
PG_CONFIG = pg_config
PGXS := $(shell $(PG_CONFIG) --pgxs)
include $(PGXS)
```

EXTENSION INSIGHT
IMPLEMENTATION

- trivadis_extension.c

```
#include "postgres.h"
#include "fmgr.h"
#include "utils/builtins.h"

PG_MODULE_MAGIC;

PG_FUNCTION_INFO_V1(trivadis_extension);

Datum
trivadis_extension(PG_FUNCTION_ARGS)
{
    PG_RETURN_TEXT_P(" stop war! ");
}
```

- trivadis_extension- -1.0.sql

```
CREATE FUNCTION trivadis_extension()
RETURNS text
AS '$libdir/trivadis_extension'
LANGUAGE C;
```

EXTENSION INSIGHT
IMPLEMENTATION

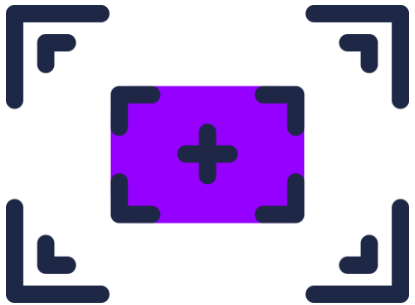
- Installation

```
$ ls -alh
-rw-r--r--.  1 root root   94 Dec 20 12:18 trivadis_extension--1.0.sql
-rw-r--r--.  1 root root  217 Dec 21 16:10 trivadis_extension.c
-rw-r--r--.  1 root root   67 Dec 21 01:15 trivadis_extension.control
$ make
...
$ make install
$ psql
...
postgres=# create extension trivadis_extension;
CREATE
```

Extension sources

EXTENSION SOURCES
FIND THE EXTENSIONS

- Don’t reinvent the wheel
- Several sources
  - Contrib package
  - PostgreSQL Extension Network (PGXN)
  - Everywhere else…google, git, …



EXTENSION SOURCES
CONTRIB

- Official community maintained set of extensions
- Same repository like official PostgreSQL packages
- Around 49 included extensions

EXTENSION SOURCES
PGXN

- PostgreSQL Extension Network (PGXN)
- https://pgxn.org/
- Over 330 extensions
- Additional tools for building and testing extension
- PGXN client to manage extension installations

EXTENSION SOURCES
INSTALLATION

- With package manager (Repositories, rpm, deb, etc.)
- Self-compiling of source code
- For Windows User
  - Only self compiling or
  - Stack Builder (by EDB)

EXTENSION SOURCES
WHAT'S ABOUT CLOUD?

- With IaaS no issue, of course ☺
- Extensions available for DBaaS
  - Depending on the offer of the cloud vendor
  - At least for Azure, AWS and Google a good selection
- Mostly extensions out of contrib

Extension handling

EXTENSION HANDLING
MANAGING EXTENSIONS

- Extension has to be installed
- Simple activation/deactivation of extension per database

```
CREATE EXTENSION <extension_name>;
DROP EXTENSION <extension name> CASCADE;
```

- Configuration in postgresql.conf
  - `shared_preload_libraries` (sometimes)
  - Optional parameters per extension

EXTENSION HANDLING
MANAGING EXTENSIONS

- Modifying an extension (example)

```
ALTER EXTENSION <extension_name>
ADD VIEW <schema_name.view_name>;
```

- Updating an extension

```
ALTER EXTENSION <extension_name>
UPDATE TO <new_version>;
```

EXTENSION HANDLING
MANAGING EXTENSIONS

- Check for installed extensions

```
[local]:5432 postgres@postgres=# \dx

                List of installed extensions

  Name   | Version |   Schema   |         Description
---------+---------+------------+------------------------------
 plpgsql | 1.0     | pg_catalog | PL/pgSQL procedural language
```

EXTENSION HANDLING
MANAGING EXTENSIONS

- Check for already installed and available extension

```
[local]:5432 postgres@postgres=# select * from pg_available_extensions;

      name     | default_version | installed_version |           comment
---------------+-----------------+-------------------+-------------------------------------
plpgsql        | 1.0             | 1.0               | PL/pgSQL procedural language
adminpack      | 2.1             |                   | administrative functions for PostgreSQL
…
pg_qualstats   | 2.0.2           |                   | An extension collecting statistics…
powa           | 4.1.2           |                   | PostgreSQL Workload Analyser-core
```

Cool Extension examples

COOL EXTENSION EXAMPLES
ESSENTIAL EXAMPLES

- Plpgsql
  - Procedural SQL functions (like pl/sql at Oracle DB)
  - Alternative languages available as extensions

- Foreign Data Wrappers
  - „Database Links“
  - Postgres_fdw and file_fdw in contrib
  - Many others (including oracle)

COOL EXTENSION EXAMPLES
ESSENTIAL EXAMPLES

- pg_stat_statements
  - Long term SQL query performance statistics
- pg_wait_sampling
  - Long term WAIT_EVENT statistics
- pg_profile
  - Snapshot based performance reports

COOL EXTENSION EXAMPLES
FAMOUS EXAMPLES

- postGIS
  - Support for spatial and geographical objects
- pgcrypto
  - Cryptographicyl functions

COOL EXTENSION EXAMPLES
SPECIAL EXAMPLES

- hstore
  - Key/value datatype
- cstore_fdw
  - Column store implementation
  - Now part of Citus extension for distributed database

System architects dilemma with Extensions

SYSTEM ARCHITECTS DILEMMA WITH EXTENSIONS
DECISIONS

- Introducing extensions for whatever reasons is a challenge
- New component, change management, etc.
- Many discussions regarding new „software product“
- Cluster restart sometimes necessary ☹

SYSTEM ARCHITECTS DILEMMA WITH EXTENSIONS
PROBLEMS

- Many several software components vs monolith?
  - The need of evaluation/test
  - Even if it is still just an addon to PostgreSQL
  - The wish to have the all-in-one solution

- Pro Arguments
  - Flexibility, lightweightness, tailored solution
  - Don‘t over overestimate it as a large component

SYSTEM ARCHITECTS DILEMMA WITH EXTENSIONS
PROBLEMS

- Sustainability, Future Developement
  - Depending on this functionality
  - How to make sure to get updates and new features?

- Pro Arguments
  - Trust the open source mindset
  - Anyhow, rely on established extensions only
  - Possibilty to continue developement for your own

40 SYSTEM ARCHITECTS DILEMMA WITH EXTENSIONS
PROBLEMS

- What about support
  - Can extension be part of existing SLAs
  - Is there any support from developer?

- Pro Arguments
  - Some vendors are supporting several extensions
  - Don‘t forget the community support

SYSTEM ARCHITECTS DILEMMA WITH EXTENSIONS
OVERALL

- Often extensions are helpful but don‘t affect the system
  - Support not that important
  - Installation uncritical
  - Deinstallation/deactivation quick and easy
- Some extension features are well expected to be in core

Final Words and Discussion

FINAL WORDS
CONCLUSION

- Extensions are very powerful
- ...and often essential
- Base implementation is simple
- Many sources and already good ideas
- Use of extensions should be discussed early in projects
- Choose and prepare your own default selection

trivadis