

rdf_fdw - Foreign Data Wrapper for RDF Triplestores

Motivation

Modern systems increasingly rely on RDF data exposed via SPARQL endpoints to enrich and interlink their data.

Today, integration usually means:

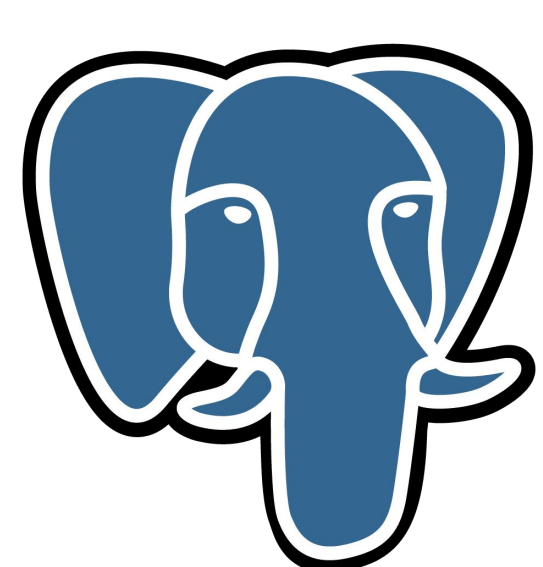
- Writing custom parsers and import pipelines
- Duplicating data across systems
- Maintaining synchronization logic when RDF data changes

This approach is fragile, hard to maintain, and quickly becomes a long-term operational burden.

rdf_fdw removes this barrier.

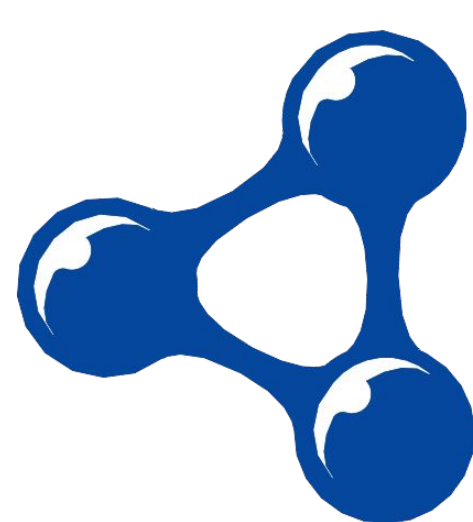
What is rdf_fdw?

rdf_fdw is a PostgreSQL Foreign Data Wrapper that enables direct access to RDF triple stores via SPARQL endpoints, making the Web of Data accessible from PostgreSQL!



SQL → SPARQL Translation
SELECT, INSERT, DELETE, and UPDATE

PostgreSQL Tuples ← RDF
Parsed RDF data into Foreign Tables



It exposes RDF data as foreign tables while pushing as much work as possible to the RDF store — without exporting or duplicating data.

Highlights

- SQL → SPARQL pushdown (filters, ordering, limits)
- Data modification support (INSERT / UPDATE / DELETE)
- Native RDF terms (rdfnode) and SPARQL 1.1 functions
- No export pipelines, no data duplication!

Example

Films and their capital costs from Wikidata.

```
CREATE SERVER wikidata
FOREIGN DATA WRAPPER rdf_fdw
OPTIONS (endpoint 'https://query.wikidata.org/sparql');

CREATE FOREIGN TABLE films (
    name      rdfnode OPTIONS (variable '?filmLabel'),
    cost      rdfnode OPTIONS (variable '?cost'),
    country   rdfnode OPTIONS (variable '?ccode')
)
SERVER wikidata OPTIONS (
    sparql $$
        SELECT ?film ?filmLabel ?cost
        WHERE {
            ?film wdt:P31/wdt:P279* wd:Q11424 ; # film
            wdt:P2130 ?cost ;                # capital cost (USD)
            wdt:P136 wd:Q157443 ;            # genre comedy
            wdt:P1981 wd:Q20644796 ;        # FSK 12
            wdt:P495 ?c .
            ?c wdt:P297 ?ccode .             # country code
            SERVICE wikibase:label {
                bd:serviceParam wikibase:language "en" .
            }
        }
    $$
);

SELECT name FROM films
-- pushed down clauses
WHERE country = 'GB'::rdfnode AND
      cost BETWEEN 3500000 AND 4000000
ORDER BY name ASC, cost DESC
FETCH FIRST ROW ONLY;
```

name
"Monty Python's Life of Brian"@en

(1 row)

How to Get Involved

- Try **rdf_fdw** with your PostgreSQL projects
- Report Issues or suggest features
- Share your use case or feedback



Any feedback us much appreciated!
https://github.com/jimjonesbr/rdf_fdw

Jim Jones

jim.jones@uni-muenster.de

Centre for Information Technology, University of Münster, Germany

wissen.leben