#### One Database To Rule 'em All

PostgreSQL SQL-MED

Percona Live Europe 2017, Dublin Stefanie Janine Stölting

@sjstoelting mail@stefanie-stoelting.de



#### SQL/MED

First defined in ISO/IEC 9075-9:2008, revised by ISO/IEC 9075-9:2016

Supported by

DB2

**MariaDB** 

With CONNECT storage engine, implementation differs to the standard

**PostgreSQL** 



#### **Implementation**

Foreign Data Wrapper

Read only

Read and write

Installation as extensions



#### **Available FDW**

```
Examples:
  Oracle (pgxn.org)
  MS SQL Server / Sybase ASE
    read-only (pgxn.org)
  MongoDB(GitHub)
  MariaDB / MySQL (pgxn.org)
  SQLite
    read-only (GitHub)
  Hadoop (HDFS)
    read-only (GitHub)
  ODBC (GitHub)
```



## Special FDW

```
file_fdw
postgres_fdw
foreign_table_exposer
```



#### Write your own FDW

#### Multicorn

Use Python and Multicorn to write your own and access lots of stuff like

**IMAP** 

HTML



#### Data source

The example data used in the live data part is available from Chinook Database:

PostgreSQL

**MySQL** 

**CSV** 

**SQLite** 



#### **Chinook Tables**

	<b>T</b> tablename
1	Artist
2	Invoice
3	Employee
4	Customer
5	Playlist
6	InvoiceLine
7	Album
8	Genre
9	PlaylistTrack
10	MediaType
11	Track

	I table_name ↔	■ column_name	I data_type ↔
1	Artist	ArtistId	integer
2	Artist	Name	character varying (120)

	I table_name ↔	■ column_name	T data_type ♣
1	Album	AlbumId	integer
2	Album	Title	character varying (160)
3	Album	ArtistId	integer

	T table_name ↔	T column_name ∜	
1	Track	TrackId	integer
2	Track	Name	character varying (200)
3	Track	AlbumId	integer
4	Track	MediaTypeId	integer
5	Track	GenreId	integer
6	Track	Composer	character varying (220)
7	Track	Milliseconds	integer
8	Track	Bytes	integer
9	Track	UnitPrice	numeric

	র GenreId %	■ Name
1	1	Rock
2	2	Jazz
3	3	Metal
4	4	Alternative & Punk
5	5	Rock And Roll
6	6	Blues
7	7	Latin
8	8	Reggae
9	9	Pop
10	10	Soundtrack
11	11	Bossa Nova
12	12	Easy Listening
13	13	Heavy Metal
14	14	R&B/Soul
15	15	Electronica/Dance



#### CTE

#### Common Table Expressions will be used in examples

Example:

```
WITH RECURSIVE t(n) AS (
     VALUES (1)
UNION ALL

SELECT n+1 FROM t WHERE n < 100
)
SELECT sum(n), min(n), max(n) FROM t;</pre>
```

#### Result:

	sum bigint		max integer
1	5050	1	100





```
-- Create the SQLite foreign data wrapper extension in the current database
CREATE EXTENSION sqlite fdw;
-- Create the mapping to the foreign SQLite file
CREATE SERVER sqlite server
FOREIGN DATA WRAPPER sqlite fdw
OPTIONS (database '/var/sqlite/Chinook Sqlite.sqlite')
-- Create the SQLite foreign table, column definitions have to match
CREATE FOREIGN TABLE sqlite artist(
     "ArtistId" integer,
     "Name" character varying(120)
SERVER sqlite server
OPTIONS(
    table 'Artist'
);
```



```
-- Select some data
SELECT * FROM sqlite_artist;
```





```
-- Create the foreign data wrapper extension in the current database CREATE EXTENSION mysql_fdw;

-- Create the mapping to the foreign MariaDB server
CREATE SERVER mariadb_server
FOREIGN DATA WRAPPER mysql_fdw
OPTIONS (host '127.0.0.1', port '3306');

-- Create a user mapping with user and password of the foreign table
-- PostgreSQL gives you options to connect this user with its own users
CREATE USER MAPPING FOR PUBLIC SERVER mariadb_server
OPTIONS (username 'stefanie', password 'secret');
```



```
create the MariaDB foreign table, column definitions have to match
CREATE FOREIGN TABLE mysql_album(
    "AlbumId" integer,
    "Title" character varying(160),
    "ArtistId" integer
)
SERVER mariadb_server
OPTIONS(
    dbname 'Chinook',
    table_name 'Album'
);

-- Select some data
SELECT * FROM mysql_album;
```

	1₁ AlbumId ॐ	T Title ♣	ᠬ͡ᠬ ArtistId ↔
1	1	For Those About To Rock We Salute You	1
2	2	Balls to the Wall	2
3	3	Restless and Wild	2
4	4	Let There Be Rock	1
5	5	Big Ones	3
6	6	Jagged Little Pill	4
7	7	Facelift	5
200 ו	row(s) fetched - 7n	ns	



	■ Name	■ Title
1	AC/DC	Let There Be Rock
2	AC/DC	For Those About To Rock We Salute You
3	Accept	Restless and Wild
4	Accept	Balls to the Wall
5	Aerosmith	Big Ones
6	Alanis Morissette	Jagged Little Pill
7	Alice In Chains	Facelift
8	Antônio Carlos Jobim	Chill: Brazil (Disc 2)
9	Antônio Carlos Jobim	Warner 25 Anos
200 r	ow(s) fetched - 5ms	



```
-- Link foreign tables into the current database and schema
IMPORT FOREIGN SCHEMA public LIMIT TO("Track")
FROM SERVER pg_localhost_chinook
INTO public
```

```
-- Try to select some data
SELECT * FROM "Track";
```

For Those About To Rock (We Salute You) Balls to the Wall Fast As a Shark Restless and Wild Princess of the Dawn Put The Finger On You	1 2 3 3 3	1 2 2 2 2	1 1 1 1	F. Baltes, R.A. Smith-Diesel, S. Kaufman, U
Fast As a Shark Restless and Wild Princess of the Dawn	3 3	2 2 2 2	1 1 1	F. Baltes, S. Kaufman, U. Dirkscneider & V F. Baltes, R.A. Smith-Diesel, S. Kaufman, U
Restless and Wild Princess of the Dawn	3 3	2 2 2	1	F. Baltes, R.A. Smith-Diesel, S. Kaufman, U
Princess of the Dawn	3	2	1	
	3	2	1	Dooff, O.D.A. Croith Diocal
Put The Finger On You	1		•	Deaffy & R.A. Smith-Diesel
0	I	1	1	Angus Young, Malcolm Young, Brian Johns
Let's Get It Up	1	1	1	Angus Young, Malcolm Young, Brian Johns
Inject The Venom	1	1	1	Angus Young, Malcolm Young, Brian Johns
Snowballed	1	1	1	Angus Young, Malcolm Young, Brian Johns
Evil Walks	1	1	1	Angus Young, Malcolm Young, Brian Johns
C.O.D.	1	1	1	Angus Young, Malcolm Young, Brian Johns
Breaking The Rules	1	1	1	Angus Young, Malcolm Young, Brian Johns
Night Of The Long Knives	1	1	1	Angus Young, Malcolm Young, Brian Johns
Spellbound	1	1	1	Angus Young, Malcolm Young, Brian Johns
Go Down	4	1	1	AC/DC
	Inject The Venom Snowballed Evil Walks C.O.D. Breaking The Rules Night Of The Long Knives Spellbound Go Down  fetched - 8ms	Snowballed 1 Evil Walks 1 C.O.D. 1 Breaking The Rules 1 Night Of The Long Knives 1 Spellbound 1 Go Down 4	Snowballed       1       1         Evil Walks       1       1         C.O.D.       1       1         Breaking The Rules       1       1         Night Of The Long Knives       1       1         Spellbound       1       1         Go Down       4       1	Snowballed         1         1         1           Evil Walks         1         1         1           C.O.D.         1         1         1           Breaking The Rules         1         1         1           Night Of The Long Knives         1         1         1           Spellbound         1         1         1           Go Down         4         1         1



```
-- Join SQLite, MariaDB, and PostgreSQL tables
SELECT artist."Name"
    , album."Title"
    , track."Name"
FROM sqlite_artist AS artist
INNER JOIN mysql_album AS album
    ON artist."ArtistId" = album."ArtistId"
INNER JOIN "Track" AS track
    ON album."AlbumId" = track."AlbumId"
;
```

	■ Name %	■ Title	■ Name
1	AC/DC	Let There Be Rock	Go Down
2	AC/DC	Let There Be Rock	Dog Eat Dog
3	AC/DC	Let There Be Rock	Let There Be Rock
4	AC/DC	Let There Be Rock	Bad Boy Boogie
5	AC/DC	Let There Be Rock	Problem Child
6	AC/DC	Let There Be Rock	Overdose
7	AC/DC	Let There Be Rock	Hell Ain't A Bad Place To Be
8	AC/DC	Let There Be Rock	Whole Lotta Rosie
9	AC/DC	For Those About To Rock We Salute You	For Those About To Rock (We Salute You)

200 row(s) fetched - 12ms

```
CREATE EXTENSION file fdw;
-- One does need a server, but afterwards every csv file is avilable
CREATE SERVER chinook csv
FOREIGN DATA WRAPPER file fdw
-- Creating a foreign table based on a csv file
-- Options are the same as in COPY
CREATE FOREIGN TABLE csv genre (
     "GenreId" integer,
    "Name" text
) SERVER chinook csv
OPTIONS (
    filename '/var/tmp/Genre.csv',
    format 'csv',
    HEADER 'true'
);
```



-- Select some data
SELECT \* FROM csv\_genre;

	₩ GenreId	■ Name
1	1	Rock
2	2	Jazz
3	3	Metal
4	4	Alternative & Punk
5	5	Rock And Roll
6	6	Blues
7	7	Latin
8	8	Reggae
9	9	Pop
10	10	Soundtrack
11	11	Bossa Nova
12	12	Easy Listening
13	13	Heavy Metal
14	14	R&B/Soul
15	15	Electronica/Dance

25 row(s) fetched - 1ms



```
-- Join SQLite, MariaDB, PostgreSQL, and CSV tables
SELECT artist."Name"
    , album."Title"
    , track."Name"
    , genre."Name"
FROM sqlite_artist AS artist
INNER JOIN mysql_album AS album
    ON artist."ArtistId" = album."ArtistId"
INNER JOIN "Track" AS track
    ON album."AlbumId" = track."AlbumId"
INNER JOIN csv_genre AS genre
    ON track."GenreId" = genre."GenreId"
;
```

	<b>I</b> Name %	■ Title	■ Name	🗷 Name 🍫		
1	AC/DC	Let There Be Rock	Go Down	Rock		
2	AC/DC	Let There Be Rock	Dog Eat Dog	Rock		
3	AC/DC	Let There Be Rock	Let There Be Rock	Rock		
4	AC/DC	Let There Be Rock	Bad Boy Boogie	Rock		
5	AC/DC	Let There Be Rock	Problem Child	Rock		
6	AC/DC	Let There Be Rock	Overdose	Rock		
7	AC/DC	Let There Be Rock	Hell Ain't A Bad Place To Be	Rock		
8	AC/DC	Let There Be Rock	Whole Lotta Rosie	Rock		
9	AC/DC	For Those About To Rock We Salute You	For Those About To Rock (We Salute You)	Rock		
200 row(s) fetched - 11ms						

200 row(s) fetched - 11ms



	I artist ↔	••• album_titles				
1	AC/DC	'For Those About To Rock We Salute You','Let There Be Rock'				
2	Accept	'Balls to the Wall','Restless and Wild'				
3	Aerosmith	Big Ones				
4	Alanis Morissette	Jagged Little Pill				
5 Alice In Chains		Facelift				
200 r	200 row(s) fetched - 24ms					





```
-- Select the mv data
SELECT *
FROM mv_album_artist
WHERE upper(artist) LIKE 'A%'
ORDER BY artist
;
```

	■ artist	••• album_titles	রি sum ॐ					
3	Academy of St. Martin in the Fields Chamber Ensemble & Sir Neville Marriner	{Sir Neville Marriner: A Celebration}	1					
4	Academy of St. Martin in the Fields, John Birch, Sir Neville Marriner & Sylvia McNair	{Fauré: Requiem, Ravel: Pavane & Others}	1					
5	Academy of St. Martin in the Fields & Sir Neville Marriner	{The World of Classical Favourites}	1					
6	Academy of St. Martin in the Fields, Sir Neville Marriner & Thurston Dart	{Bach: Orchestral Suites Nos. 1 - 4}	1					
7	Academy of St. Martin in the Fields, Sir Neville Marriner & William Bennett	NULL	[NULL]					
8	Accept	{Balls to the Wall,Restless and Wild}	2					
9	AC/DC	{For Those About To Rock We Salute You,Let There Be Rock}	2					
10	A Cor Do Som	NULL	[NULL]					
11	Adrian Leaper & Doreen de Feis	{Górecki: Symphony No. 3}	1					
26 r	26 row(s) fetched - 2ms							



```
-- SELECT the amount of albums from the MariaDB table from MariaDB, not with a foreign data
wrapper
SELECT count( * ) AS AlbumCount
FROM `Album`
;
```

	1.1 AlbumCount		
1	347		

1 row(s) fetched - 8ms



```
-- Insert data calculated from foreign tables using PostgreSQL features into another foreign table
INSERT INTO mysql_album("AlbumId", "ArtistId", "Title")
WITH album AS
          -- Generate a new album id
          SELECT MAX(album."AlbumId") + 1 AS new album id
          FROM mysql album AS album
SELECT album.new album id
     , artist."ArtistId"
     , 'Back in Black'
FROM sqlite artist AS artist, album
WHERE artist."Name" = 'AC/DC'
GROUP BY album.new_album_id
     , artist."ArtistId"
Updated Rows 1
```

1 row(s) fetched - 19ms



```
-- SELECT the amount of albums from the MariaDB table from MariaDB, not with a foreign data
wrapper
SELECT count( * ) AS AlbumCount
FROM `Album`
;
```

	1.1 AlbumCount		
1	348		
1 rc	w(s) fetched - 5ms		



```
-- Select data from the materialized view

SELECT *
FROM mv_album_artist
WHERE artist = 'AC/DC'
ORDER BY artist;

I artist  artist  artist  artist

T artist  artist
```

-- Refresh the mv to see the recently added data REFRESH MATERIALIZED VIEW mv\_album\_artist;



```
-- We can even delete data from foreign tables
DELETE FROM mysql_album
WHERE "Title" = 'Back in Black'
AND "ArtistId" = 1
:
```



```
-- Using PostgreSQL JSON with data from MariaDB and SQLite
-- Step 1: Albums with tracks as JSON
WITH albums AS
(
SELECT a. "ArtistId" AS artist_id
, a. "Title" AS album_title
, array_agg(t. "Name") AS album_tracks
FROM mysql_album AS a
INNER JOIN "Track" AS t
ON a. "AlbumId" = t. "AlbumId"
GROUP BY a. "ArtistId"
, a. "Title"
)
SELECT row_to_json(albums) AS album_tracks
FROM albums
;
```

#### 

200 row(s) fetched - 12ms



```
-- Albums including tracks with aritsts with some JSON magic
WITH albums AS
          SELECT a. "ArtistId" AS artist id
               , a. "Title" AS album title
               , array agg(t."Name") AS album tracks
          FROM mysql album AS a
          INNER JOIN "Track" AS t
               ON a."AlbumId" = t."AlbumId"
          GROUP BY a. "ArtistId"
               , a. "Title"
, js albums AS
          SELECT row to json(albums) AS album tracks
          FROM albums
SELECT a. "Name" AS artist
     , jsonb pretty(al.album tracks::jsonb) AS albums tracks
FROM sqlite artist AS a
INNER JOIN is albums AS al
    ON a. "ArtistId" = (al.album tracks→>'artist id')::int
```



	🗷 artist 🖖	T	albums_tracks	<b>%</b>
1	AC/DC	<b>{</b> ¶	"artist_id": 1,¶	"album_title": "For Those About To Roc
2	AC/DC	<b>{</b> ¶	"artist_id": 1,¶	"album_title": "Let There Be Rock",¶
3	Accept	<b>{</b> ¶	"artist_id": 2,¶	"album_title": "Balls to the Wall",¶ "a "artist_id": 1, "album_title": "For Those About To Rock We Salute Yo
4	Accept	<b>{</b> ¶	"artist_id": 2,¶	"album_title": "Restless and Wild",¶ " "album_tracks": [
5	Aerosmith	<b>{</b> ¶	"artist_id": 3,¶	"album_title": "Big Ones",¶ "album_ti "Spellbound",
6	Alanis Morissette	<b>{</b> ¶	"artist_id": 4,¶	"album_title": "Jagged Little Pill",¶ "al "Night Of The Long Knives", "Breaking The Rules",
7	Alice In Chains	<b>{</b> ¶	"artist_id": 5,¶	"album_title": "Facelift",¶ "album_tra "C.O.D.",
8	Apocalyptica	<b>{</b> ¶	"artist_id": 7,¶	"album_title": "Plays Metallica By Four "Evil Walks",
9	Audioslave	<b>{</b> ¶	"artist_id": 8,¶	"album_title": "Revelations",¶ "album" "Snowballed",
200 r	ow(s) fetched - 18ms			



-- Create the multicorn extension CREATE EXTENSION multicorn;

Name	Value
Query	Create the multicorn extension CREATE EXTENSION multicorn
Updated Rows	0
Finish time	Thu Nov 03 19:03:17 EET 2016



```
CREATE SERVER rss_srv foreign data wrapper multicorn options (
    wrapper 'multicorn.rssfdw.RssFdw'
)
;
```

Name	Value
Query	Create the server, which is simply a placeholder CREATE SERVER rss_srv foreign data wrapper multicorn options ( wrapper 'multicorn.rssfdw.RssFdw' )
Updated Rows	0
Finish time	Thu Nov 03 19:05:47 EET 2016



```
-- Create a foreign table based on an RSS feed
CREATE FOREIGN TABLE rss_postgresql_events (
    title CHARACTER VARYING,
    link CHARACTER VARYING,
    description CHARACTER VARYING,
    "pubDate" TIMESTAMPTZ,
    guid CHARACTER VARYING
) server rss_srv OPTIONS (
    url 'https://www.postgresql.org/events.rss')
:
```

Name	Value
Query	Create a foreign table based on an RSS feed CREATE FOREIGN TABLE rss_postgresql_events (     title CHARACTER VARYING,     link CHARACTER VARYING,     description CHARACTER VARYING,     "pubDate" TIMESTAMPTZ,     guid CHARACTER VARYING ) server rss_srv OPTIONS (     url 'https://www.postgresql.org/events.rss' )
Updated Rows	0
Finish time	Thu Nov 03 19:07:43 EET 2016



```
-- Query the RSS feed
SELECT *
FROM rss_postgresql_events
;
```





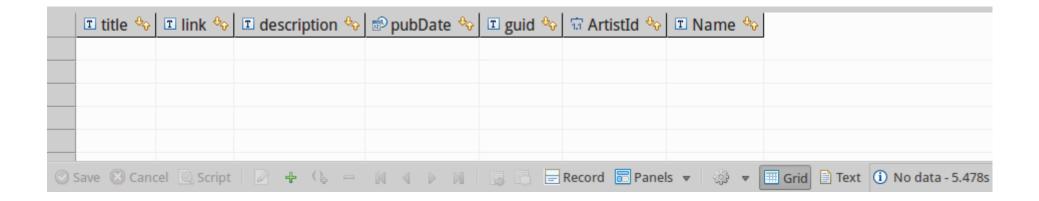
```
-- Entend the query of the RSS feed
SELECT title
    , "pubDate"::DATE AS "Conference Start Date"
    , description
FROM rss_postgresql_events
WHERE "pubDate"::DATE > NOW()::DATE
ORDER BY "pubDate" ASC
;
```

	■ title	🗈 Conference Start Date 🍫	■ description
1	Postgres Conference US Local: Philly¶	2017-07-13	¶ <a <="" href="https://www.pgconf.us/conferences/Philly201" th=""></a>
2	PGConf US Local: Seattle 2017¶	2017-08-11	¶ <h1>PGConf US in partnership with SeaPug is pleased to a</h1>
3	PGDay Austin 2017: Don't Mess with Postgres¶	2017-08-26	¶PGDay Austin 2017 offers a full day of PostgreSQL cont
4	PostgresOpen SV 2017¶	2017-09-06	¶ <a href="https://2017.postgresopen.org">PostgresOpe</a>
5	PGBR2017 The Brazilian PostgreSQL Conference¶	2017-09-14	¶PGBR2017 is the seventh edition of The Brazilian Confe
_	DCD IT 2047#	2017 10 12	Mark TI DCD TT 3 In females (4-1) de-disease De-4
	Save 🗵 Cancel 🗓 Script 🛛 🗗 🗘 🗕 🔞 🔞 🕨	Record  Panels	▼ Grid Text 10 10 row(s) fetched - 1.146s (+1ms)



```
CREATE FOREIGN TABLE rss_music_news (
    title CHARACTER VARYING,
    link CHARACTER VARYING,
    description CHARACTER VARYING,
    "pubDate" TIMESTAMPTZ,
    guid CHARACTER VARYING
) server rss_srv OPTIONS (
    url 'http://www.music-news.com/rss/UK/news?includeCover=false'
);

SELECT *
FROM rss_music_news AS r
    INNER JOIN sqlite_artist AS a
        ON r.title ilike '%' || a."Name" || '%'
;
```





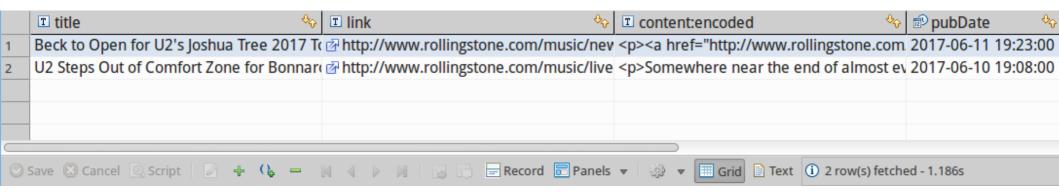
```
CREATE FOREIGN TABLE rss_rolling_stone (
    title CHARACTER VARYING,
    link CHARACTER VARYING,
    "content:encoded" CHARACTER VARYING,
    "pubDate" TIMESTAMPTZ,
    guid CHARACTER VARYING
) server rss_srv OPTIONS (
    url 'http://www.rollingstone.com/music/rss'
)
;

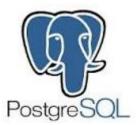
SELECT *
FROM rss_rolling_stone
;
```





```
SELECT *
FROM rss_rolling_stone
;
```



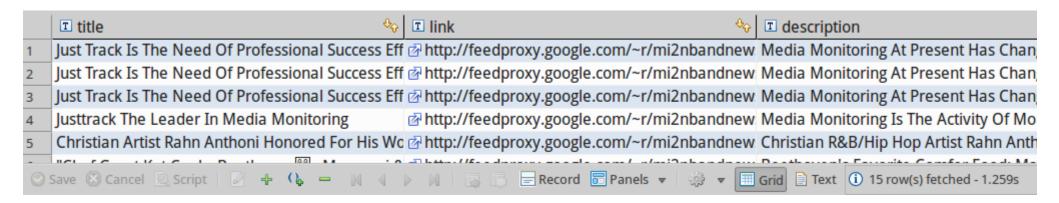


	র ArtistId ॐ	🗷 Name 🍫	I title ❖	🖟 🗷 link 😘	■ content:encoded
1	150	U2	Beck to Open for U2's Joshua Tree 2017	ি 🗹 http://www.rollingstone.com/music/ne	v



```
CREATE FOREIGN TABLE rss_mi2nbandnews (
        title CHARACTER VARYING,
        link CHARACTER VARYING,
        description CHARACTER VARYING,
        "pubDate" TIMESTAMPTZ,
        guid CHARACTER VARYING
) server rss_srv OPTIONS (
        url 'http://feeds.feedburner.com/mi2nbandnews')
;

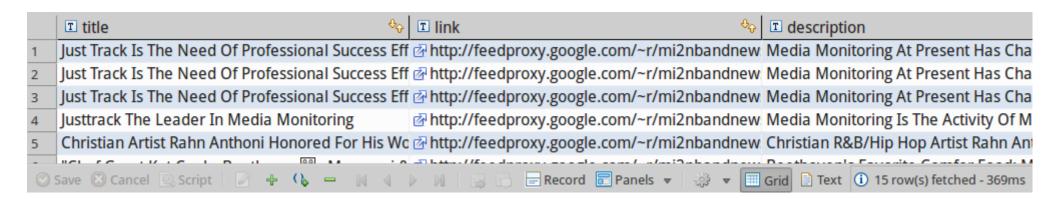
SELECT * FROM rss_mi2nbandnews;
```





```
CREATE FOREIGN TABLE rss_mi2neventnews (
    title CHARACTER VARYING,
    link CHARACTER VARYING,
    description CHARACTER VARYING,
    "pubDate" TIMESTAMPTZ,
    guid CHARACTER VARYING
) server rss_srv OPTIONS (
    url 'http://feeds.feedburner.com/mi2nmusicevents')
;

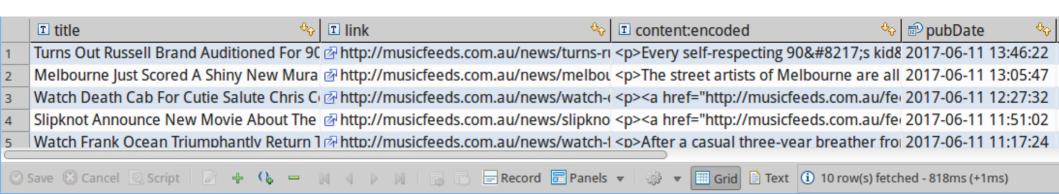
SELECT * FROM rss_mi2nbandnews;
```





```
CREATE FOREIGN TABLE rss_musicfeeds (
    title CHARACTER VARYING,
    link CHARACTER VARYING,
    "content:encoded" CHARACTER VARYING,
    "pubDate" TIMESTAMPTZ,
    guid CHARACTER VARYING
) server rss_srv OPTIONS (
    url 'http://musicfeeds.com.au/news/feed/'
)
;

SELECT * FROM rss_musicfeeds;
```





```
CREATE MATERIALIZED VIEW mv rss music newslists AS
SELECT current timestamp AS refreshed
, ROW NUMBER() OVER() AS rn
, 'Rolling Stone' AS source
 'http://www.rollingstone.com/music/rss' AS url
, r.title
, r."content:encoded" AS content
. TRUE AS encoded
, r.link
, r. "pubDate" AS published
FROM rss rolling stone AS r
UNION
SELECT current timestamp AS refreshed
, ROW NUMBER()OVER() AS rn
 'Music-News' AS source
 'http://www.music-news.com/rss/UK/news?includeCover=false' AS url
, r.title
, r.description AS content
, FALSE AS encoded
, r.link
, r. "pubDate" AS publihed
FROM rss music news AS r
UNION
SELECT current timestamp AS refreshed
, ROW NUMBER()OVER() AS rn
 'MusicFeeds' AS source
 'http://musicfeeds.com.au/news/feed/' AS url
, r.title
 r."content:encoded" AS content
. TRUE AS encoded
, r.link
, r. "pubDate" AS published
FROM rss musicfeeds AS r
UNION
SFIFCT current timestamn AS refreshed
```



```
CREATE UNIQUE INDEX udx_mv_rss_music_newslists_source_rn
    ON mv_rss_music_newslists USING btree
    (source, rn)
;

REFRESH MATERIALIZED VIEW CONCURRENTLY mv_rss_music_newslists;

SELECT *
FROM mv_rss_music_newslists AS r
    INNER JOIN sqlite_artist AS a
        ON r.title ilike '%' || a."Name" || '%'
        OR r.content ilike '%' || a."Name" || '%'
;
```

		ফি rn ॐ	I source ↔	□ url 🍫	<b>I</b> title ♣	■ content	
1	2017-06-11 23:23:28	3	MusicFeeds	☑ http://musicfeeds.com.au/news/feed/	Watch Death Cab For Cutie Salute Chris C	<a href="&lt;/th"></a>	
2	2017-06-11 23:23:28	7	MusicFeeds		Looks Like Migos Are About To Announce	Concert	
3	2017-06-11 23:23:28	8	MusicFeeds	☑ http://musicfeeds.com.au/news/feed/	The 50 Most Overlooked Songs Of The '90	<a href="&lt;/td"></a>	
4	2017-06-11 23:23:28	8	MusicFeeds		The 50 Most Overlooked Songs Of The '90	<a href="&lt;/td"></a>	
5	2017-06-11 23:23:28	18	Rolling Stone	ৌ http://www.rollingstone.com/music/rss	Katv Perrv: Tavlor Swift Tried to 'Assassina	<a href="&lt;/td"></a>	
Ø :	Save   Cancel   Script    Text   Torow(s) fetched - 464ms (+1ms)						



#### Link List

#### **PGXN** Extensions:

- mysql\_fdw, MySQL/MariaDB FDW
- sqlite\_fdw, SQLite FDW

Slide and source on Github:

https://github.com/sjstoelting/talks/

#### One Database To Rule 'em All



This document by Stefanie Janine Stölting is covered by the Creative Commons Attribution 4.0 International