

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](https://github.com))

MariaDB / MySQL (pgxn.org)

SQLite
read-only ([GitHub](https://github.com))

Hadoop (HDFS)
read-only ([GitHub](https://github.com))

ODBC ([GitHub](https://github.com))



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

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

	T table_name	T column_name	T data_type
1	Artist	ArtistId	integer
2	Artist	Name	character varying (120)

	T table_name	T 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	T data_type
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

	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;
```

- Result:

	sum bigint	min integer	max integer
1	5050	1	100



Live Data examples



Live Data examples

```
-- 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')
```

```
);
```



Live Data examples

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

	ArtistId	Name
1	1	AC/DC
2	2	Accept
3	3	Aerosmith
4	4	Alanis Morissette

200 row(s) fetched - 12ms



Live Data examples

-- 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');
```



Live Data examples

-- 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;
```

	AlbumId	Title	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 row(s) fetched - 7ms



Live Data examples

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

	T Name	T 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 row(s) fetched - 5ms



Live Data examples

```
CREATE EXTENSION postgres_fdw;
```

```
-- Create a connection to the other database on the same server
```

```
CREATE SERVER pg_localhost_chinook  
    FOREIGN DATA WRAPPER postgres_fdw  
    OPTIONS (host '127.0.0.1', port '5432', dbname 'chinook')  
;
```

```
-- Create a user mapping
```

```
CREATE USER MAPPING FOR stefanie  
    SERVER pg_localhost_chinook  
    OPTIONS (user 'stefanie', password 'password')  
;
```




Live Data examples

```
-- 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";
```

	TrackId	Name	AlbumId	MediaTypeId	GenreId	Composer
1	1	For Those About To Rock (We Salute You)	1	1	1	Angus Young, Malcolm Young, Brian Johns
2	2	Balls to the Wall	2	2	1	[NULL]
3	3	Fast As a Shark	3	2	1	F. Baltes, S. Kaufman, U. Dirksneider & V
4	4	Restless and Wild	3	2	1	F. Baltes, R.A. Smith-Diesel, S. Kaufman, U
5	5	Princess of the Dawn	3	2	1	Deaffy & R.A. Smith-Diesel
6	6	Put The Finger On You	1	1	1	Angus Young, Malcolm Young, Brian Johns
7	7	Let's Get It Up	1	1	1	Angus Young, Malcolm Young, Brian Johns
8	8	Inject The Venom	1	1	1	Angus Young, Malcolm Young, Brian Johns
9	9	Snowballed	1	1	1	Angus Young, Malcolm Young, Brian Johns
10	10	Evil Walks	1	1	1	Angus Young, Malcolm Young, Brian Johns
11	11	C.O.D.	1	1	1	Angus Young, Malcolm Young, Brian Johns
12	12	Breaking The Rules	1	1	1	Angus Young, Malcolm Young, Brian Johns
13	13	Night Of The Long Knives	1	1	1	Angus Young, Malcolm Young, Brian Johns
14	14	Spellbound	1	1	1	Angus Young, Malcolm Young, Brian Johns
15	15	Go Down	4	1	1	AC/DC

200 row(s) fetched - 8ms



Live Data examples

-- 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"  
;
```

	T Name 📄	T Title 📄	T 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



Live Data examples

```
CREATE EXTENSION file_fdw;
```

```
-- One does need a server, but afterwards every csv file is available
```

```
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'  
);
```



Live Data examples

-- 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



Live Data examples

-- 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"  
;
```

	T Name 📄	T Title 📄	T Name 📄	T 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



Live Data examples

-- Joining SQLite and MariaDB tables using PostgreSQL expressions

```
WITH album AS
(
    SELECT "ArtistId"
           , array_agg("Title") AS album_titles
    FROM mysql_album
    GROUP BY "ArtistId"
)
SELECT artist."Name" AS artist
       , album.album_titles
FROM sqlite_artist AS artist
INNER JOIN album
    ON artist."ArtistId" = album."ArtistId"
;
```

	T 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 row(s) fetched - 24ms



Live Data examples

```
-- Creates an materialized view on foreign tables
CREATE MATERIALIZED VIEW mv_album_artist AS
WITH album AS
(
    SELECT "ArtistId"
        , array_agg("Title") AS album_titles
    FROM mysql_album
    GROUP BY "ArtistId"
)
SELECT artist."Name" AS artist
    , album.album_titles
    , SUM(ARRAY_LENGTH(album_titles, 1))
FROM sqlite_artist AS artist
LEFT OUTER JOIN album
    ON artist."ArtistId" = album."ArtistId"
GROUP BY artist."Name"
    , album.album_titles
;
```



Live Data examples

-- Select the mv data

```
SELECT *  
FROM mv_album_artist  
WHERE upper(artist) LIKE 'A%'  
ORDER BY artist  
;
```

	artist	album_titles	sum
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 row(s) fetched - 2ms

Gri



Live Data examples

-- SELECT the amount of albums from the MariaDB table from MariaDB, not with a foreign data wrapper

```
SELECT count( * ) AS AlbumCount  
FROM `Album`  
;
```

	AlbumCount
1	347

1 row(s) fetched - 8ms



Live Data examples

```
-- 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



Live Data examples

-- SELECT the amount of albums from the MariaDB table from MariaDB, not with a foreign data wrapper

```
SELECT count( * ) AS AlbumCount  
FROM `Album`  
;
```

	AlbumCount
1	348

1 row(s) fetched - 5ms



Live Data examples

```
-- Select data from the materialized view
```

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

	artist	album_titles	sum
1	AC/DC	'For Those About To Rock We Salute You','Let There Be Rock'	2

1 row(s) fetched - 4ms

```
-- Refresh the mv to see the recently added data
```

```
REFRESH MATERIALIZED VIEW mv_album_artist;
```

	artist	album_titles	sum
1	AC/DC	'For Those About To Rock We Salute You','Let There Be Rock','Back in Black'	3

1 row(s) fetched - 4ms

```
-- We can even delete data from foreign tables
```

```
DELETE FROM mysql_album  
WHERE "Title" = 'Back in Black'  
AND "ArtistId" = 1  
;
```



Live Data examples

```
-- 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
;
```

	? album_tracks
1	{"artist_id":133,"album_title":"In Step","album_tracks":["Riviera Paradise","Love Me Darlin'","Scratch-N-Sniff","Wall Of Denial","Travis V
2	{"artist_id":83,"album_title":"Deixa Entrar","album_tracks":["Desaforo","Minha Gata","Medo De Escuro","Asas","Principiando/Decolag
3	{"artist_id":15,"album_title":"The Best Of Buddy Guy - The Millenium Collection","album_tracks":["Talkin' 'Bout Women Obviously","T
4	{"artist_id":78,"album_title":"Vault: Def Leppard's Greatest Hits","album_tracks":["Bringin' On The Heartbreak","Hysteria","Rock Of Ag
5	{"artist_id":205,"album_title":"Carry On","album_tracks":["You Know My Name","Disappearing Act","Silence the Voices","Finally Forev
6	{"artist_id":58,"album_title":"Come Taste The Band","album_tracks":["You Keep On Moving","This Time Around / Owed to 'G' [Instrun
7	{"artist_id":110,"album_title":"Nevermind","album_tracks":["Something In The Way","On A Plain","Stay Away","Lounge Act","Drain Yo
8	{"artist_id":234,"album_title":"Bach: The Brandenburg Concertos","album_tracks":["Concerto No.2 in F Major, BWV1047, I. Allegro"]}

200 row(s) fetched - 12ms



Live Data examples

```
-- Albums including tracks with artists 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 js_albums AS al
    ON a."ArtistId" = (al.album_tracks->'artist_id')::int
;
```



Live Data examples

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



Live Data examples

```
-- 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



Live Data examples

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

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



Live Data examples

```
-- 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



Live Data examples

-- Query the RSS feed

```
SELECT *  
FROM rss_postgresql_events  
;
```

	T title	T link	T description	pubDate	T guid
1	PGConf ASIA 2017	https://www.postgresql.org/about/even	<p>PGConf.ASIA 2017 is an internationa</p>	2017-12-04 01:00:00	https://www.postgresql.org/about/even
2	Australia PGDay 2017	https://www.postgresql.org/about/even	<p>Australia PGDay 2017 will be held in</p>	2017-11-17 01:00:00	https://www.postgresql.org/about/even
3	CHAR(17)	https://www.postgresql.org/about/even	<p>Clustering, High Availability,</p>	2017-11-07 01:00:00	https://www.postgresql.org/about/even
4	PGConf.EU 2017	https://www.postgresql.org/about/even	<p>P</p>	2017-10-24 02:00:00	https://www.postgresql.org/about/even
5	PGDay.IT 2017	https://www.postgresql.org/about/even	<p>Il PGDay.IT è la conferenza italiana c</p>	2017-10-13 02:00:00	https://www.postgresql.org/about/even
-	PGDD2017 The Brazilian Day	https://www.postgresql.org/about/even	<p>PGDD2017 is the seventh edition of</p>	2017-09-14 02:00:00	https://www.postgresql.org/about/even

Save Cancel Script + - < > Record Panels Grid Text 10 row(s) fetched - 5.441s



Live Data examples

```
-- Extend 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	<p>https://www.pgconf.us/conferences/Philly2017</p>
2	PGConf US Local: Seattle 2017	2017-08-11	<p><h1>PGConf US in partnership with SeaPug is pleased to a</p>
3	PGDay Austin 2017: Don't Mess with Postgres	2017-08-26	<p><p>PGDay Austin 2017 offers a full day of PostgreSQL con</p>
4	PostgresOpen SV 2017	2017-09-06	<p><p>PostgresOpe</p>
5	PGBR2017 The Brazilian PostgreSQL Conference	2017-09-14	<p><p>PGBR2017 is the seventh edition of The Brazilian Conf</p>
6	PGDay IT 2017	2017-10-12	<p><p>PGDay IT è la conferenza italiana dedicata a Postgre</p>

Save Cancel Script + - Record Panels Grid Text 10 row(s) fetched - 1.146s (+1ms)



Live Data examples

```
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" || '%'  
;
```

T title	T link	T description	pubDate	T guid	ArtistId	T Name	

Save Cancel Script + () - < > Record Panels Grid Text No data - 5.478s



Live Data examples

```
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  
;
```

	T title	T link	T content:encoded	pubDa
1	Beck to Open for U2's Joshua Tree 2017 T	http://www.rollingstone.com/music/nev	<p><a href="http://www.rollingstone.com	2017-06-1
2	Bobbi Kristina Brown Boyfriend Nick Gord	http://www.rollingstone.com/music/nev	<p>Nick Gordon, the boyfriend of Bobbi K	2017-06-1
3	Katy Perry on Taylor Swift Beef: 'I'm Ready	http://www.rollingstone.com/music/nev	<p><a href="http://www.rollingstone.com	2017-06-1
4	U2 Steps Out of Comfort Zone for Bonnar	http://www.rollingstone.com/music/live	<p>Somewhere near the end of almost ev	2017-06-1
5	Frank Ocean Performs First Concert in Thr	http://www.rollingstone.com/music/nev	<p><a href="http://www.rollingstone.com	2017-06-1

Save Cancel Script + (- Record Panels Grid Text 20 row(s) fetched - 1.102s



Live Data examples

```
SELECT *  
FROM rss_rolling_stone  
;
```

	T title	T link	T content:encoded	pubDate
1	Beck to Open for U2's Joshua Tree 2017 T	http://www.rollingstone.com/music/new	<p><a href="http://www.rollingstone.com	2017-06-11 19:23:00
2	U2 Steps Out of Comfort Zone for Bonnar	http://www.rollingstone.com/music/live	<p>Somewhere near the end of almost ev	2017-06-10 19:08:00

Save Cancel Script

+

+

-

⏮

⏪

⏩

⏭

📄

📄

📄

Record

📄

Panels

⚙

⌵

Grid

Text

2 row(s) fetched - 1.186s



Live Data examples

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

	ArtistId	Name	title	link	content:encoded
1	150	U2	Beck to Open for U2's Joshua Tree 2017 T	http://www.rollingstone.com/music/new	<p><a href="http://ww
2	150	U2	U2 Steps Out of Comfort Zone for Bonnar	http://www.rollingstone.com/music/live	<p>Somewhere near th

Save Cancel Script

+

+

-

⏮

⏪

⏩

⏭

📄

📄

Record

Panels

⚙

Grid

Text

2 row(s) fetched - 428ms



Live Data examples

```
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;
```

	T title	T link	T description
1	Just Track Is The Need Of Professional Success Eff	http://feedproxy.google.com/~r/mi2nbandnew	Media Monitoring At Present Has Chan
2	Just Track Is The Need Of Professional Success Eff	http://feedproxy.google.com/~r/mi2nbandnew	Media Monitoring At Present Has Chan
3	Just Track Is The Need Of Professional Success Eff	http://feedproxy.google.com/~r/mi2nbandnew	Media Monitoring At Present Has Chan
4	Justtrack The Leader In Media Monitoring	http://feedproxy.google.com/~r/mi2nbandnew	Media Monitoring Is The Activity Of Mo
5	Christian Artist Rahn Anthoni Honored For His Wo	http://feedproxy.google.com/~r/mi2nbandnew	Christian R&B/Hip Hop Artist Rahn Anth
...

Save Cancel Script + () - < > Record Panels Grid Text 15 row(s) fetched - 1.259s



Live Data examples

```
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;
```

	T title	T link	T description
1	Just Track Is The Need Of Professional Success Eff	http://feedproxy.google.com/~r/mi2nbandnew	Media Monitoring At Present Has Cha
2	Just Track Is The Need Of Professional Success Eff	http://feedproxy.google.com/~r/mi2nbandnew	Media Monitoring At Present Has Cha
3	Just Track Is The Need Of Professional Success Eff	http://feedproxy.google.com/~r/mi2nbandnew	Media Monitoring At Present Has Cha
4	Justtrack The Leader In Media Monitoring	http://feedproxy.google.com/~r/mi2nbandnew	Media Monitoring Is The Activity Of M
5	Christian Artist Rahn Anthoni Honored For His Wo	http://feedproxy.google.com/~r/mi2nbandnew	Christian R&B/Hip Hop Artist Rahn An
6

Save Cancel Script + () - < > Record Panels Grid Text 15 row(s) fetched - 369ms



```
SELECT * FROM rss_musicfeeds;
```

Save Cancel Script Record Panels Grid Text 10 row(s) fetched - 818ms (+1ms)



PostgreSQL

Live Data examples

```
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 published
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
SELECT current_timestamp AS refreshed
```



Live Data examples

```
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" || '%';
```

	refreshed	rn	source	url	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	<p><a href=
2	2017-06-11 23:23:28	7	MusicFeeds	http://musicfeeds.com.au/news/feed/	Looks Like Migos Are About To Announce	<p>Concert
3	2017-06-11 23:23:28	8	MusicFeeds	http://musicfeeds.com.au/news/feed/	The 50 Most Overlooked Songs Of The '90	<p><a href=
4	2017-06-11 23:23:28	8	MusicFeeds	http://musicfeeds.com.au/news/feed/	The 50 Most Overlooked Songs Of The '90	<p><a href=
5	2017-06-11 23:23:28	18	Rolling Stone	http://www.rollingstone.com/music/rss	Katv Perrv: Tavlör Swift Tried to 'Assassina	<p><a href=

Save Cancel Script + Record Panels Grid Text 17 row(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](#)