

PostgreSQL As Data Integration Tool

PostgreSQL SQL-MED

pgDay Paris 2019/03/12
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)

Please use the [GitHub](https://github.com) repository, the extension is outdated on PGXN

SQLite (pgxn.org)

There is another one (read-only) on [GitHub](https://github.com), which isn't in active development

Hadoop (HDFS)

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

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

There is another one on [PGXN](http://pgxn.org), which isn't in active development

Apache Kafka

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



Special FDW

file_fdw

postgres_fdw

foreign_table_exposer

Some BI systems have problems with foreign tables, that extension solves that



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

CSV

SQLite

MariaDB



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	T 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



Live Data examples



Live Data examples

-- Create the SQLite foreign data wrapper extension in the current database

CREATE EXTENSION sqlite_fdw;



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
```

```
-- IMPORTANT:
```

```
-- Take care that the postgres user has the rights to write
```

```
-- not only on the SQLite file, but on the folder, too
```

```
CREATE SERVER sqlite_server_srv
```

```
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 chinook_sqlite."Artist"(  
"ArtistId" integer OPTIONS (key 'true'),
```

```
"Name" text
```

```
)
```

```
)
```

```
SERVER sqlite_server_srv
```

```
OPTIONS(
```

```
table 'Artist'
```

```
);
```



Live Data examples

-- Select some data

```
SELECT *  
FROM chinook_sqlite."Artist"  
;
```

	123 ArtistId ↑↓	ABC Name ↑↓
1	1	AC/DC
2	2	Accept
3	3	Aerosmith
4	4	Alanis Morissette
5	5	Alice In Chains
6	6	Antônio Carlos Jobim
7	7	Apocalyptica
8	8	Audioslave



Live Data examples

```
-- Update one record inside the SQLite DB
```

```
UPDATE chinook_sqlite."Artist" SET  
"Name" = lower("Name")  
WHERE "ArtistId" = 1  
;
```

```
-- Select the updated record
```

```
SELECT *  
FROM chinook_sqlite."Artist"  
WHERE "ArtistId" = 1  
;
```



123 ArtistId	ABC Name
1	ac/dc



Live Data examples

```
-- Revert the update
UPDATE chinook_sqlite."Artist" SET
"Name" = 'AC/DC'
WHERE "ArtistId" = 1
;

-- Check the revert of the updated record
SELECT *
FROM chinook_sqlite."Artist"
WHERE "ArtistId" = 1
;
```

	123 ArtistId 	ABC Name 
1	1	AC/DC



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_srv
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_srv
OPTIONS (username 'stefanie', password 'secret')
;

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



Live Data examples

-- Select some data

```
SELECT *  
FROM chinook_mariadb."Album"  
;
```

	123 AlbumId 	ABC Title 	123 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
8	8	Warner 25 Anos	6



Live Data examples

```
-- Join SQLite with MariaDB
SELECT artist."Name"
, album."Title"
FROM chinook_sqlite."Artist" AS artist
INNER JOIN chinook_mariadb."Album" AS album
ON artist."ArtistId" = album."ArtistId"
;
```




	ABC Name	ABC 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)



Live Data examples

```
-- Select one album to check it before an update
```

```
SELECT *  
FROM chinook_mariadb."Album"  
WHERE "AlbumId" = 1  
;
```

	123 AlbumId 	ABC Title 	123 ArtistId 
1	1	For Those About To Rock We Salute You	1






Live Data examples

```
-- Update one album in MariaDB from PostgreSQL
```

```
UPDATE chinook_mariadb."Album" SET  
"Title" = 'Updated by PostgreSQL'  
WHERE "AlbumId" = 1  
;
```

```
-- Control the result of the updated album in MariaDB
```

```
SELECT *  
FROM chinook_mariadb."Album"  
WHERE "AlbumId" = 1  
;
```

	123 AlbumId 	ABC Title 	123 ArtistId 
1	1	Updated by PostgreSQL	1



Live Data examples

```
-- Revert the update in MariaDB by the same record in PostgreSQL
```

```
UPDATE chinook_mariadb."Album" AS mariadb_album SET  
"Title" = pg_album."Title"  
FROM "Album" AS pg_album  
WHERE pg_album."AlbumId" = mariadb_album."AlbumId"  
AND mariadb_album."AlbumId" = 1  
;
```

```
-- Control the result of the updated album in MariaDB
```

```
SELECT *  
FROM chinook_mariadb."Album"  
WHERE "AlbumId" = 1  
;
```

	123 AlbumId	ABC Title	123 ArtistId
1	1	For Those About To Rock We Salute You	1



Live Data examples

```
-- Create the PostgreSQL extension to link other PostgreSQL databases
CREATE EXTENSION postgres_fdw;

-- Create a connection to the other database PostgreSQL on the same server (9.6)
CREATE SERVER postgresql_9_6_localhost_chinook_srv
    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 postgresql_9_6_localhost_chinook_srv
    OPTIONS (user 'stefanie', password 'password')
;

-- Link foreign tables into the current database and schema
IMPORT FOREIGN SCHEMA public LIMIT TO("Track")
FROM SERVER postgresql_9_6_localhost_chinook_srv
INTO chinook_postgresql_9_6
;
```



Live Data examples

-- Try to select some data



```
SELECT *  
FROM chinook_postgresql_9_6."Track"  
;
```

	123 TrackId	ABC Name	123 AlbumId	123 MediaTypeId	123 GenreId	ABC Composer	123 Milliseconds	123 Bytes	123 UnitPrice
1	1	For Those About To Rock (We Salute You)	1	1	1	Angus Young, Malcolm Young, Brian Johnson	343719	11170334	0,99
2	2	Balls to the Wall	2	2	1	[NULL]	342562	5510424	0,99
3	3	Fast As a Shark	3	2	1	F. Baltes, S. Kaufman, U. Dirksneider & W. Hoffman	230619	3990994	0,99
4	4	Restless and Wild	3	2	1	F. Baltes, R.A. Smith-Diesel, S. Kaufman, U. Dirksneider & W. Hoffman	252051	4331779	0,99
5	5	Princess of the Dawn	3	2	1	Deaffy & R.A. Smith-Diesel	375418	6290521	0,99
6	6	Put The Finger On You	1	1	1	Angus Young, Malcolm Young, Brian Johnson	205662	6713451	0,99
7	7	Let's Get It Up	1	1	1	Angus Young, Malcolm Young, Brian Johnson	233926	7636561	0,99
8	8	Inject The Venom	1	1	1	Angus Young, Malcolm Young, Brian Johnson	210834	6852860	0,99



Live Data examples

```
-- Join SQLite and PostgreSQL tables
SELECT artist."Name"
, album."Title"
, track."Name"
FROM chinook_sqlite."Artist" AS artist
INNER JOIN chinook_mariadb."Album" AS album
ON artist."ArtistId" = album."ArtistId"
INNER JOIN chinook_postgresql_9_6."Track" AS track
ON album."AlbumId" = track."AlbumId"
;
```

	ABC Name 	ABC Title 	ABC 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



Live Data examples

```
-- Create the file extension
CREATE EXTENSION file_fdw;

-- One does not need a server, but afterwards every csv file is available
CREATE SERVER chinook_csv_srv
FOREIGN DATA WRAPPER file_fdw
;

-- Creating a foreign table based on a csv file
-- Options are the same as in COPY
CREATE FOREIGN TABLE chinook_csv."Genre" (
  "GenreId" integer,
  "Name" text
) SERVER chinook_csv_srv
OPTIONS (
  filename '/var/sqlite/Genre.csv',
  format 'csv',
  HEADER 'true'
)
;
```




Live Data examples

```
-- Select some data
SELECT *
FROM chinook_csv."Genre"
;
```

	123 GenreId ↑↓	ABC 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



Live Data examples

```
-- Join SQLite, two PostgreSQL servers, and a CSV tables
```

```
SELECT artist."Name"  
, album."Title"  
, track."Name"  
, genre."Name"  
FROM chinook_sqlite."Artist" AS artist  
INNER JOIN chinook_mariadb."Album" AS album  
ON artist."ArtistId" = album."ArtistId"  
INNER JOIN chinook_postgresql_9_6."Track" AS track  
ON album."AlbumId" = track."AlbumId"  
INNER JOIN chinook_csv."Genre" AS genre  
ON track."GenreId" = genre."GenreId"  
;
```

	ABC Name	ABC Title	ABC Name	ABC 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



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 chinook_mariadb."Album"
GROUP BY "ArtistId"
)
SELECT artist."Name" AS artist
, album.album_titles
, SUM(ARRAY_LENGTH(album_titles, 1))
FROM chinook_sqlite."Artist" AS artist
LEFT OUTER JOIN album
ON artist."ArtistId" = album."ArtistId"
GROUP BY artist."Name"
, album.album_titles
;
```



Live Data examples

-- Select Data from the materialized view

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

	ABC artist	album_titles	123 sum
1	Aaron Copland & London Symphony Orchestra	{A Copland Celebration, Vol. I}	1
2	Aaron Goldberg	{Worlds}	1
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



Live Data examples

```
-- Create the multicorn extension
CREATE EXTENSION multicorn;

-- Create the server, which is simply a placeholder
CREATE SERVER rss_srv foreign data wrapper multicorn options
(
    wrapper 'multicorn.rssfdw.RssFdw'
)
;

-- Create a foreign table based on an RSS feed
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'
)
;
```



Live Data examples

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

	ABC title	ABC link	ABC description
1	Oprah Winfrey to interview Michael Jackson's Leaving N	http://www.music-news.com/news/UK/119779/Oprah	The TV mogul's best friend, newswoman Gayle King, sn
2	Ryan Adams fans demand ticket refunds amid abuse all	http://www.music-news.com/news/UK/119778/Ryan-/	The singer-songwriter "unequivocally" denies the allega
3	Offset rattled by online comments	http://www.music-news.com/news/UK/119777/Offset	The rapper said he feels a responsibility to "hold his farr
4	Michael Jackson's nephew: 'My uncle's naivety was his d	http://www.music-news.com/news/UK/119775/Micha	Michael Jackson's brother Marlon has said the documer
5	John Legend desperate to be an 'EGOTSMA'	http://www.music-news.com/news/UK/119769/John-L	The singer realised that both Blake Shelton and Adam L
6	Nicki Minaj wants Tracy Chapman lawsuit dismissed	http://www.music-news.com/news/UK/119766/Nicki-I	The veteran musician denied the request to clear the sa
7	R. Kelly facing possible child support charges	http://www.music-news.com/news/UK/119765/R-Kelly	The star is already facing multiple claims of sex abuse.
8	Bebe Rexha jets to New York for family showdown over	http://www.music-news.com/news/UK/119764/Bebe-	The singer's father hasn't spoken to her for two weeks.



Live Data examples

```
-- Link the previous RSS feed to existing data
SELECT a."Name"
, r.title
, r.description
FROM rss_music_news AS r
INNER JOIN chinook_sqlite."Artist" AS a
ON r.title ilike '%' || a."Name" || '%'
;
```

	ABC Name	ABC title	ABC description
1	JET	Bebe Rexha jets to New York for family showdown over racy new video	The singer's father hasn't spoken to her for two weeks.
2	Chris Cornell	Chris Cornell's former doctor fights widow's wrongful death suit	Vicky Cornell testified before U.S. politicians to call for action to curb the opioid crisis.
3	Accept	Lady Gaga 'can't remember' Oscars acceptance speech	Elton John's children sent handwritten congratulation notes to godmother Gaga.



Live Data examples

```
-- We create several tables with RSS feeds containing music news
-- Create a foreign table based on an RSS feed
```

```
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'  
)  
;
```

```
-- Create a foreign table based on an RSS feed
```

```
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'  
)  
;
```




Live Data examples

```
-- Create a foreign table based on an RSS feed
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'
)
;
```



Live Data examples

```
-- This materialized view will contain the results of all RSS music feeds
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
, 'Music Industry News Network: Band News' AS source
, 'http://feeds.feedburner.com/mi2nbandnews' AS url
, r.title
, r.description AS content
, FALSE AS encoded
, r.link
, r."pubDate" AS published
FROM rss_mi2nbandnews AS r
UNION
SELECT current_timestamp AS refreshed
, ROW_NUMBER()OVER() AS rn
, 'Music Industry News Network: Event News' AS source
, 'http://feeds.feedburner.com/mi2nmusicevents' AS url
, r.title
, r.description AS content
, FALSE AS encoded
, r.link
, r."pubDate" AS published
FROM rss_mi2neventnews AS r
;
```



Live Data examples

```
-- The unique index will help to refresh the materilized view  
CREATE UNIQUE INDEX udx_mv_rss_music_newslists_source_rn  
ON mv_rss_music_newslists USING btree  
(source, rn)  
;
```



Live Data examples

```
-- The two materialized views are joined and return data  
-- from a lot of different sources, but this time all queried  
-- inside PostgreSQL
```

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

	refreshed	123 rn	ABC source	ABC url	ABC title
1	2019-02-27 22:51:20	23	Music-News	http://www.music-news.com/rss/UK/news?include	Lady Gaga 'can't remember' Oscars acceptance spee
2	2019-02-27 22:51:20	10	Music Industry News Network: Band News	http://feeds.feedburner.com/mi2nbandnews	The Pretty Things Retire From Electric Performances
3	2019-02-27 22:51:20	9	Music-News	http://www.music-news.com/rss/UK/news?include	Chris Cornell's former doctor fights widow's wrongfu
4	2019-02-27 22:51:20	8	Music-News	http://www.music-news.com/rss/UK/news?include	Bebe Rexha jets to New York for family showdown ov
5	2019-02-27 22:51:20	6	Music Industry News Network: Event News	http://feeds.feedburner.com/mi2nmusicevents	David Bowie: Earthly Messenger - Benefit Concert
6	2019-02-27 22:51:20	8	Music Industry News Network: Band News	http://feeds.feedburner.com/mi2nbandnews	Bongo Boy Blues Artist Sheba The Mississippi Queen



Live Data examples

```
-- Installing the pg_cron extension to schedule jobs in PostgreSQL
-- https://github.com/citusdata/pg_cron
DROP EXTENSION IF EXISTS pg_cron;
CREATE EXTENSION pg_cron;

-- The table is used for logging calls
CREATE TABLE cron.log (
log_id bigserial NOT NULL,
time_stamp_begin timestamp WITH time ZONE NOT NULL,
time_stamp_end timestamp WITH time ZONE NOT NULL,
executed text NOT NULL,
CONSTRAINT log_pk PRIMARY KEY (log_id)
)
;
```



Live Data examples

```
CREATE OR REPLACE PROCEDURE refresh_every_minute() AS $$  
DECLARE  
    ts_start timestamp WITH time ZONE DEFAULT current_timestamp;  
BEGIN  
    -- Refresh the materialized view concurrently to keep it available  
REFRESH MATERIALIZED VIEW CONCURRENTLY mv_rss_music_newslists;  
  
    -- Write a log entry into the log table  
INSERT INTO cron.log (executed, time_stamp_begin)  
VALUES (  
    'REFRESH MATERIALIZED VIEW CONCURRENTLY mv_rss_music_newslists',  
    ts_start  
);  
END;  
$$  
LANGUAGE plpgsql  
;
```



Live Data examples

```
-- Create a cron job inside PostgreSQL that will refresh the  
-- Materialized view with RSS feed data every one minute  
INSERT INTO cron.job (schedule, command, nodename, nodeport, database, username)  
VALUES ('* * * * *', 'CALL refresh_every_minute()', '', 5434, 'chinook', 'postgres')  
;
```

```
SELECT * FROM cron.job;
```

	123 jobid	ABC schedule	ABC command	ABC nodename	123 nodeport	ABC database	ABC username	✓ active
1	1	* * * * *	CALL refresh_every_minute()		5434	chinook	postgres	true

```
SELECT * FROM cron.log;
```



Live Data examples

```
-- Select the log written by the procedure  
SELECT * FROM cron.log;
```




Live Data examples

-- The refresh is running transparent in the background

```
SELECT *  
FROM mv_rss_music_newslists  
;
```

	refreshed	123 rn	ABC source	ABC url	ABC title
1	2019-03-09 20:17:50	1	Music Industry News Network: Band News	http://feeds.feedburner.com/mi2nbandnews	Iron Mike Signs With Memphis Managment Firm, N
2	2019-03-09 20:17:50	1	Music Industry News Network: Event News	http://feeds.feedburner.com/mi2nmusicevents	Art Of Sound: Youth Chorus
3	2019-03-09 20:17:50	1	Music-News	http://www.music-news.com/rss/UK/news?include	Taylor Swift's stalker has 'no regrets' over break-in
4	2019-03-09 20:17:50	2	Music Industry News Network: Band News	http://feeds.feedburner.com/mi2nbandnews	The Wing Agency Is Excited To Represent Drummer
5	2019-03-09 20:17:50	2	Music Industry News Network: Event News	http://feeds.feedburner.com/mi2nmusicevents	UG! Comedy Show!! @ Huron Club: Tuesday March 2
6	2019-03-09 20:17:50	2	Music-News	http://www.music-news.com/rss/UK/news?include	Sara Bareilles: 'I was tired of the pop cycle'
7	2019-03-09 20:17:50	3	Music Industry News Network: Band News	http://feeds.feedburner.com/mi2nbandnews	Country Music Legends Know That Music Should Be
8	2019-03-09 20:17:50	3	Music Industry News Network: Event News	http://feeds.feedburner.com/mi2nmusicevents	Live On Lincoln Presents The Jean P Jam Reggae Expe



Live Data examples

```
-- Remove the scheduled job  
SELECT cron.unschedule(1);
```



Live Data examples

```
-- PostgreSQL Conferences RSS feed
SELECT title
, "pubDate"::DATE AS "Conference Start Date"
, strip_tags(description)
FROM rss_postgresql_events
WHERE "pubDate"::DATE > NOW()::DATE
ORDER BY "pubDate" ASC
;
```

	ABC title	Conference Start Date	ABC strip_tags
1	pgDay Israel 2019	2019-03-14	For the third time in Israel PostgreSQL enthusiast from all over the world join toget
2	Nordic PGDay 2019	2019-03-19	Nordic PGDay 2019 is a one-day PostgreSQL conference, held in Copenhagen, Der
3	PGConf APAC 2019	2019-03-19	On behalf of PGConf APAC organizing committee, Ashnik announces the PGConf A
4	pgconf.de 2019	2019-05-10	PGConf.de 2019 is the sequel of the highly successful German-speaking PostgreSC
5	PGDay.IT 2019	2019-05-16	The PGDay.IT 2019 opens the doors https://2019.pgday.it/en/ It will be held in Bolo
6	PGCon 2019	2019-05-28	PGCon is an annual conference for users and developers of PostgreSQL.
7	Swiss PGDay 2019	2019-06-28	The Swiss conference on PostgreSQL This year's PGDay i Switzerland will be hoste
8	PostgresLondon	2019-07-02	PostgresLondon is the perfect opportunity for PostgreSQL users, developers, and a



Link List

Slide and sources are available on Github:

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

<https://gitlab.com/sjstoelting/talks/>

PostgreSQL As Data Integration Tool



This document by [Stefanie Janine Stölting](#) is covered by the [Creative Commons Attribution 4.0 International](#)