One Database To Rule 'em All

PostgreSQL

PHP Usergroup Berlin
November 2015
Stefanie Janine Stölting

SQL/MED

- Defined by ISO/IEC 9075-9:200
- 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 readonly (pgxn.org)
- MongoDB readonly (pgxn.org)
- MariaDB / MySQL (github.com)

Data source

The example data used in the SQL part is availabale from Chinook Database:

- PostgreSQL
- MySQL
- SQLite

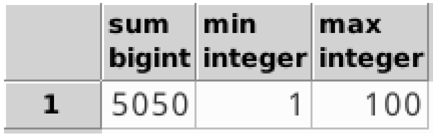
CTE

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	min	max
	bigint	integer	integer
1	5050	1	100

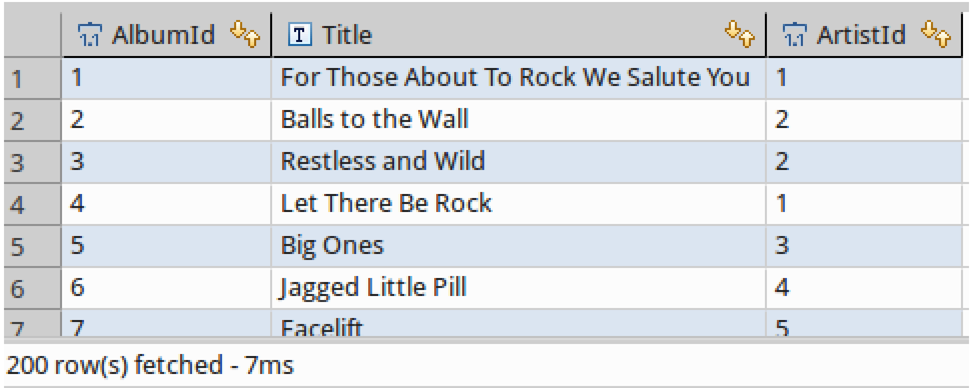
Live Data examples

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 'pg_test', 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

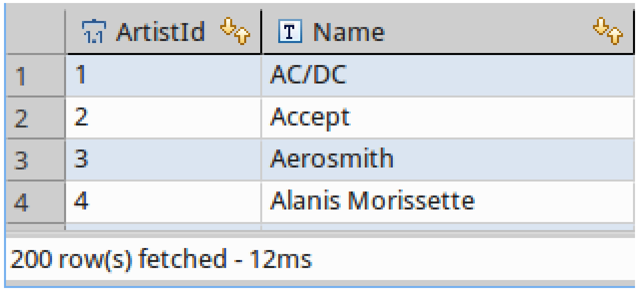
200 row(s) fetched - 7ms

Live Data examples

```
-- 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;
```

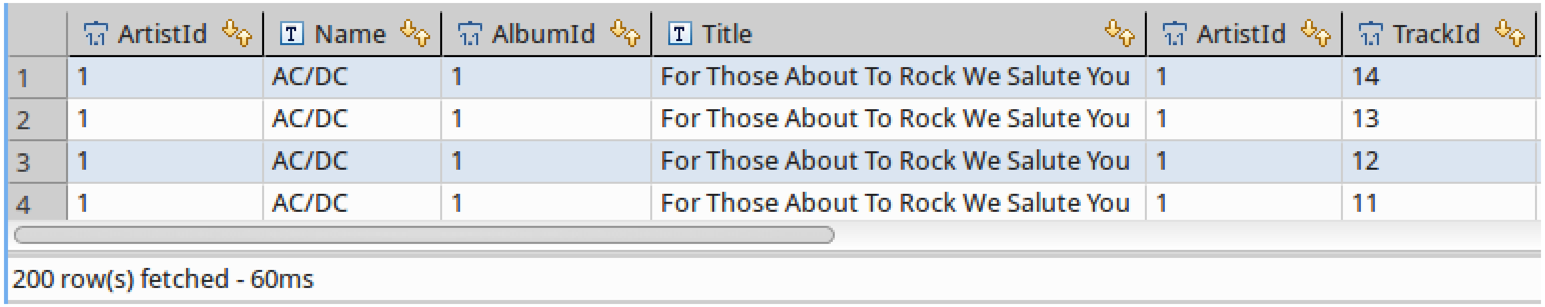


	ArtistId	Name
1	1	AC/DC
2	2	Accept
3	3	Aerosmith
4	4	Alanis Morissette

200 row(s) fetched - 12ms

Live Data examples

```
-- Join PostgreSQL, MariaDB and SQLite tables
SELECT *
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"
;
```

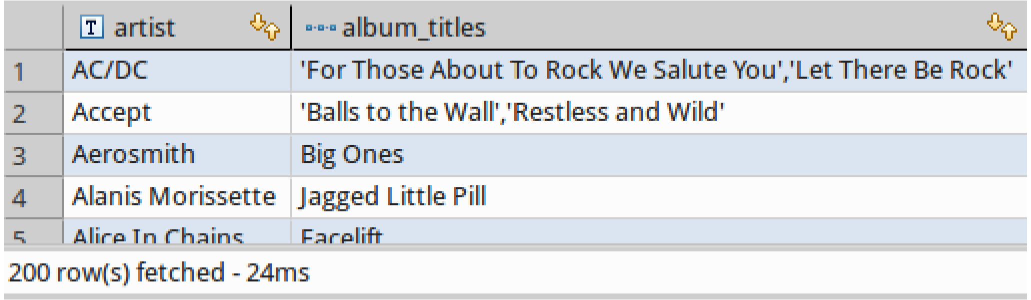


	ArtistId	Name	AlbumId	Title	ArtistId	TrackId
1	1	AC/DC	1	For Those About To Rock We Salute You	1	14
2	1	AC/DC	1	For Those About To Rock We Salute You	1	13
3	1	AC/DC	1	For Those About To Rock We Salute You	1	12
4	1	AC/DC	1	For Those About To Rock We Salute You	1	11

200 row(s) fetched - 60ms

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

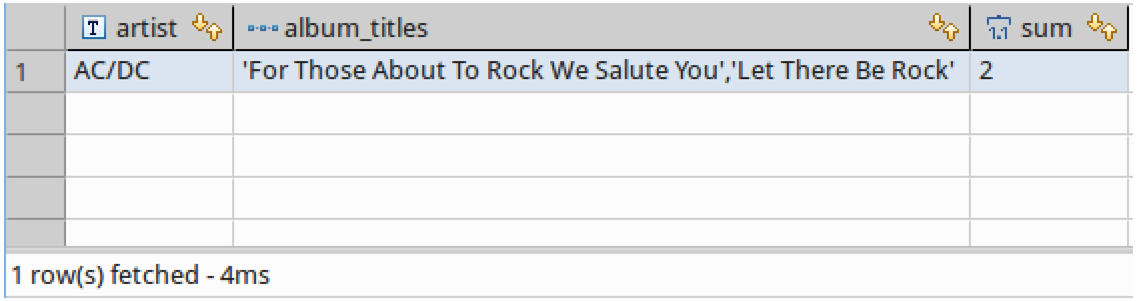
200 row(s) fetched - 24ms

Live Data examples

```sql
-- 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
;


-- Creates a unique index on a mv
CREATE UNIQUE INDEX mv_album_artist__artist ON mv_album_artist(artist);
```

Live Data examples

```
-- Select the mv data
SELECT *
FROM mv_album_artist
WHERE artist = 'AC/DC'
;
```

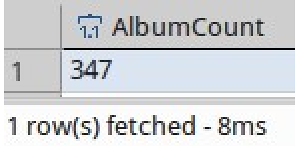


	artist	album_titles	sum
1	AC/DC	'For Those About To Rock We Salute You','Let There Be Rock'	2

1 row(s) fetched - 4ms

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`
;
```

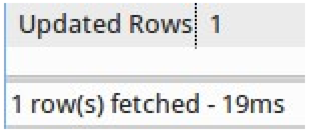


	AlbumCount
1	347

1 row(s) fetched - 8ms

Live Data examples

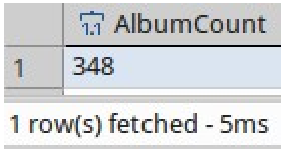
```
-- Insert data calculated from foreign tables using PostgreSQL features into another foreign table
INSERT INTO mysql_album("AlbumId", "ArtistId", "Title")
WITH album AS
(
-- Generate a new album id
SELECT MAX(album."AlbumId") + 1 AS new_album_id
FROM mysql_album AS album
)
SELECT album.new_album_id
, 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`
;
```

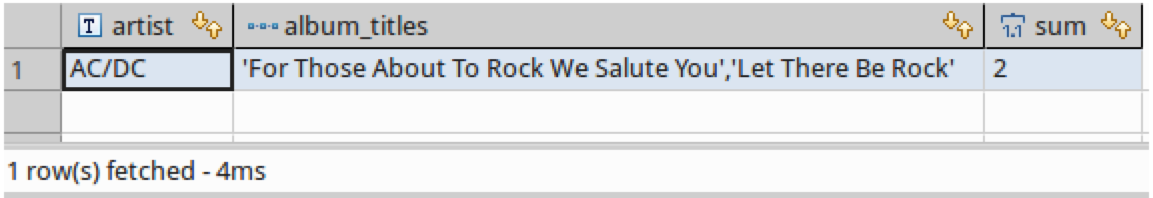


	AlbumCount
1	348

1 row(s) fetched - 5ms

Live Data examples

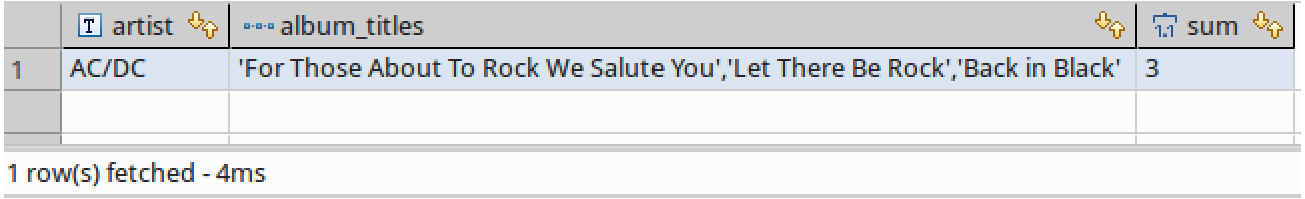
```
-- Select data from the materialized view
SELECT *
FROM mv_album_artist
WHERE artist = 'AC/DC'
ORDER BY artist
;
```



	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;
```



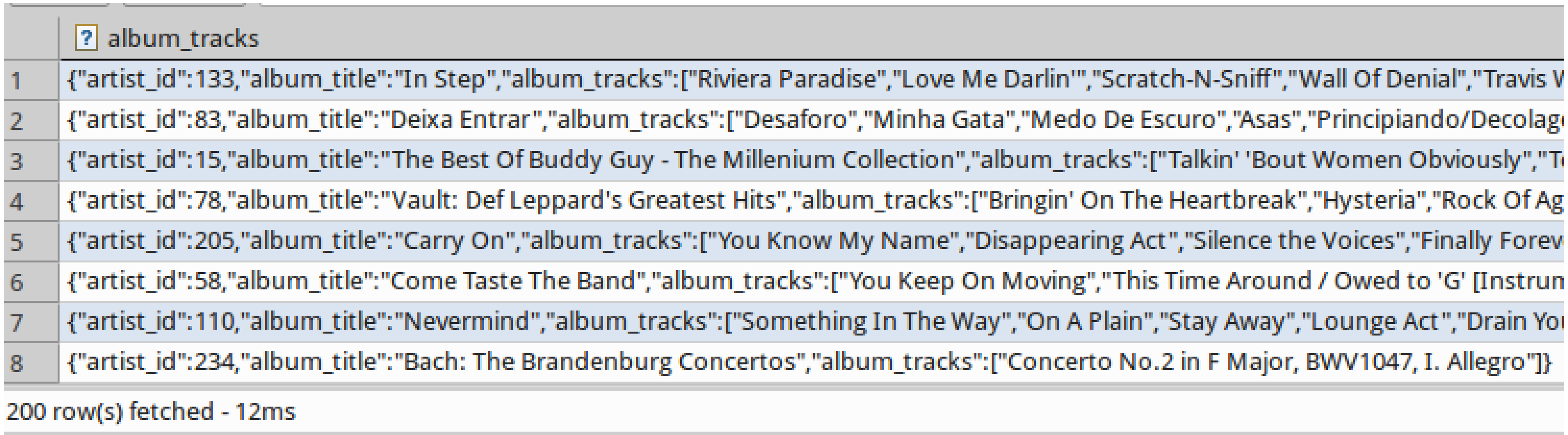
	artist	album_titles	sum
1	AC/DC	'For Those About To Rock We Salute You','Let There Be Rock','Back in Black'	3

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

```sql
-- 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
;
```



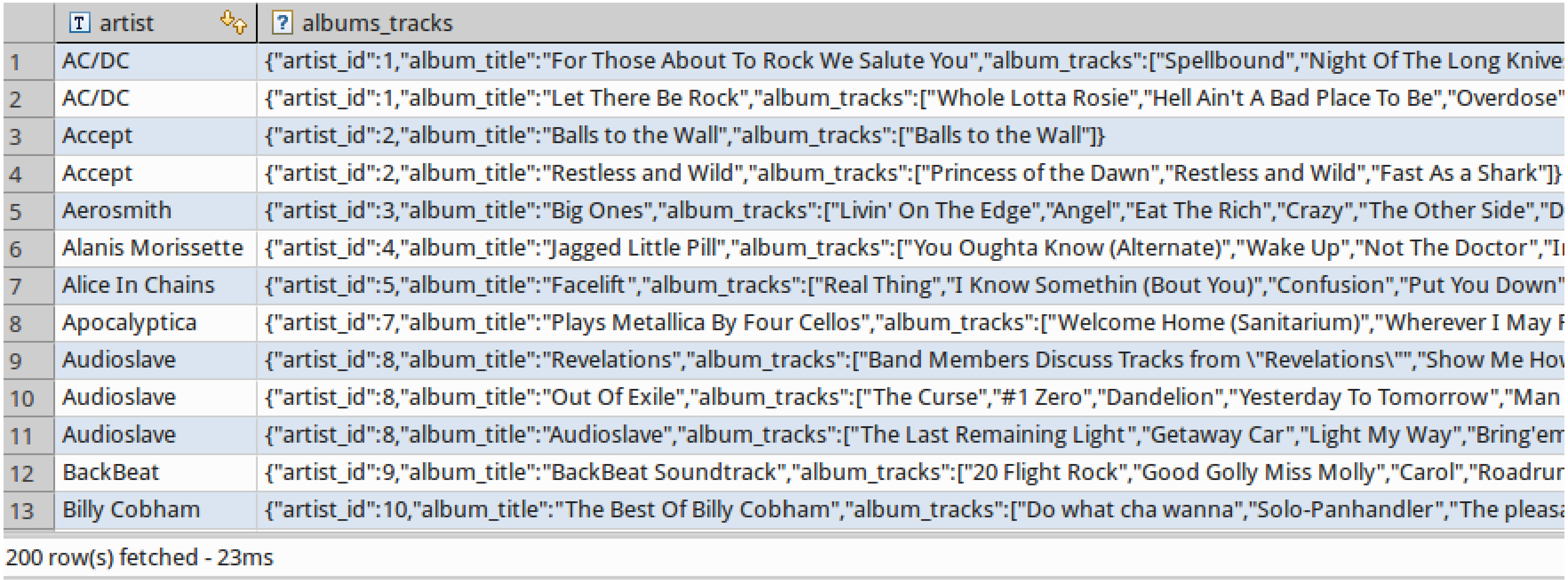
	album_tracks
1	{"artist_id":133,"album_title":"In Step","album_tracks":["Riviera Paradise","Love Me Darlin'","Scratch-N-Sniff","Wall Of Denial","Travis W
2	{"artist_id":83,"album_title":"Deixa Entrar","album_tracks":["Desaforo","Minha Gata","Medo De Escuro","Asas","Principiando/Decolag
3	{"artist_id":15,"album_title":"The 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' [Instrum
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

```
-- Step 2 Abums including tracks with aritsts
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
, al.album_tracks AS albums_tracks
FROM sqlite_artist AS a
INNER JOIN js_albums AS al
ON a."ArtistId" = CAST(al.album_tracks->>'artist_id' AS INT)
;
```

Live Data examples



	artist	albums_tracks
1	AC/DC	{"artist_id":1,"album_title":"For Those About To Rock We Salute You","album_tracks":["Spellbound","Night Of The Long Knive
2	AC/DC	{"artist_id":1,"album_title":"Let There Be Rock","album_tracks":["Whole Lotta Rosie","Hell Ain't A Bad Place To Be","Overdose"
3	Accept	{"artist_id":2,"album_title":"Balls to the Wall","album_tracks":["Balls to the Wall"]}
4	Accept	{"artist_id":2,"album_title":"Restless and Wild","album_tracks":["Princess of the Dawn","Restless and Wild","Fast As a Shark"]}
5	Aerosmith	{"artist_id":3,"album_title":"Big Ones","album_tracks":["Livin' On The Edge","Angel","Eat The Rich","Crazy","The Other Side","D
6	Alanis Morissette	{"artist_id":4,"album_title":"Jagged Little Pill","album_tracks":["You Oughta Know (Alternate)","Wake Up","Not The Doctor","I
7	Alice In Chains	{"artist_id":5,"album_title":"Facelift","album_tracks":["Real Thing","I Know Somethin (Bout You)","Confusion","Put You Down"
8	Apocalyptica	{"artist_id":7,"album_title":"Plays Metallica By Four Cellos","album_tracks":["Welcome Home (Sanitarium)","Wherever I May F
9	Audioslave	{"artist_id":8,"album_title":"Revelations","album_tracks":["Band Members Discuss Tracks from \"Revelations\"","Show Me Ho
10	Audioslave	{"artist_id":8,"album_title":"Out Of Exile","album_tracks":["The Curse","#1 Zero","Dandelion","Yesterday To Tomorrow","Man
11	Audioslave	{"artist_id":8,"album_title":"Audioslave","album_tracks":["The Last Remaining Light","Getaway Car","Light My Way","Bring'em
12	BackBeat	{"artist_id":9,"album_title":"BackBeat Soundtrack","album_tracks":["20 Flight Rock","Good Golly Miss Molly","Carol","Roadrur
13	Billy Cobham	{"artist_id":10,"album_title":"The Best Of Billy Cobham","album_tracks":["Do what cha wanna","Solo-Panhandler","The pleas

200 row(s) fetched - 23ms

Live Data examples

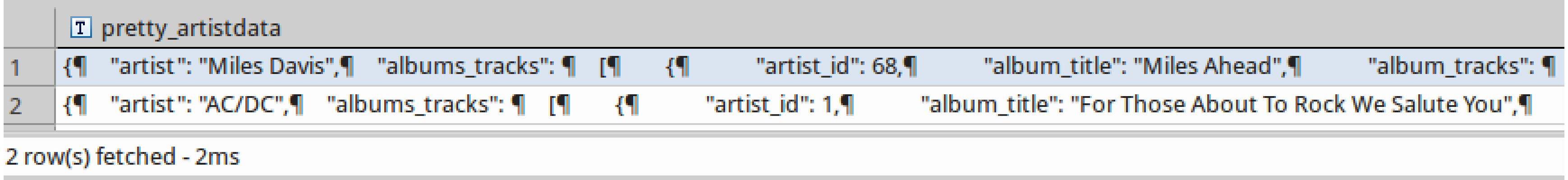
```
-- Step 3 Return one row for an artist with all albums
CREATE MATERIALIZED VIEW v_artist_data AS
WITH albums AS
(
SELECT a."ArtistId" AS artist_id
, a."Title" AS album_title
, array_agg(t."Name") AS album_tracks
FROM mysql_album AS a
INNER JOIN "Track" AS t
ON a."AlbumId" = t."AlbumId"
GROUP BY a."ArtistId"
, a."Title"
)
, js_albums AS
(
SELECT row_to_json(albums) AS album_tracks
FROM albums
)
, artist_albums AS
(
SELECT a."Name" AS artist
, array_agg(al.album_tracks) AS albums_tracks
FROM sqlite_artist AS a
INNER JOIN js_albums AS al
ON a."ArtistId" = CAST(al.album_tracks->>'artist_id' AS INT)
GROUP BY a."Name"
)
SELECT CAST(row_to_json(artist_albums) AS JSONB) AS artist_data
FROM artist_albums
;
```

Live Data examples

```
CREATE MATERIALIZED VIEW mv_artist_data AS
SELECT *
FROM v_artist_data
;

CREATE INDEX artist_data_gin ON mv_artist_data USING
GIN(artist_data);
```

Live Data examples

```
-- SELECT data from that materialized view, that does
querying
-- PostgreSQL, MariaDB, and SQLite tables in one SQL
statement
SELECT jsonb_pretty(artist_data) pretty_artistdata
FROM mv_artist_data
WHERE artist_data->>'artist' IN ('Miles Davis', 'AC/DC')
;
```



	pretty_artistdata
1	{¶ "artist": "Miles Davis",¶ "albums_tracks": ¶ [¶ {¶ "artist_id": 68,¶ "album_title": "Miles Ahead",¶ "album_tracks": ¶
2	{¶ "artist": "AC/DC",¶ "albums_tracks": ¶ [¶ {¶ "artist_id": 1,¶ "album_title": "For Those About To Rock We Salute You",¶

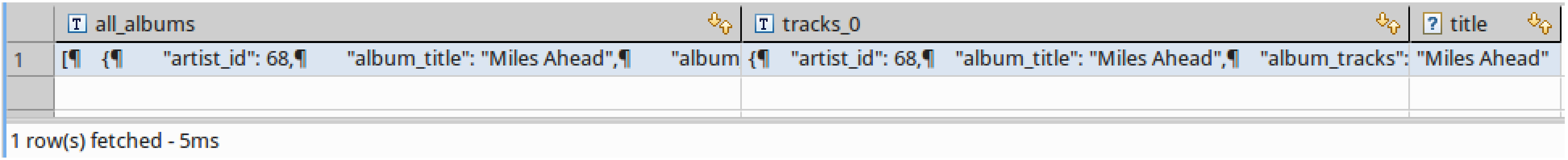
2 row(s) fetched - 2ms

Live Data examples

```
{
    "artist": "Miles Davis",
    "albums_tracks":
    [
        {
            "artist_id": 68,
            "album_title": "Miles Ahead",
            "album_tracks":
            [
                "I Don't Wanna Be Kissed (By Anyone But You) (Alternate Take)",
                "The Meaning Of The Blues/Lament (Alternate Take)",
                "Blues For Pablo (Alternate Take)",
                "Springsville (Alternate Take)",
                ...
            ]
        },
        {
        ...
        }
    ]
}
```

Live Data examples

```
-- SELECT some data using JSON methods
SELECT jsonb_pretty(artist_data#>'{albums_tracks}') AS all_albums
, jsonb_pretty(artist_data#>'{albums_tracks, 0}') AS tracks_0
, artist_data#>'{albums_tracks, 0, album_title}' AS title
FROM mv_artist_data
WHERE artist_data->'albums_tracks' @> '[{"album_title":"Miles Ahead"}]'
;
```



	all_albums	tracks_0	title
1	[¶ {¶ "artist_id": 68,¶ "album_title": "Miles Ahead",¶ "album	{¶ "artist_id": 68,¶ "album_title": "Miles Ahead",¶ "album_tracks":	"Miles Ahead"

1 row(s) fetched - 5ms

User Group / Conferences

PostgreSQL User Group Berlin

German PostgreSQL Conference 2015
Hamburg, November 26-27, 2015

FOSDEM PGDay 2016
Brussels, January 29, 2016

FOSDEM 2016, PostgreSQL devroom
Brussels, January 30-31, 2016

Link List

PGXN Extensions:
- mysql_fdw, MySQL/MariaDB FDW
- sqlite_fdw, SQLite FDW
- jsonbx, PostgreSQL 9.5 JSONB features

Slide and source on Github:
https://github.com/sjstoelting/talks

e Database To Rule 'em All