



JSON in PostgreSQL FAQ

Boriss Mejias
Holistic System Software Engineer
Air Guitar Player





JSON in PostgreSQL FAQ

Boriss Mejias
Senior Solution Architect
Air Guitar Player



Why on Earth would I want to use JSON ?

Why in the Universe would I want to use JSON ?

Schema Freedom

documents						
id	name	owner	creation	modified	size	type

id	name	owner	creation	modified	size	type	h	w	lat	long	fps	length	audio
1	file1	user1	2023-01-01	2023-01-01	1000	image	500	300	40.7128	-74.0060	30	10.5	mp4
2	file2	user1	2023-01-02	2023-01-02	1200	video	600	400	40.7128	-74.0060	25	15.0	mp4
3	file3	user2	2023-01-03	2023-01-03	800	image	450	300	40.7128	-74.0060	20	8.0	mp4
4	file4	user2	2023-01-04	2023-01-04	900	video	550	350	40.7128	-74.0060	25	12.0	mp4
5	file5	user3	2023-01-05	2023-01-05	700	image	400	300	40.7128	-74.0060	15	7.0	mp4
6	file6	user3	2023-01-06	2023-01-06	600	video	350	250	40.7128	-74.0060	20	6.0	mp4
7	file7	user4	2023-01-07	2023-01-07	500	image	300	200	40.7128	-74.0060	10	5.0	mp4
8	file8	user4	2023-01-08	2023-01-08	400	video	250	150	40.7128	-74.0060	15	4.0	mp4
9	file9	user5	2023-01-09	2023-01-09	300	image	200	100	40.7128	-74.0060	8	3.0	mp4
10	file10	user5	2023-01-10	2023-01-10	200	video	150	100	40.7128	-74.0060	12	2.0	mp4

Attributes

documents

id	name	owner	creation	modified	size	type
----	------	-------	----------	----------	------	------

green oval						
------------	------------	------------	------------	------------	------------	------------

green oval						
------------	------------	------------	------------	------------	------------	------------

green oval						
------------	------------	------------	------------	------------	------------	------------

green oval						
------------	------------	------------	------------	------------	------------	------------

green oval						
------------	------------	------------	------------	------------	------------	------------

COKE

attributes

JSON

green oval

documents							attributes		
id	name	owner	creation	modified	size	type			
●	●	●	●	●	●	●	●	●	
●	●	●	●	●	●	●	●	●	
●	●	●	●	●	●	●	●	●	
●	●	●	●	●	●	●	●	●	
●	●	●	●	●	●	●	●	●	

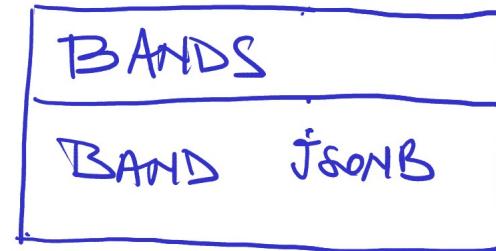
CORE



Bare Metal

Metal bands

```
{  
  "id": 37161,  
  "name": "Avatar",  
  "genre": ["Melodic Death Metal (early)",  
            "Nu-Metal (later)"],  
  
  "theme": ["Emotions", "StarCraft"],  
  "active": "2001 (as Lost Soul) | 2001-present",  
  "status": "Active",  
  "country": "Sweden",  
  "formed_in": "2001"  
}
```



Metal bands with a unique id

```
CREATE TABLE bands (
    band jsonb
);
```

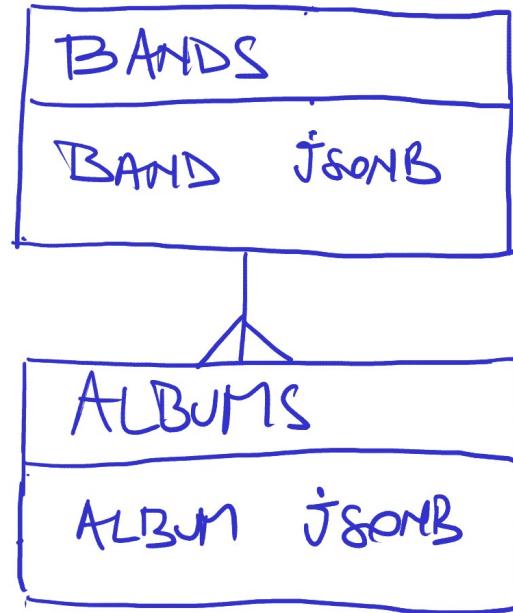
```
CREATE UNIQUE INDEX metal_band_idx
ON bands ((band->>'id'));
```

Metal albums

```
{  
    "id": 26714,  
    "band": 37161,  
    "title": "Black Waltz",  
    "year": 2012  
}  
{  
    "id": 26715,  
    "band": 37161,  
    "title": "Hail the Apocalypse",  
    "year": 2014  
}  
{  
    "id": 26716,  
    "band": 37161,  
    "title": "Feathers & Flesh",  
    "year": 2016  
}
```

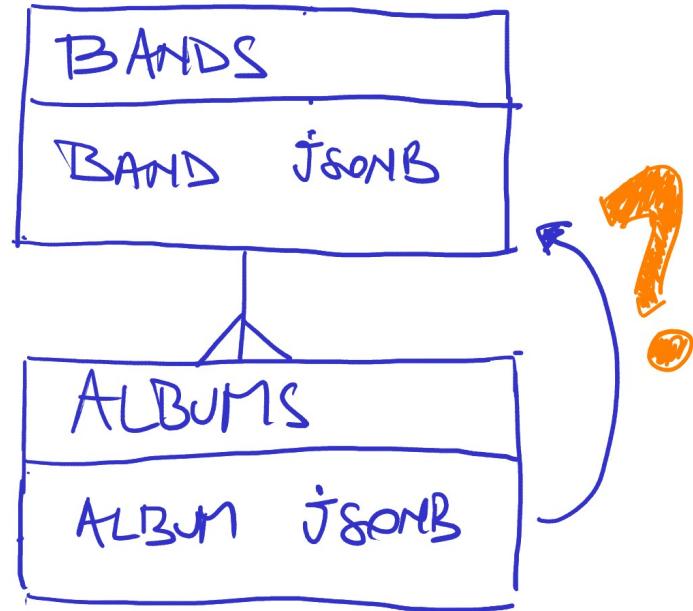
Metal albums

```
{  
    "id": 26714,  
    "band": 37161,  
    "title": "Black Waltz",  
    "year": 2012  
}  
{  
    "id": 26715,  
    "band": 37161,  
    "title": "Hail the Apocalypse",  
    "year": 2014  
}  
{  
    "id": 26716,  
    "band": 37161,  
    "title": "Feathers & Flesh",  
    "year": 2016  
}
```



Metal albums

```
{  
    "id": 26714,  
    "band": 37161,  
    "title": "Black Waltz",  
    "year": 2012  
}  
{  
    "id": 26715,  
    "band": 37161,  
    "title": "Hail the Apocalypse",  
    "year": 2014  
}  
{  
    "id": 26716,  
    "band": 37161,  
    "title": "Feathers & Flesh",  
    "year": 2016  
}
```



Metal bands with a primary key

```
CREATE TABLE bands (
    id      bigint PRIMARY KEY
    , band   jsonb
);
```

```
CREATE TABLE albums (
    id      bigint PRIMARY KEY
    , band   bigint REFERENCES bands(id),
    , title  text,
    , year   integer
);
```

Metal bands with a primary key

```
CREATE TABLE bands (
    id      bigint PRIMARY KEY
    , band   jsonb
);
```

```
CREATE TABLE albums (
    id      bigint PRIMARY KEY
    , band   bigint REFERENCES bands(id),
    , title  text,
    , year   integer
);
```



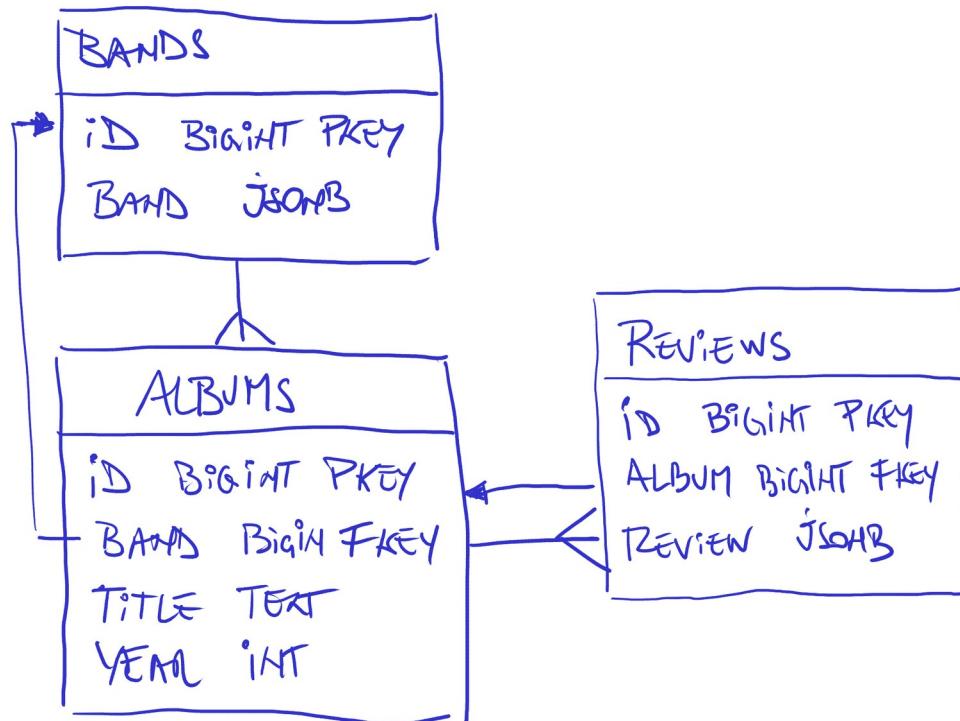
Metal albums reviews

```
{  
  "id": 18956,  
  "album": 26715,  
  "title": "Pushes the envelope further".  
  "score": 0.9  
  "content": "Sweden's rising force| Avatar..."  
}
```

Metal albums reviews

```
CREATE TABLE reviews (
    id      bigint PRIMARY KEY
    , album  bigint REFERENCES albums(id)
    , review jsonb
);
```

Bare Metal



JSON in Postgres

JSONB in Postgres

Access to JSON/JSONB

object → field ⇒ object

object ➤ field ⇒ text

object @> object ⇒ boolean

Using a Path

object #> array of text

object #>> array of text

Using a Path

object #>'{bands, 1}'

object #>>'{bands, 1}'

Updating JSON

Updating JSONB

Add and remove fields

//

-

**Replace fields with
jsonb_set(jsonb, path, new value)**

**Replace fields with
jsonb_set(jsonb, path, new value, create)**

What about search?
Can they run faster?

What about search?
Can they run faster?
Can I index JSON fields?

What about search?
Can they run faster?
Can I index JSONB fields?

Indexing JSONB fields

Single values with B-Trees

```
create index idx_bands_name_btree
on bands ((band->>'name' ));
```

Composite values with GIN (JSONB, arrays)

```
create index idx_bands_genres_gin
on bands
using gin ((band->'genre'));
```

For text use pg_trgm

```
create index idx_album_review_trgm
on reviews
using gin
((review->>'content') gin_trgm_ops);
```

Can I force a data type inside a JSON field?

Can I force a data type inside a JSONB field?

Oh Yeah!

\m/

```
create index idx_review_score_double  
on reviews  
(((review->>'score')::double precision));
```

Closing Words

Conclusion

- Metal bands also sing about Unicorns

JSON in Postgres

Awesome support for JSON
Schema-less

JSON in Postgres

Awesome support for JSON
Schema Freedom

JSON in Postgres

Awesome support for JSON

Schema Freedom

Use JSONB to manipulate the fields

Use indexes according to the queries

B-trees, GIN, pg_trgm



Thank you! Hail Slonik

