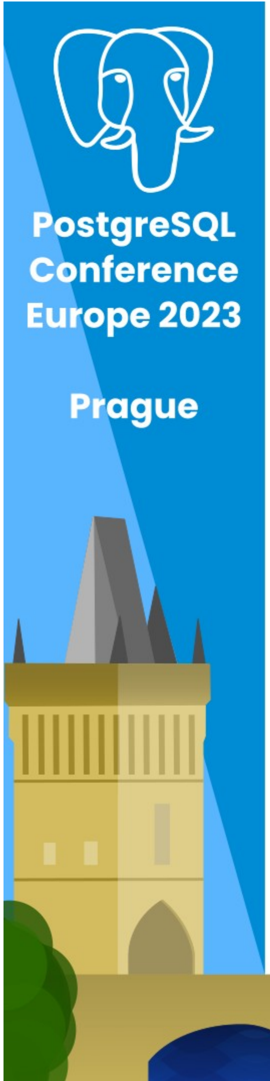
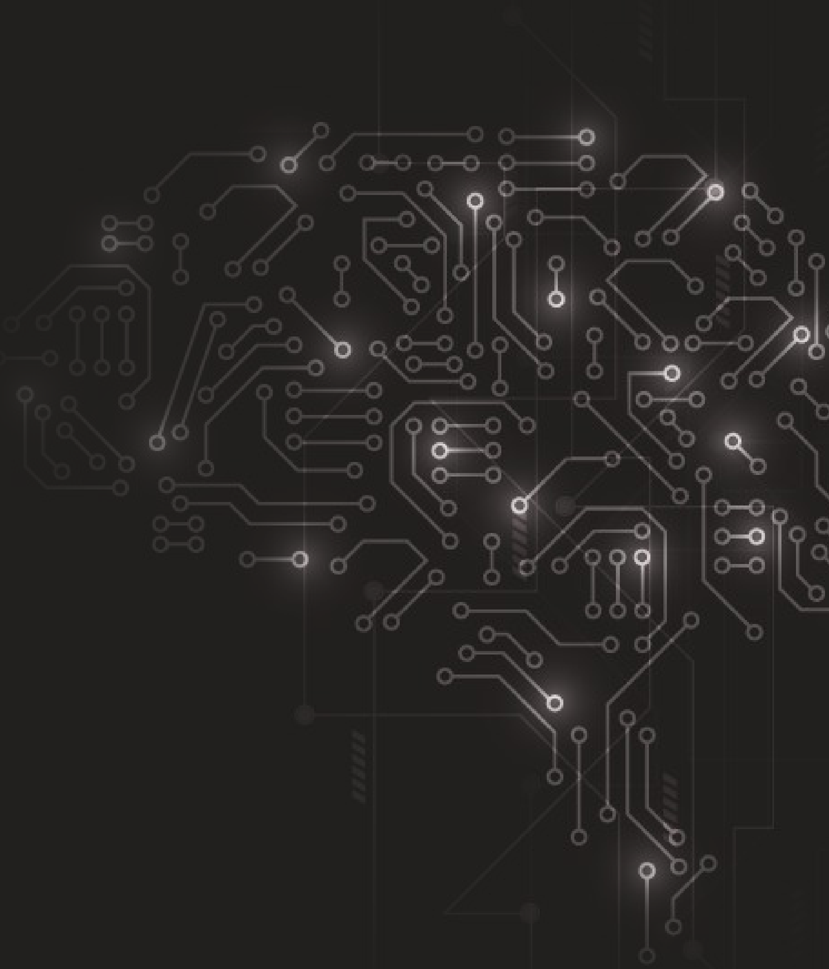


Should I use JSON in PostgreSQL?

Boriss Mejías

Should I use JSON in PostgreSQL?

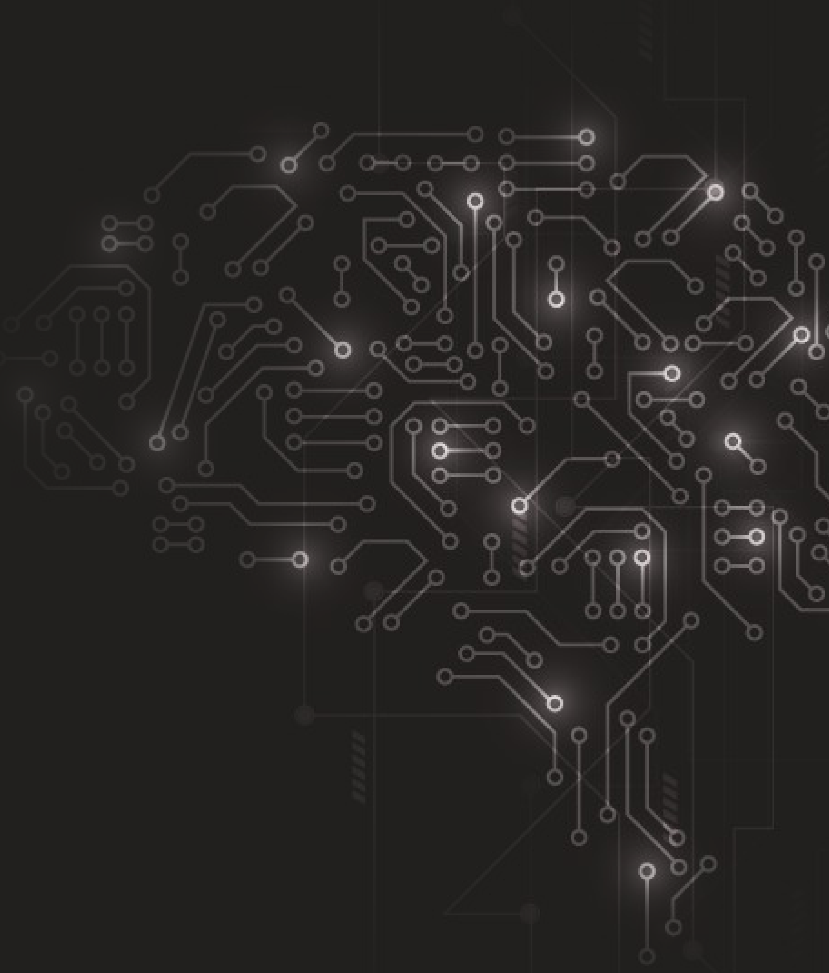
Boriss Mejías
Senior Solutions Architect
Air Guitar Player



Should I use JSON in PostgreSQL?

A therapeutic session

Boriss Mejías
Holistic System Software Engineer
Air Guitar Player



Why on Earth would you want to use JSON?

Schema Freedom

documents

id	name	owner	creation	modified	size	type

documents

id	name	owner	creation	modified	size	type	h	w

documents

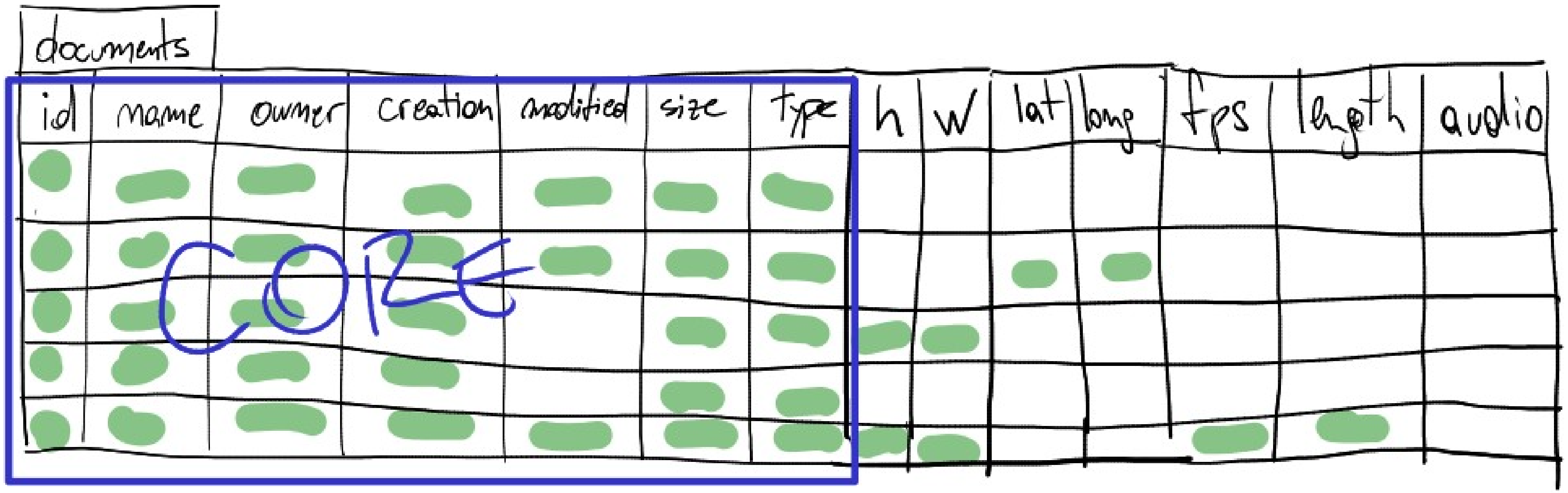
id	name	owner	creation	modified	size	type	h	w	lat	long

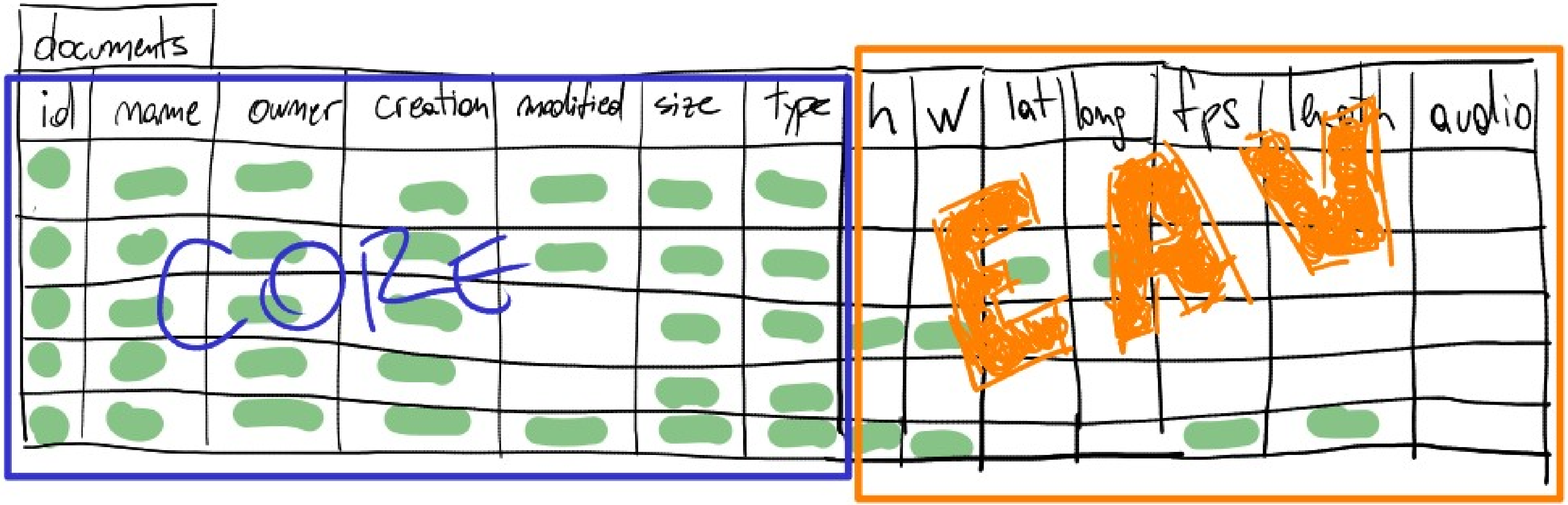
documents

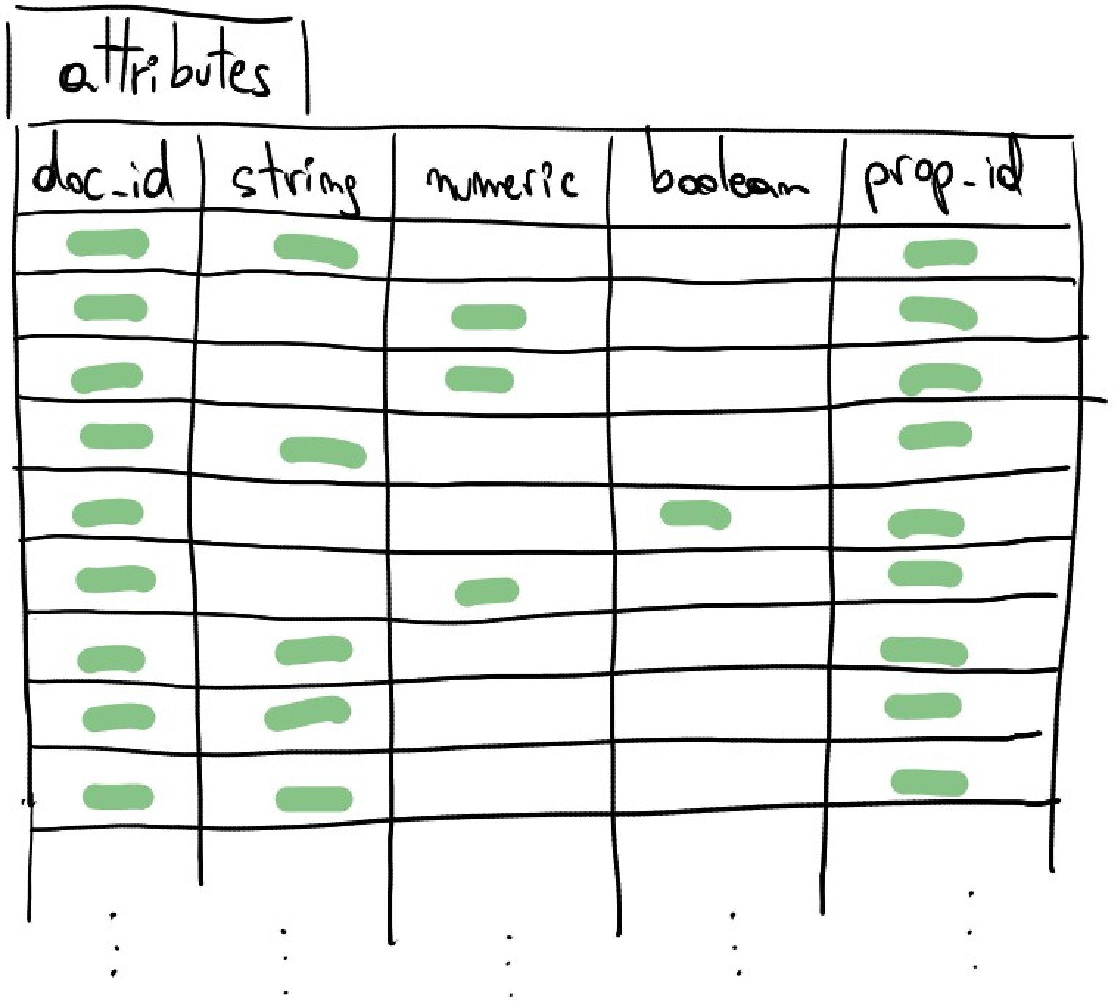
id	name	owner	creation	modified	size	type	h	w	lat	long	fps	length	audio

documents

id	name	owner	creation	modified	size	type	h	w	lat	long	fps	length	audio
🟩	🟩	🟩	🟩	🟩	🟩	🟩							
🟩	🟩	🟩	🟩	🟩	🟩	🟩			🟩	🟩			
🟩	🟩	🟩	🟩		🟩	🟩	🟩	🟩					
🟩	🟩	🟩	🟩		🟩	🟩							
🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩			🟩	🟩	

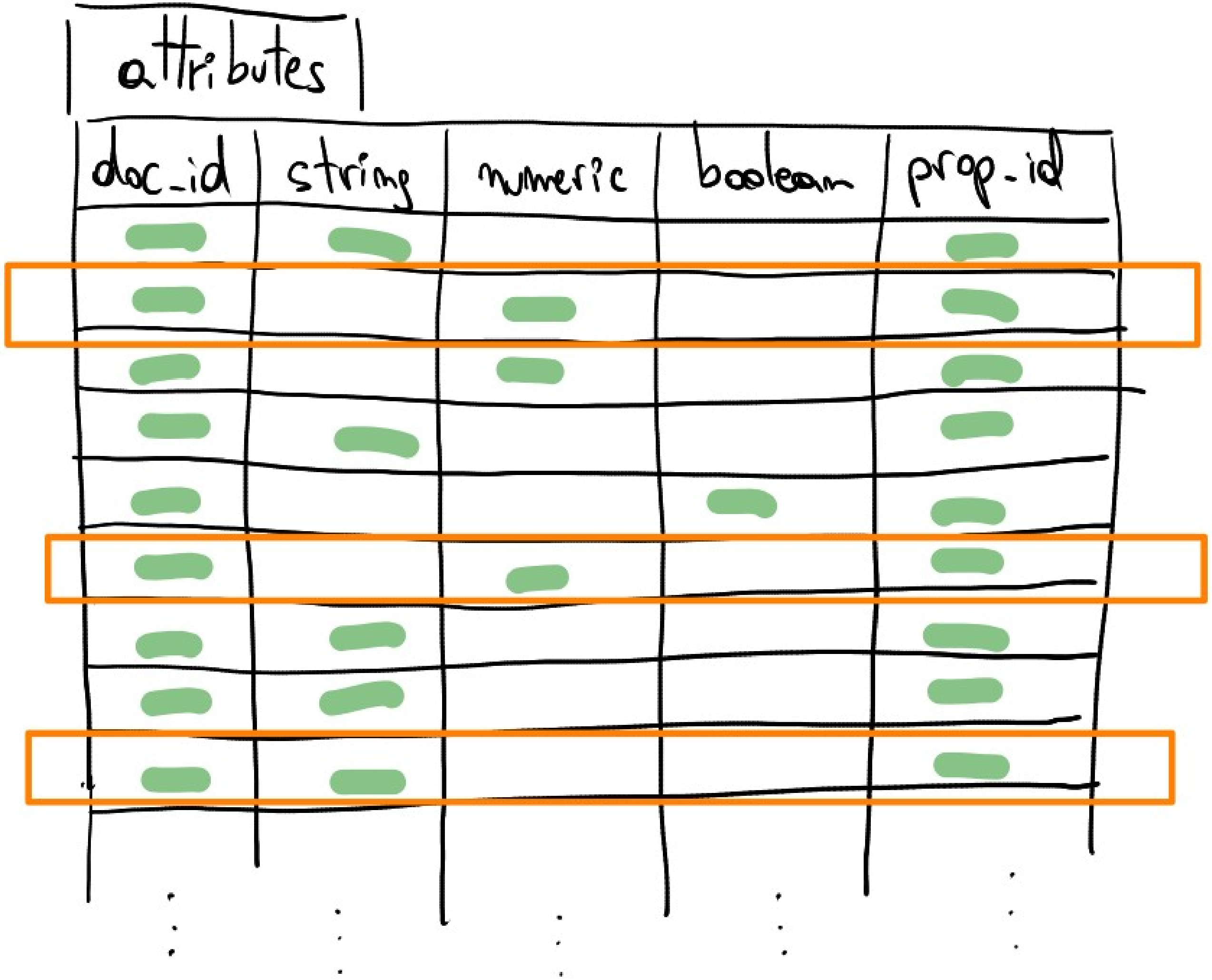

documents
id
name
owner
creation
modified
size
type
h
w
lat
long
fps
length
audio
CORE


documents
id
name
owner
creation
modified
size
type
h
w
lat
long
fps
length
audio
CORE
EAV



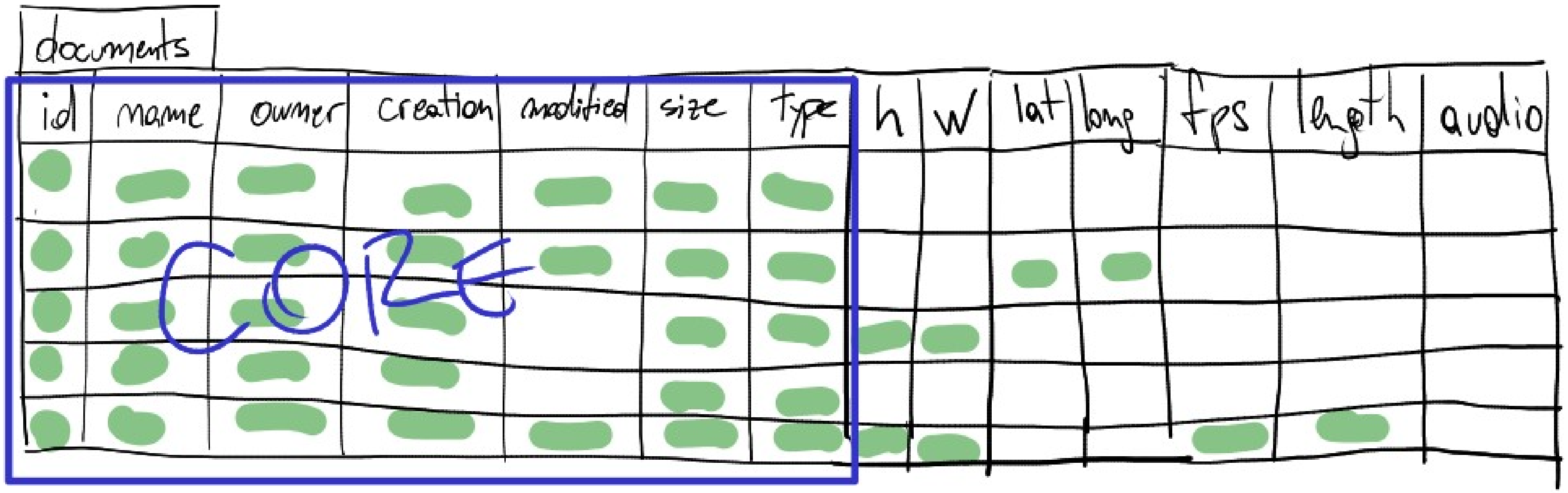
attributes

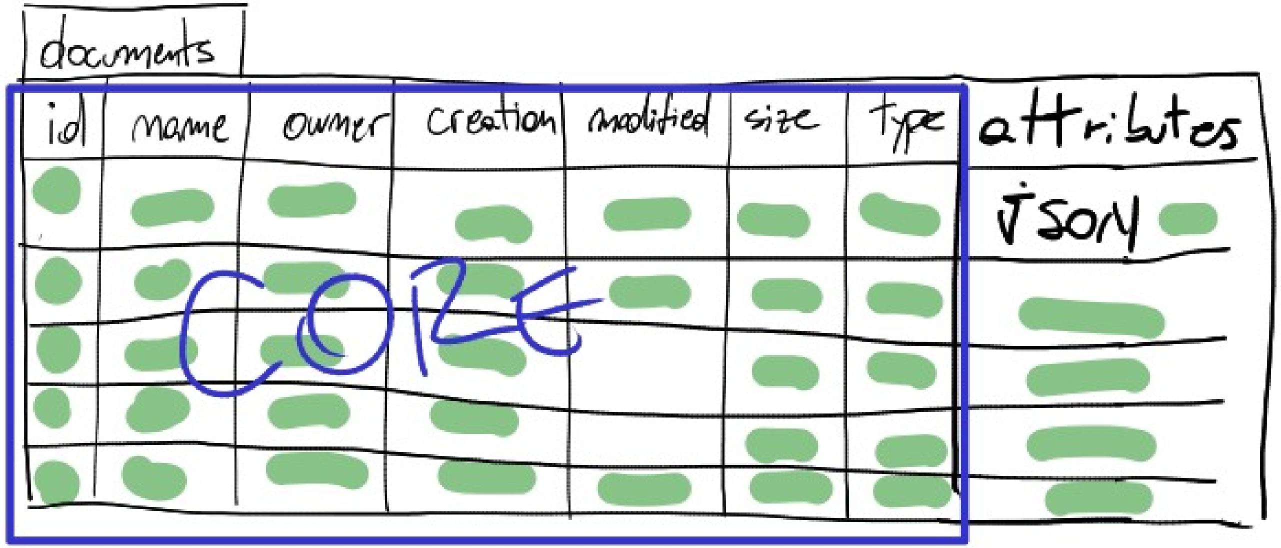
doc_id	string	numeric	boolean	prop_id
🟩	🟩			🟩
🟩		🟩		🟩
🟩		🟩		🟩
🟩	🟩			🟩
🟩			🟩	🟩
🟩		🟩		🟩
🟩	🟩			🟩
🟩	🟩			🟩
🟩	🟩			🟩
⋮	⋮	⋮	⋮	⋮

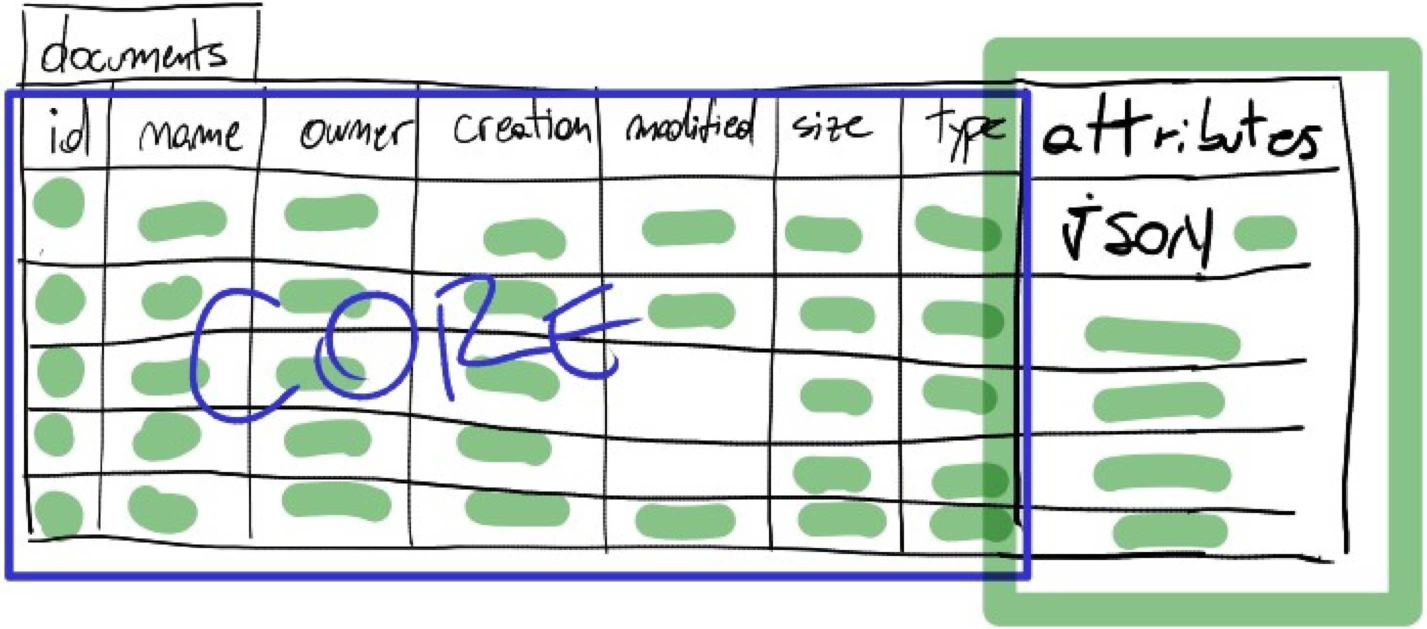


attributes

doc_id	string	numeric	boolean	prop_id
🟩	🟩			🟩
🟩		🟩		🟩
🟩		🟩		🟩
🟩	🟩			🟩
🟩			🟩	🟩
🟩		🟩		🟩
🟩	🟩			🟩
🟩	🟩			🟩
🟩	🟩			🟩
⋮	⋮	⋮	⋮	⋮


documents
id
name
owner
creation
modified
size
type
h
w
lat
long
fps
length
audio
CORE


documents
id
name
owner
creation
modified
size
type
attributes
json
CORE


documents
id
name
owner
creation
modified
size
type
attributes
json
CORE

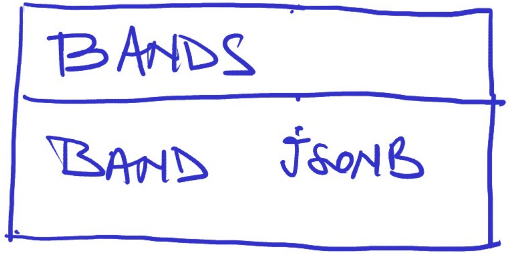
JSON

```
{ "height": 800 , "width": 600 }

{ "height": 2304 , "width": 4096 ,
  "latitude": -53,16004 , "Longitud": -70,91556 }

{ "container": "ogg" , "duration": "00:07:56",
  "tags" : [ "interview", "slonik", "postgres" ] }
```

Bare Metal
\m/

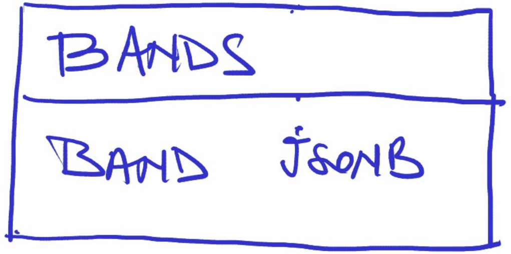
Metal bands

```
{
  "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"
}
```



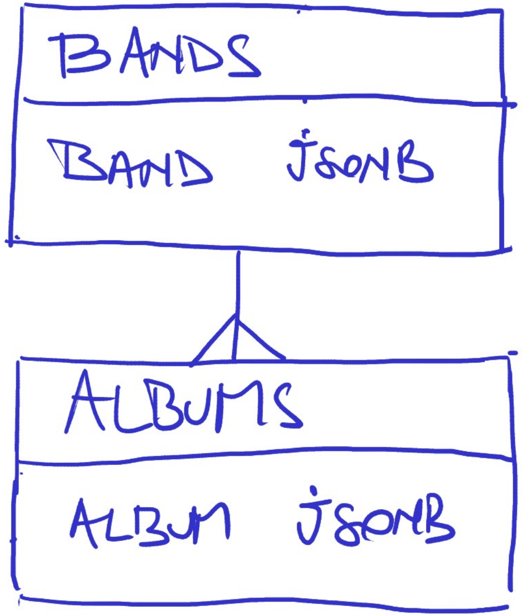
Metal bands with a unique Id

```
CREATE TABLE bands (
    band jsonb
);

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



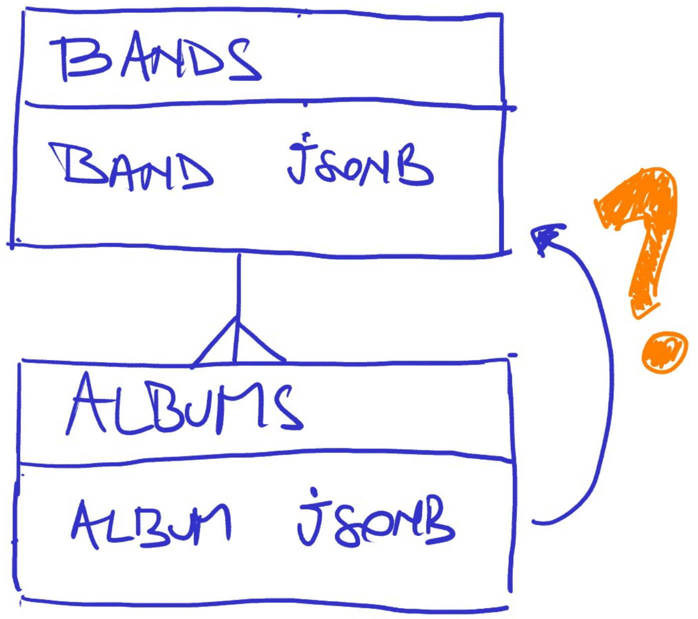
Metal albums

```
{
  "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
}
```



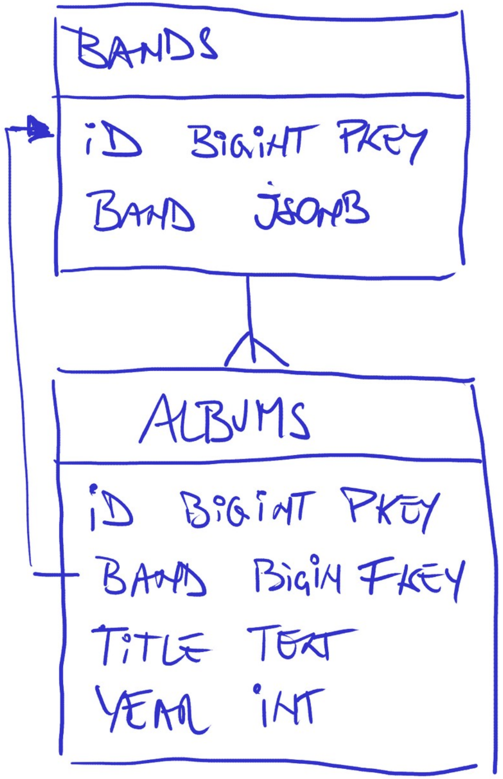
Metal albums

```
{
  "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 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
);
```



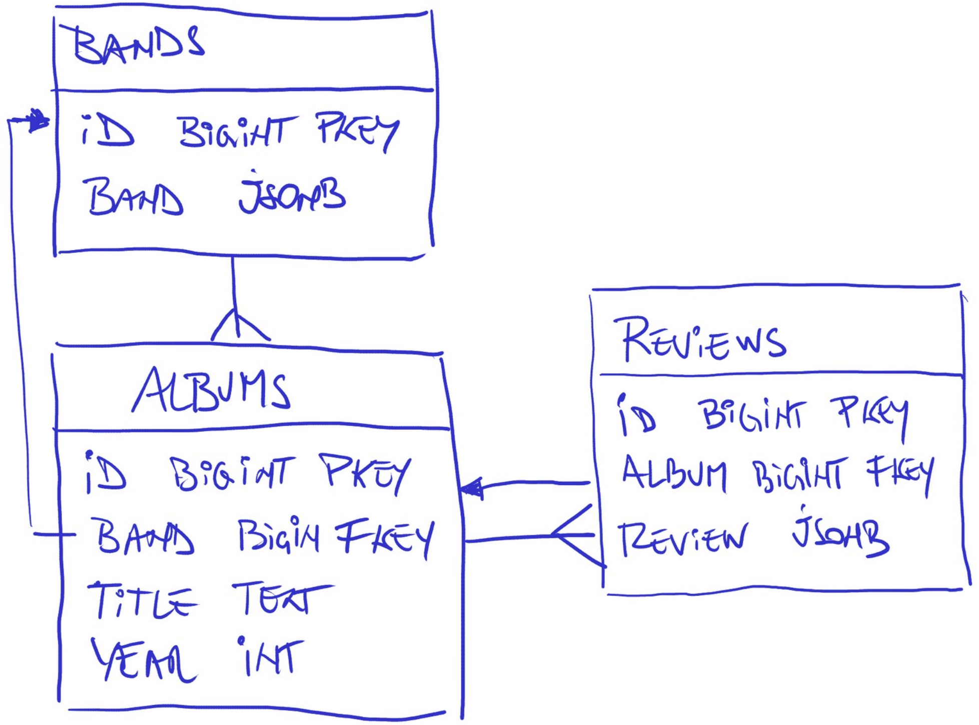
Metal album reviews

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

Metal album reviews

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

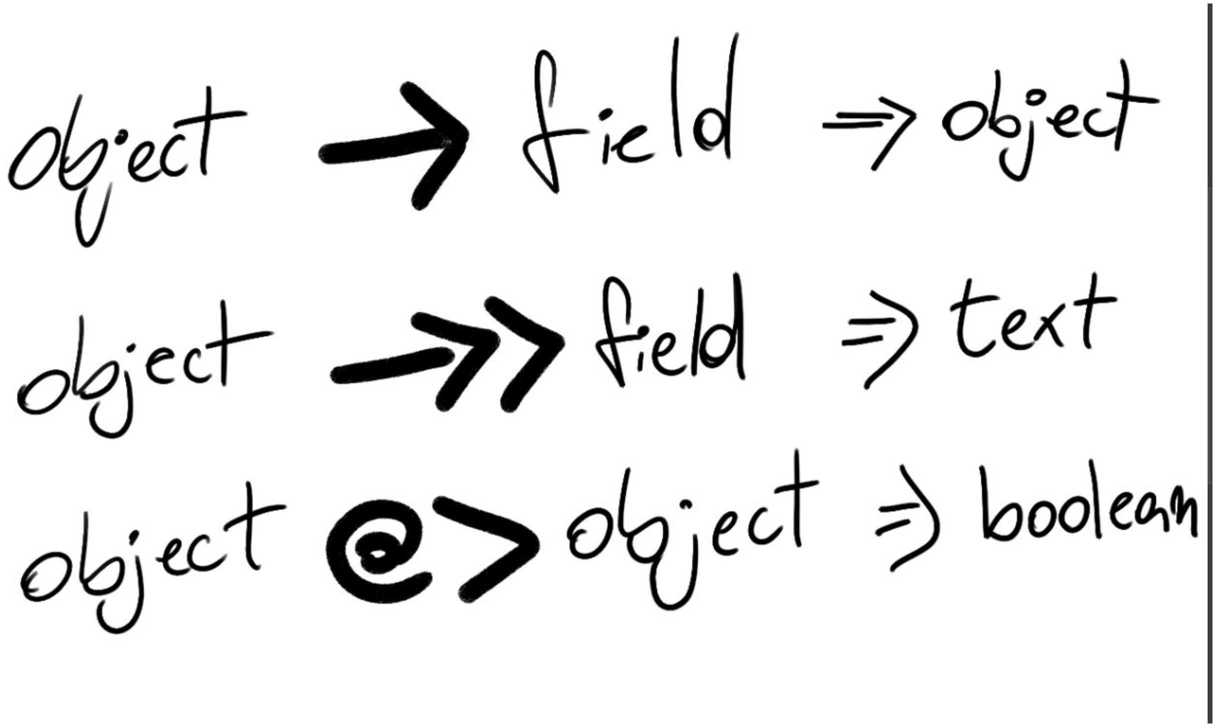
Bare Metal


BANDS
ID BIGINT PKEY
BAND JSONB
ALBUMS
ID BIGINT PKEY
BAND BIGINT FKEY
TITLE TEXT
YEAR INT
REVIEWS
ID BIGINT PKEY
ALBUM BIGINT FKEY
REVIEW JSONB

JSON in PostgreSQL

JSONB in PostgreSQL

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?
Is it fast?
Can I index JSON fields?

What about search?
Is it fast?
Can I index JSONB fields?

Indexing JSONB fields

Single values with B-Trees

```
CREATE INDEX idx_bands_name_btree
ON bands ((band->>'name'));
```

```sql
CREATE INDEX idx_bands_name_btree
ON bands ((band->>'name'));
```

This also works with JSON data type

Composite values with GIN (JSONB, arrays)

```
CREATE INDEX idx_bands_genres_gin
ON bands
USING GIN ((band->'genre'));
```

For text use pg_trgm

```
CREATE EXTENSION pg_trgm
WITH SCHEMA metal;

CREATE INDEX idx_album_review_trgm
ON reviews
USING GIN
((review->>'content') gin_trgm_ops);
```

Can you force a data type inside a JSON field?

Can I force a data type inside a JSON field?

Can I force a data type inside a JSONB field?

JSON or JSONB

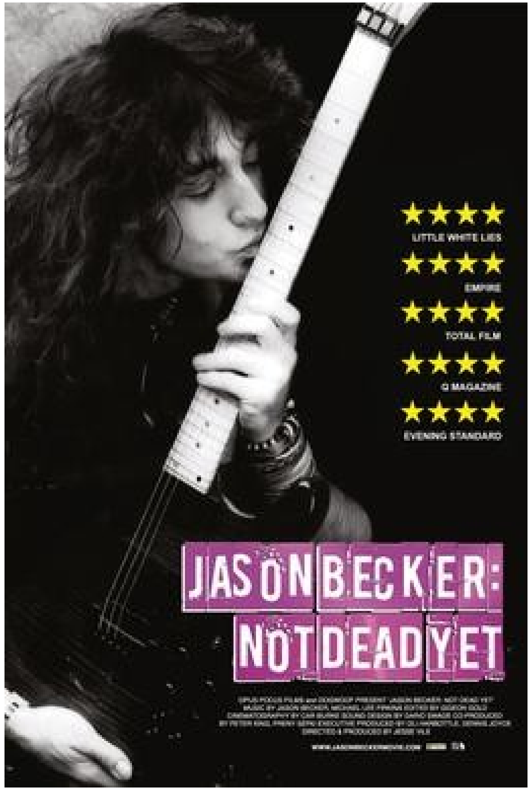


Image from Wikipedia: Jason Becker

Can I force a data type inside a JSONB field?

Index with a type cast

```
CREATE INDEX idx_review_score_double
ON reviews
(((review->>'score')::double precision));
```

Just add a constraint

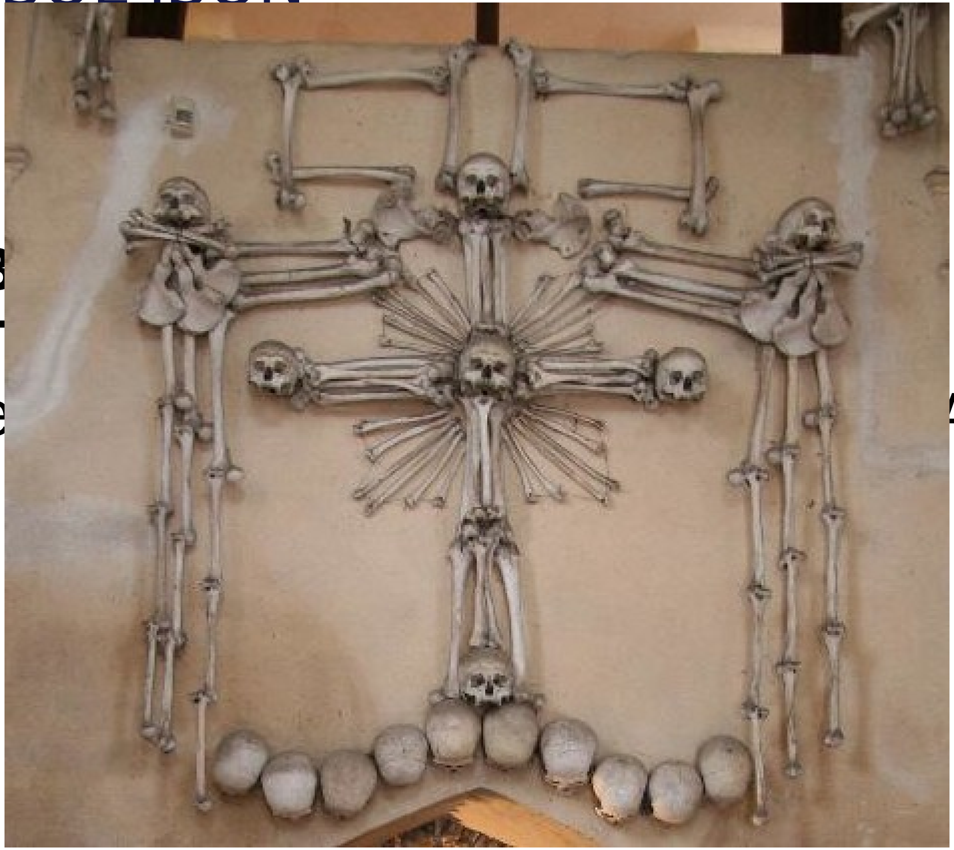
```
ALTER TABLE reviews
ADD CONSTRAINT numeric_score
CHECK (pg_input_is_valid(review->>'score',
                          'double precision'));
```

What about SQL JSON

```
ALTER TABLE reviews
ADD CONSTRAINT numeric_score
CHECK (review->'score' IS JSON SCALAR);
```

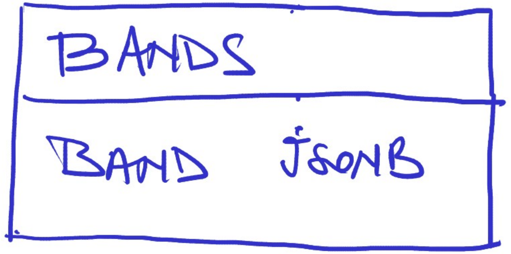
What about SQL JSON

```
ALTER TAB
ADD CONST
CHECK (re                          ALAR);
```



Metal bands

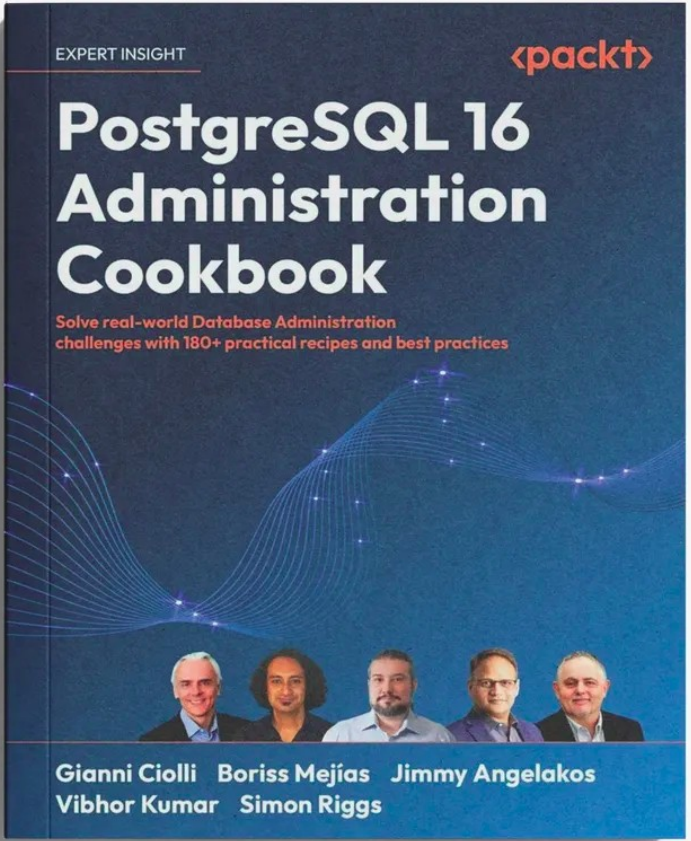
```
{
  "id": 37161,
  "name": "Avatar",
  "genre": ["Melodic Death Metal (early)",
                "Nu-Metal (later)"],

  "theme": ["Emotions", "StarCraft"],
  "active": "2001 (as Lost Soul)| 2001-present",
  "status": "Active",
  "country": "Sweden",
  "formed_in": "2001"
}
```



What about SQL JSON

```
ALTER TABLE bands
ADD CONSTRAINT array_of_genres
CHECK (band->'genre' IS JSON ARRAY);
```

There is more...



EXPERT INSIGHT

‹packt›

PostgreSQL 16 Administration Cookbook

Solve real-world Database Administration challenges with 180+ practical recipes and best practices

Gianni Ciolli   Boriss Mejías   Jimmy Angelakos
Vibhor Kumar   Simon Riggs

Closing Words

Should I use JSON in PostgreSQL?

Awesome support for JSON
Schema Freedom
Use the JSONB data type
Use indexes according to the queries
B-trees, GIN, pg_trgm

Thank you!
Hail Slonik

Boriss Mejías
Holistic System Software Engineer
Air Guitar Player
boriss.mejias@enterprisedb.com

EDB™