



# JSON and Speed

Jonathan S. Katz  
CTO, VenueBook  
PGConf EU 2013

First

Then

## **8.14. JSON Type**

Then

## **9.15. JSON Functions and Operators**

# The Catch



# However



```
CREATE UNLOGGED TABLE jsons (  
    data json  
);  
  
INSERT INTO jsons  
SELECT  
    row_to_json(ROW(x, x + 2, x + 3))  
FROM generate_series(1,1000000) x;
```

# Is It Fast?

```
EXPLAIN ANALYZE
SELECT *
FROM jsons
WHERE
    json_extract_path_text(data, 'f2') = '311';
```

Seq Scan on jsons

Filter: (json\_extract\_path\_text(data, VARIADIC '{f2}'::text[]) =  
'311'::text)

Rows Removed by Filter: 999999





Total runtime: **970.356 ms**

```
CREATE INDEX jsons_data_idx ON jsons
  (json_extract_path_text(data, 'f2'));
```

```
EXPLAIN ANALYZE
```

```
SELECT *
```

```
FROM jsons
```

```
WHERE
```

```
  json_extract_path_text(data, 'f2') = '311';
```

```
Bitmap Heap Scan on jsons
```

```
  Recheck Cond: (json_extract_path_text(data, VARIADIC
    '{f2}'::text[]) = '311'::text)
```

```
  -> Bitmap Index Scan on jsons_data_idx
```

```
    Index Cond: (json_extract_path_text(data, VARIADIC
      '{f2}'::text[]) = '311'::text)
```



Total runtime: **0.126 ms**



# Thank You

- Jonathan S. Katz
- @jkatz05