

More JSONB querying



schinckel.net/2018/01/12/more-jsonb-querying/

Occasionally, I get emails from people regarding specific queries in Postgres, usually because I have blogged about JSONB querying before.

Today, I got one: rather than just reply, I thought I'd blog about how queries could be written to solve this problem.

Our table can be a single column with JSONB data for the purposes of this.

```
CREATE TABLE priority (data JSONB);
```

We also need a bit of data to query:

```

INSERT INTO priority (data) VALUES (
'{'
  "id": "02e32a14-904c-4153-a32b-fe8d1f1bbbe1",
  "entity": "activity",
  "fields": {
    "subject": [
      {"val": "Subject", "priority": 7}
    ]
  },
  "recordStatusType": "active"
}'), (
'{'
  "id": "b33498b2-32f6-4575-b2cd-9e9a1ae2059d",
  "entity": "activity",
  "fields": {
    "subject": [
      {"val": "Subject", "priority": 4}
    ]
  },
  "recordStatusType": "active"
}'), (
'{'
  "id": "a2d327d2-7668-4dc0-ae1d-d6144130e3ec",
  "entity": "activity",
  "fields": {
    "object": [],
    "subject": [
      {"val": "Object", "priority": 1},
      {"val": "Target", "priority": 7}
    ]
  }
}'), (
'{'
  "id": "3bc8b536-00af-4fc7-881e-b88b620ac436",
  "entity": "activity",
  "fields": {
    "object": [
      {"val": "Object", "priority": 9}
    ]
  }
}')
);

```

The problem requires selection of the data rows where priority is greater than 5.

I've extended the data provided: I'm not sure if there will be multiple "fields", but I assume so. I also assume that a match for *any* priority within a subject field will be required.

Lets start with a simpler version: get the records where the first `fields->subject` priority is greater than 5 (I'll return just the `id` , to make it simpler):

```

SELECT data->'id'
FROM priority
WHERE (data#>>'{fields,subject,0,priority}'):INTEGER > 5;

"02e32a14-904c-4153-a32b-fe8d1f1bbbe1"

```

This uses the `#>>` operator - which does a path lookup, and returns a string value, that we then cast to an integer for the comparison. Note that the path lookup differs from normal Postgres' array indexing, in that it uses 0 as the first index, rather than 1.

But, we want to query for all rows where *any* subject field has a priority greater than 5.

We'll want to use the `jsonb_array_elements` (which is the JSONB equivalent of `unnest`). We can use that to get the fields themselves:

```
SELECT jsonb_array_elements(data#>'{{fields,subject}}') FROM priority;
```

Note this uses the `#>` operator, because we still want JSONB data:

```
      jsonb_array_elements
-----
{"val": "Subject", "priority": 7}
{"val": "Subject", "priority": 4}
{"val": "Object", "priority": 1}
{"val": "Target", "priority": 7}
(4 rows)
```

We can get a bit further too:

```
SELECT jsonb_array_elements(data#>'{{fields,subject}}')->'priority' FROM priority;
```

Indeed, we can get all the way to our boolean test:

```
SELECT (jsonb_array_elements(data#>'{{fields,subject}}')->'priority')::INTEGER > 5
FROM priority;
```

```
 ?column?
-----
t
f
f
t
(4 rows)
```

But we want the data rows themselves, not just the matching subject field, and this is not that useful. So, we can use the fact that `jsonb_array_elements` returns a set, and use that as a subquery in our `WHERE` clause, using the `value operator ANY()` construct:

```
SELECT data->'id'
FROM priority
WHERE 5 < ANY(SELECT (jsonb_array_elements(data#>'{{fields,subject}}')-
>>'priority')::INTEGER)
```

This means that we want only the records where 5 is less than *any* of the priority values in subject fields.

```
      ?column?
-----
"02e32a14-904c-4153-a32b-fe8d1f1bbbe1"
"a2d327d2-7668-4dc0-ae1d-d6144130e3ec"
```

I hope this helps, Paulo!