Interactive - 47 - NoSQL databases (mongoDB) v.s. JSONb in PostreSQL

One of the most common alternatives to a SQL database is MongoDB. Front end developers like MongoDB because everting is in JSON.

There are two reasons why I don't use MongoDB.

1. Concurrency control is incredibly important. Google reviewed its code base and counted the number of defects related to bad implementations of custom concurrency control. The problems were bad enough that Google banned the use of MongoDB.
2. Everything that can be done in MogoDB can be done just as fast using JSONb data in PostgreSQL. Also with PostgreSQL you get a complete relational database with tables and indexes and spacial data and... That other stuff just get thrown in.

So... Let's look at how to use JSONb in PostgreSQL.

First let's create a table. This is the same as creating a "collection" in MongoDB. The table will use a GIN index to index all of the data in the JSON data. (Run file `hw47_1.sql`)

```
drop TABLE if exists test_collection ;

CREATE TABLE test_collection (
    id serial primary key not null,
    data JSONB
);

CREATE INDEX test_collection_gin_1 ON test_collection USING gin (data);

INSERT INTO test_collection ( data ) values
    ( '{"name":"bob"}' )
;
```

PostgreSQL has a slew of operators.

Operator	Right Operand Type	Description	Example	Example Result
`->`	int	Get JSON array element (indexed from zero, negative integers count from the end)	`'[{"a":"foo"},{"b":"bar"},{"c":"baz"}]'::json->2`	`{"c":"baz"}`
`->`	text	Get JSON object field by key	`'{"a": {"b":"foo"}}'::json->'a'`	`{"b":"foo"}`
`->>`	int	Get JSON array element as text	`'[1,2,3]'::json->>2`	`3`

Operator	Right Operand Type	Description	Example	Example Result
`->>`	text	Get JSON object field as text	`'{"a":1,"b":2}'::json->>'b'`	`2`
`#>`	text[]	Get JSON object at specified path	`'{"a": {"b":{"c": "foo"}}}'::json#>'{a,b}'`	`{"c": "foo"}`
`#>>`	text[]	Get JSON object at specified path as text	`'{"a":[1,2,3],"b":[4,5,6]}'::json#>>'{a,2}'`	`3`
`@>`	JSONb	Does the left JSON value contain the right JSON path/value entries at the top level?	`'{"a":1, "b":2}'::JSONb @> '{"b":2}'::JSONb`	
`<@`	JSONb	Are the left JSON path/value entries contained at the top level within the right JSON value?	`'{"b":2}'::JSONb <@ '{"a":1, "b":2}'::JSONb`	
`?`	text	Does the string exist as a top-level key within the JSON value?	`'{"a":1, "b":2}'::JSONb ? 'b'`	

Operator	Right Operand Type	Description	Example	Example Result
?xxVb()`` | text[]	Do any of these array strings exist as top-level keys?	‘{“a”:1, “b”:2, “c”:3}’::JSONb ?xxVb() array[‘b’, ‘c’]		
|?& | text[]	Do all of these array strings exist as top-level keys?	‘[“a”, “b”]’::JSONb ?& array[‘a’, ‘b’]		
|xxVb()xxVb()| JSONb	Concatenate two JSONb values into a new			

Operator	Right Operand Type	Description	Example	Example Result

For example:

```
SELECT *
    FROM test_collection
    WHERE data->>'name' = 'bob'
;
```

Should return 1 row the name equal to `bob`.

Tags: “JSONb”,“MongoDB”

Validate: SQL-Select,“select ‘PASS’ as x”

FilesToRun: hw47_1.sql