IPB Leibniz-Institut für Pflanzenbiochemie

Short overview of JSONB and indices in postgres

Fabian Mauz

WissenschaftsCampus Pflanzenbasierte Bioökonomie Halle

Json Data Types

The **json** data type stores an exact copy of the input text. Functions must reparse on each execution, no index

jsonb data is stored in a decomposed binary format. It also supports indices.

specified in RFC 7159

Advantage against text is enforcing that each stored value is valid according to the JSON rules

```
CREATE TABLE xxx (data jsonb);
```

Operator	Description
->	Get JSON object field by key or index
->>	Get object casted as text by key or index
#>	Get JSON object at specified path
@>	Does the left JSON value contain the right JSON path/value entries at the **top** level?
?	Does the string exist as a top-level key within the JSON value?

There are more operators and functions -> special ones are set generating functions -> they produce an own table

Indices in postgres

Aditional datastructure with 2 main purposes: contraining and query optimization

Default type in postgres : Btree, but there are more (GIN,GIST,…)

Indices are stored as tuples (value[s],pointer to row) and referenced in pages of size 8kB organized as an array with adinitional tree structure for faster access

Some contstraints automatically create indices (primary key, unique, exclude)

You can see the metainformation of the index with pageinspect

```
CREATE EXTENSION pageinspect
```

Balanced Trees (B-Tree)


```
CREATE INDEX btree_example ON metadata(id);
```

Based on Lehman & Yao Btree[1]

```
SELECT * FROM bt_metap('btree_example');
```

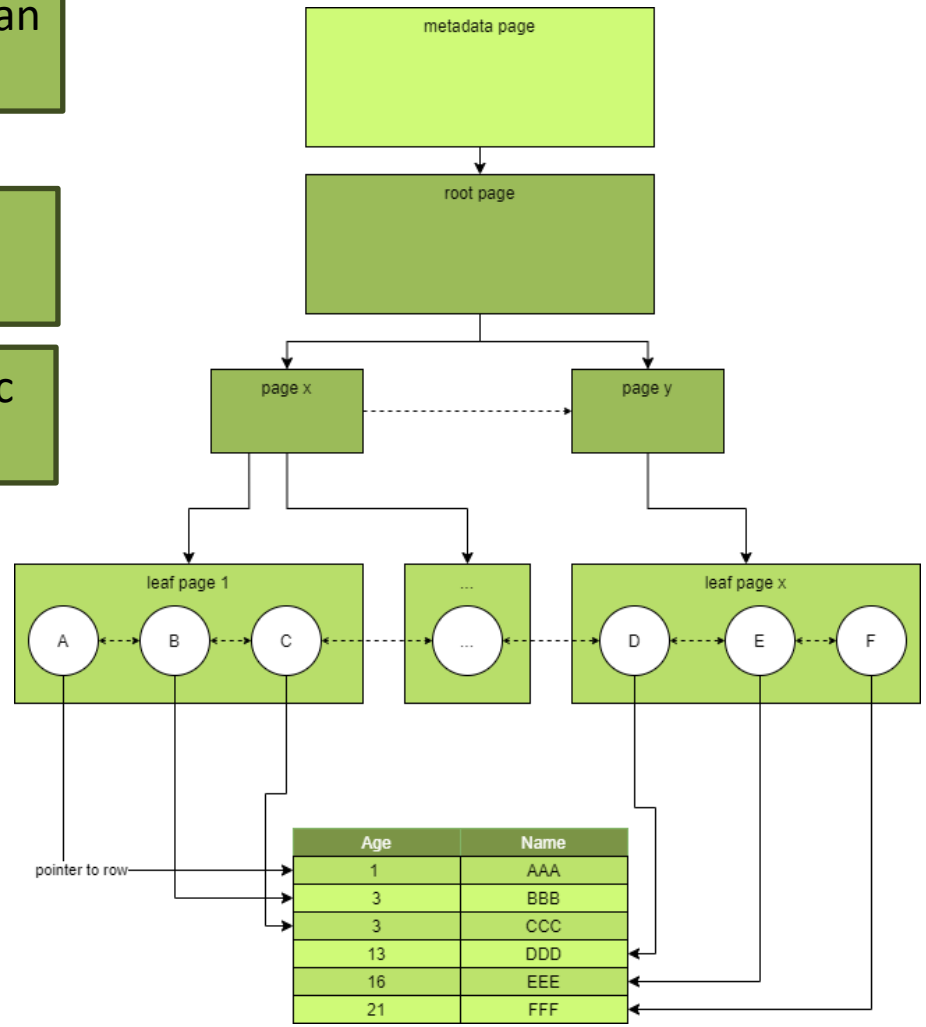
Meta infos of index

```
SELECT * FROM BT_PAGE_ITEMS('btree_example',1);
```

Infos of specific pages

If query is satisfied by values only from index, no access to “real” data

Supported operators: <,>,<=,>=,=



[1] *Efficient Locking for Concurrent Operations on B-Trees*, **P. Lehman, B. Yao**, 1981

Generalized Inverted Index (GIN)

Designed for cases where the items to be indexed are composite values

Inserting very slow -> You can disable an instant index update and do it later

```
Insert on metadata  (cost=0.00..0.01 rows=0 width=0) (actual time=0.067..0.069 rows=0 loops=1)
  ->  Result  (cost=0.00..0.01 rows=1 width=36) (actual time=0.002..0.004 rows=1 loops=1)
Planning Time: 0.034 ms
Execution Time: 0.095 ms
```

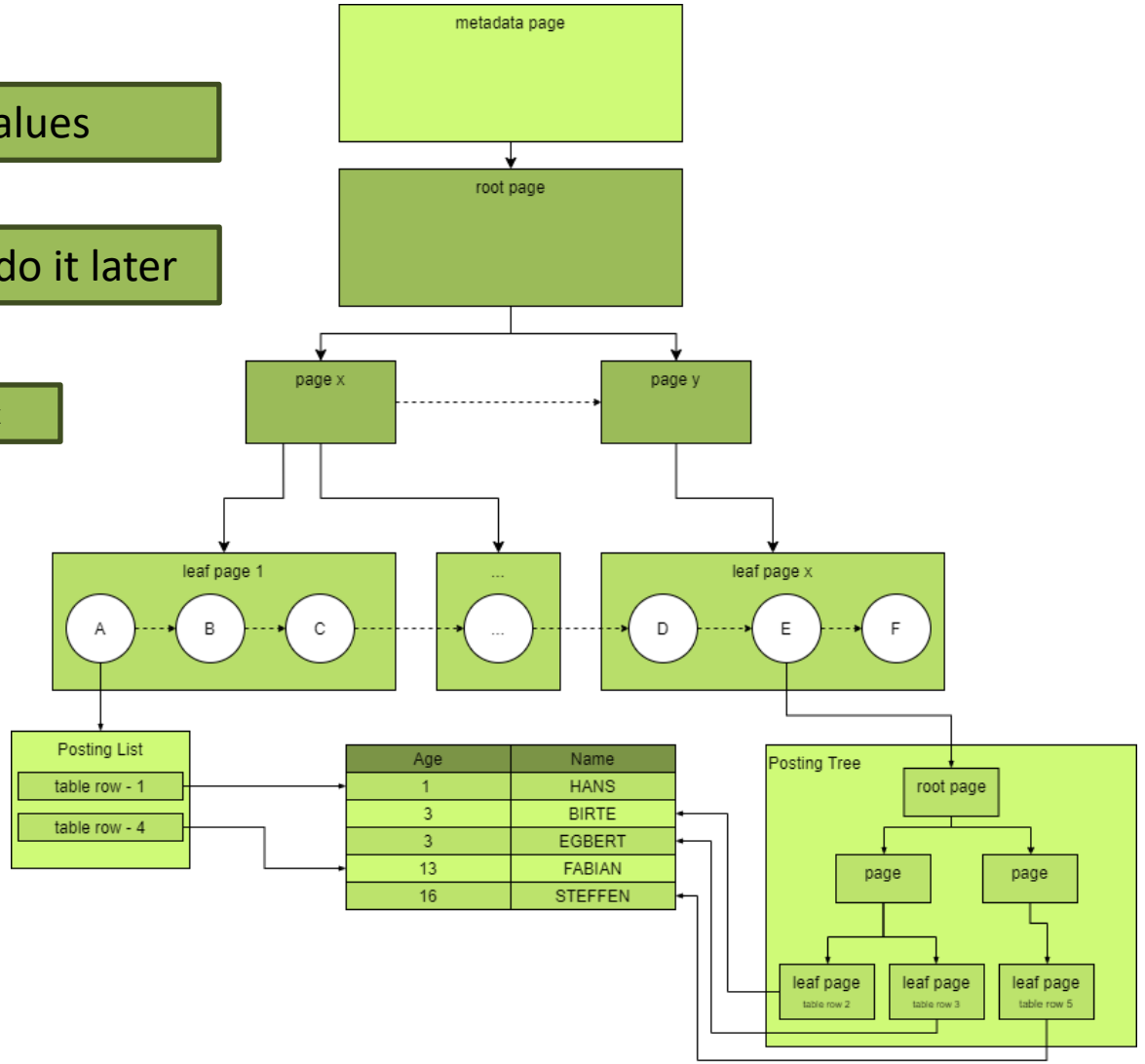
Without index

```
Insert on metadata  (cost=0.00..0.01 rows=0 width=0) (actual time=8.276..8.278 rows=0 loops=1)
  ->  Result  (cost=0.00..0.01 rows=1 width=36) (actual time=0.005..0.007 rows=1 loops=1)
Planning Time: 2.040 ms
Execution Time: 9.131 ms
```

With index

No **duplicates** in child nodes of the first B tree

Default supported operators for jsonb: @>, @?, @@, ?,?|,?&



Test environment

Postgres Version: 16
Running in a docker container

Datasource: chemotion eln metadata
dump: 233 MB

```
test=# \d metadata;
                Table "public.metadata"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 id     | integer |           |          |
 data   | jsonb   |           |          |
```

Entries: 18799


```
{
    "@id": "https://doi.org/10.14272/ZRSIZLVHDLEVQU-FMQUCBEESA-N/CHMO0000596"
    "url": "https://www.chemotion-repository.net/inchikey/ZRSIZLVHDLEVQU-FMQU
    "name": "CHMO:0000596 | distortionless enhancement with polarization trar
    "@type": "Dataset",
    "author": [{
            "name": "Nicolai Wippert",
            "@type": "Person",
            "givenName": "Nicolai",
            "familyName": "Wippert",
            "affiliation": {
                "name": "Institute of Organic Chemistry, Karlsruhe Institute
                "@type": "Organization"
            }
        }
    ],
    "creator": [{
            "name": "Nicolai Wippert",
            "@type": "Person",
            "givenName": "Nicolai",
            "familyName": "Wippert",
            "affiliation": {
                "name": "Institute of Organic Chemistry, Karlsruhe Institute
                "@type": "Organization"
            }
        }
    ],
    "license": "http://creativecommons.org/licenses/by-sa/4.0/",
    "@context": "https://schema.org",
    "publisher": {
        "url": "https://www.chemotion-repository.net",
        "logo": "https://www.chemotion-repository.net/images/repo/Chemotion-\
        "name": "chemotion-repository",
        "@type": "Organization"
    },
    "identifier": "CRD-12627",
    "description": "dataset for  distortionless enhancement with polarizatior
    "measurementTechnique": {
        "@id": "http://purl.obolibrary.org/obo/CHMO_0000596",
        "url": "https://terminology.nfdi4chem.de/ts/ontologies/chmo/terms?iri
        "name": "distortionless enhancement with polarization transfer",
        "@type": "DefinedTerm",
        "termCode": "CHMO:0000596",
        "alternateName": ["distortionless enhancement with polarisation trans
        "inDefinedTermSet": {
            "@id": "http://purl.obolibrary.org/obo/chmo.owl",
            "url": "http://purl.obolibrary.org/obo/chmo.owl",
            "name": "chmo",
            "@type": "DefinedTermSet"
        }
    },
```

UseCase 1 - Get the dataset with identifier CRD-12637

SELECT id FROM metadata WHERE data @> '{"identifier":"CRD-12627"}';

EXPLAIN ANALYZE SELECT id FROM metadata WHERE data @> '{"identifier":"CRD-12627"}';

```
---------------------------------------------------------------------------------------------------------
 Seq Scan on metadata  (cost=0.00..1340.99 rows=188 width=4) (actual time=40.912..215.034 rows=1 loops=1)
   Filter: (data @> '{"identifier": "CRD-12627"}'::jsonb)
   Rows Removed by Filter: 18798
 Planning Time: 0.031 ms
 Execution Time: 215.049 ms
(5 rows)
```

Full table scan
215 ms

Create an GIN index on the column data of table metadata

CREATE INDEX index_on_identifier ON metadata USING GIN(data);

```
 Bitmap Heap Scan on metadata  (cost=31.04..522.34 rows=188 width=4) (actual time=0.168..0.222 rows=1 loops=1)
   Recheck Cond: (data @> '{"identifier": "CRD-12627"}'::jsonb)
   Rows Removed by Index Recheck: 2
   Heap Blocks: exact=3
   ->  Bitmap Index Scan on index_on_identifier  (cost=0.00..30.99 rows=188 width=0) (actual time=0.084..0.084
ops=1)
         Index Cond: (data @> '{"identifier": "CRD-12627"}'::jsonb)
 Planning Time: 0.141 ms
 Execution Time: 0.305 ms
(8 rows)
```

Index scan
0.305 ms

UseCase 2 - Get all datasets with specific measurementTechnique


```
SELECT count(id) FROM metadata
WHERE data->'measurementTechnique' @> '{"@id": "http://purl.obolibrary.org/obo/CHMO_0000596"}';
```

916 rows returned

Full table scan
223 ms

```
CREATE INDEX index_on_measurement ON metadata USING GIN((data->'measurementTechnique'));
```

```
EXPLAIN ANALYZE SELECT count(id) FROM metadata
WHERE data->'measurementTechnique' @> '{"@id": "http://purl.obolibrary.org/obo/CHMO_0000596"}';
```

Index scan
9.128 ms

```
SELECT COUNT(id) FROM metadata
WHERE data->'measurementTechnique'->>'@id' = 'http://purl.obolibrary.org/obo/CHMO_0000596';
```

916 rows returned **BUT**
no index was used

UseCase 3 - Get all datasets from specific author

Get all datasets from Nicolai Wippert
Problem: Interresting item inside an array -> some psql magick must be done

```
SELECT COUNT(DISTINCT (id,x->‘name'))
FROM metadata CROSS JOIN LATERAL (SELECT jsonb_array_elements(data->'author')::jsonb as x)
WHERE x @> '{"name":"Nicolai Wippert"}';
```

Full table scan
372 ms

```
CREATE INDEX index_on_author ON metadata USING GIN((data->'author'));
```

No changes !!!

Literature

https://www.cybertec-postgresql.com/en/gin-just-an-index-type/

http://www.louisemeta.com/blog/indexes-gin/

https://www.postgresql.org/docs/current/datatype-json.html