

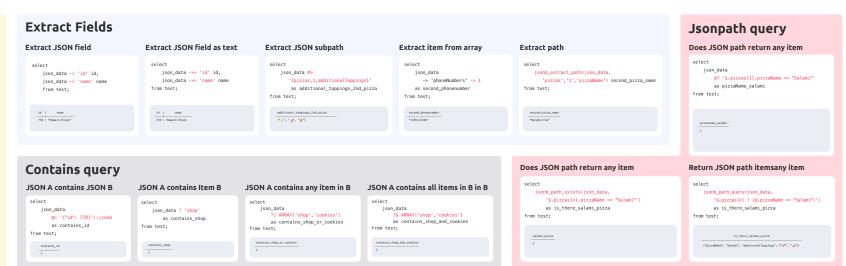


JSONB Functions Cheatsheet

@ftisiot ¥

https://www.postgresgl.org/docs/current/functions-json.html











jsonb_typeof(json_data->'id') type_id,

jsonb_typeof(json_data->'name') type_name

Types

from

Get item type

```
Tabulate
Extract to record based on type
                                                  Extract to recordset based on type
                                                   create type pizza as ("pizzaName" text,
 as (id int, name text, address text);
                                                    'additionalToppings" text[]):
                                                   select p.* from test
 from test cross join lateral
                                                   cross ioin lateral
       null::pizzaOrder, json_data) p;
                                                          null::pizza, json_data -> 'pizzas') p;
   id | name | address
   778 | Edward Olson | Unit 9398 Box 2856+
Extract to record declaring columns
                                                  Extract to recordset declaring columns
 select p.* from test
                                                   select p.* from test
 cross join lateral
                                                   cross join lateral
    jsonb_to_record(json_data)
                                                      jsonb_to_recordset(json_data -> 'pizzas')
    as p(id int, "phoneNumbers" text[]):
                                                       as p("pizzaName" text.
```

778 | {(935)583-3765±4154.(935)12345}

jsonb_object

{"name" : "francesco", "pizza" : "Margherita"}

"additionalToppings" text[]);



Concat JSON A and JSON B

as add_note

json_data

from test;



("id": 778,
 "neme": "Chard Gison",
 "hopy": "Lugis Fizza",
 "pizza": [
 ("pizzaliem": "salami", additionalToppings": ["d", "J"]),
 (pizzaliem": "Narpherita"), ...)





Return array length

from test:

json_data ->
'phoneNumbers')
from '

