



### 1. Json y Jsonb

- Json guarda en texto, Jsonb en binario
- Json mantiene espacios
- Jsonb permite hacer índices (por lo que tarda más en guardar)

### 2. Table clientes

```
CREATE TABLE clientes (
  id serial NOT NULL PRIMARY KEY,
  personales jsonb NOT NULL,
  direcciones json NOT NULL
);
CREATE INDEX ON clientes((personales->>'nif'));
```

### 3. Insert

```
insert into clientes (personales, direcciones)
values ('{
  "nif": "5095Y",
  "nombre": {
    "nombre": "jorge",
    "apellido1": "aguilera",
    "apellido2": "gonzalez"
  },
  "fechaNacimiento":{
    "dia": 21,
    "mes" : 3,
    "año": 1969
  }
}', '{
  {
    "tipo": "calle",
    "direccion": "calle ejemplo",
    "numero": 1
  },
  {
    "tipo": "avenida",
    "direccion": "avenida ejemplo",
    "numero": 223
  }
}');
```

### 4. Select Where

```
select personales->'nombre' from clientes
where personales -> 'nombre' -> 'apellido1' = 'aguilera';
```

{"nombre": "jorge", "apellido1": "aguilera", "apellido2": "gonzalez"}

### 5. Agregados

```
select min(cast(personales->'fechaNacimiento'->'dia' as int))
from clientes
```

21

### 6. Select Where Array JSON

```
select * from clientes
where direcciones -> 'tipo' = "calle";
```



ERROR: direcciones es un array

### 7. Select Where Array como JSON

Una flecha devuelve JSON

```
select elem->'tipo' from clientes
cross join json_array_elements(direcciones) elem
--
"calle"
"avenida"
```

### 8. Select Where Array como Texto

```
select elem->>'tipo' from clientes
cross join json_array_elements(direcciones) elem
--
calle
avenida
```


### 9. Where en arrays de JSON

```
select elem->>'direccion'
from clientes cross join json_array_elements(direcciones) elem
where elem->>'tipo'='calle'
```

### 10. Conclusiones

- json vs jsonb
- Usar → (json) o →> (text)

Authors :

 @jagedn  
Software developer at Puravida Software