PostreSQL
Baza danych nie musi być sztywna

Paweł Tarczykowski

Tokagen

paul@tokagen.com

https://github.com/far-rel

Tablice!

Tablice

```
CREATE TABLE my_table (
  id            SERIAL,
  int_array     integer[],
  string_array  varchar[],
  multi_dim     integer[][]
)
```

Tablice

```
SELECT int_array FROM my_table;
```

```
|  int_array  |
+-------------+
|  {1,2,3,4}  |
```

Uwaga!

Pierwszy indeks to 1!

Tablice

```
SELECT int_array[1] FROM my_table;
```

```
| int_array[1] |
+--------------+
|      1       |
```

Tablice

```
SELECT int_array[2:3] FROM my_table;


| int_array[2:3] |
+----------------+
|     {2,3}      |
```

Tablice

```
INSERT INTO my_table(int_array)
VALUES ('{1,2,3,4,5}');

INSERT INTO my_table(int_array)
VALUES (ARRAY[1,2,3,4,5]);
```

Tablice

```
UPDATE my_table SET int_array[1] = 10;

UPDATE my_table
SET int_array[1:2] = '{11,10}'
```

Tablice

```
UPDATE my_table SET int_array[0] = 10;

UPDATE my_table SET int_array[-1] = 11;
```

Tablice

```
UPDATE my_table SET int_array = int_array || 10;

UPDATE my_table SET int_array = 10 || int_array;
```

Tablice

```
SELECT ARRAY[1,2,3] @> ARRAY[1,2];

SELECT ARRAY[1,2] <@ ARRAY[1,2,3];

SELECT ARRAY[1,2] && ARRAY[2,3];
```

TABLICE

```
SELECT unnest(int_array) FROM my_table;
```

```
|  int_array  |
+-------------+
|      1      |
|      2      |
|      3      |
|      4      |
```

Tablice

```
SELECT e.*, array_agg(ed.department_id)
FROM employees e
JOIN employee_departments ed ON
e.id = ed.employee_id
GROUP BY e.id;
|  id  | name | depts |
+------+------+-------+
|  1   | John | {1,2} |
```

Hstore!

Hstore

```
CREATE EXTENSION hstore;

CREATE TABLE hstore_table (
  id              SERIAL,
  hstore_field    hstore
);
```

Hstore

```
SELECT hstore_field FROM hstore_table;


| hstore_table |
+--------------+
|  a=>b,c=>d   |
```

Hstore

```
SELECT hstore_field -> 'a' FROM hstore_table;


| hstore_table |
+--------------+
|      b       |
```

Hstore

```
INSERT INTO hstore_table
VALUES ('a=>b, c=>d');

UPDATE hstore_table
SET hstore_table = 'a=>c';
```

Hstore

```
UPDATE hstore_table SET
hstore_field = hstore_field || 'f=>g';

UPDATE hstore_table SET
hstore_field = delete(hstore_field, 'a');
```

Hstore

```
SELECT akeys(hstore_field) FROM hstore_table;

SELECT avals(hstore_field) FROM hstore_table;
```

Hstore

```
SELECT hstore(employees) FROM employees;


SELECT * FROM
populate_record(NULL::employees,'id=>1,name=>John')
```

Własne typy danych!

Enumeratory

```
CREATE TYPE order_status AS ENUM (
  'in_progress',
  'paid',
  'in_delivery',
  'delivered'
);
```

Typy złożone

```
CREATE TYPE person AS (
  first_name VARCHAR,
  last_name VARCHAR,
  age INTEGER
);
```

Typy złożone

```
CREATE TABLE orders (
  id        SERIAL,
  p         person,
  status    order_status
);
```

Typy złożone

```
INSERT INTO orders(p, status)
VALUES (('John', 'Doe', 21), 'delivered');


UPDATE orders SET p.first_name = 'Paul',
                  status = 'in_delivery';
```

Typy złożone

```
SELECT (orders.p).first_name FROM orders;
```

```
|  first_name  |
+--------------+
|     Paul     |
```

Pytania?

Dziękuję!