Upsert – inserts or update data if the new row already exists in the table

PostgreSQL UPSERT statement

The term upsert is referred to as a merge. The idea is that when you insert a new row into the table, PostgreSQL will update the row if it already exists, otherwise, PostgreSQL inserts the new row. That is why we call the action is upsert (update or insert).

PostgreSQL, you use the `INSERT ON CONFLICT` statement as follows:

```
INSERT INTO table_name(column_list) VALUES(value_list)
ON CONFLICT target action;
```

The action can be:

- `DO NOTHING` – means do nothing if the row already exists in the table.
- `DO UPDATE SET column_1 = value_1, .. WHERE condition` – update some fields in the table.

Upsert – inserts or update data if the new row already exists in the table

PostgreSQL upsert examples

```
CREATE TABLE customers (
customer_id serial PRIMARY KEY,
name VARCHAR UNIQUE,
email VARCHAR NOT NULL,
active bool NOT NULL DEFAULT TRUE
);
```

The following INSERT statement inserts some rows into the `customers` table.

```
INSERT INTO customers (NAME, email)
VALUES
('IBM', 'contact@ibm.com'),
(
'Microsoft',
'contact@microsoft.com'
),
(
'Intel',
'contact@intel.com'
);
```

```
#SELECT * FROM customers;
 customer_id |   name    |         email          | active
-------------+-----------+------------------------+--------
           1 | IBM       | contact@ibm.com        | t
           2 | Microsoft | contact@microsoft.com  | t
           3 | Intel     | contact@intel.com      | t
```

Upsert – inserts or update data if the new row already exists in the table

Suppose Microsoft changes the contact email from contact@microsoft.com to hotline@microft.com, we can update it using the UPDATE statement. However, to demonstrate the upsert feature, we use the following `INSERT ON CONFLICT` statement:

```
INSERT INTO customers (name, email)
VALUES
(
'Microsoft',
'hotline@microsoft.com'
)
ON CONFLICT (name)
DO NOTHING;
```

Suppose, you want to concatenate the new email with the old email when inserting a customer that already exists, in this case, you use UPDATE clause as the action of the INSERT statement as follows:

```
INSERT INTO customers (name, email)
VALUES
(
'Microsoft',
'hotline@microsoft.com'
)
ON CONFLICT (name)
DO
UPDATE
 SET email = EXCLUDED.email || ';' || customers.email;
```

```
#SELECT * FROM customers;
 customer_id |   name    |                    email                    | active
-------------+-----------+---------------------------------------------+--------
           1 | IBM       | contact@ibm.com                             | t
           3 | Intel     | contact@intel.com                           | t
           2 | Microsoft | hotline@microsoft.com;contact@microsoft.com | t
```