

# PostgreSQL Upsert Using INSERT ON CONFLICT statement

```
1  --The following statement creates a new table called customers
2  --to demonstrate the PostgreSQL upsert feature.
3
4  DROP TABLE IF EXISTS customers;
5
6  CREATE TABLE customers (
7      customer_id serial PRIMARY KEY,
8      name VARCHAR UNIQUE,
9      email VARCHAR NOT NULL,
10     active bool NOT NULL DEFAULT TRUE
11 );
```

Messages Data Output

NOTICE: table "customers" does not exist, skipping  
CREATE TABLE

Query Editor Query History Explain Recommendations

```
1  --The following INSERT statement inserts some rows into the customers table.
2
3  INSERT INTO
4      customers (name, email)
5  VALUES
6      ('IBM', 'contact@ibm.com'),
7      ('Microsoft', 'contact@microsoft.com'),
8      ('Intel', 'contact@intel.com');
```

Messages Data Output

INSERT 0 3

```
1  --to demonstrate the upsert feature, we use the following
2  --INSERT ON CONFLICT statement:
3
4  INSERT INTO customers (NAME, email)
5  VALUES('Microsoft','hotline@microsoft.com')
6  ON CONFLICT ON CONSTRAINT customers_name_key
7  DO NOTHING;
```

Messages Data Output

INSERT 0 0

Query Editor   Query History   Explain   Notifications

```
1  --The following statement is equivalent to the above
2  --statement but it uses the name column instead of
3  --the unique constraint name as the target of the INSERT statement.
4
5  INSERT INTO customers (name, email)
6  VALUES('Microsoft','hotline@microsoft.com')
7  ON CONFLICT (name)
8  DO NOTHING;
```

Messages   Data Output

INSERT 0 0

Query Editor   Query History   Explain   Notifications

```
1  --use the UPDATE clause as the action of the INSERT statement as follows:
2
3  INSERT INTO customers (name, email)
4  VALUES('Microsoft','hotline@microsoft.com')
5  ON CONFLICT (name)
6  DO
7      UPDATE SET email = EXCLUDED.email || ';' || customers.email;
```

Messages   Data Output

INSERT 0 1