

[Update](#) – updates existing data in a table.

PostgreSQL UPDATE syntax

```
UPDATE table
SET column1 = value1,
    column2 = value2, ...
WHERE condition;
```

First, specify the table name where you want to update the data after `UPDATE` clause.

Second, list the columns whose values you want to change in the `SET` clause. If you update values in multiple columns, you use a comma (,) to separate each pair of column and value. The columns that are not on the list retain their original values.

Third, determine which rows you want to update in the condition of the `WHERE` clause. If you omit the `WHERE` clause, all the rows in the table are updated.

PostgreSQL UPDATE examples

Let's examine the content of the `link` table:

```
SELECT * FROM link;
```

id	url	name	description	rel	last_update
1	http://www.postgresqltutorial.com	PostgreSQL Tutorial	(Null)	(Null)	(Null)
2	http://www.oreilly.com	O'Reilly Media	(Null)	(Null)	(Null)
3	http://www.google.com	Google	(Null)	(Null)	(Null)
4	http://www.yahoo.com	Yahoo	(Null)	(Null)	(Null)
5	http://www.bing.com	Bing	(Null)	(Null)	(Null)
6	http://www.facebook.com	Facebook	(Null)	(Null)	2013-06-01
7	https://www.tumblr.com/	Tumblr	(Null)	(Null)	2013-06-02
8	http://www.postgresql.org	PostgreSQL	(Null)	(Null)	2013-06-02

[Update](#) – updates existing data in a table.

PostgreSQL update table partially example

```
UPDATE link  
SET last_update = '2013-06-02'  
WHERE  
last_update IS NULL;
```

id	url	name	description	rel	last_update
6	http://www.facebook.com	Facebook	(Null)	(Null)	2013-06-01
7	https://www.tumblr.com/	Tumblr	(Null)	(Null)	2013-06-02
8	http://www.postgresql.org	PostgreSQL	(Null)	(Null)	2013-06-02
1	http://www.postgresqltutori	PostgreSQL Tutorial	(Null)	(Null)	2013-06-02
2	http://www.oreilly.com	O'Reilly Media	(Null)	(Null)	2013-06-02
3	http://www.google.com	Google	(Null)	(Null)	2013-06-02
4	http://www.yahoo.com	Yahoo	(Null)	(Null)	2013-06-02
5	http://www.bing.com	Bing	(Null)	(Null)	2013-06-02

PostgreSQL update all rows in a table

```
UPDATE link SET rel = 'nofollow';
```

id	url	name	description	rel	last_update
6	http://www.facebook.com	Facebook	(Null)	nofollow	2013-06-01
7	https://www.tumblr.com/	Tumblr	(Null)	nofollow	2013-06-02
8	http://www.postgresql.org	PostgreSQL	(Null)	nofollow	2013-06-02
1	http://www.postgresqltutori	PostgreSQL Tutorial	(Null)	nofollow	2013-06-02
2	http://www.oreilly.com	O'Reilly Media	(Null)	nofollow	2013-06-02
3	http://www.google.com	Google	(Null)	nofollow	2013-06-02
4	http://www.yahoo.com	Yahoo	(Null)	nofollow	2013-06-02
5	http://www.bing.com	Bing	(Null)	nofollow	2013-06-02

[Update](#) – updates existing data in a table.

You can also update data of a column from another column within the same table. The following statement copies the values of the `name` column to the `description` column of the `link` table:

```
UPDATE link
SET description = name;
```

id	url	name	description	rel	last_update
6	http://www.facebook.com	Facebook	Facebook	nofollow	2013-06-01
7	https://www.tumblr.com/	Tumblr	Tumblr	nofollow	2013-06-02
8	http://www.postgresql.org	PostgreSQL	PostgreSQL	nofollow	2013-06-02
1	http://www.postgresqltutori	PostgreSQL Tutorial	PostgreSQL Tutorial	nofollow	2013-06-02
2	http://www.oreilly.com	O'Reilly Media	O'Reilly Media	nofollow	2013-06-02
3	http://www.google.com	Google	Google	nofollow	2013-06-02
4	http://www.yahoo.com	Yahoo	Yahoo	nofollow	2013-06-02
5	http://www.bing.com	Bing	Bing	nofollow	2013-06-02

The following statement updates values that come from the `link` table for the columns in the `link_tmp` table:

```
UPDATE link_tmp
SET rel = link.rel,
description = link.description,
last_update = link.last_update
FROM
link
WHERE
link_tmp.id = link.id;
```

PostgreSQL update with returning clause

```
UPDATE link SET description = 'Learn PostgreSQL fast and easy', rel = 'follow'
WHERE ID = 1
RETURNING id, description, rel;
```

id	description	rel
1	Learn PostgreSQL fast and easy	follow