**UPDATE syntax**

UPDATE table

SET column1 = value1,

    column2 = value2 ,...

WHERE

   condition;

Examples:

CREATE TABLE link (

   ID serial PRIMARY KEY,

   url VARCHAR (255) NOT NULL,

   name VARCHAR (255) NOT NULL,

   description VARCHAR (255),

   rel VARCHAR (50)

);

INSERT INTO link (url, name)

VALUES

('http://www.google.com','Google'),

('http://www.yahoo.com','Yahoo'),

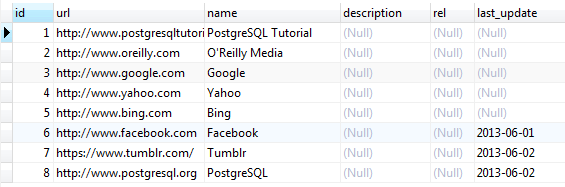
('http://www.bing.com','Bing');

SELECT

   \*

FROM

   link;



UPDATE link

SET last\_update = DEFAULT

WHERE

   last\_update IS NULL;

UPDATE link

SET rel = 'nofollow';

UPDATE link

SET description = name;

UPDATE link

SET description = 'Learn PostgreSQL fast and easy',

rel = 'follow'

WHERE

   ID = 1

RETURNING id,

   description,

   rel;

Practice:

id | name | age | address | salary

----+-------+-----+-----------+--------

1 | Paul | 32 | California| 20000

2 | Allen | 25 | Texas | 15000

3 | Teddy | 23 | Norway | 20000

4 | Mark | 25 | Rich-Mond | 65000

5 | David | 27 | Texas | 85000

6 | Kim | 22 | South-Hall| 45000

7 | James | 24 | Houston | 10000

(7 rows)

UPDATE COMPANY SET SALARY = 15000 WHERE ID = 3;

UPDATE COMPANY SET ADDRESS = 'Texas', SALARY=20000;

# TRUNCATE TABLE Statement

The TRUNCATE TABLE statement is used to remove all records from a table or set of tables in PostgreSQL. It performs the same function as a DELETE statement without a WHERE clause. If you truncate a table, the TRUNCATE TABLE statement can not be rolled back unless it is within a transaction that has not been committed. truncating a table is a fast way to clear out records from a table because it does not need to scan the table. Truncating a table is also a lot easier than dropping the table and recreating it.

TRUNCATE ONLY products;