

# POSTGRESQL - LOGICAL OPERATORS

[http://www.tutorialspoint.com/postgresql/postgresql\\_logical-operators.htm](http://www.tutorialspoint.com/postgresql/postgresql_logical-operators.htm)

Copyright © tutorialspoint.com

Consider the table [COMPANY](#) having records as follows:

```
testdb# select * from COMPANY;
 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)
```

Here are simple examples showing usage of PostgreSQL LOGICAL Operators. Following SELECT statement list down all the records where AGE is greater than or equal to 25 and salary is greater than or equal to 65000.00:

```
testdb=# SELECT * FROM COMPANY WHERE AGE >= 25 AND SALARY >= 6500;
```

Above PostgreSQL statement will produce the following result:

```
 id | name  | age | address  | salary
-----+-----+-----+-----+-----
  1 | Paul  |  32 | California | 20000
  2 | Allen |  25 | Texas     | 15000
  4 | Mark  |  25 | Rich-Mond | 65000
  5 | David |  27 | Texas     | 85000
(4 rows)
```

Following SELECT statement lists down all the records where AGE is greater than or equal to 25 **OR** salary is greater than or equal to 65000.00:

```
testdb=# SELECT * FROM COMPANY WHERE AGE >= 25 OR SALARY >= 6500;
```

Above PostgreSQL statement will produce the following result:

```
 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
  8 | Paul  |  24 | Houston   | 20000
  9 | James |  44 | Norway    |  5000
 10 | James |  45 | Texas     |  5000
(10 rows)
```

Following SELECT statement lists down all the records where AGE is not NULL which means all the records because none of the record is having AGE equal to NULL:

```
testdb=# SELECT * FROM COMPANY WHERE SALARY IS NOT NULL;
```

Above PostgreSQL statement will produce the following result:

```
 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
8 | Paul   | 24 | Houston    | 20000
9 | James  | 44 | Norway     | 5000
10 | James  | 45 | Texas      | 5000
(10 rows)
```