

# POSTGRESQL - PRIVILEGES

[http://www.tutorialspoint.com/postgresql/postgresql\\_privileges.htm](http://www.tutorialspoint.com/postgresql/postgresql_privileges.htm)

Copyright © tutorialspoint.com

Whenever an object is created in a database, an owner is assigned to it. The owner is usually the one who executed the creation statement. For most kinds of objects, the initial state is that only the owner (or a superuser) can modify or delete the object. To allow other roles or users to use it, *privileges* or permission must be granted.

Different kinds of privileges in PostgreSQL are:

- SELECT,
- INSERT,
- UPDATE,
- DELETE,
- TRUNCATE,
- REFERENCES,
- TRIGGER,
- CREATE,
- CONNECT,
- TEMPORARY,
- EXECUTE, and
- USAGE

Depending on the type of the object (table, function, etc.), privileges are applied to the object. To assign privileges to the users, the GRANT command is used.

## Syntax for GRANT

Basic syntax for GRANT command is as follows:

```
GRANT privilege [, ...]
ON object [, ...]
TO { PUBLIC | GROUP group | username }
```

- **privilege** values could be: SELECT, INSERT, UPDATE, DELETE, RULE, ALL.
- **object**: The name of an object to which to grant access. The possible objects are: table, view, sequence
- **PUBLIC**: A short form representing all users.
- **GROUP group**: A group to whom to grant privileges.
- **username**: The name of a user to whom to grant privileges. PUBLIC is a short form representing all users.

The privileges can be revoked using the REVOKE command.

## Syntax for REVOKE

Basic syntax for REVOKE command is as follows:

```
REVOKE privilege [, ...]
ON object [, ...]
```

```
FROM { PUBLIC | GROUP groupname | username }
```

- **privilege** values could be: SELECT, INSERT, UPDATE, DELETE, RULE, ALL.
- **object**: The name of an object to which to grant access. The possible objects are: table, view, sequence
- **PUBLIC**: A short form representing all users.
- **GROUP group**: A group to whom to grant privileges.
- **username**: The name of a user to whom to grant privileges. PUBLIC is a short form representing all users.

## Example

To understand the privileges, let's first create a USER as follows:

```
testdb=# CREATE USER manisha WITH PASSWORD 'password';  
CREATE ROLE
```

The message CREATE ROLE indicates that the USER "manisha" is created.

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)
```

Next, let's grants all privileges on a table COMPANY to the user "manisha" as follows:

```
testdb=# GRANT ALL ON COMPANY TO manisha;  
GRANT
```

The message GRANT indicates that all privileges are assigned to the USER.

Next, let's revoke the privileges from the USER "manisha" as follows:

```
testdb=# REVOKE ALL ON COMPANY FROM manisha;  
REVOKE
```

The message REVOKE indicates that all privileges are revoked from the USER.

You can even delete the user as follows:

```
testdb=# DROP USER manisha;  
DROP ROLE
```

The message DROP ROLE indicates USER manisha is deleted from the database.