PostgreSQL - Schema

A **schema** is a named collection of tables. A schema can also contain views, indexes, sequences, data types, operators, and functions. Schemas are analogous to directories at the operating system level, except that schemas cannot be nested. PostgreSQL statement CREATE SCHEMA creates a schema.

Syntax

The basic syntax of CREATE SCHEMA is as follows −

```
CREATE SCHEMA name;
```

Where *name* is the name of the schema.

Syntax to Create Table in Schema

The basic syntax to create table in schema is as follows −

```
CREATE TABLE myschema.mytable (
...
);
```

Example

Let us see an example for creating a schema. Connect to the database *testdb* and create a schema *myschema* as follows −

```
testdb=# create schema myschema;
CREATE SCHEMA
```

The message "CREATE SCHEMA" signifies that the schema is created successfully.

Now, let us create a table in the above schema as follows −

```
testdb=# create table myschema.company(
   ID   INT              NOT NULL,
   NAME VARCHAR (20)     NOT NULL,
   AGE  INT              NOT NULL,
   ADDRESS  CHAR (25),
   SALARY   DECIMAL (18, 2),
```

```
   PRIMARY KEY (ID)
);
```

This will create an empty table. You can verify the table created with the command given below −

```
testdb=# select * from myschema.company;
```

This would produce the following result −

```
 id | name | age | address | salary
----+------+-----+---------+--------
(0 rows)
```

Syntax to Drop Schema

To drop a schema if it is empty (all objects in it have been dropped), use the command −

```
DROP SCHEMA myschema;
```

To drop a schema including all contained objects, use the command −

```
DROP SCHEMA myschema CASCADE;
```

Advantages of using a Schema

- It allows many users to use one database without interfering with each other.
- It organizes database objects into logical groups to make them more manageable.
- Third-party applications can be put into separate schemas so they do not collide with the names of other objects.