PostgreSQL CREATE Command

1. Purpose

The CREATE command in PostgreSQL is used to create new database objects such as tables, databases, views, indexes, schemas, functions, and more. It defines and initializes new components that can later be used to store or manipulate data.

2. Syntax

a. Create Table

```
CREATE TABLE table_name (
   column_name1 data_type [constraints],
   column_name2 data_type [constraints],
   ...
);
```

b. Create Database

```
CREATE DATABASE database_name;
```

c. Create Index

```
CREATE INDEX index_name ON table_name (column_name);
```

3. Parameters

Parameter	Description
CREATE	The keyword that starts the command to create a new object.
TABLE	Specifies that you are creating a new table.
DATABASE	Specifies that you are creating a new database.
INDEX	Indicates the creation of an index for faster searching.
table_name	The name of the table you want to create.
column_name	The name of each column in the table.
data_type	The data type for each column (e.g., INTEGER, TEXT, DATE).
constraints	Optional rules (e.g., PRIMARY KEY, NOT NULL, UNIQUE).
index_name	A name for the new index.

4. Examples

Example 1: Create a Table

```
CREATE TABLE employees (
   id SERIAL PRIMARY KEY,
   name TEXT NOT NULL,
   hire_date DATE,
    salary NUMERIC(10, 2)
);
```

Explanation: This command creates a table called employees with four columns: id (auto-incrementing primary key), name (non-nullable text), hire_date (date), and salary (numeric with 2 decimal places).

Example 2: Create a Database

```
CREATE DATABASE company_db;
```

Explanation: This command creates a new database named company_db.

Example 3: Create an Index

```
CREATE INDEX idx_name ON employees (name);
```

Explanation: This creates an index named idx_name on the name column of the employees table.

5. Use Cases

Creating new databases for different applications or environments (e.g., development, testing, production). Defining new tables to store structured data in a relational format. Creating indexes to speed up queries on frequently searched columns. Setting up views, functions, or schemas for modular and reusable database logic.

6. Best Practices

■ **Always specify constraints** like NOT NULL or PRIMARY KEY when needed to enforce data integrity. ■ **Use descriptive names** for tables and columns to make your schema easy to understand. ■ **Use IF NOT EXISTS** (when supported) to avoid errors when creating objects that might already exist. ■ **Avoid using reserved keywords** (like SELECT, USER) as object names. ■ **Don’t forget to back up** before creating structures that might conflict with existing objects.

7. Related Commands

Command	Purpose
DROP	Deletes an existing object like a table or database.
ALTER	Modifies an existing database object (e.g., adding columns to a table).
INSERT	Adds data to a table created using CREATE TABLE.
SELECT	Queries data from tables.
CREATE OR REPLACE	Creates or updates objects like functions and views.