what is the function, procedure and package in pgress database.

In PostgreSQL, **functions**, **procedures**, and **packages** serve different purposes in database management. Here's how they differ:

**1. Function (FUNCTION)**

* A function in PostgreSQL is used to perform calculations and return a value.
* It is commonly used for reusable logic within queries.
* Functions can return single values (RETURNS INT), tables (RETURNS TABLE), or even complex types.

Example:

CREATE FUNCTION add\_numbers(a INT, b INT) RETURNS INT AS $$ BEGIN RETURN a + b; END; $$ LANGUAGE plpgsql;

Call the function:

SELECT add\_numbers(5, 10);

**2. Procedure (PROCEDURE)**

* A procedure is similar to a function but does **not return a value**.
* Used for operations that modify data (INSERT, UPDATE, etc.).
* Introduced in PostgreSQL 11.

Example:

CREATE PROCEDURE insert\_user(IN username TEXT) LANGUAGE plpgsql AS $$ BEGIN INSERT INTO users (name) VALUES (username); END; $$;

Call the procedure:

CALL insert\_user('Alice');

**3. Package (SCHEMA)**

* PostgreSQL does **not** have an explicit "package" concept like Oracle.
* However, **schemas** act as logical containers for functions, procedures, and tables.
* You can group database objects inside a schema to organize them efficiently.

Example:

CREATE SCHEMA fastapi\_db\_package; CREATE FUNCTION fastapi\_db\_package.get\_user\_count() RETURNS INT AS $$ BEGIN RETURN (SELECT COUNT(\*) FROM fastapi\_db\_package.users); END; $$ LANGUAGE plpgsql;

Call the function inside the schema:

SELECT fastapi\_db\_package.get\_user\_count();

**Summary:**

| **Object** | **Returns Value?** | **Modifies Data?** | **Purpose** |
| --- | --- | --- | --- |
| Function (FUNCTION) | ✅ Yes | ❌ No | Returns computed values |
| Procedure (PROCEDURE) | ❌ No | ✅ Yes | Performs operations like INSERT, UPDATE |
| Package (Schema) (SCHEMA) | ✅ Yes | ✅ Yes | Logical grouping of functions/procedures |

While PostgreSQL lacks a native package system like Oracle, **schemas** can be effectively used to organize database objects.

Let me know if you need more details! 🚀