Tablespces

Types of Tablespaces in PostgreSQL

Besides the default tablespaces, PostgreSQL allows users to create custom tablespaces. These can be categorized as:

1. **System Tablespaces**
    1. pg_default and pg_global fall under this category.
    2. They are created automatically during PostgreSQL installation.
2. **User-Defined Tablespaces**
    1. Created by the DBA to store database objects in a specific location.
    2. Useful for performance optimization, separating large tables from small ones, or storing indexes on a different disk.

    Example:-
    CREATE TABLESPACE temp_space LOCATION '/mnt/temp_space';
    ALTER DATABASE mydb SET temp_tablespaces = temp_space;
3. **Temporary Tablespaces**

- Used for storing temporary objects such as temporary tables and query sorting operations (ORDER BY, GROUP BY).
- Can help improve performance by placing temp data on a fast disk.
- Example:-

CREATE TABLESPACE temp_space LOCATION '/mnt/temp_space';

ALTER DATABASE mydb SET temp_tablespaces = temp_space;

A . How to Check Available Tablespaces:-

1. To list all tablespaces in your PostgreSQL instance, run:

SELECT spcname, spclocation FROM pg_tablespace;

Or, using the psql command-line tool:

\db

Default Tablespaces in PostgreSQL

PostgreSQL comes with two default tablespaces:

1. pg_default

    1. This is the default tablespace where most database objects (tables, indexes, etc.) are stored unless a different tablespace is explicitly specified.
    2. It is located inside the PostgreSQL data directory (PGDATA).

    Example location:-
    /var/lib/pgsql/15/data/base/

2. pg_global

- This tablespace is used for shared system catalogs, such as user and role information (pg_authid).
- It contains metadata that is accessible across all databases in a PostgreSQL instance.
- Example location:-

/var/lib/pgsql/15/data/global/

---

1. Tablespace Structure in PostgreSQL

When you create a tablespace, PostgreSQL manages it as a symbolic link inside the pg_tblspc directory of the PostgreSQL data directory.
Example:

-- Suppose you create a tablespace named my_tablespace at /mnt/data/tablespace1, PostgreSQL creates a symlink:-

/var/lib/pgsql/15/data/pg_tblspc/16384 -> /mnt/data/tablespace1

· The number (16384) is the **OID** (Object Identifier) of the tablespace.

· Inside /mnt/data/tablespace1, PostgreSQL creates subdirectories for databases that use this tablespace.

2. How Tablespaces Work with Databases

- When a new database is created, it is assigned a tablespace (default is pg_default).
- If a table or index is created inside a custom tablespace, its files are stored in that location.

- A single database can use multiple tablespaces for different objects.

3. Tablespace Permissions

Only the **superuser** or a user with CREATE privilege on a tablespace can create objects inside it.

Granting access:

GRANT CREATE ON TABLESPACE my_tablespace TO some_user;

Revoking access:

REVOKE CREATE ON TABLESPACE my_tablespace FROM some_user;

Checking tablespace permissions:-

SELECT * FROM pg_tablespace WHERE spcname = 'my_tablespace';

4. Changing the Default Tablespace for a Database

You can specify a different default tablespace when creating a
CREATE DATABASE mydb TABLESPACE my_tablespace;

To change the default tablespace for a database:-
ALTER DATABASE mydb SET default_tablespace = my_tablespace;

5. Moving Existing Tables to a Different Tablespace:-

If you need to move an existing table or index to a different tablespace:-
ALTER TABLE my_table SET TABLESPACE my_tablespace;

For indexes:-
ALTER INDEX my_index SET TABLESPACE my_tablespace;

6. Checking Disk Usage of Tablespaces:-

To check space used by each tablespace:-

SELECT spcname, pg_size_pretty(pg_tablespace_size(spcname)) FROM pg_tablespace;

To check space used by a specific tablespace:
SELECT pg_size_pretty(pg_tablespace_size('my_tablespace'));

7. Dropping a Tablespace:-

Before dropping, ensure there are no objects using the tablespace:-

DROP TABLESPACE my_tablespace;

If there are dependencies, move them to another tablespace first using ALTER TABLE or ALTER INDEX.

8. Best Practices for Using Tablespaces:-

✅Use tablespaces to **separate frequently accessed tables from less-used tables** for better performance.
✅Store **indexes in a different tablespace** to reduce disk contention.
✅Use a **fast disk (SSD) for temporary tablespaces** to improve query performance.
✅Regularly **monitor tablespace usage** to avoid running out of disk space.