**Collecting Log Files for Wraparound Warnings**

**Objective**

* Understand how PostgreSQL warns about **transaction ID wraparound**
* Simulate conditions that may lead to wraparound
* Configure PostgreSQL to **log warnings**
* View and collect **log file evidence**

**Prerequisites**

* PostgreSQL installed
* Logging enabled in postgresql.conf
* Superuser access to change configuration
* Access to the PostgreSQL data directory and log files

**Step 1: Enable Logging for Warnings**

Edit your postgresql.conf file (on Windows or Linux), usually located in:

C:\Program Files\PostgreSQL\<version>\data\postgresql.conf

Enable or modify the following:

logging\_collector = on

log\_directory = 'pg\_log' # or 'log' or any writable folder

log\_filename = 'postgresql-%Y-%m-%d.log'

log\_line\_prefix = '%t [%p]: [%l-1] user=%u,db=%d '

log\_min\_messages = warning

Restart PostgreSQL:

pg\_ctl restart -D "C:\Program Files\PostgreSQL\<version>\data"

**Step 2: Check Current Transaction Age**

Use this SQL command:

SELECT datname, age(datfrozenxid)

FROM pg\_database;

If the age(datfrozenxid) is close to **2,000,000,000**, PostgreSQL will start issuing warnings. You may not reach that naturally, so proceed to simulate.

**Step 3: Simulate Wraparound Risk (Test Only)**

**Do this only on a test database.**

Create a custom database and simulate by disabling autovacuum:

CREATE DATABASE wrap\_test;

\c wrap\_test

-- Disable autovacuum

ALTER SYSTEM SET autovacuum = off;

SELECT pg\_reload\_conf();

Insert many transactions in a loop:

DO $$

BEGIN

FOR i IN 1..1000000 LOOP

EXECUTE format('CREATE TABLE dummy\_table\_%s (id INT); DROP TABLE dummy\_table\_%s;', i, i);

END LOOP;

END $$;

This aggressively consumes transaction IDs.

**Step 4: Monitor Log File for Warnings**

Go to your log folder:

C:\Program Files\PostgreSQL\<version>\data\pg\_log\

Search the latest log file for warning messages like:

WARNING: database "wrap\_test" must be vacuumed within 2000000000 transactions

or

ERROR: database is not accepting commands to avoid wraparound data loss

**Step 5: Cleanup and Reactivate Autovacuum**

After test:

ALTER SYSTEM SET autovacuum = on;

SELECT pg\_reload\_conf();