**VACUUM FREEZE in PostgreSQL**

**What is VACUUM FREEZE?**

VACUUM FREEZE is a special form of vacuum in PostgreSQL that **marks all tuples (rows) as "frozen"** — meaning they **won’t need future freezing**, even for very old transactions.

This is important to **prevent transaction ID (XID) wraparound**, which can lead to **data loss or database shutdown** if not managed properly.

**Why Freeze?**

* PostgreSQL uses 32-bit transaction IDs.
* After ~2 billion transactions, old rows must be frozen to avoid reuse of XIDs.
* Autovacuum usually handles this automatically.
* You can **force it manually** using VACUUM FREEZE.

**Example**

**Step 1: Create a sample table and insert data**

CREATE TABLE demo\_freeze (

id SERIAL PRIMARY KEY,

name TEXT

);

INSERT INTO demo\_freeze (name)

SELECT 'row ' || g

FROM generate\_series(1, 1000) AS g;

**Step 2: Run a regular vacuum**

VACUUM demo\_freeze;

This reclaims space and updates stats but **does not force freeze**.

**Step 3: Check freeze status**

SELECT relname, relfrozenxid

FROM pg\_class

WHERE relname = 'demo\_freeze';

The relfrozenxid tells you the oldest transaction ID for unfrozen rows.

**Step 4: Run VACUUM FREEZE**

VACUUM FREEZE demo\_freeze;

Now all tuples are marked as frozen and won’t need another freeze until changed.