**Lab Exercise 19- Monitoring PostgreSQL with pg\_stat\_activity and pg\_stat\_database**

**Objective**

* Learn how to monitor active sessions and database statistics
* Understand how to detect long-running queries and performance bottlenecks

**Prerequisites**

* PostgreSQL installed and running
* Access to psql as a superuser or a role with monitoring rights

**Step 1: Connect to PostgreSQL**

Open a terminal or command prompt and connect using:

psql -U postgres -d your\_database

**Step 2: View Active Sessions with pg\_stat\_activity**

Run the following query to see all active sessions:

SELECT pid, usename, datname, client\_addr, state, query\_start, query

FROM pg\_stat\_activity;

**Explanation**

* pid: process ID of the backend
* usename: user connected
* datname: database name
* client\_addr: IP address of the client
* state: current state such as active or idle
* query\_start: timestamp of when the query began
* query: the actual SQL being run

**Step 3: Find Long-Running Queries**

To find queries running for more than 1 minute:

SELECT pid, now() - query\_start AS duration, query

FROM pg\_stat\_activity

WHERE state = 'active' AND now() - query\_start > interval '1 minute';

**Step 4: View General Database Stats with pg\_stat\_database**

Run:

SELECT datname, numbackends, xact\_commit, xact\_rollback, blks\_read, blks\_hit

FROM pg\_stat\_database;

**Explanation**

* datname: database name
* numbackends: number of active connections
* xact\_commit: number of committed transactions
* xact\_rollback: number of rolled-back transactions
* blks\_read: disk blocks read
* blks\_hit: memory blocks served from cache

**Step 5: Reset Statistics (Optional)**

To clear the collected statistics:

SELECT pg\_stat\_reset();

Note: This resets stats for all databases and should be done with caution.

**Step 6: Monitor in Real Time (Optional)**

You can run this repeatedly in psql to simulate real-time monitoring:

watch -n 2 "psql -U postgres -d your\_database -c \"SELECT pid, state, query FROM pg\_stat\_activity;\""

This is supported on Linux or WSL using watch. On Windows, you can manually re-run the query.

**Summary Table**

| **View** | **Command** | **Description** |
| --- | --- | --- |
| Active sessions | SELECT from pg\_stat\_activity | Shows current queries |
| Database stats | SELECT from pg\_stat\_database | Shows DB-level metrics |
| Long queries | SELECT with duration filter | Identifies slow queries |
| Reset stats | SELECT pg\_stat\_reset() | Clears statistics counters |