**PostgreSQL Multi-Database Setup & Sample Data Script**

**1. Install PostgreSQL**

bash

CopyEdit

sudo apt update

sudo apt install postgresql -y

**2. Set a New Password for the postgres User**

bash

CopyEdit

sudo -u postgres psql -c "ALTER USER postgres WITH PASSWORD 'Yash@250499';"

**3. Create Databases Using SQL Script**

**Option A: Create via PL/pgSQL loop**

bash

CopyEdit

nano /tmp/setup\_db1.sql

**Contents of /tmp/setup\_db1.sql:**

sql

CopyEdit

\c postgres;

DO $$

DECLARE

dbname TEXT;

BEGIN

FOR i IN 1..4 LOOP

dbname := format('app\_db%s', i);

EXECUTE format('DROP DATABASE IF EXISTS %I', dbname);

EXECUTE format('CREATE DATABASE %I', dbname);

END LOOP;

END $$;

Run the script:

bash

CopyEdit

sudo -u postgres psql -f /tmp/setup\_db1.sql

**Option B: Plain SQL Alternative (if PL/pgSQL causes issues)**

bash

CopyEdit

nano /tmp/setup\_db1\_fixed.sql

**Contents of /tmp/setup\_db1\_fixed.sql:**

sql

CopyEdit

-- Run this on 'postgres' database

DROP DATABASE IF EXISTS app\_db1;

DROP DATABASE IF EXISTS app\_db2;

DROP DATABASE IF EXISTS app\_db3;

DROP DATABASE IF EXISTS app\_db4;

CREATE DATABASE app\_db1;

CREATE DATABASE app\_db2;

CREATE DATABASE app\_db3;

CREATE DATABASE app\_db4;

Run this script:

bash

CopyEdit

sudo -u postgres psql -f /tmp/setup\_db1\_fixed.sql

**4. Create Tables and Insert Sample Data**

bash

CopyEdit

for db in app\_db1 app\_db2 app\_db3 app\_db4

do

echo "Setting up $db..."

sudo -u postgres psql -d $db <<EOF

DROP TABLE IF EXISTS data\_events, transactions, activity\_logs, system\_info;

CREATE TABLE data\_events (

id SERIAL PRIMARY KEY,

event\_date DATE,

value INT

);

CREATE TABLE transactions (

id SERIAL PRIMARY KEY,

created\_at TIMESTAMP,

amount NUMERIC

);

CREATE TABLE activity\_logs (

id SERIAL PRIMARY KEY,

log\_time TIMESTAMP,

activity TEXT

);

CREATE TABLE system\_info (

id SERIAL PRIMARY KEY,

info TEXT,

status TEXT

);

-- Insert dummy data for 30 days

INSERT INTO data\_events (event\_date, value)

SELECT CURRENT\_DATE - (i || ' days')::interval, ROUND((random()\*100)::numeric)

FROM generate\_series(0,29) AS i;

INSERT INTO transactions (created\_at, amount)

SELECT CURRENT\_DATE - (i || ' days')::interval + time '10:00', ROUND((random()\*1000)::numeric, 2)

FROM generate\_series(0,29) AS i;

INSERT INTO activity\_logs (log\_time, activity)

SELECT CURRENT\_DATE - (i || ' days')::interval + time '15:00', 'Activity ' || i

FROM generate\_series(0,29) AS i;

-- Insert non-date-based data

INSERT INTO system\_info (info, status)

SELECT 'Info ' || i, CASE WHEN i % 2 = 0 THEN 'OK' ELSE 'FAIL' END

FROM generate\_series(1,10) AS i;

EOF

done

**5. Verify Database Setup**

**List all databases:**

bash

CopyEdit

sudo -u postgres psql -c "\l"

**List all tables in each database:**

bash

CopyEdit

for db in app\_db1 app\_db2 app\_db3 app\_db4

do

echo "📂 Tables in $db:"

sudo -u postgres psql -d $db -c "\dt"

done

**6. View Sample Data from All Tables**

bash

CopyEdit

for db in app\_db1 app\_db2 app\_db3 app\_db4

do

echo "🔎 $db: data\_events"

sudo -u postgres psql -d $db -c "SELECT \* FROM data\_events LIMIT 5;"

echo "🔎 $db: transactions"

sudo -u postgres psql -d $db -c "SELECT \* FROM transactions LIMIT 5;"

echo "🔎 $db: activity\_logs"

sudo -u postgres psql -d $db -c "SELECT \* FROM activity\_logs LIMIT 5;"

echo "🔎 $db: system\_info"

sudo -u postgres psql -d $db -c "SELECT \* FROM system\_info LIMIT 5;"

done

**View full system\_info table for each DB:**

bash

CopyEdit

for db in app\_db1 app\_db2 app\_db3 app\_db4

do

echo "🔎 $db: system\_info"

sudo -u postgres psql -d $db -c "SELECT \* FROM system\_info;"

done

**7. Optional File Permission Fix (if script can't be read)**

If you face permission issues running scripts from home directory, either:

**Option 1: Move the file to /tmp (recommended)**

bash

CopyEdit

sudo mv /home/sysadmin/setup\_db1.sql /tmp/

sudo -u postgres psql -f /tmp/setup\_db1.sql

**Option 2: Change permissions (not recommended for production)**

bash

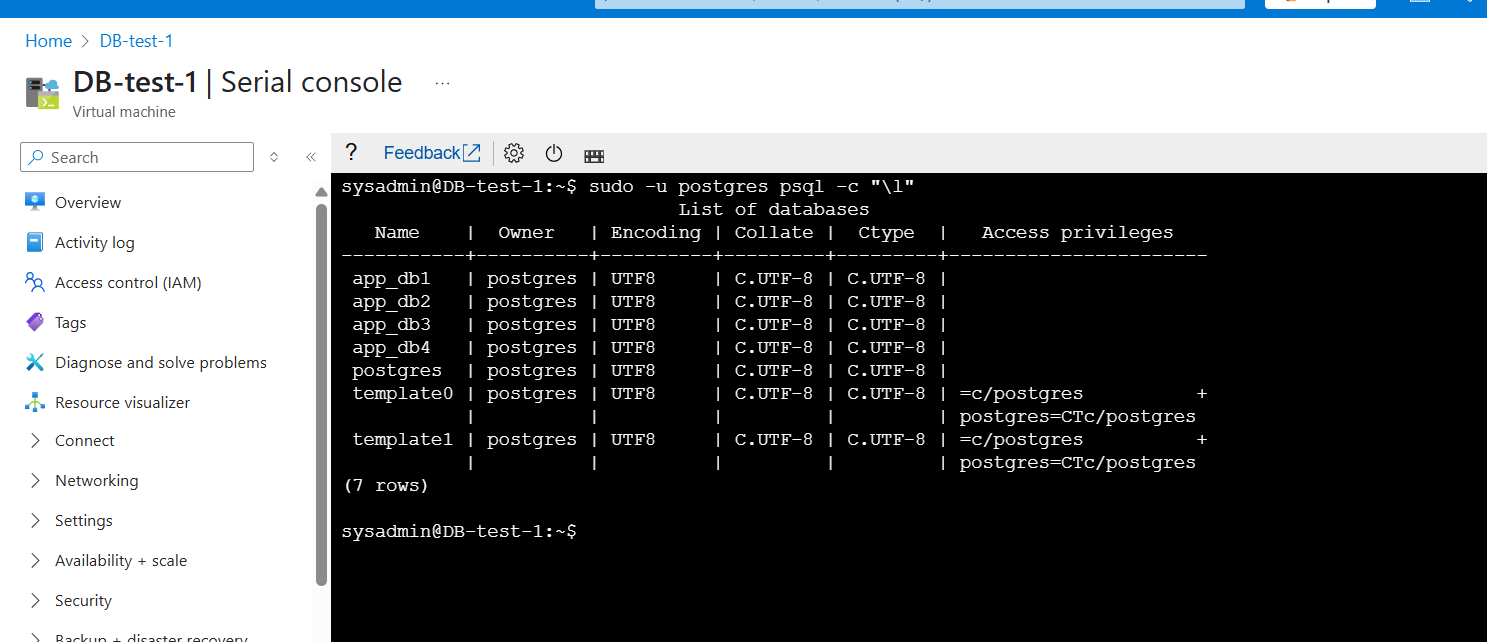
CopyEdit

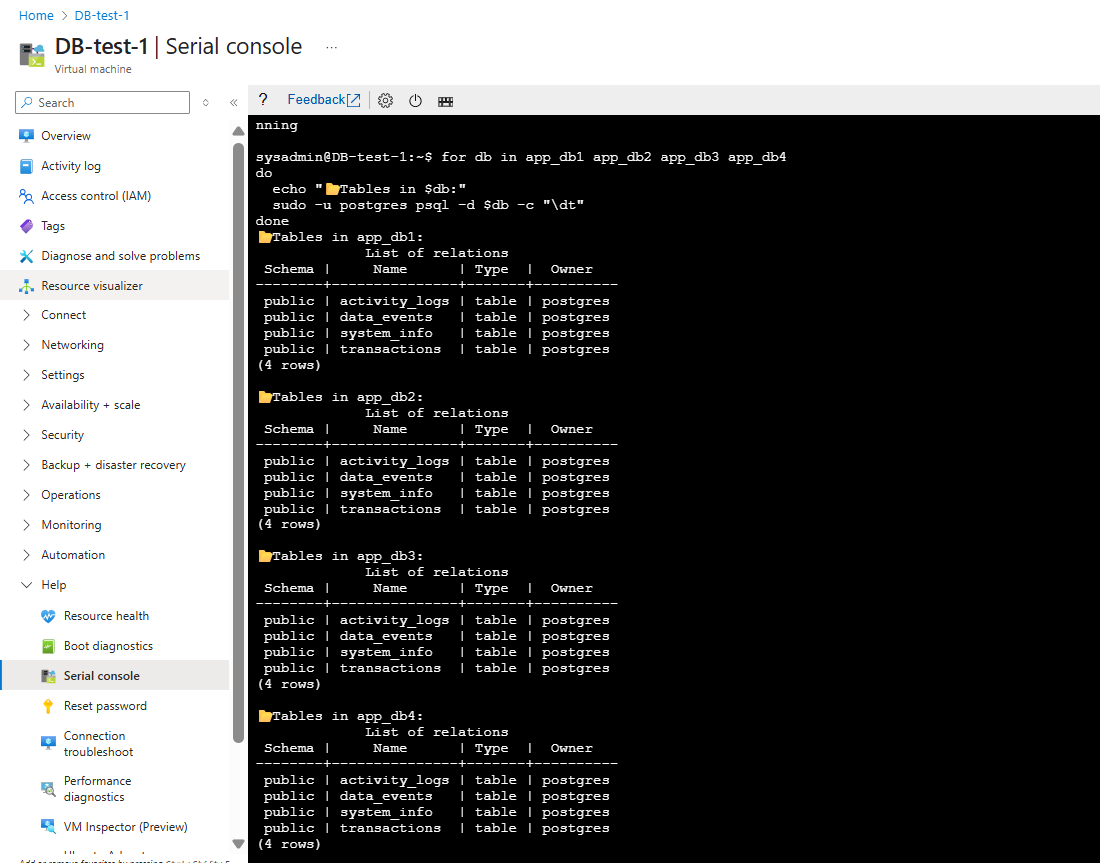
chmod +r /home/sysadmin/setup\_db1.sql

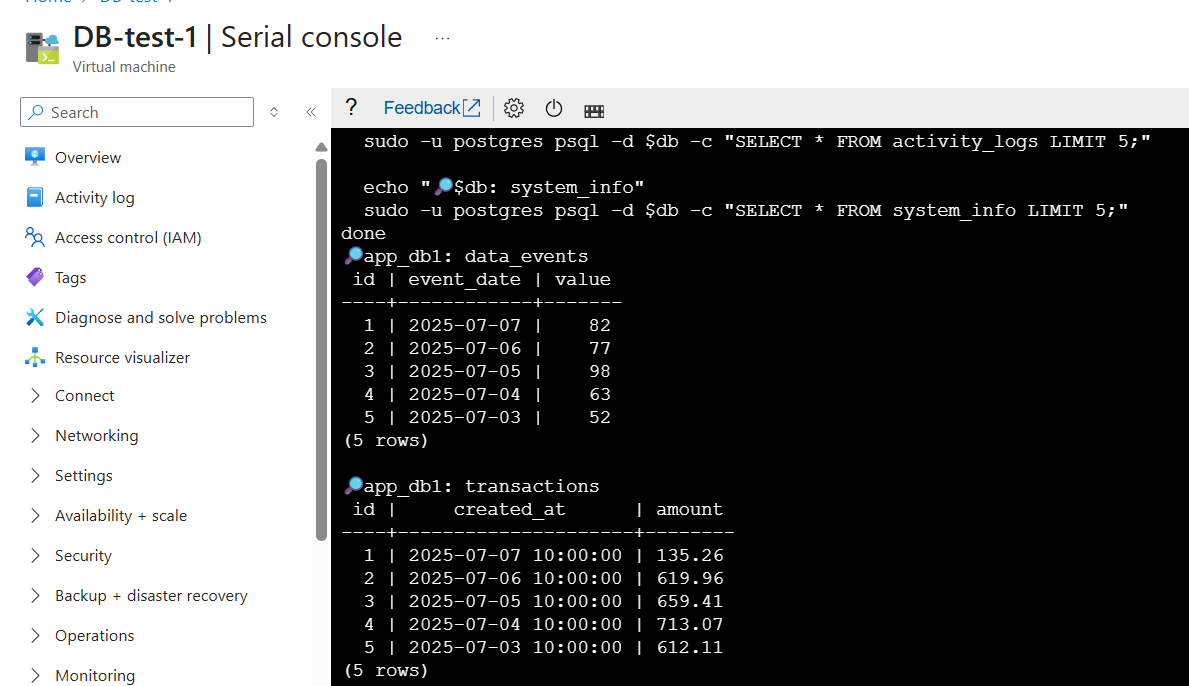
chmod +x /home/sysadmin

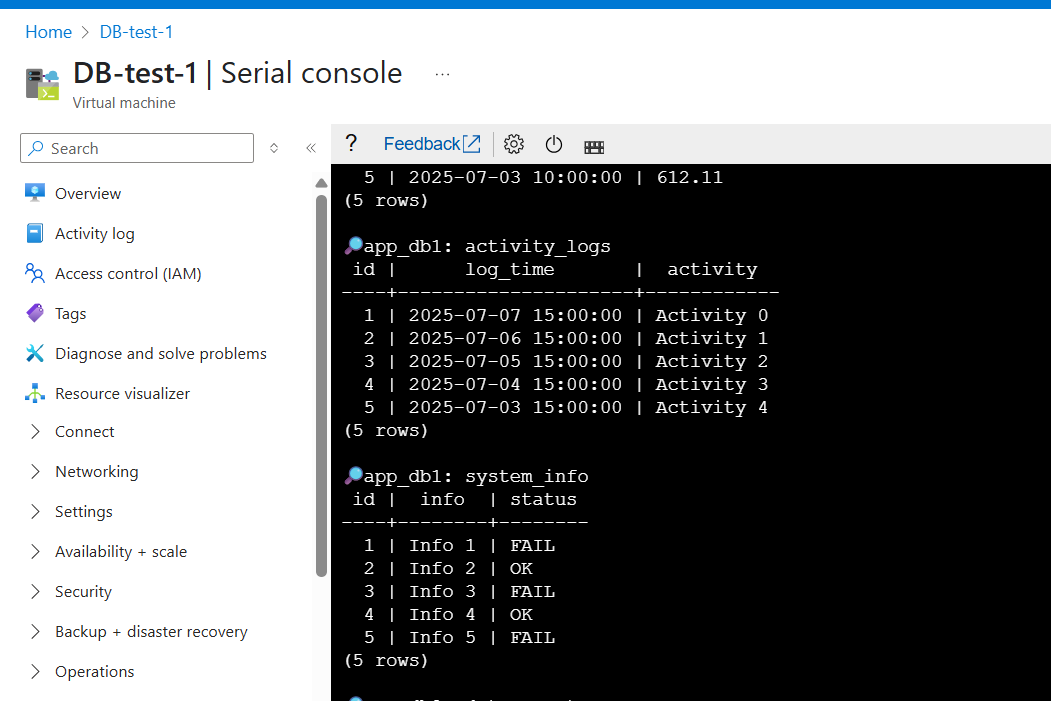
sudo -u postgres psql -f /home/sysadmin/setup\_db1.sql

**Output from DB1:**









**DB2 From Output:**

