



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## EXPERIMENT- 09

**Student Name:** Ashutosh Yadav

**UID:** 23BCS11023

**Branch:** BE-CSE

**Section/Group:** KRG 1-B

**Semester:** 05

**Date of Performance:** 28/10/25

**Subject Name:** ADBMS

**Subject Code:** 23CSP-333

**1. Aim:** To create and connect a PostgreSQL database instance on **Amazon RDS (Relational Database Service)**

### **2. Objective:**

- To understand the steps involved in launching a database instance using Amazon RDS.
- To configure a database for public access and connect it with a local client (pgAdmin).
- To perform basic SQL operations (CREATE, INSERT, SELECT).

### **3. Tools / Software**

- Amazon Web Services (AWS)
- PostgreSQL
- pgAdmin 4
- RDS (Relational Database Service)

### **4. Program:**

Step 1: Create and Configure Database Instance

1. Login to AWS Console → RDS → Create database, select Standard create and PostgreSQL under the Free Tier template.
2. Set DB identifier: ruchi-db, Username: postgre, choose db.t3.micro, 20 GB gp2 storage, and enable Public access.
3. Click Create database and wait until the status shows Available in the RDS dashboard.

The screenshot shows the AWS RDS (Relational Database Service) interface. In the left sidebar, under 'Aurora and RDS', the 'Databases' option is selected. The main area displays a table titled 'Databases (1)'. The table has columns: DB identifier, Status, Role, Engine, Region ..., and Size. One row is present, showing 'ruchi-db' as the DB identifier, 'Config...' as the Status, 'Instance' as the Role, 'PostgreSQL' as the Engine, 'eu-north-1a' as the Region, and 'db.t4g.micro' as the Size. At the top right of the table, there are buttons for 'Group resources', 'Modify', 'Actions', and a yellow 'Create database' button. The top navigation bar includes the AWS logo, a search bar, and account information: Account ID: 0607-9593-0204, Region: Europe (Stockholm), and User: Ruchi%20Sharma.



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## Step 2: Configure Security Group (Allow Local Access Only)

1. In AWS Console → go to RDS → Databases → click your DB .
2. Open the Connectivity & Security tab.
3. Under VPC security groups, click the linked group name (it opens EC2 security groups).
4. Click Edit inbound rules → Add rule
  - Type: PostgreSQL
  - Protocol: TCP
  - Port: 5432
  - Source: My IP
5. Click Save rules.

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-0d39d1bf593210da4	IPv4	PostgreSQL	TCP	5432	106.206.235.43
-	sgr-0ee4f18536cb88772	-	All traffic	All	All	sg-0570f95942

## Step 3: Connect Database Using pgAdmin

1. Open pgAdmin 4 on your local system.
2. Right-click Servers → Create → Server.
3. Under the General tab, enter the name: postgres.
4. Under the Connection tab, fill in the following details:
  - Host name/address: demo-db.xxxxxxx.rds.amazonaws.com
  - Port: 5432
  - Username: postgres
  - Check Save password.
5. Click Save to connect your RDS PostgreSQL database.

```
graph TD; Root[Default Workspace] --> Servers[Servers (2)]; Servers --> PostgreSQL[PostgreSQL 17]; PostgreSQL --> ruchiDb[ruchi-db]; ruchiDb --> Databases[Databases]; ruchiDb --> Roles[Login/Group Roles]; ruchiDb --> Tablespaces[Tablespaces]
```