



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

## EXPERIMENT- 09

**Student Name:** Charu Singla

**UID:** 23BCS12553

**Branch:** BE-CSE

**Section/Group:** KRG 1(A)

**Semester:** 05

**Date of Performance:** 30/10/25

**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: postgres, choose db.t3.micro, 20 GB gp2 storage, and enable Public access.

The screenshot shows the AWS RDS console interface. On the left, there's a sidebar with options like Dashboard, Databases (which is selected), Query editor, Performance insights, and Snapshots. The main area is titled 'Databases (1)' and shows a table with one row. The table columns are DB identifier, Status, Role, Engine, Region..., and Size. The row contains 'ruchi-db', 'Config...', 'Instance', 'PostgreSQL', 'eu-north-1a', and 'db.t4g.micro'. There are buttons for Group resources, Modify, Actions, and Create database.

3. Click Create database and wait until the status shows Available in the RDS dashboard.



# 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 (ruchi-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.

The screenshot shows the AWS VPC Security Groups Inbound Rules table for a security group named 'sg-0570f959421927738 - default'. There are two rules listed:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-0d39d1bf593210da4	IPv4	PostgreSQL	TCP	5432	106.206.235.43, sg-0570f95942
-	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: **postgre**.
4. Under the Connection tab, fill in the following details:
  - Host name/address: **ruchi-db.xxxxxxx.rds.amazonaws.com**
  - Port: **5432**
  - Username: **postgre**
  - Check Save password.
5. Click **Save** to connect your RDS PostgreSQL database.

