Exercise 4: DBeaver and AWS RDS

Duration: ~20 minutes

Objective

Deploy a managed PostgreSQL database on AWS RDS, connect with DBeaver (a professional database tool), and run the HR schema on a cloud database.

Part 1: Install DBeaver (~3 min)

1. Download from dbeaver.io
2. Install and open DBeaver

Part 2: Deploy PostgreSQL on AWS RDS (~10 min)

1. Open your **AWS Academy Learner Lab**
2. Click **Start Lab** and wait for the green indicator
3. Click **AWS** to open the Management Console
4. Navigate to **RDS** (search in the top bar)
5. Click **Create database**
6. Configure:
   - **Engine**: PostgreSQL
   - **Engine version**: PostgreSQL 16.x (latest available)
   - **Templates**: Free tier
   - **DB instance identifier**: `bdi-hr-database`
   - **Master username**: `postgres`
   - **Master password**: choose a password and **write it down**
   - **DB instance class**: `db.t3.micro`
   - **Storage**: 20 GB gp2 (default)
   - **Public access**: **Yes**
   - **VPC security group**: Choose existing -> select the **default** security group
   - **Initial database name**: `hr_database`
7. Click **Create database**
8. Wait ~5 minutes for the instance to become **Available**

Configure Security Group

1. Click on your database instance
2. Under **Connectivity & security**, click the **VPC security group** link
3. Select **Inbound rules** -> **Edit inbound rules**
4. Add rule:
   - **Type**: PostgreSQL
   - **Port**: 5432
   - **Source**: `0.0.0.0/0` (Anywhere IPv4)
5. Click **Save rules**

> ***Note:*** *In AWS Academy Learner Lab, you cannot create new security groups. Use the* ***default*** *security group and add the inbound rule there.*

> ***Warning:*** *In production, restrict access to specific IPs. We use* `0.0.0.0/0` *only for this lab.*

Part 3: Connect DBeaver to RDS (~3 min)

1. Copy the RDS **Endpoint** (e.g., `bdi-hr-database.xxxxx.us-east-1.rds.amazonaws.com`)
2. In DBeaver, click **New Database Connection** (plug icon)
3. Select **PostgreSQL** -> **Next**
4. Fill in:
   - **Host**: your RDS endpoint
   - **Port**: `5432`
   - **Database**: `hr_database`
   - **Username**: `postgres`
   - **Password**: your RDS password
5. Click **Test Connection** (install driver if prompted)
6. Click **Finish**

Part 4: Create the HR Schema on RDS (~4 min)

1. In DBeaver, open a new SQL editor (`Ctrl+]`)
2. Open and execute `hr_schema.sql`
3. Execute `hr_seed_data.sql`
4. Verify:

```sql
SELECT table_name FROM information_schema.tables
WHERE table_schema = 'public' ORDER BY table_name;

SELECT COUNT(*) FROM employee;
```

5. Run some of the KPI queries from Exercise 2 to verify everything works

Part 5: Connect the Python Scripts to RDS

Update the connection URL in the scripts or pass it directly:

```
# Update SETTINGS["postgresql"]["url"] in the scripts to your RDS endpoint, or:
export
DATABASE_URL="postgresql://postgres:YOUR_PASSWORD@YOUR_RDS_ENDPOINT:5432/hr_database"
```

Deliverable

- Screenshot of DBeaver connected to RDS showing the 5 tables
- Screenshot of a KPI query result in DBeaver