

AWS RDS Hands-On Practice

Introduction

A **database** is a structured collection of data that can be accessed, managed, and updated efficiently. In this hands-on activity, we focus on using **Amazon RDS (Relational Database Service)** to create and manage a MySQL database, connect it from an EC2 instance, and perform various SQL operations.

This practice covers:

- Creating RDS and EC2 resources
- Establishing a secure connection
- Performing SQL operations
- Backup and restore
- User creation and privilege management

Database Types

◆ Relational Databases (SQL)

Examples:

- MySQL
- MariaDB
- PostgreSQL
- Aurora

◆ Non-Relational Databases (NoSQL)

Examples:

- DynamoDB
- MongoDB

Default Databases in MySQL

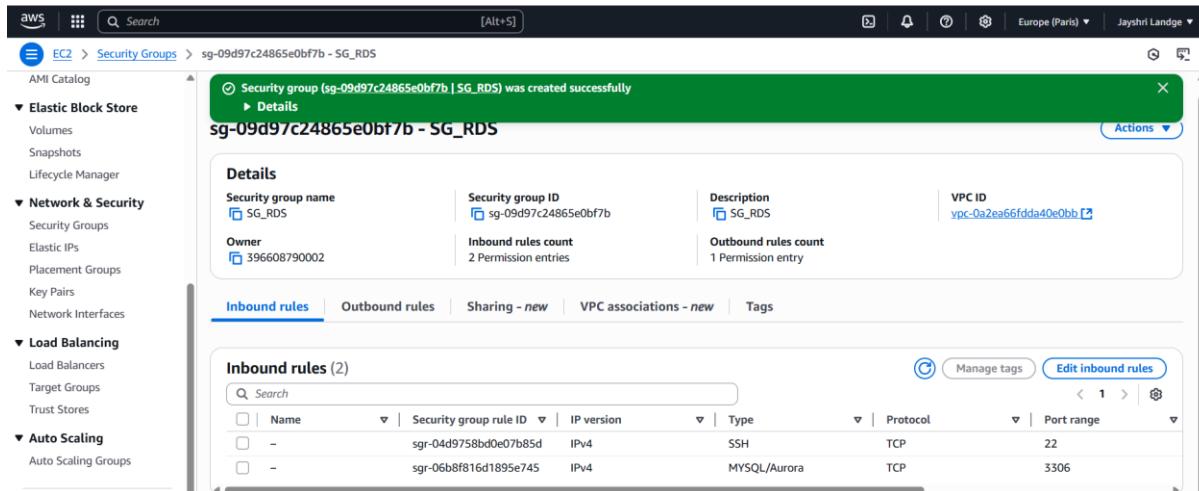
- information_schema
- performance_schema
- mysql
- sys

RDS Setup and Execution Steps:

Step 1: Create a Security Group

Create a security group with the following rules:

- **Port 22** – for SSH access
- **Port 3306** – for MySQL access



Screenshot 1: Security Group Creation

Step 2: Launch EC2 Linux Instance

- Launch a Linux-based EC2 instance
- Attach the security group created in Step 1

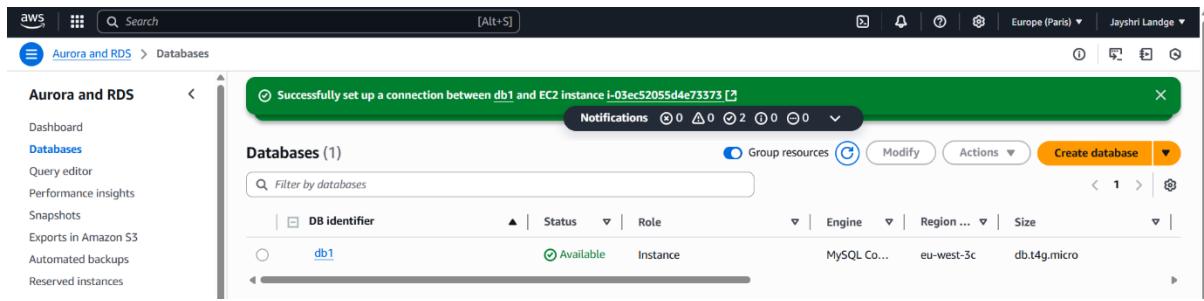


Screenshot 2: Instance Launch

Step 3: Create an RDS MySQL Database

Go to the RDS service and:

- Choose **MySQL** as the engine
- Configure instance settings
- Enable **public accessibility**
- Use the same security group



Screenshot 3: RDS Instance Creation

Step 4: Connect to EC2 Instance

Use SSH to connect to your EC2 Linux instance:

```
ssh -i your-key.pem ec2-user@your-ec2-public-ip
```

```
[ec2-user@ip-172-31-47-77:~]
[ec2-user@ip-172-31-47-77:~] $ ssh -i your-key.pem ec2-user@your-ec2-public-ip
[ec2-user@ip-172-31-47-77:~]$
```

The terminal session shows a successful SSH login to an Amazon Linux 2 instance. It displays system information, including the end-of-life date (2026-06-30) and a link to the latest version of Amazon Linux.

Screenshot 4: EC2 SSH Connection

Step 5: Install MySQL Client on EC2

```
sudo yum install -y mysql
```

```
[ec2-user@ip-172-31-47-77:~]$ sudo yum install -y mysql
[ec2-user@ip-172-31-47-77:~]$
```

The terminal session shows the execution of the command "sudo yum install -y mysql", which installs the MySQL client package on the EC2 instance.

Screenshot 5: MySQL Installation

Step 6: Connect to the RDS MySQL Database

```
mysql -h <endpoint> -u <username> -p
```

Example:

```
mysql -h db1.c3k0yseq8slt.eu-west-3.rds.amazonaws.com -u admin -padmin123
```

```
[root@ip-172-31-40-206 ~]# mysql -h db1.c3k0yseq8slt.eu-west-3.rds.amazonaws.com -u admin -padmin123
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 77
Server version: 8.0.41 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Screenshot 6: MySQL RDS Connection

SQL Commands and Practice

◆ View All Databases

```
show databases;
```

```
MySQL [(none)]> show databases;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+
4 rows in set (0.00 sec)
```

Screenshot 7: Default Databases

◆ Create a New Database

```
create database db1;
```

```
MySQL [(none)]> create database db1;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> show databases;
+-----+
| Database      |
+-----+
| db1           |
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+
5 rows in set (0.00 sec)
```

Screenshot 8: Create database

◆ Use a Database

```
use db1; (OR)
```

```
connect db1;
```

```
MySQL [(none)]> connect db1;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Connection id: 50
Current database: db1
```

Screenshot 9: Connect to database

◆ Create Employee Table

```
create table Employee(employee_id int(5), name varchar(10), designation varchar(15),
department varchar(20), salary int(10), email_id varchar(30), phone_no int(12), city
varchar(15));
```

```
MySQL [db1]> create table company_employee(employee_id int(5), name varchar(10), designation varchar(15), department varchar(20), salary int(10), email_id varchar(30), phone_no int(12), city
varchar(15));
Query OK, 0 rows affected, 3 warnings (0.05 sec)

MySQL [db1]> show tables;
+-----+
| Tables_in_db |
+-----+
| company_employee |
+-----+
1 row in set (0.02 sec)

MySQL [db1]> desc company_employee;
+-----+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| employee_id | int(5) | YES | NO | NULL | |
| name | varchar(10) | YES | NO | NULL | |
| designation | varchar(15) | YES | NO | NULL | |
| department | varchar(20) | YES | NO | NULL | |
| salary | int(10) | YES | NO | NULL | |
| email_id | varchar(30) | YES | NO | NULL | |
| phone_no | int(12) | YES | NO | NULL | |
| city | varchar(15) | YES | NO | NULL | |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.01 sec)
```

Screenshot 10: Table Creation

◆ Insert Records

```
insert into Employee values
```

```
(1, 'Rohan', 'Developer', 'IT', 60000, 'rohan@gmail.com', 9876543210, 'Pune'),
```

```
(2, 'Neha', 'HR', 'HR', 55000, 'neha@gmail.com', 9123456780, 'Mumbai'),
```

```
...;
```

```
MySQL [db1]> INSERT INTO Employee
-> VALUES
-> (1, 'Rohan', 'Developer', 'IT', 60000, 'rohan@gmail.com', 9876543210, 'Pune'),
-> (2, 'Neha', 'HR', 'HR', 55000, 'neha@gmail.com', 9123456780, 'Mumbai'),
-> (3, 'Ravi', 'Analyst', 'Finance', 58000, 'ravi@gmail.com', 9990776655, 'Mumbai'),
-> (4, 'Sneha', 'Team Lead', 'IT', 80000, 'sneha@gmail.com', 9090909090, 'Aurangabad'),
-> (5, 'Sai', 'Salesman', 'Sales', 45000, 'sai@gmail.com', 9899776655, 'Nagpur'),
-> (6, 'Roshani', 'Accountant', 'Finance', 52000, 'roshani@gmail.com', 9898999999, 'Pune'),
-> (7, 'Rahul', 'DevOps Engg', 'IT', 67000, 'rahul@gmail.com', 9777777777, 'Nagpur'),
-> (8, 'Mayur', 'Intern', 'IT', 15000, 'mayur@gmail.com', 9666666666, 'Pune');
Query OK, 8 rows affected, 8 warnings (0.00 sec)
Records: 8 Duplicates: 0 Warnings: 8

MySQL [db1]>
MySQL [db1]> select * from Employee;
+-----+-----+-----+-----+-----+-----+-----+
| employee_id | name | designation | department | salary | email_id | phone_no | city |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Rohan | Developer | IT | 60000 | rohan@gmail.com | 2147483647 | Pune |
| 2 | Neha | HR | HR | 55000 | neha@gmail.com | 2147483647 | Mumbai |
| 3 | Ravi | Analyst | Finance | 58000 | ravi@gmail.com | 2147483647 | Mumbai |
| 4 | Sneha | Team Lead | IT | 80000 | sneha@gmail.com | 2147483647 | Aurangabad |
| 5 | Sai | Salesman | Sales | 45000 | sai@gmail.com | 2147483647 | Nagpur |
| 6 | Roshani | Accountant | Finance | 52000 | roshani@gmail.com | 2147483647 | Pune |
| 7 | Rahul | DevOps Engg | IT | 67000 | rahul@gmail.com | 2147483647 | Nagpur |
| 8 | Mayur | Intern | IT | 15000 | mayur@gmail.com | 2147483647 | Pune |
+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

Screenshot 11: Insert Data

◆ Queries for Practice

- View maximum salary:

```
select max(salary) from Employee;
```

```
MySQL [db1]> select max(salary) from Employee;
+-----+
| max(salary) |
+-----+
|      80000 |
+-----+
1 row in set (0.00 sec)
```

Screenshot 12: Max function

- View minimum salary:

```
select min(salary) from Employee;
```

```
MySQL [db1]> select min(salary) from Employee;
+-----+
| min(salary) |
+-----+
|      15000 |
+-----+
1 row in set (0.00 sec)
```

Screenshot 13: Min Function

- Average and sum:

```
select avg(salary), sum(salary) from Employee;
```

```
MySQL [db1]> select sum(salary) from company_employee;
+-----+
| sum(salary) |
+-----+
|      432000 |
+-----+
1 row in set (0.00 sec)

MySQL [db1]> select avg(salary) from company_employee;
+-----+
| avg(salary) |
+-----+
| 54000.0000 |
+-----+
1 row in set (0.00 sec)
```

Screenshot 14: Average and sum Function

- Filtered queries:

```
select * from Employee where city='Mumbai';
```

```
MySQL [db1]> select * from Employee where city="Mumbai";
+-----+-----+-----+-----+-----+-----+-----+
| employee_id | name | designation | department | salary | email_id | phone_no | city |
+-----+-----+-----+-----+-----+-----+-----+
|          2 | Neha | HR           | HR          | 55000 | neha@gmail.com | 2147483647 | Mumbai |
|          3 | Ravi | Analyst       | Finance     | 58000 | ravi@gmail.com | 2147483647 | Mumbai |
+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

```
select name, city from Employee where department='Finance';
```

```
MySQL [db1]> select name from Employee where department="Finance";
+-----+
| name   |
+-----+
| Ravi   |
| Roshani|
+-----+
2 rows in set (0.00 sec)
```

Screenshot 15: Where Clause Query Output

```
select * from Employee where city="Pune" or name="Neha";
```

```
MySQL [db1]> select * from Employee where city="Pune" or name="Neha";
+-----+-----+-----+-----+-----+-----+-----+-----+
| employee_id | name      | designation | department | salary | email_id      | phone_no | city    |
+-----+-----+-----+-----+-----+-----+-----+-----+
|       1 | Rohan    | Developer   | IT          | 60000 | rohan@gmail.com | 2147483647 | Pune   |
|       2 | Neha     | HR          | HR          | 55000 | neha@gmail.com  | 2147483647 | Mumbai |
|       6 | Roshani   | Accountant | Finance    | 52000 | roshani@gmail.com| 2147483647 | Pune   |
|       8 | Mayur    | Intern     | IT          | 15000 | mayur@gmail.com | 2147483647 | Pune   |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Screenshot 16: Or Operator

```
select * from Employee where not city="Pune";
```

```
MySQL [db1]> select * from Employee where not city="Pune";
+-----+-----+-----+-----+-----+-----+-----+-----+
| employee_id | name      | designation | department | salary | email_id      | phone_no | city    |
+-----+-----+-----+-----+-----+-----+-----+-----+
|       2 | Neha     | HR          | HR          | 55000 | neha@gmail.com | 2147483647 | Mumbai |
|       3 | Ravi     | Analyst    | Finance    | 58000 | ravi@gmail.com | 2147483647 | Mumbai |
|       4 | Sneha    | Team Lead  | IT          | 80000 | sneha@gmail.com | 2147483647 | Aurangabad |
|       5 | Sai       | Salesman   | Sales      | 45000 | sai@gmail.com  | 2147483647 | Nagpur |
|       7 | Rahul    | DevOps Engg | IT          | 67000 | rahul@gmail.com | 2147483647 | Nagpur |
+-----+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Screenshot 17: Not Operator

```
select * from Employee where name="Rohan" and department="IT";
```

```
MySQL [db1]> select * from Employee where name="Rohan" and department="IT";
+-----+-----+-----+-----+-----+-----+-----+
| employee_id | name      | designation | department | salary | email_id      | phone_no | city    |
+-----+-----+-----+-----+-----+-----+-----+
|       1 | Rohan    | Developer   | IT          | 60000 | rohan@gmail.com | 2147483647 | Pune   |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Screenshot 18 : And Operator

- Update & delete:

```
update Employee set name='Shruti' where employee_id=2;
```

```
MySQL [db1]> update Employee set name="Shruti" where employee_id=2;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MySQL [db1]> select * from Employee;
+-----+-----+-----+-----+-----+-----+-----+
| employee_id | name | designation | department | salary | email_id | phone_no | city |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Rohan | Developer | IT | 60000 | rohan@gmail.com | 2147483647 | Pune |
| 2 | Shruti | HR | HR | 55000 | neha@gmail.com | 2147483647 | Mumbai |
| 3 | Ravi | Analyst | Finance | 58000 | ravi@gmail.com | 2147483647 | Mumbai |
| 4 | Sneha | Team Lead | IT | 80000 | sneha@gmail.com | 2147483647 | Aurangabad |
| 5 | Sai | Salesman | Sales | 45000 | sai@gmail.com | 2147483647 | Nagpur |
| 6 | Roshani | Accountant | Finance | 52000 | roshani@gmail.com | 2147483647 | Pune |
| 7 | Rahul | DevOps Engg | IT | 67000 | rahul@gmail.com | 2147483647 | Nagpur |
| 8 | Mayur | Intern | IT | 15000 | mayur@gmail.com | 2147483647 | Pune |
+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

Screenshot 19: Update The Table

```
alter table Employee add column experience_years int(50);
```

```
MySQL [db1]> alter table Employee add column experience_years int(50);
Query OK, 0 rows affected, 1 warning (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 1

MySQL [db1]> select * from Employee;
+-----+-----+-----+-----+-----+-----+-----+-----+
| employee_id | name | designation | department | salary | email_id | phone_no | city | experience_years |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Rohan | Developer | IT | 60000 | rohan@gmail.com | 2147483647 | Pune | NULL |
| 2 | Shruti | HR | HR | 55000 | neha@gmail.com | 2147483647 | Mumbai | NULL |
| 3 | Ravi | Analyst | Finance | 58000 | ravi@gmail.com | 2147483647 | Mumbai | NULL |
| 4 | Sneha | Team Lead | IT | 80000 | sneha@gmail.com | 2147483647 | Aurangabad | NULL |
| 5 | Sai | Salesman | Sales | 45000 | sai@gmail.com | 2147483647 | Nagpur | NULL |
| 6 | Roshani | Accountant | Finance | 52000 | roshani@gmail.com | 2147483647 | Pune | NULL |
| 7 | Rahul | DevOps Engg | IT | 67000 | rahul@gmail.com | 2147483647 | Nagpur | NULL |
| 8 | Mayur | Intern | IT | 15000 | mayur@gmail.com | 2147483647 | Pune | NULL |
+-----+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

Screenshot 20: Add The Column

```
alter table Employee drop column experience_years;
```

```
MySQL [db1]> alter table Employee drop column experience_years;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0

MySQL [db1]> select * from Employee;
+-----+-----+-----+-----+-----+-----+-----+
| employee_id | name | designation | department | salary | email_id | phone_no | city |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Rohan | Developer | IT | 60000 | rohan@gmail.com | 2147483647 | Pune |
| 2 | Shruti | HR | HR | 55000 | neha@gmail.com | 2147483647 | Mumbai |
| 3 | Ravi | Analyst | Finance | 58000 | ravi@gmail.com | 2147483647 | Mumbai |
| 4 | Sneha | Team Lead | IT | 80000 | sneha@gmail.com | 2147483647 | Aurangabad |
| 5 | Sai | Salesman | Sales | 45000 | sai@gmail.com | 2147483647 | Nagpur |
| 6 | Roshani | Accountant | Finance | 52000 | roshani@gmail.com | 2147483647 | Pune |
| 7 | Rahul | DevOps Engg | IT | 67000 | rahul@gmail.com | 2147483647 | Nagpur |
| 8 | Mayur | Intern | IT | 15000 | mayur@gmail.com | 2147483647 | Pune |
+-----+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

Screenshot 21: Drop The Column

```
delete from Employee where employee_id=8;
```

```
MySQL [db1]> delete from Employee where employee_id=8;
Query OK, 1 row affected (0.00 sec)

MySQL [db1]> select * from Employee;
+-----+-----+-----+-----+-----+-----+-----+
| employee_id | name | designation | department | salary | email_id | phone_no | city |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Rohan | Developer | IT | 60000 | rohan@gmail.com | 2147483647 | Pune |
| 2 | Shruti | HR | HR | 55000 | neha@gmail.com | 2147483647 | Mumbai |
| 3 | Ravi | Analyst | Finance | 58000 | ravi@gmail.com | 2147483647 | Mumbai |
| 4 | Sneha | Team Lead | IT | 80000 | sneha@gmail.com | 2147483647 | Aurangabad |
| 5 | Sai | Salesman | Sales | 45000 | sai@gmail.com | 2147483647 | Nagpur |
| 6 | Roshani | Accountant | Finance | 52000 | roshani@gmail.com | 2147483647 | Pune |
| 7 | Rahul | DevOps Engg | IT | 67000 | rahul@gmail.com | 2147483647 | Nagpur |
+-----+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

Screenshot 22: Query Outputs

Backup and Restore

◆ Backup with mysqldump

```
mysqldump -h <endpoint> db1 > /file1 -u admin -padmin123
```

```
[root@ip-172-31-43-24 ~]# mysqldump -h db1.c3k0yseq8slt.eu-west-3.rds.amazonaws.com db1 > /file1 -u admin -padmin123
[root@ip-172-31-43-24 ~]# cat /file1
-- MySQL dump 10.14 Distrib 5.5.68-MariaDB, for Linux (x86_64)
--
-- Host: db1.c3k0yseq8slt.eu-west-3.rds.amazonaws.com Database: db1
-- 
-- Server version 8.0.41

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;
/*!40103 SET @OLD_TIME_ZONE=@@TIME_ZONE */;
/*!40103 SET TIME_ZONE='+00:00' */;
/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40101 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

-- Table structure for table 'Employee'

DROP TABLE IF EXISTS `Employee`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `Employee` (
  `employee_id` int DEFAULT NULL,
  `name` varchar(10) DEFAULT NULL,
  `designation` varchar(15) DEFAULT NULL,
  `department` varchar(20) DEFAULT NULL,
  `salary` int DEFAULT NULL,
  `email_id` varchar(20) DEFAULT NULL,
  `phone_no` int DEFAULT NULL,
  `city` varchar(15) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

-- Dumping data for table 'Employee'
--
```

Screenshot 23: Take a Backup

◆ Restore

```
mysql -h <endpoint> db1 < /file1 -u admin -padmin123
```

```
7 mysql -h db1.c3k0yseq8slt.eu-west-3.rds.amazonaws.com db1 < /file1 -u admin -padmin123

MySQL [(none)]> connect db1;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Connection id: 50
Current database: db1

MySQL [db1]> show tables;
+-----+
| Tables_in_db1 |
+-----+
| Employee |
+-----+
1 row in set (0.00 sec)

MySQL [db1]> select * from Employee;
+-----+-----+-----+-----+-----+-----+-----+-----+
| employee_id | name | designation | department | salary | email_id | phone_no | city |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Rohan | Developer | IT | 60000 | rohan@gmail.com | 2147483647 | Pune |
| 2 | Shruti | HR | HR | 55000 | neha@gmail.com | 2147483647 | Mumbai |
| 3 | Ravi | Analyst | Finance | 58000 | ravi@gmail.com | 2147483647 | Mumbai |
| 4 | Sneha | Team Lead | IT | 80000 | sneha@gmail.com | 2147483647 | Aurangabad |
| 5 | Sai | Salesman | Sales | 45000 | sai@gmail.com | 2147483647 | Nagpur |
| 6 | Roshani | Accountant | Finance | 52000 | roshani@gmail.com | 2147483647 | Pune |
| 7 | Rahul | DevOps Engg | IT | 67000 | rahul@gmail.com | 2147483647 | Nagpur |
+-----+-----+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

Screenshot 24: Restore Backup

◆ Automate Backup with Cron

```
05 14 * * * mysqldump -h <endpoint> db1 > /file1 -u admin -padmin123
```

```
[root@ip-172-31-43-24 ~]# date
Thu Jun 26 14:00:18 UTC 2025
[root@ip-172-31-43-24 ~]# crontab -e
no crontab for root - using an empty one
crontab: installing new crontab
[root@ip-172-31-43-24 ~]# date
Thu Jun 26 14:02:45 UTC 2025
[root@ip-172-31-43-24 ~]# crontab -l
05 14 * * * mysqldump -h db1.c3k0yseq8slt.eu-west-3.rds.amazonaws.com db1 > /file1 -u admin -padmin123

[root@ip-172-31-43-24 ~]# cat /file1
-- MySQL dump 10.14 Distrib 5.5.68-MariaDB, for Linux (x86_64)
--
-- Host: db1.c3k0yseq8slt.eu-west-3.rds.amazonaws.com Database: db1
-- 
-- Server version 8.0.41
```

Screenshot 25: Backup & Cron Setup

User Management in MySQL

◆ View Users

```
select user from mysql.user;
```

```
MySQL [(none)]> select user from mysql.user;
+-----+
| user |
+-----+
| admin |
| rds_superuser_role |
| rohan |
| rohan'%
| sagar |
| mysql.infoschema |
| mysql.session |
| mysql.sys |
| rdsadmin |
+-----+
9 rows in set (0.00 sec)
```

Screenshot 26: Show User's

◆ View Privileges

```
show grants for 'rohan';
```

```
MySQL [(none)]> show grants for 'rohan';
+-----+
| Grants for rohan@% |
+-----+
| GRANT USAGE ON *.* TO `rohan`@`%` |
| GRANT SELECT, INSERT, UPDATE, DROP, ALTER ON `db1`.* TO `rohan`@`%` |
| GRANT SELECT, INSERT, UPDATE, DROP, ALTER ON `db2`.* TO `rohan`@`%` |
+-----+
3 rows in set (0.00 sec)
```

Screenshot 27: Show Grants For User

◆ Create User

```
create user 'rohan'@'%' identified by 'rohan@50';
```

```
MySQL [(none)]> create user 'rohan'@'%' identified by 'rohan@50';
Query OK, 0 rows affected (0.01 sec)
```

Screenshot 28: User Creation

◆ Grant Privileges

```
grant select, insert, update, alter, drop on db1.* to 'rohan'@'%';
```

```
flush privileges;
```

```
MySQL [(none)]> grant insert on db1.* to 'rohan'@'%';
Query OK, 0 rows affected (0.00 sec)

MySQL [(none)]> flush privileges;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
MySQL [(none)]> flush privileges;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds
MySQL [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MySQL [(none)]> grant insert on db1.* to 'rohan'@'%';
Query OK, 0 rows affected (0.00 sec)

MySQL [(none)]> grant update on db1.* to 'rohan'@'%';
Query OK, 0 rows affected (0.01 sec)

MySQL [(none)]> grant alter on db1.* to 'rohan'@'%';
Query OK, 0 rows affected (0.00 sec)

MySQL [(none)]> grant drop on db1.* to 'rohan'@'%';
Query OK, 0 rows affected (0.00 sec)

MySQL [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

Screenshot 29: Grant Privileges

◆ Grant All Privileges

```
grant all privileges on db1.* to 'sai'@'%';
flush privileges;
```

```
[root@ip-172-31-40-206 ~]# mysql -h db1.c3k0yseq8slt.eu-west-3.rds.amazonaws.com -u admin -padmin123
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 77
Server version: 8.0.41 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> create user 'sai'@'%' identified by 'sai@50';
Query OK, 0 rows affected (0.01 sec)

MySQL [(none)]> grant all privileges on db1.* to 'sai'@'%';
Query OK, 0 rows affected (0.00 sec)

MySQL [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

Screenshot 30: Grant All Privileges

◆ Delete User

```
drop user 'rohan';
```

```
MySQL [(none)]> drop user 'rohan';
Query OK, 0 rows affected (0.01 sec)

MySQL [(none)]> select user from mysql.user;
+-----+
| user      |
+-----+
| admin     |
| rds_superuser_role |
| rohan'%   |
| sagar     |
| sai       |
| mysql.infoschema |
| mysql.session |
| mysql.sys    |
| rdsadmin   |
+-----+
9 rows in set (0.00 sec)
```

Screenshot 31: Delete User

Conclusion

This hands-on practice with **Amazon RDS (MySQL)** covered the essential tasks of setting up cloud databases, connecting from EC2, executing SQL queries, performing backups, and managing user permissions. Through this exercise, a solid foundational understanding of managing databases in the cloud was achieved.