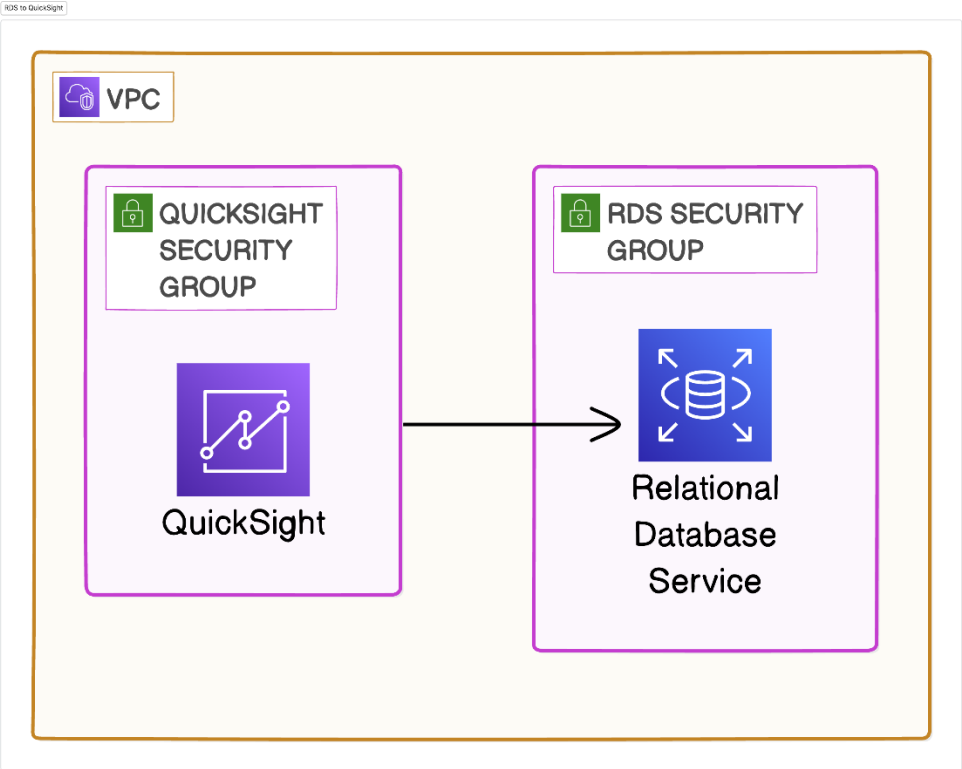
Visualise a Relational Database



Introducing Today's Project!

What is Amazon RDS?

Amazon RDS is a fully managed relational database service by AWS, offering easy setup, scalability, high availability, automated backups, and security, simplifying database management for applications.

How I used Amazon RDS in this project

In today's project, I have used Amazon RDS to set up a managed database, ensuring scalability, security, automated backups, and easy management, enabling efficient data storage for the application.

One thing I didn't expect in this project was...

One thing I didn't expect in this project is how complex managing network interfaces (ENIs) can be, especially when they remain attached to resources like Quick Sight, even after termination.

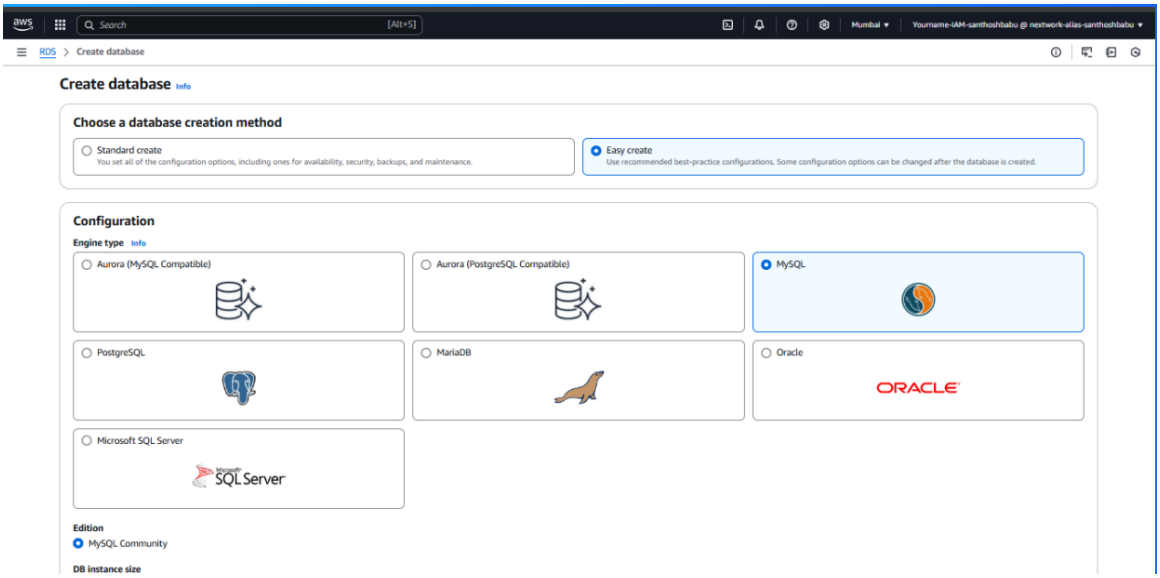
This project took me...

This project took me almost 2 hours to complete.

In the first part of my project...

Creating a Relational Database

I created my relational database by RDS console -> Databases -> Create Database -> configure as needed -> Create Database.



Understanding Relational Databases

A relational database is a system that stores data in tables with rows and columns, linked by keys. It uses SQL for querying and managing data, ensuring efficient organization and retrieval.

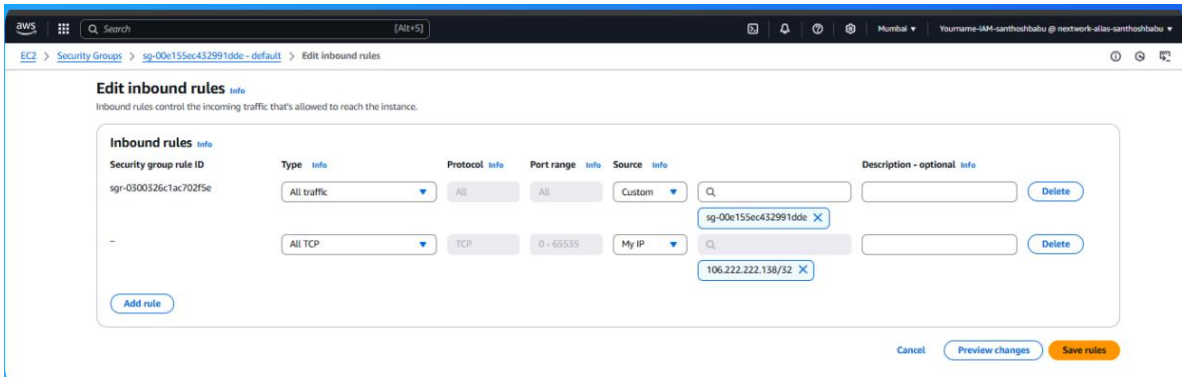
MySQL vs SQL

The difference between MySQL and SQL is SQL (Structured Query Language) is a standard language for managing and querying relational databases. MySQL is an open-source relational database management system (RDBMS) that uses SQL for querying.

Populating my RDS instance

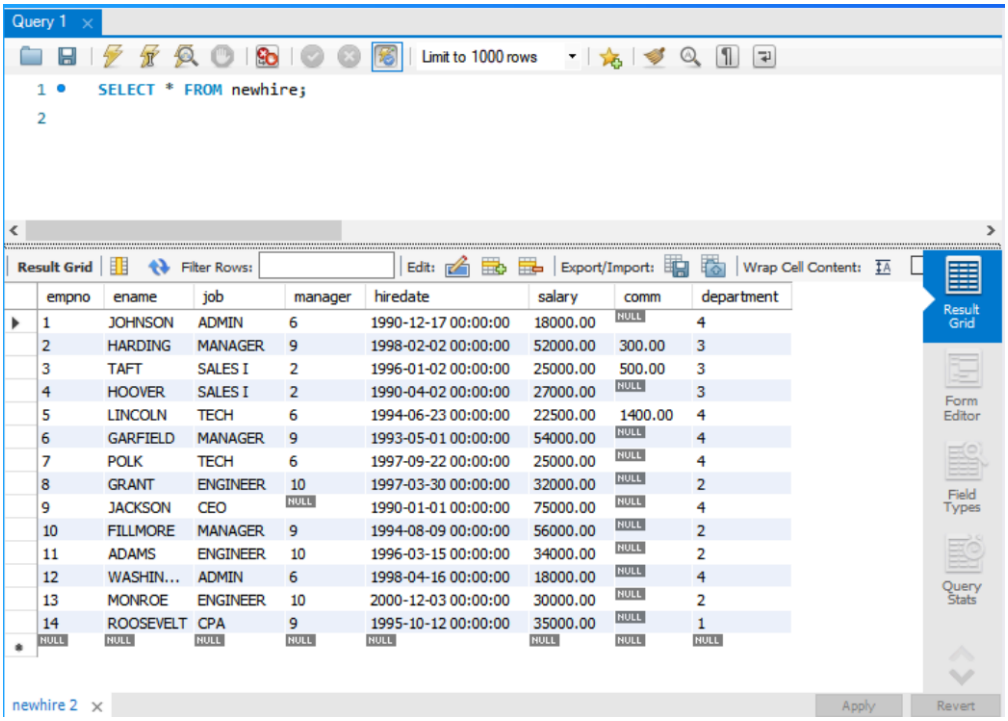
The first thing I did was make my RDS instance public because I needed external access for my application, testing, or third-party integrations. However, I ensured strong security measures were in place.

I had to update the default security group for my RDS schema because it was necessary to allow specific inbound and outbound traffic for the application or services to access the database securely.



Using MySQL Workbench

To populate my database, I created a schema and then created tables and then populated database using SQL queries.



Connecting Quick Sight and RDS

To connect my RDS instance to Quick Sight I navigated to Amazon Quick Sight - > Datasets -> New dataset -> select RDS -> configure the settings -> Validate connection -> Create data source.

This solution is risky because anyone can access it, making us vulnerable from hackers and malicious people trying to get our data.

A better strategy

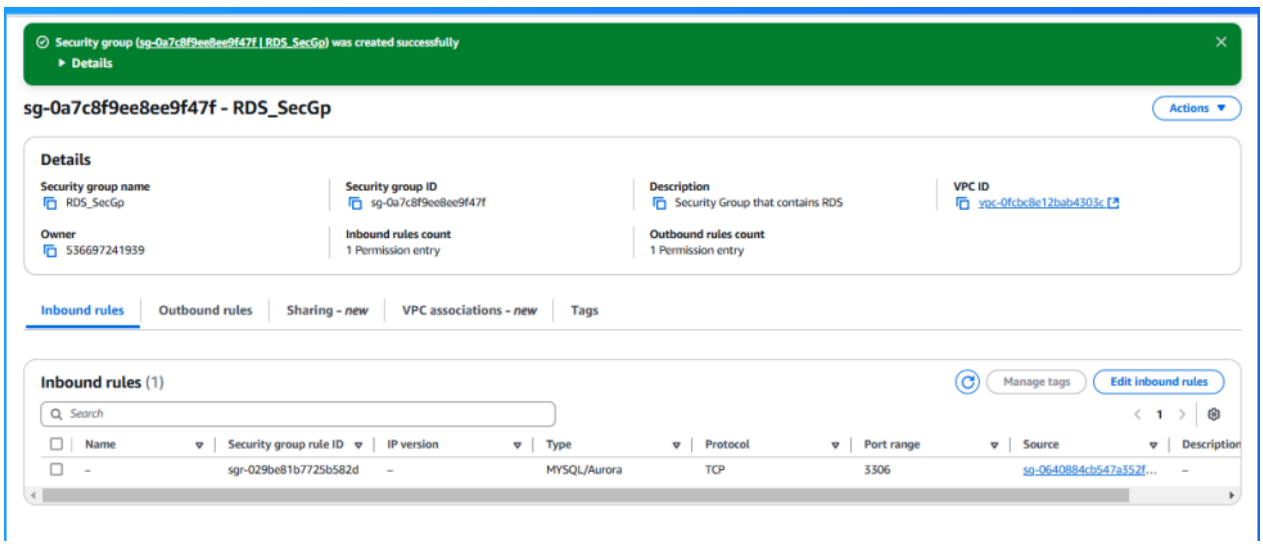
First, I made a new security group so that we can attach Quick Sight to our new security group.

Next, I connected my new security group to Quick Sight by updating the RDS security group to allow inbound traffic from Quick Sight's IP range or VPC. I ensured proper VPC and subnet access and confirmed the security group was associated with the RDS.

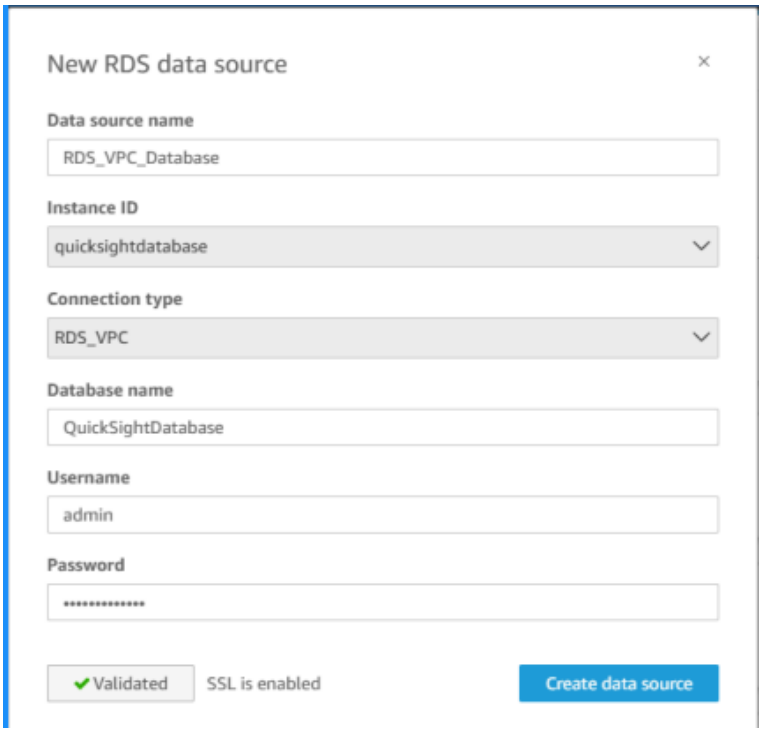
Now to secure my RDS instance

To make my RDS instance secure I created a security group for RDS.

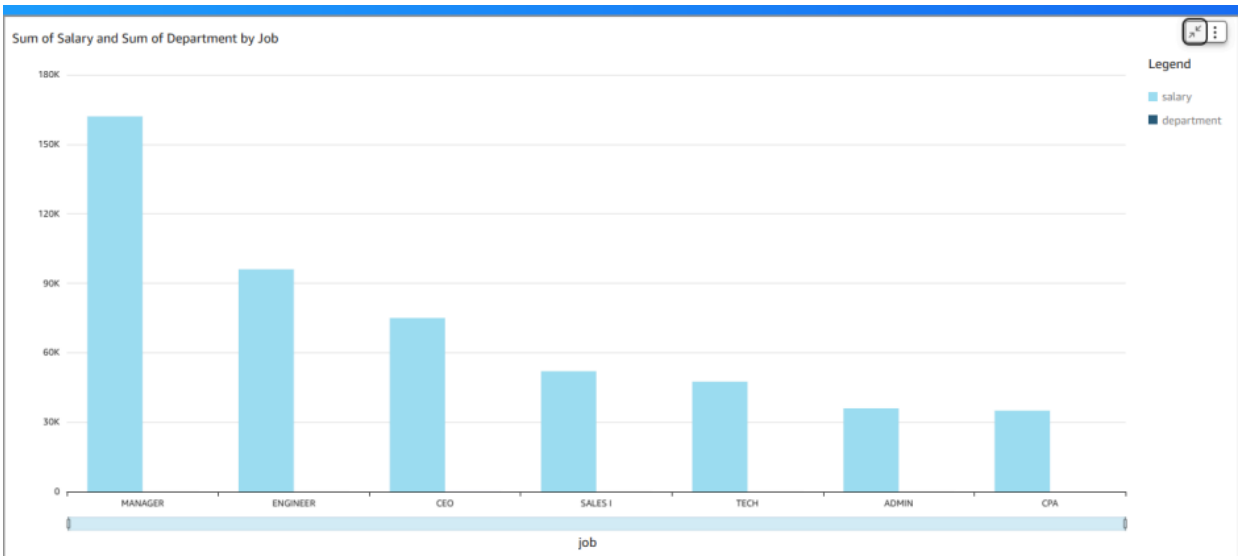
I made sure that my RDS instance could be accessed from Quick Sight by modifying the security group connection RDS.



Adding RDS as a data source for Quick Sight



This data source is different from my initial data source because initially the RDS instance is publicly available. now it is not publicly available.



Today you've learnt how to:

- 🧱 **Create a Relational Database and populate it:** You created a relational database instance in AWS and then populated it using MySQL Workbench. You also made it publicly accessible and then connected it to MySQL Workbench through the VPC.
- 😎 **Made the connection between RDS and QuickSight secure** You configured security groups for both your RDS instance and QuickSight to make sure our data stayed secure.
- 🎨 **Used the data in our RDS to create some charts:** Once everything was connected through a VPC Dataset in QuickSight, you used the data to make a stunning dashboard.