



# Visualise a Relational Database

K

Kehinde Abiuwa

The screenshot shows the MySQL Workbench interface. In the top-left pane, the 'Schemas' tree shows 'QuickSightDatabase' selected. The main area displays a 'Result Grid' for the query: 'SELECT \* FROM newhire;'. The grid contains 14 rows of data from the 'newhire' table. The bottom pane shows the 'Action Output' log with four entries.

empno	ename	job	manager	hiredate	salary	comm	deptno
1	JOHNSON	ADMIN	6	1990-12-17 00:00:00	18000.00	NULL	4
2	HARDING	MANAGER	9	1998-02-02 00:00:00	52000.00	300.00	3
3	TAFT	SALES I	2	1996-01-02 00:00:00	25000.00	500.00	3
4	HOOVER	SALES I	2	1990-04-02 00:00:00	27000.00	NULL	3
5	LINCOLN	TECH	6	1994-06-23 00:00:00	22500.00	1400.00	4
6	GARFIELD	MANAGER	9	1993-05-01 00:00:00	54000.00	NULL	4
7	POLK	TECH	6	1997-09-22 00:00:00	25000.00	NULL	4
8	GRANT	ENGINEER	10	1997-03-30 00:00:00	32000.00	NULL	2
9	JACKSON	CEO	NULL	1990-01-01 00:00:00	75000.00	NULL	4
10	FILLMORE	MANAGER	9	1994-08-09 00:00:00	56000.00	NULL	2
11	WILSON	ENGINEER	10	1992-01-01 00:00:00	32000.00	NULL	2
12	WHITE	MANAGER	10	1994-08-09 00:00:00	56000.00	NULL	2
13	SCOTT	MANAGER	10	1993-06-09 00:00:00	62000.00	200.00	5
14	BLAKE	MANAGER	10	1998-05-01 00:00:00	85000.00	500.00	5

Action Output

Time	Response	Duration / Fetch Time
20:29:38	S 0 row(s) returned	0.038 sec / 0.000020...
20:30:25	IN 14 row(s) affected Records: 14 Duplicates: 0 Warni... 0.043 sec	
20:30:37	S 14 row(s) returned	0.036 sec / 0.000031...



# Introducing Today's Project!

## What is Amazon RDS?

Amazon RDS is a managed relational database service provided by AWS that makes it easy to set up, operate, and scale databases in the cloud. It is useful because it automates time-consuming tasks like backups, patching, and maintenance, while providing high availability, security, and scalability—allowing you to focus on using the database rather than managing its infrastructure.

## How I used Amazon RDS in this project

In today's project, I used Amazon RDS to create and manage a relational database in the cloud, store and organize my sample data in tables, and securely connect it to Amazon QuickSight so I could build visualizations and analyze the data.

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

One thing I didn't expect in this project was how important security group settings are when connecting RDS to external tools like MySQL Workbench or QuickSight. It showed me that even if the database itself is set up correctly, the right permissions and network access rules are crucial for everything to work.



K

**Kehinde Abiuwa**  
NextWork Student

[nextwork.org](http://nextwork.org)

---

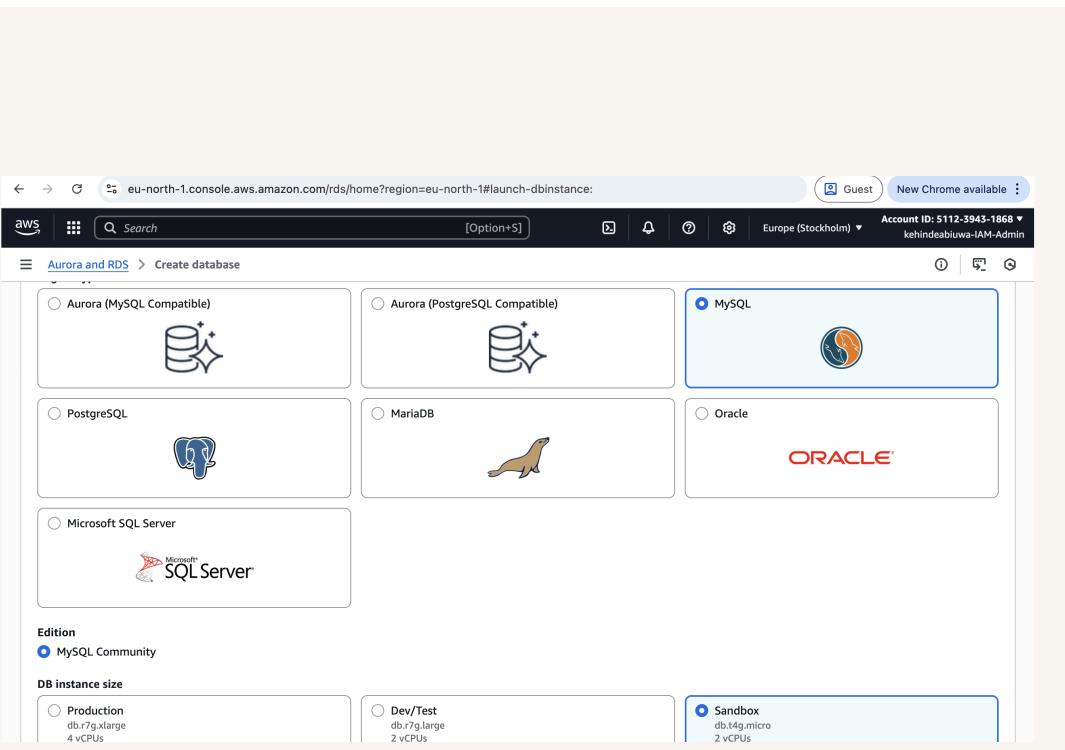
This project took me...

This project took me 1 hour and 20 minutes

# In the first part of my project...

## Creating a Relational Database

I created my relational database by using Amazon RDS in the AWS Management Console, choosing MySQL as the database engine, configuring settings like the database name, username, and password, and then launching the instance. This gave me a managed relational database that I could later connect to and populate with data.



# Understanding Relational Databases

A relational database is a type of database that organizes data into tables made up of rows and columns, where each table represents an entity (like customers, orders, or products). The “relational” part comes from the ability to define relationships between these tables using keys, which makes it easy to query, join, and manage structured data efficiently.

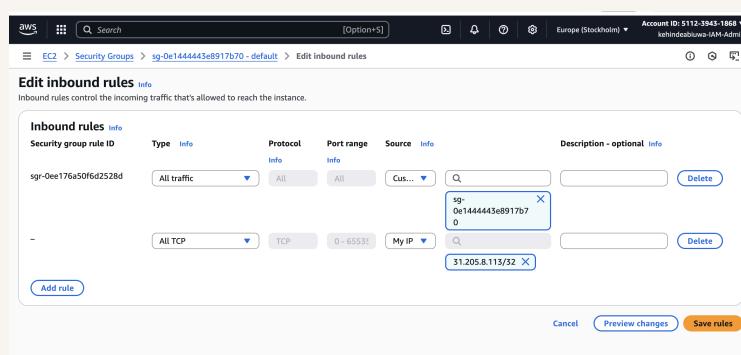
## MySQL vs SQL

The difference between MySQL and SQL is that SQL (Structured Query Language) is the standard language used to interact with relational databases, while MySQL is a database management system (DBMS) that uses SQL to store, manage, and retrieve data. In other words, SQL is the language, and MySQL is one of the popular tools that implements that language.

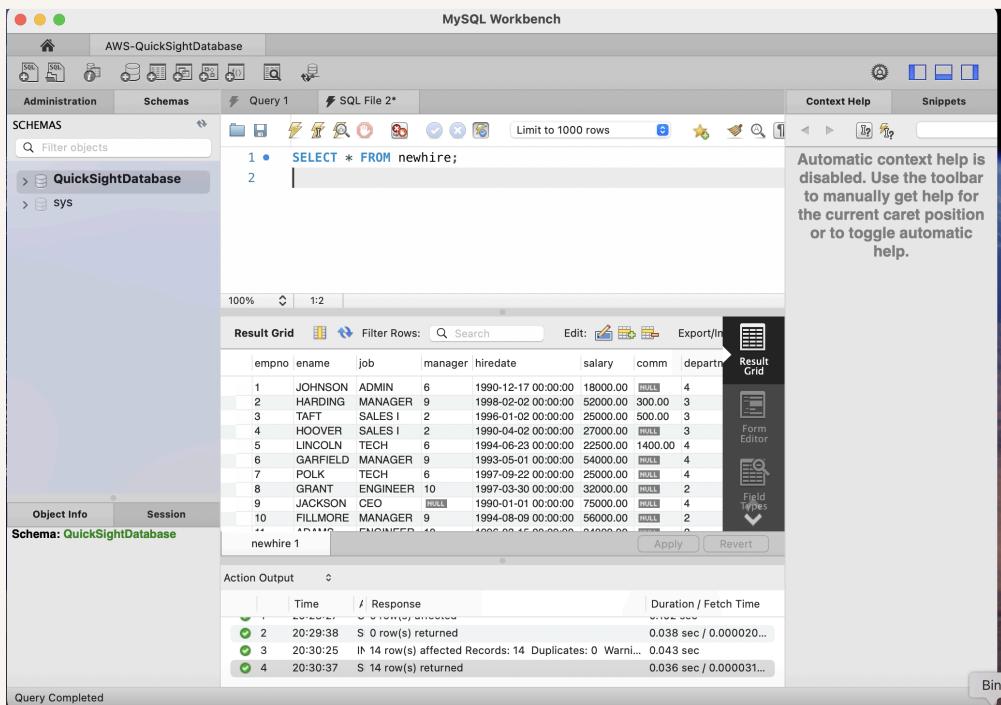
# Populating my RDS instance

The first thing I did was make my RDS instance public because I needed to connect to it from my local machine using MySQL Workbench. Making it public allowed the database to accept connections from outside the AWS network, so I could log in remotely and manage it.

I had to update the default security group for my RDS schema because by default, RDS blocks outside connections for security reasons. To connect from my local machine using MySQL Workbench, I needed to add an inbound rule that allowed traffic from my computer's IP address. This opened up the right access while still keeping the database secure from unwanted connections.



# Using MySQL Workbench



To populate my database I used SQL INSERT INTO statements in MySQL Workbench to add rows of data into my tables. After inserting the records, I ran a SELECT \* FROM <table\_name>; query to confirm that the data had been added correctly and could be retrieved from the database.



# Connecting QuickSight and RDS

To connect my RDS instance to QuickSight I updated the RDS security group to allow access from any IP range, made sure my database was publicly accessible, and then went into the QuickSight console to add a new data source. I selected RDS as the source, entered my database connection details (endpoint, port, username, and password), and tested the connection to confirm QuickSight could reach my RDS instance.

This solution is risky because allowing access from any IP range makes the RDS instance publicly exposed to the internet, which increases the chance of unauthorized access or attacks. Ideally, the security group should be restricted so that only Amazon QuickSight's managed IP addresses can connect, minimizing the attack surface and keeping the database secure.

## A better strategy

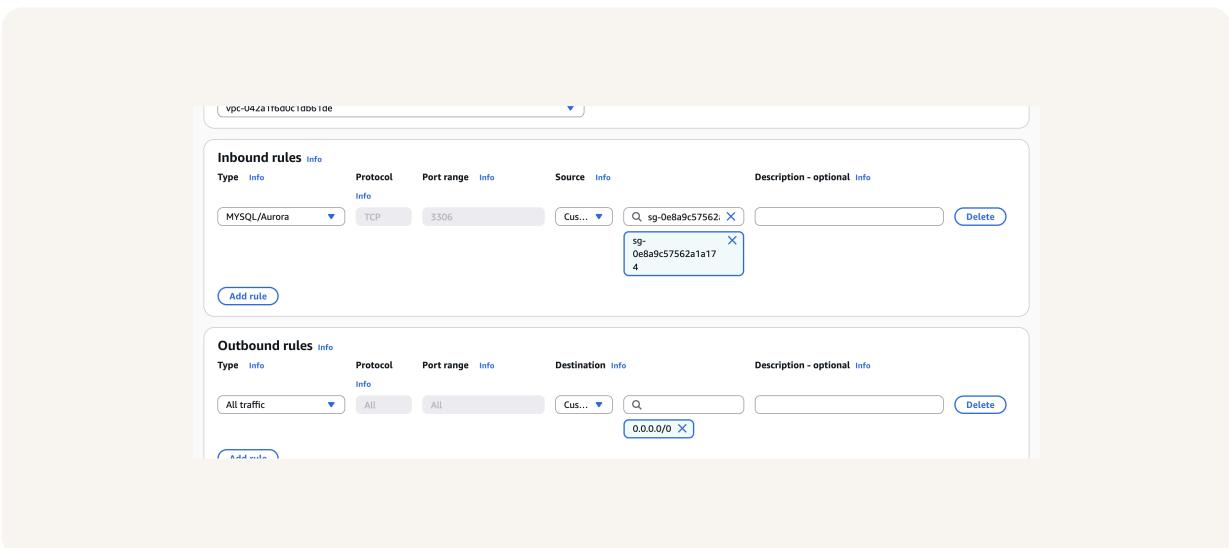
First, I made a new security group so that I could dedicate it specifically to Amazon QuickSight. This lets me tightly control access by allowing only QuickSight to connect to my RDS instance, instead of opening it up to the entire internet or unrelated resources.

Next, I connected my new security group to QuickSight by creating a new VPC connection inside the QuickSight console. I selected the same VPC where my RDS instance lives and attached the QuickSight security group I had created. This allowed QuickSight to communicate securely with my RDS database through the VPC instead of relying on open public access.

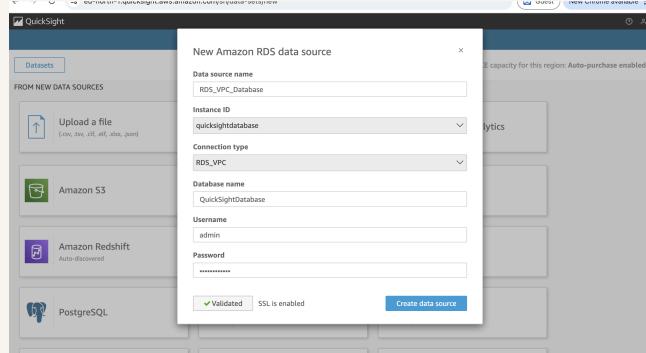
# Now to secure my RDS instance

To make my RDS instance secure I created a dedicated security group just for RDS and set its inbound rule to only allow MySQL/Aurora traffic from my QuickSight security group. This ensures that my RDS instance is private and can only be accessed by QuickSight, not from the open internet or any unauthorized source.

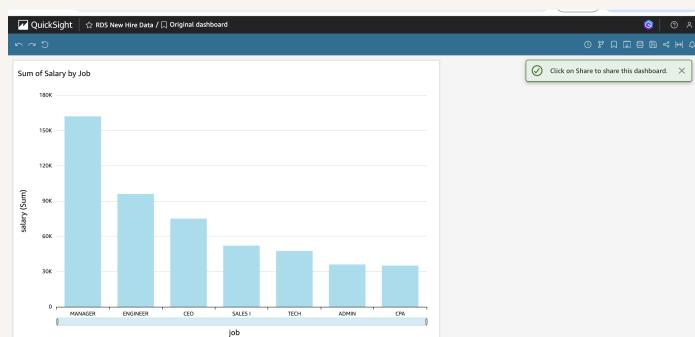
I made sure that my RDS instance could be accessed from QuickSight by attaching my new RDS security group to the database and configuring its inbound rules to only allow traffic from the QuickSight security group. This way, QuickSight could connect securely to my database without exposing it to the public internet.



# Adding RDS as a data source for QuickSight



This data source is different from my initial data source because it is now connected privately through a dedicated security group and VPC connection, instead of being publicly accessible from any IP. This makes the connection more secure and ensures that only QuickSight can access my RDS instance.





[nextwork.org](https://nextwork.org)

# The place to learn & showcase your skills

Check out [nextwork.org](https://nextwork.org) for more projects

