

# Capstone Project Use Case 1

**Data preparation using an Amazon RDS for MySQL database with AWS Glue DataBrew**

---

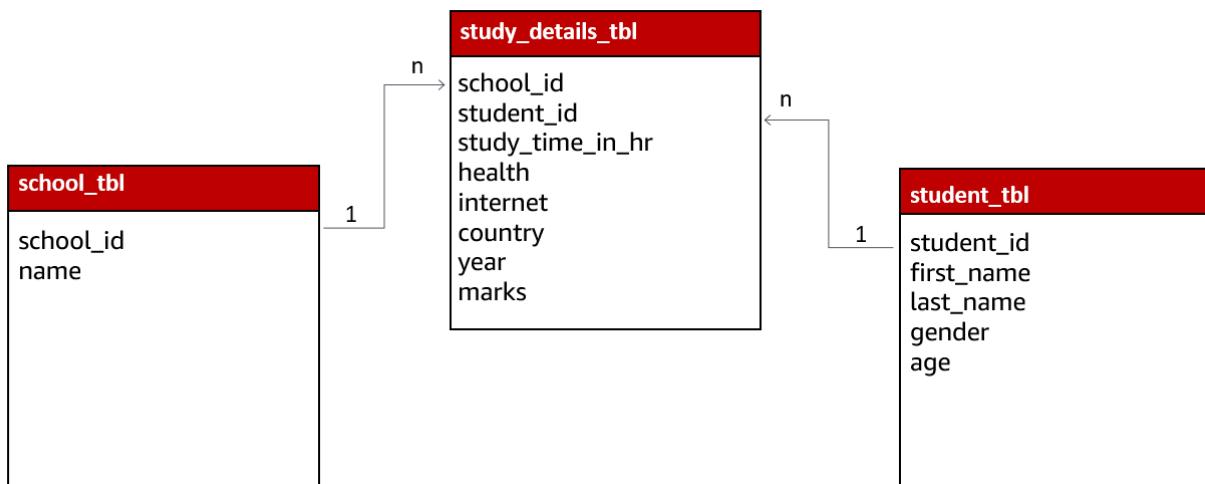
Simran Agarwal  
February 05, 2024

## Chapter 1: Use case overview

For our use case, we use three datasets:

- A school dataset that contains school details like school ID and school name
- A student dataset that contains student details like student ID, name, and age
- A student study details dataset that contains student study time, health, country, and more

The following diagram shows the relation of these tables.

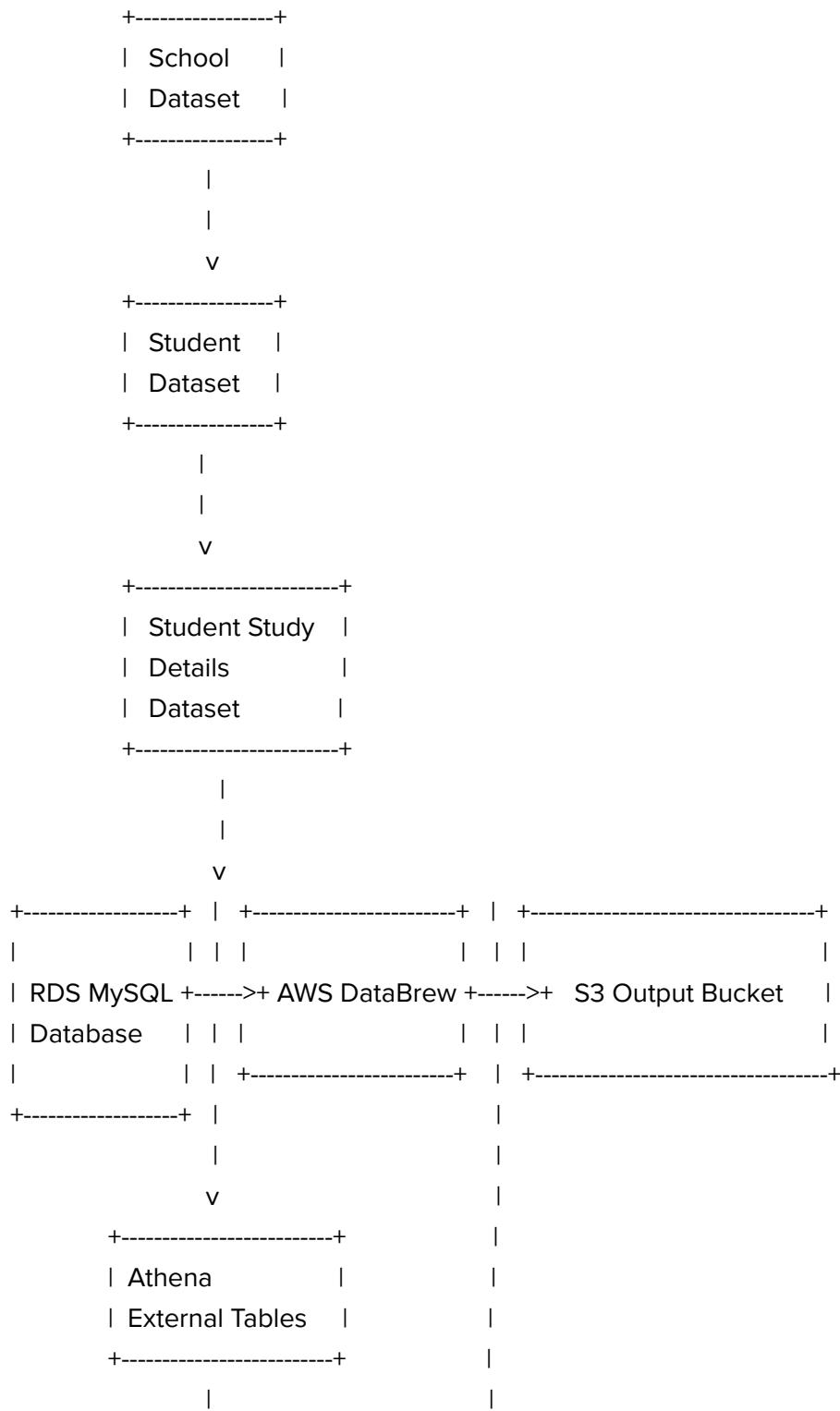


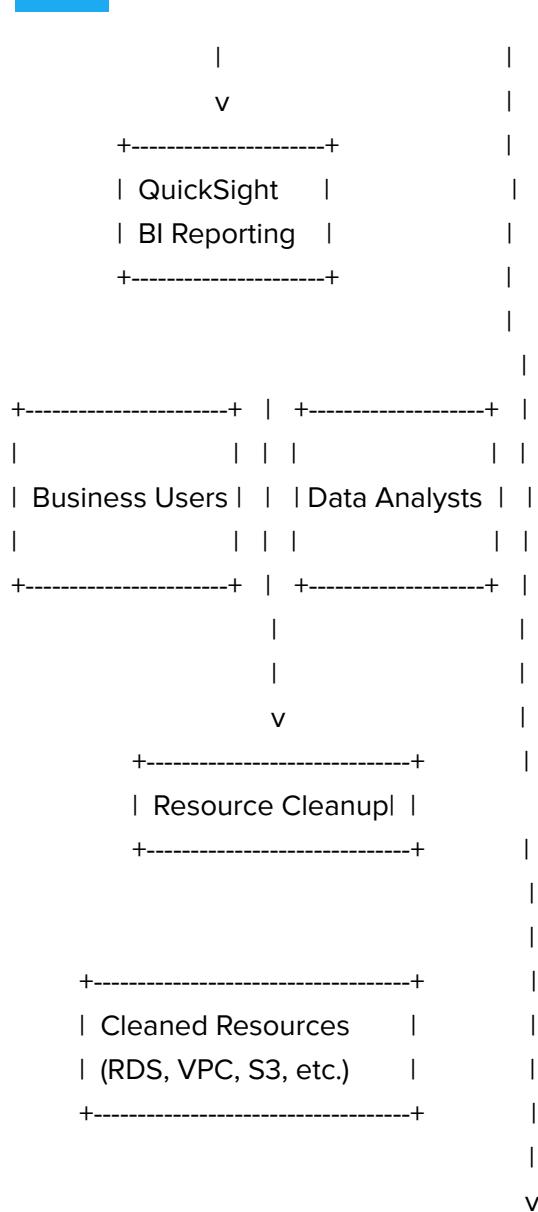
### Workflow:

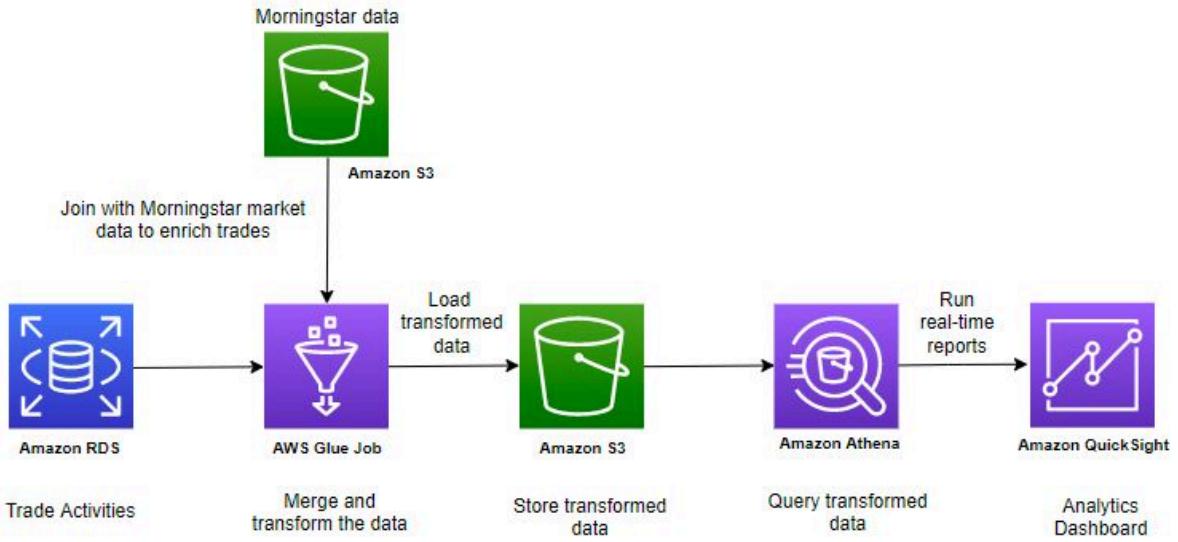
The workflow includes the following steps:

1. Create a JDBC connection for RDS and a DataBrew project. DataBrew does the transformation to find the top-performing students across all the schools considered for analysis.
2. The DataBrew job writes the final output to our S3 output bucket.
3. After the output data is written, we can create external tables on top of it with Athena create table statements.
4. Business users can use QuickSight for BI reporting, which fetches data through Athena. Data analysts can also use Athena to analyze the complete refreshed dataset.

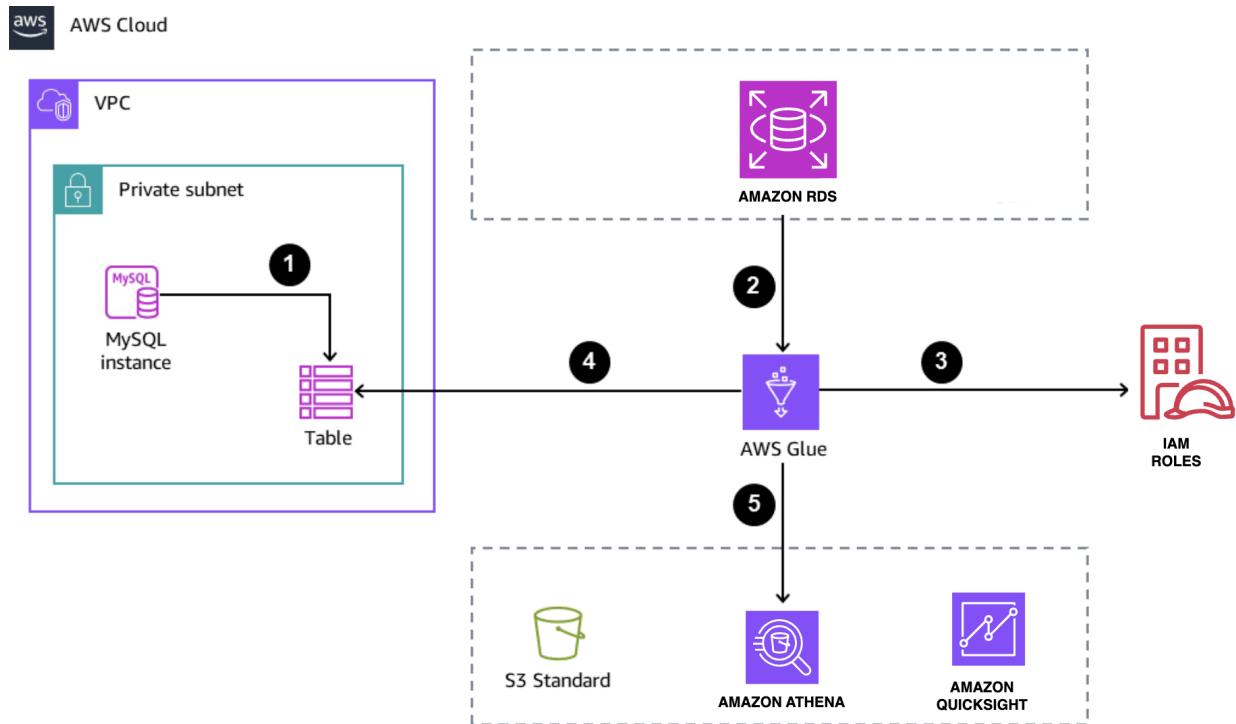
## Chapter 2: Architecture Diagram





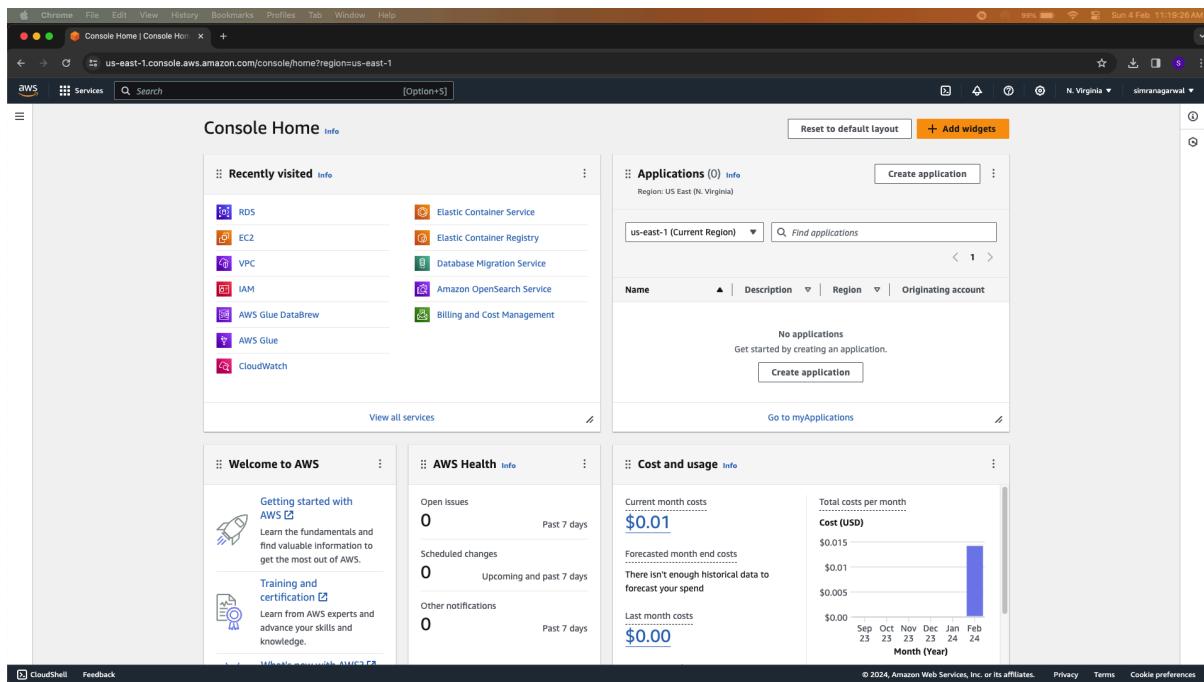


## Final Architectural Diagram:



# Chapter 3: Pre-requisites

- To complete this solution you should have an **Amazon AWS Account**.



## Chapter 4: Lab Setup

1. Create the RDS for MySQL instance to capture the student health data.

(Search for RDS in AWS Management Console and go to the RDS Dashboard)

The screenshot shows the AWS Management Console search results for 'rds'. The top result is 'RDS' under 'Services', which is highlighted. To the right, the 'RDS' dashboard is displayed. It includes a 'Create application' button, a 'Region' dropdown set to 'N. Virginia', a 'Find applications' search bar, and a message stating 'No applications'. Below this is a chart titled 'Total costs per month' showing costs from September 2023 to February 2024. The cost for February 2024 is approximately \$0.015. The chart has 'Cost (USD)' on the y-axis and 'Month (Year)' on the x-axis.

2. Click on Create Database.

The screenshot shows the AWS RDS Dashboard for the US East (N. Virginia) region. On the left, there's a sidebar with links like Dashboard, Databases, Query Editor, Performance Insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL Integrations, Events, Event subscriptions, Recommendations, and Certificate update. The main area has a banner about Multi-AZ deployment for MySQL and PostgreSQL. Below it, there's a 'Resources' section showing usage statistics for DB instances, DB Clusters, Reserved instances, and Snapshots. A large 'Create database' button is prominently displayed. To the right, there are sections for 'Recommended for you' (Amazon RDS Backup and Restore using AWS Backup, Test Your DR Strategy in Minutes, Migrate SSRS to RDS for SQL Server, Build RDS Operational Tasks) and 'Recommended services' (Customers like you also use these services). At the bottom, there are links for cloudShell, Feedback, and copyright information.

### 3. Choose a database creation method - “Standard Create” and Engine option as “MySQL”.

The screenshot shows the 'Create database' wizard. In the first step, 'Choose a database creation method', the 'Standard create' option is selected. In the second step, 'Engine options', the 'MySQL' engine type is selected. To the right, there's a detailed description of the MySQL database, listing its features such as support for up to 64 TiB, General Purpose, Memory Optimized, and Burstable Performance instance classes, automated backup, point-in-time recovery, and up to 15 Read Replicas per instance. The bottom of the screen shows standard AWS navigation links for cloudShell, Feedback, and copyright information.

#### 4. Check the Engine version and populate the fields (as shown in the diagrams below).

This screenshot shows the 'Create a new DB instance' form for MySQL. The 'DB instance identifier' is set to 'student'. Under 'Credentials Settings', the 'Master username' is 'admin'. A note indicates that managing master user credentials in AWS Secrets Manager is not supported. The 'Auto generate a password' option is checked, and a master password '\*\*\*\*\*' is entered twice. The right sidebar provides general information about MySQL and its features.

This screenshot shows the 'Configure instance options' section. Under 'DB Instance class', 'Burstable classes (includes t classes)' is selected, showing 'db.t2.micro' with 1 vCPU and 1 GiB RAM. Under 'Storage', 'General Purpose SSD (gp2)' is chosen with 20 GiB allocated. A note states that storage optimization will affect the status of the DB instance. The right sidebar contains the same MySQL information as the previous screenshot.

5. Don't connect to an EC2 compute resource (not at all needed as we will be connecting to RDS MySQL database through VS Code).

The screenshot shows the AWS RDS MySQL setup page. On the left, there's a sidebar with 'MySQL' selected. The main content area has several sections:

- Connectivity info:** A radio button is selected for "Don't connect to an EC2 compute resource".
- Virtual private cloud (VPC) info:** A dropdown menu shows "Create new VPC".
- DB subnet group info:** A dropdown menu shows "Create new DB Subnet Group".
- Public access info:** A radio button is selected for "Yes".
- VPC security group (firewall) info:** A dropdown menu shows "Choose existing".

On the right side, there's a sidebar titled "MySQL" with the following text and bullet points:

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance Instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

Also, allow the public access to access it from an external application.

## 6. Create a new subnet and VPC as well.

The screenshot shows the AWS RDS console in a web browser. The URL is `us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance;jsHermesCreate=true`. The left sidebar shows 'Services' and 'Search [Option+S]'. The main content area is titled 'DB subnet group' with a sub-section 'Info'. It says 'Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.' Below this is a dropdown menu 'Create new DB Subnet Group'. Under 'Public access' there are two options: 'Yes' (selected) and 'No'. The 'Yes' option includes a note about RDS assigning a public IP address and other resources connecting to the database. Under 'VPC security group (firewall)' there are two options: 'Choose existing' and 'Create new'. 'Create new' is selected. A text input field 'New VPC security group name' contains 'student'. Under 'Availability Zone' there is a dropdown menu 'No preference'. In the bottom right corner of the main form, there is a note about RDS Proxy and a checkbox for 'Create an RDS Proxy'. At the bottom of the form, there is a note about 'Certificate authority - optional' and a dropdown menu showing 'rds-ca-sa2048-g1 (default) Expiry: May 26, 2061'. The right side of the screen has a sidebar titled 'MySQL' with a brief description of MySQL and a bulleted list of features.

This will control the accessibility of the database from unknown sources (check the VPC and subnet settings in the VPC console).

- Check the cost of your newly created RDS Database and click on the Create button at the bottom.

The screenshot shows the AWS RDS console for creating a MySQL database. On the left, there's a sidebar with 'Monitoring' and 'Additional configuration' sections. The main area shows 'Estimated Monthly costs' with the following breakdown:

DB Instance	12.41 USD
Storage	2.30 USD
Total	14.71 USD

A note below states: "This billing estimate is based on on-demand usage as described in [Amazon RDS Pricing](#). Estimate does not include costs for backup storage, I/Os (if applicable), or data transfer." There's also a link to "Estimate your monthly costs for the DB Instance using the [AWS Simple Monthly Calculator](#)".

To the right, a panel titled "MySQL" provides an overview: "MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database." It lists several features:

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance Instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

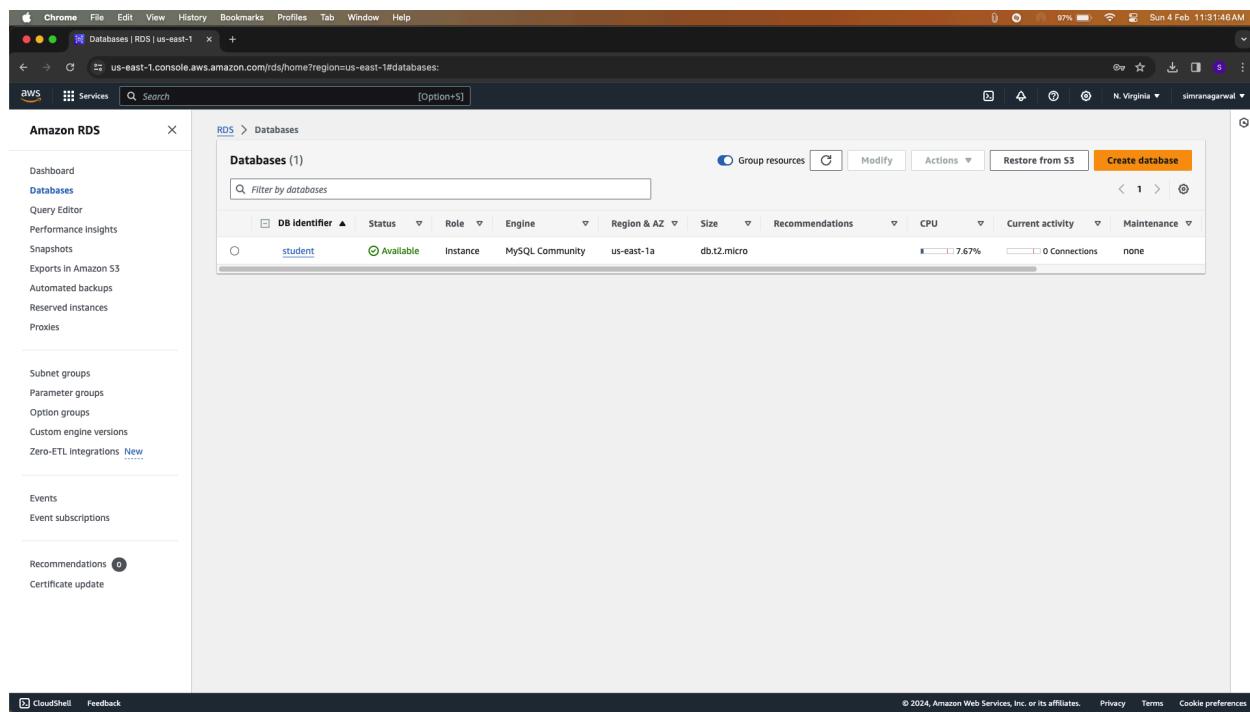
At the bottom, there are links for "CloudShell", "Feedback", and copyright information: "© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences".

RDS Database creation might take a few minutes to be successfully available for use.

The screenshot shows the Amazon RDS console in a web browser. The URL is `us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#databases:`. The left sidebar shows navigation links for Dashboard, Databases (selected), Query Editor, Performance Insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL Integrations, Events, Event subscriptions, Recommendations, and Certificate update. The main content area displays a message: "Creating database student. Your database might take a few minutes to launch. You can use settings from student to simplify configuration of suggested database add-ons while we finish creating your DB for you." Below this is a "Databases (1)" table with one row for "student". The table columns are DB identifier, Status, Role, Engine, Region & AZ, Size, Recommendations, CPU, Current activity, Maintenance, and VPC. The "student" row shows "Creating" status, MySQL Community engine, db.t2.micro size, and vpc-0517 VPC. A "Create database" button is visible at the top right of the table. The bottom of the page includes standard AWS footer links: CloudShell, Feedback, © 2024, Amazon Web Services, Inc. or its affiliates., Privacy, Terms, and Cookie preferences.

This screenshot is identical to the first one, except the database status has changed from "Creating" to "Backing-up". The rest of the interface and data remain the same.

## 8. Our RDS MySQL Database is ready to use.



The screenshot shows the AWS RDS MySQL Databases page. The left sidebar includes links for Dashboard, Databases (which is selected), Query Editor, Performance Insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL Integrations, Events, Event subscriptions, Recommendations (0), and Certificate update. The main content area displays a table titled "Databases (1)". The table has one row for a database named "student". The columns include DB identifier, Status (Available), Role, Engine (MySQL Community), Region & AZ (us-east-1a), Size, Recommendations, CPU (7.67%), Current activity (0 Connections), and Maintenance (none). A "Create database" button is located at the top right of the table. The URL in the browser is "us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#databases:".

DB identifier	Status	Role	Engine	Region & AZ	Size	Recommendations	CPU	Current activity	Maintenance
student	Available	Instance	MySQL Community	us-east-1a	db.t2.micro		7.67%	0 Connections	none

You can see the detailed configuration of the RDS instance created here.

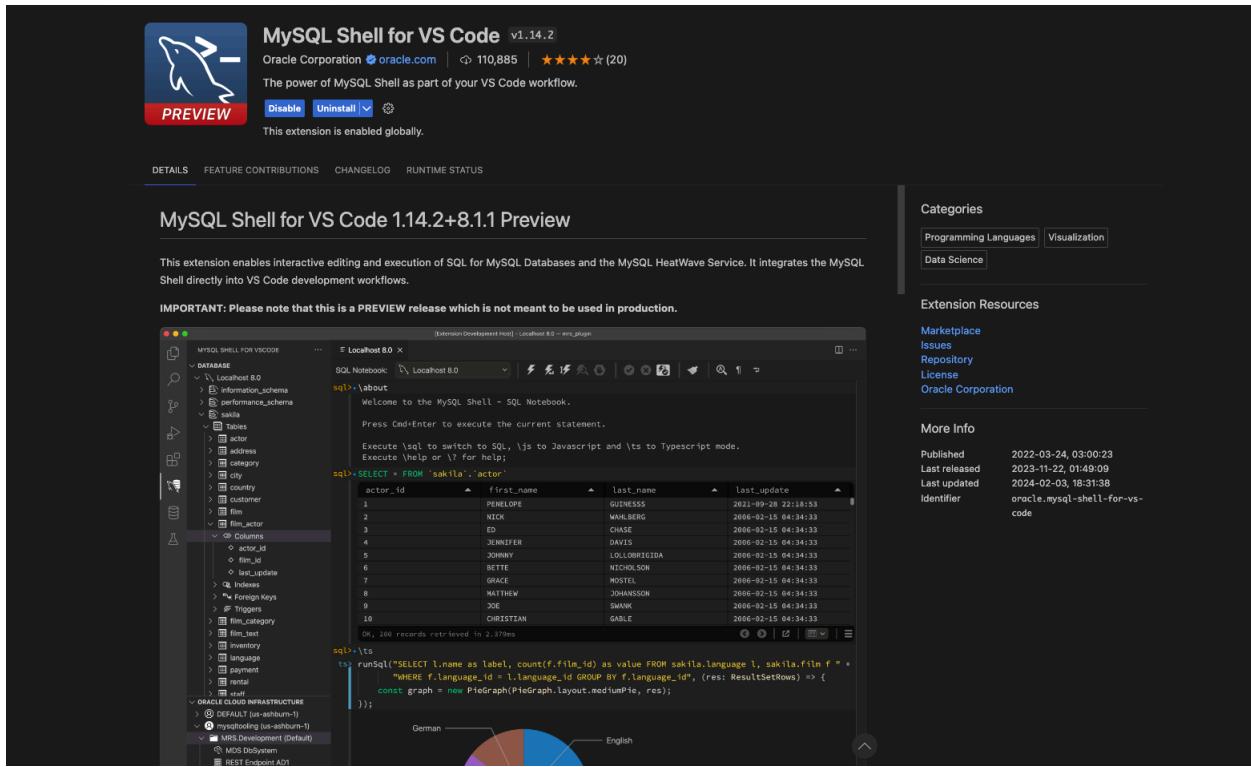
The screenshot shows the AWS RDS console interface. On the left, there's a sidebar with navigation links like Dashboard, Databases, Query Editor, and Subnet groups. The main area displays the 'student' database instance under the 'Databases' section. The 'Summary' tab is selected, showing details such as DB Identifier (student), Status (Available), Role (Instance), Engine (MySQL Community), and Recommendations. Below the summary, there are tabs for Connectivity & security, Monitoring, Logs & events, Configuration, Zero-ETL Integrations, Maintenance & backups, Tags, and Recommendations. The 'Connectivity & security' tab is active, showing information about the endpoint and port (Endpoint: student.cpuiu0qe4vxx.us-east-1.rds.amazonaws.com, Port: 3306), networking (Availability Zone: us-east-1a, VPC: vpc-0517195bd1fa9e0f0, Subnet group: default-vpc-0517195bd1fa9e0f0, Subnets: subnet-0eb9a7b0ce1668fee, subnet-0fe72df5420547a5a, subnet-00502c4eb8fabfffd, subnet-0b588ee89f5ca7862, subnet-076b15b41adefbb84c), and security (VPC security groups: student (sg-0ae497abc004b6264), Active). It also shows publicly accessible status (Yes), certificate authority (rds-ca-rsa2048-g1), and certificate authority date (May 26, 2061, 05:04 (UTC+05:30)). The DB instance certificate expiration date is February 03, 2025, 11:25 (UTC+05:30).

9. Meanwhile, install Visual Studio Code and login through your Microsoft or GitHub account (whichever you feel is preferable)

As an alternative, you can also use MySQL Workbench.

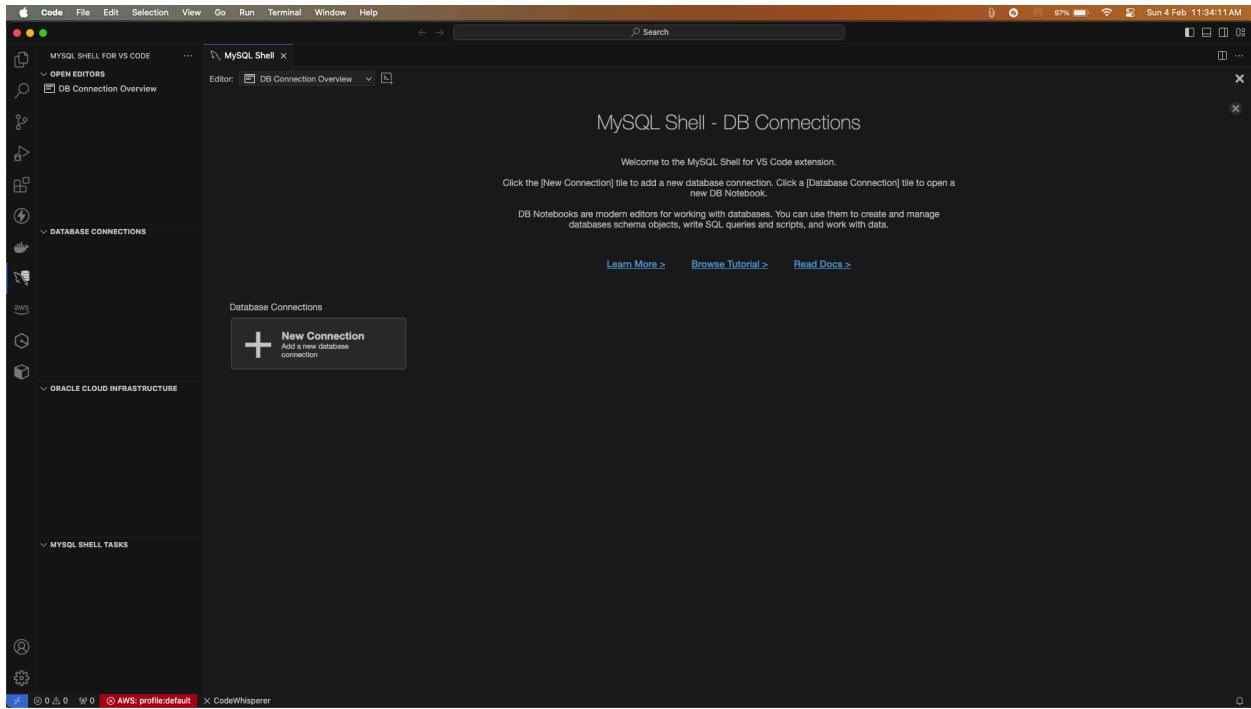
After installing and setting up VS Code, now you are ready to use the extensions.

10. Now go to the extensions tab and search for MySQL Shell for VS Code.  
 (There are many options available, but it is preferred that we use the one for Oracle Corporation.)



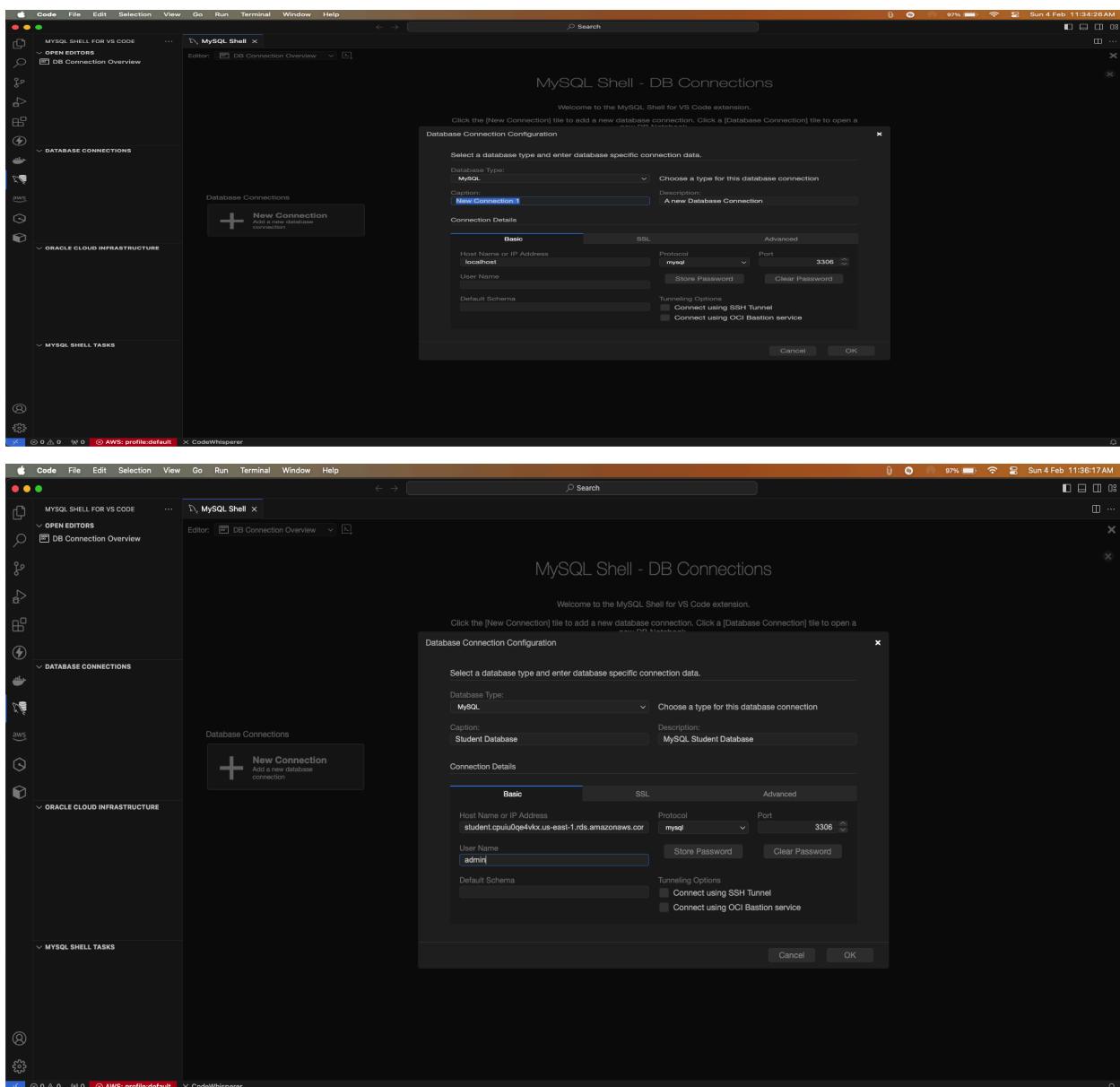
Click on install and wait for the installation to complete.

11. Now, the installation of MySQL Shell for VS Code is complete and we are ready to connect to a new database.

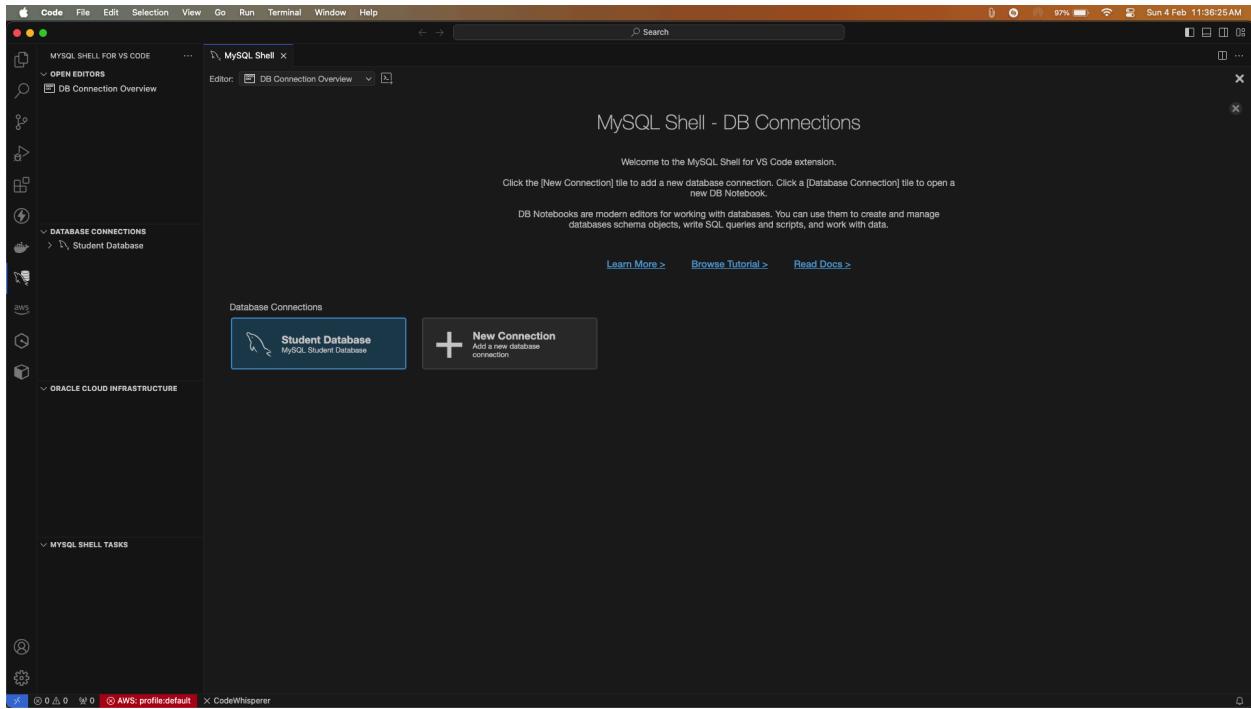


12. Fill in the details as follows:

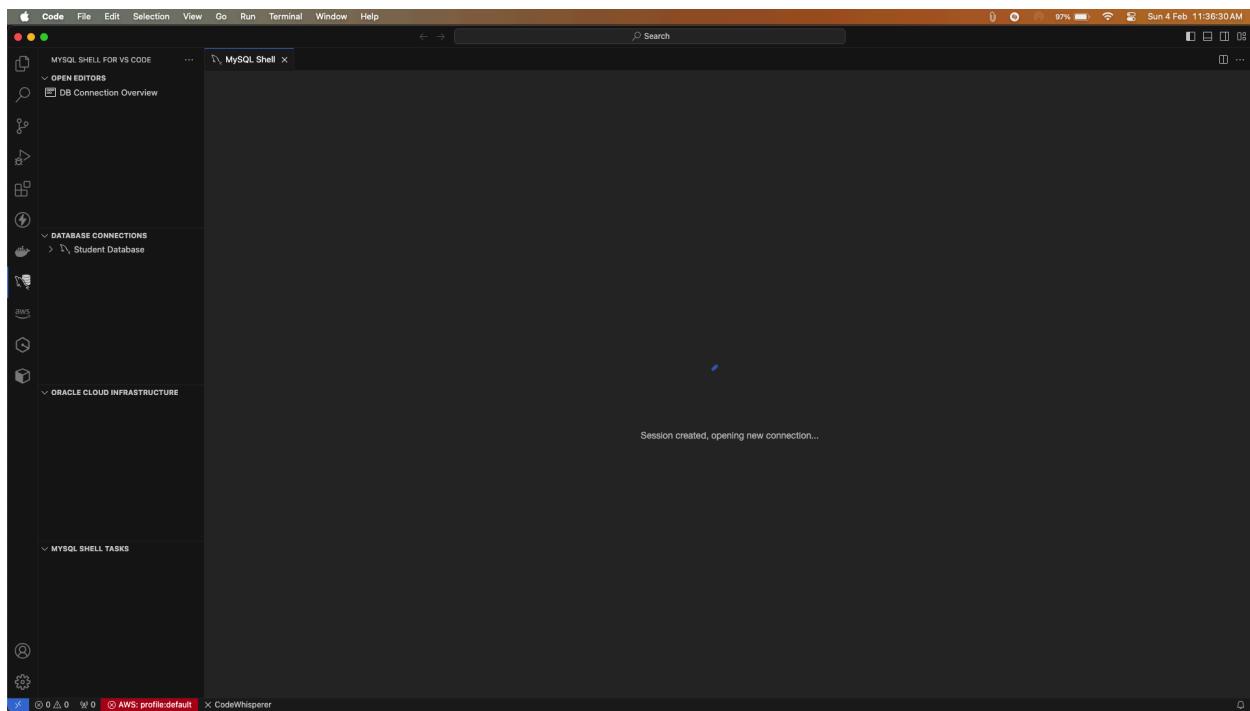
- a. Database Type: MySQL
- b. Caption: The name of the connection (Alias) and also the description.
- c. Hostname: This will be the endpoint of the RDS instance created.
- d. Username: As provided while filling up the form for creating the RDS Database (in our case it is **admin**)
- e. Default Schema: Leave it as empty as we don't have any default schema set as of now.
- f. Port: Port number for the MySQL connection. (3306).



13. As you can see we have successfully created a connection. Now, it's time to connect to the RDS Database.



14. Now click on it and enter the password, this will connect you to a database connection that we created using AWS RDS (if you have followed the process correctly so far, you will be able to reach and connect to the database)



15. Now create a database named “**student**” and create the three tables as follows with the following parameters as mentioned in the screenshot:

- a. school\_tbl
- b. student\_tbl
- c. Study\_details\_tbl

If you run the query correctly, you should see the command executed and the number of rows affected message as shown in the figure below.

The screenshot shows the MySQL Shell interface in VS Code. The left sidebar displays the MySQL Shell tasks, database connections (including a connection to 'Student Database'), and Oracle Cloud Infrastructure. The main area is titled 'Student Database' and contains a code editor with the following SQL script:

```

sql> !abort
Welcome to the MySQL Shell - DB Notebook.

Press Ctrl+Enter to execute the code block.
Execute \sql to switch to SQL, \js to JavaScript and \ts to TypeScript mode.
Execute \help or \? for help.

sql>

```

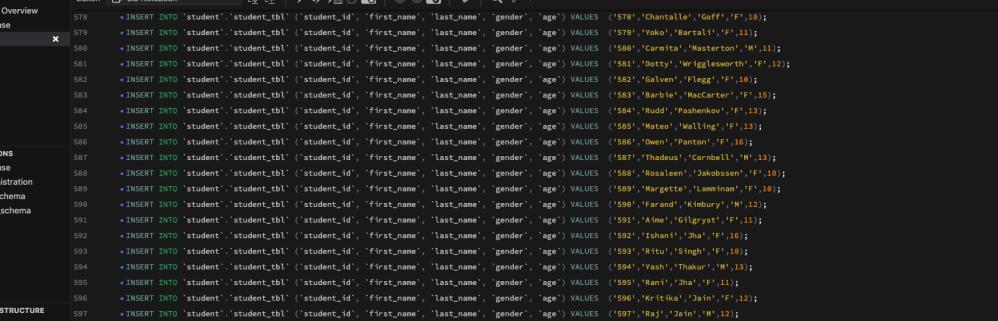
The screenshot shows the MySQL Shell interface in VS Code with the same layout. The code editor now contains the completed SQL script from the previous step, along with the results of its execution:

```

sql> !CREATE DATABASE IF NOT EXISTS `student`;
1
2
3 USE `student`;
4
5 !CREATE TABLE `school_tbl` (
6     `school_id` varchar(15) NOT NULL,
7     `name` varchar(45) DEFAULT NULL,
8     PRIMARY KEY (`school_id`)
9 );
10
11 !CREATE TABLE `student_tbl` (
12     `student_id` varchar(3) NOT NULL,
13     `first_name` varchar(50) DEFAULT NULL,
14     `last_name` varchar(50) DEFAULT NULL,
15     `gender` varchar(10) DEFAULT NULL,
16     `age` int(11) DEFAULT NULL,
17     PRIMARY KEY (`student_id`)
18 );
19
20 !CREATE TABLE `study_details_tbl` (
21     `school_id` varchar(15) NOT NULL,
22     `student_id` varchar(3) NOT NULL,
23     `study_time_in_hr` int(11) DEFAULT NULL,
24     `health` varchar(15) DEFAULT NULL,
25     `internet` varchar(10) DEFAULT NULL,
26     `country` varchar(20) DEFAULT NULL,
27     `year` varchar(4) DEFAULT NULL,
28     `marks` int(11) DEFAULT NULL,
29     PRIMARY KEY (`school_id`, `student_id`),
30     FOREIGN KEY (`school_id`) REFERENCES `school_tbl` (`school_id`),
31     FOREIGN KEY (`student_id`) REFERENCES `student_tbl` (`student_id`)
32 );
33
34
#1:OK, 0 rows affected in 315.788ms
#2:OK, 0 records retrieved in 37.37Ms
#3:OK, 0 records retrieved in 487.225ms
#4:OK, 0 records retrieved in 398.734ms
#5:OK, 0 records retrieved in 397.264ms

```

16. Now carefully insert the data into the three tables in order (as if you miss the order and try to perform the insertion in other tables first then it will give you a foreign key constraint error as all the tables are connected with a foreign key constraint).



The screenshot shows the MySQL Shell interface within VS Code. The left sidebar displays database connections (MySQL Administration, information\_schema, performance\_schema, student, sys) and Oracle Cloud Infrastructure resources. The main area is titled 'Student Database' and contains a 'DB Notebook' tab with the following SQL code:

```
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('578', 'Chantalle', 'Goff', 'F', 18);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('579', 'Yoko', 'Bartali', 'F', 11);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('580', 'Carrie', 'Masteron', 'M', 11);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('581', 'Dotty', 'Wigglesworth', 'F', 12);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('582', 'Galven', 'Flagg', 'F', 10);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('583', 'Barbie', 'MacCarter', 'F', 15);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('584', 'Rudd', 'Pashenkov', 'F', 13);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('585', 'Mateso', 'Walting', 'F', 13);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('586', 'Own', 'Panton', 'F', 16);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('587', 'Thadeus', 'Cornbell', 'M', 13);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('588', 'Rosaleen', 'Jakobsson', 'F', 10);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('589', 'Margrete', 'Lamianin', 'F', 16);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('590', 'Farand', 'Kinebury', 'M', 12);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('591', 'Aime', 'Glyrst', 'F', 11);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('592', 'Ishant', 'Jha', 'F', 16);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('593', 'Ritu', 'Singh', 'F', 10);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('594', 'Yash', 'Thakur', 'M', 13);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('595', 'Rami', 'Jha', 'F', 13);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('596', 'Krithika', 'Jain', 'F', 12);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('597', 'Raj', 'Jain', 'M', 12);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('598', 'Suresh', 'Jain', 'M', 14);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('599', 'Sonali', 'Singh', 'F', 10);
+ INSERT INTO `student`.`student_tbl` ('student_id', 'first_name', 'last_name', 'gender', 'age') VALUES ('600', 'Kavya', 'Jain', 'F', 10);

#578: Ok, 1 row affected in 393.99ms
#579: Ok, 1 row affected in 393.122ms
#580: Ok, 1 row affected in 393.122ms
#581: Ok, 1 row affected in 393.94ms
#582: Ok, 1 row affected in 393.252ms
#583: Ok, 1 row affected in 393.44ms
#584: Ok, 1 row affected in 393.745ms
#585: Ok, 1 row affected in 393.325ms
#586: Ok, 1 row affected in 393.387ms
#587: Ok, 1 row affected in 397.99ms
#588: Ok, 1 row affected in 393.325ms
#589: Ok, 1 row affected in 394.69ms
#590: Ok, 1 row affected in 393.122ms
#591: Ok, 1 row affected in 393.122ms
#592: Ok, 1 row affected in 393.94ms
#593: Ok, 1 row affected in 398.252ms
#594: Ok, 1 row affected in 393.94ms
#595: Ok, 1 row affected in 397.634ms
#596: Ok, 1 row affected in 393.646
#597: Ok, 1 row affected in 396.174ms
#598: Ok, 1 row affected in 393.646ms
#599: Ok, 1 row affected in 393.686ms
#600: Ok, 1 row affected in 396.864ms
```

First, you need to insert the data in:

- a. school\_tbl
  - b. student\_tbl
  - c. study\_details\_tbl



The screenshot shows the MySQL Shell interface within VS Code. The left sidebar displays database connections, including 'Student Database' and 'AWS'. The main area contains three tabs:

- student.school\_tbl:**

student_id	first_name	last_name	gender	age
1	Raj	Singh	M	12
10	Sonia	Jain	F	11
100	Stillmann	Pendrid	F	12
101	Jarie	Tilne	F	12
102	Zitella	Dowrey	F	12
103	Dario	Phillott	M	15
104	Rosina	Van Zon	M	17
105	Anne-cornine	Chaddock	F	15
106	Tomasina	Palfreyman	F	16
107	Cybille	Gherardi	M	13
108	Luis	Spalding	M	13
109	Delphinia	Gauthier	F	14
11	Corrianne	Santostefano	F	17
110	Cully	Jerke	M	14
...	...	...	...	...
- student.student\_tbl:**

student_id	study_time_in_hr	health	internet	country	year	marks
02-6957378	107	4	great	United States	2020	36
02-6957378	110	5	average	TRUE	2020	45
02-6957378	113	6	bad	FALSE	United States	35
02-6957378	118	4	bad	FALSE	United States	36
02-6957378	122	6	worst	FALSE	United States	54
02-6957378	123	4	bad	FALSE	United States	36
02-6957378	125	4	worst	TRUE	United States	36
02-6957378	126	7	average	TRUE	United States	2020
02-6957378	13	8	great	FALSE	United States	2020
02-6957378	133	8	good	TRUE	United States	2020
02-6957378	136	6	average	TRUE	United States	2020
02-6957378	139	5	average	FALSE	United States	2020
02-6957378	142	5	good	FALSE	United States	2020
02-6957378	148	5	worst	FALSE	United States	2020
...	...	...	...	...	...	...
- student.study\_details\_tbl:**

school_id	student_id	marks
02-6957378	107	36
02-6957378	110	45
02-6957378	113	35
02-6957378	118	36
02-6957378	122	54
02-6957378	123	36
02-6957378	125	36
02-6957378	126	63
02-6957378	13	72
02-6957378	133	72
02-6957378	136	54
02-6957378	139	45
02-6957378	142	45
02-6957378	148	45
...	...	...

17. Now after inserting the data in all three tables check all the data in all three tables using the following queries.

- select \* from student.school\_tbl;
- select \* from student.student\_tbl;
- select \* from student.study\_details\_tbl;

If you find the data in all three tables, then congratulations you are good to go.

18. Goto, VPC Console by typing VPC in the search bar of Amazon AWS and click on VPC from the search results.

The screenshot shows the AWS VPC Console home page in a web browser. The URL is [us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#Home](https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#Home). The page has a top navigation bar with tabs for RDS, Home, and VPC Console. Below the navigation is a search bar and a dropdown for 'Option+5'. On the left, there's a sidebar titled 'VPC dashboard' with a 'Create VPC' button and a 'Launch EC2 Instances' button. The sidebar also lists various VPC-related services like EC2 Global View, Virtual private cloud, Security, and Network Firewall. The main content area is titled 'Resources by Region' and shows a grid of resources with counts for US East 1 and US East 2 regions. Resources include VPCs (6), Subnets (6), Route Tables (2), Internet Gateways (1), Egress-only Internet Gateways (0), DHCP option sets (1), Endpoints (0), Instance Connect Endpoints (0), and Endpoint Services (0). To the right of the resource grid are sections for 'Service Health', 'Settings' (with links to Zones and Console Experiments), 'Additional Information' (with links to VPC Documentation, All VPC Resources, Forums, and Report an Issue), and 'AWS Network Manager' (with a link to Get started with Network Manager). At the bottom right, there's a 'Create VPN Connection' button and a 'Site-to-Site VPN Connections' section. The footer includes links for CloudShell, Feedback, and copyright information: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

19. Now open the Endpoints tab in the sub-menu of VPC, and click on “Create Endpoint”.

The screenshot shows a browser window with the AWS VPC Endpoints console. The URL is `us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#Endpoints:`. The page title is "Endpoints". On the left, there is a navigation sidebar with the following categories and sub-items:

- VPC dashboard
- EC2 Global View
- Filter by VPC:
  - Select a VPC
- Virtual private cloud
  - Your VPCs
  - Subnets
  - Route tables
  - Internet gateways
  - Egress-only internet gateways
  - Carrier gateways
  - DHCP option sets
  - Elastic IPs
  - Managed prefix lists
  - Endpoints** (selected)
  - Endpoint services
  - NAT gateways
  - Peering connections
- Security
  - Network ACLs
  - Security groups
- DNS firewall
  - Rule groups
  - Domain lists
- Network Firewall
  - Firewalls
  - Firewall policies

The main content area is titled "Endpoints" and contains a search bar and a table header with columns: Name, VPC endpoint ID, VPC ID, Service name, Endpoint type, and Status. A message "No endpoint found" is displayed below the table. At the bottom of the page, there is a "Select an endpoint" section and a footer with links to CloudShell, Feedback, and various AWS terms and conditions.

20. Under the service category Select “AWS Services” and fill out the form accordingly.

The screenshot shows the 'Create endpoint' wizard in the AWS VPC console. The current step is 'Endpoint settings'. The 'Service category' section is expanded, showing the following options:

- AWS services: Services provided by Amazon
- PrivateLink Ready partner services: Services with an AWS Service Ready designation
- AWS Marketplace services: Services that you've purchased through AWS Marketplace
- EC2 Instance Connect Endpoint: An elastic network interface that allows you to connect to resources in a private subnet
- Other endpoint services: Find services shared with you by service name

The 'Services (1/2)' section shows a table with one item:

Service Name	Owner	Type
com.amazonaws.us-east-1.s3	amazon	Gateway

The 'VPC' section is collapsed. At the bottom, there are links for CloudShell and Feedback, and a footer with copyright information: © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences.

21. Now search for “s3” and select the one with Type - Gateway.

Screenshot of the AWS VPC Endpoint creation process:

**Step 1: Services Selection**

The "Services" section shows two entries for "com.amazonaws.us-east-1.s3". The first entry is selected as a "Gateway".

Service Name	Owner	Type
com.amazonaws.us-east-1.s3	amazon	Gateway
com.amazonaws.us-east-1.s3	amazon	Interface

**Step 2: VPC Selection**

The "VPC" section shows a dropdown menu with the option "vpc-0517195bd1fa9e0f0" selected.

**Step 3: Route Tables Selection**

The "Route tables (1/1)" section shows one route table selected: "rtb-0b63295a2af1b1c80".

**Step 4: Policy Selection**

The "Policy" section shows the "Full access" policy selected.

**Step 5: Tags**

A tag named "Name" is added with the value "databrews3".

**Step 6: Create Endpoint**

The "Create endpoint" button is highlighted in orange at the bottom right.

The screenshot shows the AWS VPC Endpoints console in a web browser. The URL is [us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#Endpoints:vpcEndpointId=vpce-0ec4d3312632c2c0f](https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#Endpoints:vpcEndpointId=vpce-0ec4d3312632c2c0f). The page displays a success message: "Successfully created VPC endpoint vpce-0ec4d3312632c2c0f". Below this, a table lists the endpoint details:

Name	VPC endpoint ID	VPC ID	Service name	Endpoint type	Status
databrews3	vpce-0ec4d3312632c2c0f	vpc-0517195bd1fa9e0f0	com.amazonaws.us-east-1.s3	Gateway	Available

On the left sidebar, under the "Endpoints" section, there is a link to "vpce-0ec4d3312632c2c0f / databrews3". The "Details" tab is selected, showing the following information:

Endpoint ID	Status	Creation time	Endpoint type
vpce-0ec4d3312632c2c0f	Available	Sunday, 4 February 2024 at 12:21:35 GMT+5:30	Gateway
VPC ID	Status message	Service name	Private DNS names enabled
vpc-0517195bd1fa9e0f0	-	com.amazonaws.us-east-1.s3	No

At the bottom right of the page, there is a footer with links: "© 2024, Amazon Web Services, Inc. or its affiliates.", "Privacy", "Terms", and "Cookie preferences".

22. Now that your endpoint is created, You need to search for AWS Glue Databrew. (here we will try to create a data set and a project and then run a job which will put the output in the S3 Bucket).

The screenshot shows the AWS Glue DataBrew landing page. The top navigation bar includes 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Profiles', 'Tab', 'Window', and 'Help'. The address bar shows 'us-east-1.console.aws.amazon.com/databrew/home?region=us-east-1#landing'. The left sidebar has a 'Services' dropdown, a search bar with 'rds', and links for 'DATASETS', 'PROJECTS', 'RECIPES', 'DQ RULES', 'JOBS', and 'What's New'. The main content area features the title 'AWS Glue DataBrew' and the tagline 'Clean and normalize data up to 80% faster'. Below this is a section titled 'How it works' with a video thumbnail for 'AWS Glue DataBrew Demo Video For Beginners'. To the right, there are two call-to-action boxes: 'Create a project' (orange button) and 'Create sample project' (grey button). Further down is a 'Pricing' section with a table comparing interactive sessions and DataBrew jobs, and a 'Cost calculator' link. At the bottom, there are 'Getting started' links and standard footer links for 'CloudShell', 'Feedback', '© 2024, Amazon Web Services, Inc. or its affiliates.', 'Privacy', 'Terms', and 'Cookie preferences'.

23. Before going to Databrew we need to take care and assign some roles for the Databrew. Goto IAM Console by searching in the search bar or through AWS Home Panel.

The screenshot shows the AWS IAM Roles page. On the left, there's a navigation sidebar with options like Dashboard, Access management, Access reports, and Related consoles. The main content area has a heading "Roles (4) Info" with a sub-section "Roles Anywhere". It lists four roles:

Role name	Trusted entities	Last activity
AWSServiceRoleForRDS	AWS Service: rds (Service-Linked Role)	13 minutes ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service)	-
rds-monitoring-role	AWS Service: monitoring.rds	-

Below the table, there are three sections: "Access AWS from your non AWS workloads", "X.509 Standard", and "Temporary credentials". At the bottom right, there are links for "Manage", "CloudShell", "Feedback", and copyright information.

24. Click on “Create Role”.

Choose the trusted entity type as “AWS Service” and use case as “**Glue**” and then click Next.

Screenshot of the AWS IAM 'Create role' wizard, Step 1: Select trusted entity.

**Select trusted entity**

**Trusted entity type**

- AWS service**  
Allows AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account**  
Allows entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity**  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation**  
Allows users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy**  
Create a custom trust policy to enable others to perform actions in this account.

**Use case**  
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

**Service or use case**  
Glue

**Choose a use case for the specified service.**  
**Use case**  
 **Glue**  
Allows Glue to call AWS services on your behalf.

**Next**

Screenshot of the AWS IAM 'Create role' wizard, Step 2: Name, review, and create.

**Name, review, and create**

**Role details**

**Role name**  
Enter a meaningful name to identify this role.  
**Master Role**

**Description**  
Add a short explanation for this role.  
Allows Glue to call AWS services on your behalf.

**Step 1: Select trusted entities**

**Trust policy**

```

1- {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Principal": "*",
7       "Service": "glue.amazonaws.com"
8     },
9     "Action": "sts:AssumeRole"
10   ]
11 }
12 }

```

**Step 2: Add permissions**

**Step 2: Add permissions**

Policy name	Type	Attached as
<a href="#">AmazonRDSDataFullAccess</a>	AWS managed	Permissions policy
<a href="#">AmazonRDSFullAccess</a>	AWS managed	Permissions policy
<a href="#">AmazonS3FullAccess</a>	AWS managed	Permissions policy
<a href="#">AWSGlueConsoleFullAccess</a>	AWS managed	Permissions policy
<a href="#">AwsGlueDataBrewFullAccessPolicy</a>	AWS managed	Permissions policy
<a href="#">AWSGlueDataBrewServiceRole</a>	AWS managed	Permissions policy
<a href="#">AWSGlueServiceRole</a>	AWS managed	Permissions policy

**Step 3: Add tags**

Add tags - optional Info  
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add new tag You can add up to 50 more tags.

Cancel Previous Create role

Review the policies assigned by you to that role.

Essential ones are.

- S3 Full Access Policy - will be used to output the dataset to an S3 Bucket
- RDS Full Access Policy - will be used to read data from the RDS MySQL Database.
- Glue Databrew Access Policy - will be used to allow connecting to the Databrew.
- Glue Service Role Policy - For testing and editing the connections.

Also, you can assign roles like:

- Athena Full Access Policy - for accessing Athena and Athena to access other resources as well.
- Quick Sight Access Policy - for accessing the QuickSight.

25. Now go to the Security Group sections under the VPC Console and click on edit the inbound rules (outbound rules remain the default, there is no change).

The screenshot shows two consecutive screenshots of the AWS VPC Security Groups console. Both screenshots are from the same browser session, with the second one showing a success message.

**Screenshot 1 (Top):**

- Left Sidebar:** Shows the VPC dashboard and a detailed navigation menu under the **Security groups** section, including options like Network ACLs, DNS firewall, and Network Firewall.
- Central Content:** Displays the details for a security group named **sg-0ae497abc004b6264 - student**. It shows the security group ID as **sg-0ae497abc004b6264**, owner as **380172154144**, and a single inbound rule entry.
- Inbound Rules Table:**

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-0fe12ad883a844511	IPv4	MySQL/Aurora	TCP	3306	106.213.82.95/32	-

**Screenshot 2 (Bottom):**

- Left Sidebar:** Same as the first screenshot.
- Central Content:** Shows a green success message: **Inbound security group rules successfully modified on security group sg-0ae497abc004b6264 | student**.
- Details Section:** Shows the same security group details as the first screenshot.
- Inbound Rules Table:**

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-071ee0a398a77f1c2	IPv4	All traffic	All	All	0.0.0.0/0	-
-	sgr-0fe12ad883a844511	IPv4	MySQL/Aurora	TCP	3306	106.213.82.95/32	-

26. Now, go to the NAT Gateway section and create a NAT Gateway which will later be linked to our VPC.

**Create NAT gateway** Info

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

**NAT gateway settings**

Name - optional  
Create a tag with a key of 'Name' and a value that you specify.  
  
The name can be up to 256 characters long.

Subnet  
Select a subnet in which to create the NAT gateway.

Connectivity type  
Select a connectivity type for the NAT gateway.  
 Public  
 Private

Elastic IP allocation ID Info  
Assign an Elastic IP address to the NAT gateway.

**► Additional settings** Info

**Tags**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="student-nat"/> <input type="button" value="Remove"/>

You can add 49 more tags.

**NAT gateway nat-04023b6101ae4def7 | student-nat was created successfully.**

**nat-04023b6101ae4def7 / student-nat**

**Details**

NAT gateway ID <input type="text" value="nat-04023b6101ae4def7"/>	Connectivity type Public	State <input checked="" type="radio"/> Pending	State message <small>Info</small> —
NAT gateway ARN <input type="text" value="arn:aws:ec2:us-east-1:380172154144:natgateway/nat-04023b6101ae4def7"/>	Primary public IPv4 address —	Primary private IPv4 address —	Primary network interface ID —
VPC <input type="text" value="vpc-0517195bd1fa9e0f0"/>	Subnet <input type="text" value="subnet-0b588ee89f3ca7862"/>	Created <input type="text" value="Sunday, 4 February 2024 at 12:40:44 GMT+5:30"/>	Deleted —

**Secondary IPv4 addresses**

Private IPv4 address	Network interface ID	Status	Failure message
Secondary IPv4 addresses are not available for this nat gateway.			

27. In the security group, you need to add all the details that are mentioned in the image below:

### Inbound Rules (Must have)

- a. MySQL
- b. All TCP

The screenshot shows the AWS VPC Security Groups console. The URL is <https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#SecurityGroup:group-id=sg-0ae497abc004b6264>. The security group is named "sg-0ae497abc004b6264 - student".

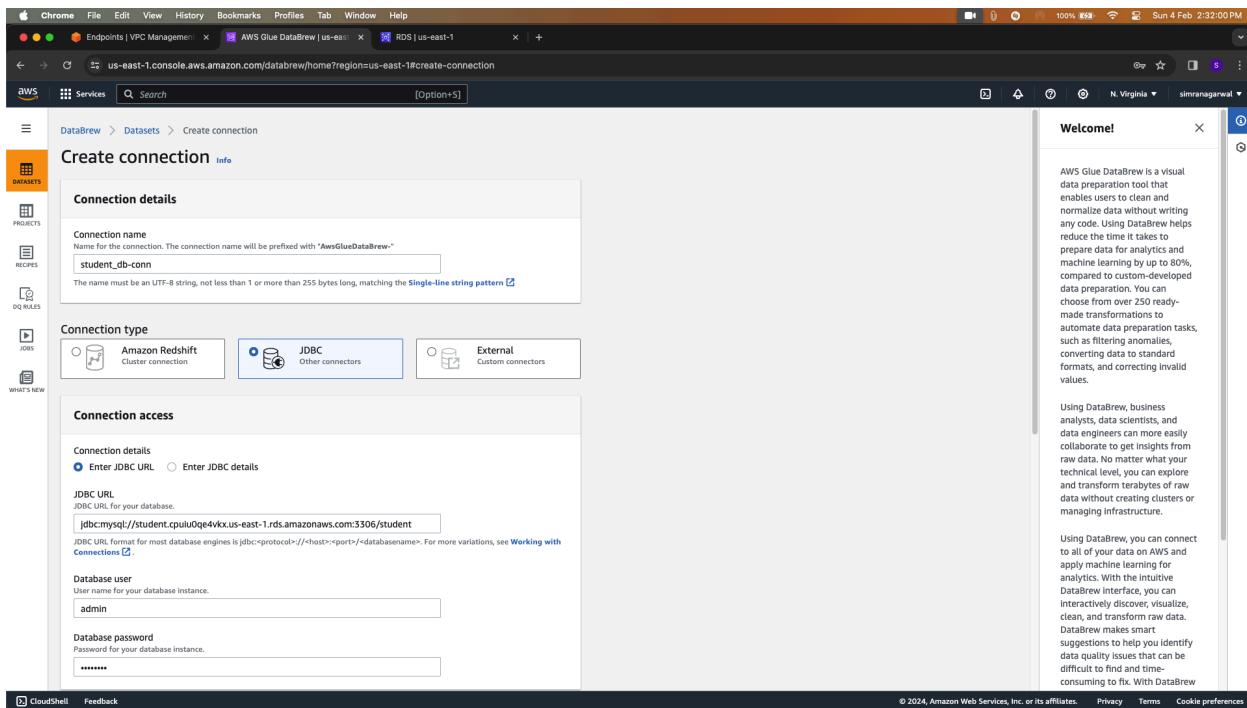
**Details:**

Security group name	sg-0ae497abc004b6264	Description	VPC ID
Owner	380172154144	Inbound rules count	vpc-0517195bd1fa9e0f0
		3 Permission entries	
		Outbound rules count	1 Permission entry

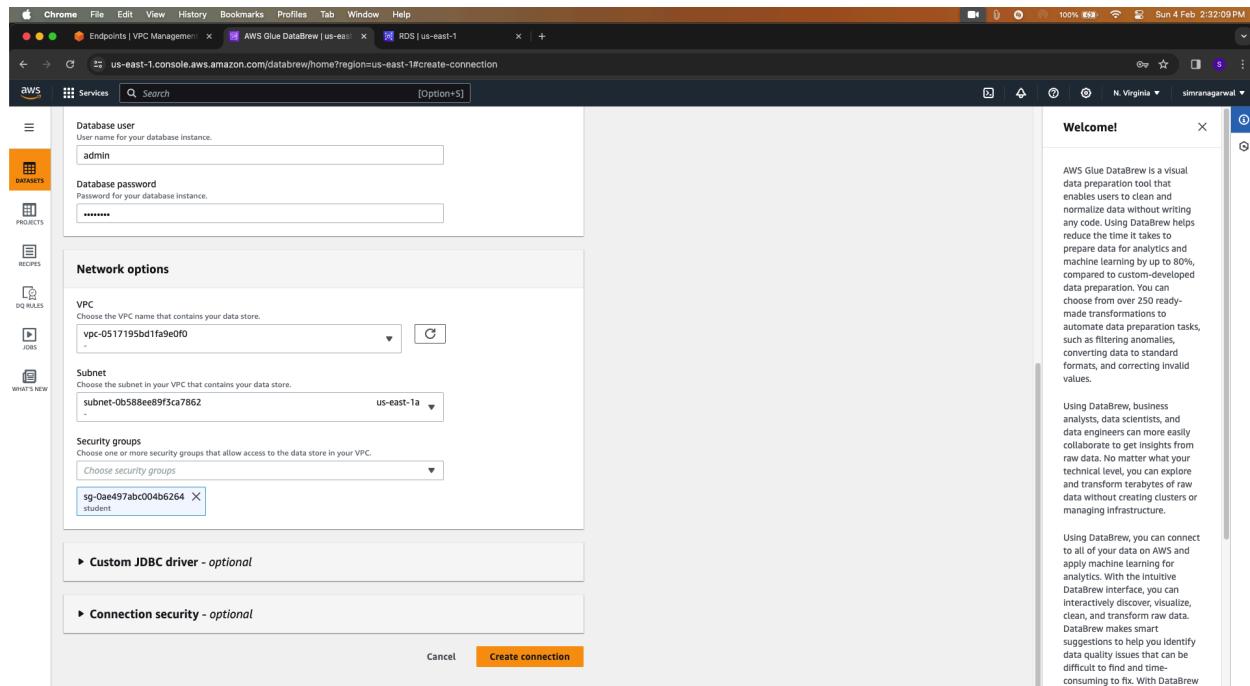
**Inbound rules (3):**

Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Description
-	sgr-071ee0a398a77f1c2	IPv4	All traffic	All	All	0.0.0.0/0	-
-	sgr-074447ff548be9a71	-	All TCP	TCP	0 - 65535	sg-0ae497abc004b62...	-
-	sgr-0fe12ad883a844511	IPv4	MySQL/Aurora	TCP	3306	106.213.82.95/32	-

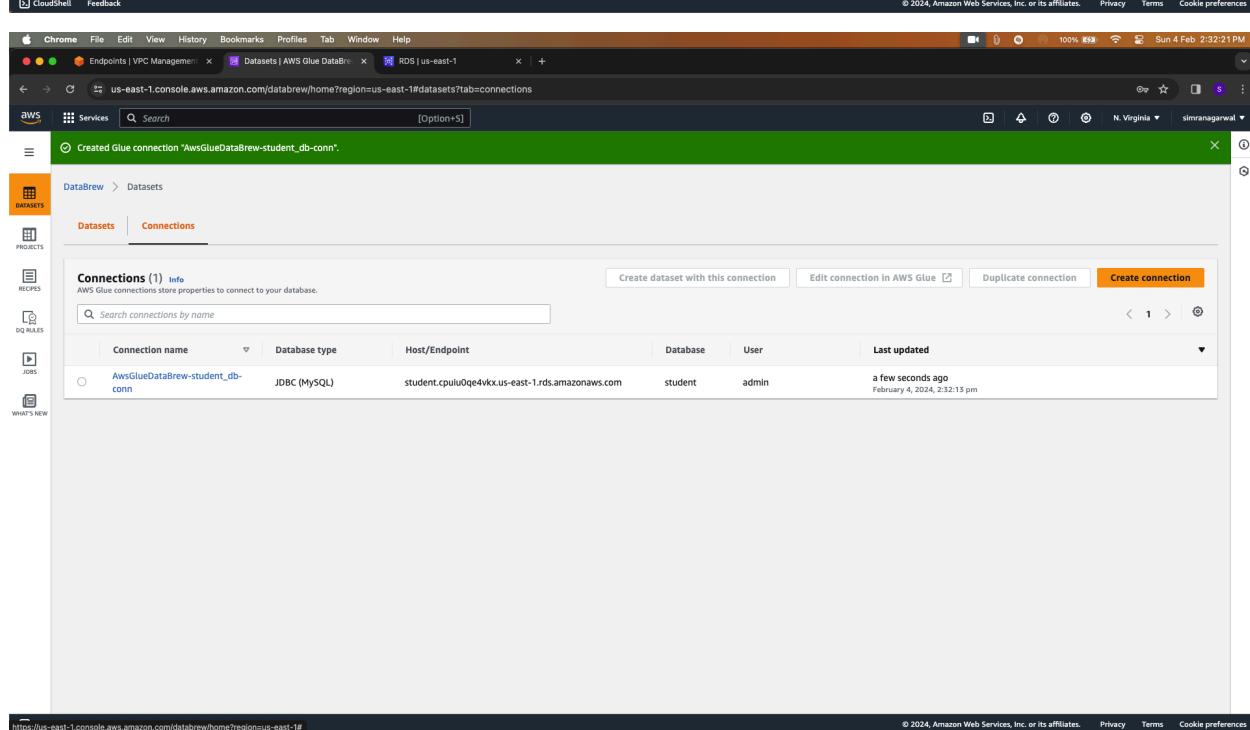
28. Go to Glue Databrew again, and create a connection, for this, you need to first click on Dataset then click on Connection and then Create Connection.



29. Proceed with JDBC Connection and fill out all the details asked in the form fields there.  
 (Please be cautious or a bit careful while you enter the JDBC URL, username and password as these are the three things which will help you connect to RDS Database from Glue DataBrew)

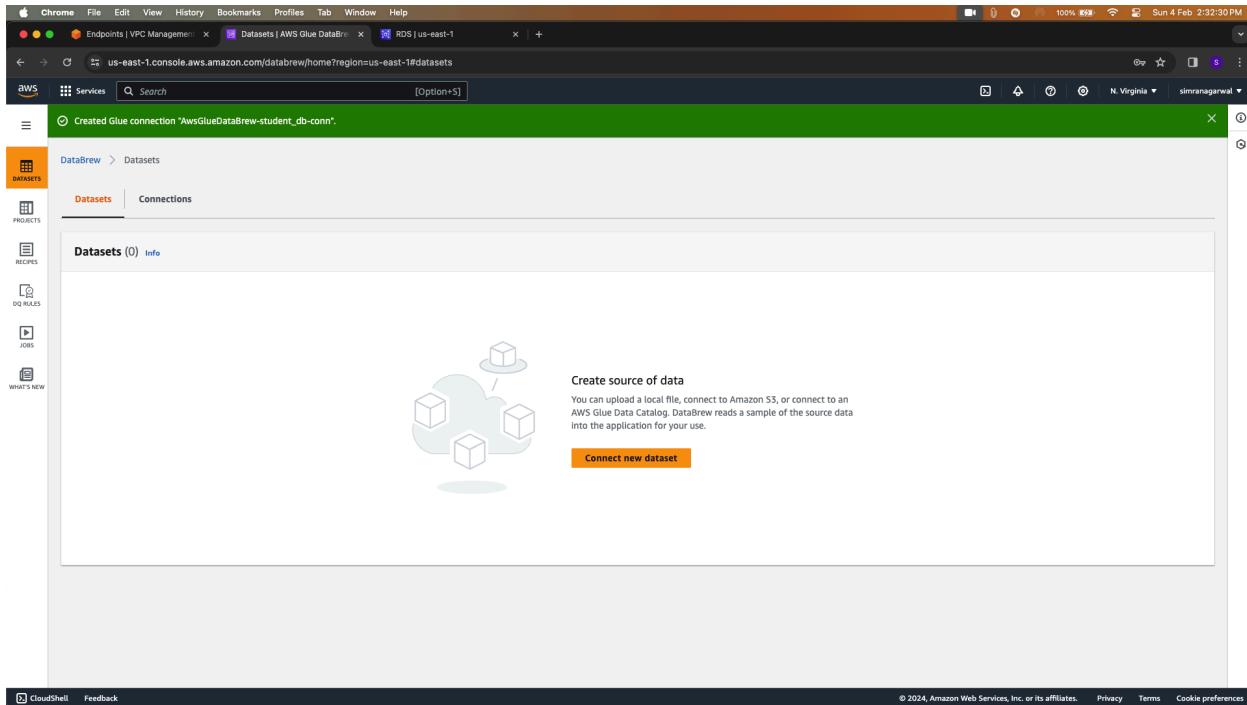


The screenshot shows the 'Create connection' dialog in AWS Glue DataBrew. It includes fields for 'Database user' (admin), 'Database password' (redacted), 'VPC' (vpc-0517195bd1fa9e0f0), 'Subnet' (subnet-0b58ee89f3ca7862), and 'Security groups' (sg-0ae497ab004db6264). Below these are sections for 'Custom JDBC driver - optional' and 'Connection security - optional'. A prominent orange 'Create connection' button is at the bottom right.

The screenshot shows the 'Connections' page in AWS Glue DataBrew. It displays a single connection entry: 'AwsGlueDataBrew-student\_db-conn'. The table includes columns for Connection name, Database type, Host/Endpoint, Database, User, and Last updated. The connection details are: Connection name 'AwsGlueDataBrew-student\_db-conn', Database type 'JDBC (MySQL)', Host/Endpoint 'student.cpuiu0qe4vxx.us-east-1.rds.amazonaws.com', User 'student', and Last updated 'a few seconds ago' (February 4, 2024, 2:32:15 pm).

30. As the connection is successfully created, it's time to create Datasets that connects or corresponds to the tables in RDS MySQL Database.



31. Now, you can create all three Datasets (corresponding to the three tables of the RDS MySQL “student” Database created by us.

- a. school-dataset for school\_tbl.
- b. student-dataset for student\_tbl.
- c. study-details-dataset for study\_details\_tbl.

**New dataset details**

**Dataset name**  
school-dataset

The dataset name must contain 1-255 characters. Valid characters are alphanumeric (A-Z, a-z, 0-9), hyphen (-), period (.), and space.

**Connect to new dataset**

**JDBC connections**

Connection name	Database type	Host/Endpoint	Database	User	Last updated
AwsGlueDataBrew-student_db-conn	MySQL	student.cpuuluQe4vk.us-east-1.rds.amazonaws.com	student	admin	a few seconds ago February 4, 2024, 2:32:13 pm

**New dataset details**

**Dataset name**  
school-dataset

The dataset name must contain 1-255 characters. Valid characters are alphanumeric (A-Z, a-z, 0-9), hyphen (-), period (.), and space.

**Connect to new dataset**

**JDBC connections**

Your selected connection  
AwsGlueDataBrew-student\_db-conn  
jdbc:mysql://student.cpuuluQe4vk.us-east-1.rds.amazonaws.com:3306/student

Table name  
Table within the database instance.  
school\_tb

Screenshot of the AWS DataBrew console showing the creation of a new dataset.

The browser window title is "us-east-1.console.aws.amazon.com/databrew/home?region=us-east-1#datasets". The URL is "https://us-east-1.console.aws.amazon.com/databrew/home?region=us-east-1#datasets".

The AWS DataBrew sidebar includes options for Databases, Projects, Recipes, DQ Rules, Jobs, and What's New.

The main content shows a success message: "Created dataset 'school-dataset'." Below it is a table titled "Datasets (1) info" with one row:

Dataset name	Data type	Data profile	Source	Location	Create date	Created by	Tags
school-dataset	Database table	-	Database connection	AwsGlueDataBrew-student_db-conn	a few seconds ago February 4, 2024, 2:32:57 pm	-	-

**New connection**

**New dataset details**

- Dataset name: student-dataset
- Description: The dataset name must contain 1-255 characters. Valid characters are alphanumeric (A-Z, a-z, 0-9), hyphen (-), period (.), and space.

**Connect to new dataset**

**JDBC connections**

- File upload
- Amazon S3
- Database connections
- Amazon Redshift
- JDBC**
- AWS Glue Data Catalog
- Data Catalog S3 tables
- Data Catalog Redshift tables
- Data Catalog RDS tables
- All AWS Glue tables
- Others

**Your selected connection**

AwsGlueDataBrew-student\_db-conn  
jdc:mysql://student.cpu0qeq4vkk.us-east-1.rds.amazonaws.com:3306/student

**Table name**

Table within the database instance.  
student\_tbl

**Welcome!**

AWS Glue DataBrew is a visual data preparation tool that enables users to clean and normalize data without writing any code. Using DataBrew helps reduce the time it takes to prepare data for analytics and machine learning by up to 80%, making it an efficient and cost-effective way to prepare data. You can choose from over 350 ready-made transformations to automate data preparation tasks, such as filtering anomalies, converting data to standard formats, and correcting invalid values.

Using DataBrew, business analysts, data scientists, and data engineers can more easily collaborate to get insights from your data. No matter what your technical level, you can explore and transform terabytes of raw data without creating clusters or managing infrastructure.

Using DataBrew, you can connect to all of your data on AWS and apply machine learning for analytics. With the intuitive DataBrew interface, you can interactively discover, visualize, clean, and transform raw data. DataBrew makes smart suggestions to help you identify data quality issues that can be difficult to find and time-consuming to fix. With DataBrew

Screenshot of the AWS Glue DataBrew console showing the creation of a new dataset.

**Created dataset "school-dataset".**

**Connect to new dataset**

**JDBC connections**

Table name: student\_tb[ ]

**Tags - optional**

**Create dataset**

**Datasets**

Dataset name	Data type	Data profile	Source	Location	Create date	Created by	Tags
student-dataset	Database table	-	Database connection	AwsGlueDataBrew-student_db-conn	a few seconds ago February 4, 2024, 2:33:22 pm	-	-
school-dataset	Database table	-	Database connection	AwsGlueDataBrew-student_db-conn	a few seconds ago February 4, 2024, 2:32:57 pm	-	-

The screenshot shows the AWS Glue DataBrew interface in a web browser. A success message 'Created dataset "student-dataset"' is displayed at the top. The main area is titled 'New connection' and shows 'New dataset details' with a dataset name 'study-detail-dataset'. Below this, under 'Connect to new dataset', a 'JDBC connections' section is selected, showing a connection named 'AwsGlueDataBrew-student\_db-conn'. The table name is set to 'study\_details\_tb'. The sidebar on the left includes sections for Projects, Recipes, DQ Rules, Jobs, and What's New, with 'DQASSETS' currently highlighted. A right-hand sidebar provides an 'Welcome!' overview of DataBrew's capabilities.

Screenshot of the AWS Glue DataBrew console showing the creation of a new dataset.

**Created dataset "student-dataset".**

**Connect to new dataset** Info

**JDBC connections** Info

AWS Glue connections store properties to connect to your database.

**Add JDBC connection**

**Your selected connection**  
AwsGlueDataBrew-student\_db-conn  
jdbcmysql://student.cpu0qge4vkk.us-east-1.rds.amazonaws.com:3306/student

**Table name**  
Table within the database instance.  
`study_details_tb`

**Tags - optional**  
Metadata that you can define and assign to AWS resources. Each tag is a simple label consisting of a customer-defined key (name) and an optional value. Using tags can make it easier for you to manage, search for, and filter resources by purpose, owner, environment, or other criteria.

**Create dataset**

**Datasets** Info

**Datasets (3)**

Dataset name	Data type	Data profile	Source	Location	Create date	Created by	Tags
study-detail-dataset	Database table	-	Database connection	AwsGlueDataBrew-student_db-conn	a few seconds ago February 4, 2024, 2:34:00 pm	-	-
student-dataset	Database table	-	Database connection	AwsGlueDataBrew-student_db-conn	a few seconds ago February 4, 2024, 2:33:22 pm	-	-
school-dataset	Database table	-	Database connection	AwsGlueDataBrew-student_db-conn	a minute ago February 4, 2024, 2:32:57 pm	-	-

32. Now when all three datasets are created in Glue DataBrew, it's time to create a project and name it "my-rds-proj".

The screenshot shows two consecutive screenshots of the AWS Glue DataBrew console.

**Screenshot 1:** The main DataBrew interface. A green banner at the top says "Created dataset 'study-detail-dataset'". The left sidebar has "PROJECTS" selected. The main area shows a placeholder image for a dataset and a button to "Create sample project".

**Screenshot 2:** The "Create project" dialog. It has tabs for "Project details" and "Recipe details". Under "Project details", the "Project name" field is filled with "my-rds-proj". Under "Attached recipe", there is a dropdown menu with "Create new recipe" selected, and a sub-menu item "my-rds-proj-recipe" is shown. There is also an option to "Import steps from recipe".

A "Welcome!" sidebar on the right provides an overview of DataBrew's features, mentioning its use for visual data preparation, machine learning, and analytics. It highlights the tool's ability to clean and normalize data without writing code, using over 250 ready-made transformations to automate tasks like filtering anomalies and correcting invalid values.

### 33. The form to create a project asks us to select a dataset.

The screenshot shows the AWS Glue DataBrew 'Create project' form. The main title is 'Created dataset "study-detail-dataset"'. The left sidebar has 'DATASETS' selected. The main area starts with a note: 'The recipe name must contain 1-255 characters. Valid characters are alphanumeric (A-Z, a-z, 0-9), hyphen (-), period (.), and space.' Below this is a checkbox for 'Import steps from recipe' and a note: 'Import recipe steps from an existing recipe into your project. The existing recipe that you chose will not be edited.' A section titled 'Select a dataset' follows, with a note: 'Select the dataset that you want to work on'. It contains three buttons: 'My datasets' (selected, showing 'Your imported datasets'), 'Sample files' (Explore example files for your dataset), and 'New dataset' (Import new dataset). Below these are three cards: 'Find datasets' (with a search bar), 'Dataset name' (dropdown), 'Data type' (dropdown), 'Source' (dropdown), and 'Create date' (dropdown). The first card shows a list of datasets:

Dataset name	Data type	Source	Create date
<input checked="" type="radio"/> study-detail-dataset	Database table	Database connection	a few seconds ago February 4, 2024, 2:34:00 pm
<input type="radio"/> student-dataset	Database table	Database connection	a minute ago February 4, 2024, 2:33:22 pm
<input type="radio"/> school-dataset	Database table	Database connection	2 minutes ago February 4, 2024, 2:32:57 pm

Below this are sections for 'Sampling - optional' (Select the type and size of your sample) and 'Tags - optional' (Metadata that you can define and assign to AWS resources). A 'Permissions' section follows, with a note: 'DataBrew needs permission to connect to data on your behalf. Use an IAM role with the required policy attached.' It shows a dropdown for 'Role name' set to 'AWSGlueDataBrewServiceRole-student'. A note below says: 'By clicking "Create project" you are authorizing DataBrew to add required permissions to access all the datasets in this project to the selected service role.' At the bottom are 'Cancel' and 'Create project' buttons. A 'Welcome!' sidebar on the right provides general information about AWS Glue DataBrew.

34. Wait for the project report to open and then we can perform operations like join and filter.

The screenshot shows two views of the AWS Glue DataBrew interface. The top view displays a 'Provisioning compute' dialog box indicating a 0% completion rate. The bottom view shows a completed dataset named 'study-detail-dataset' with 500 rows. The dataset contains columns: ABC\_school\_id, ABC\_student\_id, #\_study\_time\_in\_hr, ABC\_health, and ABC\_internet. The ABC\_internet column has values: good (116), average (104), worst (102), and all other values (178). The ABC\_health column has values: good (500), average (497), and worst (1). The ABC\_school\_id and ABC\_student\_id columns show distinct counts of 4 and 500 respectively. The #\_study\_time\_in\_hr column shows a distribution with Median at 4, Mean at 3.89, Mode at 4, and Max at 8. The bottom view also includes a 'Recipe (0)' panel and a 'Build your recipe' section.

Column	Type	Value
ABC_school_id	Distinct	4
ABC_student_id	Distinct	500
#_study_time_in_hr	Total	500
ABC_health	Distinct	5
ABC_internet	Total	500
	Distinct	2
	Unique	0

Value	Count
good	116
average	104
worst	102
all other values	178

Column	Type	Value
ABC_school_id	Distinct	4
ABC_student_id	Distinct	500
#_study_time_in_hr	Total	500
ABC_health	Distinct	5
ABC_internet	Total	500
	Distinct	2
	Unique	0

Value	Count
good	500
average	497
worst	1

35. As the report have opened and we can start performing our operations, click on Join in the bar above the report.

The screenshot shows the AWS Glue DataBrew interface. On the left, there's a sidebar with 'PROJECTS', 'RECIPES', 'DQ RULES', and 'JOBS'. The main area displays a dataset named 'study-detail-dataset' with 500 rows. The data is presented in a grid format with columns: 'ABC school\_id', 'ABC student\_id', '# study\_time\_in\_hr', 'ABC health', and 'ABC internet'. Each column has summary statistics (e.g., Total, Distinct, Unique) and visualizations like histograms and box plots. Below the grid, there's a 'SAMPLE' button. To the right, a 'RECIPES' panel shows a single step named 'my-rds-proj-recipe' with a status of 'Working version'. At the bottom, there's a 'Build your recipe' section with a 'Add step' button.

36. Select the student-dataset and click on Next.

The screenshot shows the AWS Glue DataBrew interface. The top navigation bar includes 'File', 'Edit', 'View', 'History', 'Bookmarks', 'Profiles', 'Tab', 'Window', and 'Help'. The address bar shows the URL: 'aws.us-east-1.console.aws.amazon.com/databrew/home?region=us-east-1#project-workspace?project=my-rds-proj&view=grid'. The main content area is titled 'my-rds-proj' and shows a 'Join' step. The 'Select dataset' dropdown is set to 'student-dataset'. A 'Dataset preview' table is displayed, showing the following data:

#	student_id	first_name	last_name	gender	age
424	Deena	Arnoldi	F	14	
323	Morris	Rouzet	M	11	
480	Rosamund	Corzor	F	10	
537	Westbrook	Polglase	M	10	
395	Ollie	Phizackerly	M	16	

At the bottom right of the preview table are 'Cancel' and 'Next' buttons. The footer of the page includes links for 'cloudShell', 'Feedback', and copyright information: '© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences'.

37. Select Left Join and the remove the duplicate (or the common) field from table B. Similary join the third table with the left join and excluding the duplicate / common field from table.

The screenshot shows the AWS Glue DataBrew interface for a project named "my-rds-proj". The "Join" step is currently active. In the "Step 1 Select dataset" section, "Table A (this project)" is selected. In the "Step 2 Specify join details" section, "Left join" is selected. The "Join keys" section shows "Table A student\_id" and "Table B student\_id" both mapped to "student\_id". The "Column list" section lists columns from both tables: "Table A country", "Table A year", "Table A # marks", "Table B student\_id", and "Table B first\_name". The "Joined table preview" section shows a sample of the joined data.

The screenshot shows the "Select dataset" step configuration. "school-dataset" is selected as the dataset to join. The "Dataset name" is "school-dataset", "Data source" is "JDBC (MySQL) by AWS Glue Connection", "Location" is "AwsGlueDataBrew-student\_db-conn", and "User" is "admin". The "Custom driver location" is set to "-". The "Database table name" is "school\_tbl". The "Created by" and "Last modified by" fields show the date and time of creation. The "Dataset preview" section shows a table with data from the "school\_tbl" table.

The screenshot shows the AWS Glue DataBrew interface for creating a new project named "my-rds-proj". The left sidebar includes options for DATASETS, RECIPES, DQ RULES, JOBS, and WHAT'S NEW. The main area displays a success message: "Created project 'my-rds-proj'." Below this, the "Join" step is selected. Step 1, "Select dataset", shows a single dataset named "study-detail-dataset". Step 2, "Specify join details", is currently active. It features a "Select join type" section with seven options: Inner join (selected), Left join, Right join, Outer join, Left excluding join, and Right excluding join. To the right, the "Join keys" section shows "Table A (this project)" set to "study-detail-dataset" and "Table B" set to "school-dataset". Buttons for "Add another join key", "Cancel", "Previous", and "Finish" are visible at the bottom.

Screenshot of AWS Glue DataBrew interface showing the creation of a project and data preview.

**Top Panel (Project Creation):**

- Project Name: "my-rds-proj".
- Join Info Step 1: Select dataset (AWS SCHOOL\_ID).
- Join Info Step 2: Specify join details (Left join, Right join, Outer join, Left excluding join, Right excluding join, Outer excluding join).
- Column list: Includes columns from Table A (year, marks, first\_name, last\_name, gender, age) and Table B (school\_id, name).
- Buttons: Cancel, Previous, Finish.

**Bottom Panel (Data Preview):**

- Dataset: study-detail-dataset | Sample: First n sample (500 rows).
- Sample View: Shows 500 rows of data for columns first\_name, last\_name, gender, and school\_id.
- Merge Columns Dialog: Set up to merge first\_name and last\_name into a new column named "Merged column 1".
- Transform Options: All rows (500 rows) or Filtered rows - 0 filters applied (500/500 rows).
- Buttons: Create Job, IMAGE, ACTIONS.

38. Now we will click on first name field and then click on merge with the last name field and rename the resulting field as student\_name with a separator as a “ “ (space bar).

The screenshot shows the AWS Glue DataBrew interface. On the left, there's a sidebar with 'PROJECTS', 'RECIPES', 'DQ RULES', and 'JOBS'. The main area displays a dataset named 'study-detail-dataset' with 500 rows. Three columns are shown: 'first\_name', 'last\_name', and 'gender'. A 'Merge columns' dialog is open over the grid, containing fields for 'first\_name' and 'last\_name', a 'Separator - Optional' field with a space character, and a 'New column name' field set to 'student\_name'. Below these, transformation options like 'All rows (500 rows)' or 'Filtered rows - 0 filters applied (500/500 rows)' are available. At the bottom right of the dialog is an 'Apply' button.

39. We need to filter out the high performing students, so for that we will choose the marks field and choose the filter option and within that we will choose the greater than filter and enter the value as “60”.

Resulting Report will have the details of all the high performing students who scored more than 60.

Screenshot of AWS Glue DataBrew interface showing two project configurations for "my-rds-proj".

**Project 1 (Top):**

- Dataset:** study-detail-dataset
- Sample:** First n sample (500 rows)
- Columns:** ABC\_student\_name, ABC\_gender, age
- Data Preview:** Shows 500 rows of student data with columns ABC\_student\_name, ABC\_gender, and age.
- Filter Values (Right Panel):**
  - Source column: marks
  - Filter condition: Greater than 60
  - Preview changes: Shows a histogram of marks distribution.

**Project 2 (Bottom):**

- Dataset:** study-detail-dataset
- Sample:** First n sample (185 rows)
- Columns:** ABC\_student\_name, ABC\_gender, age
- Data Preview:** Shows 185 rows of student data with columns ABC\_student\_name, ABC\_gender, and age.
- Filter Values (Right Panel):**
  - Source column: marks
  - Filter condition: Greater than 60
  - Preview changes: Shows a histogram of marks distribution.

40. Now that you are ready to run the job you first need to create an S3 bucket for the job to store the resulting dataset.

Search for S3 in AWS Console Search Bar and click on S3 Buckets.

The screenshot shows the AWS Glue DataBrew console with a search bar at the top containing the text 's3'. The search results are displayed in two main sections: 'Services' and 'Features'.

**Services** section:

- S3: Scalable Storage in the Cloud
- S3 Glacier: Archive Storage in the Cloud
- AWS Snow Family: Large Scale Data Transport
- Storage Gateway: Hybrid Storage Integration

**Features** section:

- Imports from S3
- DynamoDB feature
- Batch Operations
- S3 feature
- Buckets
- S3 feature
- Access points
- S3 feature

On the right side of the screen, there is a 'Welcome!' panel with the following text:

AWS Glue DataBrew is a visual data preparation tool that enables users to clean and normalize data without writing any code. Using DataBrew helps reduce the time it takes to prepare data for analytics and machine learning by up to 80%, compared to custom-developed data preparation. You can choose from over 250 ready-made transformations to automate data preparation tasks, such as filtering anomalies, converting data to standard formats, and correcting invalid values.

Using DataBrew, business analysts, data engineers, and data scientists can more easily collaborate to get insights from raw data. No matter what your technical level, you can explore and transform terabytes of raw data without creating clusters or managing infrastructure.

At the bottom of the screen, the URL is shown as <https://s3.console.aws.amazon.com/s3/buckets?regions=us-east-1>.

41. Click on create a bucket.

The screenshot shows the AWS S3 console interface. On the left, there is a navigation sidebar with various options like Buckets, Storage Lens, and Feature spotlight. The main content area is titled 'Amazon S3 > Buckets'. It features an 'Account snapshot' section with a link to 'View Storage Lens dashboard'. Below this is a tab bar with 'General purpose buckets' selected, followed by 'Directory buckets'. A search bar labeled 'Find buckets by name' is present. A table header includes columns for 'Name', 'AWS Region', 'Access', and 'Creation date'. A message at the bottom states 'No buckets' and 'You don't have any buckets.' A prominent orange 'Create bucket' button is located at the bottom right of the table area. The top of the browser window shows the URL 's3.console.aws.amazon.com/s3/buckets?region=us-east-1#' and the AWS logo.

42. Choose a desired name for the bucket and fill out the required the details. Remember not to block the public access.

The screenshot shows the 'Create bucket' wizard in the AWS S3 console. The 'General configuration' step is selected, showing the following fields:

- AWS Region:** US East (N. Virginia) us-east-1
- Bucket type:** General purpose (selected)
- Bucket name:** bucket090920

Below these, under 'Copy settings from existing bucket - optional', there is a 'Choose bucket' button and a note about the format: \$2://bucket/prefix.

Under 'Object Ownership', the 'ACLs disabled (recommended)' option is selected, with a note that all objects are owned by the account owner.

At the bottom of the page, there are links for 'cloudShell', 'Feedback', and copyright information: © 2024, Amazon Web Services, Inc. or its affiliates.

Screenshot of the AWS S3 Bucket Creation wizard, Step 2: Set Bucket Settings.

**Object Ownership**

**ACLs disabled (recommended)**  
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

**ACLs enabled**  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

**Object Ownership**  
Bucket owner enforced

**Block Public Access settings for this bucket**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all, in order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

**Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLS)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLS)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

**Warning:** Turning off block all public access might result in this bucket and the objects within becoming public.  
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

**Bucket Versioning**

Versions is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

**Bucket Versioning**

**Disable**

**Enable**

**Tags - optional**

You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

[Add tag](#)

**Default encryption** Info

Server-side encryption is automatically applied to new objects stored in this bucket.

Screenshot of the AWS S3 Bucket Creation Wizard:

**Tags - optional (0)**  
 You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

[Add tag](#)

**Default encryption** Info  
 Server-side encryption is automatically applied to new objects stored in this bucket.

**Encryption type** Info

- Server-side encryption with Amazon S3 managed keys (SSE-S3)
- Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)  
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).

**Bucket Key**  
 Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

- Disable
- Enable

**Advanced settings**

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

[Cancel](#) [Create bucket](#)

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS S3 Buckets List:

**Successfully created bucket "bucket090920"**  
 To upload files and folders, or to configure additional bucket settings, choose [View details](#).

[View details](#)

[Amazon S3](#) > [Buckets](#)

**Account snapshot**  
 Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

[General purpose buckets](#) [Directory buckets](#)

**General purpose buckets (1) Info**  
 Buckets are containers for data stored in S3. [Learn more](#)

Name	AWS Region	Access	Creation date
bucket090920	US East (N. Virginia) us-east-1	Objects can be public	February 4, 2024, 14:46:44 (UTC+05:30)

[Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

43. We have successfully created a bucket here.

The screenshot shows a Chrome browser window with several tabs open, including 'Endpoints | VPC Management', 'AWS Glue DataBrew | us-east-1', 'bucket090920 - S3 bucket', and 'RDS | us-east-1'. The main content area is the AWS S3 console, specifically the 'Objects' tab for the bucket 'bucket090920'. The URL in the address bar is 's3.console.aws.amazon.com/s3/buckets/bucket090920?region=us-east-1&bucketType=general&tab=objects'. The page displays a message: 'No objects' and 'You don't have any objects in this bucket.' There is a prominent orange 'Upload' button at the bottom. The top navigation bar includes links for 'Objects', 'Properties', 'Permissions', 'Metrics', 'Management', and 'Access Points'. The status bar at the bottom right shows 'Sun 4 Feb 2:46:55 PM'.

44. Now click on create a job and then select the output location as the location of the S3 bucket. After configuring all the settings for the job click on “**Create and run the job**”.

The screenshot shows the AWS Glue DataBrew interface for creating a new job. The 'Associated recipe' dropdown is open, displaying the 'Browse S3' option with the path 'S3 Buckets > bucket090920'. The 'Output' section is configured to output to 'Amazon S3' using 'PARQUET' file format. The 'Advanced job settings - optional' section is expanded, showing the 'Associated schedule' field. The 'Permissions' section indicates that the role 'AWSGlueDataBrewServiceRole-student' has been selected. At the bottom, the 'Create and run job' button is highlighted in orange.

Screenshot of the AWS DataBrew console showing the creation of a recipe job named "top-performer-student".

The top section shows the AWS Lambda service page with tabs for Endpoints, VPC Management, AWS Glue DataBrew, bucket090920 - S3 bucket, IAM Global, and RDS us-east-1.

The main view displays the "my-rds-proj" project. A dataset named "study-detail-dataset" is selected, showing its first n sample (185 rows). The schema includes columns: ABC school\_id, ABC student\_id, # study\_time\_in\_hr, ABC health, and ABC internet.

The "RECIPES" tab is active, showing a "Recipe (4)" named "my-rds-proj-recipe". The steps are:

- Left join student-dataset
- Left join school-dataset
- Merge columns first\_name, last\_name into student\_name separated by " "
- Filter values by marks

The bottom section shows the "Jobs" tab, where the "Recipe jobs (1) info" table lists the "top-performer-student" job. The job is running and was created a few seconds ago on February 4, 2024, at 2:47:58 pm.

45. After you see the job running status as succeed, you should check the S3 Bucket whether the job has created the resulting dataset in the output location (i.e. S3 Bucket) mentioned in the earlier steps.

The screenshot shows two browser tabs. The top tab is the AWS DataBrew 'Jobs' page, displaying a single 'Recipe jobs' entry:

Job name	Status	Job input	Job output	Last run	Created on
top-performer-student	Succeeded	my-rds-proj ( study-detail... + my-rds-proj-r... )	1 output	a few seconds ago	4 minutes ago

The bottom tab is the AWS S3 'Buckets' page, showing the contents of the 'bucket090920' bucket:

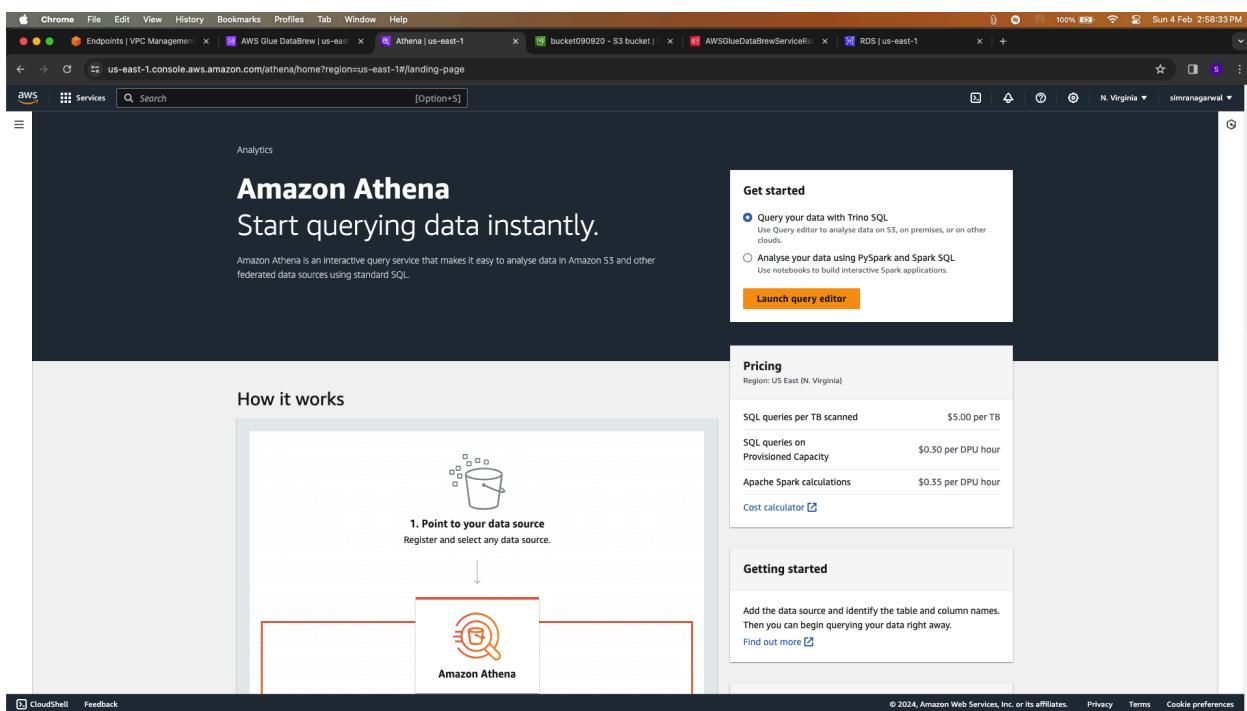
Name	Type	Last modified	Size	Storage class
top-performer-student_04Feb2024_1707038393427/	Folder	-	-	-

We have successfully created a resulting dataset in the S3 Bucket, now this S3 bucket will act as

an input source for the Athena to read and map data into its table and then use it for querying and analysis and provide the resulting data in another S3 Bucket.

46. Before starting with Amazon Athena we will be requiring an S3 Bucket for storing the output, so for that you need to create another S3 Bucket ( follow the process from step number 41.)

47. Open Amazon Athena and click on Launch query editor.



First of all, in the settings click on settings and choose the output location as the newly created S3 Bucket.

So now, we have two S3 Buckets - The first one which was the output bucket for DataBrew will act as the input bucket for Athena and the freshly created S3 Bucket will act as output bucket for Athena.

48. Now click on Create and select S3 Bucket Data and then start populating the fields in the form starting with table and database name..

The screenshot shows the AWS Athena Query Editor interface. The top navigation bar includes tabs for 'Endpoints | VPC Management', 'AWS Glue DataBrew | us-east-1', 'Query editor tabs | Athena', 'bucket090920 - S3 bucket', 'AWSGlueDataBrewServiceRole', and 'RDS | us-east-1'. The main area has a dark blue header with the title 'Amazon Athena > Query editor tabs'. Below the header, there are tabs for 'Editor', 'Recent queries', 'Saved queries', and 'Settings'. A 'Workgroup' dropdown is set to 'primary'. On the left, a sidebar shows 'Data' selected, with options like 'Create a table from data source' (which is currently 'S3 bucket data'), 'AWS Glue Crawler', 'Create with SQL', 'CREATE TABLE', 'CREATE TABLE AS SELECT', 'CREATE TABLE AS SELECT(ICEBERG)', and 'CREATE VIEW'. Below the sidebar are sections for 'Tables and views' (with 'Tables (0)' and 'Views (0)') and a 'SQL' editor with a single line of code 'SQL Ln 1, Col 1'. At the bottom, there are buttons for 'Run', 'Explain', 'Cancel', 'Clear', and 'Create'. A note at the bottom right says 'Reuse query results up to 60 minutes ago'.

Table Name: top\_performer\_study\_details, Database Name: student.

The screenshot shows the 'Create table from S3 bucket data' configuration form. The top navigation bar is identical to the previous screenshot. The main form has two main sections: 'Table details' and 'Database configuration'. In the 'Table details' section, the 'Table name' field is filled with 'top\_performer\_study\_details'. A note below it states: 'Table name must be from 1-128 characters and must be unique. Valid characters are a-z, A-Z, 0-9, \_ (underscore). Table names tend to correspond to the directory where the data will be stored.' The 'Description - Optional' field contains the placeholder 'Type something'. A note below it says: 'Table description must be from 1-1024 characters. 1024 characters remaining.' In the 'Database configuration' section, there is a note: 'Choose an existing database or create a new database. Choose to access an existing database or to create a new database in order to create a new table. Athena stores the table schema in the AWS Glue Data Catalogue.' Two radio buttons are present: 'Create a database' (selected) and 'Choose an existing database'. The 'Database name' field is filled with 'student'. A note below it states: 'Database name must be from 1-128 characters and must be unique. Valid characters are a-z, A-Z, 0-9, \_ (underscore).'

49. Choose the Dataset as the path for the first bucket which is now an input source for Athena.

The screenshot shows the 'Create table from S3 bucket' configuration page in the AWS Athena console. The 'Dataset' tab is selected. In the 'Location of input data set' section, the path 'bucket090920/top-performer-student\_04feb2024\_170703839542' is entered. Below it, a note specifies that the path must be unique and valid characters are a-z, A-Z, 0-9, and underscore. The 'Data format' section includes 'Table type: Apache Hive', 'File format: Apache Parquet', and 'SerDe library: org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe'. The 'Column details' section notes that column names must be unique and valid. At the bottom, there are links for 'cloudShell', 'Feedback', and copyright information.

50. For creating the table, you need to add all the column with their datatypes and lengths.

The screenshot shows two configurations for creating a table from an S3 bucket in the AWS Glue DataBrew console. Both configurations have the same basic structure: a top section for file format (Apache Parquet) and SerDe library (org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe), and a bottom section for Column details.

**Configuration 1 (Top):**

- File format:** Apache Parquet
- Serde library:** org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe
- Column details:**
  - Column name:** school\_id, **Column type:** varchar, **Description - Optional:** Enter description
  - Length:** 15
  - Column name:** student\_id, **Column type:** varchar, **Description - Optional:** Enter description
  - Length:** 3
  - Column name:** study\_time\_in\_hr, **Column type:** int, **Description - Optional:** Enter description

**Configuration 2 (Bottom):**

- File format:** Apache Parquet
- Serde library:** org.apache.hadoop.hive.ql.io.parquet.serde.ParquetHiveSerDe
- Column details:**
  - Column name:** school\_id, **Column type:** varchar, **Description - Optional:** Enter description
  - Length:** 15
  - Column name:** student\_id, **Column type:** varchar, **Description - Optional:** Enter description
  - Length:** 3
  - Column name:** study\_time\_in\_hr, **Column type:** int, **Description - Optional:** Enter description
  - Column name:** health, **Column type:** varchar, **Description - Optional:** Enter description
  - Length:** 15
  - Column name:** internet, **Column type:** varchar, **Description - Optional:** Enter description
  - Length:** 10
  - Column name:** country, **Column type:** varchar, **Description - Optional:** Enter description
  - Length:** 20
  - Column name:** year, **Column type:** varchar, **Description - Optional:** Enter description

The screenshot shows the 'Create table from S3 bucket' interface in the AWS Athena console. The user has defined six columns:

- Internet**: Varchar, Length 10
- country**: Varchar, Length 20
- year**: Varchar, Length 4
- marks**: Int, Length 10
- student\_name**: Varchar, Length 50
- gender**: Varchar, Length 10
- age**: Int, Length 10

Below the column definitions, there are buttons for 'Add a column' and 'Bulk add columns'. The interface also includes sections for 'Table properties - Optional' (Partitions and Bucketing) and 'Bucketing - Optional'.

After adding all the columns with their datatypes and lengths click on create table.

The screenshot shows the 'Create table' wizard in the AWS Athena console. The table definition is as follows:

```
1   name      varchar(45),
2   year      varchar(4),
3   marks     int,
4   student_name  varchar(50),
5   gender    varchar(10),
6   age       int,
7   name      varchar(45)
8 )
9 ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.ParquetHiveSerDe'
10 STORED AS INPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MappedParquetInputFormat'
11   OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.parquet.MappedParquetOutputFormat'
12 LOCATION 's3://bucket090920/top-performer-student_04Feb2024_1707038393427/'
13 TBLPROPERTIES ('classification' = 'parquet');
```

The 'Preview table query' section shows the generated SQL code. At the bottom right, there are 'Cancel' and 'Create table' buttons.

51. Now we can check whether we created the table correctly or not, this can be verified by running the following query:

“`select * from student.top_performer_study_details;`”

The screenshot shows two instances of the AWS Athena Query Editor interface. Both instances are running the same query:

```
1 | select * from student.top_performer_study_details;
```

**Query Editor 1 (Top):**

- Data Source:** AwsDataCatalog
- Database:** student
- Tables and views:** top\_performer\_study\_details
- Query Results:** Completed. Time in queue: 111 ms, Run time: 438 ms, Data scanned: 9.31 KB.

**Query Editor 2 (Bottom):**

- Data Source:** AwsDataCatalog
- Database:** student
- Tables and views:** top\_performer\_study\_details
- Results:** 221 rows returned. The table structure is as follows:

#	school_id	student_id	study_time_in_h	health	internet	country	year	student_name	gender	age	name
1	02-6957378	222	7	worst	TRUE	United States	2019	Case Mioni	M	13	Lotus Public School
2	33-9516954	565	7	bad	TRUE	India	2018	Kevina Barberow	M	15	Rose International
3	33-9516954	204	7	good	TRUE	India	2019	Amos Jagger	F	15	Rose International
4	33-9516954	214	7	good	FALSE	India	2019	Dwayne Zini	M	13	Rose International
5	02-6957378	213	8	average	TRUE	United States	2019	Kerr Kingswell	M	11	Lotus Public School
6	02-6957378	351	8	worst	FALSE	United States	2019	Isabella Somers	M	16	Lotus Public School
7	33-9516954	225	8	worst	FALSE	India	2019	Etti Lamprecht	M	10	Rose International
8	33-9516954	577	7	good	FALSE	India	2018	Darnell Matley	M	10	Rose International
9	33-9516954	121	7	worst	TRUE	India	2020	Bonny Johnston	F	10	Rose International
10	33-9516954	272	7	average	FALSE	India	2019	Donavon Pee	F	10	Rose International

52. Now search for QuickSight in AWS Console, and then click on QuickSight.

The screenshot shows a Chrome browser window with multiple tabs open, including the AWS Glue DataBrew service. The main content area displays search results for 'quicksu'. The first result is 'QuickSight' under the 'Services' category, which is highlighted with a blue border. The 'QuickSight' card includes a star icon and the text 'Fast, easy to use business analytics'. Below this, there are other service cards: 'Amazon Machine Learning', 'CodeStar', and 'Application Composer'. The 'Features' section contains cards for 'Quick start', 'Quick Setup', 'AWS B2B Data Interchange - Quick Setup', and 'Block storage disks'. The 'Resources' section has a link for 'Focused search'. To the right of the search results, there is a data preview pane showing a table with columns: country, year, student\_name, gender, age, and name. The table lists 10 rows of data from various countries and years, such as United States (2019), India (2018), and India (2019). At the bottom of the page, there are links for 'cloudShell', 'Feedback', and copyright information.

53. Now click on New Analysis, then click on Athena from the list of Datasets

The image consists of two screenshots of the AWS QuickSight web interface.

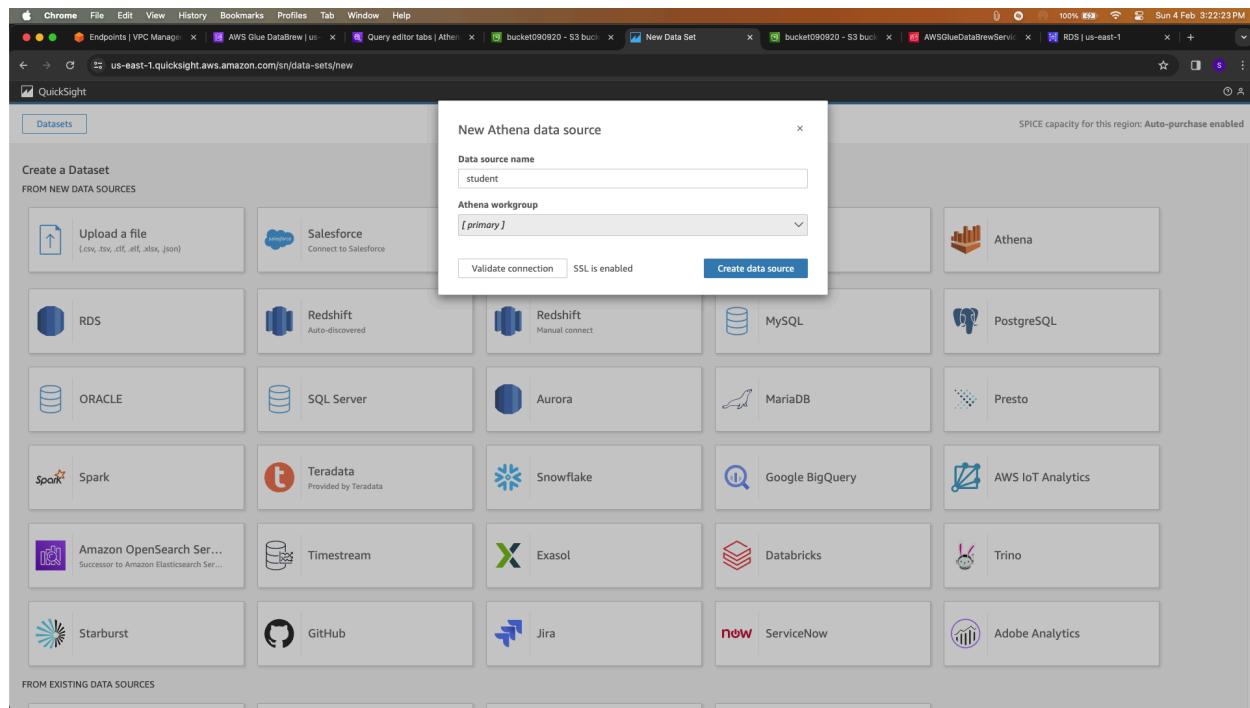
The top screenshot shows the 'Analyses' page. On the left, there is a sidebar with navigation links: Favorites, Recent, My folders, Shared folders, Dashboards, Analyses (which is selected and highlighted in blue), Datasets, Topics, and Community. The main area displays four sample analyses: 'Web and Social Media Analysis' (bar chart), 'People Overview analysis' (pie chart), 'Business Review analysis' (line chart), and 'Sales Pipeline analysis' (bar chart). Each sample has a 'SAMPLE' button and a three-dot menu icon.

The bottom screenshot shows the 'Create a Dataset' page. At the top, it says 'Create a Dataset FROM NEW DATA SOURCES'. Below this is a grid of 20 data source icons, each with a name and a brief description. The icons are arranged in five rows and four columns:

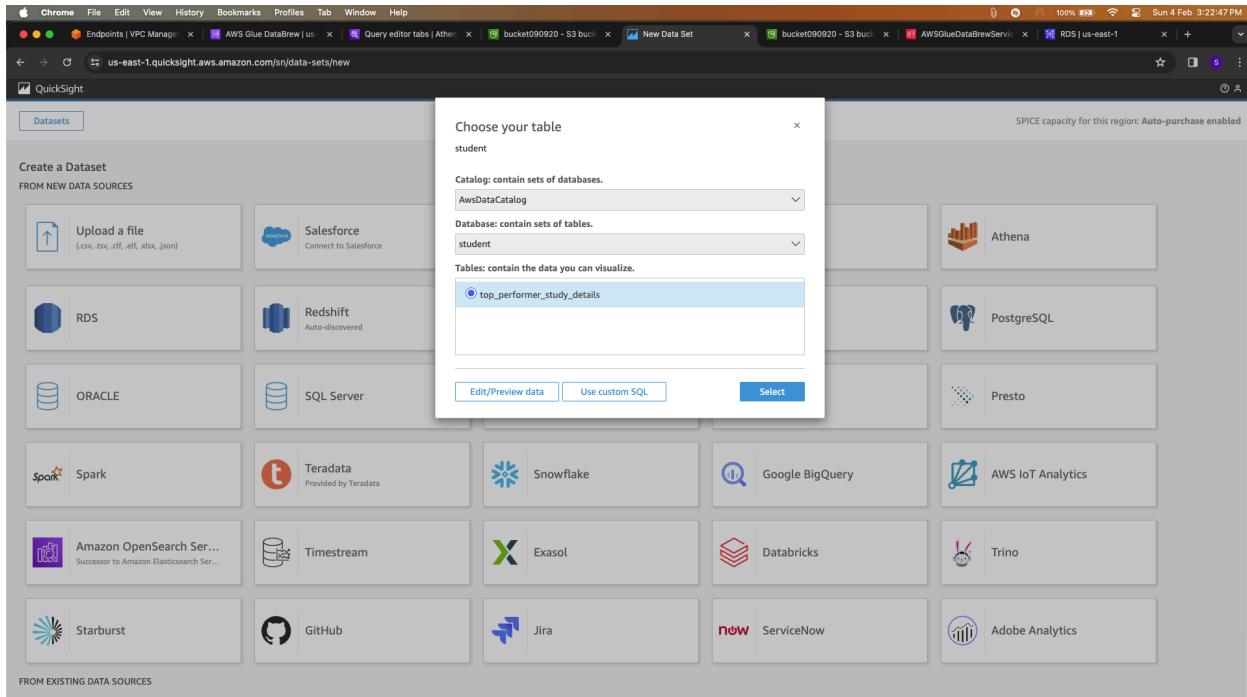
- Row 1: Upload a file (CSV, TSV, CTZ, ELF, ATLSX, JSON), Salesforce (Connect to Salesforce), S3 Analytics, S3.
- Row 2: RDS, Redshift (Auto-discovered), Redshift (Manual connect), MySQL.
- Row 3: ORACLE, SQL Server, Aurora, MariaDB.
- Row 4: Spark, Teradata (Provided by Teradata), Snowflake, Google BigQuery.
- Row 5: Amazon OpenSearch Service (Successor to Amazon Elasticsearch Service...), Timestream, Exasol, Databricks.
- Row 6: Starburst, GitHub, Jira, ServiceNow.
- Row 7: Adobe Analytics.

At the bottom of the page, there is a link: <https://us-east-1.quicksight.aws.amazon.com/sn/data-sets>.

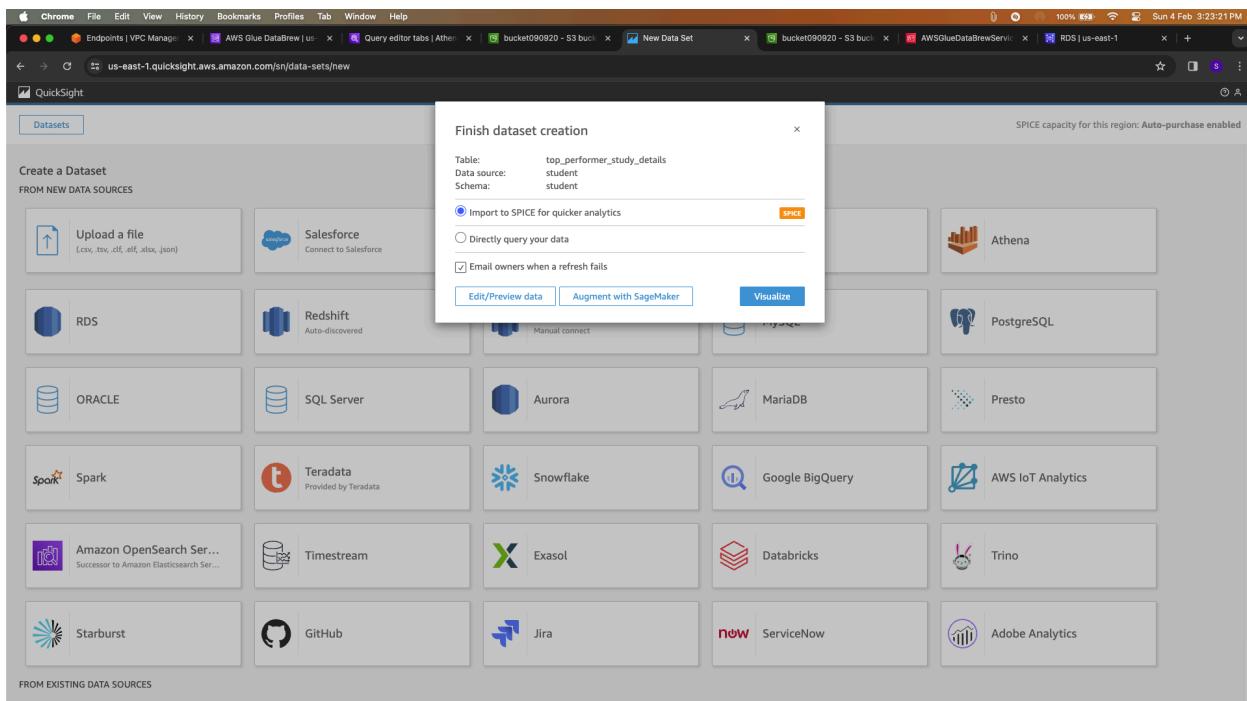
54. Enter the details like Data source name as “student” and Athena Workgroup as “primary”.



55. If we are doing it right we should see the table that we created in Athena (i.e. top\_performer\_study\_details).



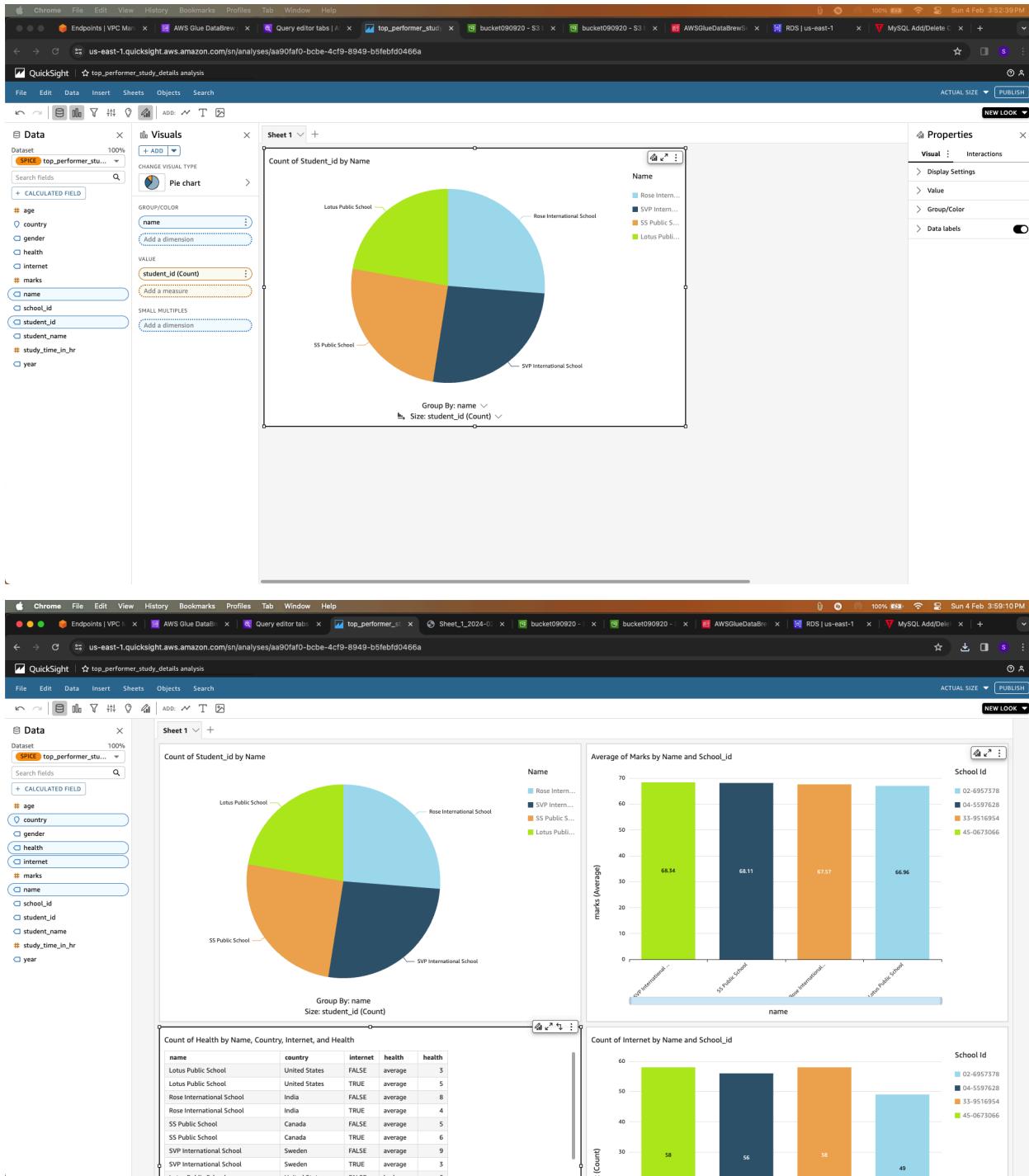
The screenshot shows the 'Choose your table' dialog box in the AWS QuickSight interface. The 'Tables' section contains a single item: 'top\_performer\_study\_details'. Below the table selection are two buttons: 'Edit/Preview data' and 'Select'. To the right of the dialog box is a grid of data source icons, including 'Athena', 'PostgreSQL', 'Presto', 'AWS IoT Analytics', 'Trino', and 'Adobe Analytics'. The background shows the 'Create a Dataset' page with various data source options like RDS, Oracle, Spark, and Amazon OpenSearch Service.



The screenshot shows the 'Finish dataset creation' dialog box. It displays the selected table ('top\_performer\_study\_details'), data source ('student'), and schema ('student'). There are three options for visualization: 'Import to SPICE for quicker analytics' (selected), 'Directly query your data', and 'Email owners when a refresh fails'. Below these options are 'Edit/Preview data' and 'Augment with SageMaker' buttons, followed by a 'Visualize' button. The background shows the same 'Create a Dataset' page as the previous screenshot.

Now click on Visualize.

56. Now prepare the different charts considering different parameters for filtering, sorting and ordering.



Our Visualization is ready and now we can proceed towards cleanup of all the resources used for this project.

## Chapter 6: Cost Analysis

To provide a monthly billing estimate for the implemented solution of data preparation using an Amazon RDS for MySQL database with AWS Glue DataBrew, we need to consider the costs associated with each component based on the assumed usage by 1000 users for a month.

### **1. Amazon RDS for MySQL:**

Database Instance: The cost of running the RDS instance will depend on factors such as instance type, storage type, storage size, and region. Assuming a moderate instance type and storage size, the cost could range from approximately \$100 to \$500 per month.

Backup Storage: Backup storage costs may add an additional 10-20% to the RDS instance cost.

Data Transfer: Assuming moderate data transfer volumes, the cost could be around \$10 to \$50 per month.

### **2. AWS Glue DataBrew:**

DataBrew Usage: The cost of using AWS Glue DataBrew is based on the amount of data processed and the duration of data preparation jobs. Assuming moderate usage by 1000 users, the monthly cost could range from approximately \$500 to \$1000.

Data Transfer Costs: Data transfer costs between Amazon RDS and AWS Glue DataBrew are minimal if both services are in the same AWS region. Assuming minimal cross-region data transfer, the cost could be negligible.

Optional Services: Additional costs may apply if intermediate data storage is utilized, such as storing data in Amazon S3 as an intermediate layer between RDS and Glue DataBrew. Costs for storing intermediate data could range from approximately \$50 to \$100 per month.

Monitoring and Logging: Costs for storing logs in CloudWatch Logs are typically minimal and may add around \$10 to \$20 per month.

---

**Total Monthly Estimate:**

Adding up the estimated costs for each component:

Amazon RDS for MySQL: \$100 - \$500

AWS Glue DataBrew: \$500 - \$1000

Data Transfer: \$10 - \$50

Optional Services (S3 storage): \$50 - \$100

Monitoring and Logging: \$10 - \$20

Total Monthly Estimate: \$670 - \$1670

Please note that these are rough estimates and actual costs may vary based on factors such as specific usage patterns, instance configurations, data volumes, and AWS pricing updates. It's recommended to use AWS Pricing Calculator or Cost Explorer to get more accurate estimates based on your specific requirements. Additionally, leveraging Reserved Instances or Savings Plans can help reduce costs for AWS services with predictable usage.

## Chapter 7: Lessons and Observations

- 1. Data Pipeline Design:** Designing a data pipeline that involves multiple AWS services (RDS, DataBrew, S3, Athena, QuickSight) requires careful planning to ensure smooth data flow and efficient processing. Understanding the capabilities and limitations of each service is crucial for designing an effective pipeline.
- 2. Resource Provisioning and Configuration:** Provisioning and configuring resources such as RDS instances, VPC, subnets, NAT Gateway, and security groups need to be done thoughtfully to ensure security, scalability, and optimal performance. Proper configuration of access control and network settings is essential for protecting sensitive data and preventing unauthorized access.
- 3. Integration and Connectivity:** Establishing connections between different services (e.g., RDS and DataBrew, DataBrew and S3) requires understanding of connectivity options (e.g., JDBC connection) and configuring appropriate access permissions. Ensuring seamless integration between services is critical for data transfer and processing.
- 4. Data Transformation and Analysis:** Utilizing AWS DataBrew for data transformation tasks such as cleaning, filtering, and aggregating data simplifies the data preparation process. Leveraging tools like Athena for SQL-based analysis enables data analysts to perform complex queries and gain insights from large datasets efficiently.
- 5. Visualization and Reporting:** QuickSight provides a user-friendly interface for creating interactive visualizations and dashboards based on data stored in Athena. Building insightful visualizations allows stakeholders to interpret data trends and make informed decisions.
- 6. Cost Management:** Monitoring resource usage and optimizing costs is essential for managing AWS expenses effectively. Understanding the pricing models of different AWS services helps in estimating and controlling costs. Implementing cost-saving measures such as resource cleanup and rightsizing resources can lead to significant cost savings in the long run.

**7. Resource Cleanup and Deletion:** Proper cleanup of resources after completing a project is crucial for avoiding unnecessary charges and maintaining a clean AWS environment. Deleting unused resources such as RDS instances, S3 buckets, IAM roles, and QuickSight analyses helps in reducing costs and minimizing security risks.

**8. Documentation and Best Practices:** Documenting the entire process, including setup steps, configurations, and lessons learned, is valuable for future reference and knowledge sharing. Following AWS best practices and guidelines ensures that the solution is scalable, secure, and well-architected.

In conclusion, building a data pipeline on AWS involves various stages such as setup, configuration, integration, data processing, analysis, visualization, and cleanup. By following best practices and leveraging AWS services effectively, organizations can streamline data workflows, gain actionable insights, and drive business success.

---

## Chapter 8: Conclusion

In conclusion, the article provides a structured approach to leveraging AWS services for student data analysis, encompassing data ingestion, transformation, analysis, and visualization. By following the outlined steps, users can effectively utilize AWS RDS, Glue Databrew, Athena, and QuickSight to gain insights from complex datasets. The emphasis on pre lab setup, including security considerations and resource cleanup, ensures a streamlined and cost-effective workflow. Overall, this use case demonstrates the power of cloud-based solutions in handling data-intensive tasks and driving informed decision-making in educational contexts.

## Chapter 9: Summary

The provided article outlines a comprehensive use case involving the analysis of student data across schools using AWS services. Three datasets are utilized: school details, student details, and student study details. The workflow begins with setting up an RDS MySQL instance, establishing a JDBC connection, and creating a DataBrew project for data transformation. DataBrew is used to identify top-performing students, with the final output written to an S3 bucket. External tables are then created using Athena, enabling business users to perform BI reporting through QuickSight. The article also covers prerequisites, and pre lab setup steps including database creation, table setup, endpoint creation, IAM role assignment, and security group configuration. It further details the process of creating Glue Databrew connections, datasets, and projects, followed by data manipulation tasks such as joining, filtering, and creating resulting datasets. Additionally, S3 bucket creation, job execution, Athena configuration, and QuickSight setup are explained. The article concludes with a cleanup section, highlighting the importance of deleting all resources used in the project to avoid unnecessary costs.

