

Lab3- Create a database for Alberta vacation company using AWS RDS and MySQL Workbench

Akilandeshwari Srinivasan

451036

CLCM3102

03-10-2023

Description:

Create a AWS Relational database and using the MySQL workbench to create database and tables for the Alberta vacation company.

Preparation:

To do this we need to create a Relational Database service in AWS and connect the MySQL workbench. Then we can write the SQL commands to create and insert data in the table and database which we have learned earlier in the class.

SQL benefits in this Alberta vacation project:

- Basically SQL helps to maintain structured data in a table format where we can easily access to our data based on needs.
- With this SQL, We have created a studentVacation Database and created two tables in it using a line of command inSQL.
- This way SQL makes the work simple and understandable.
- If the website added new features we can add it into a table using “ALTER”, “UPDATE” commands in SQL or can create new table in the db as it shows how scalable it is.

Screenshots:

1.Launch aws management console and search RDS service

The screenshot shows the AWS Management Console Home page. The URL in the address bar is `us-east-1.console.aws.amazon.com/console/home?region=us-east-1#`. The top navigation bar includes links for BVC, CTS, IRCC, Cloud, and others. The Services menu is open, and the RDS icon is highlighted. The main content area displays a "Recently visited" section with links to RDS, DynamoDB, EC2, S3, and Athena. To the right, there is a "Welcome to AWS" sidebar with sections for "Getting started with AWS", "Training and certification", and "What's new with AWS?". Below the sidebar, there are sections for "AWS Health" (with Open issues) and "Cost and usage" (showing Current month costs and Top costs for current month). The bottom of the page includes links for CloudShell, Feedback, and various legal notices.

2.Create database with standard create option and mySQL which we are going to use in our case.

Screenshot of the AWS RDS Create Database page for MySQL.

Create database

Choose a database creation method

- Standard create: You set all of the configuration options, including ones for availability, security, backups, and maintenance.
- Easy create: Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type: MySQL (selected)

- Aurora (MySQL Compatible)
- Aurora (PostgreSQL Compatible)
- MySQL
- MariaDB
- PostgreSQL
- Oracle

MySQL

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.

3. Select the free tier.

Screenshot of the AWS RDS Create Database page for MySQL, showing the selection of the Free tier template.

Hide filters

Show versions that support the Multi-AZ DB cluster [Info](#)
Create a Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

Show versions that support the Amazon RDS Optimized Writes [Info](#)
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine Version
MySQL 8.0.33

Templates
Choose a sample template to meet your use case.

Production: Use defaults for high availability and fast, consistent performance.

Dev/Test: This instance is intended for development use outside of a production environment.

Free tier: Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. [Info](#)

Availability and durability

MySQL

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.

4. Name your database and set the password.

The screenshot shows the AWS RDS MySQL instance creation wizard. In the 'Settings' section, the DB instance identifier is set to 'alberta-vacation-kilaa'. Under 'Master username', 'admin' is selected. A note states: 'If you manage the master user credentials in Secrets Manager, some RDS features aren't supported.' Below this, there's an option to 'Auto generate a password'. On the right, a sidebar titled 'MySQL' lists its features: database size up to 64 TiB, General Purpose, Memory Optimized, and Burstable Performance instance classes, automated backup and point-in-time recovery, and up to 15 Read Replicas per instance.

The screenshot continues the AWS RDS MySQL instance creation wizard. In the 'Instance configuration' section, it shows 'Amazon RDS Optimized Writes' is turned off. Under 'DB instance class', 'db.t3.micro' is selected. The sidebar on the right remains the same, listing MySQL's features.

5.give the public access as "Yes" so that it will be visible and accessible to others.

This screenshot shows the first step of creating a new MySQL database on AWS RDS. It includes sections for Public access, VPC security group (firewall), Availability Zone, and RDS Proxy. A sidebar on the right provides information about MySQL.

Public access: Yes (selected).
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group (firewall): Choose existing (selected).
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.
Existing VPC security groups: default (selected).

Availability Zone: No preference.

RDS Proxy: RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.
 Create an RDS Proxy (Info): RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS Proxy pricing](#).

MySQL: 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.

This screenshot shows the second step of the MySQL creation wizard, focusing on additional configuration and estimated monthly costs.

Additional configuration: Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Estimated monthly costs: The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

[Learn more about AWS Free Tier](#).

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page](#).

MySQL: 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.

Creating database alberta-vacation-kila
Your database might take a few minutes to launch.
You can use settings from alberta-vacation-kila to simplify configuration of [suggested database add-ons](#) while we finish creating your DB for you.

Introducing Aurora I/O-Optimized
Aurora's I/O-Optimized is a new cluster storage configuration that offers predictable pricing for all applications and improved price-performance, with up to 40% costs savings for I/O-intensive applications.

Databases (1)

DB identifier	Status	Role	Engine	Region & AZ	Size	Actions	CPU	Current acti
alberta-vacation-kila	Creating	Instance	MySQL Community	-	db.t3.micro	-	-	-

Successfully created database alberta-vacation-kila
You can use settings from alberta-vacation-kila to simplify configuration of [suggested database add-ons](#) while we finish creating your DB for you.

Introducing Aurora I/O-Optimized
Aurora's I/O-Optimized is a new cluster storage configuration that offers predictable pricing for all applications and improved price-performance, with up to 40% costs savings for I/O-intensive applications.

Databases (1)

DB identifier	Status	Role	Engine	Region & AZ	Size	Actions	CPU	Current a
alberta-vacation-kila	Backing-up	Instance	MySQL Community	us-east-1a	db.t3.micro	-	-	-

6. Once the database is created copy the endpoint for the MySQL workbench username.

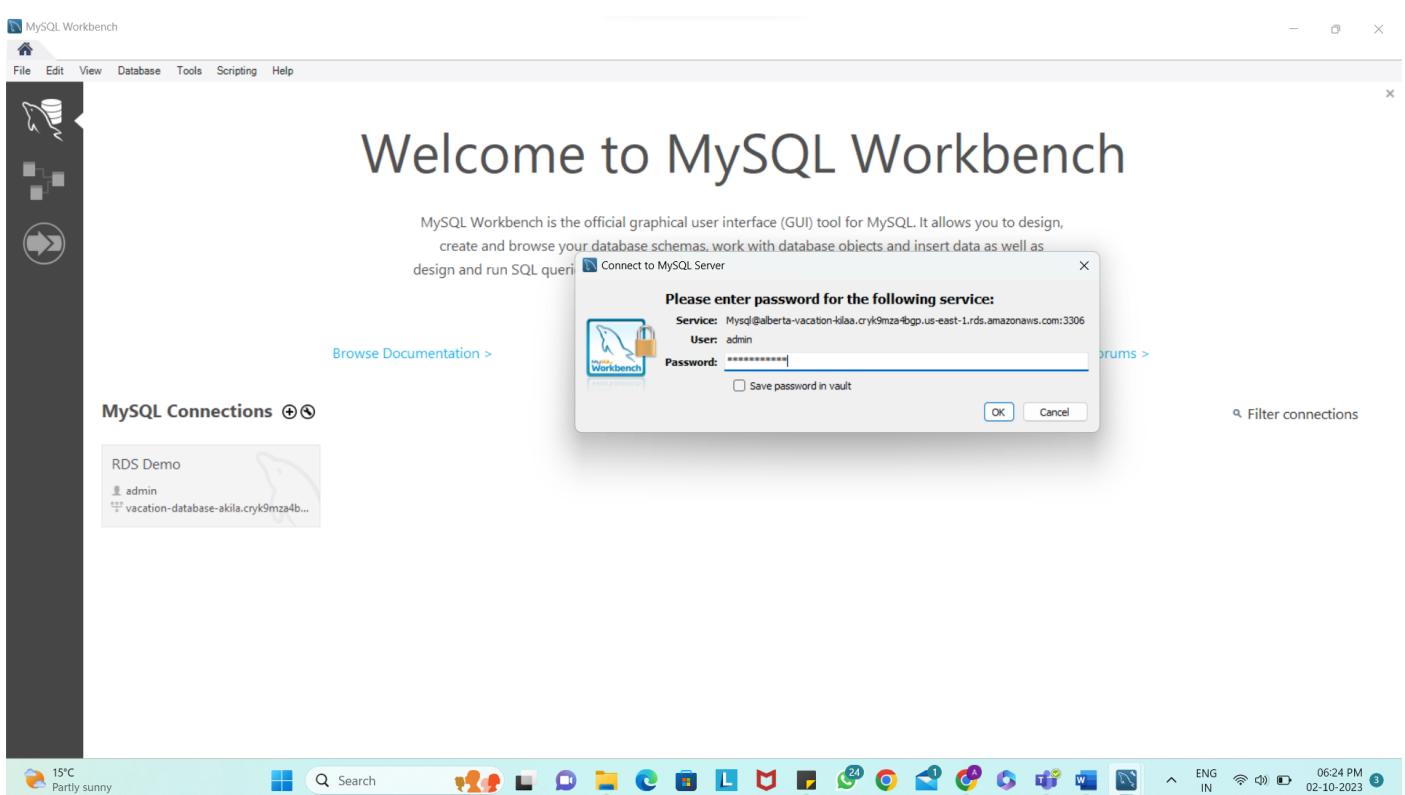
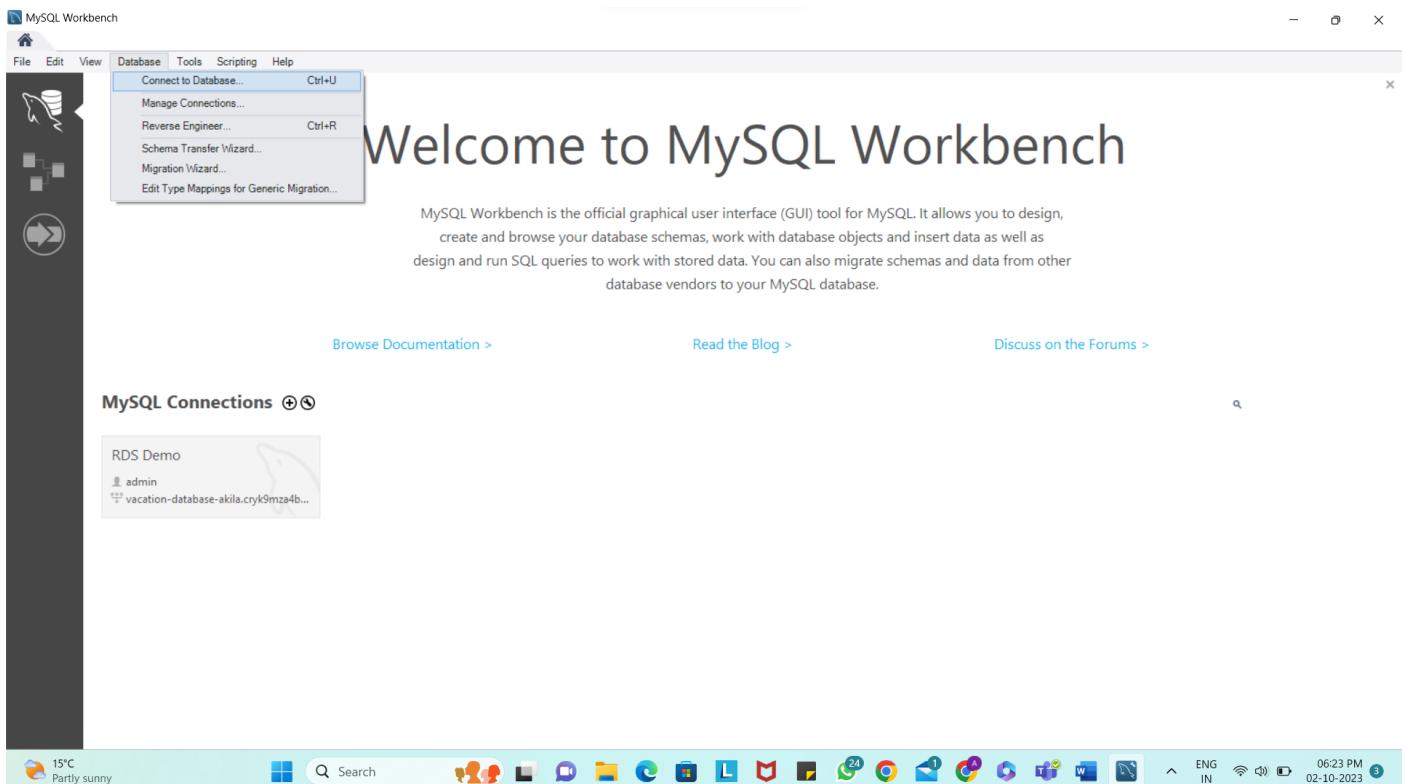
The screenshot shows the AWS RDS console for a MySQL database named 'alberta-vacation-kilaa'. The left sidebar is collapsed, showing options like Dashboard, Databases, Query Editor, and Subnet groups. The main area has a 'Summary' card with details such as DB identifier, CPU usage (10.03%), Status (Backing-up), Role (Instance), Engine (MySQL Community), and Class (db.t3.micro). Below the summary is a navigation bar with tabs: Connectivity & security, Monitoring, Logs & events, Configuration, Maintenance & backups, and Tags. The 'Connectivity & security' tab is selected, displaying information about the endpoint and port, networking (Availability Zone: us-east-1a, VPC: vpc-050524e447eb30cd0), and security (VPC security group: default sg-092bafe1ecf64eff1, Active). The status is publicly accessible.

7. Also add MySql/Aurora in inbound Rules.

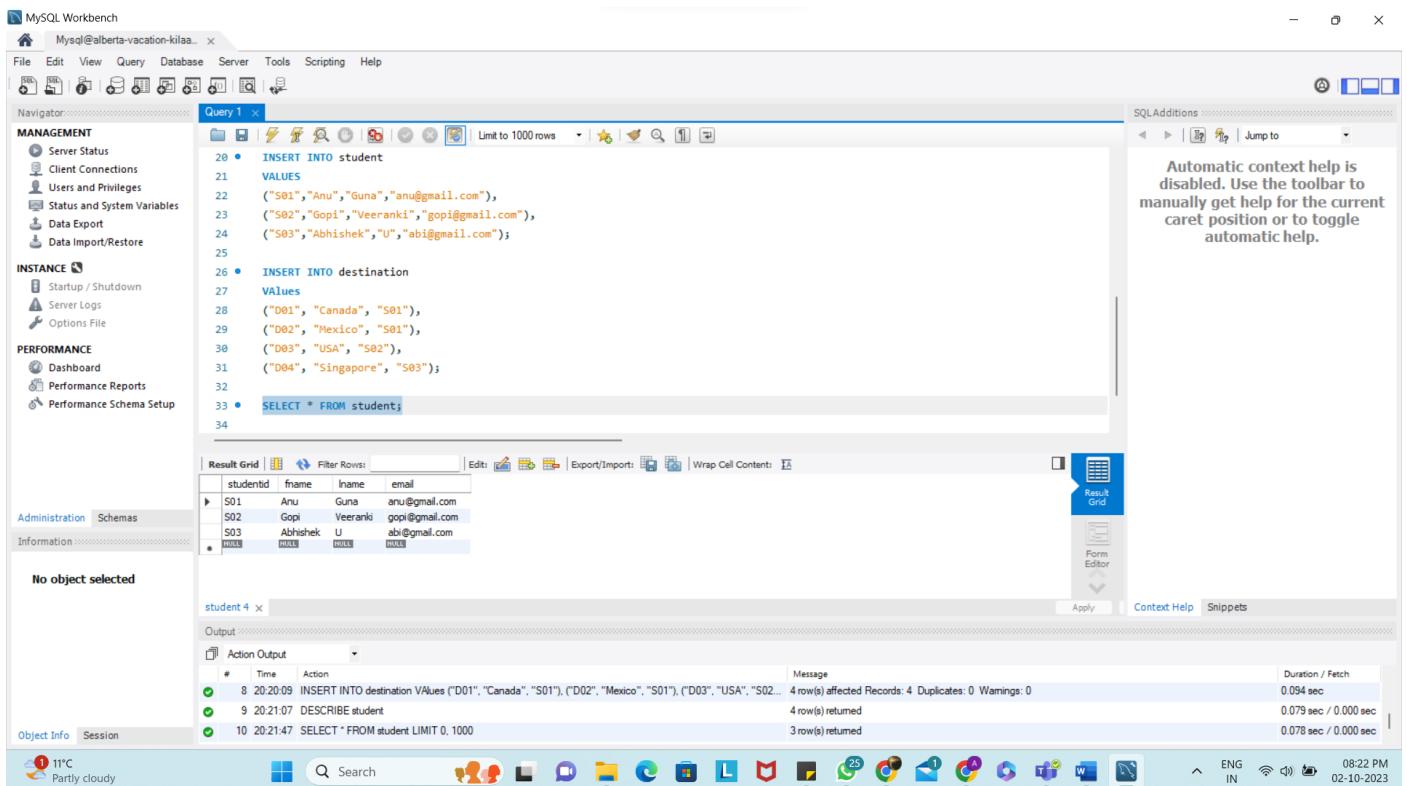
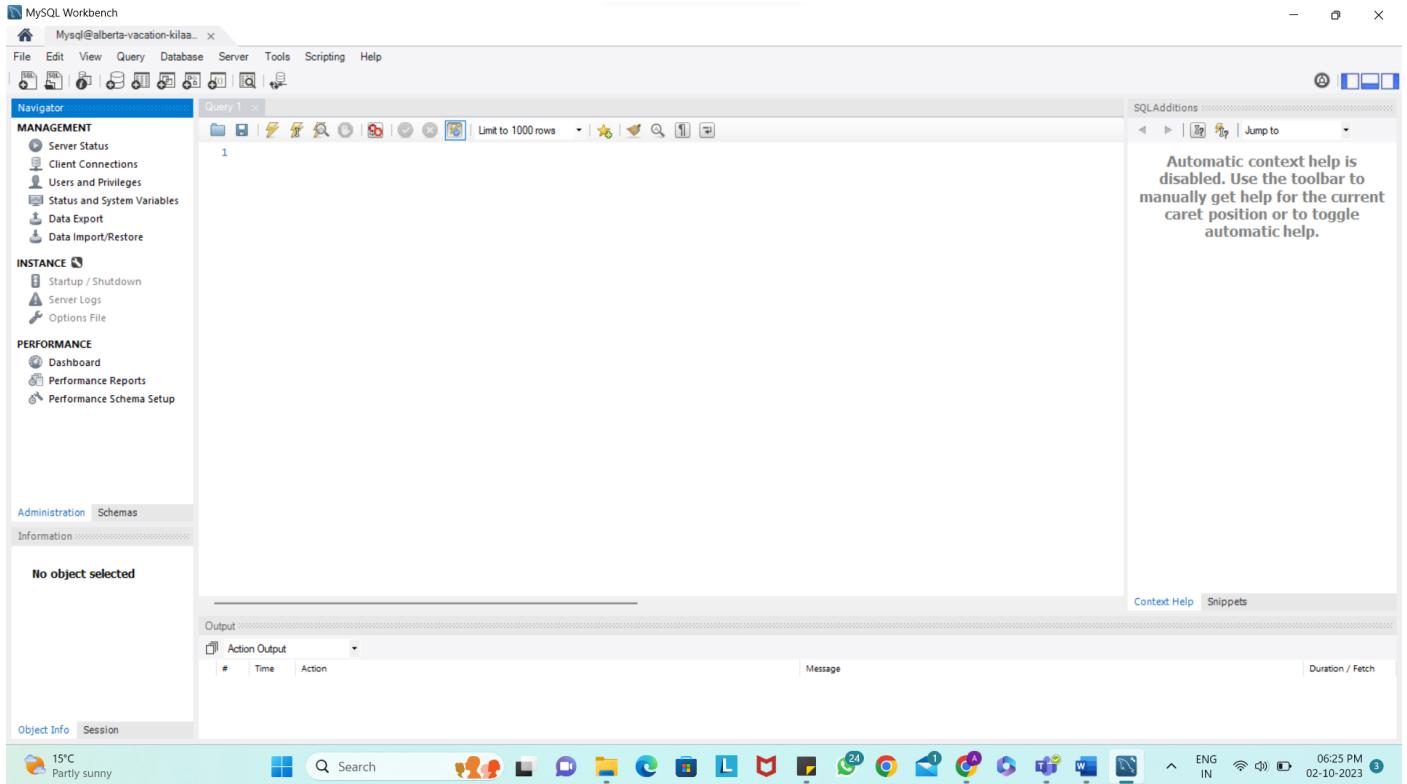
The screenshot shows the AWS EC2 Security Groups console for a security group named 'sg-092bafe1ecf64eff1 - default'. The left sidebar shows 'Security Groups' and the current selection 'sg-092bafe1ecf64eff1 - default'. The main area is titled 'Edit inbound rules' and contains a table of rules. There are two rules listed: one for 'All traffic' with a source of 'sg-092bafe1ecf64eff1' and another for 'MySQL/Aurora' with a source of '0.0.0.0/0'. A button 'Add rule' is at the bottom left. A warning message at the bottom states: '⚠️ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' The bottom navigation bar includes CloudShell, Feedback, and standard browser controls.

The screenshot shows the AWS EC2 Security Groups page. The left sidebar includes links for EC2 Dashboard, EC2 Global View, Events, Instances, Images, and Elastic Block Store. The main content area displays the 'Security Groups (1/1)' table with one row for the 'default' security group. The table columns are Name, Security group ID, Security group name, VPC ID, Description, and Owner. Below the table, tabs for Details, Inbound rules, Outbound rules, and Tags are visible, with 'Inbound rules' selected. The 'Inbound rules (1/2)' table shows two rules: one for All traffic (Protocol All, Port range All) and another for MySQL/Aurora (Protocol TCP, Port range 3306). The bottom of the screen shows the Windows taskbar with various pinned icons and system status.

8. Now open the mysql workbench and connect to database and the password which you have given while creating the RDS.



9. Now write the SQL query ; create a database and table as per required.



MySQL Workbench

Mysql@alberta-vacation-kila...

File Edit View Query Database Server Tools Scripting Help

Navigator:

- MANAGEMENT
 - Server Status
 - Client Connections
 - Users and Privileges
 - Status and System Variables
 - Data Export
 - Data Import/Restore
- INSTANCE
 - Startup / Shutdown
 - Server Logs
 - Options File
- PERFORMANCE
 - Dashboard
 - Performance Reports
 - Performance Schema Setup

No object selected

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

destinationid	destinationcountry	studentid
D01	Canada	S01
D02	Mexico	S01
D03	USA	S02
D04	Singapore	S03
NULL	NULL	NULL

SQLAdditions | Jump to

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid | Form Editor

destination 5 | Context Help Snippets

Action Output | Duration / Fetch

Time	Action	Message	Duration / Fetch
9 20:21:07	DESCRIBE student	4 row(s) returned	0.079 sec / 0.000 sec
10 20:21:47	SELECT * FROM student LIMIT 0, 1000	3 row(s) returned	0.078 sec / 0.000 sec
11 20:22:35	SELECT * FROM destination LIMIT 0, 1000	4 row(s) returned	0.094 sec / 0.000 sec

Object Info Session

11°C Partly cloudy | Search | Application Icons | 08:22 PM 02-10-2023

MySQL Workbench

Mysql@alberta-vacation-kila...

File Edit View Query Database Server Tools Scripting Help

Navigator:

- MANAGEMENT
 - Server Status
 - Client Connections
 - Users and Privileges
 - Status and System Variables
 - Data Export
 - Data Import/Restore
- INSTANCE
 - Startup / Shutdown
 - Server Logs
 - Options File
- PERFORMANCE
 - Dashboard
 - Performance Reports
 - Performance Schema Setup

No object selected

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

studentid	fname	lname	email
S01	Anu	Guna	anu@gmail.com
S02	Gopi	Veeranki	gopi@gmail.com
S03	Abhishek	U	abi@gmail.com
S04	Akila	Vasan	akila@gmail.com
S05	Mamatha	Prave	mamu@gmail.com
NULL	NULL	NULL	NULL

SQLAdditions | Jump to

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid | Form Editor

student 6 | Context Help Snippets

Action Output | Duration / Fetch

Time	Action	Message	Duration / Fetch
11 20:22:35	SELECT * FROM destination LIMIT 0, 1000	4 row(s) returned	0.094 sec / 0.000 sec
12 20:26:06	INSERT INTO student values ("S04","Akila","Vasan","akila@gmail.com"), ("S05","Mamatha","Prave","mamu@gmail.com")	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.094 sec
13 20:26:23	SELECT * FROM student LIMIT 0, 1000	5 row(s) returned	0.078 sec / 0.000 sec

Object Info Session

11°C Partly cloudy | Search | Application Icons | 08:26 PM 02-10-2023

MySQL Workbench

Mysql@alberta-vacation-kila... ×

File Edit View Query Database Server Tools Scripting Help

Navigator: SQLAdditions

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

No object selected

Administration Schemas Information

Object Info Session

Query 1 ×

```
34
35 •   SELECT * FROM destination;
36
37 •   INSERT INTO student_values
38     ("S04", "Akila", "Vasan", "akila@gmail.com"),
39     ("S05", "Manutha", "Prave", "mamu@gmail.com");
40 •   SELECT * FROM student;
41
42 •   INSERT INTO destination
43     Values
44     ("D05", "South Korea", "S04"),
45     ("D06", "Italy", "S05");
46
47 •   SELECT * FROM destination;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Apply | Context Help | Snippets

destinationid	destinationcountry	studentid
D01	Canada	S01
D02	Mexico	S01
D03	USA	S02
D04	Singapore	S03
D05	South Korea	S04
D06	Italy	S05
• NULL	NULL	NULL

destination 8 ×

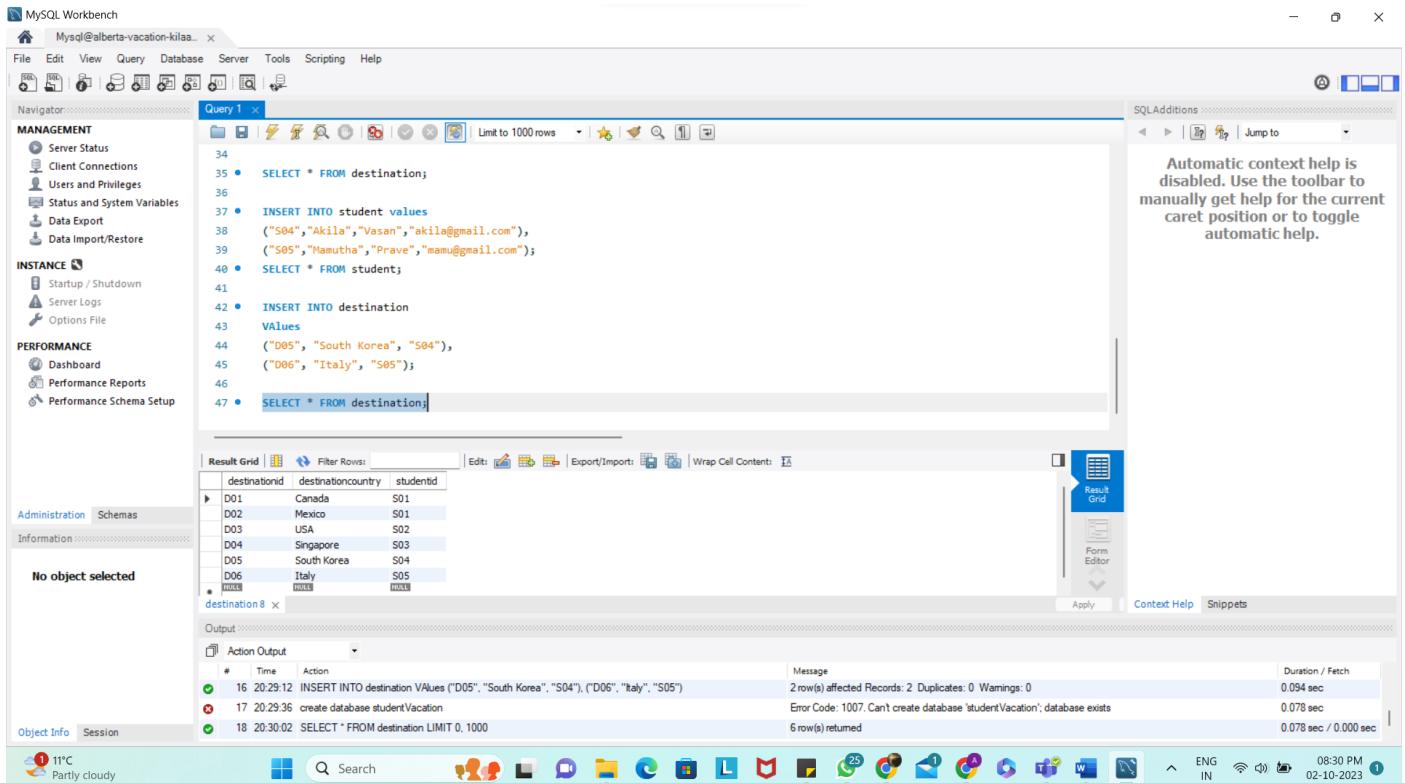
Output:

Action Output	Time	Action	Message	Duration / Fetch
16	20:29:12	INSERT INTO destination Values ("D05", "South Korea", "S04"), ("D06", "Italy", "S05")	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.094 sec
17	20:29:36	create database studentVacation	Error Code: 1007. Can't create database 'studentVacation'; database exists	0.078 sec
18	20:30:02	SELECT * FROM destination LIMIT 0, 1000	6 row(s) returned	0.078 sec / 0.000 sec

11°C Partly cloudy

Search

ENG IN 08:30 PM 02-10-2023



10.If we need to delete the RDS; click the specified one and in actions click “delete”.

The screenshot shows the AWS RDS console with the 'Databases' page. On the left, there's a sidebar with various navigation links. The main area displays a table of databases. One specific database, 'alberta-vacation-kilaa', is selected and has a context menu open over it. The menu contains several options for managing the database, including 'Delete', which is currently selected.

The screenshot shows the AWS RDS console with a modal dialog titled "Delete alberta-vacation-kila instance?". The dialog contains a warning message: "Proceeding with this action will delete the instance with all its content and can affect related resources. [Learn more](#)". Below this, there are three checkboxes:

- Create final snapshot: Describes creating a final DB Snapshot before deletion.
- Retain automated backups: Describes retaining automated backups for 1 day after deletion.
- I acknowledge that upon instance deletion, automated backups, including system snapshots and point-in-time recovery, will no longer be available.

To avoid accidental deletion, provide additional written consent. To confirm deletion, type `delete me` into the field, which contains the text "delete me". A final warning message in a box says: "We strongly recommend taking a final snapshot before instance deletion since after your instance is deleted, automated backups will no longer be available." At the bottom are "Cancel" and "Delete" buttons.

The screenshot shows the AWS RDS console with a success message: "Successfully deleted DB instance alberta-vacation-kilaa". A blue banner at the top right says "Introducing Aurora I/O-Optimized". Below it, a callout box suggests creating a Blue/Green Deployment. The main table shows zero instances found.

Amazon RDS

- 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 [New](#)
- Events
- Event subscriptions

RDS > Databases

Databases (0)

Group resources Modify Actions

Filter by databases

DB identifier	Status	Role	Engine	Region & AZ	Size	Actions	CPU	Current activity	Maintenance
No instances found									

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 11°C Partly cloudy ENG IN 08:50 PM 02-10-2023

Observation:

AWS RDS makes work easy and helps to maintain the code. Moreover it is scalable. MySQL workbench is understandable and easy to use while working with it.

Reference:

www.W3Schools.com , www.google.com ,

aws official documentation(<https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Welcome.html>)