CLOUD COMPUTING PROJECT

CUSTOMER DATA DATABASE MIGRATION FROM ON-PREMISE TO CLOUD

OBJECTIVES:

After performing the project, the user should be able to:

1. Design a platform for recording, organizing, and analytics on customers transactions data received through application(s).
2. The data stored in the DB must be organized and maintained in such a way that it caters to the needs of the future. Derive insights from the results to device better strategies and enhance customers experience.

THEORY:

You can use AWS Database Migration Service (AWS DMS) to migrate your data to and from most widely used commercial and open-source databases such as Oracle, PostgreSQL, Microsoft SQL Server, Amazon Redshift, Amazon Aurora, MariaDB, and MySQL. The service supports homogeneous migrations such as Oracle to Oracle, and also heterogeneous migrations between different database platforms, such as Oracle to MySQL or MySQL to Amazon Aurora with MySQL compatibility. The source or target database must be on an AWS service.

This pattern provides guidance for migrating an on-premises MySQL database to a MySQL database on an Amazon RDS. The pattern discusses the use of AWS Database Migration Service tools such as mysqldbcopy, mysqldump and mysql workbench migration methodology.

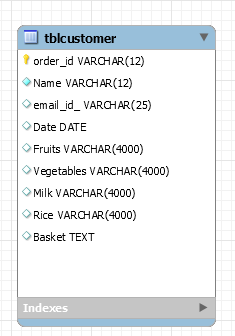
SOFTWARE NEEDED

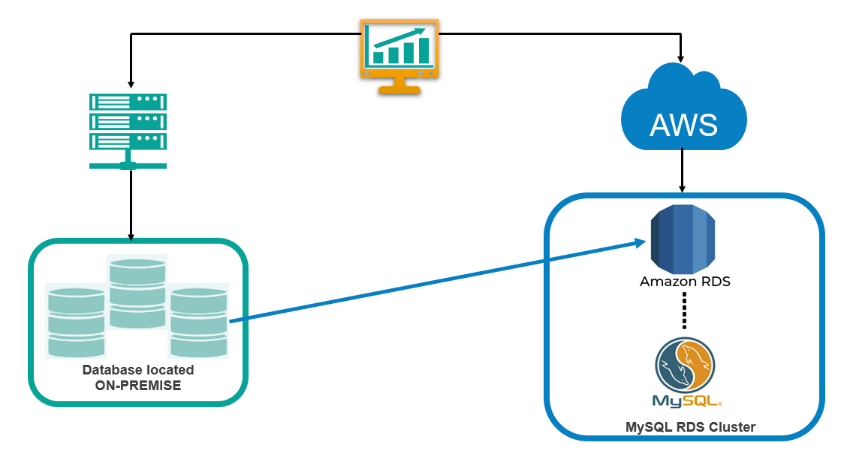
MySQL Workbench

AWS RDS MySQL Environment

Data to be used

MIGRATION DESIGN AND SCHEMA DESIGN





# **1. Steps in Importing file from .csv file to MySQL On-Premise**

1. Start Excel, select the **Data** menu tab, and then click **MySQL for Excel** to open the MySQL for Excel task pane.
2. From the **Open a MySQL Connection** area in the task pane, double-click an existing local or remote connection to display the available database schemas.
3. Select a schema from the list and click **Next** to display all database objects in the schema (tables, views, and procedures).
4. Select the table, view, or procedure with data to import and then click **Import MySQL Data**. A preview window displays the selected data and provides **Options** and Advanced Options to be used during the import operation.
5. Click **Import** to finish the operation.

The Import Data windows provides a preview of the columns to select during the import operation. You can specify both the columns and rows to import. As the following figure shows, the preview includes a small subset of the rows for the selected table or view.

Figure 1.1 Importing table data with MySQL for Excel

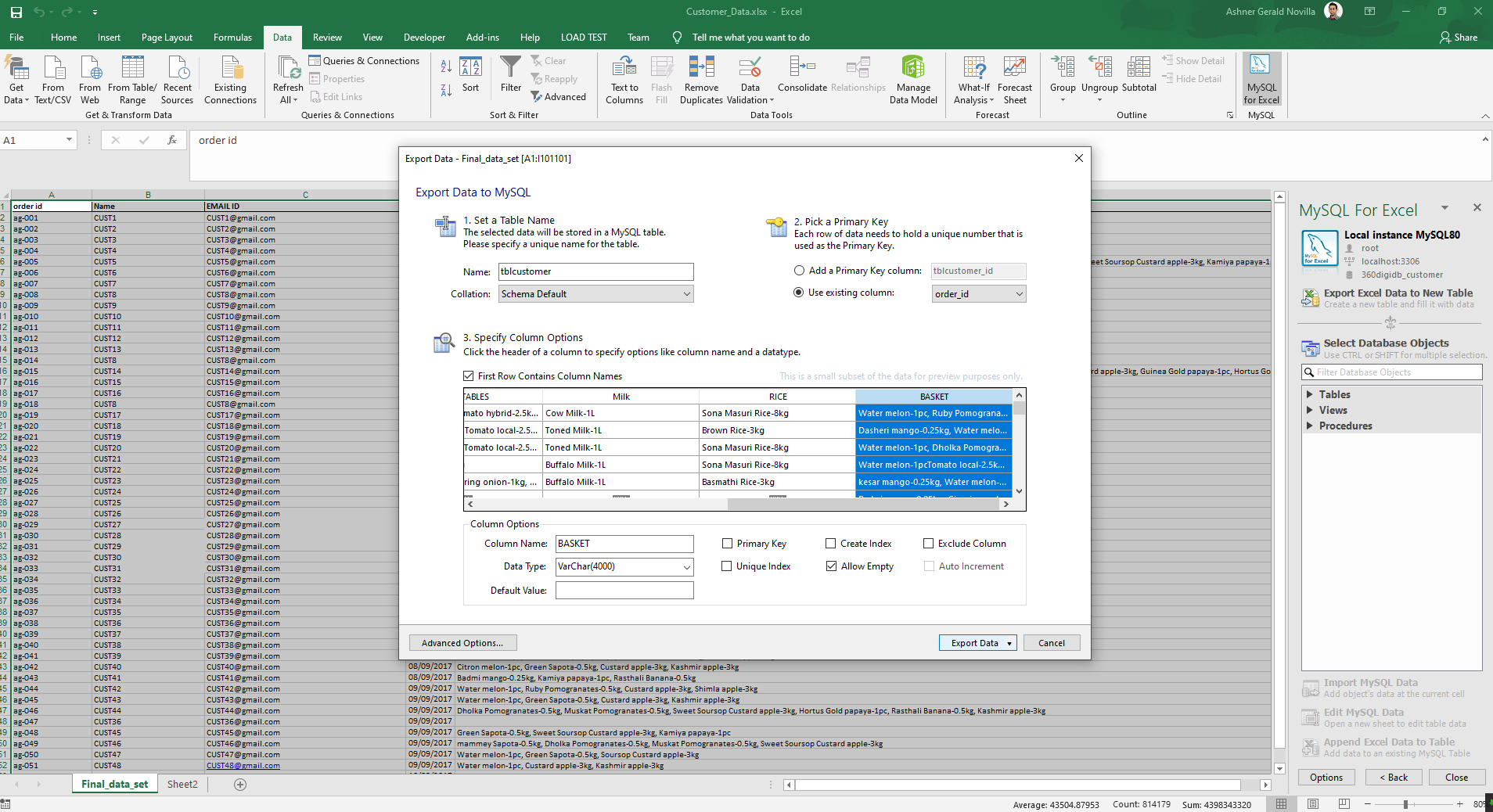
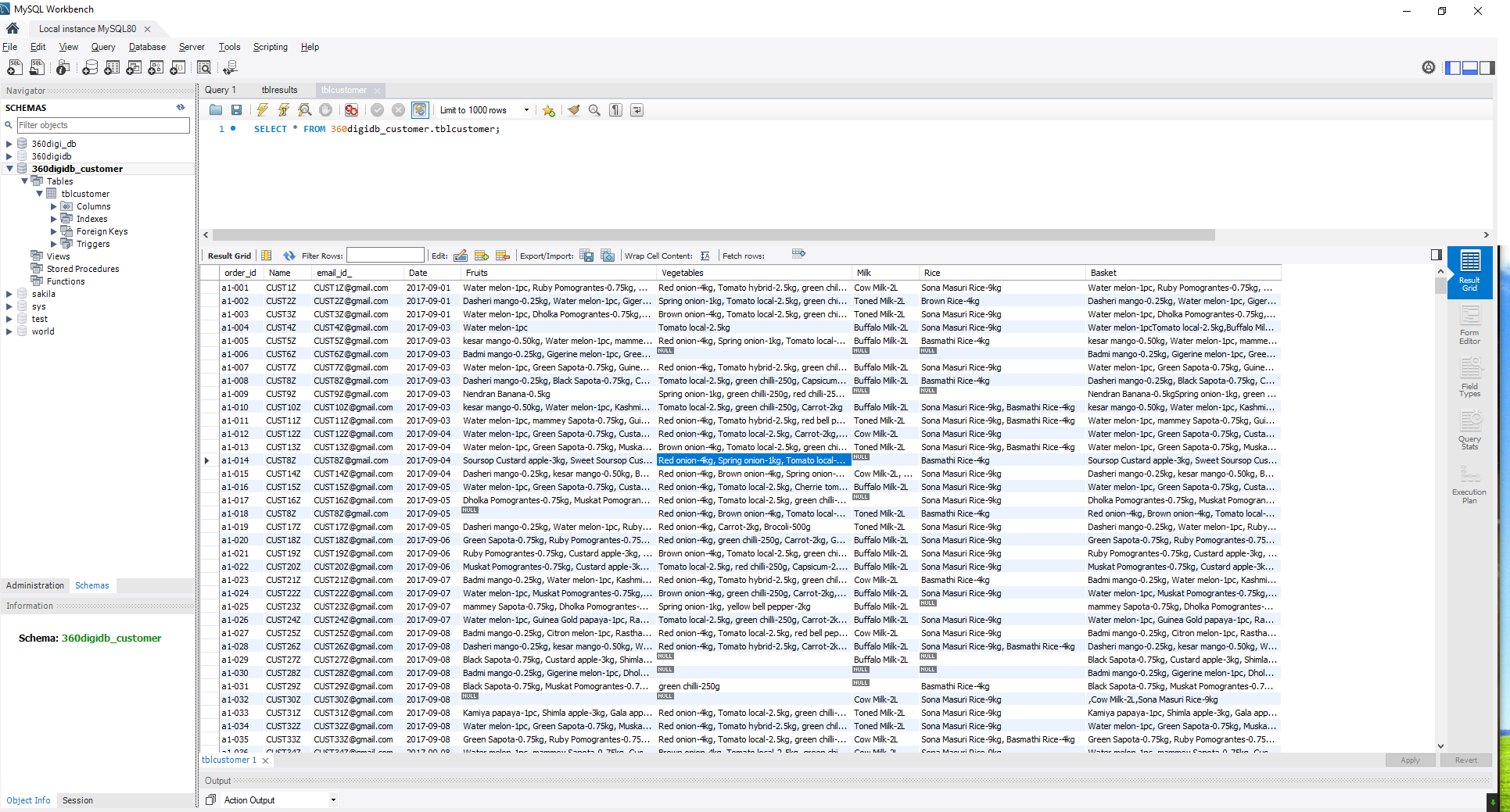


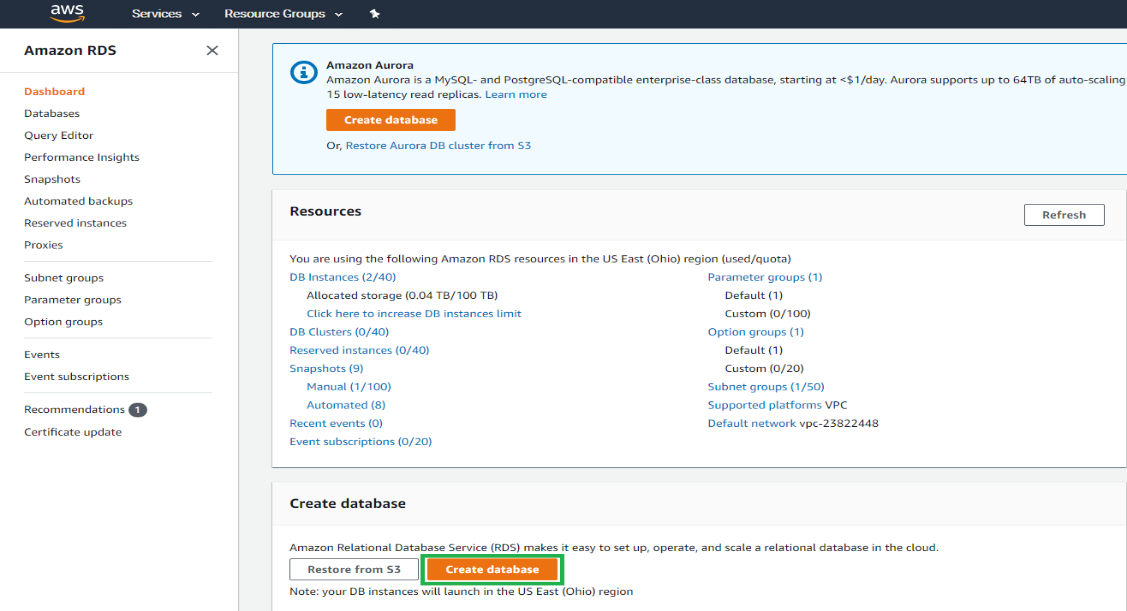
Figure 1.2 Importing table data with MySQL for Excel Successful



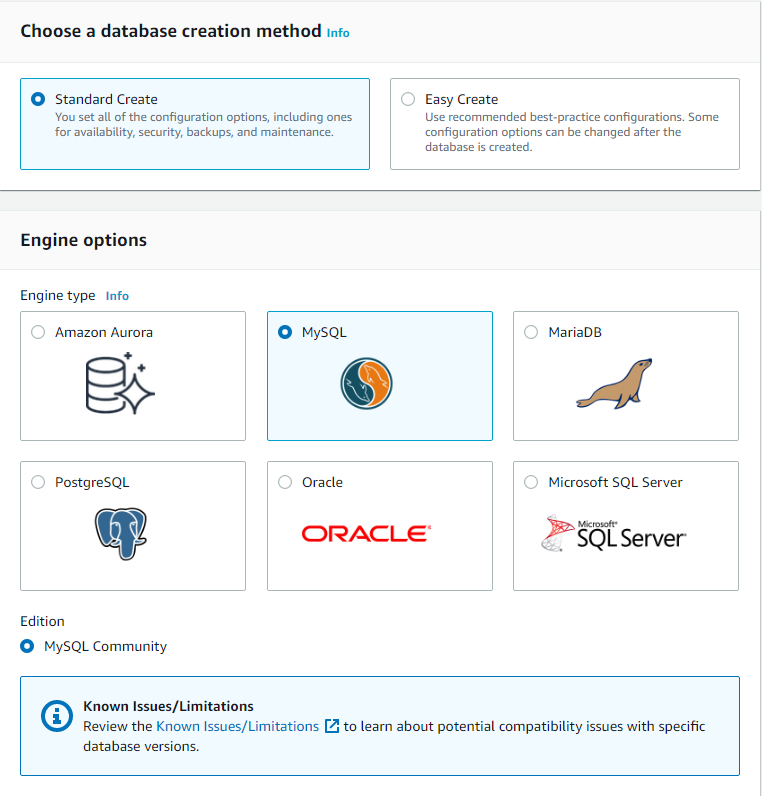
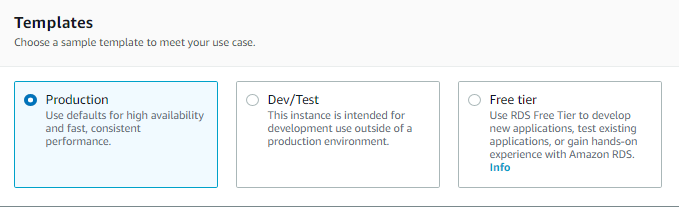
# **2. Steps in Deploying RDS MySQL in Amazon Web Services**

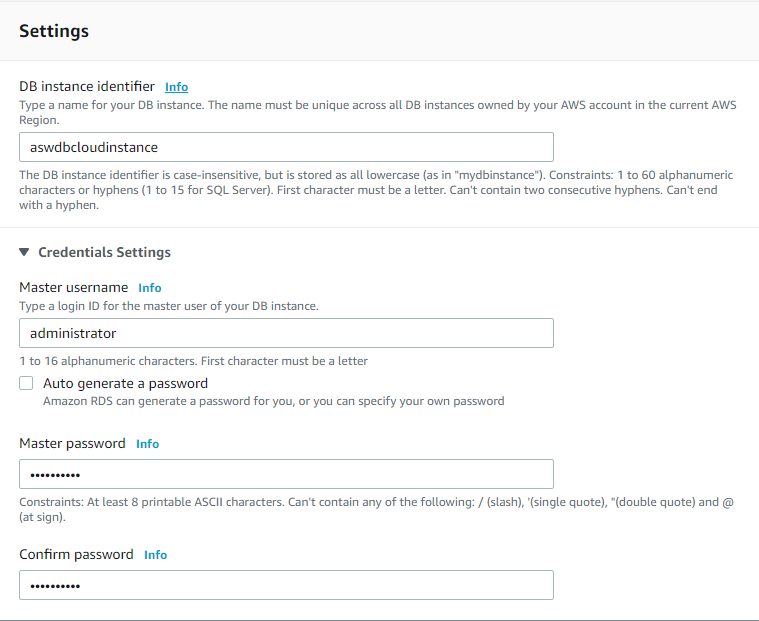
1. Sign in to the AWS Management Console and open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the upper-right corner of the Amazon RDS console, choose the AWS Region in which you want to create the DB instance.
3. In the navigation pane, choose **Databases**.
4. Choose **Create database** and make sure that **Standard Create** is chosen.

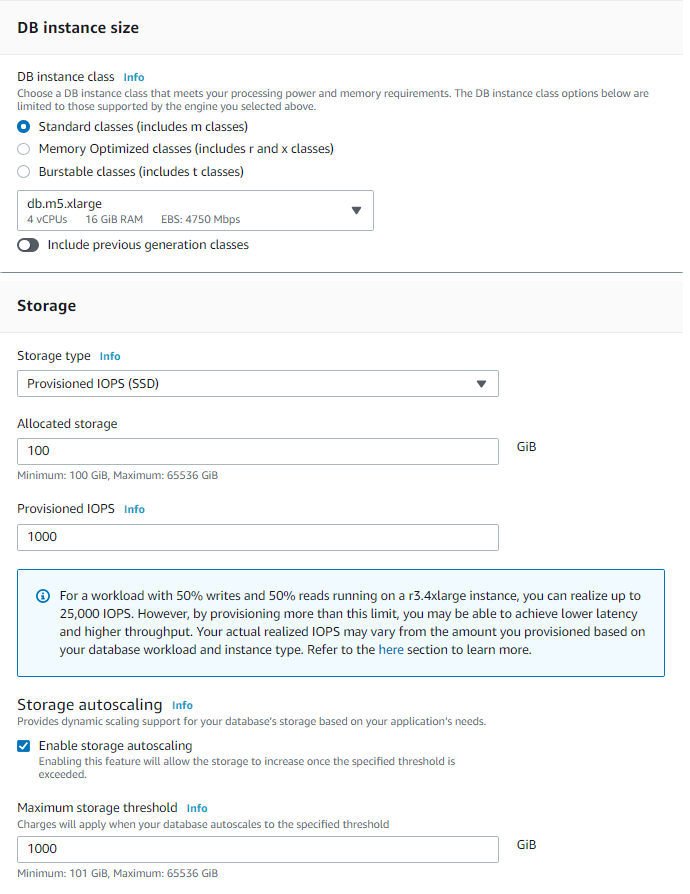
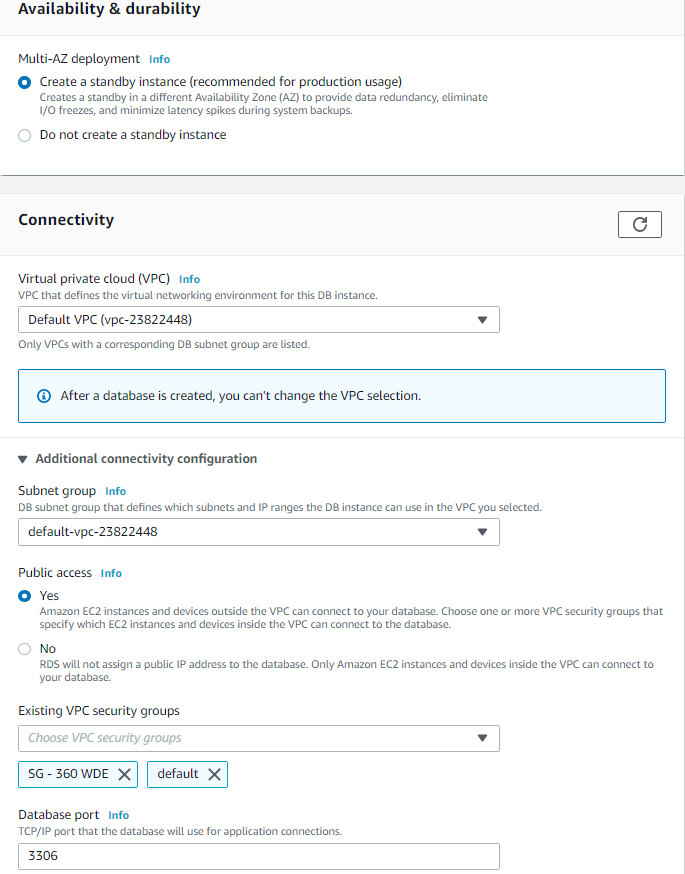
Figure 2.1 Initiating the deployment of Database in RDS

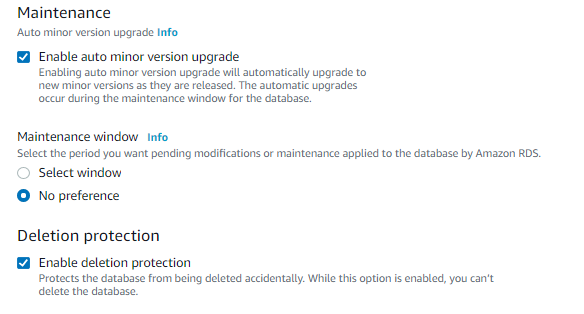
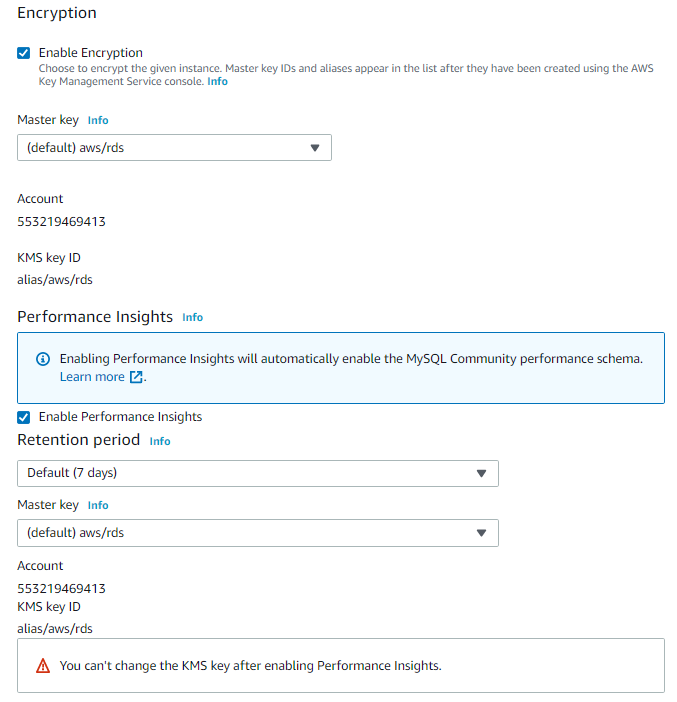
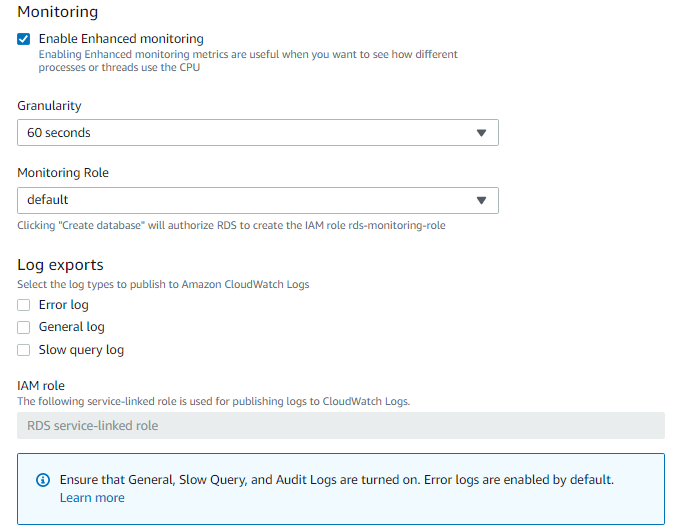


1. In **Configuration**, choose **MySQL**.
2. For **DB instance size**, choose **Free tier**.
3. For **DB instance identifier**, enter a name for the DB instance, or leave the default name.
4. For **Master username**, enter a name for the master user, or leave the default name. The **Create database** page should look similar to the following image.
5. To enter your master password, disable **Auto generate a password**, and then enter the same password in **Master password** and **Confirm password**.
6. Choose **Standard Create** to change one or more settings during database creation and set them. To change a setting with No in that column, use **Standard Create**. For settings with **Yes** in that column, you can either use **Standard Create** or modify the DB instance after it is created to change the setting.

Figure 2.2 Selecting the Database creation method and the engine option to be used. (In this project the developer will focus on using MySQL, User: administrator, Password: Amazon12#$)



Figure 2.3 Selecting the DB Instance size as the **Standard Class** and having an initial Storage of 100GB **SSD** with an autoscaling storage up to **1000 GB.** **Multi-AZ** Deployment for **Active and Standby Mode.**

 Figure 2.4 Database authentication using **Password authentication**. The additional configuration set on the image below is configured to cater the ease of **maintenance**, **back-up**, **security**, **monitoring**, **scaling** and **update** on the client side.

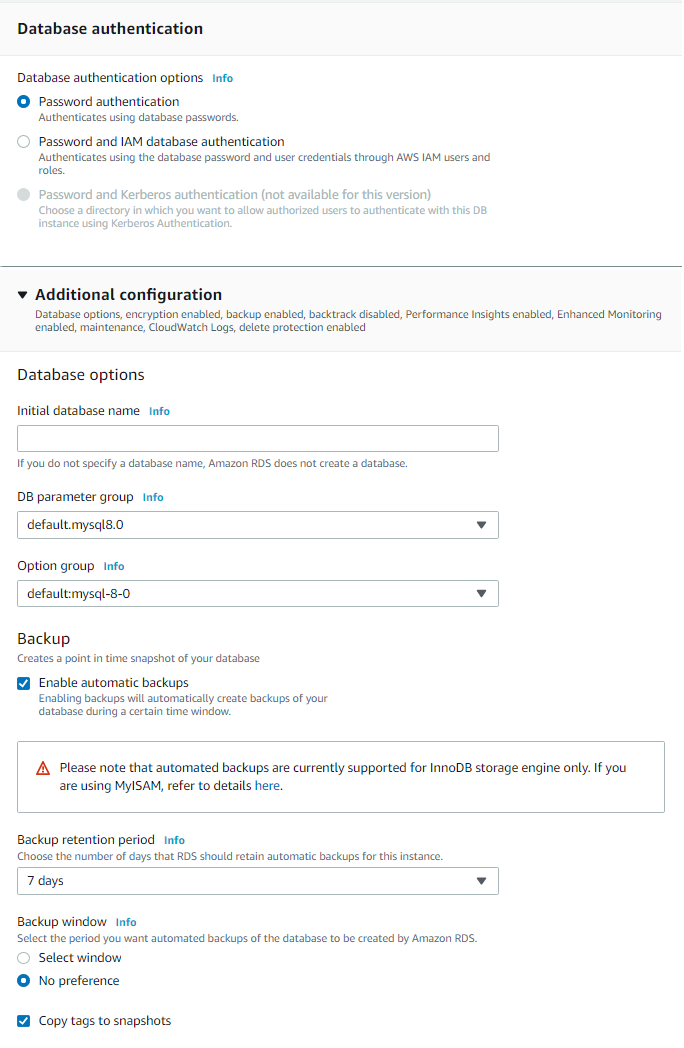
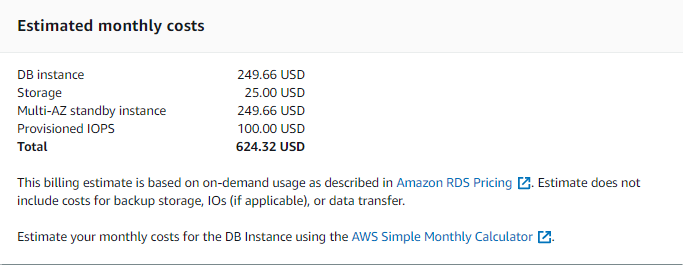


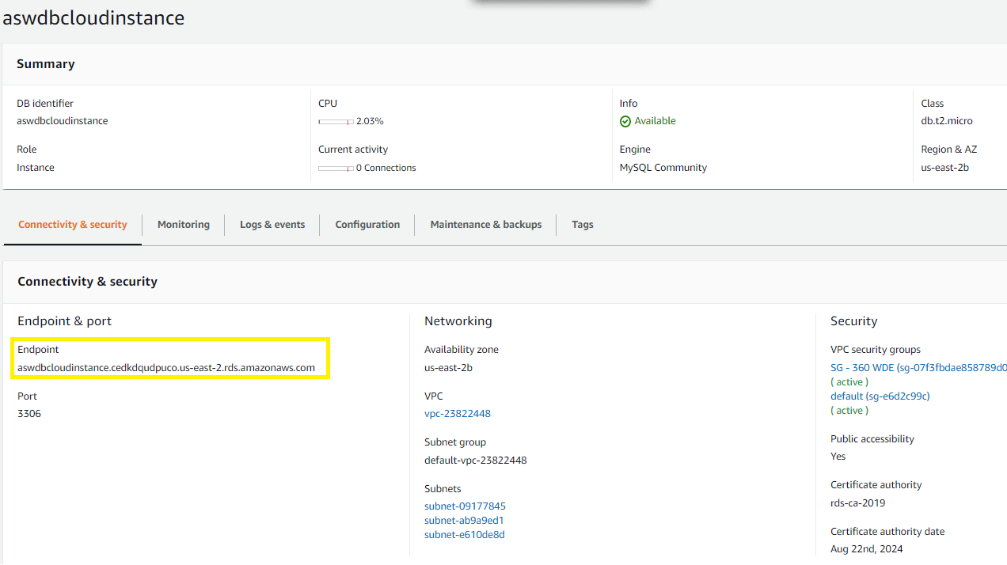
Figure 2.5 Database estimated monthly costs that the client will need to keep in mind. (link: <https://calculator.s3.amazonaws.com/index.html> for future calculation or resizing of the environment)



# **3. Connecting to a Database on a DB Instance Running the MySQL Database Engine**

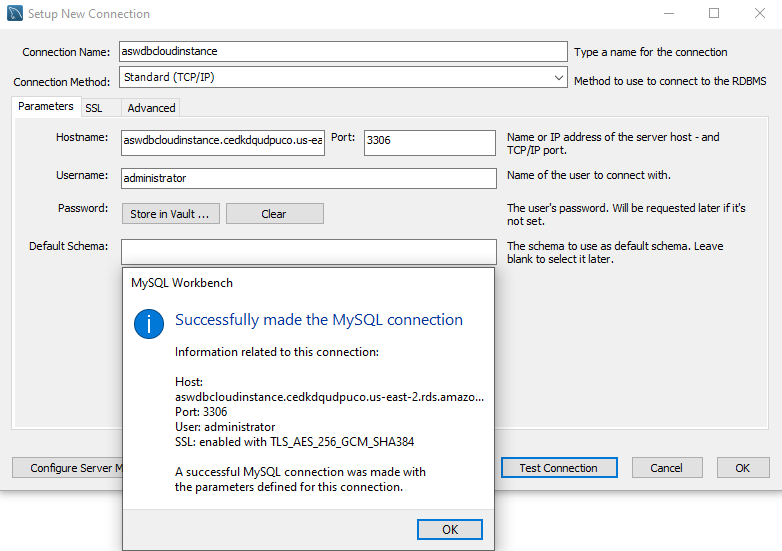
1. After Amazon RDS provisions your DB instance, you can use any standard SQL client application to connect to a database on the DB instance. In this example, you connect to a database on a MySQL DB instance using MySQL monitor commands.
2. Find the endpoint (DNS name) and port number for your DB instance.
   1. Open the RDS console and then choose Databases to display a list of your DB instances.
   2. Choose the MySQL DB instance name to display its details.
   3. On the Connectivity & security tab, copy the endpoint. Also, note the port number. You need both the endpoint and the port number to connect to the DB instance.

Figure 3.1 Database Endpoint (*this endpoint is been done using free tier mode for the developer to avoid unneeded cost of deployment*)



1. Download a SQL client that you can use to connect to the DB instance. You can connect to an Amazon RDS MySQL DB instance by using tools like the MySQL Workbench.
2. Connect to a database on a MySQL DB instance. For example, enter the following command at a command prompt on a client computer to connect to a database on a MySQL DB instance using the MySQL client. Substitute the DNS name for your DB instance for <*endpoint*>, the master user name you used for <*mymasteruser*>, and provide the master password you used when prompted for a password.

Figure 3.2 Local MySQL Workbench successfully made a connect to MySQL of RDS in AWS



# **4. Migrating an On-Premises Database to Amazon RDS MySQL**

The MySQL Workbench Migration Wizard allows you to easily and quickly migrate

databases from Microsoft SQL Server, PostgreSQL, Sybase ASE and SQL anywhere,

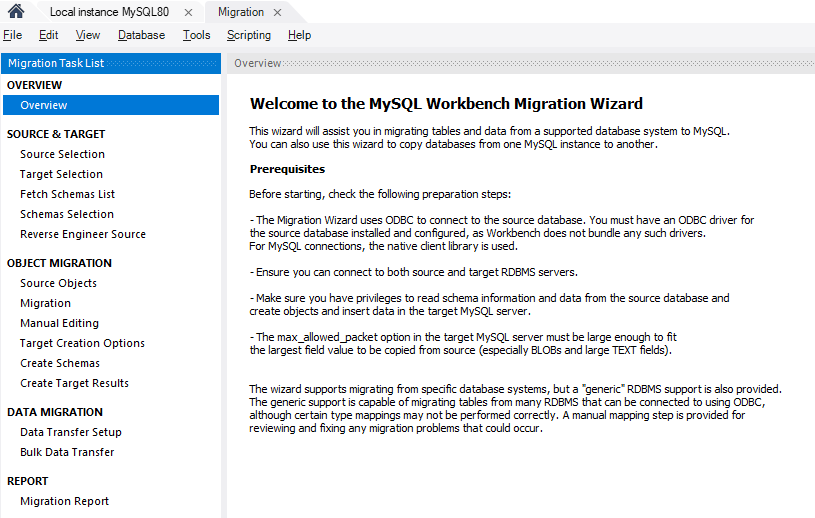
SQLite, and most ODBC-capable RDBMSs to MySQL. In addition, you can use the

module to create MySQL to MySQL database copies that can be used for tasks such as

copying a database across servers or migrating data across different versions of MySQL.

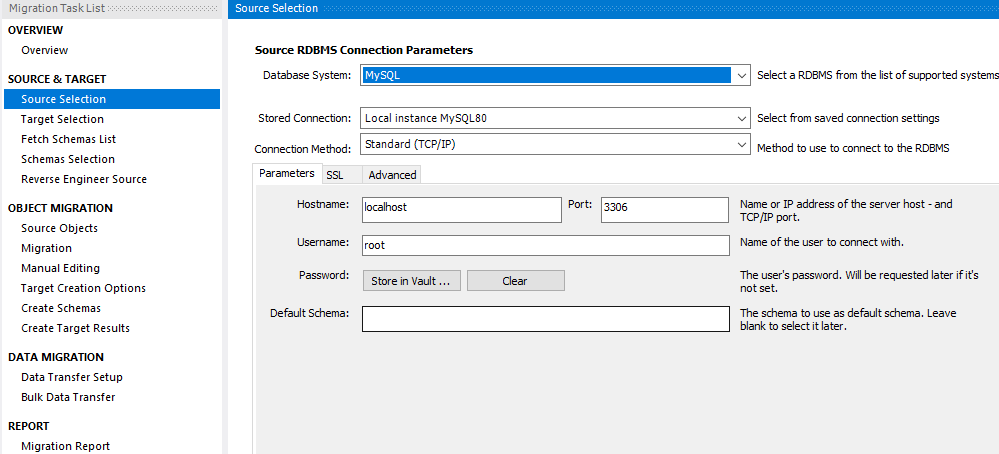
1. Check if the user has an appropriate access to a running SQL Server instance of the database you want to migrate.
2. Check if the user has an appropriate access to a running MySQL Server instance.
3. Open MySQL Workbench and start the Migration Wizard. From the main MySQL Workbench screen, you can start the Migration Wizard by clicking on the Database Migration launcher in the Workbench Central panel or through **Database –> Migrate** in the main menu.

Figure 4.1 Migration Wizard



1. Set up the parameters to connect to your source database by **clicking start.** In this page you need to provide the information about the RDBMS you are migrating.
2. If you open the **Database System** combo box, you’ll find a list of the supported RDBMSs. Select MySQL Server from the list. Just below it there’s another combo box named **Stored Connection**. It will list saved connection settings for that RDBMS. You can save connections by marking the checkbox at the bottom of the page and giving a name.

Figure 4.2 Source Connection Settings



1. Set up the parameters to connect to your target Database. Click on the **Next** button to move to the **Target Selection** page. Once there, set the parameters to connect to your MySQL Server instance. When you are done click on the **Test Connection button** and verify that you can successfully connect to it.

Figure 4.3 Target Connection Settings

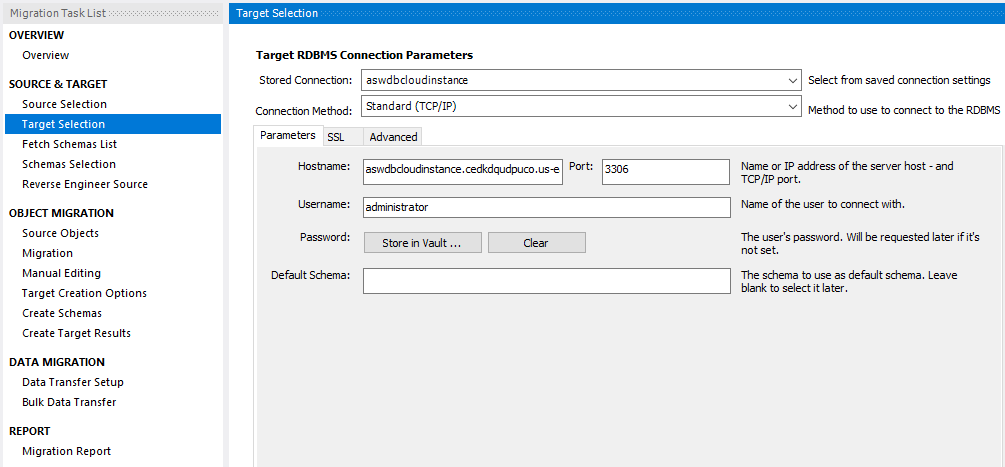
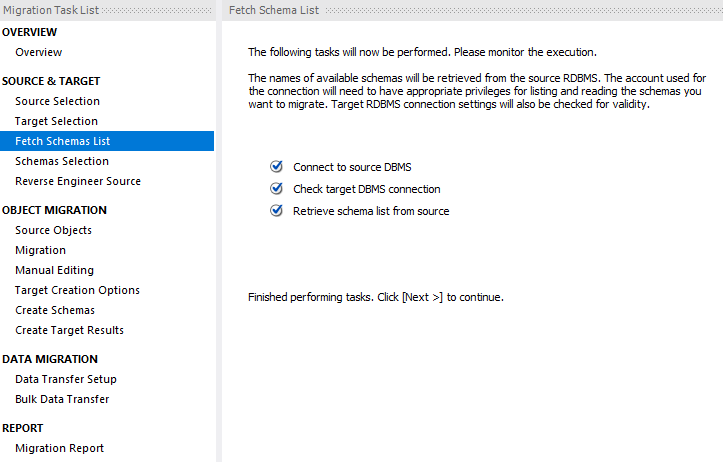


Figure 4.4 Fetch Schemas List



1. Select the Schema to Migrate Click on the Next button to move to the next page. The Migration Wizard will connect to SQL Server to fetch a list of the catalogs and schemas.

Figure 4.4 Schemas Selection (*In this project the user will be importing the data from 360digidb\_customer schema that contains the data given by the PM*)

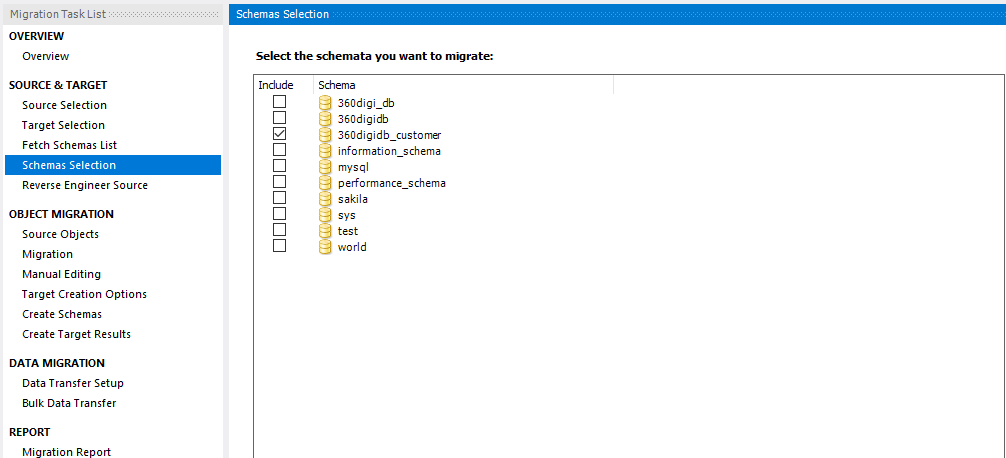
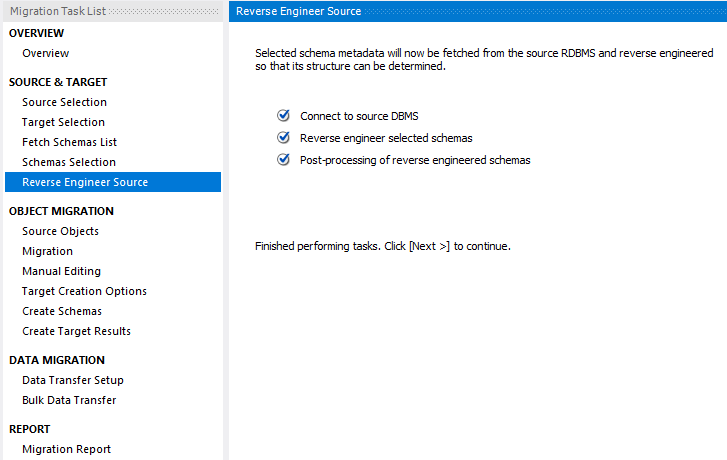
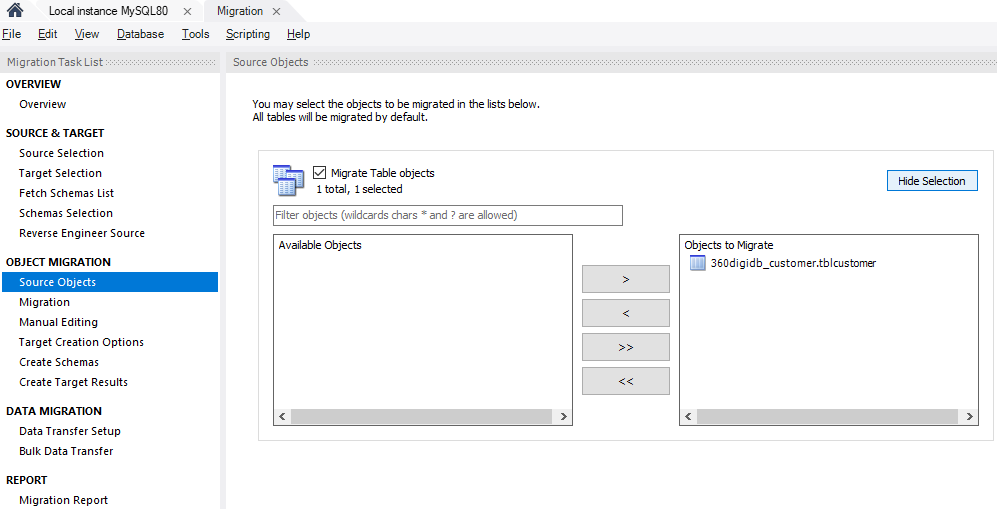


Figure 4.5 Reverse Engineer Source



1. Select the Objects to Migrate Move to the next page using the Next button. You should see the reverse engineering of the selected schema in progress. At this point the Migration Wizard is retrieving relevant information about the involved database objects (table names, table columns, primary and foreign keys, indices, triggers, views, etc.).
2. If you click on the **Show Selection** button you will be given the opportunity to select exactly what you want to migrate as shown here:



1. Review the Proposed Migration. Move to the next page. You will see the progress of the migration there. At this point the Migration Wizard is converting the objects you selected into their equivalent objects in MySQL and creating the MySQL code needed to create them in the target server.

Figure 4.5 Migration

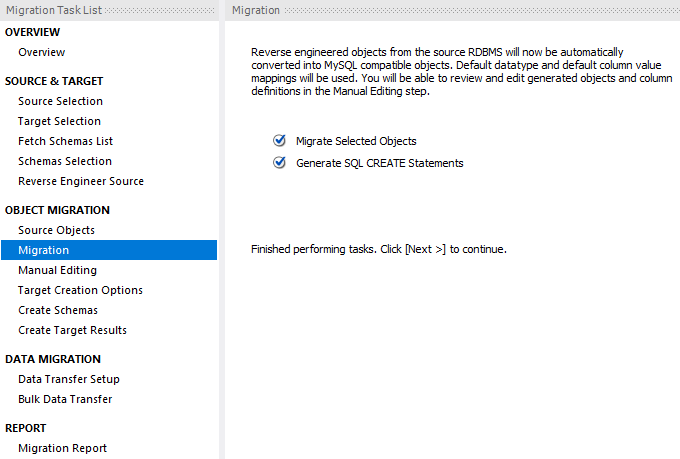


Figure 4.6 Manual Editing

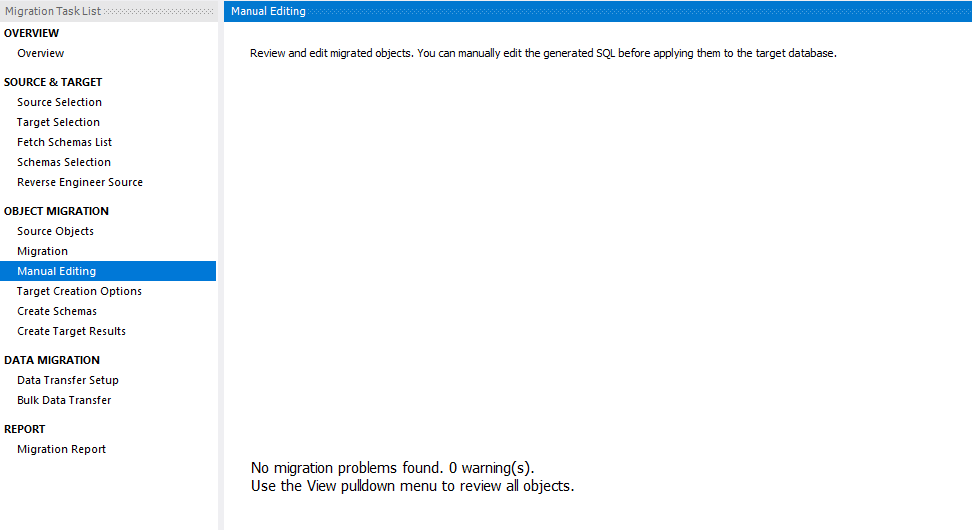


Figure 4.7 Target Creation Options

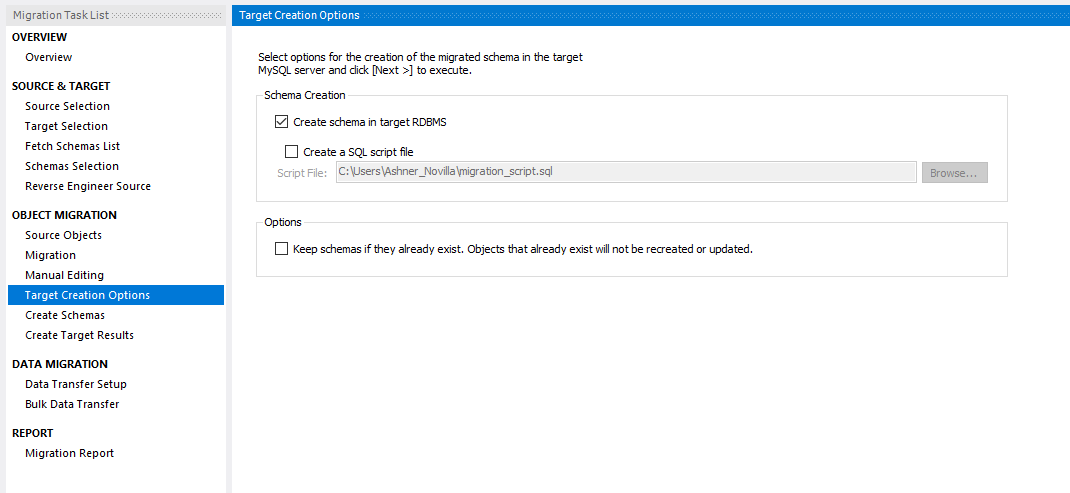


Figure 4.8 Create Schemas

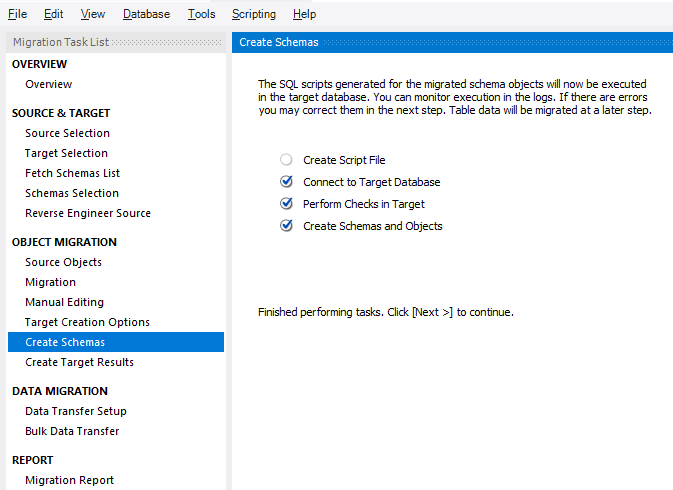
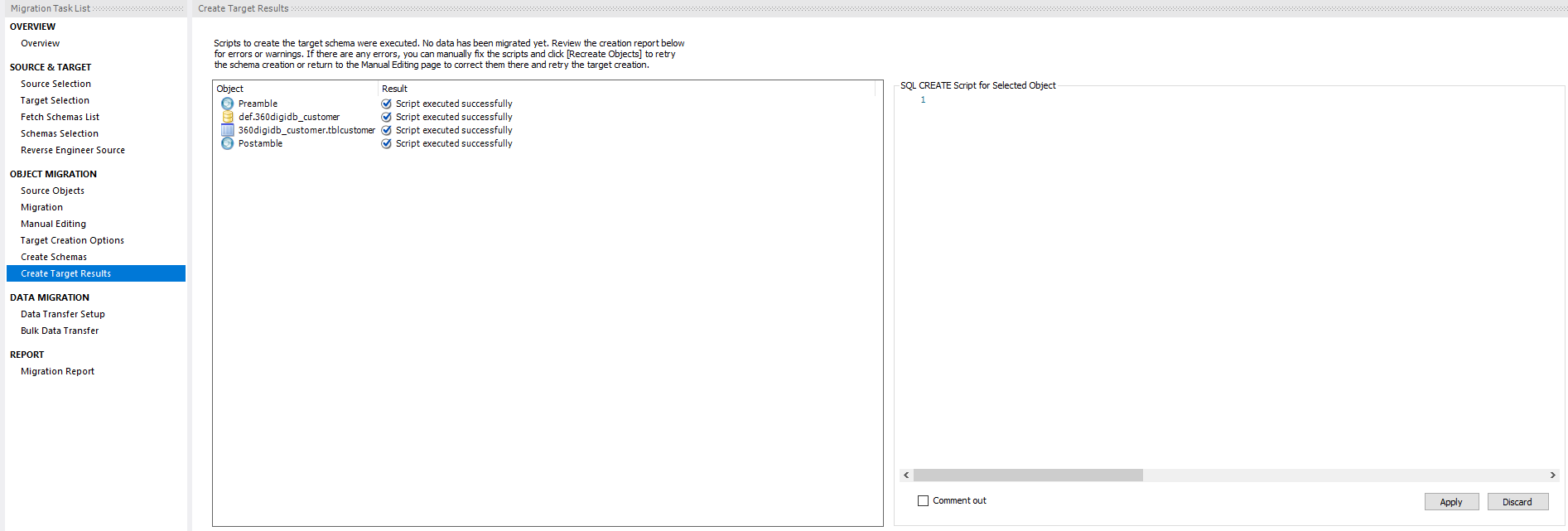


Figure 4.96 Create Target Results



1. This step in the Migration Wizard involve the transference of data from the source SQL Server database into your newly created MySQL database. The Data Transfer Setup page allows you to configure this process.

Figure 4.97 Data Transfer Setup

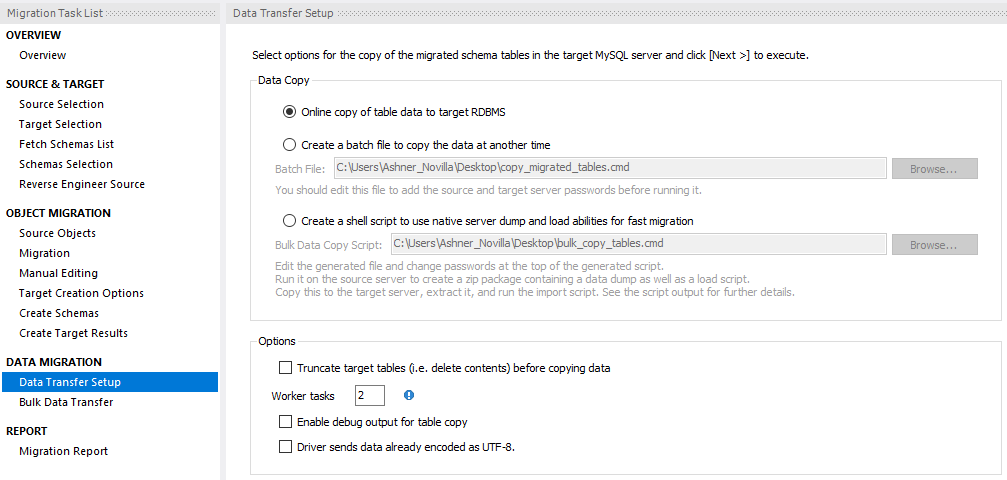
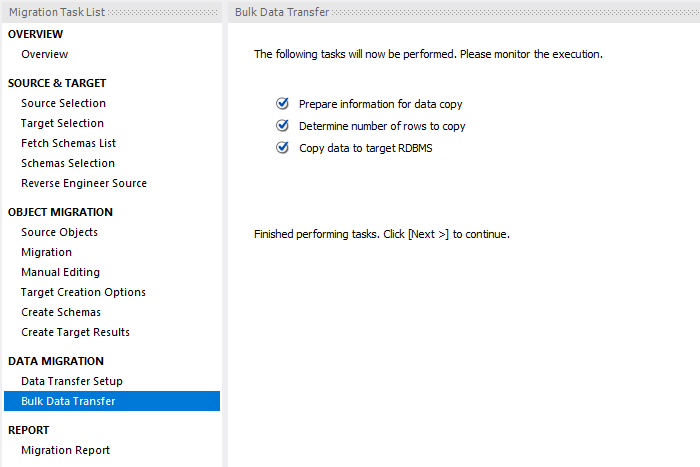


Figure 4.98 Bulk Data Transfer



1. Once it finishes, move to the next page. A report page will be displayed with a summary of the entire process:

Figure 4.98 Migration Report and Verification of Migration Result on the MySQL RDS in AWS

