

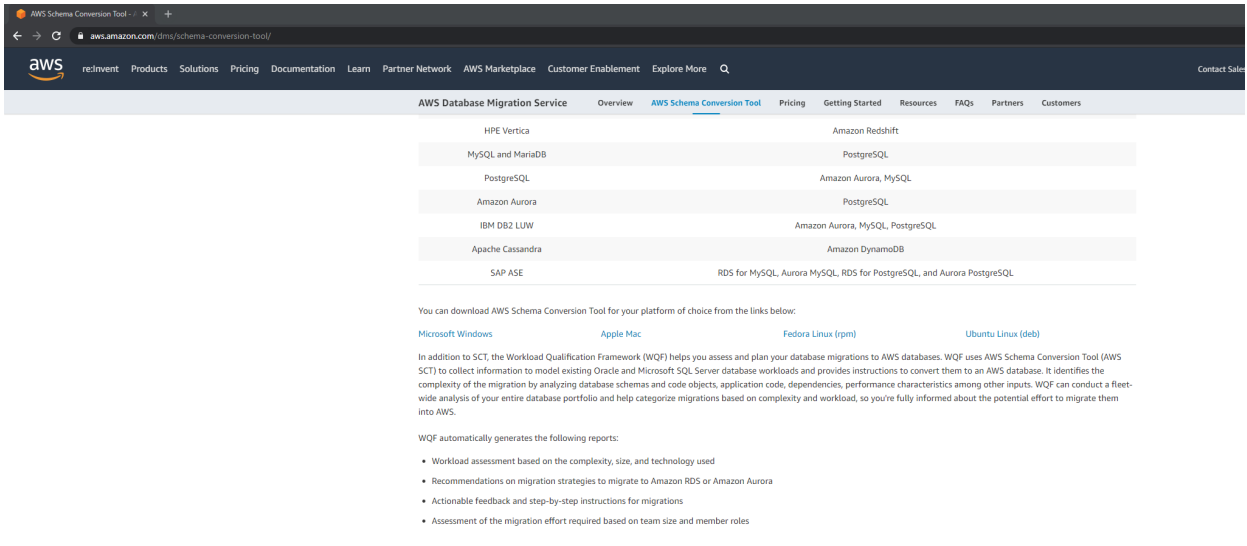
# AWS Database Migration Project

Name: Vlasits Pitsios

Email: [vlasits.pitsios@vodafone.com](mailto:vlasits.pitsios@vodafone.com)

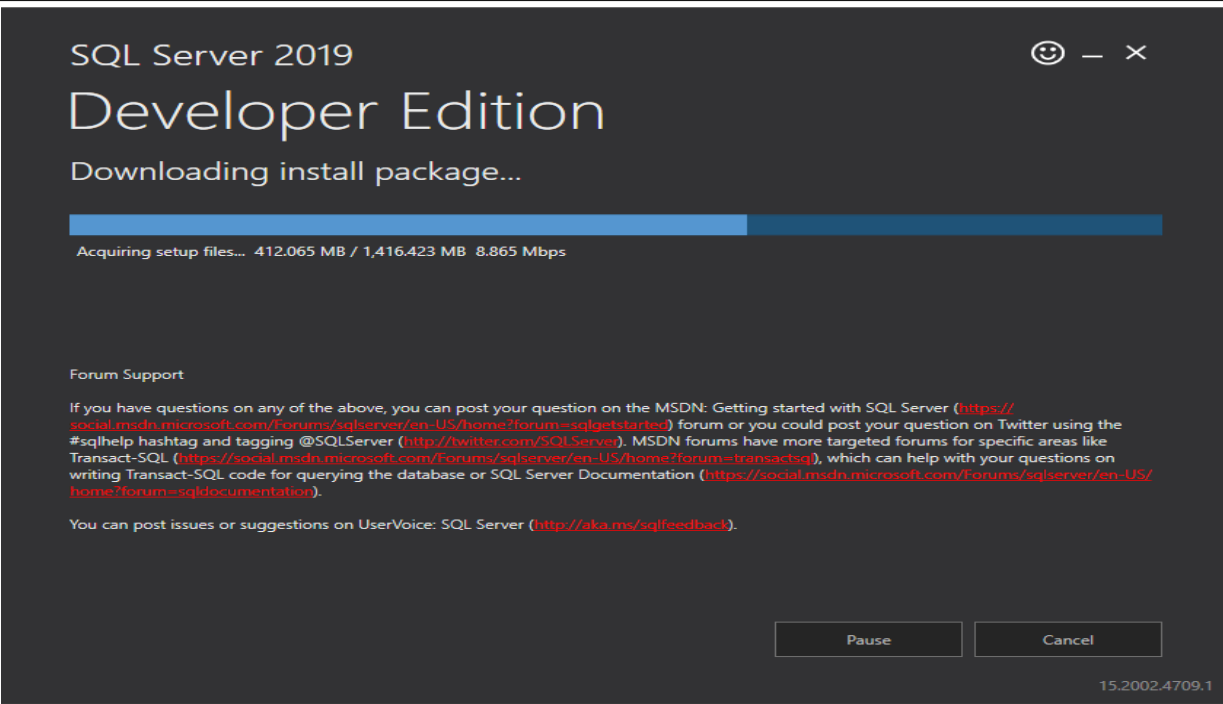
Company: Vodafone

**Main project:** In this project, the learner will install the AWS Schema Conversion Tool and use it to verify the effort required to migrate the SQL Server source database. Fix the issue and rerun the Database Assessment report and apply schema to target MySQL database.



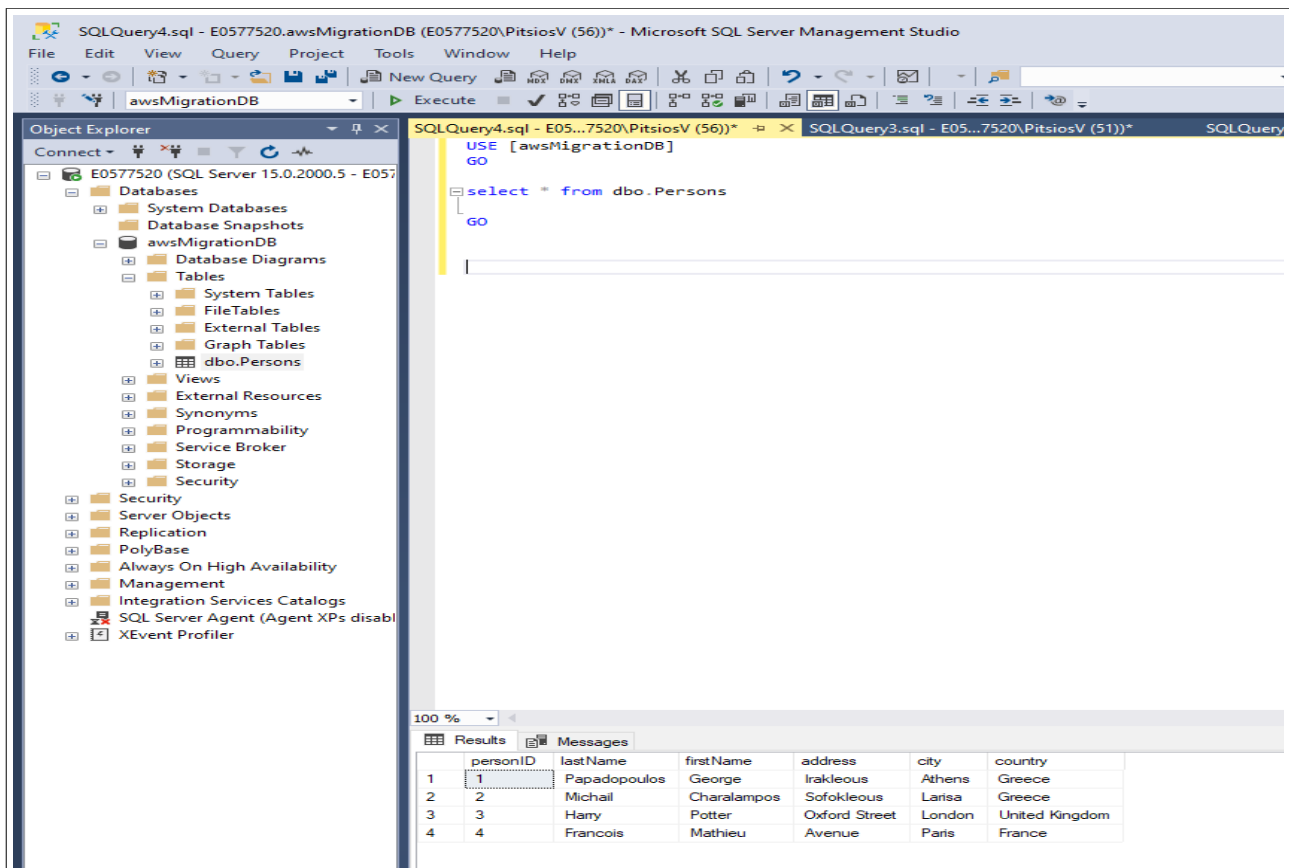
The screenshot shows the AWS Schema Conversion Tool download page. The page has a navigation bar with links like 're:Invent', 'Products', 'Solutions', 'Pricing', 'Documentation', 'Learn', 'Partner Network', 'AWS Marketplace', 'Customer Enablement', 'Explore More', and 'Contact Sales'. Below the navigation bar, there's a section titled 'AWS Database Migration Service' with a sub-section 'AWS Schema Conversion Tool'. This section lists various database migration paths, such as 'HPE Vertica to Amazon Redshift', 'MySQL and MariaDB to PostgreSQL', 'PostgreSQL to Amazon Aurora, MySQL', 'Amazon Aurora to PostgreSQL', 'IBM DB2 LUW to Amazon Aurora, MySQL, PostgreSQL', 'Apache Cassandra to Amazon DynamoDB', and 'SAP ASE to RDS for MySQL, Aurora MySQL, RDS for PostgreSQL, and Aurora PostgreSQL'. Below this list, there's a section 'You can download AWS Schema Conversion Tool for your platform of choice from the links below:' with links for 'Microsoft Windows', 'Apple Mac', 'Fedora Linux (rpm)', and 'Ubuntu Linux (deb)'. Further down, there's a section 'In addition to SCT, the Workload Qualification Framework (WQF) helps you assess and plan your database migrations to AWS databases. WQF uses AWS Schema Conversion Tool (AWS SCT) to collect information to model existing Oracle and Microsoft SQL Server database workloads and provides instructions to convert them to an AWS database. It identifies the complexity of the migration by analyzing database schemas and code objects, application code, dependencies, performance characteristics among other inputs. WQF can conduct a fleet-wide analysis of your entire database portfolio and help categorize migrations based on complexity and workload, so you're fully informed about the potential effort to migrate them into AWS.' Below this, there's a section 'WQF automatically generates the following reports:' with a bulleted list: 'Workload assessment based on the complexity, size, and technology used', 'Recommendations on migration strategies to migrate to Amazon RDS or Amazon Aurora', 'Actionable feedback and step-by-step instructions for migrations', and 'Assessment of the migration effort required based on team size and member roles'.

**Screenshot 1:** Download AWS Schema Conversion Tool from the AWS web site. Version for Microsoft windows was downloaded.

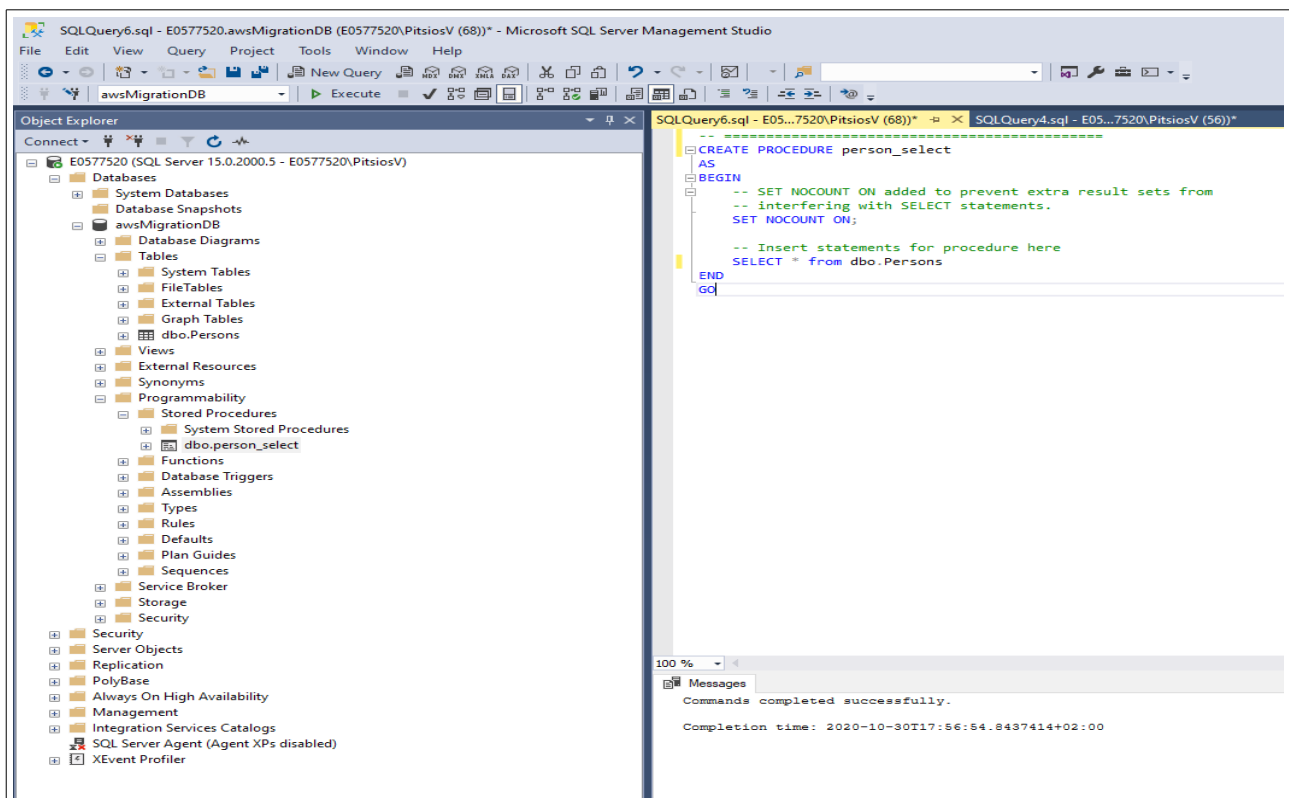


The screenshot shows the SQL Server 2019 Developer Edition installation window. The window title is 'SQL Server 2019 Developer Edition'. The main text says 'Downloading install package...'. Below this, there's a progress bar showing the download progress. The text below the progress bar says 'Acquiring setup files... 412.065 MB / 1,416.423 MB 8.865 Mbps'. Below the progress bar, there's a section 'Forum Support' with text: 'If you have questions on any of the above, you can post your question on the MSDN: Getting started with SQL Server (<https://social.msdn.microsoft.com/Forums/sqlserver/en-US/home?forum=sqlgetstarted>) forum or you could post your question on Twitter using the #sqlhelp hashtag and tagging @SQLServer (<http://twitter.com/SQLServer>). MSDN forums have more targeted forums for specific areas like Transact-SQL (<https://social.msdn.microsoft.com/Forums/sqlserver/en-US/home?forum=transactsql>), which can help with your questions on writing Transact-SQL code for querying the database or SQL Server Documentation (<https://social.msdn.microsoft.com/Forums/sqlserver/en-US/home?forum=sqldocumentation>). You can post issues or suggestions on UserVoice: SQL Server (<http://aka.ms/sqlfeedback>).' At the bottom right, there are two buttons: 'Pause' and 'Cancel'. The bottom right corner of the window shows the version number '15.2002.4709.1'.

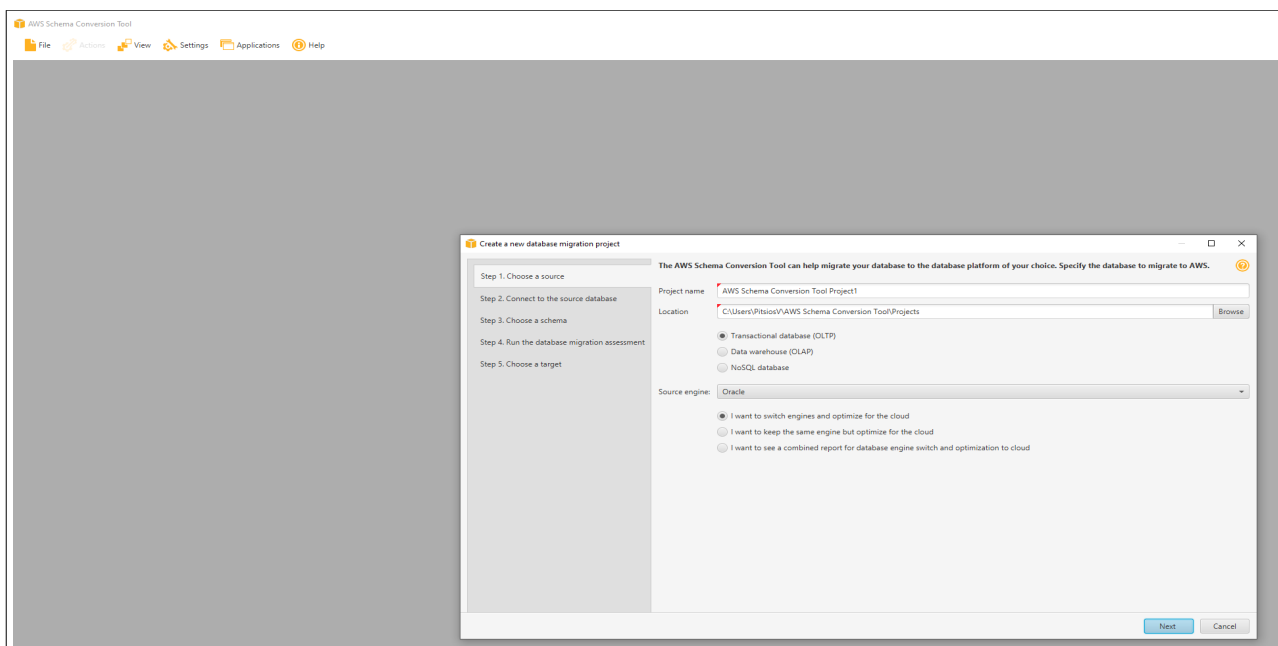
**Screenshot 2:** Download and install locally the SQL Server 2019 Developer edition.



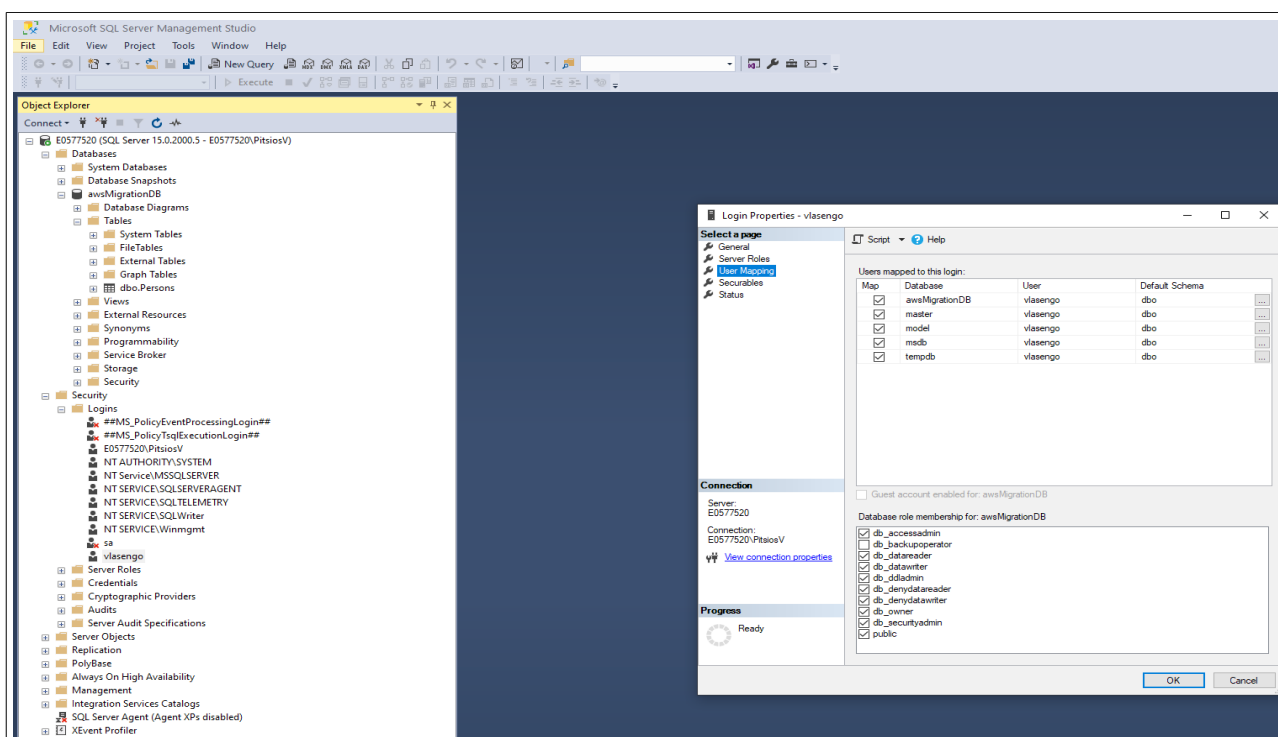
**Screenshot 3:** Database "awsMigrationDB" is created on SQL Server 2019. Insert data on SQL Server 2019 on my machine on a table called "Persons". Microsoft SQL Server Management Studio is used to connect to local SQL Server 2019.



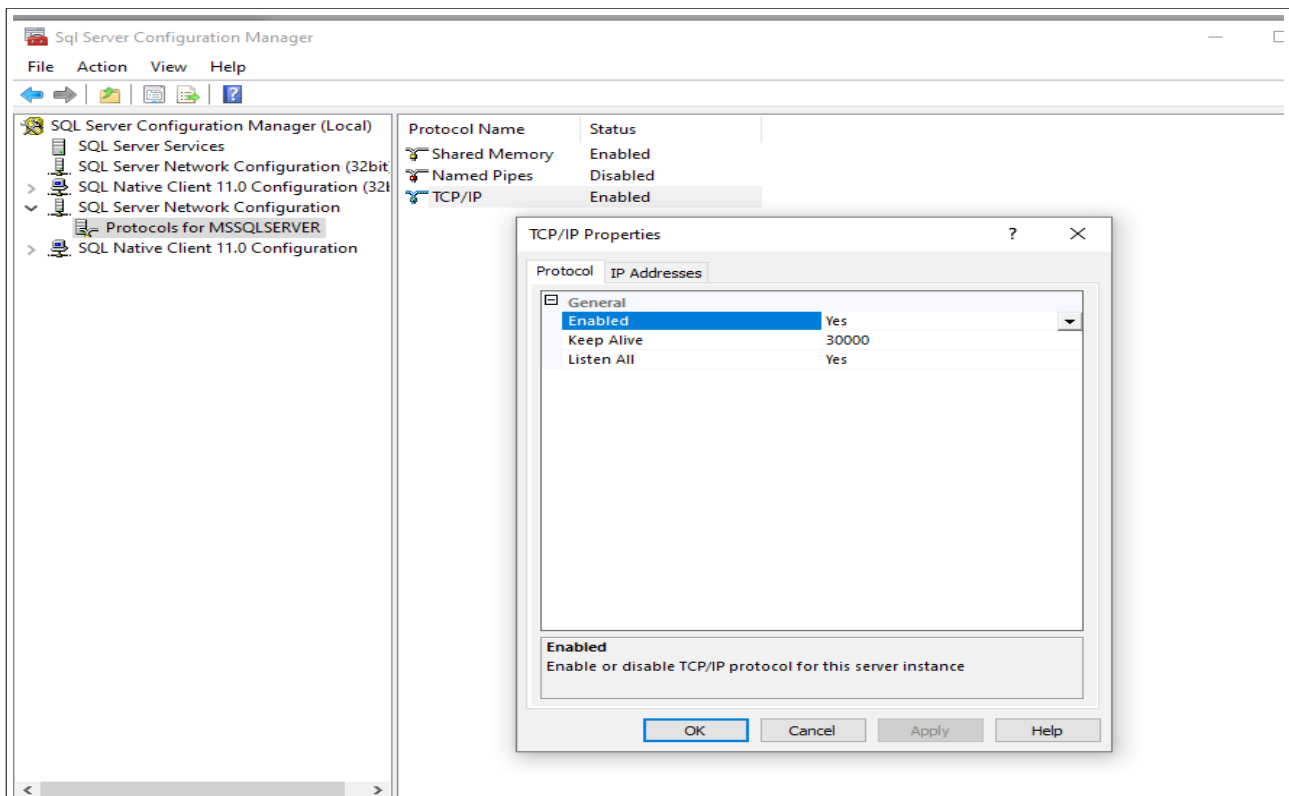
**Screenshot 4:** Create stored procedure on SQL Server 2019 on my machine called "person\_select". The stored procedure contains the command "SET NOCOUNT ON".



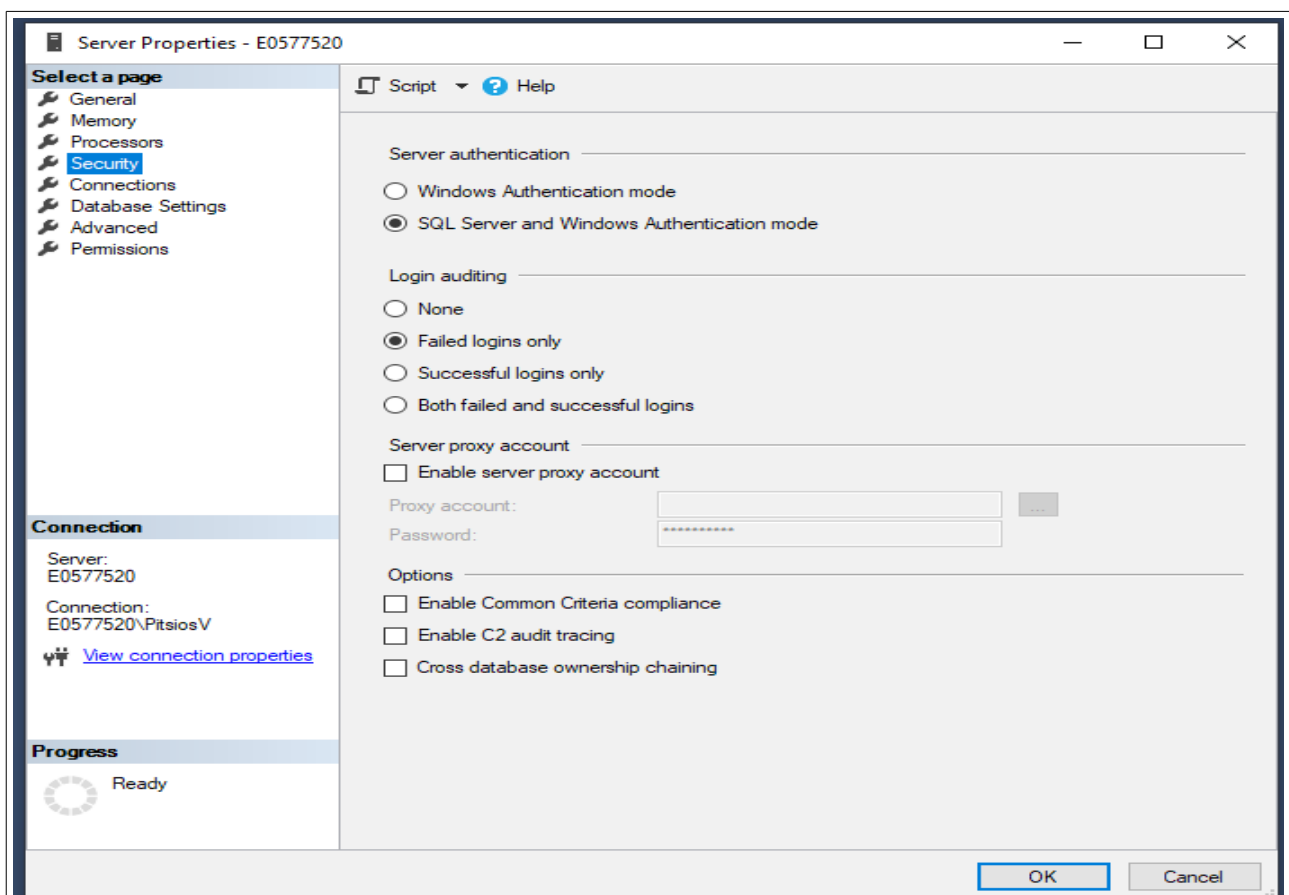
**Screenshot 5:** AWS Schema Conversion Tool installed on my machine.



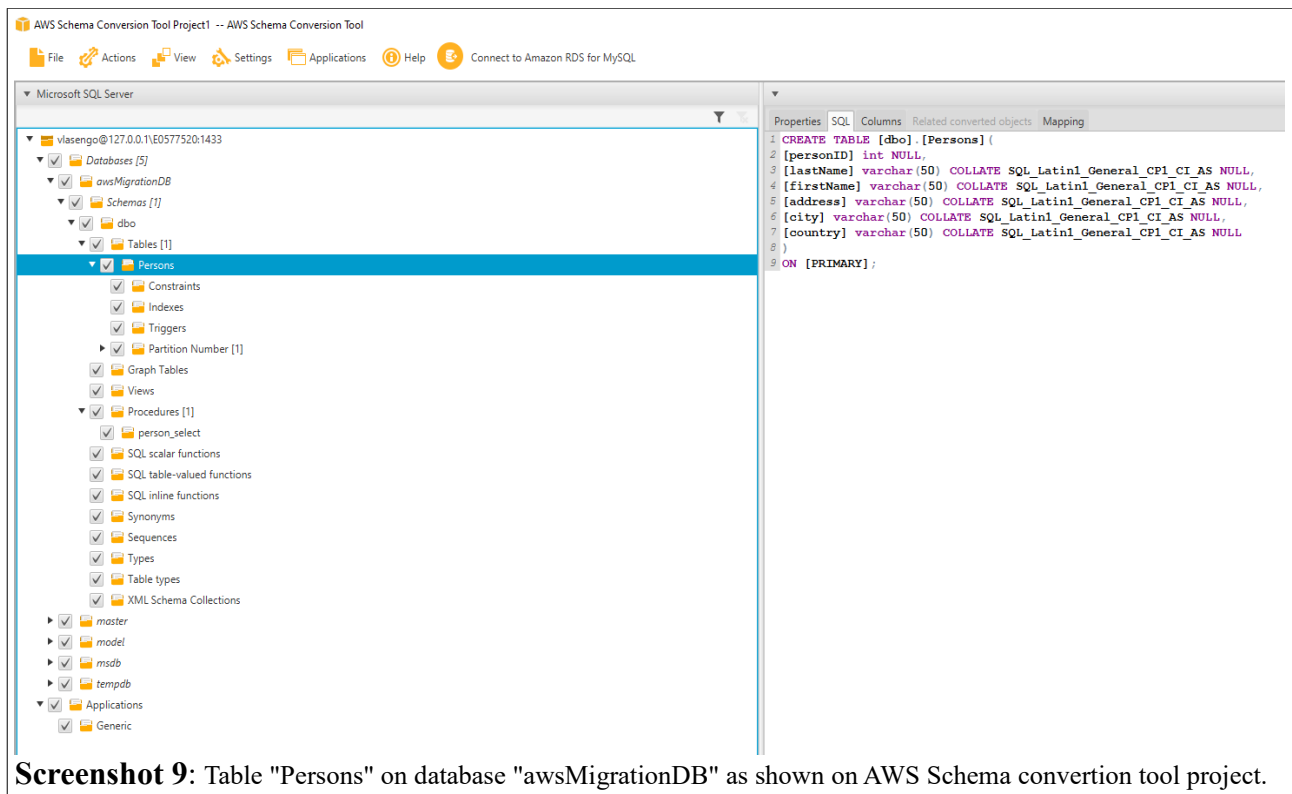
**Screenshot 6:** Create user for migration process on SQL server database using the Microsoft SQL Server Management Studio. The creation of the user is necessary in order to connect to this database from the AWS Schema Conversion Tool.



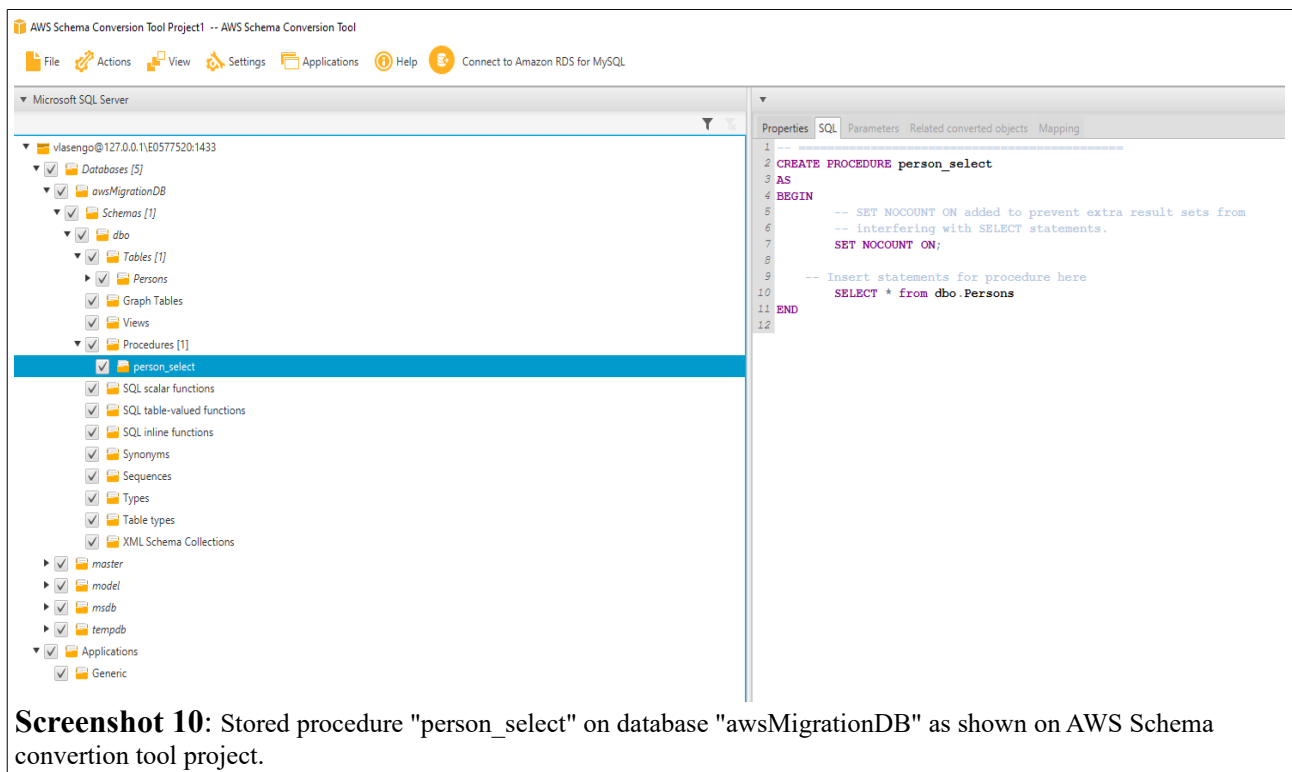
**Screenshot 7:** TCP/IP protocol must be enabled for SQL Server because the AWS Schema Conversion Tool must have access to the database. This is done on "Sql Server Configuration Manager" as shown on the screenshot.



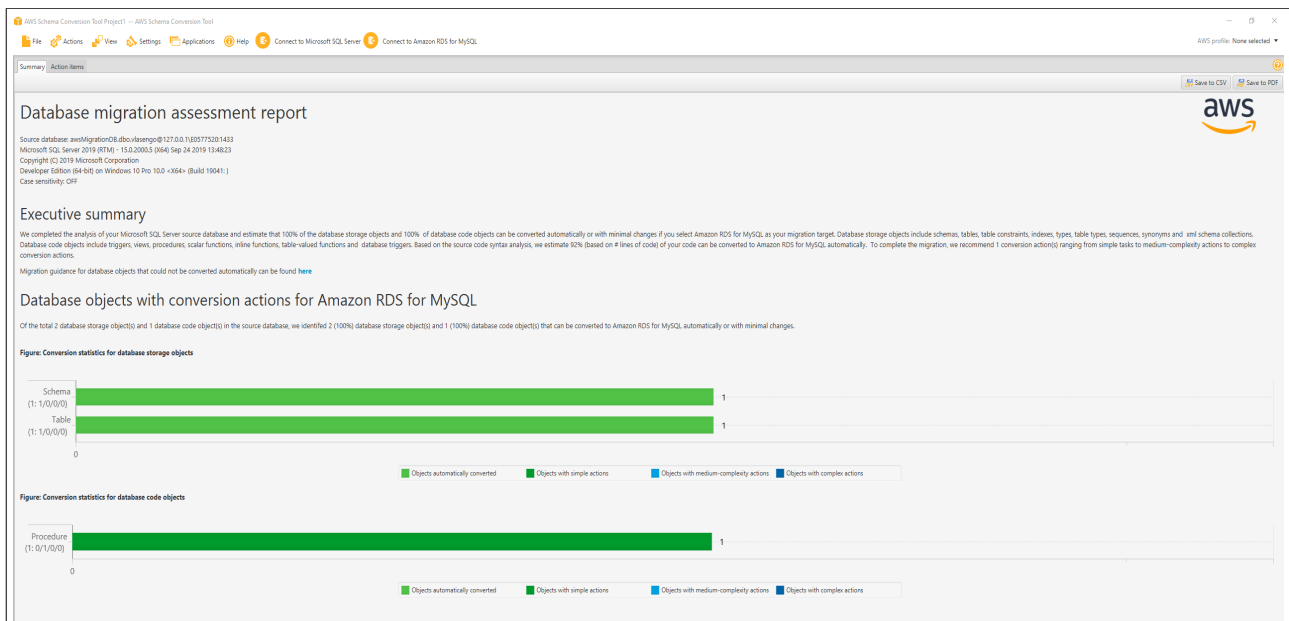
**Screenshot 8:** "SQL Server and Windows Authentication mode" must be checked because the user provided on the AWS Schema Conversion Tool must be authenticated.



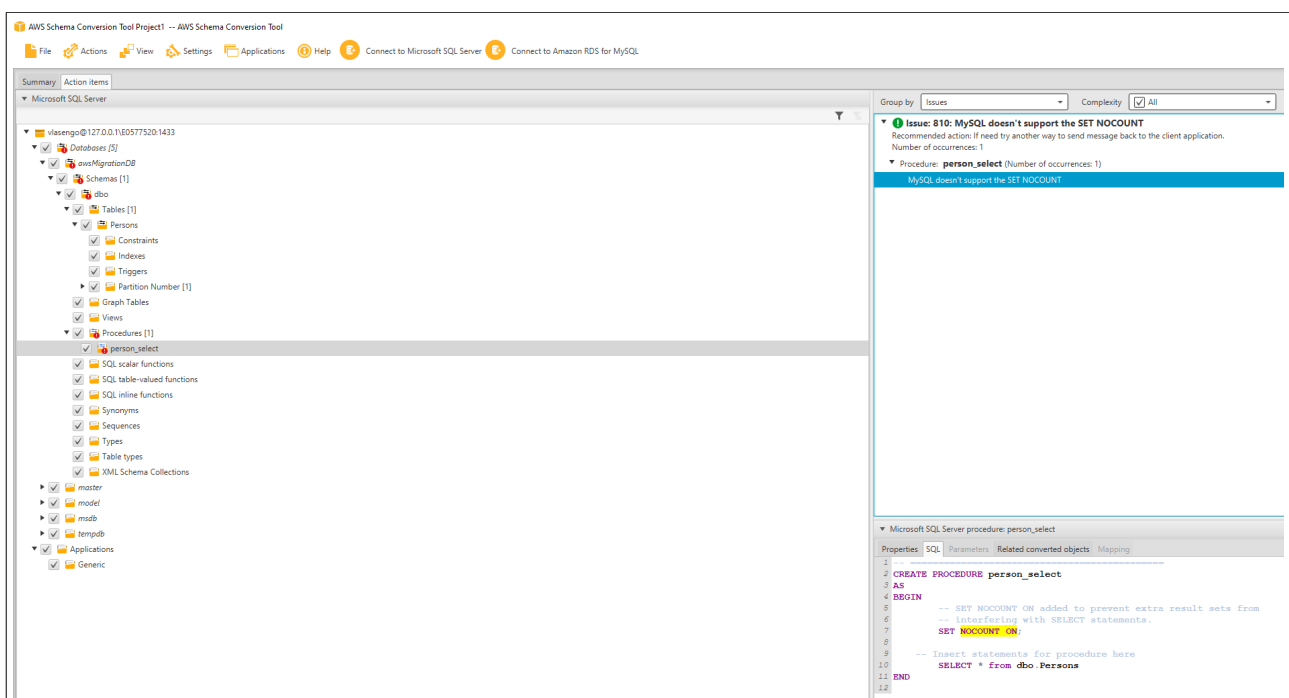
**Screenshot 9:** Table "Persons" on database "awsMigrationDB" as shown on AWS Schema conversion tool project.



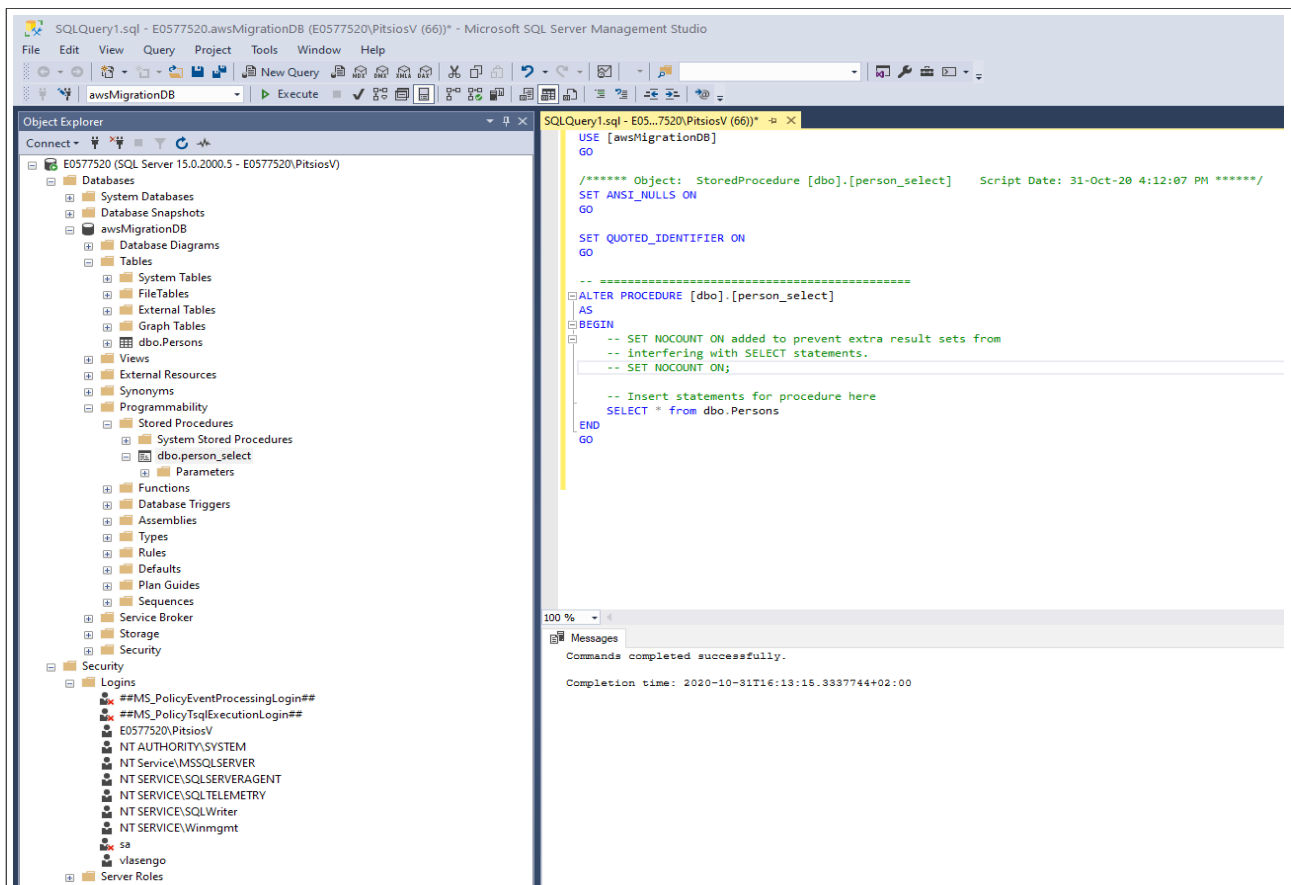
**Screenshot 10:** Stored procedure "person\_select" on database "awsMigrationDB" as shown on AWS Schema conversion tool project.



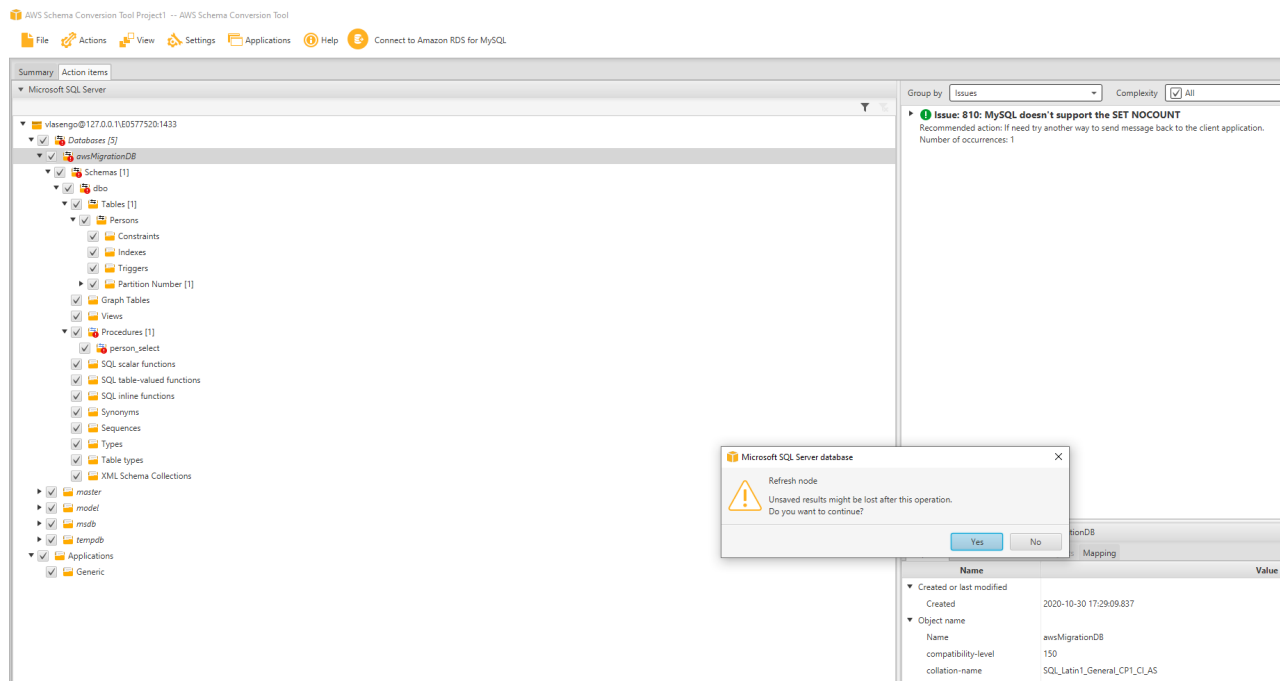
**Screenshot 11:** Migration assessment report was run on AWS Schema conversion tool project. A database schema, a table and a stored procedure identified during the run of the assessment.



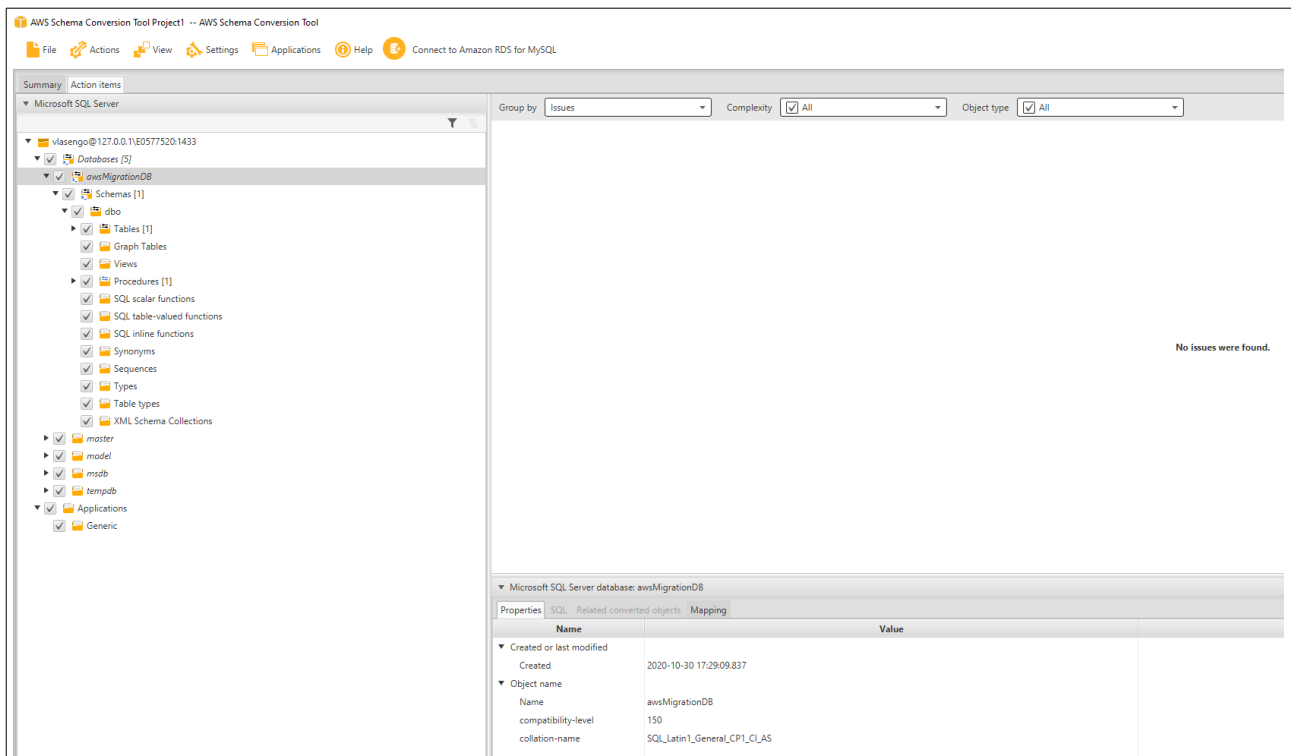
**Screenshot 12:** An issue is reported from the assessment report. Stored procedure contains the command "SET NOCOUNT" which is not supported from MySQL database. We need to fix this issue in order to proceed with the schema migration.



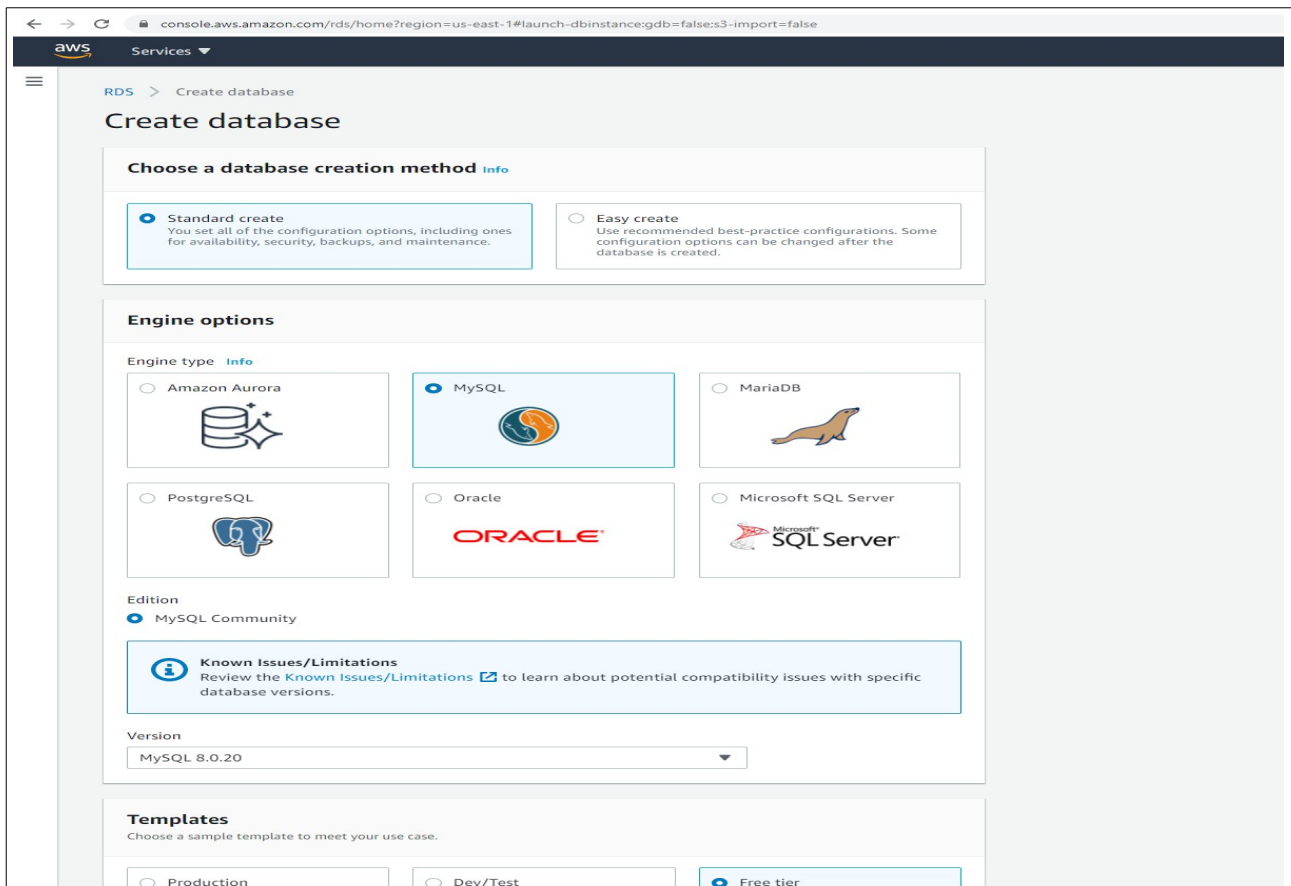
**Screenshot 13:** Issue fixed. Line "SET NOCOUNT ON" is commented out. This was done with the Microsoft SQL Server Management Studio on SQL Server 2019.



**Screenshot 14:** On AWS Schema Conversion Tool, the node must be refreshed in order to read the latest changes from the database SQL Server 2019. This is done by clicking on "awsMigrationDB" and choose "Refresh from database".

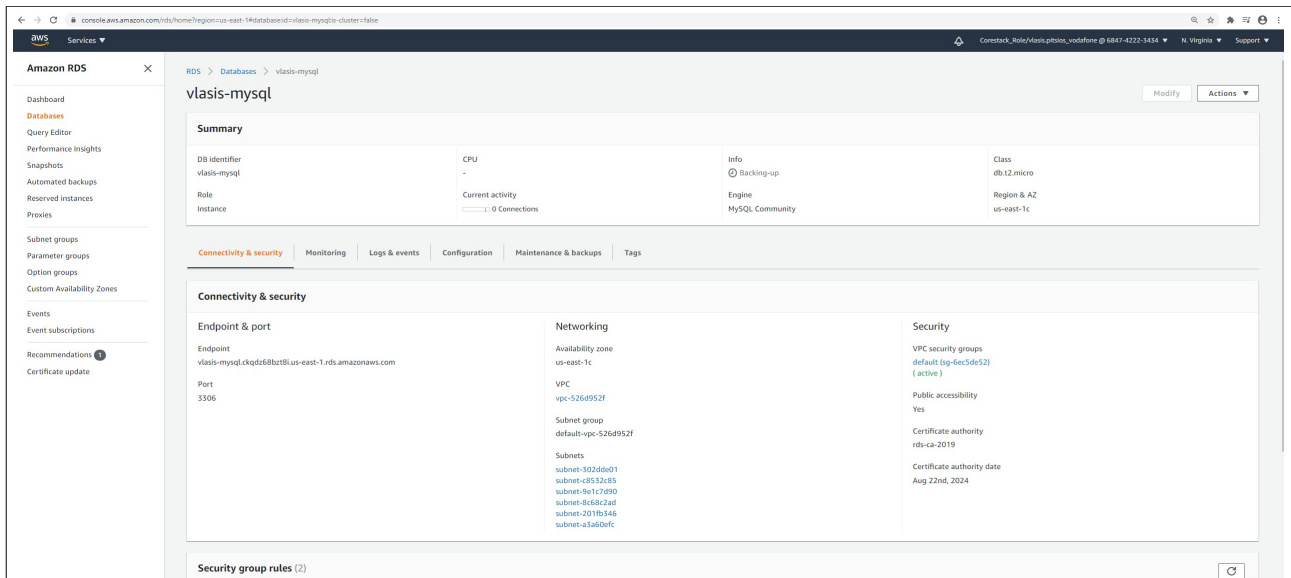


**Screenshot 15:** Assessment report was run again on AWS Schema Conversion Tool without any issues.

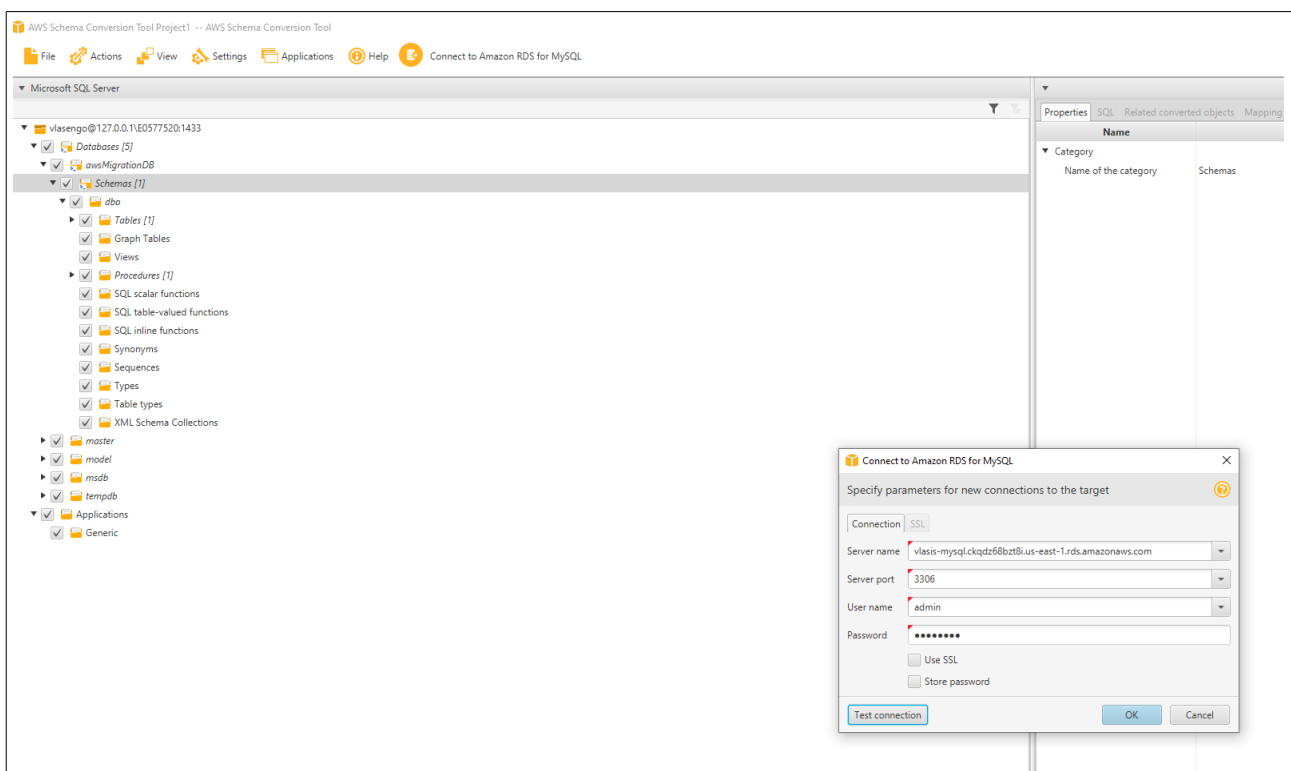


**Screenshot 16:** Amazon RDS MySQL database was created on AWS account.

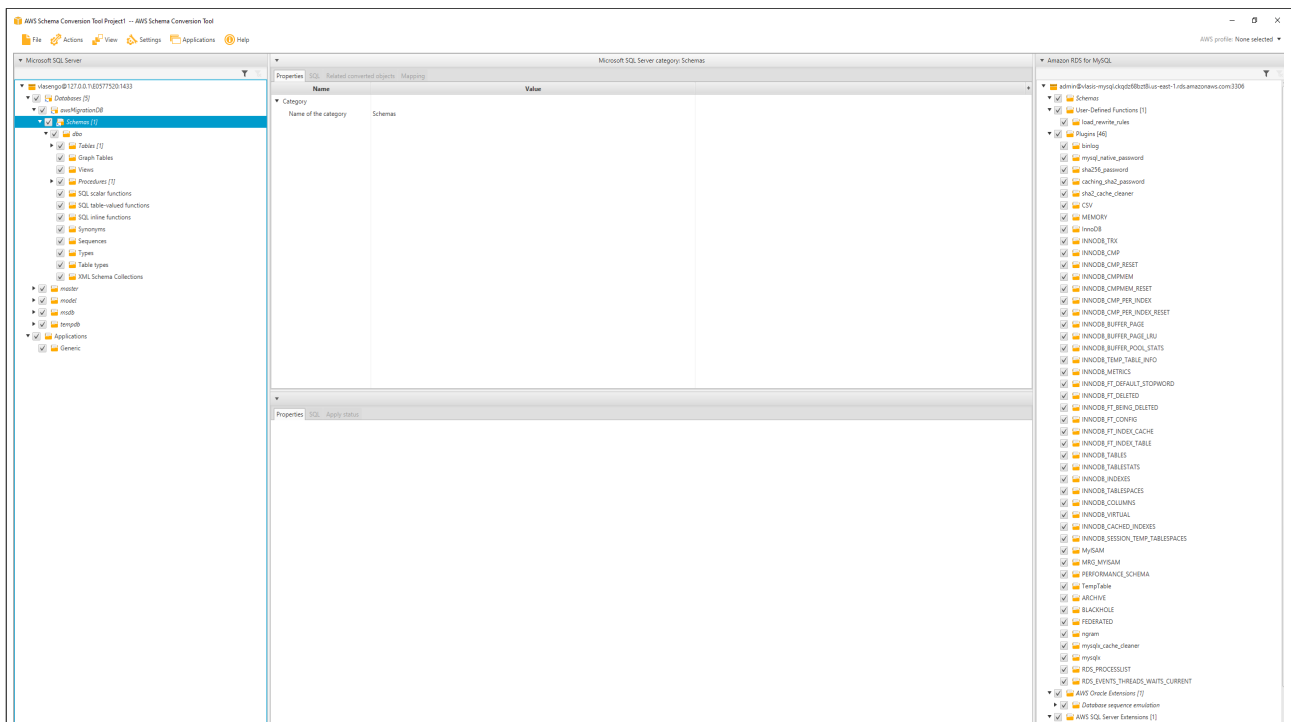




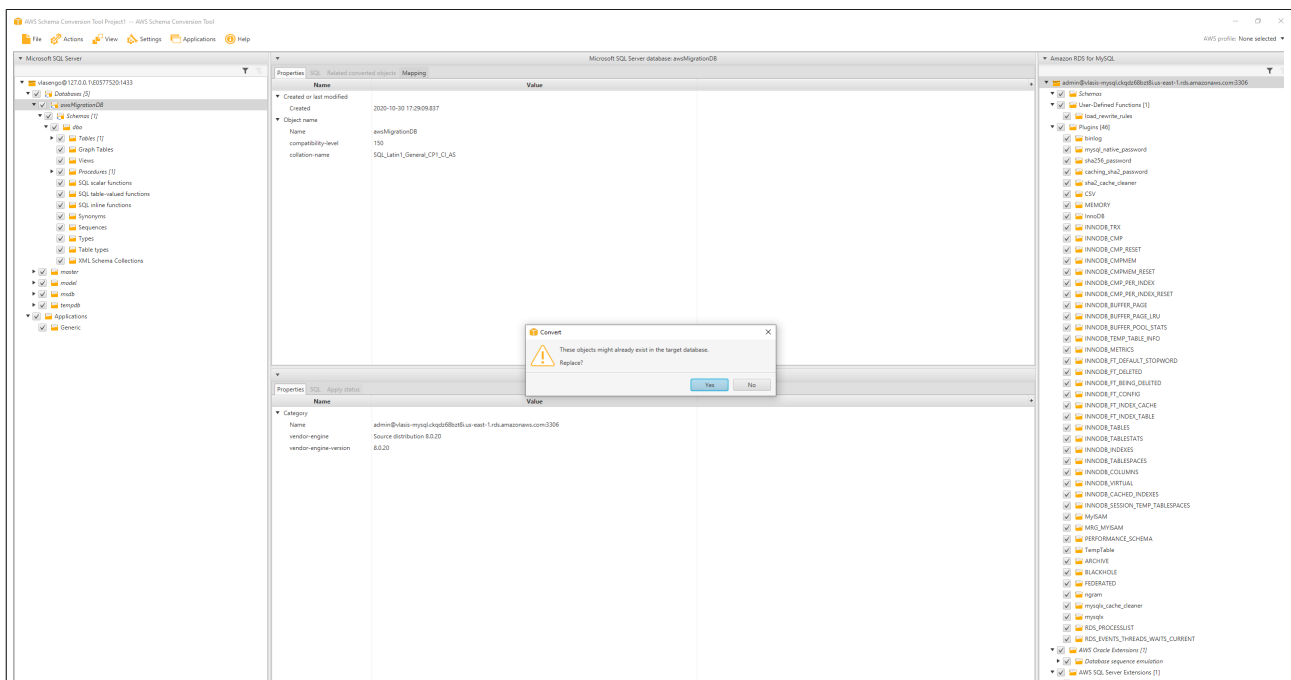
**Screenshot 17:** Amazon RDS MySQL database was created on AWS account with public accessibility on endpoint "vlasis-mysql.ckqdz68bzt8i.us-east-1.rds.amazonaws.com" and port "3306".



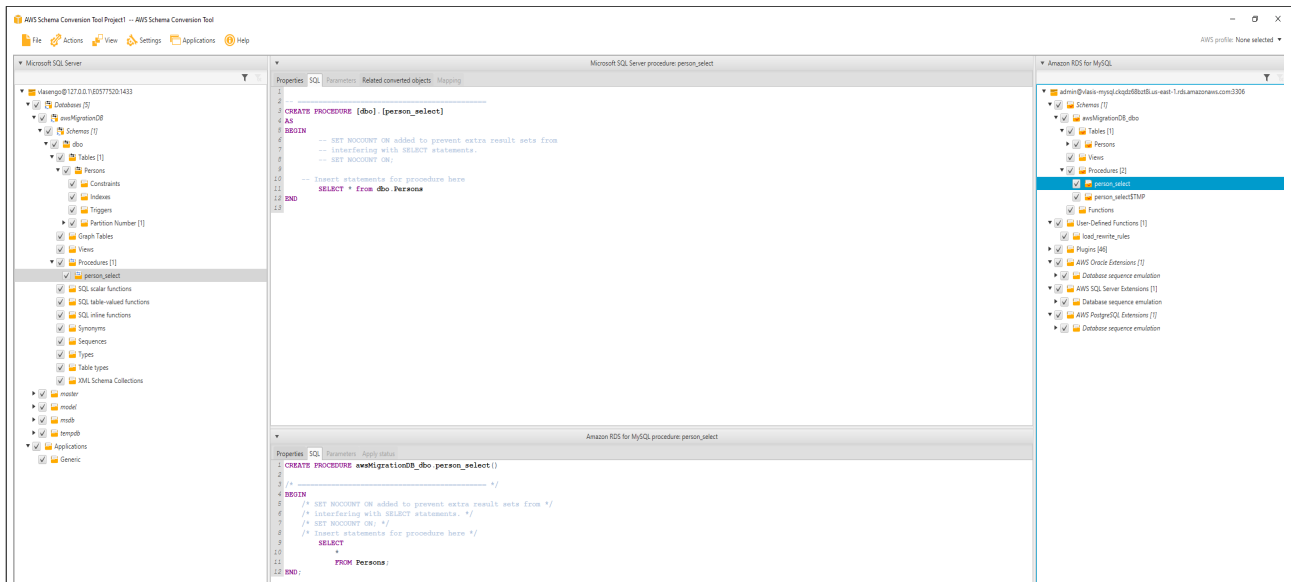
**Screenshot 18:** Connect to Amazon RDS MySQL database which created on a previous step from AWS Schema Conversion Tool on endpoint "vlasis-mysql.ckqdz68bzt8i.us-east-1.rds.amazonaws.com" and port "3306".



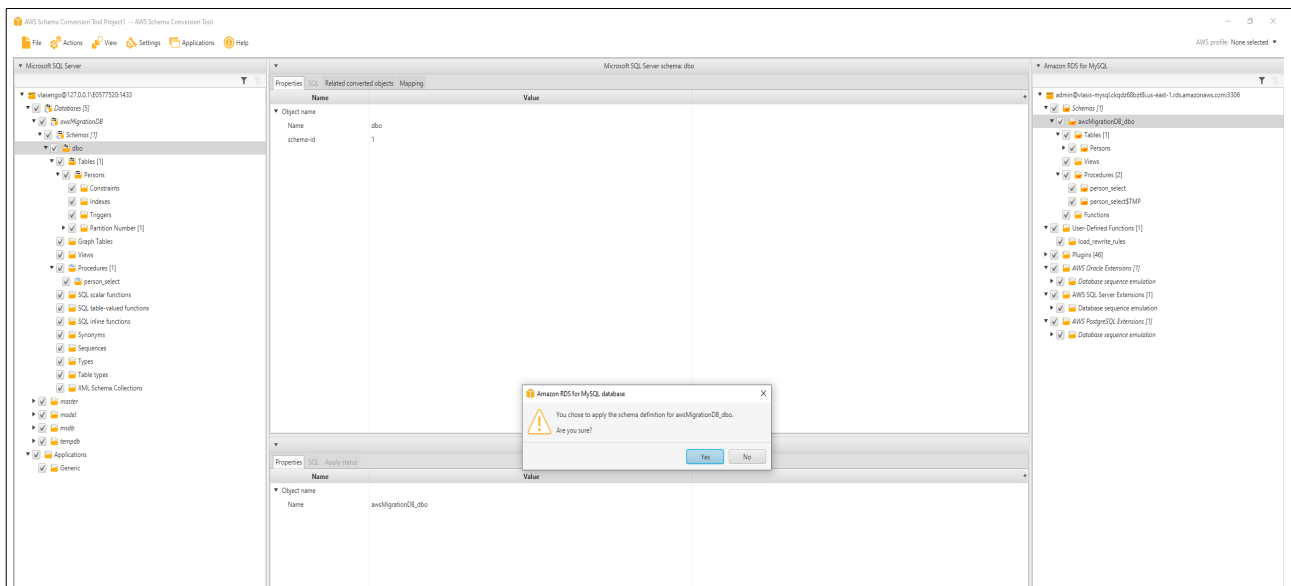
**Screenshot 19:** AWS Schema Conversion Tool screen after connecting to Amazon RDS MySQL database. On the left, the SQL Server 2019 on my local machine is presented and on the right the Amazon RDS MySQL database on AWS. Until now, the database "awsMigrationDB" is not created on Amazon RDS MySQL database, that's why is not presented on the right of the screenshot.



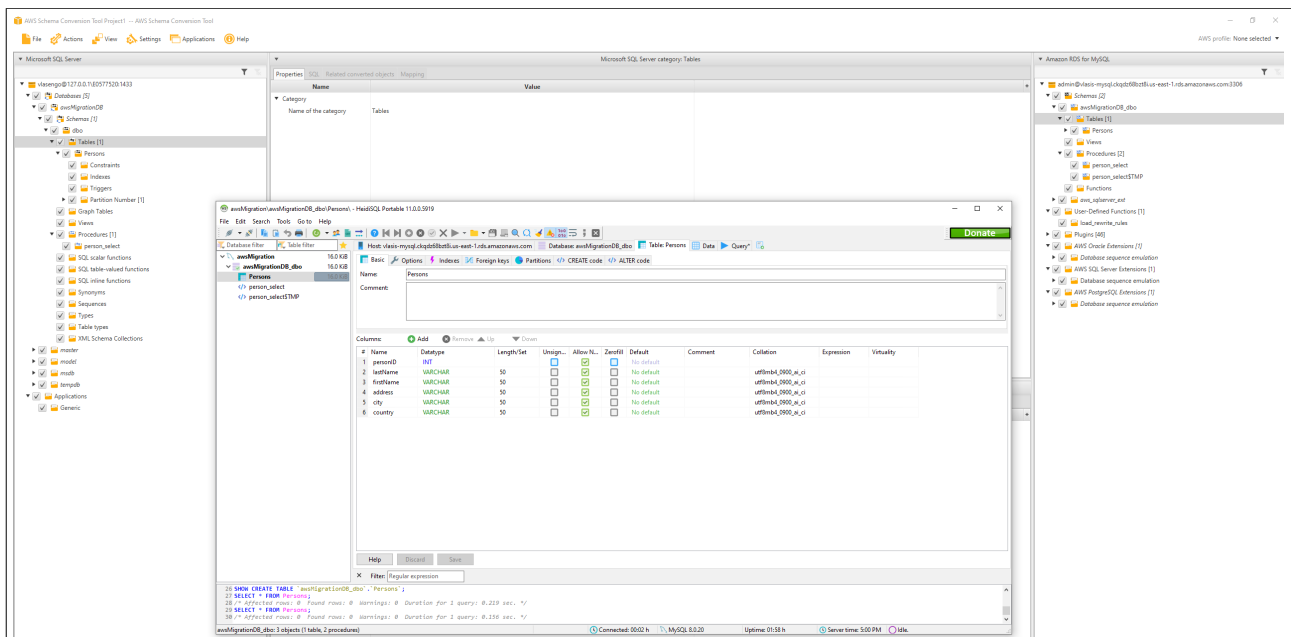
**Screenshot 20:** To create the schema on the Amazon RDS MySQL database, right click on the "awsMigrationDB" and then press "Convert Schema". AWS Schema Conversion Tool shows the pop up on this screenshot. If pressing "Yes", schema conversion is starting.



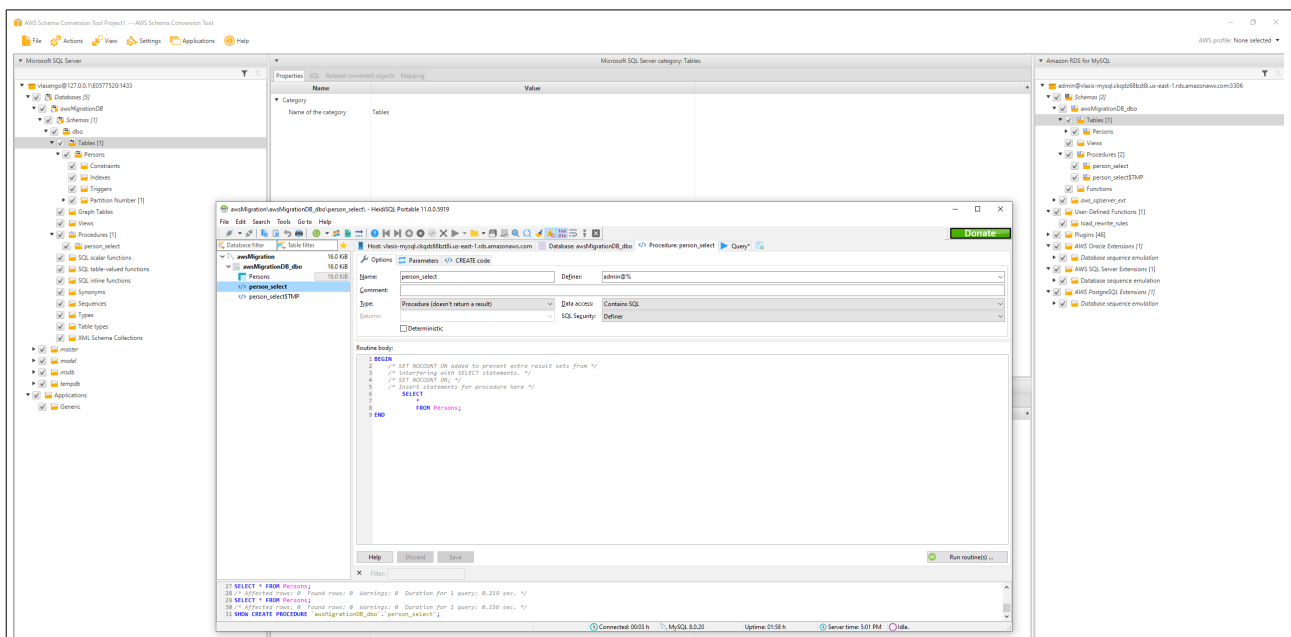
**Screenshot 21:** The schema is migrated to the Amazon RDS MySQL database. Table "Persons" and stored procedure "person\_select" are shown on the right of the image. These two database entities are created on the AWS Schema Conversion Tool, but are not applied yet on the actual database. One more step is necessary to complete the schema migration.



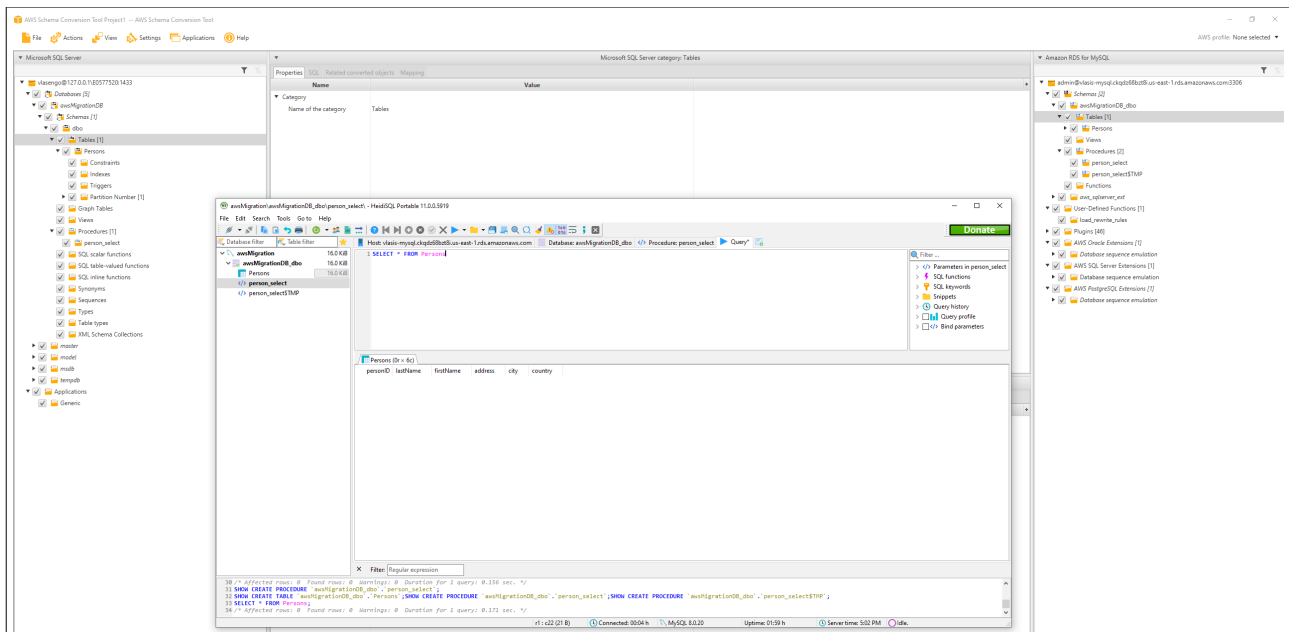
**Screenshot 22:** To apply the created schema on the Amazon RDS MySQL database right click on the "awsMigrationDB\_dbo" and then click "Apply to database". Then, the schema will be stored on the database.



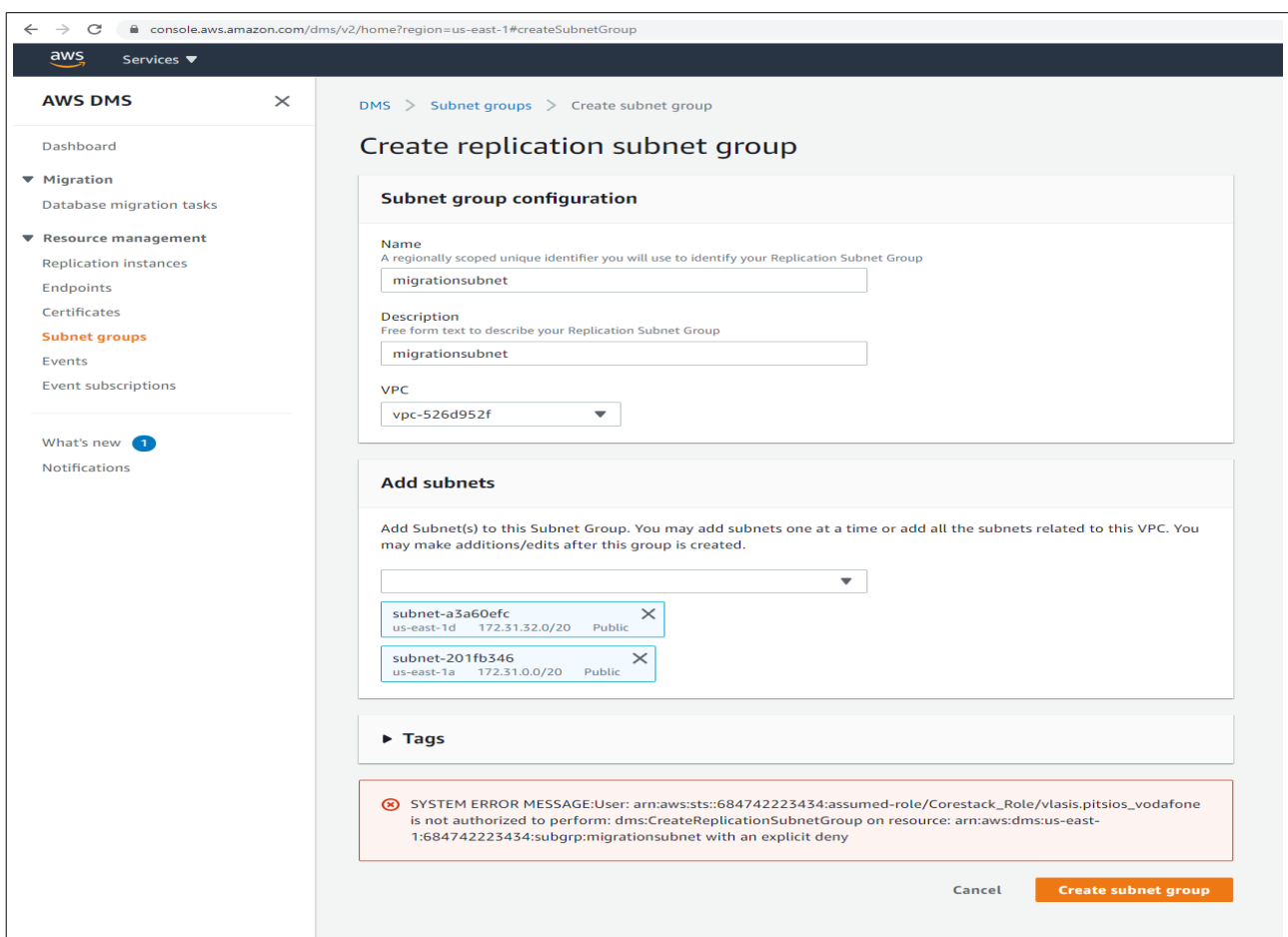
**Screenshot 23:** Using HeidiSQL client to connect to Amazon RDS MySQL database with endpoint "vlasismysql.ckqdz68bzt8i.us-east-1.rds.amazonaws.com", we verify that the table "Persons" is created.



**Screenshot 24:** Using HeidiSQL client to connect to Amazon RDS MySQL database with endpoint "vlasismysql.ckqdz68bzt8i.us-east-1.rds.amazonaws.com", we can verify that the stored procedure "person\_select" is created.



**Screenshot 25:** Table "Persons" is empty because only the database schema was migrated with the previous steps. In order to migrate the data, the next step is to use the AWS Data Migration Service (DMS). In order to do that we need to create a replication subnet group, a replication instance and a data migration task. If all the steps are completed, the data to the "Persons" will be migrated.



**Screenshot 26:** My account on AWS does not provide access to create a replication subnet group in order to proceed with the replication instance creation, which is necessary to complete the migration of the data to the Amazon RDS MySQL database.