EX.NO 10: Implement database migration in Microsoft Azure

Date:

READING MATERIALS:

Microsoft Windows Azure is a cloud operating system built on top of Microsoft datacenters' infra-Structure and provides developers with a collection of services for building applications with cloud technology. Services range from compute, storage, and networking to application connectivity, access control, and business intelligence. Any application that is built on the Microsoft technology can be scaled using the Azure platform, which integrates the scalability features into the common Microsoft technologies such as Microsoft Windows Server 2008, SQL Server, and ASP.NET.

Storage services

Compute resources are equipped with local storage in the form of a directory on the local file system that can be used to temporarily store information that is useful for the current execution cycle of a role. If the role is restarted and activated on a different physical machine, this information is lost. Windows Azure provides different types of storage solutions that complement compute services with a more durable and redundant option compared to local storage. Compared to local storage, these services can be accessed by multiple clients at the same time and from everywhere, thus becoming a general solution for storage.

Blobs

Azure allows storing large amount of data in the form of binary large objects (BLOBs) by means of the blobs service. This service is optimal to store large text or binary files. Two types of blobs are available:

• Block blobs. Block blobs are composed of blocks and are optimized for sequential access; therefore they are appropriate for media streaming. Currently, blocks are of 4 MB, and a single block blob can reach 200 GB in dimension.

• Page blobs. Page blobs are made of pages that are identified by an offset from the beginning of the blob. A page blob can be split into multiple pages or constituted of a single page. This type of blob is

optimized for random access and can be used to host data different from streaming. Currently, the maximum dimension of a page blob can be 1 TB.

Blobs storage provides users with the ability to describe the data by adding metadata. It is also possible to take snapshots of a blob for backup purposes. Moreover, to optimize its distribution, blobs storage can leverage the Windows Azure CDN so that blobs are kept close to users requesting them and can be served efficiently

Azure drive Page blobs can be used to store an entire file system in the form of a single Virtual Hard Drive (VHD) file. This can then be mounted as a part of the NTFS file system by Azure compute resources, thus providing persistent and durable storage. A page blob mounted as part of an NTFS tree is called an Azure Drive.

Tables

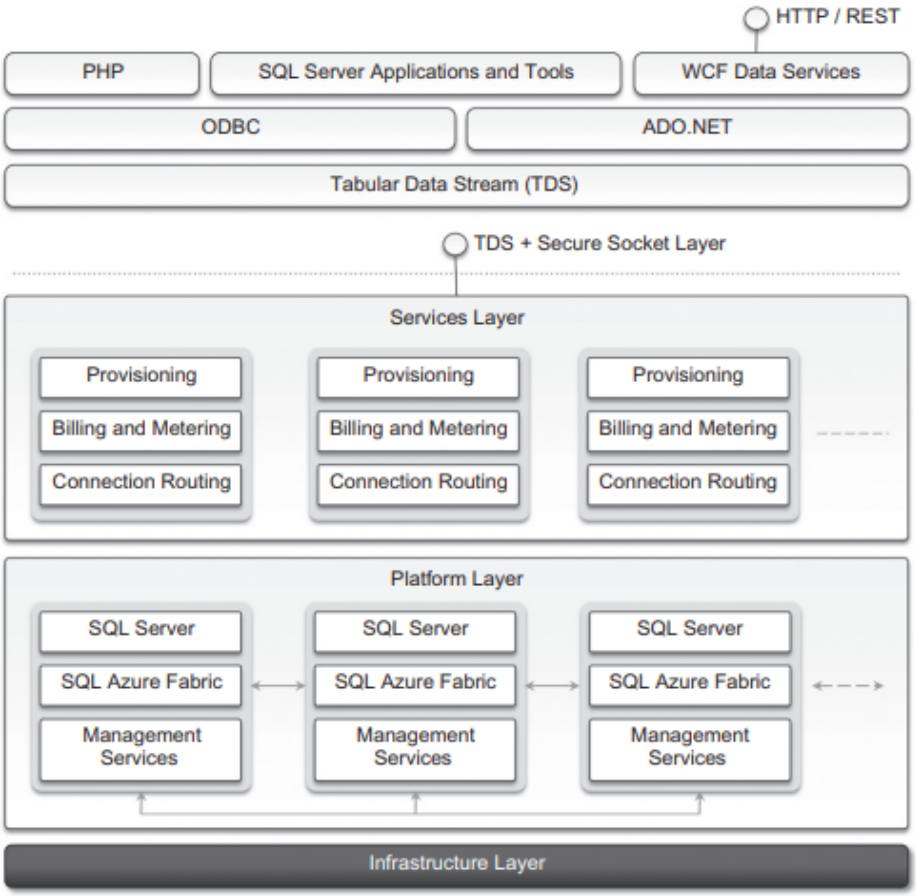
Tables constitute a semistructured storage solution, allowing users to store information in the form of entities with a collection of properties. Entities are stored as rows in the table and are identified by a key, which also constitutes the unique index built for the table. Users can insert, update, delete, and select a subset of the rows stored in the table. Unlike SQL tables, there are no schema enforcing constraints on the properties of entities and there is no facility for representing relationships among entities. For this reason, tables are more similar to spreadsheets rather than SQL tables.

The service is designed to handle large amounts of data and queries returning huge result sets. This capability is supported by partial result sets and table partitions. A partial result set is returned together with a continuation token, allowing the client to resume the query for large result sets. Table partitions allow tables to be divided among several servers for load-balancing purposes. A partition is identified by a key, which is represented by three of the columns of the table. Currently, a table can contain up to 100 TB of data, and rows can have up to 255 properties, with a maximum of 1 MB for each row. The maximum dimension of a row key and partition keys is 1 KB.

Queues Queue storage allows applications to communicate by exchanging messages through durable queues, thus avoiding lost or unprocessed messages. Applications enter messages into a queue, and other applications can read them in a first-in, first-out (FIFO) style. To ensure that messages get processed, when an application reads a message it is marked as invisible; hence it will not be available to other clients. Once the application has completed processing the message, it needs to explicitly delete the message from the queue. This two-phase process ensures that messages get processed before they are removed from the queue, and the client failures do not prevent messages from being processed. At the same time, this is also a reason that the queue does not enforce a strict FIFO model: Messages that are read by applications that crash during processing are made available again after a timeout, during which other messages can be read by other clients. An alternative to reading a message is peeking, which allows retrieving the message but letting it stay visible in the queue. Messages that are peeked are not considered processed. All the services described are geo-replicated three times to ensure their availability in case of major disasters. Geo-replication involves the copying of data into a different datacenter that is hundreds or thousands of miles away from the original datacenter.

SQL Azure

SQL Azure is a relational database service hosted on Windows Azure and built on the SQL Server technologies. The service extends the capabilities of SQL Server to the cloud and provides developers with a scalable, highly available, and fault-tolerant relational database. SQL Azure is accessible from either the Windows Azure Cloud or any other location that has access to the Azure Cloud. It is fully compatible with the interface exposed by SQL Server, so applications built for SQL Server can transparently migrate to SQL Azure. Moreover, the service is fully manageable using REST APIs, allowing developers to control databases deployed in the Azure Cloud as well as the firewall rules set up for their accessibility



Access to SQL Azure is based on the Tabular Data Stream (TDS) protocol, which is the communication protocol underlying all the different interfaces used by applications to connect to a SQL Server-based installation such as ODBC and ADO.NET. On the SQL Azure side, access to data is mediated by the service layer, which provides provisioning, billing, and connection-routing services. These services are logically part of server instances, which are managed by SQL Azure Fabric. This is the distributed database middleware that constitutes the infrastructure of SQL Azure and that is deployed on Microsoft datacenters. Developers have to sign up for a Windows Azure account in order to use SQL Azure. Once the account is activated, they can either use the Windows Azure Management Portal or the REST APIs to create servers and logins and to configure access to servers. SQL Azure servers are abstractions that closely resemble physical SQL Servers: They have a fully qualified domain name under the database.windows.net (i.e., server-name.database.windows.net) domain name. This simplifies the management tasks and the interaction with SQL Azure from client applications. SQL Azure ensures that multiple copies of each server are maintained within the Azure Cloud and that these copies are kept synchronized when client applications insert, update, and delete data on them.

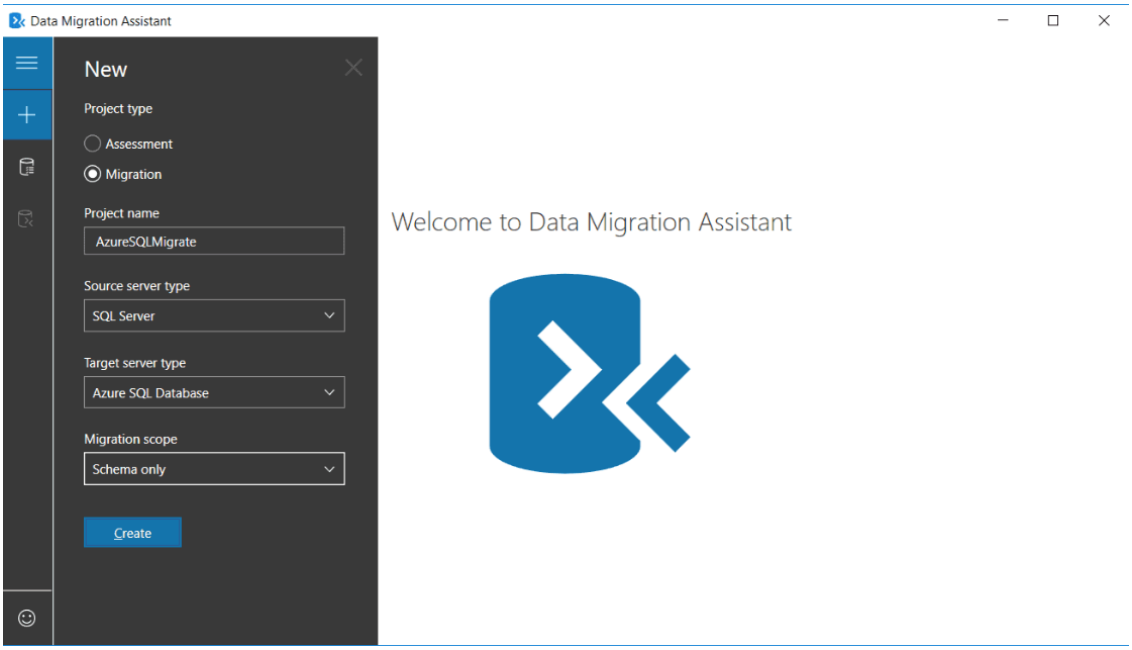
EX.NO 10: Implement database migration in Microsoft Azure

Date:

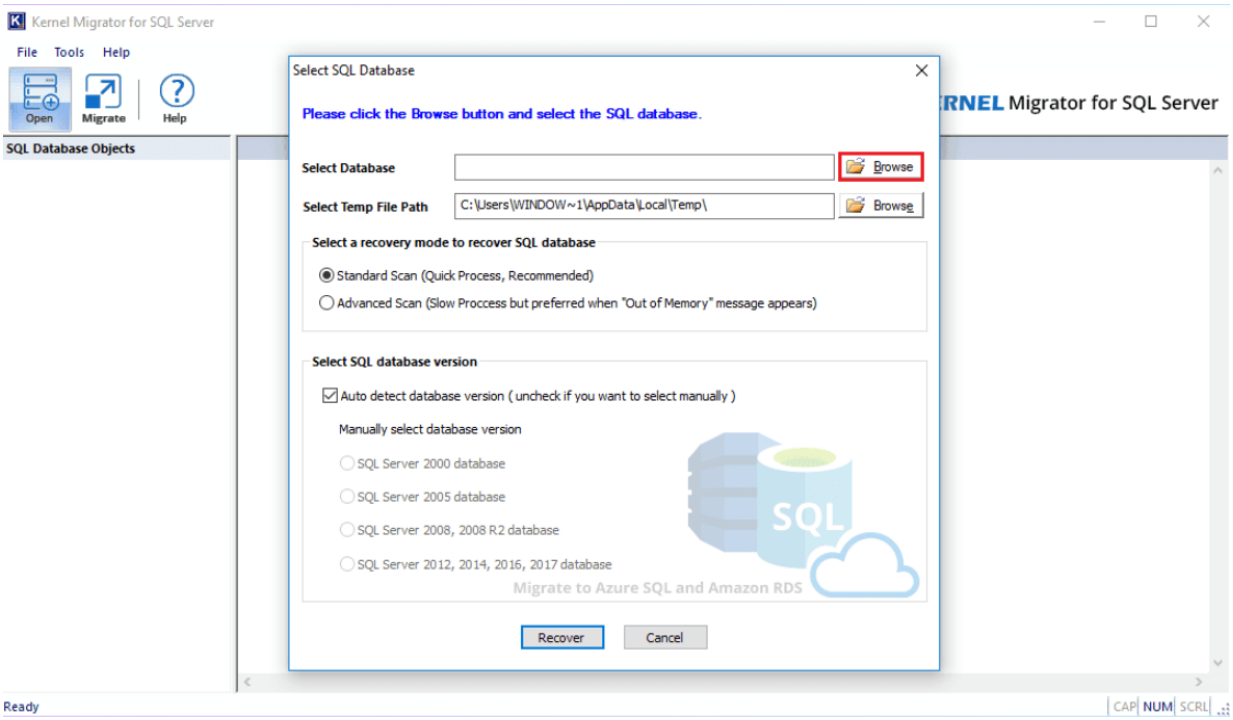
AIM:

PROCEDURE with SCREENSHOTS:

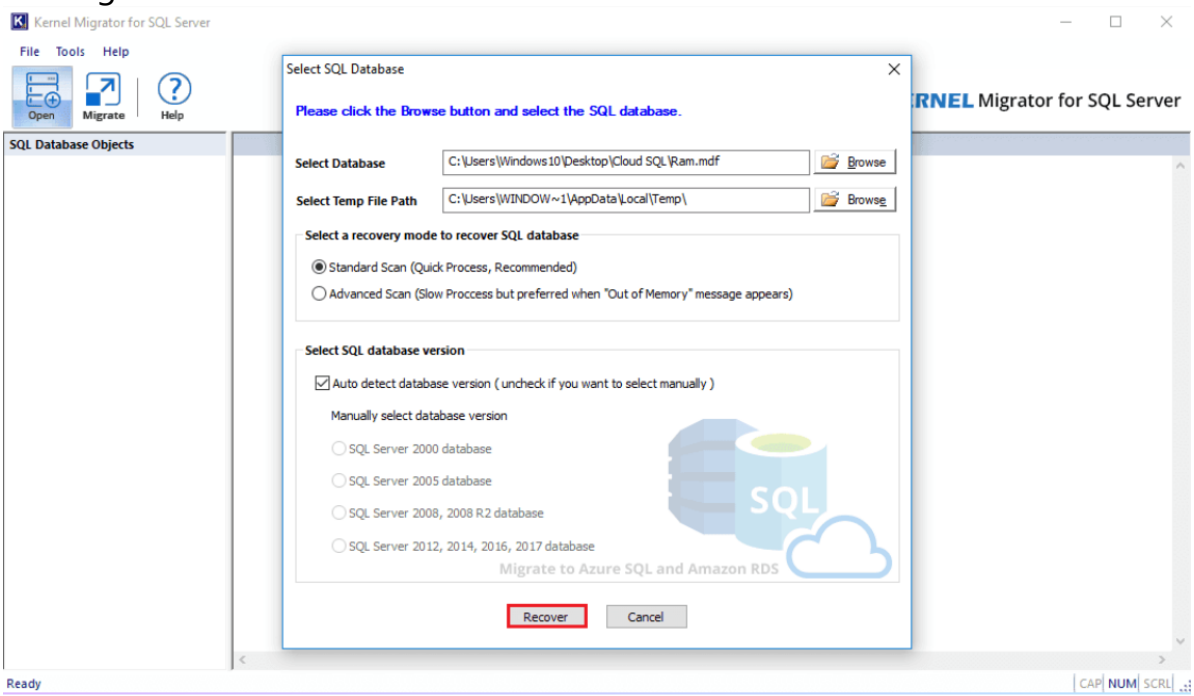
Step 1: Run the Data Migration Assistant tool, select the New (+) icon, and choose a new Migration option under the Project type.



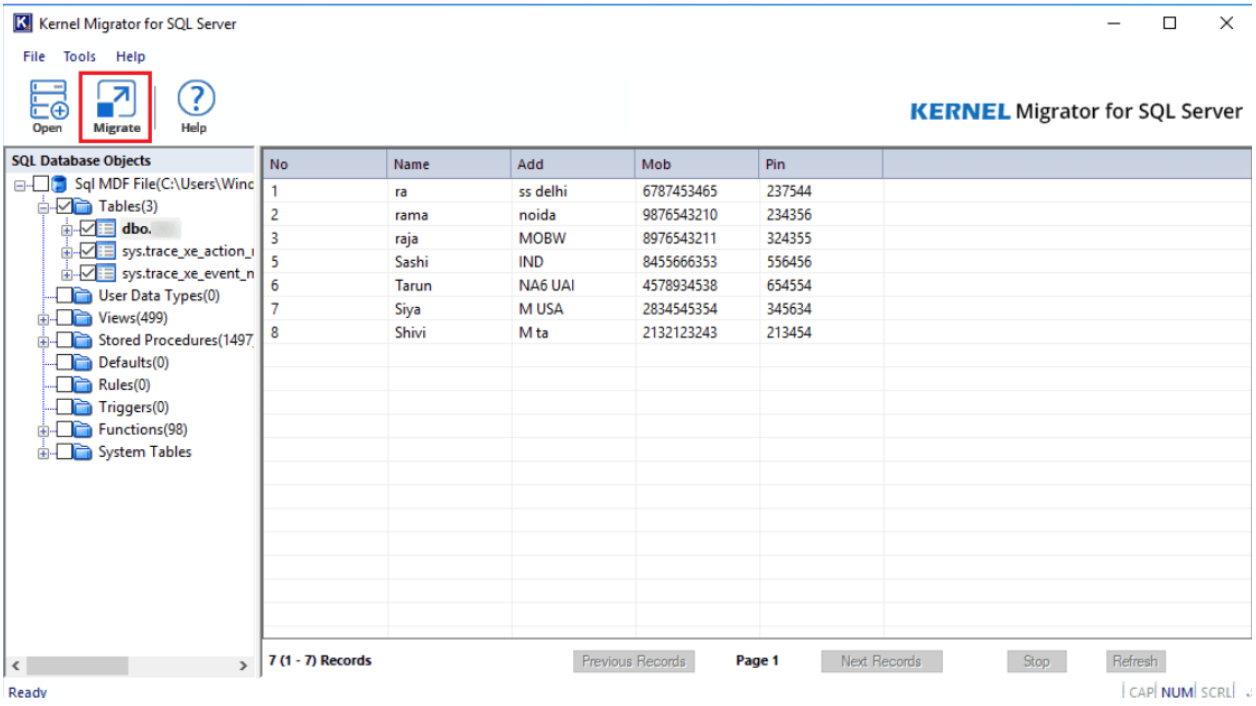
Step 2: Start the software, and the Select SQL Database will pop up. Click the Browse button.



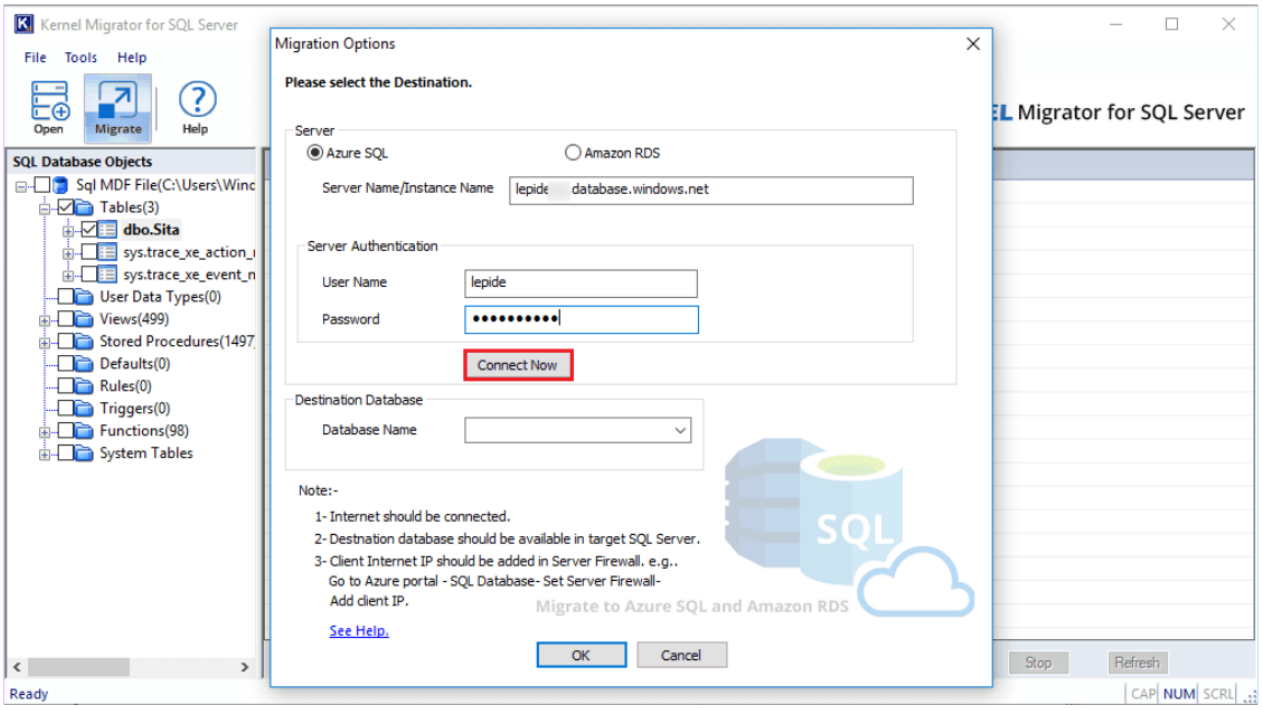
Step 3: After browsing the database, the second step requires selecting the scan mode and clicking the Recover button.



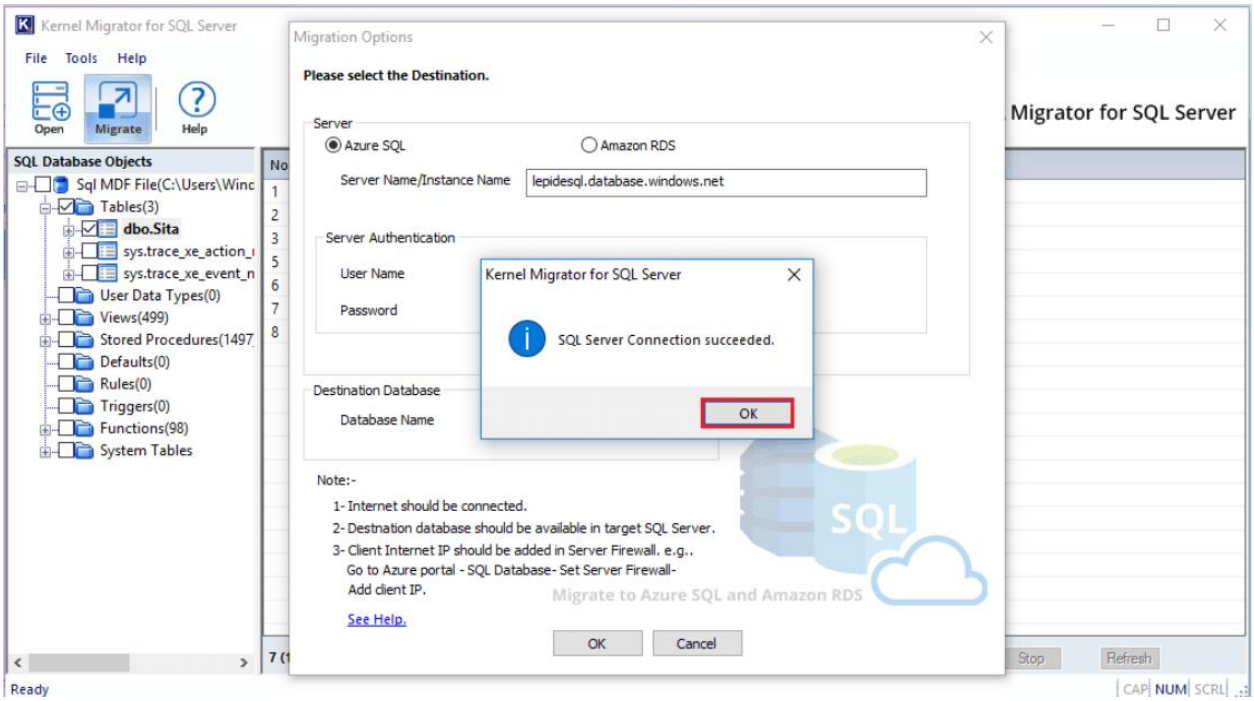
Step4: After retrieving the database, the database objects are displayed in a tree structure. Here, you can get a clear preview of the objects. Select the required objects and click the Migrate button.



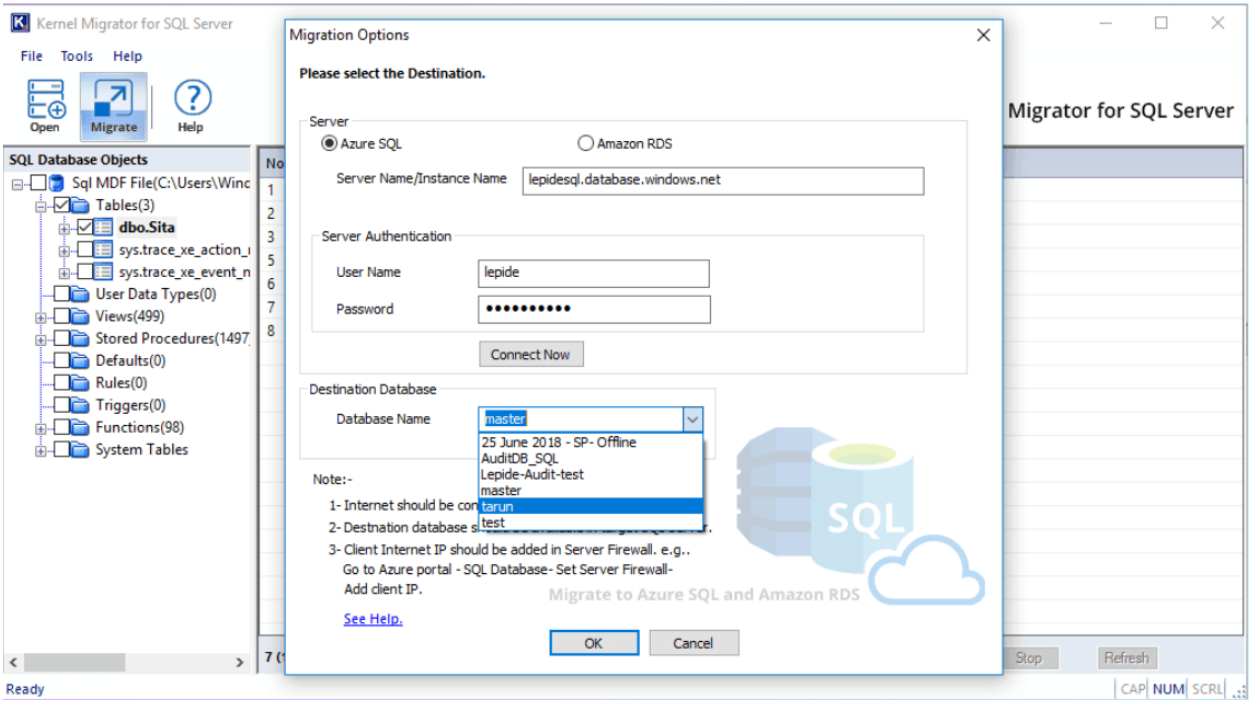
Step 5: Select the first option of Azure SQL. Then input the server name and its complete credentials. Then click the Connect Now button



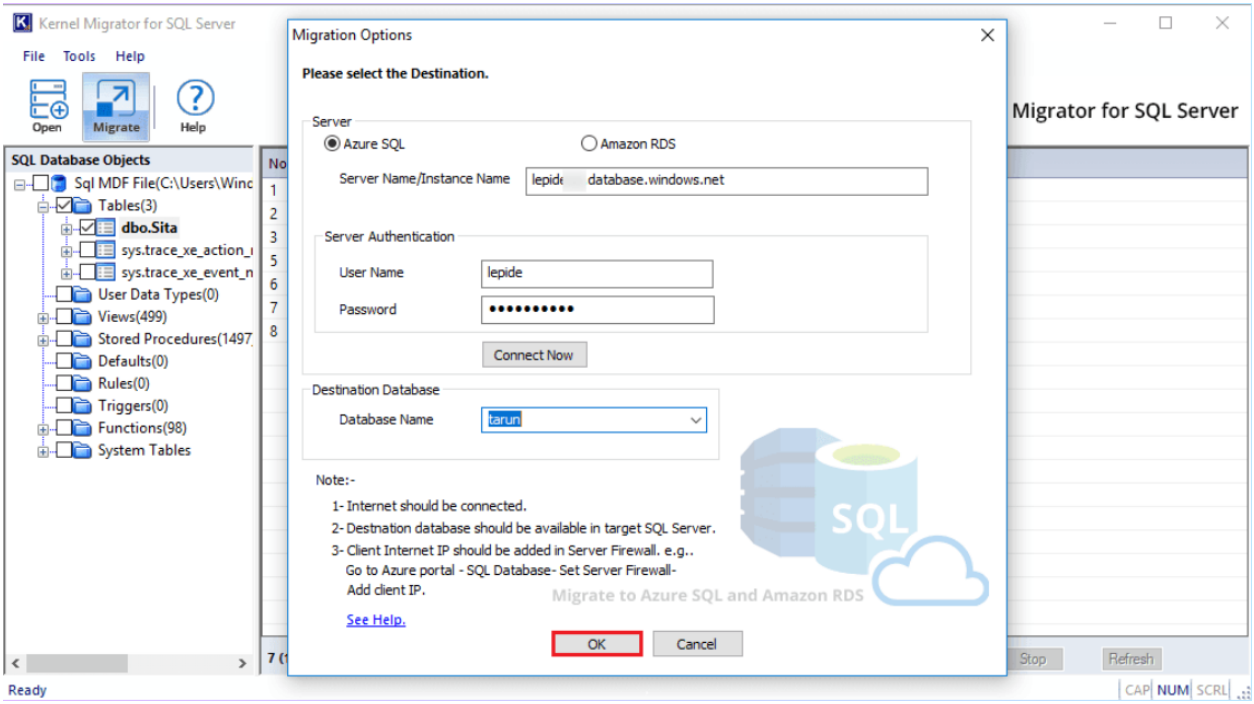
Step 6: After a successful connection, the tool will provide a successful message. Click OK



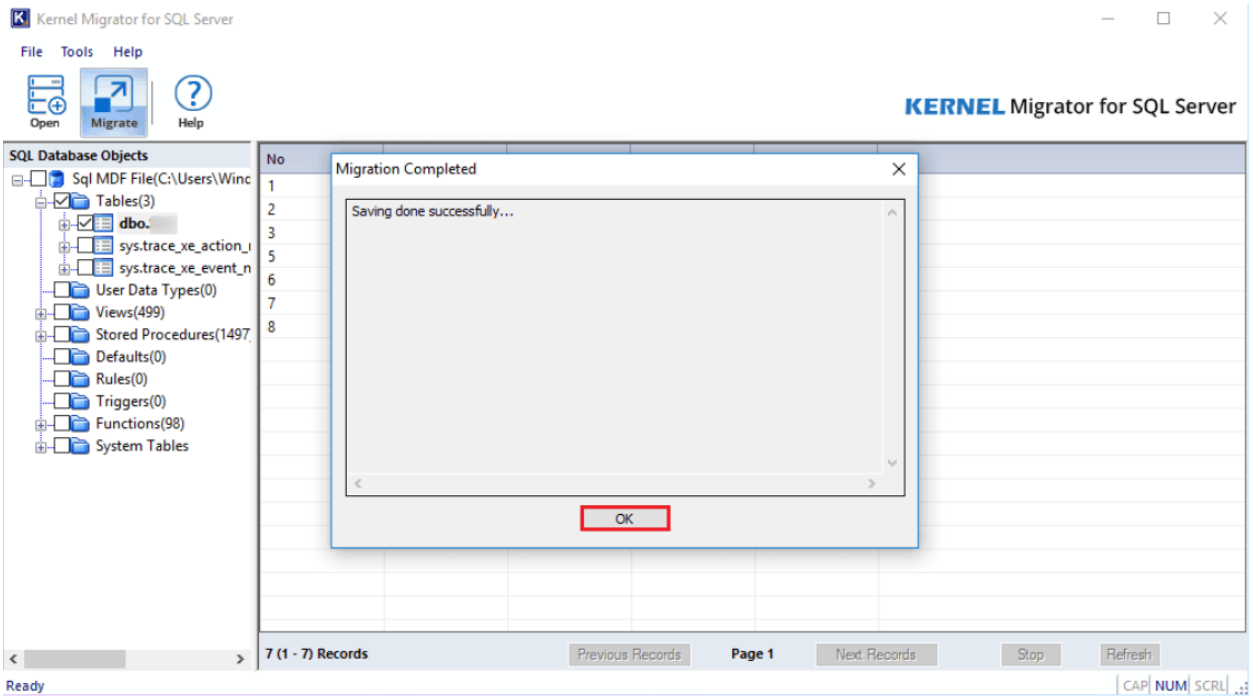
Step 7: Now, select the destination database from the drop-down list



Step 8: After selecting the database, click the **OK** button



Step 9: The tool has successfully migrated the database, and it gives a successful message. Click OK



Evaluation by faculty	
Criteria	Marks
	/20
	/25
	/20
	/10
Total	/75
Faculty Signature with Date	

RESULT: