# Project Documentation: Migrating Microsoft Access Database to SQL Server

## Overview

This project involves migrating a Microsoft Access database to SQL Server using SQL Server Migration Assistant (SSMA). The goal is to transition the backend data to SQL Server for enhanced performance, scalability, and security while optionally retaining Access as the front-end interface.

## Steps and Process

### 1. Preparation

- Backup Data: Ensure the Microsoft Access database is backed up.  
- System Requirements:  
 - A system with SQL Server installed (e.g., SQL Server 2019 Standard).  
 - SQL Server Migration Assistant (SSMA) for Access installed on the system.  
 - Administrative privileges on both the Access database and the SQL Server instance.

### 2. Setting Up the Environment

- SQL Server Configuration:  
 - Enabled TCP/IP protocol using the SQL Server Configuration Manager.  
 - Opened port 1433 in the firewall for remote access.  
 - Created a database in SQL Server to hold the migrated data.

### 3. Migration Using SSMA

#### Step 1: Install SQL Server Migration Assistant (SSMA)

Download and install SSMA for Access from the official Microsoft website.

#### Step 2: Launch SSMA

- Open SSMA and create a new project.  
- Configure the project for the appropriate version of SQL Server.

#### Step 3: Connect to the Access Database

- Add the Access database file (.accdb or .mdb) to SSMA.  
- Verify the connection by providing the necessary credentials if the database is password-protected.

#### Step 4: Connect to SQL Server

- Enter the SQL Server connection details:  
 - Server Name: Enter the IP address or server name (e.g., 192.168.100.100).  
 - Authentication: Use SQL Server Authentication with valid credentials.  
 - Database: Manually specify or select the target database for migration.

#### Step 5: Convert Schema

- Convert the Access database schema to SQL Server by:  
 - Right-clicking the Access objects and selecting 'Convert Schema.'  
 - Reviewing and resolving any compatibility issues in the Output pane.

#### Step 6: Migrate Data

- Select the converted database objects and initiate the data migration.  
- Monitor the migration progress in SSMA and ensure all data is successfully transferred.

### 4. Post-Migration Steps

#### Validate Data

- Use SQL Server Management Studio (SSMS) to verify the migrated tables and data.  
- Run queries to confirm data integrity (e.g., row counts, sample data checks).

#### Update Access Front-End (Optional)

- If using Access as the front-end, update linked tables:  
 - Go to External Data > Linked Table Manager in Access.  
 - Relink the tables to the SQL Server database using an ODBC connection.

#### Optimize Performance

- Add indexes or optimize queries in SQL Server for better performance.  
- Monitor performance using SQL Server Profiler or Dynamic Management Views (DMVs).

## Challenges and Resolutions

1. Issue: SSL certificate error during connection to SQL Server.  
 Resolution: Enabled 'Trust Server Certificate' in SSMA connection settings.  
  
2. Issue: Blank database dropdown in SSMA.  
 Resolution: Manually entered the database name during SQL Server connection setup.  
  
3. Issue: Access front-end linked table manager was grayed out.  
 Resolution: Ensured the database was writable and in a trusted location, and added an ODBC connection manually.

## Outcomes

- Successful migration of Microsoft Access database to SQL Server.  
- All tables, data, and relationships migrated and validated in SQL Server.  
- Access front-end optionally linked to SQL Server for continued use.

## Next Steps

1. Train end-users on the new setup.  
2. Plan for a full transition to SQL Server if needed, including rebuilding the front-end.  
3. Monitor database performance and optimize as necessary.

## Tools and Resources Used

- Microsoft Access  
- SQL Server 2019  
- SQL Server Management Studio (SSMS)  
- SQL Server Migration Assistant (SSMA)

## Project Notes

- Document all connection settings and configurations for future reference.  
- Schedule regular database backups on SQL Server to prevent data loss.