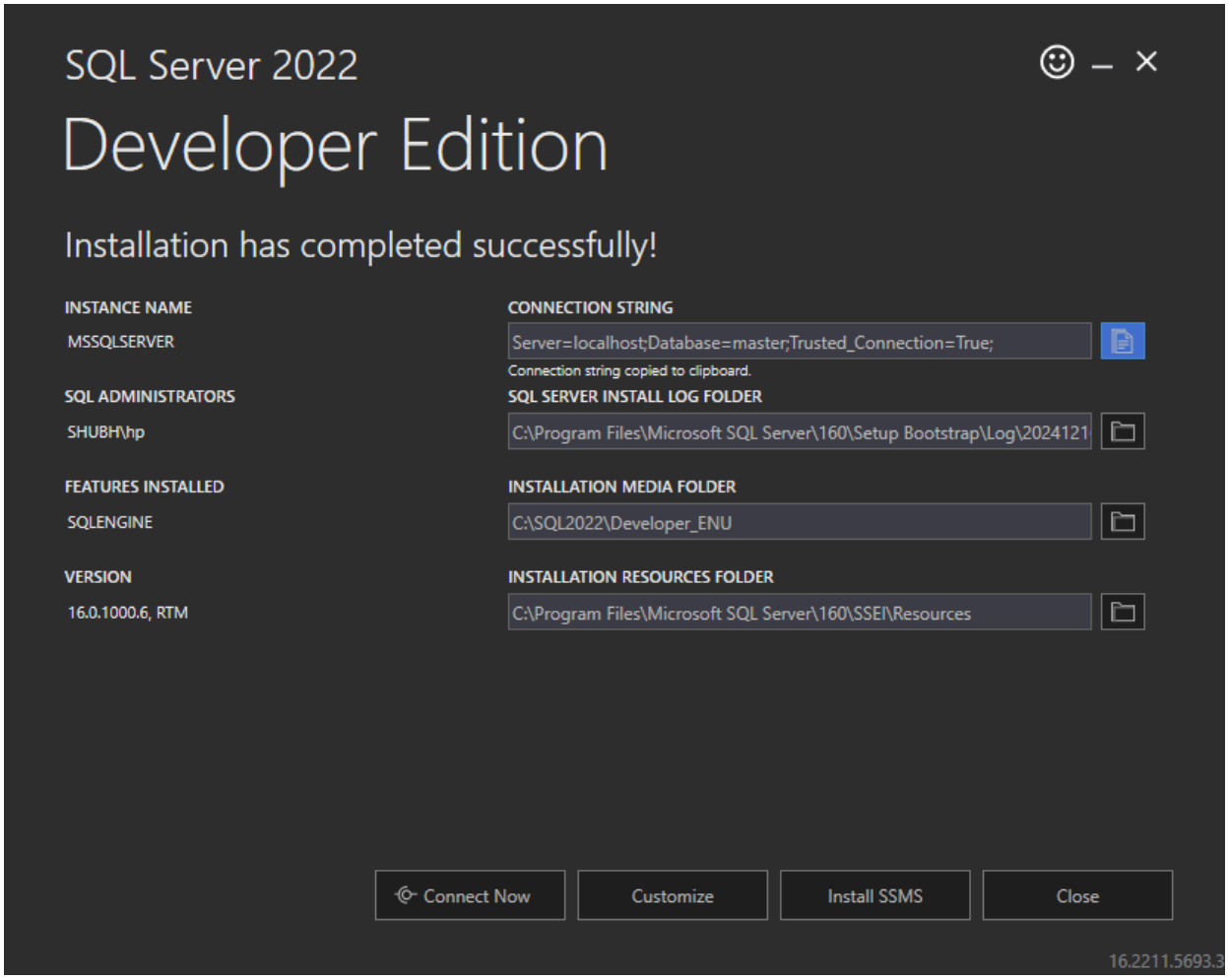
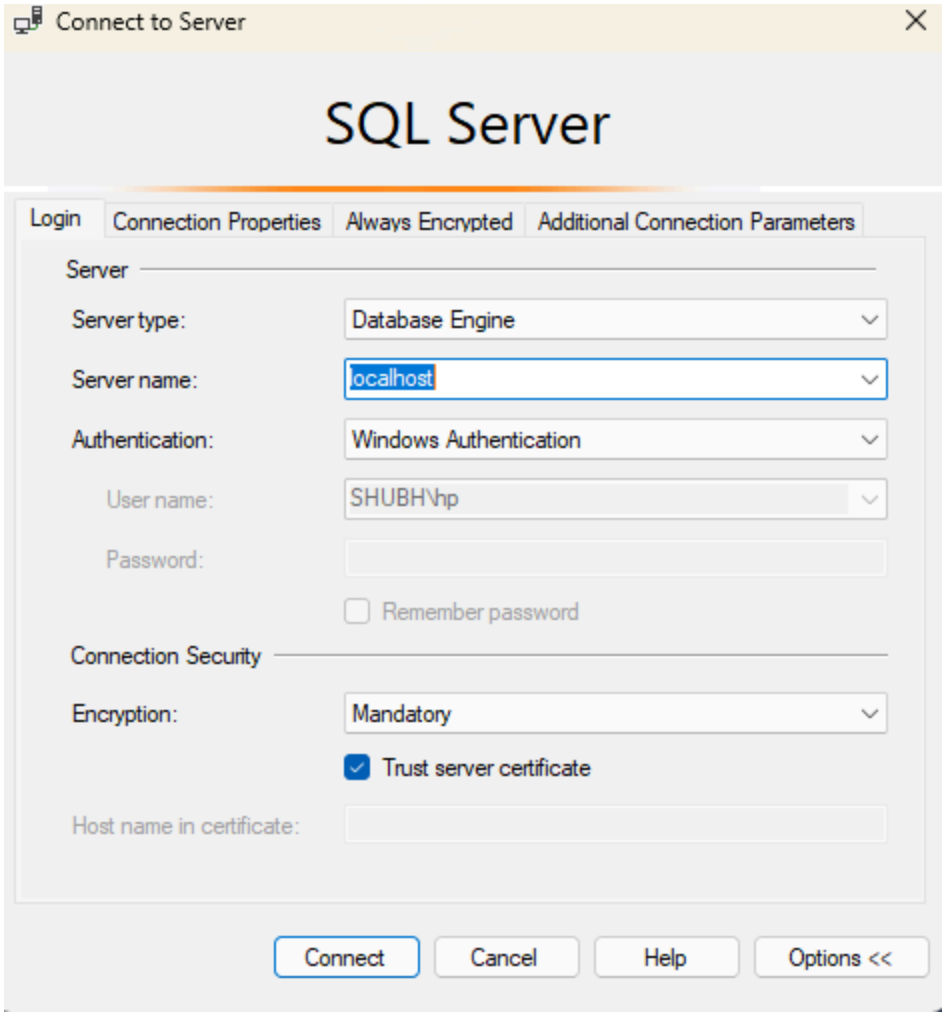
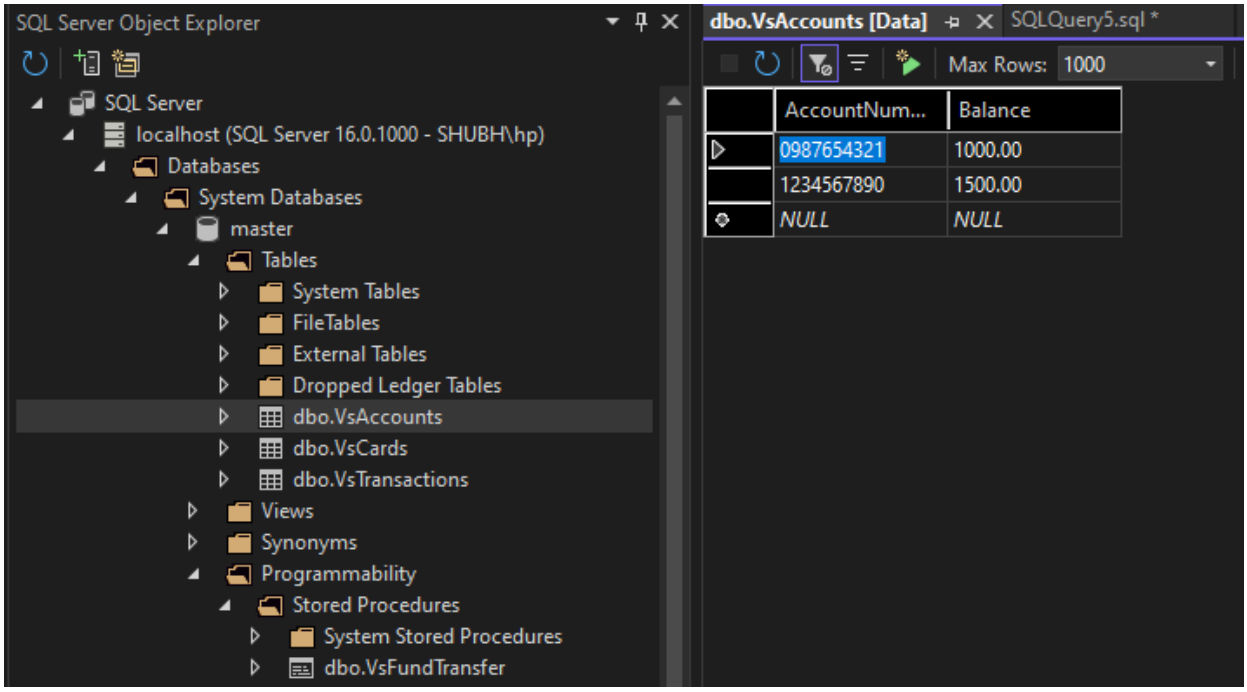
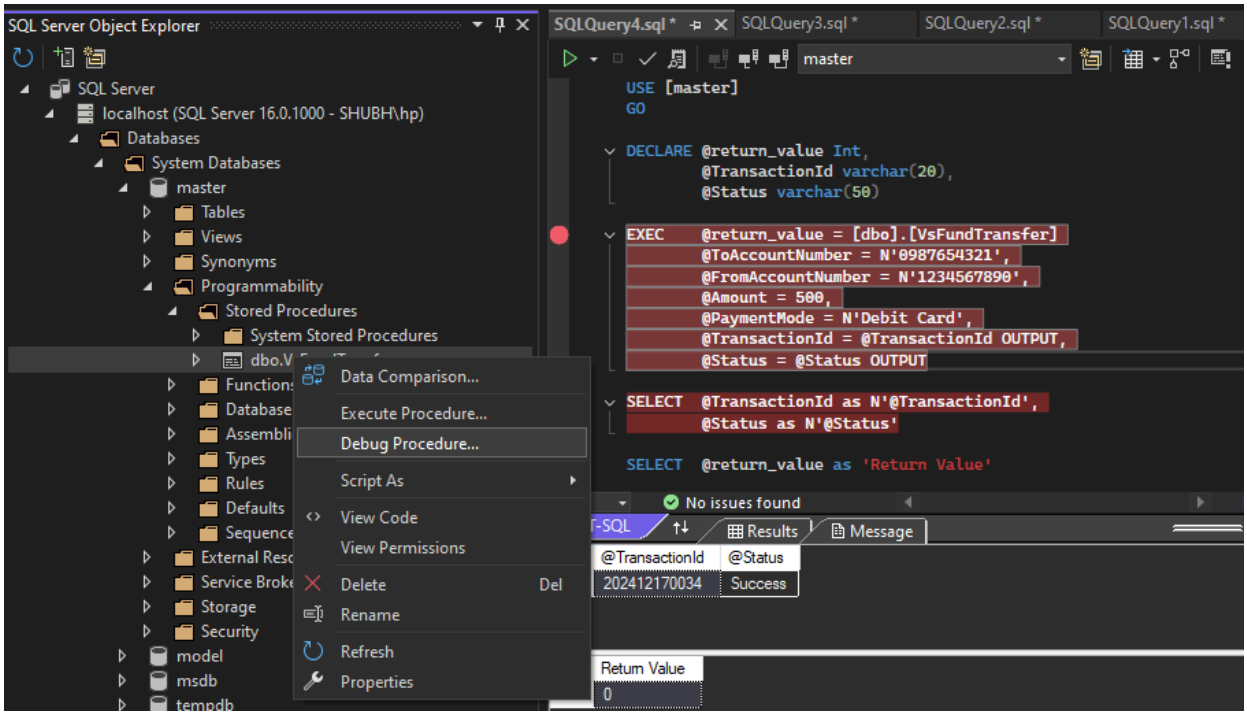
SQL server 2022 download

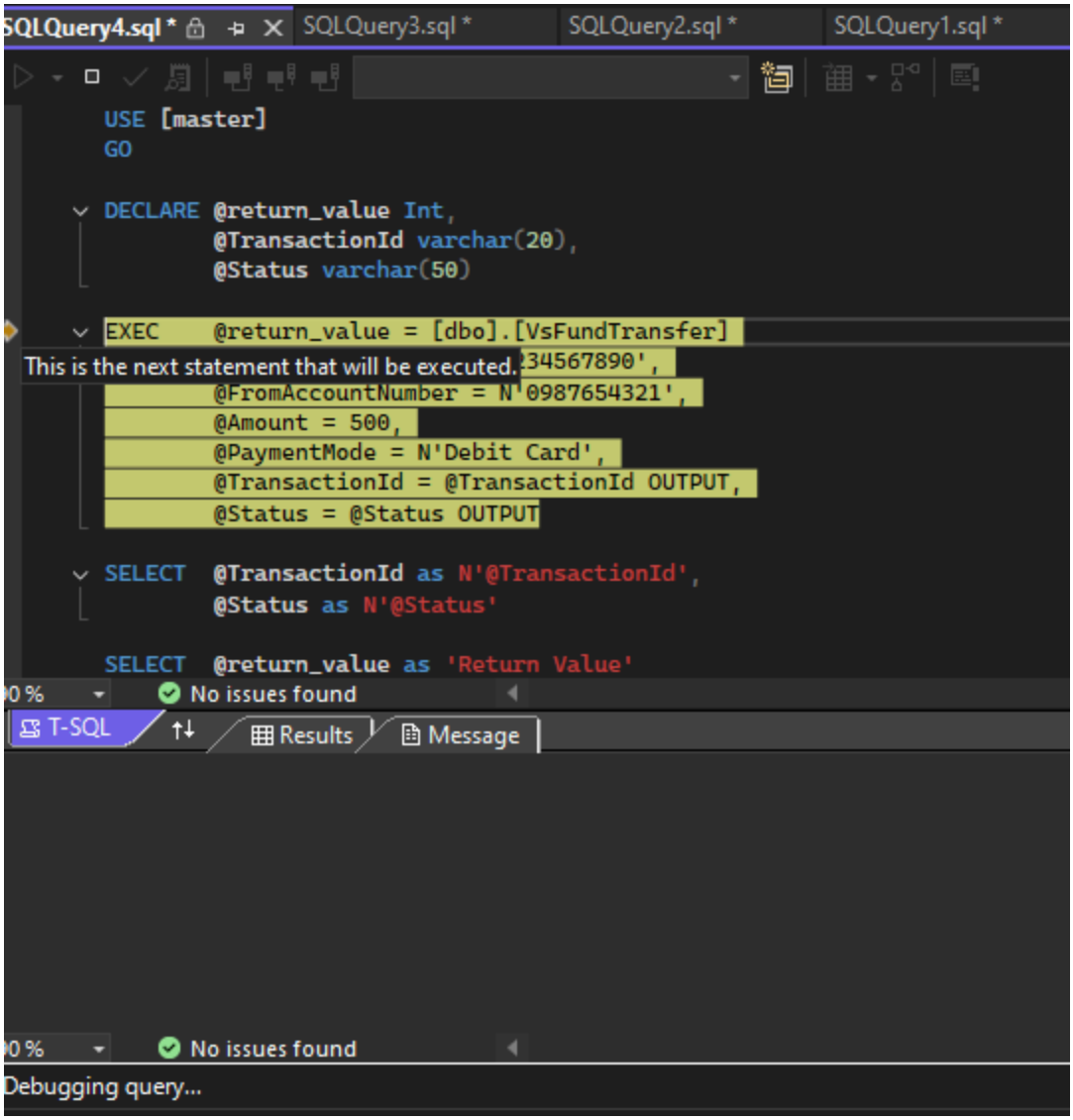


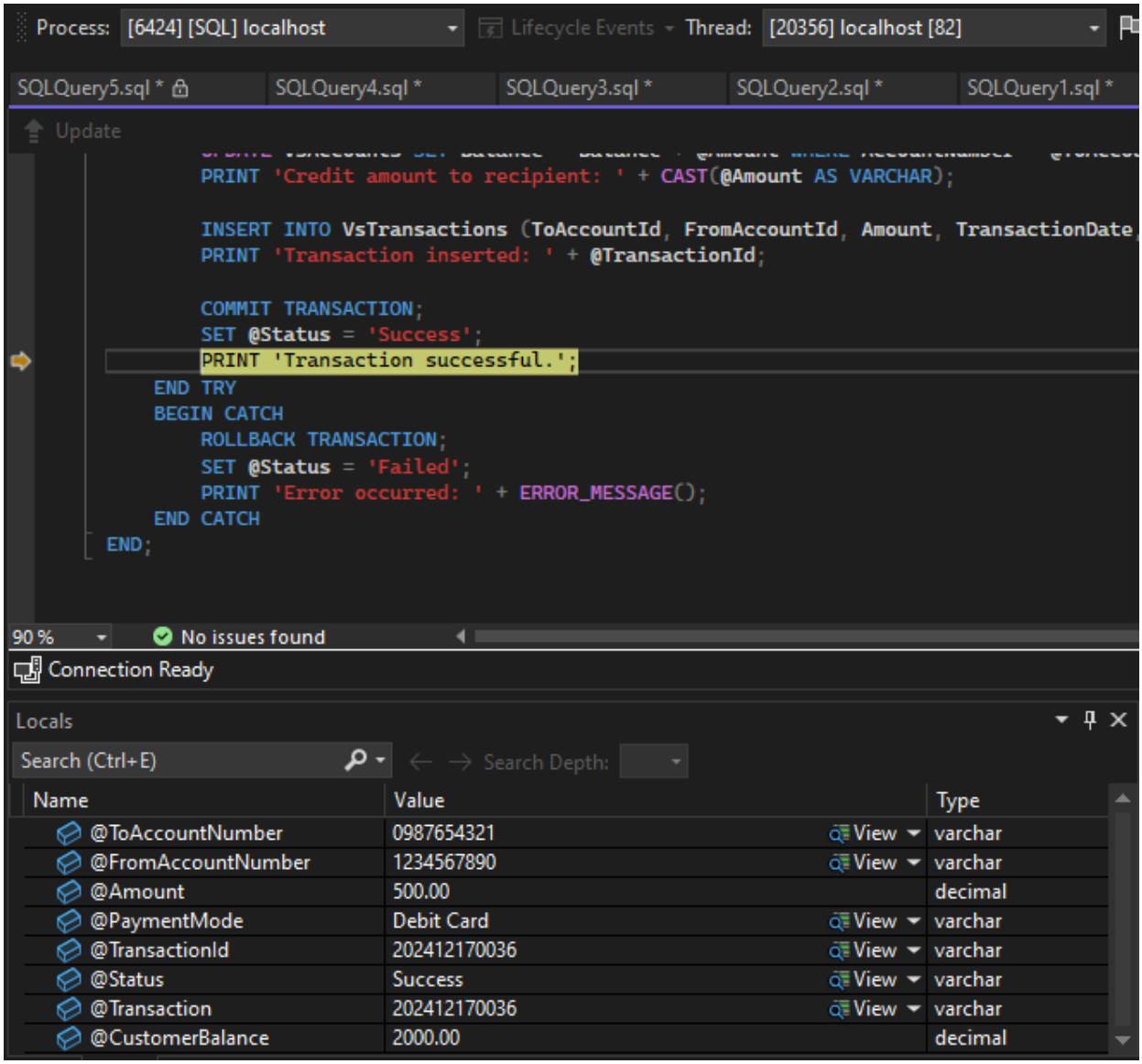
SQL Server management studio


Connect to Server
SQL Server
Login
Connection Properties
Always Encrypted
Additional Connection Parameters
Server
Server type:
Database Engine
Server name:
localhost
Authentication:
Windows Authentication
User name:
SHUBH\hp
Password:
Remember password
Connection Security
Encryption:
Mandatory
Trust server certificate
Host name in certificate:
Connect
Cancel
Help
Options <<


SQL Server Object Explorer
SQL Server
localhost (SQL Server 16.0.1000 - SHUBH\hp)
Databases
System Databases
master
Tables
System Tables
FileTables
External Tables
Dropped Ledger Tables
dbo.VsAccounts
dbo.VsCards
dbo.VsTransactions
Views
Synonyms
Programmability
Stored Procedures
System Stored Procedures
dbo.VsFundTransfer
dbo.VsAccounts [Data]
SQLQuery5.sql *
Max Rows: 1000
AccountNum...
Balance
0987654321
1000.00
1234567890
1500.00
NULL
NULL


SQL Server Object Explorer
SQL Server
localhost (SQL Server 16.0.1000 - SHUBH\hp)
Databases
System Databases
master
Tables
Views
Synonyms
Programmability
Stored Procedures
System Stored Procedures
Functions
Database
Assembli
Types
Rules
Defaults
Sequence
External Resc
Service Broke
Storage
Security
model
msdb
tempdb
Data Comparison...
Execute Procedure...
Debug Procedure...
Script As
View Code
View Permissions
Delete Del
Rename
Refresh
Properties
SQLQuery4.sql *
SQLQuery3.sql *
SQLQuery2.sql *
SQLQuery1.sql *
master
USE [master]
GO
DECLARE @return_value Int,
@TransactionId varchar(20),
@Status varchar(50)
EXEC @return_value = [dbo].[VsFundTransfer]
@ToAccountNumber = N'0987654321',
@FromAccountNumber = N'1234567890',
@Amount = 500,
@PaymentMode = N'Debit Card',
@TransactionId = @TransactionId OUTPUT,
@Status = @Status OUTPUT
SELECT @TransactionId as N'@TransactionId',
@Status as N'@Status'
SELECT @return_value as 'Return Value'
No issues found
Results
Message
@TransactionId
@Status
202412170034
Success
Return Value
0


SQLQuery4.sql *
SQLQuery3.sql *
SQLQuery2.sql *
SQLQuery1.sql *
USE [master]
GO
DECLARE @return_value Int,
@TransactionId varchar(20),
@Status varchar(50)
EXEC @return_value = [dbo].[VsFundTransfer]
This is the next statement that will be executed.
@FromAccountNumber = N'0987654321',
@Amount = 500,
@PaymentMode = N'Debit Card',
@TransactionId = @TransactionId OUTPUT,
@Status = @Status OUTPUT
SELECT @TransactionId as N'@TransactionId',
@Status as N'@Status'
SELECT @return_value as 'Return Value'
No issues found
T-SQL
Results
Message
No issues found
Debugging query...



Steps

1. Enable SQL Server Debugging in Visual Studio

In order to debug SQL Server stored procedures from Visual Studio, there are certain settings and permissions you need to verify on both Visual Studio and the SQL Server instance.

2. Verify SQL Server Instance Settings

SQL Server must have certain configurations enabled for debugging, including remote debugging if your SQL Server instance is on a different machine.

Steps to Check and Enable Debugging:

Step 1: Enable SQLCLR Debugging

Open SQL Server Management Studio (SSMS).

Run the following query to check whether SQLCLR debugging is enabled:

```
EXEC sp_configure 'clr enabled';
```

- If the `run_value` is `0`, SQLCLR is disabled.
- To enable it, run:

```
EXEC sp_configure 'show advanced options', 1;
RECONFIGURE;
EXEC sp_configure 'clr enabled', 1;
RECONFIGURE;
```

Step 2: Allow SQL Debugging in Visual Studio

1. In **Visual Studio**, right-click your project and select **Properties**.
2. In the project properties, go to the **Debug** tab.
3. Check the box for **Enable SQL Server debugging** if available.

Step 3: Ensure Permissions for SQL Debugging

1. **Check your SQL Server user permissions**:
   - The account that connects to SQL Server needs to be part of the `sysadmin` fixed server role or have `debugger` permissions.

     You can verify by running this query in SSMS:

     ```
     SELECT IS_SRVROLEMEMBER('sysadmin');
     ```

2. If the result is `0`, your account is not a sysadmin, and you will need additional privileges to debug.

Step 4: Attach the Debugger to SQL Server

If everything is set up properly, you can try attaching the debugger to `sqlservr.exe` in Visual Studio by following these steps:

1. Go to **Debug** > **Attach to Process**.
2. Look for `sqlservr.exe` in the list of processes and attach to it.
3. Make sure to select **Managed (SQL Server)** as the code type to attach.