Response 1

The error message you're seeing:

**"Access to the remote server is denied because no login-mapping exists"**

...when using a query like:

SELECT \* FROM OPENROWSET(...)

typically means **SQL Server is trying to access a linked server or remote data source**, but there is **no login mapping configured** for the current user.

**Root Cause:**

This usually happens when:

* You're using OPENROWSET with a **linked server** or remote data source that requires **explicit credentials**.
* SQL Server doesn't know **which credentials to use** to authenticate to the remote server.
* Either the **linked server has not been set up properly**, or the **login mappings for the current SQL Server login are missing**.

**Common Scenarios:**

1. **Linked Server with No Login Mapping:**
   * You might have added a linked server in SQL Server Management Studio (SSMS), but didn't configure security mappings under the *"Security"* tab (e.g., no mapping for your SQL login to a remote login/password).
2. **Using Windows Authentication:**
   * If you’re using Windows Authentication, and the remote server doesn’t allow delegation or doesn’t recognize your Windows credentials, login fails.
3. **Ad-hoc OPENROWSET Usage without Proper Authentication:**
   * If Ad Hoc Distributed Queries are disabled, or if OPENROWSET is trying to connect to a remote source without credentials, it will fail.

**Fixes:**

**Option 1: Use Explicit Credentials in OPENROWSET**

Instead of relying on linked server mappings, specify credentials directly:

ESELECT \*

FROM OPENROWSET(

'SQLNCLI',

'Server=RemoteServerName;UID=your\_username;PWD=your\_password;',

'SELECT \* FROM RemoteDatabase.dbo.TableName')

**Option 2: Configure Linked Server Security**

If using a linked server:

1. In SSMS: go to **Server Objects > Linked Servers > [YourServer] > Properties > Security**.
2. Add a login mapping:
   * For a specific SQL login, map to a remote login/password.
   * Or select "Be made using this security context" and provide default credentials.

**Option 3: Enable Ad Hoc Distributed Queries**

If using OPENROWSET without a linked server, you might need to enable the feature:

sql

CopyEdit

EXEC sp\_configure 'show advanced options', 1;

RECONFIGURE;

EXEC sp\_configure 'Ad Hoc Distributed Queries', 1;

RECONFIGURE;

⚠️ Use with caution. Only enable Ad Hoc Distributed Queries if you understand the security implications.

⚠️ **Note:** Storing credentials in plain text inside a query is a security risk. Prefer using linked servers with proper login mappings when possible.

**✅ Example Using SQLOLEDB:**

sql

CopyEdit

SELECT \*

FROM OPENROWSET('SQLOLEDB',

'Server=RemoteServerName;UID=YourUsername;PWD=YourPassword;',

'SELECT \* FROM RemoteDatabase.dbo.YourTable');

Replace:

* RemoteServerName – with the target SQL Server instance name
* YourUsername / YourPassword – with appropriate credentials
* RemoteDatabase.dbo.YourTable – with the target table

**🔍 Difference Between SQLOLEDB and SQLNCLI:**

| **Feature** | **SQLOLEDB** | **SQLNCLI (SQL Native Client)** |
| --- | --- | --- |
| Provider Type | Older OLE DB provider | Newer OLE DB provider |
| Deprecated? | Yes (deprecated by Microsoft) | Yes, replaced by **MSOLEDBSQL** |
| Supports SQL Server 2012+ | Limited support | Better for SQL Server 2005 – 2012 |
| Advanced Features | Lacks modern features | Supports MARS, XML, newer data types |
| TLS 1.2 Support | No | Partial (depending on version) |
| Current Recommendation | ❌ Don't use for new development | ❌ Also deprecated; use **MSOLEDBSQL** |

**✅ Preferred Option in 2025:**

Use **MSOLEDBSQL** (Microsoft OLE DB Driver for SQL Server), which is the **actively supported** and recommended driver:

sql

CopyEdit

SELECT \*

FROM OPENROWSET('MSOLEDBSQL',

'Server=RemoteServerName;UID=YourUsername;PWD=YourPassword;',

'SELECT \* FROM RemoteDatabase.dbo.YourTable');

Would you like to test or troubleshoot this on a specific SQL Server version?

4o

**You said:**

is trusted connection=yes is needed?

**ChatGPT said:**

Trusted\_Connection=Yes (or Integrated Security=SSPI) is **used to specify that Windows Authentication** should be used instead of SQL Server Authentication.

**✅ When to Use Trusted\_Connection=Yes:**

Use it **only** if:

* You are **not** providing a UID and PWD, and
* You want to use the **Windows credentials** of the SQL Server process (or the caller, depending on context).

**Example with Trusted\_Connection=Yes:**

sql

CopyEdit

SELECT \*

FROM OPENROWSET('SQLOLEDB',

'Server=RemoteServerName;Trusted\_Connection=Yes;',

'SELECT \* FROM RemoteDatabase.dbo.YourTable');

In this case, SQL Server will try to connect to the remote server **using the Windows credentials of the current login/session**.

**⚠️ If You're Using UID and PWD:**

**Do not** include Trusted\_Connection=Yes, because it conflicts with SQL Authentication.

For example, this is **invalid**:

sql

CopyEdit

-- ❌ Don't do this

'Server=RemoteServerName;UID=User;PWD=Password;Trusted\_Connection=Yes;'

Here are **examples of OPENDATASOURCE and OPENQUERY** with **explicit remote connection details**, assuming you want to **query a remote SQL Server directly** (without relying on a linked server’s preconfigured login mappings).

**✅ 1. OPENDATASOURCE Example (Ad Hoc Query with Connection String)**

sql

CopyEdit

SELECT \*

FROM OPENDATASOURCE('SQLOLEDB',

'Data Source=RemoteServerName;

User ID=YourUsername;

Password=YourPassword;').RemoteDatabase.dbo.YourTable;

* SQLOLEDB can be replaced with MSOLEDBSQL or SQLNCLI depending on the driver you have.
* This connects and queries directly without using a linked server.

**✅ 2. OPENQUERY Example (Requires Linked Server)**

OPENQUERY **does not allow direct credentials**, so it **requires a preconfigured linked server** with login mappings.

sql

CopyEdit

SELECT \*

FROM OPENQUERY([YourLinkedServerName],

'SELECT \* FROM RemoteDatabase.dbo.YourTable');

* To use this, you must have already added the linked server via:

sql

CopyEdit

EXEC sp\_addlinkedserver

@server = 'YourLinkedServerName',

@srvproduct = '',

@provider = 'SQLOLEDB',

@datasrc = 'RemoteServerName';

EXEC sp\_addlinkedsrvlogin

@rmtsrvname = 'YourLinkedServerName',

@useself = 'false',

@rmtuser = 'YourUsername',

@rmtpassword = 'YourPassword';

**⚠️ Notes:**

* OPENDATASOURCE is more flexible for **ad hoc one-off queries**.
* OPENQUERY is better for **predefined secure access** via linked servers.
* Make sure **Ad Hoc Distributed Queries** is enabled for OPENDATASOURCE and OPENROWSET to work:

sql

CopyEdit

sp\_configure 'show advanced options', 1;

RECONFIGURE;

sp\_configure 'Ad Hoc Distributed Queries', 1;

RECONFIGURE;