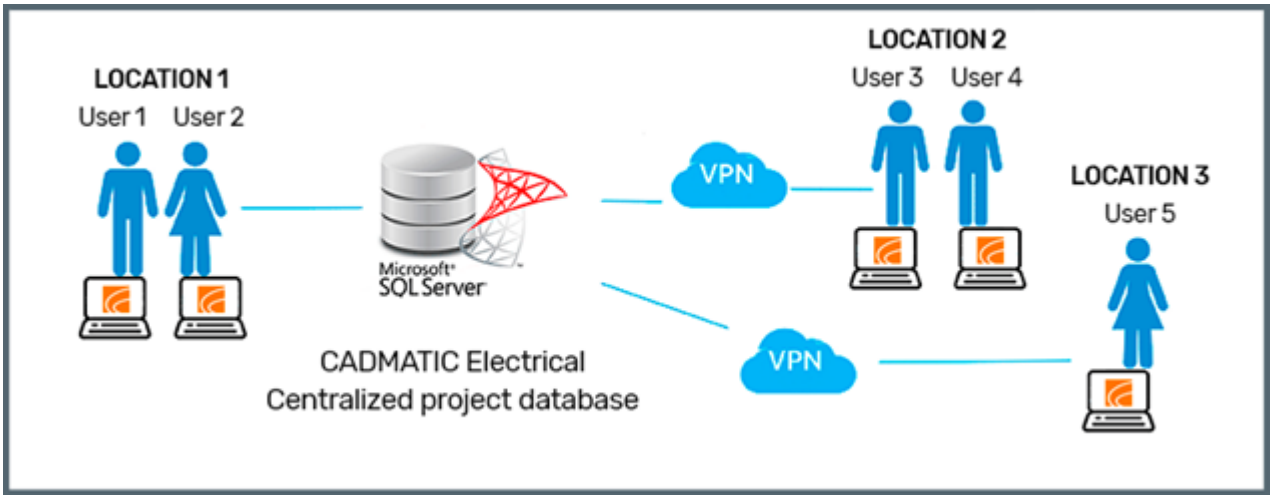
CADMATIC Electrical

Project databases on SQL Server

Contents

1. Overview

You can save the data from the selected projects on a database that is located on an SQL Server which is suitable for multi-user projects and remote access (VPN).



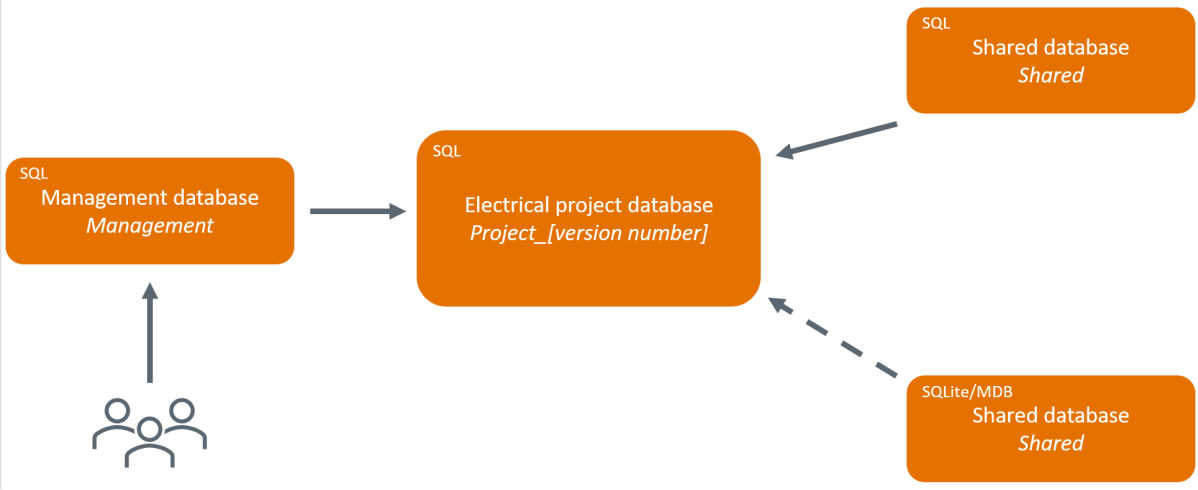
In a traditional project structure, drawings and other project files, such as the project database *EDBProject.sqlite* or *EDBProject.mdb*, are all located in the same directory on the file server. With SQL databases, you save the database information on the SQL Server, which can be in a different location on the network. Drawings and other project files are still stored on a shared drive on the file server.

While the *EDBProject.sqlite*/*EDBProject.mdb* file is project-specific in the traditional project structure, an SQL Server database usually contains data for various different projects. On SQL Server, all project data are in the same tables and they are separated with their own project number. For instance, all devices in the project are in a table called **Devices**, and you can fetch information from the table using the project number. The project number is internal to Electrical and cannot be changed.

Check the system requirements before installation. Also note that taking SQL Server into use requires understanding of database server configuration and the creation of databases, tables and users. Furthermore, the company must adapt their work practices to suit the server environment. For this reason, taking SQL Server into use should be done in cooperation with Cadmatic and the company's own ICT support.

2. Database types

With SQL, there are three different types of databases involved: a management database, a project database and shared databases. You can name these databases any way you want – in these instructions, the databases are named **Management**, **Project** and **Shared**.



- The management database (in these instructions **Management**) includes a list of different project databases, shared databases, and users. The user first connects to the management database which then routes the user to the correct project data. This allows you to have project data in various different databases without having them in the same location.
  You can divide project databases according to your needs, such as for each regional office, year, or software version. However, you should avoid dividing the databases too small since you have to create new databases in MS SQL Server Management Studio each time.
  Create empty databases in MS SQL Server Management Studio and add tables to the databases for project data with the management tool in Electrical. The program creates the *EDBManagement.econ* connection file which you must share to every user.
- The project database (in these instructions **Project**) includes the project data (devices, cables, wires, etc.). Drawing files will be stored to the project folder just like in SQLite/MDB projects. The project database can be in the same or different location as the management database. Usually, you use one database to save data of many projects due to the SQL Server structure. You cannot create new databases in Electrical, only in a separate program, such as MS SQL Server Management Studio.

If you change the project database structure in Electrical and update it, the structure changes in each of the projects that you have saved in the database.

- Shared databases (in these instructions **Shared**) contain design data that is shared within the company, such as cable types, product models and plate definitions. This design data is used in actual projects.

  You can keep shared databases as an SQLite/MDB database or move them to an SQL Server database. You can check the location of the SQLite/MDB database in the Electrical settings and the location of the SQL database in the project information in the management database for each project.

  If you move all projects to an SQL Server database, move shared databases here as well. If you have some projects in SQL format and some in SQLite/MDB format, use the SQLite/MDB database.

3. User roles

The following roles define what users are allowed to do with SQL Server and its projects. Each role in the list automatically includes the user rights of all the roles below them.

- **db_owner** – The user can create database tables and update the database structure when upgrading to a new version. This role needs to be defined for the main user.
- **EDBAdmin** – The user can create new databases, delete databases and edit database settings in SQL Server project management.
- **EDBPowerUser** – The user can create and delete SQL Server projects.
- **EDBUser** – The user can edit projects.
- **EDBReader** – The user only has read-only rights to projects.

4. Install and implement SQL Server databases

These instructions are about installing and implementing Microsoft SQL Server databases in CADMATIC Electrical. You might need to adapt the instructions to the existing IT infrastructure. Also note that firewall ports might need to be opened. Furthermore, creating admin and user groups on the AD server make managing permissions easier.

Check the system requirements before starting.

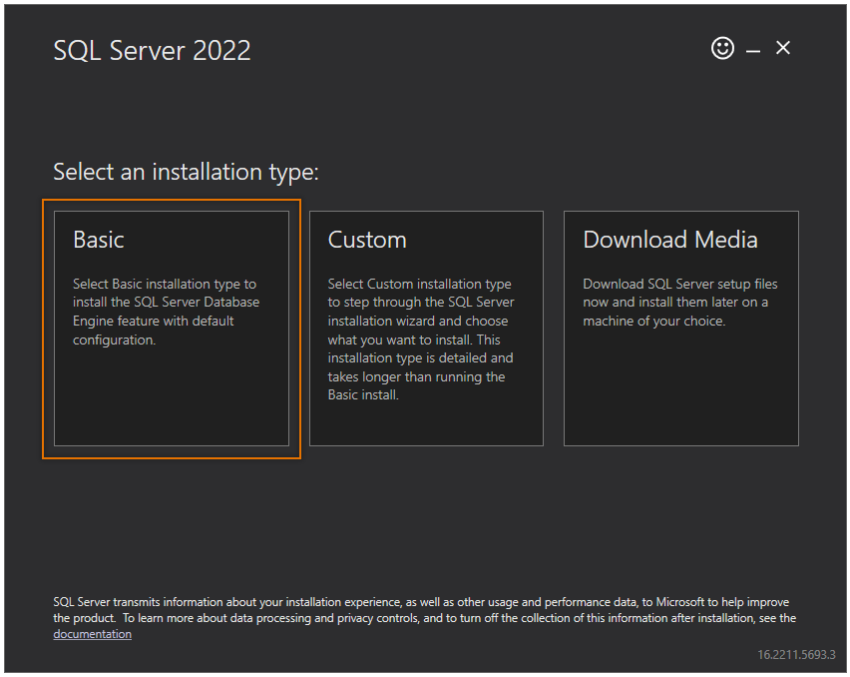
4.1. Install the server

In addition to SQL Server as the data repository, SQL Server Management Studio is needed for creating the databases and managing users and permissions. You can install it after you have installed the SQL Server program on the Windows Server.

Check the required SQL Server version from the system requirements. In this example, SQL Server 2022 is used.

Do the following:

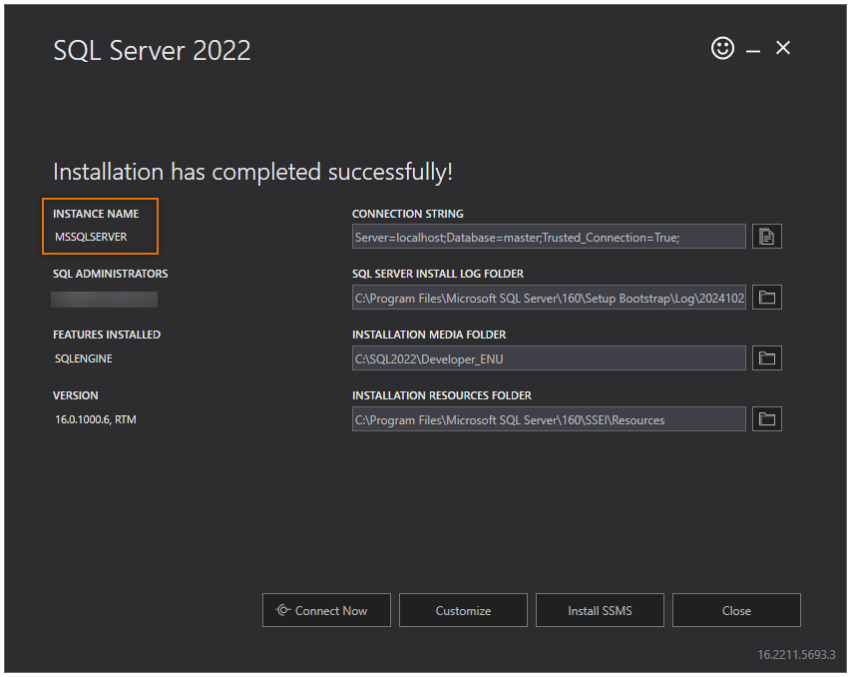
1. Double-click the installation file you downloaded.
2. Select **Basic** as the installation type.



3. Follow the instructions in the installation dialog.

   The program is installed on your computer, after which a dialog opens showing you information about the installation.

4. Save the instance name, for example, by taking a screenshot of the dialog. In this example, the instance name is **MSSQLSERVER**.



5. Install the SQL Server Management Studio by clicking **Install SSMS**. A web page opens.
6. Click the **Download SQL Server Management Studio (SSMS) [version number]** link.
7. Double-click the installation file, and install SQL Server Management Studio by following the instructions in the installation dialog.
8. Close the installation programs.

Next, configure the server.

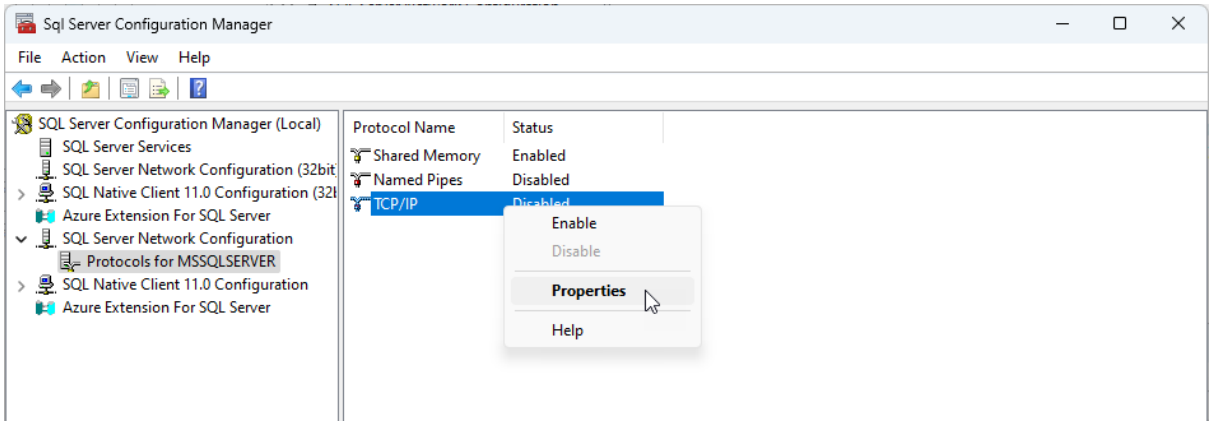
4.2. Configure the server

The port configured for the SQL Server instance affects the company's firewall settings and how you connect to the SQL Server database. The port is either static or dynamic:

- **Static port:**
  - If you configure the default port 1433 or some other static port, you only need to allow inbound traffic via that port through the Windows firewall.
  - If you configure the default port 1433, you connect to the database with the name of the server only, e.g. *SERVER01*.
  - If you configure a static port other than 1433, you connect to the database with the server name and port in the format *[server name],[port]*, e.g. *SERVER01,1910*.
- **Dynamic port (0 or no TCP port defined):**
  - The port changes with every startup so you need to allow inbound traffic via the whole range of dynamic ports through the Windows firewall.
  - You need to open the UDP port 1434.
  - SQL Server Browser needs to be running on the server.
  - You connect to the database with the server name and the instance name in the format *[server name]\[instance]*, e.g. *SERVER01\MSSQLSERVER*.

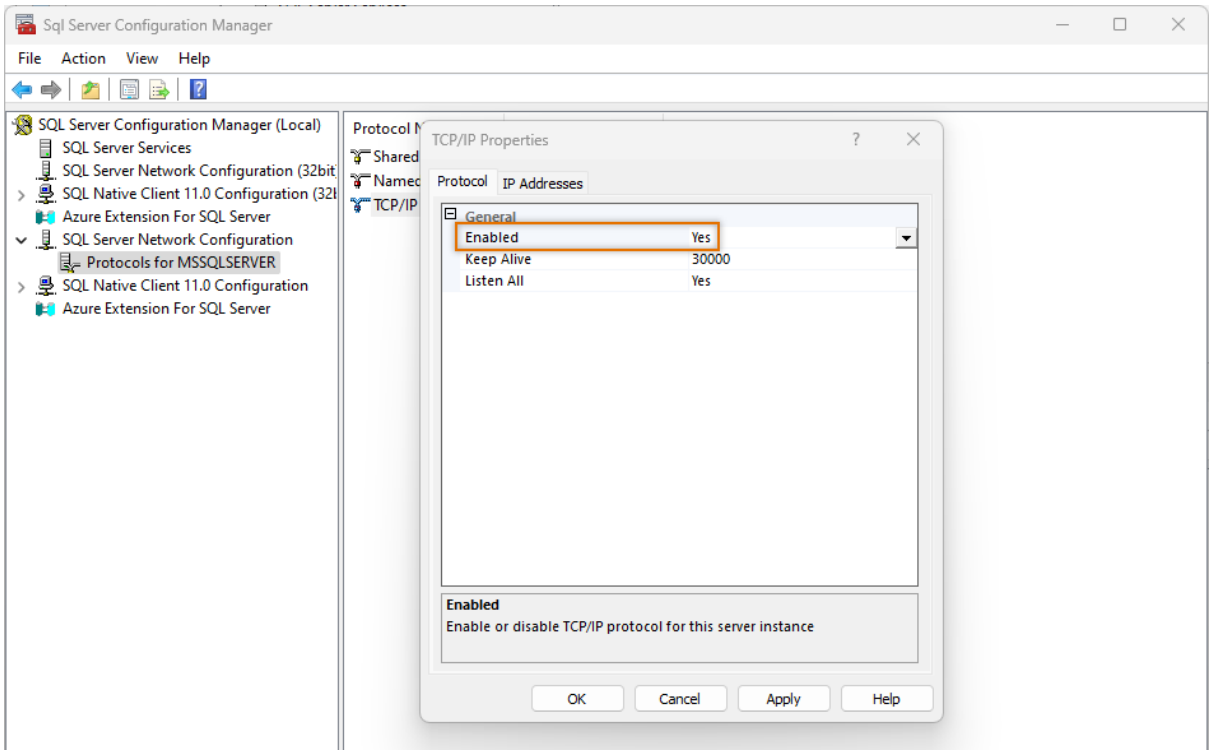
Do the following:

1. Start SQL Server Configuration Manager.
2. From the left of the dialog, select **SQL Server Network Configuration** > **Protocols for [instance name]**. In this example, the instance name is **MSSQLSERVER**.
3. Right-click **TCP/IP**, and select **Properties**.



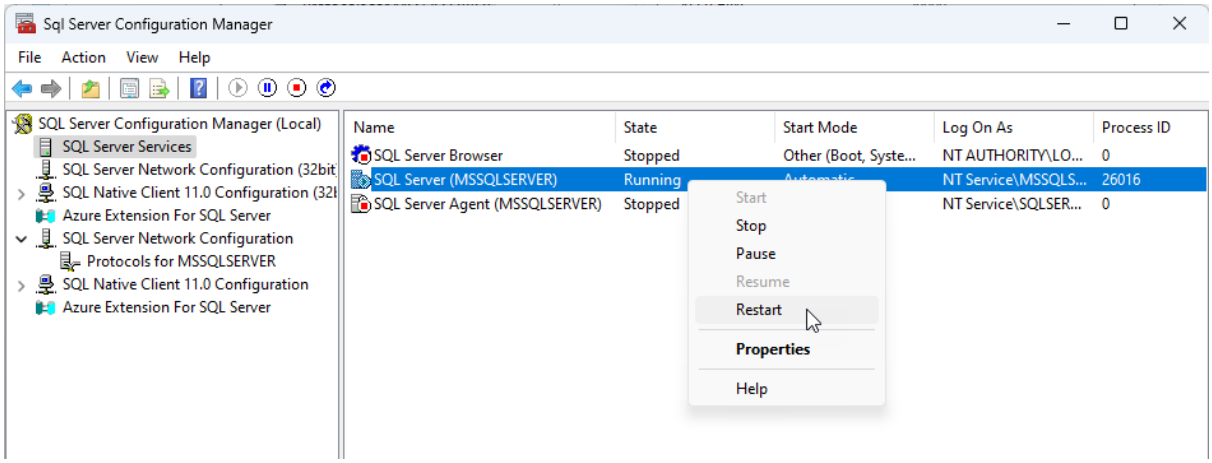
The **TCP/IP Properties** dialog opens.

4. For **Enabled**, select **Yes**.



5. Select the **IP Addresses** tab.
6. Below **IPAll**, define the port to use:
   - If you want to use a static port, enter *1433* or some other port in the **TCP Port** field. 1433 is the default communication port for SQL Server. If necessary, you can also leave the field empty in case you need to change the port later, for example – the port will then be a dynamic port that changes with every startup.
   - If you want to use a dynamic port that changes with every startup, enter 0 in the **TCP Dynamic Ports** field.
7. Click **OK**.
8. From the left of the dialog, select **SQL Server Services**.

9. Right-click **SQL Server ([instance name])**, and select **Restart**. In this example, the instance name is **MSSQLSERVER**.



The service restarts itself with your modifications. You can close the program.

Next, connect to the SQL Server database.

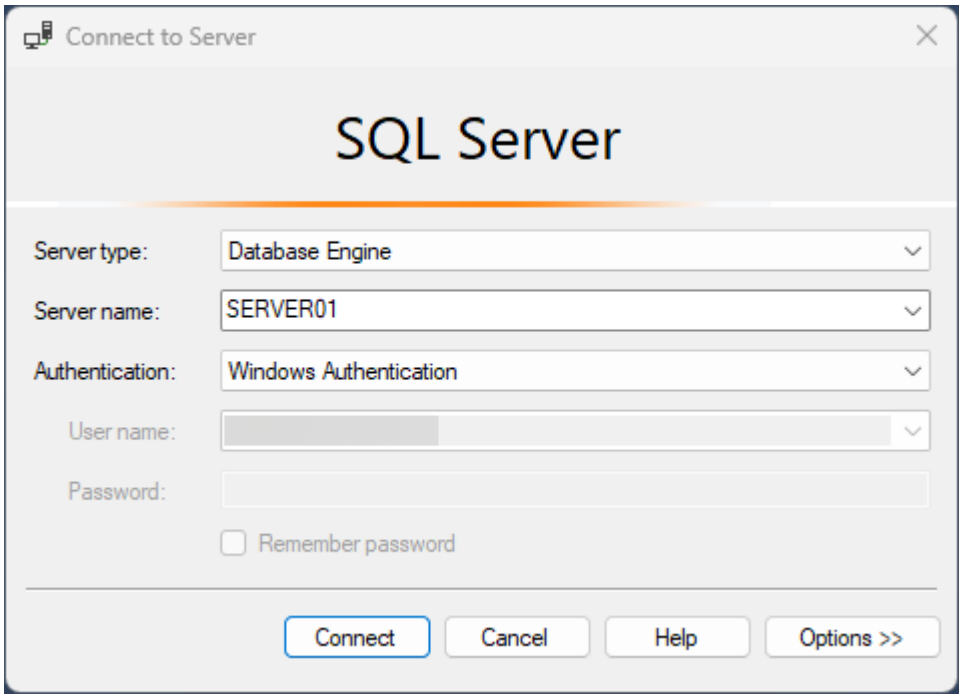
4.3. Connect to the SQL Server database

The server name you use to connect to the SQL Server database depends on the port you defined; you will either use the server name only, the server name and the port, or the server name and the instance name.

Do the following:

1. Start Microsoft SQL Server Management Studio. The **Connect to Server** dialog opens.
2. For **Server name**, enter the name of the server or the names of both the server and the instance or port that you just configured as follows:
   - If you configured the SQL Server to use the default port 1433, enter the name of the server only, e.g. *SERVER01*.
   - If you configured the SQL Server to use a static port other than 1433, enter the name in the format *[server name],[port]*, e.g. *SERVER01,1910*.
   - If you did not define a port at all or you defined a dynamic port, the port changes with every startup. Thus, enter both the server name and the instance name in the format *[server name]\[instance]*, e.g. *SERVER01\MSSQLSERVER*.

In this example, the default port 1433 is used so the connection is made with the server name *SERVER01* only.



3. For **Authentication**, select **Windows Authentication**.
4. Click **Connect**.

The program connects to the server. The **Object Explorer** window on the left shows you the folders that are on the server.

Next, configure the firewall.

4.4. Configure the Windows firewall

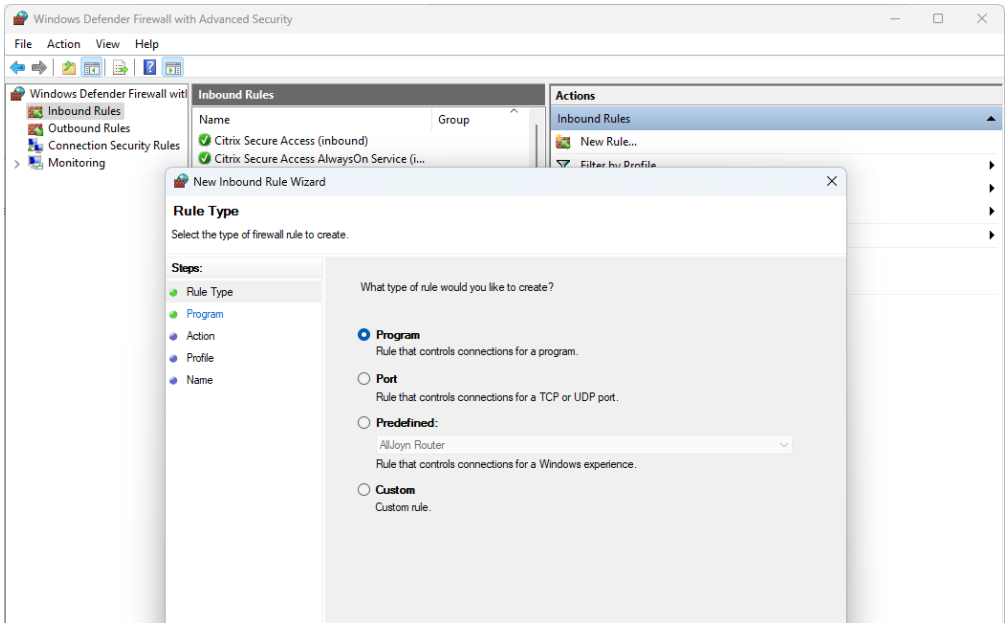
You have to allow inbound traffic via the configured port through the Windows firewall.

- If you configured the default port 1433 or some other static port, you only need to allow inbound traffic via that port through the Windows firewall.
- If you did not configure a TCP port or you configured a dynamic port (0), the port changes in each startup. You then need to note the following:
  - You need to allow inbound traffic via the whole range of dynamic ports through the Windows firewall.
  - You need to open the UDP port 1434.
  - SQL Server Browser needs to be running on the server.

In this example, the default port 1433 is used.

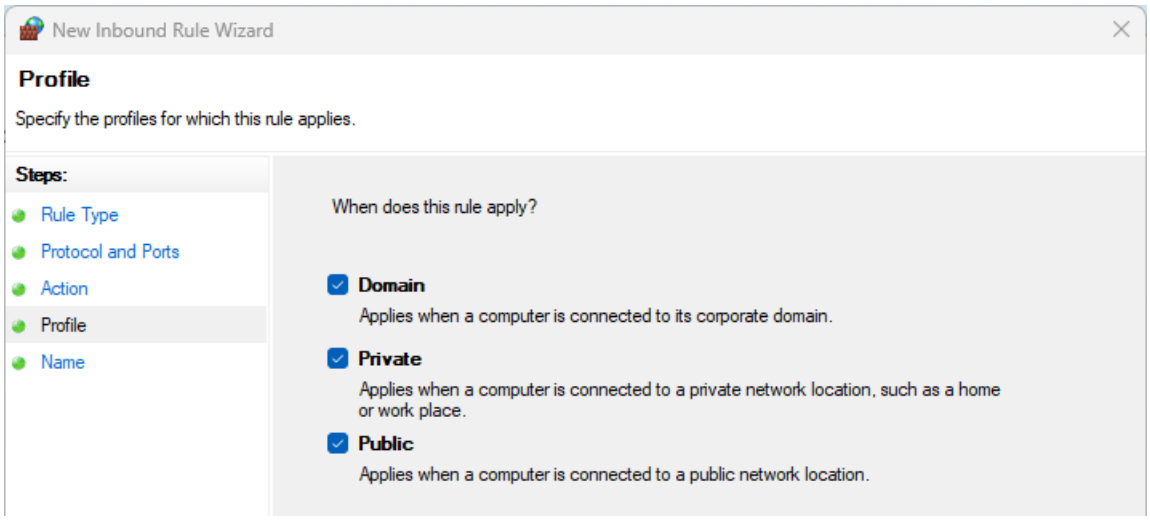
Do the following:

1. In the Windows search field, enter *wf.msc*.
2. Open the program. The **Windows Defender Firewall with Advanced Security** dialog opens.
3. On the left side of the dialog, select **Inbound Rules**.
4. On the right side of the dialog, click **New Rule**. The **New Inbound Rule Wizard** dialog opens.



5. Select **Port**.
6. Click **Next**.

7. Select **TCP**.
8. For **Specific local ports**, enter the port, such as *1433*.
9. Click **Next**.
10. Select **Allow the connection**.
11. Click **Next**.
12. Select all the profiles taking the appropriate information security measures into account. In this example, all profiles are selected.



13. Click **Next**.
14. Enter a name for the firewall rule.
15. Click **Finish**. Traffic through the firewall to the SQL Server instance is now allowed and connection outside of the server is possible.

Next, define the main user of the CADMATIC Electrical SQL Server databases and create the databases.

4.5. Define the main user and create databases

Define the main user and create databases in Microsoft SQL Server Management Studio.

You will need to define the **db_owner** user role for the main user. This role will allow the main user to create database tables and update the database structure when upgrading to a new version. For more information on the roles, see User roles.

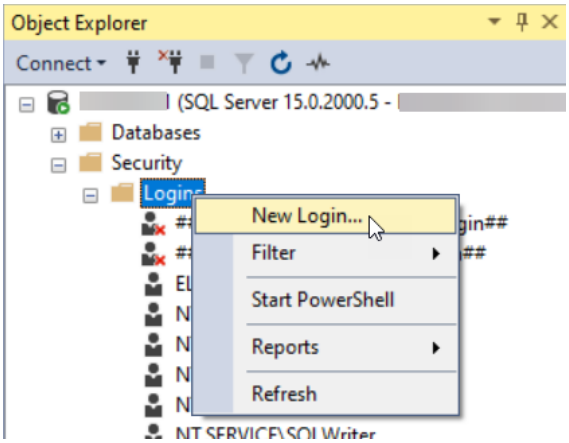
You will need to create three types of databases: a management database, a project database and shared databases.

- The management database includes a list of different project databases, shared databases, and users.
- The project database contains all the CADMATIC Electrical project data in the SQL Server instance.
- The shared databases contain shared design content, such as cable types, product information, and plate definitions.

You can name these databases any way you want. In these instructions, the databases are named **Management**, **Project** and **Shared**.

Do the following:

1. Connect to the SQL Server database.
2. Add a new user (who you will define as the main user later):
    a. In Object Explorer, open **Security** > **Logins**. This folder contains all SQL Server users and user groups.
    b. Right-click **Logins**, and select **New Login**.



The **Login - New** dialog opens.

c. For **Login name**, enter the username in the format *[domain name]\[Windows username]*.

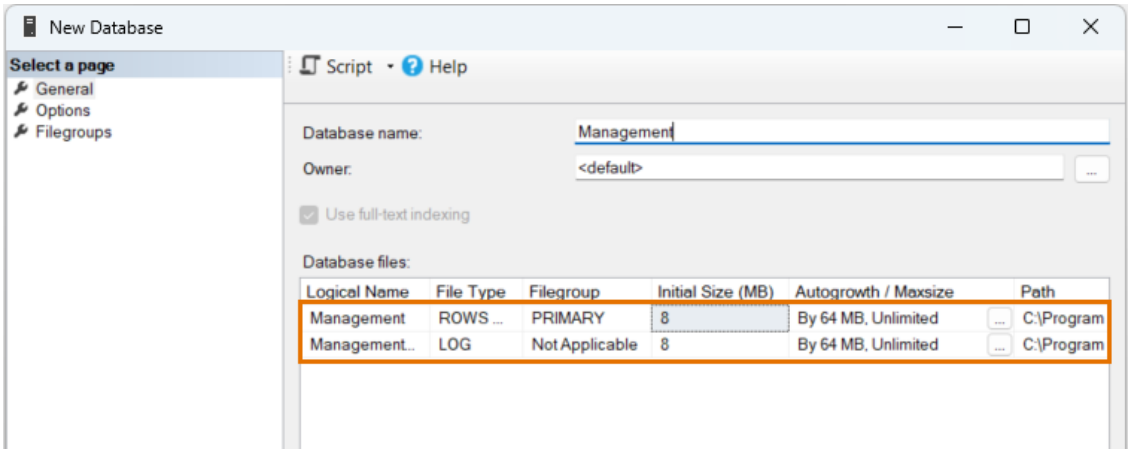
d. Click **OK**. The program creates the new user into the user folder.

3. Create the databases:

a. In Object Explorer, right-click **Databases** and select **New Database**. The **New Database** dialog opens.

b. In the **Database Name** field, enter the desired name. In this example, **Management** is used.

The program creates two files: a **ROWS** file that contains the actual database data, and a **LOG** file that contains the SQL commands that are sent to the database.



c. Define the default sizes:

i. For the ROWS file, for **Initial Size**, enter *64* as the default size.

ii. For the LOG file, enter 32 as the default size.

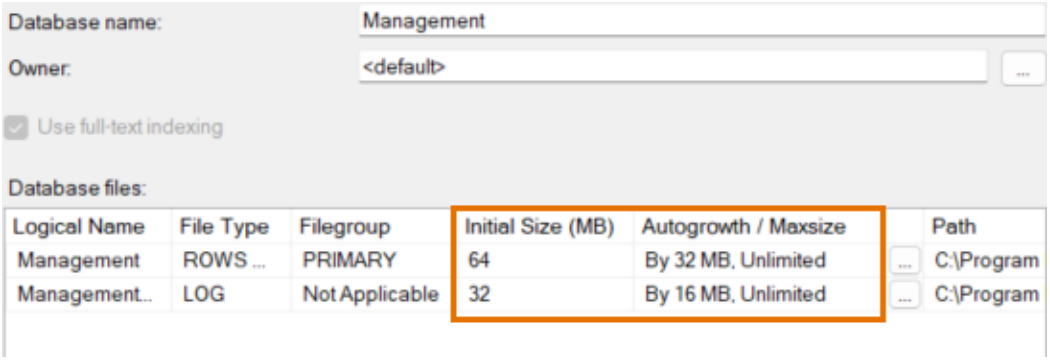
d. Define the growth rates:

i. For the ROWS file, in the **Autogrowth / Maxsize** column, click the ... button. The **Change Autogrowth for [file name]** opens.

ii. For **In Megabytes**, enter *32* as the value.

iii. Click **OK**.

iv. Enter *16* as the growth rate for the LOG file as described.



e. Check the default file location in the **Path** column. You do not need to change the location at this point, but note that the program does not create backups of the files, so you have to do that yourself.

f. Click **OK**.

g. Create two more databases (in this example named **Project** and **Shared**) as described, with the following definitions:

Database type	Type	Default size	Growth rate
Project	ROWS	512	256
	LOG	128	64
Shared	ROWS	128	64
	LOG	64	32

4. Define the main user for the databases:

a. Open **Security** > **Logins**.

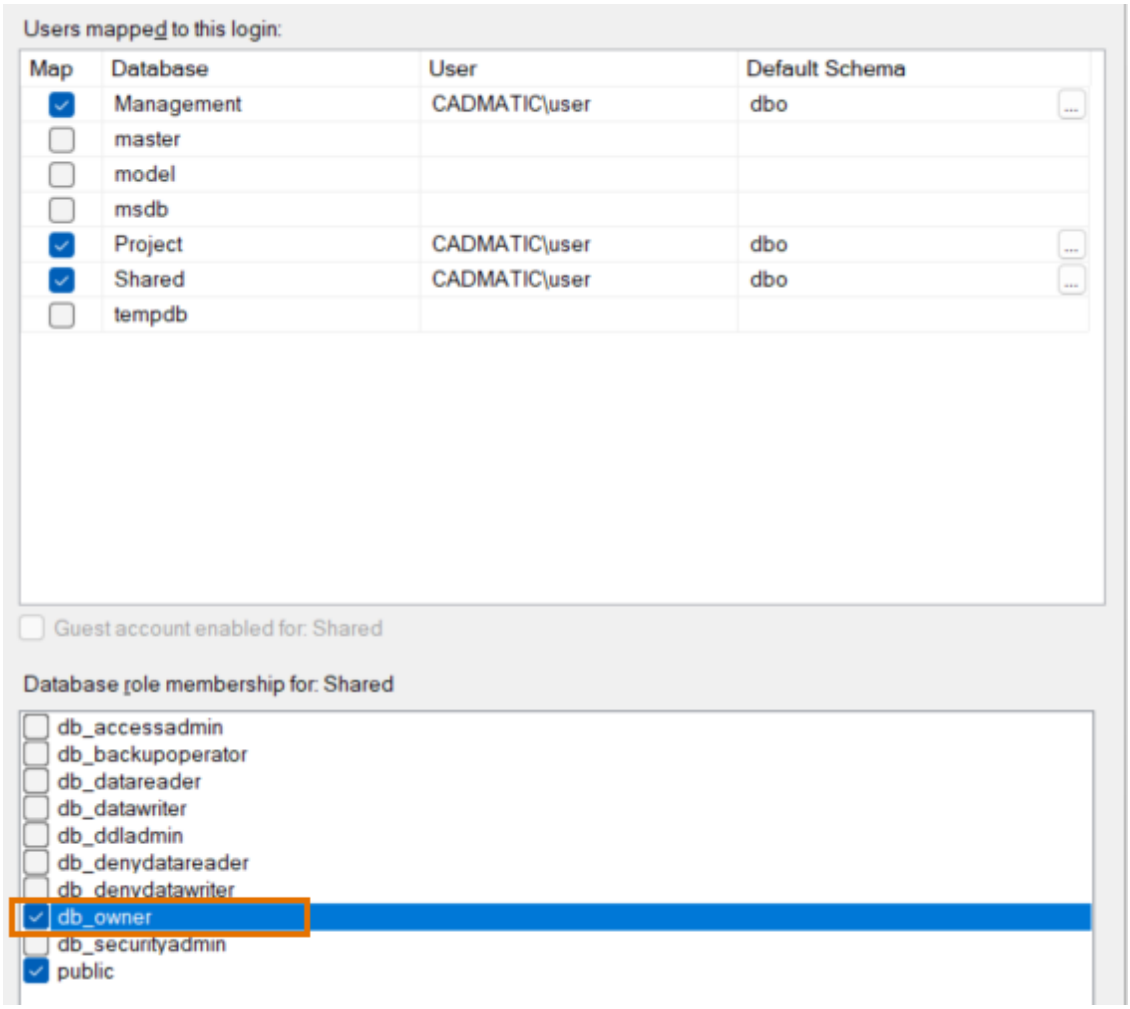
b. Right-click the user name you added in step 1, and select **Properties**. The **Login Properties - [User name]** dialog opens.

c. On the left side of the dialog, select **User Mapping**.

d. In the **Map** column, select the databases you just created.

e. For **Default Schema**, enter *dbo* as the value for each database.

f. One by one, in the list at the bottom of the dialog, select **db_owner** for each database.



g. Click **OK**. The program adds the user as the main user for the databases. If you want to check that everything is in order, select **Databases** > [**Database**] > **Security** > **Users**. The user is visible in the user list.

Next, take SQL Server project management into use.

4.6. Take SQL Server project management into use

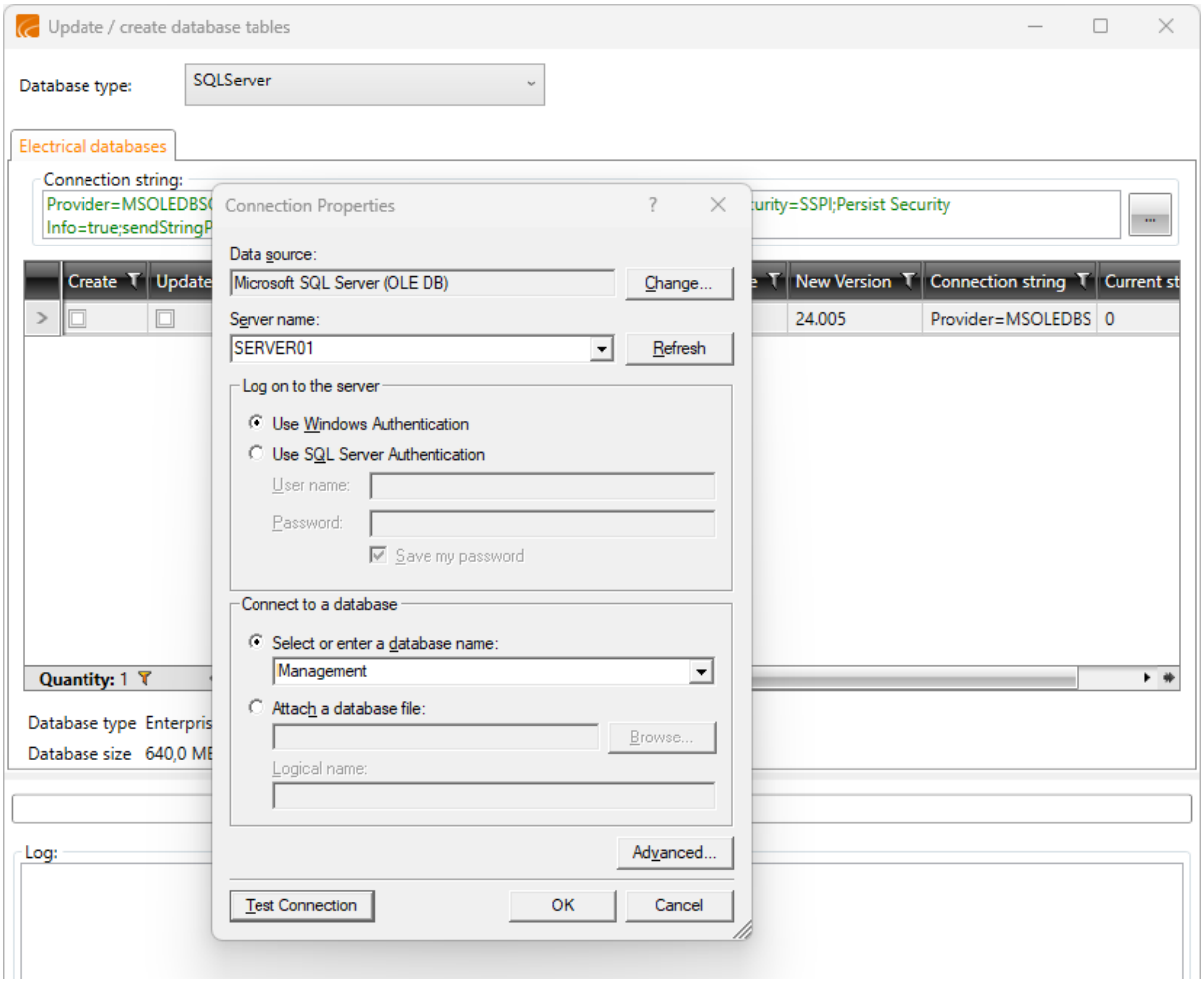
Before you implement project management, log into the workstation with the username that you defined as the main user for the SQL Server.

Do the following:

1. Start CADMATIC Electrical.
2. Select **Electrical** tab > **Projects** group > **Open** menu > **SQL Server project management**.
   When you start the project management for the first time, the program asks you about the configuration of the database.
3. Click **Yes**. The **New Database** dialog opens.
4. In the **Server** field, enter the name of the server in the same way you did when connecting to the SQL Server database, i.e. the server, server and instance, or server and port as follows:
   - If the instance uses the default port 1433, enter the name of the server only, e.g. *SERVER01*.
   - If the instance uses a static port other than 1433, enter the name in the format *[server name],[port]*, e.g. *SERVER01,1910*.
   - If you did not define a port at all or you defined a dynamic port, enter both the server name and the instance name in the format *[server name]\[instance]*, e.g. *SERVER01\MSSQLSERVER*.

   In this example, the instance uses the default port 1433 so the connection will be made with the server name *SERVER01*.
5. Test the connection by clicking **Test**. If the database connects, the **Testing database connection** dialog opens and informs you of the success of the connection.
6. Click **Yes**. The **Update / create database tables** dialog opens.
7. For **Database type**, select **SQLServer**.
8. Next to **Connection string**, click the ... button. The **Connection Properties** dialog opens.
9. In the **Server name** field, enter the server name as before, i.e. the server, server and instance, or server and port. In this example, only the server name **SERVER01** is used.

10. For **Select or enter a database name**, select the management database. In this example, the management database is called **Management**.



11. Click **OK**.
12. In the table's **Create** column, click the check box to select it.
13. Click **Update / create database tables**. The program informs you of a successful creation.
14. Click **OK**.
15. Click **Close**.
16. Close the **New database** dialog by clicking **OK**. The program informs you of the folders where the other users have to save their files so that they can start using the shared files.
17. Click **OK**. The **SQL Server project management** dialog opens.

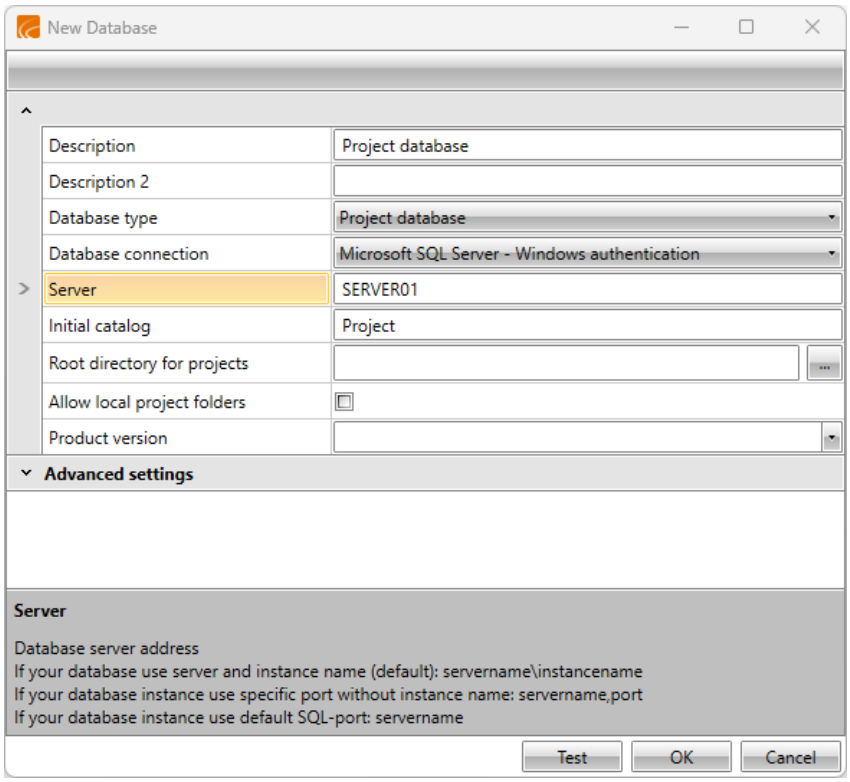
Next, create the project database structure.

4.7. Create project database structure

Do the following:

1. In the **SQL Server project management** dialog, select the **Databases** tab.
2. Click **New**. The **New Database** dialog opens.
3. Enter the desired description for the database.
4. In the **Server** field, enter the name of the server in the same way as before, i.e. the server, server and instance, or server and port as follows:
    - If the instance uses the default port 1433, enter the name of the server only, e.g. *SERVER01*.
    - If the instance uses a static port other than 1433, enter the name in the format *[server name],[port]*, e.g. *SERVER01,1910*.
    - If you did not define a port at all or you defined a dynamic port, enter both the server name and the instance name in the format *[server name]\[instance]*, e.g. *SERVER01\MSSQLSERVER*.

    In this example, the instance uses the default port 1433 so the connection will be made with the server name **SERVER01**.



5. Click **Test**. If the server connects, the **Testing database connection** dialog opens and asks about updating/creating database tables.
6. Click **Yes**. The **Update / create database tables** dialog opens.
7. In the table's **Create** column, click the check box to select it.
8. Click **Update / create database tables**. The program informs you of a successful creation.
9. Click **OK**.
10. Click **Close**.
11. Close the **New Database** dialog by clicking **OK**. The project database is now ready to be used.

Next, create the shared database structure.

4.8. Create shared database structure

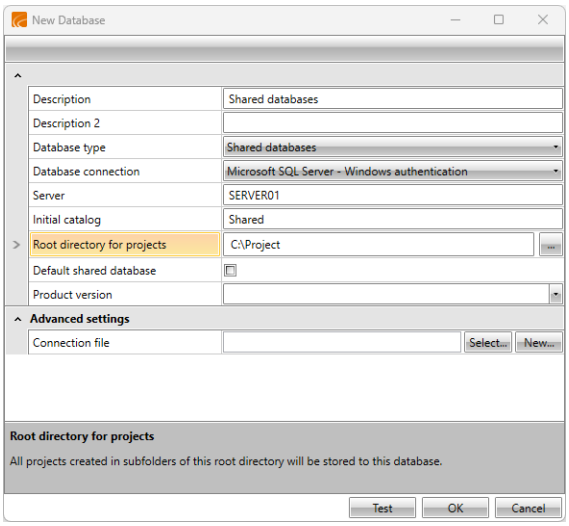
Shared databases contain shared design content, such as cable types, product information, and plate definitions. When you are working on a project, you select design content for the project from these shared databases, which then ends up as the project design content.

Do the following:

1. In the **SQL Server project management** dialog, select the **Databases** tab.
2. Click **New**. The **New Database** dialog opens.
3. Enter or select the following information for the database:
    - **Description** – *Shared databases*
    - **Database type** – **Shared databases**
    - **Server** – The server name, the server name and the port, or the server name and the instance name as before, i.e. the server, server and instance, or server and port as follows:
        - If the instance uses the default port 1433, enter the name of the server only, e.g. *SERVER01*.
        - If the instance uses a static port other than 1433, enter the name in the format *[server name],[port]*, e.g. *SERVER01,1910*.
        - If you did not define a port at all or you defined a dynamic port, enter both the server name and the instance name in the format *[server name]\[instance]*, e.g. *SERVER01\MSSQLSERVER*.

      In this example, the instance uses the default port 1433 so the connection will be made with the server name **SERVER01**.

- **Initial catalog** – The name of the shared database. In this example, the name is **Shared**.
- **Root directory for projects** – The folder whose sub-folders use this database



4. Click **Test**. The **Testing database connection** dialog opens.
5. Click **Yes**. The **Update / create database tables** dialog opens.
6. Next to **Connection string**, click the ... button. The **Connection Properties** dialog opens.
7. For **Select or enter a database name**, select the name of the shared database. In this example, the name is **EDBUserCommon**.
8. Click **OK**.
9. In the table's **Create** column, click the check box to select it.
10. Click **Update / create database tables**. The program informs you of a successful creation.
11. Click **OK**.
12. Create as many own product databases as you want by repeating the previous steps.

    If the databases are not visible on the list, do the following:

    a. Click **Close**.
    b. Select the database.
    c. Click **DB Update**. The **Update / create database tables** dialog opens and shows you the databases.
    d. Select the databases and click **Update / create database tables**.
13. Click **Close**.
14. Click **OK**.

Next, add other users to SQL Server.

4.9. Add other users

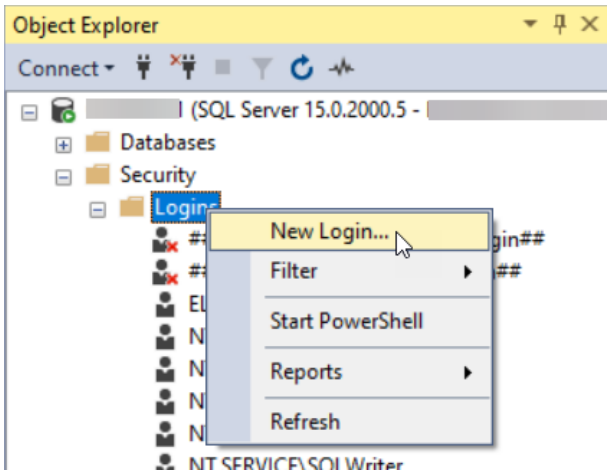
The other users are added in the same way as the main user but with roles starting with **EDB**:

- **EDBAdmin** – The user can create new databases, delete databases, and edit database settings in SQL Server project management.
- **EDBPowerUser** – The user can create and delete SQL Server projects.
- **EDBUser** – The user can edit projects.
- **EDBReader** – The user only has read-only rights to projects.

Each role in the list automatically includes the user rights of all the roles below them.

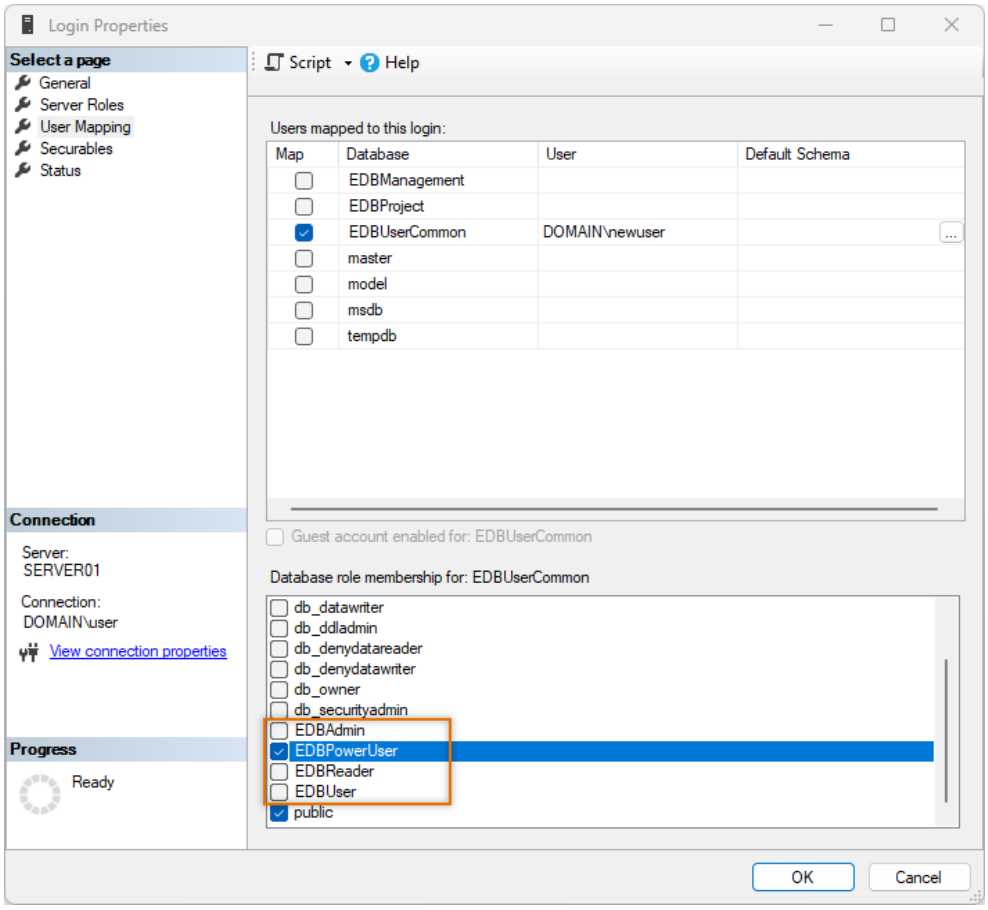
Do the following:

1. Open Microsoft SQL Server Management Studio.
2. Open the **Security** node.
3. Right-click **Logins**, and select **New Login**.



   The **Login - New** dialog opens.
4. For **Login name**, enter the username in the format *[domain name]\[Windows username]*. In this example, *DOMAIN\newuser* is used.
5. Click **OK**. The program creates the new user into the user folder.
6. Below **Security** > **Logins**, right-click the username you added and select **Properties**. The **Login Properties - [Username]** dialog opens.
7. Select **User Mapping**.
8. In the **Map** column, select the databases you want to give the user the rights to.

9. Select the desired database roles beginning with **EDB**, such as **EDBPowerUser**.



10. Click **OK**.

SQL Server is now ready for use, and you can start working on CADMATIC Electrical SQL Server projects.

5. Use and manage SQL Server projects

These instructions about managing CADMATIC Electrical SQL Server projects are intended for Electrical main users. You can manage the database data on the SQL Server by selecting **Electrical** tab > **Projects** group > **Open** menu > **SQL Server project management**.

As the main user, note that the *EDBManagement.ECON* file needs to be shared to all users.

In day-to-day work, SQL Server is not very visible; the functions work the same way for both SQLite/Access and SQL projects. However, a key difference between SQLite/MDB and SQL projects is that the SQL Server database can contain several projects. This means that when you update the project structure, the program updates the entire database and all the projects. Therefore, you should pay attention to when you update the database to make sure that everyone can still access old projects.

5.1. Manage projects

DB tool > **File** > **SQL Server project management**

Electrical tab > **Projects** group > **Open** menu > **SQL Server project management**

You can manage the database data on the SQL Server in SQL Server project management. For example, you can

- Create projects
- Copy projects from one database to another
- Copy and convert project databases between the SQLite/MDB and SQL formats
- Restore backups from the Restore database

5.1.1. Create new projects

After you have taken SQL Server into use, you can choose between the SQL and SQLite formats when creating a new project. The program saves the drawing to the location that is designated for the project. The Electrical project tree shows the type of the project.

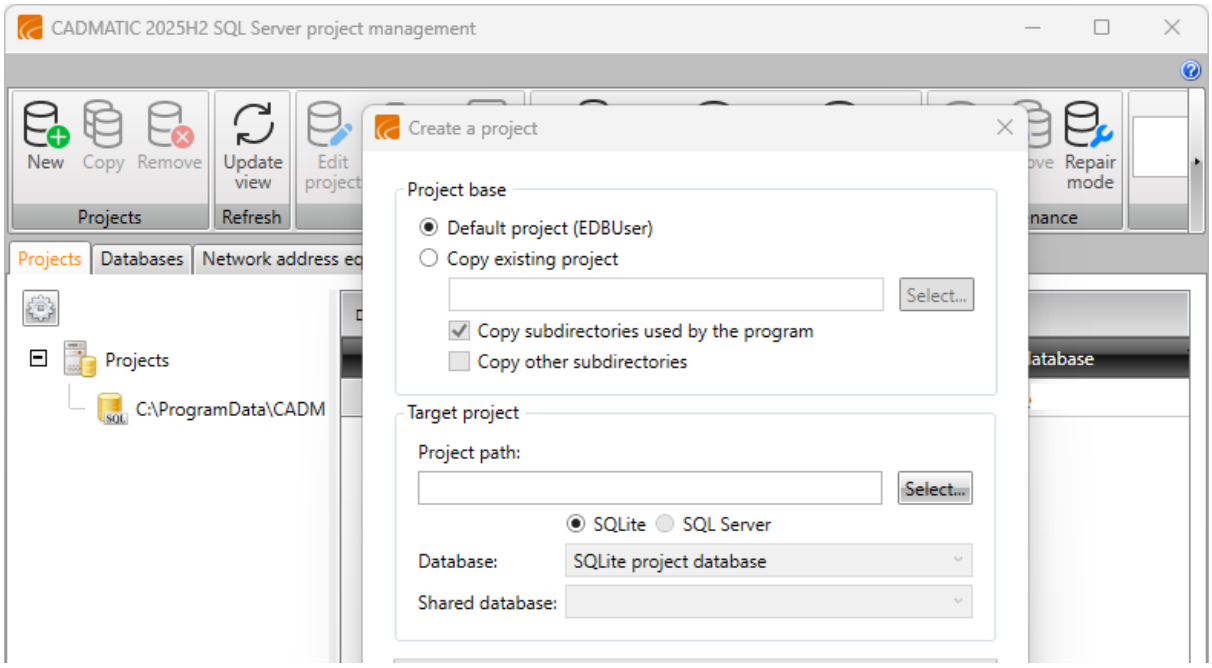
Do the following:

1. Start creating a new project in SQL Server project management by selecting **Projects** tab > **Projects** group > **New**.

   You can also create a new project in one of the following ways:

   - Create the first drawing into an empty folder.
   - In Electrical, select **Electrical** tab > **Projects** group > **New**.
   - In DB, select **File** > **New project**.

   The **Create a project** dialog opens.



2. Select whether you want to use the default project or copy an existing project as the source.

   If you want to use an existing project, do the following:

   a. Select **Copy existing project**.
   b. Click **Select**.
   c. Select whether you want to select the source from a file, a directory or the project management dialog.

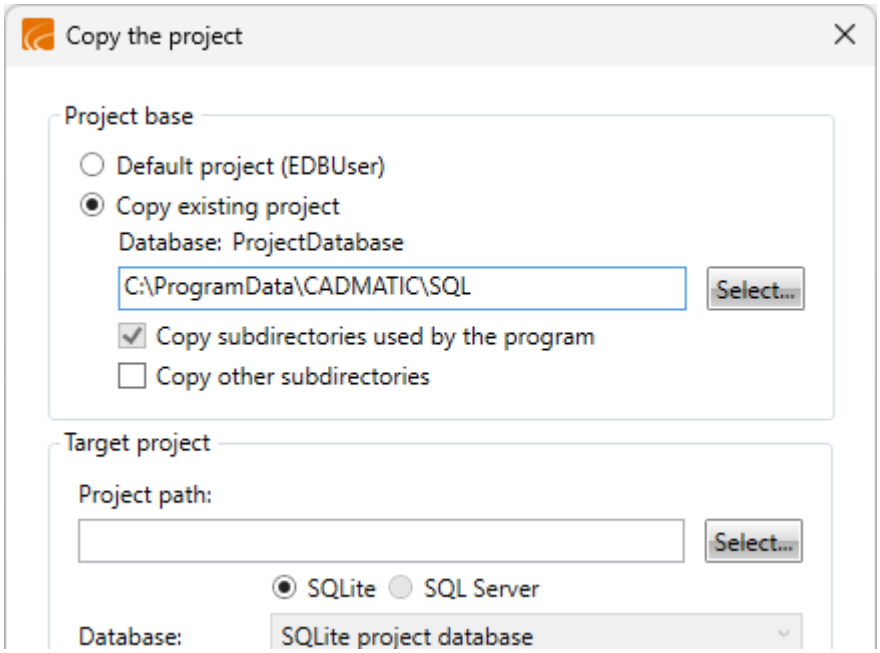
   d. Navigate to the project that you want to use as a source.
   e. Select whether you want to copy subdirectories.
3. Select the directory where you want to create the project and where the drawings are saved.
4. Select whether you want to create an SQLite database or an SQL Server database.
5. If you want to create an SQL Server database, select the database where you want to create the project.
6. Click **OK**.

The program opens a dialog where you can define the project information and settings.

5.1.2. Copy projects

Do the following:

1. In the grid, select the project you want to copy.
2. Select **Projects** tab > **Projects** group > **Copy**. The **Copy the project** dialog opens.



3. Select the directory where you want to create the project and where the drawings are saved.
4. Select whether you want to create an SQLite database or an SQL Server database.
5. If you want to create an SQL Server database, select the database where you want to create the project.
6. Click **OK**.

5.1.3. Convert project databases

When you convert projects, you can either copy or change them:

- Copying results in a copied project and drawings in their own targer folder.
- Changing only transfers the project database data between formats and keeps drawings in their current location.

5.1.3.1. Copy to SQL Server database

Do the following:

1. Select **Projects** tab > **SQLite / Access (DB) conversions** group > **Copy to SQL Server database**.
2. Browse to the project you want to copy and select it. The **Copy the project dialog** opens.
3. Select the directory where you want to create the project and where the drawings are saved.
4. Select whether you want to create an SQLite database or an SQL Server database.
5. If you want to create an SQL Server database, select the database where you want to create the project.
6. Click **OK**.

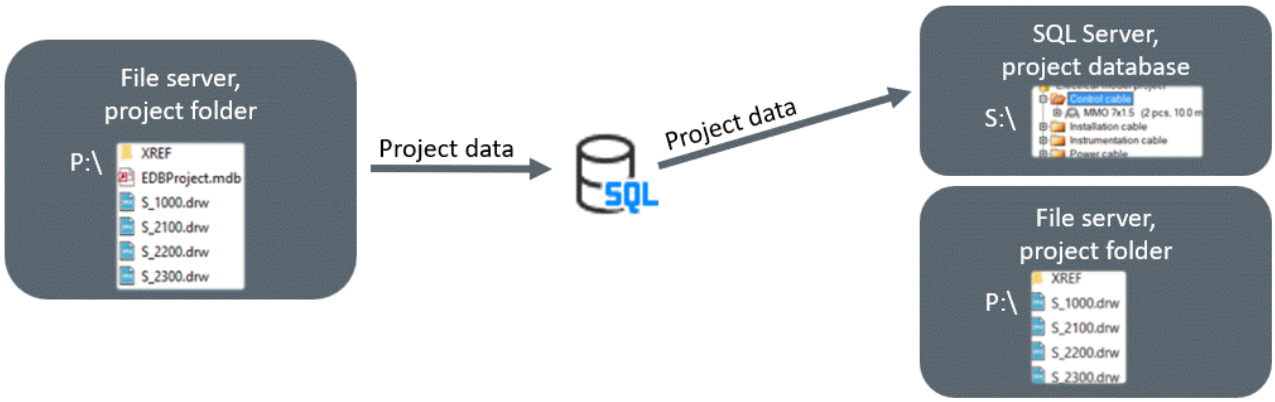
5.1.3.2. Change to SQL Server or to SQLite

With these functions, you can change the format of a project between SQL and SQLite. The program moves the database data to another format, and the drawings remain in their current location.

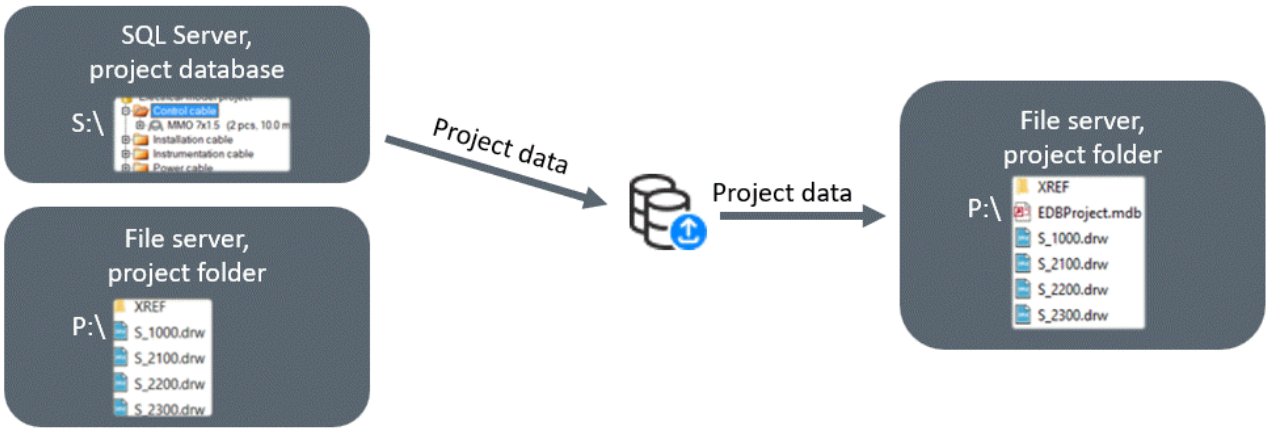
Do the following:

1. Select either an *.sqlite* file or a project from the list.
2. Select **Projects** tab > **SQLite / Access (MDB) conversions** group > **Change to SQL Server** or **Change to another database**. The **Change project to another database** or dialog opens.
3. Select the new database format.
4. Click **Move**.

Change to SQL Server



Change to SQlite



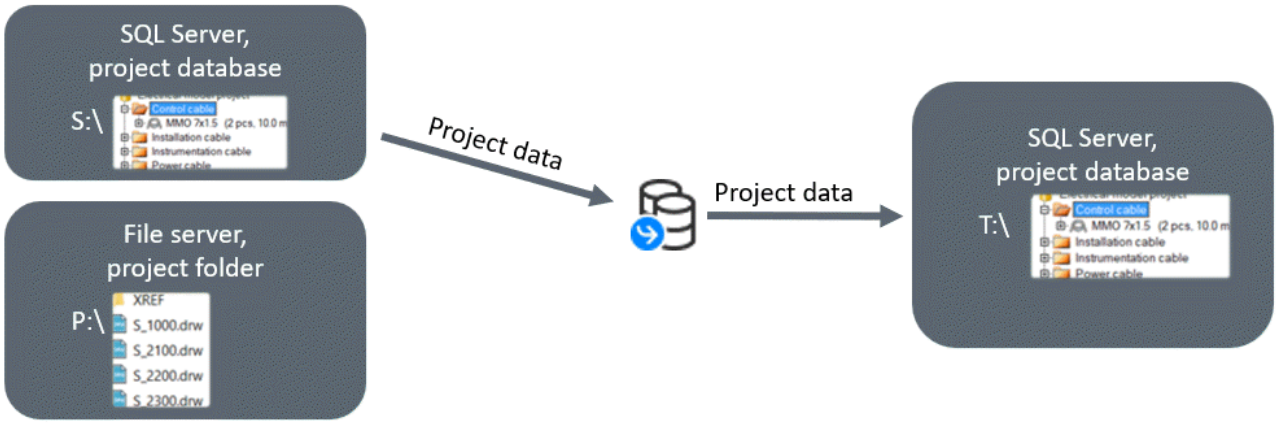
5.1.4. Move project databases

With this function, you can move project data from one database to another. The source and target databases mut be of the same format. If you want to change the format of a database, use the change functions.

Do the following:

1. Select **Move**. The **Move the project to another database** dialog opens.
2. Select the project whose data you want to move from one database to another.
3. Select the current database format.
4. Select the new database format.
5. Click **Move**.

The program moves the data to the new database, and the drawings remain in their current location.



5.1.5. Restore backups

You can restore a backup from a separate Restore database. Note that the **Restore** function restores only database data, so you have to restore the drawings separately. Arrange the backups of the drawings server and the SQL Server with your own IT organisation.

Do the following:

1. Create a recovery database of the SQL Server backup that contains the entire database.
2. Create a new recovery database in SQL Server project management.
   For **Database type**, select **Restore database**.
3. Restore the project over a selected project by clicking **Restore**.

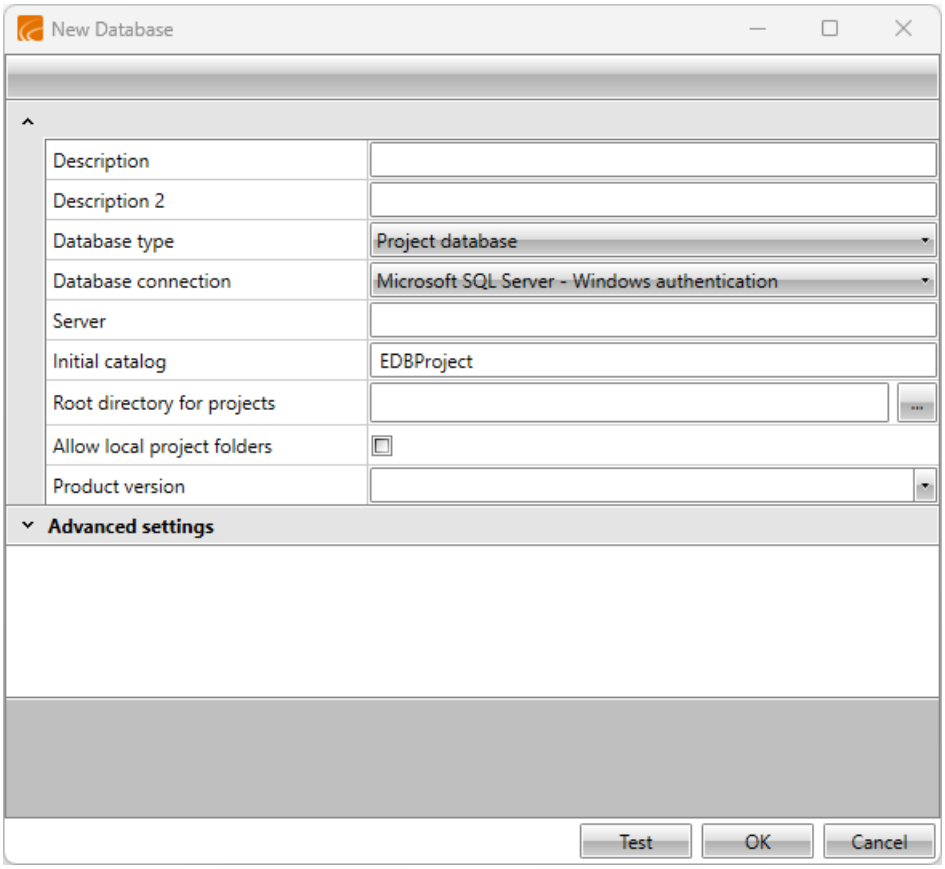
5.2. Create and manage databases

On the **Databases** tab, you can create database tables to previously created SQL Server databases. When you create a database, enter the software version and the contents of the database in its description. The description is visible when you are creating a project and selecting a database to use for it.

You can also remove database tables, for instance if you have moved all projects of the database to another database.

For instructions on how to create a database, see Take SQL Server project management into use. When you are creating a database, select the settings that are best suited for your purposes:

- Database type
  - **Project database** – Select this type when you are creating a database for normal projects.
  - **Shared databases** – Select this type when you are creating a shared database. The database contains company-wide data, such as default cable types.
  - **Restore database** – Select this type when you are restoring a backup.
- **Root directory for projects** – Select the directory whose projects use the SQL database by default.
- **Allow local project folders** – When you select this, you can import projects that are created on the internal drive of the computer into an SQL database. However, note that other users may not necessarily have access to the drawings stored on the local drive.
- **Product version** – If multiple product versions are in use, select the correct one.

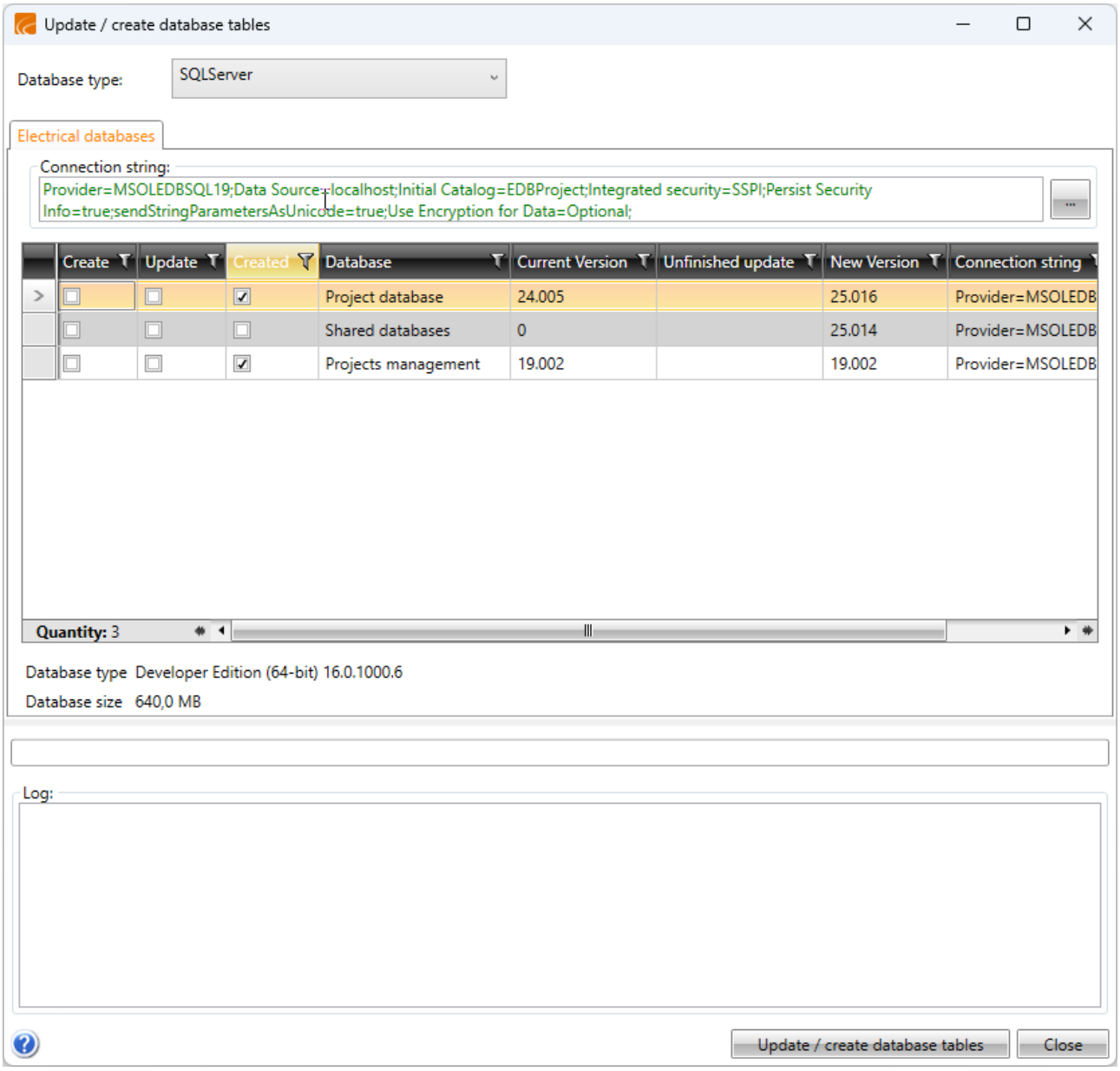


You can update the structure of the database tables to match the needs of the current software version. Note that the structure of the entire database changes and all projects contained in the database are updated.

Electrical notifies you of required updates when you open a project and when new updates are released. Updates within the same version are backwards compatible, and you can safely update them whenever the software requests them.

Important: Updates between different versions are not backwards compatible. Update the structure of the database tables only when you know that all users that participate in the project are using the most recent software version.

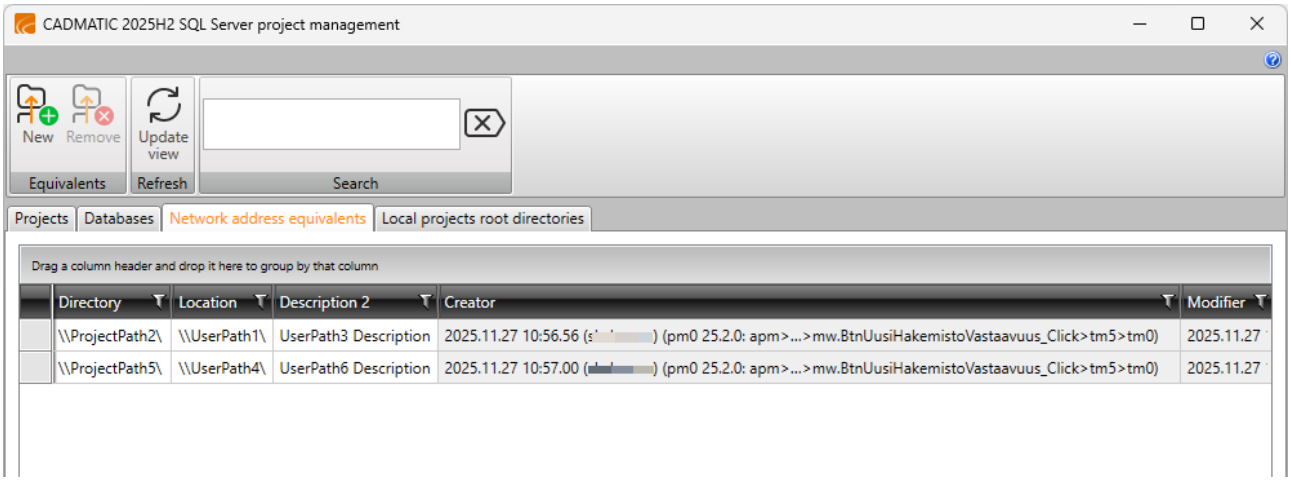
Update both project databases and shared databases in the same way as when creating them. You can only update databases that you have already created.



5.3. Manage network address equivalents

With network address equivalents, you can solve connection problems when the same network address is shown differently for different users. An IP address or a domain, for example, can be directed to the server address.

Select **Electrical** tab > **Projects** group > **Open** menu > **SQL Server project management** and then the **Network address equivalents** tab.



The program finds projects based on their hostname address. The program decodes drive mappings onto hostname addresses, so the letter of the drive mapping has no effect on the address.

If you have mapped drives using IP addresses, hostname.domain names (*server.domain*), or aliases, provide the network address equivalents to the program so that the program can establish connections properly.

If the of the path of a project folder varies between users due to the above reasons, enter these different addresses in the **Directory** column. The **Location** column tells you the path that the project has on the list on the **Projects** tab.

5.4. Move shared databases

The shared databases contain data shared within the company, such as cable types, product models, and plate definitions. This data is used in actual projects.

You can create shared databases on the SQL Server. By default, the program uses the MDB format for shared databases. Reading the database over a VPN connection might be slow if the file is stored in a network location. However, you have to get the design information from the database only once, so the connection does not slow down normal project work.

If you plan on creating all future projects on the SQL Server, you should move the shared databases there as well. Create a database (in these instructions *EDBUserCommon.sqlite*) for shared databases for each version.

Move the information on the current EDBUserCommon database to the database that you created as follows:

1. In SQL Server project management, create a project that uses a SQL Server database as a shared database.
2. Open the project in Electrical DB by selecting **Electrical** tab > **Projects** group > **Open**.
3. Select **Management** tab > **Imports** group > **Imports** menu > **Import to shared database**.
   The **Import to shared database** dialog opens.
4. Select the file that the company uses as the source database.
5. Select the rows that you want to import.
6. Click **Import**.
7. When the importing is finished, click **Close**.

6. Frequently asked questions

For how long is the free MS SQL Express sufficient?

There is no universally applicable answer to this question. The Express version limits the maximum size of a database to 10 GB. You can compare the sizes of your existing databases to this limit to gauge whether it is suitable for your situation. A project takes slightly more space in SQL format than in SQLite/MDB format, but the difference is not significant.

Note that the 10 GB size limit is database-specific, so you can create multiple project databases using the Express version. The paid version has improved functionalities for defragmentation and restoring.

How do I use an SQL project offline?

If you need to move a project to your own computer while working offline, agree on a system with the other project users. When a project is checked out, no one can make changes to the network version of it because restoring affects the entire project. Also note that your company migh use server-based shared databases or other settings.

One way to work on a project offline is as follows:

1. Check a project out:
    a. Convert the SQL project into SQLite/MDB format.
        - The drawings remain in their original locations.
        - The program creates an *EDBProject.sqlite*/*EDBProject.mdb* file into the directory.
        - The project is no longer in SQL Server project management.
    b. Copy the entire project folder and any subfolders to your computer.
    c. Let the other users know that you have checked the project out. You can compress the project into a ZIP file to prevent the others from accidentally editing the project.
2. Work on the project locally.
3. Restore the project:
    a. Copy the project folder and any subfolders to the server.
    b. If necessary, replace the old drawings on the server with new ones.
    c. If necessary, remove the ZIP file.
    d. Convert the project into SQL format.

The drawings remain in their locations and the SQLite/MDB database is removed as the data moves into SQL Server.

Why cannot I connect to the SQL Server using the management tool?

The Microsoft OLE DB Driver 19 for SQL Server (x64) driver provided with Electrical might be missing. If so, download it from Microsoft's website.

How can I test a new software version?

You can copy an existing project into SQLite format and test the new software version on it. If you end up taking the new version into use, remove the test project and update the SQL database to the new version.

What needs to be considered in version updates?

There is no single correct way of conducting a version update. Instead, you should aim to find the most suitable alternative.

First, ensure together with your company's IT expert that the SQL databases have backups. For more information, see Manage projects.

Update an existing SQL database into the new version as follows:

1. Ensure that the other users have the newest software version.
2. Update the structure of the SQL database.
3. Update any possible shared databases.
4. Let the other users know about the update.

Can I create separate databases for different software versions?

You can – however, figure out if there is a real need for it or would it be enough that you update all projects to the newest version. If all projects are in the newest format, you can fetch data from old projects into the new projects.

If you end up creating separate databases for each software version, take care of the following:

- Create the new databases on MS SQL Server Management Studio together with an IT expert.
- Make backups of new databases.
- Enter an informative description text for the databases. When you create a project and select a database for it, the description helps you to figure out which database is the correct one.
- Create a new shared database by copying and updating an existing shared database, so that you do not have to move the data separately.

- Remove the root directory definition from the databases, so that the same root directory does not appear in two different databases.