e!COCKPIT Application Note e!Cockpit

Building

Automation

WAGO-I/O-SYSTEM 750

Installing MySQL Server v5.1.72

Version 1.0.3

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
D-32423 Minden

Tel.: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69

E-Mail: info@wago.com

Web: http://www.wago.com

Technical Support

Tel.: +49 (0) 571/8 87 – 4 45 55
Fax: +49 (0) 571/8 87 – 84 45 55

E-Mail: support@wago.com

Every conceivable measure has been taken to ensure the accuracy and completeness of this documentation. However, as errors can never be fully excluded, we always appreciate any information or suggestions for improving the documentation.

We wish to point out that the software and hardware terms, as well as the trademarks of companies used and/or mentioned in the present document are generally protected by trademark or patent.

Notes about this Documentation

Copyright

Number Notation

Table 1: Number Notation

Number code	Example	Note
Decimal	100	Normal notation
Hexadecimal	0x64	C notation
Binary	'100' '0110.0100'	In quotation marks, nibble separated with dots (.)

Font Conventions

Table 2: Font Conventions

Font type	Indicates
italic	Names of paths and data files are marked in italic-type. e.g.: *C:\Programme\WAGO-I/O-CHECK*
Menu	Menu items are marked in bold letters. e.g.: **Save**
>	A greater-than sign between two names means the selection of a menu item from a menu. e.g.: **File** > **New**
Input	Designation of input or optional fields are marked in bold letters, e.g.: **Start of measurement range**
“Value”	Input or selective values are marked in inverted commas. e.g.: Enter the value “4 mA” under **Start of measurement range**.
[Button]	Pushbuttons in dialog boxes are marked with bold letters in square brackets. e.g.: **[Input]**
[Key]	Keys are marked with bold letters in square brackets. e.g.: **[F5]**

Symbols

⚠ DANGER

Personal Injury!
Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

⚠ DANGER

Personal Injury Caused by Electric Current!
Indicates a high-risk, imminently hazardous situation which, if not avoided, will result in death or serious injury.

⚠ WARNING

Personal Injury!
Indicates a moderate-risk, potentially hazardous situation which, if not avoided, could result in death or serious injury.

⚠ CAUTION

Personal Injury!
Indicates a low-risk, potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

NOTICE

Damage to Property!
Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

NOTICE

Damage to Property Caused by Electrostatic Discharge (ESD)!
Indicates a potentially hazardous situation which, if not avoided, may result in damage to property.

Note

Important Note!
Indicates a potential malfunction which, if not avoided, however, will not result in damage to property.

Information

Additional Information:
Refers to additional information which is not an integral part of this documentation (e.g., the Internet).

Legal Bases

Subject to Change

WAGO Kontakttechnik GmbH & Co. KG reserves the right to make any alterations or modifications that serve to increase the efficiency of technical progress. WAGO Kontakttechnik GmbH & Co. KG owns all rights arising from granting patents or from the legal protection of utility patents. Third-party products are always mentioned without any reference to patent rights. Thus, the existence of such rights cannot be excluded.

Personnel Qualification

The use of the product described in this document is exclusively geared to specialists having qualifications in PLC programming, electrical specialists or persons instructed by electrical specialists who are also familiar with the appropriate current standards.

Moreover, the persons cited here must also be familiar with all of the products cited in this document, along with the operating instructions. They must also be capable of correctly predicting any hazards which may not arise until the products are combined.

WAGO Kontakttechnik GmbH & Co. KG assumes no liability resulting from improper action and damage to WAGO products and third-party products due to non-observance of the information contained in this document.

Limitation of Liability

This documentation describes the use of various hardware and software components in specific example applications. The components may represent products or parts of products from different manufacturers. The respective operating instructions from the manufacturers apply exclusively with regard to intended and safe use of the products. The manufacturers of the respective products are solely responsible for the contents of these instructions.

The sample applications described in this documentation represent concepts, that is, technically feasible application. Whether these concepts can actually be implemented depends on various boundary conditions. For example, different versions of the hardware or software components can require different handling than that described here. Therefore, the descriptions contained in this documentation do not form the basis for assertion of a certain product characteristic.

Responsibility for safe use of a specific software or hardware configuration lies with the party that produces or operates the configuration. This also applies when one of the concepts described in this document was used for implementation of the configuration.

WAGO Kontakttechnik GmbH & Co. KG is not liable for any actual implementation of the concepts.

Table of Contents

1 Description

This Application Note explains how to install a MySQL Sever v5.1.72 on a PFC200 (750-820x). An exchange of Data with the different MySQL-databases, which are stored on the SD Card, can be realized with one of the following librarys:

- `WagoLibMySQL_03.lib` for CODESYS V2.3
- `WagoAppSQL_MySQL` for ***e!****COCKPIT*

This Application note also contains an IEC-61131-3 example program for ***e!***COCKPIT, which functions as a MySQL Client. In this example project, the WAGO 3-Phase Power Measurement Module (750-494) is used in recording measurement values and writing them in the previously created MySQL database.

Note

Firmware Version!
The application note is only supported for PFC200 with firmware up to and including version 11. A function is not available with firmware 12 or higher devices.

Note

Important General Information!
Once the MySQL Server has been installed on the PFC200, it can be accessed via the network by using a remote client. The settings necessary for this are described in Section 4.2.

It is also possible to access the database via a running IEC application, which runs on the identical controller (see Section 5.3).

Note

Handling the SD Card!

- The server is automatically installed on the SD card. This requires an empty PFC card that is correctly inserted in the PFC before the installation process is started.
- In the following, the expression “created SD card” is used as a synonym for a successfully set up SD card with a MySQL database.
- All databases created on the server are saved on the created SD card.
- Removing the SD card when an operation is running stops the operation or causes unpredictable MySQL Server functioning.
- Removing the SD card can lead to data loss and/or a corrupt database.
- The server can be restarted by rebooting the controller with an inserted, previously created SD card.
- If the created SD card is first inserted after the controller restart, the server will not be started. Thus, the controller should always be restarted after the created SD card has been inserted.
- Additional information about exchanging the SD card is presented in Section 6.

Note

SD Card Type!
Please only use a WAGO SD card (Item No. 758-879/000-001). This card has been specially designed for industrial applications in extreme environments. Compatibility with other storage media cannot be guaranteed.

Note

PFC200 Restrictions!
The performance of the MySQL Server on the PFC200 cannot be compared with the performance of a conventional server. The user of this application should be aware of the limited computing capacity of the PFC200.

Note

Power failure!
If a power failure occurs during ongoing operation, this can lead to data loss and/or to a corrupt database. To avoid this failure, an uninterruptible power supply (UPS) should be used.

2 Material Used

2.1 Required Libraries

Library	Description
Standard.lib	Standard library
WagoLibMySQL_03.lib	MySQL library
PowerMeasurement_494_02.lib	Library for the 3-phase power measurement module

2.2 Devices

Supplier	Quantity	Designation	Item No.
WAGO	1	Controller PFC200*	750-820x
WAGO	1	SD memory card	758-879/000-001
WAGO	1	3-phase power measurement module	750-494
WAGO	1	End module	750-600

*Support up to firmware 11

2.3 Tools

Designation	Item No.
e!COCKPIT - Workstation License	2759-101/1110-2002
MySQL Client “HeidiSQL”	-

3 Structure

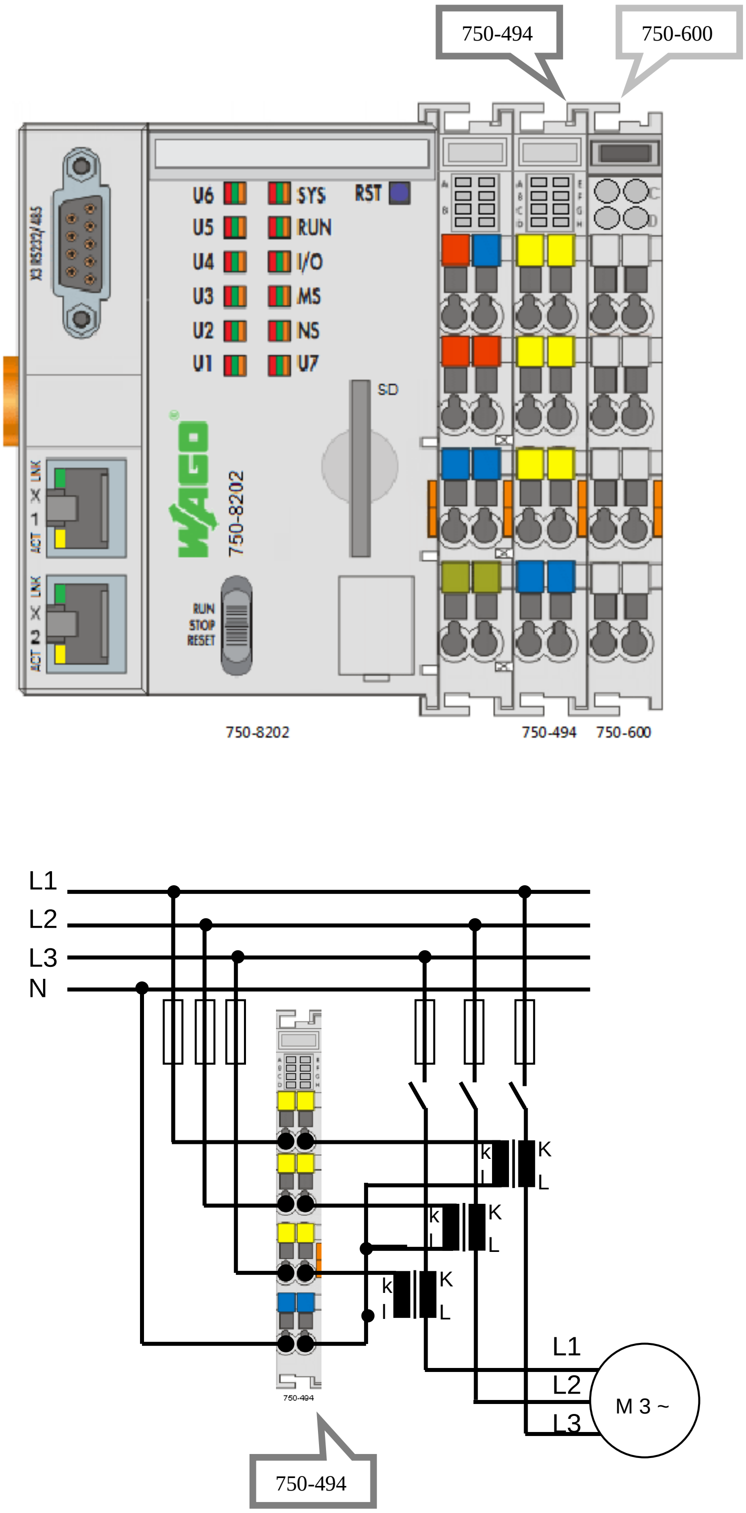


Figure 1: Connection diagram

4 Set-up

The process for installing a MySQL Server is described in the following:

1. After the ZIP file has been downloaded and unzipped, open the folder *“Setup* MySQL *Server/*.” This folder contains three IPK files:

   - *“install-mysql-server-part1.ipk”*
   - *“install-mysql-server-part2.ipk”*
   - *“install-mysql-server-part3.ipk”*

2. Insert an empty SD card in the PFC200. Because it is used as saving location for the database, make sure the card is actually empty before startup.

3. Use a Web browser to open the Web-Based Management (WBM): `“Controller-IP/wbm”`

4. Log on with the user name **admin** and the corresponding password (default password: **wago**).

5. Navigate to the tab “Software Uploads” (see Fig.2 - 1)).

6. Click the **[Browse]** button and select the first IPK file “install-mysql-server-part1.ipk.” Then click the **[Open]** button to select the file (see Fig.2 - 2), 3) & 4)).

7. Click the **[Start Upload]** button to start the upload (see Fig.2 - 5)).

8. After the upload has completed, activate the IPK file by ticking “Activate” and clicking the **[Submit]** button (see Fig.2 - 6)).

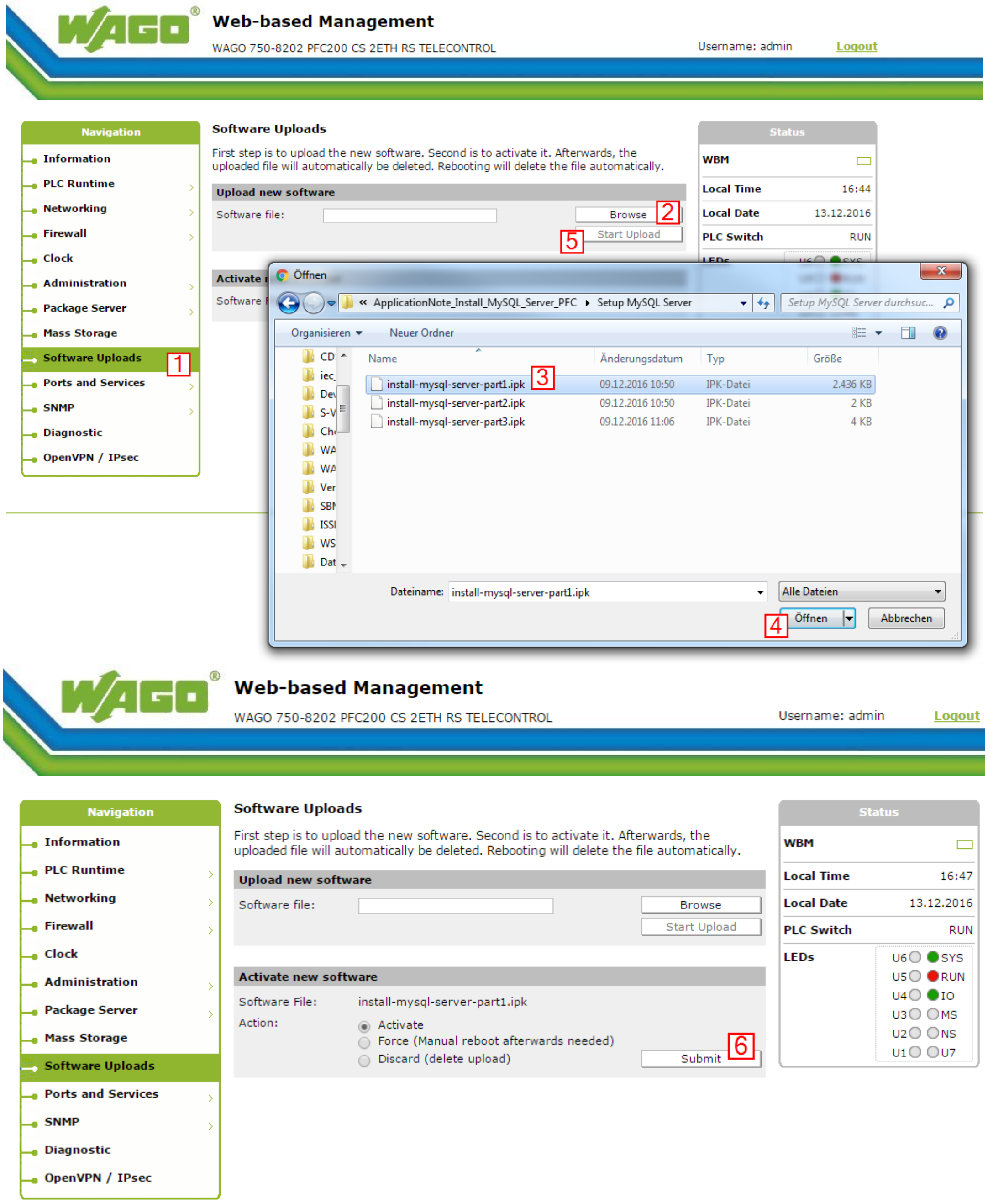
9. Repeat steps 6 through 8 for the other two IPK files.

Note

Sequence of the Upload!

It is imperative to follow the IPK file upload sequence as described under Point 1.

10. The installation process is now finished and the MySQL Server is active.



Web-based Management

WAGO 750-8202 PFC200 CS 2ETH RS TELECONTROL

Username: admin

Software Uploads

First step is to upload the new software. Second is to activate it. Afterwards, the uploaded file will automatically be deleted. Rebooting will delete the file automatically.

Upload new software

Software file:

Local Time 16:44

Local Date 13.12.2016

PLC Switch RUN

Öffnen

« ApplicationNote_Install_MySQL_Server_PFC ▸ Setup MySQL Server

Name	Änderungsdatum	Typ	Größe
install-mysql-server-part1.ipk	09.12.2016 10:50	IPK-Datei	2.436 KB
install-mysql-server-part2.ipk	09.12.2016 10:50	IPK-Datei	2 KB
install-mysql-server-part3.ipk	09.12.2016 11:06	IPK-Datei	4 KB

Dateiname: install-mysql-server-part1.ipk

Alle Dateien

Web-based Management

WAGO 750-8202 PFC200 CS 2ETH RS TELECONTROL

Username: admin

Software Uploads

First step is to upload the new software. Second is to activate it. Afterwards, the uploaded file will automatically be deleted. Rebooting will delete the file automatically.

Upload new software

Software file:

Activate new software

Software File: install-mysql-server-part1.ipk

Action:

● Activate

○ Force (Manual reboot afterwards needed)

○ Discard (delete upload)

WBM	
Local Time	16:47
Local Date	13.12.2016
PLC Switch	RUN
LEDs	U6 ○ ● SYS U5 ○ ● RUN U4 ○ ● IO U3 ○ ○ MS U2 ○ ○ NS U1 ○ ○ U7

Figure 2: Uploading the IPK Files in WBM

4.1 Default Server Settings

After the three IPK files have been activated, the PFC200 functions as a MySQL Server. The following default server settings are set:

- An individual with “root” privileges is set up:
  - Default user name: **admin**
  - Default password: **wago**

Note

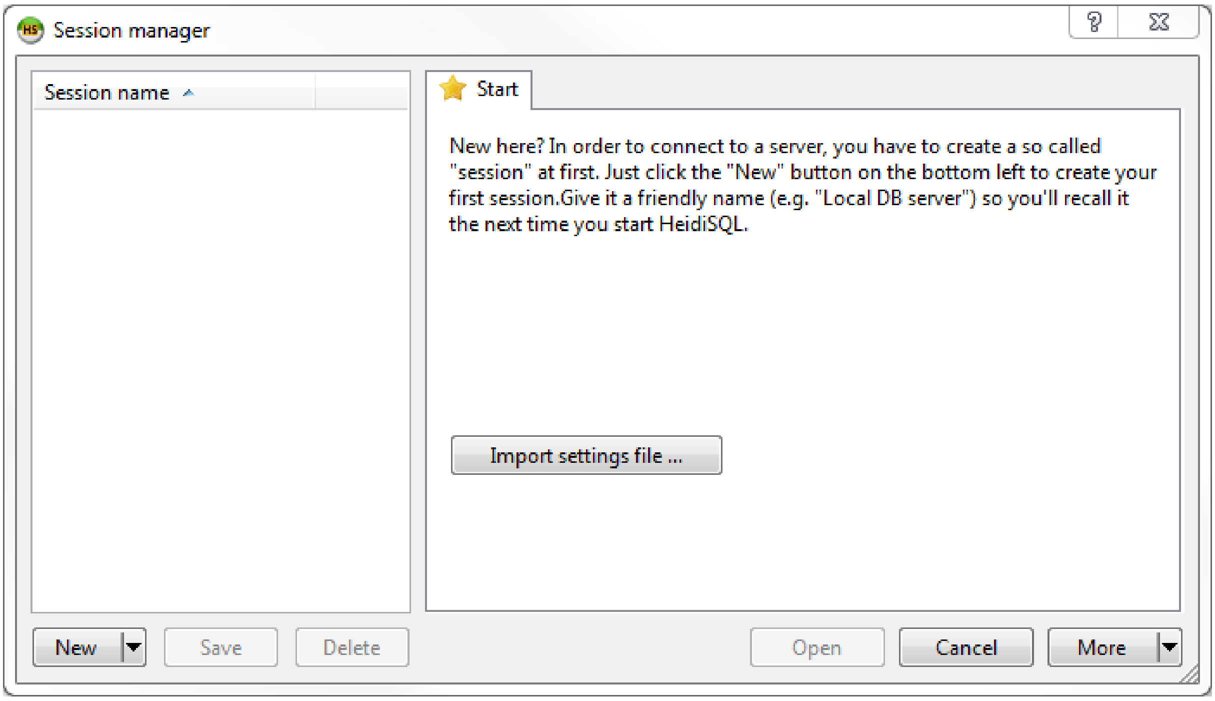
Security Information!
It is strongly recommended to change the default logon information. This procedure is described in Section 5.1.

4.2 Testing the MySQL Server

The simplest way to test the previously installed server is by installing a free MySQL Client on a Windows operating system. A sufficient number of free tools is available on the Internet.

The program “HeidiSQL” is used in this Application Note. HeidiSQL is a free tool which can be used to edit data, create and edit tables, etc.

1. After successful installation, open the program HeidiSQL. The following window is displayed:



New here? In order to connect to a server, you have to create a so called "session" at first. Just click the "New" button on the bottom left to create your first session.Give it a friendly name (e.g. "Local DB server") so you'll recall it the next time you start HeidiSQL.

Figure 3: HeidiSQL Start Window

2. To establish a connection to the server, first a new link must be set up. Click the button **[New]** at the lower left edge of the window.

3. To maintain a clear overview, rename the new session, e.g.: “PFC200_Server.” Then enter the following setup parameters:

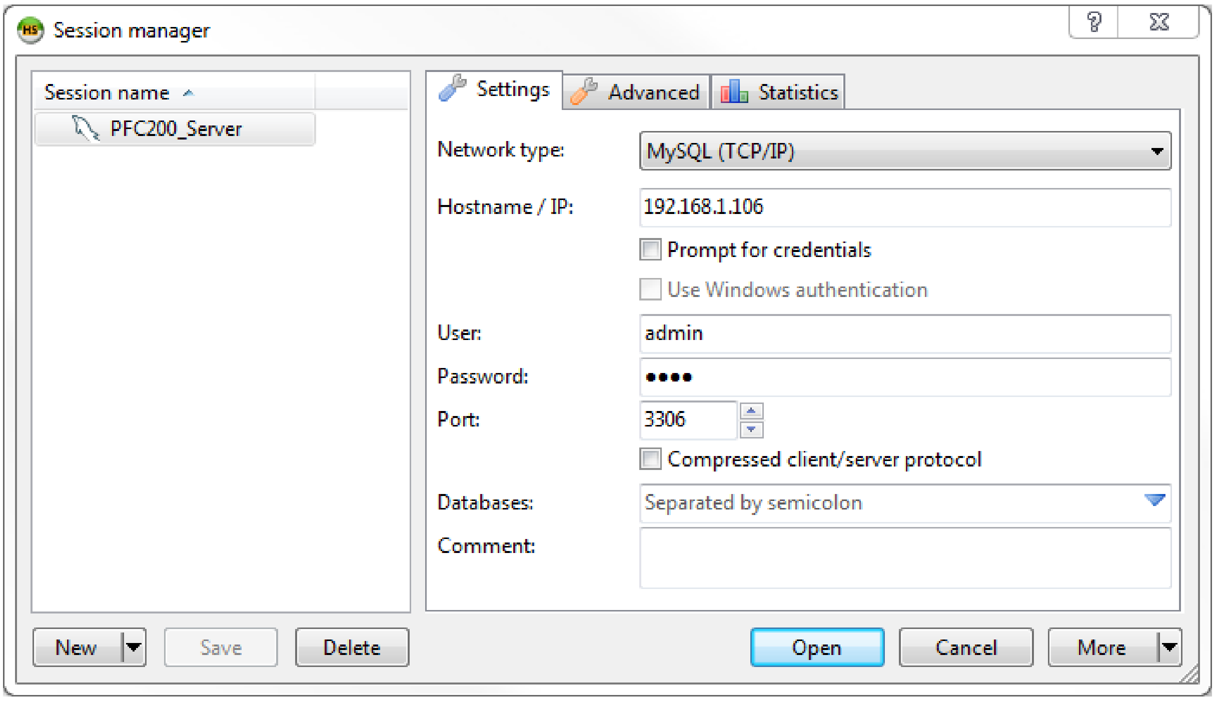
   - **Network type**: MySQL (TCP/IP)

   - **Hostname/IP:** Enter the PFC200 IP address here
     (in this example: 192.168.1.17).

   - **User:** admin

   - **Password:** wago

   - **Port:** 3306



PFC200_Server

Network type: MySQL (TCP/IP)

Hostname / IP: 192.168.1.106

□ Prompt for credentials

□ Use Windows authentication

User: admin

Password: ●●●●

Port: 3306

□ Compressed client/server protocol

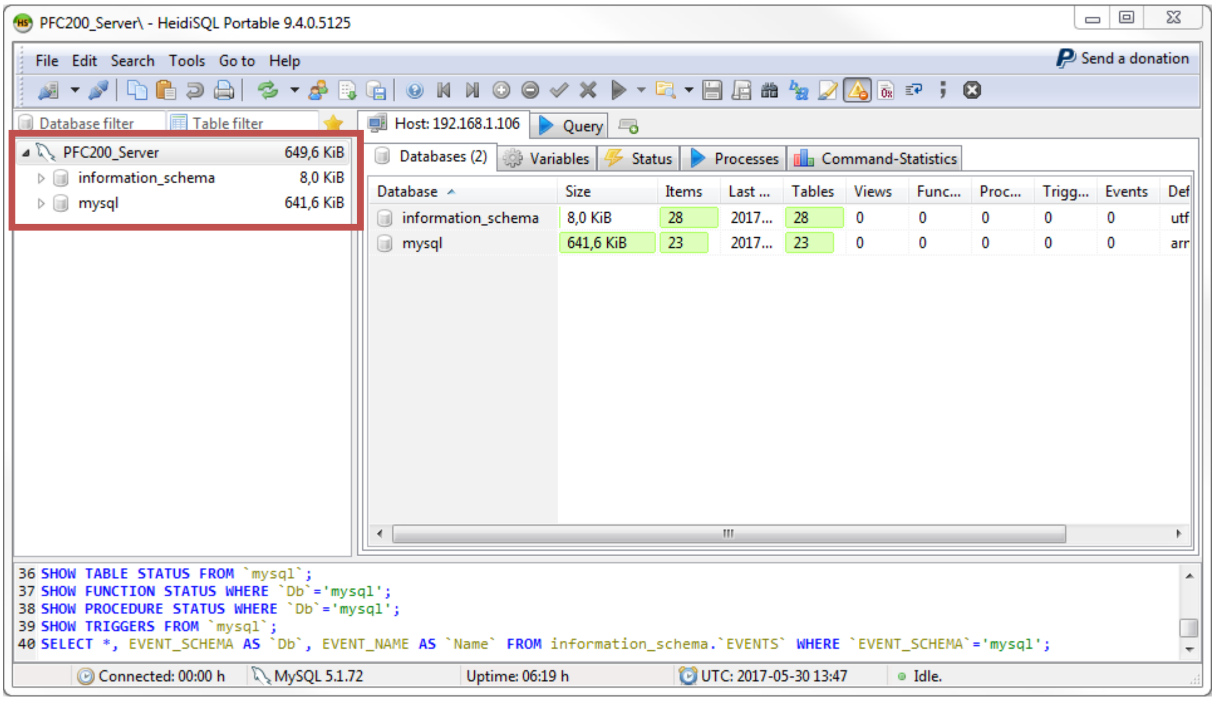
Databases: Separated by semicolon

Comment:

Figure 4: Default Server Settings

4. Then click **[Open].** If the MySQL Server has been correctly installed, now the following two databases should be visible:

   - Information_schema
   - mysql



Database	Size	Items	Last ...	Tables	Views	Func...	Proc...	Trigg...	Events	Def
information_schema	8,0 KiB	28	2017...	28	0	0	0	0	0	utf
mysql	641,6 KiB	23	2017...	23	0	0	0	0	0	arr

```
SHOW TABLE STATUS FROM `mysql`;
SHOW FUNCTION STATUS WHERE `Db`='mysql';
SHOW PROCEDURE STATUS WHERE `Db`='mysql';
SHOW TRIGGERS FROM `mysql`;
SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='mysql';
```

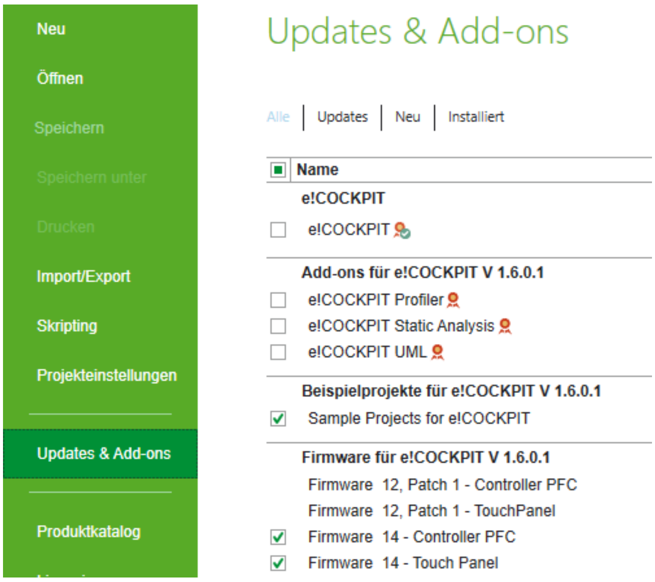
Connected: 00:00 h | MySQL 5.1.72 | Uptime: 06:19 h | UTC: 2017-05-30 13:47 | Idle.

Figure 5: Successful Connection to the PFC200_MySQL-Server

5 Example Application

NOTICE

Installation of Sample Projects for *e!COCKPIT*



■	Name
	e!COCKPIT
□	e!COCKPIT
	Add-ons für e!COCKPIT V 1.6.0.1
□	e!COCKPIT Profiler
□	e!COCKPIT Static Analysis
□	e!COCKPIT UML
	Beispielprojekte für e!COCKPIT V 1.6.0.1
☑	Sample Projects for e!COCKPIT
	Firmware für e!COCKPIT V 1.6.0.1
	Firmware 12, Patch 1 - Controller PFC
	Firmware 12, Patch 1 - TouchPanel
☑	Firmware 14 - Controller PFC
☑	Firmware 14 - Touch Panel

Figure 6: ***e!****COCKPIT* Sample Projects

Sample programs can be called up from the ***e!****COCKPIT* Backstage view by clicking the **Updates & Add-ons** button in the navigation bar.

5.1 Changing Default Logon Information

The following steps demonstrate how the default user logon information “admin” can be changed. This step is optional, but is emphatically recommended for security reasons.

Clicking the menu option **Tools** > **User manager** (see Fig. 6) opens the User manager (see Fig. 7).

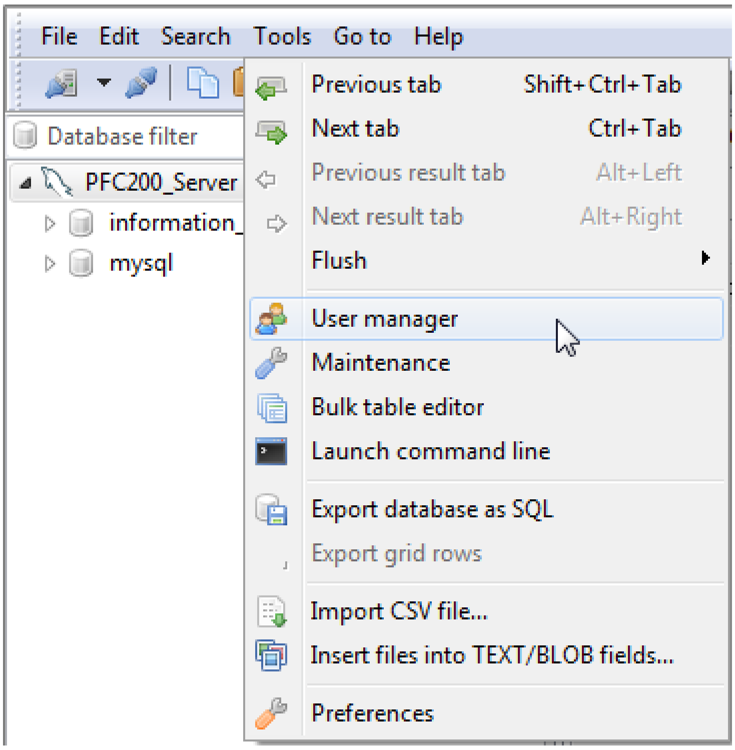
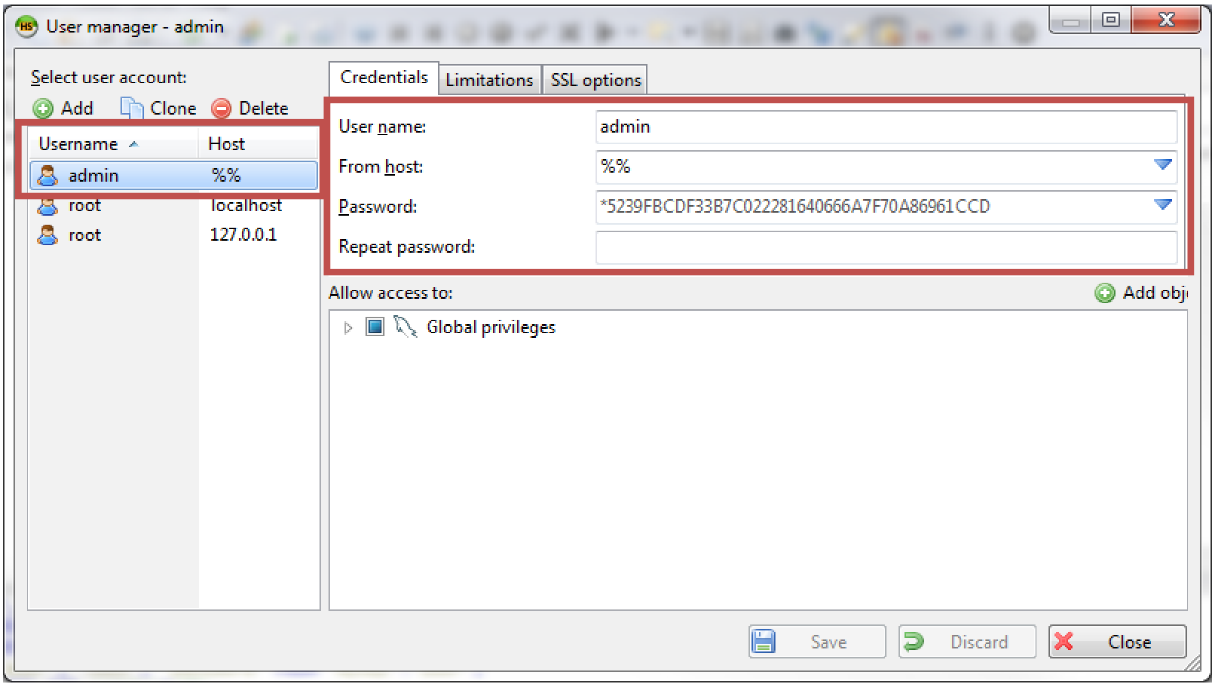


Figure 7: Menu option User manager

Click the account “admin” and enter a new user name and password in the field next to it.

Click **[Save]** to apply the settings.



Username	Host
admin	%%
root	localhost
root	127.0.0.1

User name: admin

From host: %%

Password: *5239FBCDF33B7C022281640666A7F70A86961CCD

Repeat password:

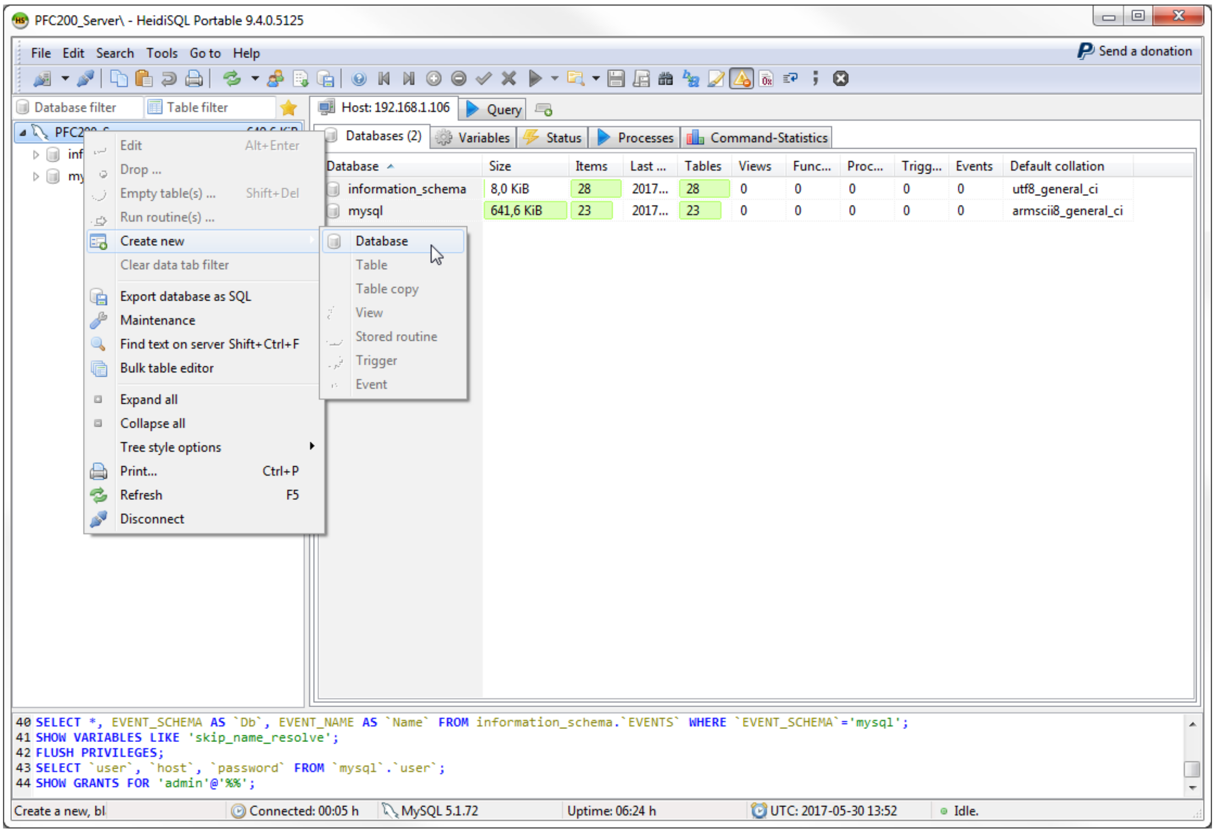
Figure 8: Changing the Logon Information for the User "admin"

5.2 Preparing the Database

The example IEC-61131-3 in this Application Note requires a previously prepared database structure to function properly. Follow these steps to create this structure:

1. Open HeidiSQL and establish a connection to the PFC200_Server (see Section 4.2).

2. Right-click on **“PFC200_Server” > Create new > Database**



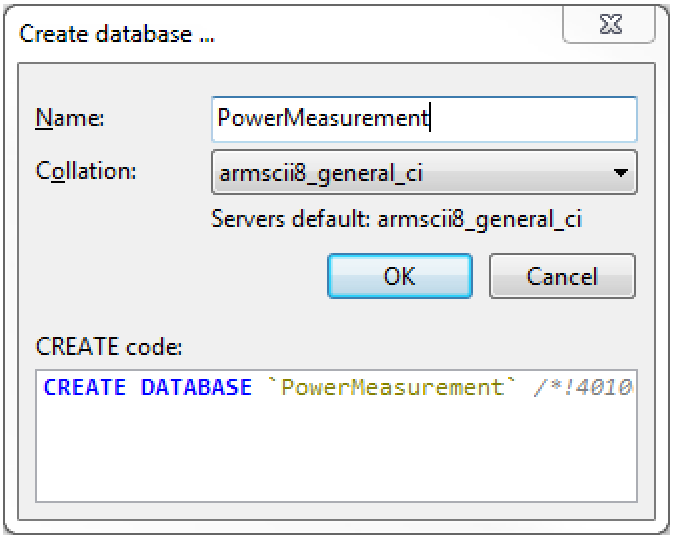
Database	Size	Items	Last ...	Tables	Views	Func...	Proc...	Trigg...	Events	Default collation
information_schema	8,0 KiB	28	2017...	28	0	0	0	0	0	utf8_general_ci
mysql	641,6 KiB	23	2017...	23	0	0	0	0	0	armscii8_general_ci

```
SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='mysql';
SHOW VARIABLES LIKE 'skip_name_resolve';
FLUSH PRIVILEGES;
SELECT `user`, `host`, `password` FROM `mysql`.`user`;
SHOW GRANTS FOR 'admin'@'%%';
```

Create a new, bl | Connected: 00:05 h | MySQL 5.1.72 | Uptime: 06:24 h | UTC: 2017-05-30 13:52 | Idle.

Figure 9: Creating a new Database

3. Enter the name “PowerMeasurement” and confirm with **[Ok]**.



Create database ...

Name: PowerMeasurement

Collation: armscii8_general_ci

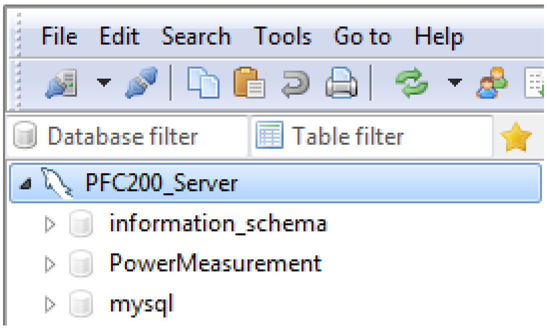
Servers default: armscii8_general_ci

CREATE code:

```
CREATE DATABASE `PowerMeasurement` /*!4010
```

Figure 10: Naming the Newly Created Database

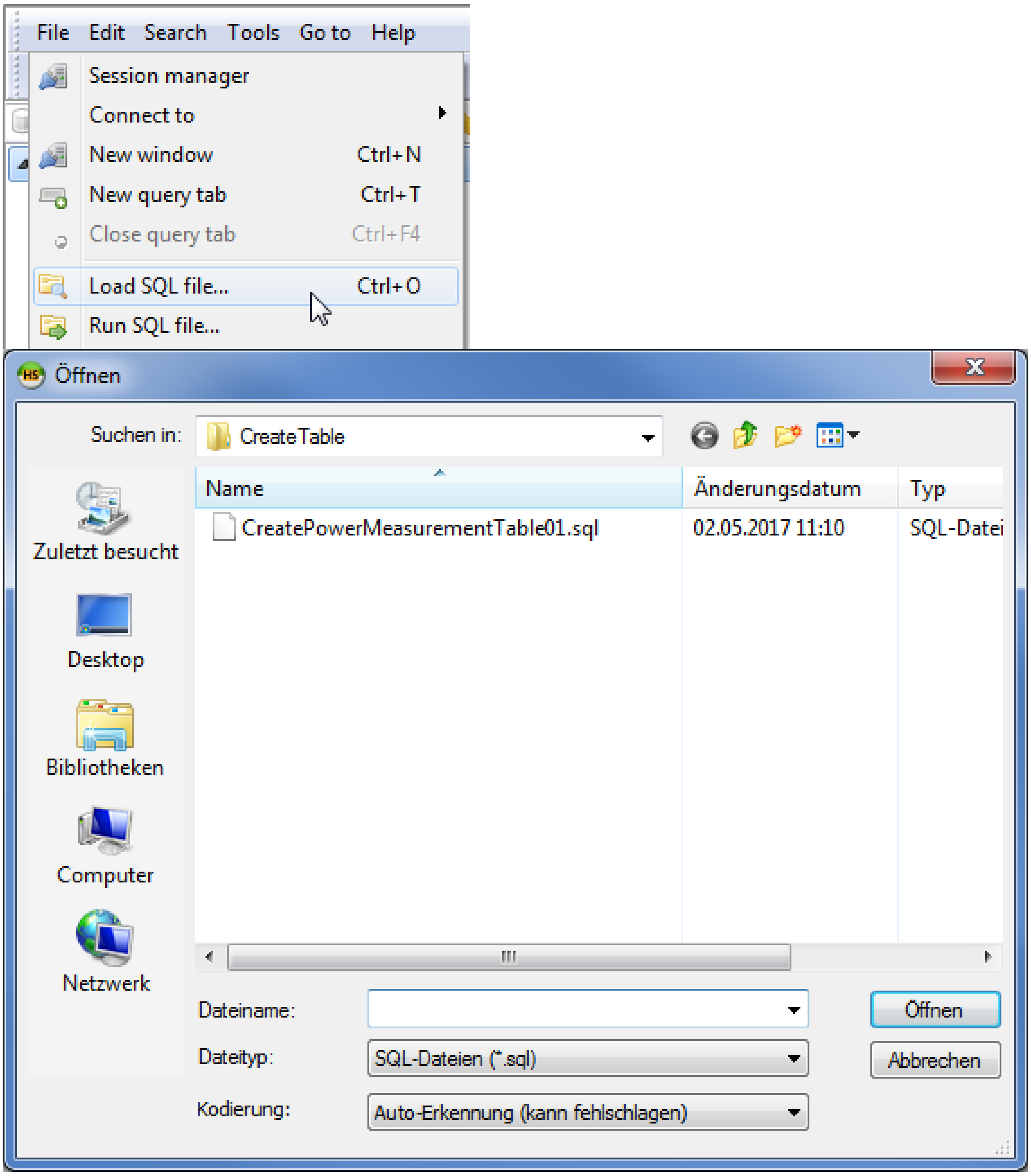
4. A new database is created under the name “PowerMeasurement.” The next step is to set up a table in this database.



PFC200_Server
- information_schema
- PowerMeasurement
- mysql

Figure 11: Created Database “PowerMeasurement”

5. To create a new file, click **File** > **Load SQL file**. Then, select the file “CreatePowerMeasurementTable01.sql” located in the folder .../`Create Table/` and confirm the selection with the **[Open]** button.



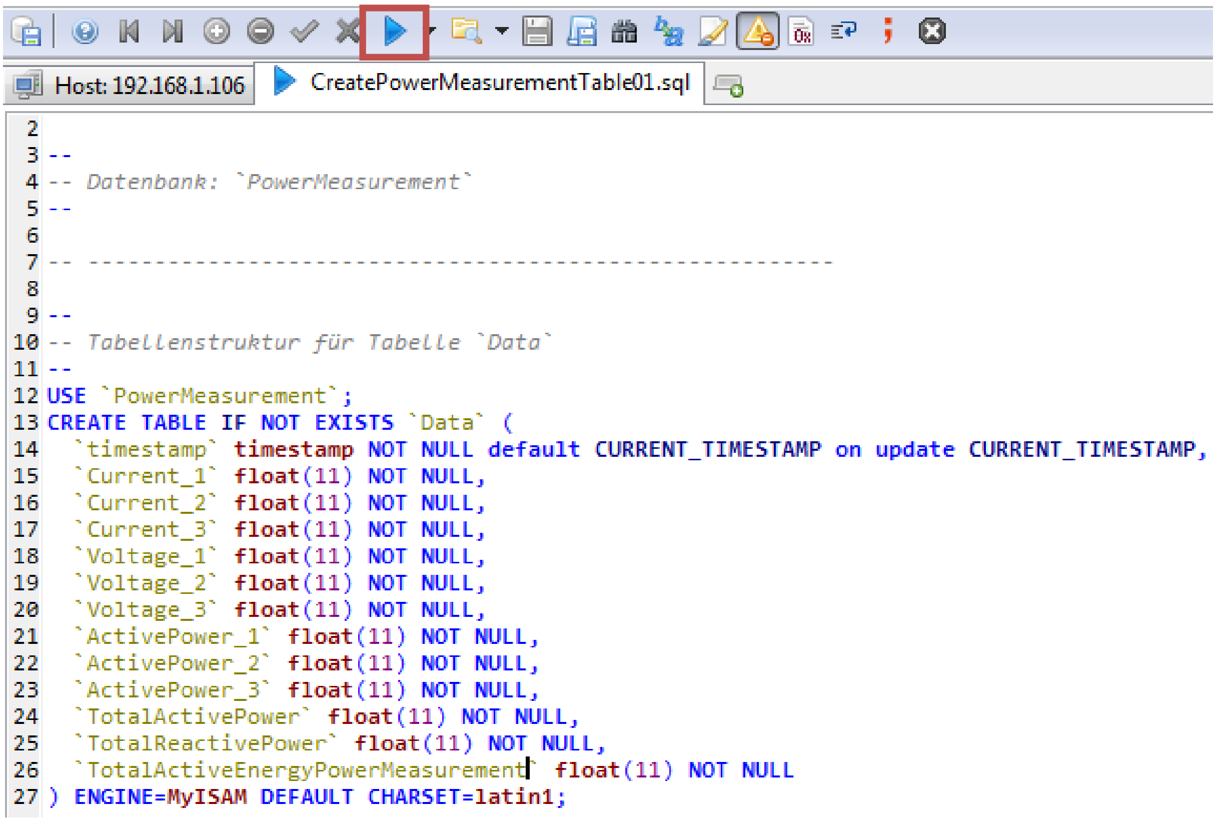
Name	Änderungsdatum	Typ
CreatePowerMeasurementTable01.sql	02.05.2017 11:10	SQL-Datei

Dateityp: SQL-Dateien (*.sql)

Kodierung: Auto-Erkennung (kann fehlschlagen)

Figure 12: Loading the SQL File

6. Next, click the **[▶]** button to execute the previously loaded SQL instructions. The code then automatically creates a new table with the name “Data” in the “PowerMeasurement” database.

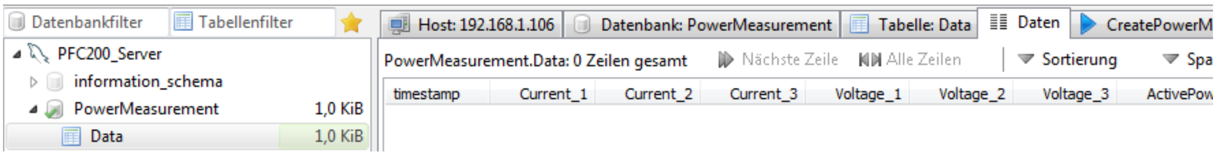


```

--
-- Datenbank: `PowerMeasurement`
--

-- --------------------------------------------------------

--
-- Tabellenstruktur für Tabelle `Data`
--
USE `PowerMeasurement`;
CREATE TABLE IF NOT EXISTS `Data` (
  `timestamp` timestamp NOT NULL default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP,
  `Current_1` float(11) NOT NULL,
  `Current_2` float(11) NOT NULL,
  `Current_3` float(11) NOT NULL,
  `Voltage_1` float(11) NOT NULL,
  `Voltage_2` float(11) NOT NULL,
  `Voltage_3` float(11) NOT NULL,
  `ActivePower_1` float(11) NOT NULL,
  `ActivePower_2` float(11) NOT NULL,
  `ActivePower_3` float(11) NOT NULL,
  `TotalActivePower` float(11) NOT NULL,
  `TotalReactivePower` float(11) NOT NULL,
  `TotalActiveEnergyPowerMeasurement` float(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Figure 13: Executing the Loaded SQL Code

The created table “Data” contains the following columns:

-	Current_1 (_2,_3)	(currents of the three phases)
-	Voltage_1 (_2,_3)	(voltages of the three phases)
-	ActivePower_1 (_2,_3)	(active power of the three phases)
-	TotalActivePower	(total active power)
-	TotalReactivePower	(total reactive power)
-	TotalActiveEnergy	(total active energy)



PowerMeasurement.Data: 0 Zeilen gesamt

timestamp	Current_1	Current_2	Current_3	Voltage_1	Voltage_2	Voltage_3	ActivePow

Figure 14: Created “Data” Table – Section

At this point, the table is empty. In the following section, the attached IEC-61131-3 application is used under ***e!***COCKPIT to write the measurement values of the 3-phase power measurement module in the database.

5.3 IEC-61131-3 Application as MySQLClient

Note

PLC Runtime!
Before the ***e!***COCKPIT application can be operated on the PFC200, the PFC's "PLC Runtime" must be set on ***e!***RUNTIME via WBM.

Please note, the MySQL Server runs on the identical PFC200 as the controller application. Therefore, the following information applies:

Note

FbMySql_Login Module Switching!
If the controller application and the MySQL Server are operating on the same PFC200, "localhost" or "127.0.0.1'" must be entered as IP address in the function block **FbMySql_Login**.

In addition to the `WagoAppPowerMeasurement` library in order to use the 3-phase power measurement module, the attached ***e!***COCKPIT example project *`WagoApp_x.x.x.x_MySQLServer_PowerMeasurement_Example_01.ecp`* also needs the `WagoAppSQL_MySQL`. This library is needed for the communication with the MySQL Server. The following function blocks from the library are used thereby:

- **FbMySql_Login** is needed to establish a connection with the database.

- **FbMySql_Execute** is needed to add data to the "Data" table and for updating.

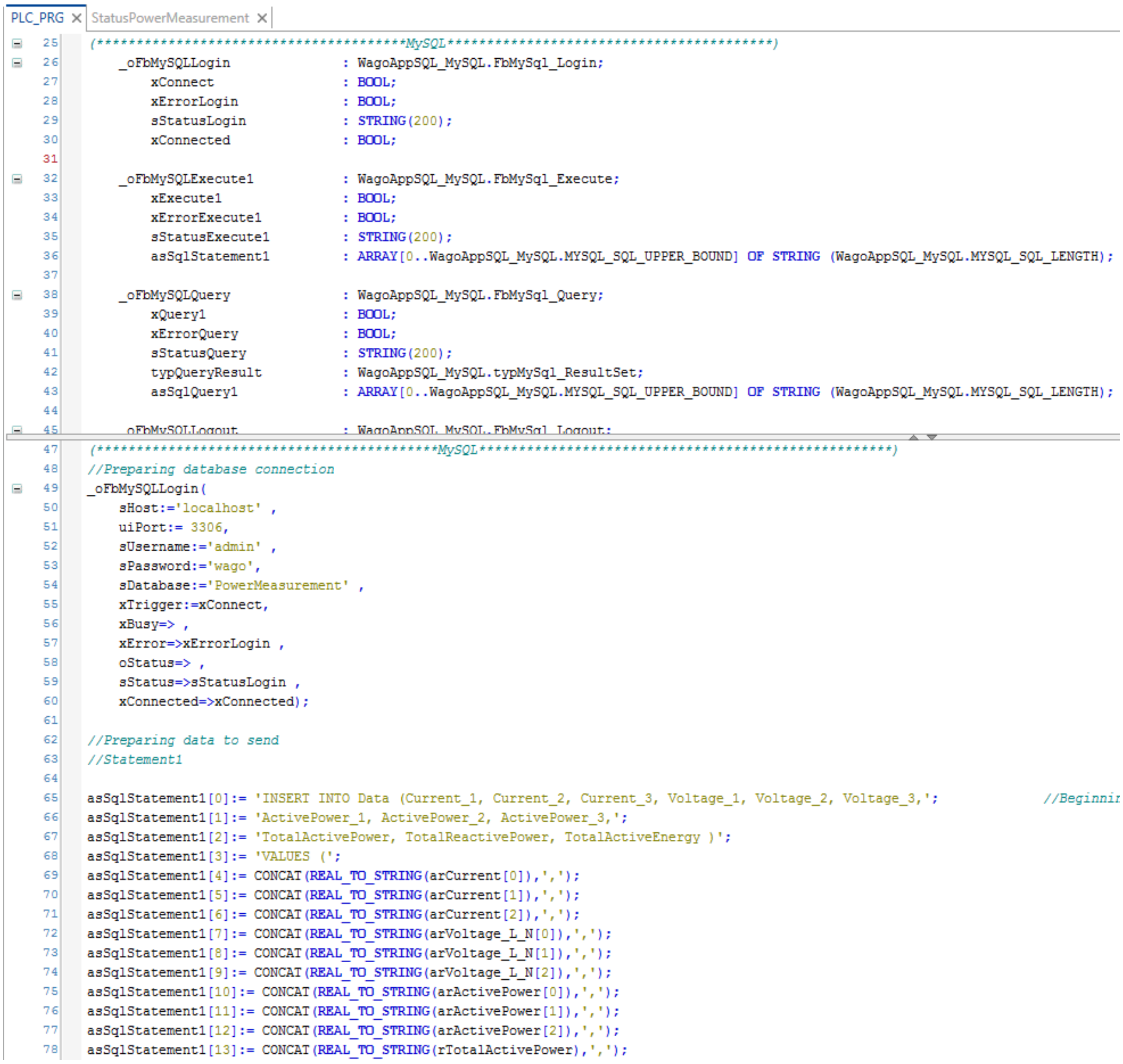
- **FbMySql_Query** is needed to read out entries from the "Data" table.

- **FbMySql_GetStringValue** is needed to convert query data in IEC files.

- **FbMySql_Logout** is needed to terminate the connection to the database.

The example program source code is divided into two sections. The first connection contains module call-ups for parameterizing the 3-phase power measurement module and for the subsequent measurement value recording. However, this Installation note is focused on the MySQL functions, so the parameterization and wiring of these modules are not explained further. If you have any questions or uncertainty, refer to the documentation of the `WagoAppPowerMeasurement`.

The following description deals with the second section of the example program and the MySQL functions it contains.

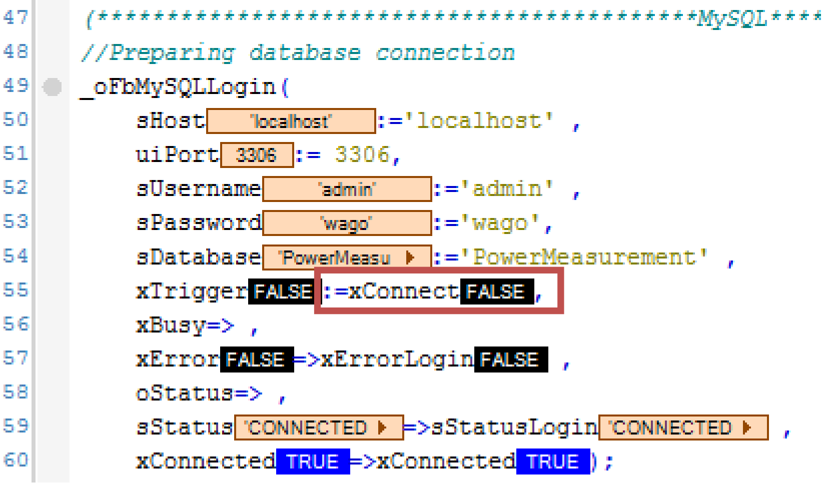


```
(*************************************MySQL****************************************)
    _oFbMySQLLogin              : WagoAppSQL_MySQL.FbMySql_Login;
        xConnect                : BOOL;
        xErrorLogin             : BOOL;
        sStatusLogin            : STRING(200);
        xConnected              : BOOL;

    _oFbMySQLExecute1           : WagoAppSQL_MySQL.FbMySql_Execute;
        xExecute1               : BOOL;
        xErrorExecute1          : BOOL;
        sStatusExecute1         : STRING(200);
        asSqlStatement1         : ARRAY[0..WagoAppSQL_MySQL.MYSQL_SQL_UPPER_BOUND] OF STRING (WagoAppSQL_MySQL.MYSQL_SQL_LENGTH);

    _oFbMySQLQuery              : WagoAppSQL_MySQL.FbMySql_Query;
        xQuery1                 : BOOL;
        xErrorQuery             : BOOL;
        sStatusQuery            : STRING(200);
        typQueryResult          : WagoAppSQL_MySQL.typMySql_ResultSet;
        asSqlQuery1             : ARRAY[0..WagoAppSQL_MySQL.MYSQL_SQL_UPPER_BOUND] OF STRING (WagoAppSQL_MySQL.MYSQL_SQL_LENGTH);

     _oFbMySQLLogout            : WagoAppSQL_MySQL.FbMySql_Logout;
(******************************************MySQL******************************************************)
//Preparing database connection
_oFbMySQLLogin(
    sHost:='localhost' ,
    uiPort:= 3306,
    sUsername:='admin' ,
    sPassword:='wago',
    sDatabase:='PowerMeasurement' ,
    xTrigger:=xConnect,
    xBusy=> ,
    xError=>xErrorLogin ,
    oStatus=> ,
    sStatus=>sStatusLogin ,
    xConnected=>xConnected);

//Preparing data to send
//Statement1

asSqlStatement1[0]:= 'INSERT INTO Data (Current_1, Current_2, Current_3, Voltage_1, Voltage_2, Voltage_3,';          //Beginnir
asSqlStatement1[1]:= 'ActivePower_1, ActivePower_2, ActivePower_3,';
asSqlStatement1[2]:= 'TotalActivePower, TotalReactivePower, TotalActiveEnergy )';
asSqlStatement1[3]:= 'VALUES (';
asSqlStatement1[4]:= CONCAT(REAL_TO_STRING(arCurrent[0]),',');
asSqlStatement1[5]:= CONCAT(REAL_TO_STRING(arCurrent[1]),',');
asSqlStatement1[6]:= CONCAT(REAL_TO_STRING(arCurrent[2]),',');
asSqlStatement1[7]:= CONCAT(REAL_TO_STRING(arVoltage_L_N[0]),',');
asSqlStatement1[8]:= CONCAT(REAL_TO_STRING(arVoltage_L_N[1]),',');
asSqlStatement1[9]:= CONCAT(REAL_TO_STRING(arVoltage_L_N[2]),',');
asSqlStatement1[10]:= CONCAT(REAL_TO_STRING(arActivePower[0]),',');
asSqlStatement1[11]:= CONCAT(REAL_TO_STRING(arActivePower[1]),',');
asSqlStatement1[12]:= CONCAT(REAL_TO_STRING(arActivePower[2]),',');
asSqlStatement1[13]:= CONCAT(REAL_TO_STRING(rTotalActivePower),',');
```

Figure 15: Source Code of the Example Program “Example_MySQL_PowerMeasurement.pro” – Section

1. After a connection to the sever has been successfully established and the application has been started, a connection to the database must be set up. To do so, set “TRUE” for parameter ***“xConnect”*** at the ***“xTrigger”*** input of the module **_oFbMySQLLogin** (double-click, then **[CTRL]**+**[F7]**; see Fig. 15).

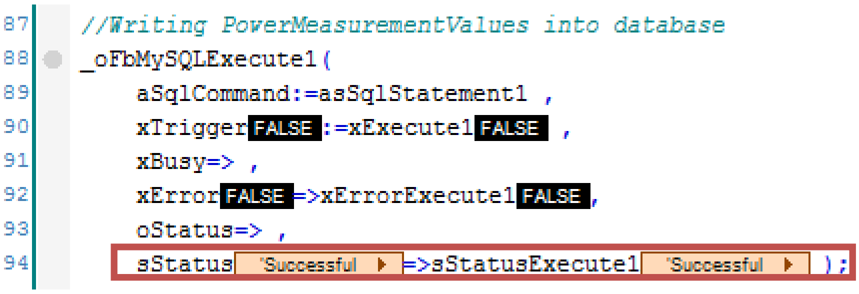
2. Information about the current status of the logon function can be found under the variables ***“sStatusLogin”*** and ***“xConnected”*** (if connection is successful: “TRUE”).



```
(**********************************************MySQL****
//Preparing database connection
_oFbMySQLLogin(
    sHost 'localhost' :='localhost' ,
    uiPort 3306 := 3306,
    sUsername 'admin' :='admin' ,
    sPassword 'wago' :='wago',
    sDatabase 'PowerMeasu :='PowerMeasurement' ,
    xTrigger FALSE :=xConnect FALSE ,
    xBusy=> ,
    xError FALSE =>xErrorLogin FALSE ,
    oStatus=> ,
    sStatus 'CONNECTED =>sStatusLogin 'CONNECTED ,
    xConnected TRUE =>xConnected TRUE );
```

Figure 16: _oFbMySQLLogin Module

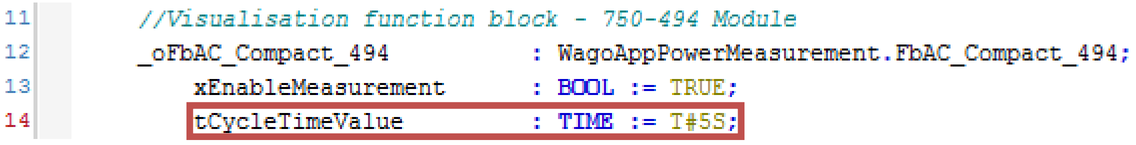
3. After a successful connection to the server has been established, the SQL instruction to transfer the measurement values is executed in a 5-second cycle. The **_oFbMySqlExecute1** module delivers feedback about the data transfer status in the ***"sStatusExecute1"*** string.



```
//Writing PowerMeasurementValues into database
_oFbMySQLExecute1(
    aSqlCommand:=asSqlStatement1 ,
    xTrigger FALSE :=xExecute1 FALSE ,
    xBusy=> ,
    xError FALSE =>xErrorExecute1 FALSE ,
    oStatus=> ,
    sStatus 'Successful =>sStatusExecute1 'Successful );
```

Figure 17: _oFbMySQLExecute1 Module

The pause time between transfers (initially, five seconds) can be set to any value in the parameter ***"tCycleTimeValue"***.



```
    //Visualisation function block - 750-494 Module
    _oFbAC_Compact_494          : WagoAppPowerMeasurement.FbAC_Compact_494;
        xEnableMeasurement      : BOOL := TRUE;
        tCycleTimeValue         : TIME := T#5S;
```

Figure 18: Setting the Transmission Interval (tCycleTimeValue)

Note

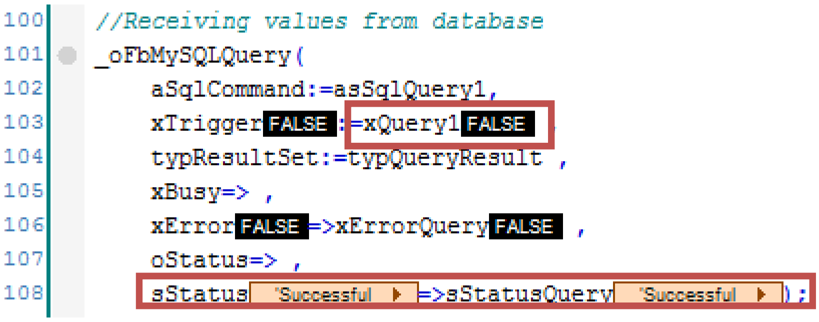
Transmission Interval!
The time for the transmission interval should not be set too low. The resulting high transmission rate would lead to unnecessary loads on the controller and quickly exhaust the memory capacity on the SD card.

4. Next, the optional second SQL instruction (**_oFbMySQLQuery**) can be executed to call up the measurement values saved in the database. To do so, set the signal ***"xQuery1"*** in the module to "TRUE." The status of the execution will then be present in the ***"sStatusQuery1"*** string. The results of the queries are filed in the ***"stTableData.astRow[]***" array.

Note

Maximum Rows of Data to be Called Up!

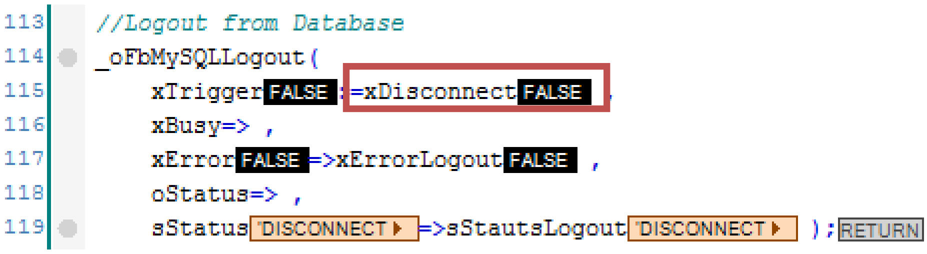
The maximum number of rows from the database table that can be called up can be set with the global variable ***"MAX_ROWS.***" This parameter is set to thirty rows as default, but can be adjusted as required.



```
//Receiving values from database
_oFbMySQLQuery(
    aSqlCommand:=asSqlQuery1,
    xTrigger FALSE :=xQuery1 FALSE ,
    typResultSet:=typQueryResult ,
    xBusy=> ,
    xError FALSE =>xErrorQuery FALSE ,
    oStatus=> ,
    sStatus 'Successful' =>sStatusQuery 'Successful' );
```

Figure 19: _oFbMySQLQuery Module

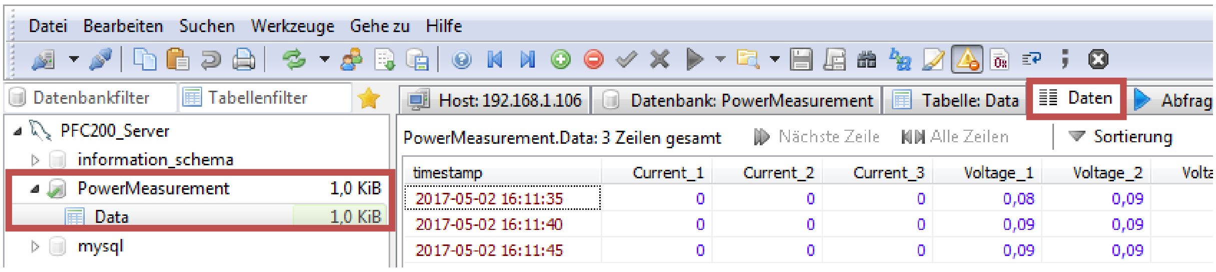
5. Close the connection to the database by forcing the signal ***"xDisconnect"*** to the value "TRUE" in the **_oFbMySQLLogout** module.



```
//Logout from Database
_oFbMySQLLogout(
    xTrigger FALSE :=xDisconnect FALSE ,
    xBusy=> ,
    xError FALSE =>xErrorLogout FALSE ,
    oStatus=> ,
    sStatus 'DISCONNECT' =>sStautsLogout 'DISCONNECT' );RETURN
```

Figure 20: _oFbMySQLLogout Module

6. As an option, the result can then be checked with HeidiSQL. To do so, reconnect to the server; the data that was just written in can be viewed in the "Data" table under the previously created "PowerMeasurement" database.



PowerMeasurement.Data: 3 Zeilen gesamt

timestamp	Current_1	Current_2	Current_3	Voltage_1	Voltage_2	Volta
2017-05-02 16:11:35	0	0	0	0,08	0,09	
2017-05-02 16:11:40	0	0	0	0,09	0,09	
2017-05-02 16:11:45	0	0	0	0,09	0,09	

Figure 21: Overview of the Data in "HeidiSQL" – Section

6 Exchanging the SD Card

Before new data is written in the database, check that sufficient capacity remains on the SD card used.

If the remaining memory capacity is insufficient, first check whether older data can be deleted to create enough capacity. If this is not possible, the SD card can be exchanged as described in the following steps.

1. Insert a new SD card in the PFC200.

2. Restart the controller.

3. Follow the instructions in Sections 4 and 5 again.

WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Telefon: +49 (0) 571/8 87 – 0
Telefax: +49 (0) 571/8 87 – 1 69
E-Mail: info@wago.com
Internet: http://www.wago.com