

SQL ID 109779336

# LSQL Library: Getting Started

Obiettivo di questa guida rapida è quello di guidare l'utente nelle impostazioni preliminari necessarie per utilizzare correttamente la libreria LSQL. Per ulteriori informazioni o approfondimenti si rimanda all'application example "Connecting a S7-1200 PLC / S7-1500 PLC to a SQL Database", ID: 109779336

**SIEMENS**

# Sommario

<b>Introduzione</b>	<b>3</b>
<b>Integrazione nel progetto</b>	<b>4</b>
Requisiti	4
Impostazioni preliminari	4
Lsql_Microsoft	5
AnalyzeTokens	8
DeserializeRows	10
FAQ e problemi comuni	12
1) Impossibile connettersi al server	12
2) AnalyzeToken. Errore 16#8609	13
3) Impossibile inserire dati su colonne di tipo "nchar"	14
4) Corretta interpretazione del tipo "DateTime" da database SQL	15
5) Stored Procedure. Esecuzione negata	16

SQL

# Introduzione

## SW utilizzati

1. Tia Portal V18
2. Microsoft SQL Server Management Studio <https://docs.microsoft.com/en-us/sql/ssms/download-SQL-Server-management-studio-ssms>
3. Libreria LSQL (Esempio applicativo) <https://support.industry.siemens.com/cs/document/109779336/connecting-a-s7-1200-plc-s7-1500-plc-to-a-sql-database?dti=0&lc=en-WW>

L'installazione del Server Management Studio (4), qualora il server SQL sia già operativo, non è obbligatoria ma consente di avere accesso al database SQL e semplificare eventuali operazioni di debug.

Lo scopo di questo documento non vuole essere sostituire il manuale fornito insieme all'esempio applicativo (link 3), del quale si consiglia in ogni caso la lettura, ma semplicemente guidare l'utente nelle impostazioni preliminari.

## LIBRERIA LSQL

# Integrazione nel progetto

## Requisiti

- S7-1500 Fw V2.5 / S7-1200 Fw V4.4
- Microsoft SQL Server già configurato<sup>1</sup>
- S7-1500 e Server devono essere nella stessa subnet
- Porta 1433 abilitata nel firewall

Per i requisiti completi ed eventuali limitazioni si rimanda alla guida dell'esempio applicativo scaricabile al link indicato in Introduzione.

## Impostazioni preliminari

Dopo aver importato l'intero contenuto della libreria nel proprio progetto è necessario richiamare all'interno dell' OB1 e parametrizzare i seguenti tre blocchi:

- **FB "LSql\_Microsoft"**. Necessario per gestire la connessione al database.
- **FB "AnalyzeTokens"**. Elabora la risposta del server SQL in modo da eseguire i comandi. La "risposta" è suddivisa in tre parti: TDS header, packet data, DONE token. Per ulteriori informazioni si rimanda sempre all'esempio applicativo.
- **FB "DeserializeRows"**. Elabora i dati serializzati contenuti nelle righe del database e li memorizza in funzione del tipo di dati usato in una struttura che deve essere adattata al singolo use-case.

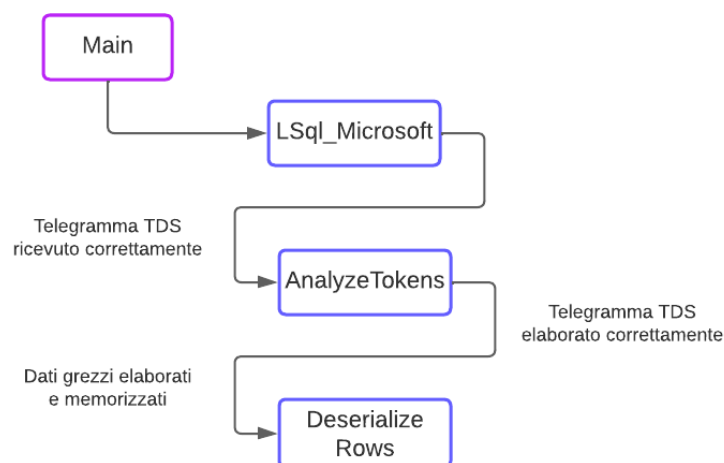


Figura 1 - Ordine di call delle FB

<sup>1</sup> Se necessario, l'esempio applicativo (vedi Introduzione) contiene una guida base per installare ed inizializzare un server SQL. L'argomento non verrà perciò trattato qui.

# Lsql\_Microsoft

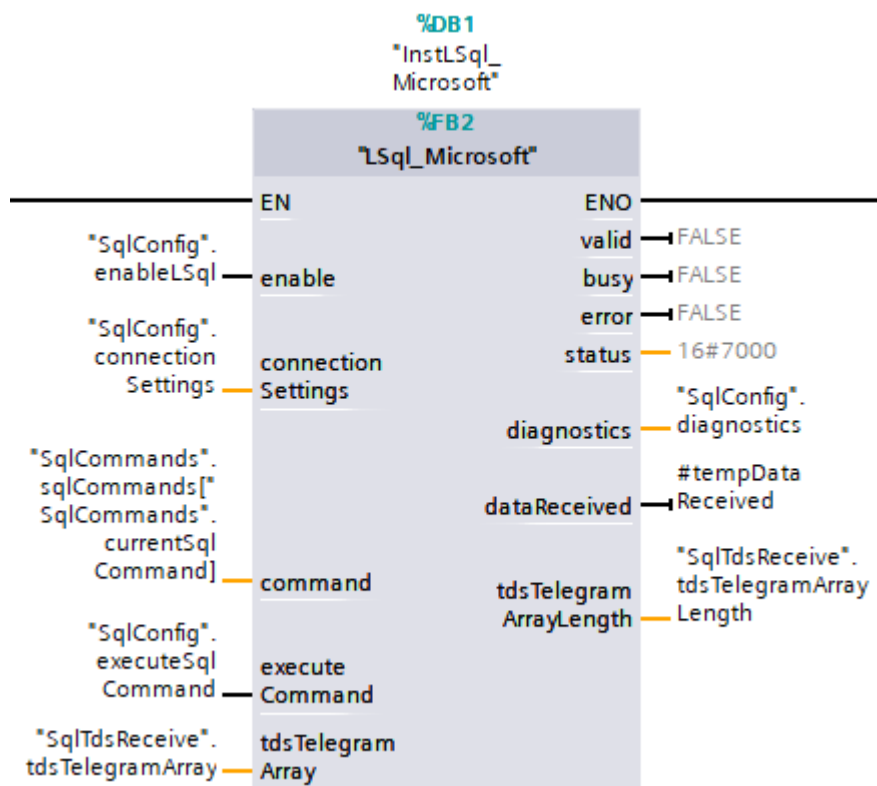


Figura 2 - Lsql\_Microsoft

Per informazioni dettagliate a riguardo si rimanda sempre all'esempio applicativo.

Analizzando brevemente gli input troviamo:

- **Enable:** abilita e lancia la comunicazione con il database SQL. Se la comunicazione viene stabilita correttamente gli output "Valid" e "Busy" rimangono alzati
- **Execute Command:** lancia il comando SQL richiesto, l'FB legge il fronte di salita.
- **Connection settings:** di tipo "Lsql\_typeConnectionSettings" contiene le informazioni necessarie per connettersi al server SQL, vediamole nel dettaglio.

Static		
connectionSettings	"LSql_typeConnectionSettings"	
interfaceSettings	TCON_IP_v4	
InterfaceId	HW_ANY	64
ID	CONN_OUC	16#10
ConnectionType	Byte	16#0B
ActiveEstablish...	Bool	true
RemoteAddress	IP_V4	
ADDR	Array[1..4] of Byte	
ADDR[1]	Byte	192
ADDR[2]	Byte	168
ADDR[3]	Byte	43
ADDR[4]	Byte	134
RemotePort	UInt	1433
LocalPort	UInt	0
loginInformation	"LSql_typeLoginInformation"	
hostName	String	"
userName	String	'SQL_S71500'
password	String	'[REDACTED]'
appName	String	"
serverName	String	'SQLEXPRESS'
libraryName	String	"
language	String	"
databaseName	String	'SQLFromPLC'
sspi	String	"
attachDbfile	String	"
changePassword	String	"

Figura 3 – ConnectionSettings

1) Hardware identifier dell'interfaccia connessa alla rete del server SQL.

Name	Type	Hardware identi.	Used by
Local~Configuration	Hw_SubModule	33	PLC_1
Local~Display	Hw_SubModule	54	PLC_1
Local~Exec	Hw_SubModule	52	PLC_1
Local	Hw_SubModule	49	PLC_1
Local~PROFINET_interface_1	Hw_Interface	64	PLC_1
Local~PROFINET_interface_2	Hw_Interface	72	PLC_1

Figura 4 - Hw Identifier

- 2) Indirizzo IP del server SQL.
- 3) Username e Password. Informazioni di login per il server SQL, l'utente indicato deve esistere sul database SQL e deve avere i diritti di lettura e scrittura.

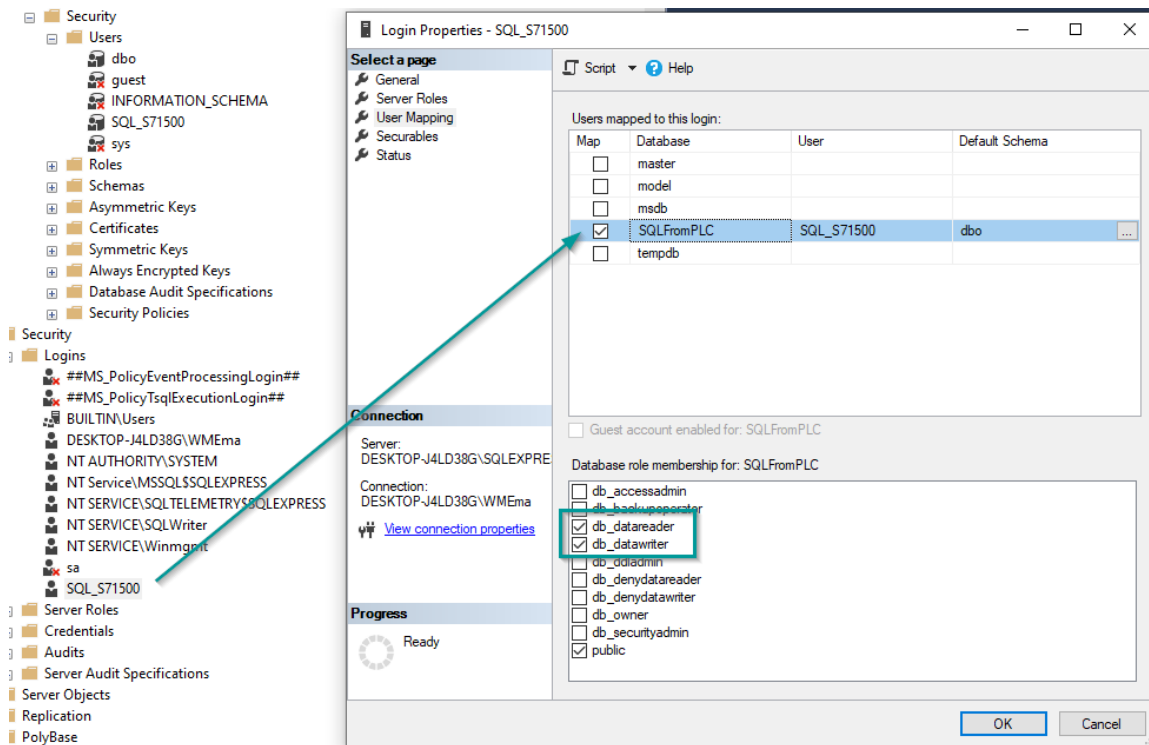


Figura 5 - Microsoft SQL Server Management Studio. Informazioni Utente

- 4) Nome del Server
- 5) Nome del Database

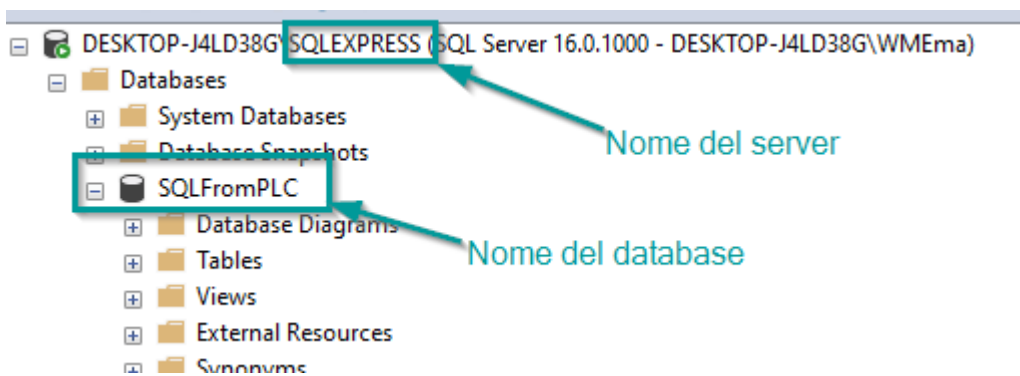


Figura 6 - Microsoft SQL Server Management Studio. Server e Database

- **Command:** di tipo stringa, nell'esempio applicativo viene utilizzato un array di stringhe contenente alcuni esempi di comandi. L'utilizzo di un array non è obbligatorio.

sqlCommands	Array[0..25] of String
sqlCommands[0]	'insert into PLCDATA_1 values (7,8, 9)'
sqlCommands[1]	'Update PLCDATA_1 set IntegerValue1 = 7, IntegerValue2 = 7, IntegerValue3 = 7 where IntegerVal...
sqlCommands[2]	'Select IntegerValue1 from PLCDATA_1'
sqlCommands[3]	'Select Fruit from PLCDATA_2 where Amount != 0'
sqlCommands[4]	'Select Fruit from PLCDATA_2 where color = \$'red\$''
sqlCommands[5]	'insert into PLCDATA_3 values (7, \$'2020-01-01 10:23:24.125\$)'
sqlCommands[6]	'SELECT TOP (3) * FROM PLCDATA_2 ORDER BY Fruit'
sqlCommands[7]	'execute myProcedureIn @input1=30, @input2=6, @input3=84'
sqlCommands[8]	'execute myProcedureSelect'
sqlCommands[9]	'declare @myout1 char(30), @myout2 int execute myProcedureInOut @in1=\$'pear\$', @in2=\$'yell...
sqlCommands[10]	'INSERT INTO [dbo].[errors] ((Text)) VALUES (\$'remotetest\$)'
sqlCommands[11]	'SELECT TOP (10) ValueReal1, ValueReal2, ValueReal3, ValueReal4, ValueReal5, ValueReal6, ValueR...
sqlCommands[12]	'SELECT * FROM KompressionLog ORDER BY StentNr'
sqlCommands[13]	'execute SelectProc1'
sqlCommands[14]	'insert into TestDateTime values (\$'01.01.1990 12:00:00\$)'
sqlCommands[15]	'exec SetDateTime @Input1 = \$'07.30.1990 12:00:00\$''
sqlCommands[16]	'SELECT TOP (100) MesId, MesId2, MesId3 FROM TestJulian'
sqlCommands[17]	'SELECT TOP (1) DateTime FROM TestDateTime'
sqlCommands[18]	'SELECT TOP 2 * FROM FunctionTest ORDER BY ID'
sqlCommands[19]	'INSERT INTO [dbo].[FunctionTest] DEFAULT VALUES'
sqlCommands[20]	'use [SQLFromPLC] declare @myout1 bit execute NachtragDatensaeetze @inDateiPfad= \$'C:\Progra...
sqlCommands[21]	'UPDATE [dbo].[KompressionLog] SET LotNumber = \$'Test234\$' WHERE KammerId = \$'3\$''
sqlCommands[22]	'insert into PLCDATA_2 (Amount, Color, Fruit, Fresh, Country, CountryCode, City, Taste) values (10...

Figura 7 – SqlCommands

**N.B:** Lo scopo di questa guida non include una spiegazione dei comandi, in generale si consiglia almeno una conoscenza base del linguaggio SQL. Nell'esempio applicativo vengono illustrati alcuni dei comandi base.

## AnalyzeTokens

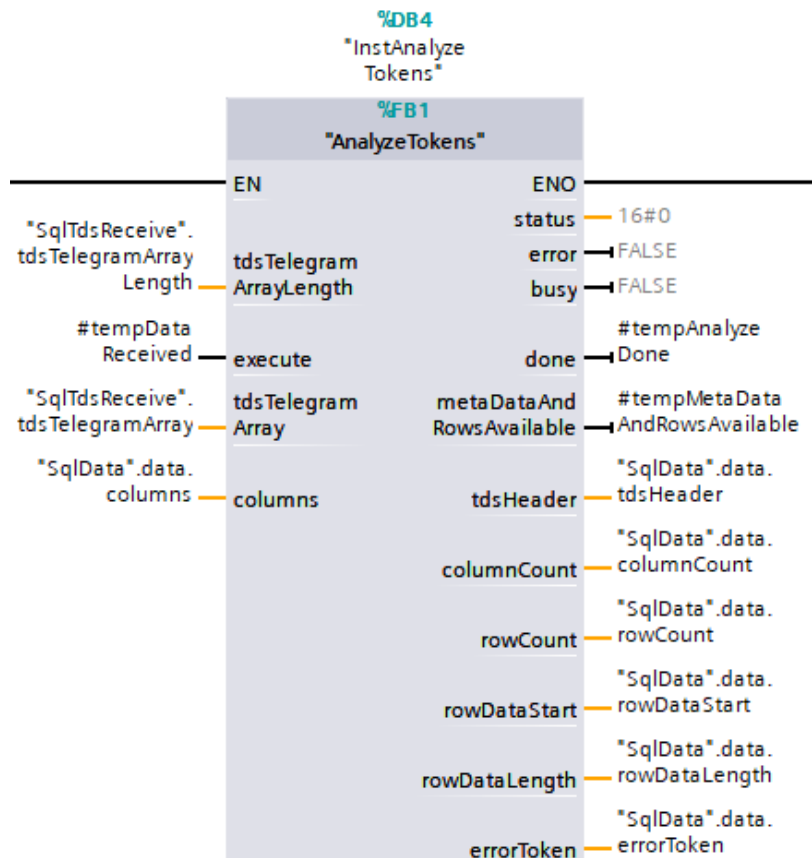


Figura 8 – AnalyzeTokens



Senza approfondire il funzionamento dell'istruzione (fare riferimento all'esempio applicativo per maggiori informazioni) sottolineo l'importanza dello "errorToken" grazie al quale, in caso di errore nell'esecuzione di un comando, è possibile ricevere la risposta di diagnostica del server SQL.

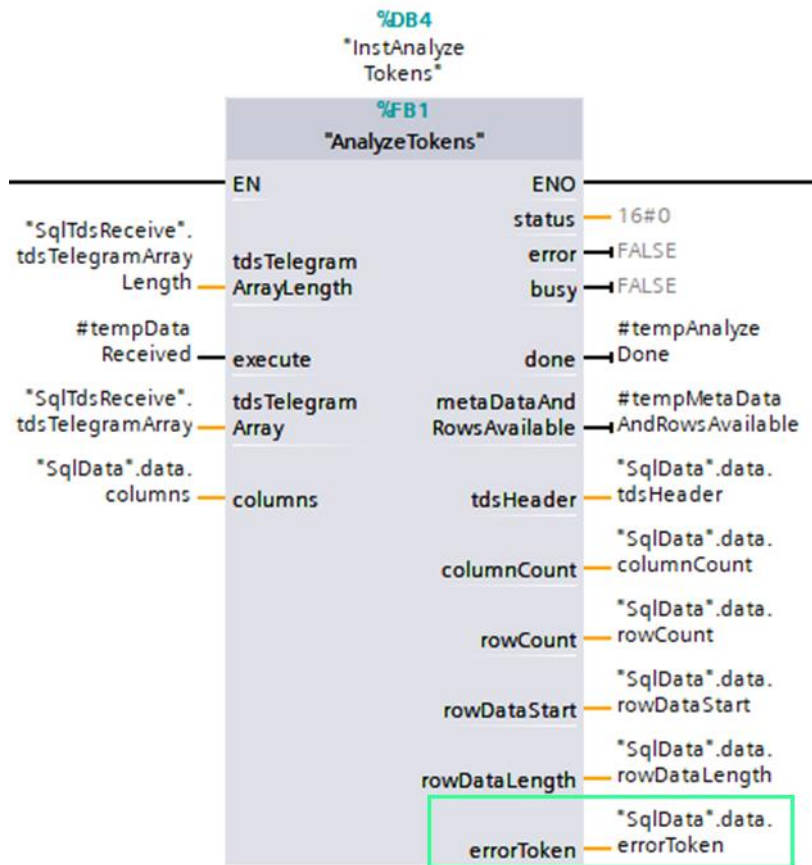


Figura 9 - ErrorToken

rows	Array[0..*LSQL_NU...			
errorToken	*LSql_typeErrorTok...			
TokenLength	UInt	0		122
SQLErrorNumber	UDInt	0		207
State	Byte	16#0		16#01
Class	USInt	0		16
ErrorMessageL...	UInt	0		28
ErrorMessage	String	"		'Invalid column name '\$Apple\$'.'
ServerNameLe...	USInt	0		26
ServerName	String	"		'DESKTOP-J4LD38GISQLEXPRESS'
ProcNameLeng...	USInt	0		0
ProcName	String	"		"
LineNumber	UDInt	0		1

Figura 10 - ErrorToken. Esempio

# DeserializeRows

Per poter utilizzare correttamente il comando di "SELECT" ovvero per poter leggere i dati presenti nel database SQL e memorizzarli sul PLC è necessario adattare alcune componenti dell'esempio applicativo in funzione del nostro database.

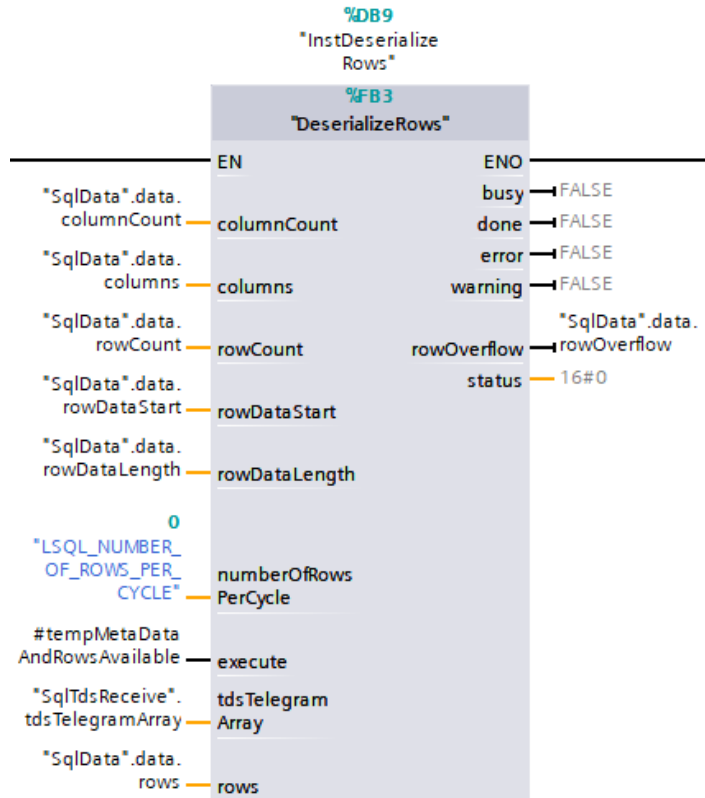


Figura 11 - DeserializeRows

I dati ricevuti vengono salvati in "SqlData.data.rows", quest'ultimo va configurato in modo da essere allineato al database che vogliamo leggere. Questo è fattibile modificando l'udt "typeTokenRow".

typeTokenRow			
Name	Data type	Default value	Accessible f...
Amount	Int	0	<input type="checkbox"/>
Color	String[30]	"	<input type="checkbox"/>
Fruit	String[50]	"	<input type="checkbox"/>
Fresh	Bool	false	<input type="checkbox"/>
Country	String	"	<input type="checkbox"/>
CountryCode	String[5]	"	<input type="checkbox"/>
City	String	"	<input type="checkbox"/>
Taste	String[20]	"	<input type="checkbox"/>

Figura 12 - PLC Data Type

dbo.PLCDATA_2	
Columns	
Amount	(int, null)
Color	(nchar(30), null)
Fruit	(nchar(50), null)
Fresh	(bit, null)
Country	(ntext, null)
CountryCode	(nchar(5), null)
City	(ntext, null)
Taste	(nchar(20), null)

Figura 13 - Tabella SQL

A questo punto, aprendo "DeserializeRows" sarà possibile individuare alcuni esempi per leggere stringhe, interi e booleani, la libreria fornisce anche le funzioni necessarie per elaborare correttamente tipi di dato DWORD.

Partendo dagli esempi sarà quindi sufficiente correggere il nome della colonna in funzione del nostro database e duplicare/eliminare i tipi di dati presenti.

Nelle figure seguenti vengono evidenziati i campi che devono essere personalizzati, viene inoltre riportato un esempio di conversione a "REAL" utilizzando la funzione "LGF\_MergeBytesToDWord" fornita dalla libreria LSQ.

```

IF #columns[#tempColumnLoopCounter].columnName = 'Amount' THEN
  REGION example INT
    //merge next two BYTE to WORD and then convert to INT
    #rows[#tempRowLoopCounter].Amount := WORD_TO_INT("LGF_MergeBytesToWord"(byte0 := #tdsTelegramArray[#statByteAddressCounter], byte1 := #tdsTele
  END_REGION example Int

ELSIF #columns[#tempColumnLoopCounter].columnName = 'Fresh' THEN
  REGION example Bool
    //convert BYTE to BOOL
    #rows[#tempRowLoopCounter].Fresh := BYTE_TO_BOOL(#tdsTelegramArray[#statByteAddressCounter]);
  END_REGION example Bool

ELSIF #columns[#tempColumnLoopCounter].columnName = 'Color' THEN
  REGION example String
    //move Bytes to temporary array
    MOVE_BLK(IN := #tdsTelegramArray[#statByteAddressCounter],
    COUNT := UDINT_TO_UINT(#tempValueLength),
    OUT => #tempByteCharArray[0]);
    //convert BYTE array to STRING
    #rows[#tempRowLoopCounter].Color := "LSql_BytesToString"(columnType := #columns[#tempColumnLoopCounter].columnType, byteArray := #tempByteChara
  END_REGION example String

```

Figura 14 – DeserializeRows. Integer, Bool e String

```

ELSIF #columns[#tempColumnLoopCounter].columnName = 'Real' THEN
  REGION example Real
    //convert BYTE array to DWORD
    #tReal := "LGF_MergeBytesToDWord"(byte0 := #tdsTelegramArray[#statByteAddressCounter], byte1 := #tdsTelegramArray[#statByteAddressCounter + 1],
    byte2 := #tdsTelegramArray[#statByteAddressCounter + 2], byte3 := #tdsTelegramArray[#statByteAddressCounter + 3]);
    //convert DWORD to REAL
    #rows[#tempRowLoopCounter].Test := DWORD_TO_REAL(#tReal);
  END_REGION example Real

```

Figura 15 - DeserializeRows. Real

A questo punto la configurazione è completa e, se tutto è stato effettuato correttamente, è possibile comunicare con il database SQL.

# FAQ e problemi comuni

## 1) Impossibile connettersi al server

Qualora la connessione con il server non sia possibile, verificare che il server SQL sia raggiungibile e/o le informazioni di login siano corrette. Un possibile metodo è utilizzare un file “.udl” su un PC connesso alla stessa rete del PLC e del server SQL.

1. Creare un nuovo file di testo sul pc e cambiare l'estensione in .udl

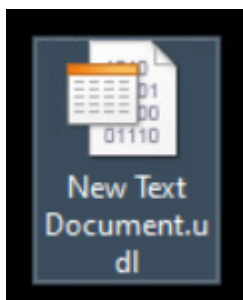


Figura 16 - File .udl

2. Aprire il file e, nella tab “Connessione”, inserire gli stessi dati inseriti in “ConnectionSettings” e premere “Verifica connessione”

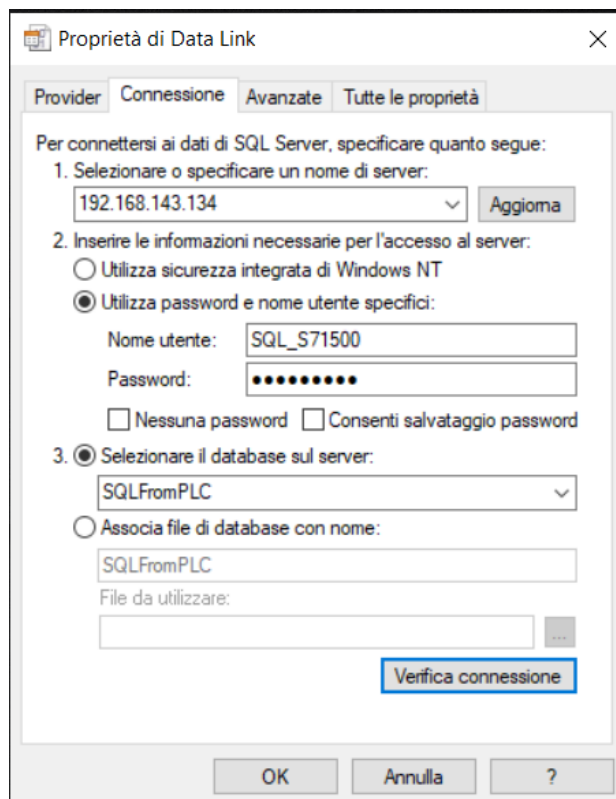


Figura 17 - Test Connessione

3. Se neanche tramite questo test è possibile raggiungere il server il problema non è legato al PLC ma bisogna verificare le informazioni inserite e/o la configurazione di rete.

Alcuni degli errori e delle cause possibili sono riportati in seguito.

ERRORE	POSSIBILE CAUSA
<b>SERVER SQL INESISTENTE O ACCESSO NEGATO</b>	Verificare che il server sia sulla stessa sotto rete del PLC/PC. Verificare che non ci siano firewall attivi e/o che la porta 1433 sia sbloccata
<b>IMPOSSIBILE APRIRE IL DATABASE "XXXXX" RICHIESTO DALL'ACCOUNT DI ACCESSO. ACCESSO NON RIUSCITO.</b>	Verificare che i dati di login siano corretti (es. verificare nome del database). Verificare che, lato server SQL, l'utente sia effettivamente abilitato per accedere al database richiesto.

## 2) AnalyzeToken. Errore 16#8609

Un errore 16#8609 nell'istruzione AnalyzeToken indica che è stato individuato un token di errore nella risposta dal database. Fare riferimento ai dati contenuti nell'error Token.



Figura 18 – ErrorToken

Errori tipici possono essere:

ERRORE	POSSIBILE CAUSA
<b>SINTASSI NON CORRETTA IN PROSSIMITÀ DI ...</b>	Nella query inviata al database è presente un errore di sintassi. Verificare la stringa inviata.
<b>IL NOME DI COLONNA ... NON È VALIDO</b>	Verificare di non avere indicato il nome di una colonna errata. Se non è stato specificato nessun nome di colonna verificare che il tipo di dato che stiamo inviando corrisponda a quello atteso dalla tabella. (vedi FAQ 3 per esempio relativo agli nchar)

### 3) Impossibile inserire dati su colonne di tipo "nchar"

Qualora la tabella sulla quale si vuole inviare una query abbia una colonna il cui tipo di dato è "nchar" è importante fare attenzione al modo in cui viene inviato il valore.

Column Name	Data Type	Allow Nulls
PrimaryKey	nchar(10)	<input type="checkbox"/>
test	int	<input checked="" type="checkbox"/>

Figura 19 - Esempio di tabella con nchar

SQL, infatti, riconosce come valori di tipo nchar solo quei valori indicati tra singoli apici. Il mancato inserimento degli apici restituisce un errore di "nome colonna non valido" (vedi Figura 20).

errorToken	*LSql_typeErrorTok...			
TokenLength	UInt	0	146	
SQLErrorNumber	UDInt	0	207	
State	Byte	16#0	16#01	
Class	USInt	0	16	
ErrorMessageL...	UInt	0	40	
ErrorMessage	String	"	"	'Il nome di colonna '\$pippo\$' non è valido.'
ServerNameLe...	USInt	0	26	
ServerName	String	"	"	'DESKTOP-48IG9LE\SQLEXPRESS'
ProcNameLeng...	USInt	0	0	
ProcName	String	"	"	"
LineNumber	UDInt	0	1	

Figura 20 - Errore nchar

SqlCommands				
	Nome	Tipo di dati	Valore di avvio	Valore di controllo
1	Static			
2	currentSqlCommand	USInt	6	0
3	sqlCommands	Array[0..25] of String		
4	sqlCommands[0]	String	'insert into PLC...	'insert into testP values (pippo, 100)'

Figura 21 - Sintassi errata

Per poter inserire questo carattere nella stringa gestita dal PLC è obbligatorio prederlo con il simbolo \$.

In Figura 22 troverete un esempio di sintassi corretta per inserire un valore nchar in una tabella tramite la libreria LSQL.

SqlCommands				
	Nome	Tipo di dati	Valore di avvio	Valore di controllo
1	Static			
2	currentSqlCommand	USInt	6	0
3	sqlCommands	Array[0..25] of String		
4	sqlCommands[0]	String	'insert into PLC...	'insert into testP values ('\$pippo\$', 100)'

Figura 22 - Sintassi corretta

## 4) Corretta interpretazione del tipo "DateTime" da database SQL

Quando si cerca di leggere (comando SELECT) una colonna di tipo "DateTime" da database SQL è importante fare attenzione al modo in cui viene elaborata. Il protocollo TDS rappresenta infatti i dati "datetime" in modo diverso dal "Date\_And\_Time" Siemens e richiede quindi una conversione da implementare "manualmente" all'interno dell'istruzione "DeserializeRows"

Il protocollo TDS è definito come segue<sup>2</sup>:

- 4-byte che rappresentano il numero di giorni trascorsi dal 1° gennaio 1900.
- 4-byte che rappresentano il numero di trecentesimi di secondo (300 "impulsi" ogni secondo) trascorsi dalle 12:00 dello stesso giorno.

Lato Siemens invece, i tipi di dato "Date" e "Time\_Of\_Day" sono definiti come segue (vedi Help Online di Tia):

- Date: numero di giorni trascorsi dal 1° gennaio 1990
- TOD: numero di millisecondi trascorsi dalle 00:00:00 dello stesso giorno.

È quindi necessario inserire un offset di 90 anni per quanto riguarda la data e convertire il tempo da "trecentesimi di secondo" a millisecondi. A questo punto sarà possibile concatenare "Date" e "TOD" in un'unica variabile di tipo "Date\_and\_Time".

Nell'immagine seguente è riportato un esempio di conversione.

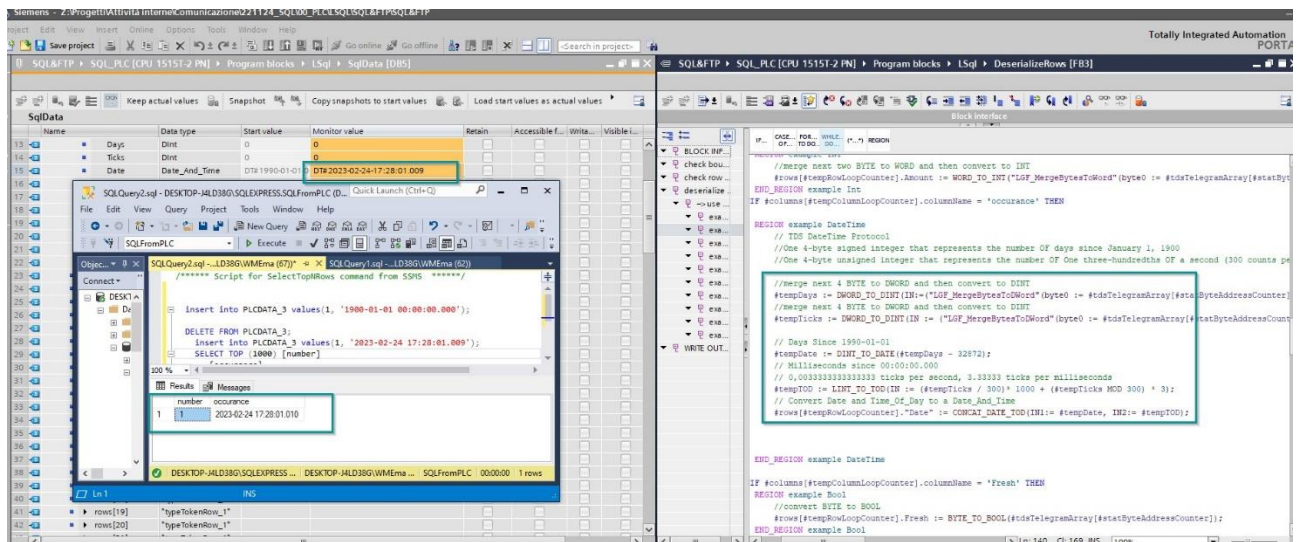


Figura 23 - Conversione DateTime

**N.B.** L'esempio riportato in figura non è da considerarsi parte della libreria ufficiale.

<sup>2</sup> [\[MS-TDS\]: Dates and Times | Microsoft Learn](#)





**Siemens S.p.A.**  
**Via Vipiteno 4**  
**20128 Milano**  
**Tel. 02 243 1**  
**Mail: [infodesk.it@siemens.com](mailto:infodesk.it@siemens.com)**

Con riserva di modifiche e salvo errori.

Il presente documento contiene solo descrizioni generali o informazioni su caratteristiche non sempre applicabili, nella forma descritta, al caso concreto o che possono cambiare a seguito di un ulteriore sviluppo dei prodotti. Le caratteristiche desiderate sono vincolanti solo se espressamente concordate all'atto di stipula del contratto.

Tutte le denominazioni dei prodotti possono essere marchi oppure denominazioni di prodotti della Siemens AG o di altre ditte fornitrici, il cui utilizzo da parte di terzi per propri scopi può violare il diritto dei proprietari.