

Тема: Приклад застосування функції SQL Delta HMI

Застосовувана модель	Серія DOP-100, DOP-300S
Ключове слово	SQL

**Дистриб'ютор в Україні****Україна: ТОВ "Системи реального часу - Україна"**www.delta-electronics.com.ua

вул. Святослава Хороброго, 29-А, 49001, м.Дніпро

Пошта: sales@rts.ua

ТЕЛ : +38 0562 392223 / +38 068 2392223

Зміст

1. #	ВСТУП	3 #
2. #	СЦЕНАРІЙ.....	3 #
3. #	КОНФІГУРАЦІЯ АПАРАТНОГО/ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	3 #
4. #	ОПЕРАЦІЙНІ КРОКИ.....	5 #
5. #	ФУНКЦІОНАЛЬНЕ ТЕСТУВАННЯ	20 #
6. #	ОПИС ФУНКЦІЇ	23 #
7. #	ВИСНОВОК	25 #

1. Вступ

Повна назва SQL – структурована мова запитів. Це мова програмування та запитів до бази даних, яка використовується для доступу до даних, запитів, оновлення та керування реляційними системами баз даних. Глобальна база даних SQL включає Oracle, DB2, Sybase, SQL Server, MySQL, MS Access тощо.

Наразі Delta HMI підтримує SQL Server та MySQL.

2. Сценарій

Його використовують на текстильних та пакувальних фабриках для виробництва різних стилів, що відповідають індивідуальним потребам. Тому параметри обробки, що відповідають цим різним стилям, повинні бути зафіксовані, а історія виробництва або пов'язана з нею інформація повинна бути передана до заводської системи (ERP, MES тощо).

Такі дані, як параметри обробки, історія виробництва тощо, є дуже великими. Наразі існування бази даних SQL є необхідним, а людський комп'ютер надає API, пов'язані з SQL, для отримання інформації SQL, а також може передавати дані до SQL.

У цій статті SQL Server використовується як SQL-сервер, а людина-комп'ютер як клієнт. Завдяки концепції бази даних дані створюються та зчитуються.

3. Конфігурація апаратного/програмного забезпечення

Апаратне забезпечення :

№	Пристрій	Тип	Кількість
1	ПК	Windows 10 Enterprise версії 22H2	1
2	HMI	DOP-3S07S3E2	
3	Кабель	RJ45	1

Програмне забезпечення:

№	Пристрій	Тип	Кількість
1	Програмне забезпечення для редагування HMI	DIAScreen	1

Інформація про програмне забезпечення, пов'язане з SQL:

№	Пристрій	Тип	Зауваження
1	SQL Server	Потрібна версія 2014 або вище	Немає даних
2	MySQL	Потрібна версія 8.0.39 або вище	Немає даних
3	CCMC	Потрібна версія 18.4 або вище	Щоб встановити параметри, пов'язані з SQL (ім'я бази даних, дані для входу тощо)

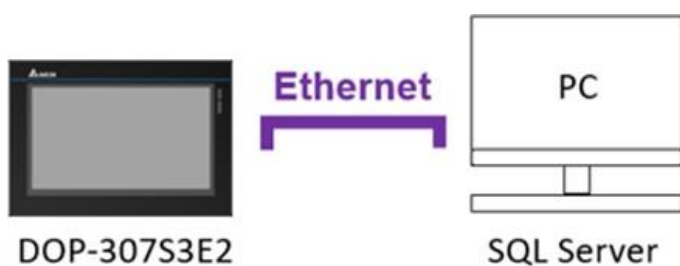
Версія програмного забезпечення HMI: DIAScreen 1.4.3.45

Версія прошивки HMI: Ver-1.0132.8.1

Завантаження SQL Server: <https://www.microsoft.com/zh-tw/download/details.aspx?id=42299>

Завантаження SSMS: <https://learn.microsoft.com/zh-tw/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16>

Архітектура:

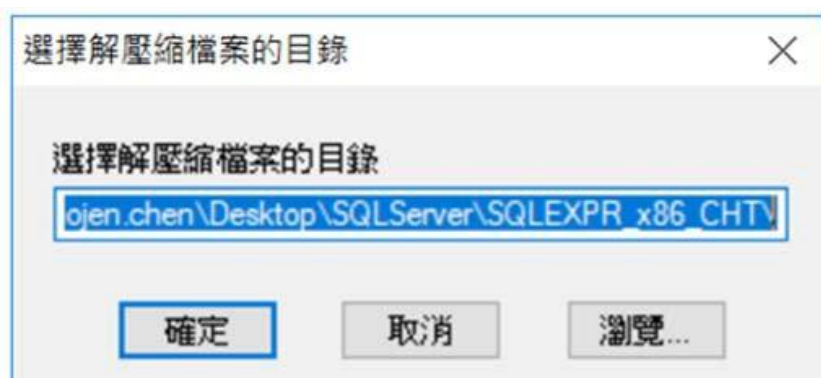


4. Етапи виконання

У цій статті показано, як HMI підключається до комп'ютера (SQL Server) через API, пов'язані з SQL, та отримує дані бази даних.

● Встановлення програмного забезпечення

1. Створіть SQL-сервер.
2. Виконайте SQLEXPRESS_x86_CHT.exe.



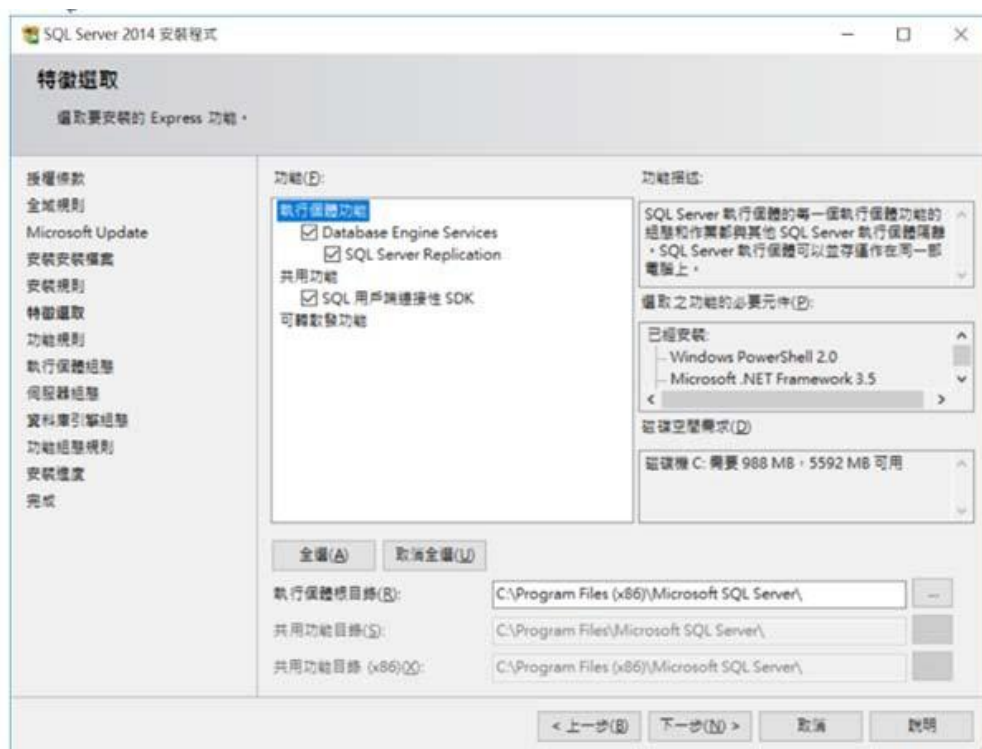
3. З'явиться екран встановлення, виберіть елемент у червоній рамці.



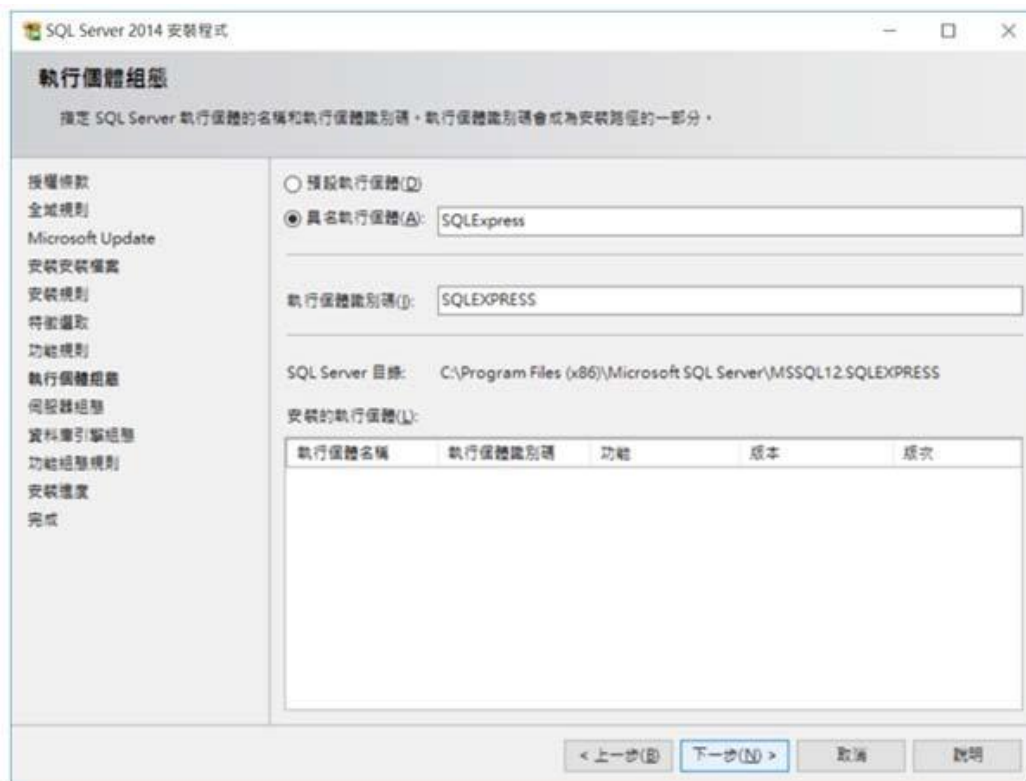
4. Поставте позначку в червоному полі та перейдіть до наступного кроку.



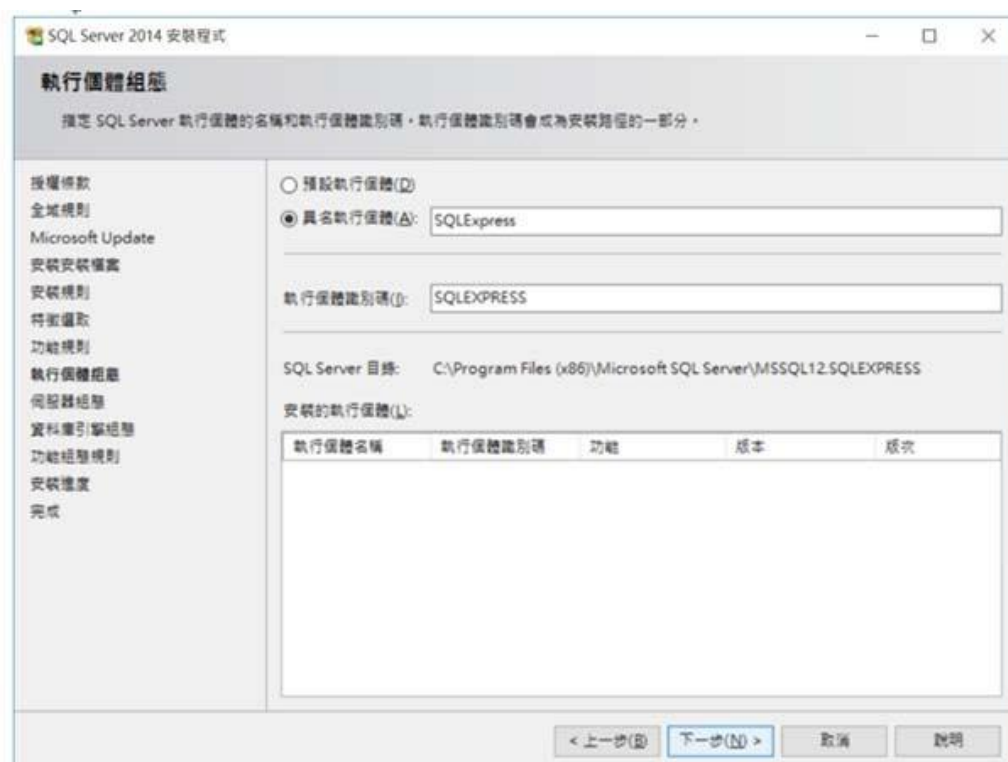
5. Перейдіть до наступного кроку, не змінюючи опцій.



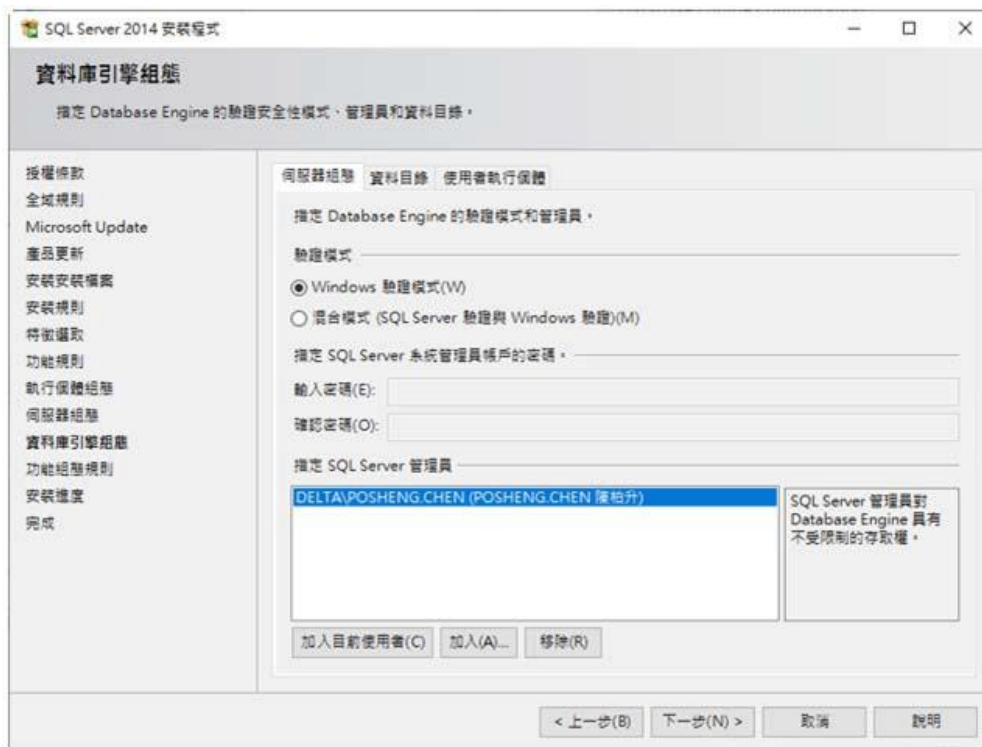
6. Визначте ім'я SQL Server та перейдіть до наступного кроку.



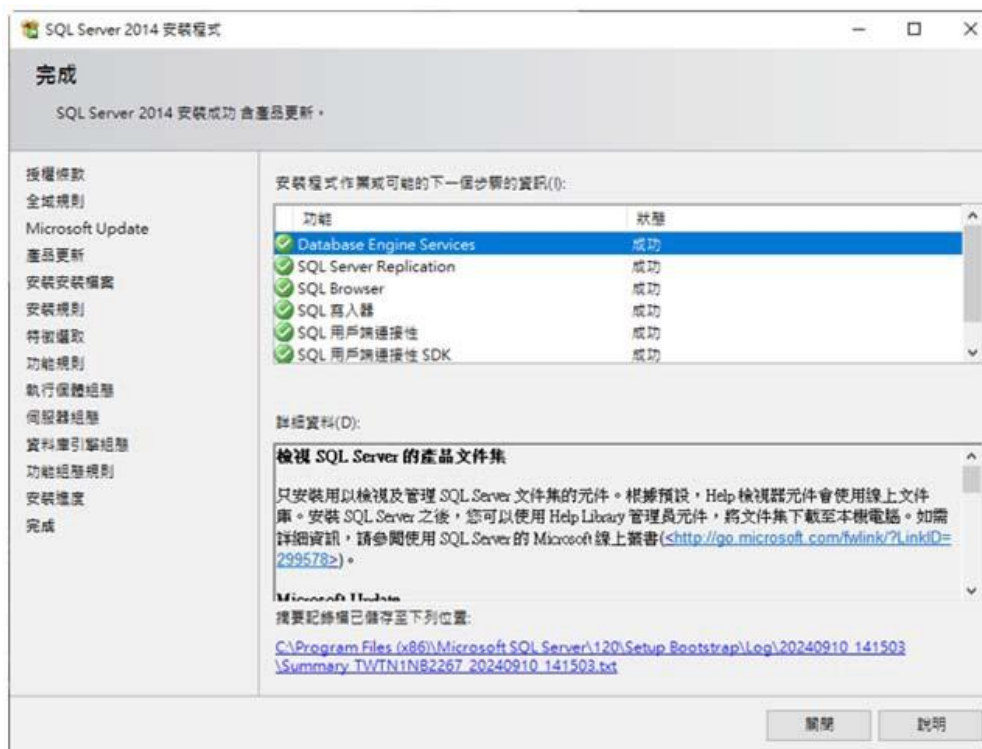
7. Перейдіть до наступного кроку, не змінюючи опцій.



8. Перейдіть до наступного кроку, не змінюючи опцій.



9. Інсталяцію SQL Server завершено.



● Встановлення SSMS

1. Виконайте SSMS-Setup-CHT.exe.
2. Виконайте встановлення.



3. Встановлення завершено.



● Створення SQL-сервера

1. Налаштуйте SQL Server.

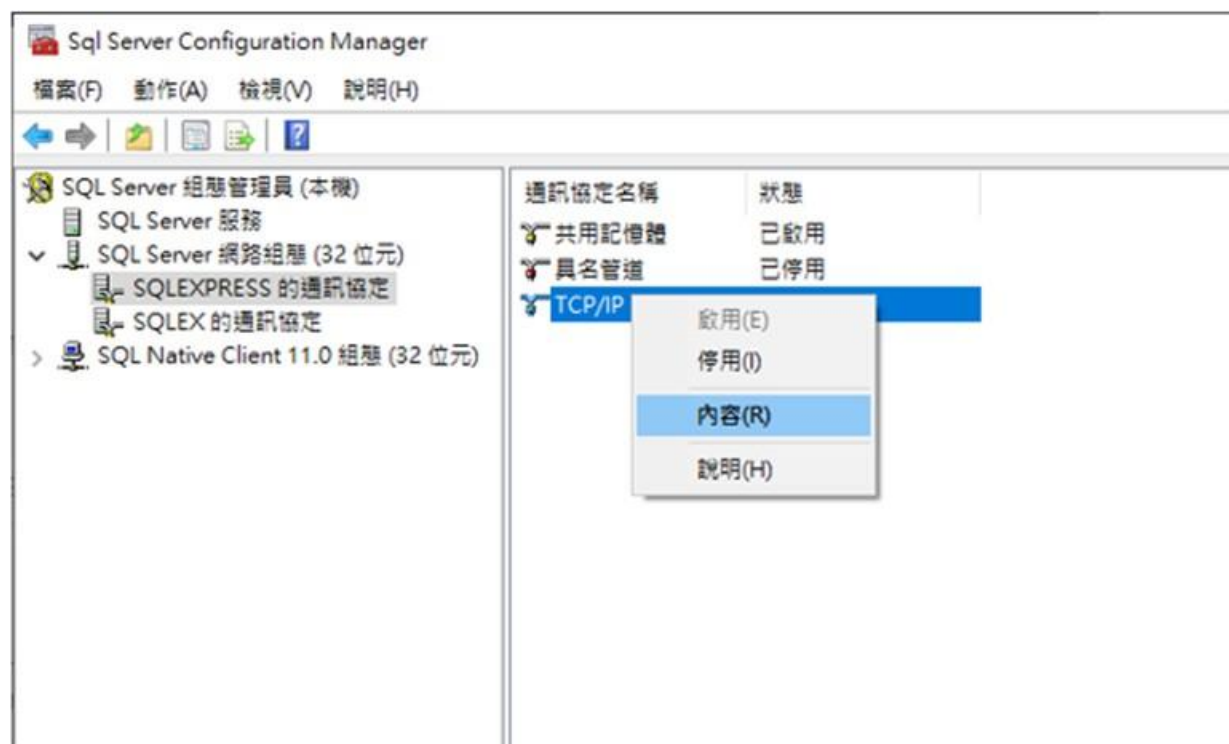
Виберіть **Microsoft SQL Server 2014 > Адміністратор конфігурації SQL Server 2014** .



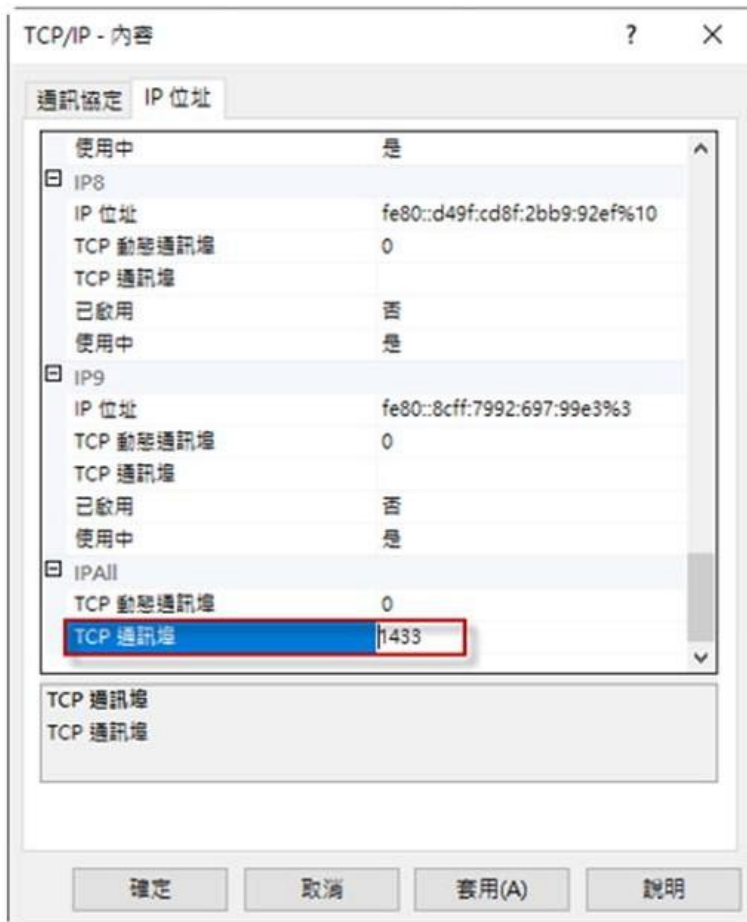
2. Розгорніть усі вузли, натисніть протокол **SQLEXPRESS** і підтвердіть, що статус TCP/IP праворуч увімкнено (Примітка: Якщо він все ще вимкнено, ви можете натиснути на TCP/IP, клацнути правою кнопкою миші, відкрити меню та вибрати Увімкнути).



3. Після увімкнення знову натисніть **TCP/IP**, клацніть правою кнопкою миші, і з'явиться меню, у якому можна вибрати вміст.



4. Прокрутіть спливаюче вікно вниз і встановіть **TCP-порт на 1433** .

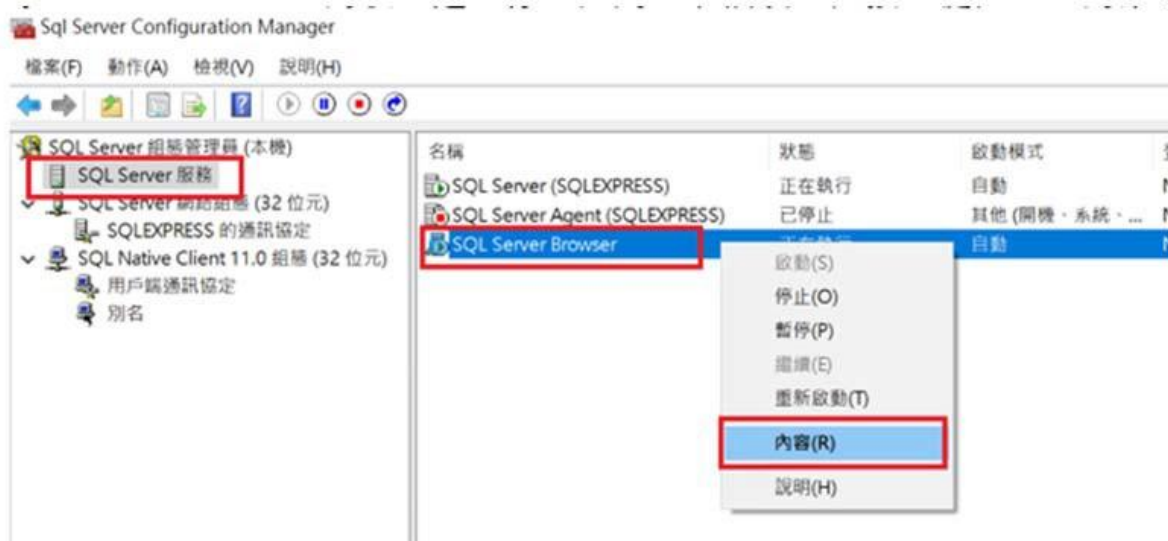


5. Натисніть **SQL Native Client** > Client Protocol та перевірте, чи увімкнено TCP/IP праворуч. Якщо він не увімкнено, натисніть на елемент і клацніть правою кнопкою миші, щоб відкрити меню, і виберіть Увімкнути.

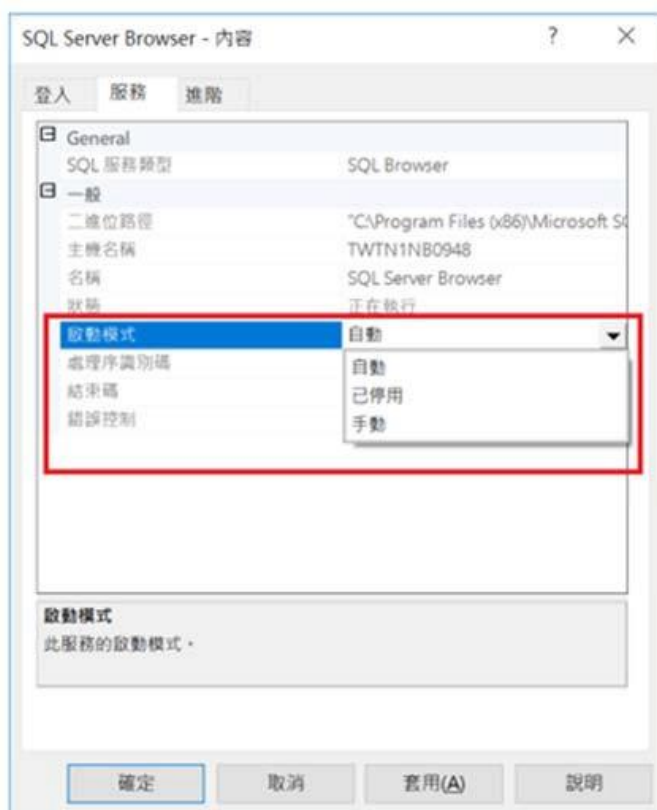


6. Клацніть **службу SQL Server** і підтвердіть, що стан SQL Server (SQLEXPRESS) і SQL Server Browser праворуч – виконання.

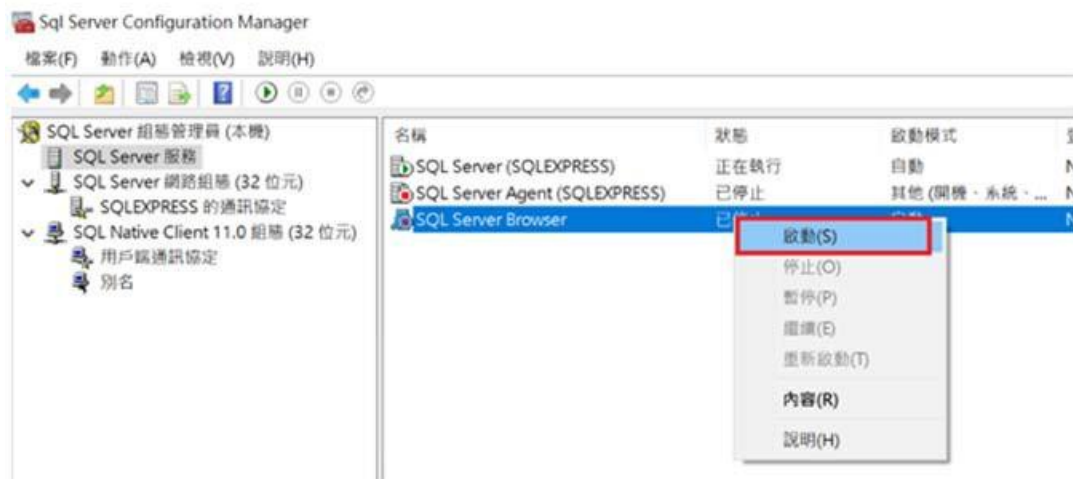
Якщо браузер SQL Server перебуває у стані зупинки, виберіть елемент, клацніть правою кнопкою миші та виберіть вміст у спливаючому меню.



7. Виберіть Авто для режиму запуску.

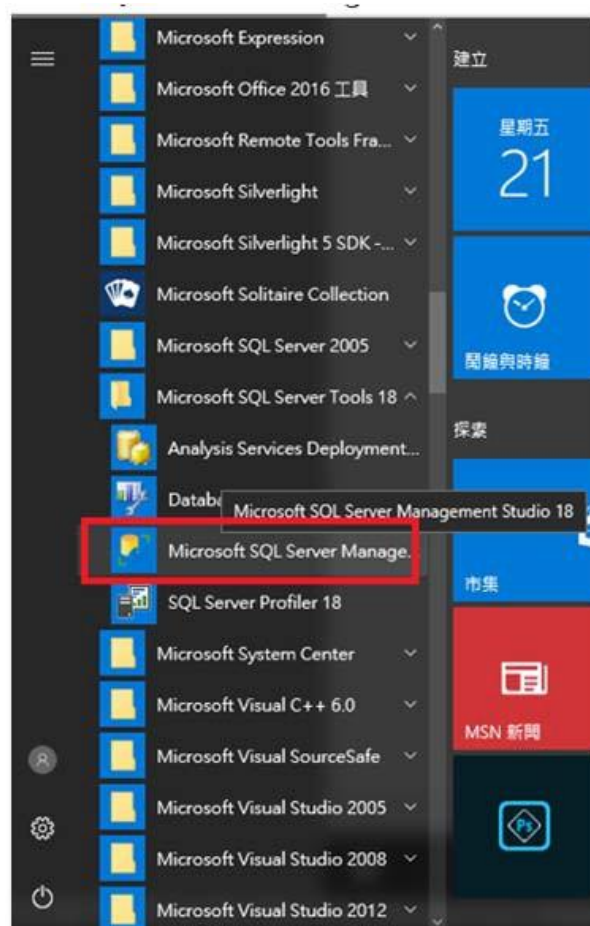


8. Після натискання кнопки **ОК** перевірте, чи браузер SQL Server запущено. Якщо браузер SQL Server зупинено, виберіть елемент, клацніть правою кнопкою миші та виберіть у спливаючому меню пункт «Запустити».

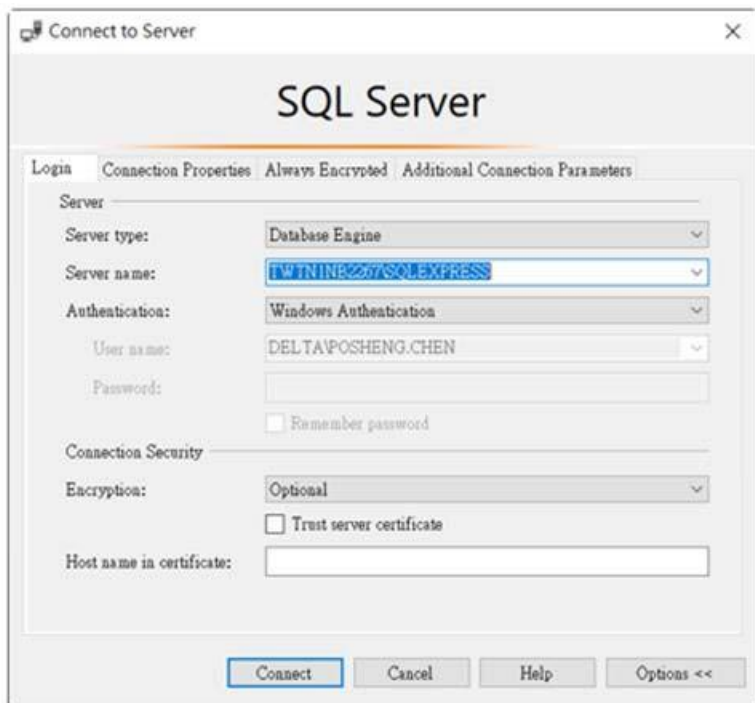


● Підключення до SQL Server та створення облікового запису й бази даних

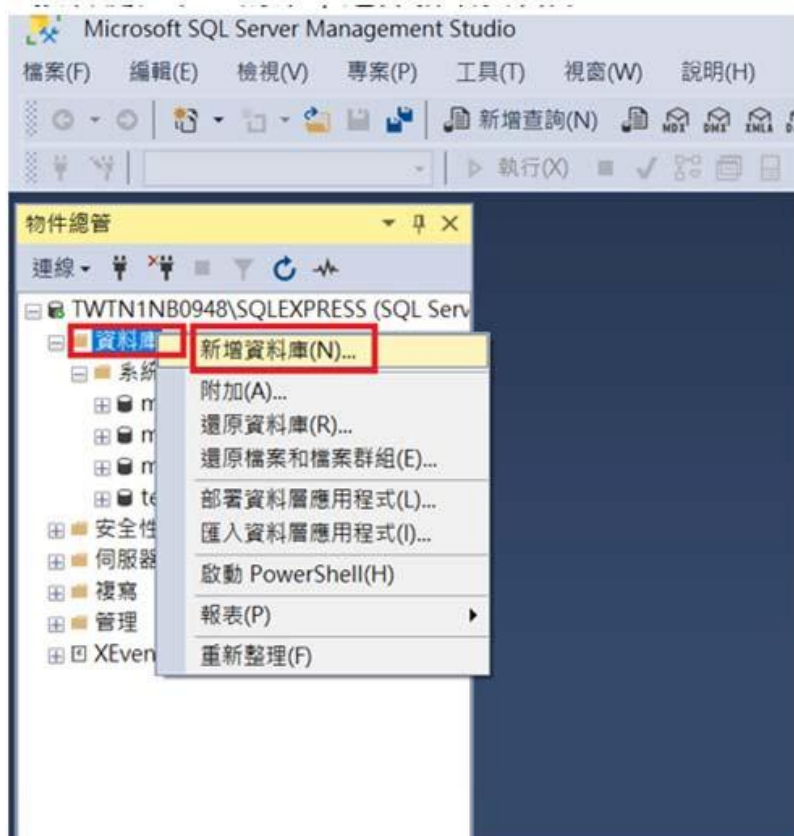
1. Виберіть Microsoft SQL Server Management Studio 18 як набір програм.



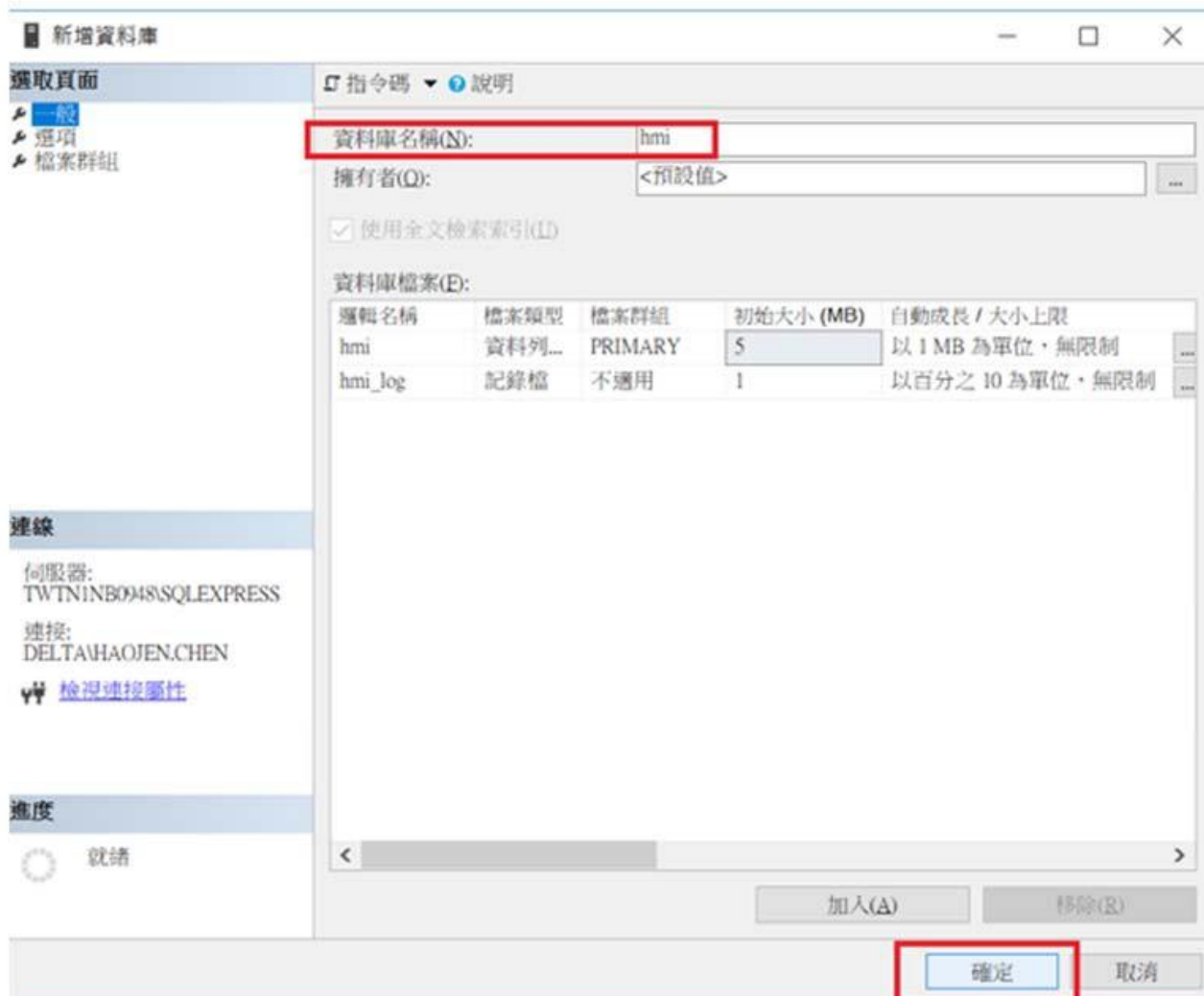
2. Підключіться до SQL Server (Примітка: Введіть пароль, встановлений під час інсталяції).



3. Клацніть базу даних, клацніть правою кнопкою миші та виберіть у спливаючому меню пункт Додати базу даних.



4. Введіть назву бази даних і натисніть кнопку «OK» .



新增資料庫

選取頁面

- 一般
- 選項
- 檔案群組

指令碼 說明

資料庫名稱(N): **hmi**

擁有者(O): <預設值>

☒ 使用全文檢索引引(I)

資料庫檔案(D):

邏輯名稱	檔案類型	檔案群組	初始大小 (MB)	自動成長 / 大小上限
hmi	資料列...	PRIMARY	5	以 1 MB 為單位, 無限制
hmi_log	記錄檔	不適用	1	以百分之 10 為單位, 無限制

連線

伺服器:
TWTNINB048\SQLEXPRESS

連接:
DELTA\HAOJEN.CHEN

[檢視連接屬性](#)

進度

就緒

加入(A) 移除(R)

確定 取消

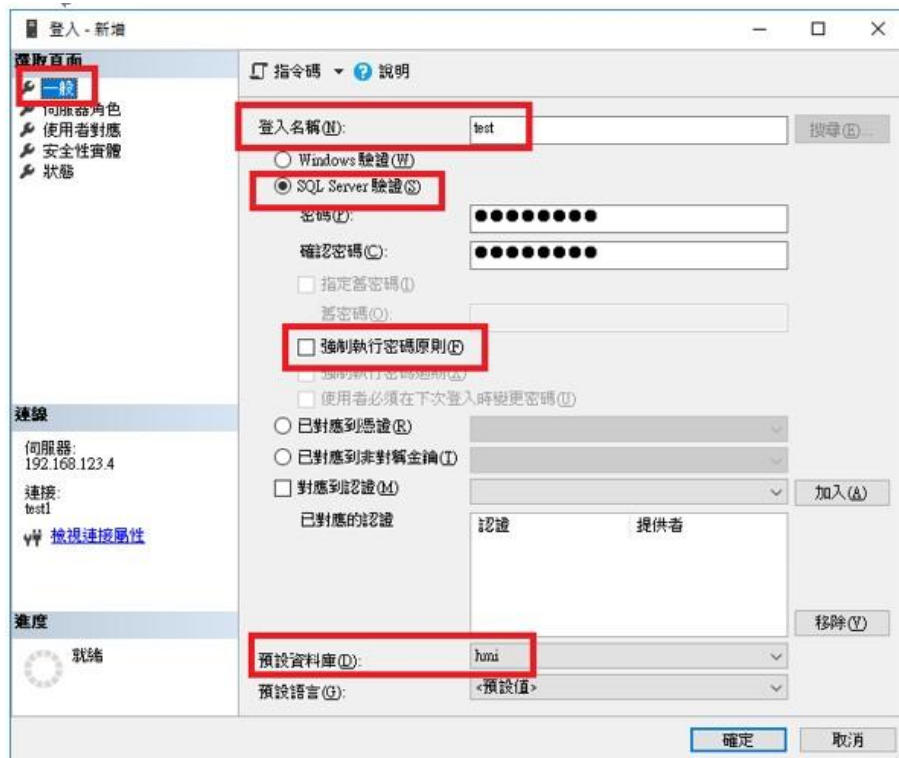
5. Розгорніть елемент безпеки, виберіть **«Вхід»** . Клацніть правою кнопкою миші та виберіть «Новий вхід» у спливаючому меню.



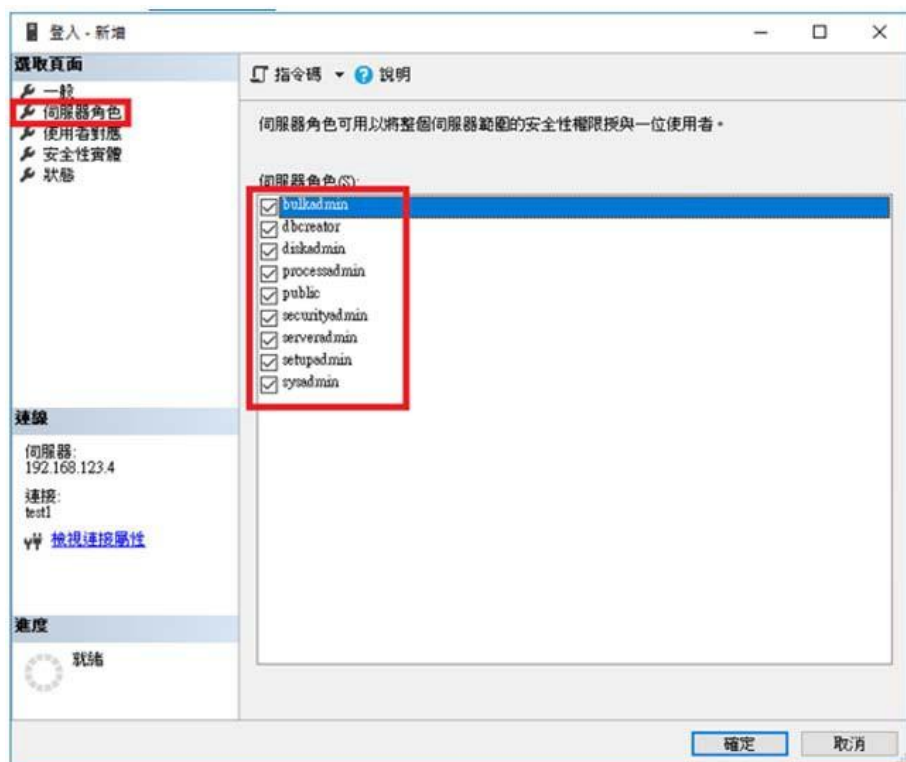
6. На загальній сторінці введіть своє ім'я для входу (номер облікового запису).

- Виберіть **«Автентифікація SQL Server»** і введіть пароль
- Зніміть прапорець «Застосувати політику паролів»
- Виберіть щойно створену базу даних у базі даних за замовчуванням.

➤ Натисніть OK

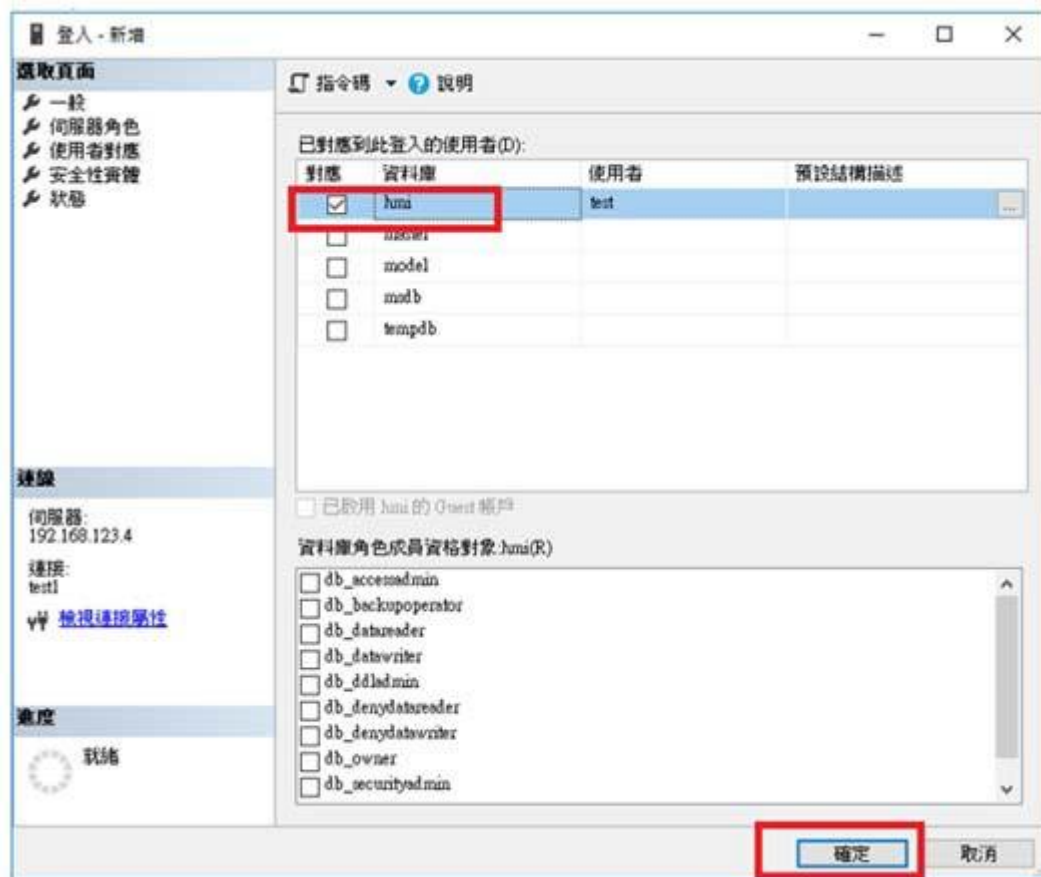


7. Для ролі сервера позначте всі пункти праворуч.



8. У розділі «Листування користувачів» перевірте новостворену базу даних.

Після завершення натисніть кнопку «OK» .



5. Функціональне тестування

Після виконання вищезазначених кроків налаштування SQL Server завершено. Будь ласка, скористайтеся тестом проекту в розділі 7.

Налаштування Луа:

```
while true do

    -- Add loop code here (cyclic run)

    -- 使用的資料表名稱
    tableName = "Recipe310"

    -- 連線資料
    dbType = 1 -- 0: MySQL 1: SQL Server
    serverIP = "192.168.123.4"
    serverPort = 1433
    usedDB = "hmi"
    account = "test"
    password = "Aa123456"
```

tableName змінюється на таблицю даних, яку потрібно створити

Тип бази даних встановлено на 1

serverIP встановлено на IP-адресу хоста бази даних,

serverPort встановлено на 1433.

usedDB встановлюється на ім'я новоствореної бази даних

обліковий запис налаштовано на щойно створений обліковий запис для входу

пароль встановлено на пароль, пов'язаний з новоствореним обліковим записом для входу

Опис функцій для наступних кнопок:



Запуск \$100.2 виконає частину створення таблиці даних LUA.

```
if (mem.inter.ReadBit(100, 2) == 1) then
    r1, id = sql.Connect(dbType, serverIP, serverPort, usedDB, account, password)

    error , size = sql.Connect(id)

    -- 建立一個有6個欄位的表格 (第一個欄位為索引)
    if (r1 == 1) then
        cmd = "CREATE table "
        cmd = cmd .. tableName
        cmd = cmd .. "(id INT NOT NULL, "
        cmd = cmd .. "material_1 VARCHAR(45) NULL, "
        cmd = cmd .. "material_2 VARCHAR(45) NULL, "
        cmd = cmd .. "material_3 VARCHAR(45) NULL, "
        cmd = cmd .. "material_4 VARCHAR(45) NULL, "
        cmd = cmd .. "material_5 VARCHAR(45) NULL, "
        cmd = cmd .. "PRIMARY KEY (id)) "

        r1, r2 = sql.ExecCommand(id, cmd);
    end
end
```

Запуск \$100.3 призведе до виконання LUA для створення віртуальних даних та їх розміщення в таблиці даних.

```
if (mem.inter.ReadBit(100, 3) == 1) then
    r1, id = sql.Connect(dbType, serverIP, serverPort, usedDB, account, password)

方法2 利用組合API
    if (r1 == 1) then
        for rowIndex=1, 5, 1 do
            sql.BeginInsertCommand()
            sql.SetInsertPattern("id", rowIndex)
            sql.SetInsertPattern("material_1", math.random(100, 1000))
            sql.SetInsertPattern("material_2", math.random(100, 1000))
            sql.SetInsertPattern("material_3", math.random(100, 1000))
            sql.SetInsertPattern("material_4", math.random(100, 1000))
            sql.SetInsertPattern("material_5", math.random(100, 1000))
            sql.EndInsertCommand(id, tableName)
```

Запуск \$100.0 призведе до виконання LUA-інструкції для зчитування даних таблиці та поміщення їх у частину \$0~\$24.

```
if (mem.inter.ReadBit(100, 0) == 1) then
    r1, id = sql.Connect(dbType, serverIP, serverPort, usedDB, account, password)
    if (r1 == 1) then
        cmd = "select * from "
        cmd = cmd .. tableName
        r1, r2 = sql.ExecCommand(id, cmd)
        r1, rowCount = sql.RecordCount(id) -- 資料總筆數
        r1, colCount = sql.FieldCount(id) -- 每筆資料欄位數

        memIndex = 0
        for rowIndex=0, rowCount-1, 1 do
            r1, t = sql.RecordData(id, rowIndex) -- 取得第rowIndex筆

            for colIndex=1, colCount-1, 1 do
                mem.inter.Write(memIndex, t[colIndex+1]) -- t[1].t[2].. , t base on 1
                memIndex = memIndex + 1
            end
        end
    end
end
end
```

Запуск \$100.1 призведе до виконання LUA-інструкції для запису вмісту \$0~\$24 у частину таблиці даних.

```

if (mem.inter.ReadBit(100, 1) == 1) then
    r1, id = sql.Connect(dbType, serverIP, serverPort, usedDB, account, password)
    if (r1 == 1) then
        memIndex = 0
        for rowIndex=1, 5, 1 do
            方法1 單一欄位更新
            memValue = mem.inter.Read(memIndex)
            memIndex = memIndex + 1
            r1, r2 = sql.ExecUpdateCommand(id, tableName, "material_1", memValue, "id", rowIndex)

            memValue = mem.inter.Read(memIndex)
            memIndex = memIndex + 1
            r1, r2 = sql.ExecUpdateCommand(id, tableName, "material_2", memValue, "id", rowIndex)

            memValue = mem.inter.Read(memIndex)
            memIndex = memIndex + 1
            r1, r2 = sql.ExecUpdateCommand(id, tableName, "material_3", memValue, "id", rowIndex)

            memValue = mem.inter.Read(memIndex)
            memIndex = memIndex + 1
            r1, r2 = sql.ExecUpdateCommand(id, tableName, "material_4", memValue, "id", rowIndex)

            memValue = mem.inter.Read(memIndex)
            memIndex = memIndex + 1
            r1, r2 = sql.ExecUpdateCommand(id, tableName, "material_5", memValue, "id", rowIndex)

```

6. Опис функцій

Інструкція	Опис
sql.Connect Підключення до бази даних	Візерунок: r1, r2 = sql.Connect(dbType, хост, порт, ім'я бази даних, користувач, пароль) dbType 0: MySql dbType 1: SQL Server Приклад: r1, r2 = sql.Connect(0, "192.168.123.4", 3306, "hmi", "тест", "12345678") Повернене значення: r1 : успіх (1) / невдача (0) r2 : ідентифікатор підключення (> 0) / код результату (< 0)
sql.ExecCommand Виконання інструкцій SQL	Візерунок: r1, r2 = sql.ExecCommand (id, sql_command) id: значення r2 з sql.Connect Приклад: r1, r2 = sql.ExecCommand(1, "вибрати * з тесту") Повернене значення: r1 : успіх (1) / невдача (0) r2 : код результату (< 0)
sql.RecordCount Отримати кількість транзакцій, повернутих останнім викликом sql.ExecCommand	Візерунок: r1, r2 = sql.RecordCount(id) Приклад: r1, r2 = sql.RecordCount (1) Повернене значення: r1 : успіх (1) / невдача (0) r2 : кількість записів (> 0) / код результату (< 0)
sql.FieldCount Отримати загальну кількість полів, повернутих останнім викликом sql.ExecCommand	Візерунок: r1, r2 = sql.FieldCount (ідентифікатор) Приклад: r1, r2 = sql.FieldCount (1) Повернене значення:

	r1 : успіх (1) / невдача (0) r2 : кількість полів (> 0) / код результату (< 0)
sql.RecordData Отримати дані, повернуті останнім викликом sql.ExecCommand	Візерунок: r1, r2 = sql.RecordData (id, recordIndex) recordIndex: 第几笔资料 Приклад: r1, r2 = sql.RecordData(1, 2) Повернене значення: r1 : успіх (1) / невдача (0) r2 : дані запису (> 0) / код результату (< 0) В и к о р и с т о в у й т е recorddata[1], recorddata[2]...recorddata[FiledCount]
sql.CloseConnect Закрити з'єднання з базою даних	Візерунок: r1, r2 = sql.CloseConnect (ідентифікатор) Приклад: r1, r2 = sql.ЗакритиЗв'язок (1) Повернене значення: r1 : успіх (1) / невдача (0) r2 : код результату (< 0)
sql.ExecUpdateCommand Оновлення даних таблиці	Візерунок: r1, r2 = sql.ExecUpdateCommand(id, tableName, fieldName, value, where поле умови, where поле умови) Приклад: r1, r2 = sql.ExecUpdateCommand(id, tableName, "material_1", math.random(100, 1000), "id", rowIndex) Повернене значення: r1 : успіх (1) / невдача (0) r2 : код результату (< 0)
sql.BeginInsertCommand Ініціалізація команди вставки таблиці даних	Візерунок: sql.BeginInsertCommand() Приклад: sql.BeginInsertCommand()
Sql.SetInsertPattern Встановлює вміст даних для вставки	Шаблон: sql.SetInsertPattern(поле, значення) Приклад: sql.SetInsertPattern("матеріал_1", 100) Повернене значення: r1 : успіх (1) / невдача (0) r2 : код результату (< 0)
sql.EndInsertCommand Виконання команди вставки таблиці даних	Візерунок: sql.EndInsertCommand(id, tableName) Приклад: sql.EndInsertCommand(1, "рецепт100") Повернене значення: r1 : успіх (1) / невдача (0) r2 : код результату (< 0)

7. Висновок

У цій статті демонструється архітектура **ПК як SQL Server, HMI як SQL Client** та описується, як оновлювати дані бази даних через SQL API.

Демонструвати:

