

C#과 MS SQL 연동 (Select 및 Insert문)

- C#과 MS SQL을 연결하는 방법을 습득한다.
- WinForm과 MS SQL을 연동하는 방법을 습득한다.

- 동작 확인
- MSSQL 설치
- C# WinForm 프로젝트 만들기
- WinForm 만들기
- C#과 MSSQL 기능 연동하기
 - 변수명 지정
 - 함수 생성
 - MSSQL 연결
 - 프로그램 연동
 - 화면 표시

Section 01 동작 확인

- 프로그램 실행 화면

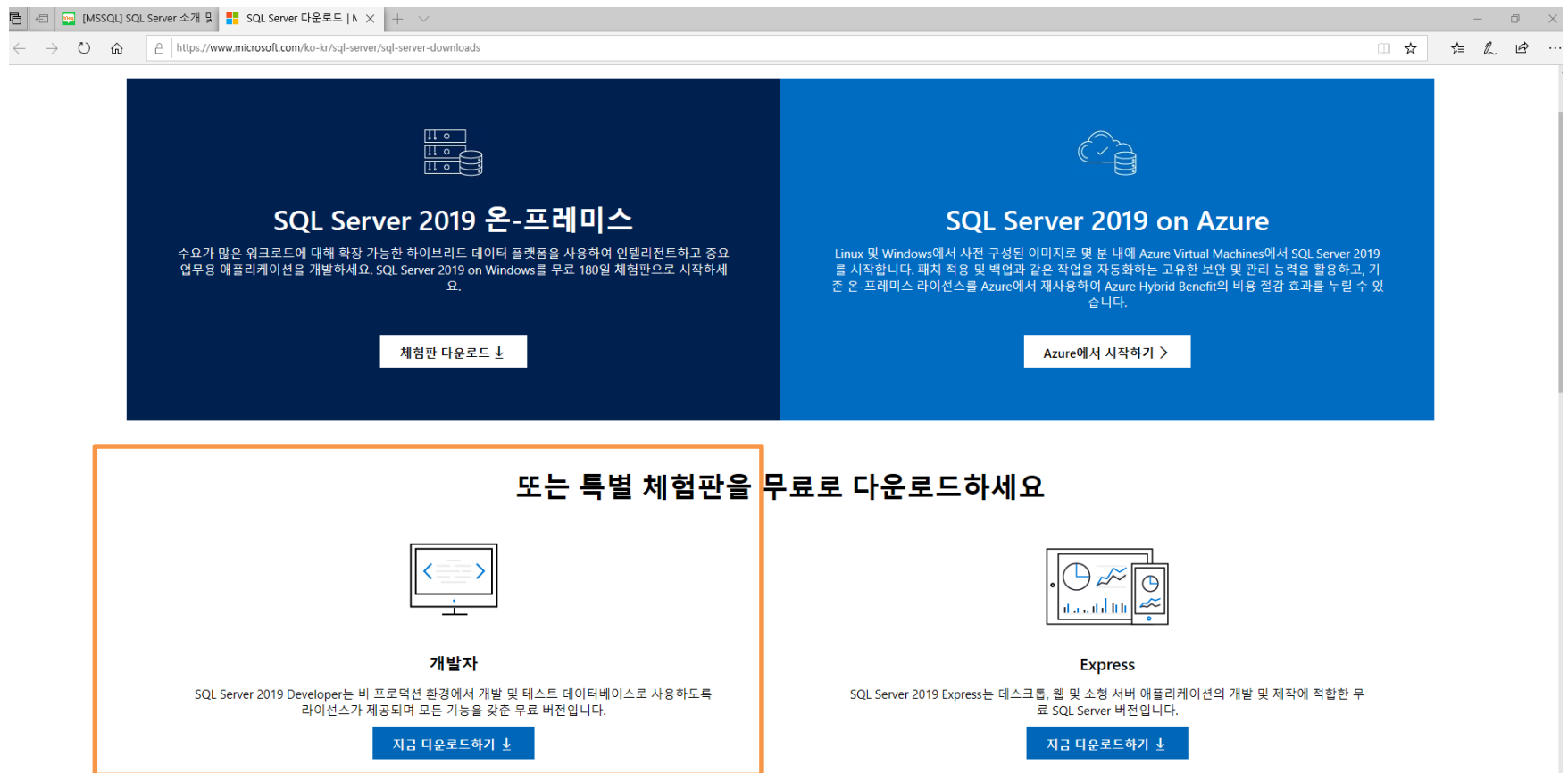
	CUST_ID	BIRTH_DT
▶	11	lee
	111	222
	123	222
	adsf	fda
	new	222222
	new2	121212
	new3	e32
	new4	4321
	sfdg	sfg
	xfgvfg	dfgdfg
	xghdgh	cbncbn

조회

ID BIRTH

Section 02 MSSQL 설치

- MSSQL 설치 및 연결
- 아래 사이트에서 SQL Server 다운로드
- <https://www.microsoft.com/ko-kr/sql-server/sql-server-downloads>



The screenshot shows the Microsoft SQL Server 2019 download page. The browser address bar shows the URL: <https://www.microsoft.com/ko-kr/sql-server/sql-server-downloads>.

SQL Server 2019 온-프레미스
수요가 많은 워크로드에 대해 확장 가능한 하이브리드 데이터 플랫폼을 사용하여 인텔리전트하고 중요 업무용 애플리케이션을 개발하세요. SQL Server 2019 on Windows를 무료 180일 체험판으로 시작하세요.

SQL Server 2019 on Azure
Linux 및 Windows에서 사전 구성된 이미지로 몇 분 내에 Azure Virtual Machines에서 SQL Server 2019를 시작합니다. 패치 적용 및 백업과 같은 작업을 자동화하는 고유한 보안 및 관리 능력을 활용하고, 기존 온-프레미스 라이선스를 Azure에서 재사용하여 Azure Hybrid Benefit의 비용 절감 효과를 누릴 수 있습니다.

또는 특별 체험판을 무료로 다운로드하세요

개발자
SQL Server 2019 Developer는 비 프로덕션 환경에서 개발 및 테스트 데이터베이스로 사용하도록 라이선스가 제공되며 모든 기능을 갖춘 무료 버전입니다.

Express
SQL Server 2019 Express는 데스크톱, 웹 및 소형 서버 애플리케이션의 개발 및 제작에 적합한 무료 SQL Server 버전입니다.

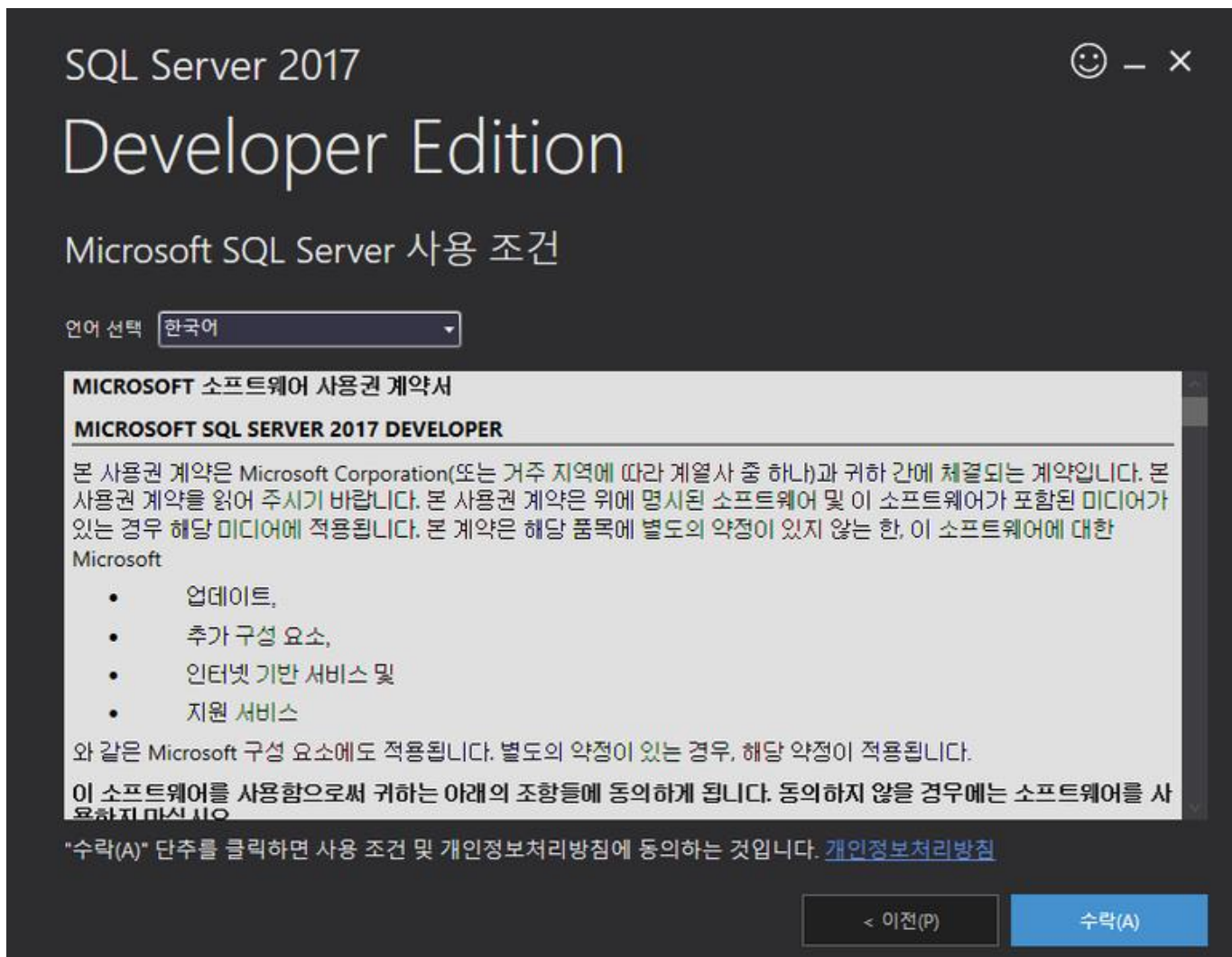
Section 02 MSSQL 설치

■ 설치 유형은 기본 구성을 권장



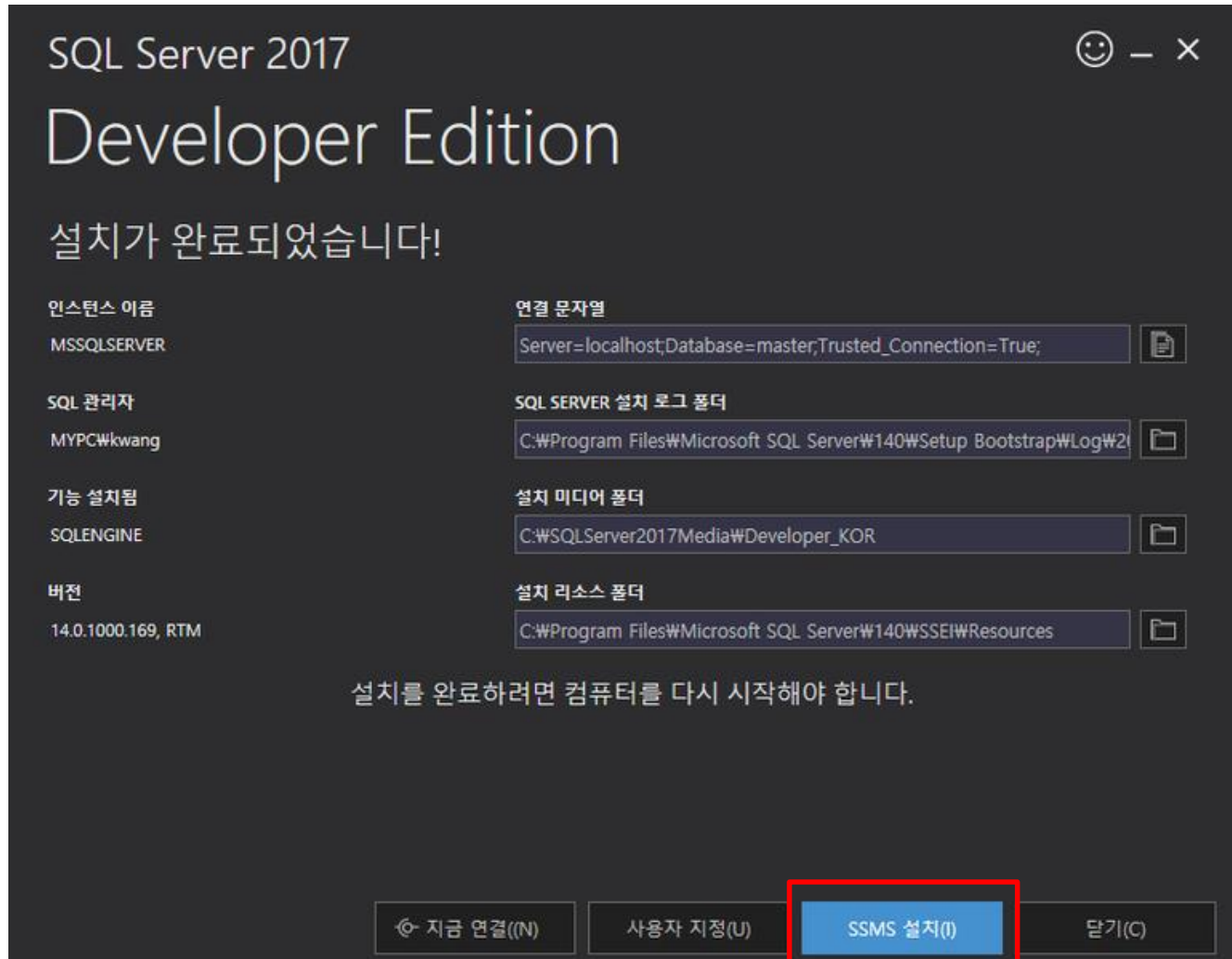
Section 02 MSSQL 설치

■ 언어 선택



Section 02 MSSQL 설치

- 설치 완료 후 인스턴스 이름과 폴더 등을 확인
- DB 관리툴 설치(SQL Server Management Studio)



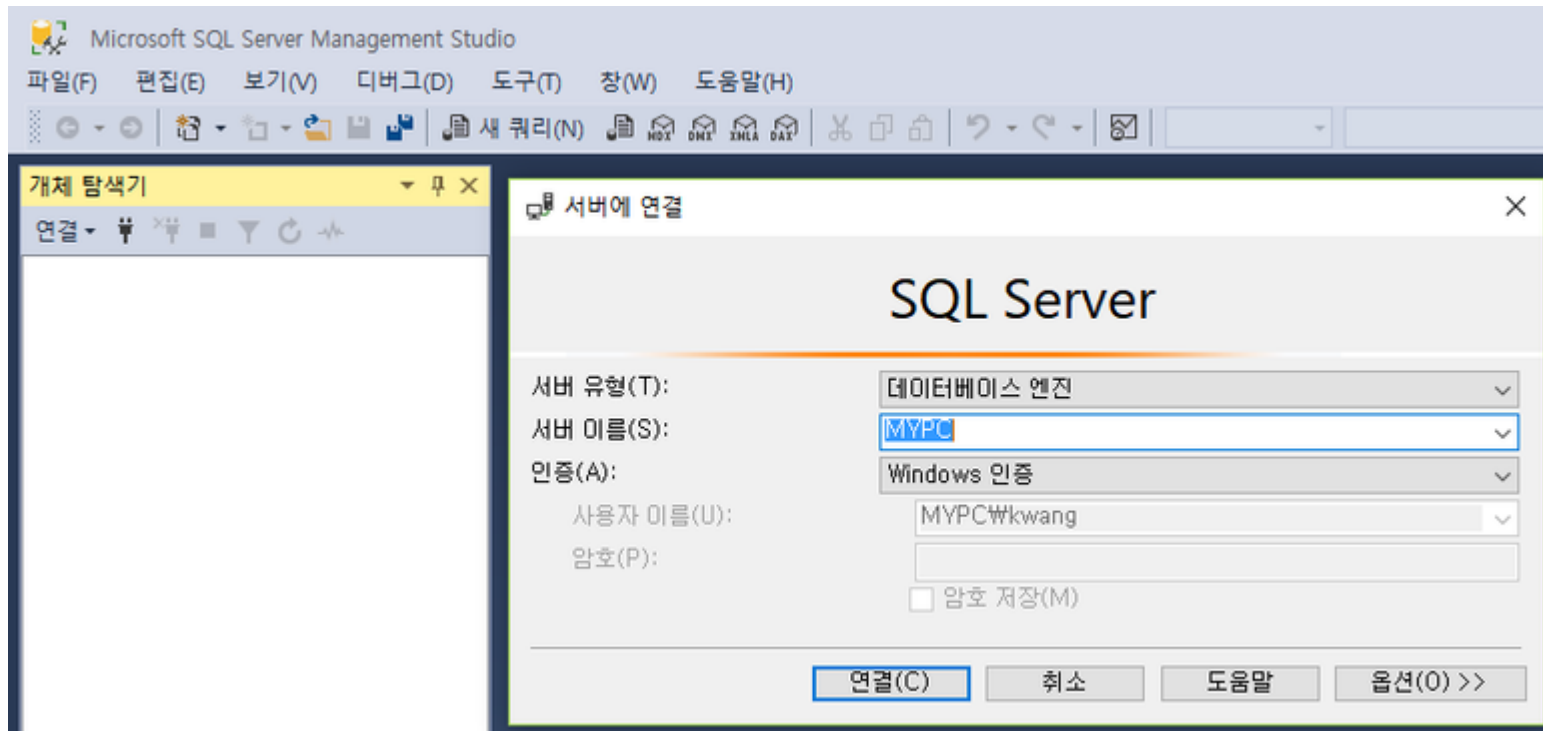
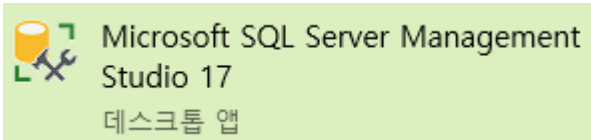
Section 02 MSSQL 설치

■ 설치 완료 후 재부팅



Section 02 MSSQL 설치

- S는(SQL Server Management Studio) 실행
- 서버이름 입력 후 연결
- Local 서버인 경우에는 점(.)만 입력해도 연결 가능



Section 02 MSSQL 설치

■ 데이터베이스 생성

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the '데이터베이스' (Databases) folder is expanded, and a context menu is open with '새 데이터베이스(N)...' (New Database...) selected. The main pane displays the '새 데이터베이스' (New Database) dialog box.

새 데이터베이스(N)...

데이터베이스 이름(N): MYDB1
소유자(O): <기본값>

☒ 전체 텍스트 인덱싱 사용(U)

데이터베이스 파일(F):

논리적 이름	파일 형식	파일 그룹	처음 크기(MB)	자동 증가/최대 크기
MYDB1	행 데이터	PRIMARY	8	64MB 단위로, 제한 없음
MYDB1_log	로그	해당 사항 없음	8	64MB 단위로, 제한 없음

연결

서버: MYPC
연결: MYPC\kwang
[연결 속성 보기](#)

진행률

준비 완료

Buttons: 추가(A), 제거(R), 확인, 취소

Section 02 MSSQL 설치

■ 새 테이블 생성(C#과 연동할 테이블)

The screenshot displays the SQL Server Enterprise Manager interface. On the left, the 'Tables' folder under 'MYDB1' is selected. The main pane shows the 'dbo.Table_1*' table structure with the following columns:

열 이름	데이터 형식	Null 허용
CUST_ID	nchar(10)	<input checked="" type="checkbox"/>
BIRTH_DT	nchar(8)	<input checked="" type="checkbox"/>

Below the table structure, the '열 속성' (Column Properties) window is open for the 'CUST_ID' column. It shows the following properties:

속성	값
(일반)	
(이름)	CUST_ID
Null 허용	예
기본값 또는 바인딩	
길이	10
데이터 형식	nchar
테이블 디자이너	
DTS 게시됨	아니요

On the right, the '속성' (Properties) window is open for the 'dbo.TB_CUST' table. It shows the following properties:

속성	값
[테이블] dbo.TB_CUST	
(이름)	TB_CUST
데이터베이스 이름	MYDB1
서버 이름	mypc
설명	
스키마	dbo
테이블 디자이너	
ID 열	
기본 데이터 공간 사양	PRIMARY
복제됨	아니요
인덱싱 가능	예
잠금 에스컬레이션	테이블
텍스트/이미지 파일 그룹	PRIMARY
행 GUID 열	

Section 02 MSSQL 설치

■ 테이블이 제대로 생성되었는지 여부 확인

The screenshot displays the SQL Server Enterprise Manager interface on the left and a SQL Query window on the right.

Enterprise Manager (Left): The tree view shows the hierarchy: MYPC (SQL Server 14.0.1000.169 - MYPC#kwang) > 데이터베이스 > MYDB1 > 테이블 > **dbo.TB_CUST**. The table **dbo.TB_CUST** is highlighted with a red box. A context menu is open over it, showing options: 새 테이블(T)... (New Table...), 디자인(G) (Design), and 상위 1000개 행 선택(W) (Select Top 1000 Rows).

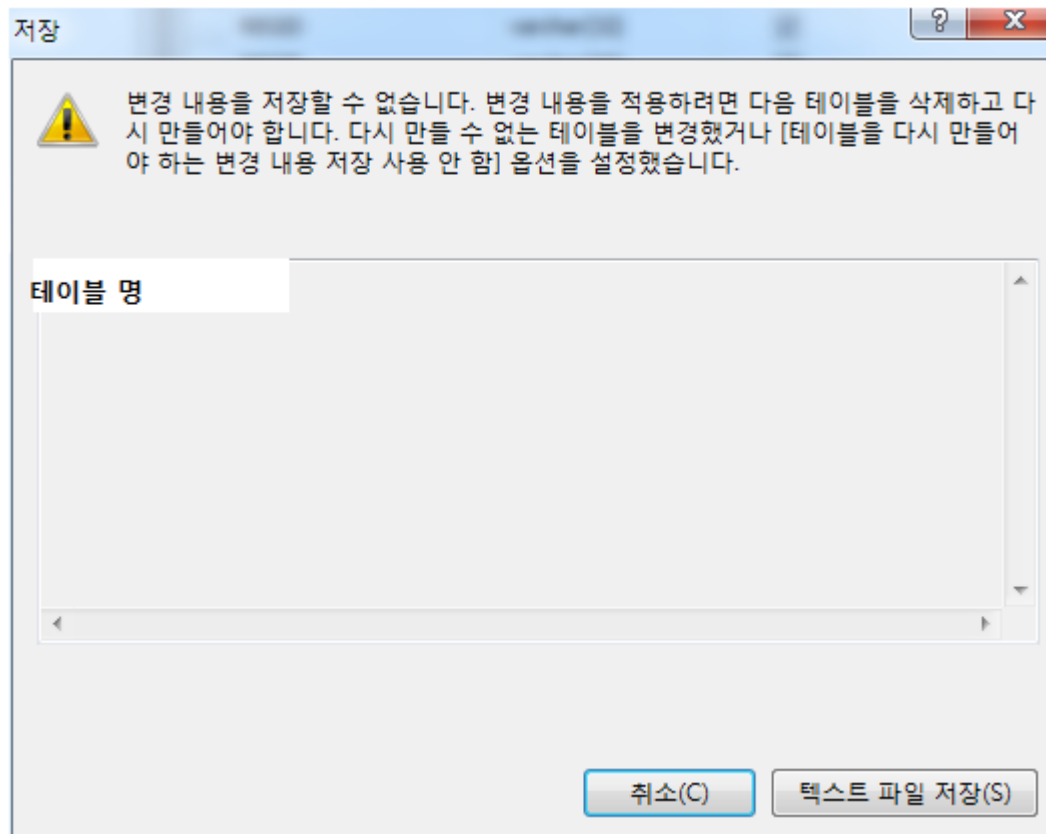
SQL Query Window (Right): The title bar indicates the query is for 'MYPC.MYDB1 - db...B1 - dbo.TB_CUST'. The query text is as follows:

```
SELECT TOP (1000) [CUST_ID]
, [BIRTH_DT]
FROM [MYDB1].[dbo].[TB_CUST]
```

The query text is enclosed in a red box. Above the query, a comment line reads: `/***** SSMS의 SelectTopNRows 명령 스크립트 *****/`.

■ 테이블 속성 변경 관련 에러메시지가 나타날 경우

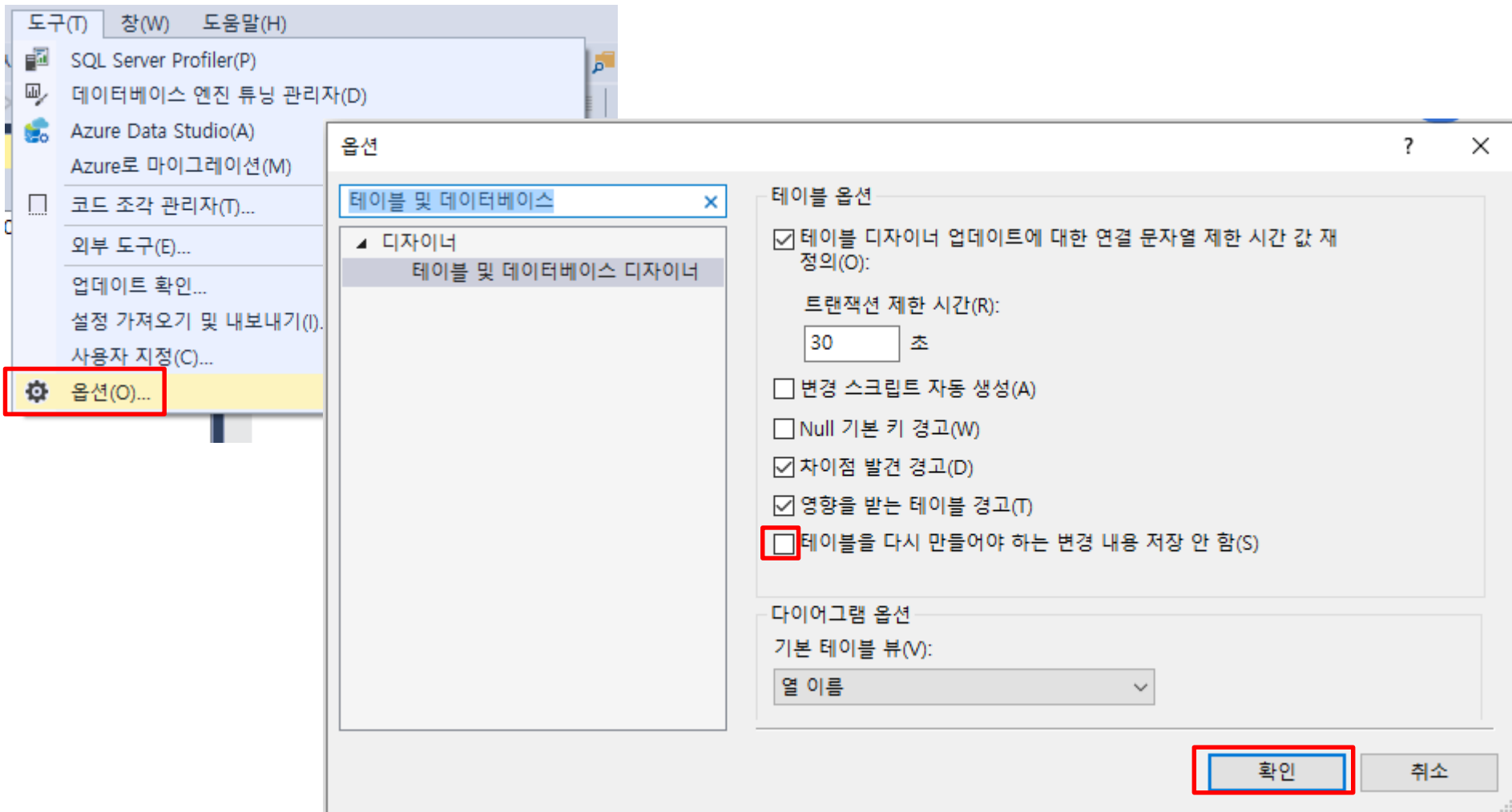
- '변경 내용을 저장할 수 없습니다. 변경 내용을 적용하려면 다음 테이블을 삭제하고 다시 만들어야 합니다. 다시 만들 수 없는 테이블을 변경했거나 [테이블을 다시 만들어야 하는 변경 내용 저장 사용 안 함] 옵션을 설정했습니다.'



Section 02 MSSQL 설치

■ 테이블 속성 변경 관련 에러메시지가 나타날 경우

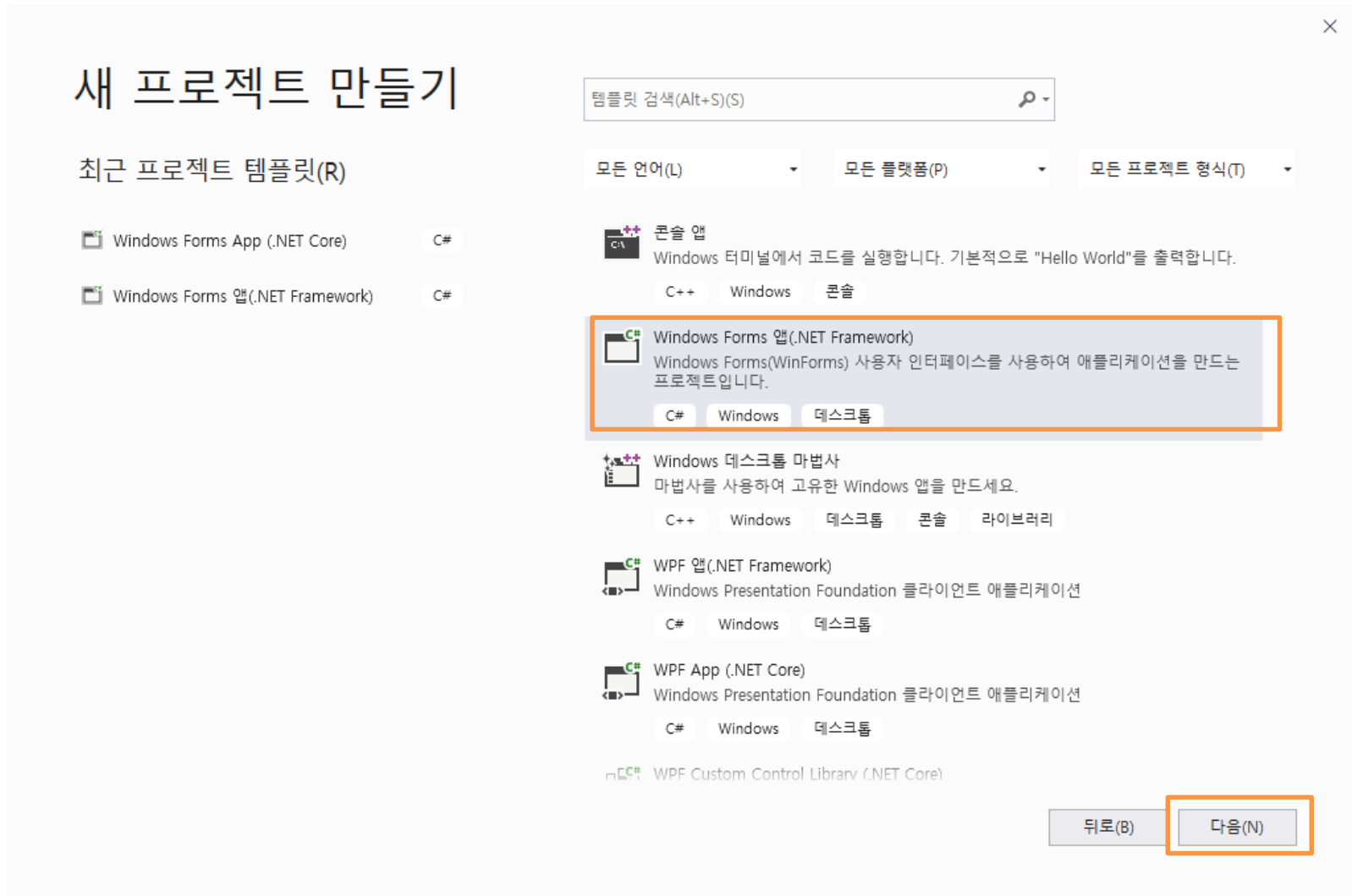
- 도구-> 옵션에서 테이블 및 데이터베이스 검색 후 표시된 부분 체크 해제후 확인



Section 03 C# WinForm 프로젝트 만들기

■ 프로젝트 기본 구성

- Windows Forms App(.NET Framework)으로 새프로젝트 생성



Section 03 C# WinForm 프로젝트 만들기

- 프로젝트 기본 구성
 - Windows Forms App으로 새 프로젝트 생성

새 프로젝트 구성

Windows Forms 앱(.NET Framework) C# Windows 데스크톱

프로젝트 이름(N)
WindowsFormsApp1

위치(L)
F:\source\csharp\example

솔루션 이름(M) ⓘ
WindowsFormsApp1

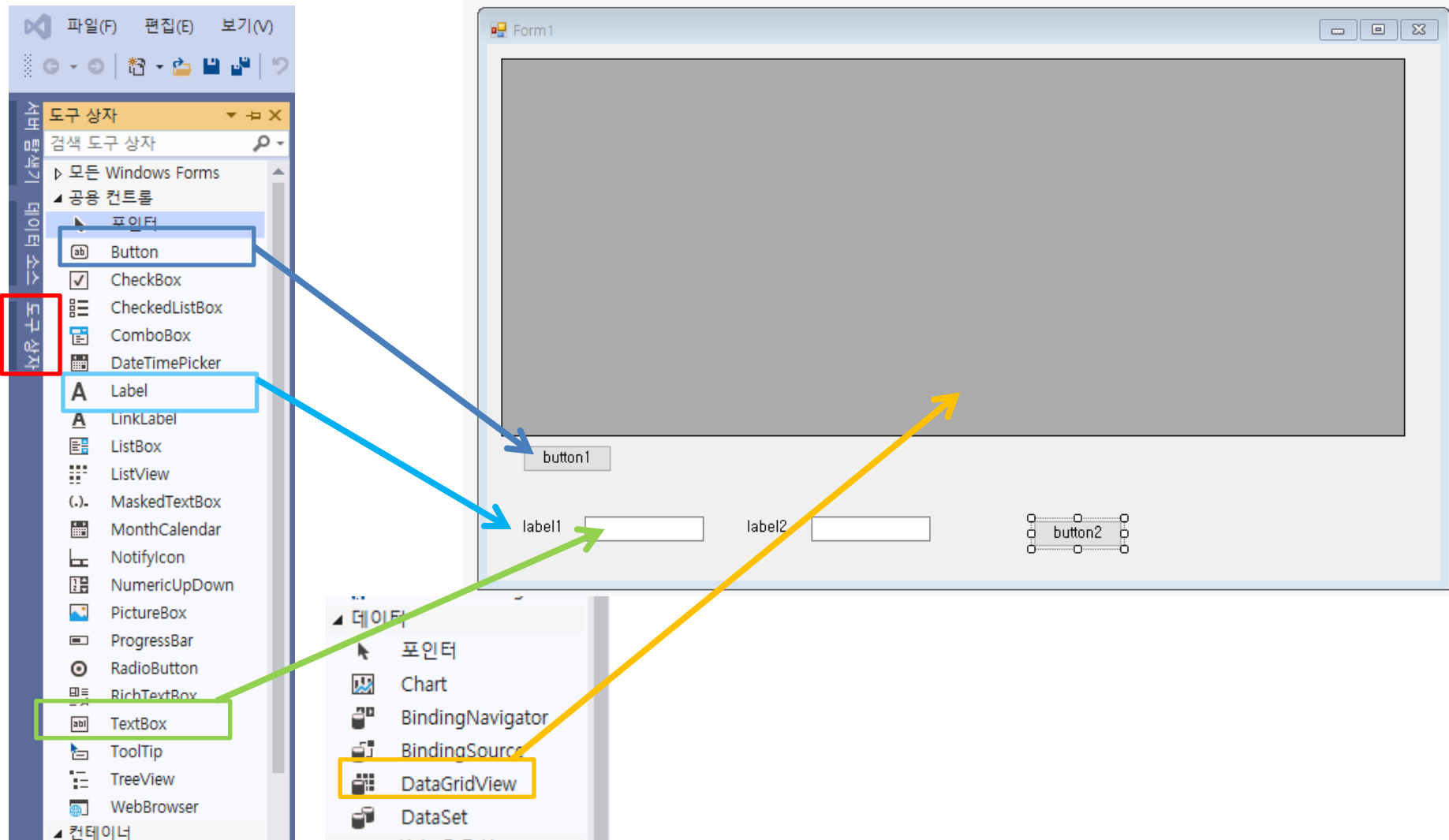
☐ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

프레임워크(F)
.NET Framework 4.7.2

뒤로(B) 만들기(C)

Section 04 WinForm 만들기

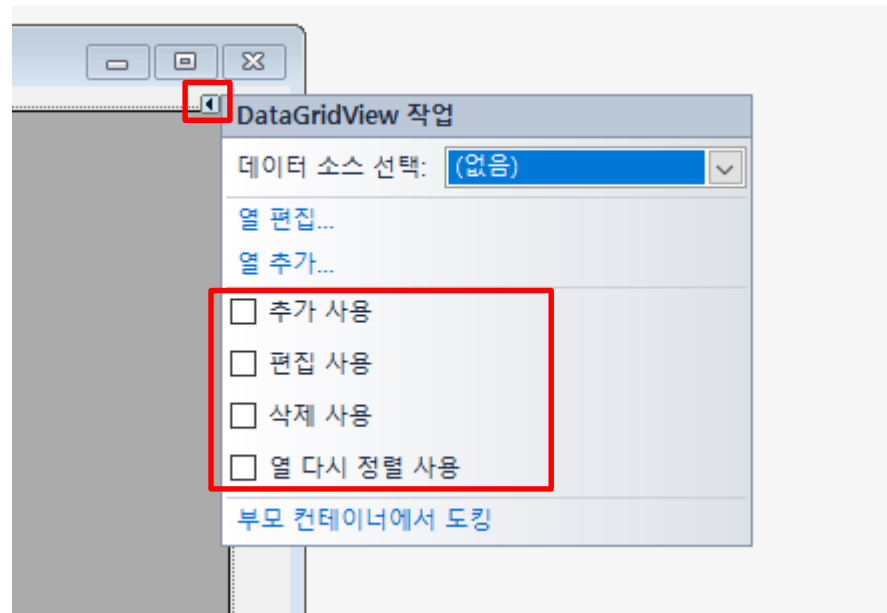
- 프로그램 화면에 필요한 요소들 추가
 - 도구상자에서 Drag & Drop하여 화면과 같이 구상



Section 04 WinForm 만들기

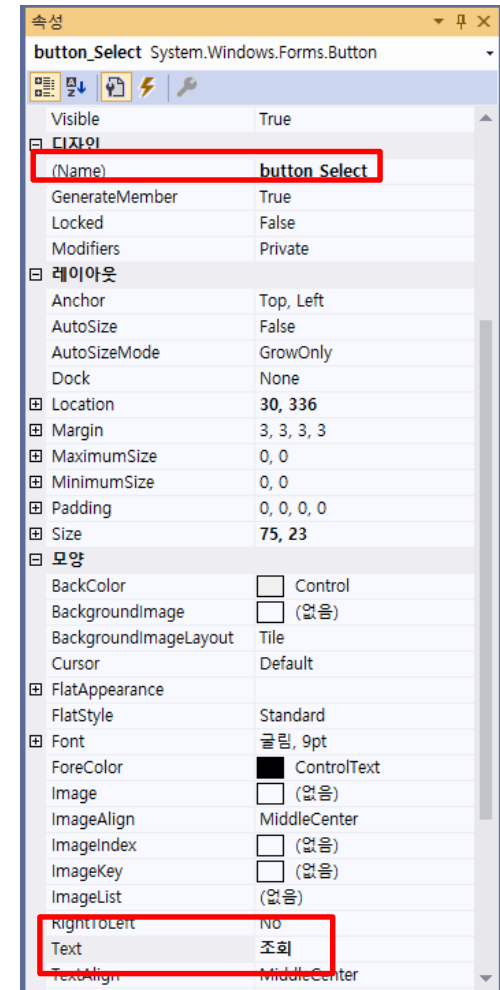
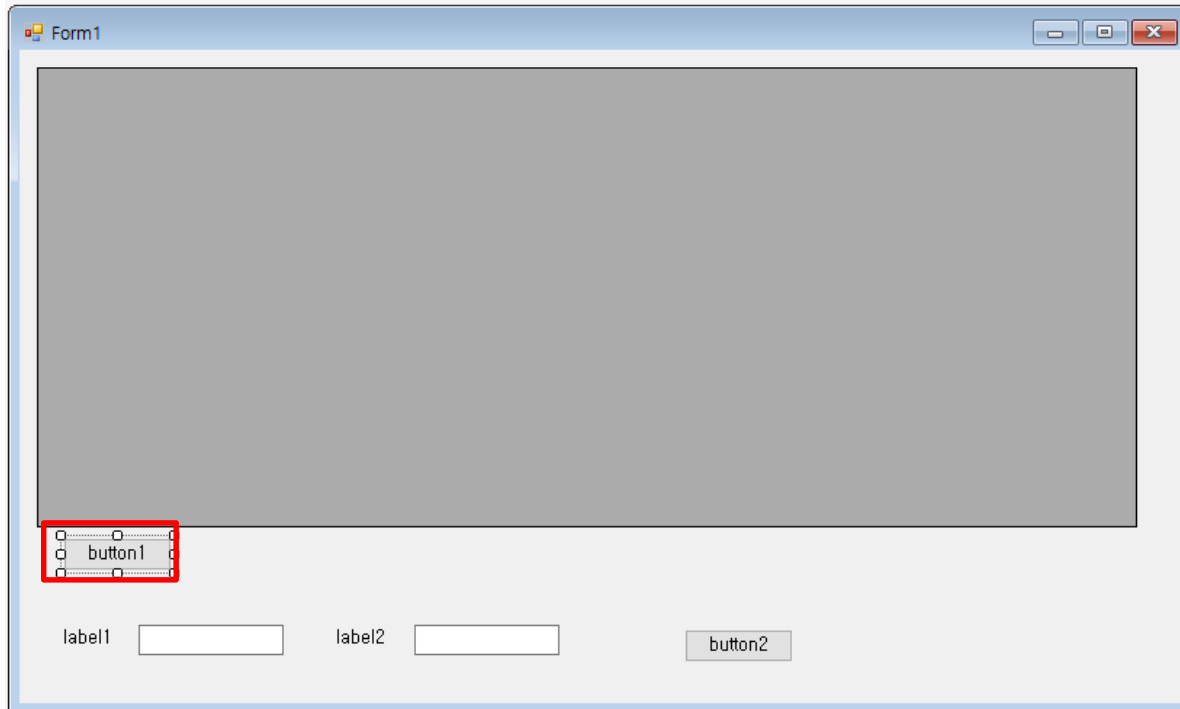
■ DataGridView

- 데이터들을 표형식으로 나타내줌
- 화면 아래의 버튼들을 통해서 데이터들을 조회하고 추가할 것이므로, 모든 체크박스들 체크 해제



Section 05 C#과 MSSQL 기능 연동하기 – 변수명 지정

- 버튼 및 텍스트박스들의 텍스트 및 이름 변경
 - 데이터들을 표형식으로 나타내줌
 - 화면 아래의 버튼들을 통해서 데이터들을 조회하고 추가할 것이므로, 모든 체크박스들 체크 해제
 - Name : 코드상에서 사용되는 변수명
 - Text : 화면에 표시되는 글자



Section 05 C#과 MSSQL 기능 연동하기 - 변수명 지정

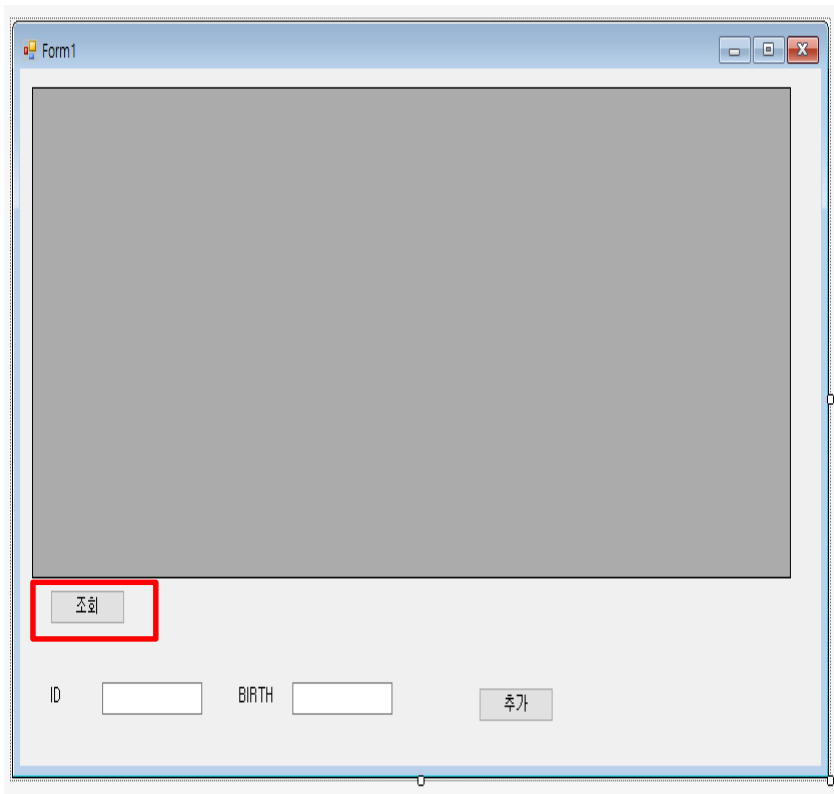
- 아래 그림과 같이 Text 및 Name 변경
 - ※ 별도 표기 없는 부분은 자유롭게 작성

The screenshot shows a Windows Form titled "Form1" with a standard Windows XP-style title bar. The form contains a large gray rectangular area. Below this area, there are several controls:

- A button labeled "조회" (Search) with a callout box showing: Text : 조회, Name : button_Select.
- Two text boxes for input. The first is labeled "ID" with a callout box: Text : ID, Name : textBox_ID. The second is labeled "BIRTH" with a callout box: Text : BIRTH, Name : textBox_Birth.
- A button labeled "추가" (Add) with a callout box showing: Text : 추가, Name : button_Insert.

Section 05 C#과 MSSQL 기능 연동하기 - 함수생성

- 조회 및 추가 버튼 더블 클릭
- 더블 클릭시 해당 버튼 클릭시 수행할 함수가 생성됨

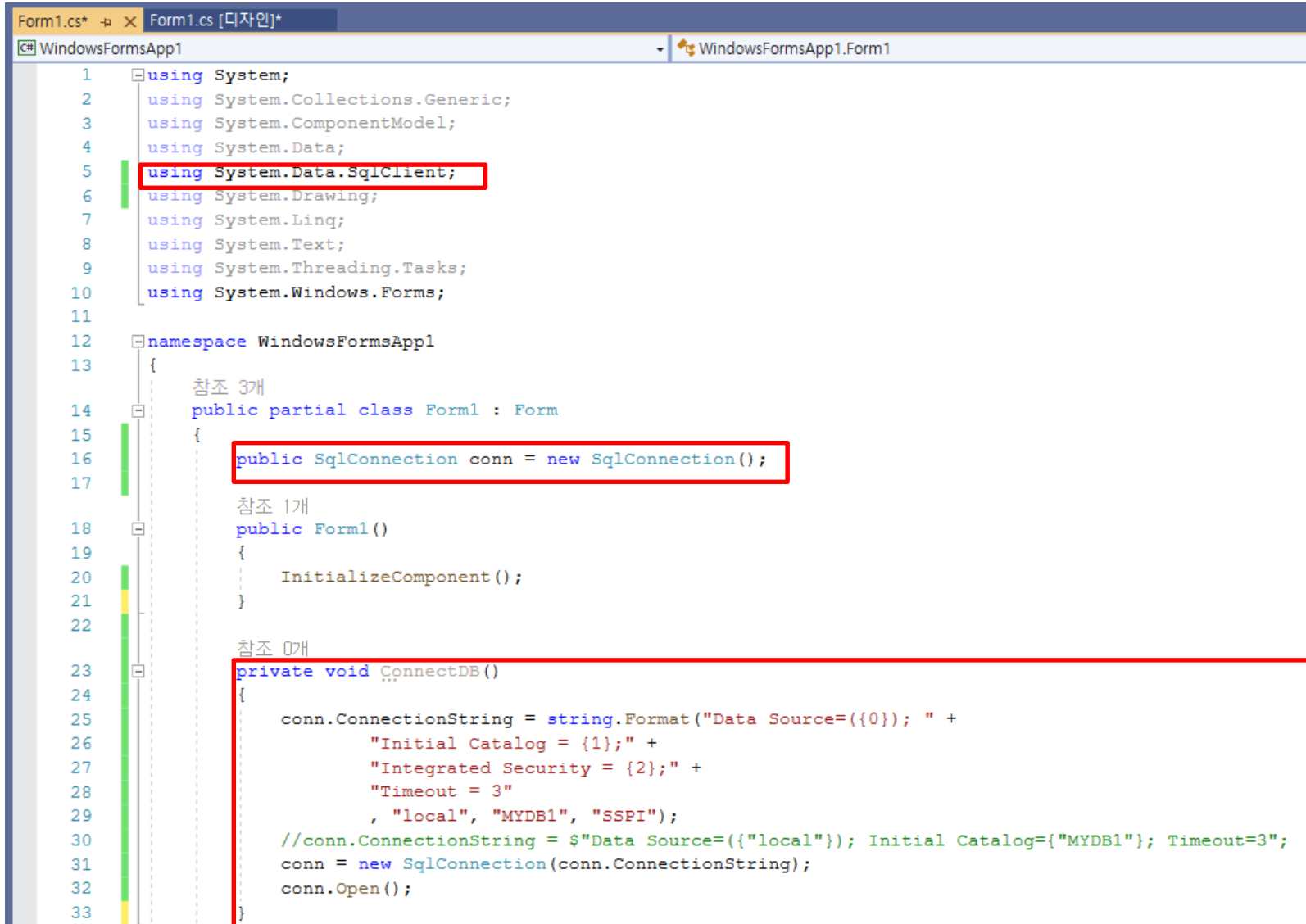


```
Form1.cs* -> Form1.cs [디자인]*
C# WindowsFormsApp1 -> WindowsFormsApp1

5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace WindowsFormsApp1
12 {
13     참조 3개
14     public partial class Form1 : Form
15     {
16         참조 1개
17         public Form1()
18         {
19             InitializeComponent();
20         }
21
22         private void button_Select_Click(object sender, EventArgs e)
23         {
24         }
25     }
}
```

Section 05 C#과 MSSQL 기능 연동하기 – MSSQL 연결

- MSSQL을 연결시 사용할 객체 및 함수 선언



```
Form1.cs* -> Form1.cs [디자인]*
C# WindowsFormsApp1 -> WindowsFormsApp1.Form1

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Data.SqlClient;
6  using System.Drawing;
7  using System.Linq;
8  using System.Text;
9  using System.Threading.Tasks;
10 using System.Windows.Forms;
11
12 namespace WindowsFormsApp1
13 {
14     참조 3개
15     public partial class Form1 : Form
16     {
17         public SqlConnection conn = new SqlConnection();
18
19         참조 1개
20         public Form1()
21         {
22             InitializeComponent();
23         }
24
25         참조 0개
26         private void ConnectDB()
27         {
28             conn.ConnectionString = string.Format("Data Source={0}; " +
29                 "Initial Catalog = {1}; " +
30                 "Integrated Security = {2}; " +
31                 "Timeout = 3"
32                 , "local", "MYDB1", "SSPI");
33             //conn.ConnectionString = $"Data Source=({\"local\"}); Initial Catalog=({\"MYDB1\"}); Timeout=3";
34             conn = new SqlConnection(conn.ConnectionString);
35             conn.Open();
36         }
37     }
38 }
```

Section 05 C#과 MSSQL 기능 연동하기 - 프로그램 연동(조회)

- '조회' 버튼 클릭시 Query_Select 함수 수행
- Select 기본 쿼리문 사용

참조 2개

```
private void Query_Select()
{
    ConnectDB();

    //SQL 명령어 선언
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = conn;
    cmd.CommandText = "SELECT * FROM TB_CUST";

    //DataAdapter와 DataSet으로 DB table 불러오기
    SqlDataAdapter da = new SqlDataAdapter(cmd); //select 구문이 들어감
    DataSet ds = new DataSet();
    da.Fill(ds, "TB_CUST"); //SELECT * FROM TB_CUST의 결과가 da에 입력됨

    //dataGridView에 DB에서 가져온 데이터 입력하기
    dataGridView1.DataSource = ds;
    dataGridView1.DataMember = "TB_CUST";
    conn.Close(); //연결 해제
}
```

참조 1개

```
private void button_Select_Click(object sender, EventArgs e)
{
    Query_Select();
}
```


Section 05 C#과 MSSQL 기능 연동하기 - 프로그램 연동(추가)

- '추가' 버튼 클릭시 Query_Insert 함수 수행

참조 1개

```
private void Query_Insert()  
{  
    ConnectDB();  
    string sqlcommand = "Insert Into TB_CUST (CUST_ID, BIRTH_DT) values (@parameter1,@parameter2)";  
    SqlCommand cmd = new SqlCommand();  
  
    cmd.Connection = conn;  
    cmd.CommandType = CommandType.Text;  
    //Column 명은 별도의 파라미터 형태로 선언함  
    //SQL Injection을 방지하고자 함 (SQL Injection : 유효하지 않은 데이터를 이용한 공격) 예: +나 ' 기호를 이용한 공격  
    cmd.Parameters.AddWithValue("@parameter1", textBox_ID.Text);  
    cmd.Parameters.AddWithValue("@parameter2", textBox_Birth.Text);  
    cmd.CommandText = sqlcommand;  
    cmd.ExecuteNonQuery(); //쿼리 실행  
    conn.Close();  
}
```

참조 1개

```
private void button_Insert_Click(object sender, EventArgs e)  
{  
    Query_Insert(); //테이블에 입력한 데이터 추가  
    Query_Select(); //추가된 데이터가 포함된 테이블 출력  
}
```



Thank You
