

데이터베이스 및 설계

MySQL을 위한 MFC를 사용한 ODBC 프로그래밍
(MFC ODBC Consumer 사용)



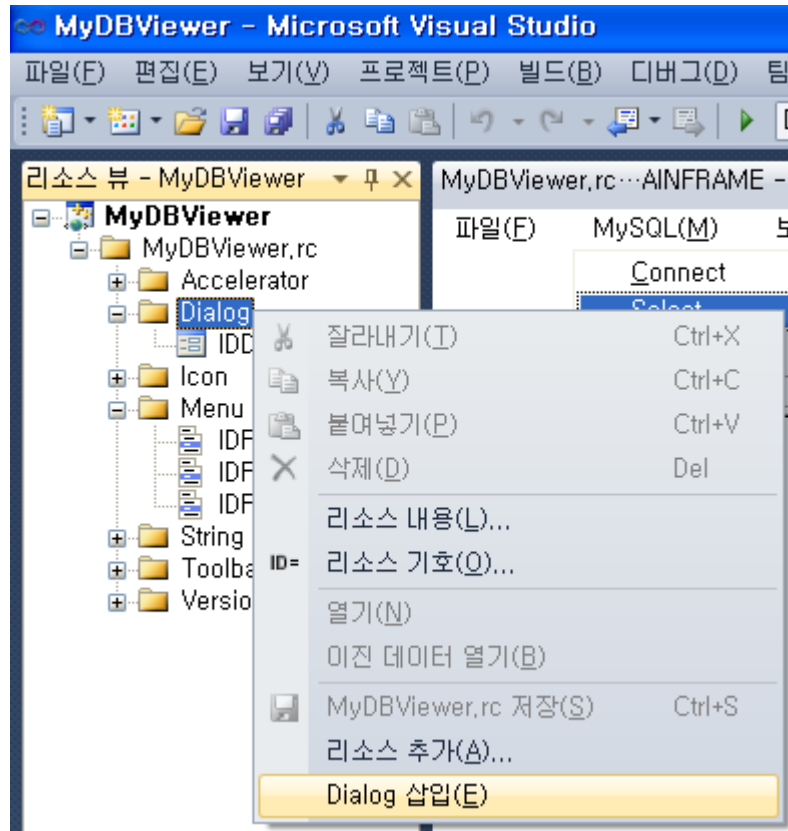
2012.06.04.

오 병 우

컴퓨터공학과
금오공과대학교

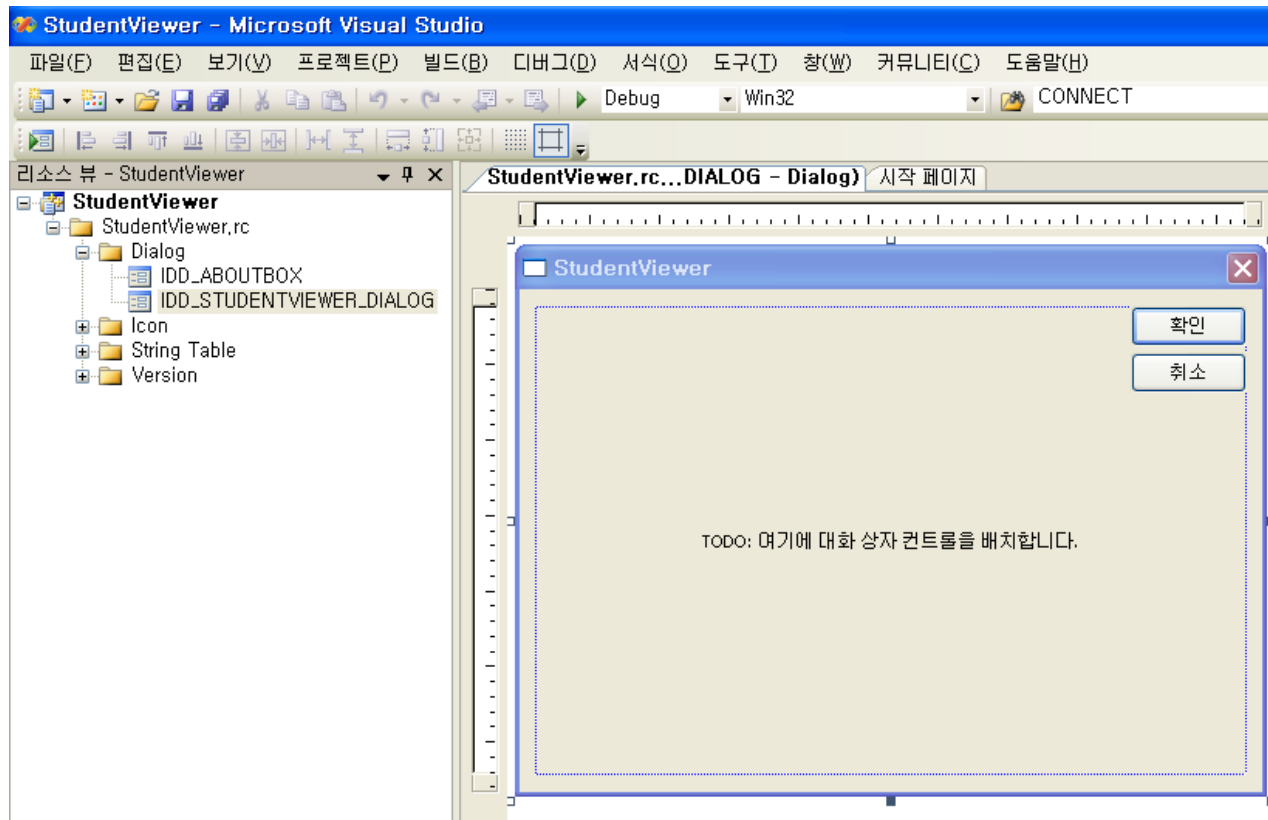
MyDBViewer 프로젝트에 추가

Dialog 삽입

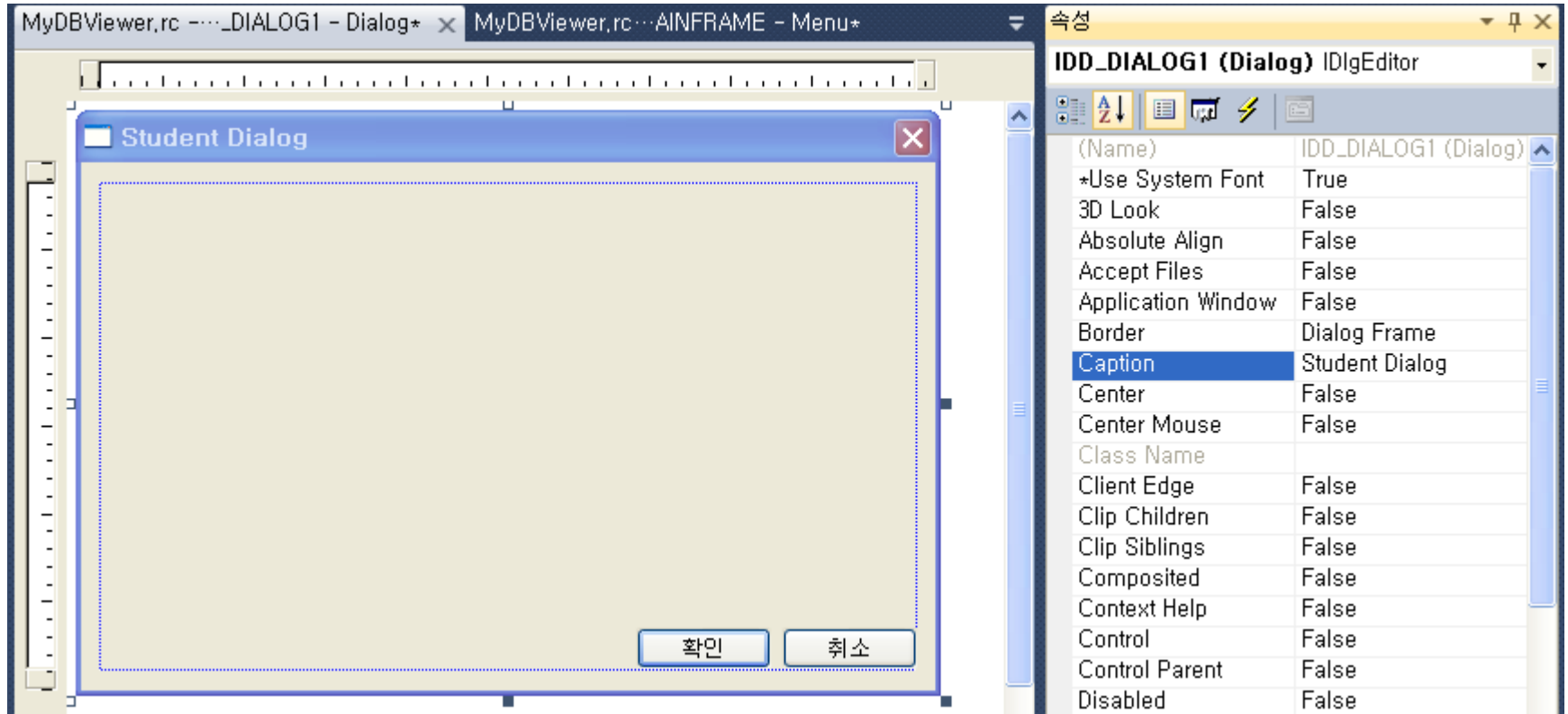


다이얼로그

리소스 뷰

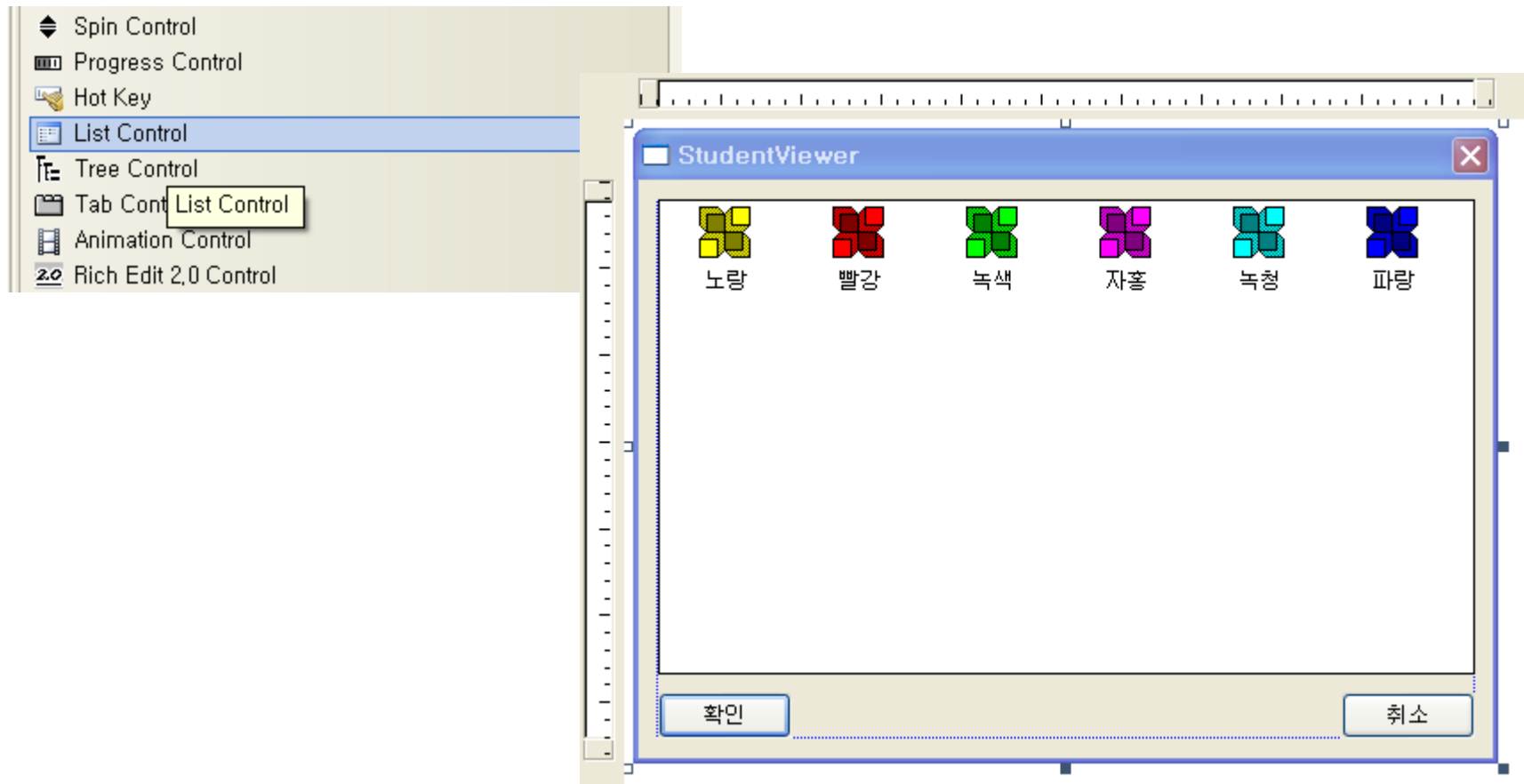


캡션 변경



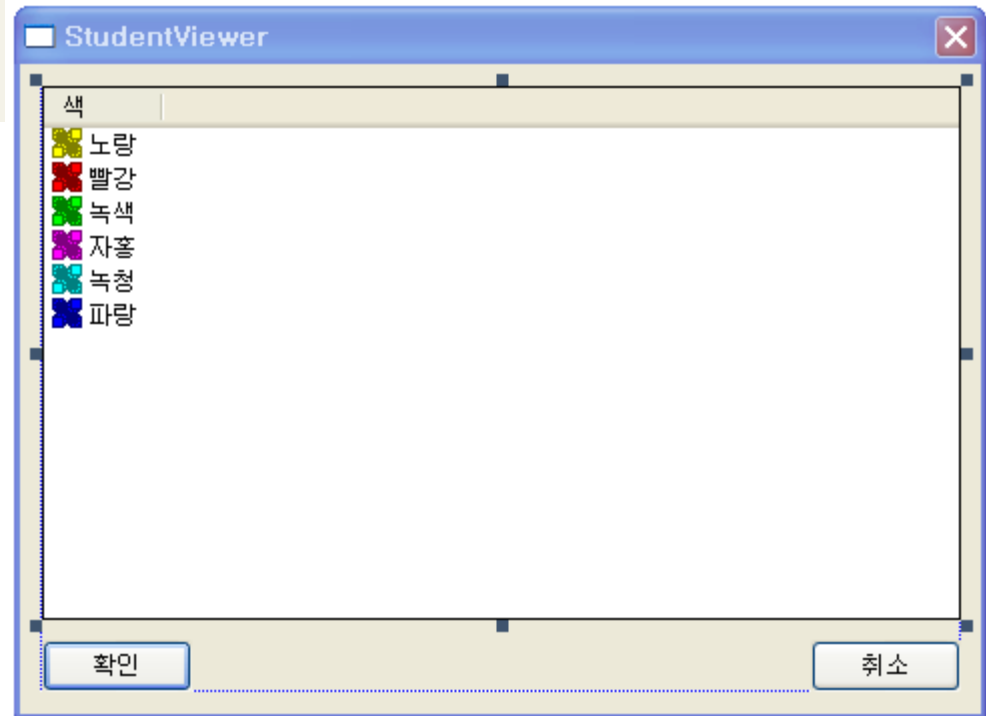
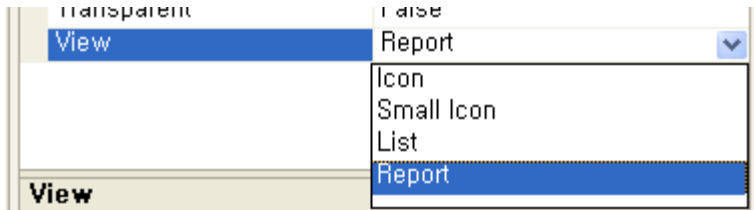
리스트 컨트롤 추가

ListControl

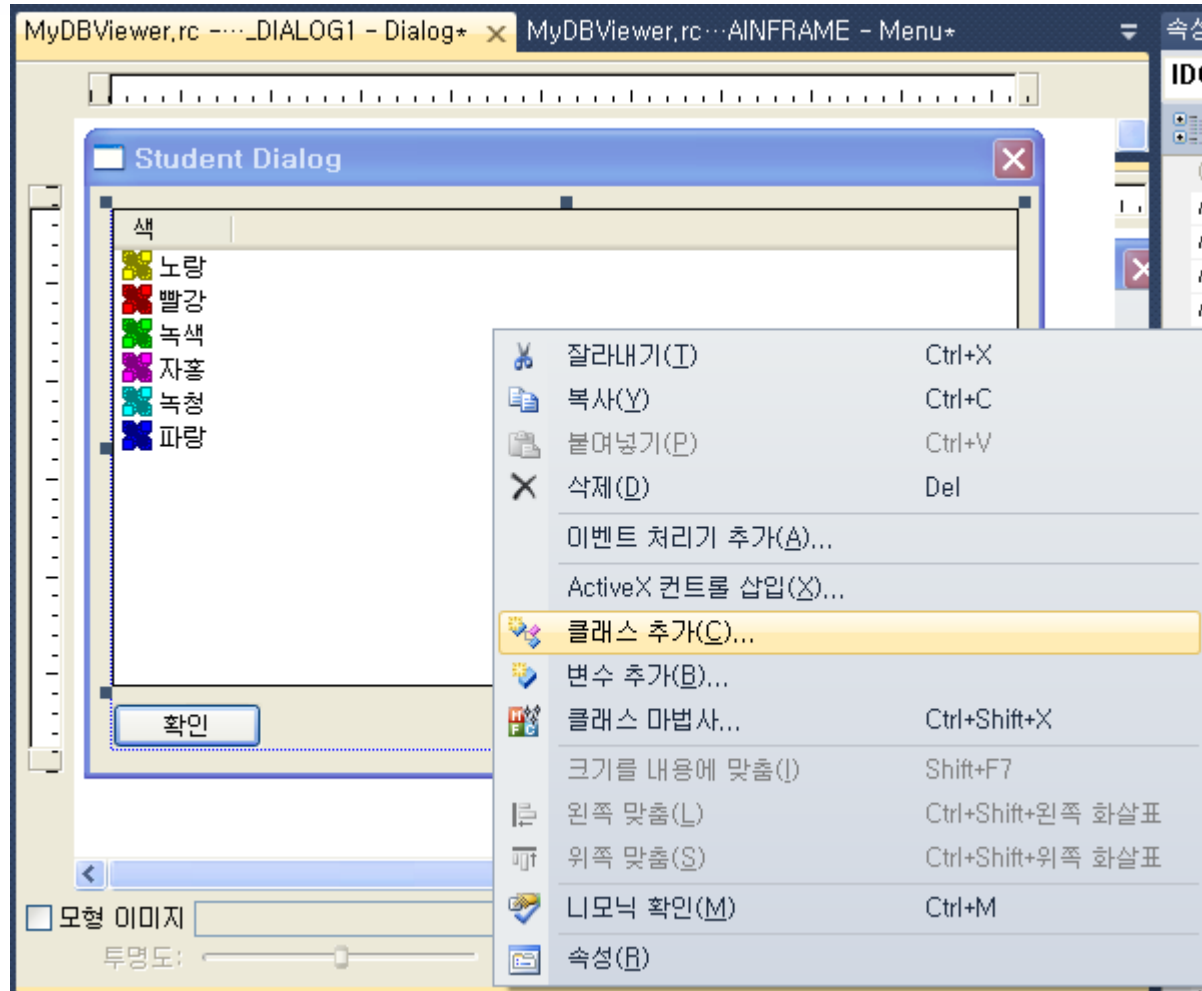


View

Report



Dialog 파생 클래스 추가



클래스 추가 마법사

MFC 클래스 추가 마법사 - MyDBViewer

MFC 클래스 추가 마법사 시작

이름
문서 템플릿 속성

클래스 이름(L): CStudentDlg

기본 클래스(B): CDialogEx

대화 상자 ID(D): IDD_DIALOG1

.h 파일(H): StudentDlg.h

.cpp 파일(P): StudentDlg.cpp

Active accessibility(Y)

DHTML 리소스 ID(S): IDR_HTML_STUDENTDLG

.HTM 파일(M): StudentDlg.htm

자동화:

- 없음(N)
- 자동화(A)
- 형식 ID로 생성 가능(E)

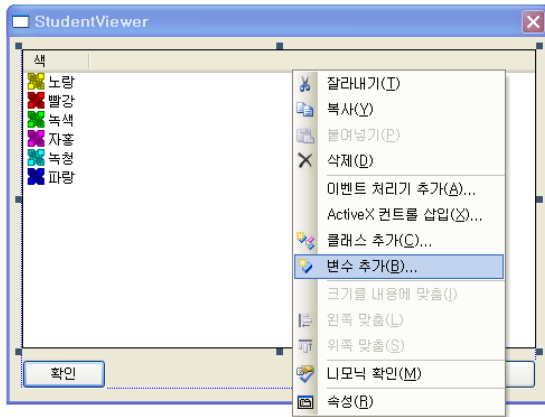
형식 ID(I): MyDBViewer.StudentDlg

DocTemplate 리소스 생성(G)

< 이전 다음 > 마침 취소

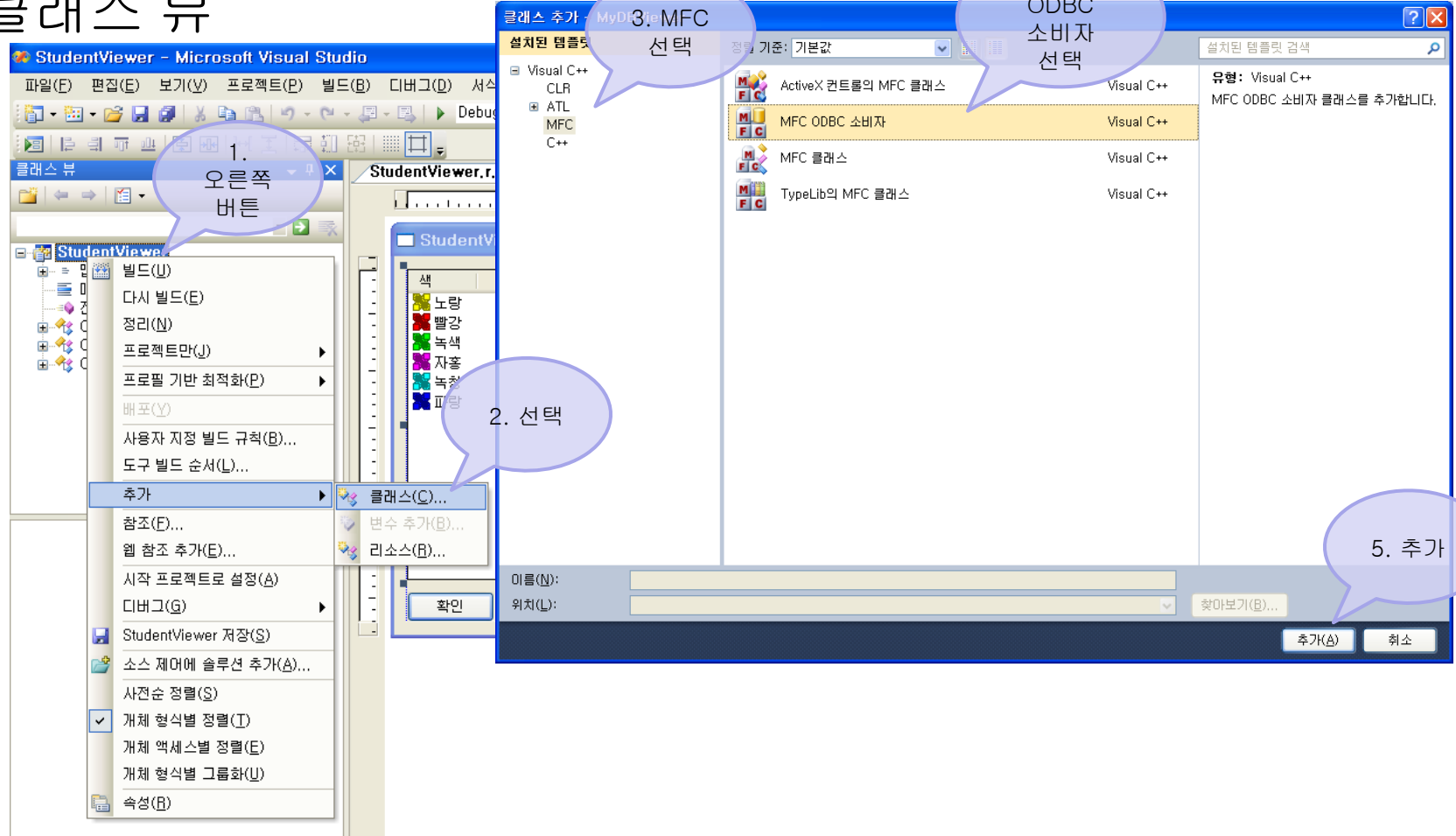
Control Type Member Variable

- 리스트 선택한 후에 Ctrl+더블클릭
- 또는 오른쪽 버튼
 - ◆ 변수 추가



MFC ODBC 소비자 클래스 추가

클래스 뷰



1. 오른쪽 버튼

2. 선택

3. 선택

4. MFC ODBC 소비자 선택

5. 추가

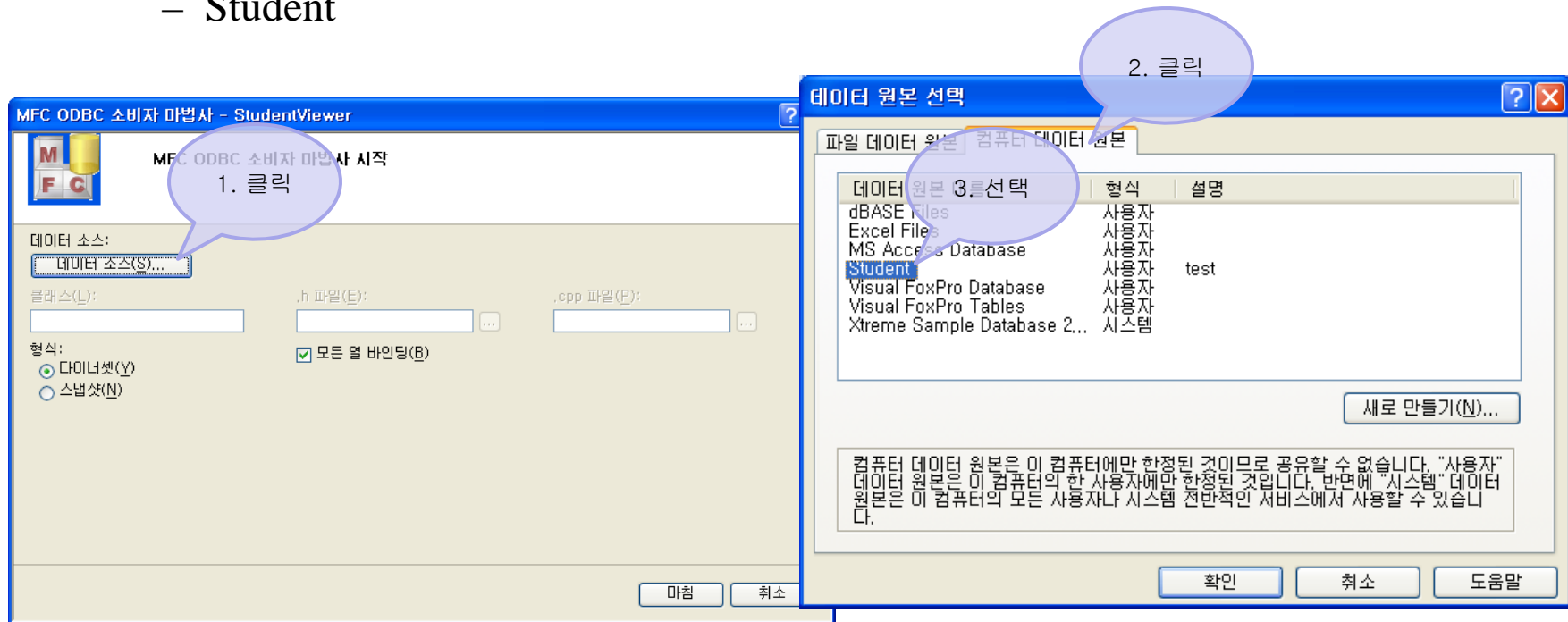
클래스 이름	유형
ActiveX 컨트롤의 MFC 클래스	Visual C++
MFC ODBC 소비자	Visual C++
MFC 클래스	Visual C++
TypeLib의 MFC 클래스	Visual C++

MFC ODBC Consumer Wizard

데이터 소스 선택

◆ DSN 선택

– Student



MFC ODBC 소비자 마법사 - StudentViewer

MFC ODBC 소비자 마법사 시작

1. 클릭

데이터 소스:
데이터 소스(S)...

클래스(L):
.h 파일(E):
.cpp 파일(P):

형식:
 데이터셋(Y)
 스냅샷(N)
 모든 열 바인딩(B)

마침 취소

데이터 원본 선택

파일 데이터 원본 컴퓨터 데이터 원본

2. 클릭

데이터 원본 3를 선택	형식	설명
dBASE Files	사용자	
Excel Files	사용자	
MS Access Database	사용자	
Student	사용자	test
Visual FoxPro Database	사용자	
Visual FoxPro Tables	사용자	
Xtreme Sample Database 2...	시스템	

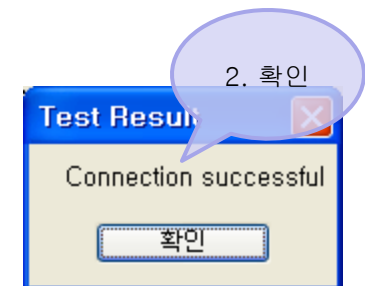
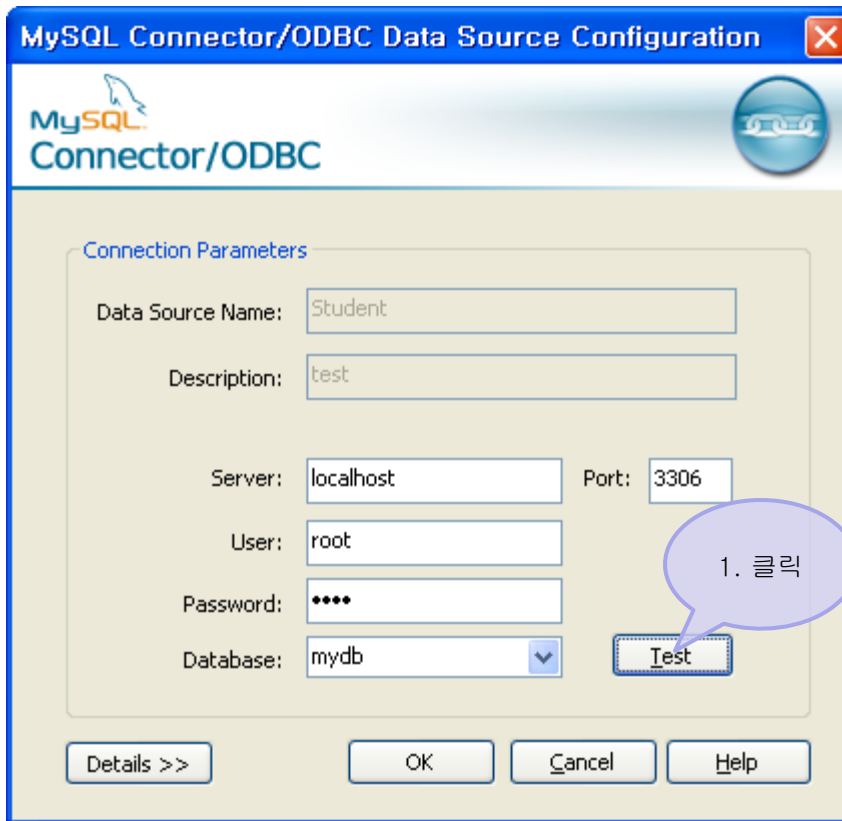
새로 만들기(N)...

컴퓨터 데이터 원본은 이 컴퓨터에만 한정된 것이므로 공유할 수 없습니다. "사용자" 데이터 원본은 이 컴퓨터의 한 사용자에만 한정된 것입니다. 반면에 "시스템" 데이터 원본은 이 컴퓨터의 모든 사용자나 시스템 전반적인 서비스에서 사용할 수 있습니다.

확인 취소 도움말

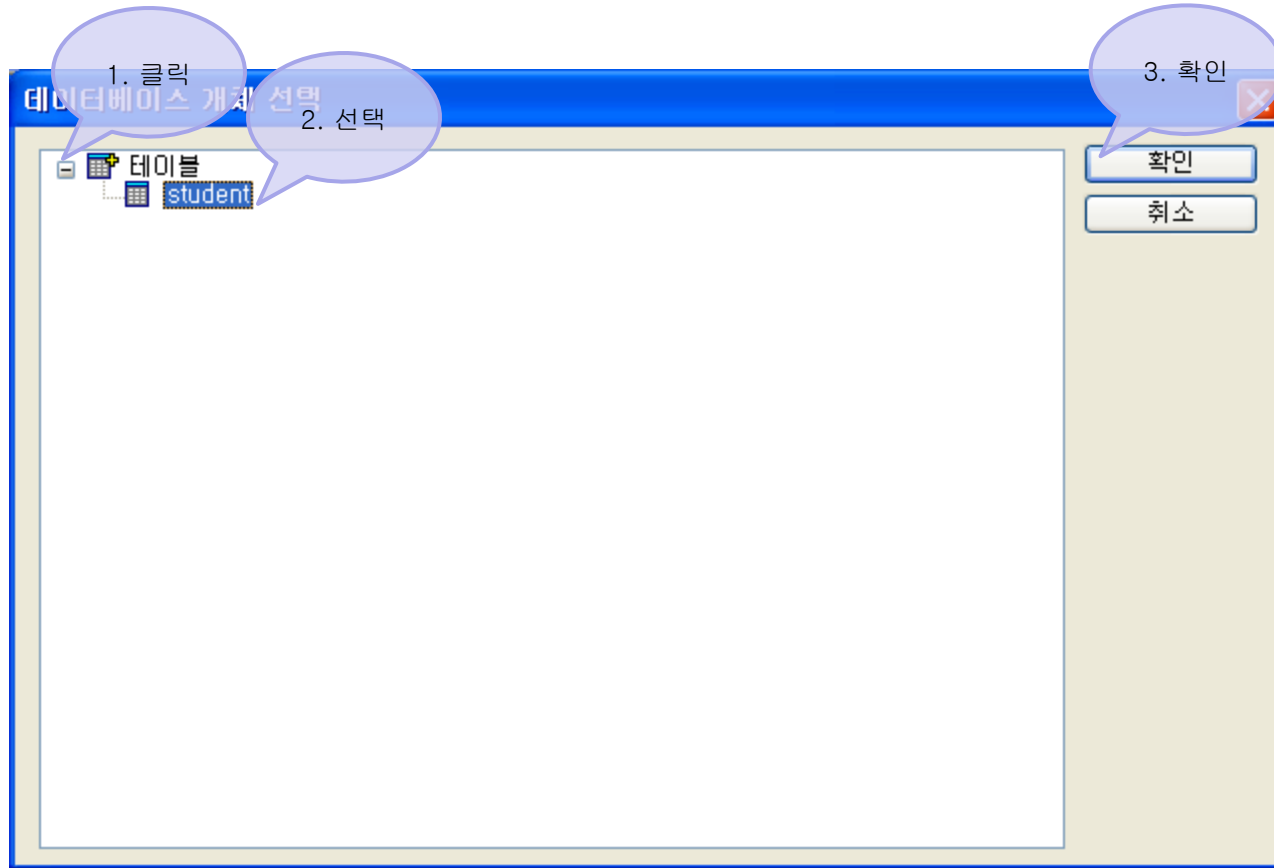
ODBC 연결 테스트

- 예전에 만들어 둔 데이터 소스 연결 테스트
 - ◆ ODBC 5.1 버전



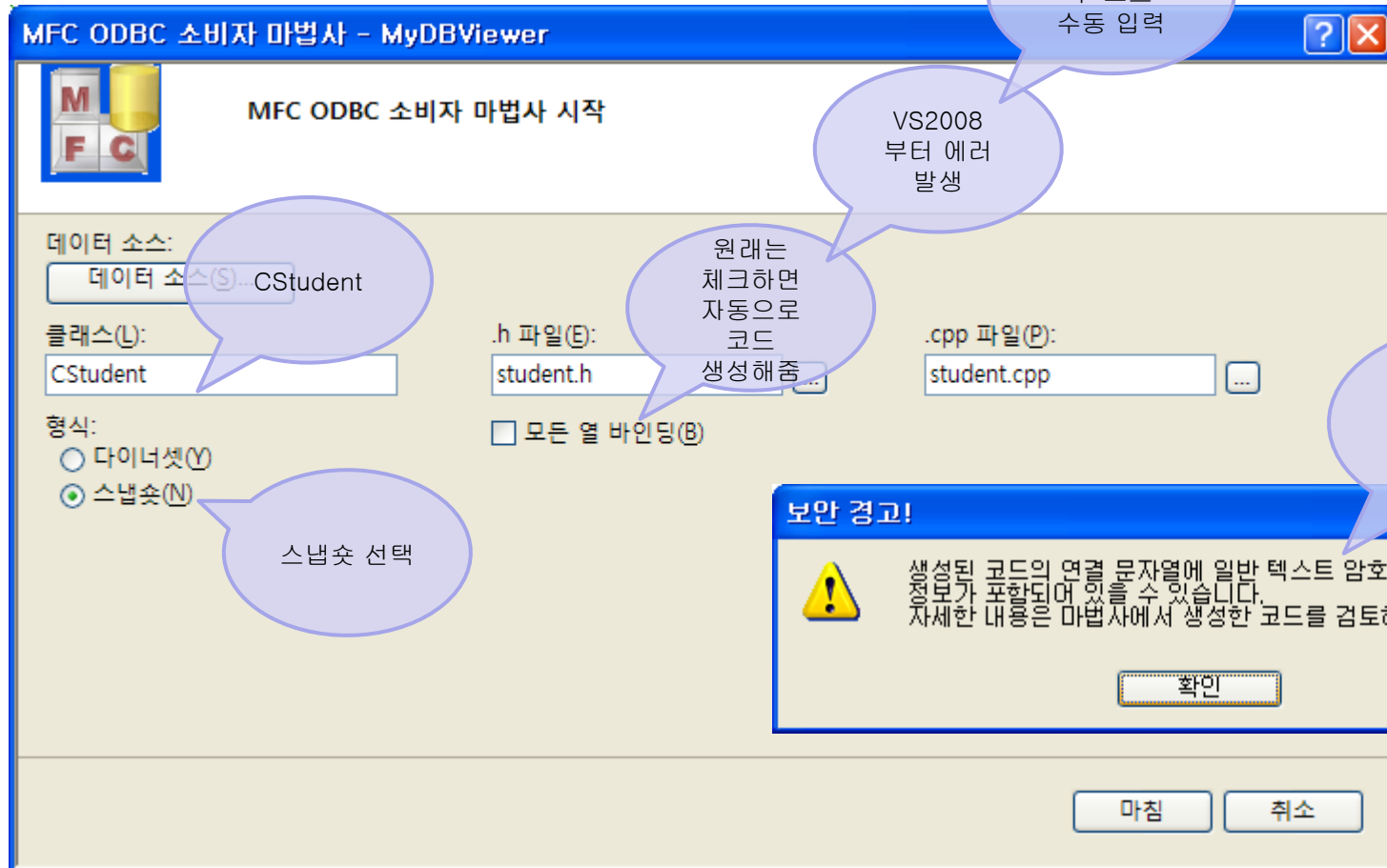
테이블 선택

Student



클래스 이름

스냅샷, 모든 열 바인딩 체크 해제



에러 발생

F7 누르면 에러 발생

- ◆ F4 눌러서 에러 발생한 곳으로 이동
- ◆ 경고 주석 처리

```

CStudent::CStudent(CDatabase* pdb)
    : CRecordset(pdb)
{
    // ...
    type = snapshot;
}
// #error Security Issue: The connection string may contain a password
// 아래 연결 문자열에 일반 텍스트 암호 및/또는
// 다른 중요한 정보가 포함되어 있을 수 있습니다.
// 보안 관련 문제가 있는지 연결 문자열을 검토한 후에 #error를(를) 제거하십시오.
// 다른 형식으로 암호를 저장하거나 다른 사용자 인증을 사용하십시오.
CString CStudent::GetDefaultConnect()

```

주석 처리

Column (열) Binding 수동 입력

Student.h 수정

```
class CStudent : public CRecordset
{
public:
    CStudent(CDatabase* pDatabase = NULL);
    DECLARE_DYNAMIC(CStudent)

// 필드/매개 변수 데이터

// 아래의 문자열 형식(있을 경우)은 데이터베이스 필드의 실제 데이터 형식을
// 나타냅니다(CStringA: ANSI 데이터 형식, CStringW: 유니코드 데이터 형식).
// 이것은 ODBC 드라이버에서 불필요한 변환을 수행할 수 없도록 합니다.
// 원할 경우 이들 멤버를 CString 형식으로 변환할 수 있으며
// 그럴 경우 ODBC 드라이버에서 모든 필요한 변환을 수행합니다.
// (참고: 유니코드와 이들 변환을 모두 지원하려면 ODBC 드라이버
// 버전 3.5 이상을 사용해야 합니다).
```

```
long        m_SNO;
CStringW    m_SNAME;
CStringW    m_MAJOR;
double      m_GPA;
```

변수
추가

유니코드
미사용시

```
CString    m_SNAME;
CString    m_MAJOR;
```


Column (열) Binding 수동 입력

Student.cpp 수정

◆ 생성자

```

#include "stdafx.h"
#include "student.h"
IMPLEMENT_DYNAMIC(CStudent, CRecordset)

CStudent::CStudent(CDatabase* pdb)
    : CRecordset(pdb)
{
    m_SNO = 0;
    m_SNAME = L"";
    m_MAJOR = L"";
    m_GPA = 0.0;

    m_nFields = 4;
    //m_nFields = 0;
    m_nDefaultType = dynaset;
}
    
```

유니코드
미사용시

```

m_SNAME = "";
m_MAJOR = "";
    
```

Column (열) Binding 수동 입력

- Student.cpp 수정
 - ◆ GetDeafultSQL()

```

CString CStudent::GetDefaultSQL()
{
    return _T("Select * from student");
    //return _T("[student]"); // 이렇게 해도 됨
}
    
```

Column (열) Binding 수동 입력

Student.cpp 수정

◆ DoFieldExchange()

```

void CStudent::DoFieldExchange(CFieldExchange* pFX)
{
    pFX->SetFieldType(CFieldExchange::outputColumn);
    // RFX_Text() 및 RFX_Int() 같은 매크로는 데이터베이스의 필드
    // 형식이 아니라 멤버 변수의 형식에 따라 달라집니다.
    // ODBC에서는 자동으로 열 값을 요청된 형식으로 변환하려고 합니다
    RFX_Long(pFX, _T("[SNO]"), m_SNO);
    RFX_Text(pFX, _T("[SNAME]"), m_SNAME);
    RFX_Text(pFX, _T("[MAJOR]"), m_MAJOR);
    RFX_Double(pFX, _T("[GPA]"), m_GPA);
}

```

ShowTable() 함수 추가

함수 추가.
헤더파일에도
public으로
추가

맨 위에
헤더파일
(student.h)
include 추가

유니코드
미사용시

```
int CStudentDlg::ShowTable()
{
    CStudent db;

    if (!db.Open())
        return ERROR_CONNECTION_UNAVAIL;

    m_List.DeleteAllItems();
    //db.Execute("set names euckr");

    int i = 0;
    CString str;
    //CStringW str;
    while (!db.IsEOF()) {

        str.Format(_T("%d"), db.m_SNO);
        m_List.InsertItem(i, str);

        //str = db.m_SNAME; //m_List.SetItemText(i, 1, str);
        m_List.SetItemText(i, 1, db.m_SNAME);

        //str = db.m_MAJOR; //m_List.SetItemText(i, 2, str);
        m_List.SetItemText(i, 2, db.m_MAJOR);

        str.Format(_T("%.2f"), db.m_GPA);
        m_List.SetItemText(i, 3, str);

        db.MoveNext();
        i++;
    }

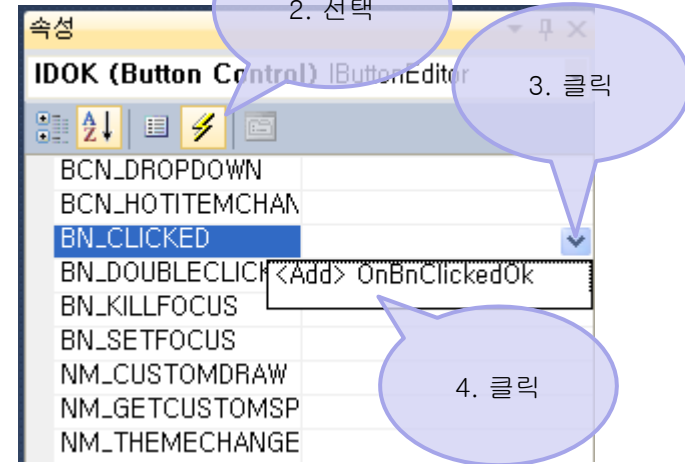
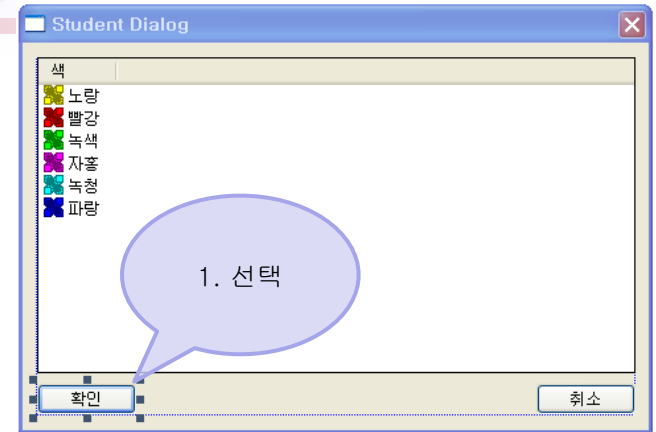
    db.Close();
    return S_OK;
}
```

이벤트 처리기 추가

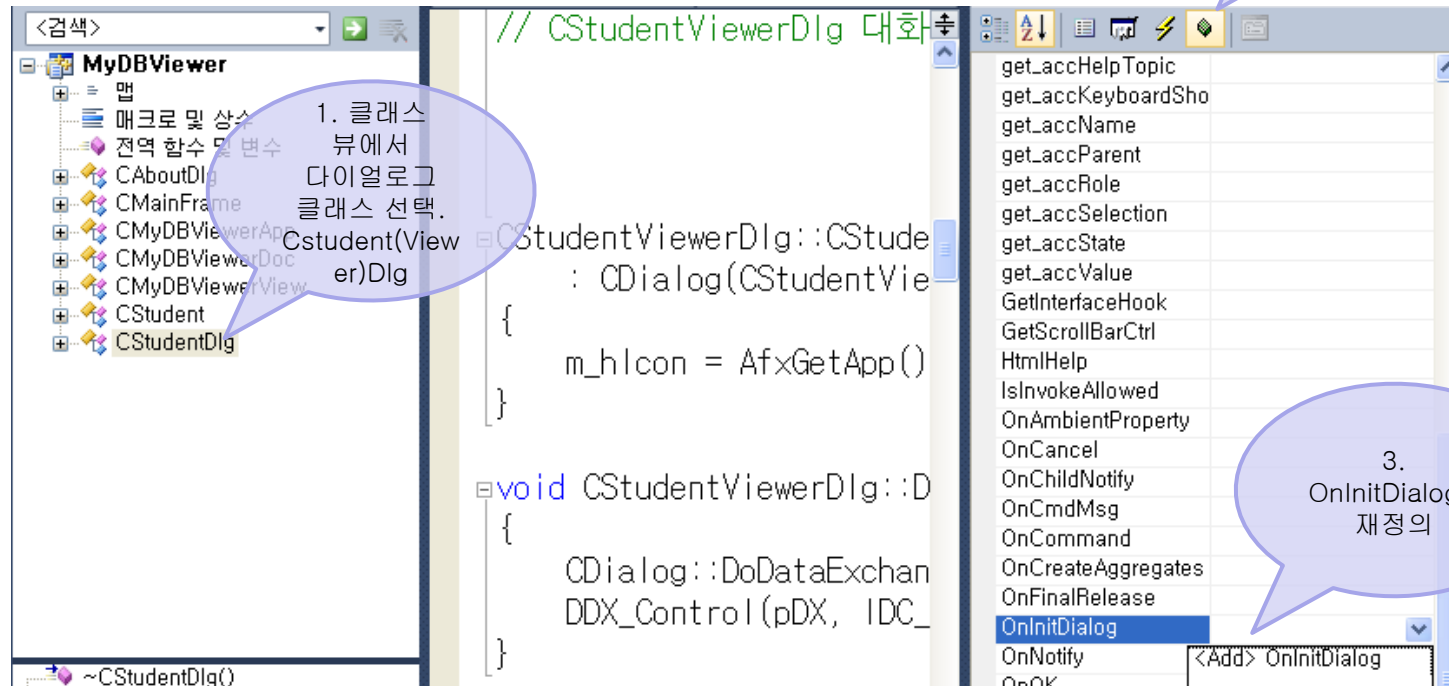
```
void CStudentDlg::OnBnClickedOk()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
    //CDialogEx::OnOK();
    ShowTable();
}
```

수정

// TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.
 //CDialogEx::OnOK();
 ShowTable();



Dialog에 OnInitDialog() 재정의



The screenshot shows the Visual Studio IDE with the following components:

- Left Panel (Solution Explorer):** Shows a project named 'MyDBViewer'. A callout bubble points to the 'CStudentDlg' class, with the text: "1. 클래스 뷰에서 다이얼로그 클래스 선택." (1. Class selection in the class view).
- Center Panel (Code Editor):** Shows the implementation of the dialog class:


```
// CStudentViewerDlg 대화 상자
class CStudentViewerDlg : CDialog(CStudentViewerDlg)
{
public:
    CStudentViewerDlg(CStudentViewerDlg* pParent, CWnd* pWnd): CDialog(CStudentViewerDlg, pParent, pWnd)
    {
        m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
    }

protected:
    void DoDataExchange(CDataExchange* pDX)
    {
        CDialog::DoDataExchange(pDX);
        DDX_Control(pDX, IDC_EDIT1, m_edit1);
    }
};
```
- Right Panel (Class View):** Shows the list of methods for the class. A callout bubble points to the 'OnInitDialog()' method, with the text: "2. 재정의 선택" (2. Selection of override). Another callout bubble points to the 'OnInitDialog()' method in the list, with the text: "3. OnInitDialog() 재정의" (3. OnInitDialog() override).

코드 추가

```

BOOL CStudentDlg::OnInitDialog()
{
    CDialogEx::OnInitDialog();

    // TODO: 여기에 추가 초기화 작업을 추가합니다.

    m_List.InsertColumn(0, _T("SNO"), LVCFMT_CENTER, 80);
    m_List.InsertColumn(1, _T("SNAME"), LVCFMT_CENTER, 80);
    m_List.InsertColumn(2, _T("MAJOR"), LVCFMT_CENTER, 120);
    m_List.InsertColumn(3, _T("GPA"), LVCFMT_CENTER, 80);

    return TRUE; // return TRUE unless you set the focus to a control
    // 예외: OCX 속성 페이지는 FALSE를 반환해야 합니다.
}

```

다이얼로그 실행

● MyDBViewerDoc.cpp

◆ Dialog를 사용하기 위해서 헤더 파일 import

```

// MyDBViewerDoc.cpp : CMyDBViewerDoc 클래스의 구현
//

#include "stdafx.h"
// SHARED_HANDLERS는 미리 보기, 축소판 그림 및 검색 필터 처리
// 해당 프로젝트와 문서 코드를 공유하도록 해 줍니다.
#ifdef SHARED_HANDLERS
#include "MyDBViewer.h"
#endif

#include "MyDBViewerDoc.h"

#include "studentDlg.h"
    
```


다이얼로그 실행

OnMysqlSelect() 수정

```

void CMyDBViewerDoc::OnMysqlSelect()
{
    //UpdateAllViews(NULL);
    CStudentDlg dlg;
    dlg.DoModal();
}
    
```

결과

- MySQL – Select, 확인 버튼 클릭

