

MFC 환경에서 데이터 베이스 접근하기

OLE-DB

충남대학교 영상시스템 연구실

Database

Database

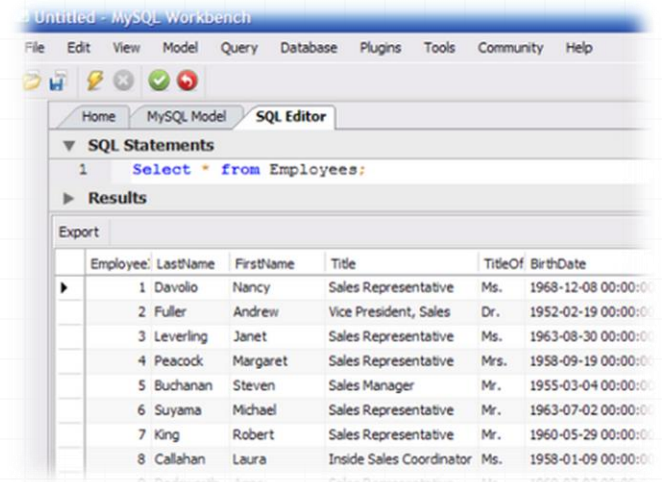
- 데이터를 효율적으로 처리하기 위해서 개발
- 자료를 구조화해서 저장시켜놓은 높은 수준의 자료들의 집합체
- 자료 중복 문제 해결, 검색과 갱신의 효율화

DBMS (Database Management System)

- 데이터베이스 내의 데이터를 접근할 수 있도록 해주는

소프트웨어 도구

- Oracle, MS-SQL, My-SQL, MS-Access, SQL-Lite 등



OLE-DB

○ 데이터 베이스와의 연결 방법

- ODBC
- OLE-DB
- ADO.NET

○ OLE-DB (Object Linking and Embedding, Database)

- 마이크로소프트사가 개발한 API로, 통일된 방식으로 저장된 여러 종류의 데이터에 접근하기 위해 만들어졌다.

○ API

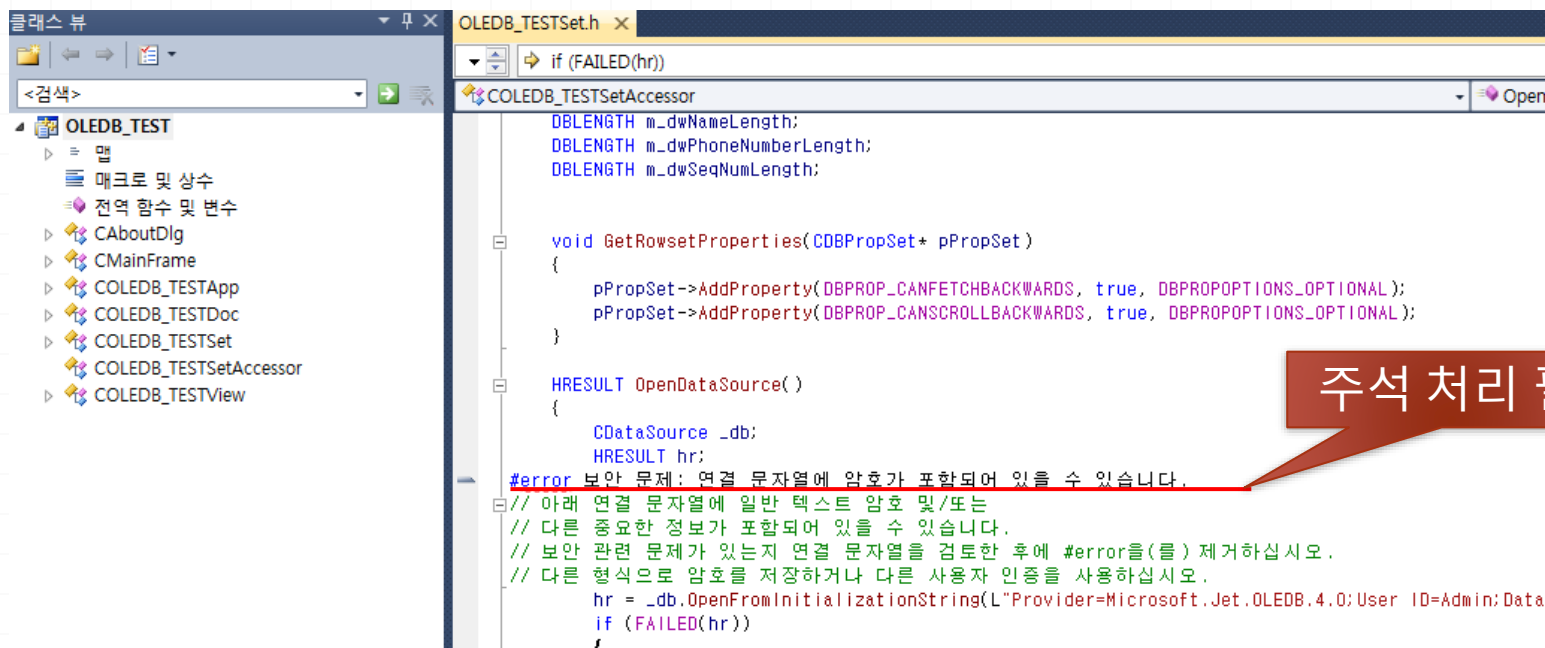
- 프로그램이 운영 체제에 어떤 처리를 위해서 호출할 수 있는 서브루틴 또는 함수의 집합

MFC 환경에서 데이터 베이스 접근하기

- Database와 연결된 프로젝트 생성

MFC 환경에서 데이터 베이스 접근하기

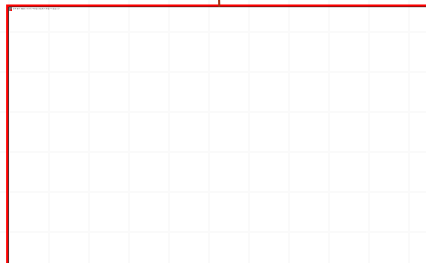
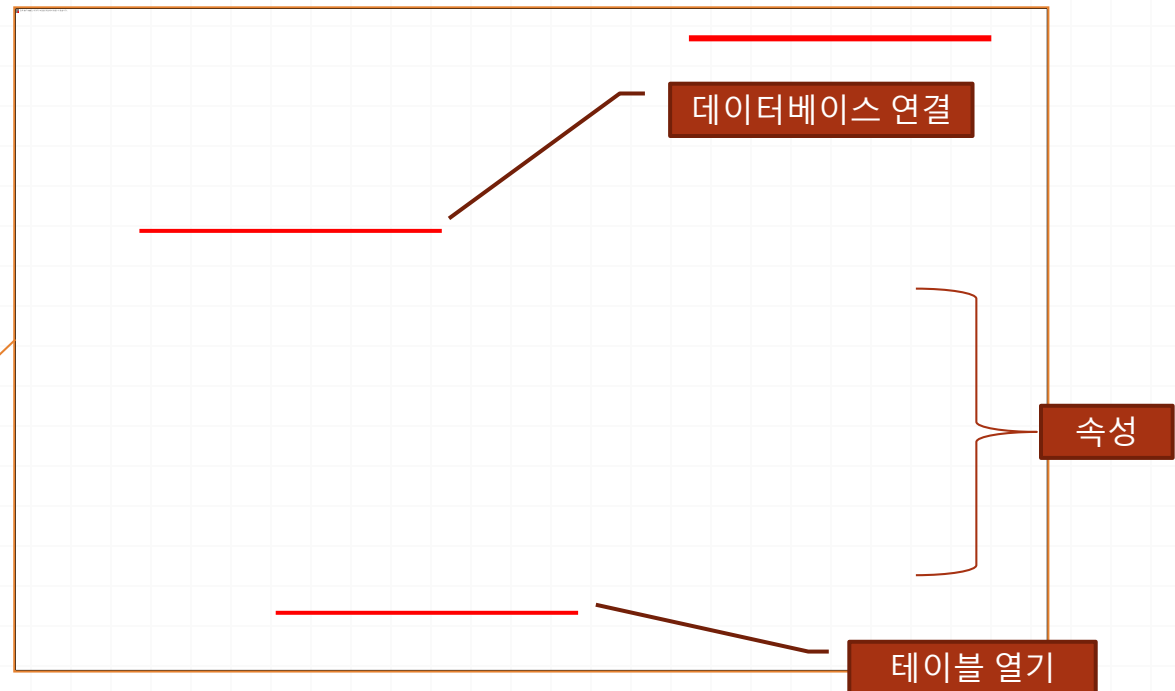
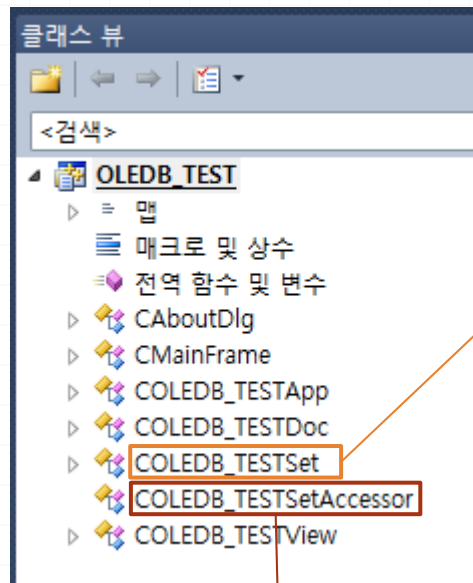
Database와 연결된 프로젝트 생성



MFC 환경에서 데이터 베이스 접근하기

프로젝트 분석

테이블 객체화



MFC 환경에서 데이터 베이스 접근하기

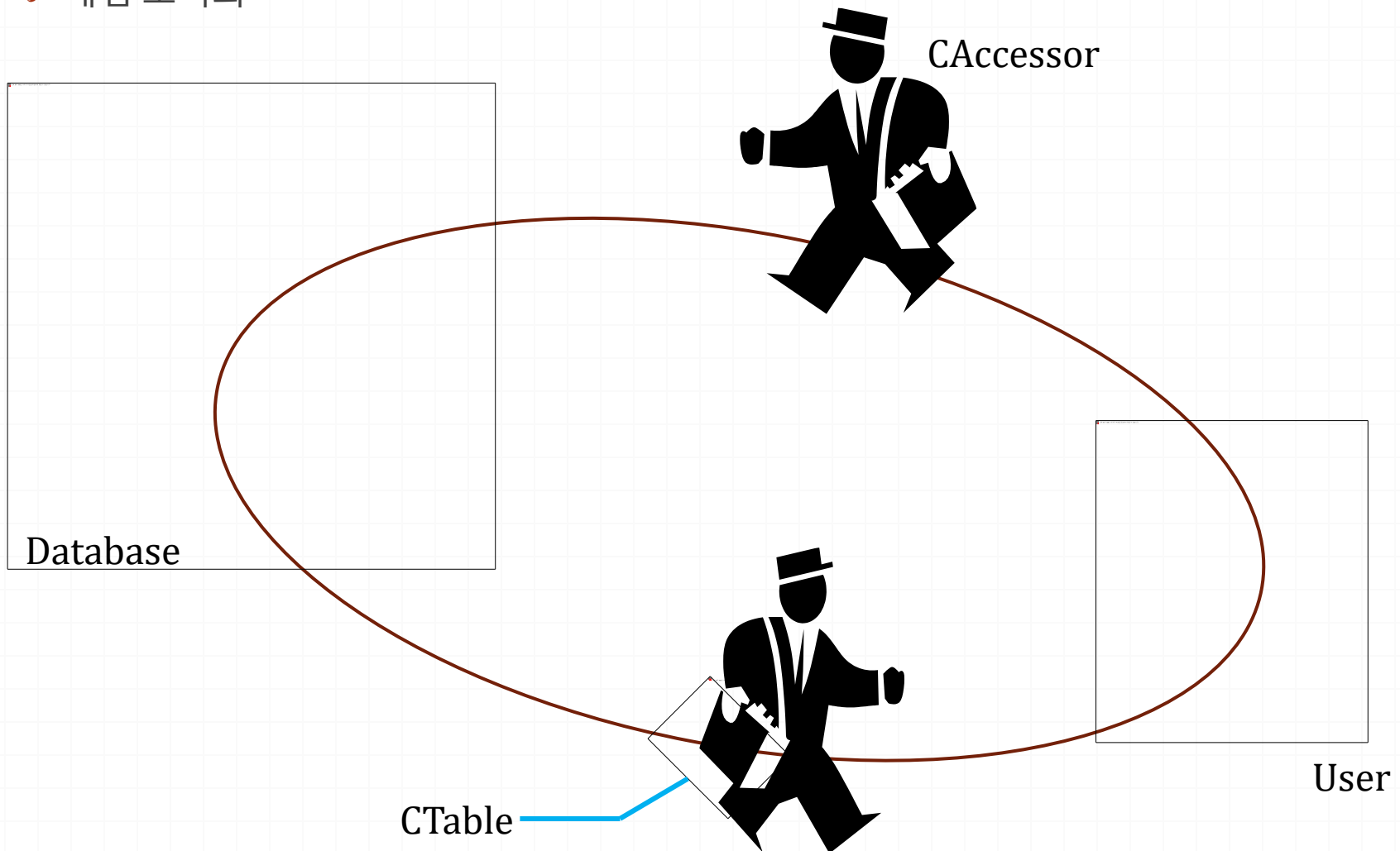
프로젝트 분석

```
HRESULT OpenDataSource()  
{  
    CDataSource _db;  
    HRESULT hr;  
    hr = _db.OpenFromInitializationString(L"Provider=Microsoft.Jet.OLEDB.4.0;User ID=Admin  
    if (FAILED(hr))  
    {  
        return hr;  
    }  
    return m_session.Open(_db);  
}
```

```
void GetRowsetProperties(CDBPropSet* pPropSet)  
{  
    pPropSet->AddProperty(DBPROP_CANFETCHBACKWARDS,  
        true, DBPROPOPTIONS_OPTIONAL);  
    pPropSet->AddProperty(DBPROP_CANSCROLLBACKWARDS,  
        true, DBPROPOPTIONS_OPTIONAL);  
}
```

MFC 환경에서 데이터 베이스 접근하기

o 개념 도식화



MFC 환경에서 데이터 베이스 접근하기

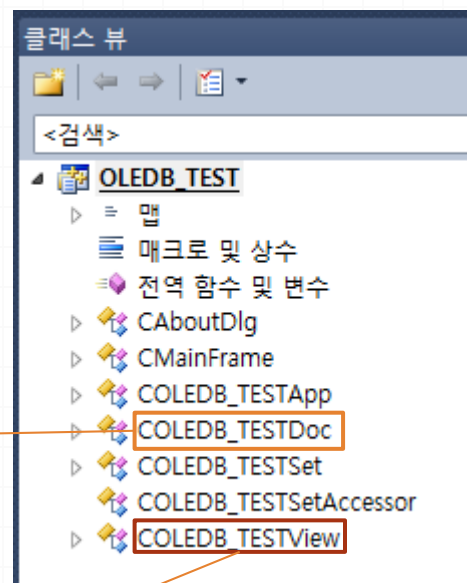
프로젝트 분석

OLEDB_TESTDoc.h

```
class COLEDB_TESTDoc : public CDocument
{
protected: // serialization에서만 만들어집니다.
    COLEDB_TESTDoc();
    DECLARE_DYNCREATE(COLEDB_TESTDoc)

    // 특성입니다.
public:
    COLEDB_TESTSet m_OLEDB_TESTSet;

    // 자여입니다
```



OLEDB_TESTView.h

```
class COLEDB_TESTView : public COleDBRecordView
{
protected: // serialization에서만 만들어집니다.
    COLEDB_TESTView();
    DECLARE_DYNCREATE(COLEDB_TESTView)

public:
    enum{ IDD = IDD_OLEDB_TEST_FORM };
    COLEDB_TESTSet* m_pSet;

    // 특성입니다
```

OLEDB_TESTView.cpp

```
void COLEDB_TESTView::OnInitialUpdate()
{
    m_pSet = &GetDocument()->m_OLEDB_TESTSet;
    {
        CWaitCursor wait;
        HRESULT hr = m_pSet->OpenAll();
        if (FAILED(hr))
        {
            // 레코드 집합을 열지 못했습니다. 레
```


MFC 환경에서 데이터 베이스 접근하기

0 레코드 추가, 갱신, 삭제를 위한 준비

1. 데이터 베이스에 대한 접근 권한 변경

```
void GetRowsetProperties(CDBPropSet* pPropSet)
{
    pPropSet->AddProperty(DBPROP_CANFETCHBACKWARDS, true, DBPROPOPTIONS_OPTIONAL);
    pPropSet->AddProperty(DBPROP_CANSROLLBACKWARDS, true, DBPROPOPTIONS_OPTIONAL);

    pPropSet->AddProperty(DBPROP_UPDATABILITY,
        DBPROPVAL_UP_CHANGE |
        DBPROPVAL_UP_INSERT |
        DBPROPVAL_UP_DELETE);
}
```

2. 상속 클래스 변경

```
//class COLEDB_TESTSet : public CTable<CAccessor<COLEDB_TESTSetAccessor> >
class COLEDB_TESTSet : public CCommand<CAccessor<COLEDB_TESTSetAccessor> >
{
public:
    HRESULT OpenAll()
    {
        HRESULT hr;
        hr = OpenDataSource();
        if (FAILED(hr))
            return hr;
        __if_exists(GetRowsetProperties)
        {
            CDBPropSet propset(DBPROPSET_ROWSET);
```

CTable : 단순 조회

CCommand : SQL 문(명령)을 실행할 수 있는 클래스

3. OpenRowset 함수 수정

```
HRESULT OpenRowset(DBPROPSET *pPropSet = NULL)
{
    //HRESULT hr = Open(m_session, L"UserData", pPropSet);
    HRESULT hr = Open(m_session, "SELECT * FROM UserData", pPropSet);
}
def ndpic
```

MFC 환경에서 데이터 베이스 접근하기

○ 레코드 추가

```
HRESULT RunSQL(CString strSQL, DBPROPSET *pPropSet = NULL){
    Close();
    ReleaseCommand();
    HRESULT hr = Open(m_session, strSQL, pPropSet);
    return hr;
}
```

```
void COLEDB_TESTView::OnBnClickedAdd()
{
    CCommand<CDynamicAccessor> Cmd;

    CDBPropSet propset(DBPROPSET_ROWSET);
    propset.AddProperty(DBPROP_IRowsetChange, true);
    propset.AddProperty(DBPROP_UPDATABILITY,
        DBPROPVAL_UP_CHANGE |
        DBPROPVAL_UP_INSERT |
        DBPROPVAL_UP_DELETE);

    HRESULT hr = Cmd.Open(m_pSet->m_session,
        _T("SELECT * FROM UserData"), &propset);
    if(FAILED(hr))
    {
        AfxMessageBox(_T("ERROR: Failed@"));
    }

    TCHAR* pszValue = NULL;
    Cmd.SetStatus(1, DBSTATUS_S_IGNORE);

    //이름
    pszValue = (TCHAR*)Cmd.GetValue(2);
    wprintf(pszValue, _T("%s"), _T("김철수"));
    Cmd.SetLength(2, lstrlen(pszValue) * 2);
    Cmd.SetStatus(2, DBSTATUS_S_OK);
}
```

MFC 환경에서 데이터 베이스 접근하기

○ 레코드 수정

```
//상태가 DBSTATUS_S_OK 인 필드의 값을 수정
hr = Cmd.SetData( );
if(FAILED(hr))
{
    AfxMessageBox(_T("ERROR: Failed to modify record!"));
}

//데이터베이스에 업데이트
Cmd.Update( );

Cmd.Close( );
Cmd.ReleaseCommand( );

if(m_List.GetItemCount( ) > 0)      m_List.DeleteAllItems( );
m_pSet->RunSQL(_T("SELECT * FROM UserData"));
ListupAllRecords( );
}
```

MFC 환경에서 데이터 베이스 접근하기

○ 레코드 삭제

```
void COLEDB_TESTView::OnBnClickedButtonDelete()  
{  
    CCommand<CDynamicAccessor> Cmd;  
  
    COBPropSet propset(DBPROPSET_ROWSET);  
    propset.AddProperty(DBPROP_IRowsetChange, true);  
    propset.AddProperty(DBPROP_UPDATABILITY,  
        DBPROPVAL_UP_CHANGE |  
        DBPROPVAL_UP_INSERT |  
        DBPROPVAL_UP_DELETE);  
  
    //유용길 탐색  
    HRESULT hr = Cmd.Open(m_pSet->m_session,  
        _T("SELECT * FROM UserData WHERE Name = '유용길'"),  
        &propset);  
    if(FAILED(hr))  
    {  
        AfxMessageBox(_T("ERROR: Failed to execute SQL!"));  
    }  
  
    //유용길은 하나밖에 없으므로 맨 처음 데이터가 유일한 데이터  
    Cmd.MoveFirst();  
  
    //유용길 제거  
    hr = Cmd.Delete();  
    if(FAILED(hr))  
    {  
        AfxMessageBox(_T("ERROR: Failed to delete record!"));  
    }  
}
```