데이터베이스 II

(12주차)

학습개요

- 학습 목표
  - OLE DB 프로그램 구조를 이해한다.
  - MFC를 이용한 OLE DB 프로그래밍 기법을 익힌다.

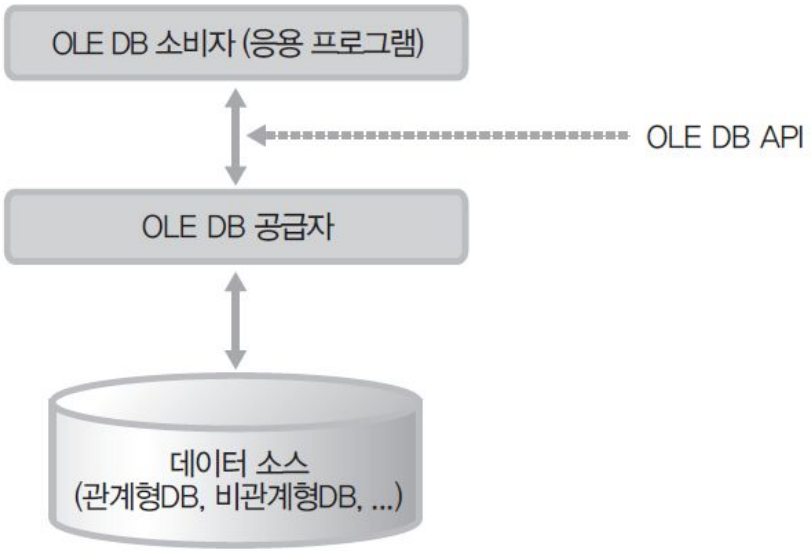
- 학습 내용
  - OLE DB 프로그램 구조
  - OLE DB 프로그래밍 (콘솔)
  - OLE DB 프로그래밍 (GUI)
  - 실습

OLE DB 프로그래밍

- OLE DB 구조
  - ODBC와 마찬가지로 단일 API를 사용해 다양한 데이터에 접근
  - MS의 COM 기술에 기반하므로 윈도우 운영체제에 종속적
  - 전통적인 DBMS로 제공되는 관계형 데이터베이스는 물론이고 그 밖의 다양한 데이터에도 접근

OLE DB 프로그래밍

- OLE DB 구조



  - OLE DB 소비자 : OLE DB 인터페이스를 통해 데이터에 액세스하는 시스템이나 응용 프로그램
  - OLE DB 공급자 : 실제 OLE DB 인터페이스를 구현하는 소프트웨어 컴포넌트
  - 데이터 소스 : OLE DB 공급자의 종류에 따라 다양한 형태

MFC OLE DB 프로그래밍 (콘솔)

- OLE DB는 복잡한 COM 기술을 기반으로 하므로 구현이 어렵고 많은 코드를 작성해야 하는 문제가 있음
- 하지만 MFC에서는 복잡한 부분을 미리 템플릿 클래스 형태로 구현하여 제공

OLE DB 코드 분석

- 레코드 출력

```
m_pSet->MoveFirst();
do {
    _tprintf(_T("%d %s %2d %s₩n"),
    m_pSet->m_ST_NUM,
    m_pSet->m_ST_NAME,
    m_pSet->m_ST_MAN,
    m_pSet->m_ST_PHONE);
} while(m_pSet->MoveNext() == S_OK);
```

OLE DB 코드 분석

- 레코드 추가 (1/2)

```
m_pSet->MoveLast(); // 맨 끝으로 이동한다.

// 레코드 추가 전에 반드시 상태 변수를 초기화해야 한다!
m_pSet->m_dwST_NUMStatus = DBSTATUS_S_OK;
m_pSet->m_dwST_NAMEStatus = DBSTATUS_S_OK;
m_pSet->m_dwST_MANStatus = DBSTATUS_S_OK;
m_pSet->m_dwST_PHONEStatus = DBSTATUS_S_OK;

// 문자열과 같은 가변 필드는 길이를 초기화해야 한다!
m_pSet->m_dwST_NAMELength = sizeof(_T("홍길동"));
m_pSet->m_dwST_PHONELength = sizeof(_T("010-1234-5678"));
```

OLE DB 코드 분석

- 레코드 추가 (2/2)

```
// 필드 변수에 새로운 레코드의 필드 값을 넣는다.
m_pSet->m_ST_NUM = 3010;
wsprintf(m_pSet->m_ST_NAME, _T("홍길동"));
m_pSet->m_ST_MAN = -1;
wsprintf(m_pSet->m_ST_PHONE, _T("010-1234-5678"));

// 레코드를 추가한다.
hr = m_pSet->Insert();
if(FAILED(hr)){
    AfxMessageBox(_T("레코드를 추가하지 못했습니다."), MB_OK);
}
```

OLE DB 코드 분석

- 레코드 변경 (1/2)

```
m_pSet->MoveLast(); // 맨 끝으로 이동한다.

// 레코드 변경 전에 반드시 상태 변수를 초기화해야 한다!
m_pSet->m_dwST_NUMStatus = DBSTATUS_S_OK;
m_pSet->m_dwST_NAMEStatus = DBSTATUS_S_OK;
m_pSet->m_dwST_MANStatus = DBSTATUS_S_OK;
m_pSet->m_dwST_PHONEStatus = DBSTATUS_S_OK;

// 문자열과 같은 가변 필드는 길이를 초기화해야 한다!
m_pSet->m_dwST_PHONELength = sizeof(_T("010-5678-1234"));
```

OLE DB 코드 분석

- 레코드 변경 (2/2)

```
// 필드 변수에 변경할 레코드의 필드 값을 넣는다.
wsprintf(m_pSet->m_ST_PHONE, _T("010-5678-1234"));

// 레코드를 변경한다.
hr = m_pSet->SetData();
if(FAILED(hr)){
    AfxMessageBox(_T("레코드를 변경하지 못했습니다."), MB_OK);
}
```

OLE DB 코드 분석

- 레코드 삭제

```
m_pSet->MoveLast(); // 맨 끝으로 이동한다.

// 레코드를 삭제한다.
hr = m_pSet->Delete();
if(FAILED(hr)){
    AfxMessageBox(_T("레코드를 삭제하지 못했습니다."), MB_OK);
}
```

실습

MFC OLE DB 프로그래밍 (GUI)

- 뷰의 기본 클래스로 COleDBRecordView 선택
- CRecordView 대신 COleDBRecordView 클래스 사용
- CRecordset 대신 CTable/CAccessor/CRowset 클래스 사용

실습

학습정리

- OLE DB는 ODBC와 마찬가지로 단일 API를 사용해 다양한 데이터에 접근하지만, MS의 COM 기술에 기반하므로 윈도우 운영체제에 종속적입니다.
- OLE DB는 전통적인 DBMS로 제공되는 관계형 데이터베이스는 물론이고 그 밖의 다양한 데이터에도 접근이 가능합니다.
- OLE DB는 OLE DB 인터페이스를 통해 데이터에 액세스하는 OLE DB 소비자, 실제 OLE DB 인터페이스를 구현하는 OLE DB 공급자, OLE DB 공급자의 종류에 따른 다양한 형태의 데이터 소스로 구성됩니다.
- MFC를 이용한 OLE DB GUI 프로그래밍을 할 경우 뷰의 기본 클래스로 COleDBRecordView 선택하고, ODBC의 CRecordView 대신 COleDBRecordView 클래스를, CRecordset 대신 CTable / CAccessor / CRowset 클래스를 사용합니다.