

ODBC 3

Mit SQL Recordset erstellen

1. SQL-Anweisung erstellen

a) Datensätze sortieren nach der Schuhgröße

```
CString sSQL = "SELECT * FROM [Personen] ORDER BY [Schuhgröße]";
```

b) alle Datensätze mit Name ‚Kohnle‘ auswählen

```
CString sSQL = "SELECT * FROM [Personen] WHERE [Name] = 'Kohnle'";
```

c) alle Datensätze auswählen, deren Name mit ‚Ko‘ anfängt

```
CString sSQL = "SELECT * FROM [Personen] WHERE [Name] LIKE 'Ko%'";
```

d) alle Datensätze mit Schuhgröße ‚42‘ auswählen

```
CString sSQL = "SELECT * FROM [Personen] WHERE [Schuhgröße] = 42";
```

2. Recordset öffnen

```
m_pSet->Open(CRecordset::dynaset, sSQL); //statt dynaset evtl snapshot
```

Recordset leer ?

```
if(m_pSet->IsEOF() && m_pSet->IsBOF()){
    // kein Datensatz vorhanden
}else{
    // mindestens ein Datensatz vorhanden
}
```

Navigation

```
m_pSet->MoveFirst();           // zum ersten Datensatz
m_pSet->MovePrevious();       // zum vorherigen Datensatz
m_pSet->MoveNext();           // zum nächsten Datensatz
m_pSet->MoveLast();           // zum letzten Datensatz
m_pSet->SetAbsolutePosition(1); // zum ersten Datensatz
```

aktuellen Datensatz anzeigen

```
UpdateData(FALSE);
```

aktuellen Datensatz verändern

```
m_pSet->Edit();
m_pSet->m_Name = "..."; //entsprechende Membervariablen verwenden!!!
...
m_pSet->Update();
```

aktuellen Datensatz löschen

```
m_pSet->Delete();
```

alle Datensätze durchlaufen

```
m_pSet->MoveFirst();
while(!m_pSet->IsEOF()){
    ... = m_pSet->m_Name; //entsprechende Memberv. verwenden!!!
    ...
    m_pSet->MoveNext();
}
```

Recordset schliessen

```
m_pSet->Close();
```

neuen Datensatz erstellen

1. aktuellen Datensatz speichern

```

if(m_pSet->CanUpdate() && !m_pSet->IsBOF() && !m_pSet->IsDeleted()){
    m_pSet->Edit();
    UpdateData(TRUE);
    m_pSet->Update();
}

```

2. neuen Datensatz erstellen

```

m_pSet->AddNew();
m_pSet->m_Name = "Neu"; //entsprechende Membervariablen verwenden!!!
...
m_pSet->Update();

```

3. neuen Datensatz anzeigen

```

if(m_pSet->CanRestart()){
    m_pSet->Requery();
    m_pSet->MoveLast();
}
UpdateData(FALSE);

```

Bookmarks (in C...View)

1. Recordset so öffnen, dass Bookmarks verwendet werden können

```

m_pSet->Open(CRecordset::dynaset, "SELECT * FROM [Tabelle]",
            CRecordset::useBookmarks);

```

2. Lesezeichen speichern

```

CDBVariant Lesezeichen;
m_pSet->GetBookmark(Lesezeichen);

```

3. zu Lesezeichen springen

```

m_pSet->SetBookmark(Lesezeichen);

```

Datumswerte verwenden (RFX_Date funktioniert nicht)

1. C...Set.h (Datentyp der Variablen ändern)

```

COleDateTime m_Datum;

```

2. Klassenassistent (Membervariable für Eingabefeld)

```

COleDateTime m_oledtDatum;

```

3. C...View

```

void C...View::DoDataExchange(CDataExchange* pDX) {
    CRecordView::DoDataExchange(pDX);
    if(pDX->m_bSaveAndValidate == FALSE)
        m_oledtDatum = m_pSet->m_Datum;

    //{{AFX_DATA_MAP(C...View)
    DDX_ ...// mindestens 1 DDX-Funktion ist notwendig!
    //}}AFX_DATA_MAP

    if(pDX->m_bSaveAndValidate == TRUE)
        m_pSet->m_Datum = m_oledtDatum;
}

```

4. C...Set Konstruktor (Initialisierung ändern):

```

m_Datum = COleDateTime::GetCurrentTime();

```

Zeiger von C...Doc auf C...View

```

POSITION Pos = GetFirstViewPosition();
C...View* pView;
pView = (C...View*)GetNextView(Pos);

```