CHƯƠNG 6
LẬP TRÌNH CƠ SỞ DỮ LIỆU

1. Khái niệm về kỹ thuật DAO

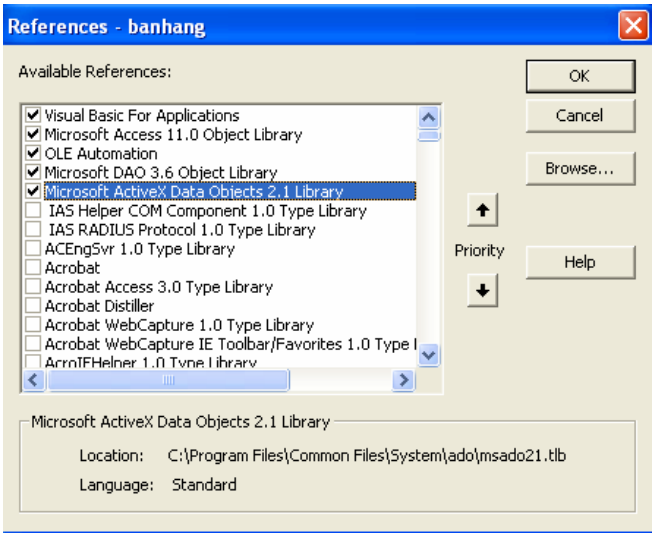
DAO (Data Access Objects – Các đối tượng truy xuất dữ liệu) là tập hợp bao gồm lớp các đối tượng có thể dùng để lập trình truy cập và xử lý dữ liệu trong các hệ CSDL. Ở đây CSDL Access, ngôn ngữ lập trình VBA.

1.1 Nạp thư viện DAO

Để nạp thư viện DAO3.6 vào làm việc, hãy thực hiện như sau:

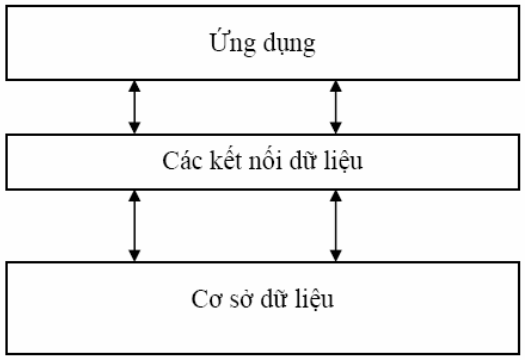
Bước 1: Mở cửa sổ lập trình VBA;

Bước 2: Chọn thực đơn **Tools | References ..** Hộp thoại sau xuất hiện:



Hãy chọn (tích) mục **Microsoft DAO 3.6 Object Libraly** trên danh sách *Available References*; chọn xong, nhấn **OK** để đóng lại.

1.2 Sơ đồ liên kết quản lý cơ sở dữ liệu



Trong đó:

- Tầng ứng dụng: bao gồm những giao diện người sử dụng cũng như những công cụ đơn giản mà người lập trình có thể dùng để xử lý dữ liệu theo các bài toán.

- Tầng Kết nối dữ liệu: bao gồm tập hợp các công cụ, phương thức để kết nối tới những dữ liệu cần làm việc trong CSDL. Ở đây, tầng kết nối bao gồm các chuẩn Microsoft Jet 4.0 và các lớp đối tượng DAO.

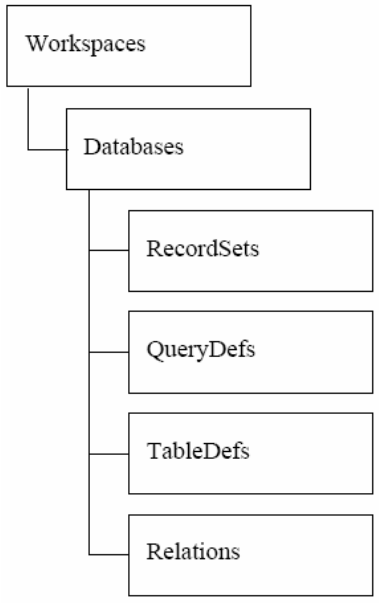
- Tầng Cơ sở dữ liệu: bao gồm các bảng, các query trong cơ sở dữ liệu thực tại.

Như vậy để lập trình trên một CSDL phải sử dụng các đối tượng, các phương thức ở tầng kết nối như là những công cụ để có thể truy cập được vào CSDL tác nghiệp xử lý. Tầng kết nối đó chính là Jet 4.0 và DAO 3.6 mà chúng ta sẽ được tìm hiểu dưới đây.

2. Lớp đối tượng DAO

Cấu trúc một CSDL bao gồm nhiều thành phần, đòi hỏi lập trình cũng cần có những thành phần tương ứng để làm việc. Lớp các thành phần tương ứng để có thể lập trình được trên toàn bộ cấu trúc CSDL là lớp các đối tượng DAO. Chúng có tên gọi, có những tập thuộc tính, các phương thức làm việc và có quan hệ mật thiết với nhau.

2.1 Cây phân cấp lớp các đối tượng DAO



Trong đó:

• **Workspaces** – định nghĩa tập hợp các vùng làm việc. Đây có thể coi là lớp làm việc cao nhất. Về lý thuyết có thể khai báo một vài vùng làm việc (Workspace), những trên thực tế chỉ cần khai báo một vùng làm việc và vùng

này luôn được khai báo ngầm định cho CSDL hiện tại. Nên sẽ không cần bàn nhiều đến lớp các WorkSpace này.

• **Databases** - định nghĩa tập hợp các CSDL Access cần làm việc trên một dự án

• **RecordSets**- định nghĩa các tập hợp bản ghi (Records) cần làm việc.

• **QueryDefs** - định nghĩa tập hợp các Query để làm việc. QueryDefs và Recordsets là khả năng truy xuất, xử lý dữ liệu (Data Manipulation) của DAO.

• **TableDefs** - định nghĩa tập hợp các bảng (Table) cần làm việc. Đây là khả năng định nghĩ dữ liệu (Data-Definition Language);

• **Relations** - định nghĩa tập hợp các quan hệ (Relationship) cần làm việc.

Mỗi lớp các đối tượng trên sẽ bao gồm tất cả các đối tượng đối tượng cùng loại trong một đối tượng mẹ đang mở:

- Databases sẽ bao gồm tất cả các CSDL đang đực mở trong vùng làm việc hiện tại;

- RecordSets sẽ bao gồm tập hợp tất cả các Recordset đang được mở trên CSDL hiện tại.

Khi đó, để tham chiếu đến một đối tượng cụ thể cần làm việc, có thể dùng chỉ số (số thứ tự của đối tượng đó trên tập hợp tất cả các đối tượng đó) hoặc dùng tên gọi đối tượng đó để tham chiếu.

Ví dụ: liệt kê tên của tất cả các Recordset đang sử dụng trong CSDL db.

```
Dim db As DAO.Database
For i = 0 To db.Recordsets.Count
    MsgBox db.Recordsets(i).Name
Next
```

Để làm việc tới một đối tượng cụ thể, cần phải tham chiếu từ lớp các đối tượng mẹ của nó.

Ví dụ: Để hiển thị giá trị của trường (Field) hoten trên tập hợp các bản ghi (Recordset) rs1 làm như sau:

```
MsgBox rs1.Fields("hoten").Value
```

hoặc

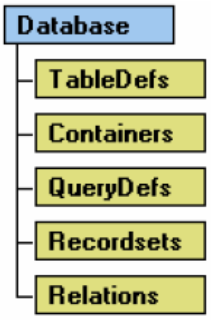
```
MsgBox rs1.Fields![hoten].Value
```

2.2 Đối tượng Database

Database là đối tượng dùng làm việc với một CSDL (trong trường hợp này có

thể hiểu một CSDL như một tệp Access .MDB).

Lớp các đối tượng con của Database được thể hiện qua sơ đồ sau:



Khai báo

```
Dim db As DAO.Database
' Gán db cho một CSDL cụ thể
Set db = OpenDatabase("C:\Baitap\qlbh.mdb")
'Đặc biệt, lệnh gán db cho CSDL hiện tại như sau:
Set db = CurrentDb
```

Khi không làm việc với CSDL nào đó, có thể ra lệnh đóng để giải phóng bộ
nhớ bằng cách:

db.Close

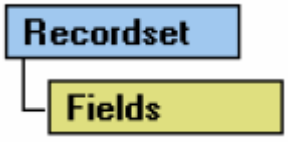
Sau khi lệnh này thực thi, tất cả các đối tượng con của db nếu đang mở sẽ được
đóng lại để giải phóng bộ nhớ. Bản thân db cũng được giải phóng bộ nhớ (bằng Nothing), tất nhiên tệp CSDL và dữ liệu vẫn còn nguyên trên đĩa.

2.3 Đối tượng RecordSet

2.3.1 Khai báo

Recordset là đối tượng dùng để miêu tả tập hợp các bản ghi của một bảng, của
một query hoặc tập các bản ghi kết quả của việc thi hành một câu lệnh SQL nào đó.

Lớp các đối tượng con của Recordset được thể hiện qua sơ đồ sau:



Khai báo:

```
Set rs=db.OpenRecordset(<Name>)
```

Trong đó:

- **Set rs = db.OpenRecordset** là lệnh để tạo ra tập hợp các bản ghi từ CSDL db gán vào biến kiểu recordset *rs*;

- <*Name*> là một xâu ký tự chỉ ra nguồn dữ liệu sẽ trả về cho Recordset. Xâu này có thể là tên một bảng, một Query hoặc một câu lệnh SQL.

Mỗi biến Recordset khi làm việc, phải được chỉ ra Database xuất xứ của nó (phải được tham chiếu từ một biến kiểu Database đã được khai báo).

Ví dụ: Gán tập hợp các bản ghi từ bảng canbo vào biến Recordset

```
Dim rs As DAO.Recordset
Set rs = db.OpenRecordset("canbo")
```

Ví dụ: Gán tập hợp các bản ghi từ một câu lệnh chọn dữ liệu SQL vào biến Recordset (ở đây là các thông tin *hoten, ngaysinh* của tất cả các cán bộ *nữ* từ bảng *canbo)*.

```
Dim rs As DAO.Recordset
Set rs = db.OpenRecordset("SELECT hoten, ngaysinh FROM
canbo
WHERE gioitinh = False")
```

2.3.2 Một số thuộc tính của Recordset

a, Thuộc tính Name

Trả về xâu ký tự trong tham số <*name*> của lệnh gọi Recordset.

Ví dụ: `MsgBox rs.Name`

b, Thuộc tính AbsolutePosition

Cho biết vị trí bản ghi hiện tại (được tính từ 0). Trong trường hợp không có bản ghi nào trên recordset hoặc con trỏ bản ghi đang nằm ở EOF- sẽ không thể lấy được giá trị thuộc tính này. Do vậy để sử dụng thuộc tính này thường phải đi kèm thuộc tính kiểm tra có tồn tại bản ghi nào hay không (RecordCount > 0) và con trỏ bản ghi có ở cuối tệp chưa (EOF = False).

c, Thuộc tính RecordCount

Cho biết tổng số bản ghi trả về trên Recordset.

d, Thuộc tính EOF

Cho biết con trỏ bản ghi hiện tại có nằm ở EOF hay không? Nếu có giá trị thuộc

tính này là True, trái lại là False.

e, Thuộc tính Fields

Dùng tham chiếu tới các trường (Fields) trên tập hợp các bản ghi mà Recordset trả về. Thực tế Field cũng là một đối tượng và cũng có bộ thuộc tính và các phương thức của nó. Với Field của Recordset thông thường người ta hay sử dụng thuộc tính *Value*. Nếu không chỉ định thuộc tính cụ thể nào cho Field, VBA vẫn hiểu ngầm định đó là Value.

Ví dụ: Hiển thị giá trị trường *hoten* trong Recordset *rs*

```
Msgbox rs.Fields("hoten").Value
```

hoặc

```
Msgbox rs.Fields("hoten")
```

2.3.3 Một số phương thức của Recordset

a, Phương thức Close

Để đóng Recordset, giải phóng bộ nhớ. Chỉ thực hiện hành động này khi không làm việc với Recordset nào đó.

b, Phương thức di chuyển bản ghi của Recorset

- Phương thức MoveFirts: Dịch chuyển con trỏ về bản ghi đầu tiên
- Phương thức MoveLast: Dịch chuyển con trỏ về bản ghi cuối cùng
- Phương thức MoveNext: Dịch đến bản ghi kề sau
- Phương thức MovePrevious: Dịch đến bản ghi kề trước

Ví dụ: duyệt và hiển thị toàn bộ *Hoten* của bảng *canbo*

```
Dim db As DAO.Database
Dim rs As DAO.Recordset
Set db = CurrentDb
Set rs = db.OpenRecordset("canbo")
If rs.RecordCount > 0 Then
    rs.MoveFirst
    While rs.EOF = False
        MsgBox rs.Fields("hoten").Value
        rs.MoveNext
    Wend
End If
```

c, Phương thức AddNew, Update

Để thêm mới một bản ghi vào Recordset. Qui trình thêm một bản ghi mới như sau:

1. Ra lệnh Addnew
2. Gán giá trị cho các trường của bản ghi mới
3. Ra lệnh Update

.

Ví dụ: thêm mới một hồ sơ cán bộ mới vào bảng *canbo*

```
Dim db As DAO.Database
Dim rs As DAO.Recordset
Set db = CurrentDb
Set rs = db.OpenRecordset("canbo")
```

```
rs.AddNew
rs.Fields("canboID") = "CB0001"
rs.Fields("hoten") = "Lang Văn Tiền"
rs.Fields("ngaysinh") = #10/08/1989#
rs.Fields("gioitinh") = True
rs.Fields("chucvuID") = "CV001"
rs.Update
```

d, Phương thức Edit, Update

Phương thức Edit để sửa dữ liệu một bản ghi nào đó trên recordset. Qui trình để sửa một bản ghi như sau:

1. Định vị tới bản ghi cần sử trên recordset
2. Ra lệnh Edit
3. Gán giá trị mới cho các trường cần sửa
4. Ra lệnh Update

Ví dụ: sửa hồ sơ cán bộ có mã *CB0001*

```
Dim rs As DAO.Recordset
Set db = CurrentDb
Set rs = db.OpenRecordset("SELECT * FROM canbo WHERE
canboID='CB000565'")
If rs.RecordCount > 0 Then
     rs.MoveFirst
     rs.Edit
     rs.Fields("hoten") = "Nguyễn Văn Tiền"
     rs.Fields("ngaysinh") = #22/11/1990#
     rs.Update
End If
```

e, Phương thức Delete

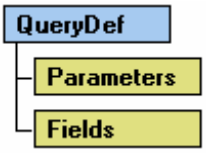
Để xoá bản ghi hiện tại ra khỏi Recordset. Khi đó bản ghi hiện tại sẽ bị xoá bỏ khỏi CSDL. Cần thận trọng mỗi khi ra lệnh này. Thông thường các lệnh một nút Xoá bản ghi của một mẫu nhập liệu (nhập vào biến Recordset *rs*) như sau:

```
Private Sub cmDelete_Click()
     Dim tbao
     tbao = MsgBox("Đã chắc chắn xoá chưa?", vbYesNo
     + vbCritical)
     If tbao = vbYes Then
          rs.Delete
          rs.MoveNext
```

```
        End If
    End Sub
```

2.4 Đối tượng QueryDef

Đối tượng Querydef dùng để tham chiếu tới các Query có sẵn (Buil-in) trên CSDL Access, hoặc cũng có thể lập trình tạo các Query từ các câu lệnh SQL.



Để tạo và kích hoạt một query trên VBA bằng cách thực thi câu lệnh SQL chúng ta thực hiện như sau:

```
'Khai báo một biến kiểu Database và một biến kiểu QueryDef
    Dim db As DAO.Database
    Dim qr As DAO.QueryDef
'Ra lệnh tạo một Query mới, có tên rỗng (chỉ ở trong bộ nhớ)
    Set qr = db.CreateQueryDef(<tên query>)
'Gán chuỗi lệnh SQL vào thuộc tính SQL của query
    qr.SQL = "lệnh SQL cần thi hành"
'Ra lệnh thi hành query
    qr.Execute
'giải phóng bộ nhớ
    qr.Close
```

Trong đó:

- Bắt buộc phải khai báo một biến kiểu QueryDef để làm việc (biến *qr*);
- Phải có một biến Database đã được khai báo sẵn (biến *db*);
- Lệnh **`Set qr = db.CreatQueryDef(<tên query>)`** để tạo một query mới lên CSDL. *<tên query>* sẽ được hiển thị trên danh sách trong thẻ Queries trên cửa sổ Database. Nếu **`<tên query>=""`**, query này sẽ chỉ tồn tại trong bộ nhớ.

Tuỳ thuộc vào mục đích công việc mà có đặt tên query hay không, nếu chỉ đơn thuần tạo một query để xử lý công việ nào đó rồi giải phóng, nên đặt **`<tên query>=""`**;

- Lệnh **`qr.SQL=<câu lệnh SQL>`** để gán lệnh SQL cần thực thi vào Query. Tuỳ thuộc vào câu lệnh SQL này mà query sẽ thực hiện những gì.

Ví dụ: xoá tất cả các bản ghi trên bảng cán bộ:

```
qr.SQL = "DELETE * FROM canbo"
```

- Lệnh **`qr.Excute`** để thi hành câu lệnh SQL đã được thiết lập. Lệnh này tương đương nhấn nút Run đối với một query trên chế độ thiết kế;

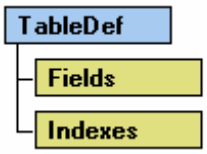
- Lệnh **`qr.Close`** để đóng query hiện tại và giải phóng bộ nhớ khi không cần làm việc đến nữa.

Ví dụ: Tạo DELETE query để xoá danh sách những cán bộ có tuổi lớn hơn 60 ra khỏi bảng *canbo* (cán bộ đã nghỉ hưu)

```
Dim db As DAO.Database
Dim qr As DAO.QueryDef
Set db = CurrentDb
Set qr = db.CreateQueryDef("")
qr.SQL="DELETE * FROM canbo WHERE Year(Date())- "
& " Year(Ngaysinh)>=60"
qr.Execute
qr.Close
```

2.5 Đối tượng TableDef

Đối tượng TableDef được dùng để tham chiếu tới các bảng dữ liệu (Table) trên CSDL. Thông qua đối tượng này có thể thiết kế, chỉnh sửa được cấu trúc các bảng dữ liệu trong chế độ Run-time của VBA như trên chế độ thiết kế bảng Design View.



2.5.1 Một số thuộc tính quan trọng của TableDef:

- *Thuộc tính Name:* Cho biết tên bảng được gán vào biến kiểu TableDef

- *Thuộc tính RecordCount:* Cho biết tổng số bản ghi hiện có trên bảng được gán bởi biến TableDef

- *Thuộc tính DateCreated:* Cho biết thời gian tạo ra bảng được gán vào biến kiểu TableDef

- *Thuộc tính Fields:* Để tham chiếu tới các trường của bảng. Đây là thuộc tính hay được sử dụng nhất đối với TableDef. Thực chất, Field ở đây là một đối tượng, do đó cũng có tập các thuộc tính và phương thức riêng cho thuộc tính này.

Ví dụ: thủ tục hiển thị tên của tất các các trường trong một bảng (ngầm định trên một CSDL đã được khai báo và gán biến *db* - kiểu Database).

```
Sub LietKeTenTruong(tenbang As String)
Dim tbl As DAO.TableDef
```

```
Set tbl = db.TableDefs(tenbang)
    For i = 0 To tbl.Fields.Count - 1
        MsgBox tbl.Fields(i).Name
    Next
End Sub
```

2.5.2 Một số phương thức của TableDef

a, Phương thức CreateTableDef

Để tạo ra một bảng mới bằng VBA. Cú pháp tạo bảng mới như sau:

```
Set tbl = db.CreateTableDef(<Tên bảng mới>)
    '----------------------------------------
    '....Các thủ tục tạo trường mới cho bảng
    '----------------------------------------
db.TableDefs.Append tbl
```

Trong đó:

- *db* – là biến kiểu Database đã được gán bởi CSDL cần làm việc (bảng mới sẽ được tạo ra trên CSDL này);

- <*Tên bảng mới*> là tên bảng cần tạo.

- Lệnh **db.TableDefs.Append tbl** là lệnh ghi cấu trúc bảng đang khai báo lên CSDL đã chỉ định.

b, Phương thức CreateField

Để tạo ra các trường cho một bảng kiểu TableDef nào đó. Để thêm một trường mới lên bảng, sử dụng cú pháp sau:

tbl.Fields.Append tbl.**CreateField***(<tên trường>,<KiểuDL>,<độlớn>)*

Trong đó:

- <*tên trường*> - tên trường mới cần tạo;

- <*KiểuDL*> - là một tuỳ chọn để khai báo kiểu dữ liệu của trường cần tạo. Kiểu dữ liệu được khai báo theo các hằng số như sau:

- *dbBoolean* Boolean
- *dbByte* Byte
- *dbChar* Currency
- *dbDate* Date/Time
- *dbDecimal* Decimal
- *dbDouble* Double
- *dbFloat* Float
- *dbGUID* GUID
- *dbInteger* Integer

- *dbLong* Long
- *dbMemo* Memo
- *dbNumeric* Numeric
- *dbSingle* Single
- *dbText* Text
- *dbTime* Time

- <*Độ lớn*> là một tuỳ chọn để khai báo độ lớn dữ liệu nếu cần.

Ví dụ:

```
Sub TaoBangMoi()
    On Error GoTo Loi
    Dim tbl As DAO.TableDef
    Set tbl = db.CreateTableDef("NewTable")
    tbl.Fields.Append tbl.CreateField("ID", dbInteger)
    tbl.Fields.Append tbl.CreateField("Name", dbText)
    tbl.Fields.Append tbl.CreateField("Age", dbByte)
    tbl.Fields.Append tbl.CreateField("DateBirth", dbDate)
    tbl.Fields.Append tbl.CreateField("Comment", dbMemo)
    db.TableDefs.Append tbl
    Exit Sub
    Loi:
    If Err.Number = 3010 Then
        MsgBox "Đã tồn tại bảng có tên " + tbl.Name
    End If
End Sub
```

1.6 Đối tượng Relation

Đối tượng Relation dùng để tạo kết nối (RelationShip) giữa 2 bảng trong CSDL Access.

Ví dụ: tạo kết nối giữa 2 bảng hoadon và khach trong CSDL Quản lý bán hàng.

```
Sub CreatRelationShip()
    On Error GoTo Loi
    Dim db As DAO.Database
    Dim rls As DAO.Relation
    Set db = CurrentDb
    Set rls = db.CreateRelation("TaoQuanHe", "khach", "hoadon",
    dbRelationUpdateCascade)
```

```
        rls.Fields.Append rls.CreateField("khachID")
        rls.Fields("khachID").ForeignName = "khachID"
        db.Relations.Append rls
        Loi:
        If Err.Number = 3012 Then
            MsgBox "Đã tồn tại quan hệ này !"
        End If
    End Sub
```

Trong trường hợp đã tồn tại kết nối này, một thông báo lỗi tiếng Việt "*Đã tồn tại quan hệ này !*" xuất hiện.

3. Đặt lọc dữ liệu

Đặt lọc là lớp bài toán phổ dụng trong thực tế. Với bài toán này phải có những yêu cầu cụ thể về lọc dữ liệu (điều kiện lọc). Kết quả trả về sẽ là một tập hợp các bản ghi, có thể được kết xuất trên form hoặc được in ra máy in dưới dạng report.

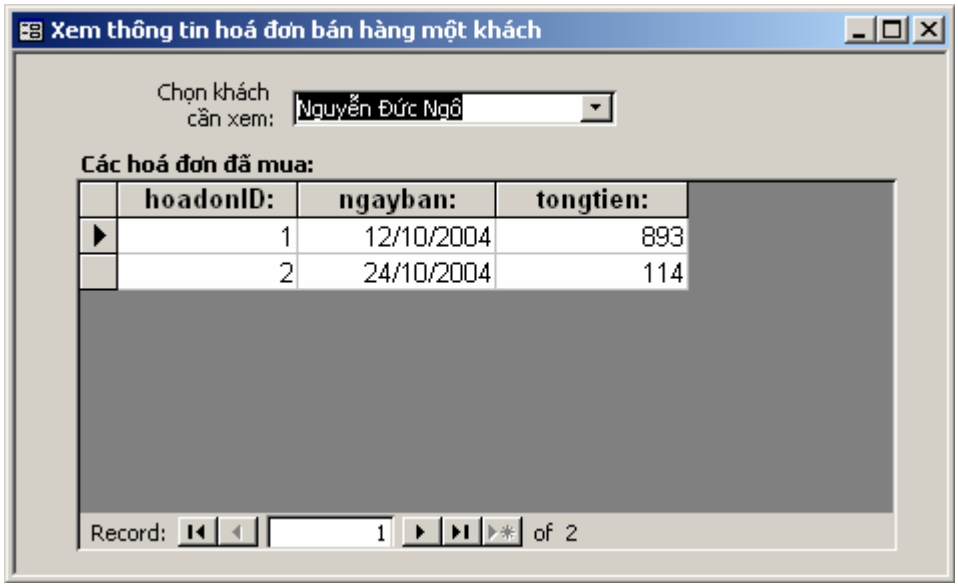
3.1 Quy trình thực hiện

Bước 1: Xây dựng Subform - form sẽ chứa những kết quả lọc được.

Bước 2: Xây dựng Mainform - form chứa những thiết lập điều kiện để lọc.

Bước 3: Thực hiện lọc ra các bản ghi thoả mãn các điều kiện trên Mainform và hiển thị kết quả lên Subform.

Ví dụ: Thực hiện form lọc dữ liệu sau:

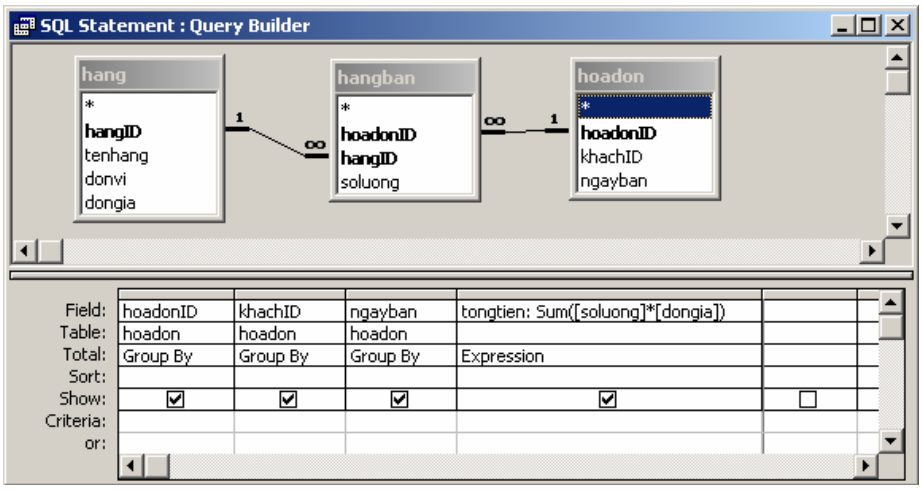


Bước 1: Xây dựng Form con

Sử dụng các kỹ năng thông thường để tạo một form con đáp ứng được các kết quả theo như bài toán. Cụ thể từng bước như sau:

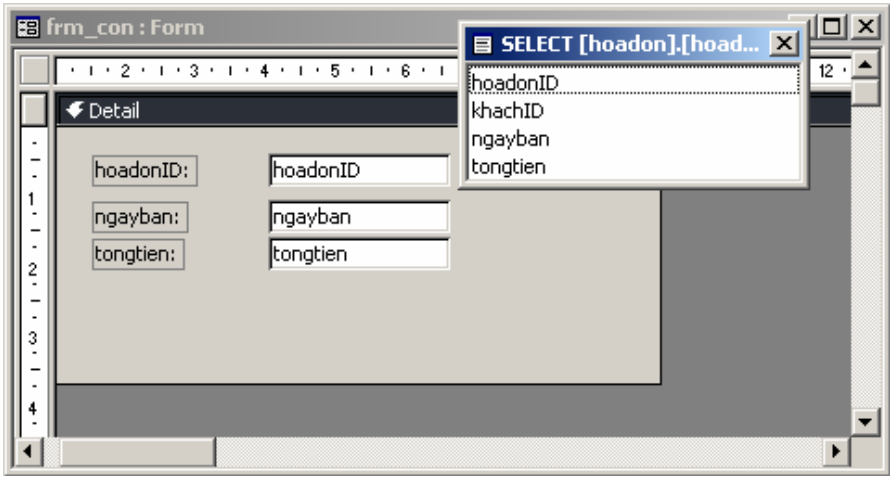
- Tạo mới form ở chế độ Design view;

- Thiết lập thuộc tính Record Source cho form là một Total Query như sau:



- Thiết lập thuộc tính Defaul View cho form con là *Datasheets*;

- Mở cửa sổ FieldList và thiết kế form như sau:



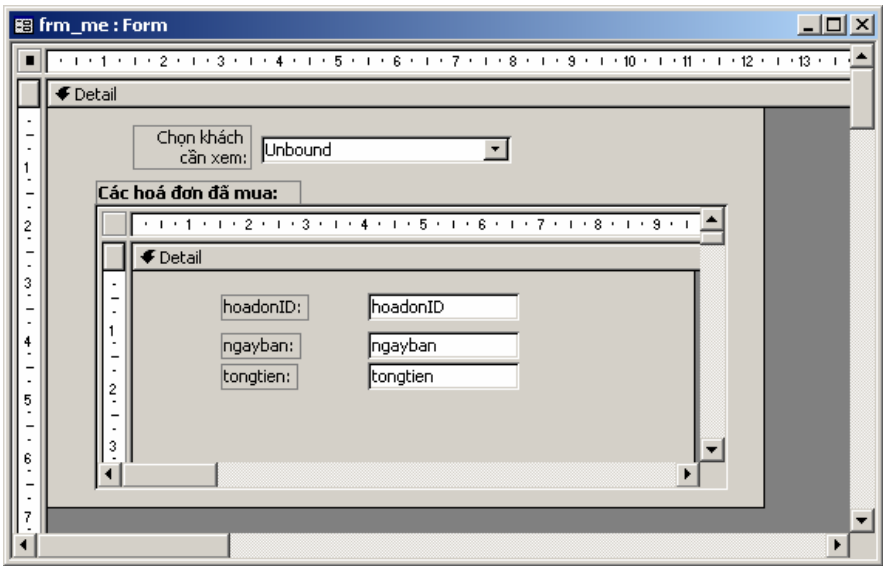
- Đóng thiết kế form con lại và ghi ten là *frm_formcon*.

Bước 2: Xây dựng form mẹ

- Tạo mới form ở chế độ Design view;

- Đưa hộp Combobox từ thành công cụ Toolbox lên form mẹ (giả sử tên (Name) của Combo này là *Combo0*). Sử dụng tính năng Combobox Wizard của Access để làm việc này. Mục đích là đưa danh sách các khách hàng từ bảng khach vào hộp Combo để chọn, phục vụ nhu cầu lọc dữ liệu.

- Sử dụng đối tượng Sub-form/Sub-report trên thanh công cụ Toolbox để đưa form con vừa tạo lên form mẹ. Ngầm định tên của subform này là frm_formcon.



Bước 3: Thiết lập lệnh lọc dữ liệu trên form mẹ

Công việc của bước này là làm sao để sau khi chọn tên một khách hàng ở hộp Combobox, danh sách các hoá đơn mua hàng của khách đó sẽ được hiển thị lên form con. Muốn thế, việc lập trình lọc dữ liệu ở đây phải được thực hiện trong thủ tục đáp ứng sự kiện **Combo0_Click**.

Giải thuật đáp ứng sự kiện Combo0_Click:

- Tạo một biến Recordset để thi hành câu lệnh SQL đưa ra danh sách kết quả thoả mãn điều kiện đặt lọc:

```
"SELECT hoadonID, khachID, ngayban, Sum([soluong]*[dongia])"
+ " AS tongtien FROM "
+ " hoadon INNER JOIN (hang INNER JOIN hangban ON "
+ " hang.hangID = hangban.hangID) ON hoadon.hoadonID ="
+ " hangban.hoadonID WHERE Trim(khachID)='"+Trim(Combo0)"
+ " GROUP BY hoadonID, khachID, ngayban "
```

- Gán thuộc tính Recorset của form con là biến kiểu recordset vừa tạo ra (chứa kết quả đã lọc).

- Ra lệnh làm tươi dữ liệu cho form con.

Chú ý: trước đó phải khai báo một biến kiểu Database toàn cục trong form và định nghĩa nó ở thủ tục Form_Load()

Code:

```
Dim db As DAO.Database
Private Sub Form_Load()
    Set db = CurrentDb
End Sub
Private Sub Combo0_Click()
    Dim rs As DAO.Recordset
```

```
    Set rs = db.OpenRecordset("SELECT hoadonID, khachID, "
    + " ngayban, Sum([soluong]*[dongia]) AS tongtien FROM"
    + " hoadon INNER JOIN (hang INNER JOIN hangban ON "
    + " hang.hangID = hangban.hangID) ON hoadon.hoadonID ="
    + " hangban.hoadonID WHERE Trim(khachID)='"+Trim(Combo0)"
    + " GROUP BY hoadonID, khachID, ngayban ")
    Set frm_formcon.Form.Recordset = rs
    frm_formcon.Requery
End Sub
```