

MACRO



Trần Quang
quangt@cse.hcmut.edu.vn

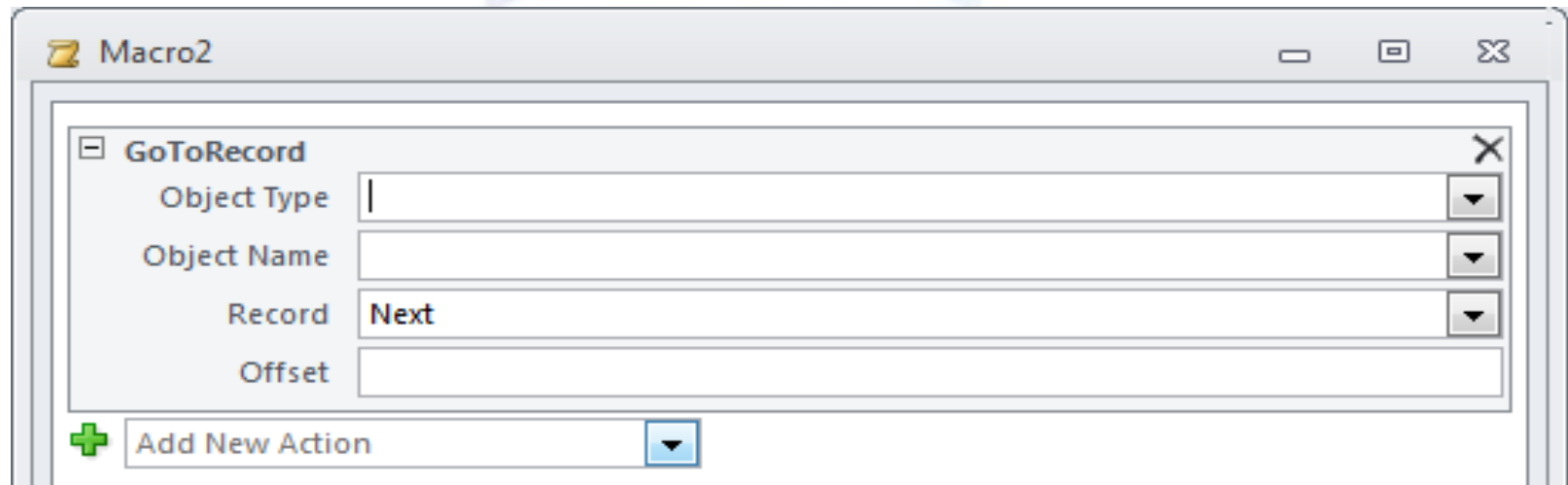
NỘI DUNG

- Giới thiệu về macro
- Cách tạo macro
- Gắn macro vào các sự kiện trên form



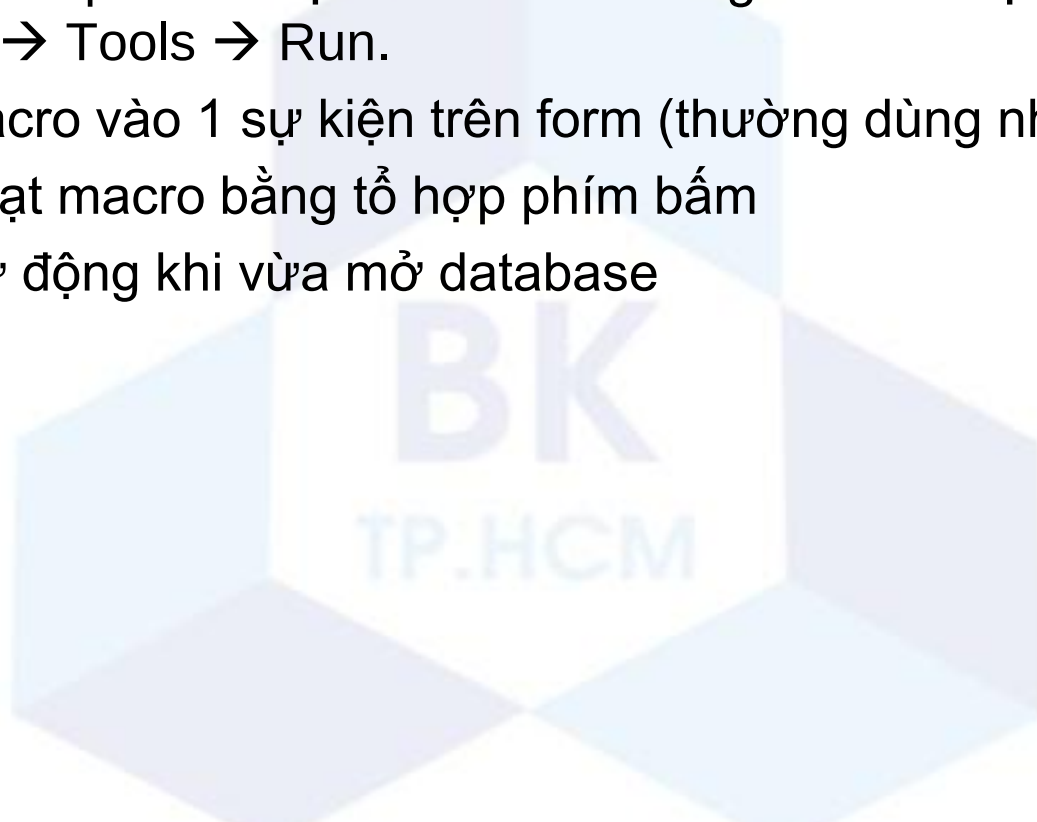
Giới thiệu

- Macro là một danh sách các hành động (*action*) sẽ được thực hiện khi chúng ta gọi macro.
- Tạo Macro:
 - Chọn Create → Macros & Code → Macro.
 - Chọn hành động từ Add New Action.
 - Điền các tham số cho hành động vừa chọn.



Giới thiệu

- Thực hiện Macro: có 4 cách.
 - Chạy trực tiếp bằng cách double-click vào tên của Macro trên thanh Navigation pane. Hoặc nếu macro đang mở thì chọn Macro Tools | Design → Tools → Run.
 - Gắn macro vào 1 sự kiện trên form (thường dùng nhất)
 - Kích hoạt macro bằng tổ hợp phím bấm
 - Chạy tự động khi vừa mở database



Hiệu chỉnh Macro

- Mở macro ở chế độ Design view
 - Thay đổi các tham số của hành động
 - Thay đổi thứ tự thực hiện của các hành động
 - Thêm 1 hành động mới.
 - Xóa bớt 1 hành động
 - Thêm các chú thích
- Mở rộng và thu gọn khung thiết kế hành động bằng cách click chuột vào nút +/- nằm bên trái tên hành động

Hiệu chỉnh Macro

Chon

☐ **If** [chon]=1 **Then**

☐ **OpenTable**

Table Name

View

Data Mode

 Add New Action

[Add Else](#) [Add Else If](#)

End If

☐ **If** [chon]=2 **Then**

OpenTable

Table Name

View

Data Mode

End If

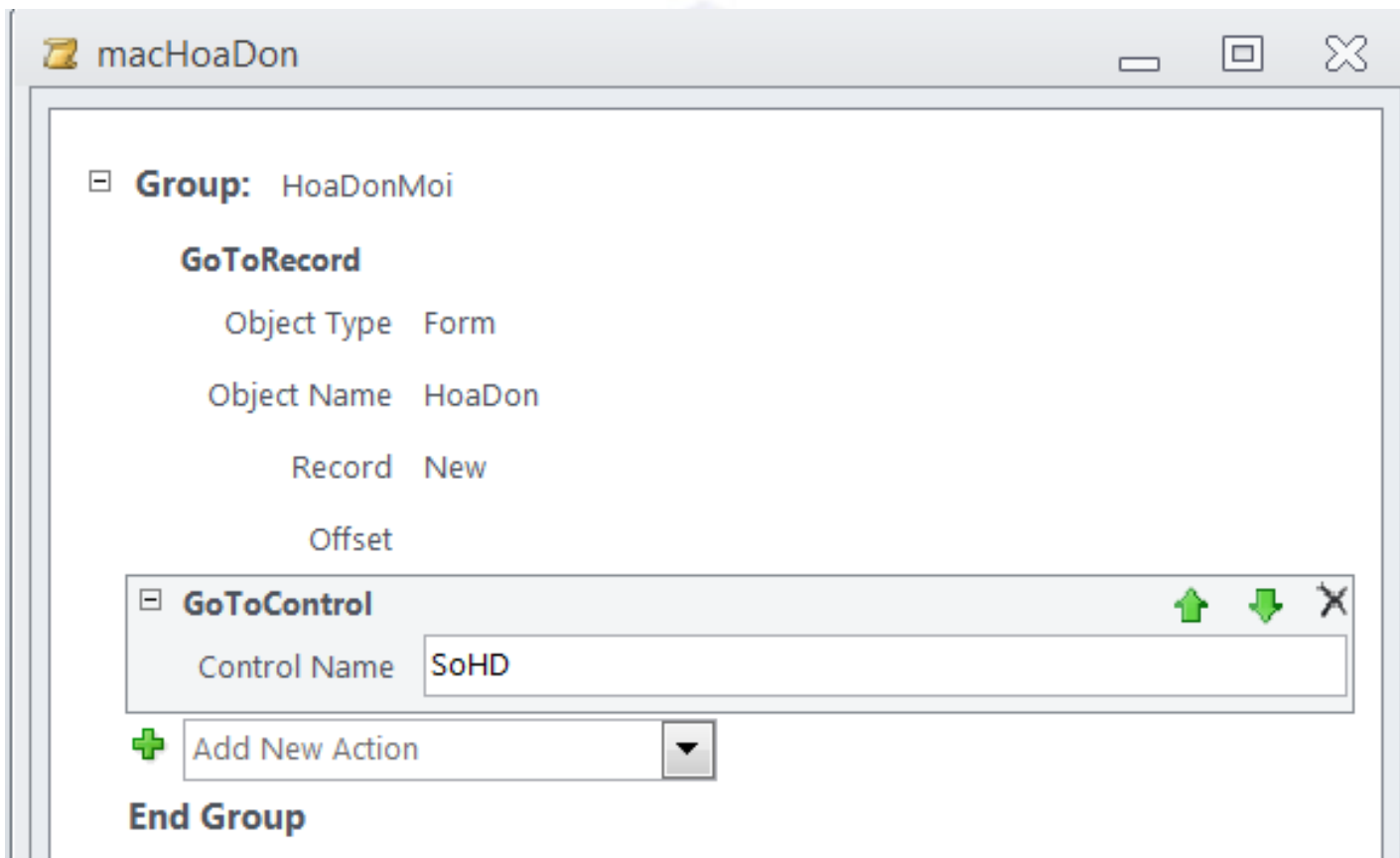
 Add New Action

Macro nhóm

- Macro nhóm (grouping macro): là việc chia 1 macro ra thành nhiều phần nhỏ, mỗi phần bao gồm 1 đến nhiều hành động.
- Để tạo 1 nhóm:
 - Chọn Group trong danh sách Add New Action.
 - Gõ vào tên nhóm.
 - Thêm các hành động vào nhóm.
 - Mỗi nhóm bắt đầu bằng chuỗi “Group” và tên nhóm, kết thúc nhóm bằng chuỗi “End Group”. Khoảng giữa là các hành động của nhóm.

Macro nhóm

■ Ví dụ



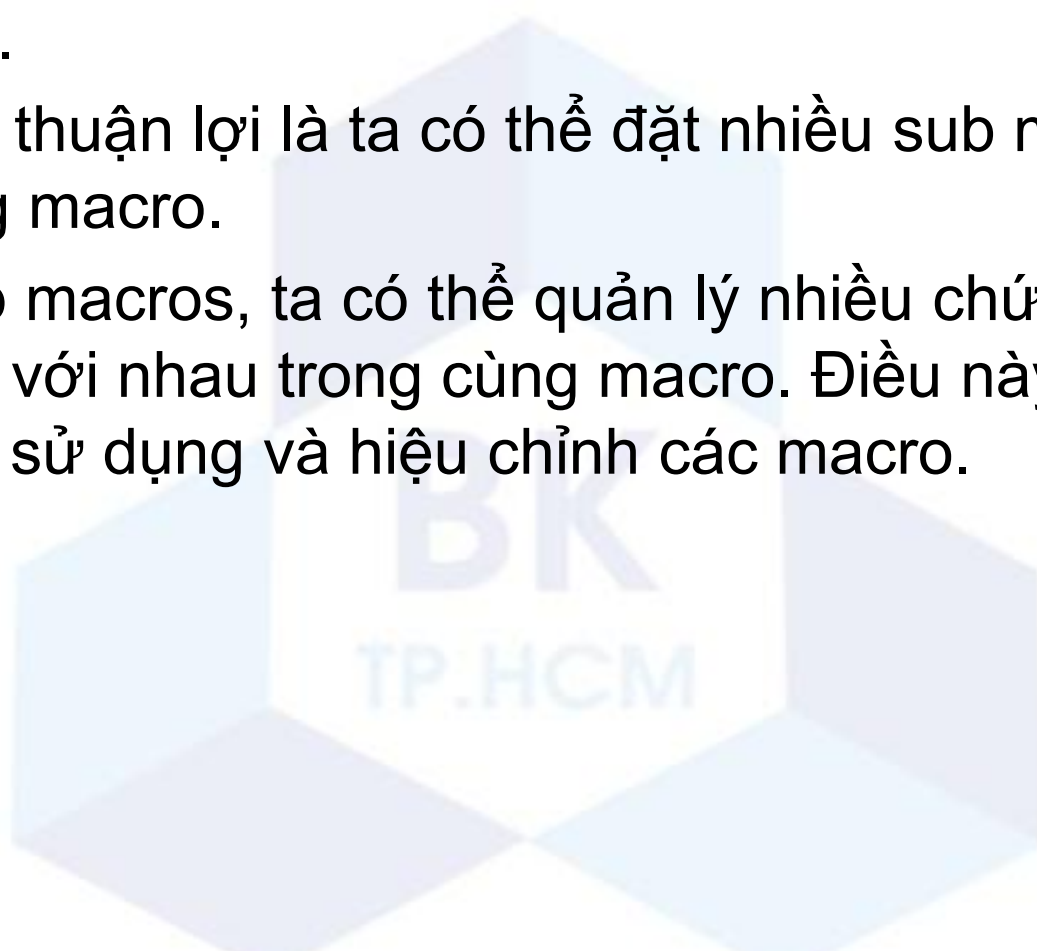
Thực hiện từng bước macro

- Thực hiện các bước sau để kiểm tra từng bước việc thực hiện của macro:
 - Mở macro ở chế độ Design view.
 - Chọn Macro Tools | Design → Tools → Single Step.
 - Chọn Macro Tools | Design → Tools → Run
 - Chọn Step, Continue, hoặc Stop All Macros, tùy theo yêu cầu của chúng ta cho bước kế.

BK
TP.HCM

Sub Macros

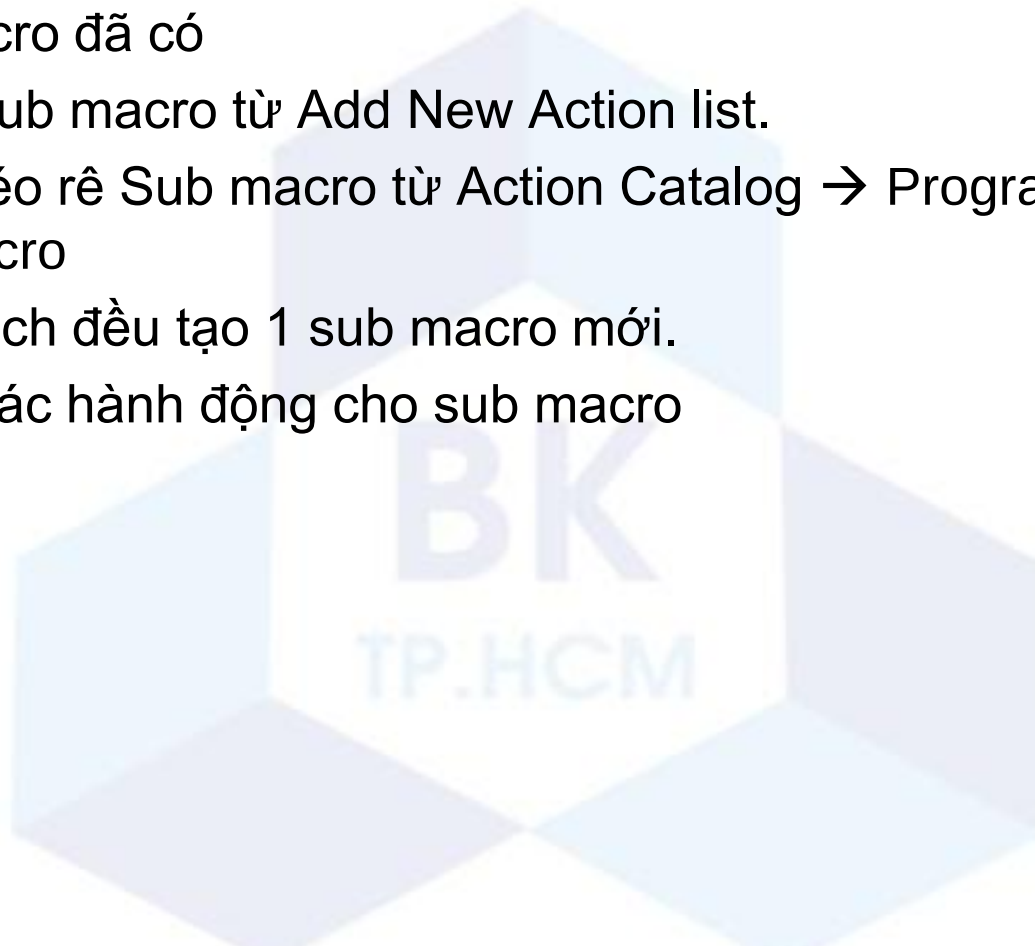
- Sub macro là 1 nhóm các hành động được gom lại và đặt cho 1 tên.
- Một điểm thuận lợi là ta có thể đặt nhiều sub macro vào 1 đối tượng macro.
- Dùng sub macros, ta có thể quản lý nhiều chức năng có liên quan với nhau trong cùng macro. Điều này giúp ta dễ tìm kiếm, sử dụng và hiệu chỉnh các macro.



Sub Macros

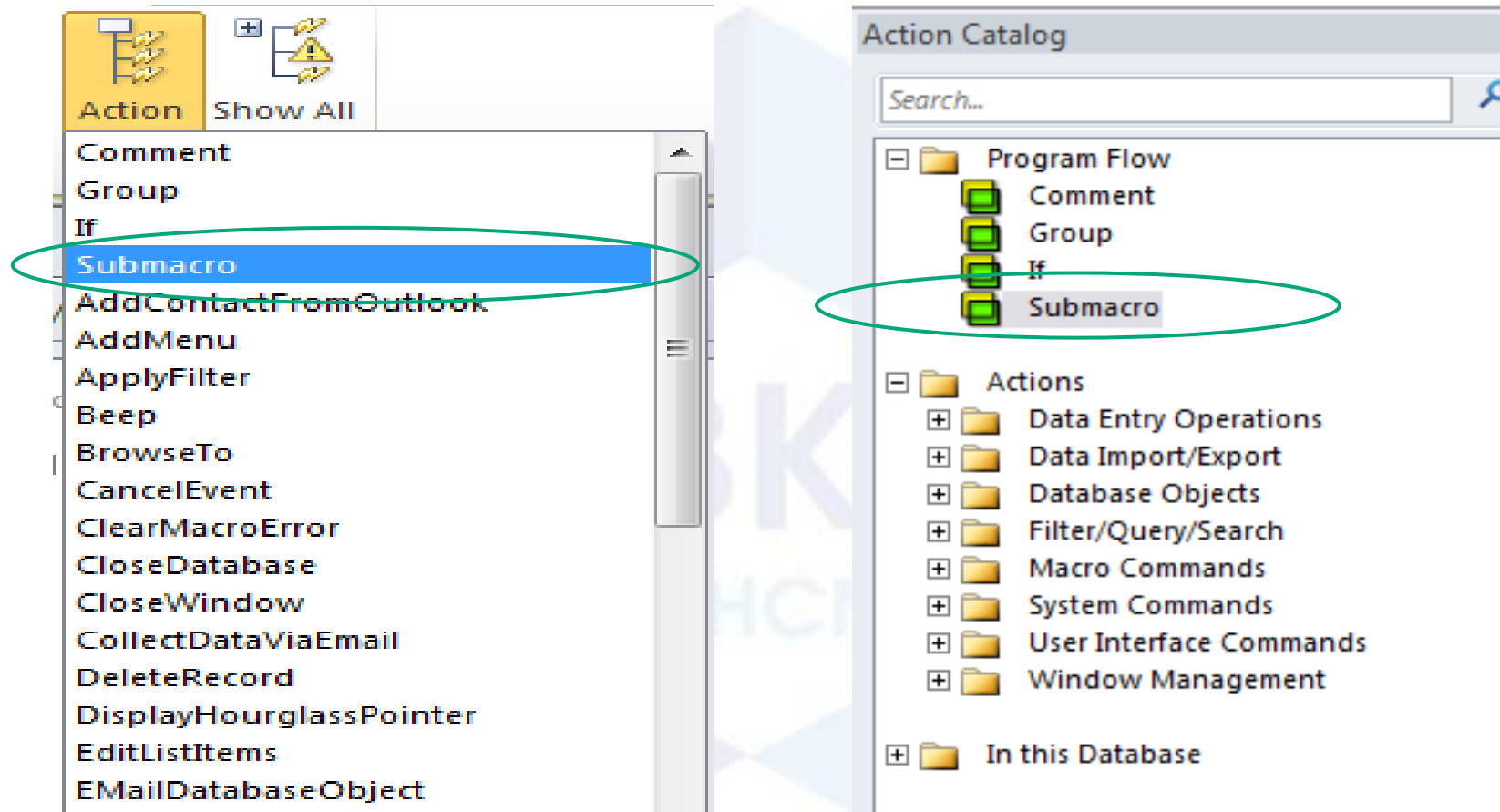
■ Để tạo 1 sub macro:

- Mở macro đã có
- Chọn Sub macro từ Add New Action list.
- Hoặc kéo rê Sub macro từ Action Catalog → Program Flow và thả vào macro
- Cả 2 cách đều tạo 1 sub macro mới.
- Thêm các hành động cho sub macro



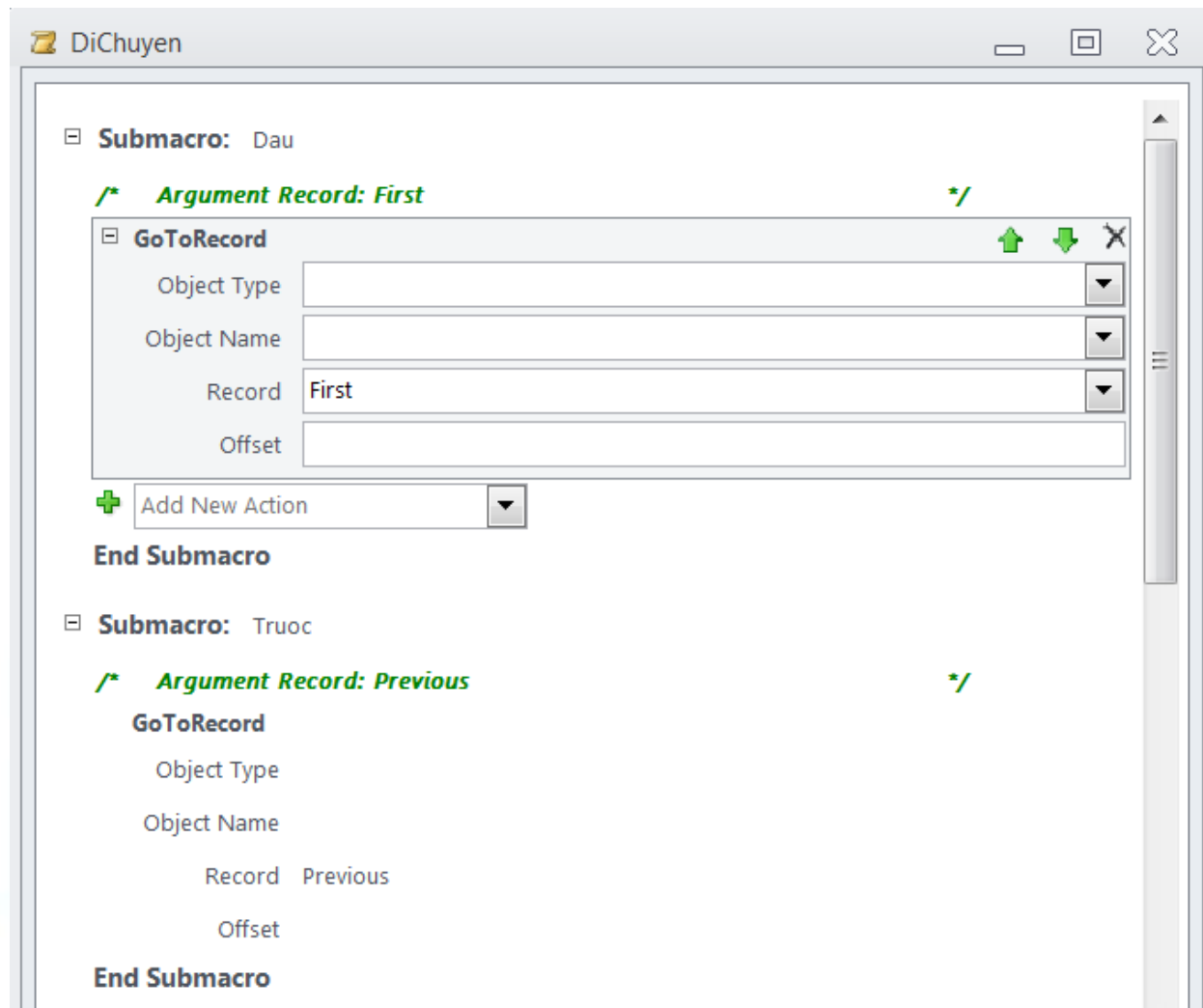
Sub Macros

- Hai cách để thêm 1 Sub macro:



Sub Macros

■ Ví dụ:



Cấu hình Startup Macro

- Macro AutoExec:
 - Là macro tự động chạy ngay khi ta vừa mở database
 - Macro thường bao gồm các hành động sau:
 - Mở 1 vài form hoặc report quan trọng (ví dụ: mở màn hình chính của ứng dụng).
 - Imports dữ liệu từ các tập tin khác.
 - Chạy các câu query để làm sạch hệ thống (ví dụ xóa các dữ liệu trong các table tạm, ...).
 - Tên của macro là AutoExec

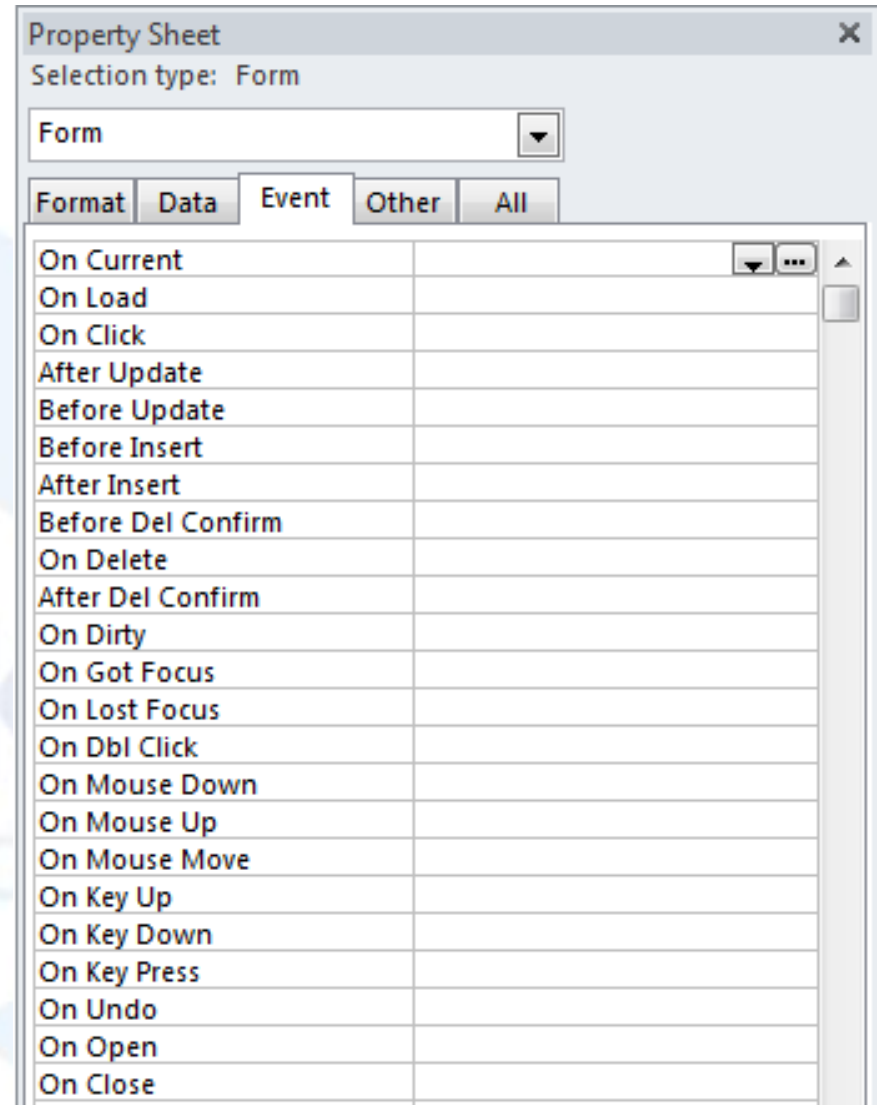
Kết nối macro với form

Trên form có 3 loại sự kiện:

- Control events: đây là các sự kiện thông dụng nhất. Sự kiện này xuất hiện khi ta thực hiện 1 hành động nào đó trên control (ví dụ: button, textbox, ...).
- Section events: form được chia thành nhiều phần (*sections*), mỗi section có các sự kiện riêng trên đó.
- Form events: bao gồm các sự kiện xảy ra khi form mở ra, đóng lại, khi di chuyển từ mẫu tin này sang mẫu tin khác hoặc khi cập nhật dữ liệu.

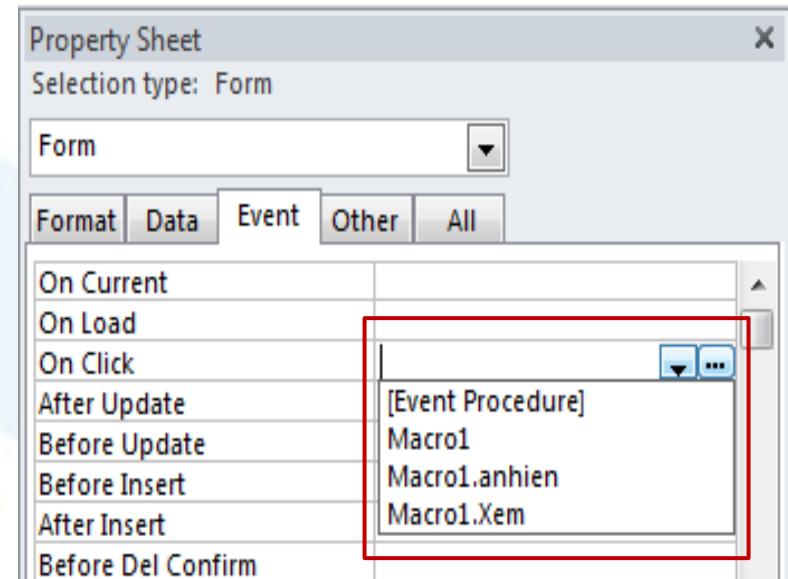
Kết nối macro với form

- Để xem danh sách các sự kiện xảy ra trên form:
 - Mở form ở chế độ Design mode.
 - Chọn thành phần có sự kiện mà ta cần kiểm tra.
 - Ở cửa sổ property chọn tab Event.
 - Chọn sự kiện phù hợp.



Gắn Macro vào sự kiện

- Tạo và lưu macro.
- Mở form ở chế độ Design view
- Chọn control, section, hoặc form cần gắn macro.
- Mở Property Sheet, chọn tab Events → chọn sự kiện phù hợp
- Click hộp drop-down và chọn macro phù hợp.



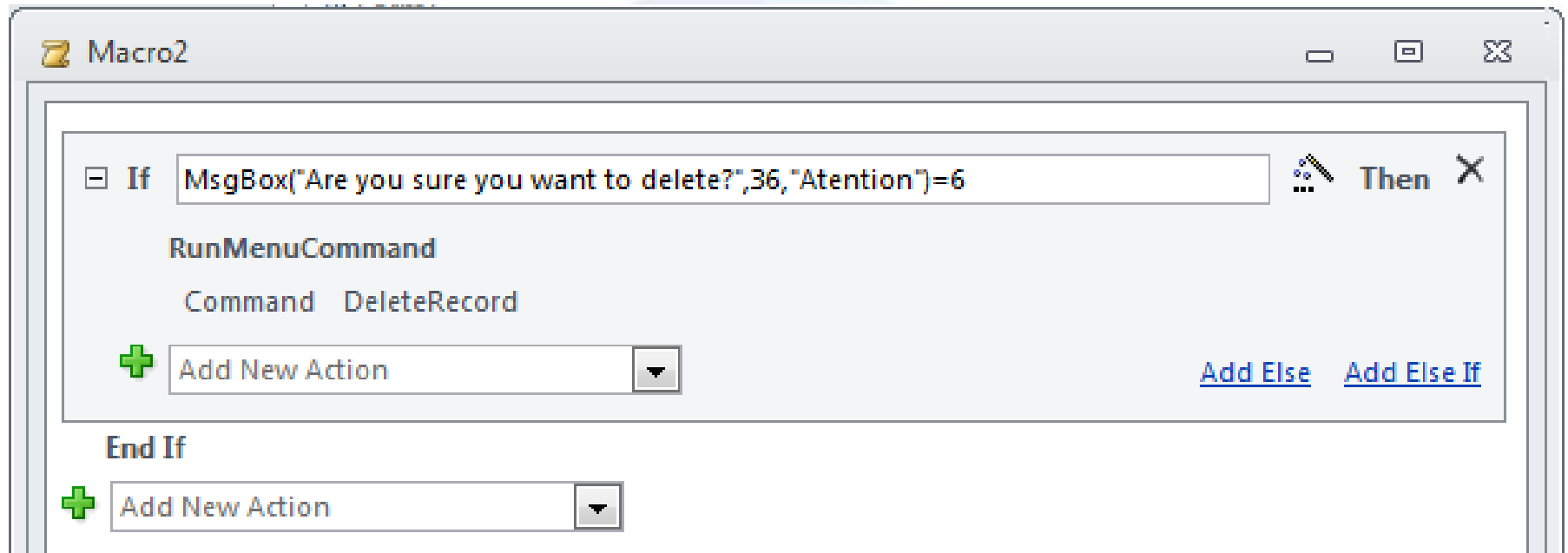
Macro điều kiện (Conditional Macros)

Các hành động được thực hiện tùy thuộc vào điều kiện

- Để tạo macro điều kiện, ta dùng hành động IF:
 - Chọn hành động If từ Add New Action
 - Kéo thả hành động If từ Program Flow của Action Catalog.
- Hành động If cần một tham số là 1 biểu thức điều kiện.
- Nếu kiểm tra điều kiện này đúng (True), Access sẽ thực hiện các hành động bên dưới
- Nếu kiểm tra điều kiện này sai (False), Access sẽ bỏ qua các hành động bên dưới

Macro điều kiện (Conditional Macros)

- Ví dụ:



Macro điều kiện (Conditional Macros)

Trong biểu thức điều kiện có thể sử dụng:

- Toán tử so sánh: <, > <=, >=, <>, =
- Toán tử luận lý: Not, And, Or.
- Hàm:
 - IsNull()
 - Len()
 - ...



Data Macros

- Data Macros là macro được tự thực hiện khi có một sự kiện xảy ra trên table
- Các sự kiện có thể là thêm 1 mẫu tin, hoặc cập nhật dữ liệu trong table
- Data macros có thể hỗ trợ 1 tập các hành động.
- Data macros có thể gắn với cơ sở dữ liệu, ngay cả khi chuyển sang platform khác.

Data Macros

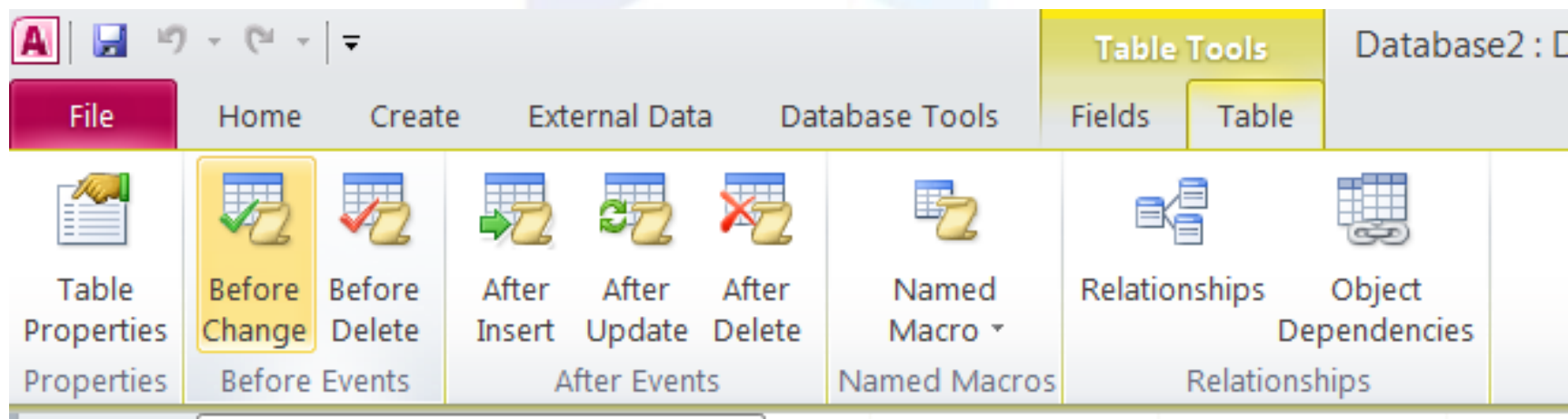
■ Các sự kiện trên table:

Sự kiện	Mô tả
Before Change	Xuất hiện trước khi Access lưu dữ liệu cập nhật hoặc thêm mẫu tin mới thực sự vào table. Ta có thể dùng thuộc tính IsInsert để phân biệt (True cho insert and False cho Update)
Before Delete	Xuất hiện trước khi Access xóa 1 mẫu tin từ table
After Insert	Xuất hiện sau khi hoàn tất việc thêm 1 mẫu tin
After Update	Xuất hiện sau khi hoàn tất việc cập nhật 1 mẫu tin
After Delete	Xuất hiện sau khi hoàn tất việc xóa 1 mẫu tin

Data Macros

Để tạo data macro:

- Mở table ở chế độ Datasheet view.
- Chọn Table Tools → Table → chọn sự kiện kích hoạt.
- Thêm các hành động



Data Macros

- Ví dụ: tạo data macro để cấm việc đưa dữ liệu trống vào field EmailAddress:
 - Mở table ở chế độ Datasheet view.
 - Chọn Table Tools → Table → Before Events → Before Change.
 - Thêm vào các hành động:
 - Hành động IF để kiểm tra quy tắc có bị phá vỡ không.
`IsNull([EmailAddress]).`
 - Hành động RaiseError để hủy thay đổi và đưa ra thông báo lỗi

Danh sách các hành động thường sử dụng

■ Open Form (Mở 1 Form) :

- Form Name : Chọn Form cần mở.
- View : Dạng trình bày Form.

Form : Mở Form ở dạng Form View.

Design : Mở Form ở dạng Design View.

Print Preview : Mở Form ở dạng Print Preview.

Datasheet : Mở Form ở dạng Datasheet View.

- Filter Name : Tên Query sàng lọc dữ liệu hiển thị trong Form.
- Where Condition : Điều kiện sàng lọc dữ liệu hiển thị trong Form.
- Data Mode : Chế độ hiển thị dữ liệu (tương tự như trong Open Table Action).

Danh sách các hành động thường sử dụng

■ Open Report (Mở 1 báo biểu) :

- Report Name : Chọn Report cần in.

- View : Chế độ in.

Print Preview : In ra màn hình.

Print : In ra máy in.

- Filter Name : Tên Query sàng lọc dữ liệu hiển thị trong Report.

- Where Condition : Điều kiện sàng lọc dữ liệu hiển thị trong Report

Danh sách các hành động thường sử dụng

■ Close Window: Đóng một Object

- Object Type : Loại Object muốn đóng. Nếu để trống mặc nhiên là cửa sổ hiện hành.
- Object Name : Tên của Object cần đóng.
- Save :

Prompt : Hiện thị hộp thoại hỏi ý kiến có lưu trữ không khi có sửa đổi đối với mỗi Object khi đóng.

Yes : Lưu trữ tất cả mọi sửa đổi mà không hiện thị hộp thoại hỏi ý kiến.

No : Đóng mà không lưu trữ những sửa đổi trước đó

Danh sách các hành động thường sử dụng

- Set Value : Gán giá trị cho 1 Field, 1 số giá trị hay thuộc tính trên 1 Form hay 1 Report.
 - Item : Tên Field, đối tượng hay thuộc tính muốn gán giá trị.
 - Expression : biểu thức cần giá trị cho Item.
- * Chú ý : Tại 2 tham số trên, khi chỉ tên đối tượng hay thuộc tính của nó ở 1 Form hay Report khác phải dùng cú pháp đầy đủ như sau :
 - Trong Form : [Forms]![Tên Form]![Tên Field của đối tượng]
 - Chỉ định thuộc tính của một đối tượng :
[Forms/Reports]![Tên Form/Report]![Tên đối tượng].[Tên thuộc tính]

Danh sách các hành động thường sử dụng

- Hàm MsgBox(Prompt, Title, Button)



Các giá trị của thông số BUTTON

Constant	Value	Description
vbOkOnly	0	Displays 1 button - OK (default value)
vbOkCancel	1	Displays 2 buttons - OK and Cancel
vbAbortRetryIgnore	2	Displays 3 buttons - Abort, Retry, Ignore
vbYesNoCancel	3	Displays 3 buttons - Yes, No, and Cancel
vbYesNo	4	Displays 2 buttons - Yes and No
vbRetryCancel	5	Displays 2 buttons - Retry and Cancel
vbCritical	16	Displays a critical message
vbQuestion	32	Displays a question

Giá trị trả về của hàm MessageBox

Hằng	Giá trị	Mô tả
vbOK	1	Value signifies that the OK button was pressed
vbCancel	2	Value signifies that the Cancel button was pressed
vbAbort	3	Value signifies that the Abort button was pressed
vbRetry	4	Value signifies that the Retry button was pressed
vbIgnore	5	Value signifies that the Ignore button was pressed
vbYes	6	Value signifies that the Yes button was pressed
vbNo	7	Value signifies that the No button was pressed