Using Macros and SQL in Access
Instructor: Aarthi Natarajan

Based on slides by
Robert Grauer, Keith Mast, Mary Anne Poatsy

Objectives

- Understand the purpose of a macro
- Create a stand-alone macro
- Use the Macro Builder
- Attach an embedded macro to an event
- Identify when to use a data macro

Objectives (continued)

- Create an event-driven data macro
- Create a named data macro
- Understand the fundamentals of SQL
- Interpret an SQL Select Statement
- Use an SQL Select Statement as a record source

Understand the Purpose of a Macro

- **Macro** – a series of actions that can be programmed to automate tasks
- **Event** – occurs when a user enters, edits, or deletes data; events can also occur when users open, use, and close forms and reports

Understand the Purpose of a Macro (continued)

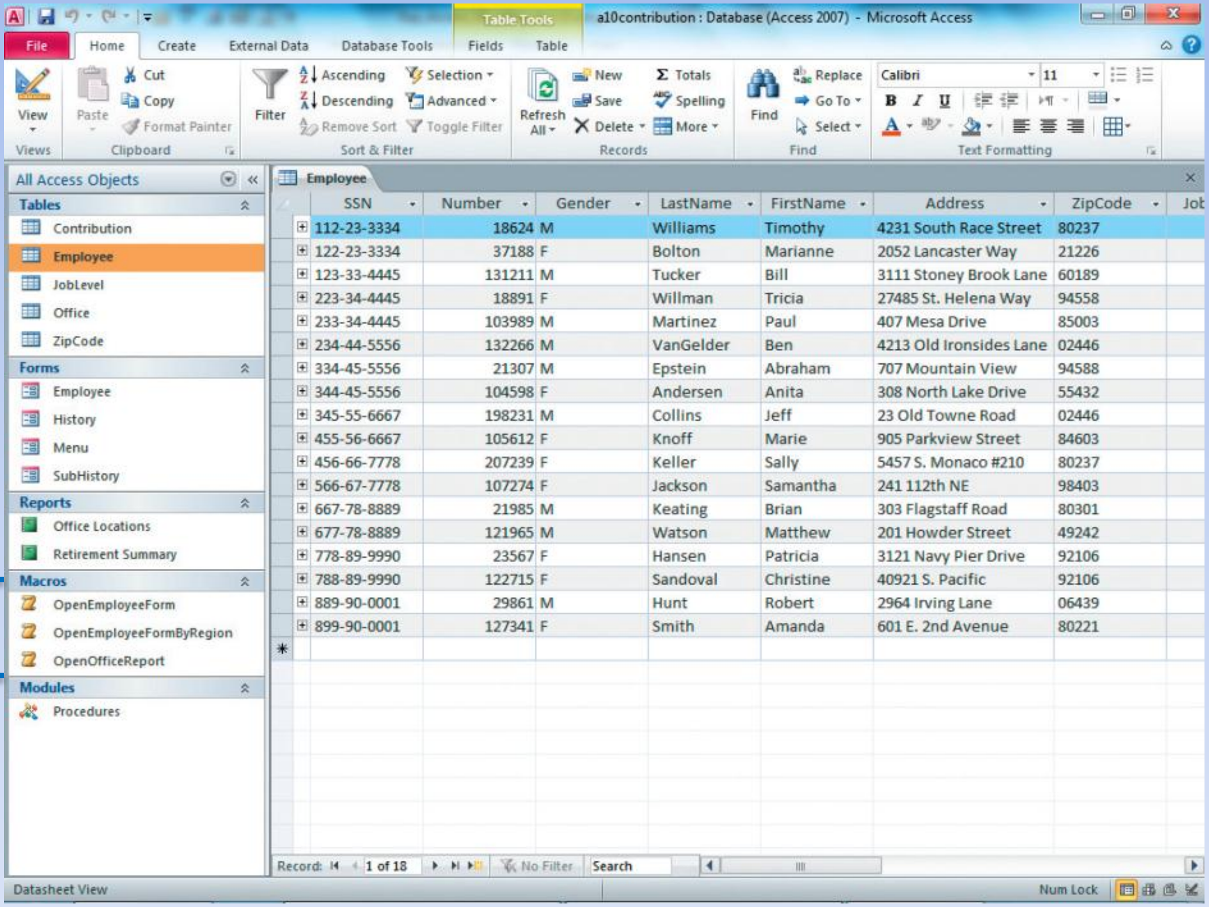
- Two types of macro:
  - **Stand-alone macro** – a database object that you create and can use independently of other controls or objects
  - **Embedded macro** – a macro that is embedded in a particular object/control, and executes when the event occurs/control is clicked

Purpose of a Macro (continued)

- Two examples of an event attached to a control:
  - **After Update event**
    - Triggered each time you enter (update) data into a field on a form
  - **On Close event**
    - Happens whenever you close a form or a report
- When triggered Access will execute the steps stored in the macro.

Understand the Purpose of a Macro (continued)

Figure 10.1 Stand-Alone Macro



SSN	Number	Gender	LastName	FirstName	Address	ZipCode	Job
112-23-3334	18624	M	Williams	Timothy	4231 South Race Street	80237	
122-23-3334	37188	F	Bolton	Marianne	2052 Lancaster Way	21226	
123-33-4445	131211	M	Tucker	Bill	3111 Stoney Brook Lane	60189	
223-34-4445	18891	F	Willman	Tricia	27485 St. Helena Way	94558	
233-34-4445	103989	M	Martinez	Paul	407 Mesa Drive	85003	
234-44-5556	132266	M	VanGelder	Ben	4213 Old Ironsides Lane	02446	
334-45-5556	21307	M	Epstein	Abraham	707 Mountain View	94588	
344-45-5556	104598	F	Andersen	Anita	308 North Lake Drive	55432	
345-55-6667	198231	M	Collins	Jeff	23 Old Towne Road	02446	
455-56-6667	105612	F	Knoff	Marie	905 Parkview Street	84603	
456-66-7778	207239	F	Keller	Sally	5457 S. Monaco #210	80237	
566-67-7778	107274	F	Jackson	Samantha	241 112th NE	98403	
667-78-8889	21985	M	Keating	Brian	303 Flagstaff Road	80301	
677-78-8889	121965	M	Watson	Matthew	201 Howder Street	49242	
778-89-9990	23567	F	Hansen	Patricia	3121 Navy Pier Drive	92106	
788-89-9990	122715	F	Sandoval	Christine	40921 S. Pacific	92106	
889-90-0001	29861	M	Hunt	Robert	2964 Irving Lane	06439	
899-90-0001	127341	F	Smith	Amanda	601 E. 2nd Avenue	80221	

Stand-alone macros appear in Navigation Pane

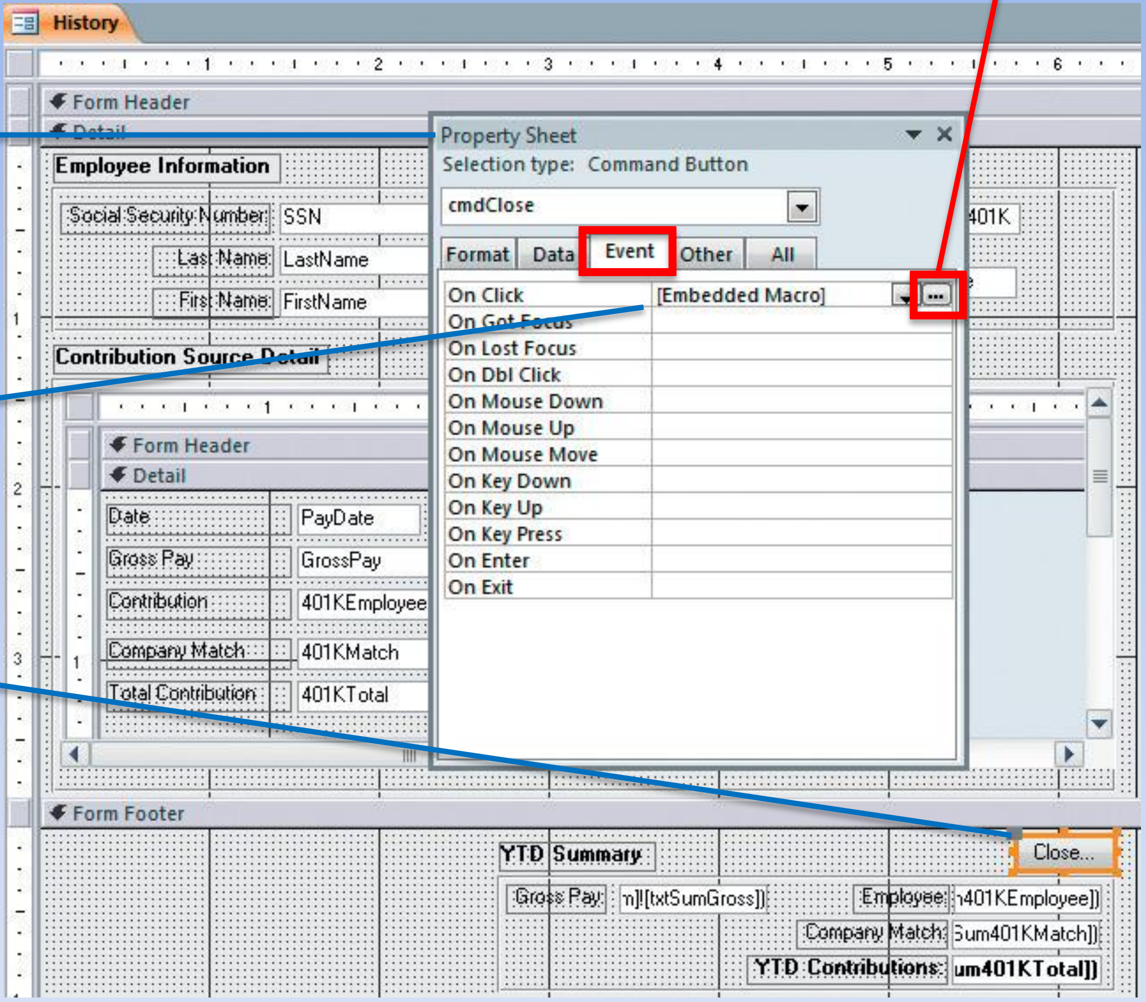
Embedded Macro

Macro Builder Button

Command Button Property Sheet

Embedded macro attached to On Click event

Command button control whose Property Sheet appears above



Property Sheet

Selection type: Command Button

cmdClose

Format | Data | Event | Other | All

On Click	[Embedded Macro]
On Got Focus	
On Lost Focus	
On Dbl Click	
On Mouse Down	
On Mouse Up	
On Mouse Move	
On Key Down	
On Key Up	
On Key Press	
On Enter	
On Exit	

Understand the Purpose of a Macro (continued)

- Two methods to create an embedded macro:
  - Access wizard
  - Add an existing macro to an existing control

Create a Stand-Alone Macro

- To start the Macro Builder:
  - Click the Create tab
  - Click Macro in the Macros & Code group
- When you finished building the macro:
  - Click Save
  - Type a descriptive name for the macro
- To run a stand-alone macro:
  - Double-click the macro name in the Navigation Pane or select the macro and press Enter

Create a Stand-Alone Macro (continued)

- To edit a macro:
  - Right-click on the macro name in the Navigation Pane
  - Select Design View from the shortcut menu
- After modifying the macro:
  - Save
  - Close the macro

Use the Macro Builder

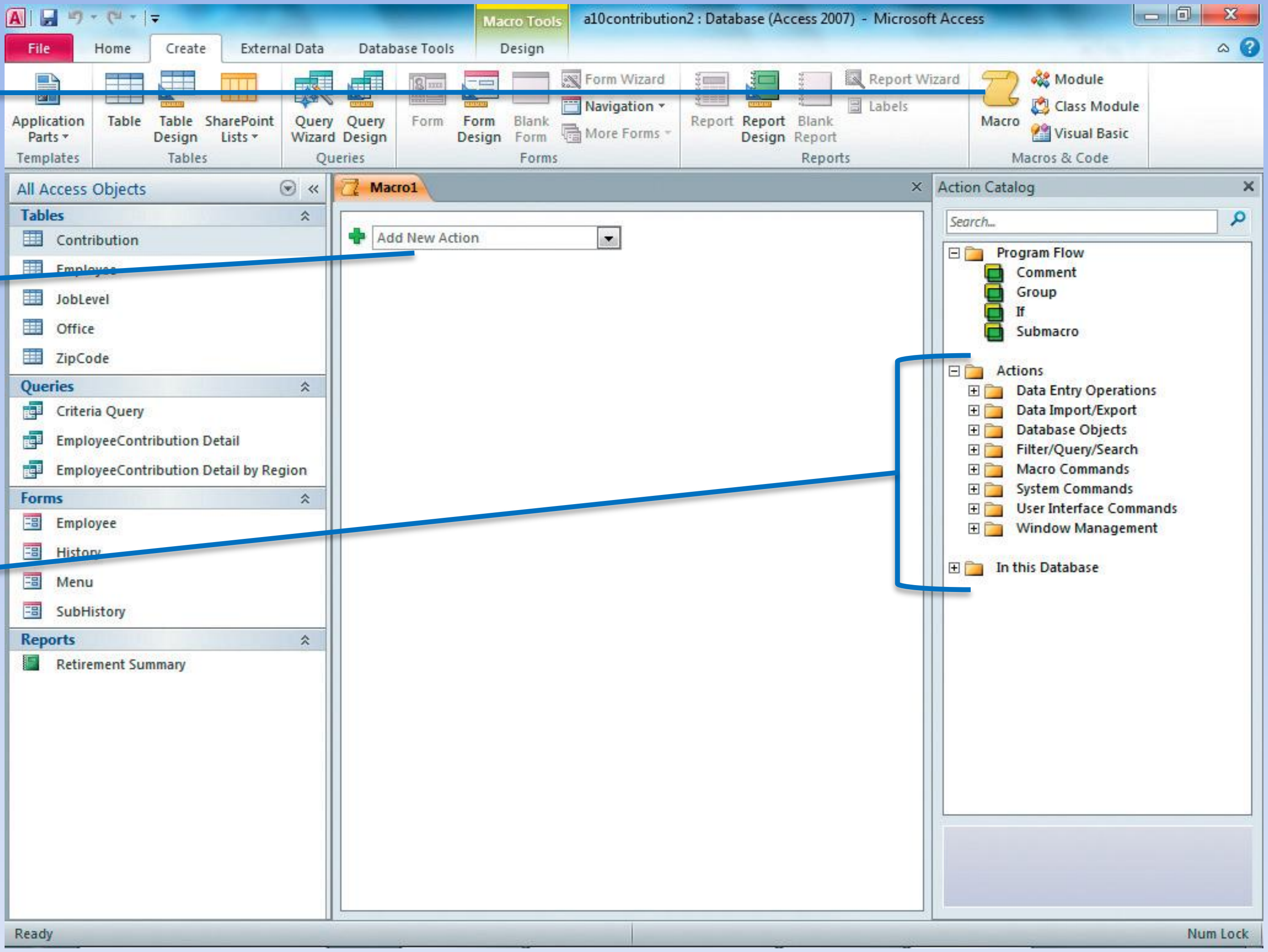
- **Macro Builder** – enables you to create and edit macros
- **Action –** A macro performs a specific series of actions.
- **Argument** – a variable, constant, or expression that is needed to produce the output for an action

Using the Macro Builder

Click Macro to open the Macro Builder

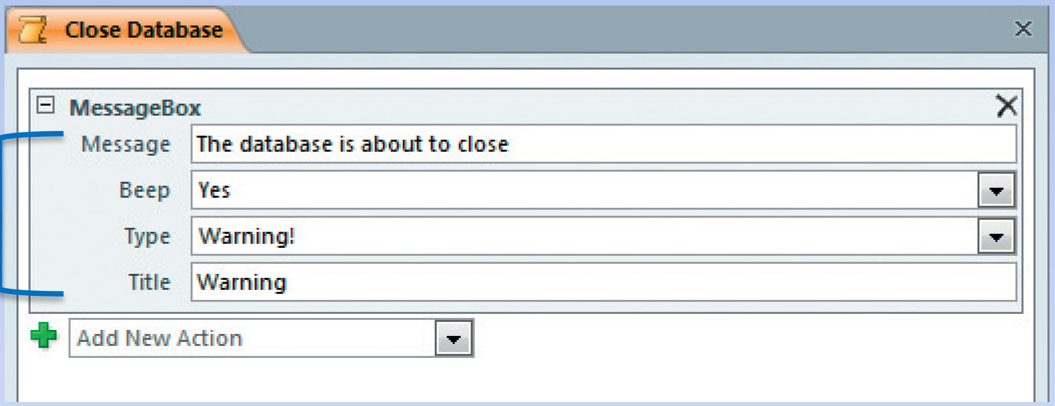
Add macro actions here

Use the Action Catalog to find and add actions



Using the Macro Builder

MessageBox action has four arguments: Message, Beep, Type, and Title.



Message	The database is about to close
Beep	Yes
Type	Warning!
Title	Warning

Macros Demo 1: Stand-alone macro

Open Query automatically

- Open the a10h1sunshine database
- Click Create > Macros&Code > Macro. Macro Builder opens.
- Click Add New Action arrow, select OpenQuery. Fill in the arguments by clicking arrows at right of box, and selecting Contribution Summary by Job Title, Datasheet, and Read Only.
- Save Macro as *Open 4 Objects*. Close the macro. (So far, only 1 object)
- Double-click the macro to run it. The query opens. Close it.

Macros Demo 2: Modify a Macro

Use the Macro Builder to modify a stand-alone macro

- Right-click the Open 4 Objects macro, and open it in Design view. Macro Builder opens.
- Click the Add New Action box, and type OpenForm, Enter.
- Arguments are Contribution Details, Form, Normal.
- Click the Add New Action box, and type OpenForm, Enter.
- Arguments are Employees, Form, Normal.
- Click the Add New Action box, and type OpenReport, Enter.
- Arguments are Detail Report, Report, Normal.
- Save the macro. Click Run. All four objects open.
- Close the four objects. Close the macro.

Attach an Embedded Macro to an Event

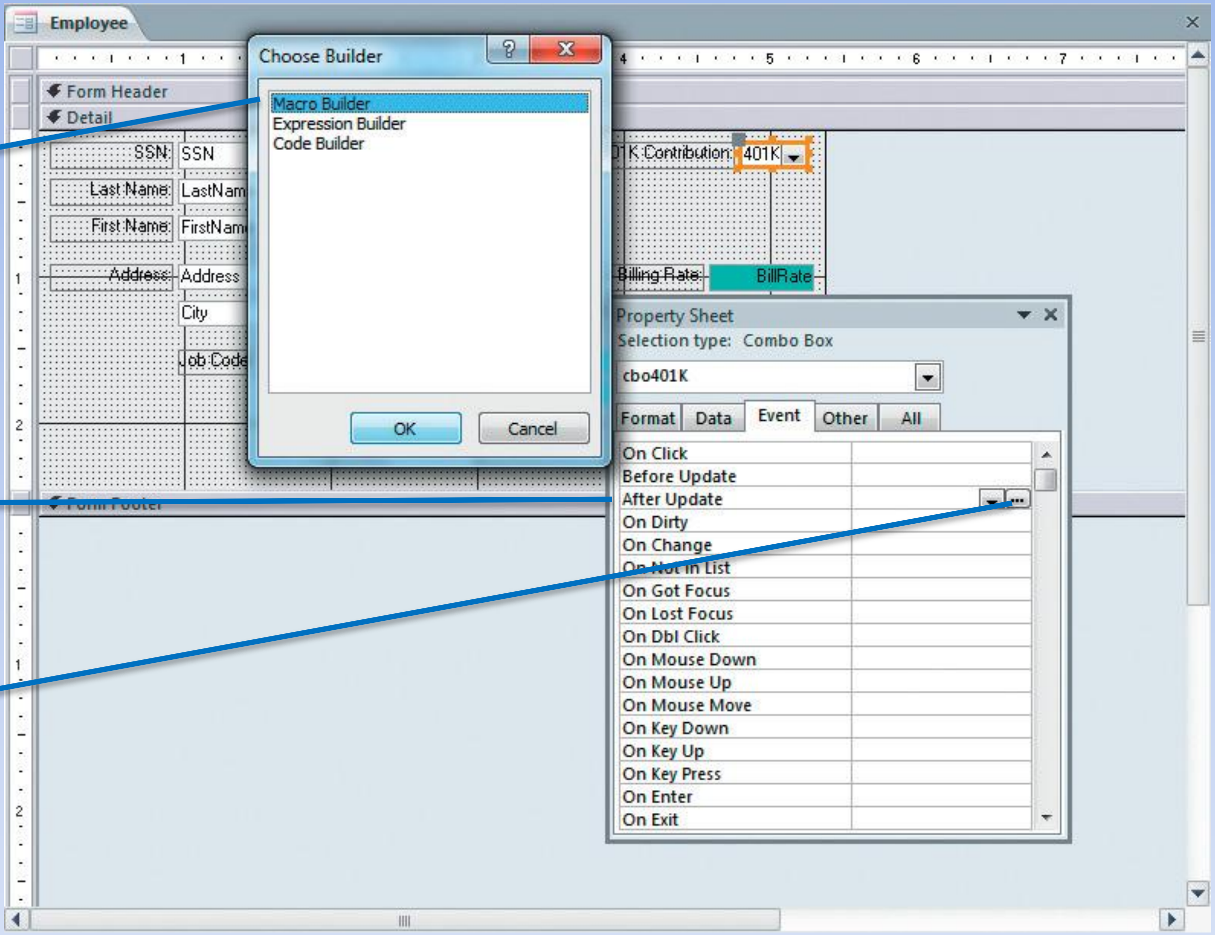
- Embedded macros:
  - Attached to an event of a control on a form or report, or to an event of the form or report object itself
  - Open in Design view to attach an embedded macro
- To attach a macro to an event:
  - Add a control that automatically creates an embedded macro using the Command Button Wizard
  - Attach an embedded macro to an event using the Property Sheet

Attach an Embedded Macro to an Event

3. Select the Macro Builder in the Choose Builder dialog box

1. After Update event is selected

2. Click Build



Macros Demo 3: Embedded Macro

Attach an Embedded Macro to an Event using Command Button wizard

- Right-click the Main Menu form, and select Design View
- Click Button control in Controls group. XXXX Then click between "Contribution Details" & "Detail Report" buttons. The Command Button wizard opens.
- Select Miscellaneous from the Categories list, then select Run Query from the Actions list. Click Next.
- Click Parameter Query (look in Nav. pane Queries), then Next. Click Text option, then type "Parameter Query" as the button text. Click Next.
- Type cmdParameterQuery, click Finish.
- Increase button size to match other buttons. Do Format>Font Size>12.
- Switch to Form View, click the new button, type 70000, Enter.
- Close query, switch to Design view of Main Menu form. Click Parameter Query button, then Property Sheet, then Event tab, then Build on RHS of On Click property. Macro Builder opens, allowing you to modify the embedded macro, if needed. Close the Macro Builder.
- Save and close the Main Menu form.

Macros Demo 4: Embedded Macro

Attach an Embedded Macro to a Report Event: Prompt the Employees

- Right-click Detail Report in the Navigation Pane, and select Design View
- Click the Event tab on the Property Sheet
- Click Build (…) on the RHS of the On Open property box. The Macro Builder option should be selected by default. Click OK.
- Click in the New Action box, and then type MessageBox, Enter.
- Type Please deliver this report to Terry every Friday.
- Verify that Yes is in the Beep box. Select Information in the type box. Type "Check employee contributions" in the Title box.
- Save and close the macro, then click Print Preview in the Views group. The message should appear. Or double click Detail Report in the Navigation Window - the message should appear.
- Click OK, close Print Preview, close and save the report.

Identify When to Use a Data Macro

- Data macros are only used with table events and not with other objects
- A form based on a table that contains a data macro will inherit the logic of the table
- Examples of when to add a data macro to a table:
  - Verify that a customer has no outstanding invoices
  - Keep a log of any changes made to a specific table
  - Send a confirmation email

Create an Event-Driven Data Macro

- Event-driven Data Macros
  - Triggered when a table event, such as After Delete or Before Change, occurs
  - Occur naturally as users enter, edit, and delete table data
- Named Data Macros
  - Can be accessed from anywhere in the database
  - Can accept parameters

Create a Named Data Macro

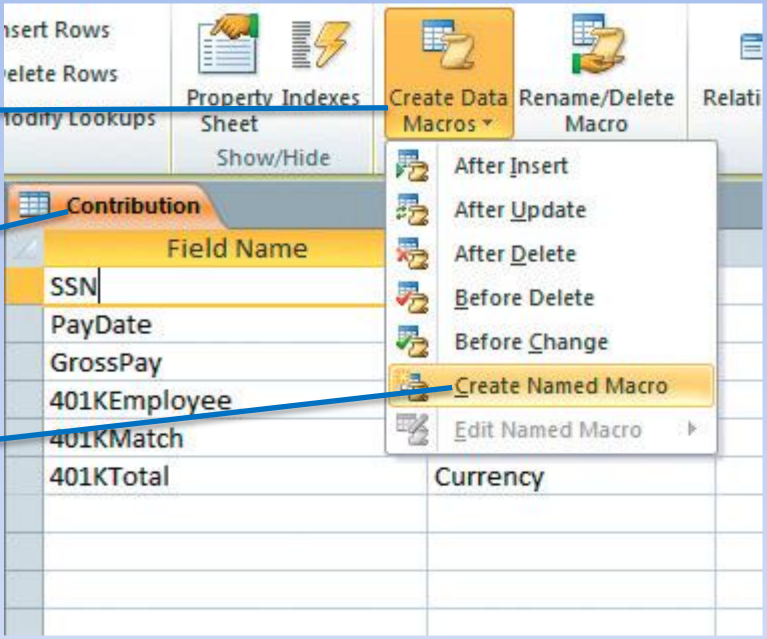
- Named Data Macro
  - Created and edited in table Design view
  - Save the macro with a descriptive name, such as DataMacro-Email
  - Can be run from within another macro

Create a Named Data Macro

Create Data Macros

Design view of the Contribution table

Create Named Macro



Working with Data Macros Demo 1

Identify when to use a data macro

- Open Employee table in Design view.
- Add a new field, EligibleForLeave, data type Yes/No, to the table – save the table.
- Switch to Datasheet view. Update the first 10 records to indicate whether they are eligible (≥1 year service).
- Tedious. Let's get a macro to do it automatically.

Working with Data Macros Demo 2

Create a Before Change data macro

- Switch to Design view. Click Design tab > Field, Record & Event > Create Data Macros > Before Change event.
- The Macro Builder appears. Drag the If statement from the Program Flow folder in the Action Catalog to the Add New Action box in the macro.
- Type (Date()-[HireDate])>365 in the Conditional expression box.
- Select SetField from If Action > Add New Action Arrow
- Type el then press tab in the Name box → EligibleForLeave
- Type Yes in the Value box. Save & close macro. Save table.

Working with Data Macros Demo 3

Test and modify a data macro

- Switch to Datasheet view. Type 0.03 in the 401k field of record 11 (Sally Keller) then click in the 401k field of record 12. Sally is eligible, and the macro automatically ticks the box.
- Change the date in record 12 to be sometime within the last year. Not eligible, no tick.
- Tick the box in an ineligible record. Macro doesn't override. ☹
- Switch to Design view. Click Create Data Macros in Field, Record & Table group, click Before Change event.
- Click If statement, click Add Else link. Type s tab→SetField
- el tab→EligibleFoLeave in Name box, No in Value box.
- Save macro, close macro, save table.
- Datasheet view, locate fudged record, check EligibleForLeave, move to another record. Macro now unchecks the record.

Working with Data Macros Demo 4

Build a Named Data Macro for Table Employees

- Switch to Design View. Click Create Data Macros > Create Named Macro. Macro Builder appears.
- Drag SendEmail action from Data Actions folder to Add New Action box.
- Type your email address in To, Subject=Sunshine Database Alert, Body=The Employee table has been altered.
- Save macro as DataMacro-SendEmail. Close the macro.
- Save and close the Employee table

Working with Data Macros Demo 5

Attach a Named Data Macro to a Form

- Open the Employees form in Design view.
- Click After Update on the Event tab of the Property Sheet.
- Click the Build (…) button of After Update, click Macro Builder if necessary, OK.
- Click the Add New Action arrow, and select RunDataMacro.
- Select Employee.DataMacro-SendEmail using the RunDataMacro arrow. Save and close the macro.
- Save Employees form, switch to Form view.
- Retype value in 401k field of first record, press tab twice. Access tries to send the email.

Summary of Macro Types in Access

Macro Type	Characteristics
Stand-alone	Appears in navigation pane; double-click to run it.
Embedded	Not in nav. pane. Runs when a particular event occurs in a form or report.
Data Macros	New in Access 2010. Associated with table events. Two types: see below
Event-driven	Run when a specific event (e.g. Before Change) in a specific table occurs
Named	Not triggered by a single event – can be invoked (possibly with parameters) by calling it within another macro attached to an event.

Understand the Fundamentals of SQL

- Structured Query Language (SQL)
  - Industry standard language for defining, manipulating, and retrieving the data in a database
  - Developed at IBM by Donald Chamberlin and Raymond Boyce in the early 1970s
  - Used to create, query, and modify tables
  - Microsoft has its own version of SQL for Access

Interpret an SQL Select Statement

- **SQL Select Statement** – is used to retrieve data from the tables in a database
- **SQL Keywords** are words (sometimes phrases) that have predefined meanings in SQL – rather like **if** … … **else** … in Javascript

Interpret an SQL Select Statement (continued)

- Four basic SQL keywords:
  - **SELECT** - Instructs Access to return the specific fields from one or more tables
  - **FROM** - Specifies the table(s) that will be searched
  - **WHERE** - Specifies the criterion or criteria that records must meet to be included in the results
  - **ORDER BY** - Is used to sort the records by a certain field in either ascending or descending order

Basic Structure of an SQL Statement

- Basic structure of an SQL statement:

```
SELECT field names
FROM table name
WHERE condition that must be met
ORDER BY field name;
```

Sample SQL

-
  ```
  SELECT Contribution.SSN,
         Contribution.PayDate,
         Contribution.GrossPay
  FROM Contribution
  WHERE (((Contribution.GrossPay) >=6000))
  ORDER BY Contribution.GrossPay DESC;
  ```

-
  ```
  SELECT * Contribution
   FROM Contribution;
  ```

Note: * selects all fields from the table
Note: You can view queries you create in Design View as SQL.

Using SQL Demo 1

Fundamentals of SQL

- Click Create > Query Design. Show Table box appears.
- Close Show Table box without adding any tables.
- The SQL command appears in the Results group (top left).
- Click the SQL command. Type SELECT * FROM Employee; in the SQL window, then click Run.
- Click View arrow > SQL view. Revise SQL statement to SELECT * FROM Employee WHERE JobCode=100; then click Run. Only 7 records appear (down from 18).
- Click View arrow > SQL view. Revise to SELECT * FROM Employee WHERE JobCode=100 ORDER BY HireDate; then click Run. Same 8 records, different order.
- Save query as SQL Practice Statement. Close query.

Using SQL Demo 2

Create query, view equivalent SQL

- Click Create > Query Design. Show Table box appears.
- Add Contribution and Employee tables. Close Show Table.
- Add all fields from Contribution to the query, and CountyID from Employee. Run the query. 219 records.
- Switch to Design view, type 1 in CountyID criteria row. Run. 48 records. Back to Design, select Descending in EmployeeTotal sort row. Run query. Switch to SQL view:

```
SELECT Contribution.SSN, Contribution.PayDate, Contribution.GrossPay,
Contribution.EmployeeContribution, Contribution.CompanyMatch,
Contribution.EmployeeTotal, Employee.CountyID
FROM Employee INNER JOIN Contribution ON Employee.SSN = Contribution.SSN
WHERE (((Employee.CountyID)=1))
ORDER BY Contribution.EmployeeTotal DESC;
```

Summary

- In this section, you learned how to use the following advanced techniques in Microsoft Access 2010:
  - Macros: to verify data entry and automate tasks
  - SQL statements

Reference

Grauer, Mast & Poatsy: *Access 2010 Comprehensive*, chapter 10: Using Macros and SQL in Access.