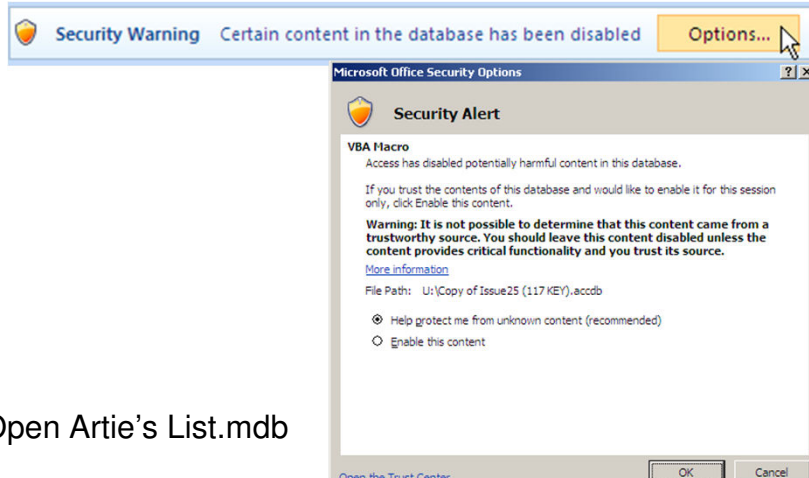


# Entry Skills Boot Camp

Databases, Tables, Relationships,  
Queries, Forms

## Access Security

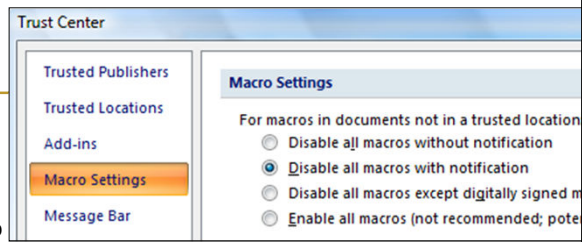
- Macros/Visual Basic code can be **malicious**
  - for safety, macros and VB code can be disabled, optionally with a warning



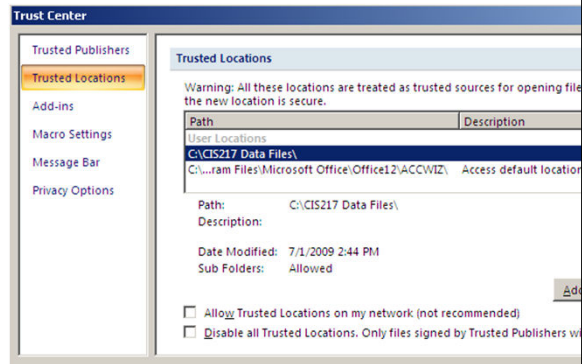
- Open Artie's List.mdb

## Access Security

- Can adjust macro security settings via
  - File | Options | Trust Center | Trust Center Settings... | Macro Settings

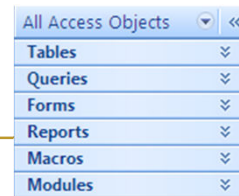


- Trusted Location
  - Access will execute macros/VB code if the file is in a Trusted Location
  - File | Options | Trust Center | Trust Center Settings... | Trusted Locations



## Access Database Objects

- Database
  - a collection of related tables and other objects
- Table
  - an object that stores data in rows (records) and columns (fields)
- Query
  - an object that selects requested fields and records from 1 or more tables
- Form
  - an object used to view/enter/change records on-screen in custom layout
- Report
  - an object you can use to print records in a custom layout



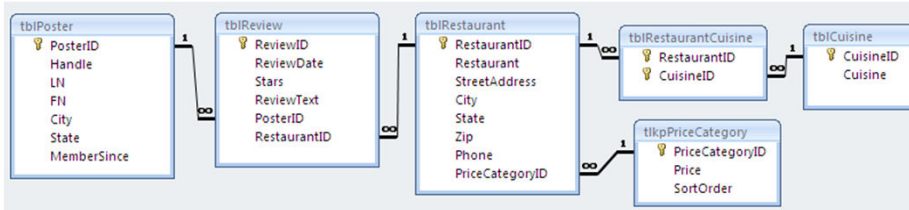
## 7 Types of Access Objects

All Access Objects	<<
Tables	∨
Queries	∨
Forms	∨
Reports	∨
Macros	∨
Modules	∨

- Data Access Page
  - a special type of **Web page** that works like an Access form
  - were new in Access 2000 – deprecated in Access 2007
- Macro
  - an object containing a series of **actions** you want Access to carry out
- Module
  - an object that stores Visual Basic **procedures** written to automate, customize, and enhance a database
- All objects are stored in a single .ACCDB/.MDB file
- Explore Artie's List.mdb

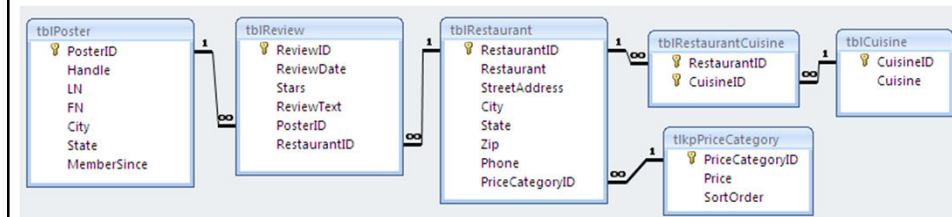
## Inter-Table Relationships

- Primary Key
  - a field, or combination of fields, that **uniquely** identifies each table record
    - no duplicates
    - composite primary key
- Foreign Key
  - a field, or combination of fields, in one table whose values must match the values of the primary key of some table
- Join
  - using a **common field** to link two tables that contain related data
  - storing data in separate, joinable tables avoids **data redundancy**



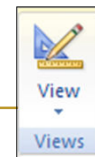
## One-to-Many Relationship (1:M)

- The most common type of inter-table relationship
- Each record in Table A can have **many** matching records in Table B but each record in Table B has **only one** matching record in Table A
  - e.g.: each **Restaurant** has one **PriceCategory** but each **PriceCategory** can apply to many **Restaurants**
  - e.g.: each **Review** is for one **Restaurant** but each **Restaurant** can have many **Reviews**
- Implementation
  - include the **primary key** of the one table as a **foreign key** in the many table



## Two Views of a Table

- Datasheet View
  - see/change the table's **data** as a grid
  - rows = records columns = fields
  - can view, insert, update, delete table records
- Design View
  - see/change the table's **structure**



## Entering Records

- Generally want users to add records using a Form
  - not a table's Datasheet
  - forms are preferred... they're flexible and powerful
- Record Navigation buttons
- Current Record Box
- New Record button
- [Tab] to move between fields
- Pencil symbol
  - indicates a **dirty** record... one whose data has changed

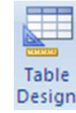
Contacts	
Contact ID	1
First Name	Hardy
Last Name	Griffin
Company	Ballard Brewery

## Saving



- Save Button
  - saves changes to an object's **layout** or **structure**
  - also saves data changes (in Access 2007)
- Access automatically saves changes to the current record's data when you
  - move to another record
  - close the form/datasheet
  - press [Shift]+[Enter]
  - use Home | Records | Save Record
- Practice Time
  - add yourself as a new **Poster**, entering the current date/time in the MemberSince field

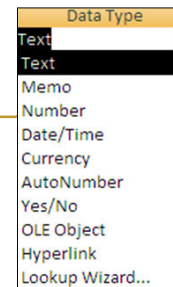
## Creating Table in Design View



- Each record in a table has the same field structure
- Creating a table involves
  - specifying each field's name, data type, other properties
  - declaring a primary key
  - naming the table object
- Advantages of setting a primary key for each table
  - each record can be **uniquely** identified
  - makes it possible for a table to participate in **relationships**
  - Access automatically prevents duplicates
  - Access requires the primary key be provided (not be left null)
  - Access **displays** rows in primary key sequence (by default)
- Create tab | Table Design
  - **Table Design** grid on top and **Field Properties** beneath



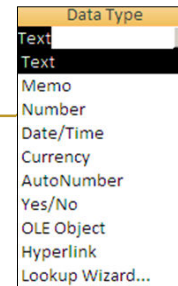
## Assigning Field Data Types



- Text
  - for storing text entries (no calculations)
  - defaults to 255 bytes
  - stored as a variable-length string
- Memo
  - longer, free-form text entries, typically sentences or paragraphs
  - < 1GB, of which you can display 65,535 characters in a control
- Number
  - for storing numeric values that are non-monetary
  - can be used for calculations
- Date/Time
  - important in many business applications
  - stored internally as 8-byte floating point **serial number in date.time**
    - 37145.375 corresponds to 9/11/2001 at 9:00 am
    - 37146.500 corresponds to 9/12/2001 at 12:00 pm (lunchtime)
  - can be used for date/time calculations

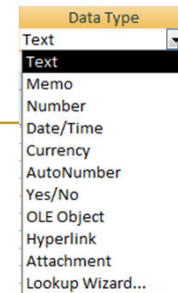
## Assigning Field Data Types

- Currency
  - for monetary values
  - can be used for calculations
- AutoNumber
  - a field containing values that Access automatically increments as each new record is added to table
  - guarantees a unique field value for each record... for a PK
  - we can't control its values
- Yes/No (logical)
  - presence/absence of an attribute
  - true/false on/off
- OLE Object
  - objects such as pictures, sound, files from other applications
  - not directly modifiable in Access




## Assigning Field Data Types

- Hyperlink
  - a path to a document, web page, or other destination
- Attachment
  - new Access 2007 (.accdb only, ACE)
  - attach files (image, spreadsheet, document) to a record
  - can view and edit attached files, depending on field properties
  - provide greater flexibility than OLE; compressed to store more efficiently
- Lookup Wizard
  - not actually a data type
  - used to help set properties for a foreign key field to make it easy to link a record in one table to a related record in another

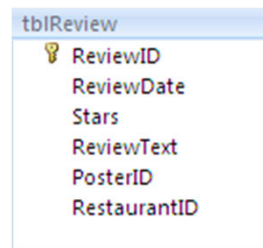



## Assigning Field Sizes

- **Field Size property**
  - controls the maximum number of characters that can be entered into a Text field, or the range and kind of values that can be stored by a Number or AutoNumber field
  - is predefined for other data types (e.g.: Date/Time, Currency)
  - balance field storage needs vs. resources consumed
- **Demo**
  - use [Access Help](#) to learn about the Field Size property
    -  to launch Help
    - press [F1] to get context-sensitive help
  - special focus: Number field size options
    - Byte
    - Integer
    - Long Integer
    - Single
    - Double
    - Decimal

## Practice Time: Create a Table

- **Create tblReview**
  - use appropriate data types & field size
  - ReviewID: automatically assigned
  - Stars: whole number 1-5
  - ReviewText: can be very long (avg = 325 chars)
  - PosterID and RestaurantID: number (long integer)
  - set the table's primary key
  - save the table's structure
- **Enter one complete review using your PosterID and RestaurantID=28 (Voce Ristorante & Lounge)**



tblReview	
	ReviewID
	ReviewDate
	Stars
	ReviewText
	PosterID
	RestaurantID

## General Properties Tab

### ■ Practice Time

- view Help topics for field properties
  - Caption
  - Default Value
  - Validation Rule
  - Validation Text

Field Name	Data Type
RestaurantID	AutoNumber
Restaurant	Text
StreetAddress	Text
City	Text
State	Text
Zip	Text
Phone	Text
PriceCategoryID	Number

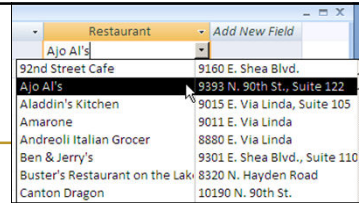
  

Field Properties	
Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	Price Category
Default Value	
Validation Rule	Is Not Null
Validation Text	You must select a Price Category for this rest
Required	No
Indexed	Yes (Duplicates OK)
Smart Tags	
Text Align	General

## Practice Time: Modifying a Table's Structure

- Make the following changes to **tblRestaurant**
  - Reduce Field Sizes: Zip to 10, State to 2
  - Add new Homepage field to store web site URL
    - switch to Datasheet View, navigate to record for Voce Ristorante & Lounge, enter [www.voceristoranteandlounge.com](http://www.voceristoranteandlounge.com) as its Homepage, click it to view
- Make the following changes to the **tblReview**
  - Specify `=Date( )` as the **Default Value** for ReviewDate
    - switch to Datasheet, verify that new record shows today's date
  - Set **Caption** so ReviewText heading displays as Review Text (two words)
    - switch to Datasheet, check the column heading
  - Use **Is Not Null** as **Validation Rule** for RestaurantID and for PosterID.  
Also include an appropriate helpful **Validation Text** for each
    - switch to Datasheet, test both Validation Rules & Validation Text: enter ReviewText but no PosterID or RestaurantID, then try to save the record. Click OK to dismiss the error message, then enter 14 as RestaurantID. Try to save again. Click OK to dismiss the error message, then enter your PosterID. Save the review.

## Lookup Field: Concepts



- Selecting from a **combo box** is more convenient
  - no need to memorize RestaurantID codes... just pick restaurant by name
- **Lookup Wizard**
  - sets a **foreign key** field's **properties** so you see descriptive text but store a foreign key value that matches the primary key in the related table
  - creates an **inter-table relationship** between the two tables
- **Demo**
  - use Lookup Wizard to have tblReview RestaurantID display restaurant name and address, sorted by name, and hide column of RestaurantIDs
- **Practice Time**
  - use Lookup Wizard to have tblReview RestaurantID as above
  - use Lookup Wizard to have tblReview PosterID display sorted list of poster Handles, hiding the column of PosterIDs
  - enter another complete review, pick any restaurant , pick any poster other than yourself

## Lookup Properties Tab

The screenshot shows the 'Field Properties' window for the 'RestaurantID' field in the 'tblReview' table. The 'Lookup' tab is active, displaying the following properties:

Property	Value
Display Control	Combo Box
Row Source Type	Table/Query
Row Source	SELECT tblRestaurant.RestaurantID, tblRestaurant.Restaurant, tblRestaurant.Stre...
Bound Column	1
Column Count	3
Column Heads	No
Column Widths	0";2";2"
List Rows	16
List Width	4"
Limit To List	Yes

Below the 'Field Properties' window, a table shows the 'Field' properties for the 'RestaurantID' field:

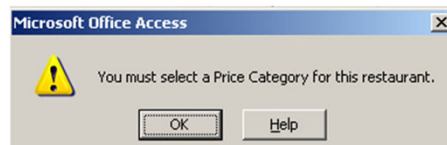
Field:	RestaurantID	Restaurant	StreetAddress
Table:	tblRestaurant	tblRestaurant	tblRestaurant
Sort:		Ascending	
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Criteria:			
or:			

## Referential Integrity Concepts

- Each non-null foreign key value must match a primary key value in the related table
  - each Review's RestaurantID must match RestaurantID of actual Restaurant
- Bottom line: no unmatched foreign keys
- Prevents **orphaned** records in a related table
  - cannot add new record to the child table with an unmatched foreign key
    - demo: try to change a restaurant's PriceCategory to Expensive
  - cannot modify or delete the primary key value in the primary table and leave the corresponding foreign key(s) orphaned
    - demo: attempt to delete Under \$20 from tlkpPriceCategory
  - Cascade Delete option
    - Access also deletes the matching row(s) in the related table
  - Cascade Update option
    - Access also changes foreign key values in matching rows in the related table
  - business rules will determine whether it is appropriate to use cascade option

## Referential Integrity Concepts

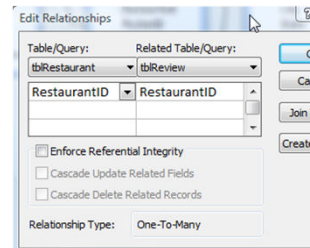
- When no foreign key value is entered, Access has no value to check and referential integrity is not at issue
  - whether to allow a Null in a foreign key depends on business rules
    - can use Validation Rule and Validation Text field properties to prevent Null foreign keys



## Relationships Window



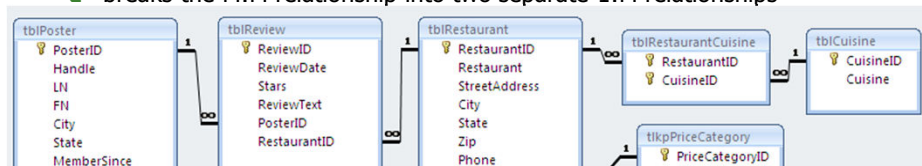
- To create/delete/modify inter-table relationships
  - create: drag PK from the One table & drop on FK in the Many table
  - delete: click join line, press [Del]
  - modify: double-click the join line
- Option to Enforce Referential Integrity
  - Cascade Updates option, Cascade Deletes option



- Practice Time
  - Modify relationship between **tblPoster** and **tblReview** to enforce referential integrity and cascade updates but not cascade deletes
  - Modify relationship between **tblRestaurant** and **tblReview** to enforce referential integrity and cascade updates but not cascade deletes

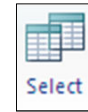
## Many:Many Relationships (M:M)

- Each record in Table A can have **many** matching records in table B and each record in Table B can have **many** matching records in Table A
  - e.g.: each **Restaurant** can offer many **Cuisines** and each **Cuisine** can be offered by many **Restaurants**
  - e.g.: each **Poster** can review many **Restaurants** and each **Restaurant** can be reviewed by many **Posters**
- Implementation
  - create a third table (**junction table**, intersection table) which includes the primary keys from table A and table B as foreign keys
    - consider using them as a composite primary key in the junction table
  - breaks the M:M relationship into two separate 1:M relationships




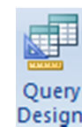
## Query Concepts

- **Select Query** (other types of queries later in course)
  - used to view, analyze and edit data from one or more tables
- **Select queries can**
  - choose which fields to see – place desired fields onto design grid
  - choose which records to see – specify criteria
  - sort records more flexibly
  - pull data from multiple related tables
  - make changes to data in tables
  - perform calculations
  - serve as record source for form, report, or another query



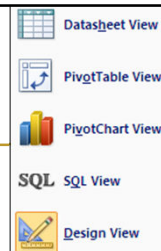
## Creating and Running a Select Query

- **Create tab | Query Design**
  - query can have fields from multiple tables
- **Inner Join**
  - a row in one table is included in query results if there is at least one **matching** row in the other joined table
  - rows that have no matching rows in the other table won't be shown
-  **button**
  - executes the query, then displays query's results in Datasheet View
- **Practice Time**
  - Create a query that shows each restaurant's name, any review dates, stars, review text and the handle and last name of the poster. Save as **qryRestaurantReviews** when correct.
    - note how few rows there are...



## Query Views

- Design View
  - create/modify query's structure
  - specify which fields to display, which records to display, how to sort, perform calculations
- Datasheet View
  - view the set of records that answers the question you posed
  - Select queries generally produce a **dynaset**
    - a **dynamic** view of a **set** of records
  - can insert/update/delete rows in a dynaset and the changes are stored in the underlying table
- SQL View
  - displays the SQL statement for the current query
  - when you create a query in Design view, Access constructs the equivalent SQL statement
  - when Access saves a query it stores the text of the SQL statement



## Sorting Query Results

- Query Results are displayed in PK sequence by default
  - query results may be easier to interpret when sorted by other field(s)
- Access allows up to 10 different sort fields
  - each can be either Ascending or Descending
  - unique vs. non-unique sort keys
  - leftmost sort field is **primary sort**, sort field to right is **secondary sort...**
    - may need to drag fields to new locations on the grid
- Practice Time
  - Create a query that shows (left-to-right) each restaurant's name, address, city, state, zip, and phone, sorted primarily by zip, and secondarily by restaurant name. Save as `qryRestaurantsByZip` when working correctly.

## Record Selection Criteria

- **Condition**
  - a criterion that controls which records are selected
- **Are placed in Criteria row(s)**
  - date values are enclosed within # # e.g.: <#12/25/09#
  - text values are enclosed within " " e.g.: ="Trollen"
    - case *insensitive* by default
- **Exact Match (=)**
  - a record is selected for the results set only if its field value exactly matches the specified value
  - **Practice Time**
    - Create a query that shows each poster's Handle, MemberSince date, City, and State but only for Arizona members. Save as `qryAZPosters` when correct.

## Record Selection Criteria

- **Range of Values Match (<, <=, >, >=, Between... And)**
  - record is selected for the results only if field value is in specified range
  - **Practice Time**
    - Create a query that shows each poster's Handle, City, and MemberSince date, but displays only for posters who joined after 1/1/2008. Sort the results with the most recently joined member on top. Save as `qryNewestPosters` when working correctly.
- **Is Null**
  - a record is selected for the query results only if the field *lacks* a value
  - **Practice Time**
    - Create a query that shows each restaurant's ID, Name, and Phone but only for those which have no phone number. Save as `qryRestaurantsWithoutPhone` when correct.

## Defining Multiple Selection Criteria

- Combine two or more simple conditions using logical operators (AND, OR)
- **AND**
  - place conditions on the **same** Criteria row
  - all conditions on the row must be true for the record to be selected
  - **Demo:** query to display AZ posters who have joined since 1/1/2008
- **OR**
  - place conditions on **different** Criteria rows
  - at least one condition row must be true for the record to be selected
  - **Demo:** display AZ posters or any poster who joined after 1/1/2008
- **Practice Time**
  - Ian lives in 85251. He would like to see each restaurant's Name, Address, Zip and Price (from tkpPriceCategory), for every restaurant in 85251. He would also consider restaurants in 85260 as long as their Price is "Under \$20". Sort by Zip code. Save as **qryIan'sOptions**.

## Calculated Column Concepts

- **Calculated Column**
  - an expression that Access evaluates and displays in the query results
    - eg: **Late Fee:** `[Invoice Amt]*0.03`
  - multi-word field names must be enclosed within [brackets]
  - the column appears in query results but is **not stored** in a table
  - don't store values that can be calculated from fields that you do store
    - wastes space
    - would need to update the calculated value when the other field changes
  - since you can **calculate the value on the fly**, you don't need to store it
- **Concatenation operator**
  - & symbol - performs string addition
  - `LabelLine2: [City] & ", " & [State] & " " & [Zip]`

## Practice Time: Calculated Columns

- Create a query that shows each poster's Handle, full name, MemberSince, and calculates how many days they have been a member.
  - FullName: [LN] & ", " & [FN]
  - DaysAsMember: Date()-[MemberSince]
- Format the DaysAsMember column to have commas and no decimal digits
- Save as qryPosterTenure when correct

## Uses of Forms

- Forms serve many roles in an application
  - data entry screens (bound forms)
    - to view, add, edit, delete records
  - application flow (unbound forms)
    - Switchboard form (Callahan 1)
    - command buttons on forms for jumping to related forms/reports (Callahan 9)
  - custom dialog boxes (unbound forms)
    - to gather information from user (Callahan 8)

The screenshot shows a 'Subscribers' form with the following fields and values:

First Name	Hard	Subscriber ID
Last Name	Griffin	
Company	Ballard Brewery	
Address	115 Leary Wy.	
City/State/Zip	Seattle WA 98117	
Phone	(206) 555-5317	Paid Through

Buttons: Go To Record, Payment History, Enter Payment

Record: 1 of 20, No Filter, Search

The screenshot shows two windows:

**Main Switchboard - Contact Management**

- Enter/View Contacts
- Enter/View Other Information...
- Preview Reports...
- Change Switchboard Items
- Exit this database

**Print Mailing Labels**

Mailing Type | Label Type

- Journal Mailing (all current subscribers)
- Invoice Mailing (subscriptions expiring within 3 months)
- Lapsed Subscription Mailing (subscriptions that ran out in past 3 months)

Buttons: Print Labels, Preview Labels, Cancel

## Creating a Simple Form

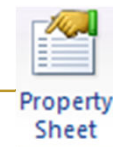
### ■ Form

- a customized on-screen view of table data, commonly displaying one record at a time
- the most powerful and flexible way to present data to users
  - table datasheets are plain and limited
- generally want form to resemble the paper form that captures the source data

### ■ Practice Time

- Use Form Wizard to create a new form
  - display all fields from tblPoster
  - Columnar layout
  - Northwind Style
  - save as frmPostersAndReviews

## Form Design Window



### ■ Property

- a characteristic of an object, such as name, width, color
- determine an object's appearance & behavior
- all objects have properties
  - form properties
  - section properties
  - control properties

### ■ Property Sheet

- displays the selected object's properties
- open via Properties Sheet Button or [F4] or [Alt]+[Enter]

### ■ Record Source Property

- specifies the source of data for a form (or report)
  - if form's data comes from a single table, base the form on the table
  - if form's data comes from multiple tables, base the form on a query
  - what if RecordSource is blank?

## Form Sections

- **Form Header**
  - appears at top of screen, won't scroll off the screen as scroll through Detail section
  - for logo, title, navigation combo box, command buttons
- **Detail**
  - displays the records from the underlying table or query
  - can display either one record at a time or several (DefaultView property)
- **Form Footer**
  - appears at bottom of screen, won't scroll off the screen
  - typically used for command buttons or instructions
- **Page Header**
  - information to be printed at top of each page (e.g.: title, column headings)
- **Page Footer**
  - information to be printed at bottom of each page (e.g.: date or page numbers)
- **Sections have Properties**
  - egs: Height, Back Color
  - each section has its own Height, but every section has same Width

The screenshot shows a 'Contacts' form with a dropdown menu set to 'Griffin, Hardy'. The form contains several input fields: Contact ID (1), First Name (Hardy), Last Name (Griffin), Company (Ballard Brewery), Dear (Red), Address (115 Leary Wy.), City (Seattle), State/Province (WA), Postal Code, and Country. On the right side, there are labels for Title, Work, Mobile, and Fax. At the bottom, there are buttons for 'Calls...', 'Dial...', and 'Add', along with a 'Page: 1 2' indicator and a 'Records: 1 of 22' status bar.

## Controls

- An object placed on a form to display field data, decorate the form, or provide other functionality
  - egs: text box, combo box, label, command button
- 3 categories of controls
  - **Bound control**
    - used to display/enter/edit field values
    - a control whose source of data is a field in a table or query
    - ControlSource property
  - **Unbound control**
    - a control that doesn't have a source of data
    - used to display static information, lines, pictures
  - **Calculated control**
    - a control whose source of data is an expression
    - e.g.: a text box with =Sum(InvoiceAmt) as its ControlSource property

## Control Types

- Label
  - use stand-alone labels for instructions or other static text
  - also accompany bound controls
  - **Caption** property
- Text Box
  - user types a value
  - **ControlSource** property
- Combo Box
  - user selects from a list or types an entry
- Check Box control
  - for yes/no true/false on/off

The screenshot shows a 'Contacts' window with a search bar containing 'Griffin, Hardy'. Below the search bar is a form with the following fields:

Contact ID	1	
First Name	Hardy	Title
Last Name	Griffin	Work
Company	Ballard Brewery	Work
Dear	Red	Mobile
Address	115 Leary Wy.	Fax
City	Seattle	
State/Province	WA	
Postal Code		
Country		

At the bottom, there are buttons for 'Calls...', 'Dial...', and 'Page: 1 2'. A status bar at the very bottom shows 'Record: 1 of 22', 'Unfiltered', and a search icon.

## Control Types

- List Box control
  - user can only pick from list
  - can allow multiple selections
- Option Group control
  - user selects from a set of **mutually exclusive** options
  - typically consists of a **frame** and a set of option buttons
- Option Button control
  - typically used in groups to force user to select 1 of a set of mutually exclusive choices

The screenshot shows a 'Go To Record' dialog box with a table of subscribers:




Name	Last Name	City	State
Griffin	Hardy	Seattle	WA
O'Halloran	Kevin	New York	NY
Stepanian	Kara	Boston	MA
Hunting	David	New York	NY
Christensen	Bonnie	Dallas	TX
O'Keefe	Maura	New York	NY
Geloff	Kevin	Seattle	WA
MacDougall	Rob	Los Angeles	CA
Sundberg	Deborah	Atlanta	GA
Doyle	Judy	New York	NY
Griffin	Gilbert	New York	NY

Buttons for 'Show Filter' and 'Cancel' are visible on the right.

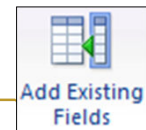
The screenshot shows a 'Filter Options' dialog box with two radio button options:

- All Contacts
- New York Contacts

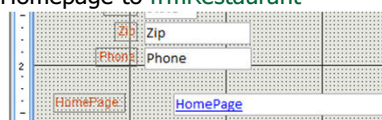
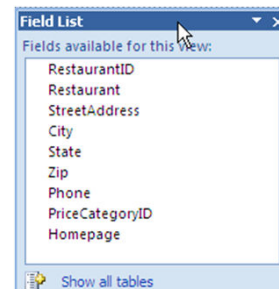
## Control Types

- Image control 
  - a frame for displaying a static image on a form or report
  - Insert Picture dialog box appears to specify source of image
- Subform/SubReport control 
  - form that is embedded on another form (the main form)
  - a main form/subform combination is sometimes referred to as a master/detail form, or a parent/child form
- Tab control 
  - a control that contains several pages, each with a tab, and each containing a group of related controls
  - helps use screen space efficiently



## Adding Fields to a Form



- Field List
  - lists the fields available in the form's Record Source
  - open via Add Existing Fields Button or [Alt]+[F8]
- Drag field(s) from Field List onto grid
  - each bound control has an associated label
  - when created from the Field List, bound controls inherit properties from the underlying table field
    - egs: Format, Input Mask
- To select several fields from Field List
  - [Shift] [Ctrl]
- Practice Time
  - add Homepage to frmRestaurant



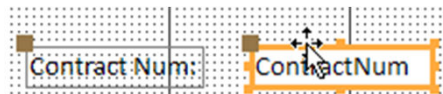
## Working with Controls

- Be able to move/size/copy/delete/align controls
- Techniques to select multiple controls
  - hold [Shift] while clicking
  - Select All button 
  - use Select Objects to lasso several objects 
  - click in the Horizontal Ruler
  - click in the Vertical Ruler
- Control Layout
  - a group of controls that can be manipulated together
  - Tabular or Stacked
- Handles
  - appear when a control and its associated label are selected
  - move handle and 7 sizing handles



## Moving and Resizing Controls

- Move text box and its associated label together



- Move text box but not its associated label



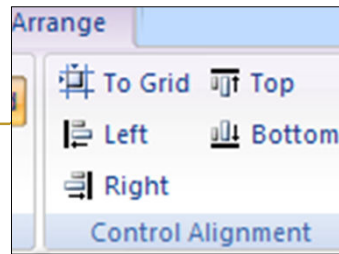
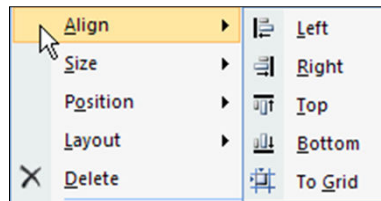
- Resize a selected control



## Aligning Controls

- Can mutually align controls via:

- Arrange Tab
- right-click a selected control



RestaurantID	<input type="text"/>	Price C
Restaurant	<input type="text" value="Andreoli Italian Grocer"/>	C
Street Address	<input type="text" value="8880 E. Via Linda"/>	
City	<input type="text" value="Scottsdale"/>	
State	<input type="text" value="AZ"/>	
Zip	<input type="text" value="85258"/>	
Phone	<input type="text" value="480-614-1980"/>	
Home Page	<input type="text"/>	

- Practice Time

- Modify frmRestaurant as illustrated
  - Homepage text box : move up; left-align with ones above, then resize
  - Homepage label: 2 separate words & remove colon, right-align with other labels; right align its text within the label
- Save frmRestaurant

## Changing the Tab Order



- Tab Order

- determines the order in which users navigate through controls by pressing [Tab]
- each section has its own tab order

- Two ways to set the Tab Order

- Manually assign each control's Tab Index property
  - 0, 1, 2, etc
- Arrange tab | Control Layout Group | Tab Order...
  - can either drag-and-drop or use Auto Order button
  - sets each control's Tab Index property for you

- Practice Time

- set the Tab Order for frmRestaurant
- Switch to Form View to test that the tab order works correctly
- Save frmRestaurant when done

## Practice Time

- Modify `frmPostersAndReviews`, as illustrated

Posters and Reviews

PosterID	<input type="text" value="1"/>	City	<input type="text" value="Mesa"/>
Handle	<input type="text" value="amgirl"/>	State	<input type="text" value="AZ"/>
Poster	<input type="text" value="McCracken"/> <input type="text" value="Aurora"/>	Member Since	<input type="text" value="03/19/2004 8:27"/>

- Save `frmPostersAndReviews` when done

## Subform Concepts

- Vividly illustrates a 1:Many relationship between records stored in separate tables
  - `tblPoster` has 1:Many relationship with `tblReview`
  - `tblPoster` records on main form
  - `tblReview` records on subform
  - 2 sets of Navigation Buttons
- The main form and subform are each saved as separate Form objects

Posters and Reviews

PosterID	<input type="text" value="24"/>	City	<input type="text" value="Scotts"/>
Handle	<input type="text" value="Skipper"/>	State	<input type="text" value="AZ"/>
Poster	<input type="text" value="Trollen"/> <input type="text" value="Tom"/>	Member Since	<input type="text" value="07/04/"/>

ReviewID	RestaurantID	Review Date	Stars	Review Text
1	Voce Ristorante & Lounge	7/5/2009	5	An elegant venue; the music wonderful!
3	El Paso Bar-B-Que Co.	7/20/2009	2	Food was cold, beer was war
*(New)		7/22/2009		

Record: 1 of 2 of 2 | No Filter | Search

Record: 1 of 23 of 23 | No Filter | Search

## Adding a Subform

- Subform Wizard
  - guides you to specify subform's source of data and fields, builds a form and embed it on the main form as a subform
  - sets two subform properties to synch subform records with main form
    - Link Child Fields property
    - Link Master Fields property
- Practice Time
  - use Wizard to embed a subform which shows all fields from tblReview except PosterID, as illustrated
  - apply Calibri 9 pt to subform; adjust subform column widths, increase datasheet row height, as illustrated
  - resize and place the subform, as illustrated
  - switch to Form View to verify function and appearance
  - save frmPostersAndReviews when done

## Compacting a Database

- When you delete an object, it is not actually removed from database
- Compacting creates a new .accdb/.mdb and then copies non-deleted objects into the new file
  - requires free space on disk
- Microsoft encourages regular compaction to promote database health
- File | Compact and Repair Database
- Can set an option to Compact on Close
  - File | Options | Current Database | Compact on Close
- Compacting also
  - physically rearranges table records in primary key sequence
  - rebuilds indexes
  - recompiles each query's execution plan