#### Northeastern University College of Computer and Information Science

Managing Data in Relational Databases

#### DATA, DATABASES, AND QUERIES

# What is this About

- Storing, accessing, searching, and viewing data are important in any business.
- While spreadsheets work well for small amounts of data, databases are used for larger data collections.
- Learning how to access data is an important skill when working with databases.
- We will learn how to formulate queries in the Microsoft Access database system.

2

#### Reasons to Move to a Database

- Too much data in individual files
   Difficult to manage data
- Have multiple uses for data
   Need multiple "views" of the data
- Need to share the data
  - Numerous people are entering, deleting, viewing data
- Need to control the data

   Control data values and consistency

# Value of Knowledge

- What we learn here is applicable to many other database that are used by businesses:
  - Oracle
  - Sybase
  - Microsoft SQL Server
  - JavaDB

. . .

• We will also learn how to decompose problems and think logically.

#### About Microsoft Access

- Not available with Mac OS
- You can use myApps (remotely login to a virtual Northeastern session). (see the course website under Resources -> Software)

#### • DOWNLOAD AND SAVE!

- You must download and save Access files BEFORE starting to work on them or you will lose your work
- .laccdb files: file locking is controlled by a locking file with the file name extension .laccdb.
  - Do not submit this file! Close your file first and the .laccdb file goes away.

#### **Relational Databases**

 Microsoft Access is a *relational database* which means that it stores data in tables

- Each table stores information about a single subject

- Each table contains rows; one row for each record, *i.e.*, a contact, order, product, etc.
- Each column (or field) contains different kinds of information about the subject
- Each row in a table has a unique identifier (or key), *e.g.*, OrderID, ProductID, ContactID, etc.

#### **Relational Databases**

- Each table in the database contains information related to a single subject and only that subject.
- You can manipulate data about two classes of information (such as customers and orders) based on related data values
- Example: it would be redundant to store all customer information with every order.
  - In a relational DB, the table for orders contains one field that stores data such as a customer ID which can be used to connect each order with the appropriate customer information.

7

#### **Relational Databases**

- Microsoft Access is a *relational database* which means that it stores data in **tables**.
- Tables contains **records**; one row for each record, *e.g.*, a contact, order, product, etc.
- Tables have **attributes**; each record has a value for every attribute, *e.g.*, name, price, address...
- Each row in a table has a unique identifier attribute called a key, e.g., OrderID, ProductID, ContactID, etc.

8

Microsoft Access Tutorial: Data, Databases, and Queries

#### LAYOUT OF THE ORDERS DATABASE

## The Orders Database

- We will be using a database that contains data for orders placed by customers for our examples.
- This database stores the following information:
  - For each order, we know what was ordered, how many of that item was ordered, and at what price.
  - For each order, we know who placed that order.
  - For each customer (called a contact), we store where he/she lives.
  - For each product, we track its description and price.

#### The Database Layout

• These are all of the tables in the database:



# The Database Layout

For each product, we know its description and price.

For each order, we know what was ordered, how many of that item were ordered, and at what price.



#### A Sample Order

#### Order

#### 00001

#### **Customer Contact**

Contact ID:	C0004
Name:	Colon, Nicholas
Address:	9020 N.W. 75 Street
	Coral Springs, FL 33065

Order Date: 4/15/1999

Product ID	Product Name	Quantity	Uni	itPrice	Ε	ExtendedPrice
P0013	DVD Disks	1	\$	23.00	\$	23.00
P0014	HD Floppy Disks	4	\$	9.99	\$	39.96
P0027	Norton Anti-Virus	1	\$	115.95	\$	115.95

**Order Total:** *\$* 178.91

# Where Does The Data Come From?

Order	<b>00001</b>	ders.Order	ID			
Customer Contact	Contacts					
Contact ID:	C0004					
Name:	Colon, Nicholas					
Address:	9020 N.W. 75 Street	TipCodoc				
	Coral Springs, FL 33065	Zipcoaes		Exter	dedPr	ice =
				Quan	tity * (	UnitPrice
Order Date:	4/15/1999 ← Orders.C	DrderDate				
		<b>a</b>			V	
Product ID	Product Name	Quantity	Un	itPrice	Extend	ledPrice
P0013	DVD Disks	1	\$	23.00	\$	23.00
P0014	HD Floppy Disks	4	\$	9.99	\$	39.96
P0027	Norton Anti-Virus	1	\$	115.95	\$	115.95
Lineltems					,	
			Or	der Total	<u>Ş</u>	178.91
		Total Ord	der	Amou	nt 🦯	<i>/</i> 1
CS1100	Microsoft Access - In	troduction		15	Northeast College of Co	ern University mputer and Information Science

# A closer look at the Contacts table

#### • The Design view

#### - Design your table in this view:

	Field Name	Data Type	Description (Optional)	٠
₽Þ	ContactID	AutoNumber		
	FirstName	Short Text		
	LastName	Short Text		
	Address	Short Text		
	ZipCode	Short Text		
	PhoneNumber	Short Text		
				Ŧ

**Field Properties** 

General Lookup		
Field Size	Long Integer	
New Values	Increment	
Format		
Caption		
Indexed	Yes (No Duplicates)	
Text Align	General	

The field description is optional. It helps you describe the field and is also displayed in the status bar when you select this field on a form. Press F1 for help on descriptions.

# Field Data Types

- Short Text alphanumeric data up to 255 characters
- Long Text– alphanumeric data up to 1 gigabyte
- Number numeric data
- Date/Time dates and times
- Currency monetary data
- AutoNumber unique value generated by Access for each new record (not editable)

# A closer look at the Contacts table

- The Datasheet view
  - Enter new data in this view

	Contacts									
2		ContactID 👻	FirstName 🕞	LastName 👻	Address 👻	ZipCode 🕞	PhoneNumt +	Clic		
	÷	1	Peter	Levoy	17 Halpern Ave	02536	508 904 6588			
	÷	2	Jane	Wu	68 Trust Cir	09056	910 332 4654			
	+	3	Jim	Wobek	5 Main Ln	02536				
	+	4	Student	Name	100 Main St	09056				
*		(New)								

#### Relationships

- From the **Database Tools** tab, define how the data in tables is related, such as ID fields in tables that should match.
  - Example: ContactID in the Contact table is related to ContactID in the Orders table – a one-to-many relationship.



Microsoft Access Tutorial: Data, Databases, and Queries

#### **RETRIEVING DATA WITH QUERIES**

# Queries

- Data is retrieved through queries.
- Queries are formulated in a specialized language called SQL (pronounced *SEQUEL*).
- Microsoft Access makes it easy to create SQL queries through a simple drag-and-drop interface called the *Query Builder*.
- Queries are eventually integrated into reports, forms, programs, or executed by themselves.

#### **Queries on Tables**

- Queries retrieve data from one or more tables.
- You can specify which
  - rows to include in the result through filters (WHERE clause in SQL terminology)
  - columns to include in the results
- The result of a query is a table that can be used in other queries (as subqueries).

# **Creating Queries in Access**

- To create a query:
  - Select the Create tab
  - Pick Query Design



 Select the tables to include in the query or simply close the dialog and drag the needed tables into the query designer

# Running a Query

• To run a query, click on:



• To return to the query design, click on:



# Example: Simple Query

• Find the contact id, first name, and last name for each contact.

Contacts * ContactID FirstName LastName Address ZipCode PhoneNum	ber	
ContactID	FirstName	LastName
Contacts	Contacts	Contacts
	V	
	Contacts * ContactID FirstName Address ZipCode PhoneNum ContactID Contacts	Contacts         *         ContactID         FirstName         LastName         Address         ZipCode         PhoneNumber

$\angle$	ContactID	•	FirstName 👻	LastName 🕞	
		1	Benjamin	Lee	
		2	Eleanor	Milgrom	
		3	Neil	Goodman	
		4	Nicholas	Colon	
		5	Michael	Ware	
		6	Jeffrey	Muddell	
		7	Ashley	Geoghegan	
		8	Serena	Sherard	
		9	Luis	Couto	
		10	Dorok	Andorson	

# **Removing Duplicates**

- Queries often result in duplicate rows.
- These are removed by "grouping rows" with the same value as a single row.
- To do a Group By, follow these steps:
  - Select the function button  $\sum_{\text{Totals}}$  in the ribbon
  - Select "Group By" for each field

Field:	ContactID	FirstName	LastName 🔍
Table:	Contacts	Contacts	Contacts
Total:	Group By	Group By	Group By
Sort:			
Show:	<b>v</b>	<b>V</b>	<b>V</b>
Criteria:			
or:			

# Example: Group By

- Find the dates on which orders were placed.
- Here's the result *without* a Group By:



#### Example: Group By

• Here's the same query *with* a Group By:



Order Date 🔻	
2/4/1998	
3/15/1999	./
4/15/1999	
4/18/1999	
4/20/1999	
4/21/1999	
4/22/1999	
4/24/1999	
4/25/1999	

Note that the duplicate rows have been "collapsed" into groups and only the group is displayed

# **Duplicates with Group By**

Group By collapses all rows that contain the same data across all columns.

2	FirstName 📼	LastName 👻	ZipCode 🕞	Ord	erID 🔻
	Benjamin	Lee	45501		O0005
	Benjamin	Lee	45501		O0006
	Benjamin	Lee	45501		O0009
	Benjamin	Lee	45501		O0011
	Eleanor	Milgrom	33063		O0007
	Eleanor	Milgrom	33063		O0008
	Eleanor	Milgrom	33063		O0010
	Eleanor	Milgrom	33063		O0036
	Neil	Goodman	33065		O0002
	Nicholas	Colon	33065		O0001
	Nicholas	Colon	33065		O0013

 OrderIDs are not the same in this example so names will show up more than once even if using Group By.

# **Eliminating Duplicates**

• Duplicates can be eliminated by specifying that the query should only return unique records.



# Joining Tables

- A "join" is a query operation in which rows are selected that have a common value for some row.
- To find contacts that actually placed an order<sup>1</sup>, the ContactID column value must be the same in the Contacts and the Orders table.
- This essentially finds all contacts who placed at least one order.

<sup>1</sup>There may be contacts in the Contacts table that are not linked to any order, *i.e.*, they never placed an order.

Microsoft Access - Introduction

# Example: Group By and Join

- Find the first name, last name, and zip code of all contacts that placed an order.
- Here's the result *without* a Group By:



FirstName 🔻	LastName 🔹	ZipCode 👻
Benjamin	Lee	45501
Eleanor	Milgrom	33063
Neil	Goodman	33065

Note the duplicate rows

# Example: Group By and Join

- Find the first name, last name, and zip code of all contacts that placed an order.
- Here's the result *with* a Group By:



2	FirstName 👻	LastName 👻	ZipCode 🕞
	Alan	Turing	33065
	Ashley	Geoghegan	33070
	Benjamin	Lee	45501
	Derek	Anderson	33120
	Eleanor	Milgrom	33063

All rows with the same first name, last name, and zip code have been collapsed into a single "group"

# Filtering

• Selecting rows that meet certain criteria is done through a WHERE clause.





Lists all of the line items (ID only) that have a Quantity > 2.

#### **Selection Criteria**

 Selection criteria are specified as an algebraic relationship, but queries are generally stated as a narrative, so we need to "translate".

Narrative	Algebraic Term
At least X	>= X
No more than X	< X
More than X	> X
No less than X	>= X
Less than X	< X
Up to X	< X
At most X	<= X