



CIS 412

DATABASE MANAGEMENT SYSTEMS

Chapter 2

pp. 44 - Functions to End of Chapter

FUNCTIONS (1/3)

- **Built-in functions**
 - Called **aggregate functions** in Access
- | | |
|------------------------|------------------------------|
| ○ Count | ○ StDev (standard deviation) |
| ○ Sum | ○ Var (variance) |
| ○ Avg (average) | ○ First |
| ○ Max (largest value) | ○ Last |
| ○ Min (smallest value) | |

FUNCTIONS (2/3)

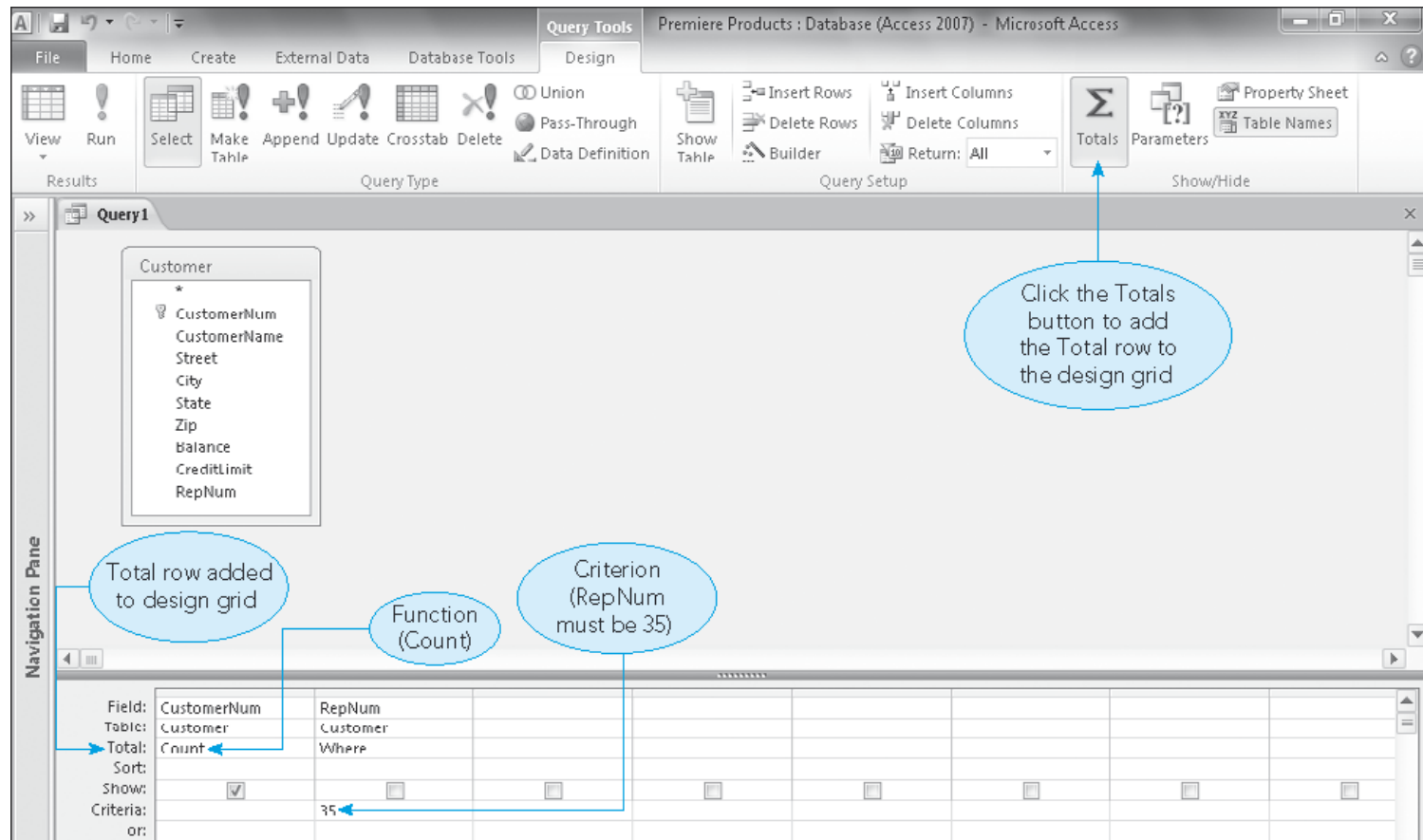


FIGURE 2-17: Query to count records

FUNCTIONS (3/3)

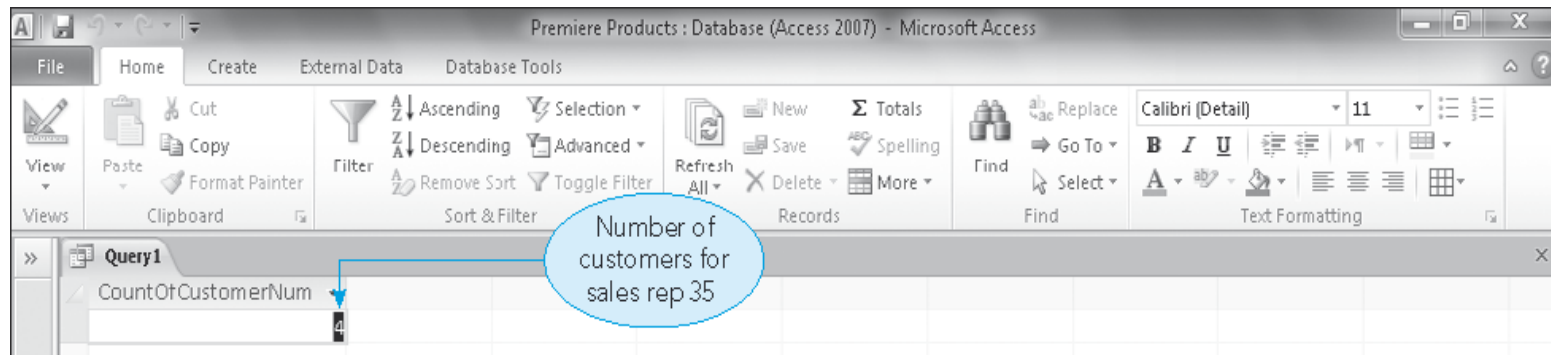


FIGURE 2-18: Query results

GROUPING (1/2)

- **Grouping:** creating groups of records that share some common characteristic
- To group records in Access:
 - Select Group By operator in the Total row for the field on which to group

GROUPING (2/2)

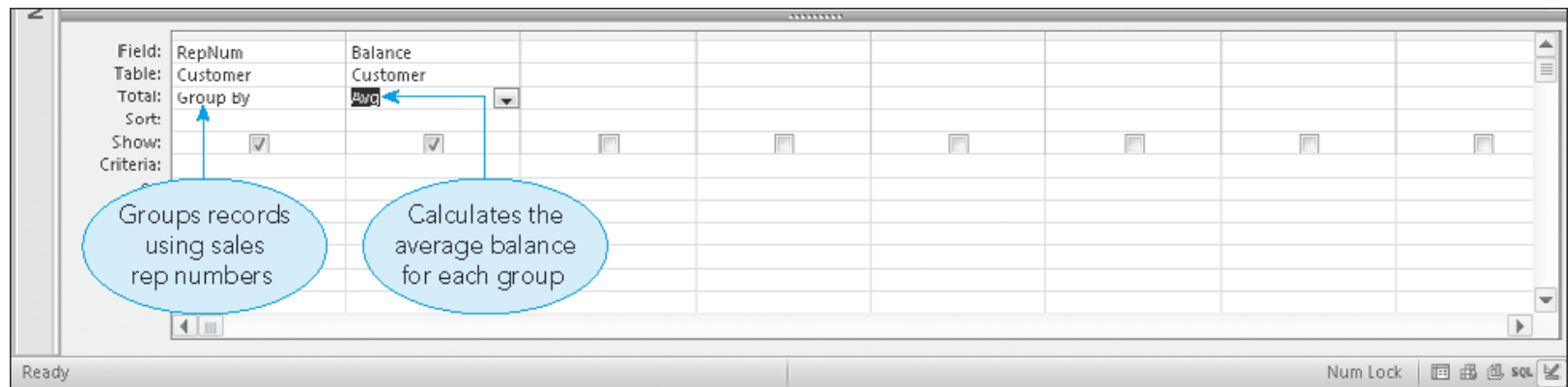


FIGURE 2-21: Query to group records

SORTING (1/2)

- **Sorting:** listing records in query results in an ordered way
- **Sort key:** field on which records are sorted
- **Major sort key**
 - Also called the **primary sort key**
 - First sort field, when sorting records by more than one field
- **Minor sort key**
 - Also called the **secondary sort key**
 - Second sort field, when sorting records by more than one field

SORTING (2/2)

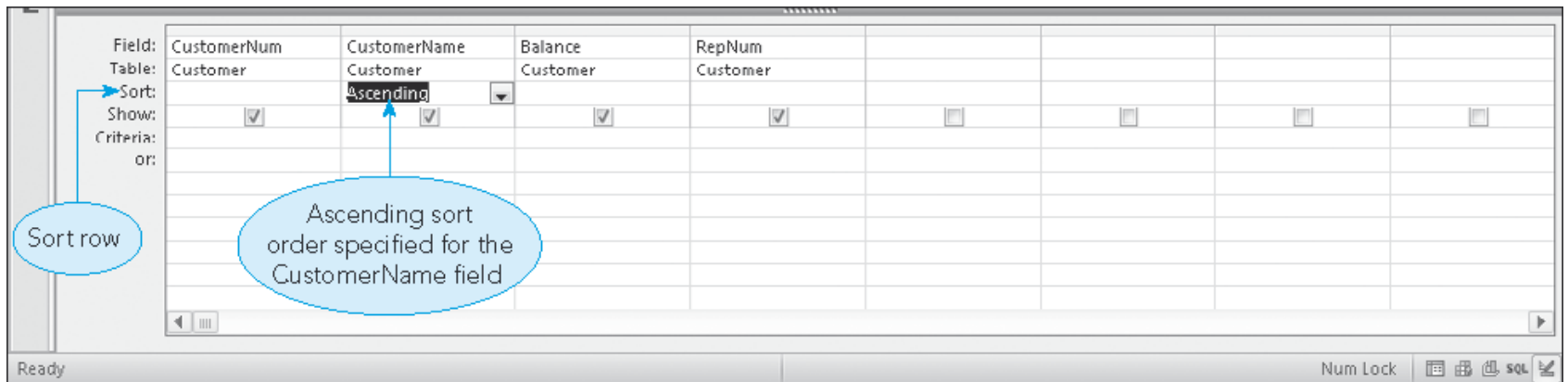


FIGURE 2-23: Query to sort records

SORTING ON MULTIPLE KEYS (1/2)

- Specifying more than one sort key in a query
- Major (primary) sort key
 - Sort key on the left in the design grid
- Minor (secondary) sort key
 - Sort key on the right in the design grid

SORTING ON MULTIPLE KEYS (2/2)

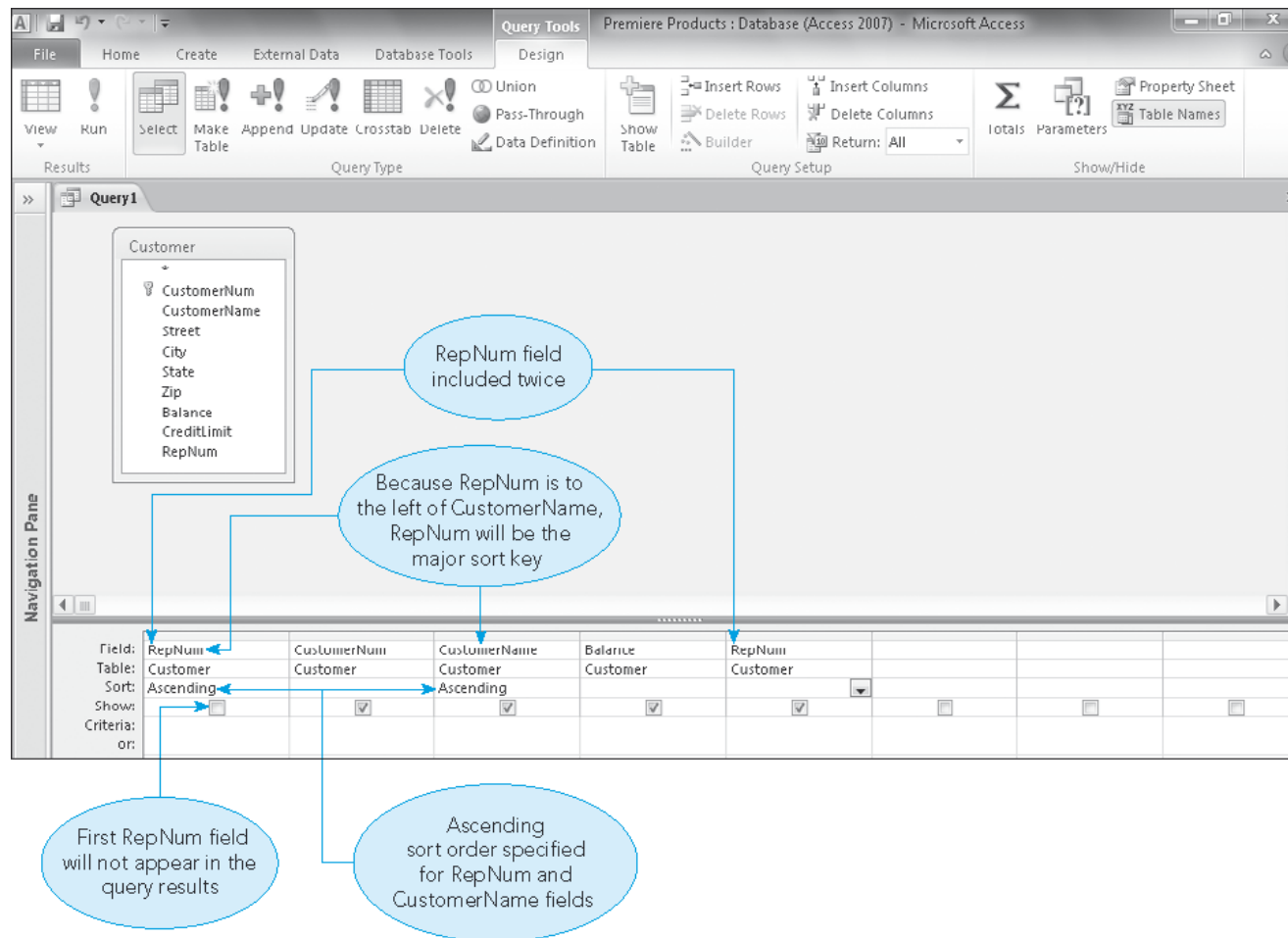


FIGURE 2-27: Correct query design to sort by RepNum and then by CustomerName

JOINING TABLES (1/2)

- Queries to select data from more than one table
- **Join** the tables based on matching fields in corresponding columns
- **Join line**
 - Line drawn by Access between matching fields in the two tables
 - Indicates that the tables are related

JOINING TABLES (2/2)

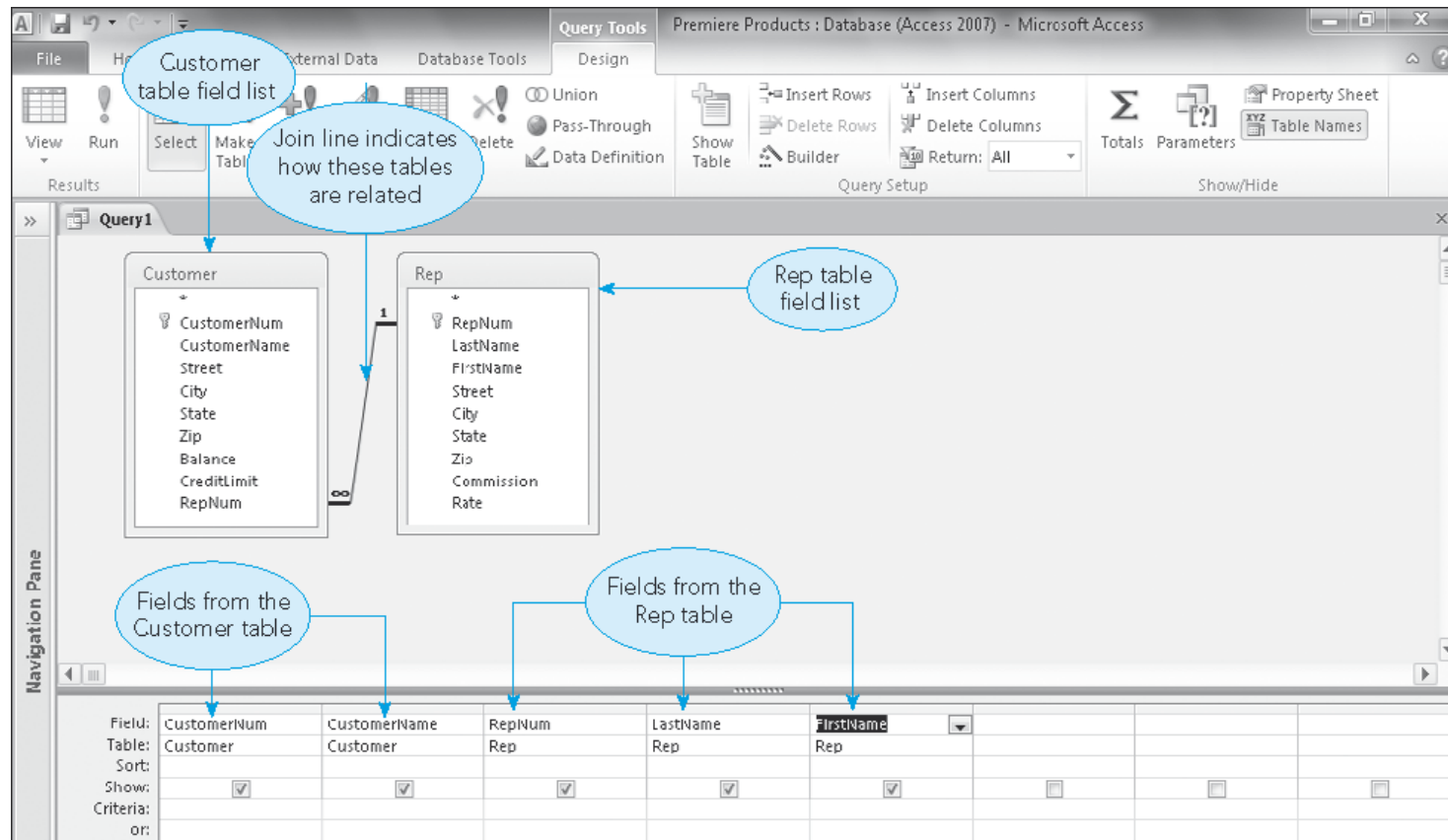


FIGURE 2-29: Query design to join two tables

JOINING MULTIPLE TABLES

- Joining three or more tables is similar to joining two tables
- To join three or more tables:
 - Add the field lists for all tables in the join to upper pane
 - Add the fields to appear in query results to design grid in the desired order

USING AN UPDATE QUERY (1/2)

- **Update query:** a query that changes data
 - Makes a specified change to all records satisfying the criteria in the query
- To change a query to an update query:
 - Click Update button in the Query Type group on the Query Tools Design tab
- Update To row is added when an update query is created
 - Used to indicate how to update data selected by the query

USING AN UPDATE QUERY (2/2)

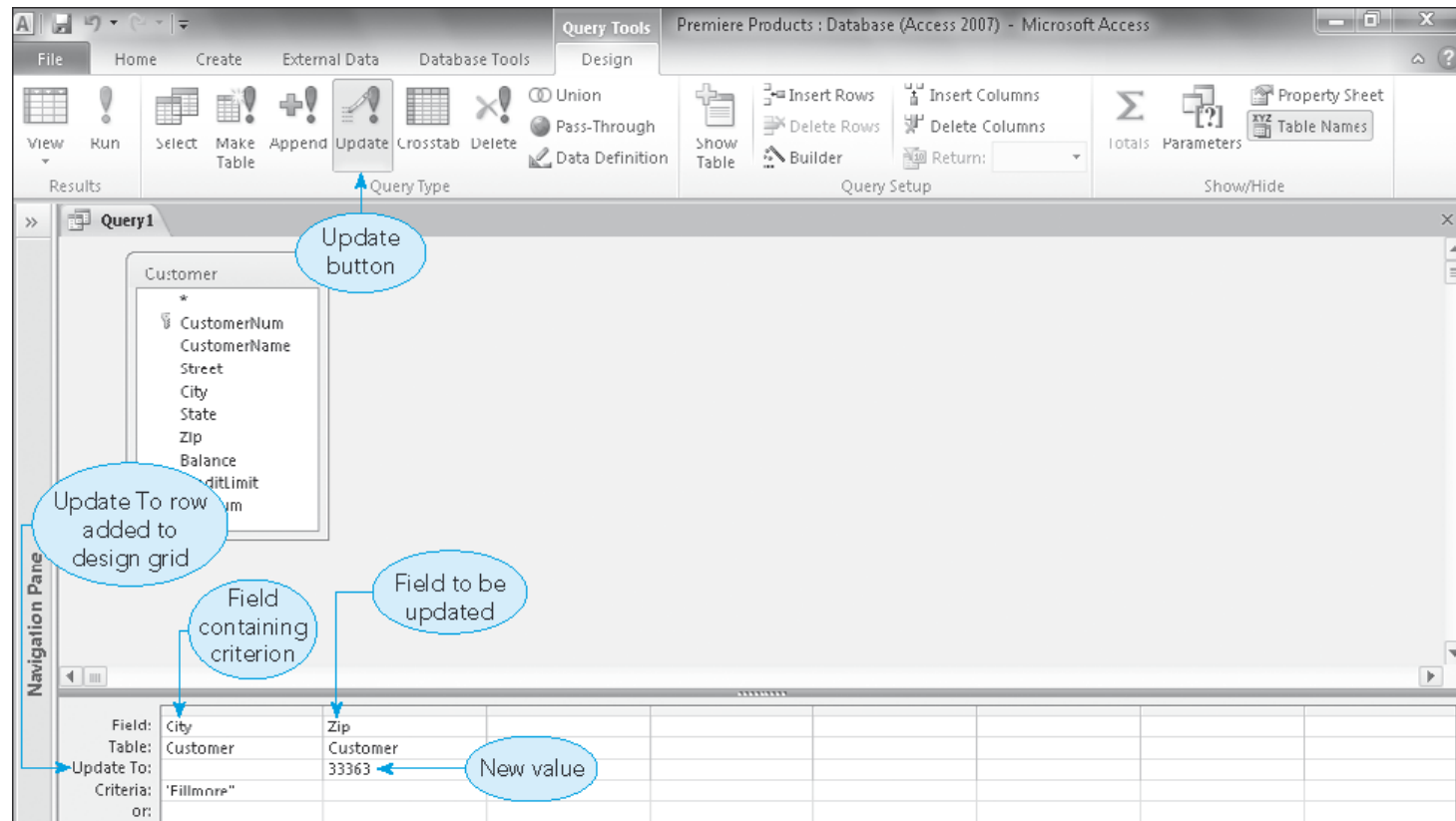


FIGURE 2-35: Query design to update data

USING A DELETE QUERY (1/2)

- **Delete query:** permanently deletes all records satisfying the criteria entered in the query
- To change query type to a delete query:
 - Click Delete button in the Query Type group on the Query Tools Design tab
- Delete row is added
 - Indicates this is a delete query

USING A DELETE QUERY (2/2)

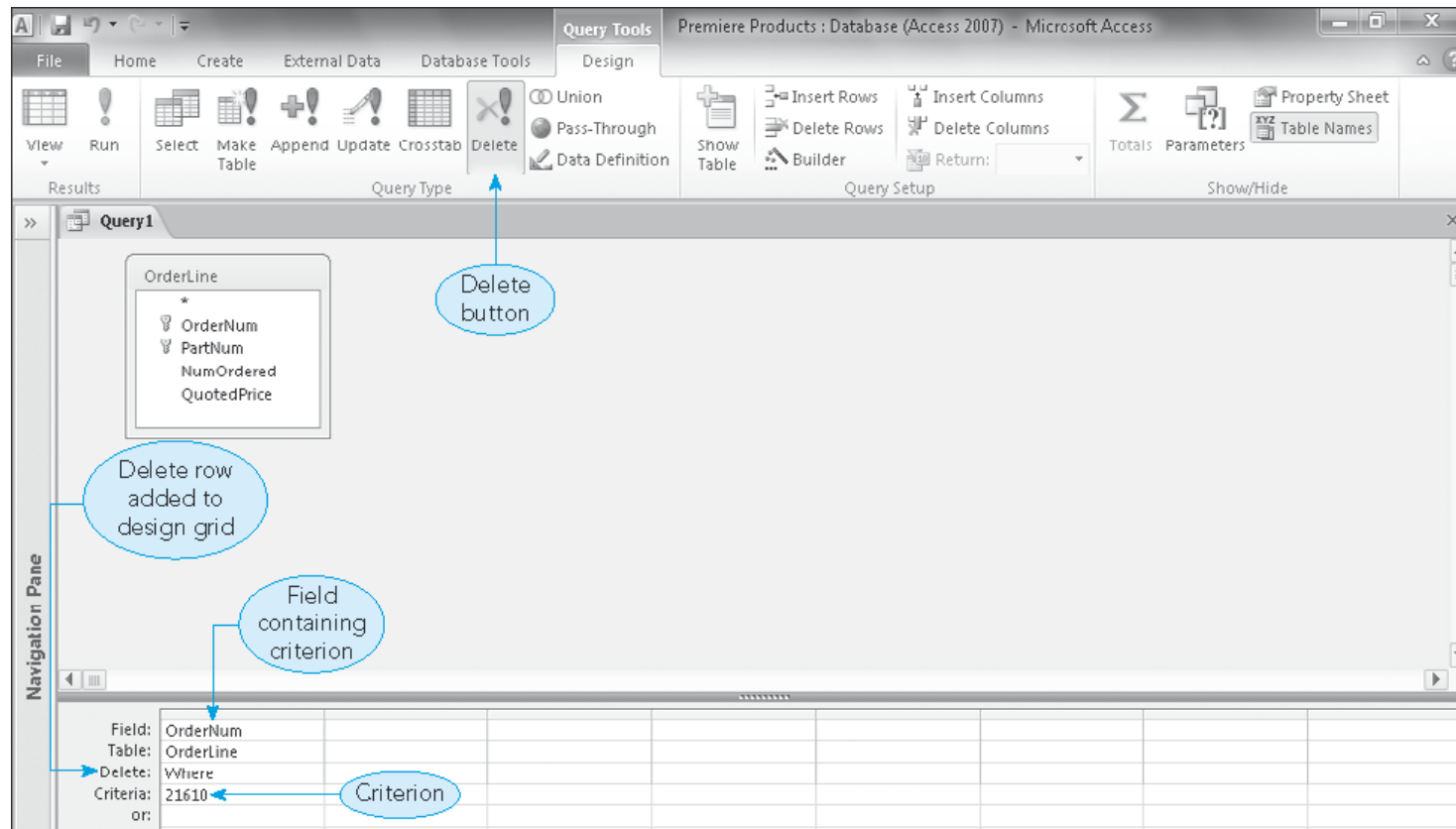


FIGURE 2-36: Query design to delete records

USING A MAKE-TABLE QUERY (1/2)

- **Make-table query:** creates a new table using results of a query
- Records added to new table are separate from the original table
- To change the query type to a make-table query:
 - Click Make Table button in the Query Type group on the Query Tools Design tab
 - In Make Table dialog box, enter the new table's name and choose where to create it

USING A MAKE-TABLE QUERY (2/2)

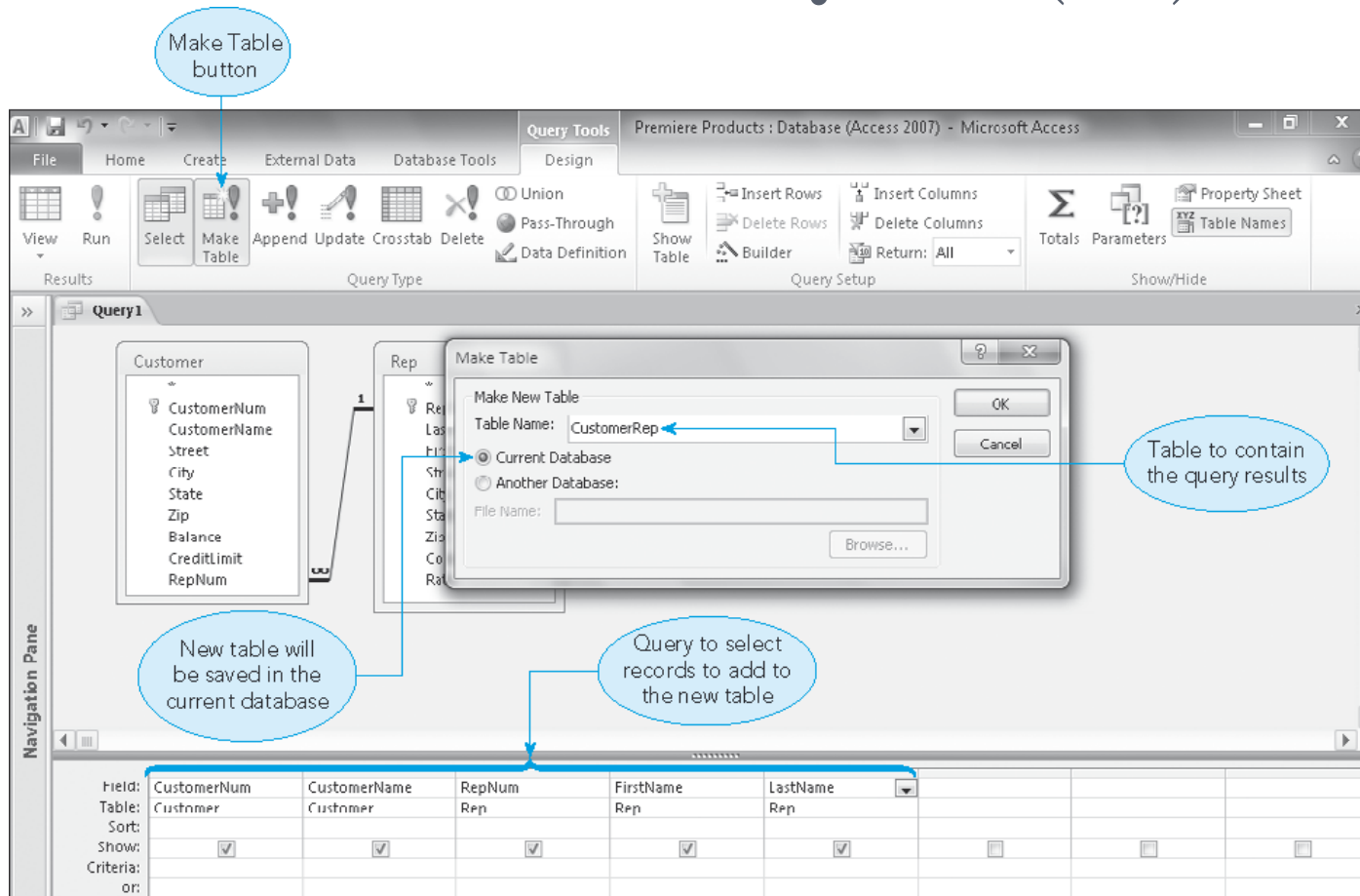


FIGURE 2-38: Make Table dialog box



CIS 372

DATABASE MANAGEMENT SYSTEMS

Chapter 3

The Relational Model 2: SQL

INTRODUCTION

- **SQL (Structured Query Language)**
 - Allows users to query a relational database
 - Must enter **commands** to obtain the desired results
 - Standard language for relational database manipulation

GETTING STARTED WITH MICROSOFT OFFICE ACCESS 2007 AND 2010 (1/2)

- If you are using the Access 2007 or 2010 version of the Premiere Products database provided with the Data Files for this text:
 - Tables in the database have already been created
 - You will not need to execute the CREATE TABLE commands to create the tables or the INSERT commands to add records to the tables

GETTING STARTED WITH MICROSOFT OFFICE ACCESS 2007 AND 2010 (2/2)

- To execute SQL commands shown in the figures in Access 2007 or Access 2010:
 - Open the Premiere Products database
 - Click the Create tab on the Ribbon
 - Click the Query Design button in the Other group
 - Click the Close button in the Show Table dialog box
 - Click the View button arrow in the Results group on the Query Design Tools tab, then click SQL View
 - The Query1 tab displays the query in SQL view, ready for you to type your SQL commands

TABLE CREATION – SQL SERVER AND MySQL

- **SQL CREATE TABLE** command
 - Creates a table by describing its layout
- Typical restrictions placed on table and column names by DBMS
 - Names cannot exceed 18 characters
 - Names must start with a letter
 - Names can contain only letters, numbers, and underscores (_)
 - Names cannot contain spaces

SIMPLE RETRIEVAL (1/3)

- **SELECT-FROM-WHERE:** SQL retrieval command
 - **SELECT clause:** lists fields to display
 - **FROM clause:** lists table or tables that contain data to display in query results
 - **WHERE clause** (optional): lists any conditions to be applied to the data to retrieve
- **Simple condition:** field name, a comparison operator, and either another field name or a value

SIMPLE RETRIEVAL (2/3)

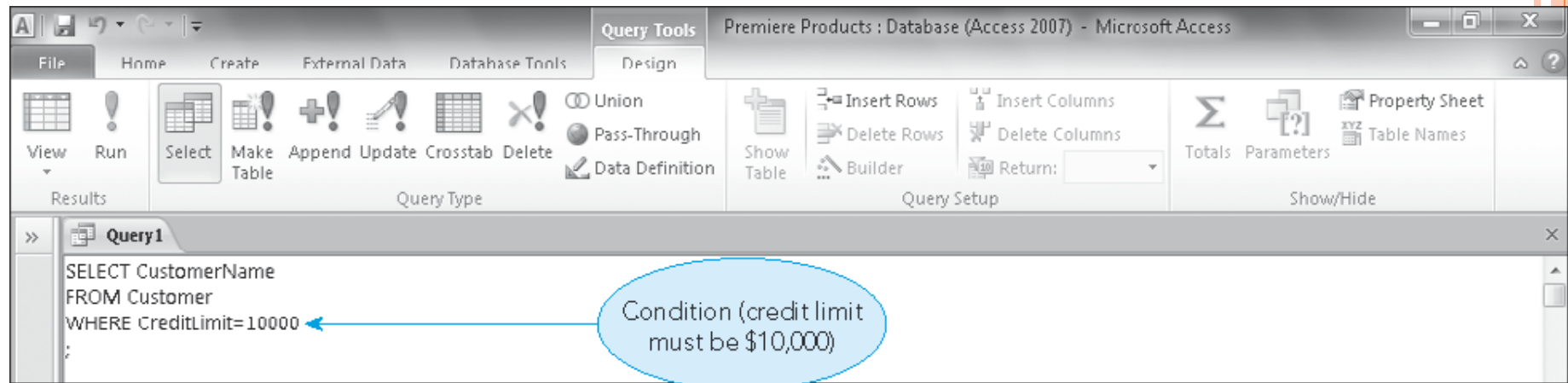


FIGURE 3-6: SQL query with WHERE condition

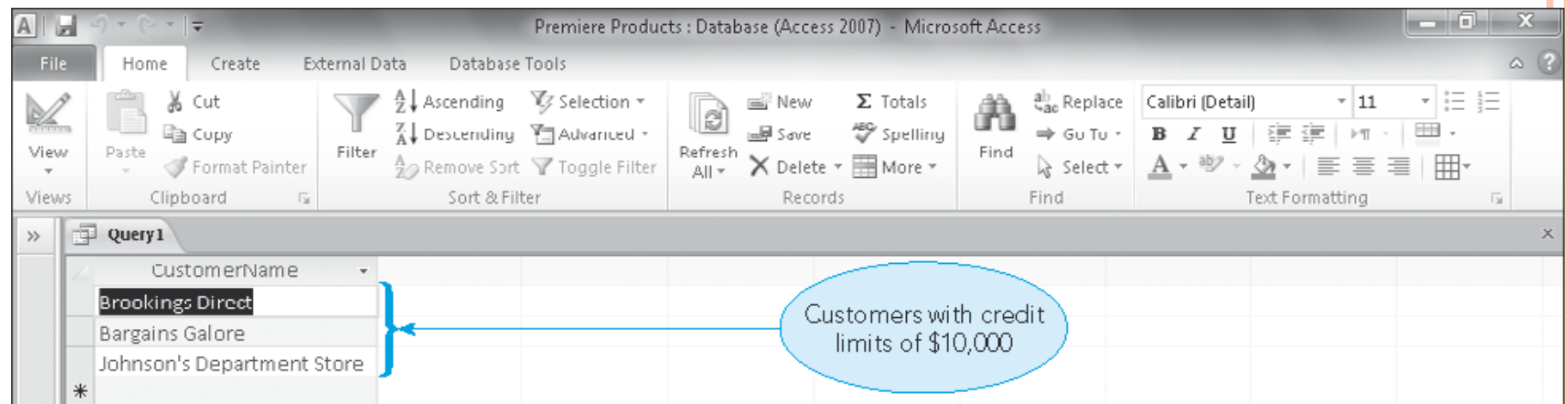


FIGURE 3-7: Query results

SIMPLE RETRIEVAL (3/3)

Comparison Operator	Meaning
=	Equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
< >	Not equal to (used by most implementations of SQL)
!=	Not equal to (used by some implementations of SQL)

FIGURE 3-8: Comparison operators used in SQL commands

COMPOUND CONDITIONS (1/5)

○ **Compound condition**

- Connecting two or more simple conditions using one or both of the following operators: AND and OR
- Preceding a single condition with the NOT operator

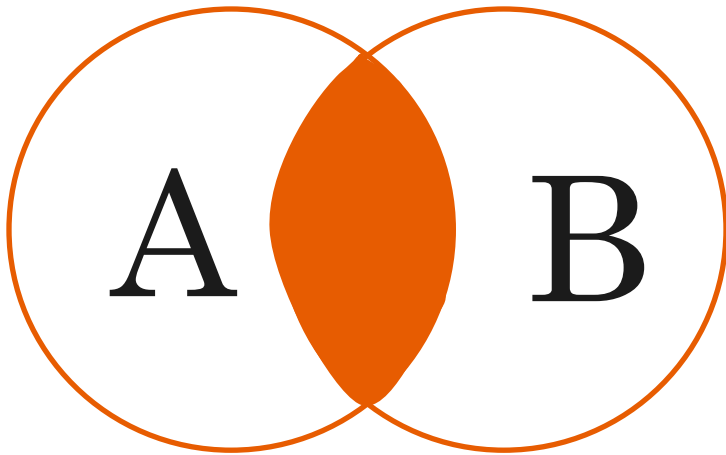
○ Connecting simple conditions using AND operator

- *All* of the simple conditions must be true for the compound condition to be true

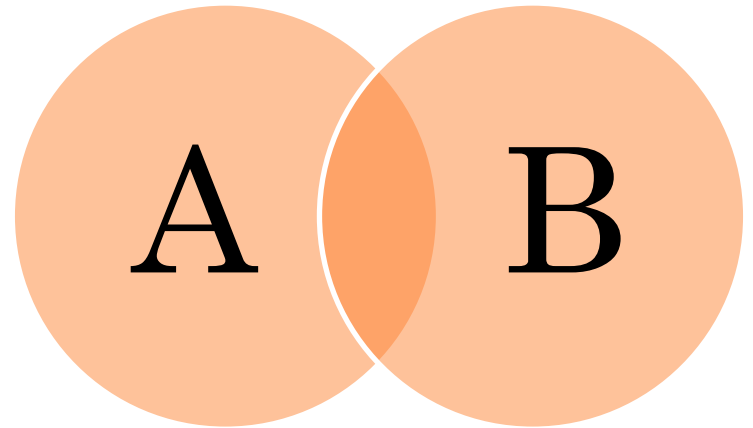
○ Connecting simple conditions using OR operator

- *Any* of the simple conditions must be true for the compound condition to be true

COMPOUND CONDITIONS (2/5)



A and B



A or B

COMPOUND CONDITIONS (3/5)

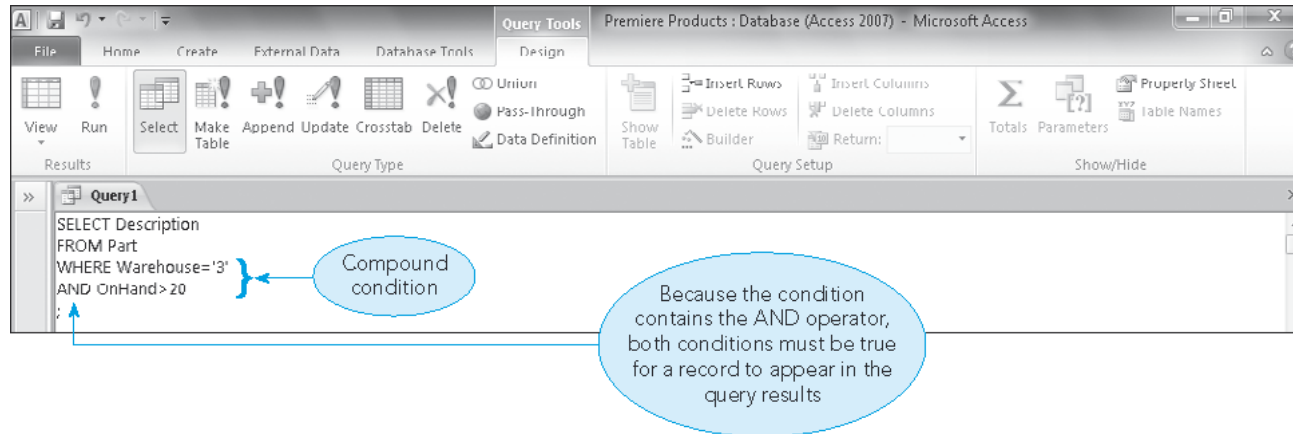


FIGURE 3-15: Compound condition that uses the AND operator

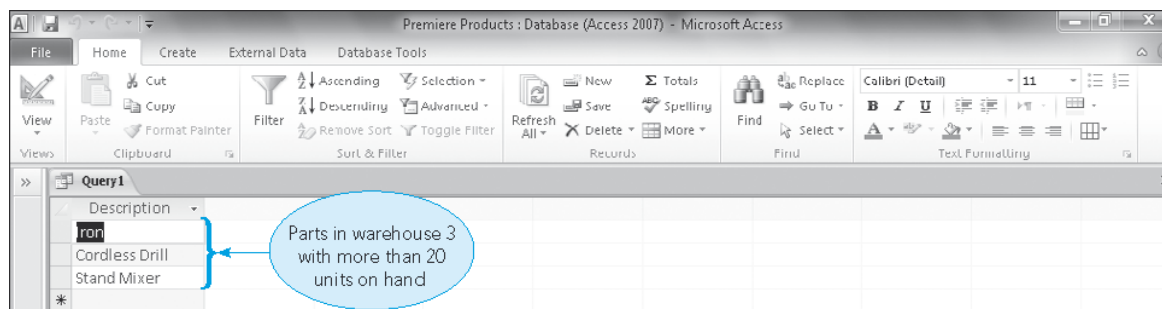


FIGURE 3-16: Query results

COMPOUND CONDITIONS (4/5)

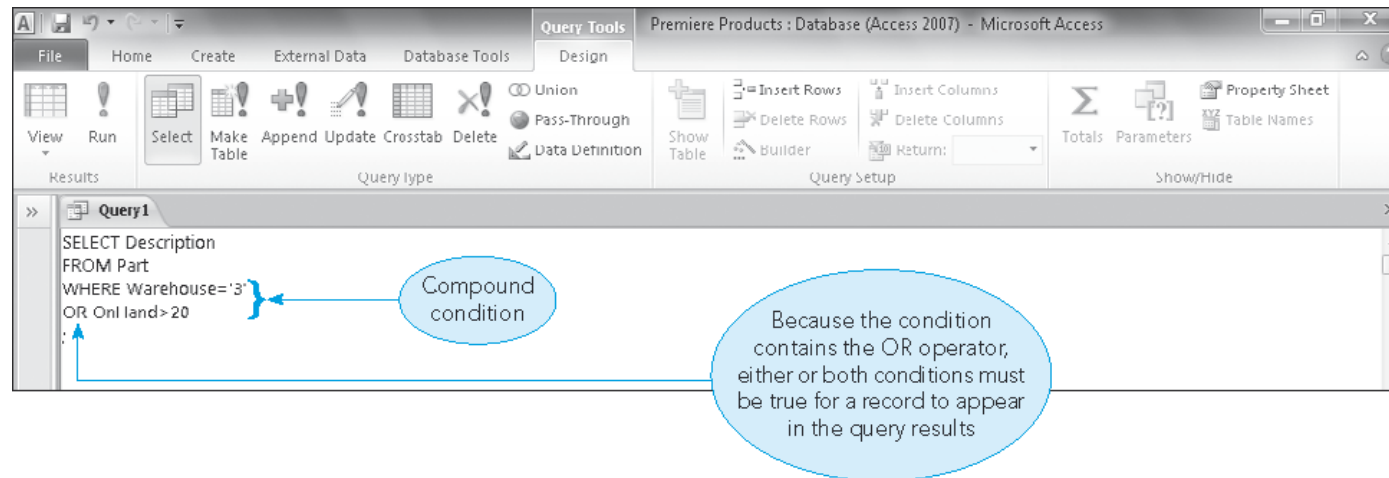


FIGURE 3-17: Compound condition that uses the OR operator

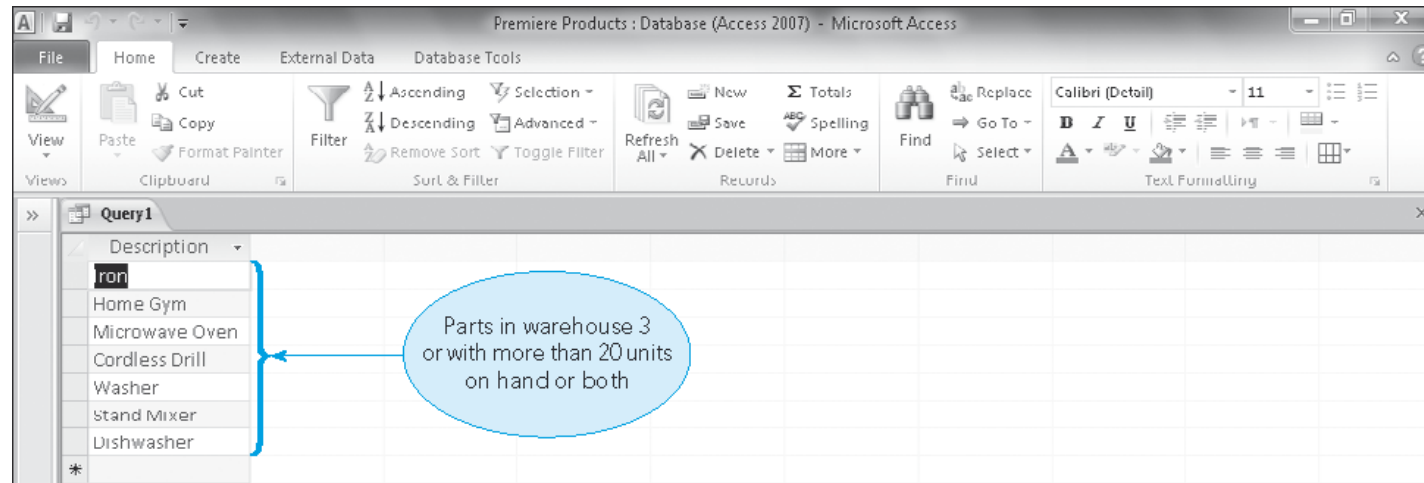


FIGURE 3-18: Query results

COMPOUND CONDITIONS (5/5)

- Preceding a condition by NOT operator
 - Reverses the truth or falsity of the original condition
- BETWEEN operator
 - Value must be between the listed numbers

COMPUTED FIELDS (1/2)

- Computed field or calculated field
 - Field whose values you derive from existing fields
 - Can involve:
 - Addition (+)
 - Subtraction (-)
 - Multiplication (*)
 - Division (/)

COMPUTED FIELDS (2/2)

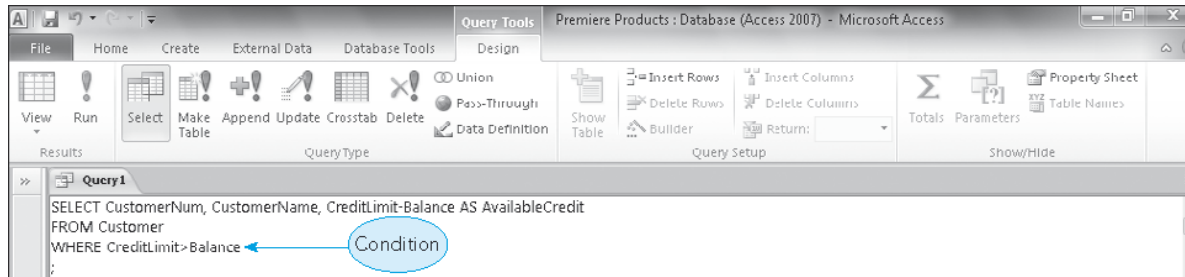


FIGURE 3-25: SQL query with a computed field and condition

Query1

CustomerNum	CustomerName	AvailableCredit
148	Al's Appliance and Sport	\$950.00
282	Brookings Direct	\$9,568.50
356	Ferguson's	\$1,715.00
462	Bargains Galore	\$6,588.00
524	Kline's	\$2,238.00
608	Johnson's Department Store	\$7,894.00
687	Lee's Sport and Appliance	\$2,149.00
725	Deerfield's Four Seasons	\$7,252.00

Only customers with credit limits that exceed their balances are listed

FIGURE 3-26: Query results

USING SPECIAL OPERATORS (LIKE AND IN) (1/2)

- Wildcards in Access SQL
 - Asterisk (*): collection of characters
 - Question mark (?): any individual character
- Wildcards in MySQL
 - Percent sign (%): any collection of characters
 - Underscore (_): any individual character
- To use a wildcard, include the LIKE operator in the WHERE clause
- IN operator provides a concise way of phrasing certain conditions

USING SPECIAL OPERATORS (LIKE AND IN) (CONTINUED)

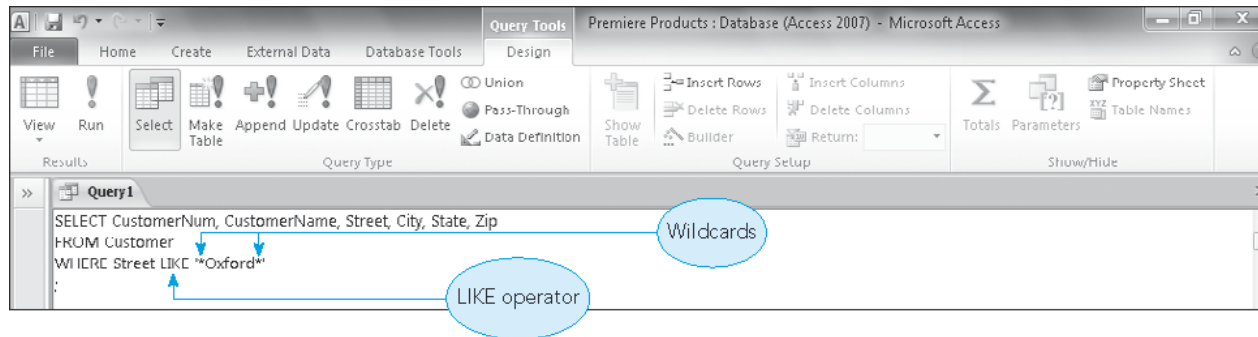


FIGURE 3-27: SQL query with a LIKE operator

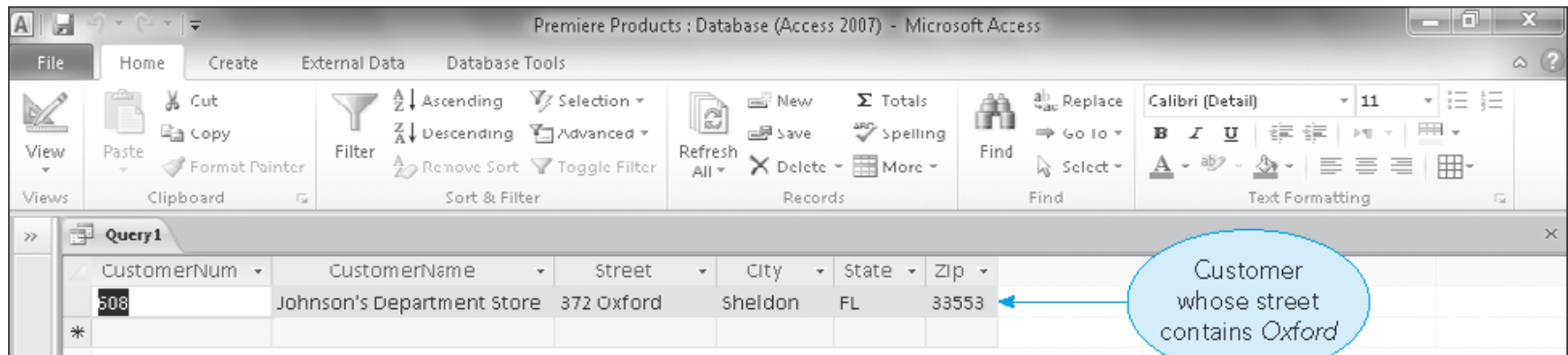


FIGURE 3-28: Query results

USING SPECIAL OPERATORS (LIKE AND IN)

(2/2)

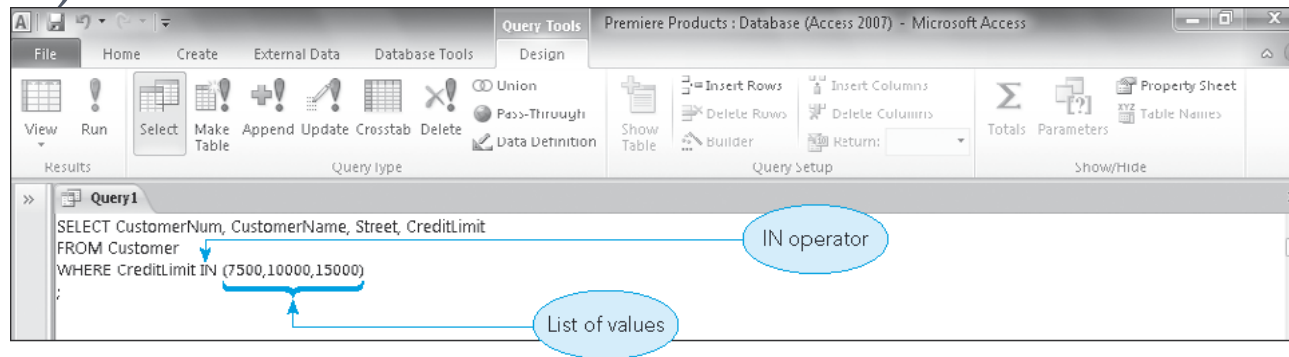


FIGURE 3-28: SQL query with an IN operator

Query1

CustomerNum	CustomerName	Street	CreditLimit
148	Al's Appliance and Sport	2837 Greenway	\$7,500.00
282	Brookings Direct	3827 Devon	\$10,000.00
356	Ferguson's	382 Wildwood	\$7,500.00
462	Bargains Galore	3829 Central	\$10,000.00
524	Kline's	838 Ridgeland	\$15,000.00
600	Johnson's Department Store	1/2 Oxford	\$10,000.00
725	Deerfield's Four Seasons	282 Columbia	\$7,500.00
842	All Season	28 Lakeview	\$7,500.00

Only customers with credit limits of \$7,500, \$10,000, or \$15,000 are listed

FIGURE 3-29: Query results

SORTING (1/2)

- Sort data using the **ORDER BY** clause
- Sort key: field on which to sort data
- When sorting data on two fields:
 - Major sort key (or primary sort key): more important sort key
 - Minor sort key (or secondary sort key): less important sort key

SORTING (2/2)

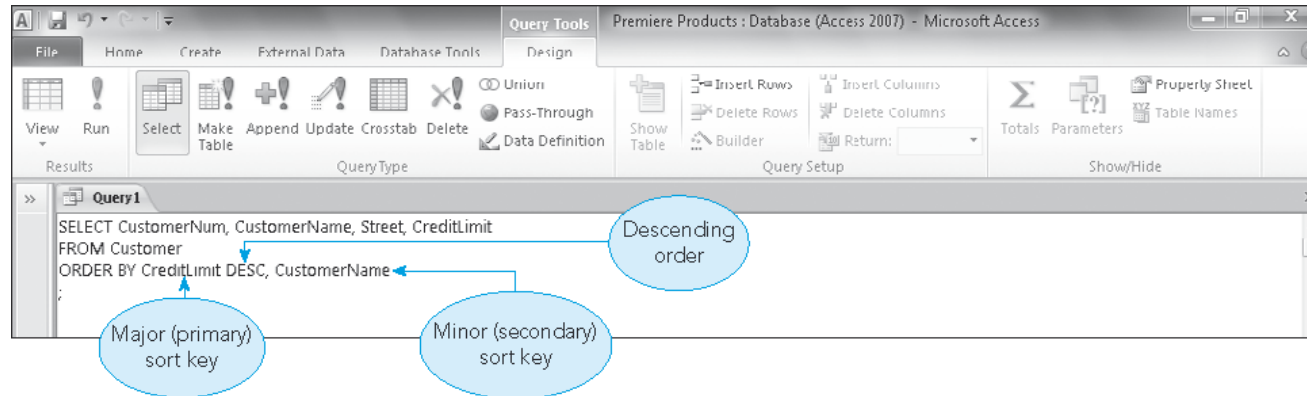


FIGURE 3-33: SQL query to sort data on multiple fields

Within credit limit, customers are sorted by name

Customers are sorted by credit limit in descending order

CustomerNum	CustomerName	Street	CreditLimit
524	Kline's	838 Ridgeland	\$15,000.00
462	Bargains Galore	3829 Central	\$10,000.00
282	Brookings Direct	3827 Devon	\$10,000.00
608	Johnson's Department Store	372 Oxford	\$10,000.00
842	All Season	28 Lakeview	\$7,500.00
148	Al's Appliance and Sport	2837 Greenway	\$7,500.00
725	Deertfield's Four Seasons	282 Columbia	\$7,500.00
356	Ferguson's	382 Wildwood	\$7,500.00
687	Lee's Sport and Appliance	282 Evergreen	\$5,000.00
408	The Everything Shop	1828 Raven	\$5,000.00

FIGURE 3-34: Query results

BUILT-IN FUNCTIONS (1/2)

- Built-in functions (aggregate functions) in SQL
 - COUNT: calculates number of entries
 - SUM or AVG: calculates sum or average of all entries in a given column
 - MAX or MIN: calculates largest or smallest values respectively

BUILT-IN FUNCTIONS (2/2)

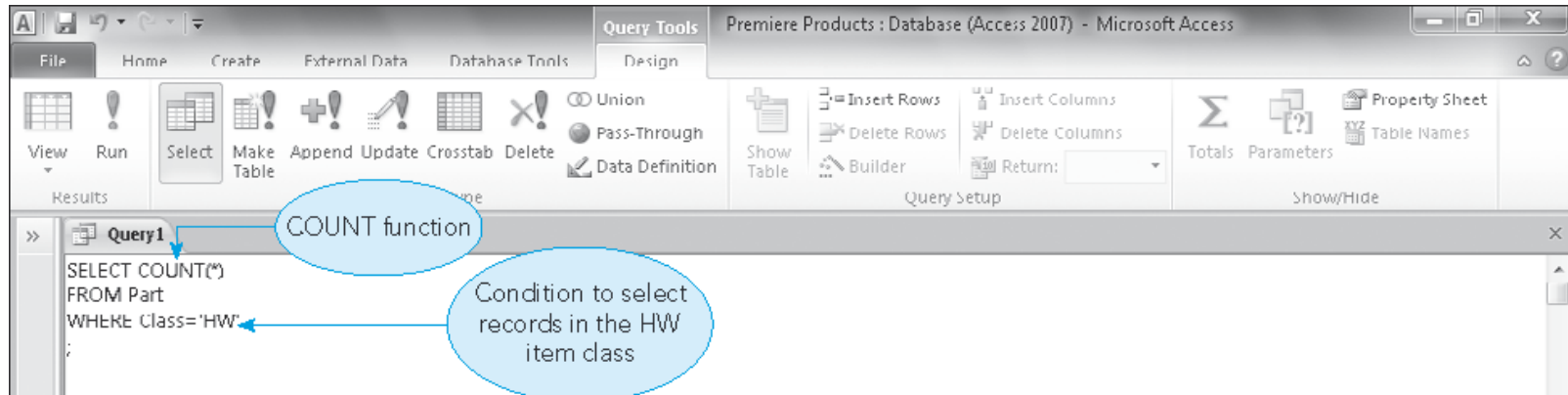


FIGURE 3-35: SQL query to count records

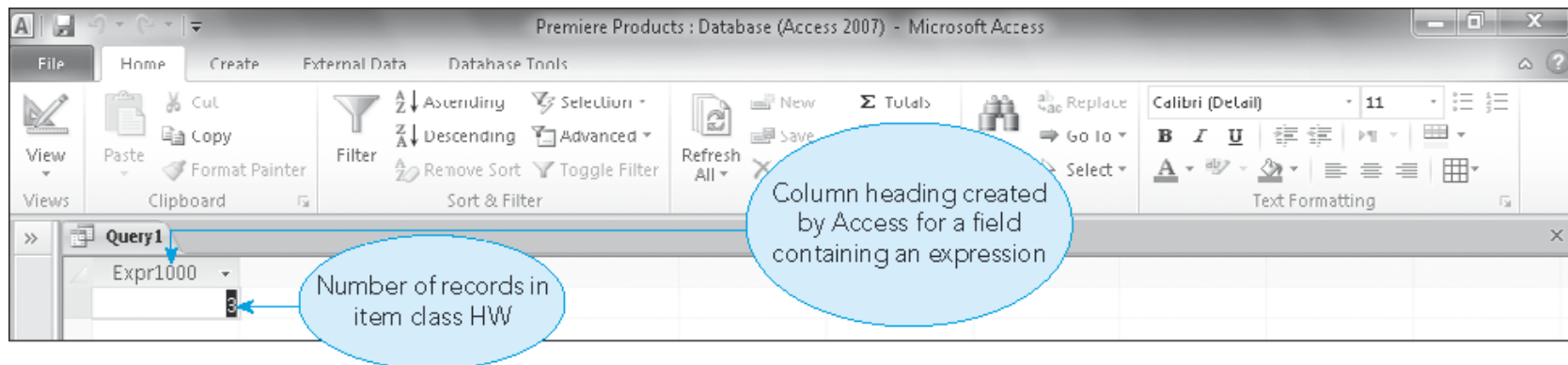


FIGURE 3-36: Query results

SUBQUERIES (1/2)

- **Subquery:** inner query
- Subquery is evaluated first
- Outer query is evaluated after the subquery

SUBQUERIES (2/2)

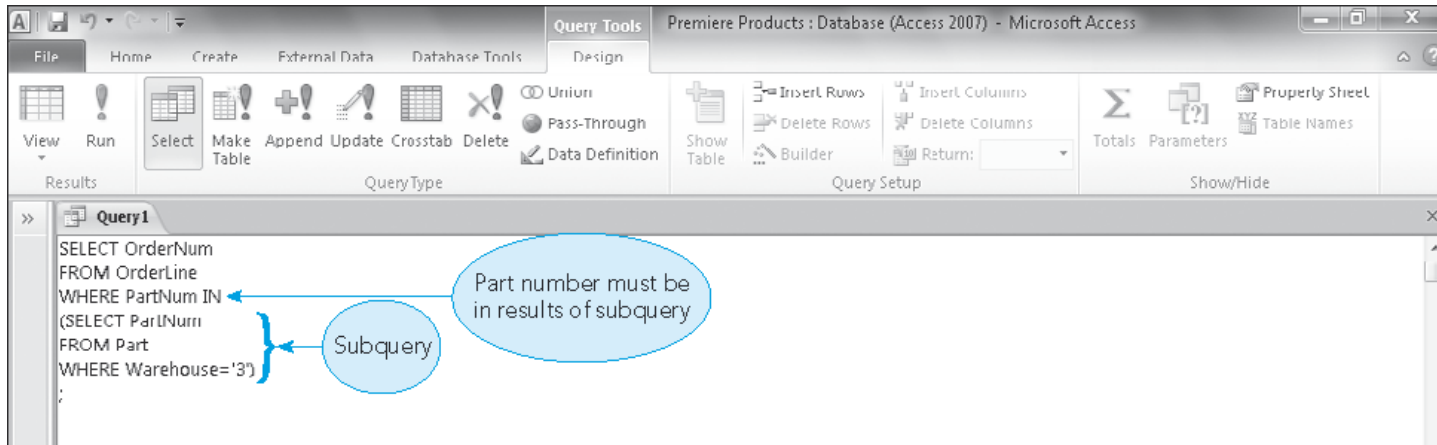


FIGURE 3-41: SQL query with a subquery

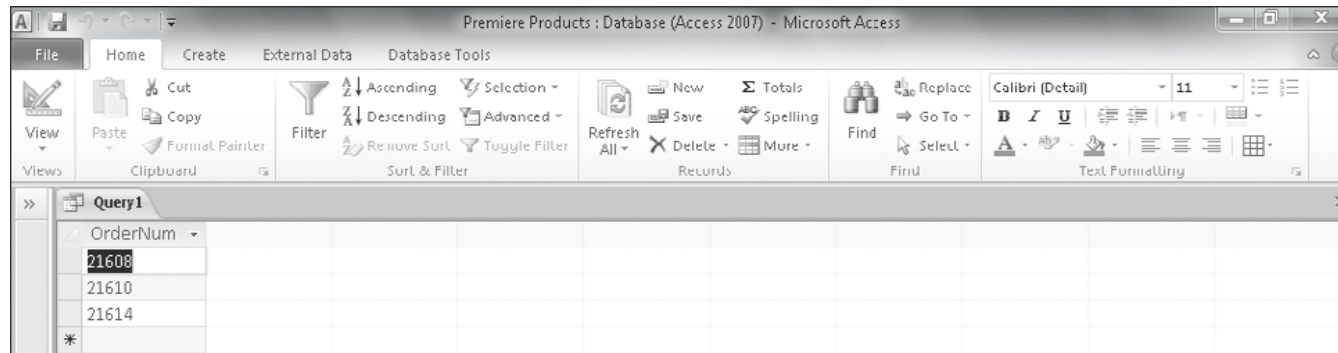


FIGURE 3-42: Query results

GROUPING (1/2)

- Create groups of records that share a common characteristic
- **GROUP BY clause** indicates grouping in SQL
- **HAVING clause** is to groups what the **WHERE** clause is to rows

GROUPING (2/2)

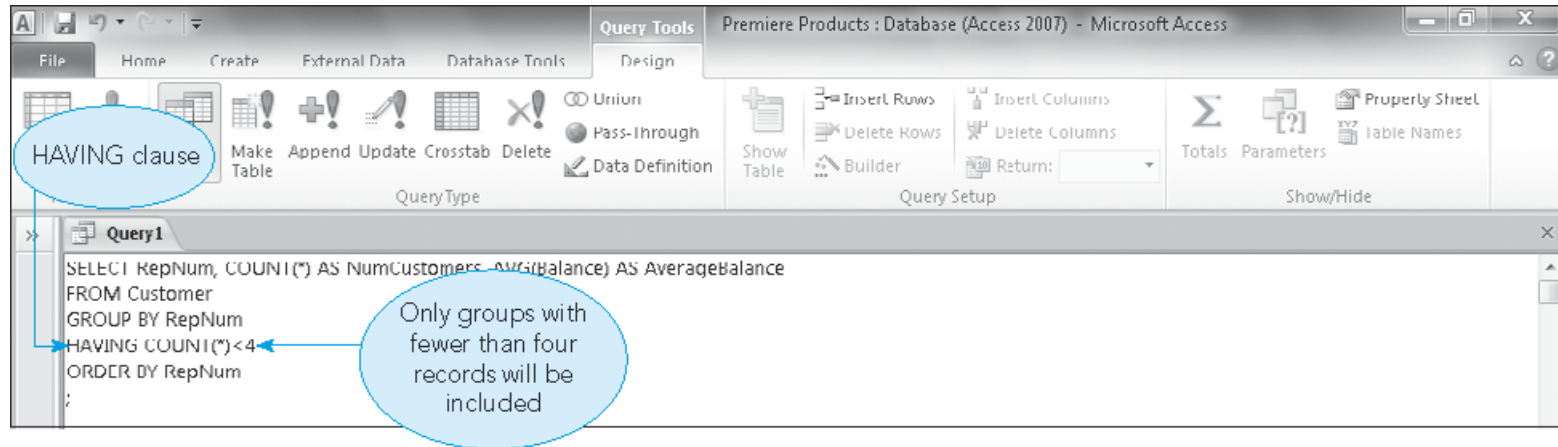


FIGURE 3-45: SQL query to restrict the groups that are included

Figure 3-46 shows the results of the query in the Datasheet view. The results are as follows:

RepNum	NumCustomers	AverageBalance
20	3	\$9,177.67
65	3	\$3,767.67

FIGURE 3-46: Query results

JOINING TABLES (1/3)

- Queries can locate data from more than one table
- Enter appropriate conditions in the WHERE clause
- To join tables, construct the SQL command as:
 1. SELECT clause: list all fields you want to display
 2. FROM clause: list all tables involved in the query
 3. WHERE clause: give the condition that will restrict the data to be retrieved to only those rows from the two tables that match

JOINING TABLES (2/3)

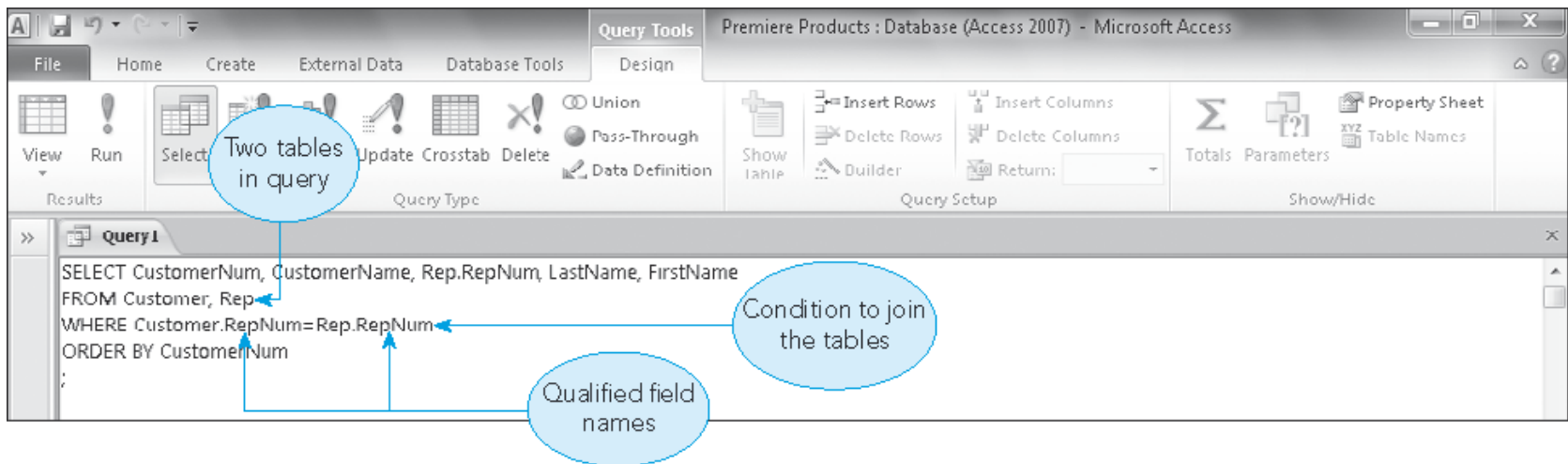


FIGURE 3-49: SQL query to join tables

JOINING TABLES (3/3)

The screenshot shows the Microsoft Access interface for a database named 'Premiere Products : Database (Access 2007)'. The 'Query1' table is displayed, showing the results of a join between the 'Customer' and 'Rep' tables. The table has five columns: 'CustomerNum', 'CustomerName', 'RepNum', 'LastName', and 'FirstName'. The records are sorted by 'CustomerNum' in ascending order. Annotations with blue arrows point to specific parts of the table: 'Fields from the Customer table' points to the 'CustomerNum' and 'CustomerName' columns; 'Fields from the Rep table' points to the 'RepNum', 'LastName', and 'FirstName' columns; and 'Records ordered by CustomerNum' points to the 'CustomerNum' column.

CustomerNum	CustomerName	RepNum	LastName	FirstName
148	Al's Appliance and Sport	20	Kaiser	Valerie
282	Brookings Direct	35	Hull	Richard
356	Ferguson's	65	Perez	Juan
408	The Everything Shop	35	Hull	Richard
462	Bargains Galore	65	Perez	Juan
524	Kline's	20	Kaiser	Valerie
608	Johnson's Department Store	65	Perez	Juan
687	Lee's Sport and Appliance	35	Hull	Richard
725	Deerfield's Four Seasons	35	Hull	Richard
842	All Season	20	Kaiser	Valerie

FIGURE 3-50: Query results

UNION (1/2)

- Union of two tables is a table containing all rows in the first table, the second table, or both tables
- Two tables involved must be union compatible
 - Same number of fields
 - Corresponding fields must have same data types

UNION (2/2)

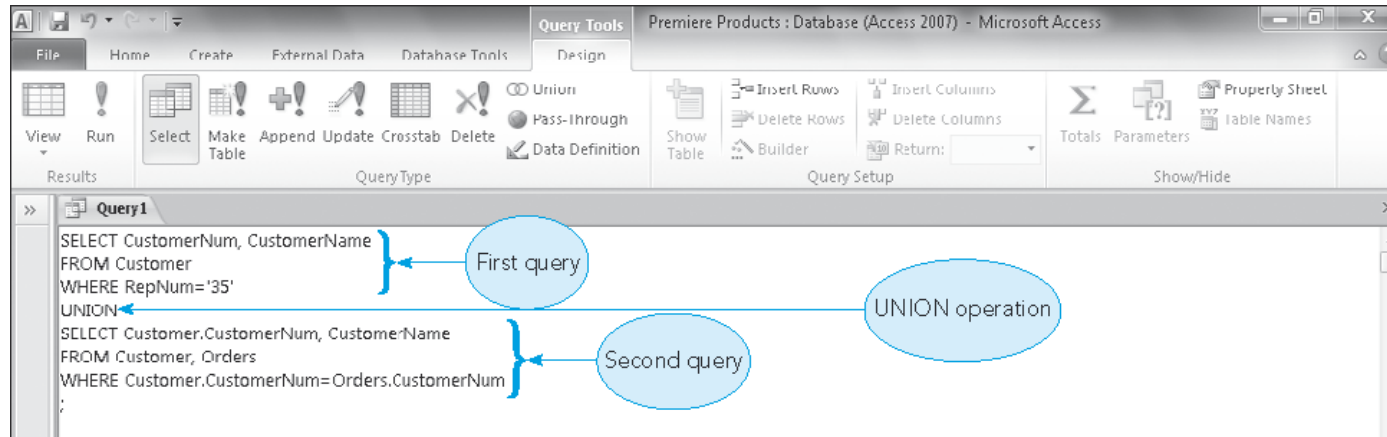


FIGURE 3-55: SQL query to perform a union

CustomerNum	CustomerName
148	Al's Appliance and Sport
282	Brookings Direct
356	Ferguson's
408	The Everything Shop
608	Johnson's Department Store
687	Lee's Sport and Appliance
725	Deerfield's Four Seasons

FIGURE 3-56: Query results

UPDATING TABLES (1/3)

- **UPDATE** command makes changes to existing data
- **INSERT** command adds new data to a table
- **DELETE** command deletes data from the database

UPDATING TABLES (2/3)

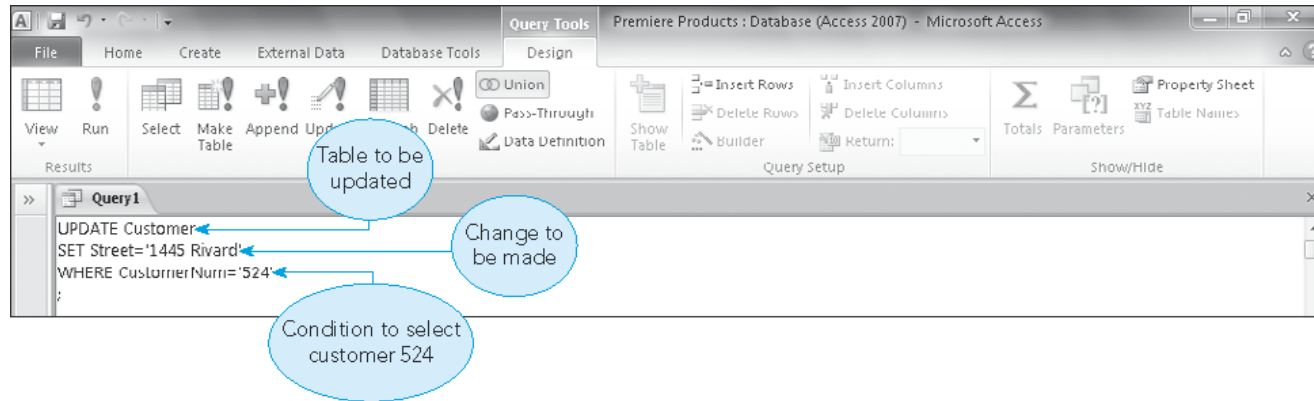


FIGURE 3-57: SQL query to update data

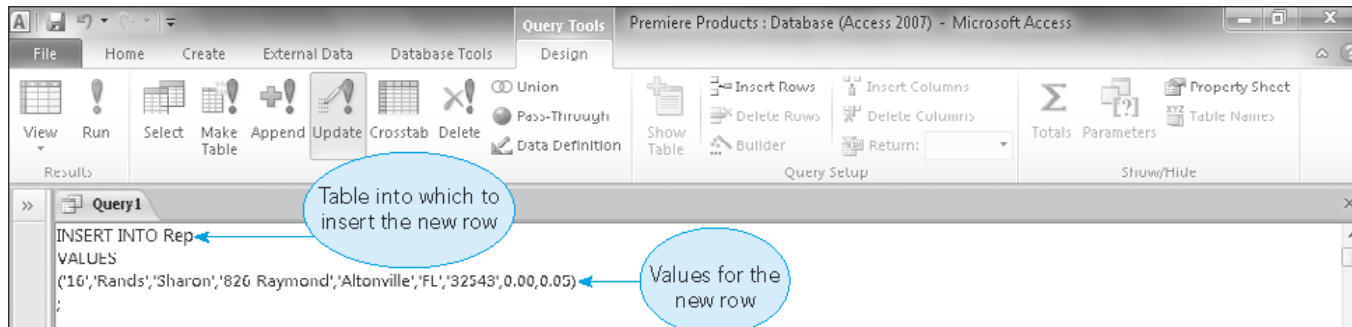


FIGURE 3-58: SQL query to insert a row

UPDATING TABLES (3/3)

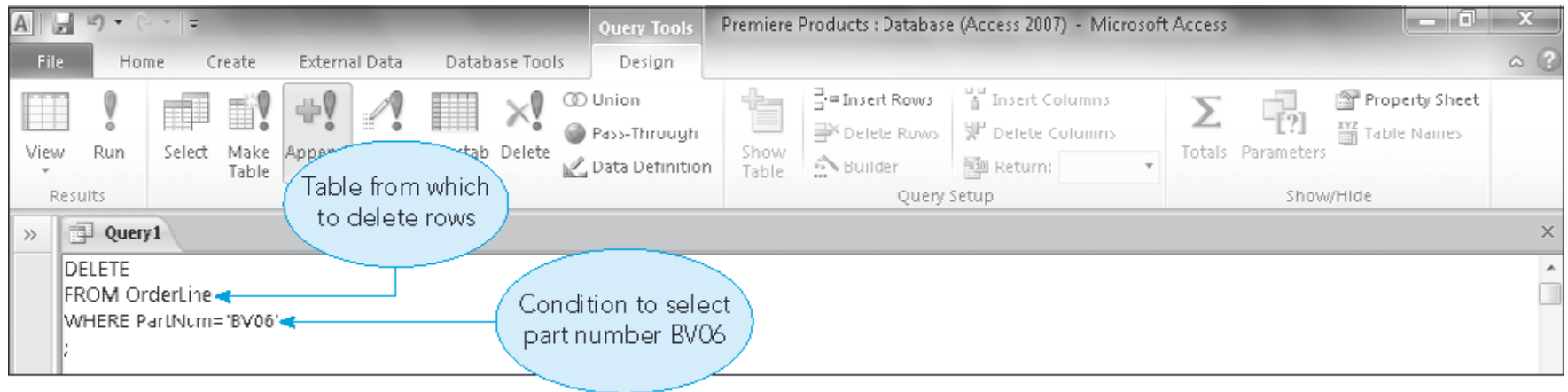


FIGURE 3-59: SQL query to delete rows

CREATING A TABLE FROM A QUERY (1/3)

○ **INTO clause**

- Saves the results of a query as a table
- Specified before FROM and WHERE clauses

○ **MySQL**

- Create the new table using a CREATE TABLE command
- Use an INSERT command to insert the appropriate data into the new table

CREATING A TABLE FROM A QUERY (2/3)

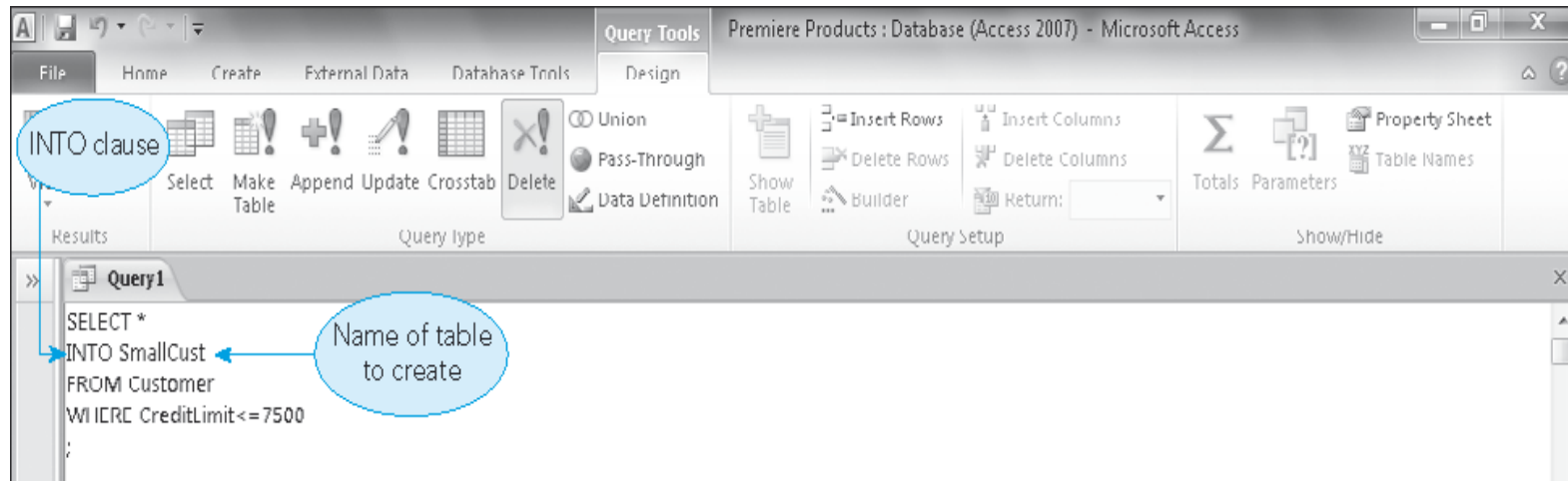


FIGURE 3-60a: Query to create a new table (Access)

CREATING A TABLE FROM A QUERY (3/3)

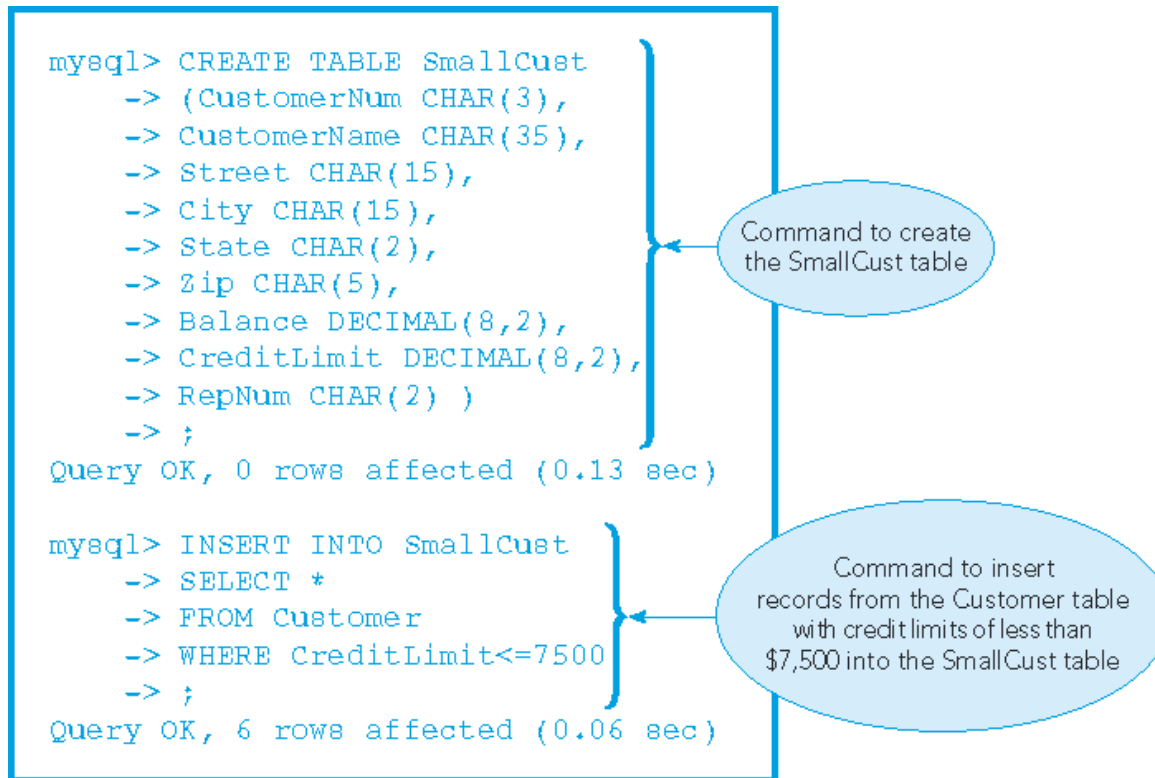


FIGURE 3-60b: Query to create a new table (for Oracle and MySQL)