CGS 2545: Database Concepts
Spring 2012

Chapter 7 – Advanced SQL

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
http://www.cs.ucf.edu/courses/cgs2545/spr2012

Department of Electrical Engineering and Computer Science
Computer Science Division
University of Central Florida

Processing Multiple Tables – Joins

- The ability to combine, or **join**, tables on common attributes is one of the most important advantages that relational databases have over other types of databases.
- A join is performed when data are retrieved from more than one table at a time.
- There are several different types of join operations as illustrated on the next page.
- In general, a join operation causes two or more tables with a common domain to be combined into a single table or view.

Processing Multiple Tables – Joins

- Theta-join – a join in which the joining condition is based on equality between values in the common columns; common columns appear redundantly in the result table. If all join conditions are equality, then the operation is known as an equi-join.
- Natural join – an equi-join in which one of the duplicate columns is eliminated in the result table.
- Outer join – a join in which rows that do not have matching values in common columns are nonetheless included in the result table (as opposed to *inner* join, in which rows must have matching values in order to appear in the result table).
- Union join – includes all columns from each table in the join, and an instance for each row of each table.

The common columns in joined tables are usually the primary key of the dominant table and the foreign key of the dependent table in 1:M relationships.

Theta-Join and Equijoin Operators

Type: binary

Symbol/general form: $r \rhd\lhd_{(predicate)} s$

Schema of result relation: concatenation of operand relations

Definition: $r \rhd\lhd_{(predicate)} s \equiv \sigma_{(predicate)}(r \times s)$

Examples:

$r \rhd\lhd_{(color='blue' \text{ AND } size=3)} s$ $r \rhd\lhd_{(color='blue' \text{ AND } size>3)} s$

an equijoin

a theta-join

- The theta-join operation is a shorthand for a Cartesian product followed by a selection operation.
- The equijoin operation is a special case of the theta-join operation that occurs when all of the conditions in the predicate are equality conditions.
- Neither a theta-join nor an equijoin operation eliminates extraneous tuples by default. Therefore, the elimination of extraneous tuples must be handled explicitly via the predicate.

Theta-Join Operator Examples

R

A	B	C	D
a	a	yes	1
b	d	no	7
c	f	yes	34
a	d	no	6

S

E	F	G	H
a	a	yes	1
b	r	yes	3
c	f	yes	34
m	n	no	56

$r = R \triangleright\triangleleft_{(R.B < S.F)} S$

A	B	C	D	E	F	G	H
a	a	yes	1	b	r	yes	3
a	a	yes	1	c	f	yes	34
a	a	yes	1	m	n	no	56
b	d	no	7	b	r	yes	3
b	d	no	7	c	f	yes	34
b	d	no	7	m	n	no	56
c	f	yes	34	b	r	yes	3
c	f	yes	34	m	n	no	56
a	d	no	6	b	r	yes	3
a	d	no	6	c	f	yes	34
a	d	no	6	m	n	no	56

Natural Join Operator

Type: binary

Symbol/general form: $r * s$

Schema of result relation: concatenation of operand relations with only one occurrence of commonly named attributes

Definition: $r * s \equiv r \rhd\lhd_{(r.commonattributes = s.commonattributes)} s$

Examples: $s * spj * p$

- The natural-join operation performs an equijoin over all attributes in the two operand relations which have the same attribute name.
- The degree of the result relation of a natural-join is sum of the degrees of the two operand relations less the number of attributes which are common to both operand relations. (In other words, one occurrence of each common attribute is eliminated from the result relation.)
- The natural join is probably the most common of all the forms of the join operation. It is extremely useful in the removal of extraneous tuples. Those attributes which are commonly named between the two operand relations are commonly referred to as the *join attributes*.

Natural Join Operator Examples

R

A	B	C	D
a	a	yes	1
b	r	no	7
c	f	yes	34
a	m	no	6

S

B	M	G	H
a	a	yes	1
b	r	yes	3
a	f	yes	34
m	n	no	56

r = R * S

A	B	C	D	M	G	H
a	a	yes	1	a	yes	1
a	a	yes	1	f	yes	34
a	m	no	6	n	no	56

r = R * T

A	B	C	D	G	H
b	r	no	7	yes	30

T

A	B	G	H
a	f	no	31
b	r	yes	30

Outer Join Operator

Type: binary

Symbol/general form: left-outer-join: $r \supset\!\lhd s$ right-outer-join: $r \rhd\!\subset s$

full outer join: $r \supset\!\lhd\!\rhd\!\subset s$

Schema of result relation: concatenation of operand relations

Definition:

$r \supset\!\lhd s$ ≡ natural join of r and s with tuples from r which do not have a match in s included in the result. Any missing values from s are set to null.

$r \rhd\!\subset s$ ≡ natural join of r and s with tuples from s which do not have a match in r included in the result. Any missing values from r are set to null.

$r \supset\!\lhd\!\rhd\!\subset s$≡ natural join of r and s with tuples from both r and s which do not have a match are included in the result. Any missing values are set to null.

Examples: Let r(A,B) = {(a, b), (c, d), (b,c)} and let

s(A,C) = {(a, d), (s, t), (b, d)}

then $r \supset\!\lhd s$ = (A,B,C) = {(a,b,d), (b,c,d), (c,d,null)},

$r \rhd\!\subset s$ = (A,B,C) = {(a,b,d), (b,c,d), (s,null,t)}, and

$r \supset\!\lhd\!\rhd\!\subset s$ = (A,B,C) = {(a,b,d), (b,c,d), (s,null,t), (c,d,null)},

Outer Join Operator Examples

R

A	B	C
1	2	3
4	5	6
7	8	9

$r = R \supset\triangleright\triangleleft\subset S$

A	B	C	D
1	2	3	10
1	2	3	11
4	5	6	null
7	8	9	null
null	6	7	12

S

B	C	D
2	3	10
2	3	11
6	7	12

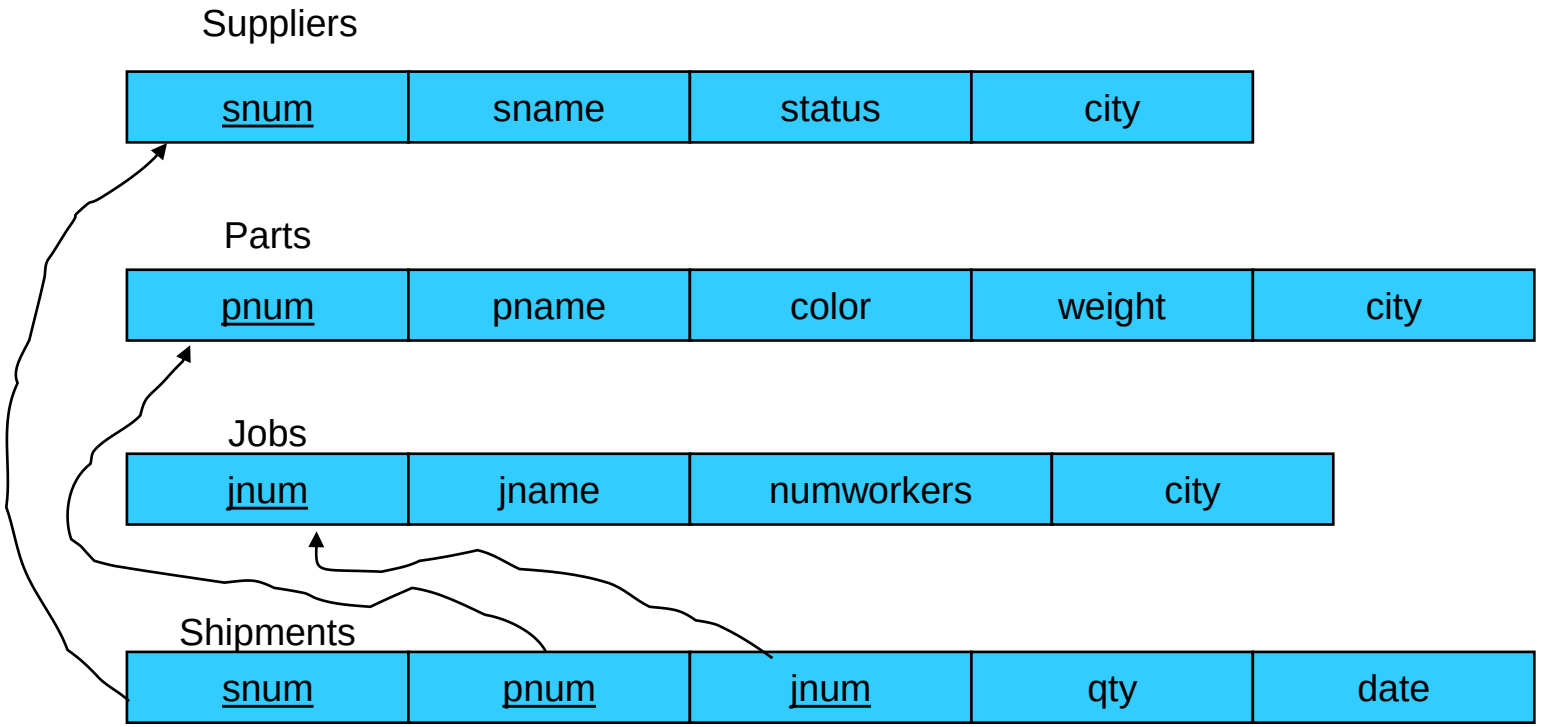
$r = R \supset\triangleleft S$

A	B	C	D
1	2	3	10
1	2	3	11
4	5	6	null
7	8	9	null

$r = R \triangleright\subset S$

B	C	D	A
2	3	10	1
2	3	11	1
6	7	12	null

An Example Database



The last three pages of this set of notes contain screen shots for these for tables from a sample database using these tables. They might help to make some of the following examples more clear.

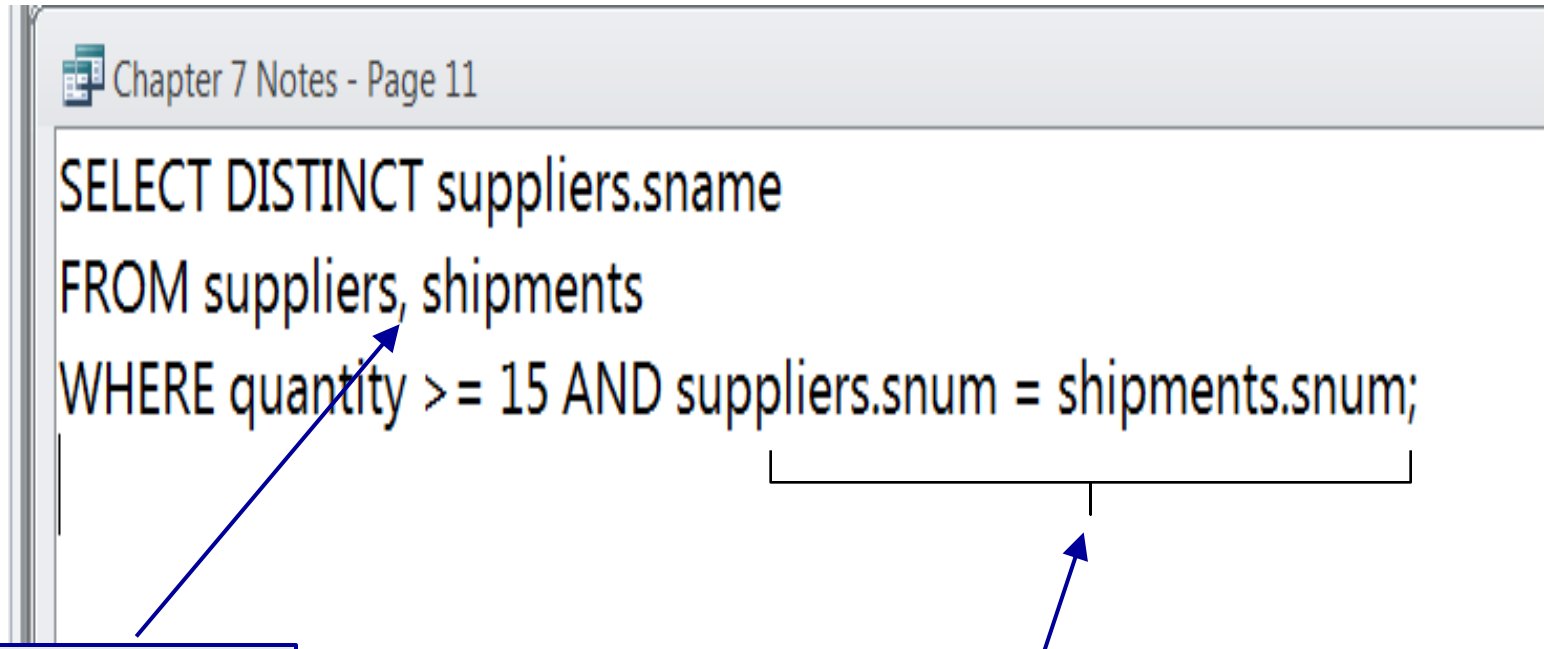
Natural Join Example 1

Query: List only the names (remove duplicates) of those suppliers who have a shipment with a quantity >= 15.

```
SELECT DISTINCT sname
FROM Suppliers NATURAL JOIN Shipments
WHERE quantity >= 15;
```

Note that Access does not support the natural join syntax shown above. In Access this query is expressed as shown on the next page.

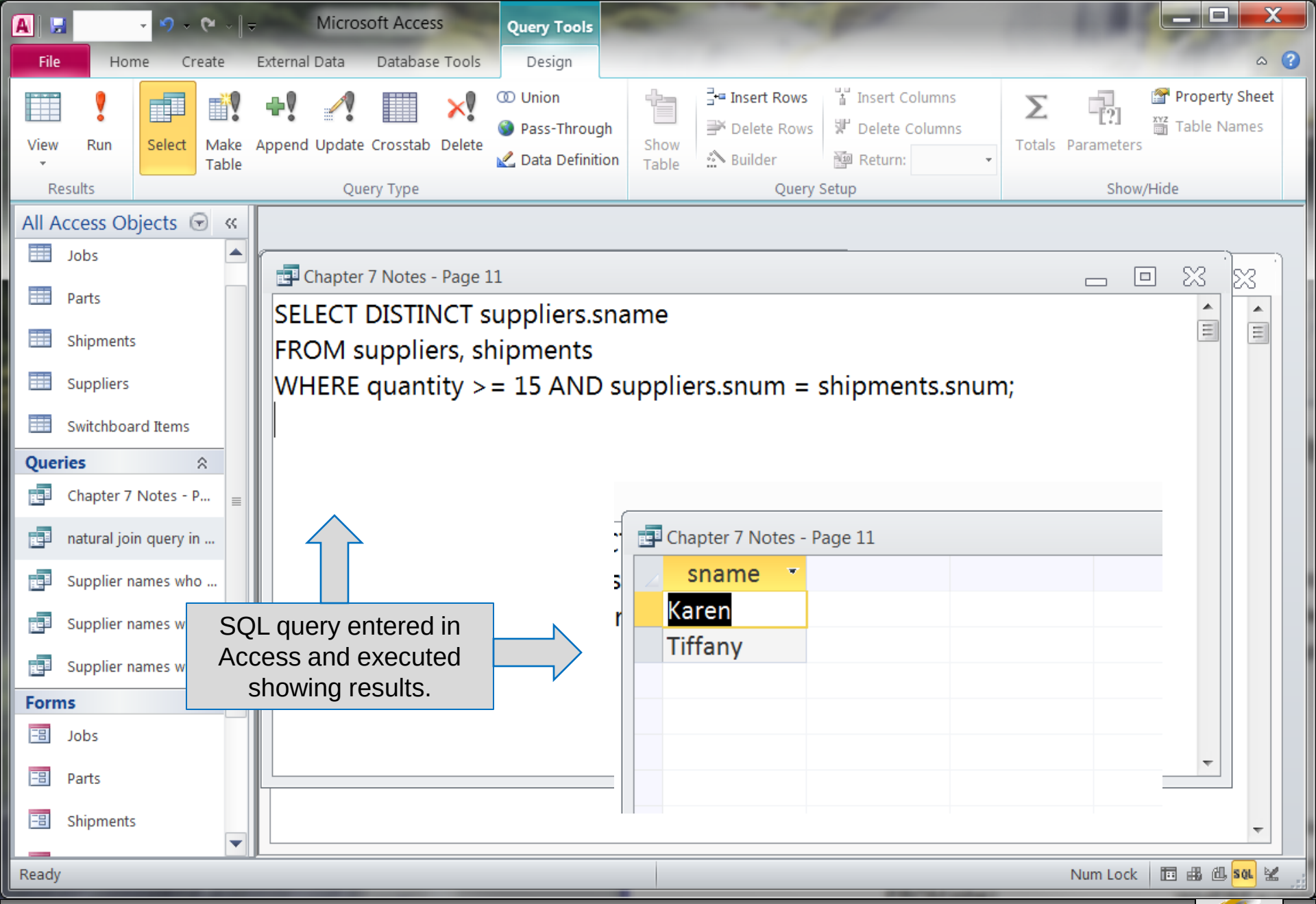
Natural Join Example 1 In Access



This comma is the generic join operator in SQL.

This condition is called an "implicit join condition". It specifies the criteria on which the join is to occur.

In this case, the primary key in suppliers and the foreign key (part of the primary key) in the shipments table must be the same.


Microsoft Access
Query Tools
File
Home
Create
External Data
Database Tools
Design
View
Run
Select
Make Table
Append
Update
Crosstab
Delete
Union
Pass-Through
Data Definition
Show Table
Insert Rows
Delete Rows
Builder
Insert Columns
Delete Columns
Return:
Totals
Parameters
Property Sheet
Table Names
Results
Query Type
Query Setup
Show/Hide
All Access Objects
Jobs
Parts
Shipments
Suppliers
Switchboard Items
Queries
Chapter 7 Notes - P...
natural join query in ...
Supplier names who ...
Supplier names w
Supplier names w
Forms
Jobs
Parts
Shipments
Chapter 7 Notes - Page 11
SELECT DISTINCT suppliers.sname
FROM suppliers, shipments
WHERE quantity >= 15 AND suppliers.snum = shipments.snum;
SQL query entered in Access and executed showing results.
Chapter 7 Notes - Page 11
sname
Karen
Tiffany
Ready
Num Lock
SQL

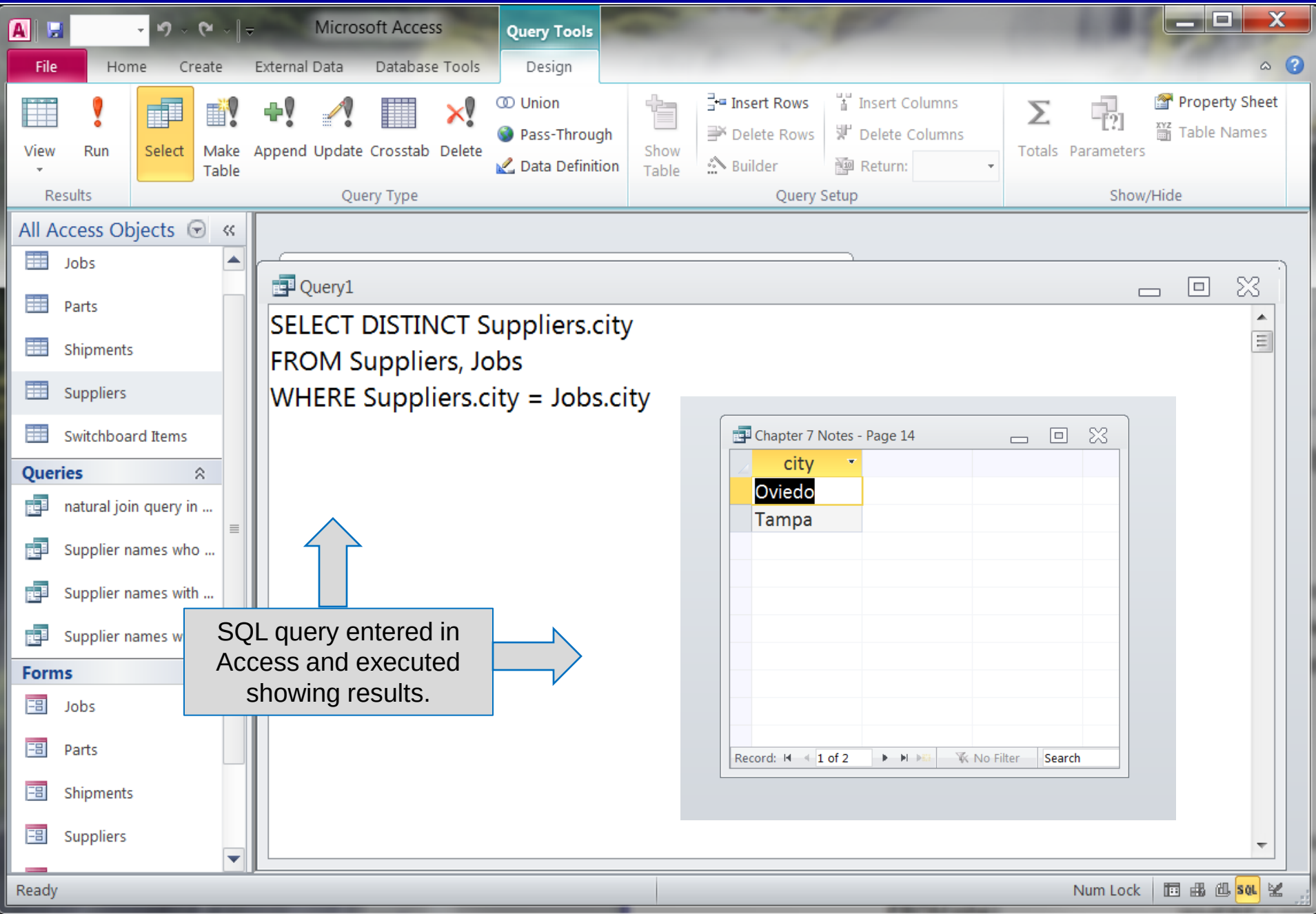
Natural Join Example 2

Query: List only the names (remove duplicates) of those cities in which both a supplier and a job are located.

```
SELECT DISTINCT Supplier.city
FROM Suppliers NATURAL JOIN Jobs;
```

Access Version:

```
SELECT DISTINCT Supplier.city
FROM Suppliers, Jobs
WHERE Suppliers.city = Jobs.city;
```



Microsoft Access
Query Tools
File
Home
Create
External Data
Database Tools
Design
View
Run
Select
Make Table
Append
Update
Crosstab
Delete
Union
Pass-Through
Data Definition
Show Table
Insert Rows
Delete Rows
Builder
Insert Columns
Delete Columns
Return:
Totals
Parameters
Property Sheet
Table Names
Results
Query Type
Query Setup
Show/Hide
All Access Objects
Jobs
Parts
Shipments
Suppliers
Switchboard Items
Queries
natural join query in ...
Supplier names who ...
Supplier names with ...
Supplier names w
Forms
Jobs
Parts
Shipments
Suppliers
Query1
SELECT DISTINCT Suppliers.city
FROM Suppliers, Jobs
WHERE Suppliers.city = Jobs.city
SQL query entered in Access and executed showing results.
Chapter 7 Notes - Page 14
city
Oviedo
Tampa
Record: 1 of 2
No Filter
Search
Ready
Num Lock
SQL

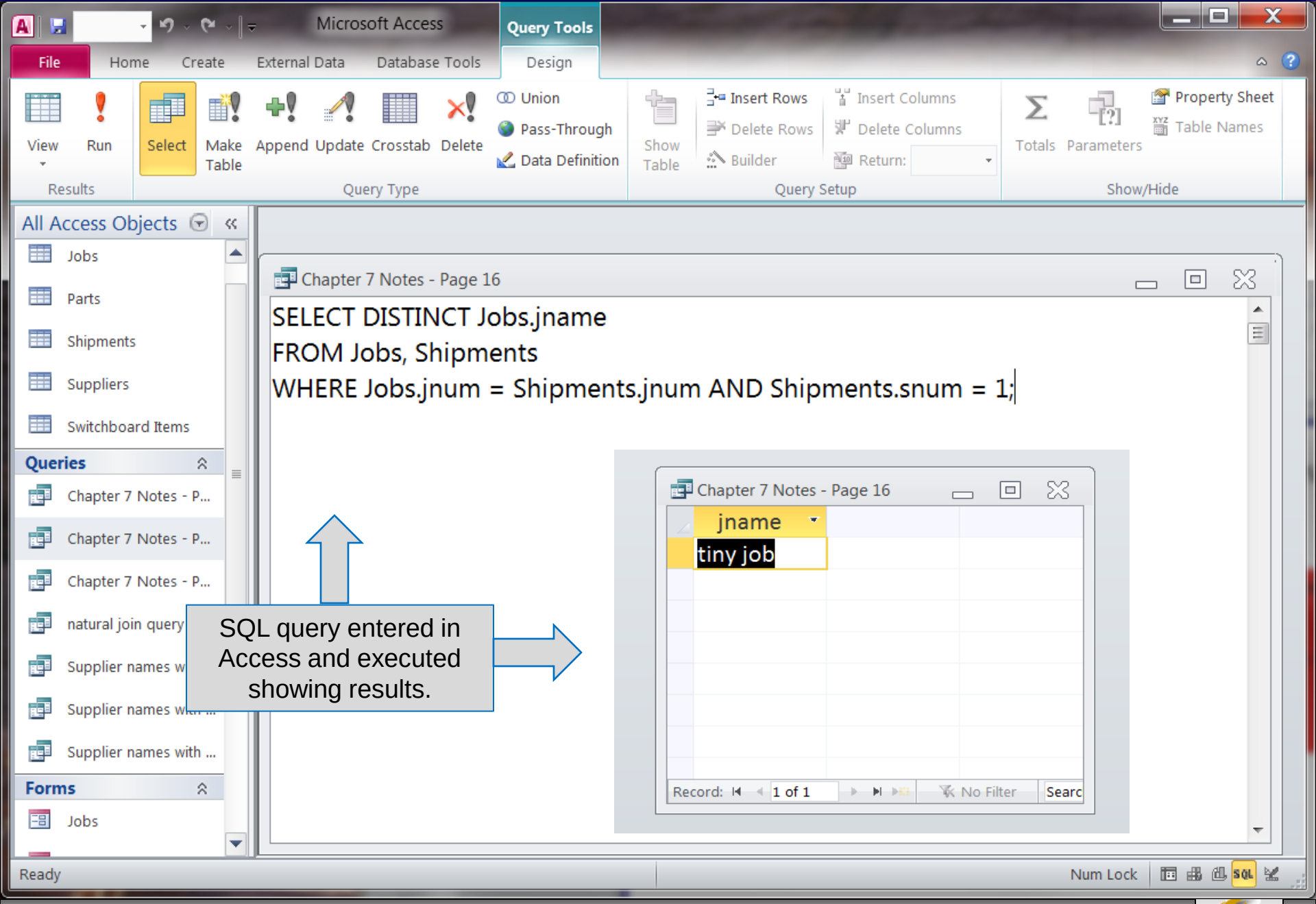
Natural Join Example 3

Query: List only the names (remove duplicates) of those jobs which receive a shipment from supplier number 1.

```
SELECT DISTINCT Jobs.jname
FROM Jobs NATURAL JOIN Shipments
WHERE Shipments.snum = 1;
```

Access Version:

```
SELECT DISTINCT Jobs.jname
FROM Jobs, Shipments
WHERE Jobs.jnum = Shipments.jnum
    AND Shipments.snum = 1;
```


Microsoft Access
Query Tools
File
Home
Create
External Data
Database Tools
Design
View
Run
Select
Make Table
Append
Update
Crosstab
Delete
Union
Pass-Through
Data Definition
Show Table
Insert Rows
Delete Rows
Builder
Insert Columns
Delete Columns
Return:
Totals
Parameters
Property Sheet
Table Names
Results
Query Type
Query Setup
Show/Hide
All Access Objects
Jobs
Parts
Shipments
Suppliers
Switchboard Items
Queries
Chapter 7 Notes - P...
Chapter 7 Notes - P...
Chapter 7 Notes - P...
natural join query
Supplier names w
Supplier names with ...
Supplier names with ...
Forms
Jobs
Chapter 7 Notes - Page 16
SELECT DISTINCT Jobs.jname
FROM Jobs, Shipments
WHERE Jobs.jnum = Shipments.jnum AND Shipments.snum = 1;
SQL query entered in Access and executed showing results.
Chapter 7 Notes - Page 16
jname
tiny job
Record: 1 of 1
No Filter
Searc
Ready
Num Lock
SQL

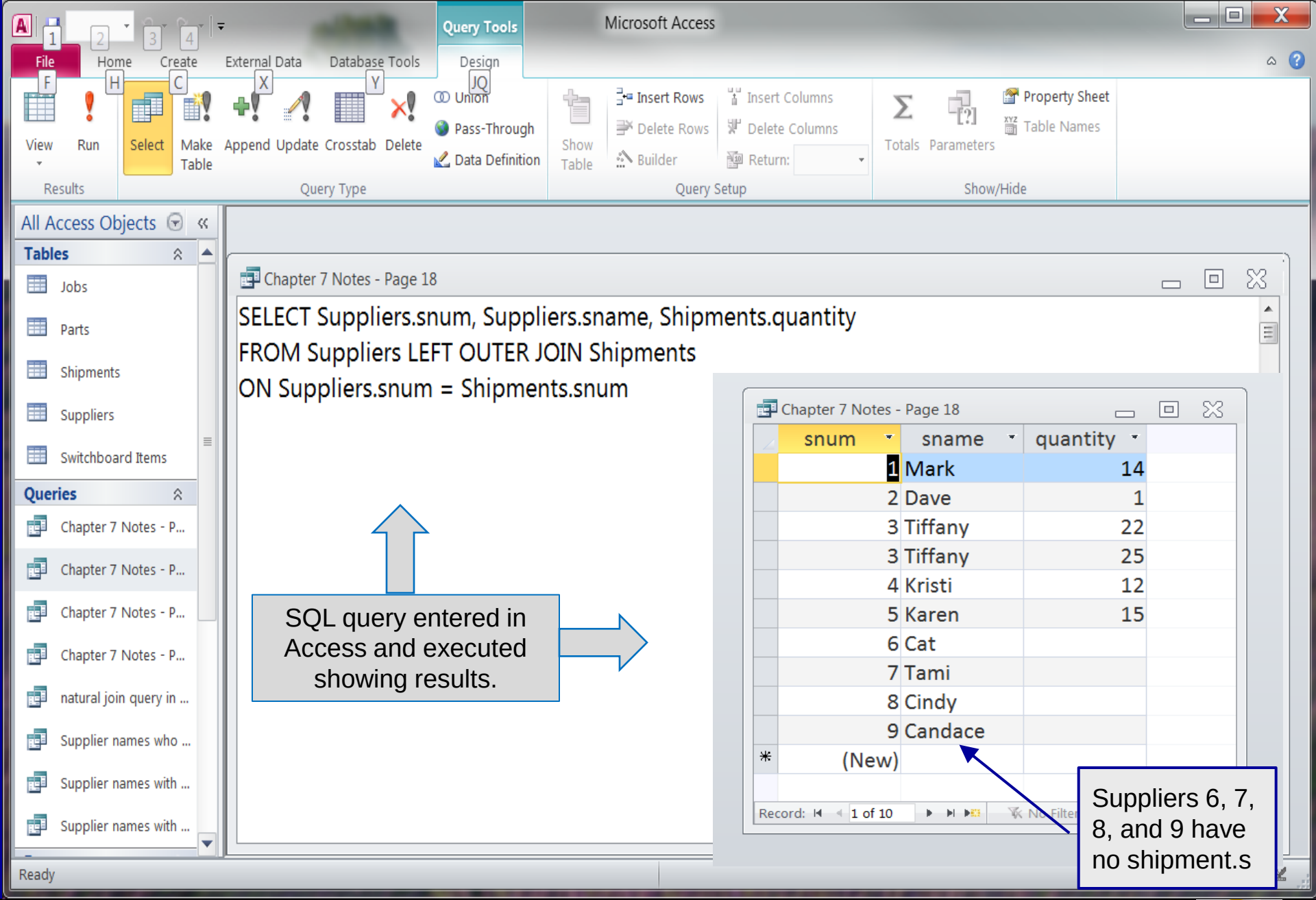
Left Outer Join Example

- List the supplier numbers and names along with the quantity of each order a supplier has and include supplier information even for suppliers who have no shipments.

Access version:

```
SELECT Suppliers.snum, Suppliers.sname,
  Shipments.quantity
FROM Suppliers LEFT OUTER JOIN Shipments
ON Suppliers.snum = Shipments.snum;
```

LEFT OUTER JOIN syntax with ON keyword instead of WHERE causes supplier information to appear even if there is no corresponding shipment information for that supplier.


Microsoft Access
Query Tools
File
Home
Create
External Data
Database Tools
Design
View
Run
Select
Make Table
Append
Update
Crosstab
Delete
Union
Pass-Through
Data Definition
Show Table
Insert Rows
Delete Rows
Builder
Insert Columns
Delete Columns
Return:
Totals
Parameters
Property Sheet
Table Names
Results
Query Type
Query Setup
Show/Hide
All Access Objects
Tables
Jobs
Parts
Shipments
Suppliers
Switchboard Items
Queries
Chapter 7 Notes - P...
Chapter 7 Notes - P...
Chapter 7 Notes - P...
Chapter 7 Notes - P...
natural join query in ...
Supplier names who ...
Supplier names with ...
Supplier names with ...
Chapter 7 Notes - Page 18
SELECT Suppliers.snum, Suppliers.sname, Shipments.quantity
FROM Suppliers LEFT OUTER JOIN Shipments
ON Suppliers.snum = Shipments.snum
SQL query entered in Access and executed showing results.
Chapter 7 Notes - Page 18
snum
sname
quantity
1 Mark 14
2 Dave 1
3 Tiffany 22
3 Tiffany 25
4 Kristi 12
5 Karen 15
6 Cat
7 Tami
8 Cindy
9 Candace
(New)
Record: 1 of 10
Suppliers 6, 7, 8, and 9 have no shipment.s
Ready

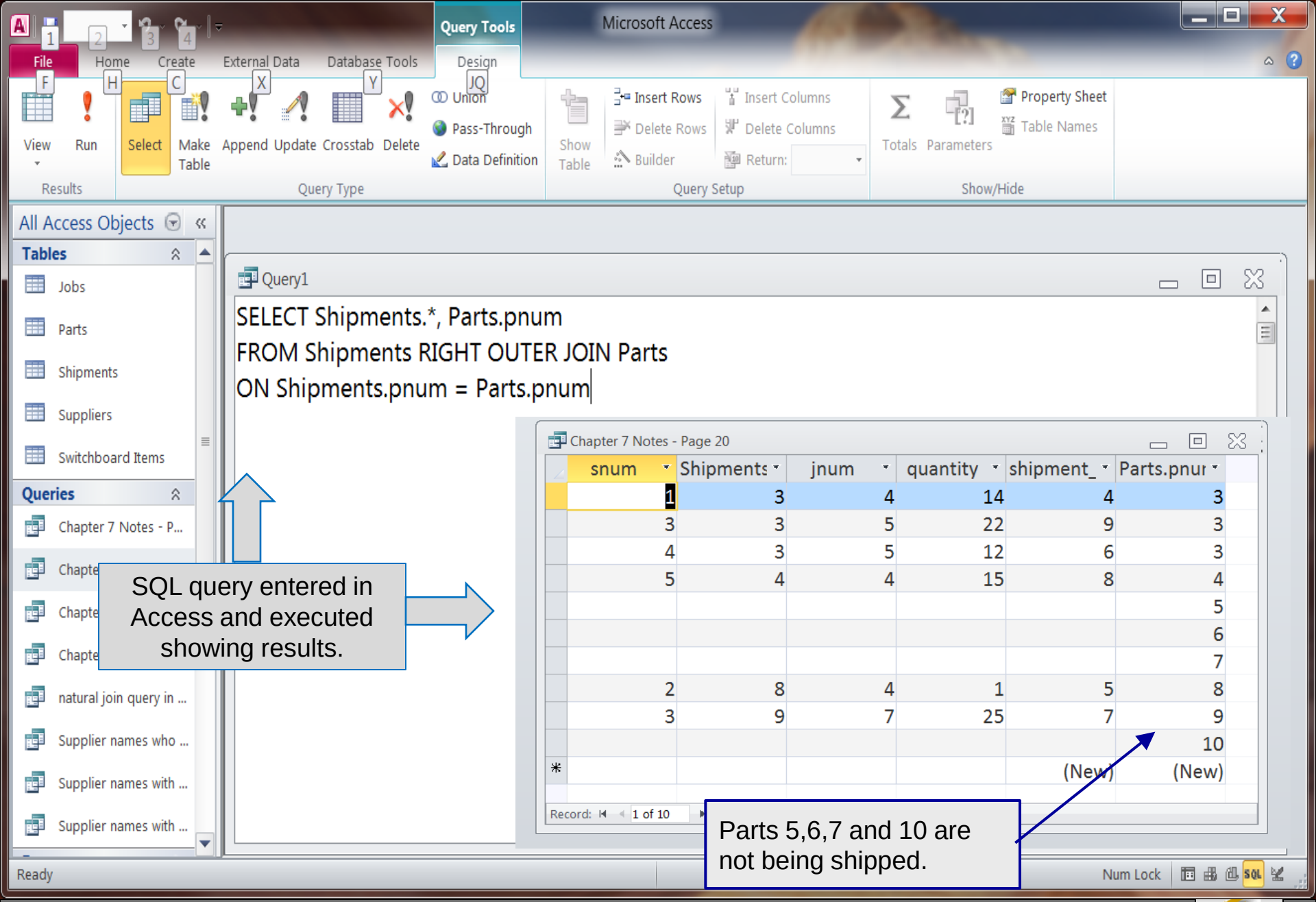
Right Outer Join Example

- List all the information about each shipment and the part number of every part that is not shipped by any supplier.

Access version:

```
SELECT Shipments.*, Parts.pnum
FROM Shipments RIGHT OUTER JOIN Parts
ON Shipments.pnum = Parts.pnum;
```

RIGHT OUTER JOIN syntax with ON keyword instead of WHERE causes part information to appear even if there is no corresponding shipment information for that part.


Query Tools
Microsoft Access
File
Home
Create
External Data
Database Tools
Design
View
Run
Select
Make Table
Append
Update
Crosstab
Delete
Union
Pass-Through
Data Definition
Show Table
Insert Rows
Delete Rows
Builder
Insert Columns
Delete Columns
Return:
Totals
Parameters
Property Sheet
Table Names
Results
Query Type
Query Setup
Show/Hide
All Access Objects
Tables
Jobs
Parts
Shipments
Suppliers
Switchboard Items
Queries
Chapter 7 Notes - P...
natural join query in ...
Supplier names who ...
Supplier names with ...
Supplier names with ...
Query1
SELECT Shipments.*, Parts.pnum
FROM Shipments RIGHT OUTER JOIN Parts
ON Shipments.pnum = Parts.pnum
Chapter 7 Notes - Page 20
snum
Shipments
jnum
quantity
shipment_
Parts.pnur
1 3 4 14 4 3
3 3 5 22 9 3
4 3 5 12 6 3
5 4 4 15 8 4
5
6
7
2 8 4 1 5 8
3 9 7 25 7 9
10
(New) (New)
Record: 1 of 10
SQL query entered in Access and executed showing results.
Parts 5,6,7 and 10 are not being shipped.
Ready
Num Lock

Multiple Table Join Example 1

- List the supplier name and city for every supplier who has a shipment of a blue part.

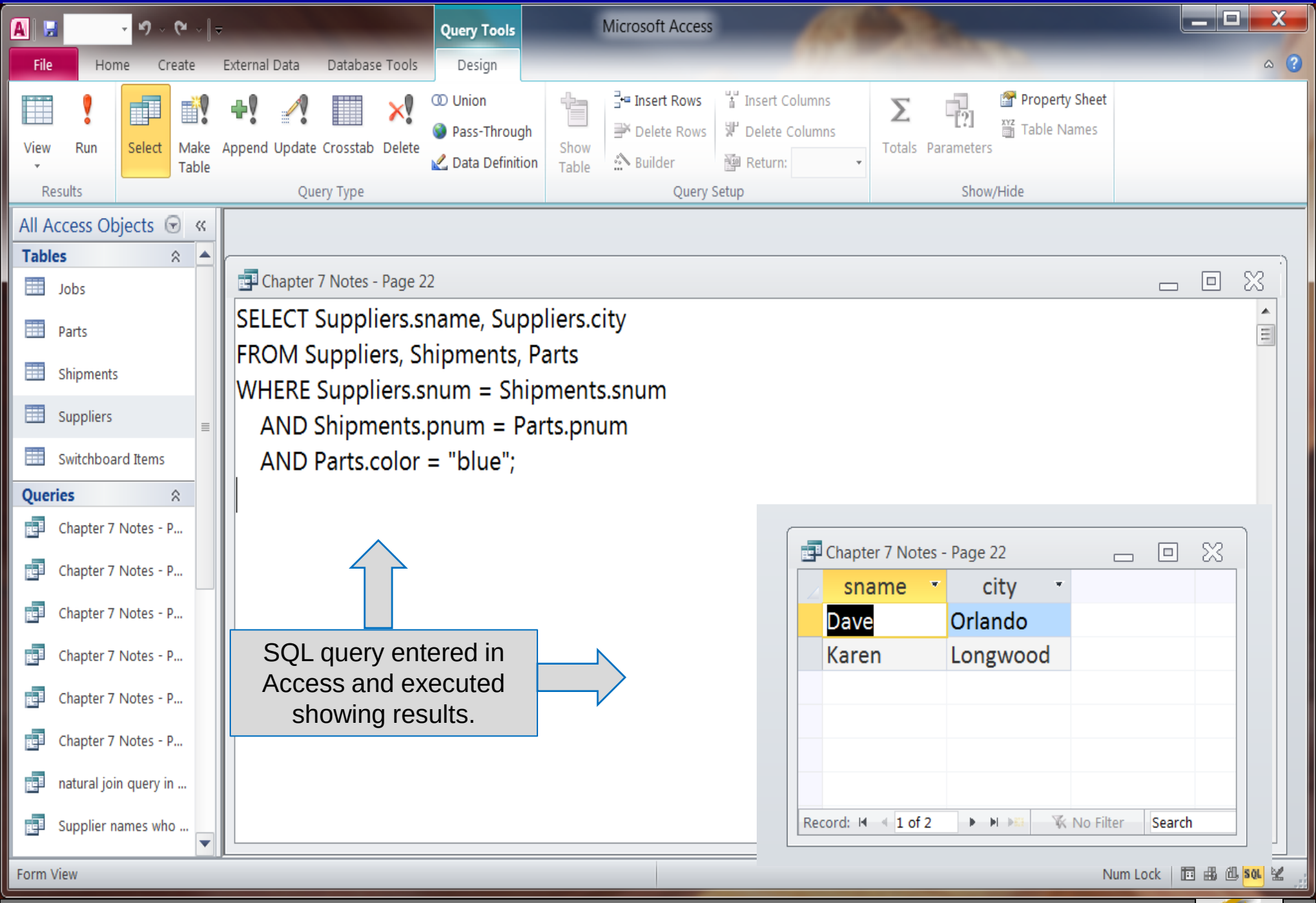
SQL Version:

```
SELECT Suppliers.sname, Suppliers.city
FROM Suppliers NATURAL JOIN Shipments NATURAL JOIN Parts
WHERE Parts.color = "blue";
```

Access Version:

```
SELECT Suppliers.sname, Suppliers.city
FROM Suppliers,Shipments,Parts
WHERE Suppliers.snum = Shipments.snum
    AND Shipments.pnum = Parts.pnum
    AND Parts.color = "blue";
```

Each pair of tables requires an implicit join condition in the WHERE clause, matching primary keys against foreign keys


Query Tools
Microsoft Access
File
Home
Create
External Data
Database Tools
Design
View
Run
Select
Make Table
Append
Update
Crosstab
Delete
Union
Pass-Through
Data Definition
Show Table
Insert Rows
Delete Rows
Builder
Insert Columns
Delete Columns
Return:
Totals
Parameters
Property Sheet
Table Names
Results
Query Type
Query Setup
Show/Hide
All Access Objects
Tables
Jobs
Parts
Shipments
Suppliers
Switchboard Items
Queries
Chapter 7 Notes - P...
natural join query in ...
Supplier names who ...
Chapter 7 Notes - Page 22
SELECT Suppliers.sname, Suppliers.city
FROM Suppliers, Shipments, Parts
WHERE Suppliers.snum = Shipments.snum
AND Shipments.pnum = Parts.pnum
AND Parts.color = "blue";
SQL query entered in Access and executed showing results.
sname
city
Dave
Orlando
Karen
Longwood
Record: 1 of 2
No Filter
Search
Form View
Num Lock

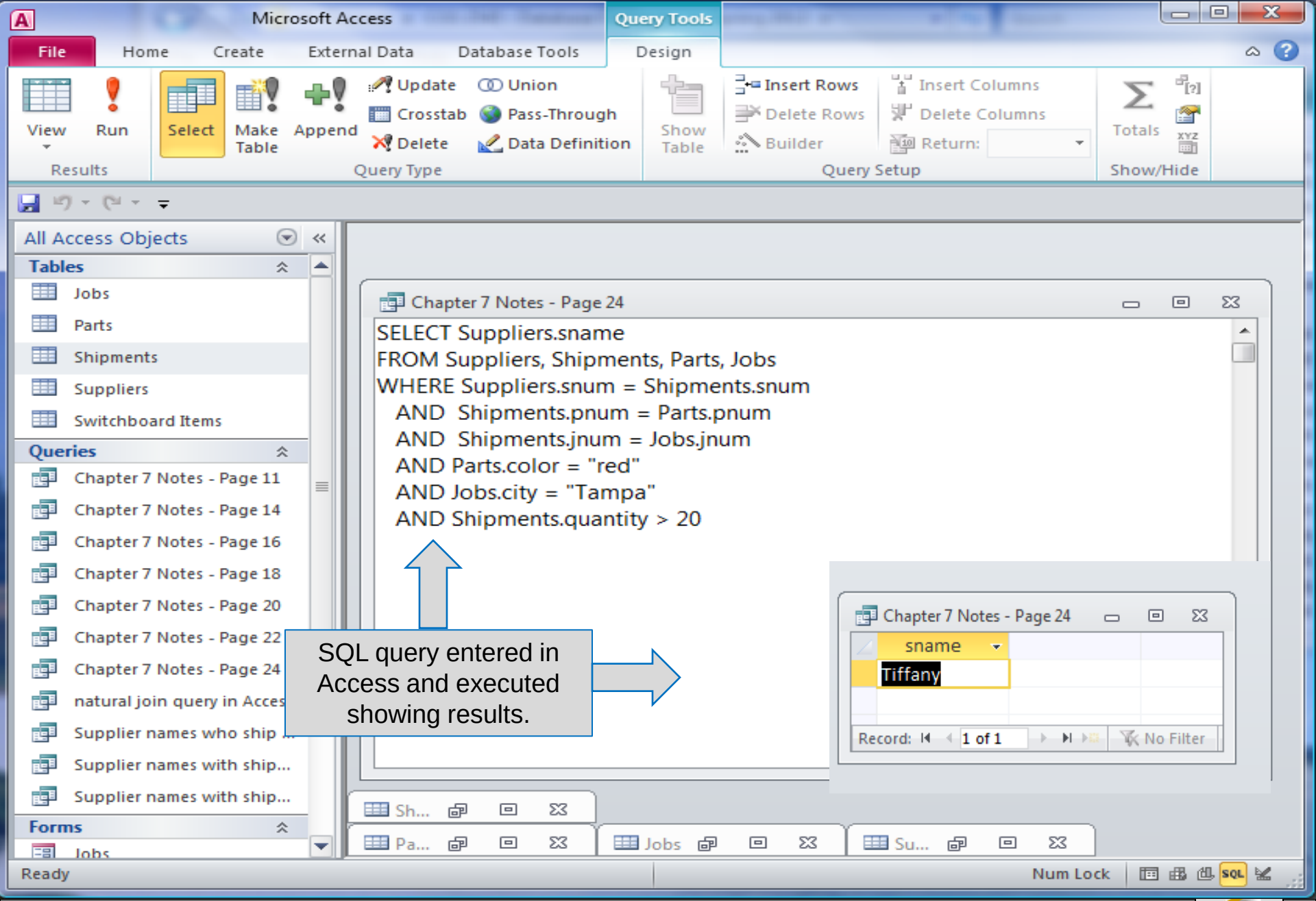
Multiple Table Join Example 2

- List the supplier names for those suppliers who supply a red part to any job in Tampa in a quantity > 20.

SQL Version:

```
SELECT Suppliers.sname
FROM Suppliers NATURAL JOIN Shipments NATURAL JOIN Parts
     NATURAL JOIN Jobs
WHERE Parts.color = "red" AND Jobs.city = "Tampa"
    AND Shipments.quantity > 20;
```

Access Version: (see next page)


Microsoft Access
Query Tools
File
Home
Create
External Data
Database Tools
Design
View
Run
Select
Make Table
Append
Update
Union
Crosstab
Pass-Through
Delete
Data Definition
Show Table
Insert Rows
Delete Rows
Builder
Insert Columns
Delete Columns
Return:
Totals
Results
Query Type
Query Setup
Show/Hide
All Access Objects
Tables
Jobs
Parts
Shipments
Suppliers
Switchboard Items
Queries
Chapter 7 Notes - Page 11
Chapter 7 Notes - Page 14
Chapter 7 Notes - Page 16
Chapter 7 Notes - Page 18
Chapter 7 Notes - Page 20
Chapter 7 Notes - Page 22
Chapter 7 Notes - Page 24
natural join query in Acces
Supplier names who ship ...
Supplier names with ship...
Supplier names with ship...
Forms
Jobs
Chapter 7 Notes - Page 24
SELECT Suppliers.sname
FROM Suppliers, Shipments, Parts, Jobs
WHERE Suppliers.snum = Shipments.snum
AND Shipments.pnum = Parts.pnum
AND Shipments.jnum = Jobs.jnum
AND Parts.color = "red"
AND Jobs.city = "Tampa"
AND Shipments.quantity > 20
SQL query entered in Access and executed showing results.
Chapter 7 Notes - Page 24
sname
Tiffany
Record: 1 of 1
No Filter
Sh...
Pa...
Jobs
Su...
Ready
Num Lock
SQL

Processing Multiple Tables Using Subqueries

- A subquery is formed by placing a query inside a query, i.e., placing a SELECT statement (the inner query) inside a SELECT statement (the outer query).
- A subquery can occur in several different location: The options are:
  - In a condition of the WHERE clause.
  - As a "table" of the FROM clause.
  - Within the HAVING clause.
- Subqueries can be:
  - Noncorrelated – executed once for the entire outer query.
  - Correlated – executed once for each row returned by the outer query.

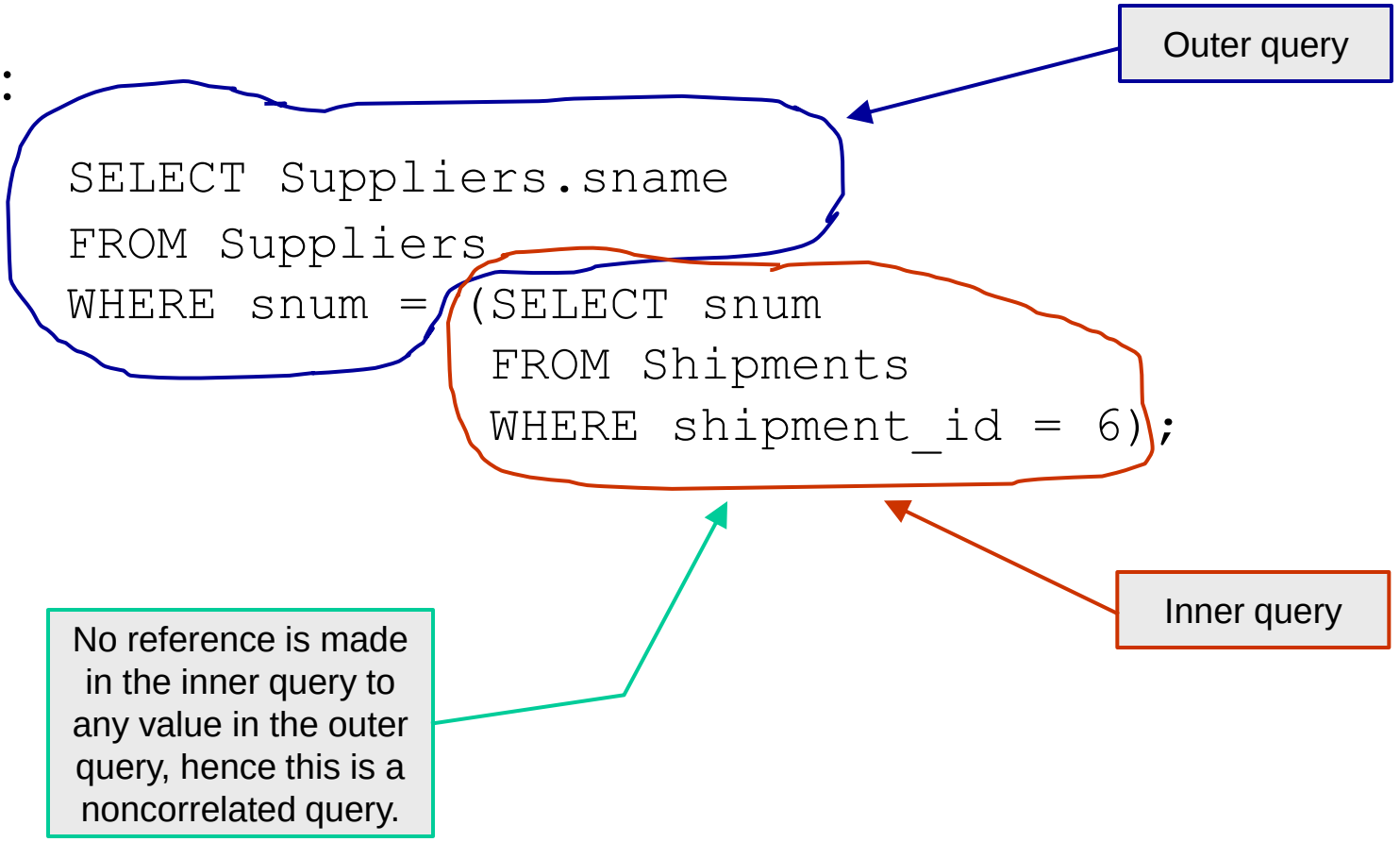
Correlated vs. Noncorrelated Subqueries

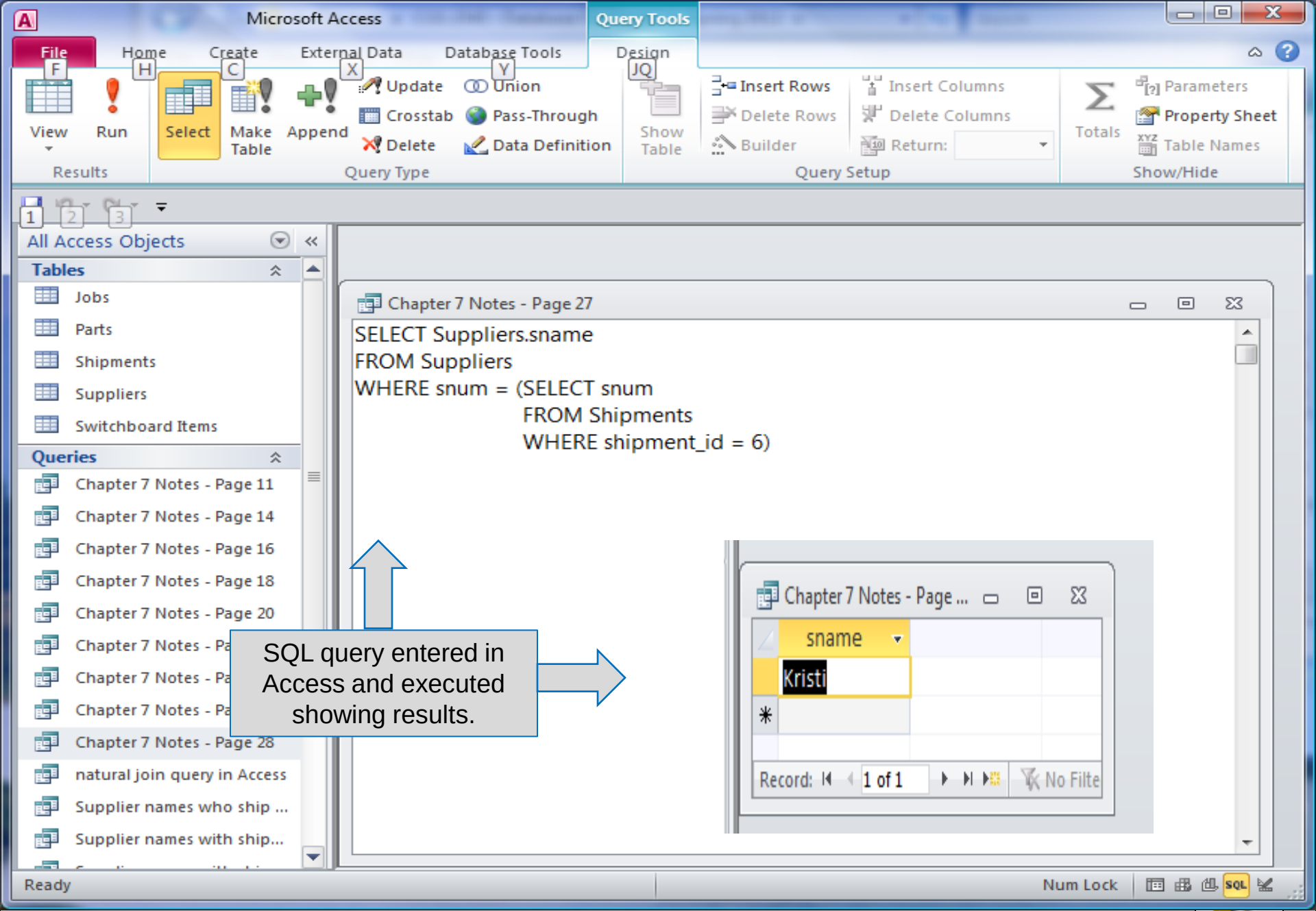
- Noncorrelated subqueries:
  - Do not depend on data from the outer query.
  - Execute once for the entire outer query.
- Correlated subqueries:
  - Make use of data from the outer query.
  - Execute once for each row of the outer query.
  - Can use the EXISTS operator.

Subquery Example 1 (Where clause)

- List the name of the supplier who shipped shipment number 6.

SQL:




Microsoft Access
Query Tools
File
Home
Create
External Data
Database Tools
Design
View
Run
Select
Make Table
Append
Update
Union
Crosstab
Pass-Through
Delete
Data Definition
Show Table
Insert Rows
Delete Rows
Builder
Insert Columns
Delete Columns
Return:
Totals
Parameters
Property Sheet
Table Names
Results
Query Type
Query Setup
Show/Hide
All Access Objects
Tables
Jobs
Parts
Shipments
Suppliers
Switchboard Items
Queries
Chapter 7 Notes - Page 11
Chapter 7 Notes - Page 14
Chapter 7 Notes - Page 16
Chapter 7 Notes - Page 18
Chapter 7 Notes - Page 20
Chapter 7 Notes - Page 28
natural join query in Access
Supplier names who ship ...
Supplier names with ship...
Chapter 7 Notes - Page 27
SELECT Suppliers.sname
FROM Suppliers
WHERE snum = (SELECT snum
FROM Shipments
WHERE shipment_id = 6)
SQL query entered in Access and executed showing results.
Chapter 7 Notes - Page ...
sname
Kristi
Record: 1 of 1
No Filte
Ready
Num Lock

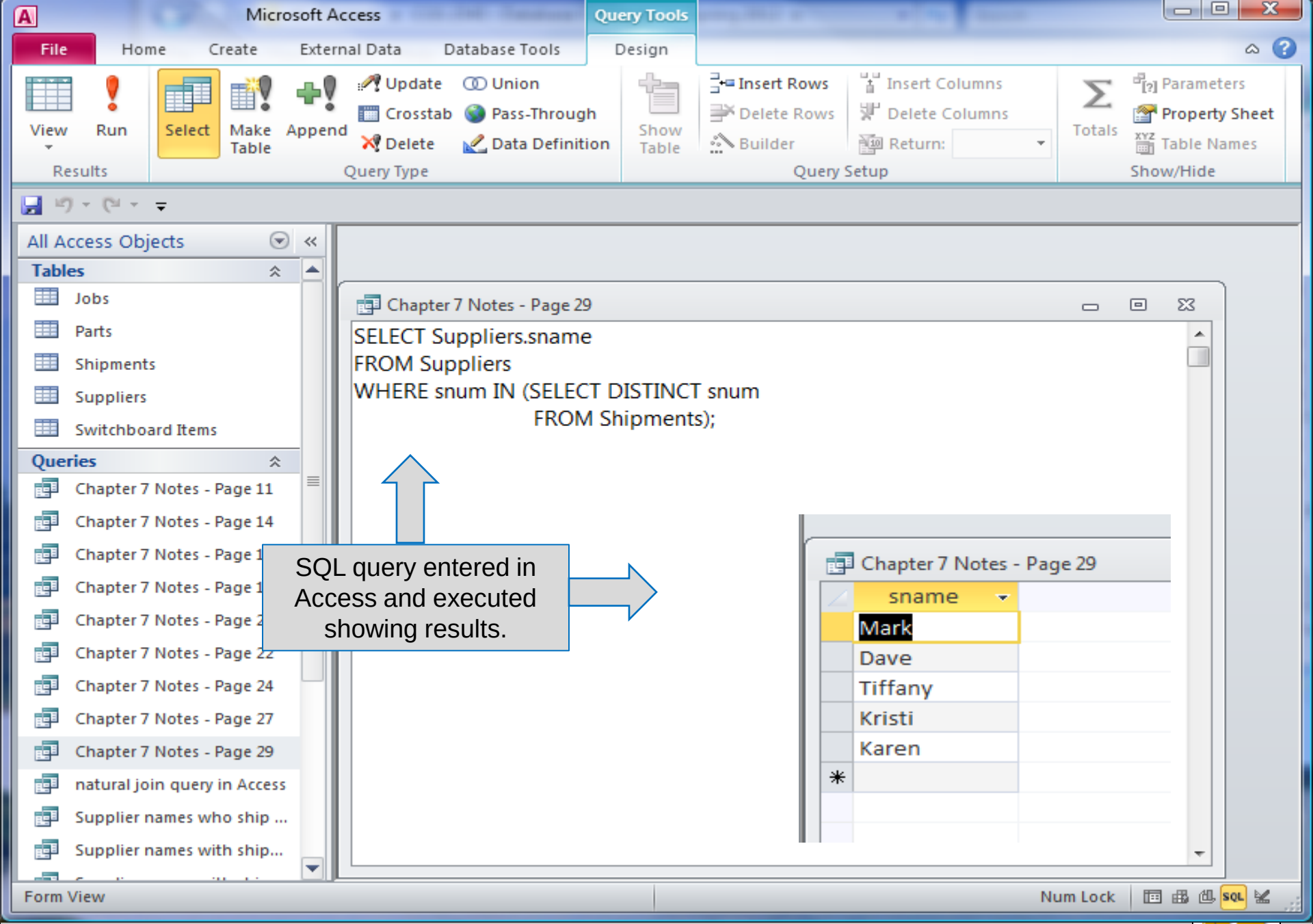
Subquery Example 2 (Where clause)

- List the names of those suppliers who at least one shipment.

SQL:

```
SELECT Suppliers.sname
FROM Suppliers
WHERE snum IN (SELECT DISTINCT snum
               FROM Shipments);
```

The IN operator is a set operator that checks to see if the left-hand operand (a value or set member instance) is contained in the right-hand operand (a set). The IN operator returns true or false.

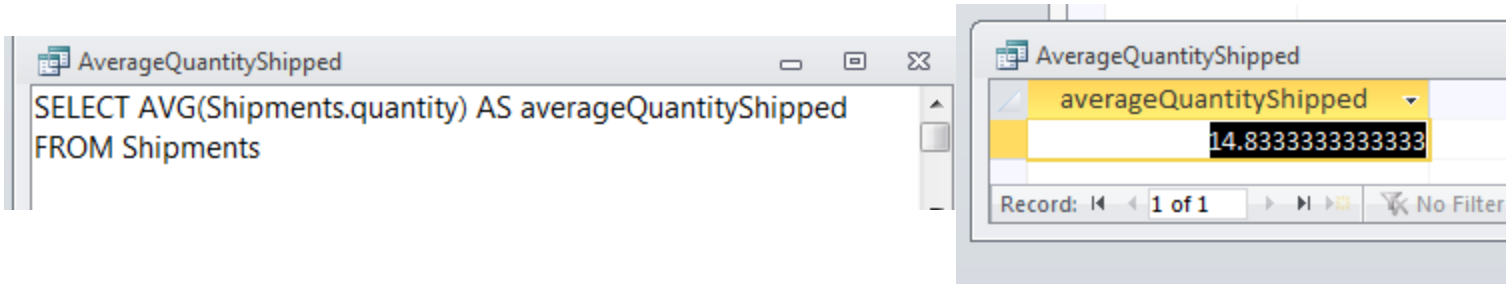

Microsoft Access
Query Tools
File
Home
Create
External Data
Database Tools
Design
View
Run
Select
Make Table
Append
Update
Union
Crosstab
Pass-Through
Delete
Data Definition
Show Table
Insert Rows
Delete Rows
Builder
Insert Columns
Delete Columns
Return:
Totals
Parameters
Property Sheet
Table Names
Results
Query Type
Query Setup
Show/Hide
All Access Objects
Tables
Jobs
Parts
Shipments
Suppliers
Switchboard Items
Queries
Chapter 7 Notes - Page 11
Chapter 7 Notes - Page 14
Chapter 7 Notes - Page 1
Chapter 7 Notes - Page 1
Chapter 7 Notes - Page 2
Chapter 7 Notes - Page 22
Chapter 7 Notes - Page 24
Chapter 7 Notes - Page 27
Chapter 7 Notes - Page 29
natural join query in Access
Supplier names who ship ...
Supplier names with ship...
Chapter 7 Notes - Page 29
SELECT Suppliers.sname
FROM Suppliers
WHERE snum IN (SELECT DISTINCT snum
FROM Shipments);
SQL query entered in Access and executed showing results.
Chapter 7 Notes - Page 29
sname
Mark
Dave
Tiffany
Kristi
Karen
Form View
Num Lock
SQL

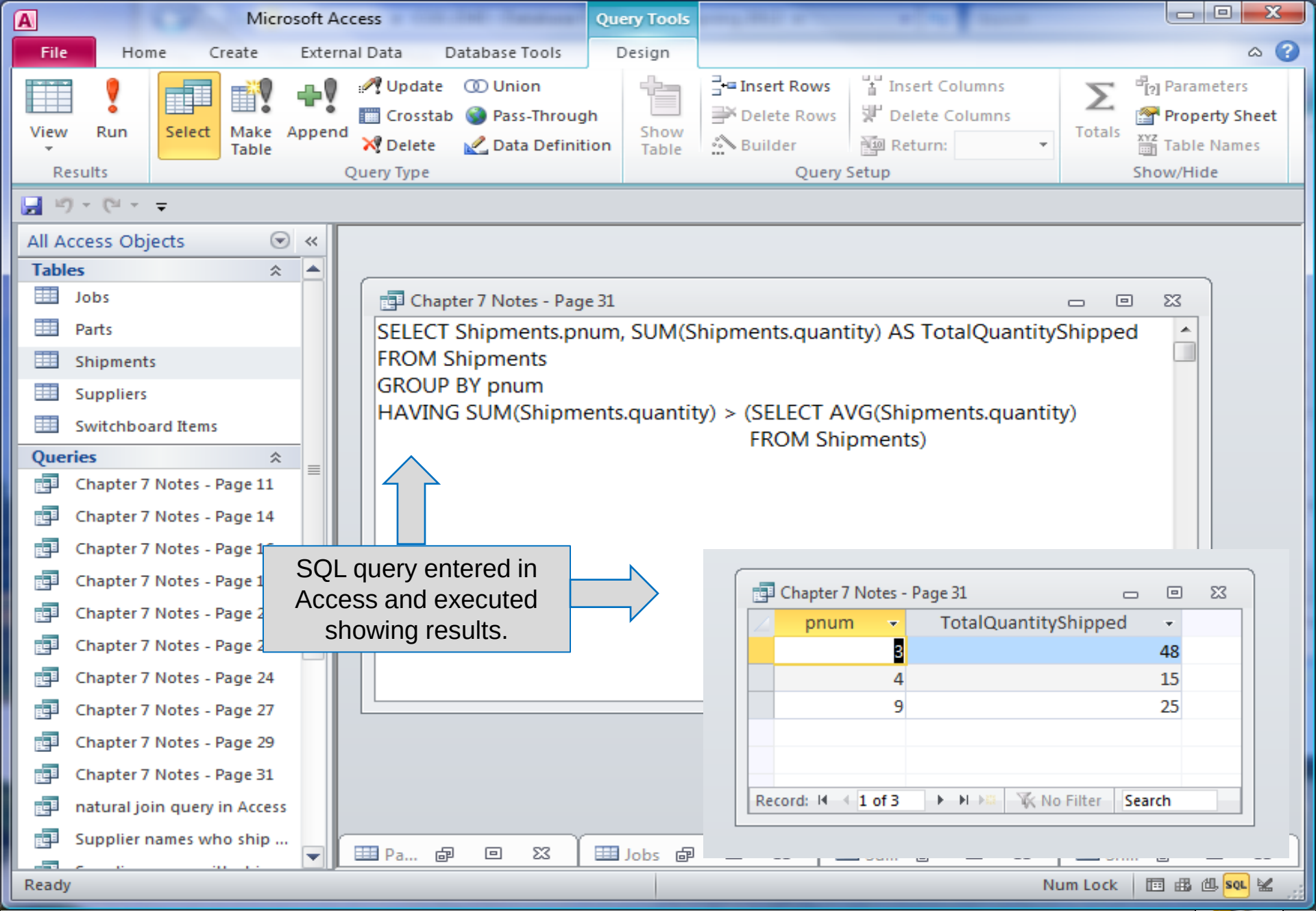
Subquery Example 3 (Having clause)

- List the part names and the total quantity shipped of that part for parts that are supplied in quantities greater than the average quantity of all parts supplied.

SQL/Access Version:

```
SELECT Shipments.pnum, SUM(Shipments.quantity) AS
      TotalQuantityShipped
FROM Shipments
GROUP BY pnum
HAVING SUM(Shipments.quantity) >
          (SELECT AVG(Shipments.quantity)
           FROM Shipments);
```




Microsoft Access
Query Tools
File
Home
Create
External Data
Database Tools
Design
View
Run
Select
Make Table
Append
Update
Union
Crosstab
Pass-Through
Delete
Data Definition
Show Table
Insert Rows
Delete Rows
Builder
Insert Columns
Delete Columns
Return:
Totals
Parameters
Property Sheet
Table Names
Results
Query Type
Query Setup
Show/Hide
All Access Objects
Tables
Jobs
Parts
Shipments
Suppliers
Switchboard Items
Queries
Chapter 7 Notes - Page 11
Chapter 7 Notes - Page 14
Chapter 7 Notes - Page 24
Chapter 7 Notes - Page 27
Chapter 7 Notes - Page 29
Chapter 7 Notes - Page 31
natural join query in Access
Supplier names who ship ...
Chapter 7 Notes - Page 31
SELECT Shipments.pnum, SUM(Shipments.quantity) AS TotalQuantityShipped
FROM Shipments
GROUP BY pnum
HAVING SUM(Shipments.quantity) > (SELECT AVG(Shipments.quantity)
FROM Shipments)
SQL query entered in Access and executed showing results.
Chapter 7 Notes - Page 31
pnum
TotalQuantityShipped
3
48
4
15
9
25
Record: 1 of 3
No Filter
Search
Jobs
Ready
Num Lock

Subquery Example 4 (From clause)

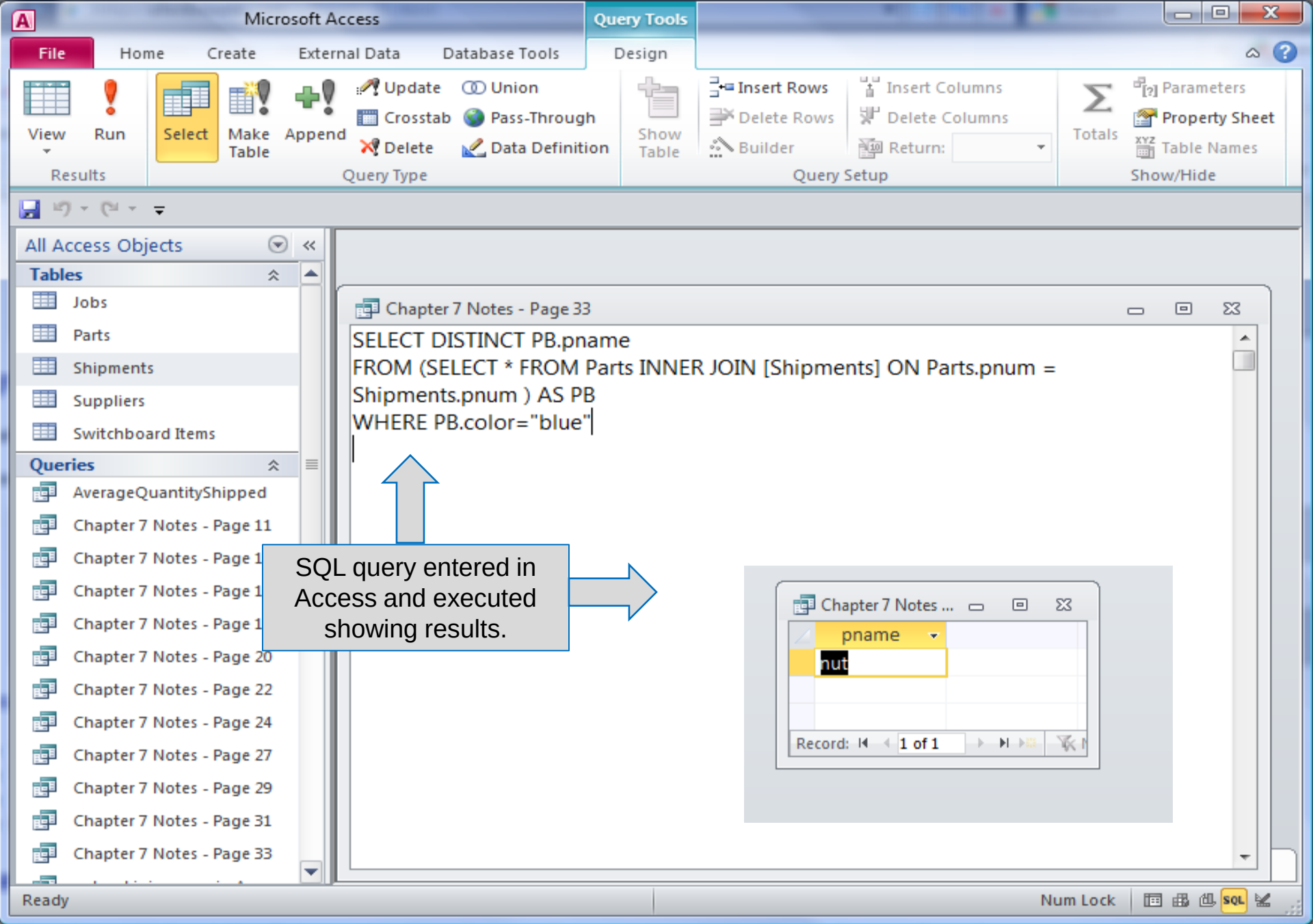
- List the unique part names for all the blue parts that are shipped.

SQL Version:

```
SELECT DISTINCT pname
FROM Shipments NATURAL JOIN (SELECT pnum
                             FROM Parts
                             WHERE color = "blue");
```

Access Version:

```
SELECT DISTINCT PB.pname
FROM (SELECT * FROM Parts INNER JOIN [Shipments]
      ON Parts.pnum = Shipments.pnum ) AS PB
WHERE PB.color="blue"
```



Microsoft Access
Query Tools
File
Home
Create
External Data
Database Tools
Design
View
Run
Select
Make Table
Append
Update
Union
Crosstab
Pass-Through
Delete
Data Definition
Results
Query Type
Show Table
Insert Rows
Delete Rows
Builder
Insert Columns
Delete Columns
Return:
Query Setup
Totals
Parameters
Property Sheet
Table Names
Show/Hide
All Access Objects
Tables
Jobs
Parts
Shipments
Suppliers
Switchboard Items
Queries
AverageQuantityShipped
Chapter 7 Notes - Page 11
Chapter 7 Notes - Page 1
Chapter 7 Notes - Page 1
Chapter 7 Notes - Page 1
Chapter 7 Notes - Page 20
Chapter 7 Notes - Page 22
Chapter 7 Notes - Page 24
Chapter 7 Notes - Page 27
Chapter 7 Notes - Page 29
Chapter 7 Notes - Page 31
Chapter 7 Notes - Page 33
Chapter 7 Notes - Page 33
SELECT DISTINCT PB.pname
FROM (SELECT * FROM Parts INNER JOIN [Shipments] ON Parts.pnum =
Shipments.pnum) AS PB
WHERE PB.color="blue"
SQL query entered in Access and executed showing results.
Chapter 7 Notes ...
pname
nut
Record: 1 of 1
Ready
Num Lock
SQL

Processing a noncorrelated subquery

1. The subquery executes and returns the customer IDs from the ORDER_T table
2. The outer query on the results of the subquery

```
SELECT CUSTOMER_NAME
        FROM CUSTOMER_T
            WHERE CUSTOMER_ID IN
```

```
(SELECT DISTINCT CUSTOMER_ID
FROM ORDER_T);
```

1. The subquery (shown in the box) is processed first and an intermediate results table created:

CUSTOMER_ID
1
8
15
5
3
2
11
12
4

9 rows selected.

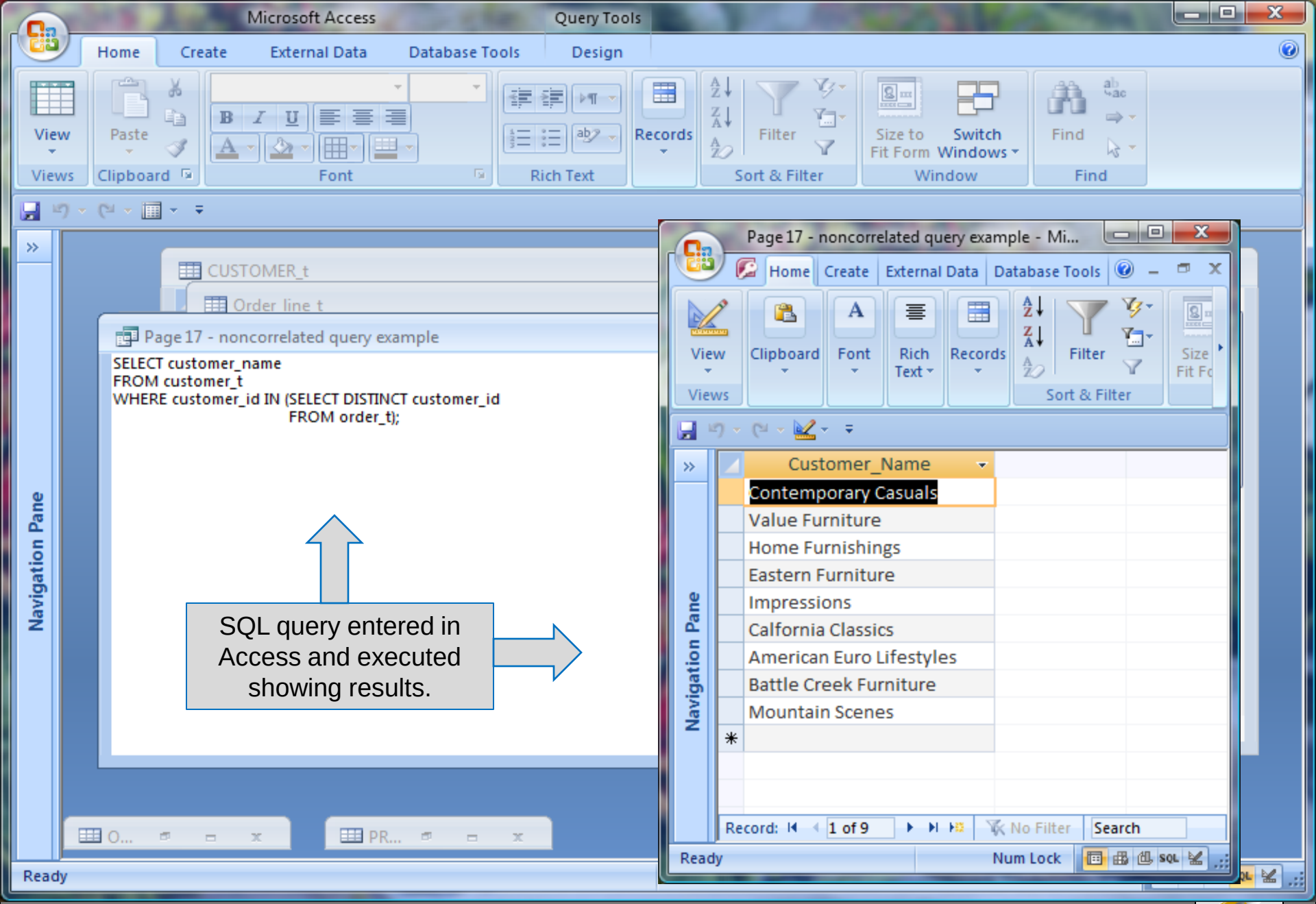
No reference to data in outer query, so subquery executes once only

2. The outer query returns the requested customer information for each customer included in the intermediate results table:

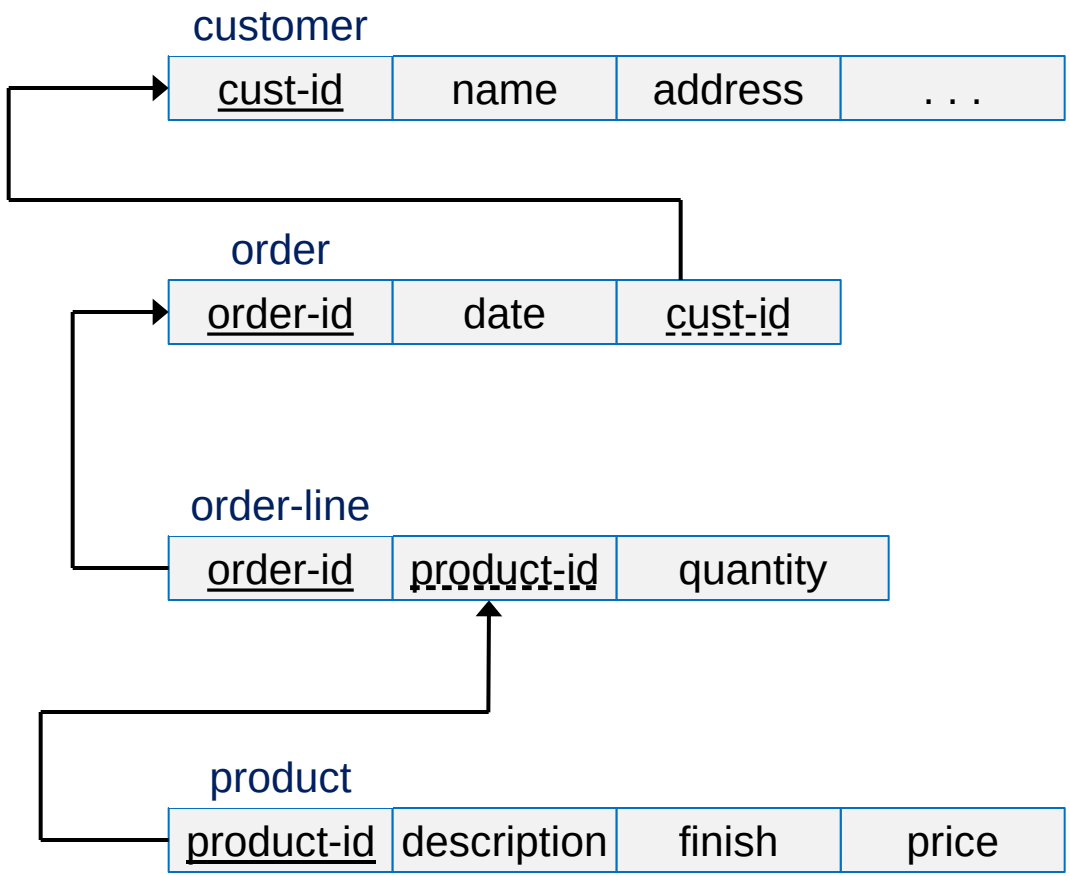
CUSTOMER_NAME
Contemporary Casuals
Value Furniture
Home Furnishings
Eastern Furniture
Impressions
California Classics
American Euro Lifestyles
Battle Creek Furniture
Mountain Scenes

9 rows selected.

These are the only customers that have IDs in the ORDER_T table


Microsoft Access
Query Tools
Home
Create
External Data
Database Tools
Design
Page 17 - noncorrelated query example
SELECT customer_name
FROM customer_t
WHERE customer_id IN (SELECT DISTINCT customer_id
FROM order_t);
SQL query entered in Access and executed showing results.
Page 17 - noncorrelated query example - Mi...
Customer_Name
Contemporary Casuals
Value Furniture
Home Furnishings
Eastern Furniture
Impressions
Calfornia Classics
American Euro Lifestyles
Battle Creek Furniture
Mountain Scenes
Navigation Pane
Ready
Num Lock

Correlated Subquery Example



Use this database for the next couple of examples.

Correlated Subquery Example

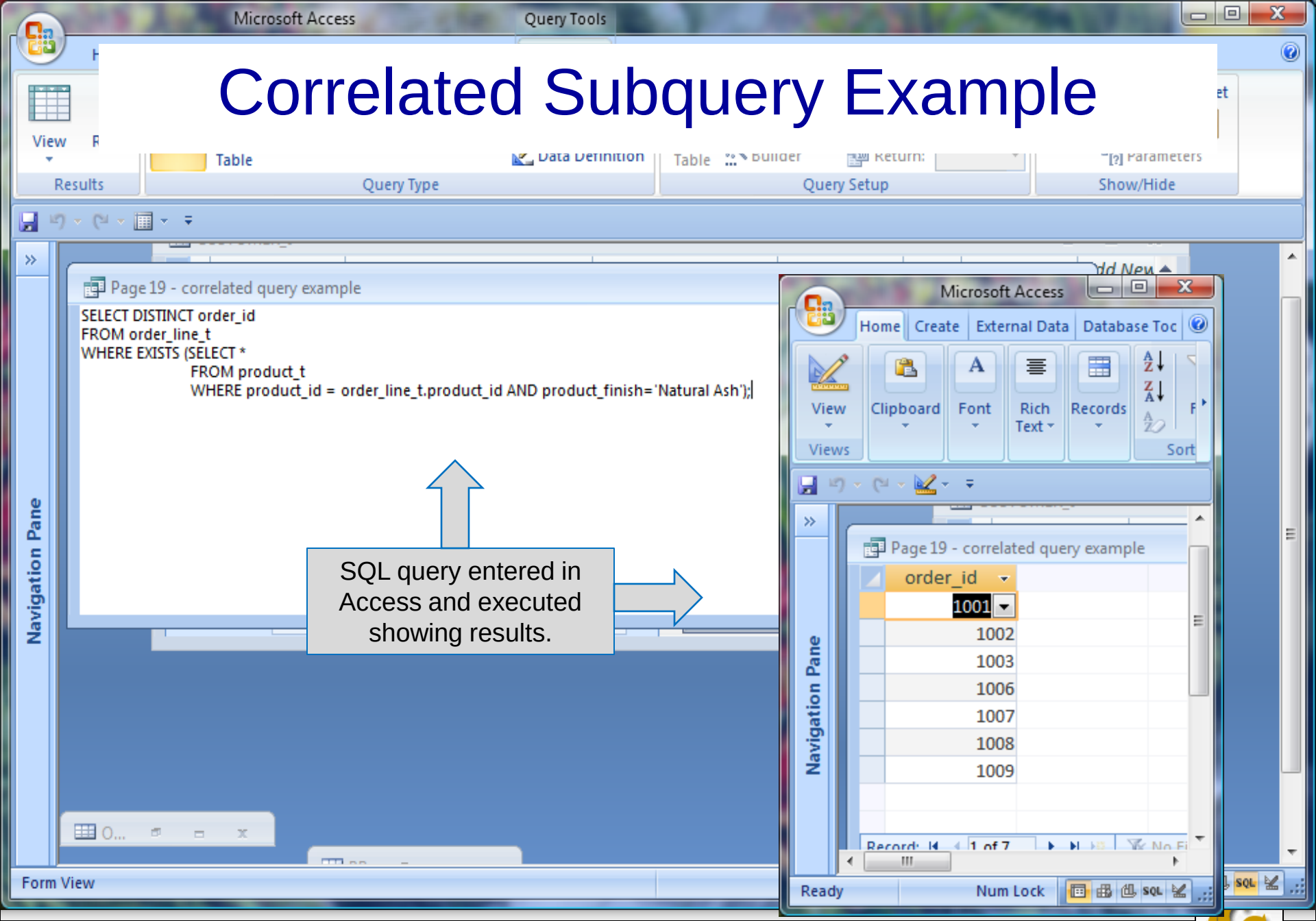
- Show all orders that include products finished in natural ash

The EXISTS operator will return a TRUE value if the subquery resulted in a non-empty set, otherwise it returns a FALSE

```
SELECT DISTINCT order-id FROM order-line
WHERE  EXISTS
   (SELECT * FROM product
    WHERE product-id = order-line.product-id
    AND finish = 'Natural ash');
```

The subquery is testing for a value that comes from the outer query

Correlated Subquery Example



Another Subquery Example

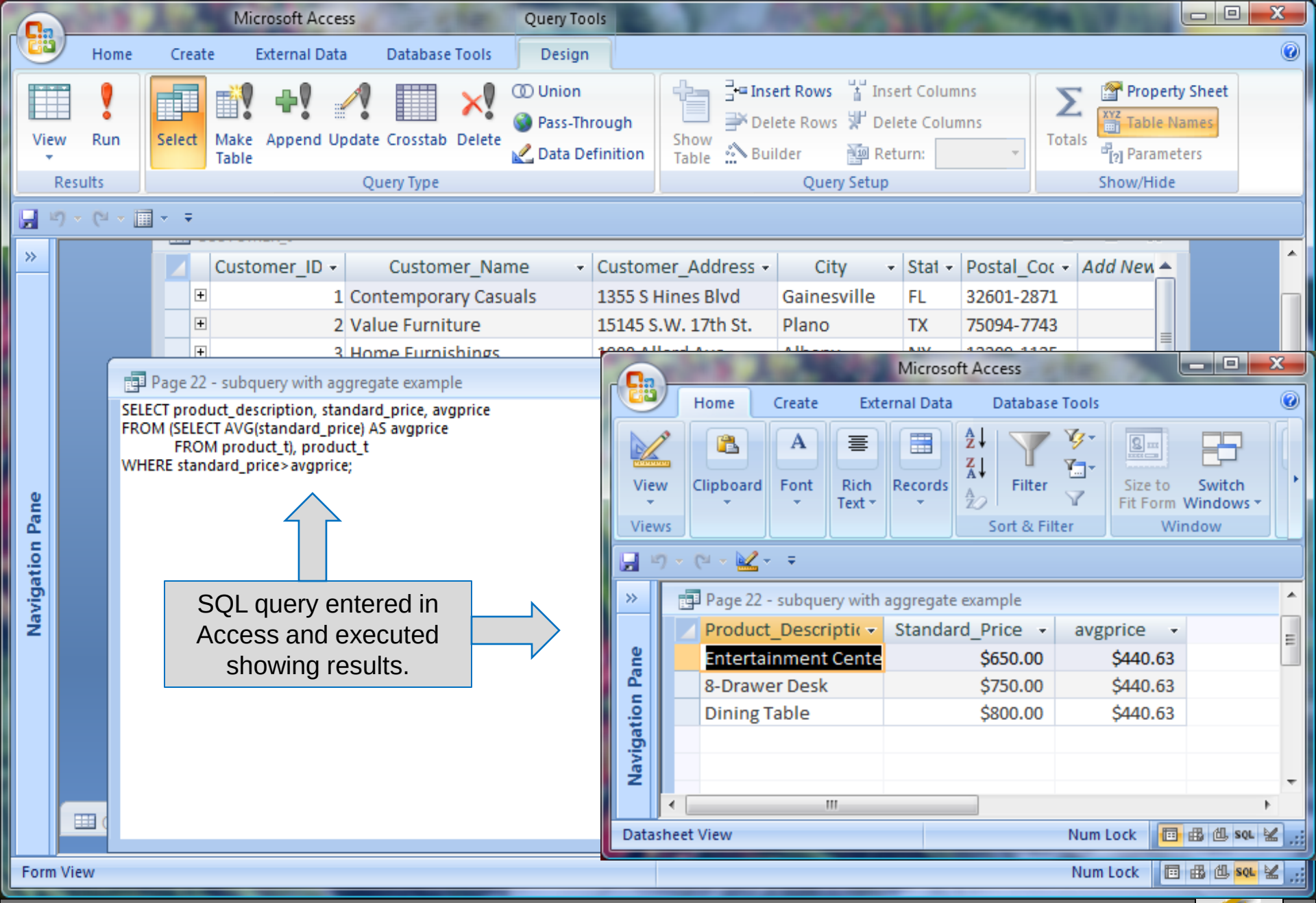
- Show all products whose price is higher than the average

Subquery forms the derived table used in the FROM clause of the outer query

One column of the subquery is an aggregate function that has an alias name. That alias can then be referred to in the outer query

```
SELECT description, price, avg-price
FROM
 (SELECT AVG(price) AS avgprice FROM product), product
WHERE price > avgprice;
```

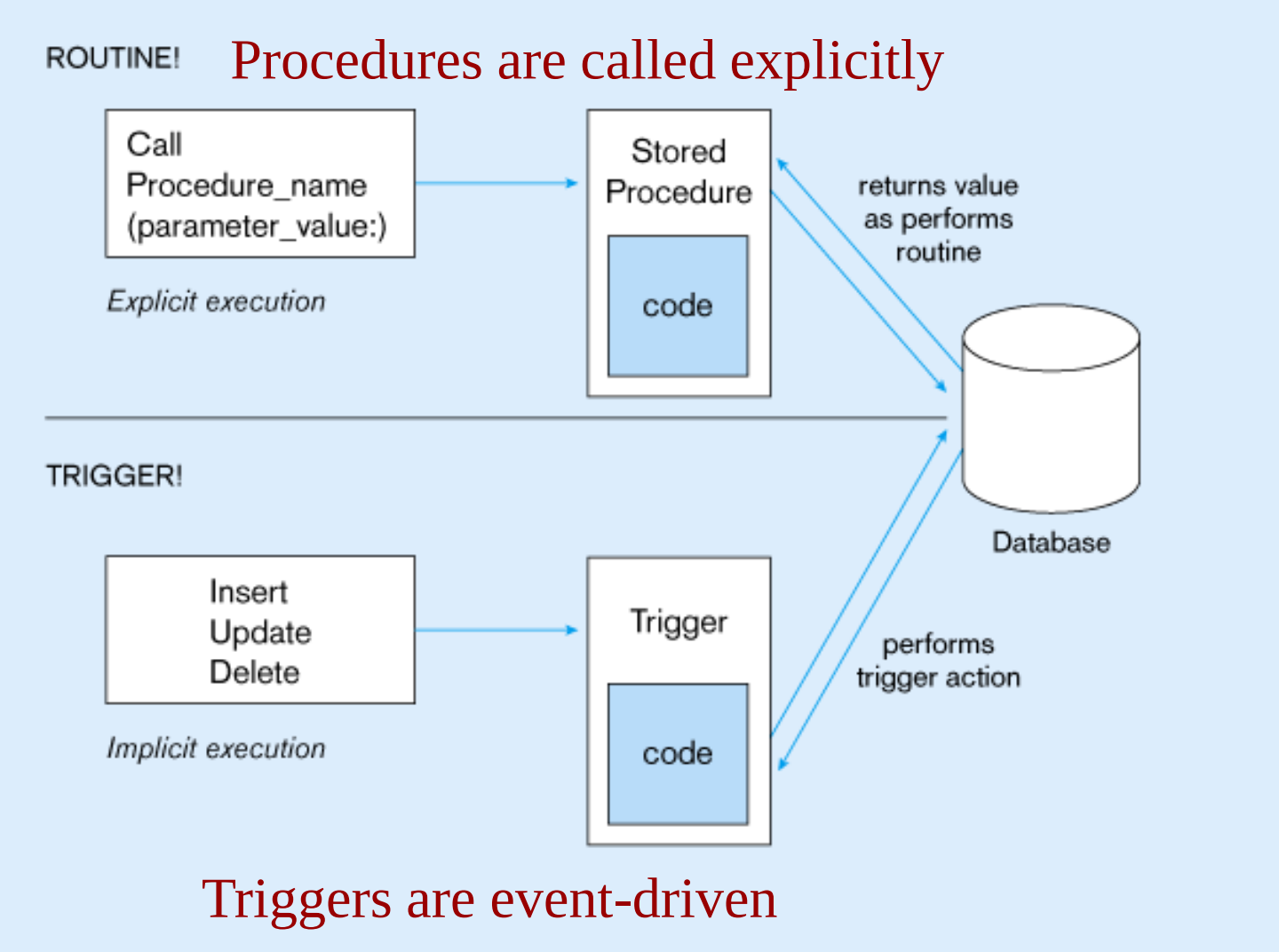
The WHERE clause normally cannot include aggregate functions, but because the aggregate is performed in the subquery its result can be used in the outer query's WHERE clause


Microsoft Access
Query Tools
Home
Create
External Data
Database Tools
Design
View
Run
Select
Make Table
Append
Update
Crosstab
Delete
Union
Pass-Through
Data Definition
Insert Rows
Delete Rows
Builder
Insert Columns
Delete Columns
Return:
Show Table
Totals
Property Sheet
Table Names
Parameters
Results
Query Type
Query Setup
Show/Hide
Customer_ID
Customer_Name
Customer_Address
City
Stat
Postal_Coc
Add New
1 Contemporary Casuals 1355 S Hines Blvd Gainesville FL 32601-2871
2 Value Furniture 15145 S.W. 17th St. Plano TX 75094-7743
3 Home Furnishings
Navigation Pane
Page 22 - subquery with aggregate example
SELECT product_description, standard_price, avgprice
FROM (SELECT AVG(standard_price) AS avgprice
FROM product_t), product_t
WHERE standard_price>avgprice;
SQL query entered in Access and executed showing results.
Microsoft Access
Home
Create
External Data
Database Tools
View
Clipboard
Font
Rich Text
Records
Filter
Size to Fit Form
Switch Windows
Views
Sort & Filter
Window
Page 22 - subquery with aggregate example
Product_Descripti
Standard_Price
avgprice
Entertainment Cente $650.00 $440.63
8-Drawer Desk $750.00 $440.63
Dining Table $800.00 $440.63
Navigation Pane
Datasheet View
Num Lock
SQL
Form View
Num Lock
SQL

Routines and Triggers

- **Routines**
  - Program modules that execute on demand
  - **Functions** – routines that return values and take input parameters
  - **Procedures** – routines that do not return values and can take input or output parameters
- **Triggers**
  - Routines that execute in response to a database event (INSERT, UPDATE, or DELETE)

Triggers contrasted with stored procedures



Oracle PL/SQL trigger syntax

```
CREATE [OR REPLACE] TRIGGER trigger_name
        {BEFORE AFTER} {INSERT | DELETE | UPDATE} ON table_name
        [FOR EACH ROW [WHEN (trigger_condition)]]
        trigger_body_here;
```

SQL:20XX Create routine syntax

```
{CREATE PROCEDURE | CREATE FUNCTION} routine_name
([parameter [{,parameter} . . .]])
[RETURNS data_type result_cast]    /* for functions only */
[LANGUAGE {ADA | C | COBOL | FORTRAN | MUMPS | PASCAL | PLI | SQL}]
[PARAMETER STYLE {SQL | GENERAL}]
[SPECIFIC specific_name]
[DETERMINISTIC | NOT DETERMINISTIC]
[NO SQL | CONTAINS SQL | READS SQL DATA | MODIFIES SQL DATA]
[RETURN NULL ON NULL INPUT | CALL ON NULL INPUT]
[DYNAMIC RESULT SETS unsigned_integer]      /* for procedures only */
[STATIC DISPATCH]                           /* for functions only */
routine_body
```


Embedded and Dynamic SQL

- Embedded SQL
  - Including hard-coded SQL statements in a program written in another language such as C or Java
- Dynamic SQL
  - Ability for an application program to generate SQL code on the fly, as the application is running

Suppliers Table Instance

Suppliers

snum	sname	status	city	Click to Ad
1	Mark	4	Oviedo	
2	Dave	30	Orlando	
3	Tiffany	2	Winter Springs	
4	Kristi	1	Orlando	
5	Karen	3	Longwood	
6	Cat	4	Oviedo	
7	Tami	3	Winter Springs	
8	Cindy	2	Tampa	
9	Candace	17	London	
(New)		0		

Record: 1 of 9 No Filter Search

Field Name	Data Type
snum	AutoNumber
sname	Text
status	Number
city	Text

Parts Table Instance

pnum	pname	color	weight	city
3	bolt	red	3	Orlando
4	nut	blue	14	Tampa
5	flange	red	7	Miami
6	clamp	black	3	Orlando
7	nut	red	4	Orlando
8	nut	blue	5	Tampa
9	switch	green	3	Oviedo
10	gasket	brown	1	Tampa
(New)			0	

Field Name	Data Type
pnum	AutoNumber
pname	Text
color	Text
weight	Number
city	Text

Jobs Table Instance

jnum	jname	numworkers	city
4	tiny job	1	Oviedo
5	small job	4	Tampa
6	bigger job	15	Jacksonville
7	huge job	45	Miami
(New)		0	

Field Name	Data Type
jnum	AutoNumber
jname	Text
numworkers	Number
city	Text

Shipments Table Instance

snum	pnum	jnum	quantity	shipment_ID	Cli
1	3	4	14	4	
2	8	4	1	5	
3	3	5	22	9	
3	9	7	25	7	
4	3	5	12	6	
5	4	4	15	8	
*	0	0	0	0	(New)

Field Name	Data Type
snum	Number
pnum	Number
jnum	Number
quantity	Number
shipment_ID	AutoNumber