

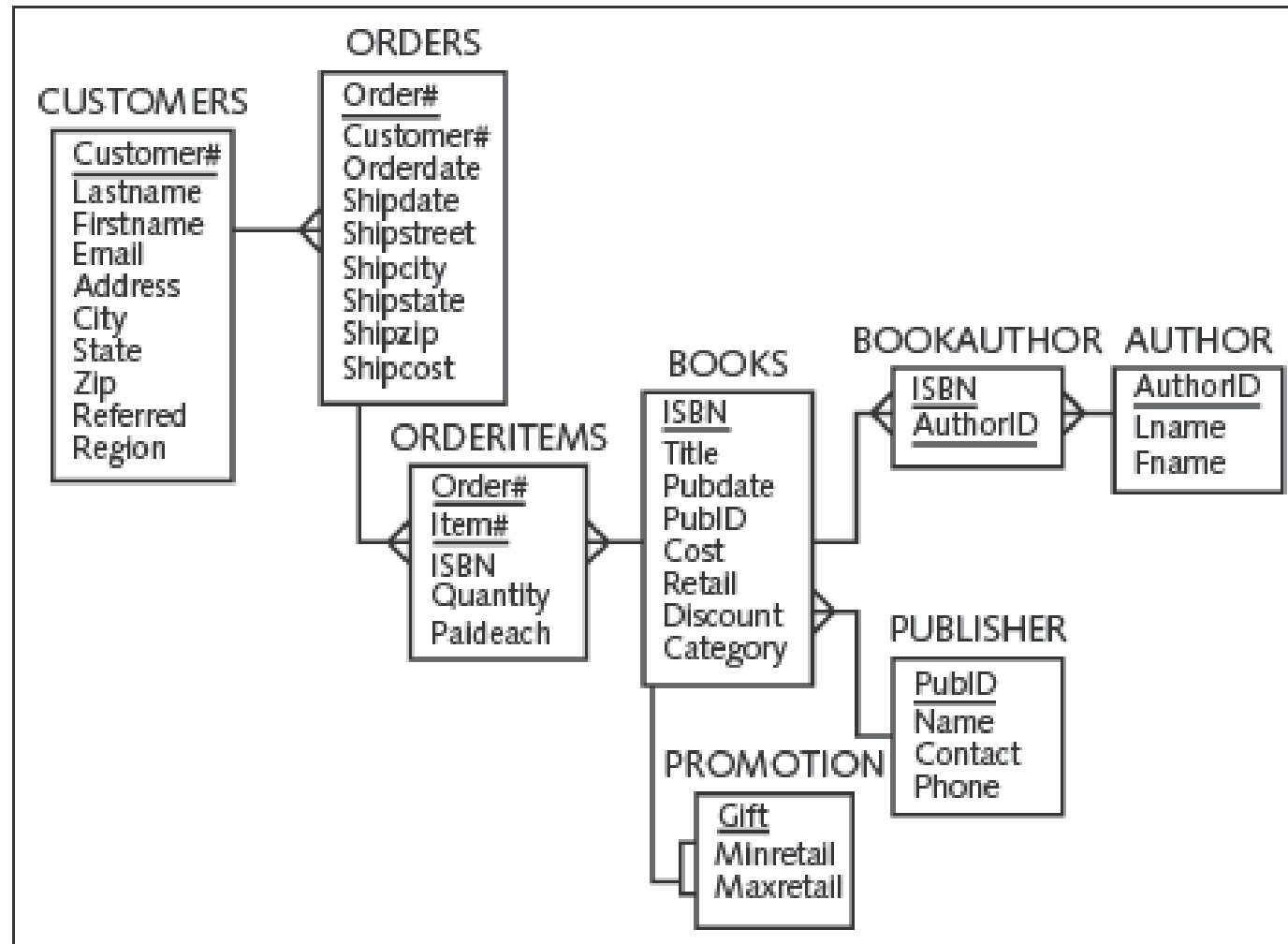
# Database Systems

## Restricting Rows and Sorting

CS 630 Database Systems

Professor Nardi

# Normalized JustLee Books Database...



# So Far...

- Give Me a List of All Customers...

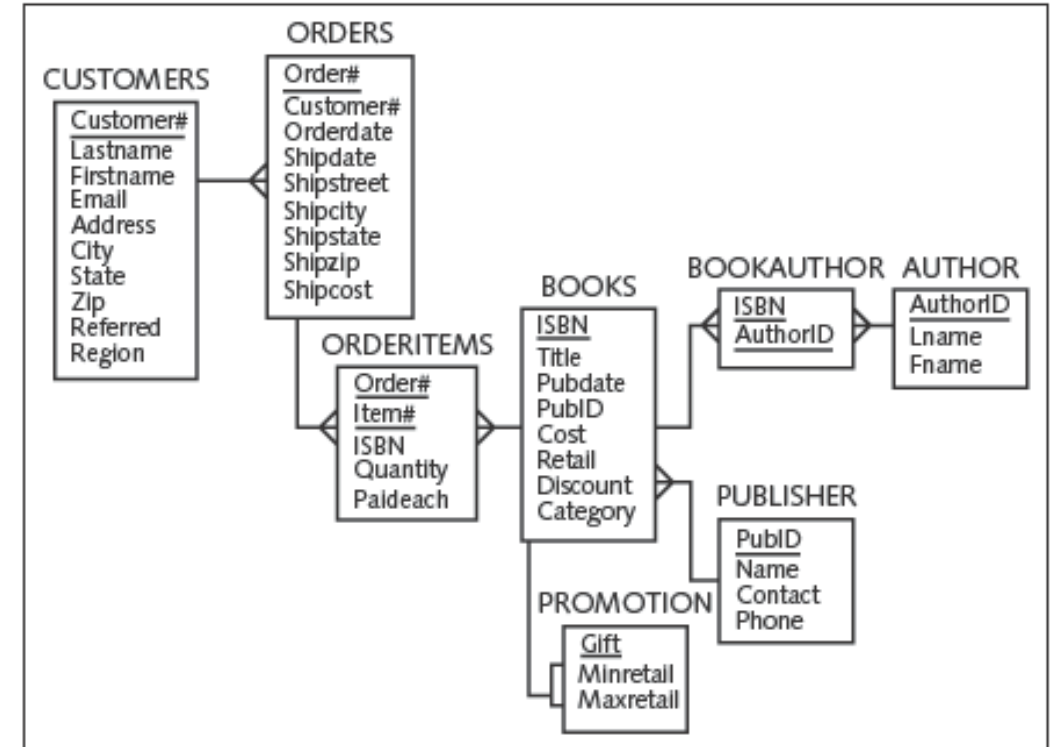
```
SELECT Lastname || ' ' || Firstname  
FROM Customers;
```

- Give Me a List of All Publisher Contacts...

```
SELECT Contact  
FROM Publisher;
```

- Give Me a List of Authors and Their IDs...

```
SELECT AuthorID, Lname, Fname  
FROM AUTHOR;
```

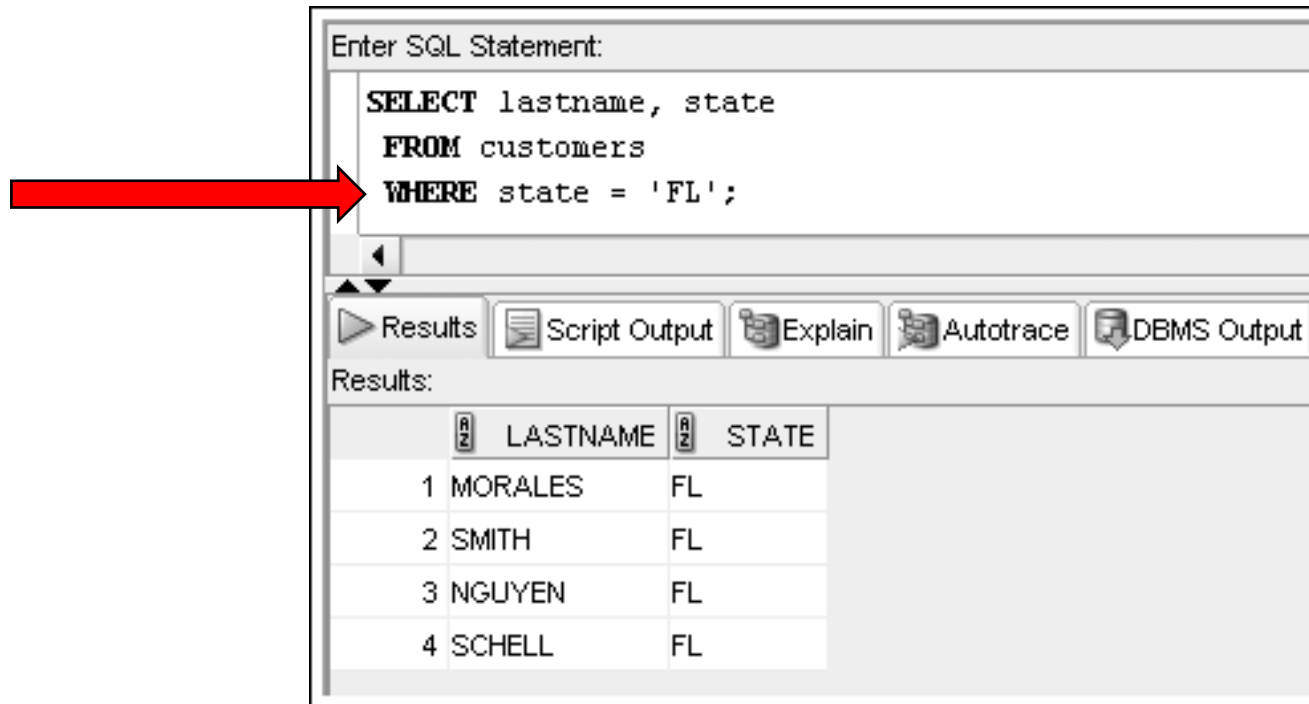


# WHERE Clause Syntax...

- WHERE Clause Used to Retrieve Rows Based on a Stated Condition...
- Requires:
  - Column Name...
  - Comparison Operator...
  - Value or Column For Comparison...
- Values Are Case Sensitive...

# WHERE Clause Example...

- List WHERE Clause After FROM Clause...
- Enclose Nonnumeric Data in Single Quotes...



The screenshot shows a SQL query editor window. The title bar is "Enter SQL Statement:". The text area contains the following SQL query:

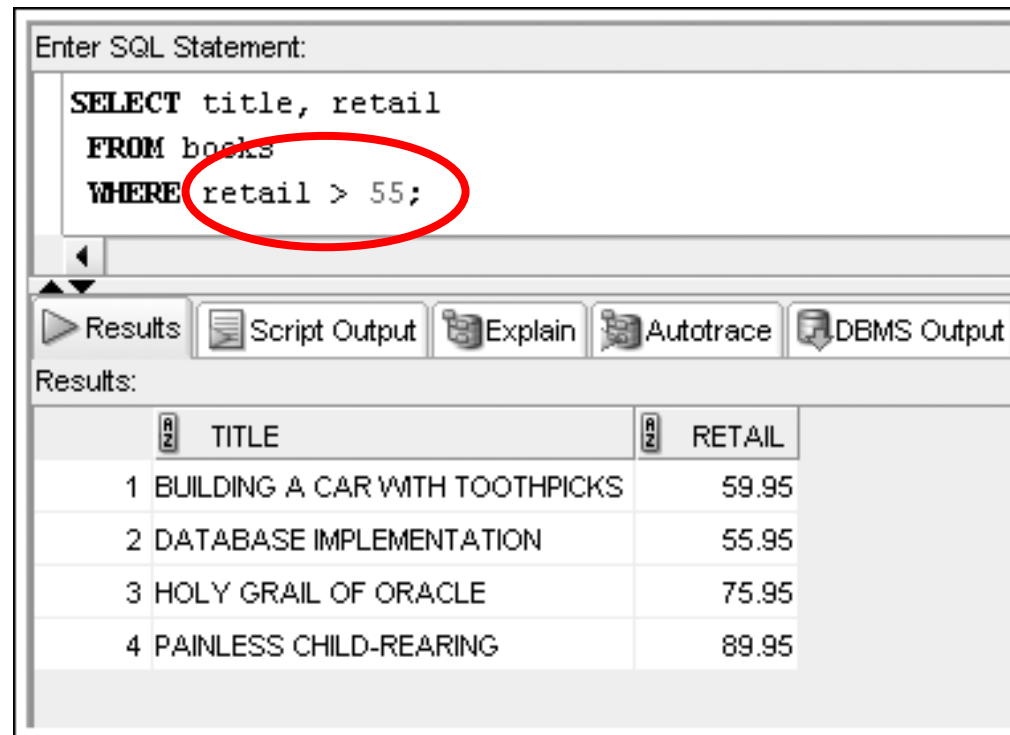
```
SELECT lastname, state
FROM customers
WHERE state = 'FL';
```

A red arrow points to the WHERE clause. Below the text area are five buttons: "Results", "Script Output", "Explain", "Autotrace", and "DBMS Output". The "Results" button is selected. Below the buttons, the "Results:" section displays a table with the following data:

	LASTNAME	STATE
1	MORALES	FL
2	SMITH	FL
3	NGUYEN	FL
4	SCHELL	FL

# Comparison Operators...

- Indicates How the Data Should Relate to the Given Search Value...



The screenshot shows a web-based SQL interface. At the top, there is a text area labeled "Enter SQL Statement:" containing the following query:

```
SELECT title, retail
FROM books
WHERE retail > 55;
```

The `WHERE` clause and the comparison operator `>` are circled in red. Below the text area is a row of buttons: "Results" (selected), "Script Output", "Explain", "Autotrace", and "DBMS Output". Below the buttons, the "Results:" section displays a table with two columns: "TITLE" and "RETAIL". The table contains four rows of data.

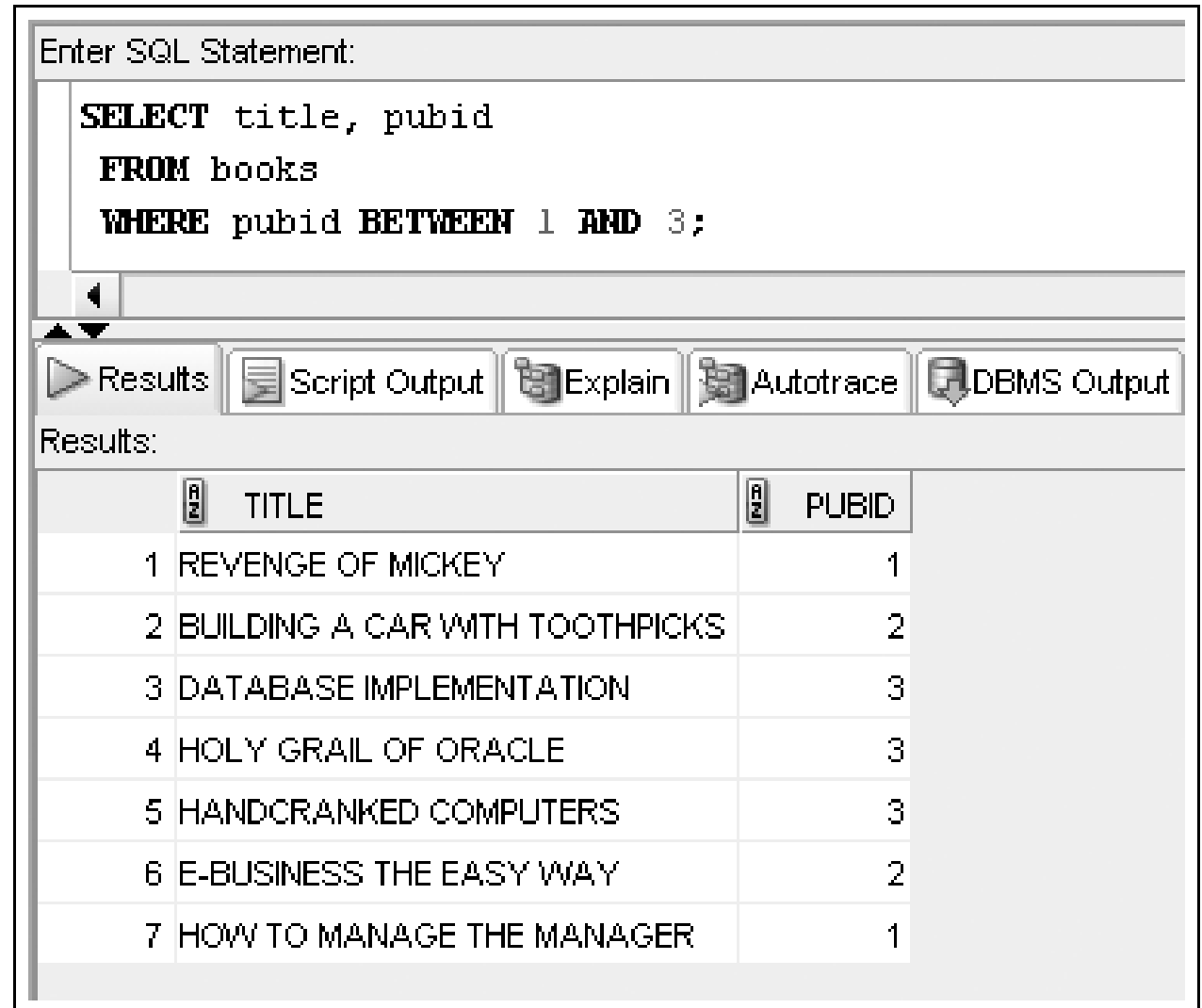
	TITLE	RETAIL
1	BUILDING A CAR WITH TOOTHPICKS	59.95
2	DATABASE IMPLEMENTATION	55.95
3	HOLY GRAIL OF ORACLE	75.95
4	PAINLESS CHILD-REARING	89.95

# Arithmetic Comparison Operators...

- = Equality or “Equal To”...For Example...Cost = 59.95...
- > Greater Than...For Example...Cost > 20...
- < Less Than...For Example...Cost < 20...
- <> Not Equal To...For Example...Cost <> 20...Can Also Use != or ^=...
- <= Less Than OR Equal To...For Example...Cost <= 20...
- >= Greater Than OR Equal To...For Example...Cost >= 20...

# Other Comparison Operators – BETWEEN...AND...

- Finds Values in a Specified Range...
- Range Is Inclusive...
- BETWEEN 5 AND 10...
- Can Also Use NOT BETWEEN...
- NOT BETWEEN 5 and 10...



The screenshot shows a SQL query execution window. The query entered is:

```
SELECT title, pubid
FROM books
WHERE pubid BETWEEN 1 AND 3;
```

The results are displayed in a table with two columns: TITLE and PUBID. The table contains 7 rows of data.

	TITLE	PUBID
1	REVENGE OF MICKEY	1
2	BUILDING A CAR WITH TOOTHPICKS	2
3	DATABASE IMPLEMENTATION	3
4	HOLY GRAIL OF ORACLE	3
5	HANDCRANKED COMPUTERS	3
6	E-BUSINESS THE EASY WAY	2
7	HOW TO MANAGE THE MANAGER	1



# Other Comparison Operators – IN...

- Returns Records That Match a Value in a Specified List...
- List Must Be in Parentheses...
- Values Are Separated by Commas...
- Can Also Use NOT IN...
- Show Books NOT Published by Publishers 1 or 3 or 4...

Enter SQL Statement:

```
SELECT title, pubid  
FROM books  
WHERE pubid IN (1,2,5);
```

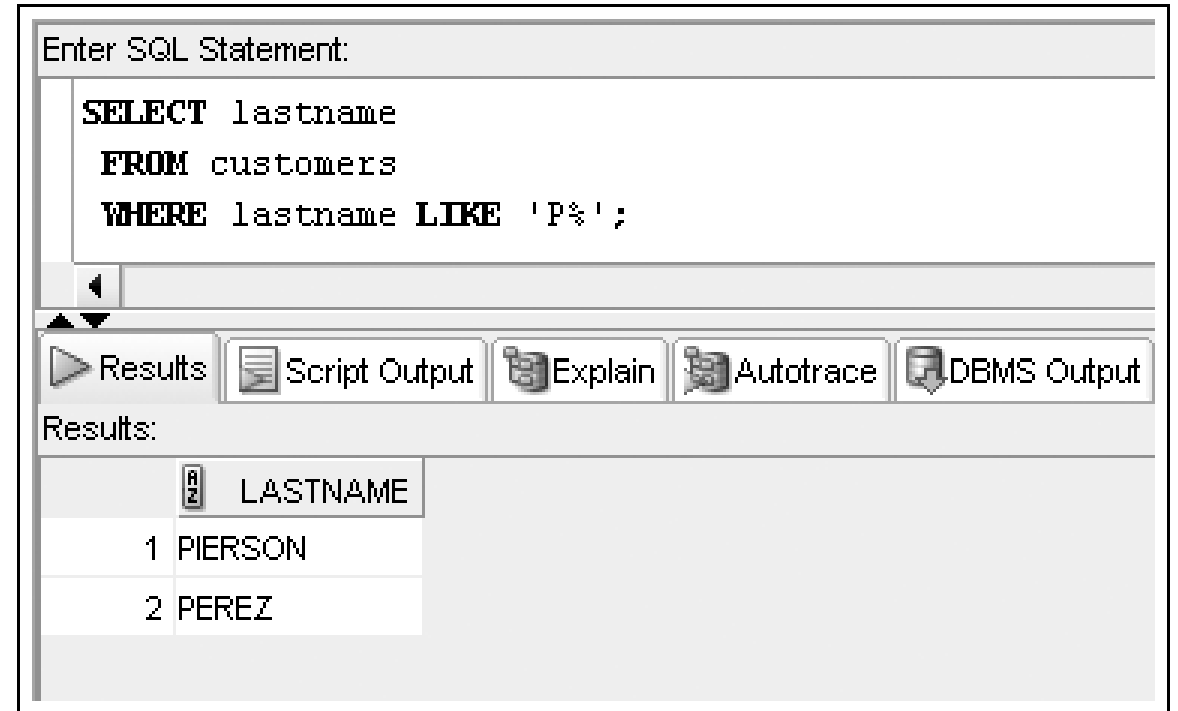
Results Script Output Explain Autotrace DBMS Output

Results:

	R 2	TITLE	R 2	PUBID
1		REVENGE OF MICKEY		1
2		BUILDING A CAR WITH TOOTHPICKS		2
3		E-BUSINESS THE EASY WAY		2
4		PAINLESS CHILD-REARING		5
5		BIG BEAR AND LITTLE DOVE		5
6		HOW TO MANAGE THE MANAGER		1
7		SHORTEST POEMS		5

# Other Comparison Operators – LIKE...

- Performs Pattern Searches...
- Used With Wildcard Characters...
  - Underscore (\_) For Exactly One Character In The Indicated Position...
  - Percent Sign (%) Represents Any Number Of Characters...
- Can Also Use NOT LIKE...
- Returns Records That Do NOT Contain a Specific Pattern...

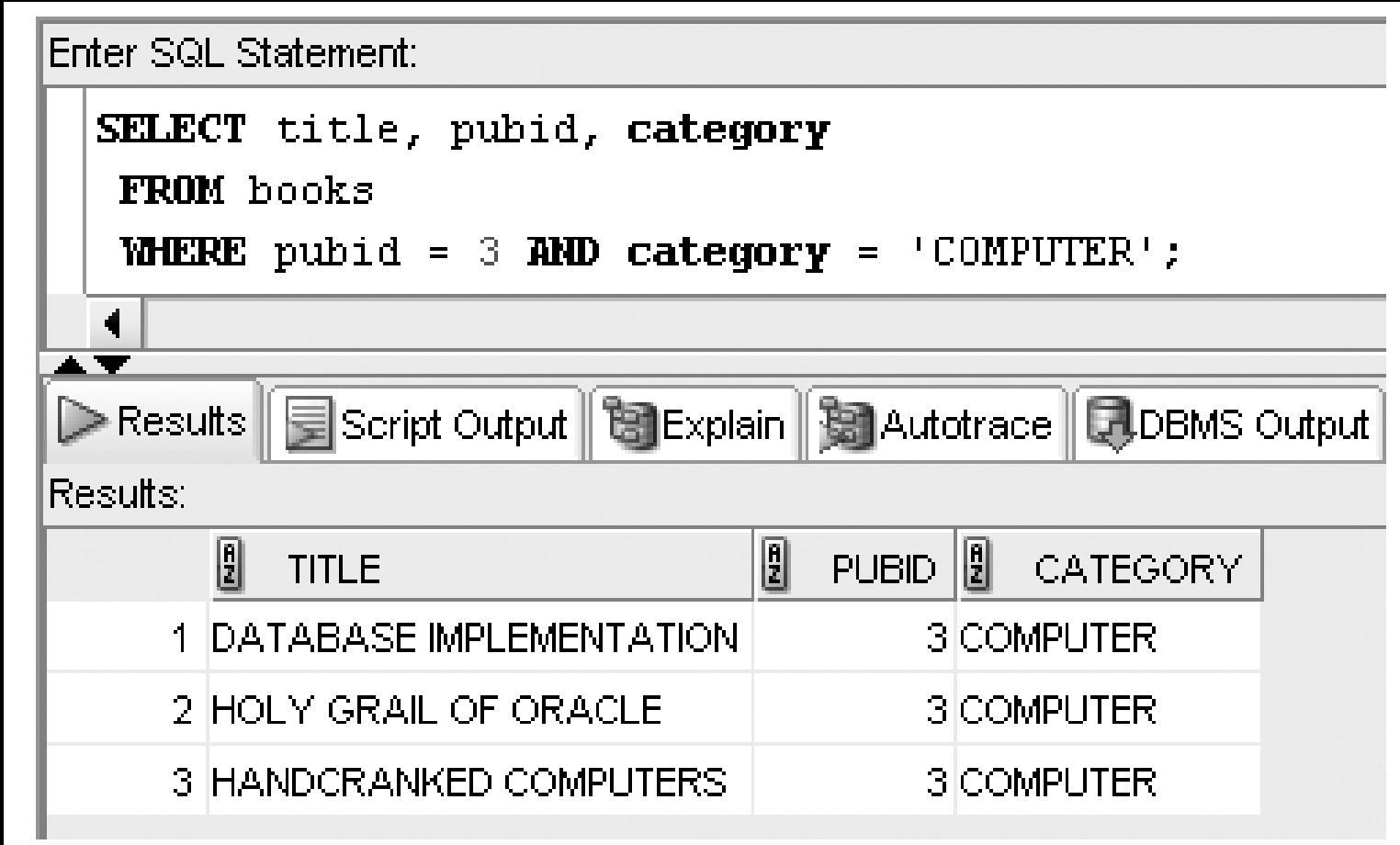


# Logical Operators...

- Used to Combine Conditions...
- Evaluated in Order of NOT, AND, OR...
  - NOT – Reverses Meaning...
  - AND – Both Conditions MUST Be TRUE...
  - OR – At Least One Condition Must Be True...

# AND Logical Operator Example...




- BOTH Conditions Must Be True...
- PUBID MUST Be 3 AND CATEGORY MUST Be COMPUTER...



The screenshot shows a web-based SQL interface. At the top, there is a text area labeled "Enter SQL Statement:" containing the following query:

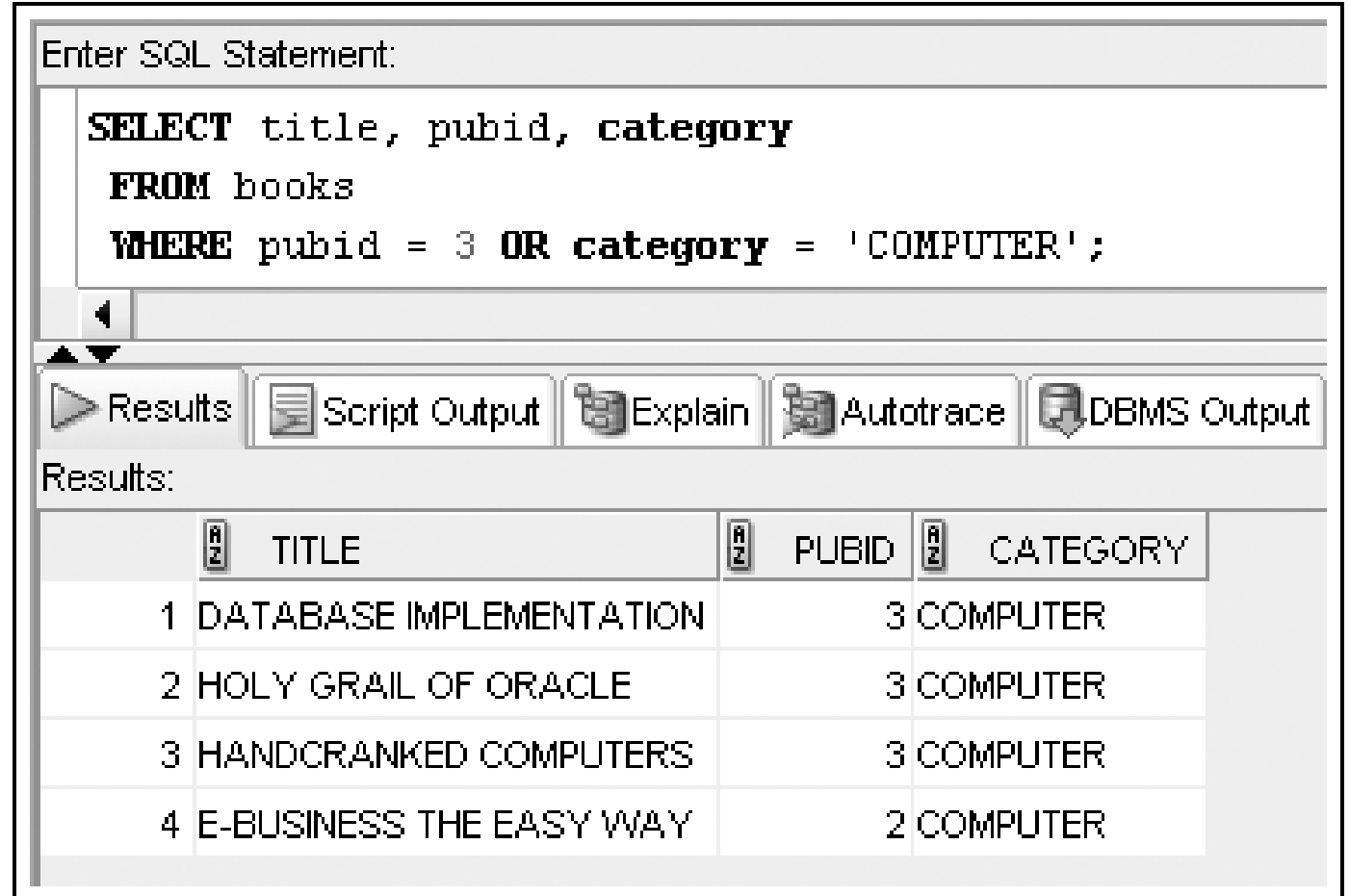
```
SELECT title, pubid, category
FROM books
WHERE pubid = 3 AND category = 'COMPUTER';
```

Below the text area is a row of buttons: "Results" (selected), "Script Output", "Explain", "Autotrace", and "DBMS Output". Below the buttons, the "Results:" section displays a table with the following data:

	 TITLE	 PUBID	 CATEGORY
1	DATABASE IMPLEMENTATION	3	COMPUTER
2	HOLY GRAIL OF ORACLE	3	COMPUTER
3	HANDCRANKED COMPUTERS	3	COMPUTER

# OR Logical Operator Example...

- One of the Conditions Must Be True...
- PUBID Must Be 3 OR CATEGORY Must Be COMPUTER...



The screenshot shows a database query interface. At the top, there is a text box labeled "Enter SQL Statement:" containing the following SQL code:

```
SELECT title, pubid, category
FROM books
WHERE pubid = 3 OR category = 'COMPUTER';
```

Below the text box is a toolbar with five buttons: "Results" (selected), "Script Output", "Explain", "Autotrace", and "DBMS Output". Below the toolbar, the word "Results:" is displayed. Below "Results:" is a table with three columns: "TITLE", "PUBID", and "CATEGORY". The table contains four rows of data:

	TITLE	PUBID	CATEGORY
1	DATABASE IMPLEMENTATION	3	COMPUTER
2	HOLY GRAIL OF ORACLE	3	COMPUTER
3	HANDCRANKED COMPUTERS	3	COMPUTER
4	E-BUSINESS THE EASY WAY	2	COMPUTER

# Multiple Logical Operators Example...

- Resolved in Order of NOT, AND, OR

Enter SQL Statement:

```
SELECT *  
FROM books  
WHERE category = 'FAMILY LIFE'  
OR pubid = 4  
AND cost > 15;
```

Results Script Output Explain Autotrace DBMS Output OWA Output

Results:

	ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	DISCOUNT	CATEGORY
1	1059831198	BODYBUILD IN 10 MINUTES A DAY	21-JAN-05	4	18.75	30.95	(null)	FITNESS
2	0401140733	REVENGE OF MICKEY	14-DEC-05	1	14.2	22	(null)	FAMILY LIFE
3	2491748320	PAINLESS CHILD-REARING	17-JUL-04	5	48	89.95	4.5	FAMILY LIFE
4	0299282519	THE WOK WAY TO COOK	11-SEP-04	4	19	28.75	(null)	COOKING
5	0132149871	HOW TO GET FASTER PIZZA	11-NOV-06	4	17.85	29.95	1.5	SELF HELP

# Multiple Logical Operators Example...

- Resolved in Order of NOT, AND, OR

Enter SQL Statement:

```
SELECT *  
FROM books  
WHERE category = 'FAMILY LIFE'  
OR pubid = 4  
AND cost > 15;
```

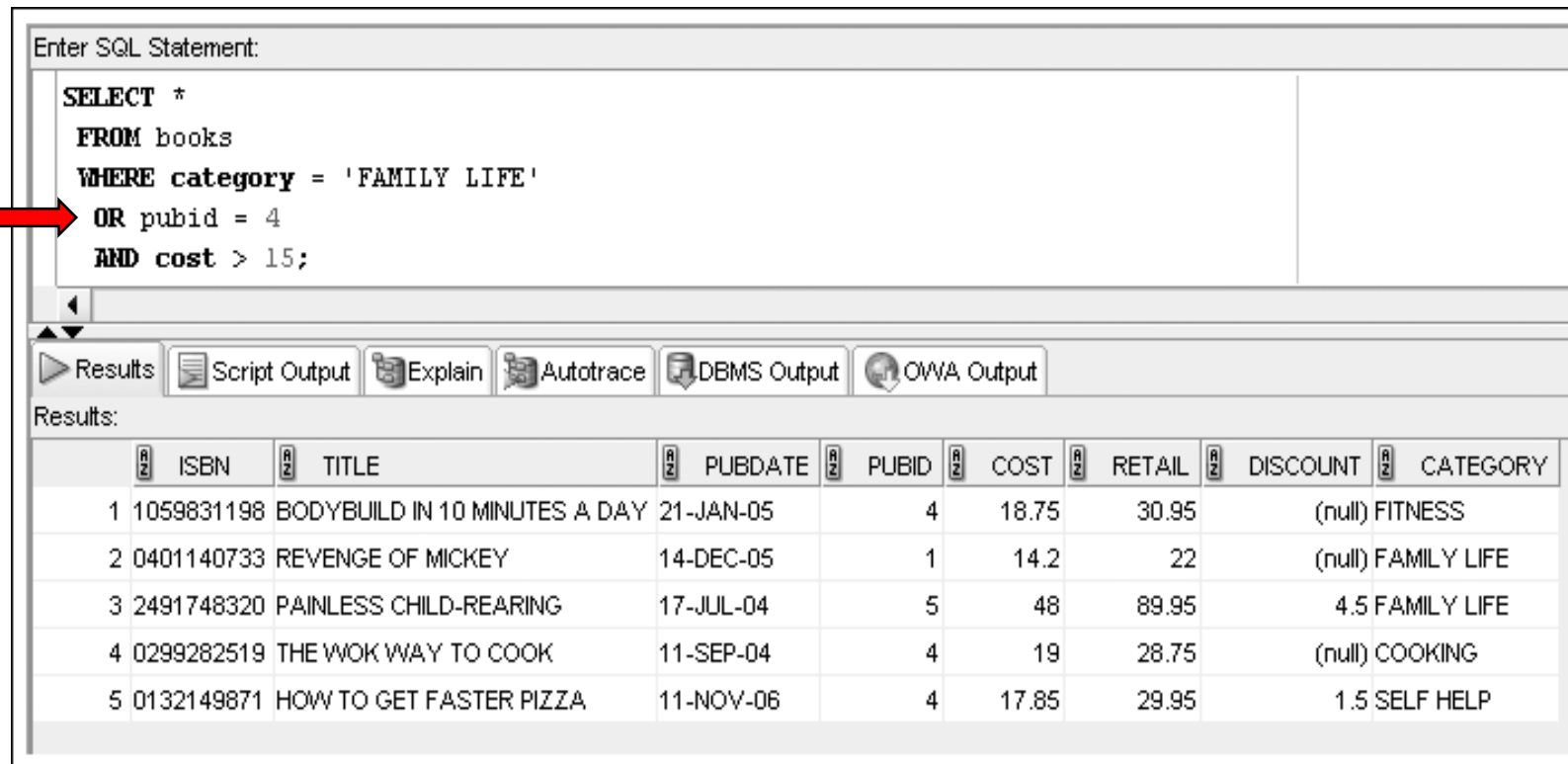
Results Script Output Explain Autotrace DBMS Output OWA Output

Results:

	ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	DISCOUNT	CATEGORY
1	1059831198	BODYBUILD IN 10 MINUTES A DAY	21-JAN-05	4	18.75	30.95	(null)	FITNESS
2	0401140733	REVENGE OF MICKEY	14-DEC-05	1	14.2	22	(null)	FAMILY LIFE
3	2491748320	PAINLESS CHILD-REARING	17-JUL-04	5	48	89.95	4.5	FAMILY LIFE
4	0299282519	THE WOK WAY TO COOK	11-SEP-04	4	19	28.75	(null)	COOKING
5	0132149871	HOW TO GET FASTER PIZZA	11-NOV-06	4	17.85	29.95	1.5	SELF HELP

# Multiple Logical Operators Example...

- Resolved in Order of NOT, AND, OR



The screenshot shows an SQL query execution interface. A red arrow points to the 'OR' operator in the query. The query is as follows:

```
Enter SQL Statement:  
  
SELECT *  
FROM books  
WHERE category = 'FAMILY LIFE'  
OR pubid = 4  
AND cost > 15;
```

Below the query, there are tabs for 'Results', 'Script Output', 'Explain', 'Autotrace', 'DBMS Output', and 'OWA Output'. The 'Results' tab is selected, showing a table of results.

	ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	DISCOUNT	CATEGORY
1	1059831198	BODYBUILD IN 10 MINUTES A DAY	21-JAN-05	4	18.75	30.95	(null)	FITNESS
2	0401140733	REVENGE OF MICKEY	14-DEC-05	1	14.2	22	(null)	FAMILY LIFE
3	2491748320	PAINLESS CHILD-REARING	17-JUL-04	5	48	89.95	4.5	FAMILY LIFE
4	0299282519	THE WOK WAY TO COOK	11-SEP-04	4	19	28.75	(null)	COOKING
5	0132149871	HOW TO GET FASTER PIZZA	11-NOV-06	4	17.85	29.95	1.5	SELF HELP



# Multiple Logical Operators Example...

- Use Parentheses to Override the Order of Evaluation...
- ALWAYS USE PARENTHESIS...LEAVE NOTHING TO CHANCE!!!...

Enter SQL Statement:

```
SELECT *  
FROM books  
WHERE (category = 'FAMILY LIFE'  
OR pubid = 4)  
AND cost > 15;
```

Results Script Output Explain Autotrace DBMS Output OWA Output

Results:

	ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	DISCOUNT	CATEGORY
1	1059831198	BODYBUILD IN 10 MINUTES A DAY	21-JAN-05	4	18.75	30.95	(null)	FITNESS
2	2491748320	PAINLESS CHILD-REARING	17-JUL-04	5	48	89.95	4.5	FAMILY LIFE
3	0299282519	THE WOK WAY TO COOK	11-SEP-04	4	19	28.75	(null)	COOKING
4	0132149871	HOW TO GET FASTER PIZZA	11-NOV-06	4	17.85	29.95	1.5	SELF HELP

# Treatment of NULL Values – Part 1...

- AGAIN...NULL is the Absence of Data...
- Requires Use of IS NULL Operator...
- Can Also Use IS NOT NULL...

The screenshot shows a SQL query execution window. The query entered is:

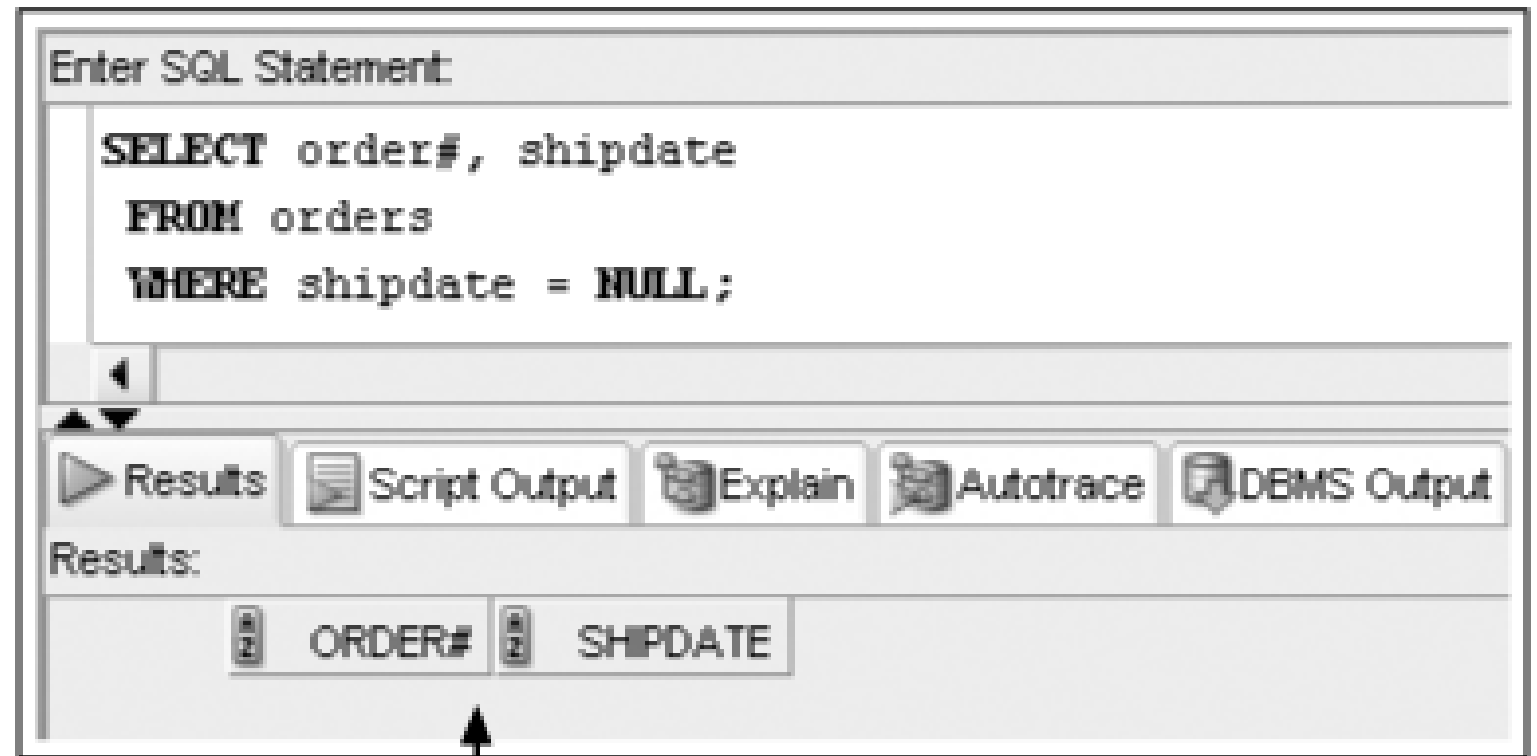
```
SELECT order#, shipdate  
FROM orders  
WHERE shipdate IS NULL;
```

Below the query, there are buttons for 'Results', 'Script Output', 'Explain', 'Autotrace', and 'DBMS Output'. The 'Results' button is selected, and the results are displayed in a table:

	ORDER#	SHIPDATE
1	1012	(null)
2	1015	(null)
3	1016	(null)
4	1018	(null)
5	1019	(null)
6	1020	(null)

# Treatment of NULL Values – Part 2...

- A Common Error Is Using “= NULL”...
- This Does NOT Return an Oracle Error...But Also Does Not Return Any Rows...



No rows returned

# ORDER BY Clause Syntax...

- ORDER BY Clause Presents Data in Sorted Order...
- Ascending Order is Default...
- Use DESC Keyword to Override Column Default...
- 255 Columns Maximum...
- Ascending Order, Values Will Be Listed in the Following Sequence:
  - Numeric Values
  - Character Values
  - NULL Values
- In Descending Order, Sequence Is Reversed...

# ORDER BY Example...

Enter SQL Statement:

```
SELECT lastname, firstname, state, city
FROM customers
WHERE state IN('FL','CA')
ORDER BY state DESC, city;
```

Results Script Output Explain Autotrace DBMS Output

Results:

	A Z	LASTNAME	A Z	FIRSTNAME	A Z	STATE	A Z	CITY
1		NGUYEN		NICHOLAS		FL		CLERMONT
2		MORALES		BONITA		FL		EASTPOINT
3		SHELL		STEVE		FL		MIAMI
4		SMITH		LEILA		FL		TALLAHASSEE
5		DAUM		MICHELL		CA		BURBANK
6		PEREZ		JORGE		CA		BURBANK
7		THOMPSON		RYAN		CA		SANTA MONICA

# ORDER BY Can Reference Column Position...

- What Is This Doing?...

Enter SQL Statement:

```
SELECT lastname, firstname, state, city
FROM customers
WHERE state IN('FL','CA')
ORDER BY 3 DESC, 4;
```

Results Script Output Explain Autotrace DBMS Output

Results:

	AZ	LASTNAME	AZ	FIRSTNAME	AZ	STATE	AZ	CITY
1		NGUYEN		NICHOLAS		FL		CLERMONT
2		MORALES		BONITA		FL		EASTPOINT
3		SCHELL		STEVE		FL		MIAMI
4		SMITH		LEILA		FL		TALLAHASSEE
5		DAUM		MICHELL		CA		BURBANK
6		PEREZ		JORGE		CA		BURBANK
7		THOMPSON		RYAN		CA		SANTA MONICA

# Syntax Summary – Part 1...

Element	Description	Example
<b>Optional SELECT clauses</b>		
WHERE clause	Specifies a search condition	SELECT * FROM customers WHERE state = 'GA';
ORDER BY clause	Specifies the display order of query results	SELECT * FROM publisher ORDER BY name;
<b>Mathematical Comparison Operators</b>		
=	“Equality” operator—requires an exact match of the record data and search value	WHERE cost = 55.95
>	“Greater than” operator— requires a record to be greater than the search value	WHERE cost > 55.95
<	“Less than” operator— requires a record to be less than the search value	WHERE cost < 55.95
<>, !=, ^=	“Not equal to” operator— requires a record to not match the search value	WHERE cost <> 55.95 <i>or</i> WHERE cost != 55.95 <i>Or</i> WHERE cost ^= 55.95
<=	“Less than or equal to” operator—requires a record to be less than or an exact match with the search value	WHERE cost <= 55.95
>=	“Greater than or equal to” operator—requires a record to be greater than or an exact	WHERE cost >= 55.95

# Syntax Summary – Part 2...

Element	Description	Example
<b>Other Comparison Operators</b>		
[NOT] BETWEEN x AND y	Searches for records in a specified range of values	WHERE cost BETWEEN 40 AND 65
[NOT] IN (x,y,...)	Searches for records matching one of the items in the list	WHERE cost IN (22, 55.95,13.50)
[NOT] LIKE	Searches for records matching a search pattern—used with wildcard characters	WHERE lastname LIKE '_A%'
IS[NOT] NULL	Searches for records with a NULL value in the indicated column	WHERE referred IS NULL
<b>Wildcard Characters</b>		
%	Percent sign wildcard represents any number of characters	WHERE lastname LIKE '%R%'
–	Underscore symbol wildcard represents exactly one character in the indicated position	WHERE lastname LIKE
<b>Logical Operators</b>		
AND	Combines two conditions together—record must match both conditions	WHERE cost > 20 AND retail < 50
OR	Requires a record to match only one of the search conditions	WHERE cost > 20 OR retail < 50



# Questions...