COURSE TECHNOLOGY
CENGAGE Learning™

Oracle SQL

SQL SELECT Statements

Objectives

- Identify keywords, mandatory clauses, and optional clauses in a SELECT statement
- Select and view all columns of a table
- Select and view one column of a table
- Display multiple columns of a table
- Use a WHERE clause to restrict the rows returned by a query
- Create a search condition using mathematical comparison operators
- Use the BETWEEN…AND comparison operator to identify records within a range of values
- Specify a list of values for a search condition using the IN comparison operator

Objectives (continued)

- Use a column alias to clarify the contents of a particular column
- Perform basic arithmetic operations in the SELECT clause
- Remove duplicate lists using either the DISTINCT or UNIQUE keyword
- Use concatenation to combine fields, literals, and other data
- Search for patterns using the LIKE comparison operator
- Identify the purpose of the % and _ wildcard characters
- Join multiple search conditions using the appropriate logical operator
- Perform searches for NULL values
- Specify the order for the presentation of query results using an ORDER BY clause

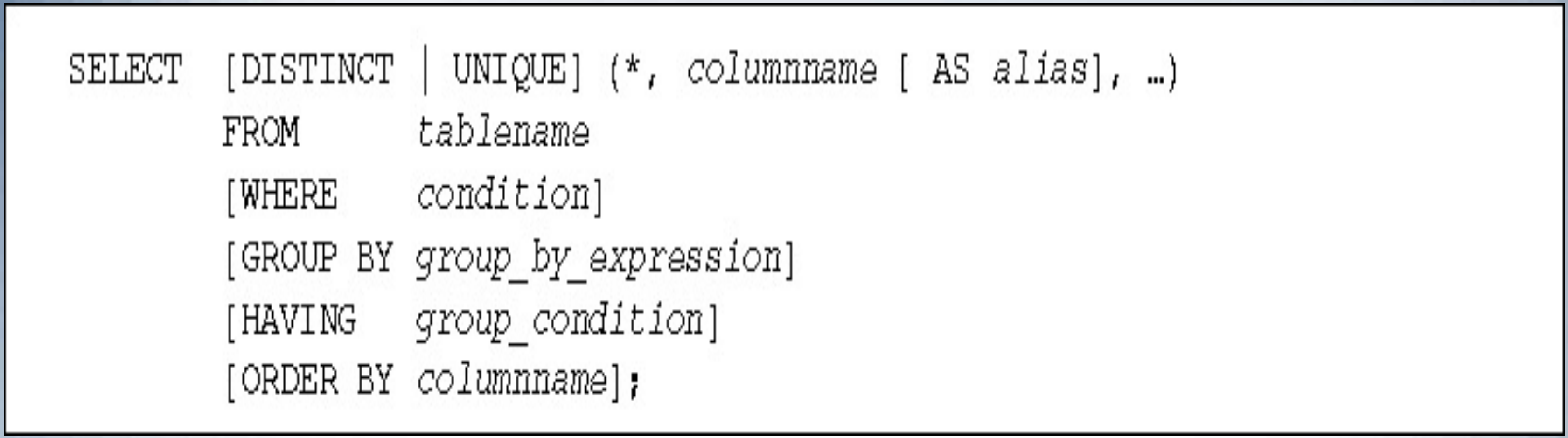
Create the JustLee Database

- Use the provided script to create the database so you can follow the chapter examples
- Verify table contents using the DESCRIBE command

SELECT Statement Syntax

- SELECT statements are used to retrieve data from the database
- A SELECT statement is referred to as a query
- Syntax gives the basic structure, or rules, for a command
- Optional clauses and keywords are shown in brackets

SELECT Statement Syntax (continued)



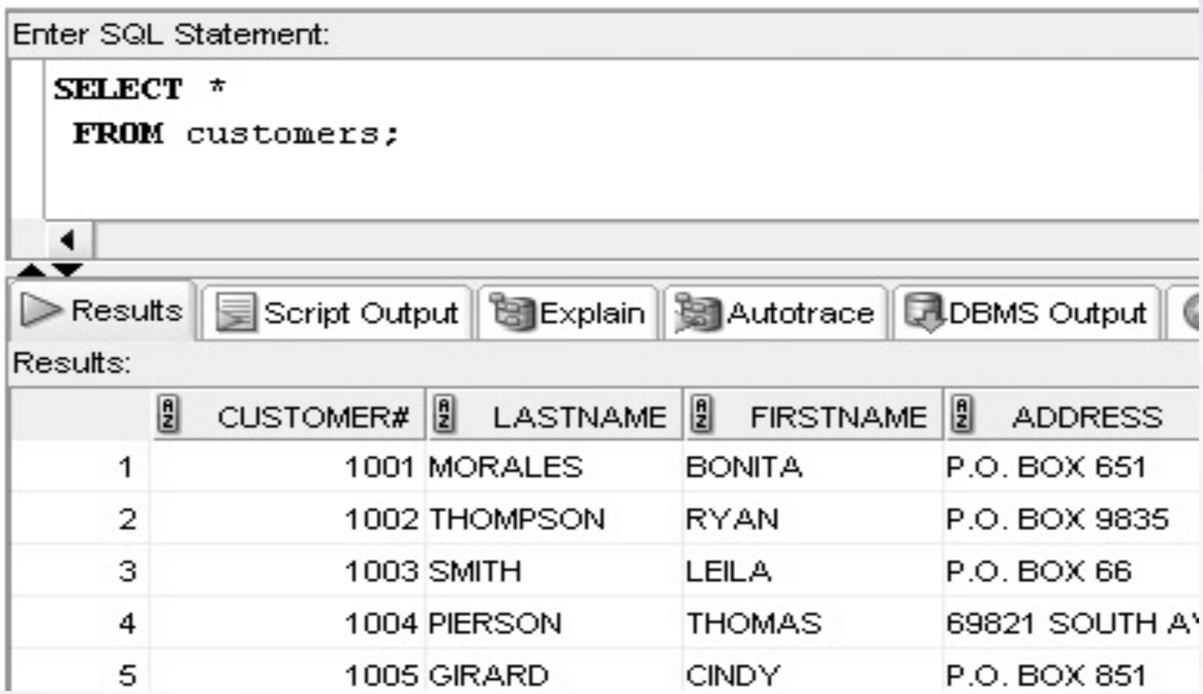
```
SELECT  [DISTINCT | UNIQUE] (*, columnname [ AS alias], …)
        FROM      tablename
        [WHERE    condition]
        [GROUP BY group_by_expression]
        [HAVING   group_condition]
        [ORDER BY columnname];
```

SELECT Statement Syntax (continued)

- SELECT and FROM clauses are required
- SELECT clause identifies column(s)
- FROM clause identifies table(s)
- Each clause begins with a keyword

Selecting All Data in a Table

- Substitute an asterisk for the column names in a SELECT clause

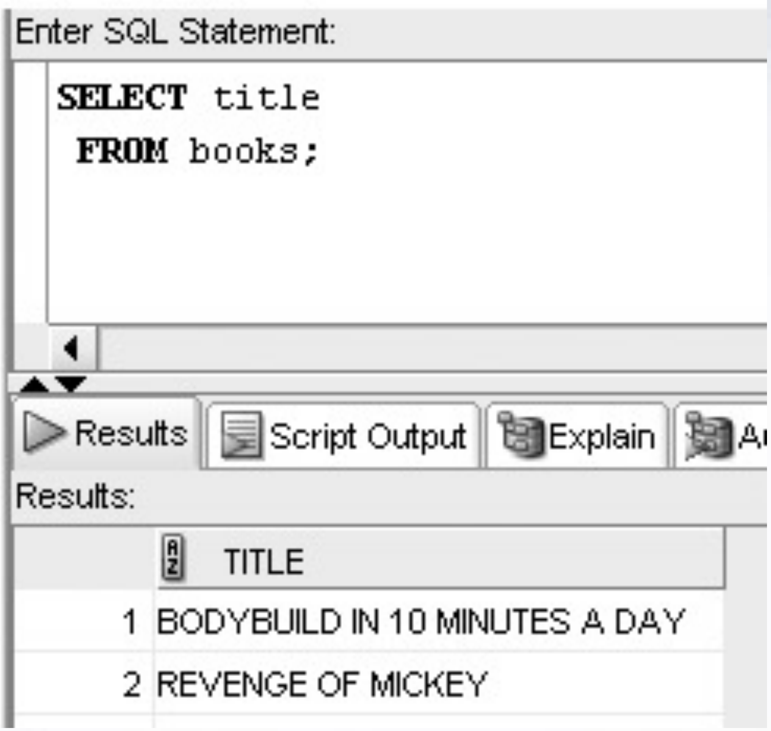


```
SELECT *
 FROM customers;
```

	CUSTOMER#	LASTNAME	FIRSTNAME	ADDRESS
1	1001	MORALES	BONITA	P.O. BOX 651
2	1002	THOMPSON	RYAN	P.O. BOX 9835
3	1003	SMITH	LEILA	P.O. BOX 66
4	1004	PIERSON	THOMAS	69821 SOUTH A
5	1005	GIRARD	CINDY	P.O. BOX 851

Selecting One Column from a Table

- Enter column name in SELECT clause



Enter SQL Statement:

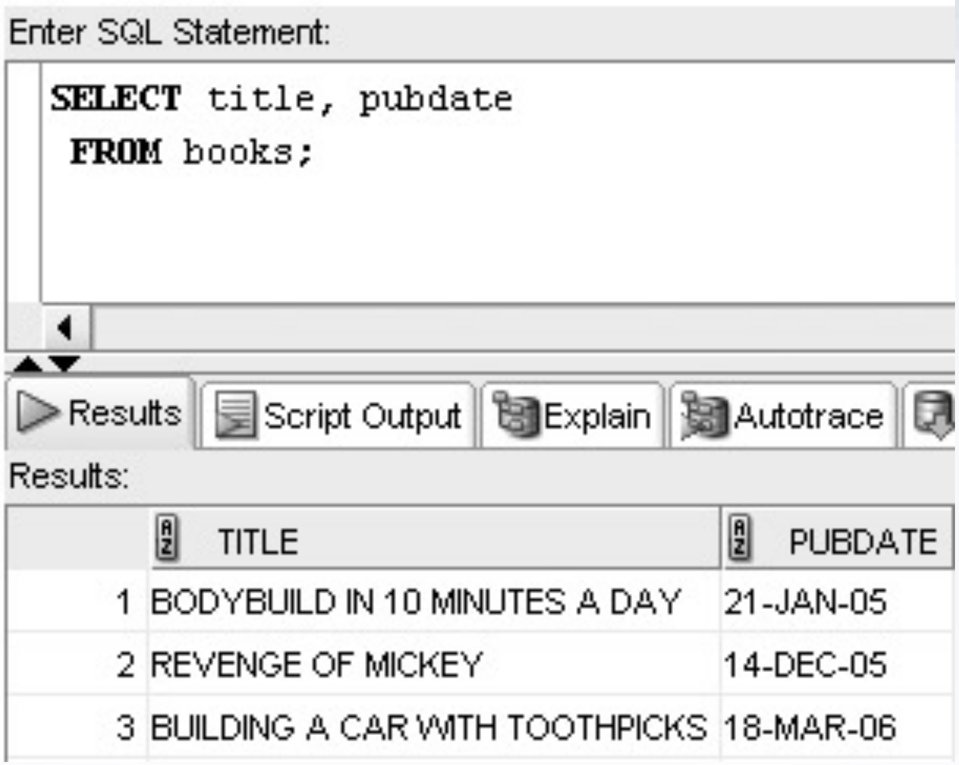
```
SELECT title
 FROM books;
```

Results:

```
      TITLE
    1 BODYBUILD IN 10 MINUTES A DAY
    2 REVENGE OF MICKEY
```

Selecting Multiple Columns from a Table

- Separate column names with a comma



```
SELECT title, pubdate
 FROM books;
```

	TITLE	PUBDATE
1	BODYBUILD IN 10 MINUTES A DAY	21-JAN-05
2	REVENGE OF MICKEY	14-DEC-05
3	BUILDING A CAR WITH TOOTHPICKS	18-MAR-06

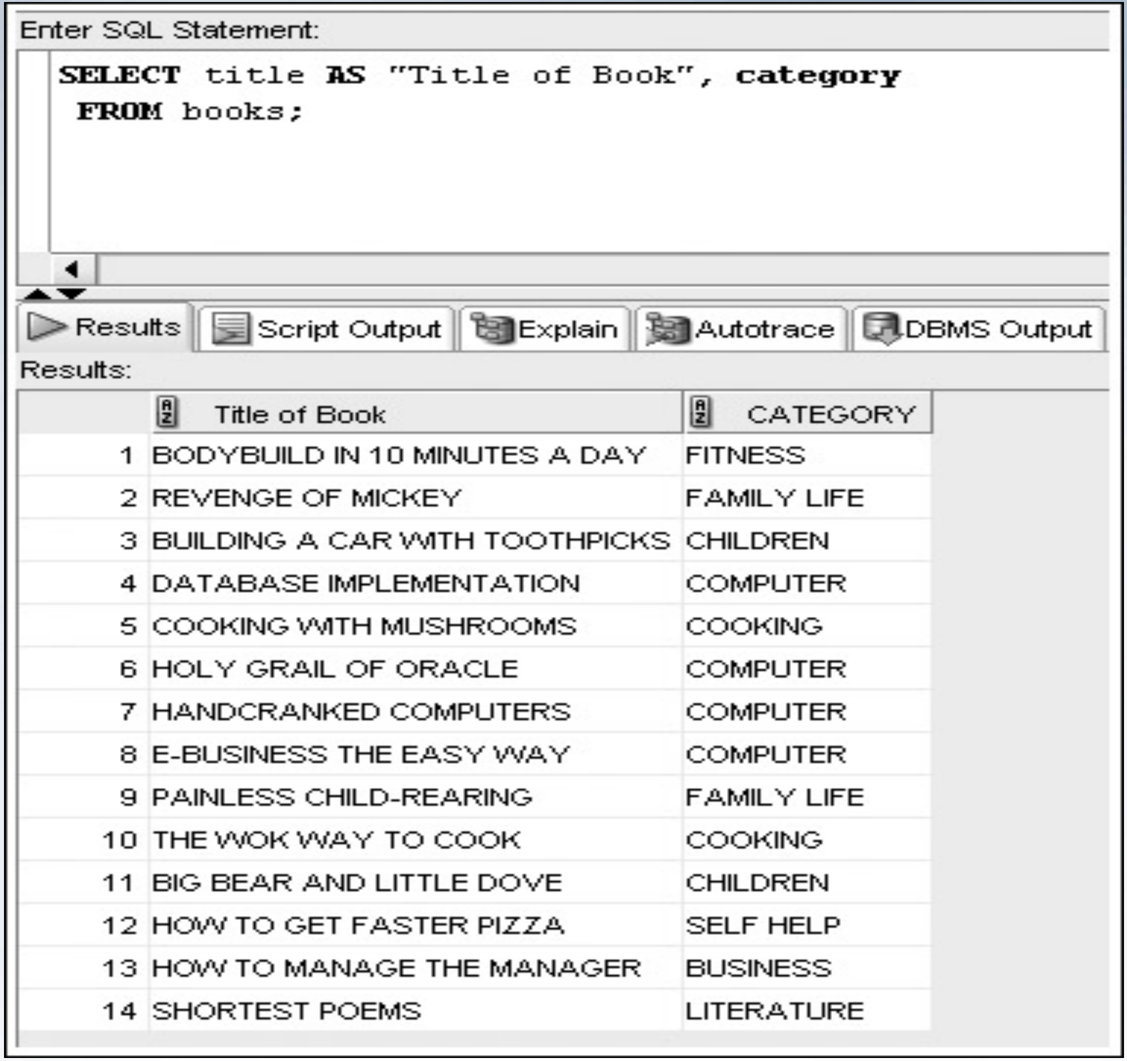
Operations within the SELECT Statement

- Column alias can be used for column headings
- Perform arithmetic operations
- Suppress duplicates
- Concatenate data

Using Column Aliases

- List the alias after the column heading
- AS keyword is optional
- Enclose in double quotation marks:
  - If it contains blank space(s)
  - If it contains special symbol(s)
  - To retain case

Column Alias Example



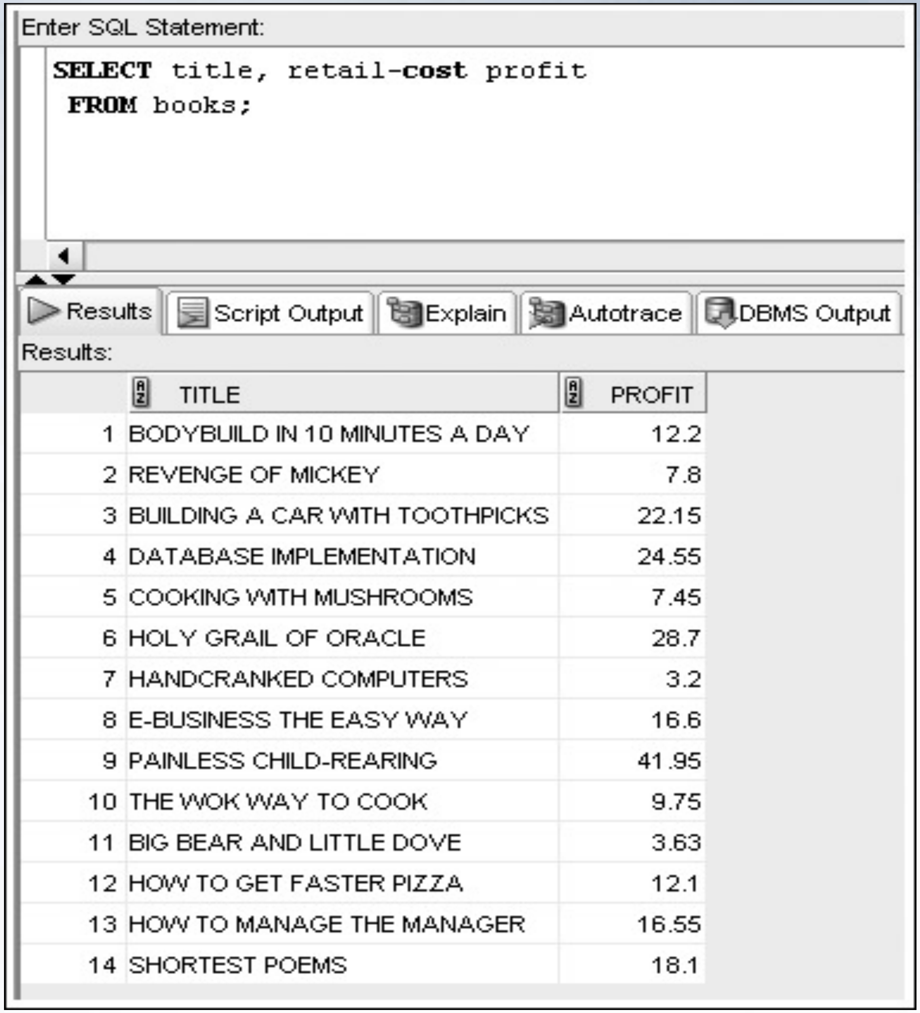
```
SELECT title AS "Title of Book", category
 FROM books;
```

	Title of Book	CATEGORY
1	BODYBUILD IN 10 MINUTES A DAY	FITNESS
2	REVENGE OF MICKEY	FAMILY LIFE
3	BUILDING A CAR WITH TOOTHPICKS	CHILDREN
4	DATABASE IMPLEMENTATION	COMPUTER
5	COOKING WITH MUSHROOMS	COOKING
6	HOLY GRAIL OF ORACLE	COMPUTER
7	HANDCRANKED COMPUTERS	COMPUTER
8	E-BUSINESS THE EASY WAY	COMPUTER
9	PAINLESS CHILD-REARING	FAMILY LIFE
10	THE WOK WAY TO COOK	COOKING
11	BIG BEAR AND LITTLE DOVE	CHILDREN
12	HOW TO GET FASTER PIZZA	SELF HELP
13	HOW TO MANAGE THE MANAGER	BUSINESS
14	SHORTEST POEMS	LITERATURE

Using Arithmetic Operations

- Arithmetic operations
  - Executed left to right
  - Multiplication and division are solved first
  - Addition and subtraction are solved last
  - Override order with parentheses

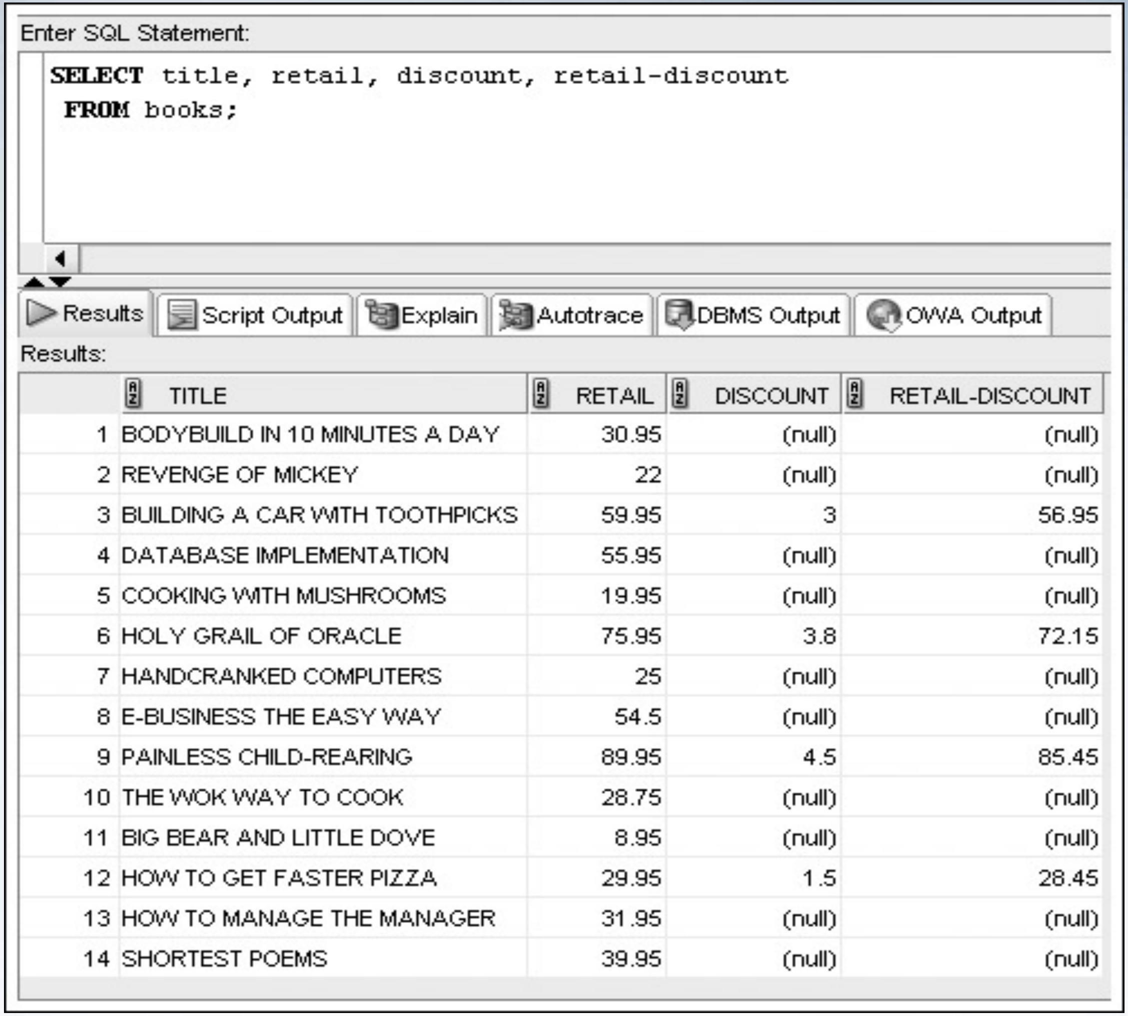
Example Arithmetic Operation with Column Alias



```
SELECT title, retail-cost profit
 FROM books;
```

	TITLE	PROFIT
1	BODYBUILD IN 10 MINUTES A DAY	12.2
2	REVENGE OF MICKEY	7.8
3	BUILDING A CAR WITH TOOTHPICKS	22.15
4	DATABASE IMPLEMENTATION	24.55
5	COOKING WITH MUSHROOMS	7.45
6	HOLY GRAIL OF ORACLE	28.7
7	HANDCRANKED COMPUTERS	3.2
8	E-BUSINESS THE EASY WAY	16.6
9	PAINLESS CHILD-REARING	41.95
10	THE WOK WAY TO COOK	9.75
11	BIG BEAR AND LITTLE DOVE	3.63
12	HOW TO GET FASTER PIZZA	12.1
13	HOW TO MANAGE THE MANAGER	16.55
14	SHORTEST POEMS	18.1

NULL Values

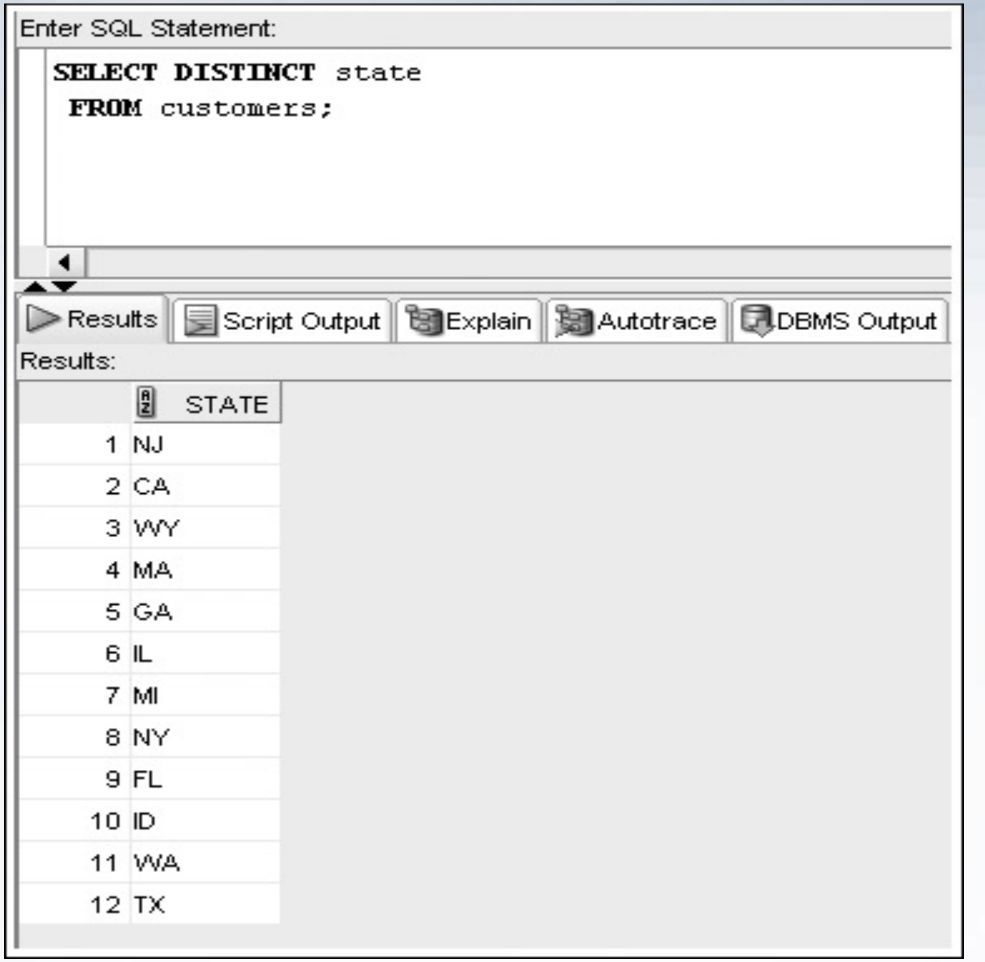


```
SELECT title, retail, discount, retail-discount
 FROM books;
```

	TITLE	RETAIL	DISCOUNT	RETAIL-DISCOUNT
1	BODYBUILD IN 10 MINUTES A DAY	30.95	(null)	(null)
2	REVENGE OF MICKEY	22	(null)	(null)
3	BUILDING A CAR WITH TOOTHPICKS	59.95	3	56.95
4	DATABASE IMPLEMENTATION	55.95	(null)	(null)
5	COOKING WITH MUSHROOMS	19.95	(null)	(null)
6	HOLY GRAIL OF ORACLE	75.95	3.8	72.15
7	HANDCRANKED COMPUTERS	25	(null)	(null)
8	E-BUSINESS THE EASY WAY	54.5	(null)	(null)
9	PAINLESS CHILD-REARING	89.95	4.5	85.45
10	THE WOK WAY TO COOK	28.75	(null)	(null)
11	BIG BEAR AND LITTLE DOVE	8.95	(null)	(null)
12	HOW TO GET FASTER PIZZA	29.95	1.5	28.45
13	HOW TO MANAGE THE MANAGER	31.95	(null)	(null)
14	SHORTEST POEMS	39.95	(null)	(null)

Using DISTINCT and UNIQUE

- Enter DISTINCT or UNIQUE after SELECT keyword to suppress duplicates



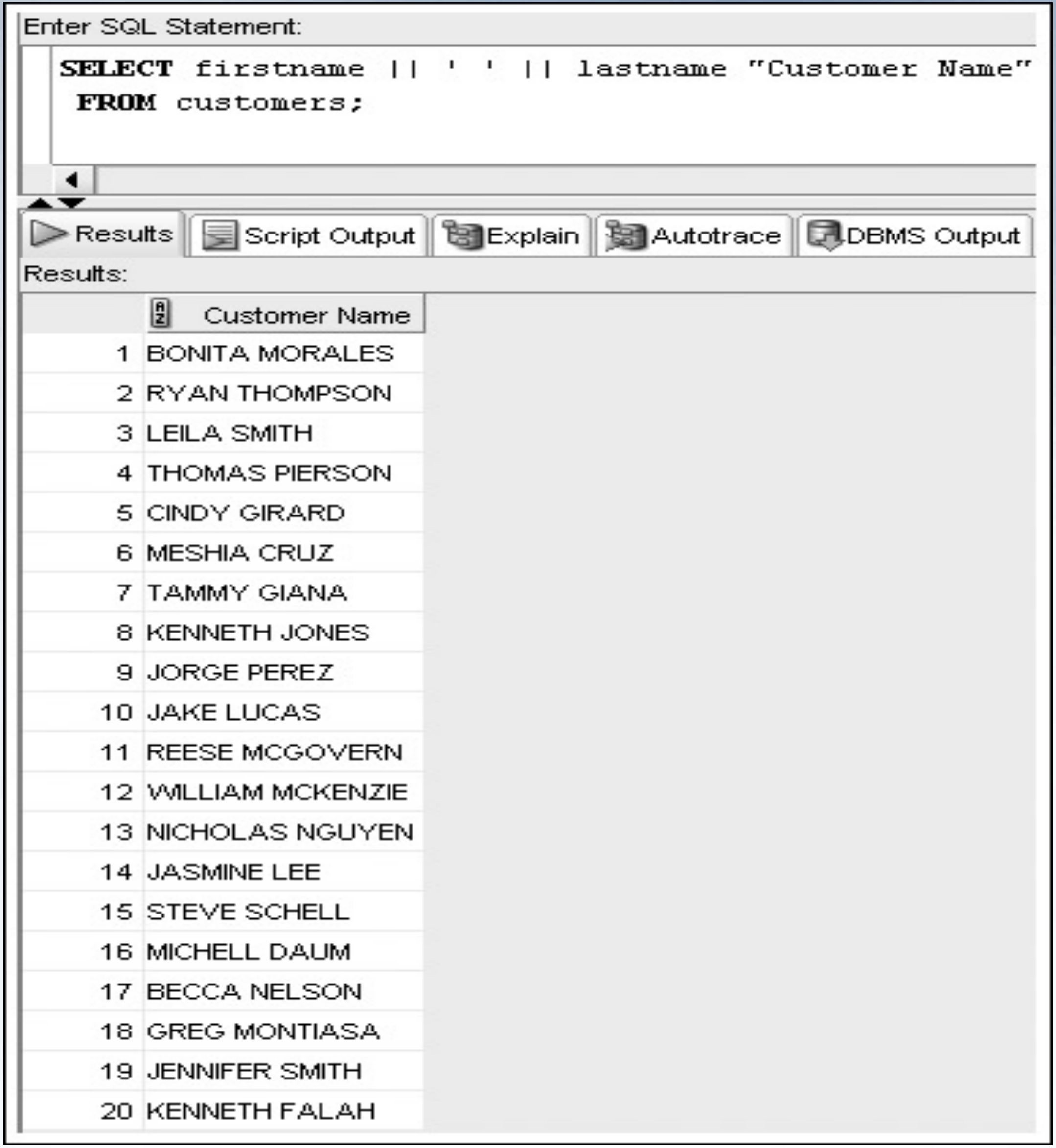
```
SELECT DISTINCT state
 FROM customers;
```

```
    STATE
 1  NJ
 2  CA
 3  WY
 4  MA
 5  GA
 6  IL
 7  MI
 8  NY
 9  FL
10  ID
11  WA
12  TX
```

Using Concatenation

- You can combine data with a string literal
- Use the concatenation operator, ||
- It allows the use of column aliases

Concatenation Example



Enter SQL Statement:

```
SELECT firstname || ' ' || lastname "Customer Name"
 FROM customers;
```

Results:

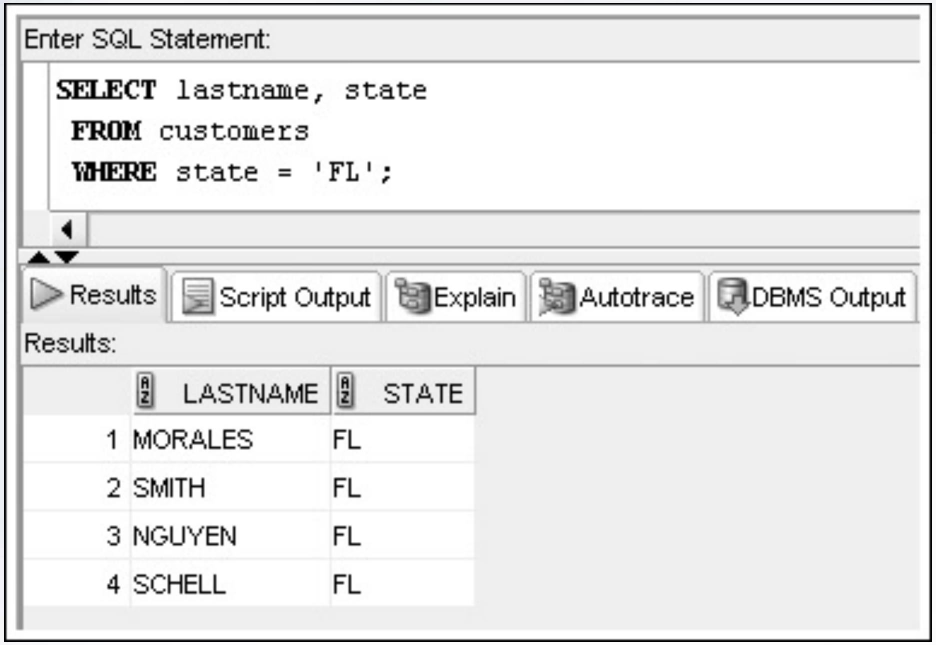
	Customer Name
1	BONITA MORALES
2	RYAN THOMPSON
3	LEILA SMITH
4	THOMAS PIERSON
5	CINDY GIRARD
6	MESHIA CRUZ
7	TAMMY GIANA
8	KENNETH JONES
9	JORGE PEREZ
10	JAKE LUCAS
11	REESE MCGOVERN
12	WILLIAM MCKENZIE
13	NICHOLAS NGUYEN
14	JASMINE LEE
15	STEVE SCHELL
16	MICHELL DAUM
17	BECCA NELSON
18	GREG MONTIASA
19	JENNIFER SMITH
20	KENNETH FALAH

WHERE Clause Syntax

- A WHERE clause is 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



Enter SQL Statement:

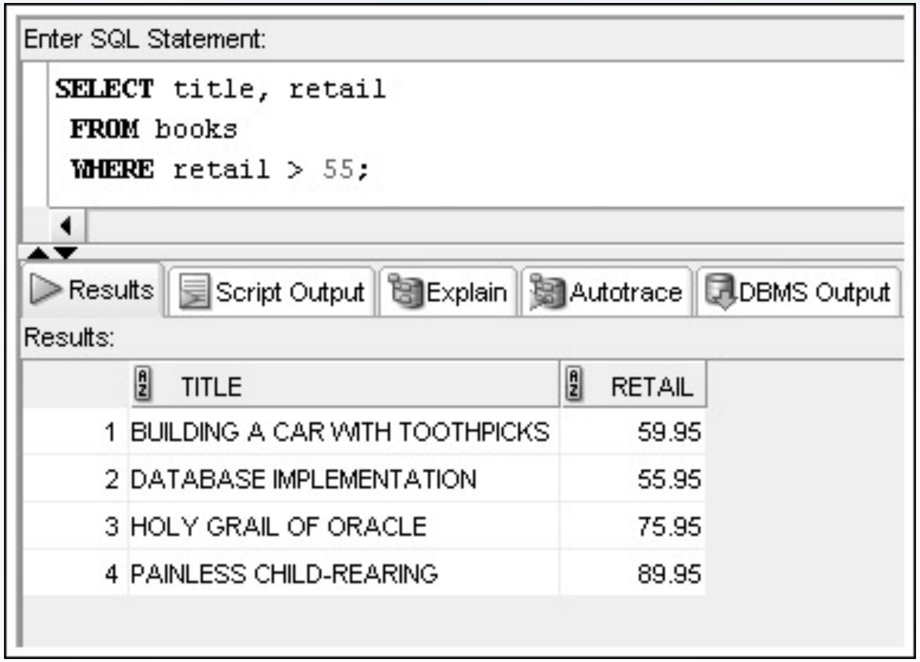
```
SELECT lastname, state
 FROM customers
 WHERE state = 'FL';
```

Results:

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

Comparison Operators

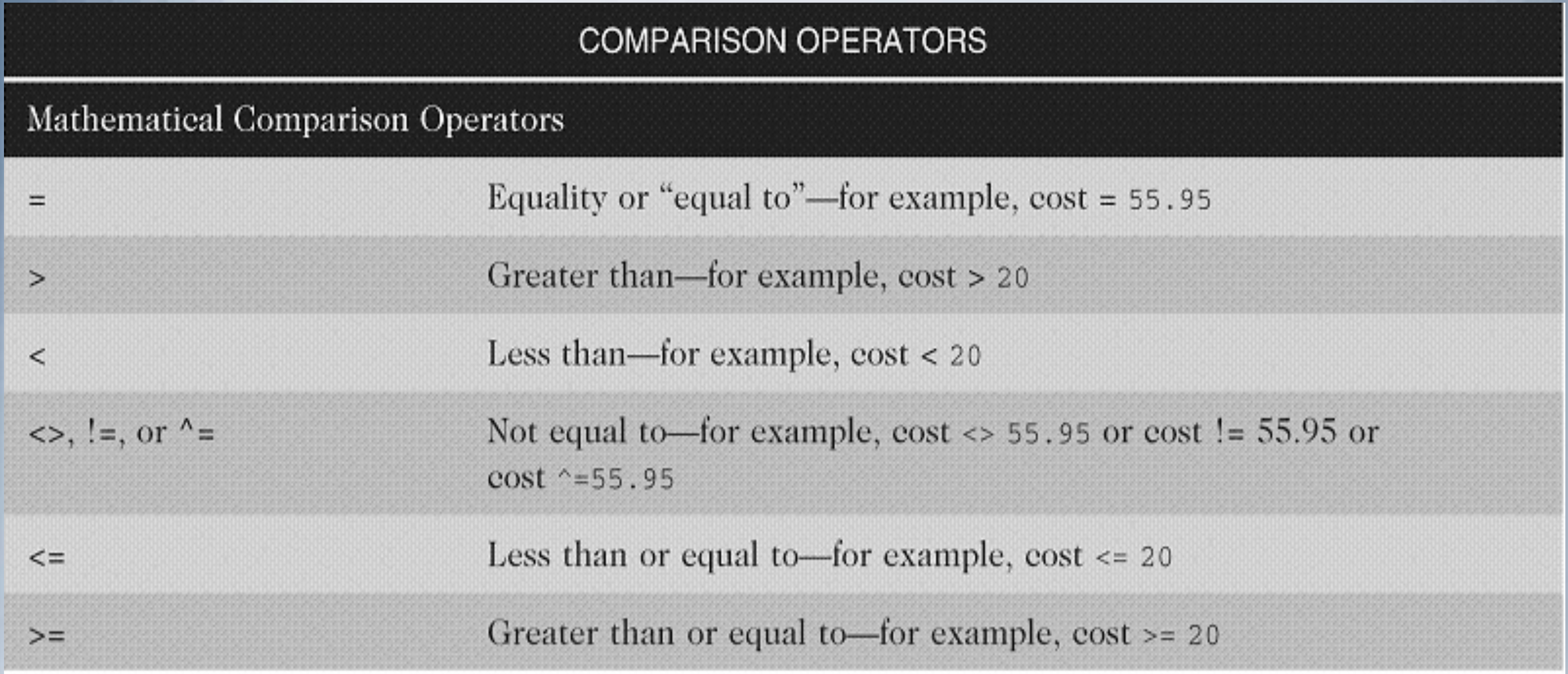
- Indicate how the data should relate to the given search value



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

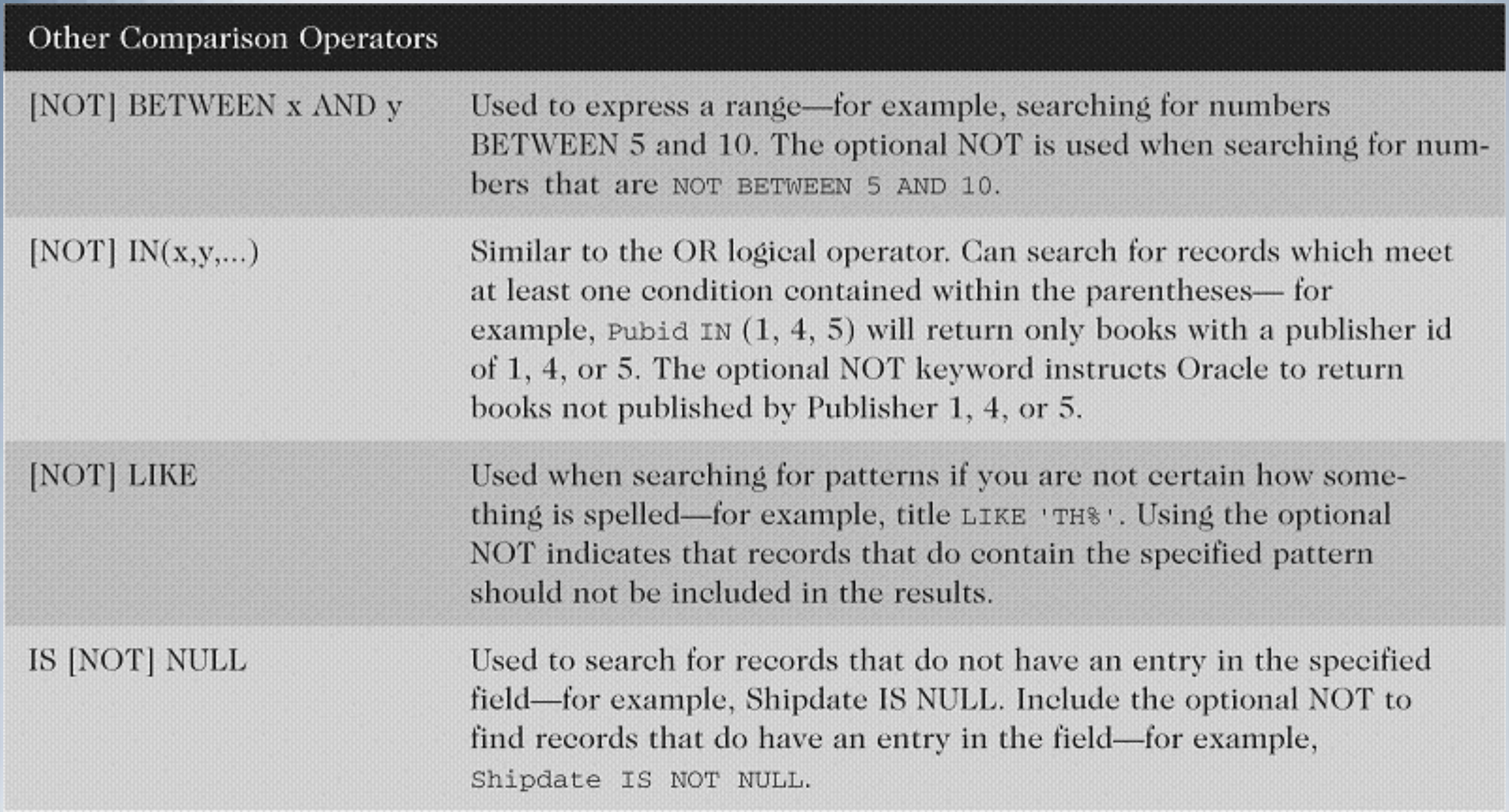
	TITLE	RETAIL
1	BUILDING A CAR WITH TOOTHPICKS	59.95
2	DATABASE IMPLEMENTATION	55.95
3	HOLY GRAIL OF ORACLE	75.95
4	PAINLESS CHILD-REARING	89.95

Arithmetic Comparison Operators



COMPARISON OPERATORS	
Mathematical Comparison Operators	
=	Equality or "equal to"—for example, cost = 55.95
>	Greater than—for example, cost > 20
<	Less than—for example, cost < 20
<>, !=, or ^=	Not equal to—for example, cost <> 55.95 or cost != 55.95 or cost ^=55.95
<=	Less than or equal to—for example, cost <= 20
>=	Greater than or equal to—for example, cost >= 20

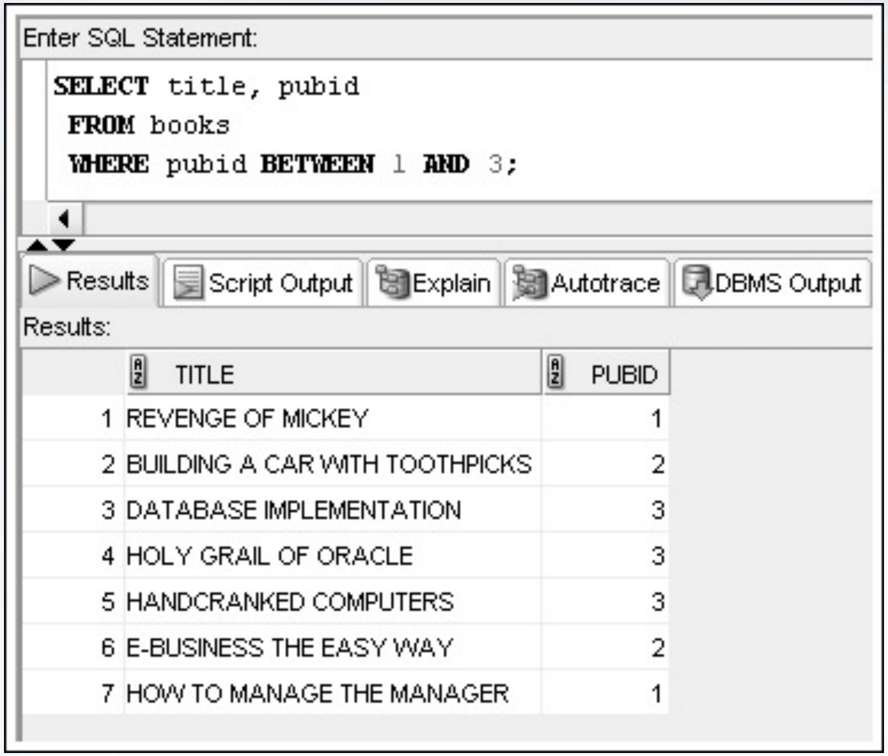
Other Comparison Operators



Other Comparison Operators	
[NOT] BETWEEN x AND y	Used to express a range—for example, searching for numbers BETWEEN 5 and 10. The optional NOT is used when searching for numbers that are `NOT BETWEEN 5 AND 10`.
[NOT] IN(x,y,...)	Similar to the OR logical operator. Can search for records which meet at least one condition contained within the parentheses— for example, `Pubid IN` (1, 4, 5) will return only books with a publisher id of 1, 4, or 5. The optional NOT keyword instructs Oracle to return books not published by Publisher 1, 4, or 5.
[NOT] LIKE	Used when searching for patterns if you are not certain how something is spelled—for example, title `LIKE 'TH%'`. Using the optional NOT indicates that records that do contain the specified pattern should not be included in the results.
IS [NOT] NULL	Used to search for records that do not have an entry in the specified field—for example, Shipdate IS NULL. Include the optional NOT to find records that do have an entry in the field—for example, `Shipdate IS NOT NULL`.

BETWEEN…AND Operator

- Finds values in a specified range



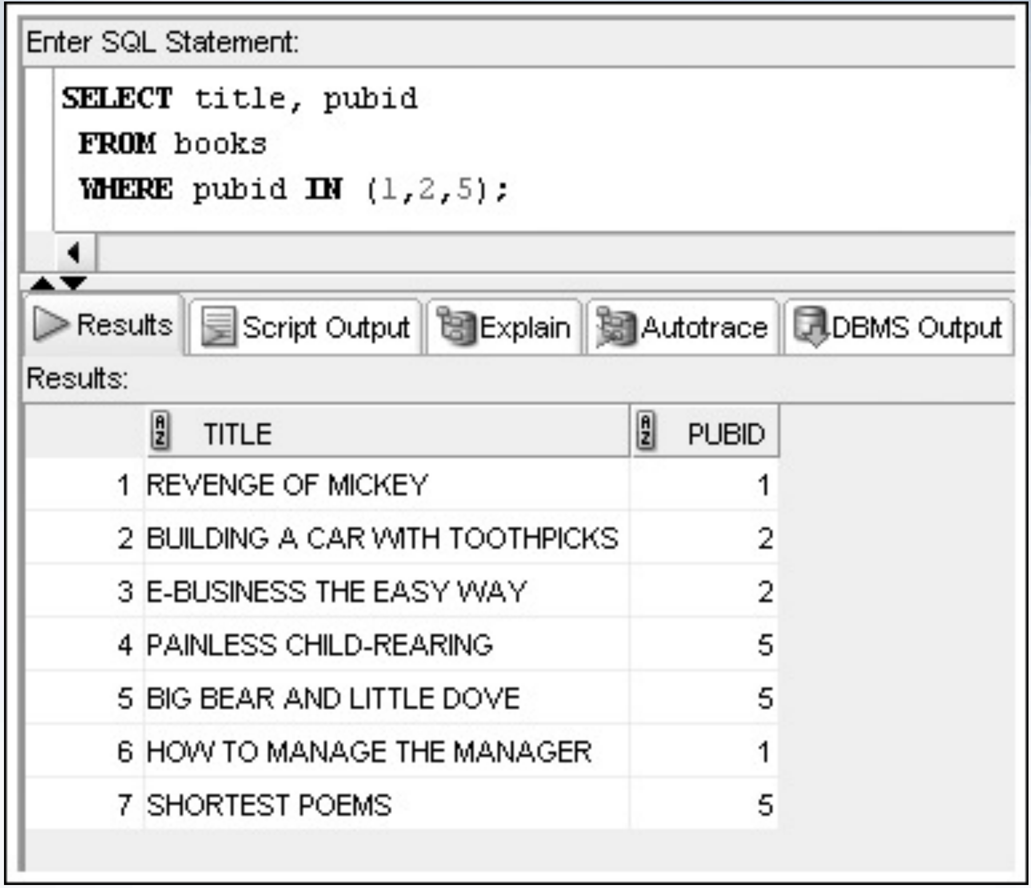
```
SELECT title, pubid
 FROM books
 WHERE pubid BETWEEN 1 AND 3;
```

	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

IN Operator

- Returns records that match a value in a specified list
- List must be in parentheses
- Values are separated by commas

IN Operator Example



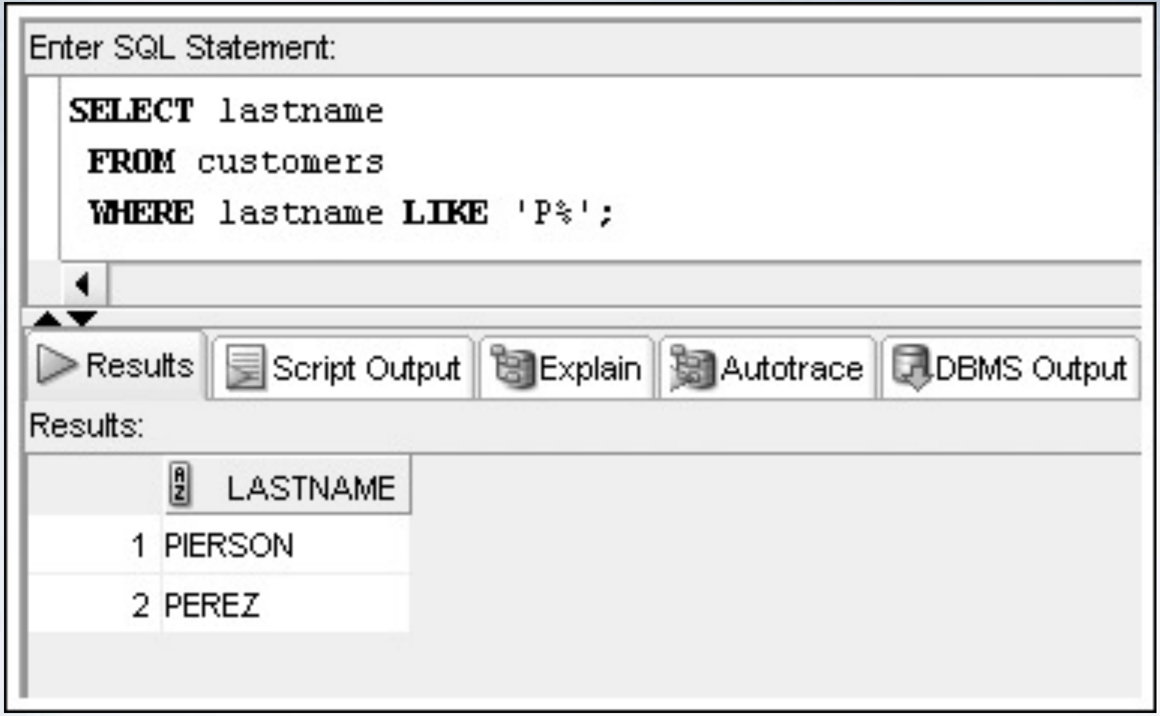
```
SELECT title, pubid
 FROM books
 WHERE pubid IN (1,2,5);
```

	TITLE	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

LIKE Operator

- Performs pattern searches
- Used with wildcard characters
  - Underscore (_) for exactly one character in the indicated position
  - Percent sign (%) represents any number of characters

LIKE Operator Example



Enter SQL Statement:

```
SELECT lastname
 FROM customers
 WHERE lastname LIKE 'P%';
```

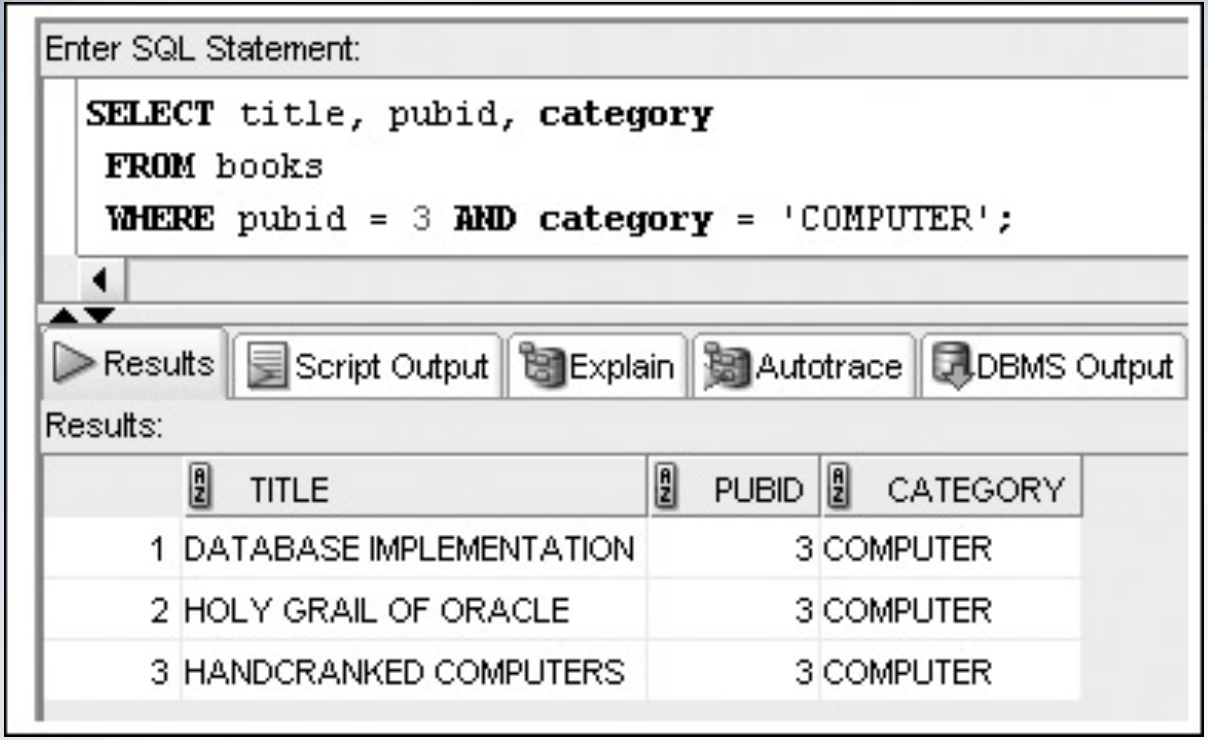
Results:

	LASTNAME
1	PIERSON
2	PEREZ

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

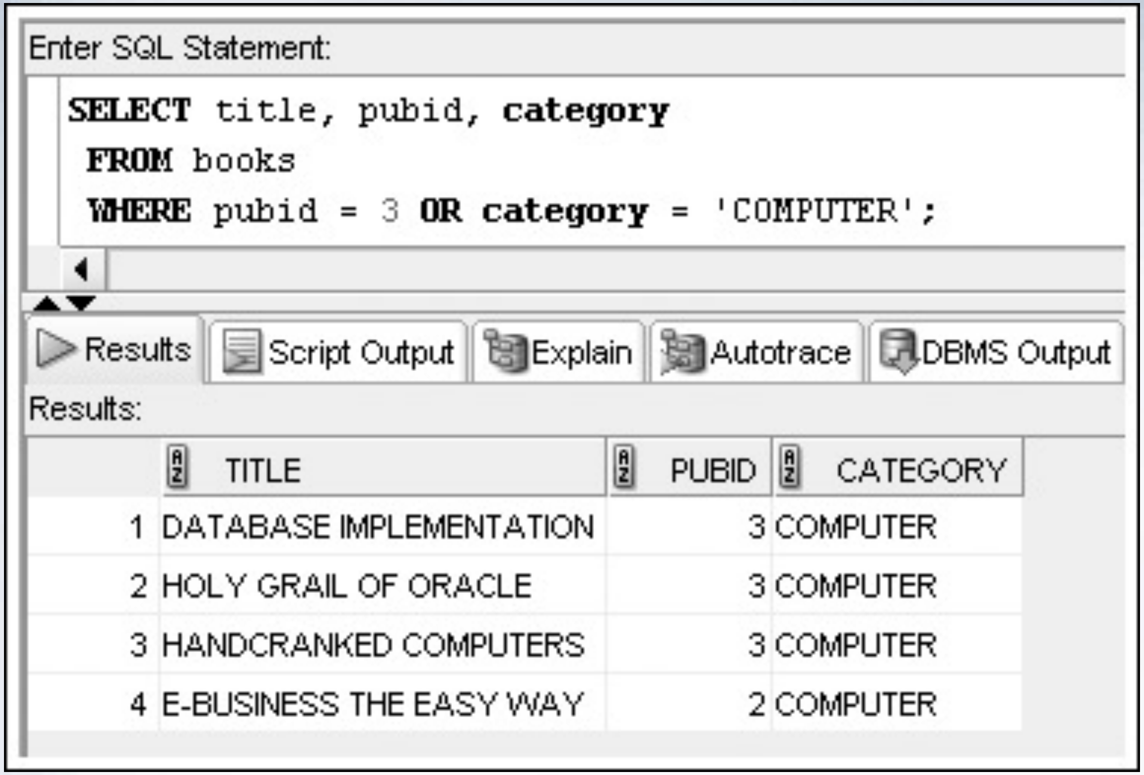
AND Logical Operator Example



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

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

OR Logical Operator Example



Enter SQL Statement:

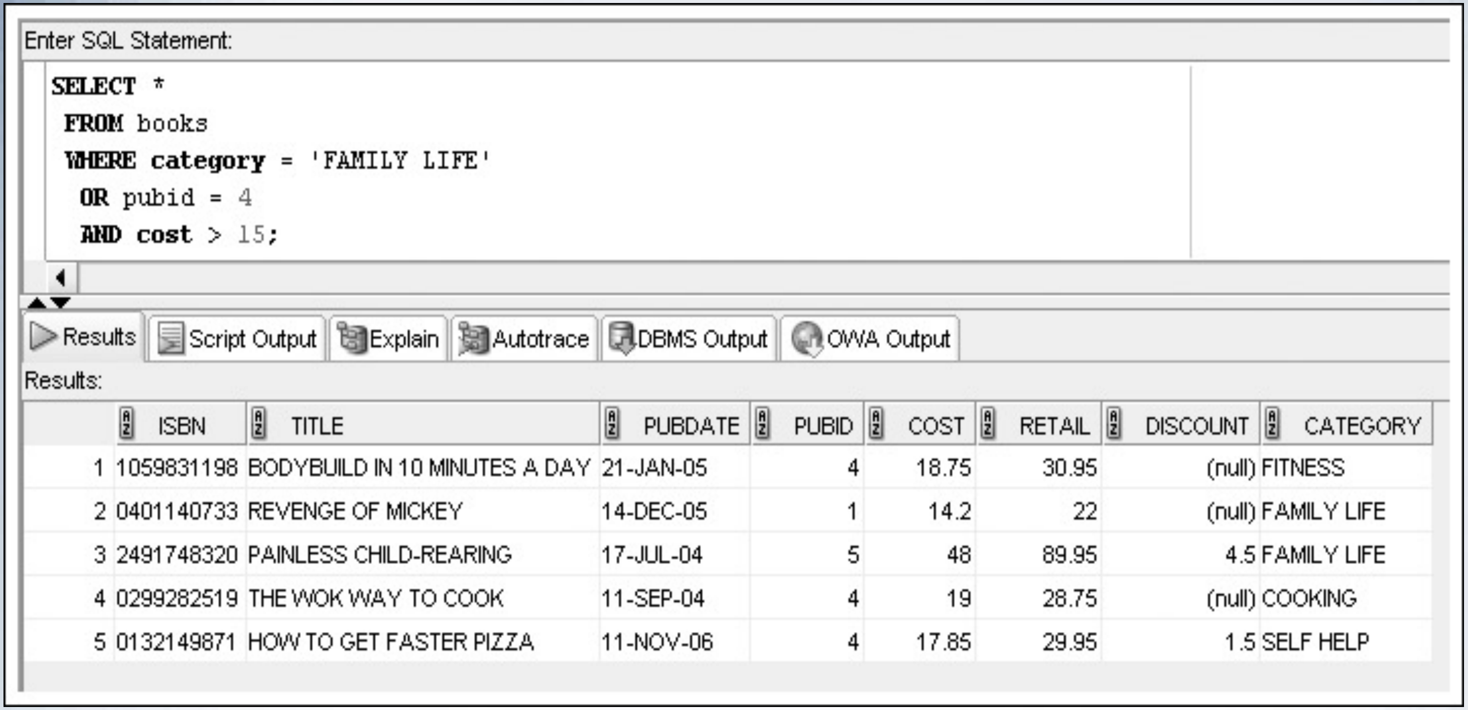
```
SELECT title, pubid, category
 FROM books
 WHERE pubid = 3 OR category = 'COMPUTER';
```

Results:

	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

- Resolved in order of NOT, AND, OR

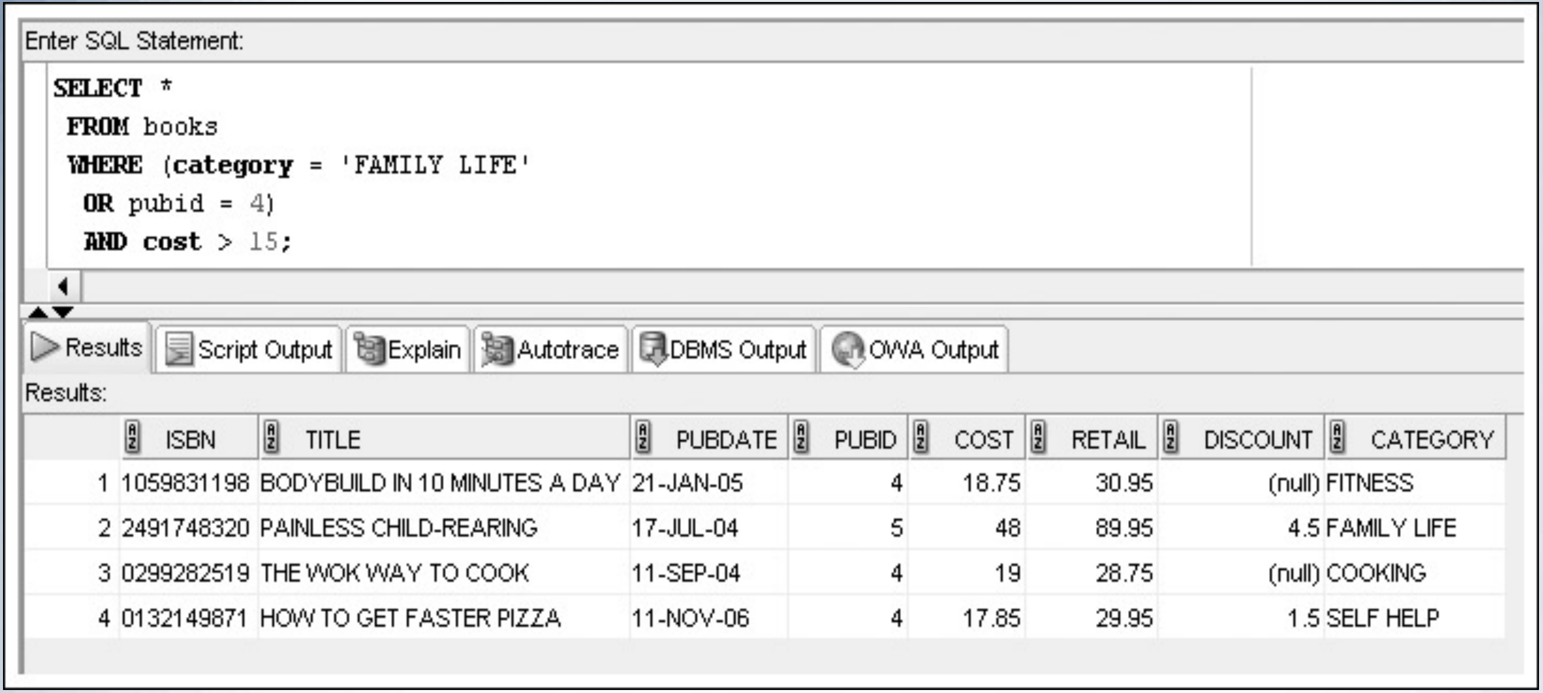


```
SELECT *
 FROM books
 WHERE category = 'FAMILY LIFE'
  OR pubid = 4
  AND cost > 15;
```

	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

- Use parentheses to override the order of evaluation



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

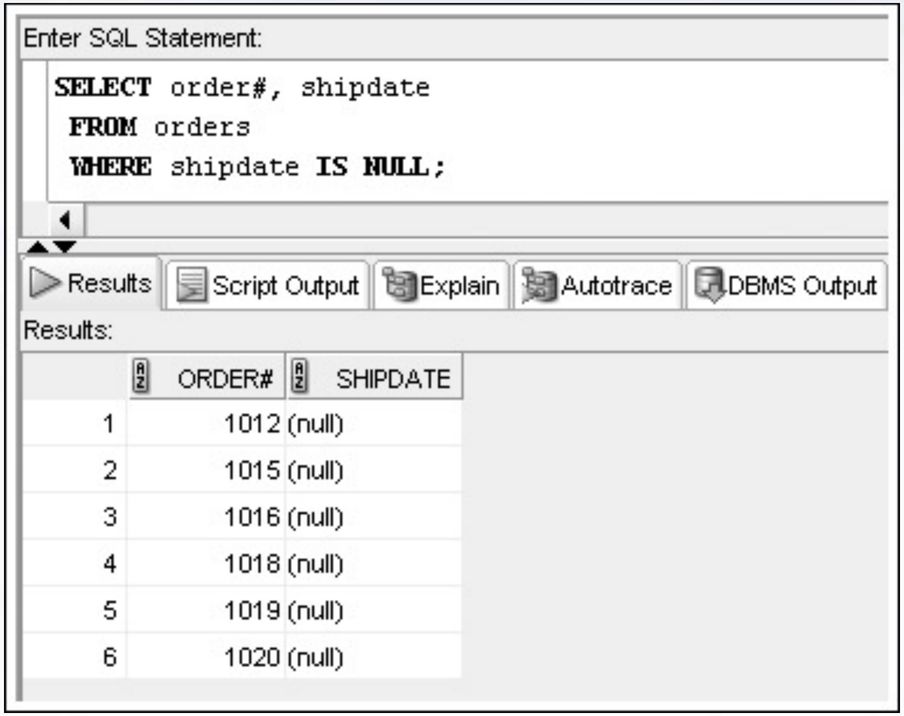
	ISBN	TITLE	PUBDATE	PUBID	COST	RETAIL	DISCOUNT	CATEGORY
1	1059831198	BODYBUILD IN 10 MINUTES A DAY	21-JAN-05	4	18.75	30.95	(null)	FITNESS
2	2491748320	PAINLESS CHILD-REARING	17-JUL-04	5	48	89.95	4.5	FAMILY LIFE
3	0299282519	THE WOK WAY TO COOK	11-SEP-04	4	19	28.75	(null)	COOKING
4	0132149871	HOW TO GET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

Resolving Multiple Types of Operators

1. Arithmetic operators
2. Comparison operators
3. Logical operators

Treatment of NULL Values

- Absence of data
- Requires use of IS NULL operator



Enter SQL Statement:

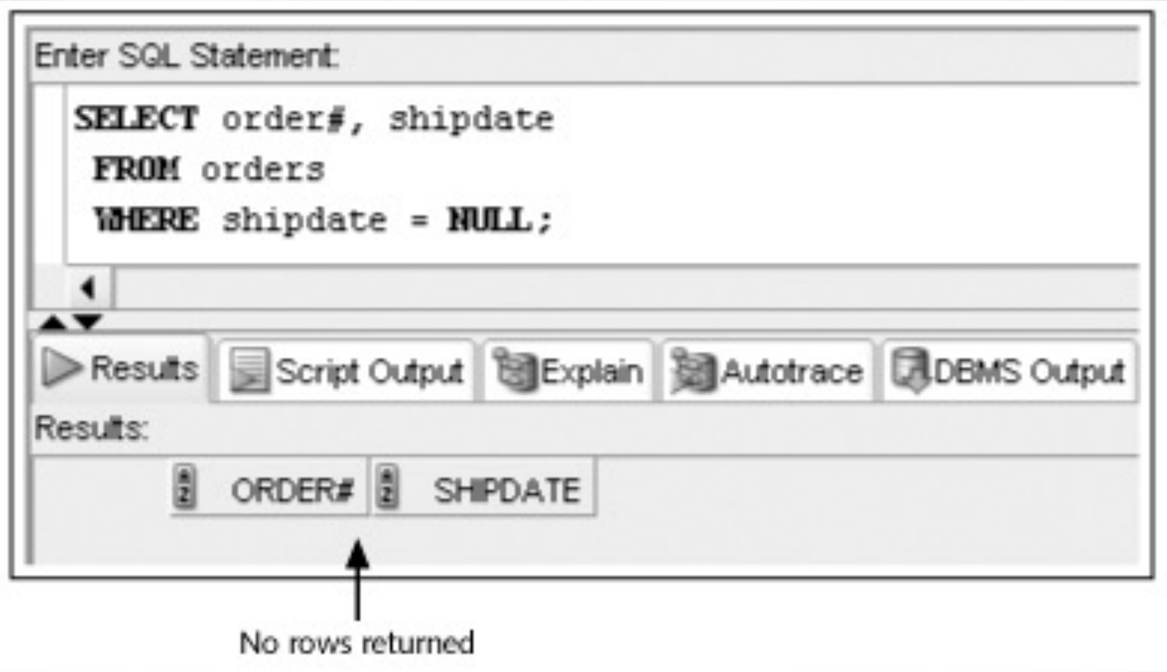
```
SELECT order#, shipdate
 FROM orders
 WHERE shipdate IS NULL;
```

Results:

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

Treatment of NULL Values (continued)

- A common error is using = NULL, which does not raise an Oracle error but also does not return any rows



Enter SQL Statement:

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

No rows returned

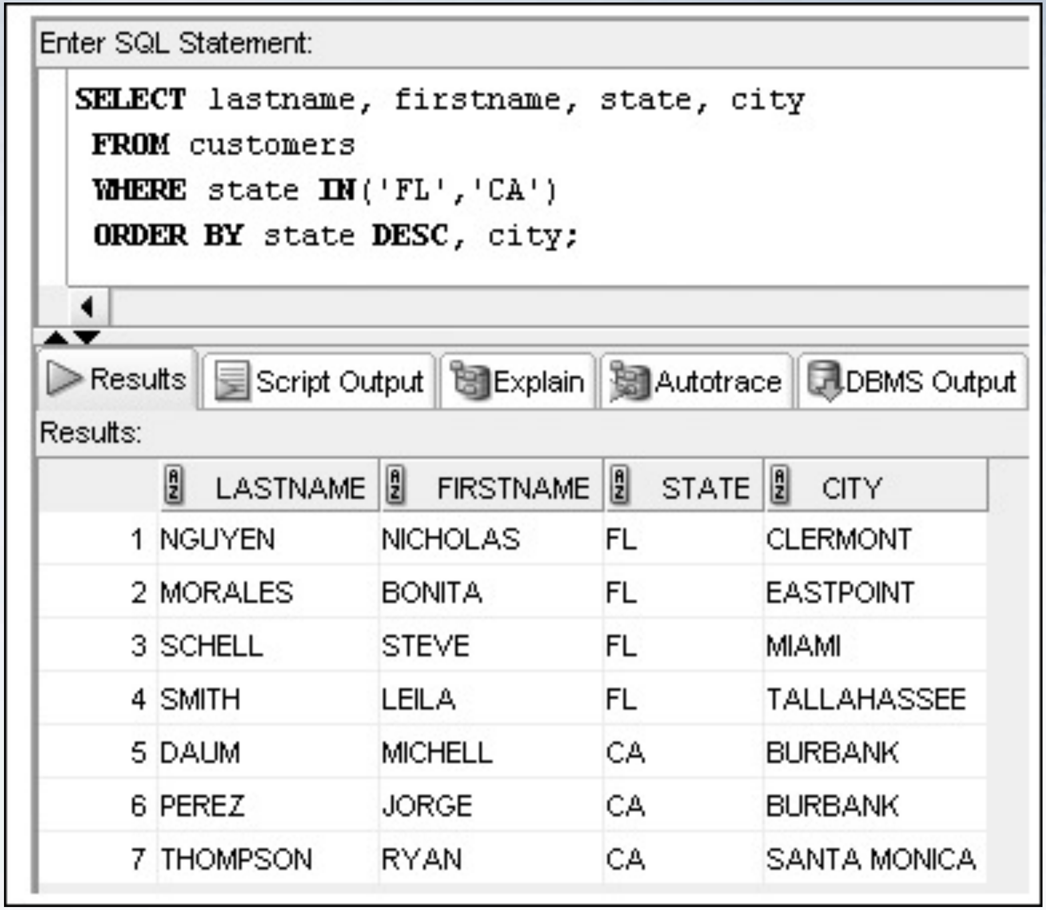
ORDER BY Clause Syntax

- The ORDER BY clause presents data in sorted order
- Ascending order is default
- Use DESC keyword to override column default
- 255 columns maximum

ORDER BY Clause Syntax Sort Sequence

- In 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



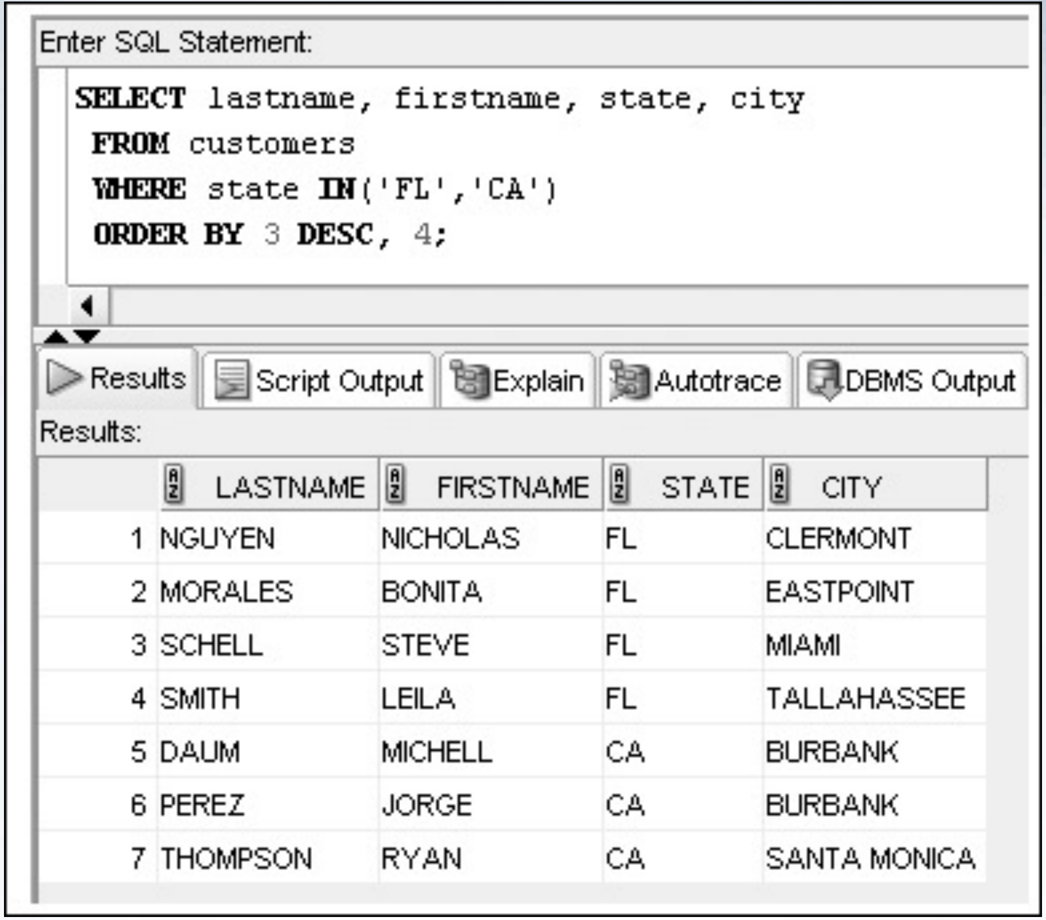
Enter SQL Statement:

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

Results:

	LASTNAME	FIRSTNAME	STATE	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

ORDER BY Can Reference Column Position



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

	LASTNAME	FIRSTNAME	STATE	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

Summary

- A basic query in Oracle 11*g* SQL includes the SELECT and FROM clauses, the only mandatory clauses in a SELECT statement
- To view all columns in the table, specify an asterisk (*) or list all of the column names individually in the SELECT clause
- To display a specific column or set of columns, list the column names in the SELECT clause (in the order in which you want them to appear)
- When listing column names in the SELECT clause, a comma must separate column names

Summary (continued)

- A column alias can be used to clarify the contents of a particular column; if the alias contains spaces or special symbols, or if you want to display the column with any lowercase letters, you must enclose the column alias in double quotation marks (" ")
- Indicate the table name following the FROM keyword
- Basic arithmetic operations can be performed in the SELECT clause
- NULL values indicate an absence of a value

Summary (continued)

- To remove duplicate listings, include either the DISTINCT or UNIQUE keyword
- To specify which table contains the desired columns, you must list the name of the table after the keyword FROM
- Use vertical bars (||) to combine, or concatenate, fields, literals, and other data

Summary (continued)

- The WHERE clause can be included in a SELECT statement to restrict the rows returned by a query to only those meeting a specified condition
- When searching a nonnumeric field, the search values must be enclosed in single quotation marks
- Comparison operators are used to indicate how the record should relate to the search value
- The BETWEEN...AND comparison operator is used to search for records that fall within a certain range of values

Summary (continued)

- The LIKE comparison operator is used with the percent and underscore symbols (% and _) to establish search patterns
- Logical operators such as AND and OR can be used to combine several search conditions
- When using the AND operator, all conditions must be TRUE for a record to be returned in the results
  - However, with the OR operator, only one condition must be TRUE
- A NULL value is the absence of data, not a field with a blank space entered

Summary (continued)

- Use the IS NULL comparison operator to match NULL values; the IS NOT NULL comparison operator finds records that do not contain NULL values in the indicated column
- You can sort the results of queries by using an ORDER BY clause; when used, the ORDER BY clause should be listed last in the SELECT statement
- By default, records are sorted in ascending order; entering DESC directly after the column name sorts the records in descending order
- A column does not have to be listed in the SELECT clause to serve as a basis for sorting