Database Programming with SQL

2-2
Limit Rows Selected
Objectives

This lesson covers the following objectives:

- Apply SQL syntax to restrict the rows returned from a query
- Demonstrate application of the WHERE clause syntax
- Explain why it is important, from a business perspective, to be able to easily limit data retrieved from a table
- Construct and produce output using a SQL query containing character strings and date values
Purpose

• Have you ever had "information overload"?
• The television is on, your Mom is asking you how school went today, the phone rings, and the dog is barking.
• Wouldn't it be nice to be able to restrict the amount of information you have to process at one time?
• In SQL, this is the job of the WHERE clause.
• It is important to be able to choose the information you need to see from a table.
• Tables can have millions of rows of data, and it is a waste of resources to search and return data you don't need or want.
SELECT Statement

• You use SELECT to retrieve information from the database.
• A SELECT statement must include at a minimum a SELECT clause and a FROM clause.
• The WHERE clause is optional.

```
SELECT*|[DISTINCT] column | expression alias|...
FROM table
[WHERE condition(s)];
```
WHERE Clause

• When retrieving data from the database, you may need to limit the rows of data that are displayed.

• You can accomplish this using the WHERE clause.

• A WHERE clause contains a condition that must be met, and it directly follows the FROM clause in a SQL statement.

• The syntax for the WHERE clause is:

  WHERE  column_name  comparison_condition comparison_value

• Note: An alias cannot be used in the WHERE clause!
WHERE Clause

• Examine the following SQL statement from the Employees database:

```sql
SELECT employee_id, first_name, last_name
FROM employees;
```

<table>
<thead>
<tr>
<th>EMPLOYEE_ID</th>
<th>FIRST_NAME</th>
<th>LAST_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Steven</td>
<td>King</td>
</tr>
<tr>
<td>101</td>
<td>Neena</td>
<td>Kochhar</td>
</tr>
<tr>
<td>102</td>
<td>Lex</td>
<td>De Haan</td>
</tr>
</tbody>
</table>

• By adding a WHERE clause, the rows are limited to those rows where the value of employee_id is 101.

```sql
SELECT employee_id, first_name, last_name
FROM employees
WHERE employee_id = 101;
```

<table>
<thead>
<tr>
<th>EMPLOYEE_ID</th>
<th>FIRST_NAME</th>
<th>LAST_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Neena</td>
<td>Kochhar</td>
</tr>
</tbody>
</table>
Comparison Operators in the WHERE Clause

• As you saw on the previous slide, the = sign can be used in the WHERE clause.

• In addition to the "equal to" operator (=), other comparison operators can be used to compare one expression to another:
  – =    equal to
  – >   greater than
  – >=  greater than or equal to
  – <    less than
  – <=  less than or equal to
  – <>  not equal to (or != or ^=)
Comparison Operators in the WHERE Clause

• In the example below, the department_id column is used in the WHERE clause, with the comparison operator = .

• All employees with a department_id of 90 are returned.

```
SELECT employee_id, last_name, department_id
FROM employees
WHERE department_id = 90;
```

<table>
<thead>
<tr>
<th>EMPLOYEE_ID</th>
<th>LAST_NAME</th>
<th>DEPARTMENT_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>King</td>
<td>90</td>
</tr>
<tr>
<td>101</td>
<td>Kochhar</td>
<td>90</td>
</tr>
<tr>
<td>102</td>
<td>De Haan</td>
<td>90</td>
</tr>
</tbody>
</table>
Character and Date Strings in the WHERE Clause

• Character strings and dates in the WHERE clause must be enclosed in single quotation marks ' '.

• Numbers, however, should not be enclosed in single quotation marks.
Character and Date Strings in the WHERE Clause

• Look at the following example from the Employees database.

• The WHERE clause contains a string and is enclosed in single quotation marks.

```sql
SELECT first_name, last_name
FROM   employees
WHERE  last_name = 'Taylor';
```
Character and Date Strings in the WHERE Clause

• What do you think will happen if the WHERE clause is written as:

```sql
WHERE last_name = 'jones';
```

• All character searches are case-sensitive.

• Because the employees table stores all the last names in the proper case, no rows are returned in this example.
Character and Date Strings in the WHERE Clause

• This is an important point to remember.
• In another lesson, you will learn to use other SQL keywords UPPER, LOWER, and INITCAP that will make it easier to avoid a case-sensitive mistake.
Comparison Operators in the WHERE Clause

• Comparison operators can be used in all of the following ways in the WHERE clause:

WHERE hire_date < '01/Jan/2000'
WHERE salary >= 6000
WHERE job_id = 'IT_PROG'

• In the following example from the Employees database, which rows will be selected?

• Will salaries of 3000 be included in the result set?

SELECT last_name, salary
FROM employees
WHERE salary <= 3000;
Terminology

Key terms used in this lesson included:

• WHERE Clause
• Comparison Operators
Summary

In this lesson, you should have learned how to:

• Apply SQL syntax to restrict the rows returned from a query
• Demonstrate application of the WHERE clause syntax
• Explain why it is important, from a business perspective, to be able to easily limit data retrieved from a table
• Construct and produce output using a SQL query containing character strings and date values