Database Design

Working with Columns, Characters, and Rows

ORACLE ACADEMY

Objectives

This lesson covers the following objectives:

- Apply the concatenation operator to link columns to other columns, arithmetic expressions, or constant values to create a character expression
- Use column aliases to rename columns in the query result
- Enter literal values of type character, number, or date into a SELECT statement
- Define and use DISTINCT to eliminate duplicate rows
- Edit, execute, and save SQL statements in Oracle Application Express

Purpose

If you were writing an article about the Olympics, you might want to know how many different countries and how many different athletes from each country were being represented.

Having to go through lists and lists of participant names could be very tedious. Fortunately, using SQL, your job could take less than a minute. In addition, you could format your output to read like a sentence. You will find these SQL features very useful.

DESCRIBE

Use the DESCRIBE (DESC) command to display the structure of a table.

```
DESCRIBE <table_name>;
```

DESC returns the table name, table schema, tablespace name, indexes, triggers, constraints, and comments, as well as the data types, primary and foreign keys, and nullable columns.

Table	**Name**	**Type**	**Length**	**Precision**	**Scale**	**Primary Key**	**Nullable**	**Default**	**Comments**

DESCRIBE (cont.)

```
DESC departments;
```

Table	Name	Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comments
DEPARTMENTS	DEPARTMENT_ID	Number		4	0	1			Primary key column of departments table.
	DEPARTMENT_NAME	Varchar2	30						A not null column that shows name of a department...
	MANAGER_ID	Number		6	0		√		Manager_id of a department. Foreign key to employee_id column of employees table...
	LOCATION_ID	Number		4	0		√		Location id where a department is located. Foreign key to location_id column of locations table.

DESCRIBE (cont.)

This is important information when inserting new rows into a table because you must know the type of data each column will accept and whether or not the column can be left empty.

The Concatenation Operator

Concatenation means to connect or link together in a series. The symbol for concatenation is 2 vertical bars sometimes referred to as “pipes.” Values on either side of the || operator are combined to make a single output column. The syntax is:

```
string1 || string2 || string_n
```

When values are concatenated, the resulting value is a character string.

The Concatenation Operator (cont.)

In SQL, the concatenation operator can link columns to other columns, arithmetic expressions, or constant values to create a character expression. The concatenation operator is used to create readable text output.

In the following example, the department_id is concatenated to the department_name.

```
SELECT department_id || department_name
FROM departments;
```

DEPARTMENT_ID||DEPARTMENT_NAME
10Administration
20Marketing
50Shipping
60IT
…

The Concatenation Operator (cont.)

In this variation on the previous example, the || ' ' || is used to make a space between the department_id and department_name. The empty set of single quotation marks creates a space between the column values.

```
SELECT department_id || '  '  ||department_name
FROM departments;
```

```
DEPARTMENT_ID||''||DEPARTMENT_NAME
10 Administration
20 Marketing
50 Shipping
60 IT
…
```

Concatenation and Column Aliases

Column aliases are useful when using the concatenation operator so that the default SELECT line does not appear as the column heading.

```
SELECT department_id || '  ' ||
department_name  AS "Department Info"
FROM departments;
```

Department Info
10 Administration
20 Marketing
50 Shipping
60 IT
…

```
SELECT first_name ||'  ' ||
last_name AS "DJs on Demand Clients"
FROM d_clients;
```

DJs on Demand Clients
Hiram Peters
Serena Jones
Lauren Vigil

Concatenation and Literal Values

A literal value is a fixed data value such as a character, number, or date. The following are examples of literal values:

- dollars
- 1000
- January 1, 2009

Using concatenation and literal values, you can create output that looks like a sentence or statement.

Concatenation and Literal Values (cont.)

Literal values can be included in the SELECT list with the concatenation operator. Characters and dates must be enclosed in a set of single quotes ' '. Every row returned from a query with literal values will have the same character string in it.

Concatenation and Literal Values (cont.)

In the following example, King earns 24000 dollars a month. The strings, 'has a monthly salary of' and 'dollars. ', are examples of literals. If you were to create a SQL statement to produce output in this format, it would be written as:

```
SELECT last_name ||' has a monthly salary of ' || salary ||  '
dollars.' AS Pay
FROM employees;
```

PAY
King has a monthly salary of 24000 dollars.
Kochhar has a monthly salary of 17000 dollars.
De Haan has a monthly salary of 17000 dollars.
Whalen has a monthly salary of 4400 dollars.
Higgins has a monthly salary of 12000 dollars.
Gietz has a monthly salary of 8300 dollars.
…

Concatenation and Literal Values (cont.)

Note the space character following the opening quote and preceding the ending quote. What happens if you remove the spaces?

Concatenation and Literal Values (cont.)

You can also include numbers as literal values. In the following example, the number 1 is concatenated to the strings, ' has a ' and ' year salary of '.

```
SELECT last_name ||
                          ' has a ' || 1 ||
                          ' year salary of ' ||
                          salary*12 || ' dollars.' AS Pay
FROM employees;
```

PAY
King has a 1 year salary of 288000 dollars.
Kochhar has a 1 year salary of 204000 dollars.
De Haan has a 1 year salary of 204000 dollars.
Whalen has a 1 year salary of 52800 dollars.
Higgins has a 1 year salary of 144000 dollars.
…

Using DISTINCT to Eliminate Duplicate Rows

Many times, you will want to know how many unique instances of something exist. For example, what if you wanted a list of all of the departments for which there are employees. You could write a query to select the department_ids from the employees table:

```
SELECT department_id
FROM employees;
```

DEPARTMENT_ID
90
90
90
10
110
110
80
80
80
…

Note all of the duplicate rows. How can you modify the statement to eliminate the duplicate rows?

Using DISTINCT to Eliminate Duplicate Rows (cont.)

Unless you indicate otherwise, the output of a SQL query will display the results without eliminating duplicate rows. In SQL, the DISTINCT keyword is used to eliminate duplicate rows.

```
SELECT DISTINCT department_id
FROM employees;
```

DEPARTMENT_ID
90
20
110
80
50
10
60

The DISTINCT qualifier affects all listed columns and returns every distinct combination of the columns in the SELECT clause. The keyword DISTINCT must appear directly after the SELECT keyword.

EXECUTE, SAVE, and EDIT in Oracle Application Express

Now that you have been using Oracle Application Express to create and execute statements, it would be nice if you could save those statements for later so you could run them again or perhaps edit them slightly and then save a new copy of the statement.

Oracle Application Express has facilities to do just that. Your teacher will demonstrate these facilities for you, and you can find further information in the Oracle Application Express User Guide.

Summary

In this lesson, you should have learned how to:

- Apply the concatenation operator to link columns to other columns, arithmetic expressions, or constant values to create a character expression
- Use column aliases to rename columns in the query result
- Enter literal values of type character, number, or date into a SELECT statement
- Define and use DISTINCT to eliminate duplicate rows
- Edit, execute, and save SQL statements in Oracle Application Express