Database Programming with SQL

5-3
Conditional Expressions
Objectives

This lesson covers the following objectives:

• Compare and contrast the DECODE and CASE functions

• Construct and execute a SQL query that correctly uses the DECODE and CASE functions

• Construct and execute two methods for implementing IF-THEN-ELSE conditional logic
Purpose

• Being able to make decisions is essential in data modeling.
• Modelers have to decide which business functions need to be modeled and which do not.
• The data-modeling process requires that designers analyze information to identify entities, resolve relationships, and select attributes.
• A typical decision could be:
  – IF a business needs to track data over time, THEN time may need to be an entity or ELSE time should be an attribute.
How Functions are Evaluated

• This decision-making process in programming is not much different from the process that we use in everyday life.

• Think of the last time you had to make an if-then-else kind of decision.

• IF I get my homework done before 9:00 p.m., I can watch television, ELSE I can't watch television.

• In SQL, these kinds of choices involve conditional-processing methods.

• Knowing how to use conditional processing makes decision making to get the data you want easier.
Conditional Expressions

• The two conditional expressions are CASE and DECODE.

• You have already studied NULLIF, which is logically equivalent to the CASE expression in that CASE compares two expressions.

• NULLIF compares two expressions, and if the two expressions are equal, then return null; if they are not equal, then return the first expression.
Conditional Expressions

• There are two sets of commands or syntax that can be used to write SQL statements:
  – ANSI/ISO SQL 99 compliant standard statements
  – Oracle proprietary statements

• The two sets of syntax are very similar, but there are a few differences.

• In this course, you will learn to use both sets of SQL statements, but the use of ANSI/ISO SQL 99 syntax is recommended.
Conditional Expressions

• CASE and DECODE are examples of one of these differences.
• CASE is an ANSI/ISO 99 SQL 99 compliant statement.
• DECODE is an Oracle Proprietary statement.
• Both statements return the same information using different syntax.
CASE Expression

• The CASE expression basically does the work of an IF-THEN-ELSE statement.

• Data types of the CASE, WHEN, and ELSE expressions must be the same.

• The syntax for a CASE expression is:

```sql
CASE expr WHEN comparison_expr1 THEN return_expr1
    [WHEN comparison_expr2 THEN return_expr2
     WHEN comparison_exprn THEN return_exprn
     ELSE else_expr]
END
```
CASE Syntax

- The query checks the department_id.
  - IF it is 90, then return 'Management'
  - IF it is 80, then return 'Sales'
  - IF it is 60, then return 'It'
  - ELSE return 'Other dept.'

```
SELECT last_name,
CASE department_id
  WHEN 90 THEN 'Management'
  WHEN 80 THEN 'Sales'
  WHEN 60 THEN 'It'
  ELSE 'Other dept.'
END AS "Department"
FROM employees;
```
DECODE Expression

- The DECODE function evaluates an expression in a similar way to the IF-THEN-ELSE logic.
- DECODE compares an expression to each of the search values.
- The syntax for DECODE is:

  ```sql
  DECODE(column1|expression, search1, result1
          [, search2, result2,...,]
          [, default])
  ```

- If the default value is omitted, a null value is returned where a search value does not match any of the values.
DECODE Expression

- Examine the example:

```sql
SELECT last_name,
       DECODE(department_id,
               90, 'Management',
               80, 'Sales',
               60, 'It',
               'Other dept.' )
AS "Department"
FROM employees;
```

- This query returns exactly the same results as the previous CASE example, but using different syntax.
Terminology

Key terms used in this lesson included:

• CASE
• Conditional expression
• DECODE
Summary

In this lesson, you should have learned how to:

• Compare and contrast the DECODE and CASE functions
• Construct and execute a SQL query that correctly uses the DECODE and CASE functions
• Construct and execute two methods for implementing IF-THEN-ELSE conditional logic