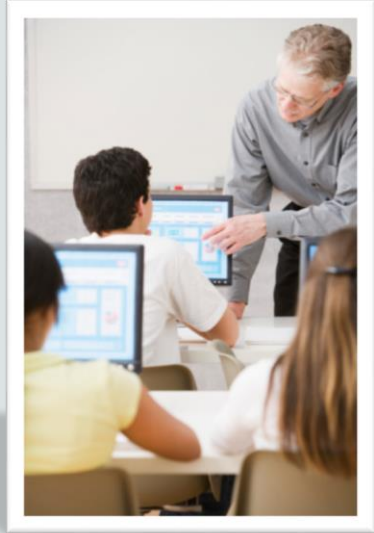




Programación de Bases de Datos con SQL

5-3

Expresiones Condicionales



Objetivos

En esta lección se abordan los siguientes objetivos:

- Comparar y contrastar las funciones DECODE y CASE
- Crear y ejecutar una consulta SQL que utiliza correctamente las funciones DECODE y CASE
- Crear y ejecutar dos métodos para implantar la lógica condicional IF-THEN-ELSE

Objetivo

- Poder tomar decisiones es esencial en el modelado de datos.
- Los modeladores tienen que decidir qué funciones de negocio se deben modelar y cuáles no.
- En el proceso de modelado de datos es necesario que los diseñadores analicen la información para identificar las entidades, resolver las relaciones y seleccionar los atributos.
- Una decisión típica podría ser:
 - (IF) Si una empresa necesita realizar un seguimiento de los datos a lo largo del tiempo, (THEN) En ese caso, el tiempo puede que necesite ser una entidad o (ELSE) En caso contrario, el tiempo debería ser un atributo.

Método de Evaluación de las Funciones

- Este proceso de toma de decisiones en programación no es muy diferente del proceso que se utiliza en la vida diaria.
- Piense en la última vez que tuvo que tomar una decisión del tipo if-then-else.
- Si (IF) termino mis deberes antes de las 9:00 p.m., puedo ver la televisión; en caso contrario (ELSE), no puedo ver la televisión.
- En SQL, estos tipos de elecciones implican métodos de procesamiento condicionales.
- Saber cómo utilizar el procesamiento condicional facilita la toma de decisiones para obtener los datos que desee.

Expresiones Condicionales

- Las dos expresiones condicionales son CASE y DECODE.
- Ya ha estudiado NULLIF, que es el equivalente lógico a la expresión CASE, ya que CASE compara dos expresiones.
- NULLIF compara dos expresiones y, si las dos expresiones son iguales, devuelve null; si no son iguales, devuelve la primera expresión.

Expresiones Condicionales

- Hay dos juegos de comandos o sintaxis que se pueden utilizar para escribir sentencias SQL:
 - Sentencias estándar compatibles con ANSI/ISO SQL 99
 - Sentencias propiedad de Oracle
- Los dos juegos de sintaxis son muy similares, pero hay unas cuantas diferencias.
- En este curso, aprenderá a utilizar ambos juegos de sentencias SQL, pero se recomienda utilizar la sintaxis ANSI/ISO SQL 99.

Es importante que los alumnos también aprendan sintaxis propiedad de Oracle, ya que es posible que se encuentren con esta sintaxis en bases de datos anteriores.

Expresiones Condicionales

- CASE y DECODE son ejemplos de una de estas diferencias.
- CASE es una sentencia compatible con ANSI/ISO 99 SQL 99.
- DECODE es una sentencia propiedad de Oracle.
- Ambas sentencias devuelven la misma información utilizando sintaxis diferente.

Expresión CASE

- La expresión CASE básicamente realiza la función de una sentencia IF-THEN-ELSE.
- Los tipos de dato de las expresiones CASE, WHEN y ELSE deben ser iguales.
- La sintaxis de una expresión CASE es:

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

Sintaxis de CASE

- La consulta comprueba el valor `department_id`.
 - Si (IF) es 90, (then) devuelve 'Management'
 - Si (IF) es 80, (then) devuelve 'Sales'
 - Si (IF) es 60, (then) devuelve 'It'
 - En caso contrario (ELSE), devuelve '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;
```

LAST_NAME	Department
King	Management
Kochhar	Management
De Haan	Management
Whalen	Other dept.
Higgins	Other dept.
Gietz	Other dept.
Zlotkey	Sales
Abel	Sales
Taylor	Sales
Grant	Other dept.
Mourgos	Other dept.
Rajs	Other dept.
Davies	Other dept.
Matos	Other dept.
Vargas	Other dept.
Hunold	It
Ernst	It
Lorentz	It
Hartstein	Other dept.
Fay	Otro depart.

Expresión DECODE

- La función DECODE evalúa una expresión de una forma similar a la lógica IF-THEN-ELSE.
- DECODE compara una expresión con cada uno de los valores de búsqueda.
- La sintaxis de DECODE es:

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

- Si se omite el valor por defecto, se devuelve un valor nulo donde un valor de búsqueda no coincida con ninguno de los valores.

Expresión DECODE

- Examine el ejemplo:

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

- Esta consulta devuelve exactamente los mismos resultados que el ejemplo CASE anterior, pero utilizando sintaxis diferente.

LAST_NAME	Department
King	Aplicaciones de
Kochhar	Aplicaciones de
De Haan	Aplicaciones de
Whalen	Other dept.
Higgins	Other dept.
Gietz	Other dept.
Zlotkey	Sales
Abel	Sales
Taylor	Sales
Grant	Other dept.
Mourgos	Other dept.
Rajs	Other dept.
Davies	Other dept.
Matos	Other dept.
Vargas	Other dept.
Hunold	It
Ernst	It
Lorentz	It
Hartstein	Other dept.
Fay	Other dept.

Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- CASE
- Expresión condicional
- DECODE

Resumen

En esta lección, debe haber aprendido lo siguiente:

- Comparar y contrastar las funciones DECODE y CASE
- Crear y ejecutar una consulta SQL que utiliza correctamente las funciones DECODE y CASE
- Crear y ejecutar dos métodos para implantar la lógica condicional IF-THEN-ELSE

