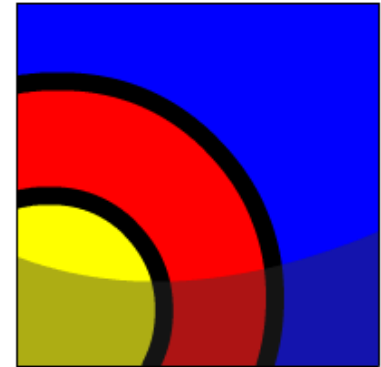


Conditional Expressions

What Will I Learn?

In this lesson, you will learn 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





Why Learn It?

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 the designers to 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.





Why Learn It?

This decision process is not much different from those we make 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.





Tell Me / Show Me

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. If the two expressions are equal, then return null; if they are not equal, then return the first expression.

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.



Tell Me / Show Me

The syntax for CASE is:

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

```
SELECT id, loc_type, rental_fee,  
    CASE loc_type WHEN 'Private Home' THEN 'No Increase'  
    WHEN 'Hotel' THEN 'Increase 5%' ELSE rental_fee  
    END AS "REVISED_FEES"  
FROM d_venues;
```

ID	LOC_TYPE	RENTAL_FEE	REVISED_FEES
100	Private Home	0	No increase
105	Private Home	0	No increase
101	Private Home	0	No increase
95	School Hall	75/hour	75/hour
99	National Park	400/flat fee	400/flat fee
220	Hotel	300/per person	Increase 5%



Tell Me / Show Me

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:

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

ID	LOC_TYPE	RENTAL_FEE	REVISED_FEES
100	Private Home	0	No increase
105	Private Home	0	No increase
101	Private Home	0	No increase
95	School Hall	75/hour	75/hour
99	National Park	400/flat fee	400/flat fee
220	Hotel	300/per person	Increase 5%

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



Tell Me / Show Me

DECODE Expression (continued)

Examine the example:

```
SELECT id, loc_type, rental_fee,  
       DECODE( loc_type, 'Private Home', 'No  
Increase',  
              'Hotel', 'Increase 5%',  
              rental_fee )  
AS "REVISED_FEES"  
FROM   d_venues;
```

ID	LOC_TYPE	RENTAL_FEE	REVISED_FEES
100	Private Home	0	No increase
105	Private Home	0	No increase
101	Private Home	0	No increase
95	School Hall	75/hour	75/hour
99	National Park	400/flat fee	400/flat fee
220	Hotel	300/per person	Increase 5%

Tell Me / Show Me

Terminology

Key terms used in this lesson include:

Conditional expression

CASE

DECODE

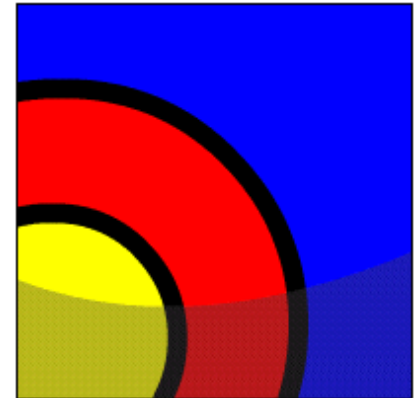




Summary

In this lesson you have learned 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



Summary

Practice Guide

The link for the lesson practice guide can be found in the course outline.

