Database Design

Comparison Operators

Objectives

This lesson covers the following objectives:

- Apply the proper comparison operator to return a desired result
- Demonstrate proper use of BETWEEN, IN, and LIKE conditions to return a desired result
- Distinguish between zero and NULL, the latter of which is unavailable, unassigned, unknown, or inapplicable
- Explain the use of comparison conditions and NULL

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Purpose

We use comparisons in everyday conversation without really thinking about it.

- "I can meet you BETWEEN 10:00 a.m. and 11:00 a.m."
- "I'm looking for a pair of jeans LIKE the ones you are wearing."
- "If I remember correctly, the best concert seats are IN rows 100, 200, and 300.“

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Purpose (cont.)

The need to express these types of comparisons also exists in SQL. Comparison conditions are used to find data in a table meeting certain conditions. Being able to formulate a SELECT clause to return specific data is a powerful feature of SQL.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Comparison Operators

You are already familiar with the comparison operators such as equal to (=), less than (<), and greater than (>). SQL has other operators that add functionality for retrieving specific sets of data.

These include:

- BETWEEN…AND
- IN
- LIKE

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

BETWEEN...AND

The BETWEEN...AND operator is used to select and display rows based on a range of values. When used with the WHERE clause, the BETWEEN...AND condition will return a range of values between and inclusive of the specified lower and upper limits.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

BETWEEN…AND (cont.)

Note in the example from the DJs on Demand database, the values returned include the lower-limit value and the upper-limit value. Values specified with the BETWEEN condition are said to be inclusive. Note also that the lower-limit value must be listed first.

```
SELECT title, year
FROM   d_cds
WHERE year BETWEEN 1999 AND 2001;
```

Note that the output included the lower-limit and upper-limit values.

TITLE	YEAR
Party Music for All Occasions	2000
Songs from My Childhood	1999
Carpe Diem	2000
Here Comes the Bride	2001

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

BETWEEN…AND (cont.)

Using BETWEEN...AND is the same as using the following expression:

```
WHERE year >= 1999 AND year <= 2001
```

In fact, there is no performance benefit in using one expression over the other. We use BETWEEN...AND for simplicity in reading the code.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

IN

The IN condition is also known as the "membership condition." It is used to test whether a value is IN a specified set of values.

For example, IN could be used to identify students whose identification numbers are 2349, 7354, or 4333 or people whose international phone calling code is 1735, 82, or 10.

```
SELECT title, type_code
FROM  d_songs
WHERE type_code IN (77,12);
```

TITLE	TYPE_CODE
Its Finally Over	12
I'm Going to Miss My Teacher	12
Hurrah for Today	77
Let's Celebrate	77

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

IN (cont.)

In this example, the WHERE clause could also be written as a set of OR conditions:

```
SELECT title, type_code
FROM   d_songs
WHERE  type_code IN ( 77, 12 );
…
WHERE type_code = 77 OR type_code = 12;
```

As with BETWEEN...AND, the IN condition can be written using either syntax just as efficiently.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

LIKE

Have you ever gone shopping to look for something that you saw in a magazine or on television but you weren't sure of the exact item? It's much the same with database searches.

A manager may know that an employee's last name starts with "S" but doesn't know the employee's entire name. Fortunately, in SQL, the LIKE condition allows you to select rows that match either characters, dates, or number patterns. Two symbols -- the (%) and the underscore (_) -- called wildcard characters, can be used to construct a search string.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

LIKE (cont.)

The percent (%) symbol is used to represent any sequence of zero or more characters. The underscore (_) symbol is used to represent a single character.

In the example shown below, all employees with last names beginning with any letter followed by an "o" and then followed by any other number of letters will be returned.

```
SELECT last_name
FROM   employees
WHERE  last_name LIKE '_o%';
```


Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

LIKE (cont.)

```
SELECT last_name
FROM   employees
WHERE  last_name LIKE '_o%';
```

Which of the following last names could have been returned from the above query?

1. Sommersmith
2. Oog
3. Fong
4. Mo

If you said 1, 2, 3, and 4, you are correct!

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

LIKE (cont.)

One additional option that is important: When you need to have an exact match for a string that has a % or _ character in it, you will need to indicate that the % or the _ is not a wildcard but is part of the item you're searching for.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

LIKE (cont.)

The ESCAPE option backward slash (\) is used to indicate that the underscore or % is part of the name, not a wildcard value. For example, if the database had stored CD track numbers as TRA_6, the WHERE clause would need to be written as:

```
WHERE track LIKE 'TRA\_%'
```


Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

IS NULL, IS NOT NULL

Remember NULL? It is the value that is unavailable, unassigned, unknown, or inapplicable. Being able to test for NULL is often desirable.

You may want to know all the dates in June that, right now, do not have a concert scheduled. You may want to know all of the clients who do not have phone numbers recorded in your database.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

IS NULL, IS NOT NULL (cont.)

The IS NULL condition tests for unavailable, unassigned, or unknown data. IS NOT NULL tests for data that is available in the database.

In the example on the next slide, the WHERE clause is written to retrieve all the last names and manager IDs of those employees who do not have a manager.

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

IS NULL, IS NOT NULL (cont.)

```
SELECT last_name, manager_id
FROM   employees
WHERE  manager_id IS NULL;
```

Read the following and explain what you expect will be returned:

```
SELECT first_name, last_name, auth_expense_amt
FROM d_partners
WHERE auth_expense_amt IS NULL;
```


Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

Summary

In this lesson, you should have learned how to:

- Apply the proper comparison operator to return a desired result
- Demonstrate proper use of BETWEEN, IN, and LIKE conditions to return a desired result
- Distinguish between zero and NULL, the latter of which is unavailable, unassigned, unknown, or inapplicable
- Explain the use of comparison conditions and NULL

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.