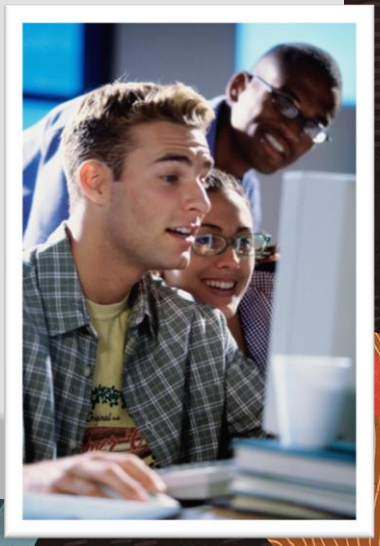

ORACLE
Academy

Database Programming with SQL

7-2

Oracle Nonequijoins and Outer Joins



ORACLE
Academy

Objectives

- This lesson covers the following objectives:
  - Construct and execute a SELECT statement to access data from more than one table using a nonequijoin
  - Create and execute a SELECT statement to access data from more than one table using an Oracle outer join

Purpose

- What happens if you want to retrieve data from a table that has no corresponding column in another table?
- For instance, your math percentage grade of 92 is stored in the GRADES column in one table; the letter grade is stored in the LETTER_GRADE column in another table
- How can we join the number grade with the letter grade?
- When data is recorded using a range, retrieving it is the job of a nonequijoin

Purpose

- The Oracle joins you've studied so far returned rows that had a matching value in both tables
- Those rows that didn't satisfy these conditions were just left out
- Sometimes, however, you want all the data from one of the tables to be returned even if no data matches in the other table
- In this lesson will also look at the Oracle Outer Joins to solve this issue

Nonequijoin

- Example:
  - Suppose we want to know the grade_level for each employee's salary
  - The job_grades table does not have a common column with the employees table
  - Using a nonequijoin allows us to join the two tables

job_grades table

GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
A	1000	2999
B	3000	5999
C	6000	9999
D	10000	14999
E	15000	24999
F	25000	40000

Nonequijoin

- Since there is no exact match between the two columns in each table, the equality operator = can't be used
- Although comparison conditions such as <= and >= can be used, BETWEEN...AND is a more effective way to execute a nonequijoin
- A nonequijoin is equivalent to an ANSI JOIN ON (where the condition used is something other than equals)

Nonequijoin

```
SELECT last_name, salary, grade_level, lowest_sal,
highest_sal
FROM employees, job_grades
WHERE (salary BETWEEN lowest_sal AND highest_sal);
```

LAST_NAME	SALARY	GRADE_LEVEL	LOWEST_SAL	HIGHEST_SAL
Vargas	2500	A	1000	2999
Matos	2600	A	1000	2999
Davies	3100	B	3000	5999
Rajs	3500	B	3000	5999
Lorentz	4200	B	3000	5999
Whalen	4400	B	3000	5999
Mourgos	5800	B	3000	5999
Fay	6000	C	6000	9999
…				

Outer Join

- An outer join is used to see rows that have a corresponding value in another table plus those rows in one of the tables that have no matching value in the other table
- To indicate which table may have missing data using Oracle Join Syntax, add a plus sign (+) after the table's column name in the WHERE clause of the query

Left Outer Join

- This query will return all employee last names, including those that are assigned to a department and those that are not
- The same results could be obtained using an ANSI LEFT OUTER JOIN

```
SELECT e.last_name,
d.department_id,
 d.department_name
FROM employees e, departments d
 WHERE e.department_id =
    d.department_id(+);
```

LAST_NAME	DEPT_ID	DEPT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing
Vargas	50	Shipping
…		
Higgins	110	Accounting
Grant	-	-

Right Outer Join

- This outer join would return all department IDs and department names, both those that have employees assigned to them and those that do not
- The same results could be obtained using an ANSI RIGHT OUTER JOIN

```
SELECT e.last_name,
d.department_id,
 d.department_name
FROM employees e, departments d
 WHERE e.department_id(+) =
    d.department_id;
```

LAST_NAME	DEPT_ID	DEPT_NAME
Whalen	10	Administration
Hartstein	20	Marketing
Fay	20	Marketing
Mourgos	50	Shipping
…		
Gietz	110	Accounting
-	190	Contracting

ORACLE Academy

Full Outer Join

- It is not possible to have the equivalent of a FULL OUTER JOIN by adding a (+) sign to both columns in the join condition
- Attempting this results in an error

```
SELECT e.last_name, d.department_id, d.department_name
FROM employees e, departments d
WHERE e.department_id(+) =  d.department_id(+);
```

ORA-01468: a predicate may reference only one outer-joined table

It is possible to perform a full outer join using Set Operators. These are covered later in the course.

Outer Join

- The syntax variations of the outer join are shown

```
SELECT table1.column, table2.column
FROM table1, table2
WHERE table1.column = table2.column(+);
```

```
SELECT table1.column, table2.column
FROM table1, table2
WHERE table1.column(+) = table2.column;
```

```
SELECT table1.column, table2.column
FROM table1, table2
NEVER table1.column(+) = table2.column(+);
```

X

Outer Join and ANSI equivalents

- The table below shows ANSI/ISO SQL: 99 joins and their equivalent Oracle outer joins

ANSI/ISO SQL	Oracle Syntax
LEFT OUTER JOIN departments d ON (e.department_id = d.department_id);	WHERE e.department_id = d.department_id(+);
RIGHT OUTER JOIN departments d ON (e.department_id = d.department_id);	WHERE e.department_id(+) = d.department_id;
FULL OUTER JOIN departments d ON (e.department_id = d.department_id);	No direct equivalent.

Terminology

- Key terms used in this lesson included:
  - Nonequijoin
  - BETWEEN…AND
  - Outer Joins

Summary

- In this lesson, you should have learned how to:
  - Construct and execute a SELECT statement to access data from more than one table using a nonequijoin
  - Create and execute a SELECT statement to access data from more than one table using an Oracle outer join


ORACLE
Academy