Database Programming with PL/SQL

Review of SQL Joins

ORACLE ACADEMY

Objectives

In this lesson, you will review how to construct and execute SELECT statements:

- To access data from more than one table using an equality operator
- To access data from more than one table using non-equality comparison operators
- To access data from more than one table using LEFT and RIGHT outer joins
- To access data from more than one table creating a cartesian product using a cross join

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

Purpose

Taking time to review previously learned material helps you to reinforce basic concepts and prepares you for more complicated constructs.

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

Associating Tables

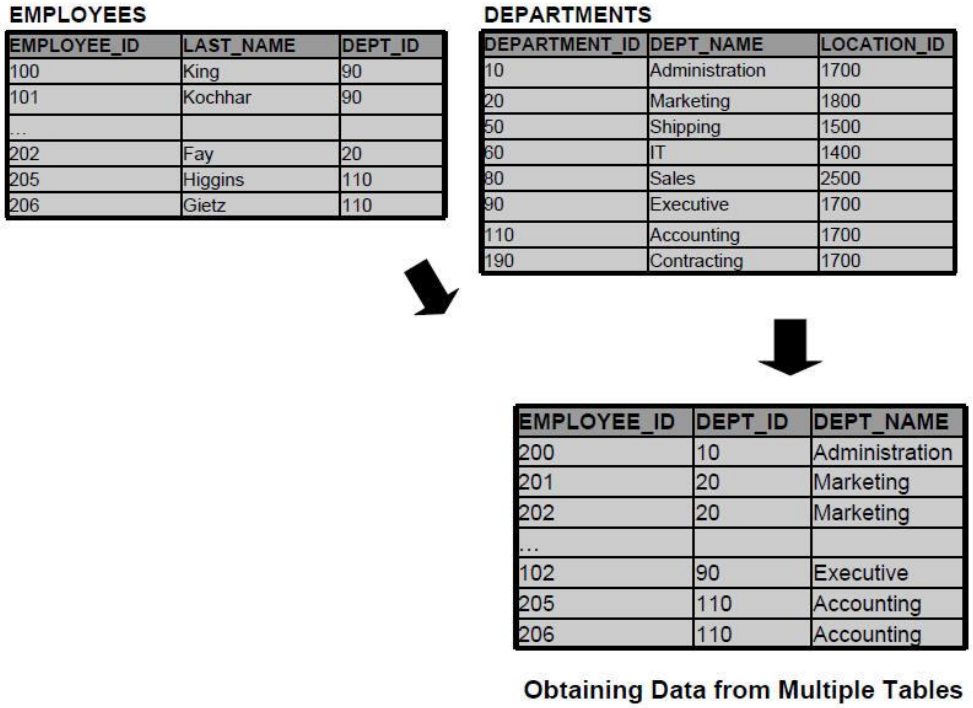
Up to now, your experience using SQL has been limited to querying and returning information from one database table at a time.

This would not be a problem if all data in the database were stored in only one table. But you know from data modeling that separating data into individual tables and being able to associate the tables with one another is the heart of relational database design.

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

Associating Tables (cont.)

Fortunately, SQL provides join conditions that enable information to be queried from separate tables and combined in one report.



EMPLOYEES

EMPLOYEE_ID	LAST_NAME	DEPT_ID
100	King	90
101	Kochhar	90
...		
202	Fay	20
205	Higgins	110
206	Gietz	110

DEPARTMENTS

DEPARTMENT_ID	DEPT_NAME	LOCATION_ID
10	Administration	1700
20	Marketing	1800
50	Shipping	1500
60	IT	1400
80	Sales	2500
90	Executive	1700
110	Accounting	1700
190	Contracting	1700

EMPLOYEE_ID	DEPT_ID	DEPT_NAME
200	10	Administration
201	20	Marketing
202	20	Marketing
...		
102	90	Executive
205	110	Accounting
206	110	Accounting

Obtaining Data from Multiple Tables

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

Natural Join

A natural join combines rows that have equal values for the specified columns. This is sometimes called a "simple" join. The basic syntax is:

```
SELECT table1.column, table2.column ...
FROM table1, table2
WHERE table1.column1 = table2.column2;
```


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

Natural Join (cont.)

The example shown joins two tables, wf_world_regions and wf_countries, to display the region name alongside the country name.

```
SELECT region_name, country_name, airports
FROM wf_world_regions NATURAL
JOIN wf_countries
ORDER BY country_name;
```

REGION_NAME	COUNTRY_NAME	AIRPORTS
Caribbean	Anguilla	3
Oceania	Antarctica	28
Caribbean	Antiqua and Barbuda	3
Caribbean	Aruba	1
Western Europe	Bailiwick of Guernsey	2
Western Europe	Bailiwick of Jersey	1

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

Nonequijoin

Nonequijoin combines tables that have no exact matching columns. This join is often used when a column value in table A falls between a range of values specified by two columns in table B. Examine the following tables:

```
SELECT employee_id,
last_name,salary
FROM employees;
```

EMPLOYEE_ID	LAST_NAME	SALARY
100	King	24000
101	Kochlar	17000
102	De Haan	17000
103	Hunold	9000
104	Ernst	6000
107	Lorentz	4200

```
SELECT *
FROM job_grades;
```

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

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

Nonequijoin (cont.)

In this example, the grade for each student is matched up with the equivalent grade code.

```
SELECT e.employee_id, e.last_name, j.grade_level
   FROM employees e JOIN job_grades j
   ON e.salary BETWEEN j.lowest_sal AND j.highest_sal;
```

EMPLOYEE_ID	LAST_NAME	GRADE_LEVEL
144	Vargas	A
143	Matos	A
142	Davies	B
141	Rajs	B
107	Lorentz	B
200	Whalen	B

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

INNER AND OUTER JOINS

In ANSI-99 SQL, a join of two or more tables that return only matched rows is called an inner join.

When a join returns the unmatched rows as well as matched rows, it is called an outer join.

Outer join syntax uses the terms “left, full, and right.” These names are associated with the order of the table names in the FROM clause of the SELECT statement.

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

LEFT AND RIGHT OUTER JOINS

In the example shown of a left outer join, note that the table name listed to the left of the words “left outer join" is referred to as the "left table.”

This query will return all matched rows as well as all employee last names even if they aren’t assigned to a department.

```
SELECT e.last_name, d.department_id, d.department_name
FROM employees e
LEFT OUTER JOIN departments d
ON (e.department_id = d.department_id);
```

LAST_NAME	DEPT_ID	DEPT_NAME
King	90	Executive
Kochhar	90	Executive
…		
Whalen	10	Administration
Hartstein	20	Marketing
Fay	20	Marketing
Higgins	110	Accounting
Gietz	110	Accounting
Grant		

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

LEFT AND RIGHT OUTER JOINS (cont.)

This right outer join would return all department IDs and department names even if no employees were assigned to them.

```
SELECT e.last_name, d.department_id, d.department_name
FROM employees e
RIGHT OUTER JOIN departments d
ON (e.department_id = d.department_id);
```

LAST_NAME	DEPT_ID	DEPT_NAME
King	90	Executive
Kochhar	90	Executive
…		
Whalen	10	Administration
Hartstein	20	Marketing
Fay	20	Marketing
Higgins	110	Accounting
Gietz	110	Accounting
	190	Contracting

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

CROSS JOIN

An Oracle Proprietary Cartesian Product joins every row in one table to every row in the other table.

The ANSI/ISO SQL: 1999 SQL equivalent of the Cartesian product is the cross join.

The results returned from both types of joins are the same. The results set represents all possible combinations of columns from both tables. This could potentially be very large!

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

Cross Join Example

```
SELECT name, event_date, loc_type, rental_fee
FROM d_events CROSS JOIN d_venues;
```

NAME	EVENT_DATE	LOC_TYPE	RENTAL_FEE
Peters Graduation	14-MAY-04	Private Home	0
Peters Graduation	14-MAY-04	Private Home	0
Peters Graduation	14-MAY-04	Private Home	0
Peters Graduation	14-MAY-04	School Hall	75/hour
Peters Graduation	14-MAY-04	National Park	400/flat fee
Peters Graduation	14-MAY-04	Hotel	300/per person
Vigil Wedding	28-APR-04	Private Home	0
Vigil Wedding	28-APR-04	Private Home	0
Vigil Wedding	28-APR-04	Private Home	0
Vigil Wedding	28-APR-04	School Hall	75/hour
Vigil Wedding	28-APR-04	National Park	400/flat fee
Vigil Wedding	28-APR-04	Hotel	300/per person

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

Terminology

Key terms used in this lesson included:

- Natural Join
- ON Clause
- LEFT Outer Join
- RIGHT Outer Join
- Cross Join

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

Summary

In this lesson, you should have learned how to:

- To access data from more than one table using an equality operator
- To access data from more than one table using non-equality comparison operators
- To access data from more than one table using LEFT and RIGHT outer joins
- To access data from more than one table creating a cartesian product using a cross join

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