10-06-2024 Displaying Data from Multiple Tables(joins)

Types of Joins

Oracle Proprietary Joins (8*i* and prior):

- Equijoin
- Nonequijoin
- Outer join
- Self join

SQL: 1999 Compliant Joins:

- Cross joins
- Natural joins
- Using clause
- Full or two sided outer joins
- Arbitrary join conditions for outer joins

Retrieving Records with Equijoins

1.SELECT employees.employee_id, employees.last_name,

employees.department_id, departments.department_id,

departments.location_id

FROM employee7, department7

WHERE employees.department_id = departments.department_id;

Additional Search Conditions

2. SELECT last_name, employees.department_id,department_name FROM employees, departments WHERE employees.department_id = departments.department_id AND last_name = 'Matos';

Using Table Aliases

3. SELECT e.employee_id, e.last_name, e.department_id,d.department_id, d.location_id

FROM employee7 e, department7 d WHERE e.department_id = d.department_id;

4.SELECT e.last_name, d.department_name, l.city FROM employees e, departments d, locations lWHERE e.department_id = d.department_id AND d.location_id = l.location_id;

Retrieving Records with Nonequijoins

5.SELECT e.last_name, e.salary, j.grade_level FROM employees e, job_grades j

WHERE e.salary BETWEEN j.lowest_sal AND j.highest_sal;

Outer Joins

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

Joining a Table to Itself

7.SELECT worker.last_name || ' works for '|| manager.last_name FROM employees worker, employees manager WHERE worker.manager_id = manager.employee_id;

Creating Cross Joins

8.SELECT last_name, department_name FROM employees CROSS JOIN departments;

Retrieving Records with Natural Joins

9.SELECT department_id, department_name,location_id, city FROM departments

NATURAL JOIN locations;

USING CLAUSE

10.SELECT e.employee_id, e.last_name, d.location_idFROM employees e JOIN departments d USING (department_id);

ON CLAUSE

11.SELECT e.employee_id, e.last_name, e.department_id,d.department_id, d.location_id

FROM employees e JOIN departments dON (e.department_id = d.department_id);

LEFT OUTER JOIN

12.SELECT e.last_name, e.department_id, d.department_name

FROM employees eLEFT OUTER JOIN departments d

ON (e.department_id = d.department_id)

RIGHT OUTER JOIN

13.SELECT e.last_name, e.department_id, d.department_name

FROM employees e

RIGHT OUTER JOIN departments d

ON (e.department_id = d.department_id);

FULL OUTER JOIN

14.SELECT e.last_name, e.department_id, d.department_name

FROM employees e

FULL OUTER JOIN departments d

ON (e.department_id = d.department_id);