10/9/25

EXERCISE-7

Displaying data from multiple tables

Objective AIM: To display data from multiple tables using SQL Queries

After the completion of this exercise, the students will be able to do the following:
• Write SELECT statements to access data from more than one table using equality and nonequality joins
• View data that generally does not meet a join condition by using outer joins
• Join a table to itself by using a self join
Sometimes you need to use data from more than one table.

Cartesian Products
• A Cartesian product is formed when:
– A join condition is omitted
– A join condition is invalid
– All rows in the first table are joined to all rows in the second table
• To avoid a Cartesian product, always include a valid join condition in a WHERE clause.
A Cartesian product tends to generate a large number of rows, and the result is rarely useful. You should always include a valid join condition in a WHERE clause, unless you have a specific need to combine all rows from all tables.
Cartesian products are useful for some tests when you need to generate a large number of rows to simulate a reasonable amount of data.

Example:
To displays employee last name and department name from the EMPLOYEES and DEPARTMENTS tables.

```
SELECT last_name, department_name dept_name
FROM   employees, departments;
```

Types of Joins

• Equijoin
• Non-equijoin
• Outer join
• Self join
• Cross joins
• Natural joins
• Using clause
• Full or two sided outer joins
• Arbitrary join conditions for outer joins

Joining Tables Using Oracle Syntax

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

Write the join condition in the WHERE clause.
• Prefix the column name with the table name when the same column name appears in more than one table.

This query retrieves all rows in the EMPLOYEES table, even if there is no match in the DEPARTMENTS table. It alslso retrieves all rows in the DEPARTMENTS table, even if there is no match in the EMPLOYEES table.

Find the Solution for the following:

1. Write a query to display the last name, department number, and department name for all employees.

Select e.lname, e. dept - id, d. dept - name FROM employees e JOIN departments d ON e.department - id = d. department -id;

2. Create a unique listing of all jobs that are in department 80. Include the location of the department in the output.

Select distinct e. job_id, l. city FROM employees e JOIN departments d ON e. department - id = d. department -id JOIN locations l ON d. location - id = l. location - id WHERE e. department - id = 80;

3. Write a query to display the employee last name, department name, location ID, and city of all employees who earn a commission

Select e. last - name, d. department - name, d. location - id, l. city, FROM employees e JOIN departments d ON e. department - id = d. department - id JOIN locations l ON d. location - id = l. location - id WHERE e. commission - pct is NOT NULL;

8. Display the employee last name and department name for all employees who have an a(lowercase) in their last names. P

Select e. lname, d. dname FROM employees e JOIN departments d ON e. dept - id = d. dept - id WHERE e. lname LIKE '%a%';

5. Write a query to display the last name, job, department number, and department name for all employees who work in Toronto.

Select e. lname, e. job - id, e. dept - id, d. dept - name FROM employees e JOIN departments d ON e. dept - id = d. dept - id JOIN locations l ON d. location - id = l. location - id WHERE l. city = 'Toronto';

6. Display the employee last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, Respectively

Select e. l - name AS employee, e. emp - id AS 'EMP#', m. lname AS Manager, e. manager - id AS 'Mgr#' FROM employees e JOIN employees m ON e. manager - id = m. employee - id;

7

f_name
Rahul
Ben
King
Sengathir
Rajesh

8 ,

Employee	dept_number
Patel	345
Raj	998

9

f_name	job_name	dept_name	Salary	Grade level
Ben	Sales representative	Sales	1400	C
Sengathir	Stock	Sales	7000	O
Rahul	Stock	Stock	50000	A

7. Modify lab4_6.sql to display all employees including King, who has no manager. Order the results by the employee number.

```
Select e.lname AS employee, e.emp-id AS 'Emp #',
m.lname AS Manager, e.manager-id AS 'Mgr #' FROM
employees e LEFT JOIN employees m ON e.manager-id
=m.emp-id ORDER BY e.emp-id;
```

8. Create a query that displays employee last names, department numbers, and all the employees who work in the same department as a given employee. Give each column an appropriate label

```
Select e.lname, e.dept-id, e2.lname AS colleague FROM
employees e JOIN employees e2 ON e.dept-id = e2.dept-id
WHERE e.lname='Taylor';
```

9. Show the structure of the JOB_GRADES table. Create a query that displays the name, job, department name, salary, and grade for all employees

```
DESC job-grades;
Select e.lname, e.job-id, d.dept-name, e.salary, j.grade-level,
FROM employees e JOIN departments d ON e.dept-id=d.dept-id
JOIN job-grades j ON e.salary BETWEEN j.low-sal AND j.high-sal;
```

10. Create a query to display the name and hire date of any employee hired after employee Davies.

```
Select lname, hire-date FROM employees WHERE hire-date >
(select hire-date FROM employees WHERE last-name='Davies');
```

11. Display the names and hire dates for all employees who were hired before their managers, along with their manager's names and hire dates. Label the columns Employee, Emp Hired, Manager, and Mgr Hired, respectively.

```
Select e.lname AS Employee, e.hire-date AS 'Emp Hired',
m.lname AS Manager, m.hire-date AS 'Mgr Hired' FROM
employees e JOIN employees m ON e.manager-id=
m.emp-id WHERE e.hire-date < m.hire-date;
```

10

f_name	hire_date
Ralph	4/5/1998
Priyan	3/5/2022
Sengathir	3/8/2006

11)

Employee	Emp Hired	Manager	Mgr. Hired
Ralph	4/5/1998	Prem	9/6/2006
Ben	5/5/1994	Harish	8/7/2023

1)

L_name	Dept_name	Dept_no
Patel	Stock	545
Raj	Sales	448
Kumar	Sales	543

2)

Job_code	Dept_location
125	China

3) P_name Dept_name Dept_location city

raj sales 65 china

4) last_Name Dept_name

Patel Stock

5) last_Name Job_name Dept_no

~~Bartlett~~ Dexter Sales_representative 342

6) Employee Emp No Manager Manager_id

Raj 80 Mohan 238

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	4
Total (15)	14
Faculty Signature	

RESULT:

Thus the data is displayed from multiple tables using SQL Query.