Date: 24/9/25

EXERCISE-10

USING THE SET OPERATORS

AIM: To use the set operators in displaying specific data.

Objectives

After the completion this exercise, the students should be able to do the following:

- Describe set operators
- Use a set operator to combine multiple queries into a single query
- Control the order of rows returned

The set operators combine the results of two or more component queries into one result.

Queries containing set operators are called *compound queries*.

Operator	Returns
UNION	All distinct rows selected by either query
UNION ALL	All rows selected by either query, including all duplicates
INTERSECT	All distinct rows selected by both queries
MINUS	All distinct rows that are selected by the first SELECT statement and not selected in the second SELECT statement

The tables used in this lesson are:

- EMPLOYEES: Provides details regarding all current employees

- JOB_HISTORY: Records the details of the start date and end date of the former job, and the job identification number and department when an employee switches jobs

UNION Operator

Guidelines

- The number of columns and the data types of the columns being selected must be identical in all the SELECT statements used in the query. The names of the columns need not be identical.

- UNION operates over all of the columns being selected.

- NULL values are not ignored during duplicate checking.

- The IN operator has a higher precedence than the UNION operator.

- By default, the output is sorted in ascending order of the first column of the SELECT clause.

Example:

Display the current and previous job details of all employees. Display each employee only once.

```
SELECT employee_id, job_id FROM employees UNION SELECT employee_id, job_id
FROM job_history;
```

• The number of columns and the data types of the columns being selected by the SELECT statements in the queries must be identical in all the SELECT statements used in the query. The names of the columns need not be identical.
• All of the columns in the WHERE clause must be in the SELECT clause for the MINUS operator to work.

Example:

Display the employee IDs of those employees who have not changed their jobs even once.

```
SELECT employee_id,job_id
FROM employees
MINUS
SELECT employee_id,job_id
FROM job_history;
```

Find the Solution for the following:

1. The HR department needs a list of department IDs for departments that do not contain the job ID ST_CLERK. Use set operators to create this report.

```
Select dept-id from departments
MINUS
Select dept-id from employees where job-id = 'ST_CLERK';
```

2. The HR department needs a list of countries that have no departments located in them. Display the country ID and the name of the countries. Use set operators to create this report.

```
Select country-id , country-name FROM countries
MINUS
select country-id , country-name FROM locations;
```

3. Produce a list of jobs for departments 10, 50, and 20, in that order. Display job ID and department ID using set operators.

```
Select job-id , dept-id FROM employees where dept-id
IN(10,50,20) ORDER BY CASE dept-id
WHEN 10 THEN1
WHEN 50 THEN2 WHEN 20 THEN3 ELSEx
END;
```

4. Create a report that lists the employee IDs and job IDs of those employees who currently have a job title that is the same as their job title when they were initially hired by the company (that is, they changed jobs but have now gone back to doing their original job).

```
select emp-id , job-id FROM employees
INTERSECT
Select emp-id , job-id FROM job-history;
```

5. The HR department needs a report with the following specifications:

- Last name and department ID of all the employees from the EMPLOYEES table, regardless of whether or not they belong to a department.

- Department ID and department name of all the departments from the DEPARTMENTS table, regardless of whether or not they have employees working in them Write a compound query to accomplish this.

```
select l_name, TO_CHAR(dept_id) FROM employees
UNION
select dept_name, TO_CHAR(dept_id) FROM departments;
```

Evaluation Procedure	Marks awarded
Query(5)	5
Execution (5)	5
Viva(5)	3
Total (15)	13
Faculty Signature	

RESULT:

Thus the specific data is displayed using set operators.

1)

- Departments Table:

dept_id	departnat_name
10	Admin
20	HR
50	Sales

- Employees Table:

emp_id	job_id	dept_id
100	SA_REP	50
101	ST_CLERK	20

sample output:

dept_id
10
50

2) Sample Input:

- Countries Table:

country_id	country_name
US	United States
UK	United Kingdom
IN	India

- Departments Table:

dept_id	dept_name	country_id
10	Admin	US
20	HR	IN

Sample output:

country_id	country_name
UK	United Kingdom

3) Sample Input:
· Employees Table

emp_id	job_id	dept_id
100	SA_REP	50
101	ST_CLERK	20
102	IT_PROG	10

Sample output:

job_id	dept_id
IT_PROG	10
ST_CLERK	20
SA_REP	50

Practice Exercise

NULL Functions

1. Create a report that shows the Global Fast Foods promotional name, start date, and end date from the f_promotional_menus table. If there is an end date, temporarily replace it with "end in two weeks". If there is no end date, replace it with today's date.

```
Select promotal_name, start_date, NVL (end_date, SYSDATE+14)
AS end_date_extended FROM f_promotional_menus;
```

2. Not all Global Fast Foods staff members receive overtime pay. Instead of displaying a null value for these employees, replace null with zero. Include the employee's last name and overtime rate in the output. Label the overtime rate as "Overtime Status".

```
Select l_name, NVL (overtime_rate, 0) AS "Overtime
Status" FROM global_fast_foods_employees;
```

3. The manager of Global Fast Foods has decided to give all staff who currently do not earn overtime an overtime rate of $5.00. Construct a query that displays the last names and the overtime rate for each staff member, substituting $5.00 for each null overtime value.

```
Select l_name, NVL (overtime_rate, 5.00) AS overtime_rate
FROM global_fast_foods_staff;
```

4. Not all Global Fast Foods staff members have a manager. Create a query that displays the employee last name and 9999 in the manager ID column for these employees.

```
Select l_name, NVL (manager_id, 9999) AS manager_id
FROM global_fast_foods_staff_members;
```

5. Which statement(s) below will return null if the value of v_sal is 50?

a. SELECT nvl(v_sal, 50) FROM emp;
b. SELECT nvl2(v_sal, 50) FROM emp;
c. SELECT nullif(v_sal, 50) FROM emp;
d. SELECT coalesce (v_sal, Null, 50) FROM emp;

6. What does this query on the Global Fast Foods table return?

```
SELECT COALESCE(last_name, to_char(manager_id)) as NAME
FROM f_staffs;
```

It will return the first non-null value from the provided list. It will display the last_name if it is not null. If the last_name is null, it will display the manager_id converted to a character string.

7a. Create a report listing the first and last names and month of hire for all employees in the EMPLOYEES table (use TO_CHAR to convert hire_date to display the month).

```
Select f_name, l_name, TO_CHAR (hire_date, 'month')
AS "Month of Hire" FROM employees;
```

b. Modify the report to display null if the month of hire is September. Use the NULLIF function.

```
Select f_name, l_name, NVLIF (TO_CHAR (hire_date, 'month'),
'September') AS "Month of Hire" FROM employees;
```

8. For all null values in the specialty column in the DJs on Demand d_partners table, substitute "No Specialty." Show the first name and specialty columns only.

```
Select f_name, l_name, NVL (Specialty, 'NO Specialty')
AS Specialty FROM d_partners;
```

Conditional Expressions

1. From the DJs on Demand d_songs table, create a query that replaces the 2-minute songs with "shortest" and the 10-minute songs with "longest". Label the output column "Play Times".

Select title, CASE duration WHEN 2 THEN 'shortest' WHEN 10 THEN 'longest' ELSE TO-CHAR (duration) END AS "Play Times" FROM d_songs;

2. Use the Oracle database employees table and CASE expression to decode the department id. Display the department id, last name, salary and a column called "New Salary" whose value is based on the following conditions:
If the department id is 10 then 1.25 * salary If
the department id is 90 then 1.5 * salary
If the department id is 130 then 1.75 * salary
Otherwise, display the old salary.

Select dept_id, l_name, salary, case dept_id WHEN 10 then 1.25 * salary when 90 then 1.5 * salary when 130 then 1.75 * salary ELSE salary END AS "New Salary" FROM employees;

3. Display the first name, last name, manager ID, and commission percentage of all employees in departments 80 and 90. In a 5th column called "Review", again display the manager ID. If they don't have a manager, display the commission percentage. If they don't have a commission, display 99999.

Select f_name, l_name, NVL (manager_id, 99999) AS manager_id, NVL (commission_pct, 9999) AS commission_pct, CASE when manager_id is NULL Then TO-CHAR (commission_pct) ELSE TO-CHAR (manager_id) END AS "Review" FROM employees WHERE dept_id IN (80, 90);

Cross Joins and Natural Joins

Use the Oracle database for problems 1-4.

1. Create a cross-join that displays the last name and department name from the employees and departments tables.

```
Select l_name, dept_name FROM employees
CROSS JOIN departments;
```

2. Create a query that uses a natural join to join the departments table and the locations table. Display the department id, department name, location id, and city.

```
Select dept_id, dept_name, location_id, city
FROM departments
NATURAL JOIN locations;
```

3. Create a query that uses a natural join to join the departments table and the locations table. Restrict the output to only department IDs of 20 and 50. Display the department id, department name, location id, and city.

```
Select dept_id, dept_name, location_id, city FROM
departments NATURAL JOIN locations WHERE dept_id
IN (20,50);
```

Evaluation Procedure	Marks awarded
Practice Evaluation (5)	5
Viva(5)	4
Total (10)	9
Faculty Signature	