

- 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 dept where dept_id IS NOT NULL MINUS
select dept_id from emp 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 from locations JOIN dept ON location_id = dept.location_id;
```

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 = 10 UNION ALL
select job_id, dept_id from employees where dept_id = 50 UNION ALL
select job_id, dept_id from employees where dept_id = 20;
```

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 employee_id, job_id from employees INTERSECT select
employee_id, job_id from job_history jh where jh.start_date =
(select MIN(start_date) from job_history where employee_id =
SELECT jh.employee_id;
```

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

select l-name, dept-id, null as dept-name from employees union all
select null as l-name, dept-id, dept-name from departments order by
dept-id, l-name;

| Evaluation Procedure | Marks awarded |
|----------------------|---------------|
| Query(5) | 5 |
| Execution (5) | 5 |
| Viva(5) | 5 |
| Total (15) | 15 |
| Faculty Signature | B.M |

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 promotional-name, start-date, NVL(TO-CHAR(end-date, 'end in two')), TO-CHAR(SYSDATE, 'DD-MON-XXXX') AS end-date-status 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 last-name, NVL(overtime-rate, 0) AS "overtime status" FROM f_staffs;

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 last-name, NVL(overtime-rate, 5.00) AS overtime-rate from f_staffs;

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 last-name, NVL(manager-id, 9999) AS manager-id from f_staffs;

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;

select coalesce(last-name, to-char(manager-id))
AS NAME from f_staffs;

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 first_name, last_name, TO_CHAR(hire_date, 'MONTH') AS hire_month FROM employees;

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

select first_name, last_name, NULLIF(TO_CHAR(hire_date, 'MONTH'), 'SEPTEMBER') AS hire_month 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 first_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 song-name, case when play-time = 2 then
'shortest' when play-time = 10 then 'longest' else to
char(play-time) 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 $1.25 * \text{salary}$
If the department id is 90 then $1.5 * \text{salary}$

If the department id is 130 then $1.75 * \text{salary}$
Otherwise, display the old salary.

select dept-id, l-name, salary, case when
dept-id = 10 then salary + 1.25 when dept-id = 90 then
salary + 1.5 when dept-id = 130 then salary + 1.75
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, manager-id, commission-pct
case when manager-id is not null then
manager-id when commission-pct else 99999
end as "Review" from employees where
department-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 e.last_name, d.department_name from emp e cross
join dept d;

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
dept 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) | |
| Viva(5) | |
| Total (10) | |
| Faculty Signature | |