EXERCISE-9

Sub queries

Find the Solution for the following:

1. The HR department needs a query that prompts the user for an employee last name. The query then displays the last name and hire date of any employee in the same department as the employee whose name they supply (excluding that employee). For example, if the user enters Zlotkey, find all employees who work with Zlotkey (excluding Zlotkey).

SELECT e.last_name, e.hire_date FROM employees e JOIN employees target ON e.department_id = target.department_id WHERE target.last_name = :employee_last_name AND e.employee_id != target.employee_id;

2. Create a report that displays the employee number, last name, and salary of all employees who earn more than the average salary. Sort the results in order of ascending salary.

SELECT employee_id, last_name, salary FROM employees WHERE salary > (SELECT AVG(salary) FROM employees) ORDER BY salary ASC;

3. Write a query that displays the employee number and last name of all employees who work in a department with any employee whose last name contains a *u*.

SELECT employee_id, last_name FROM employees WHERE department_id IN (SELECT department_id FROM employees WHERE last_name LIKE '%u%');

4. The HR department needs a report that displays the last name, department number, and job ID of all employees whose department location ID is 1700.

SELECT last_name, department_id, job_id FROM employees WHERE department_id IN (SELECT department_id FROM departments WHERE location_id = 1700);

5. Create a report for HR that displays the last name and salary of every employee who reports to King.

SELECT e.last_name, e.salary FROM employees e JOIN employees mgr ON e.manager_id = mgr.employee_id WHERE mgr.last_name = 'King';

6. Create a report for HR that displays the department number, last name, and job ID for every employee in the Executive department.

SELECT department_id, last_name, job_id FROM employees WHERE department_id = (SELECT department_id FROM departments WHERE department_name = 'Executive');

7. Modify the query 3 to display the employee number, last name, and salary of all employees who earn more than the average salary and who work in a department with any employee whose last name contains a *u*.

SELECT employee_id, last_name, salary FROM employees WHERE salary > (SELECT AVG(salary) FROM employees) AND department_id IN (SELECT department_id FROM employees WHERE last_name LIKE '%u%');

Practice Questions

1. Ellen Abel is an employee who has received a $2,000 raise. Display her first name and last name,her current salary, and her new salary. Display both salaries with a $ and two decimal places. Label her new salary column AS New Salary.

SELECT first_name, last_name, TO_CHAR(salary, '$9999.99') AS current_salary, TO_CHAR(salary + 2000, '$9999.99') AS "New Salary" FROM employees WHERE first_name = 'Ellen' AND last_name = 'Abel';

2. On what day of the week and date did Global Fast Foods' promotional code 110 Valentine's Special begin?

SELECT TO_CHAR(start_date, 'Day, DD-Mon-YYYY') AS start_day_date FROM promotions WHERE promo_code = 110;

3. Create one query that will convert 25-Dec-2004 into each of the following (you will have to convert 25-Dec-2004 to a date and then to character data):

December 25th, 2004

DECEMBER 25TH, 2004

25th december, 2004

SELECT TO_CHAR(TO_DATE('25-Dec-2004', 'DD-Mon-YYYY'), 'Month DDth, YYYY') AS "December 25th, 2004",TO_CHAR(TO_DATE('25-Dec-2004', 'DD-Mon-YYYY'), 'MONTH DDTH, YYYY') AS "DECEMBER 25TH, 2004",TO_CHAR(TO_DATE('25-Dec-2004', 'DD-Mon-YYYY'), 'DDth month, YYYY') AS "25th december, 2004"FROM dual;

4. Create a query that will format the DJs on Demand d_packages columns, low-range and high-range package costs, in the format $2500.00.

SELECT TO_CHAR(low_range, '$9999.99') AS low_range, TO_CHAR(high_range, '$9999.99') AS high_range FROM d_packages;

5. Convert JUNE192004 to a date using the fx format model.

SELECT TO_DATE('JUNE192004', 'fxMONTHDDYYYY') AS formatted_date FROM dual;

6. What is the distinction between implicit and explicit datatype conversion? Give an example of each.

Implicit Datatype Conversion:

Happens automatically by the database when needed.

Explicit Datatype Conversion:

Requires a function to explicitly convert the datatype.

7. Why is it important from a business perspective to have datatype conversions?

Datatype conversions are crucial for ensuring that data is accurately interpreted and processed in various contexts. Proper datatype conversion allows for:

Accurate Calculations: Ensuring numbers are treated as numbers for arithmetic operations.

Correct Comparisons: Enabling valid comparisons between data of different types.

Data Integrity: Preventing data corruption by enforcing the correct datatype.

System Interoperability: Facilitating data exchange between systems with different datatype requirements.

These factors support business operations by maintaining data accuracy, reliability, and consistency across applications and systems.

EXERCISE-10

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 department_id FROM departments MINUS

SELECT department_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 c.country_id, c.country_name

FROM countries c JOIN locations l ON c.country_id = l.country_id JOIN departments d ON

l.location_id = d.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, department_id FROM employees WHERE department_id = 10) UNION

(SELECT job_id, department_id FROM employees WHERE department_id = 50)UNION

(SELECT job_id, department_id FROM employees WHERE department_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 job_historyWHERE job_id = (SELECT job_id FROM

employees WHERE job_history.employee_id = employees.employee_id) AND start_date = (SELECT

MIN(start_date) FROM job_history WHERE job_history.employee_id = employees.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 accomplish this.

SELECT e.last_name, e.department_id FROM employees e UNION SELECT d.department_id,

d.department_name FROM departments d;

Practice Exercise

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 weeks'), TO_CHAR(SYSDATE, 'DD-Mon-YYYY')) AS end_date 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 Global_Fast_Foods_Staff;

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 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 last_name, NVL(manager_id, 9999) as "Manager ID"
FROM Global_Fast_Foods_Staff;

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;

Statement c will return null if v_sal is 50. It uses the NULLIF function.

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;

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 month_of_hire
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 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.

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
  CASE
    WHEN duration = 2 THEN 'Shortest'
    WHEN duration = 10 THEN 'Longest'
    ELSE duration
  END as "Play Times"
FROM d_songs;
```

2. Use the Oracle database employees table and CASE expression to decode the department id.
on Display the department id, last name, salary and a column called “New Salary” whose value is based 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 department_id, last_name, salary,
  CASE
    WHEN department_id = 10 THEN 1.25 * salary
    WHEN department_id = 90 THEN 1.5 * salary
    WHEN department_id = 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 first_name, last_name, manager_id, commission_pct,
  CASE
```

```
    WHEN manager_id IS NULL THEN commission_pct
    WHEN commission_pct IS NULL THEN 99999
    ELSE manager_id
  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 employees e
CROSS JOIN departments 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 d.department_id, d.department_name, l.location_id, l.city
FROM departments d
NATURAL JOIN locations l;
```

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 d.department_id, d.department_name, l.location_id, l.city FROM departments d
NATURAL JOIN locations l
WHERE d.department_id IN (20, 50);
```