



Práctica #9. Práctica sobre estructuras de control

Instrucciones: Cree las siguientes consultas. Adjunte, para cada ejercicio, la captura de pantalla del resultado.

1. Por medio de un bloque anónimo, presentar el nombre, salario y la comisión del empleado cuyo número de empleado es 150. Guardar el bloque PL/SQL como *lab_02_01.sql*. Salida de la información:

Num_emp	Nombre	Salario	Comisión
150	Juan	\$\$\$\$\$	\$. \$\$

```
DECLARE
    v_num number;
    v_name VARCHAR2(20);
    v_salary number;
    v_comision number;
BEGIN
    SELECT
        EMPLOYEE_ID, FIRST_NAME, SALARY, commission_pct
    INTO
        v_num, v_name, v_salary, v_comision
    FROM
        hr.EMPLOYEES
    WHERE employee_id = 150;

    DBMS_OUTPUT.PUT_LINE('Num_emp' || v_num || 'Nombre' || v_name || 'Salario' || v_salary || 'Comisión' || v_comision);
END;
```

```
[2026-02-04 11:59:29] completed in 11 ms
Num_emp      Nombre      Salario      Comisión
150          Peter       10000       .3
```

2. Cargar el script *lab_02_01.sql*, agregar los datos del empleado 147 y realizar las comparaciones entre los empleados 150 y 147 en su comisión y salario. Guardar el bloque anónimo como *lab_02_02.sql*. Presentar la siguiente salida de información.

Num_emp	Nombre	Sueldo Neto.
150	Juan	salario + (salario * comisión)
147	Pedro	salario + (salario * comisión)

Entonces.

Juan gana más que Pedro.

ó

Pedro gana más que Juan.

```

DECLARE
    v_num_1 number;
    v_name_1 VARCHAR2(20);
    v_salary_1 number;
    v_comision_1 number;
    v_neto_1 number;
    v_num_2 number;
    v_name_2 VARCHAR2(20);
    v_salary_2 number;
    v_comision_2 number;
    v_neto_2 number;
BEGIN
    SELECT
        EMPLOYEE_ID, FIRST_NAME, SALARY, commission_pct
    INTO
        v_num_1, v_name_1, v_salary_1, v_comision_1
    FROM
        hr.EMPLOYEES
    WHERE employee_id = 150;

    SELECT
        EMPLOYEE_ID, FIRST_NAME, SALARY, commission_pct
    INTO
        v_num_2, v_name_2, v_salary_2, v_comision_2
    FROM
        hr.EMPLOYEES
    WHERE employee_id = 147;

    v_neto_1 := (v_salary_1 + (v_salary_1 * v_comision_1));
    v_neto_2 := (v_salary_2 + (v_salary_2 * v_comision_2));

    DBMS_OUTPUT.PUT_LINE('Num_emp      Nombre      Sueldo Neto');
    DBMS_OUTPUT.PUT_LINE(v_num_1 || '      ' || v_name_1 || '      ' || v_neto_1);
    DBMS_OUTPUT.PUT_LINE(v_num_2 || '      ' || v_name_2 || '      ' || v_neto_2);

    IF v_neto_1 > v_neto_2 THEN
        DBMS_OUTPUT.PUT_LINE(v_name_1 || ' gana mas que ' || v_name_2);
    ELSE
        DBMS_OUTPUT.PUT_LINE(v_name_2 || ' gana mas que ' || v_name_1);
    END IF;
END;

```

```
[2026-02-04 11:56:33] completed in 17 ms
Num_emp      Nombre      Sueldo Neto
150          Peter        13000
147          Alberto       15600
Alberto gana mas que Peter
```

3. Cargar el script *lab_02_02.sql*, agregar los datos del empleado 158. Por medio de una estructura IF-THEN-ELSIF, hacer las comparaciones necesarias en el salario neto para cada uno de los empleados para indicar quién de ellos tiene el salario neto más alto. Guardar el bloque PL/SQL como *lab_02_03.sql*

```
DECLARE
    v_num_1 number;
    v_name_1 VARCHAR2(20);
    v_salary_1 number;
    v_comision_1 number;
    v_neto_1 number;
    v_num_2 number;
    v_name_2 VARCHAR2(20);
    v_salary_2 number;
    v_comision_2 number;
    v_neto_2 number;
    v_num_3 number;
    v_name_3 VARCHAR2(20);
    v_salary_3 number;
    v_comision_3 number;
    v_neto_3 number;
BEGIN
    SELECT
        EMPLOYEE_ID, FIRST_NAME, SALARY, commission_pct
    INTO
        v_num_1, v_name_1, v_salary_1, v_comision_1
    FROM
        hr.EMPLOYEES
    WHERE employee_id = 150;

    SELECT
        EMPLOYEE_ID, FIRST_NAME, SALARY, commission_pct
    INTO
        v_num_2, v_name_2, v_salary_2, v_comision_2
    FROM
        hr.EMPLOYEES
    WHERE employee_id = 147;

    SELECT
        EMPLOYEE_ID, FIRST_NAME, SALARY, commission_pct
    INTO
        v_num_3, v_name_3, v_salary_3, v_comision_3
```

```

FROM
    hr.EMPLOYEES
WHERE employee_id = 158;

v_neto_1 := (v_salary_1 + (v_salary_1 * v_comision_1));
v_neto_2 := (v_salary_2 + (v_salary_2 * v_comision_2));
v_neto_3 := (v_salary_3 + (v_salary_3 * v_comision_3));

DBMS_OUTPUT.PUT_LINE('Num_emp      Nombre      Sueldo Neto');
DBMS_OUTPUT.PUT_LINE(v_num_1 || ' ' || v_name_1 || ' ');
v_neto_1);
DBMS_OUTPUT.PUT_LINE(v_num_2 || ' ' || v_name_2 || ' ');
v_neto_2);
DBMS_OUTPUT.PUT_LINE(v_num_3 || ' ' || v_name_3 || ' ');
v_neto_3);

IF v_neto_1 > v_neto_2 and v_neto_1 > v_neto_3 THEN
    DBMS_OUTPUT.PUT_LINE(v_name_1 || ' gana mas que ' || v_name_2 || ' y
|| v_name_3);
ELSIF v_neto_2 > v_neto_1 and v_neto_2 > v_neto_3 THEN
    DBMS_OUTPUT.PUT_LINE(v_name_2 || ' gana mas que ' || v_name_1 || ' y
|| v_name_3);
ELSE
    DBMS_OUTPUT.PUT_LINE(v_name_3 || ' gana mas que ' || v_name_1 || ' y
|| v_name_2);
END IF;
END;

```

[2026-02-04 12:11:45] completed in 14 ms

Num_emp	Nombre	Sueldo Neto
150	Peter	13000
147	Alberto	15600
158	Allan	12150

Alberto gana mas que Peter y Allan

4. Por medio de una estructura LOOP, consultar los siguientes datos: número de empleado, nombre, fecha de contratación y nombre del departamento de todos los empleados cuyo número de empleado están entre 150 y 200. Guardar el script como *lab_02_04.sql*.

```

DECLARE
    v_num NUMBER := 150;
    v_name VARCHAR2(20);
    v_date DATE;
    v_dept VARCHAR2(20);

```

```

BEGIN
    LOOP
        EXIT WHEN v_num > 177;

        SELECT e.employee_id, e.first_name, e.hire_date,
d.department_name
        INTO v_num, v_name, v_date, v_dept
        FROM hr.employees e
        JOIN hr.departments d ON e.department_id = d.department_id
        WHERE e.employee_id = v_num;

        DBMS_OUTPUT.PUT_LINE(v_num || ' ' || v_name || ' ' || v_date ||
' ' || v_dept);

        v_num := v_num + 1;
    END LOOP;
END;

```

END,

[2026-02-04 12:38:10] completed in 33 ms

150 Peter 30/01/05 Sales
151 David 24/03/05 Sales
152 Peter 20/08/05 Sales
153 Christopher 30/03/06 Sales
154 Nanette 09/12/06 Sales
155 Oliver 23/11/07 Sales
156 Janette 30/01/04 Sales
157 Patrick 04/03/04 Sales
158 Allan 01/08/04 Sales
159 Lindsey 10/03/05 Sales
160 Louise 15/12/05 Sales
161 Sarah 03/11/06 Sales
162 Clara 11/11/05 Sales
163 Danielle 19/03/07 Sales
164 Mattea 24/01/08 Sales
165 David 23/02/08 Sales
166 Sundar 24/03/08 Sales
167 Amit 21/04/08 Sales
168 Lisa 11/03/05 Sales
169 Harrison 23/03/06 Sales
170 Tayler 24/01/06 Sales
171 William 23/02/07 Sales
172 Elizabeth 24/03/07 Sales
173 Sundita 21/04/08 Sales
174 Ellen 11/05/04 Sales

5. Cargar el script *lab_02_04.sql*, y cambiar la estructura por un FOR LOOP. Guardar el resultado como *lab_02_05.sql*

```
DECLARE
    v_iteracion NUMBER;
    v_num NUMBER;
    v_name VARCHAR2(20);
    v_date DATE;
    v_dept VARCHAR2(20);
BEGIN
    FOR v_iteracion IN 150..177 LOOP
        SELECT EMPLOYEE_ID, FIRST_NAME, HIRE_DATE, DEPARTMENT_NAME
        INTO v_num, v_name, v_date, v_dept
        FROM hr.EMPLOYEES, hr.DEPARTMENTS
        WHERE EMPLOYEE_ID = v_iteracion
        AND hr.EMPLOYEES.DEPARTMENT_ID = hr.DEPARTMENTS.DEPARTMENT_ID;
        DBMS_OUTPUT.PUT_LINE(v_num || ' '|| v_name || ' '|| v_date || ' '
        || v_dept);
    END LOOP;
END;

```

[2026-02-04 12:21:16] completed in 11 ms

150 Peter 30/01/05 Sales
151 David 24/03/05 Sales
152 Peter 20/08/05 Sales
153 Christopher 30/03/06 Sales
154 Nanette 09/12/06 Sales
155 Oliver 23/11/07 Sales
156 Janette 30/01/04 Sales
157 Patrick 04/03/04 Sales
158 Allan 01/08/04 Sales
159 Lindsey 10/03/05 Sales
160 Louise 15/12/05 Sales
161 Sarath 03/11/06 Sales
162 Clara 11/11/05 Sales
163 Danielle 19/03/07 Sales
164 Mattea 24/01/08 Sales
165 David 23/02/08 Sales
166 Sundar 24/03/08 Sales
167 Amit 21/04/08 Sales
168 Lisa 11/03/05 Sales
169 Harrison 23/03/06 Sales
170 Tayler 24/01/06 Sales
171 William 23/02/07 Sales
172 Elizabeth 24/03/07 Sales
173 Sundita 21/04/08 Sales
174 Ellen 11/05/04 Sales
175 Alyssa 19/03/05 Sales
176 Jonathon 24/03/06 Sales
177 Jack 23/04/06 Sales

6. Cargar el script *lab_02_05.sql*, y cambiar la estructura por un WHILE LOOP. Guardar como *lab_02_06.sql*

```
SET SERVEROUTPUT ON;
DECLARE
    v_iteracion NUMBER := 150;
    v_num NUMBER;
    v_name VARCHAR2(20);
    v_date DATE;
    v_dept VARCHAR2(20);
```

```

BEGIN
    WHILE v_iteracion < 178 LOOP
        SELECT EMPLOYEE_ID, FIRST_NAME, HIRE_DATE, DEPARTMENT_NAME
        INTO v_num, v_name, v_date, v_dept
        FROM hr.EMPLOYEES, hr.DEPARTMENTS
        WHERE EMPLOYEE_ID = v_iteracion
        AND hr.EMPLOYEES.DEPARTMENT_ID = hr.DEPARTMENTS.DEPARTMENT_ID;
        DBMS_OUTPUT.PUT_LINE(v_num || ' '|| v_name || ' ' || v_date || ' '
        || v_dept);
        v_iteracion := v_iteracion + 1;
    END LOOP;
END;

```

[2026-02-04 12:29:25] completed in 15 ms

150 Peter 30/01/05 Sales
151 David 24/03/05 Sales
152 Peter 20/08/05 Sales
153 Christopher 30/03/06 Sales
154 Nanette 09/12/06 Sales
155 Oliver 23/11/07 Sales
156 Janette 30/01/04 Sales
157 Patrick 04/03/04 Sales
158 Allan 01/08/04 Sales
159 Lindsey 10/03/05 Sales
160 Louise 15/12/05 Sales
161 Sarath 03/11/06 Sales
162 Clara 11/11/05 Sales
163 Danielle 19/03/07 Sales
164 Mattea 24/01/08 Sales
165 David 23/02/08 Sales
166 Sundar 24/03/08 Sales
167 Amit 21/04/08 Sales
168 Lisa 11/03/05 Sales
169 Harrison 23/03/06 Sales
170 Tayler 24/01/06 Sales
171 William 23/02/07 Sales
172 Elizabeth 24/03/07 Sales
173 Sundita 21/04/08 Sales
174 Ellen 11/05/04 Sales
175 Alyssa 19/03/05 Sales
176 Jonathon 24/03/06 Sales
177 Jack 23/04/06 Sales

7. Por medio de un bloque anónimo, obtener la siguiente información del departamento de ventas: nombre del empleado y nombre del jefe inmediato. Usar la estructura de control CASE para validar al jefe inmediato y obtener su nombre en base al siguiente tabulador:

MANAGER_ID	NOMBRE DEL JEFE
100	Steven
145	John
146	Karen
147	Alberto

Guardar el bloque como *lab_02_07.sql*. Obtener la siguiente salida:

Juan trabaja para Steven.

```
DECLARE
    v_nom_emp VARCHAR2(50);
    v_manager_id NUMBER;
    v_nom_jefe VARCHAR2(50);
BEGIN
    FOR num IN 100..200 LOOP
        SELECT first_name, manager_id
        INTO v_nom_emp, v_manager_id
        FROM employees
        WHERE employee_id = num;
        CASE v_manager_id
            WHEN 100 THEN v_nom_jefe := 'Steven';
            WHEN 145 THEN v_nom_jefe := 'John';
            WHEN 146 THEN v_nom_jefe := 'Karen';
            WHEN 147 THEN v_nom_jefe := 'Alberto';
            ELSE v_nom_jefe := 'Desconocido';
        END CASE;
        IF v_nom_jefe != 'Desconocido' THEN
            DBMS_OUTPUT.PUT_LINE(v_nom_emp || ' trabaja para ' ||
v_nom_jefe);
        END IF;
    END LOOP;
END;
```

[2026-02-04 12:35:28] completed in 13 ms

```
Neena trabaja para Steven
Lex trabaja para Steven
Den trabaja para Steven
Matthew trabaja para Steven
Adam trabaja para Steven
Payam trabaja para Steven
Shanta trabaja para Steven
Kevin trabaja para Steven
John trabaja para Steven
Karen trabaja para Steven
Alberto trabaja para Steven
Gerald trabaja para Steven
Eleni trabaja para Steven
Peter trabaja para John
David trabaja para John
Peter trabaja para John
Christopher trabaja para John
Nanette trabaja para John
Oliver trabaja para John
Janette trabaja para Karen
Patrick trabaja para Karen
Allan trabaja para Karen
Lindsey trabaja para Karen
Louise trabaja para Karen
Sarah trabaja para Karen
Clara trabaja para Alberto
Danielle trabaja para Alberto
Mattea trabaja para Alberto
David trabaja para Alberto
Sundar trabaja para Alberto
Amit trabaja para Alberto
```