1. Reescriba la siguiente instrucción IF analice el uso de CASE si es posible

IF calc\_totals (1994, company\_id\_in => 1005) AND

NOT overdue\_balance (company\_id\_in => 1005)

THEN

display\_sales\_figures (1005);

ELSE

contact\_vendor;

END IF;

**Respuesta:**

Es posible aplicar la cláusula Case, pero sería innecesario ya que solo se cuenta con dos casos posibles de solución en esta ocasión un **if** bastara.

1. **Reescriba la siguiente instrucción IF analice el uso de CASE si es posible**

IF total\_sales <= 0

THEN

no\_revenue := TRUE;

ELSE

no\_revenue := FALSE;

END IF;

IF total\_sales <= 0

THEN

no\_revenue := TRUE;

ELSIF total\_sales > 0

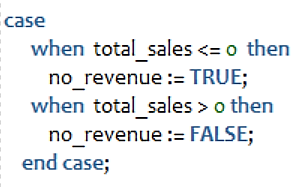
THEN

no\_revenue := FALSE;

END IF;

**Respuesta:**

**Simplificando expresión usando el CASE**



1. **Reescriba la siguiente instrucción IF analice el uso de CASE si es posible**

IF salary < 10000

THEN

bonus := 2000;

ELSE

IF salary < 20000

THEN

bonus := 1500;

ELSE

IF salary < 40000

THEN

bonus := 1000;

ELSE

bonus := 500;

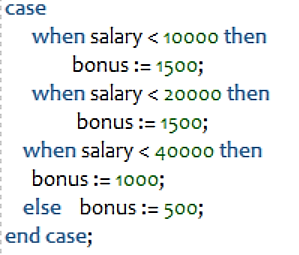
END IF;

END IF;

END IF;

**Respuesta:**

**Simplificando expresión usando el CASE**



1. ¿Identifique qué procedimiento nunca se ejecutará en esta declaración IF?

IF (order\_date > SYSDATE) AND order\_total >= min\_order\_total

THEN

fill\_order (order\_id, 'HIGH PRIORITY');

ELSIF (order\_date < SYSDATE) OR

(order\_date = SYSDATE)

THEN

fill\_order (order\_id, 'LOW PRIORITY');

ELSIF order\_date <= SYSDATE AND order\_total < min\_order\_total

THEN

queue\_order\_for\_addtl\_parts (order\_id);

ELSIF order\_total = 0

THEN

disp\_message (' No items have been placed in this order!');

END IF;

**Respuesta:**

Si pasa el primer filtro quiere decir que order\_date es menor

Al llegar al segundo filtro tiene que ejecutarse el procedimiento ya que es menor por lo tanto La llamada a **queue\_order\_for\_addtl\_parts** nunca se ejecutará.

1. ¿Cuántas veces se ejecuta el siguiente bucle?

FOR year\_index IN REVERSE 12.. 1

LOOP

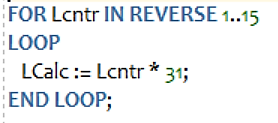
calc\_sales (year\_index);

END LOOP:

**Respuesta:**

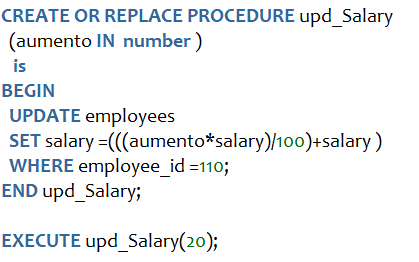
No se realiza ningún ciclo ya que el índice siempre tiene que ser un número menor que el otro.

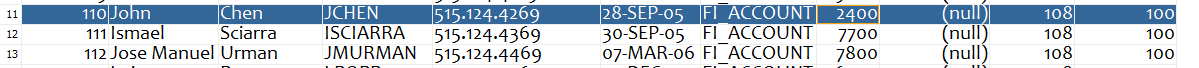
**Ejemplo:**



1. Escriba un bloque PL / SQL para calcular el salario final del empleado cuya ID es 110., dado como parámetro el porcentaje de incentivo a incrementar

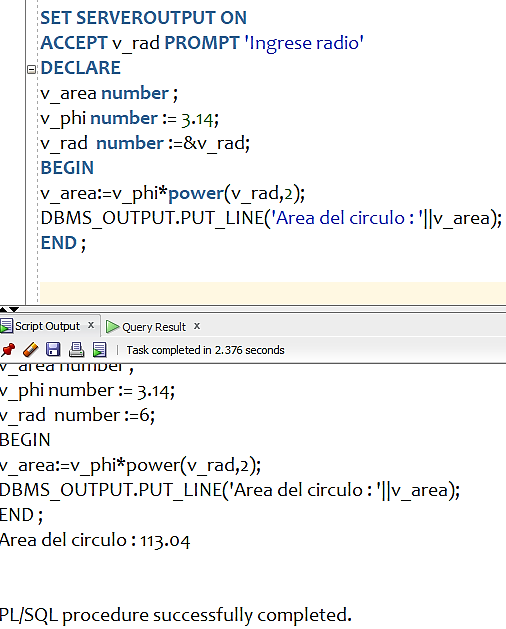
**Respuesta:**

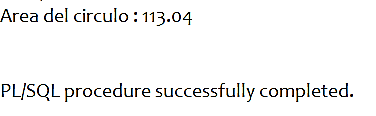




1. Escriba un bloque PL/SQL para calcular el área de un circulo (los valores necesarios para calcular la fórmula del área deben ser solicitados por pantalla)

**Respuesta:**

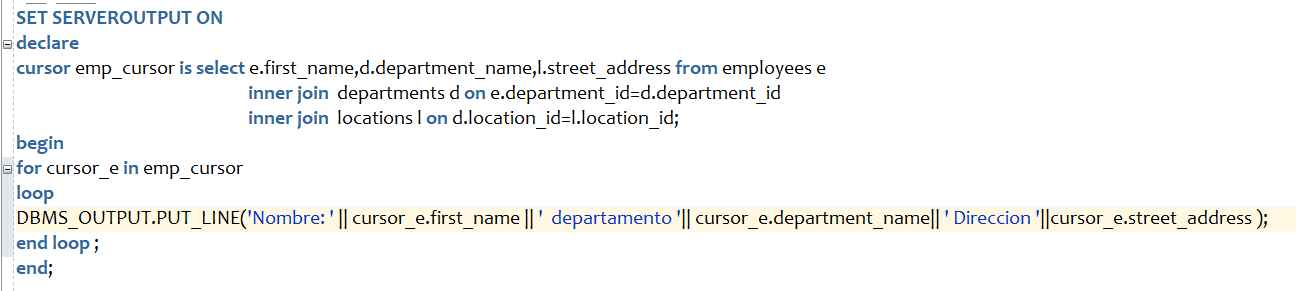




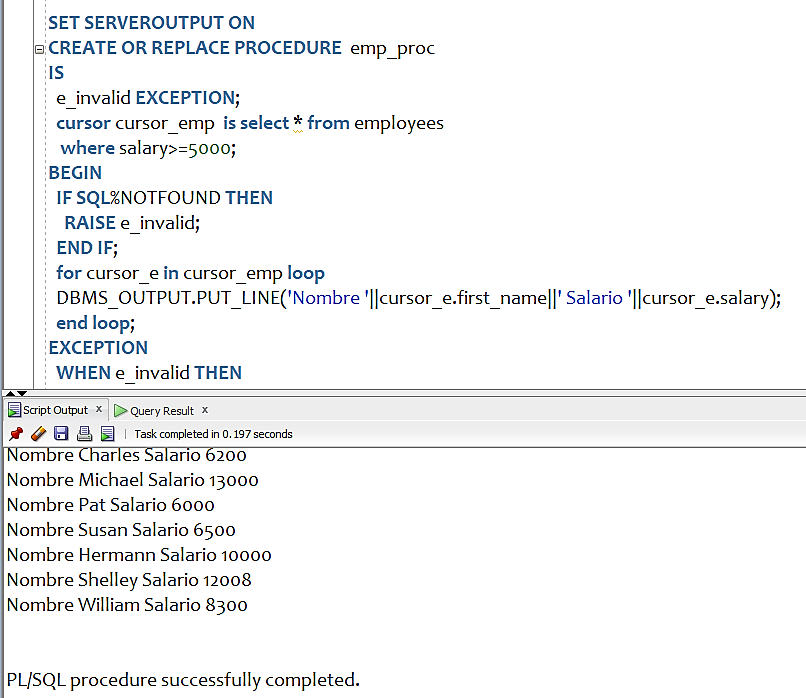
1. Seleccione el tipo de bucle (FOR, WHILE, simple LOOP) apropiado para cumplir con cada uno de los siguientes requisitos:

* Muestra el nombre y la dirección de cada empleado devuelto por un cursor.
* Escanee la lista de empleados en la tabla de PL / SQL, contando todos los salarios superiores a $ 50,000. Sin embargo, ni siquiera inicie la exploración si la tabla está vacía o si hoy es sábado o si el primer empleado de la tabla PL / SQL es el presidente de la empresa.

**Respuesta: a**



**Respuesta b:**



1. Identifique los problemas con los siguientes bucles. ¿Cómo cambiaría el bucle para mejorarlo?

FOR i IN 1 .. 100 LOOP

calc\_totals (i);

IF i > 75

THEN

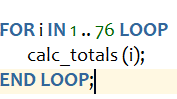
EXIT;

END IF;

END LOOP;

**Respuesta:**

Eliminar la sentencia if del bucle for y realizar el conteo desde 1 a 76.



OPEN emp\_cur;

FETCH emp\_cur INTO emp\_rec;

WHILE emp\_cur%FOUND

LOOP

calc\_totals (emp\_rec.salary);

FETCH emp\_cur INTO emp\_rec;

EXIT WHEN emp\_rec.salary > 100000;

END LOOP;

CLOSE emp\_cur;

**Respuesta:**

FOR a\_counter IN lo\_val .. hi\_val

LOOP

IF a\_counter > lo\_val \* 2

THEN

hi\_val := lo\_val;

END IF;

END LOOP;

**Respuesta:**

Es una mala práctica intentar intercambiar los valores del bucle

DECLARE

CURSOR emp\_cur IS SELECT salary FROM emp;

emp\_rec emp\_cur%ROWTYPE

BEGIN

OPEN emp\_cur;

LOOP

FETCH emp\_cur INTO emp\_rec;

EXIT WHEN emp\_cur%NOTFOUND;

calc\_totals (emp\_rec.salary);

END LOOP;

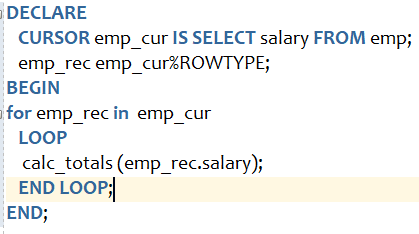
CLOSE emp\_cur;

END;

**Respuesta:**

No compilara ya que falta un; en **emp\_rec emp\_cur%ROWTYPE**

Se puede reducir el código con un simple for loop



WHILE no\_more\_data

LOOP

read\_next\_line (text);

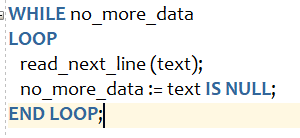
no\_more\_data := text IS NULL;

EXIT WHEN no\_more\_data;

END LOOP;

**Respuesta:**

Es innecesario utilizar EXIT WHEN no\_more\_data; para salir del bucle.



FOR month\_index IN 1 .. 12

LOOP

UPDATE monthly\_sales

SET pct\_of\_sales = 100

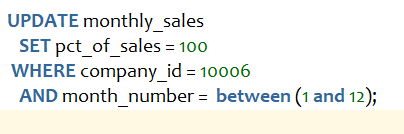
WHERE company\_id = 10006

AND month\_number = month\_index;

END LOOP;

**Respuesta:**

Es innecesario utilizar un foor loop



DECLARE

CURSOR emp\_cur IS SELECT ... ;

BEGIN

FOR emp\_rec IN emp\_cur

LOOP

calc\_totals (emp\_rec.salary);

END LOOP;

IF emp\_rec.salary < 10000

THEN

DBMS\_OUTPUT.PUT\_LINE ('Give ''em a raise!');

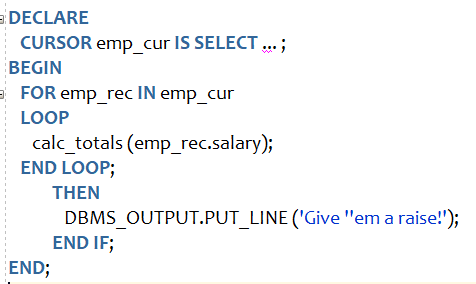
END IF;

CLOSE emp\_cur;

END;

**Respuesta:**

Es puede utilizar emp\_rec.salary luego de que el bucle haya finalizado además no se puede cerrar 2 veces el cursor ya q el for lo cierra automaticamente



DECLARE

CURSOR checked\_out\_cur IS

SELECT pet\_id, name, checkout\_date

FROM occupancy

WHERE checkout\_date IS NOT NULL;

BEGIN

FOR checked\_out\_rec IN checked\_out\_cur

LOOP

INSERT INTO occupancy\_history (pet\_id, name, checkout\_date)

VALUES (checked\_out\_rec.pet\_id,

checked\_out\_rec.name,

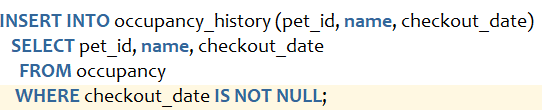
checked\_out\_rec.checkout\_date);

END LOOP;

END;

**Respuesta:**

Mejor usar una sentencia normal;



1. ¿Cuántas veces se ejecuta el siguiente bucle WHILE?

DECLARE

end\_of\_analysis BOOLEAN := FALSE;

CURSOR analysis\_cursor IS SELECT ...;

analysis\_rec analysis\_cursor%ROWTYPE;

next\_analysis\_step NUMBER;

PROCEDURE get\_next\_record (step\_out OUT NUMBER) IS

BEGIN

FETCH analysis\_cursor INTO analysis\_rec;

IF analysis\_rec.status = 'FINAL'

THEN

step\_out := 1;

ELSE

step\_out := 0;

END IF;

END;

BEGIN

OPEN analysis\_cursor;

WHILE NOT end\_of\_analysis

LOOP

get\_next\_record (next\_analysis\_step);

IF analysis\_cursor%NOTFOUND AND

next\_analysis\_step IS NULL

THEN

end\_of\_analysis := TRUE;

ELSE

perform\_analysis;

END IF;

END LOOP;

END;

1. Vuelva a escribir el siguiente bucle para que no utilice un bucle en absoluto

FOR i IN 1 .. 2

LOOP

IF i = 1

THEN

give\_bonus (president\_id, 2000000);

ELSIF i = 2

THEN

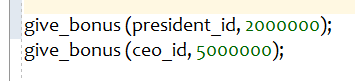
give\_bonus (ceo\_id, 5000000);

END IF;

END LOOP;

**Respuesta:**

Lo mejor llamar 2 veces a la misma funcion;



1. ¿Qué declaración eliminarías de este bloque? ¿Por qué?

DECLARE

CURSOR emp\_cur IS

SELECT ename, deptno, empno

FROM emp

WHERE sal < 2500;

emp\_rec emp\_cur%ROWTYPE;

BEGIN

FOR emp\_rec IN emp\_cur

LOOP

give\_raise (emp\_rec.empno, 10000);

END LOOP;

END;

**Respuesta:**

emp\_rec emp\_cur%ROWTYPE; nunca se utiliza.

1. ¿Qué declaraciones relacionadas con el cursor faltan en el siguiente bloque?

DECLARE

CURSOR emp\_cur IS SELECT \* FROM emp;

BEGIN

OPEN emp\_cur;

FETCH emp\_cur INTO emp\_rec;

END;

**Respuesta:**

EXIT WHEN emp\_cursor%NOTFOUND;

CLOSE CURSOR;

1. ¿Qué estructura debe eliminarse del siguiente bloque?

DECLARE

CURSOR emp\_cur IS SELECT \* FROM emp;

emp\_rec emp\_cur%ROWTYPE;

BEGIN

FOR emp\_rec IN emp\_cur

LOOP

give\_raise (emp\_rec.empno);

END LOOP;

END;

**Respuesta:**

emp\_rec emp\_cur%ROWTYPE;

1. ¿Qué mensaje se muestra en el siguiente bloque si la instrucción SELECT no devuelve una fila?

PROCEDURE display\_dname (emp\_in IN INTEGER) IS

department# dept.deptno%TYPE := NULL;

BEGIN

SELECT deptno INTO department#

FROM emp

WHERE empno = emp\_in;

IF department# IS NULL

THEN

DBMS\_OUTPUT.PUT\_LINE ('Dept is not found!');

ELSE

DBMS\_OUTPUT.PUT\_LINE ('Dept is ' || TO\_CHAR (department#));

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND

THEN

DBMS\_OUTPUT.PUT\_LINE ('No data found');

END;

1. ¿Qué mensaje se muestra en el siguiente bloque si no hay empleados en el departamento 15?

PROCEDURE display\_dept\_count

IS

total\_count INTEGER := 0;

BEGIN

SELECT COUNT(\*) INTO total\_count

FROM emp

WHERE deptno = 15;

IF total\_count = 0

THEN

DBMS\_OUTPUT.PUT\_LINE ('No employees in department!');

ELSE

DBMS\_OUTPUT.PUT\_LINE

('Count of employees in dept 15 = ' || TO\_CHAR (total\_count));

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND

THEN

DBMS\_OUTPUT.PUT\_LINE ('No data found');

END;

1. ¿Cómo cambiaría la instrucción SELECT en los siguientes bloques de cursor para que el bloque pueda mostrar la suma de los salarios en cada departamento?

DECLARE

CURSOR tot\_cur IS

SELECT deptno, SUM (sal)

FROM emp

GROUP BY deptno;

BEGIN

FOR tot\_rec IN tot\_cur

LOOP

DBMS\_OUTPUT.PUT\_LINE

('Total is: ' || tot\_rec.total\_sales);

END LOOP;

END;

1. Reescribe el siguiente bloque para usar un parámetro de cursor. Luego, vuelva a escribir para usar un módulo local, así como un parámetro de cursor.

DECLARE

CURSOR dept10\_cur IS

SELECT dname, SUM (sal) total\_sales

FROM emp

WHERE deptno = 10;

dept10\_rec dept10\_cur%ROWTYPE;

CURSOR dept20\_cur IS

SELECT dname, SUM (sal)

FROM emp

WHERE deptno = 20;

dept20\_rec dept20\_cur%ROWTYPE;

BEGIN

OPEN dept10\_cur;

FETCH dept10\_cur INTO dept10\_rec;

DBMS\_OUTPUT.PUT\_LINE

('Total for department 10 is: ' || tot\_rec.total\_sales);

CLOSE dept10\_cur;

OPEN dept20\_cur;

FETCH dept20\_cur INTO dept20\_rec;

DBMS\_OUTPUT.PUT\_LINE

('Total for department 20 is: ' || tot\_rec.total\_sales);

CLOSE dept20\_cur;

END;