

PREPARACIÓN RESOLUCIÓN CASO TIPO PRUEBA.

PRUEBA 2 PROGRAMACIÓN BASE DE DATOS

01-06-23

① - CURSOR QUE RECUPERA LAS PROFESIONES

```
DECLARE
CURSOR c1 IS
SELECT *
FROM PROFESION
WHERE COD-PROFESION IN ( SELECT COD-PROFESION
FROM PROFESIONAL);
```

COD-PROFESION	NOMBRE-PROFESION	ASIGNACION
1	CONTADOR AUDITOR	12,3
2	" GENERAL	8
3	INGENIERO INFOR	14,36
4	"	21,34
5	"	
6	"	

SOLUCION PRUEBA PROPUESTA

```
DECLARE
CURSOR C-PROFESION IS
SELECT NOM-PROF, ASIG
FROM PROFESION
ORDER BY NOMBRE-PROFESION;
```

	NOMBRE-PROFESION	ASIGNACION
1	ABOGADO	
2	ARQUITECTO	
3		
4		
5		
6		
7		
8...		

Res: APARECEN 2 PROFESIONES MAS,
YA QUE 2 PROF NO TIENEN
COD?

② CURSOR QUE RECUPERA DATOS DE LOS PROFESIONALES DE UNA PROFESION DETERMINADA, QUE POSEAN UN PROMEDIO SUPERIOR AL PROMEDIO DE HONORARIOS EN EL MES PROCESADO.

CURSOR PARAMETRIZABLE, RECIBE COMO PARAMETRO EL NOMBRE DE LA PROFESION Y DEVUELVE A LOS TRABAJADORES Y SUS ATRIBUTOS.

```
CURSOR C-PROFESIONAL (P-PROFESION VARCHAR 2) IS
SELECT numrun-prof, P.nombre, Pr.nombre-profesion, p.cod-contrato, p.cod-comuna
COUNT(*) ASESORIAS, SUM(A.honorarios) honorarios
```

```
FROM Profesional P JOIN profesion PR USING (cod-profesion)
JOIN asesoria A USING (numrun-prof)
```

```
WHERE to_char(A.inicio-asesoria, 'YYYYMM') = :b-fecha '202106'
AND PR.nombre-profesion = P-profesion.
```

GROUP BY

ORDER BY ; (8 COLUMNAS).

					SUELDO
					ASESORIAS
					HONORARIOS
RUN	NOMBRE	NOMBRE-PROFESION	COD-TPCONTRATO	COD-COMUNA	PUNTAJE ...

PARTE 2

-- VARIABLES ESCALARES (11 TOTAL)

```
V_msg VARCHAR2(300);  
V_msgusr "  
V_codemp NUMBER(2);  
V_comuna COMUNA.VOM.COMUNA %TYPE;  
V_ASIGNMOV NUMBER(8) := 0;  
... +6 (TOTAL 7 CON VALOR := 0);
```

-- VARIABLES ESCALARES ACUMULADORAS (7 TOTAL)

```
V_TOT-ASESORIAS NUMBER(8) := 0;  
V-TOT
```

-- VARRAY

```
TYPE t-descontos IS VARRAY(6) OF NUMBER;
```

```
V-desctos t-descontos := t-descontos(0.02, 0.04, 0.05, 0.07, 0.09, 0.15);
```

-- EXCEPCION PARA EL ERROR DE USUARIO
e_asignacion_limite EXCEPTION;

BEGIN

-- TRUNCAR TABLAS

```
EXECUTE IMMEDIATE 'TRUNCATE TABLE errores p';  
" " " detalle_asignacion_mes';  
" " " resumen_mes_profesion';  
" ' DROP SEQUENCE sq-error';  
" ' CREATE SEQUENCE sq-error';
```

-- cursor que recorre las profesiones

```
FOR r-profesion IN c-profesion LOOP
```

-- Reiniciamos las variables acumuladoras

```
V_tot-asesorias := 0;
```

```
...  
6
```

```
;
```


FOR r-profesion IN c-profesion LOOP

-- reiniciamos las variables acumuladoras

vector que habíamos parametrizado

FOR r-profesional IN c-profesional (r-profesion, nombre_profesion) LOOP

-- Asignacion por traslado

SELECT nom_comuna, codemp_comuna
INTO v-comuna, v-codemp-comuna
FROM COMUNA
WHERE cod_comuna = r-profesional.cod_comuna;

OBTIENE EL DATO NOMBRE LA COMUNA Y EL CODIGO EMPRESARIAL



IF v-comuna != 'PROVIDENCIA' THEN

EVALUA EL NOMBRE Y COD SEGUN LA REGLA DE NEGOCIO



v-asigmov := CASE v-codemp-comuna

WHEN 10 THEN ROUND(
r-profesional.honorarios * v_detalle)
↑
vector
v array

CONDICIONES
1. DISTINTO DE 'PROVIDENCIA' Y ≠ 'NULL'
1
2
3
4
2. = 'NULL'
\$25.000 END;

en varray son 6 valores pero en solución propuesta corregir, solo usamos 5 valores

END IF;

-- DETERMINAMOS ASIGNACION POR EVALUACION DEL PROFESIONAL

OSO SE ABRE OTRO BEGIN

BEGIN

SELECT porcentaje / 100
INTO v-pcteva
FROM Evaluation
WHERE r-profesional.puntaje BETWEEN

regla negocio

eva-punt-min AND
eva-punt-max;

OBTIENE PORCENTAJE DE EVALUACION, TODO CORRECTO

¡OJO! PURA CACHATIVA!

EXCEPTION

WHEN no_data_found THEN

v-msg := SQLERRM -- fuente? de soda no se especifica significado

v-pcteva := 0;

v-msgusr := 'NO se encontro % evaluacion' || TO_CHAR(r-profesional, run, '09G999G999');

INSERT INTO errores-p

VALUES (sq-erro.NEXTVAL, v-msg, v-msgusr);

WHEN too-many-rows THEN

END; v-asigeva := ROUND(r-profesional.honorarios * v-pcteva);

3

calcula el valor en dinero de asignacion por evaluacion

d) - Incentivo tipo de contrato (suma honorarios, % repuestos)

```
SELECT incentivo (porcentaje)
INTO v-pcttipoc
FROM tipo-contrato
WHERE cod_tpcontrato = r-profesional. cod_tpcontrato;

# OBTIENE EL DATO
```

```
v-asigtipoc := ROUND (r-profesional. honorarios * v-pcttipoc / 100);

# HACE EL CALCULO.
```

e) - Determinamos el monto de asignación profesional con un cruce de información extraída de ambos cursors y consultada en los respectivos tablas para obtener sueldo y porcentaje

```
v-asigprof := ROUND (r-profesional. sueldo * r-profesion. asignacion / 100);
```

f) - SUMAMOS TOTAL DE ASIGNACIONES

```
v-asig-prof := v-asigmov + v-asigtipoc + v-asigprof; -
```

OTO! SE ABRE OTRO BEGIN Y SE CALCULA LA EXCEPCION
- - - Preparamos esemotto para levantar le exp user.

BEGIN

```
IF v-asig-prof > b-totaleslimite THEN
  RAISE e-asignacion-limite;
```

END IF;

EXCEPTION

```
WHEN e-asig-lim THEN
```

```
v-msg := '...'
```

```
INSERT INTO errores-p
```

```
VALUES (sq-error.NEXTVAL, v-msg, '...' || v-asig - TOCHAR ...));
```

REGLA NEGLO → v-asig-prof := b-totaleslimite;
END;

* tener cuidado al momento de cerrar los acumulador de insertar datos de manera correcta.

* se realiza INSERT de tablas:

1. errores - p durante el transcurso del código
2. detalle - asignacion - mes :
3. resumen - mes - profesion :

-- Insertamos en la tabla detalle

INSERT INTO detalle - asignacion - mes

VALUES (SUBSTR (:b_fecha, -2), SUBSTR (:b_fecha, 1, 4),
r_profesional.RVN, r_profesional.nombre, r_profesional.nombre - profesion,
r_profesional.asesorias, r_profesional.honorarios, v_asigmov, v_asiggera,
v_asigtipoc, v_asigprof, v_asignaciones - profesional);

~~ACUMULAMOS EN LAS VARIABLES SUMATORIAS~~ **IMPORTANT!** ↓

v_tot_asesorias := v_tot_asesorias + r_profesional.asesorias;
honorarios := v_tot_honorarios + r_profesional.honorarios
mov
gera
+ v_asigmov
tipo con
profesion
totales

~~END LOOP;~~ # se cierra loop detalle empleado cursor.

- INSERT EN TABLA RESUMEN

INSERT INTO resumen - mes - profesion

VALUES (:b_fecha, r_profesion.nombre - profesion, v_tot_asesorias, v_tot_honorarios, v_tot_asigmov,
v_tot_asiggera, v_tot_tipoc, v_tot_asigprof, v_tot_totales);

~~END LOOP;~~
~~COMMIT;~~ → END;

CRISTÓBAL G. GONZ.