

### Taller - PLSQL.

Ejecute el siguiente código:

```
alter table ENTREGA_ELEMENTOS
add DURACION NUMBER DEFAULT 0;

update ENTREGA_ELEMENTOS set DURACION = 30;

alter table EMPLEADOS
add CLAVE VARCHAR2(250) DEFAULT '---';
```

Realice los siguientes ejercicios:



Cree un bloque anónimo que visualice los números pares del 0 al 100.



Cree un bloque anónimo que visualice los números primos del 0 al 100.



Cree un bloque anónimo que solicite dos números y permita conocer cuál de los dos números es mayor o indicar si es el caso si son iguales.



Cree un bloque anónimo que ingrese 1000 registros a la siguiente tabla:

```
CREATE TABLE TMP_MISDATOS
(
  ID INTEGER PRIMARY KEY,
  DESCRIPCION VARCHAR2(30) NOT NULL
);
```



**Nota: El campo ID debe ser un número primo y el campo descripción debe tener la siguiente cadena de texto: 'El identificador es 2', donde el numero es el id que generarón.**



Cree un bloque anónimo que actualice la fecha de devolución de la tabla ENTREGA\_ELEMENTOS, donde la duración (días) de cada elemento esta en el campo duración.



Cree un procedimiento para el punto 1 con el nombre de: pares.



Cree un procedimiento para el punto 2 con el nombre de: primos.



Cree un procedimiento para el punto 3 con el nombre de: comparar\_numeros, este procedimiento debe recibir por parámetros los dos números a comparar.



Cree un procedimiento para actualizar la columna email de la tabla empleados con la nueva dirección de correo asignada por la empresa. La nueva dirección asignada debe estar conformada por:

- a. La inicial del primer nombre + el primer apellido + el segundo apellido + unillanos.edu.co.

Ejemplo: Ricardo Andrés Sandoval Morales -> [rsandovalmorales@unillanos.edu.co](mailto:rsandovalmorales@unillanos.edu.co)

Además, debe generar la clave del correo la cual debe estar conformada por:








Inicial del primer nombre + inicial del segundo nombre + la identificación



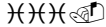



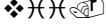
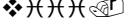
- b. Debe validar que no se repitan las direcciones, si se repita alguna, debe agregar un numero al final del nombre de la dirección de correo, el numero es un consecutivo que debe iniciar en 1.



Cree una función para conocer el precio promedio de un elemento de protección, la función debe recibir como parámetro el código del elemento.

## ADMINISTRACION DE BASES DE DATOS





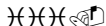


-  Cree una función que determine si un trabajador actualmente se le han entregado todos los elementos de protección que le fueron asignados, la función debe retornar un valor entero(1 o 0).
-  Realizar una consulta que muestre: la identificación, el nombre del empleado, el cargo y una columna que indique si le entregaron (S/N) los elementos de protección.
-  Cree una función para conocer si un estudiante está matriculado actualmente, se entiende por matriculado que el estudiante por lo menos halla inscrito una materia en un periodo y año determinado. La función debe recibir como parámetro el código del estudiante, periodo y año, debe retornar un valor booleano.
-  Cree una función que devuelva el año de la fecha que tienen el servidor de Oracle. La función no recibe parámetros, pero devuelve un valor numérico
-  Cree una función que determine el semestre del año en que se encuentra actualmente, tener como base la fecha del servidor de Oracle. La función no recibe parámetros, pero devuelve un valor numérico.
-  Cree una función para conocer el promedio de carrera de un estudiante, solo tener en cuenta las materias aprobadas, la función debe recibir como parámetro el código del estudiante.
-  Cree un procedimiento que muestre en pantalla la siguiente información de un estudiante:

-  Código.
-  Facultad.
-  Programa.
-  Nombres y apellidos.
-  Promedio de carrera de materias aprobadas.
-  Estudiante está matriculado actualmente(S/N).
-  Año actual.
-  Periodo actual.

Además, esta información debe quedar almacenada en la siguiente tabla:

```
CREATE TABLE TMP_ESTUDIANTES
(
  CODIGO          INTEGER NOT NULL,
  FACULTAD        VARCHAR2(50) NOT NULL,
  PROGRAMA        VARCHAR2(50) NOT NULL,
  ESTUDIANTE      VARCHAR2(200) NOT NULL,
  PROMEDIO        INTEGER (5,3) NOT NULL,
  MATRICULADO     VARCHAR2(1)  NOT NULL,
  AÑO             INTEGER      NOT NULL,
  PERIODO         INTEGER      NOT NULL
);
```

Pero, solo para aquellos estudiantes cuyo promedio de carrera sea mayor o igual a 3.8. Cada vez que se llame al procedimiento, este debe vaciar la tabla **TMP\_ESTUDIANTES**.

-  Cree una función que retorne el número de estudiantes inscritos por materia, la función debe recibir como parámetros el código de la materia, el código del programa, código de facultad, el año y periodo.
-  Cree un dato compuesto de tipo tabla, que almacene la información del punto 18 (Solo la primera parte), con base en esta información debe mostrar:
  -  El índice de la tabla y los datos.
  -  Total de datos de la tabla.
  -  Validar si la clave 10 existe, y si existe debe mostrar la información que contiene, sino, debe indicar que no existe.
  -  Mostrar la información de la primera y la última clave.
  -  Eliminar los índices de la tabla comprendidos entre 1 y 5, utilice la función **delete(m,n)**.