# PROCEDIMIENTOS

Elaborar un procedimiento almacenado que reciba como parámetro una cantidad salarial. Este procedimiento imprimirá los pedidos gestionados por empleados cuyos salarios sean mayores que el salario proporcionado.

Crear una función que reciba un empleado y debe devolver cuantos empleados son los que tienen mayor salario.

**ESTRUCTURA DE PROCEDIMIENTO**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**FUNCION**

**ESTRUCTURA DE FUNCION**

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Update

Texto, Carta

Descripción generada automáticamente

Crea un procedimiento para dar de alta a unas habitaciones que tiene una fk en tipo habitación y comprobar que ese tipo de habitación exista si no existe no dará de alta ese tipo de habitación.

Texto

Descripción generada automáticamente

EJERCICIO TEMPORADAS

MANUALMENTE A LA TEMPORADA DARLE DE ALTA MANUALMENTE

Y TENIENDO MINIMO UNA TEMPORADA DAR Alta a los precios HABITACION

Si se comprueba

1-Que ese tipo existe

Entonces luego pasara a comprobar si

2- esa temporada existe

Si existe pasara a comprobar

3- que el precio sea mayor a cero

Si todo esto existe dará de alta la temporada

Si no existe el tipo de habitación

Dará como resultado en un p\_salida =-1

Si no existe esa temporada

Dará como resultado en un p\_salida =-2

Si el precio es 0

Dará como resultado en un p\_salida =-3

**CREATE OR REPLACE PROCEDURE pr\_AltaPrecio**

**(p\_codTemporada in number,P\_tipoHabitacion in number,p\_precio in number,P\_SALIDA OUT INT)**

**as**

**V\_TEMPORADA INT;**

**V\_TIPO INT;**

**begin**

**/\*ASEGURAR DE QUE ESE TIPO DE HABITACION EXISTE\*/**

**SELECT count(\*) INTO V\_TIPO**

**FROM TIPOHABITACION**

**WHERE CATEGORIA = P\_tipoHabitacion;**

**IF V\_TIPO <> 0 THEN /\* EXISTE EL TIPO DE HABITACION SI SE PUEDE DAR DE ALTA LA TEMPORADA\*/**

**/\*COMPROBAR QUE EL TIPO EXISTE\*/**

**SELECT count(\*) INTO V\_TEMPORADA**

**FROM TEMPORADA**

**WHERE TEMPORADA =p\_codTemporada;**

**IF V\_TEMPORADA <> 0 THEN**

**IF p\_precio > 0 then**

**INSERT INTO PRECIOHABITACION (TIPOHABITACION,TEMPORADA,PRECIO)**

**VALUES (P\_tipoHabitacion,p\_codTemporada,p\_precio);**

**else**

**P\_SALIDA:=-3;**

**END IF;**

**else**

**P\_SALIDA:=-2;**

**END IF;**

**else**

**P\_SALIDA:=-1;**

**end IF;**

**end;**

Solo hemos creado una temporada con el código 1 y un tipo de habitación con el código 1

Pondremos un tipo de habitación no existente para que de -1

Imagen que contiene Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Pondremos un tipo de temporada que no existente para que de -2

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Pondremos el precio de la habitacion a 0 para que de -3

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

EJERCICIO 3

ALTA RESERVAS,añadir una reserva a la base de datos

Primeramente vamos a observar en la base de datos que tablas están relacionadas con RESERVAS en este caso tenemos LINEARESERVAS y CLIENTES.

La tabla principal será RESERVAS ya que las fk de LINEARESERVAS nos enseña que tiene fk a RESERVAS y

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente