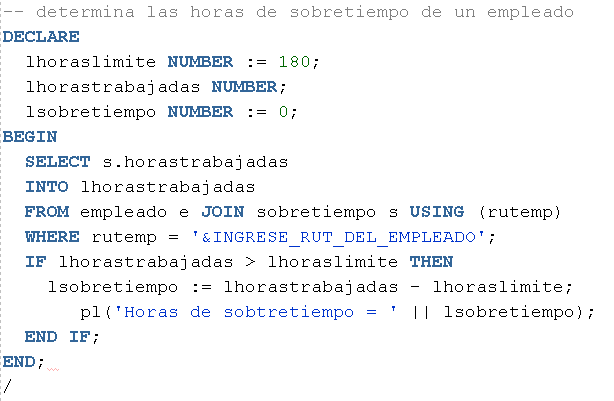
Guión de Autoestudio – Estructuras de Control

**Instrucciones**

Crea un nuevo usuario y ejecuta el script **CreaBaseEmpleados** que está disponible en el sitio. Una vez creada la base, no olvides crear y compilar el procedimiento PL para mostrar las salidas.

**Controlando el flujo con un IF simple**

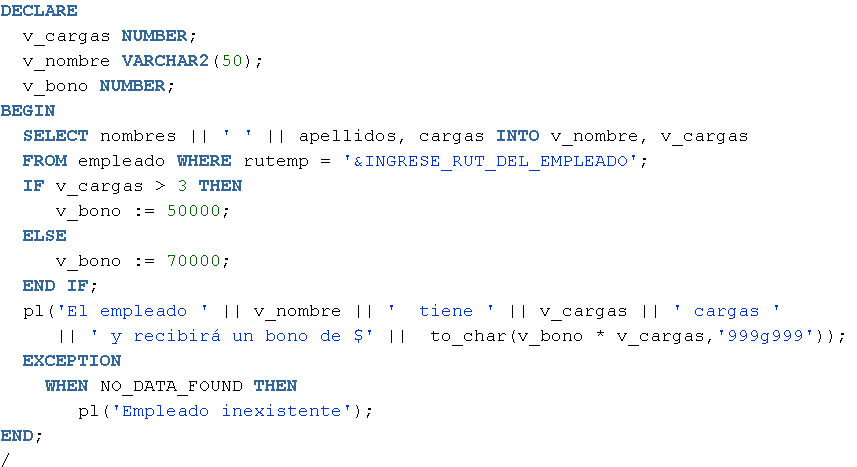
Digita y ejecuta el código de la imagen inferior. El bloque establece que el límite de horas de un empleado es 180. Luego, en el sector ejecutable, se recuperan las horas trabajadas por el empleado cuyo rut se indique en la variable de sustitución. Luego se pregunta si las horas trabajadas exceden el límite y se calculan y muestran las horas de sobretiempo.



**Agregando ELSE a un IF**

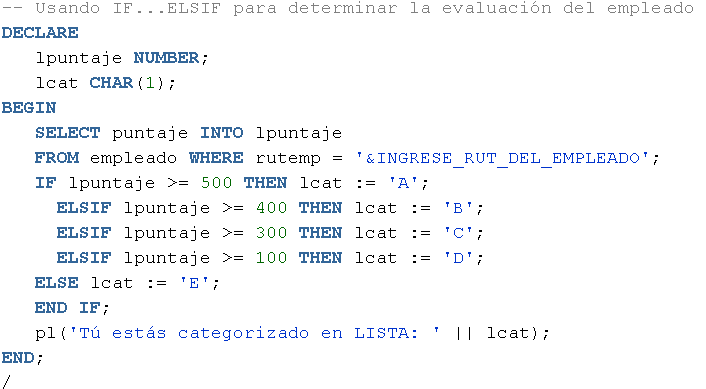
Digita y ejecuta el código de la imagen inferior. En la sección ejecutable del bloque se recuperan el nombre y el número de cargas del empleado cuyo rut se indique en la variable de sustitución. Luego se pregunta si el número de cargas es mayor que 3, en cuyo caso el valor del bono se establece en $50.000 por carga; en caso contrario, el monto del bono será equivalente a $70.000.

Luego se calcula y muestra el monto del bono que corresponde al empleado recuperado. Por último, se controla que el rut ingresado exista con la excepción respectiva.



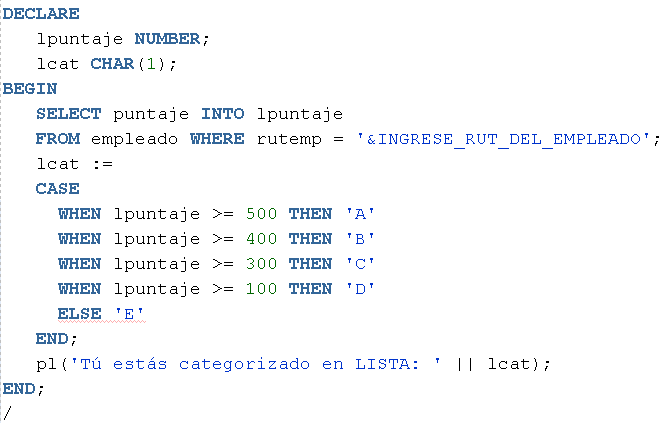
**Usando ELSIF .. THEN**

En la sección ejecutable, el bloque recupera el puntaje del empleado cuyo rut se ingrese en la variable de sustitución. Luego el bloque evalúa el puntaje recuperado y puebla la variable **lcat** con la letra de la lista que le corresponde. Más tarde se indica cuál es la categoría del empleado.



**Uso de la expresión CASE**

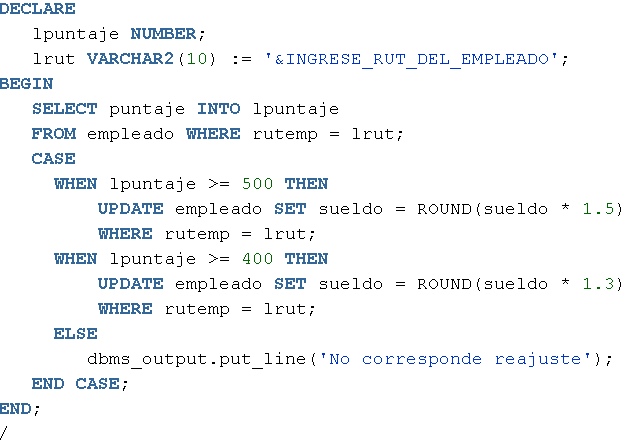
En la sección ejecutable, el bloque recupera el puntaje del empleado cuyo rut se ingrese en la variable de sustitución. Luego el bloque evalúa el puntaje recuperado y puebla la variable **lcat** con la letra de la lista que le corresponde. Más tarde se indica cuál es la categoría del empleado.



**Usando la sentencia o instrucción CASE**

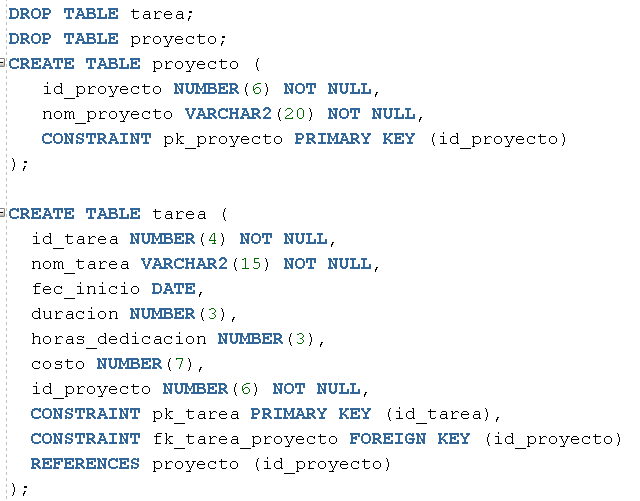
Recuerda que la sentencia CASE se diferencia de la expresión, porque la primera ejecuta algunas de las acciones declaradas de acuerdo cómo se evalúe la condición especificada.

En el ejemplo se recupera el puntaje del empleado cuyo rut se pase como parámetro. Luego se evalúa el puntaje recuperado. Si este es mayor que 500 se otorga un aumento de salario de un 50%, si es mayor que 400 el aumento es de un 30%. En cualquier otro caso no aplica.



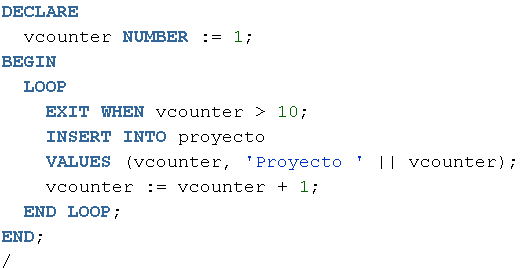
**Preparando el entorno para los ejemplos de estructuras iterativas (Mostrar IMG 1)**

En un nuevo usuario o en el actual, digita el código de la imagen y luego ejecútalo para crear la base de datos.



**Comenzando con un LOOP BASICO**

Se requiere insertar 10 proyectos en la tabla **proyecto** usando un LOOP BÁSICO. El bloque declara una variable numérica inicializada en 1 para usarla como un contador. El número de proyecto debes obtenerlo del contador, para el nombre, puedes usar un valor genérico, por ejemplo, “Proyecto 1”. El bucle debe terminar cuando el contador valga 10. Digita y ejecuta el código de la imagen.



**Probando un bucle WHILE .. LOOP.**

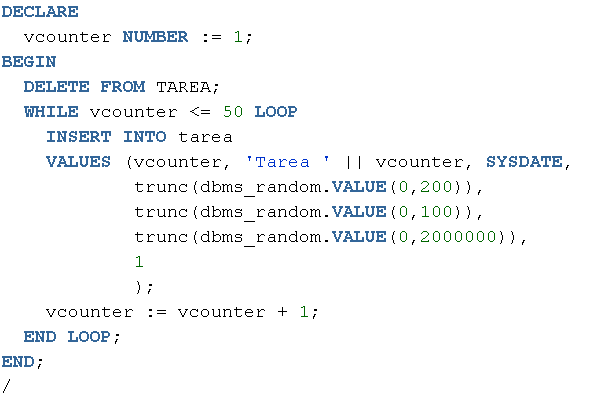
Se desea crear un bloque que inserte 50 tareas para alguno de los proyectos existentes en la tabla proyecto. Elegiremos el proyecto número 1, para este caso.

La id de la tarea debe obtenerse del contador, el nombre de la tarea, que puede ser genérico, “Tarea 1” por ejemplo, debe construirse concatenando la cadena “Tarea 1” y el contador. La duración debe ser un numero aleatorio entre 0 y 200, las horas de dedicación un número aleatorio entre 0 y 100, el costo un número aleatorio entre 0 y 2000000. Los números obtenidos deben redondearse o truncarse. El número de proyecto debe ser el de un proyecto existente. Si pruebas varias veces el código usa el comando DELETE como lo hiciste antes.

Para obtener un número aleatorio puedes usar la función value del paquete dbms\_random con la siguiente sintaxis:

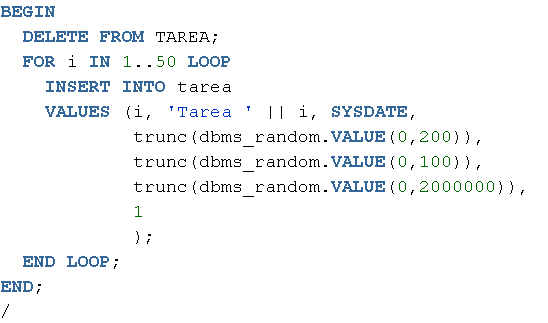
**TRUNC(DBMS\_RANDOM.VALUE(numero\_inicial, numero\_final))**

Digita y ejecuta el código de la imagen inferior.



**Probando un bucle FOR .. LOOP.**

Intenta adaptar el ejemplo anterior para que funcione con un bucle FOR .. LOOP. Si no logras hacerlo, entonces usa el código de la imagen.



**Bucles anidados, en reversa y etiquetas**.

Digita el código de los ejemplos para que veas en acción bucles anidados.

