

Para la realización de los ejercicios de esta unidad nos basaremos en el caso de estudio expuesto en los contenidos de la misma en el Anexo I. La tarea que te pedimos que realices consta de 2 actividades:

Ejercicio 1

Crear un procedimiento que permita cambiar a todos los agentes de una familia determinada (familia origen) a otra familia (familia destino).

El procedimiento tendrá la siguiente cabecera `CambiarAgentesFamilia (id_FamiliaOrigen, id_FamiliaDestino)`, donde cada uno de los argumentos corresponde a un identificador de Familia. Cambiará la columna `Identificador de Familia` de todos los agentes, de la tabla `AGENTES`, que pertenecen a la Familia con código `id_FamiliaOrigen` por el código `id_FamiliaDestino`.

Previamente comprobará que ambas familias existen y que no son iguales.

Para la comprobación de la existencia de las familias se puede utilizar un cursor variable, o contar el número de filas y en caso de que no exista, se visualizará el mensaje correspondiente mediante una excepción del tipo `RAISE_APPLICATION_ERROR`. También se mostrará un mensaje en caso de que ambos argumentos tengan el mismo valor.

El procedimiento visualizará el mensaje "Se han trasladado XXX agentes de la familia XXXXXX a la familia ZZZZZZ" donde XXX es el número de agentes que se han cambiado de familia, XXXXXX es el nombre de la familia origen y ZZZZZZ es el nombre de la familia destino.

Ejercicio 2.

Queremos controlar algunas restricciones a la hora de trabajar con agentes:

- ✓ La longitud de la clave de un agente no puede ser inferior a 6.
- ✓ La habilidad de un agente debe estar comprendida entre 0 y 9 (ambos inclusive).
- ✓ La categoría de un agente sólo puede ser igual a 0, 1 o 2.
- ✓ Si un agente tiene categoría 2 no puede pertenecer a ninguna familia y debe pertenecer a una oficina.
- ✓ Si un agente tiene categoría 1 no puede pertenecer a ninguna oficina y debe pertenecer a una familia.
- ✓ Todos los agentes deben pertenecer a una oficina o a una familia pero nunca a ambas a la vez.

Se pide crear un disparador para asegurar estas restricciones. El disparador deberá lanzar todos los errores que se puedan producir en su ejecución mediante errores que identifiquen con un mensaje adecuado por qué se ha producido dicho error.

Algunas de las restricciones implementadas con el disparador se pueden incorporar a la definición del esquema de la tabla utilizando el Lenguaje de Definición de Datos (Check, Unique,...). Identifica cuáles son y con qué tipo de restricciones las implementarías.