

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

Teniendo en cuenta que en el caso de estudio una de las tablas es FAMILIA, que su estructura es (identificador(pk), nombre, familia(fk), oficina(fk)), que contiene información acerca de una familia, su nombre, el identificador de la familia de la que depende (su madre) y su oficina.

Queremos crear un procedimiento que tendrá la siguiente cabecera mover_familia (id_origen, id_destino), donde cada uno de los argumentos corresponde a un identificador de familia. El procedimiento intentará que la familia a la que corresponde id_origen, pase a ser la hija de la familia identificada por id_destino, o de otra manera, su familia madre será la especificada en el argumento id_destino.

Para realizar esta operación, será necesario comprobar que ambas familias existen y que no son iguales.

Para la comprobación de la existencia de dichas familias se puede utilizar un cursor variable 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.

Además si la familia origen pertenecía a una oficina deberá dejar de pertenecer a esa oficina, ya no pertenecerá a ninguna y sólo ser hija de la familia destino.

Ejercicio 2.

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

- ✓ El usuario y la clave de un agente no pueden ser iguales.
- ✓ 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 pertenece a una oficina directamente, su categoría debe ser igual 2.
- ✓ Si un agente no pertenece a una oficina directamente, su categoría no puede ser 2.
- ✓ No puede haber agentes que no pertenezcan a una oficina o a una familia.
- ✓ No puede haber agentes que pertenezcan a una oficina y a una familia a la vez.

Debes 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.