

Video Club

Tarea 3

Objetivos

- Experimentar en el desarrollo de **procedimientos almacenados** para resolver problemas sencillos directamente en la base de datos.
- Experimentar en el uso de **triggers** como mecanismo de control del cumplimiento de restricciones de integridad, diferir tareas sobre la base de datos, etc.
- Practicar el uso del lenguaje PL/pgSQL.

Uso de procedimientos almacenados

1) Premio al personal de las sucursales.

Cada mes junto con su sueldo se les entrega un bono a aquellos empleados que hayan logrado ciertos objetivos.

Si los empleados superan los 30 alquileres realizados en el mes se les entrega un bono de \$1000. Además, una vez superados los 50 alquileres, se agregan al bono \$5 por cada alquiler.

Ejemplo: Si un empleado en un mes realiza 57 alquileres le corresponde un bono de \$1035

Se pide: Implementar procedimientos almacenados que permitan calcular el valor del bono para cada empleado.

En primera instancia se debe implementar un procedimiento que dado un empleado, un mes y un año calcula el valor del bono que le corresponde al empleado en el mes - año dado.

Y a continuación se debe implementar un procedimiento que dado un mes y un año almacene en una tabla **bonos** una tupla por cada empleado de cada sucursal que contiene el mes, año, datos del empleado y el valor del bono para ese empleado en ese mes.

En los siguientes cuadros se resumen los procedimientos a implementar:

Nombre	premioper	
Parámetros	Entrada	idpersonal (integer), mes (integer), año (integer)
	Salida	bono (integer)
Respetar el orden de los parámetros de entrada		
Mensajes de error	'Empleado incorrecto'	
	'Mes incorrecto'	
Acción	Para el empleado dado en idpersonal calcula el valor del bono en el mes y año dado. Se debe controlar que el empleado sea correcto (pertenzca a una sucursal) y que el número del mes sea un mes válido. En caso de error se mostrará el mensaje de error que corresponda de los indicados y no se calcula el bono.	

Nombre	premiogral	
Parámetros	Entrada	mes (integer), año (integer)
	Salida	Respetar el orden de los parámetros de entrada
Mensaje de error	'Mes incorrecto'	
Acción	Para cada empleado de cada una de las sucursales calcula el valor del bono en el mes - año dado y carga una tupla en la tabla bonos . Se debe controlar que el número del mes sea un mes válido. En caso de error se mostrará el mensaje de error indicado y no se calcula el bono	

El esquema correspondiente a la tabla **bonos** es:

bonos(mes,año,idpersonal,nombrepersonal,bonomes), donde:

Atributo	Tipo	Contiene
mes	text	Nombre del mes (Enero, Febrero,...Diciembre)
año	integer	Valor del año
idpersonal	integer	Identificador del empleado para el cual se calcula el valor del bono
nombrepersonal	text	Nombre del empleado, con el mismo formato que en la consulta 8 de la tarea 1
bonomes	integer	Valor del bono que le corresponde al empleado en ese mes – año.

Se debe entregar: un archivo llamado parte1.sql que contenga la creación de las tablas y los procedimientos y/o funciones necesarias para esta parte.

Uso de triggers

2) Registro de operaciones sobre películas

Interesa llevar un registro sobre las operaciones de insert, update y delete que se realizan sobre la tabla de películas.

Para esto se trabaja sobre una tabla operpel, que contiene una tupla por cada operación realizada sobre la tabla independientemente de la cantidad de tuplas que se afecten en la operación. El esquema de la tabla operpel es:

operpel (id, accion, fecha, usuario), donde

Atributo	Tipo	Contiene/comentario
id	integer	Valores de una secuencia
accion	char	Valores posibles: I,U,D que representa las operaciones de insert, update y delete respectivamente
fecha	date	Fecha en que se realiza la operación
usuario	text	Usuario que realiza la operación

Datos sobre la secuencia a utilizar para el atributo id:

Nombre	idlogpel
Valor inicial	1
Incremento	1

Se pide: Implementar el trigger que permita realizar el registro antes descrito.

En el siguiente cuadro se resumen las características del trigger a implementar.

Nombre Trigger	grabar_operaciones_pel
Tabla	peliculas
Acción	Inserta una tupla completa en la tabla operpel con los datos de la operación que se realiza sobre la tabla películas
Observación	Se registran únicamente las operaciones que tienen efecto sobre la tabla películas, si una operación falla o no tiene efecto sobre la tabla NO se debe registrar

Se debe entregar: un archivo llamado parte2.sql que contenga la creación de la secuencia, creación de tablas y la creación de los triggers necesarios para esta parte.

3) Registro de operaciones sobre alquileres y pagos

Unas de las tablas más sensibles de la base de datos son las tablas **alquileres** y **pagos** por lo que se desea llevar un registro de las tuplas afectadas por las operaciones que se realizan sobre estas tablas.

Para esto se trabaja sobre una tabla **regalqpag** cuyo esquema es:

regalqpag (idop, accion, tabla, fecha, usuario, idpel, idcli, idsuc, fechaalq) donde:

Atributo	Tipo	Contiene/comentario
idop	integer	Valores de una secuencia
accion	char	Valores posibles: I,U,D que representa las operaciones de insert, update y delete respectivamente
tabla	name	Nombre de la tabla sobre la que se realiza la operación 'alquileres' o 'pagos' según el caso.
fecha	date	Fecha en que se realiza la operación
usuario	text	Usuario que realiza la operación
idpel	integer	<u>A partir de una operación en la tabla alquileres:</u> - Identificador de la película que corresponde al alquiler <u>A partir de una operación en la tabla pagos:</u> - Identificador de la película alquilada sobre la que se realiza un pago. En ambos casos en caso de tratarse de una operación de update se deben registrar los valores nuevos.
idcli	integer	<u>A partir de una operación en la tabla alquileres:</u> - Identificador del cliente que realiza el alquiler <u>A partir de una operación en la tabla pagos:</u> - Identificador del cliente que realizó el alquiler correspondiente al pago que se realiza. En ambos casos en caso de tratarse de una operación de update se deben registrar los valores nuevos.
idsuc	integer	<u>A partir de una operación en la tabla alquileres:</u> - Identificador de la sucursal donde se realiza el alquiler <u>A partir de una operación en la tabla pagos:</u> - Identificador de la sucursal donde se realizó el alquiler correspondiente al pago que se realiza. En ambos casos en caso de tratarse de una operación de

		update se deben registrar los valores nuevos.
fechaalq	date	<p>A partir de una operación en la tabla alquileres:</p> <ul style="list-style-type: none"> - fecha en la que se realiza el alquiler <p>A partir de una operación en la tabla pagos:</p> <ul style="list-style-type: none"> - fecha en la que se realizó el alquiler correspondiente al pago que se realiza. <p>En ambos casos en caso de tratarse de una operación de update se deben registrar los valores nuevos.</p>

Datos sobre la secuencia a utilizar para el atributo idop

Nombre	logidseq
Valor inicial	1
Incremento	1

Observar que a diferencia del registro anterior en este caso interesa tener una tupla en la tabla **regalqpag** por cada tupla afectada por cada una de las operaciones realizadas sobre las tablas alquileres y pagos.

Se pide: Implementar triggers que permitan realizar el registro antes descrito.

En los siguientes cuadros se resumen las características de los triggers a implementar.

Nombre Trigger	grabar_operaciones_alq
Tabla	Alquileres
Acción	Inserta una tupla completa en la tabla regalqpag por cada tupla afectada por la operación que representa a la acción realizada sobre la tabla alquileres
Observación	Se registran únicamente las operaciones que tienen efecto sobre la tabla alquileres, si una operación falla o no tiene efecto sobre la tabla NO se debe registrar

Nombre Trigger	grabar_operaciones_pag
Tabla	Pagos
Acción	Inserta una tupla completa en la tabla regalqpag por cada tupla afectada por la operación que representa a la acción realizada sobre la tabla pagos
Observación	Se registran únicamente las operaciones que tienen efecto sobre la tabla pagos, si una operación falla o no tiene efecto sobre la tabla NO se debe registrar

Se debe entregar: un archivo llamado parte3.sql que contenga la creación de la secuencia, creación de tablas y la creación de los triggers necesarios para esta parte.

4) Disponibilidad de ejemplares para el alquiler

Para la realización de un alquiler es necesario que la sucursal donde se pretende realizar el alquiler tenga disponible un ejemplar de la película que el cliente desea llevar.

Se pide: Implementar triggers que solo permitan realizar alquileres si efectivamente la sucursal cuenta con el ejemplar necesario.

En el siguiente cuadro se resumen las características del trigger a implementar.

Nombre Trigger	disponiblealq
Tabla	alquileres
Acción	Solo permite la inserción de un alquiler en caso de que la sucursal cuente con el ejemplar necesario.
Mensaje de error	'No hay ejemplares disponibles'
Observación	Únicamente se controla la restricción de disponibilidad, cualquier otra restricción será controlada por el mecanismo que corresponda. En caso de no haber disponibilidad se debe mostrar un mensaje que lo indique.

Se debe entregar: un archivo llamado parte4.sql que contenga la creación del trigger necesario para esta parte.

5) Control de la política de alquiler

Tal como se planteó en la consulta 10 de la tarea 2 existe una política de alquiler que dice que no se debe realizar un nuevo alquiler a un cliente si al momento de realizarse este alquiler tiene más de un alquiler sin devolver cuyo plazo de préstamo ya venció y para el cual no haya realizado ningún pago.

Se pide: Implementar el trigger que solo permita realizar alquileres si se cumple con la política definida.

En el siguiente cuadro se resumen las características del trigger a implementar.

Nombre Trigger	cumplepol
Tabla	alquileres
Acción	Solo permite la inserción de un alquiler en caso de que se cumpla la política definida
Mensaje de error	'Hay alquileres pendientes atrasados y sin pagos'
Observación	Únicamente se controla la restricción de la política, cualquier otra restricción será controlada por el mecanismo que corresponda. En caso de no cumplir la política se debe mostrar un mensaje que lo indique.

Se debe entregar: un archivo llamado parte5.sql que contenga la creación del trigger necesario para esta parte.

6) Boletín con novedades

Cada cierto tiempo se envía a los clientes del video club un boletín con las ingresos de películas que se realizan en su sucursal.

Por cada película ingresada se notifica a los clientes de la sucursal que en el último año han realizado algún alquiler de una película que cumple por lo menos una de estas condiciones:

- tiene por lo menos una categoría en común con la nueva película
- tiene por lo menos un actor en común con la nueva película.

Para poder realizar este boletín cada vez que se ingresa una película en una sucursal (en el inventario de la sucursal) se genera una tupla en una tabla **boletines** que tiene el siguiente esquema:

boletines(fecha, idcliente, idpelicula, titulo, nombrecliente, mail, categorias, actores) donde:

atributo	tipo	Contiene
fecha	date	Fecha de ingreso de la película a la sucursal
idcliente	integer	Identificador del cliente al que hay que notificar del ingreso
idpelicula	integer	Identificador de la película ingresada
titulo	character varying(255)	Título de la película ingresada
nombrecliente	text	Nombre del cliente a notificar. Los valores de este atributo se obtienen como en la consulta 8 de la tarea 1.
mail	character varying(50)	Mail del cliente a notificar
categorias	text	Es una cadena que contiene el nombre de todas las categorías de la película separados por coma. Ejemplo: Action,Comedy,Horror. Observar que la cadena termina en el nombre de la última categoría sin ningún otro carácter como por ejemplo punto y que no hay blancos entre los nombres de las categorías
actores	text	Es una cadena que contiene el nombre completo de todos los actores de la película. El nombre de cada actor se forma como en la consulta 8 de la tarea 1 y la cadena tiene el mismo formato que el atributo categorías. Ejemplo: DAN TORN,SANDRA KILMER

Al momento de enviar los boletines se obtienen los datos de esta tabla y se borran todas sus tuplas para comenzar a generar los datos para el siguiente boletín.

Se pide: Implementar un trigger que genere los datos correspondientes a la tabla **boletines** por cada ingreso de una película a una sucursal.

En el siguiente cuadro se resumen las características del trigger a implementar.

Nombre Trigger	boletin
Tabla	inventario
Acción	Inserta una tupla completa en la tabla: boletines con la información necesaria para la notificación a cada cliente que corresponda para cada ingreso de películas en una sucursal.
Observación	Se registran únicamente los datos correspondientes a ingresos que efectivamente se realizan, si un ingreso falla por cualquier problema NO se debe registrar pero no es necesario indicarlo.

Se debe entregar: un archivo llamado parte6.sql que contenga la creación de tablas y la creación del trigger necesario para esta parte.

Consideraciones generales:

- No utilizar vistas.
- No utilizar secuencias salvo en los lugares que se pide explícitamente.
- En todas las funciones y o procedimientos a definir utilizar el lenguaje PL/pgSQL
- Respetar todos los nombres dados, para secuencias, atributos, tablas, triggers, funciones, etc.
- Respetar el orden dado de los parámetros en las funciones y/o procedimientos en los casos que existen.
- No definir triggers más allá de los pedidos explícitamente.
- Es posible definir todas las funciones y/o procedimientos (no de triggers) que desee.