

AMPLIACION de BASES DE DATOS

(Profesor : Héctor Gómez Gauchía)

Práctica 2 SEMANA 4: PLSQL dinámico con procedimientos y triggers

Resultado:

- Instrucciones y Resultados de las pruebas de cada apartado en un fichero de texto.
- los ficheros .sql de lo que desarrolles en sql y plsql,
- Haz lista de dudas concretas sin resolver sobre tus respuestas para consultar con el profesor

Modo de entrega: No se Entrega

- Si terminas alguno de los apartados indicado en pizarra, avisa al profesor para puntuar la parte de participación en clase.
- Los conceptos de esta práctica se evalúan en un examen de control (tal y como se indica en la Ficha Docente), cuya fecha se avisará a tiempo.

MATERIALES para estudiar la práctica

- Teoría: Tema PL/SQL / Desarrollo-PL-SQL.pdf
- Ejemplos de PLSQL triggers y procs.: Tema PL/SQL / Ejercicios . . . / PLSQL-ejemplos/ . . . / PLSQLejemplosEjecutablesTeoria
- Ejemplos de PLSQL Dinámico : Tema PL/SQL / Ejercicios . . . / PLSQL-ejemplos/ . . . / PLSQL-dinamico-para-CV.rar

APARTADO 1: Partir la tabla `Invierte` en las tablas `inversiones_XXXXXX` para cada empresa `XXXXXX`

Si has hecho la práctica 2 semana 3, deberías tener ya las nuevas tablas `inversiones_XXXXXX` con datos.

Si **no** la has hecho: ejecuta la solución de esa práctica, que está subida al campus V, en el orden adecuado. Creas todos los procedimientos y después ejecutas el `partir_tabla_invierte`

APARTADO 2: Hacer un trigger para cada tabla `inversiones_XXXXXX` para acumular todo lo insertado en cada tabla en una nueva tabla `SumaEmpresa`

1.- Crea en el editor una única tabla vacía `SumaEmpresa`, que tiene estos atributos (`nombreE,Cantidad`), donde tendremos una fila por cada empresa con la suma de las cantidades de todas las inversiones de esa empresa (ya indicamos en las pruebas del paso 3 cómo obtener las filas).

2.- Hacer un procedimiento `crear_triggers` que haga lo siguiente: crear un trigger `trig_suma_XXXXXX` para cada tabla `inversiones_XXXXXX`. Para saber los nombres `XXXXXX` de las empresas que tienes actualmente, debes obtenerlas de la tabla `tabs`, donde están todas las tablas cuyo nombre empieza por 'inversiones_' y luego lo cortas para quedarte solo con los nombres de las empresas.

El trigger `trigger_XXXXXX` se activa cuando se inserte una fila en la tabla `inversiones_XXXXXX` coincidiendo ambos `XXXXXX`. Lo que hace en la tabla `SumaEmpresa`, es acumular en una fila, la cantidad de la inversión. El atributo "nombreE" será el nombre de empresa `XXXXXX`. Esta tabla solo debe tener una fila para cada empresa. Si no existe la fila, el trigger la crea.

3.- Para probar estos triggers crea un procedimiento `pru_tri_suma` que haga lo siguiente:

- Borrar todas las filas (no la tabla) de la tabla `SumaEmpresa`.
- Haz un bucle para borrar todas las filas (no la tabla) de cada tabla `inversiones_XXXXXX`.

- Ejecuta `partir_tabla_invierte`
- Imprime cada fila de la tabla `SumaEmpresa`.

APARTADO 3: hacer un trigger que sirva de log de las operaciones en la tabla `SumaEmpresa`

1.- Crea en el editor una tabla vacía `LogSumas`, que tiene estos atributos: fecha y hora del sistema, LITERAL, Cantidad y NombreE. La fecha y hora es del tipo DATE y las obtienes con la función `sysdate`. Si quieres más precisión (para usar ese atributo como PK) usa el tipo timestamp y función `sysimestamp`. LITERAL es el nombre de la operación "INSERT" o "UPDATE" que activó el trigger.

2.- Crea un trigger `tri_log` que

- Se active con cada inserción en la tabla `SumaEmpresa` y que haga lo siguiente:
 - Crea una fila en la tabla `LogSumas`, con los atributos adecuados indicados arriba.
- Se active con cada update en la tabla `SumaEmpresa` y que haga lo siguiente
 - Crea una fila en la tabla `LogSumas`, con los atributos adecuados indicados arriba.

En cualquier caso debe dar una excepción y un mensaje estandar si hay una cantidad superior a cien millones de euros.

3.- Para probar este trigger:

- Modifica el procedimiento `pru_tri_suma` para que lo primero que haga es comprobar si está deshabilitado el trigger `tri_log`, si lo está debe habilitarlo.
- Ejecuta el procedimiento `pru_tri_suma` y comprueba el resultado
- Añade una inversión con cantidad superior a cien millones de euros para probar ese caso.

APARTADO 4: (PARA NOTA) Provoca un ciclo en la activación de triggers

- Crea un trigger más que, junto con los anteriores ya creados provoque un ciclo en la activación.
- Describe lo que pasa.

APARTADO 5: (PARA NOTA) No se deben repetir filas el proc. `partir_tabla_invierte`

Comprueba que, si se ejecuta dos veces `partir_tabla_invierte`, no debe insertar dos veces la misma fila (aunque tenga el número secuencial distinto, se considera que repites fila).