

AMPLIACION de BASES DE DATOS

(Profesor : Héctor Gómez Gauchía)

Práctica 2 SEMANA 3: DESNORMALIZACION con PLSQL dinámico

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 Dinámico : Tema PL/SQL / Ejercicios . . . / PLSQL-ejemplos/ . . . / PLSQL-dinamico-para-CV.rar

APARTADO 1: DESNORMALIZACION usando PLSQL dinámico

En la BDejemplo queremos aplicar *desnormalización* mediante *Particionamiento Horizontal* de la tabla `Invierte`, y así crear varias tablas: una tabla `inversiones_XXXXXX` con la partición de las inversiones para *cada* empresa `XXXXXX` dada en los parámetros del procedimiento a desarrollar. Además, como esperamos tener más empresas que se incorporen al sistema, queremos automatizar el proceso: cada vez que haya una nueva inversión, en vez de usar la tabla `Invierte`, llamaremos al procedimiento `Gestion_Inversion` (datos de la inversión) que se encarga de los pasos descritos después. **(debes crear exactamente estos elementos y generar datos adecuados para probar todos los casos posibles razonables).**

Cada tabla `inversiones_XXXXXX` tiene estos atributos (NumInv, DNI, Cantidad, Tipo). Donde NumInv es un número secuencial único para cada tabla (será la PK) que se obtiene de la secuencia `sec_XXXXXX`.

Procedimiento `Gestion_Inversion` (parámetros con datos de la inversión "actual").

PASOS dentro del procedimiento:

1. Comprueba que existe la empresa `XXXXXX` (la "actual") en la tabla `Empresa`
2. Llama al procedimiento `crea_sec_inversion`: crea la secuencia para la "empresa actual" (ver debajo la descripción)
3. Llama al procedimiento `crea_tabla_inversiones` para la empresa actual. (ver debajo la descripción)
4. Crear la fila actual en la tabla `inversiones_XXXXXX` con los datos pasados por parámetro y la secuencia `sec_XXXXXX`
5. Después se imprime por consola para ver el resultado en las pruebas.

PRUEBAS: crea un bloque sin nombre para probarlo, que tenga varias llamadas al procedimiento, una para cada situación posible.

Procedimiento crea_sec_inversion

- Comprueba que no esté creada la secuencia `sec_XXXXXX` (con la Tabla “user_objects”). Si está creada, termina sin hacer nada.
- Crea una secuencia `sec_XXXXXX` para esa empresa `XXXXXX`. Se usa para dar un valor único a cada fila en la tabla inversiones de esa empresa.

Procedimiento crea_tabla_inversiones

- Comprueba que no esté creada la tabla `inversiones_XXXXXX` (con la Tabla “tabs”). Si está creada termina sin hacer nada.
- Crea la tabla `inversiones_XXXXXX`.
- Atributos de la tabla: los mismos que la tabla `Invierte`, quitando el nombre de Empresa y añadiendo el atributo `NumSec`, que es el número único generado con la `sec_XXXXXX`

Procedimiento Partir Tabla Invierte

- Leer cada fila de la tabla original `Invierte` y, llamando a los procedimientos creados ir generando cada fila en su tabla `inversiones_XXXXXX` correspondiente, es decir que `XXXXXX` sea su empresa.