

# AMPLIACION de BASES DE DATOS

( Profesor : Héctor Gómez Gauchía)

## Práctica 2 Aptdo 1: DESNORMALIZACION con Procedimientos de PLSQL dinámico

- **Modo de entrega:** No se Entrega

**(Se rellena el CUESTIONARIO de la práctica cuando esté disponible en el CampusV)**

- 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.

- **RESPUESTAS:** En un fichero nuevo word, copia el enunciado de cada apartado en él. Después incluye cada respuesta en el apartado correspondiente. **También:**

- Incluye las filas que están implicadas en cada apartado, ya sea porque estaban provocando un problema o porque las hayas insertado porque te lo pedía.
- Incluye el contenido de los ficheros .sql de los apartados con PLSQL o con instrucciones sql.
- Para organizar bien el fichero, puedes insertar saltos de página.
- Siempre que trabajes, haz [Lista de Dudas concretas](#) para consultar con el profesor, online o por email o en clase o en el laboratorio.

→ **Conservar Respuestas para ponerlas en el Cuestionario**

### **MATERIALES para consultar al hacer la práctica**

- **VER PISTAS:** en la carpeta de esta práctica: / prac2enu-Aptdo-1-PISTAS-CV.pdf
- Teoría: Tema PL/SQL / Desarrollo-PL-SQL.pdf
- Las operaciones que se piden están en Ejemplos de PLSQL Dinámico : ver en Tema PL/SQL / Ejercicios . . . / PLSQL-ejemplos/ PLSQL-dinamico-para-CV.zip

### **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 cuyo nombre es XXXXXX*, dado 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 debajo.

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`.

**SE PIDE HACER ESTOS PROCEDIMIENTOS y PRUEBAS: (debes crear exactamente estos elementos en el orden indicado con los pasos indicados y generar filas adecuadas para probar todos los casos posibles razonables).**

ANTES: copia la tabla `Invierte` con los datos originales que ejecutes `prueba_gestion`, la necesitarás para más adelante a otra con otro nombre y si hay errores.

#### **1º.- Procedimiento crea sec inversion(nombreEmpresa Invierte.NombreE%TYPE)**

- 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.

### 2º.- Procedimiento crea\_tabla\_inversiones(nombreEmpresa Invierte.NombreE%TYPE)

- 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 `NumInv`, la PK, que es el número único secuencial generado con la `sec_XXXXXX`

### 3º.- Procedimiento Gestion\_Inversion (parámetros: atributos de la inversión “actual”, los mismos que en tabla Invierte).

PASOS dentro del procedimiento:

1. Comprueba que existe la empresa `XXXXXX` (la “actual”) en la tabla `Empresa` de la BDejemplo
2. Llama al procedimiento `crea_sec_inversion`: crea la secuencia para la empresa “actual”
3. Llama al procedimiento `crea_tabla_inversiones` para la empresa “actual”.
4. Crear la fila actual en la tabla `inversiones_XXXXXX` con los datos pasados por parámetro y para el atributo `NumInv` se asigna el valor de la secuencia `sec_XXXXXX` incrementada en una unidad.
5. Después se imprime por consola para ver el resultado en las pruebas.

**3º.-bis PRUEBAS:** crea un procedimiento `prueba_gestion` para probarlo, que tenga varias llamadas al `Gestion_Inversion`, una para cada situación posible (como hicimos en la práctica anterior) y su mensaje definiendo el caso que se prueba.

→ ANTES: Necesitas crear en la hoja de trabajo filas con insert directamente en la tabla `Invierte` para tener suficientes filas para cubrir todos los casos.

### 4º.- Procedimiento Partir\_Tabla\_Invierte(sin parámetros)

- Leer cada fila de la tabla original `Invierte` y, llamando a `Gestion_Inversion`, va generando cada fila en su tabla `inversiones_XXXXXX` correspondiente, donde `XXXXXX` es el valor del atributo `NOMBREE` de esa fila que es la empresa.
- (Cuando el paso anterior esté Probado y funcione bien) Borra el contenido de la tabla original `Invierte`, sin borrar la tabla.

### 5º.- (para NOTA) Procedimiento Suma\_inversiones(DNICliente)

- Para el cliente dado, obtener el total de dinero invertido sumando todas las inversiones de todas las tablas `inversiones_XXXXXX` de ese cliente. Mostrar una línea con el total que tiene invertido en cada empresa (`XXXXXX`) y el total de todas juntas.