

PRÁCTICA 8
TRANSPARENCIA DE DISTRIBUCIÓN PARA INSTRUCCIONES DML

Notas:

- Para todas las actividades de esta práctica, considerar el esquema de fragmentación de la práctica anterior.
- La práctica puede realizarse en equipo de 2 personas. Para este caso cada integrante deberá realizar todas las actividades en sus respectivos ambientes.

1.1. OBJETIVO

Comprender y poner en práctica estrategias para implementar los niveles de transparencia de distribución y transaccionalidad en una base de datos distribuida para operaciones DML.

1.2. ACTIVIDADES PREVIAS.

Revisar el documento practica8-previo.pdf

1.3. IMPLEMENTACIÓN DE TRANSPARENCIA DML Y TRANSACCIONALIDAD.

- Considerar nuevamente el esquema de fragmentación y la solución de fragmentación para la instrucción `select` realizadas en prácticas anteriores.
- En esta ocasión se realizará la transparencia de operaciones solo para los siguientes escenarios:
 - Transparencia para operaciones `insert` y `delete` solo para las tablas `país`, `oficina` y `empleado`. Para operaciones `update` el trigger deberá lanzar una excepción indicando que la operación `update` aún no está soportada.
 - Transparencia para operaciones `insert`, `delete` y `update` únicamente para la tabla `empleado`.

1.3.1. Privilegios para crear triggers

- Crear Scripts SQL `s-01-<iniciales>-privilegios-usuarios.sql` **C1. Incluir en el reporte únicamente** el contenido del archivo.
- En esta ocasión se requieren permisos en ambas PDBs para que el usuario correspondiente pueda crear triggers.
- El script se deberá conectar a cada PDB y como usuario SYS otorgar el privilegio correspondiente.

1.3.2. Definición de tablas temporales para manejo de columnas BLOB.

- Crear Scripts SQL `s-02-<iniciales>-tablas-temporales.sql` **C2. Incluir en el reporte únicamente** el contenido del archivo.
- El script deberá definir 2 tablas temporales: `t_empleado_insert` y `t_empleado_update` que serán empleadas por el trigger que implementará transparencia DML.
- El script deberá conectarse únicamente a las PDBs que requieran hacer accesos remotos para obtener un objeto BLOB.

1.3.3. Definición del trigger Instead Of para la vista PAIS.

- Crear Scripts SQL `s-03-<iniciales>-pais-[n<{1|2}>]-trigger.sql` Notar que el término `[n<{1|2}>]` se puede omitir ya que el código de este trigger es idéntico en ambos nodos. Solo se requiere un script por integrante. El nombre del archivo será `s-03-<iniciales>-pais-trigger.sql` **C3. Incluir en el reporte** el contenido de todos los scripts

1.3.4. Definición del trigger Instead Of para la vista OFICINA.

- Crear Scripts SQL `s-03-<iniciales>-oficina-n<{1|2}>-trigger.sql` Observar que se requiere un archivo para cada PDB ya que el código del trigger es diferente al tratarse de una fragmentación horizontal derivada. **C4. Incluir en el reporte** el contenido de todos los scripts.

1.3.5. Definición del trigger instead of para la vista EMPLEADO.

- Crear scripts `s-03-<iniciales>-empleado-n<{1|2}>-trigger.sql` Observar que en este caso si se requiere un trigger diferente por cada PDF debido al manejo del dato BLOB. **C5. Incluir en el reporte** el contenido de todos los scripts.

1.3.6. Ejecución de triggers.

- Crear Scripts SQL s-04-<iniciales>-main-triggers.sql El script se deberá conectar a cada PDB y ejecutar la compilación y creación de los triggers. **C6. Incluir en el reporte** únicamente el contenido del archivo.

Ejemplo:

```
--@Autor:          Jorge A. Rodriguez C
--@Fecha creación: dd/mm/yyyy
--@Descripción:    Creación de triggers
Prompt creando triggers en S1
connect consultora_bdd/consultora_bdd@jrcbd_s1

Prompt creando trigger para pais
@s-03-jrc-pais-trigger.sql
show errors

Prompt creando trigger para oficina
@s-03-jrc-n1-oficina-trigger.sql
show errors

Prompt creando trigger para empleado
@s-03-jrc-n1-empleado-trigger.sql
show errors

Prompt creando triggers en S2
connect consultora_bdd/consultora_bdd@jrcbd_s2

Prompt creando trigger para pais
@s-03-jrc-pais-trigger.sql
show errors

Prompt creando trigger para oficina
@s-03-jrc-n2-oficina-trigger.sql
show errors

Prompt creando trigger para empleado
@s-03-jrc-n2-empleado-trigger.sql
show errors

Prompt Listo!
exit
```

- Observar que se usa la instrucción `show errors` para mostrar posibles errores de compilación.

1.4. VALIDACIÓN DE RESULTADOS.

- Antes de iniciar con la validación de resultados, asegurarse de crear la función `load_blob_from_file` que se menciona en el previo ya que será empleada en esta sección. Para ello, realizar las siguientes acciones:
 - Crear un script s-05-main-funciones.sql El script deberá conectarse a cada PDB e invocar a un script llamado s-05-load-blob-from-file.sql el cual contendrá la definición de la función.

1.4.1. Preparar scripts de prueba.

Para verificar el correcto funcionamiento del concepto de transparencia de operaciones DML y transaccionalidad, realizar las siguientes acciones:

- De la carpeta compartida BDD obtener los siguientes archivos y carpetas:
 - s-06-main-operaciones-dml.sql
 - s-06-operaciones-dml.sql
 - carpeta fotos (Contiene imágenes de prueba).
- Copiar tanto los archivos como la carpeta `fotos` al mismo directorio donde se encuentran todos los scripts SQL de la práctica.
- Observar que el archivo s-06-main-operaciones-dml.sql se conecta a cada PDB y ejecuta el segundo Script.
 - Antes de ejecutar el archivo, abrirlo y configurar las siguientes variables según corresponda:

```
-----Configuracion de variables, modificar segun corresponda
set verify off
define pdb_1 = 'jrcbd_s1'
define pdb_2 = 'jrcbd_s2'
define usuario = 'consultora_bdd'

--evitar estas 2 lineas en sistemas reales.
define sys_pwd='system'
define usuario_pwd = 'consultora_bdd'
```

- Revisar cuidadosamente el segundo script. Observar que realizan diversas operaciones DML en cada PDB como si se tratara de una BD centralizada.
- Observar también, el manejo transaccional distribuido que se hace empleando las instrucciones `commit`, `rollback`.
- Ejecutar el script, verificar que no se produzcan errores. **NO modificar** el script `s-06-operaciones-dml.sql` **C6. Incluir en el reporte únicamente** el resultado de la ejecución del script.

1.5. CONTENIDO DEL REPORTE.

- Introducción
- Objetivo
- Desarrollo de la práctica:
 - C1. Script `s-01-<iniciales>-privilegios-usuarios.sql`. Solo contenido.
 - C2. Script SQL `s-02-<iniciales>-tablas-temporales.sql` Solo contenido.
 - C3. Script `s-03-<iniciales>-revista-n<{1|2}>-trigger.sql` Solo contenido.
 - C4. Script `s-03-<iniciales>-suscriptor-n<{1|2}>-trigger.sql` Solo contenido.
 - C5. Script `s-04-<iniciales>-main-triggers.sql` Contenido y resultado de ejecución.
 - C6. Script `s-06-main-operaciones-dml.sql`. Únicamente el resultado de su ejecución.
- Opcionalmente se pueden incluir algunas pantallas.
- Conclusiones (una sola por equipo), comentarios, recomendaciones.
- Bibliografía.