



Temario "B" → Prueba Practica

- 1- Considere que T_0 sea una transacción que transfiere $\$ 50.00$ desde la cuenta "A" a la cuenta "B" y T_1 sea una transacción que retira $\$ 100.00$ de la cuenta "C".

T_0 : leer(A)

$A := A - 50$

escribir(A)

leer(B)

$B := B + 50$

escribir(B)

T_1 : leer(C)

$C := C - 100$

escribir(C)

Assumimos que las transacciones se ejecutan secuencialmente, primero T_0 y después T_1 y que los saldos de A, B y C antes de la ejecución eran de $\$ 1000.00$, $\$ 2000.00$ y $\$ 700.00$. Supongamos que el sistema sufre una caída antes que se completen las transacciones, Indique el método de recuperación en cada uno de los casos que se presentan utilizando el esquema de Actualización diferida de Registro.

a) $\langle T_0 \text{ Iniciado} \rangle$
 $\langle T_0, A, 950 \rangle$
 $\langle T_0, B, 2050 \rangle$

b) $\langle T_0 \text{ Iniciada} \rangle$
 $\langle T_0, A, 950 \rangle$
 $\langle T_0, B, 2050 \rangle$
 $\langle T_0 \text{ Comprometida} \rangle$
 $\langle T_1 \text{ Iniciada} \rangle$
 $\langle T_1, C, 600 \rangle$



- c) $\langle T_0 \text{ Iniciada} \rangle$
 $\langle T_0, A, 950 \rangle$
 $\langle T_0, B, 2050 \rangle$
 $\langle T_0, \text{Comprometida} \rangle$
 $\langle T_1 \text{ Iniciada} \rangle$
 $\langle T_1, C, 600 \rangle$
 $\langle T_1 \text{ Comprometida} \rangle$

- a) **R//** En el primer caso presentado no es necesario realizar ninguna acción para rehacer la misma debido que no existe ningún commit en el momento en el que la base de datos sufre la caída por lo que no existió ningún write.
- b) **R//** En este segundo caso se debe ejecutar un **redo(T_0)** es decir se debe de rehacer la transacción T_0 esto debido a que existe un commit y es posible que se halla realizado un write a nivel de página sin afectar la parte física en cuanto a la transacción T_1 no es necesario realizar ninguna acción ya que no existe un commit.
- c) **R//** En el último caso es necesario realizar un rehacer secuencialmente debido a que ambas transacciones cuentan con un commit, el orden sería **redo(T_0)** y seguidamente se deberá realizar un **redo(T_1)**



Problema 2) → Defina los comandos a utilizar para las siguientes operaciones.

- 1- Conectarse a la BDD con el usuario **SYSTEM**
 - * Ingresar a SQL Command Line o SQL Plus.
 - * Ingresar el comando: **connect system@XE;**
 - * Ingresar la contraseña: ********
- 2- Creación de usuario **progra01**, asignando una contraseña.
 - * **Create user progra01 identified by progra01**
default tablespace **USERS**
temporary tablespace **TEMP**
QUOTA unlimited on USERS;
- 3- Crear un rol **desarrollo** con privilegios para: iniciar sesión, crear usuarios, crear y eliminar tablas.
 - * **Create role desarrollo;**
 - * **grant create session to desarrollo;**
 - * **grant create user to desarrollo;**
 - * **grant create table to desarrollo;**
 - * **grant delete any table to desarrollo;**
- 4- Asignar al usuario **progra01** el rol **desarrollo**.
 - * **grant desarrollo to progra01;**
- 5- Conectarse a la BDD con el usuario **progra01** y crear la tabla **Estudiante** con los atributos **codigo <integer>** y **nombre <varchar2 de 50 posiciones>**, luego modificar la tabla para -



- agregar al final un atributo fechanacimiento <datetime>.
- * Desconectarse de **system** con el comando: **disconnect**.
 - * conectarse con el comando: **connect progra01@XE**
 - * colocar la contraseña: **progra01**
 - Crear la Tabla Estudiante
 - * **create table ESTUDIANTE (codigo **INTEGER** not null, nombre **VARCHAR2(50)** not null);**
 - **Alter table estudiante add fechanacimiento datetime not null;**
→ **ORA-00902: invalid datatype** Lanzo el siguiente error por lo que se cambio el tipo de dato por **date**.
 - * **Alter table Estudiante add fechanacimiento **DATE** not null;**
Se agrego el atributo correctamente.
- 6- Conectarse a BDD con el usuario **system** y agregar el permiso que haga falta al rol **desarrollo** para que funcionen correctamente las instrucciones anteriores para el usuario **progra01**.
- No fue necesario agregar ningun permiso para el paso No.5.
- 7- Elimine el privilegio de eliminar tablas al rol **desarrollo**.
- * **Revoke delete any table from desarrollo;**
- 8- Conceda el privilegio de crear vistas sobre el rol **desarrollo**, de tal manera que cualquier usuario que pertenezca a este rol pueda conceder permiso a otros que el haya creado.
- * **Grant create view to desarrollo with admin option;**



9- Conectarse a la BDD con el usuario **progra01** y crear un usuario **progra02**

* Create user **progra02** identified by **progra02**
default tablespace **USERS**
temporary tablespace **TEMP**
QUOTA unlimited on **USERS**;

- Si fue posible crear el usuario debido a que **progra01** tiene el rol de desarrollo el cual tiene el permiso de crear usuarios, el cual se agrego en el punto 3.

10- Eliminar el usuario **progra01**.

- Para eliminar el usuario es necesario acceder con system ya que no se dieron permisos para eliminar usuarios.
Drop user progra01 cascade;



Problema 3)

Datos Iniciales

$B = 512$ bytes

$r = 30,000$ registros (tabla)

$R = 115$ bytes

$$R = 30 + 9 + 9 + 40 + 9 + 8 + 1 + 4 + 4 + 1 = \underline{\underline{115 \text{ bytes}}}$$

- 1 → Determinar el factor de bloqueo para el archivo (bfr): $\text{floor}(B/R)$
 $bfr = \text{floor}(B/R) \Rightarrow \text{floor}(512/115) \Rightarrow 4$

$$R // bfr = 4 \text{ registros / bloque}$$

- 2 → Determinar el número de bloques necesarios para almacenar el archivo (b): $\text{ceiling}(r/bfr)$
 $b = \text{ceiling}(r/bfr) \Rightarrow \text{ceiling}(30,000/4) \Rightarrow 7,500$

$$R // b = 7,500 \text{ bloques}$$

Aproximación con búsqueda binaria

$$\text{ceiling}(\log_2(b)) \Rightarrow \text{ceiling}(\log_2(7,500)) \Rightarrow 13 \text{ operaciones E/S}$$

- 3 → Supongamos que el archivo está ordenado físicamente según el campo clave NSS y que se desea construir un índice primario

- 4 → Calcular el tamaño del registro que formará parte del índice

$$R_i = NSS + \text{Puntero} \Rightarrow 9 + 6 \Rightarrow \underline{\underline{15 \text{ bytes}}}$$



- 4 → Calcular el factor de bloque del índice
 $bfri = \text{floor}(B/R_i) \Rightarrow \text{floor}(512/15) \Rightarrow \underline{34 \text{ reg. índice por bloque}}$
- 5 → Determine el número de bloques del índice
 $bi = \text{ceiling}(ri/bfri) \Rightarrow \text{ceiling}(7500/34) \Rightarrow \underline{221 \text{ bloques}}$
 $ri = b = \underline{7500}$
- 6 → Cuantos accesos son necesarios para una búsqueda binaria en el índice
 $\text{ceiling}(\log_2(bi)) \Rightarrow \text{ceiling}(\log_2(221)) \Rightarrow \underline{8 \text{ operaciones E/S}}$
- 7 → Cual es el número total de accesos a bloques de disco para localizar un registro en el archivo de datos a través del índice, tomando en cuenta su respuesta al inciso anterior.
R// 8 operaciones E/S + 1 \Rightarrow 9 operaciones E/S

En este caso buscar por medio del índice solo necesita 9 operaciones de entrada y salida una mejora en cuanto a las 13 operaciones que se necesitarían sin el índice