Mecanismos de Recuperación

Índice

- Aspectos generales sobre recuperación
- Tipos de fallos
- Fallos con pérdida de memoria volátil
  - Actualización inmediata
  - Actualización diferida
- Fallos con pérdida de memoria estable
- Mecanismos de recuperación en ORACLE

Bibliografía

- ***Fundamentals of Database Systems (4.edición 2004)***
  Fundamentos de Sistemas de Bases de Datos (3. Edición 2002)
  R.A. Elmasri, S. B. Navathe
  Addison Wesley 2002

- ***Fundamentos de Bases de Datos (4. edición)***
  A. Silberschatz, H. F. Korth, S. Sudarshan
  Mc. Graw Hill 2002

- ***Database System Implementation***
  H. García Molina, J.D. Ullman, J. Widom
  Prentice Hall 2000

Propiedades de la Transacción

Principio ACID (su cumplimiento debe estar asegurado por el SGBD***)***

- Se ejecuta como unidad (*Atomicity*) Gestor de transacciones, Gestor de recuperación
- Preserva la consistencia(*Consistency*) Gestor de Rest. de integridad
- Una transacción no muestra los cambios que produce hasta que finaliza (Isolation) Gestor de Control de Concurrencia
- Si termina correctamente, sus cambios permanecen (*Durability*) Gestor de Recuperaciones

Aspectos generales sobre recuperación

- Los sistemas de Bases de Datos deben **asegurar la disponibilidad** de los datos a aquellos usuarios que tienen derecho a ello por lo que proporcionan mecanismos que permiten recuperar la BD contra fallos lógicos o físicos que destruyen los datos en todo o en parte

Aspectos generales sobre recuperación

- Mecanismo de recuperación: responsable de la *restauración* de la BD al estado consistente previo al fallo.

  También debe proporcionar *alta disponibilidad*, esto es, debe minimizar el tiempo durante el que la BD no se puede usar después de un fallo.

Aspectos generales sobre recuperación

El principio básico en el que se apoya la recuperación de la Base de Datos es la

"Redundancia Física"

En muchos casos los procesos de recuperación y de control de concurrencia están interrelacionados. En general, cuanto mayor sea el grado de concurrencia que deseemos alcanzar, mayor tiempo consumirá la tarea de recuperación.

Aspectos generales sobre recuperación

- Para fines de recuperación el sistema necesita mantenerse al tanto de cuando la transacción se inicia, termina y se confirma o aborta.

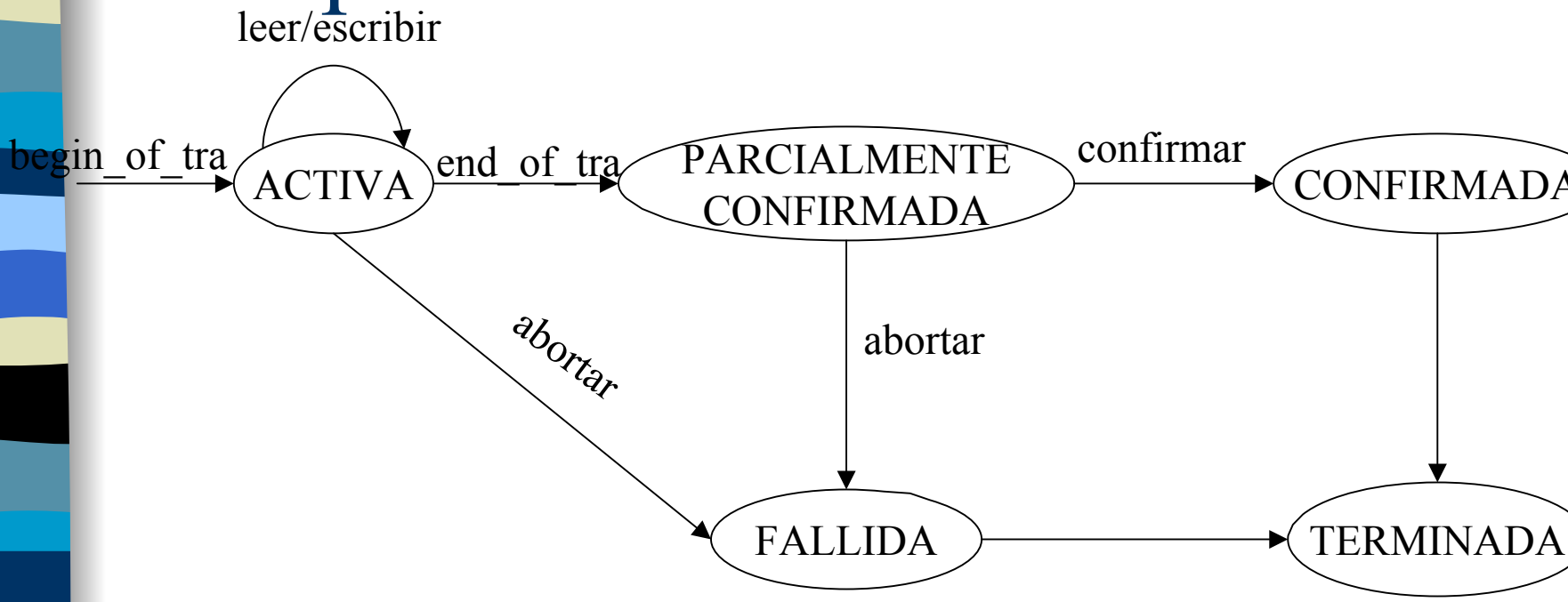
- El gestor de recuperación se mantiene al tanto de las siguientes operaciones (esta información se almacena en el diario):

  BEGIN_TRANSACTION

  READ O WRITE

  END_TRANSACTION
  COMMIT_TRANSACTION
  ROLLBACK O ABORT

Aspectos generales sobre recuperación



- DTS (Diagrama de Transición de Estados) para la ejecución de transacciones

Parcialmente Confirmada:
el SGBD verifica que no hay interferencias dañinas con otras transacciones.

Tipos de Fallos

- Fallo del ordenador (caída del sistema)
- Error de la transacción (ej. Overflow, violación restricción)
- Errores de los usuarios (ej. el usuario borra accidentalmente una tabla)
- Imposición de control de concurrencia (ej. estado de bloqueo mortal)
- Fallo disco
- Catástrofes físicas (Ej. inundación)

Fallos

- Los fallos pueden afectar a las transacciones en sus propiedades ACID. Deben existir algoritmos que garanticen la consistencia de la BD y la atomicidad de las transacciones a pesar de los fallos.
- Solución:
  - Mecanismos de control de concurrencia
  - Mecanismos de recuperación

Fallos referentes al SGBD

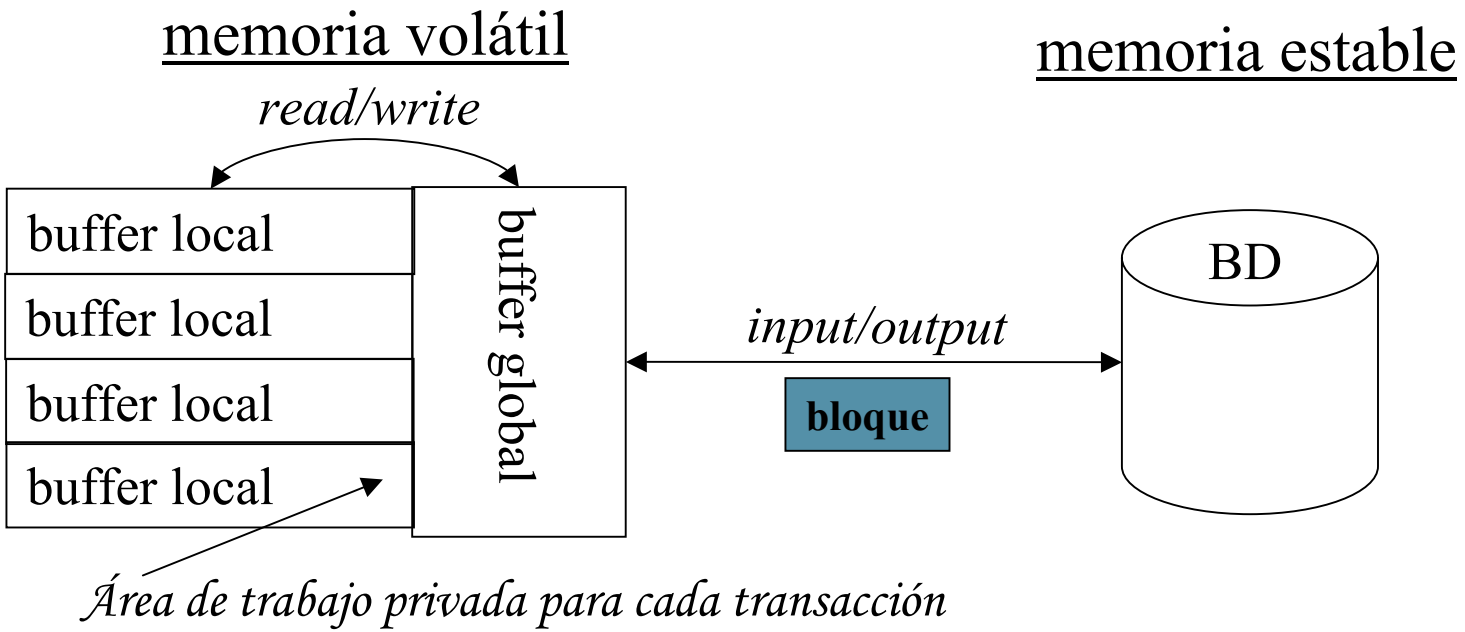
- Dos tipos importantes:
  - Los que provocan la pérdida del contenido de la *memoria estable (discos)*
  - Los que provocan la perdida de la *memoria volátil (memoria principal)*, debidos a interrupción de suministro eléctrico o por funcionamiento anormal del hardware

Estrategia de Recuperación Típica

- Si hay daños en una porción de la BD (ej. debido a un fallo del disco) el método de recuperación:
  - restaurará una copia anterior de la BD (que puede estar en cinta) y
  - reconstruirá un estado más actual, volviendo a aplicar operaciones almacenadas en el diario.

- Cuando la BD no presenta daños físicos pero se ha vuelto inconsistente, la estrategia consiste en
  - invertir los cambios que provocaron la inconsistencia. Se trabaja con el diario, no se necesita una copia archivada.

Operaciones básicas



- Las operaciones básicas de acceso a una BD que una transacción puede incluir son:
  - leer-elemento (*READ*)
  - escribir-elemento (*WRITE*)
  - *INPUT*
  - *OUTPUT*

Operaciones básicas

Nivel Transacciones

READ(x) -- SELECT

- buscar si X está en el buffer global
- si no, ocasionar un *input* para traer el bloque que contiene a X
- copiar X del buffer global al buffer local

WRITE(x) -- INSERT, UPDATE, DELETE

- copiar X del buffer local al buffer global
- SGBD transfiere el bloque actualizado desde el buffer al disco (en algún momento)

Nivel Gestor de Buffer

INPUT

- Copia el bloque que contiene el elemento X en el buffer global

OUTPUT

- Copia el bloque que contiene a X al disco

El Diario o Bitácora

- Objetivo: recuperar fallos con pérdida
- Fichero gestionado por el SGBD
- Entradas:
  - [start_transaction, T] ← identificador de la transacción
  - [commit, T]
  - [abort, T]
  - [read_item, T, dato, <valorLeido>]
  - [write_item, T, dato, <valorAntigüo>, <valorNuevo>]
- Operaciones:
  - REDO: se escriben los <valorNuevo> en la BD
  - UNDO: se repone los <valorAntigüo> en la BD

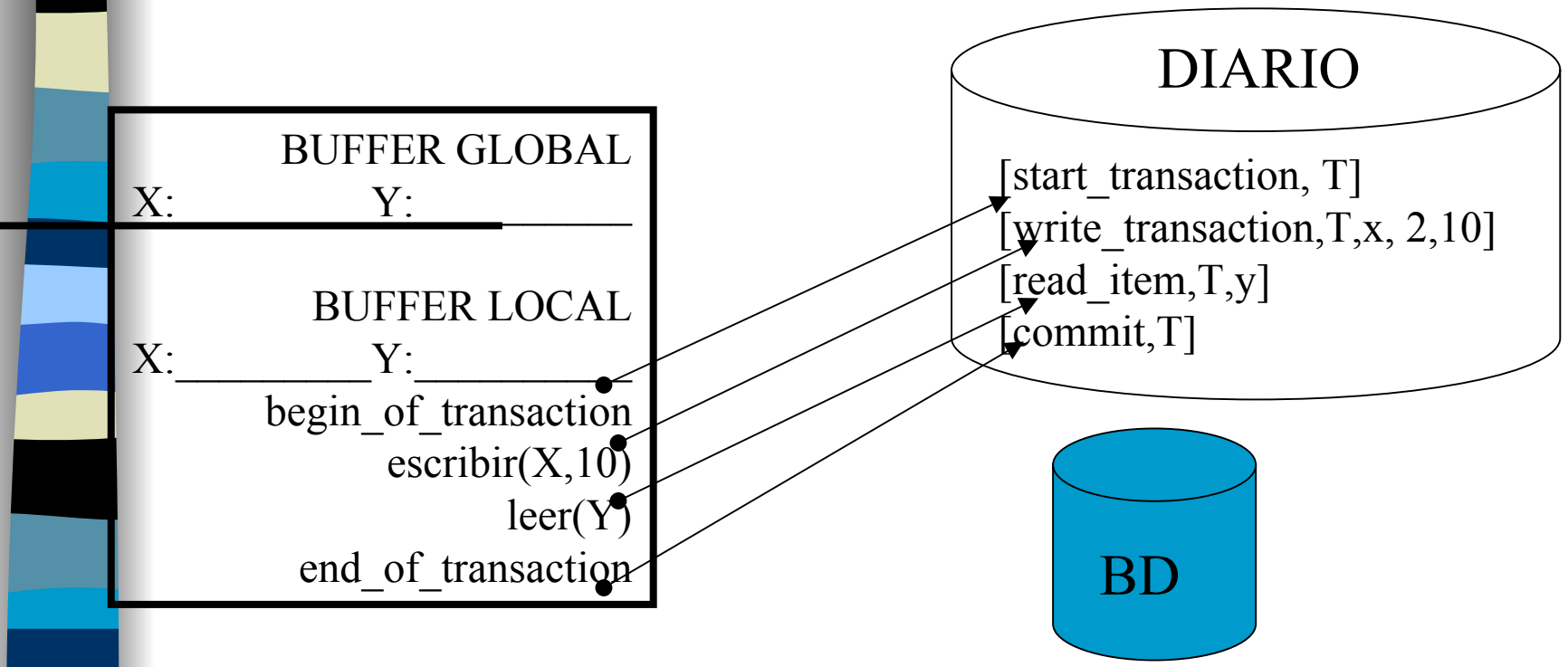
Fichero Diario

- Puede surgir un problema en caso de que se realice un cambio en la BD y no en el fichero diario; por ello normalmente se obliga a que los registros que se modifican se escriban antes en el fichero diario que en la BD, para poder anular así, en caso de necesidad, las transacciones (*log write-ahead protocol*)

Fallos con pérdida de memoria volátil

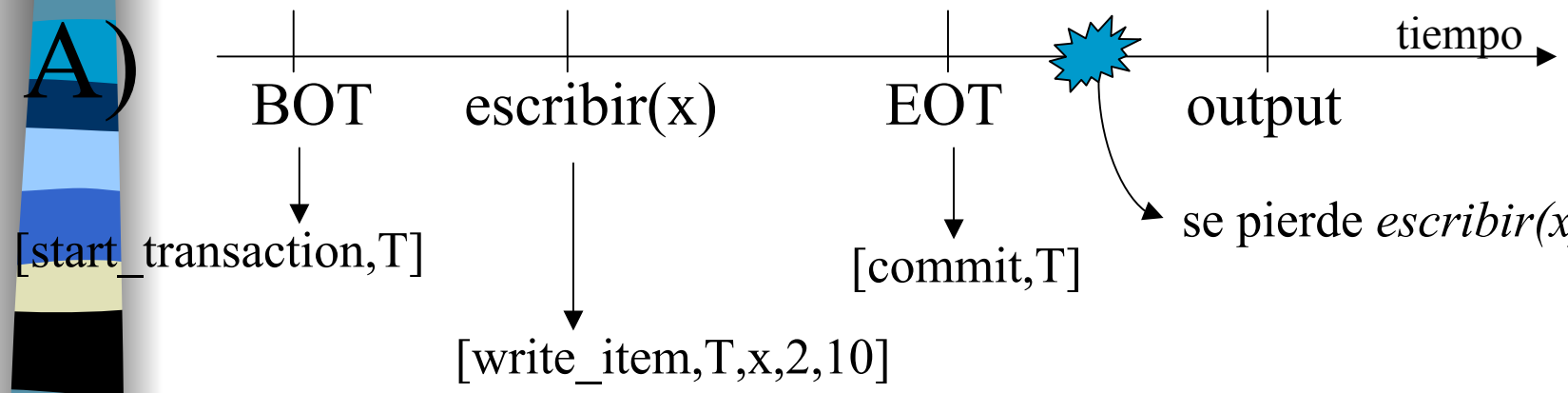
- Actualización inmediata
  - *escribir (X)* se hace directamente en el Buffer Global.
- Actualización diferida
  - *escribir (X)* se hace sobre el buffer local. Sólo al terminar la transacción se vuelca sobre el buffer global.

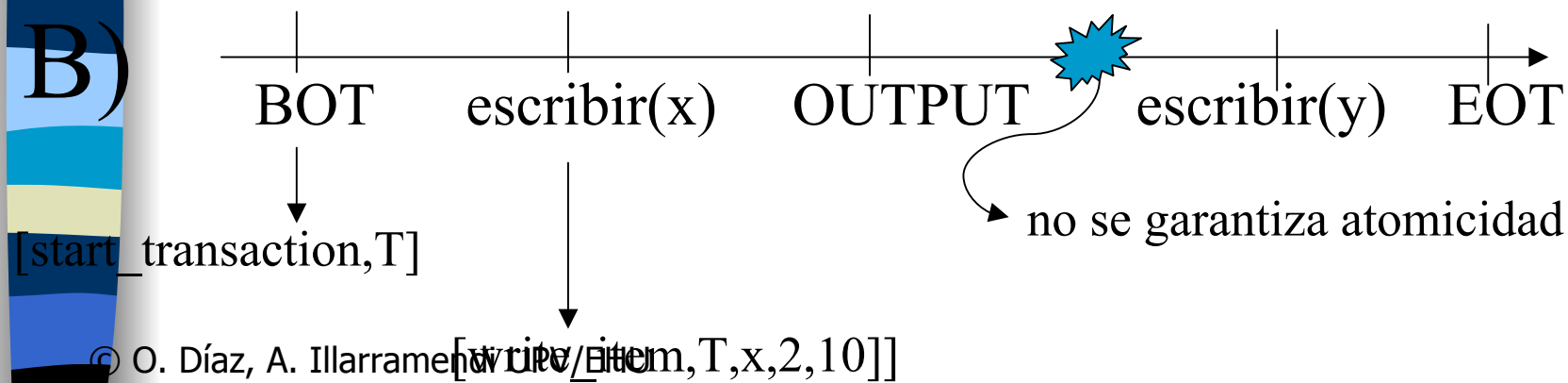
Actualización inmediata



- Actualizaciones inmediatas: *escribir (X)* se hace directamente en el Buffer Global.
- Al hacer el output de los registros del diario, T pasa a parcialmente confirmada
- Ante un fallo con pérdida, si la entrada *[commit,T]*
  - (caso A). aparece en el diario, entonces *REDO(T)*
  - (caso B). no aparece en el diario, entonces *UNDO(T)*

Error con actualizaciones inmediatas

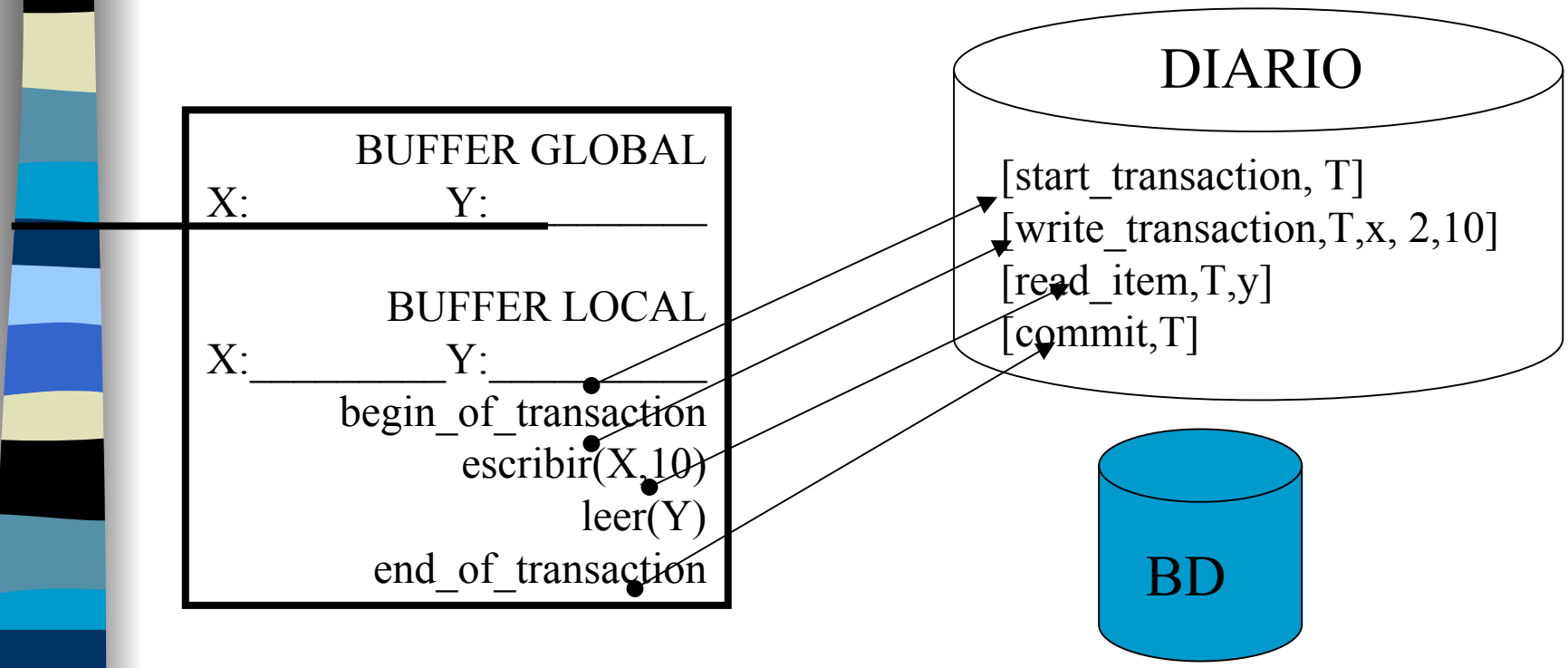




Error con actualizaciones inmediatas

- Se pueden producir “rollbacks” en cascada lo que implica un costo en proceso y en tiempo

Actualización Diferida



•Actualizaciones diferidas: escribir (X) se hace sobre el buffer local. Sólo al terminar la transacción se vuelca sobre el buffer global.

•Al hacer el output de los registros del diario, T pasa a parcialmente confirmada.

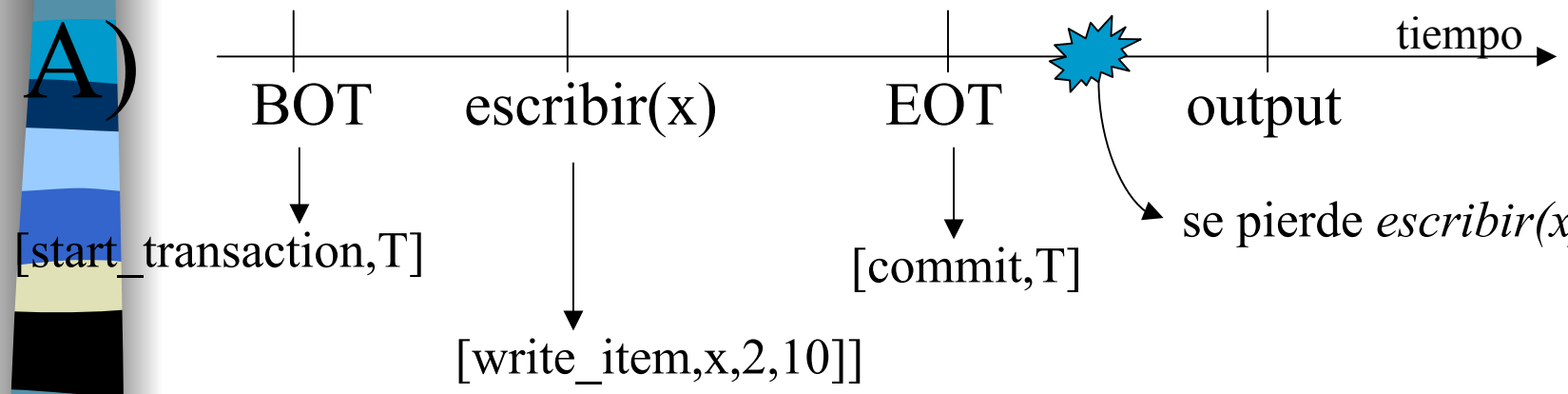
•Ante un fallo con pérdida, si la entrada *[commit,T]:*

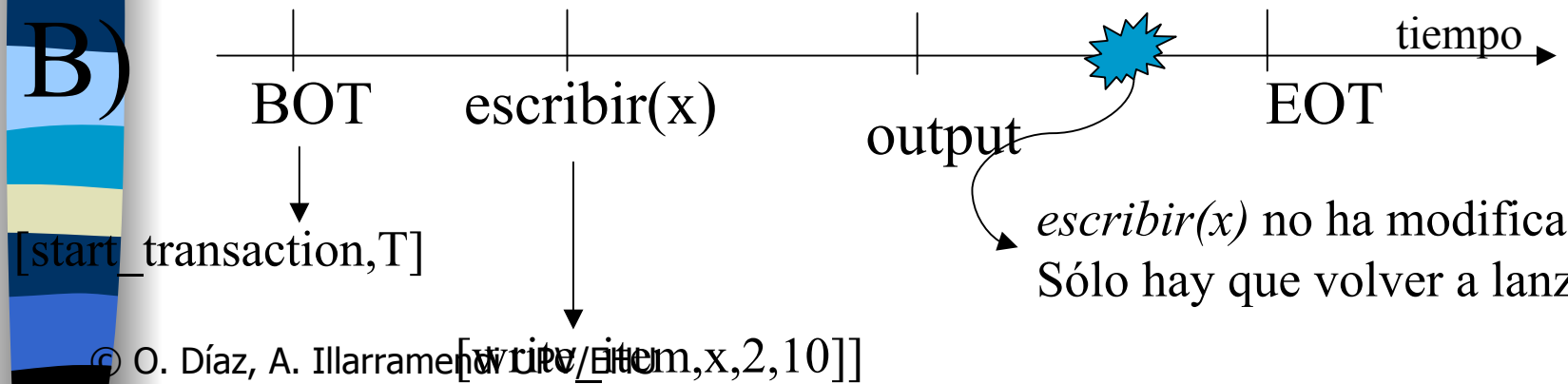
- (caso A). aparece en el diario, entonces *REDO(T)*
- (caso B). no aparece en el diario, entonces *"ignorar y volver a lanzar T"*

Actualización Diferida

- No puede usarse en la práctica a menos que las transacciones sean cortas y cada transacción cambie pocos elementos.
- Para otro tipo de transacciones existe el potencial de agotamiento del espacio del buffer.
- No se realiza la operación UNDO

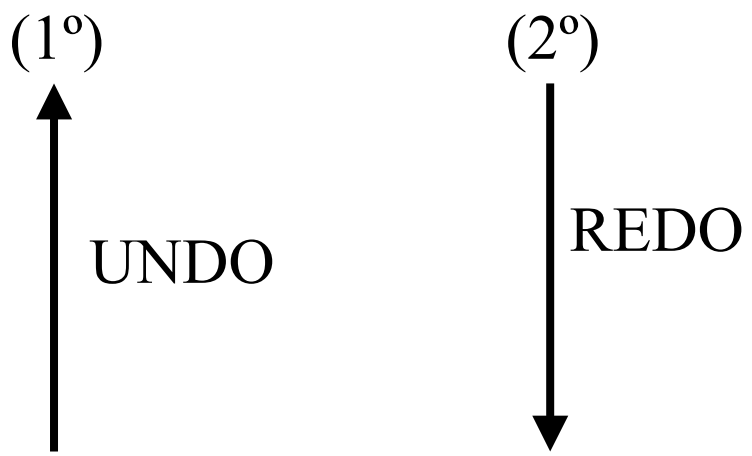
Error con actualizaciones diferidas





Proceso de recuperación

[start_transaction,T1]
.....
[commit,T1]
[start_transaction,T2]
.....
[start_transaction,T3]
.....
[commit,T3]
[start_transaction,T4]



Puntos de verificación/recuperación (*checkpoint*)

- Objetivo: evitar revisar todo el diario
- Nueva entrada en el diario: [checkpoint]
- ¿Cuándo? cada vez que se graba los datos (*output*)
- Mientras se lleva a cabo un checkpoint no se permite que ninguna transacción realice acciones de actualización.
- Proceso de recuperación
  - las transacciones con su [commit,T] antes del último [checkpoint] no hay que hacer REDO

	con [commit, T]	sin [commit, T]
antes de su [end ckpt]	ignorar T	rollback (undo + V.A.E)
después de su [end ckpt]	redo	V.A.E.*

Puntos de verificación

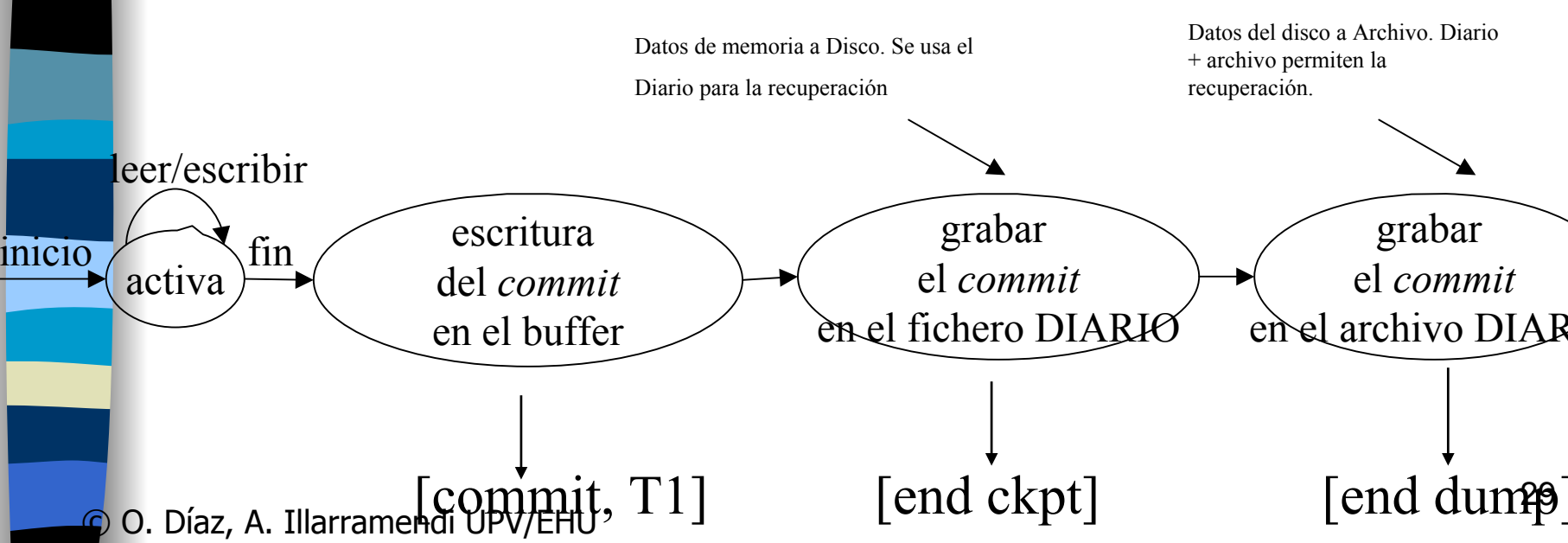
- Establecer un punto de verificación consiste en las siguientes acciones:
  - Suspensión de la ejecución de las transacciones temporalmente
  - Escritura forzada de todos los buffers de la memoria principal que han sido modificados a disco
  - Escribir un registro (*punto de verificación*) al diario y escritura forzada del diario a disco.
  - Reactivar las transacciones en ejecución

Puntos de Verificación

- El gestor de recuperación debe decidir en qué intervalos establecer un punto de control. El intervalo puede medirse en tiempo (ej. cada *m* minutos), o en un número *t* de transacciones confirmadas desde el último punto de control.

Proceso de confirmación

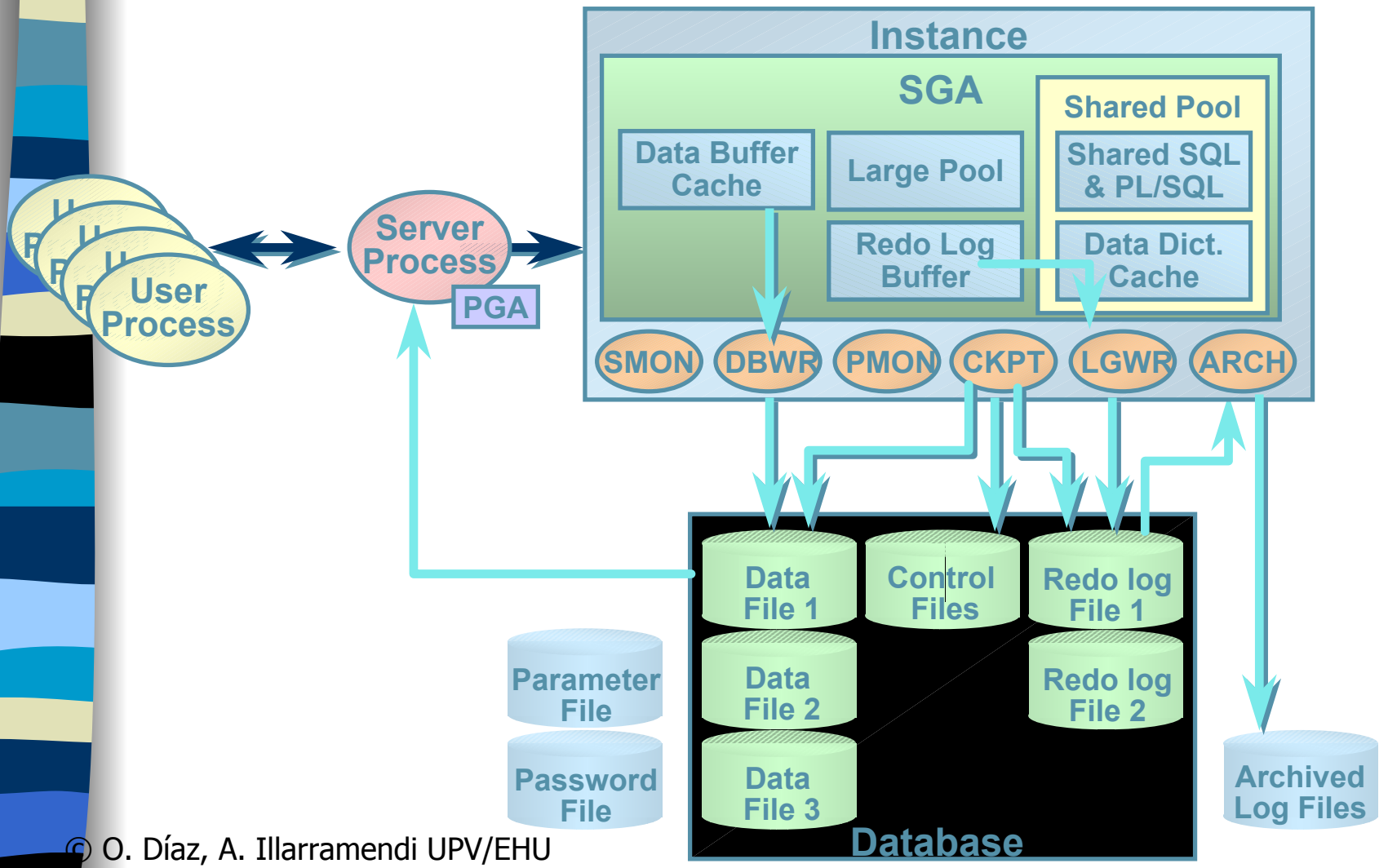
- Una trans. T esta confirmada si bajo ninguna circunstancia, se pueden perder los cambios de T
- Circunstancias adversas:
  - pérdida memoria volátil. Solución: diario
  - pérdida memoria estable. Solución: archivo



Estructura del SGBD Oracle

- Servidor Oracle: BD + instancia Oracle.
- Base de datos. Entre otros, contiene:
  - ficheros con datos (*data files*)
  - ficheros para recuperación (*redo log files*)
  - ficheros de control con información sobre la estructura física de base de datos, fecha de creación, etc (*control file*)
  - fichero de parametrización para iniciar Oracle (*parameter file*)
- Instancia Oracle:
  - cjto. de procesos que se ejecutan en background
  - zona de memoria global (*System Global Area, SGA*)
    - buffer de datos
    - buffer del diario
    - buffer compartido con planes de execución preguntas SQL
  - zona de memoria del proceso (*Program Global Area, PGA*)
  - *Los mecanismos de recuperación y backup de Oracle usan l elementos mencionados*.

Arquitectura Oracle



Instancia Oracle

- Se crea una instancia al lanzar el proceso de inicio de la BD, después de que se lee el fichero de parámetros.

- Siempre comienzan 5 procesos:
  - PMON,SMON,DBWR,LGWR,CKPT

Procesos de Background

- **DBWR**. Escribe datos que se encuentran en el “Data Buffer Cache” en los ficheros de datos.
- **LGWR**. Escribe datos que se encuentran en el “Redo Log Buffer” en los ficheros Redo.
- **SMON, PMON**. Determinan cuando es necesaria la recuperación y la activan.
- **CKPT**. Asegura que los contenidos de los buffers se escriben en los ficheros.
- **ARCH**. Archiva ficheros Redo.

Zona de Memoria

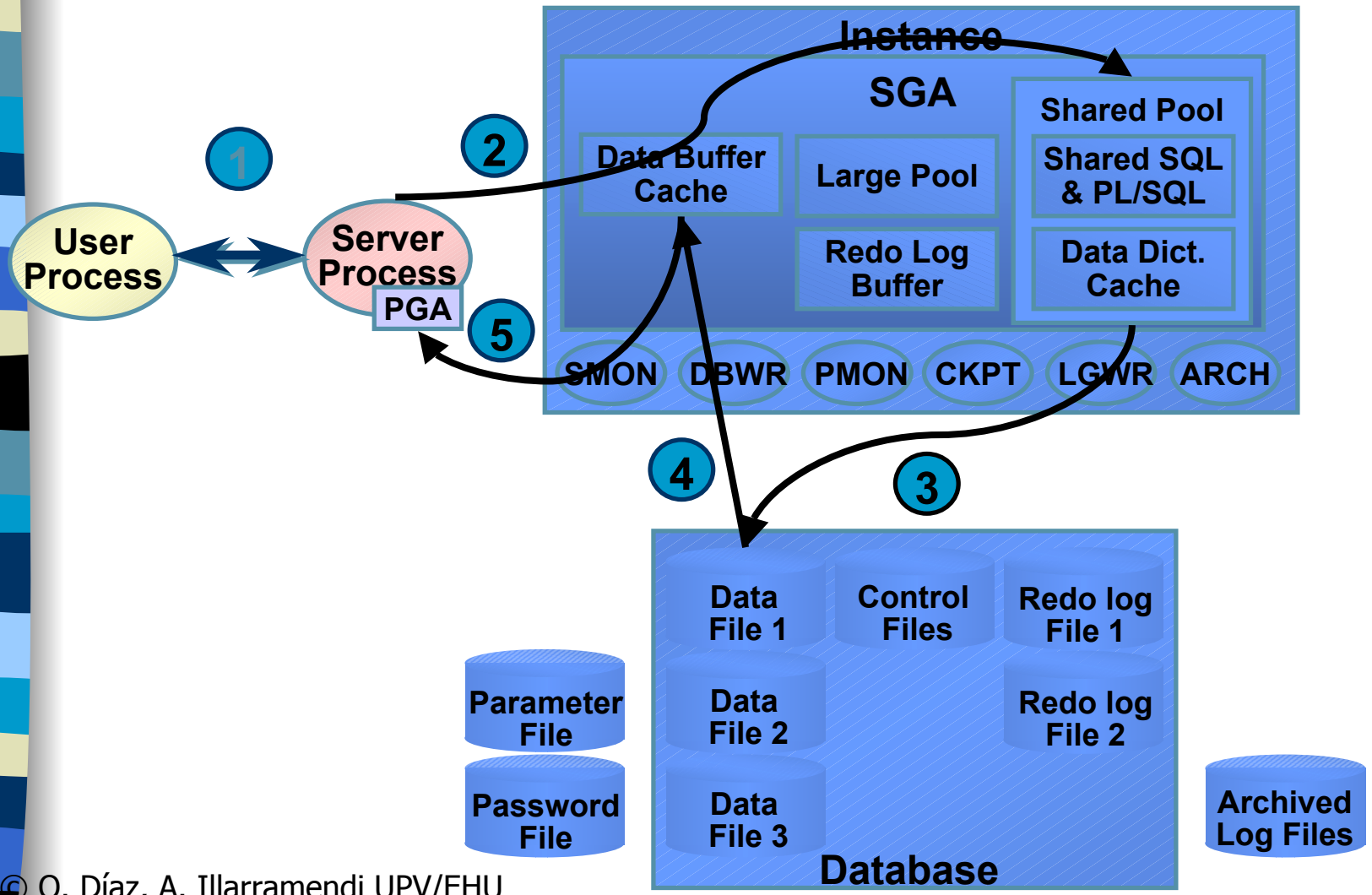
- **Data Buffer Cache**. Almacena datos traídos desde los ficheros de datos. Los datos se modifican en este buffer y después se escriben en los ficheros
- **Redo Log Buffer**. Contiene datos que se van a pasar a los ficheros redo. Es circular
- **Large Pool**. Memoria que se usa durante los procesos de restauración.
- **Shared Pool**. Almacena versiones compiladas de sentencias SQL y procedimientos.

Otros elementos

- **Ficheros de Control**. Almacenan la estructura física de la BD
- **Parameter File**. Almacena parámetros necesarios para inicializar una instancia (ej. tamaño de los buffers)
- **Password**. Almacena información sobre los usuarios que pueden recuperar la BD
- **Archived Log**. Copias de los ficheros logs.

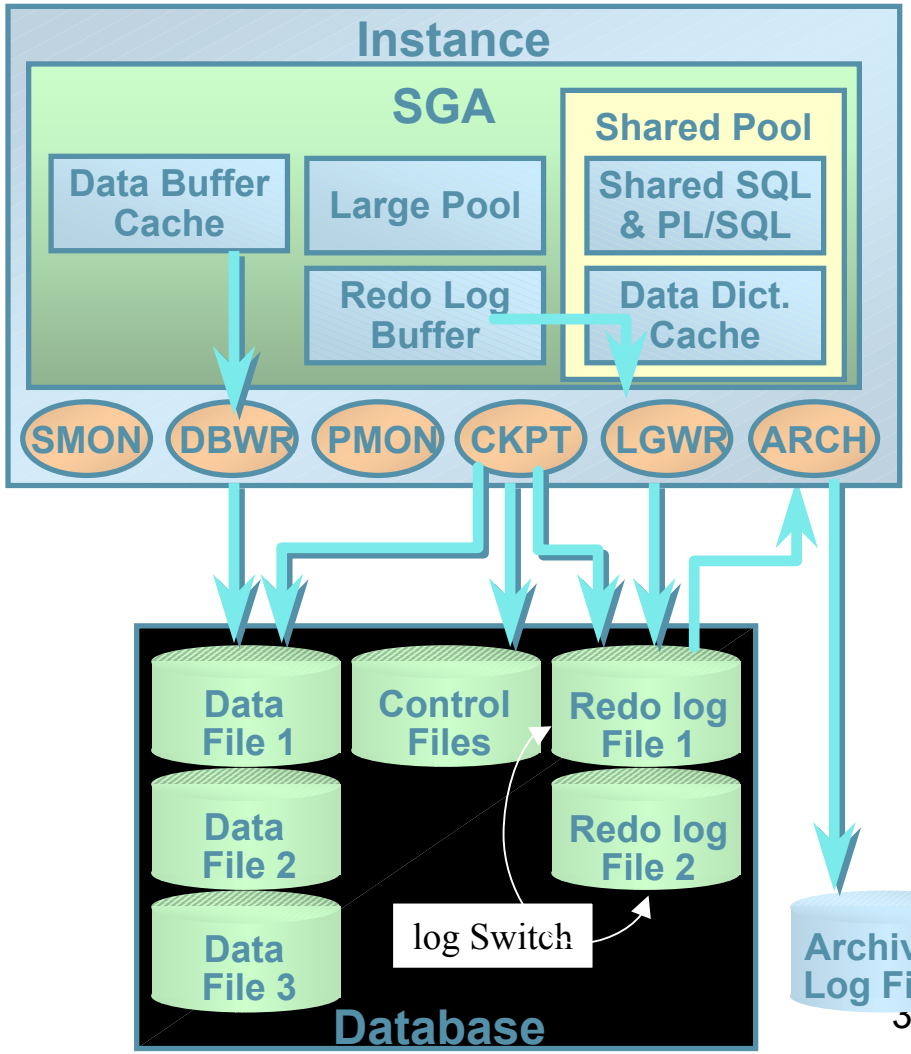
- **User Process**. Se crea cuando el usuario activa una herramienta como el SQLWorksheet.
- **Server Process**. Acepta las entradas del User Process y realiza los pasos necesarios para completar la petición del usuario.

El proceso de acceso a los datos



Procesos de una instancia Oracle

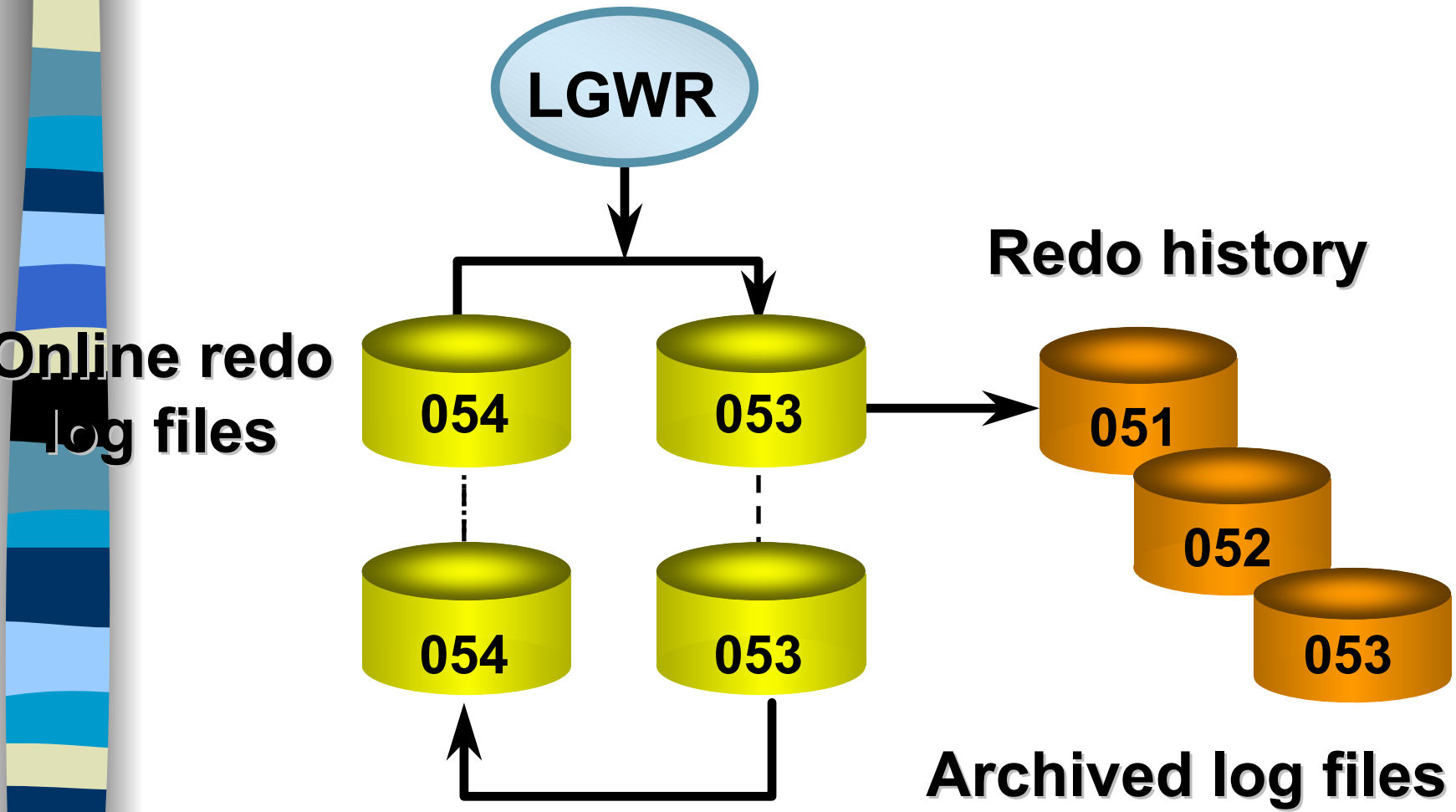
- DBWR (Database writer)
- CKPT (Checkpoint)
- LGWR (Log Writer)
- ARCH (Archive)



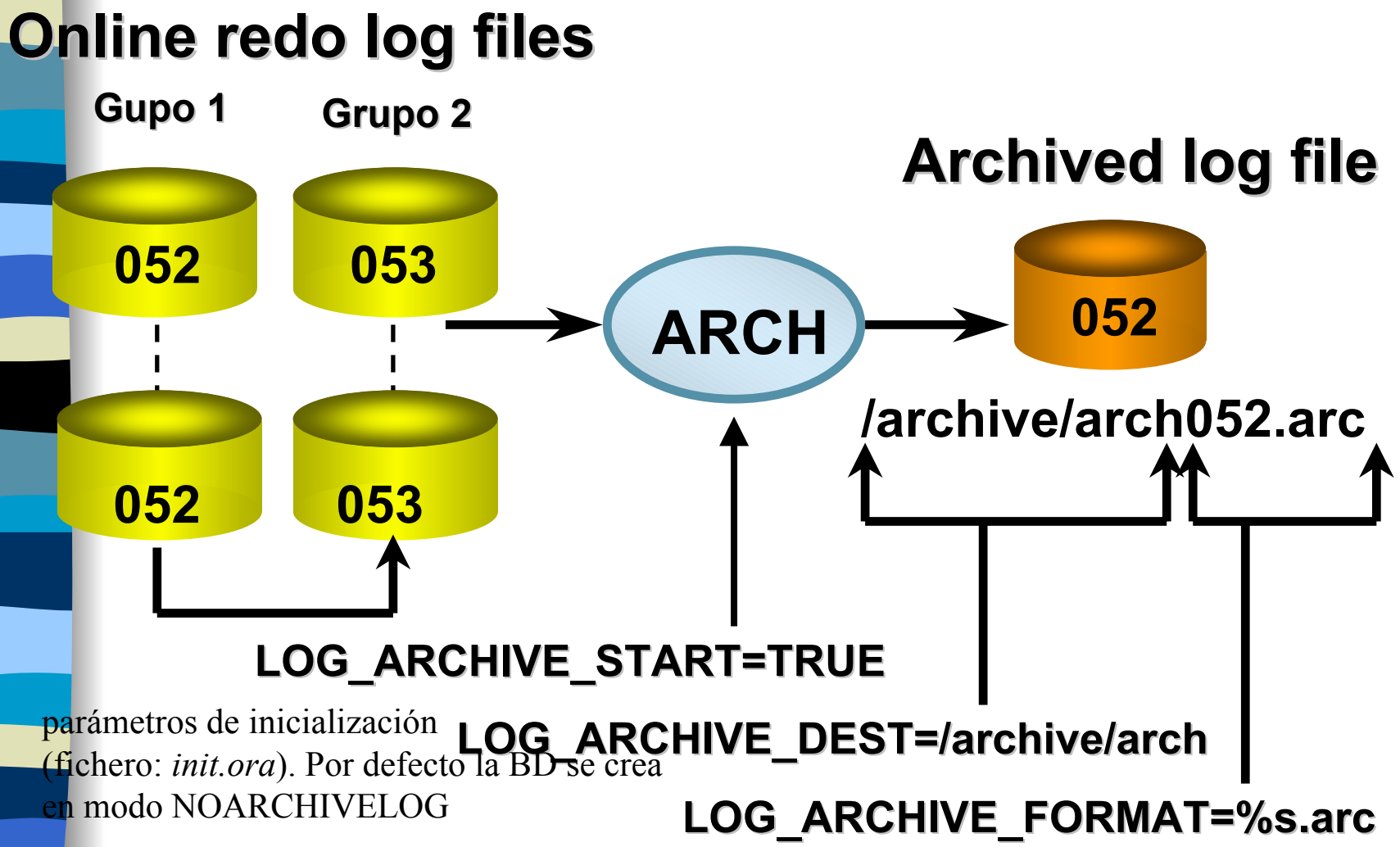
Proceso de recuperación

- Si se produce un fallo, el sistema recurre a los fichero de *log* para recuperar un estado consistente
- Pero, ¿hasta cuándo debe el sistema "recordar"? ¿cómo de grande tiene que ser el fichero de log?
  - escritura circular. El sistema va alternando entre dos o más ficheros. Cuando termina con el último, empieza a sobre-escribi el primero.
- Pero ¿qué ocurre si el error se produce en los ficheros de *log*?
  - escritura doble (multiplexada)

Gestión de *logs*. Modo de archivo

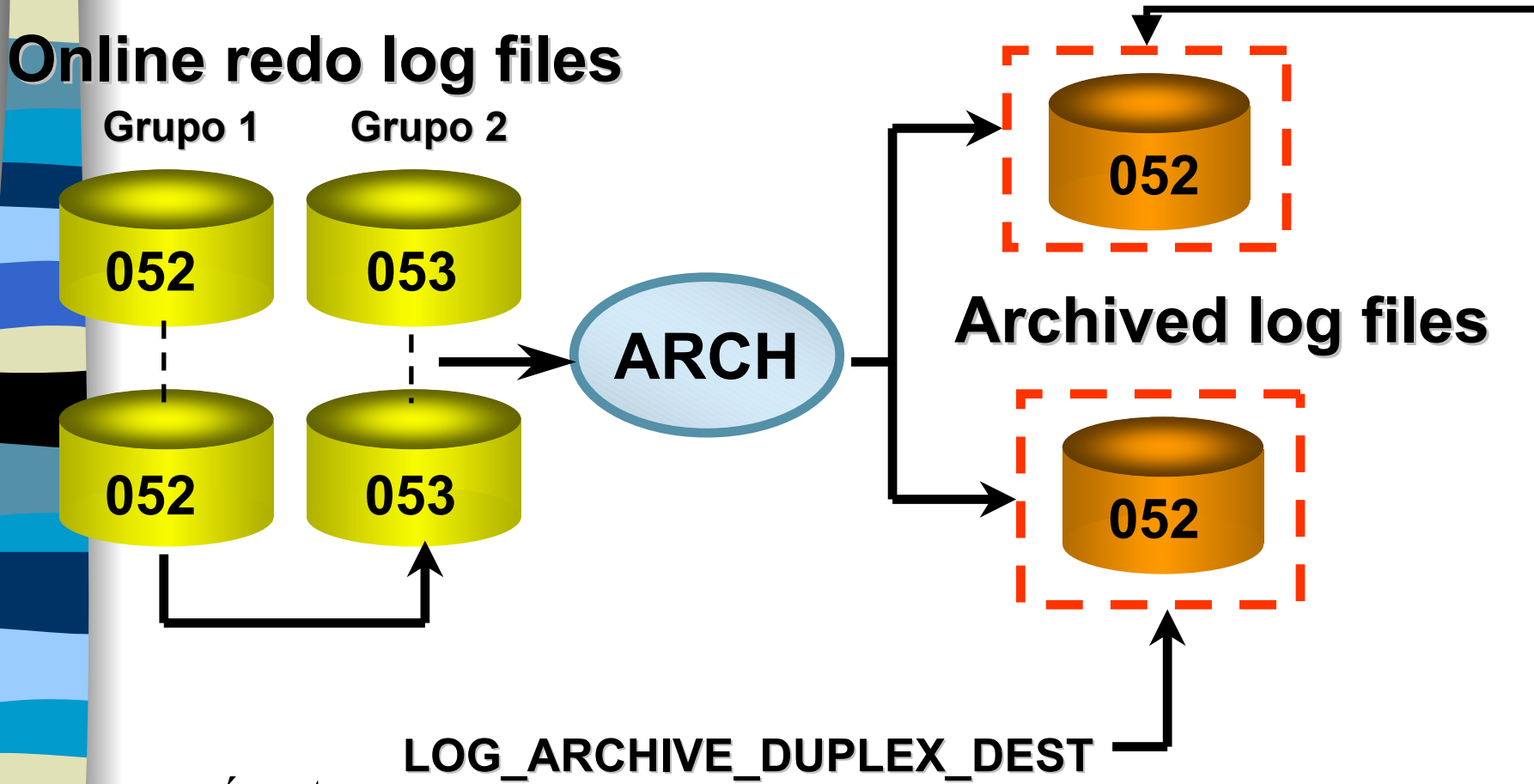


Gestión de *logs*. Modo de archivo



parámetros de inicialización (fichero: *init.ora*). Por defecto la BD se crea en modo NOARCHIVELOG

Gestión de *logs*. Duplicar el archivo



parámetros de inicialización fichero *init.ora*

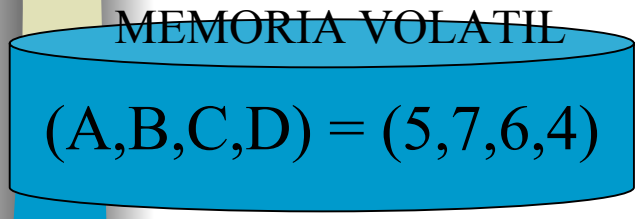
Fallos con pérdida de memoria estable. Proceso de *archivar*

- Idea básica: volcar periódicamente el contenido entero de la BD en almacenamiento estable : archivar

- ¿De qué se realiza el archivo?
  - de todos los datos: vuelco total *(full dump)*
  - sólo de los datos modificados desde el último proceso de archivar: vuelco incremental *(incremental dump)*

- ¿Cuándo se realiza el archivo?
  - con el SGBD parado (archivo “en frio”) puede requerir mucho tiempo.
  - con el SGBD funcionando (archivo “en caliente”)

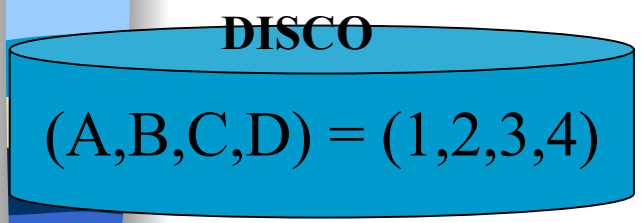
Proceso de archivar “en caliente”



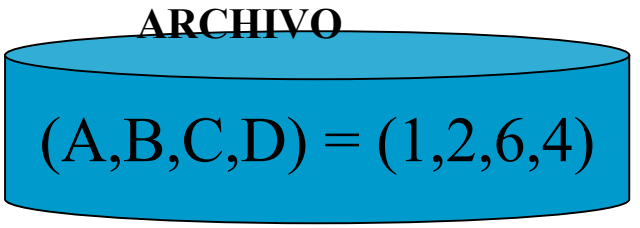
OUTPUT

Proceso de OUTPUT	Proceso de ARCHIVO
	copiar A
grabar(A,5)	
	copiar B
grabar(C,6)	
	copiar C
grabar(B,7)	
	copiar D

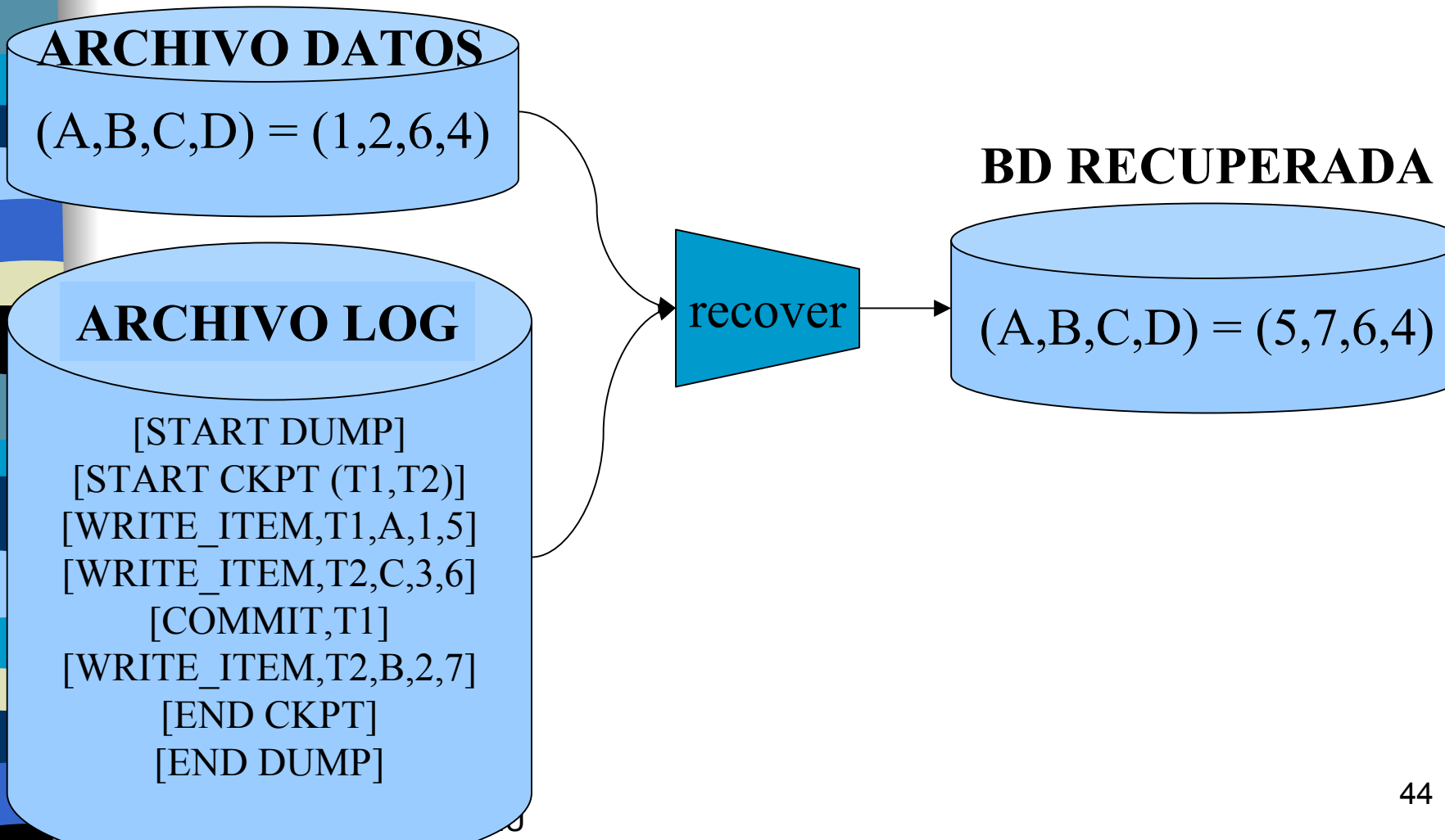
tiempo



ARCHIVAR



Recuperación “en caliente”



Recuperación con el archivo de *logs*

- ¿Dónde se encuentra los ficheros por recuperar?
  - sql> select * from v$recover_file;

FILE#	ONLINE	ERROR	CHANGE#	TIME
2		offline	288772	02-DEC-97

- Copiar el último back-up
  - unix> cp /disk1/backup/df2.dbf disk2/data/
- Recuperar este fichero a partir de los datos del archivo del diario (*log*)
  - recover datafile 'disk2/data/df2.dbf'