



# *Administración de Bases de Datos (Ingeniería Informática)*

## **Tema 8. GESTIÓN DE LA RECUPERABILIDAD**

1. Auditoría
2. Copia de Seguridad. Backup
3. Restauración. Recovery
4. Gestión del Deshacer: *Flashback*

- **Auditoría**: Controla los accesos y usos de los objetos de la BD, asociando cada acción a un único usuario. Por tanto nos proporciona una capacidad forense, dentro de la parte de seguridad retroactiva.
- La auditoría se encuentra activada por defecto en *mixed mode* (tradicional y unificada). Se puede comprobar accediendo a la vista:
  - `select * from unified_audit_trail;`
  - Y con `SELECT VALUE FROM V$OPTION WHERE PARAMETER = 'Unified Auditing';` -- FALSE indica que es mixta
- Son muchos los campos que aparecen en cada registro. Ejecutar:
  - `select column_name,comments from dba_col_comments where table_name='UNIFIED_AUDIT_TRAIL';`
- Aunque idealmente deberíamos utilizar solo la Unificada, podemos desactivar cualquiera de ella (y de hecho deberíamos hacerlo).

- **Auditoría**: La auditoría de Oracle 18 ha sufrido cambios importantes con respecto a la versión 11g.
  - Políticas de auditoría unificada. Podemos:
    - Crear tantas como queramos
    - Utilizar políticas unificadas por defecto
      - `ORA_SECURECONFIG` activada por defecto es una política que aglutina todas las opciones por defecto que ya estaban presentes en versiones de Oracle DB 11g y anteriores. Incluye muchas opciones y solapa a otras políticas.
      - `ORA_DATABASE_PARAMETER` audita la configuración de parámetros comunes como `ALTER DATABASE`, `ALTER SYSTEM` y `CREATE SPFILE`. No activada por defecto.
      - `ORA_ACCOUNT_MGMT` audita las configuraciones efectuadas sobre cuentas de usuario y privilegios, como `CREATE USER`, `ALTER USER`, `GRANT` y `REVOKE`. No activada por defecto.
  - Crear políticas fine-grained.

## UNIFIED\_AUDIT\_TRAIL

- En Oracle 18 los eventos auditados están unificados, pero provienen de distintas fuentes:
  - Registros de auditoría de las políticas unificadas y del comando AUDIT (Standard Audit)
  - Registros de auditoría fine-grained del paquete DBMS\_FGA PL/SQL
  - Oracle Database Real Application Security, Oracle Recovery Manager, Oracle Database Vault, Oracle Label Security, Oracle Data Mining, Oracle Data Pump, Oracle SQL\*Loader Direct Load
- La fuente de eventos de auditoría unificada es la vista UNIFIED\_AUDIT\_TRAIL.
  - A parte de SYS, los usuarios con los roles AUDIT\_ADMIN and AUDIT\_VIEWER pueden consultarla.
    - AUDIT\_ADMIN permite adicionalmente crear políticas de auditoría.
    - AUDIT\_VIEWER generalmente para auditores externos.

- No debe auditarse todo. Eso tiene un impacto en la BD.
- Evalúa la razón porque la que quieres auditar
  - Elegir las instrucciones para cumplir esas razones
- Utiliza una política de mínimos
  - Por ejemplo, sí sólo quieres auditar lo que ocurre durante una semana, configura una ventana de tiempo
- Asegúrate que los eventos que auditas cumplen con la política de tu empresa.
- No dejes escapar eventos que siempre serán interesantes
  - Logon, uso de privilegios de sistema, etc.
- Periódicamente archiva los registros de auditoría y purga el audit trail
- Ayúdate de los ficheros de log de la BD si fuera necesario realizar una investigación.

- Los pasos a realizar para establecer una auditoría unificada son:
  1. Establece una política de auditoría con `CREATE AUDIT POLICY`
  2. Usa el comando `AUDIT` para activarla y opcionalmente aplicarla (o excluirla de) a uno o más usuarios.
  3. Consulta el `unified_audit_trail` para buscar registros
  4. De manera periódica archiva los registros del trail y púrgalos (consultar documentación)
- No obstante, si la política es fine-grained, el procedimiento es distinto, ya que hay que hacer uso del paquete `DBMS_FGA`.
  - Permite políticas de auditoría más específicas aún
    - Grabar los accesos a una tabla determinada entre las 9p.m. y 6 a.m. o en Sábado y Domingo.
    - Grabar los eventos cuando la conexión provenga de una IP externa.
    - Select o update de una columna específica.
    - Etc.

# Auditoría de la Base de Datos

- En las políticas unificadas se pueden auditar:
  - Todas las actividades llevadas a cabo por una cuenta, rol o privilegio
  - Actividades sobre objetos (e.g. borrado de una tabla)
  - Application context values establecidos por las aplicaciones de BD.
  - Las actividades de Oracle Database Real Application Security, Oracle Recovery Manager, Oracle Data Mining, Oracle Data Pump, Oracle SQL\*Loader direct path events, Oracle Database Vault y Oracle Label Security.

```
CREATE AUDIT POLICY policy_name
  { {privilege_audit_clause [action_audit_clause] [role_audit_clause]}
    | { action_audit_clause [role_audit_clause] }
    | { role_audit_clause }
  }
  [WHEN audit_condition EVALUATE PER
  {STATEMENT|SESSION|INSTANCE}]
```



# Auditoria de la Base de Datos

- *privilege\_audit\_clause*: Describe las opciones relacionadas con el uso de privilegios
  - `PRIVILEGES privilege1 [, privilege2]`
  - La vista `SYSTEM_PRIVILEGE_MAP` nos da un listado de todos los privilegios de sistema auditables
- *action\_audit\_clause*: Describe las opciones relacionadas con el uso de objetos (standard actions) y acciones de sistema
  - `{standard_actions | component_actions} [, component_actions ]`
  - `component_actions`: para las acciones de componentes Oracle: Oracle Label Security, Oracle Database Real Application Security, Oracle Database Vault, Oracle Data Pump o Oracle SQL\*Loader (ver documentación de referencia)
  - `select * from AUDITABLE_SYSTEM_ACTIONS where component='Standard'`; para las acciones de sistema
- *role\_audit\_clause* describe las opciones relacionadas con el uso de privilegios de sistema concedidos directamente a los roles.
  - `ROLES role1 [, role2]`
- *WHEN audit\_condition EVALUATE PER*: Permite especificar una función para crear una condición.
  - `STATEMENT` evalúa la condición para cada comando.
  - `SESSION` evalúa la condición solo una vez en cada sesión, la primera vez que se use la política (y se cachea el resultado).
  - `INSTANCE` evalúa la condición solo una vez cada vez que se inicia la instancia de BD, la primera vez que se use la política (y se cachea el resultado).

## ***Acciones sobre objetos que se pueden auditar***

| Type of Object                                      | Actions  |
|---|--|
| Directory   | AUDIT, GRANT, READ   |
| Function  | AUDIT, EXECUTE (Notes 1 and 2), GRANT  |
| Java Schema Objects<br>(Source, Class,<br>Resource) | AUDIT, EXECUTE, GRANT  |
| Library   | EXECUTE, GRANT   |
| Materialized Views                                  | ALTER, AUDIT, COMMENT, DELETE, INDEX, INSERT, LOCK, SELECT, UPDATE                           |
| Mining Model  | AUDIT, COMMENT, GRANT, RENAME, SELECT  |
| Object Type   | ALTER, AUDIT, GRANT  |
| Package   | AUDIT, EXECUTE, GRANT  |
| Procedure   | AUDIT, EXECUTE (Notes 1 and 2), GRANT  |
| Sequence  | ALTER, AUDIT, GRANT, SELECT  |
| Table   | ALTER, AUDIT, COMMENT, DELETE, FLASHBACK, GRANT, INDEX, INSERT, LOCK, RENAME, SELECT, UPDATE |
| View  | AUDIT, DELETE, FLASHBACK, GRANT, INSERT, LOCK, RENAME, SELECT, UPDATE                        |

## ***Auditoria de la Base de Datos. Ejemplos***

```
CREATE AUDIT POLICY table_pol  
PRIVILEGES CREATE ANY TABLE, DROP ANY TABLE  
ROLES emp_admin, sales_admin;
```

```
CREATE AUDIT POLICY role_dba_audit_pol  
ROLES DBA;
```

```
CREATE AUDIT POLICY oe_orders_pol  
ACTIONS UPDATE ON OE.ORDERS  
WHEN 'SYS_CONTEXT("USERENV", "IDENTIFICATION_TYPE") = "EXTERNAL"  
EVALUATE PER STATEMENT;
```

```
CREATE AUDIT POLICY dml_pol  
ACTIONS DELETE on hr.employees,  
        INSERT on hr.employees,  
        UPDATE on hr.employees,  
        ALL on hr.departments;
```

## Auditoria de la Base de Datos

- Una vez creada la política, para que tenga efecto ha de activarse.
  - `AUDIT POLICY hr_mgr_audit_pol BY HR_MGR;`
    - Activa la política `hr_mgr_audit_pol` para el usuario `HR_MGR`
  - `AUDIT POLICY role_connect_audit_pol EXCEPT rlee, jrandolph;`
    - Activa la política `role_connect_audit_pol` para todos los usuarios excepto para los usuarios `rlee`, `jrandolph`.
  - `AUDIT POLICY admin_audit_pol BY USERS WITH GRANTED ROLES DBA, CDB_DBA;`
    - Activa la política `admin_audit_pol` para los usuarios con los roles (asignados directamente) `DBA` y `CBD_DBA`.
- La auditoría tiene efecto para los usuarios que se conectan **después** de la activación de la política.
- Si el uso de un privilegio de sistema y una acción sobre un objeto determinado coinciden en la misma o distintas políticas, solo se genera un registro de auditoría.

## ***Auditoria de la Base de Datos***

- Para desactivar una política NOAUDIT
  - E.g. NOAUDIT POLICY hr\_mgr\_audit\_pol BY HR\_MGR;
- Para eliminar una política (si fuera necesario) DROP
  - E.g DROP AUDIT POLICY hr\_mgr\_audit\_pol
- Para alterar una política
  - ALTER AUDIT POLICY table\_pol ADD ROLES dba;
- Para quitar privilegios de una política
  - ALTER AUDIT POLICY table\_pol DROP PRIVILEGES CREATE ANY TABLE;
- Para quitar una condición de una política
  - ALTER AUDIT POLICY order\_updates\_pol CONDITION DROP;
- Para modificar la condición(es) de una política
  - ALTER AUDIT POLICY emp\_updates\_pol CONDITION 'UID = 102' EVALUATE PER STATEMENT;

## ***Auditoría de la Base de Datos***

- Existen distintas vistas para acceder a la información relativa a la auditoría
- **AUDIT\_UNIFIED\_POLICIES**: Políticas unificadas creadas en la BD.
- **AUDIT\_UNIFIED\_ENABLED\_POLICIES**: Políticas unificadas activadas en la BD.
- **AUDIT\_UNIFIED\_POLICY\_COMMENTS**: Muestra la descripción de cada política unificada que se introdujera con el comando COMMENT.
- **UNIFIED\_AUDIT\_TRAIL**: Muestra todos los registros de auditoría.
- **ALL\_AUDIT\_POLICIES**: Muestra toda la información de las políticas fine-grained.
- **ALL\_DEF\_AUDIT\_OPTS**: Muestra toda la información de auditoría de objetos que se aplicará cuando se cree un objeto.

# Backup y Recovery

- **Fallos en la BD:** Existen muchos elementos que, de fallar, pueden llevar a pérdidas de información de la BD.
  - **Tipos de Fallos:** En el *hardware*, en Oracle, en programas de los usuarios y por errores de los usuarios.
  - El **peor caso** en el que se puede encontrar un administrador es la pérdida de uno o varios **discos** en los que se encuentren **datafiles**.
    - En este caso, el **administrador** no tendrá más remedio que utilizar una **Copia de Seguridad o BACKUP**, que puede ser total o parcial.
- Más fáciles de resolver son los casos en que no existe pérdida de integridad ni en los *datafiles* de la BD ni en los demás ficheros de control.
  - En este caso puede ser suficiente reparar la causa del fallo, reiniciar la máquina y levantar de nuevo las instancias.
  - Oracle llevará a cabo un proceso de recuperación que, caso de terminar correctamente, será capaz de situar a la BD en el momento en que la máquina falló.





# Backup y Recovery

- **Tipos de Backup:**

- **Backup Lógico:** Copiar los datos de la BD usando el programa Data Pump Export.
  - Se almacenan en un fichero binario. Puede ser parcial
- **Backup Físico:** Copiar los ficheros físicos de la BD.
  - Se puede hacer usando las utilidades del **S.O.**, o bien, la herramienta **Recovery Manager: RMAN**.

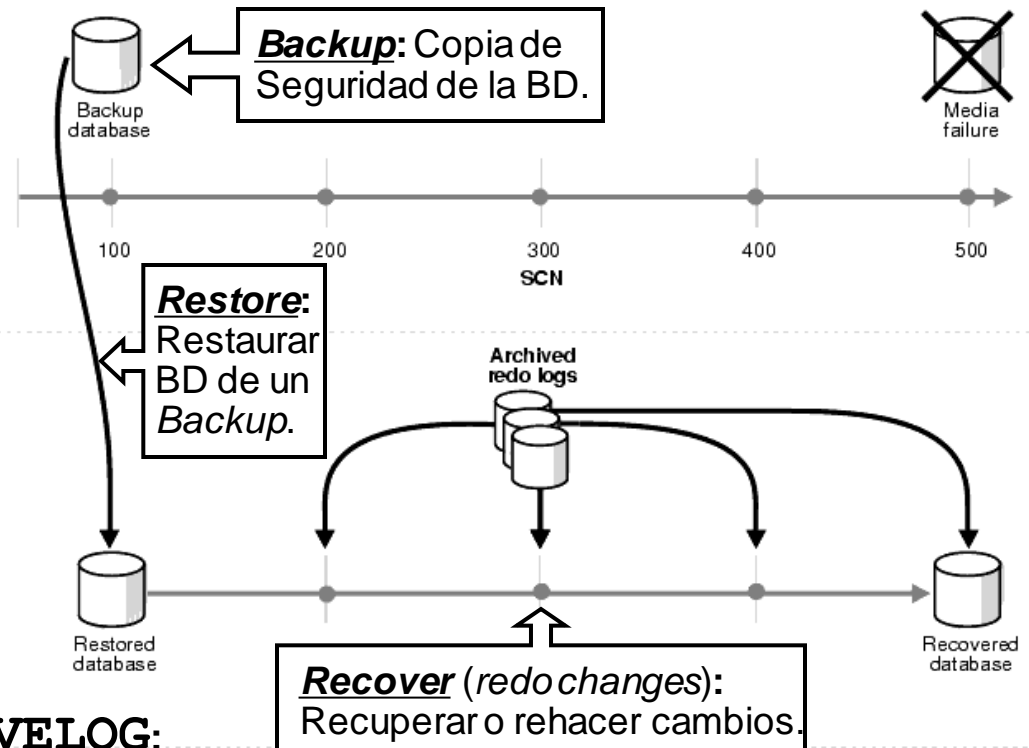
**RMAN** también permite recuperar datos de un *backup incremental*, que es el que contiene sólo los datos cambiados desde el último *backup*.

Se usarán los **registros de rehacer archivados** (archived redo logs) usando **RMAN** o las sentencias **RECOVER** or **ALTER DATABASE RECOVER**.

Con la BD montada y SIN abrir

La BD debe estar en modo **ARCHIVELOG**:

```
ALTER DATABASE [ARCHIVELOG | NOARCHIVELOG] ;
```



# Backup y Recovery

- **Registros de Transacciones:**
  - Oracle guarda los cambios a la BD en memoria, en el buffer del registro de rehacer (*redo log buffer*): Registros de rehacer (*redo records*).
  - El proceso **LGWR** de Oracle escribe ese buffer en disco frecuentemente en los **Ficheros del Registro de Rehacer Actuales** (*Online Redo Log Files*):
    - Suele hacerse en dos ficheros como mínimo de forma circular: Cuando se llena uno, cambia al otro alternativamente.
    - Si el sistema está en modo **ARCHIVELOG**, todos los ficheros del registro de rehacer (*redo logs files*) son copiados y guardados en disco en los llamados Ficheros del Registro de Rehacer Archivados (***Archived Redo Log Files***) cuando estos se han llenado y deben ser sobrescritos.
    - En modo **NOARCHIVELOG** no se guarda ese histórico de cambios y no podrá, por tanto, hacerse una recuperación total (*full point-in-time recovery*), si falla el sistema (a menos que hayamos hecho una *copia consistente*)
- **Características de una Buena Estrategia para Backup-Recovery:**
  - Tener varias copias de los ficheros de rehacer en distintos discos, y hacer frecuentes *backups*.
  - Guardar los *backups* en sitios seguros diferentes para evitar grandes desastres (incendios, inundaciones...).
  - Un sistema sofisticado es mantener una BD de réplica que se use en caso de cualquier tipo de fallo en la BD original.

# Backup y Recovery Físico

- **Estructuras de Datos Importantes para Backup-Recovery:**
  - **Datafiles (ficheros):** Almacenan los datos.
    - En el primer bloque (página) se almacena la cabecera (tamaño, tablespace, fecha de creación, SCN...).
    - Al abrir una BD Oracle compara esa información con la del Fichero de Control para determinar si es necesario un *Recovery*.
  - **Fichero de Control:** Cada BD tiene su fichero de control. Contiene:
    - Nombres en el S.O. de los ficheros de la BD y del registro de rehacer (actuales y archivados), que son leídos al montar la BD,
    - *Checkpoint*: Punto del registro de rehacer, el **mayor SCN**, en el que todos los cambios hechos a la BD antes de ese punto han sido guardados en la BD
  - – También contiene el *checkpoint* (mayor SCN) de cada *datafile*.
  - Otros datos importantes: nombre de la BD, fecha de creación, información sobre *backups*...
  - **Segmentos de Rollback:** Son estructuras lógicas guardadas en *datafiles* y contienen la información a cada modificación **no confirmada**.
    - El proceso **DBWn** puede escribir los datos en disco aunque no hayan sido confirmados, por lo que los segmentos de rollback son necesarios para garantizar la consistencia a nivel de sentencia y de transacción.
    - Oracle también los necesita para deshacer cambios no confirmados.

# Backup y Recovery

- Ficheros del Registro de Rehacer Actuales (*Online Redo Log Files*):
  - Guarda todos los cambios hechos a la BD.
  - Al confirmar los cambios, Oracle guarda el SCN.
  - Pueden ser multiplexados: Se guarda varias copias en diferentes discos, para evitar fallos en estos ficheros.
- Ficheros del Registro de Rehacer Archivados (*Archived Redo Log Files*):
  - Son los ficheros **Actuales** ya completos que se archivan en uno o más destinos, si la BD está en modo **ARCHIVELOG**.
  - En modo ARCHIVELOG:
    - Se puede hacer un *Recovery* de la BD con diversas opciones (completo, incompleto...).
    - Se puede hacer un *hot backup* (con la BD abierta y disponible).
    - Se puede mantener una BD de réplica (*standby database*), transmitiéndole y ejecutando estos ficheros.
    - Inconvenientes: Espacio en disco y tareas de gestión de estos ficheros.
  - En modo NOARCHIVELOG:
    - Sólo se puede hacer un *backup* con la BD cerrada, tras su finalización (*Shutdown*).
    - El único *Recovery* posible es un *Restore* de la BD, perdiendo todos los cambios hechos desde el último *Backup*.

# Backup y Recovery

- **Tipos de Backup Físico según su contenido:**
  - **Backup de la BD Completa:** Es el *backup* más típico.
  - **Tablespace Backup:** Copia de todos los ficheros de cierto *tablespace*.
    - Requiere que la BD esté en modo **ARCHIVELOG**, porque al recuperar ese *tablespace* será necesario hacer que sea coherente con el resto de *tablespaces*.
      - **Excepciones:** Si el *tablespace* está en modo de sólo lectura (*read-only*) o desactivado (*offline*). En esos casos, como el *tablespace* no puede modificarse, tampoco hay que rehacer cambios en él.
  - **Datafile Backup:** Copia de un *datafile*.
    - Lo normal es necesitar que exista copia también de todos los *datafiles* del *tablespace*, por lo que es mejor usar el tipo anterior.
    - Requiere que la BD esté en modo **ARCHIVELOG**.
      - **Excepciones:** Que esté en sólo lectura (*read-only*) o desactivado (*offline*).
  - **Backup del Fichero de Control:**
    - Se hace con la BD montada.
    - Sentencia: **ALTER DATABASE BACKUP CONTROLFILE.**
    - Oracle permite mantener varias copias del fichero de control (*multiplex*).

# Backup y Recovery

- **Backup Inconsistente:** Backup de uno o más ficheros, hecho mientras la BD está abierta (*open backup*) o cuando se ha caído anormalmente (*inconsistent closed backup*).
  - En este caso, no todos los ficheros han sido validados con el mismo SCN:
    - Requiere recuperación (*recovery*) de los datos posiblemente perdidos.
      - Puede usarse `V$RECOVER_FILE` para determinar qué ficheros recuperar.
    - Oracle no abrirá la BD hasta que estos SCN sean consistentes, o sea, hasta que todos los cambios de los registros de rehacer actuales (*online redo logs*) se hayan hecho en los *datafiles*.
  - Por tanto, si usamos este tipo de *backups* es buena medida:
    - Forzar a Oracle a archivar todos los registros no archivados
      - **BD abierta** : `ALTER SYSTEM ARCHIVE LOG CURRENT;`
      - **BD montada** : `ALTER SYSTEM ARCHIVE LOG ALL;`
    - Hacer copia de seguridad del fichero de control y de los registros de rehacer archivados producidos desde que se empezó el *backup*:
      - Esto asegura que el *backup* será útil aunque sea inconsistente.

# Backup y Recovery

- **Backup Consistente**: *Backup* de uno o más ficheros, hecho después de cerrar correctamente la BD.
  - Si se cerró mal (por fallo o por usar la sentencia **SHUTDOWN ABORT** los ficheros de la BD serán **INCONSISTENTES**, excepto que la BD se abriera en modo de sólo lectura.
  - **Al restaurar una BD (Restore)**: Oracle determina si el *backup* es o no consistente comparando las cabeceras de los *datafiles* con la información de cada datafile que tiene en el Fichero de Control.
    - Aquí, todos los ficheros han sido validados con el *checkpoint* del mismo **SCN**.
      - **Excepciones**: Si el *tablespace* está en modo de sólo lectura (*read-only*) o desactivado (*offline*) puede tener un SCN más antiguo. Como el *tablespace* no puede modificarse, Oracle no tiene nada que rehacer.
    - La BD puede abrirse **sin recuperar datos** (sin *recovery*).
  - En modo **NOARCHIVELOG** esta es la única opción de *backup* válida:
    - Los ficheros pueden ser inconsistentes y para que sean consistentes o cerramos la BD o tenemos la información para rehacer (*redo log*).
    - En este modo, un *backup* inconsistente sólo será útil si los ficheros de rehacer actuales están disponibles cuando se restaure la BD, lo cual es bastante improbable: **RMAN** no permite hacer *backup* de una BD caída anormalmente que corre en modo **NOARCHIVELOG**:
      - Si tu BD está en **modo NOARCHIVELOG** debes tener una **copia de seguridad consistente** (útil sin hacer *recovery*).



# Backup y Recovery

- **Tres Métodos de Backup:**
  - **1. RMAN (Recovery Manager): Backup físico** a partir de Oracle 8.
    - Requiere usar RMAN para operaciones de *Restore* y *Recovery*.
    - El único que admite *backup* incremental.
    - Almacena información sobre el *backup* en el fichero de control.
  - **2. Manualmente: Backup físico.**
    - Requiere especificar los nombres de los ficheros manualmente.
    - Puede escribirse un *script* para automatizar la operación.
  - **3. Usando la utilidad Data Pump Export: Backup lógico.**
    - Guarda información sobre los objetos del esquema, sin información sobre los *tablespaces*, fichero de control...
    - Requiere utilizar **Data Pump Import** para recuperar los datos.
- La **Recuperación** (*recovery*) **no** tiene que ser siempre de **toda la BD** y dejar la BD hasta el momento **actual**:
  - Puede interesar hacer una recuperación de una parte sólo y hasta un punto determinado en el tiempo (*incomplete recovery*), porque ciertos datos o una tabla completa se hayan borrado o corrompido...
    - **TSPITR** (*tablespace point-in-time recovery*): Recuperar uno o varios *tablespaces* a cierto punto en el tiempo.
  - Se puede hacer especificando el punto exacto en el tiempo, el SCN o el número de secuencia en el registro (con RMAN).



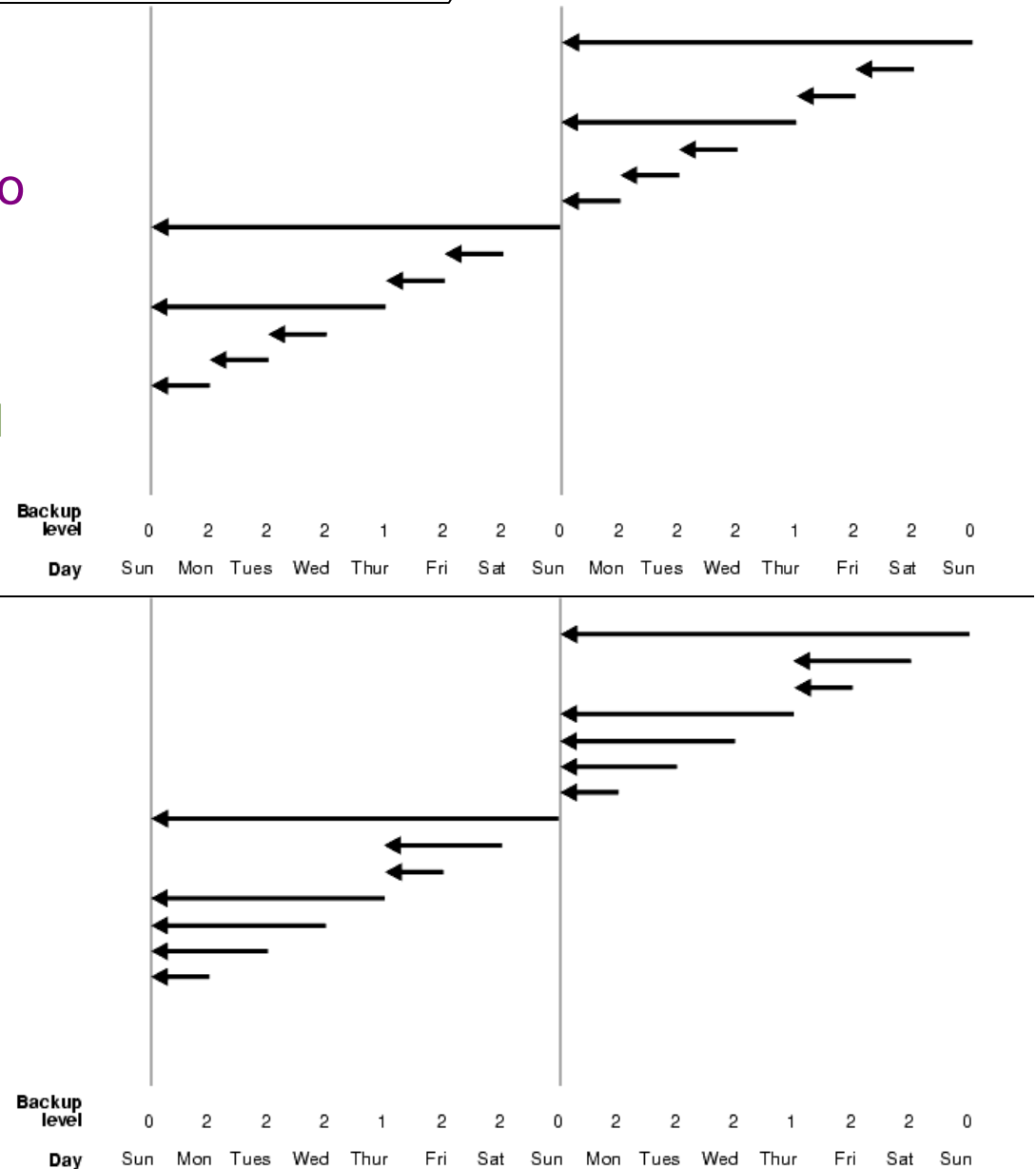
# 1. RMAN: *Recovery Manager*

- **RMAN**: Herramienta para *backup*, copiar, restaurar (*restore*) y recuperar (*recover*) ficheros de datos, de control y registros de rehacer archivados. **Permite, entre otras tareas:**
  - Crear *scripts* con las operaciones frecuentes de *backup/recovery*.
  - Crear *backups* incrementales, copiando sólo los datos que hayan cambiado desde el último *backup*.
  - Crear un duplicado de la BD de producción para pruebas.
  - Usar el catálogo (*recovery catalog*) para automatizar tareas de *restore/recovery*.
  - Encontrar los *datafiles* que requieren *backup*, usando el límite especificado por el usuario sobre la cantidad de rehacer que debe aplicarse para *recovery*.
  - Comprobar si cierto *backup* puede ser restaurado.
  - Crear informes sobre las operaciones de *backup*.
  - ***Backup* abierto y cerrado:** *Backup* de cualquier parte de la BD cuando está abierto, o cuando la BD está montada pero no abierta. Los *backups* cerrados pueden ser:
    - **Consistentes:** Con la BD montada, no abierta y cerrada correctamente (sin caídas o cierres incorrectos) y que, por tanto, son recuperados sin *recovery*.
    - **Inconsistentes:** Backup de cualquier parte cuando la BD está abierta o cerrada incorrectamente. Requiere *recovery* para que sea consistente.

# 1. RMAN: *Recovery Manager*

## – Tipos de Backup:

- **Total (Full):** De todo.
- **Incremental Multinivel:** Sólo de lo que haya cambiado desde el último *backup*.
  - Cada Incremental tiene su **nivel  $n$**  (0, 1, 2...), donde el 0 es un *backup* total.
  - A **nivel  $n$**  se hace *backup* de los cambios hechos desde el *backup* incremental a **nivel  $n$** , o inferior.
- **Backup Incremental Acumulativo Multinivel:**
  - El **nivel  $n$**  de este tipo de *backup*, copia los cambios hechos desde el *backup* más reciente a un **nivel inferior**.



## 2. Backup usando el S.O.

- **Obtener una lista con los Ficheros:**
  - **1. Ficheros que forman la BD** (columna `name` de `V$DATAFILE`), o cada uno con su *tablespace*, consultando `DBA_DATA_FILES`, o bien:

```
SELECT t.name "Tablespace", f.name "Datafile"
FROM V$TABLESPACE t, V$DATAFILE f
WHERE t.ts# = f.ts# ORDER BY t.name;
```
  - **2. Ficheros del Registro de Rehacer Actuales** (*online redo log files*):

```
SELECT member FROM V$LOGFILE;
```
  - **3. Ficheros de Control actuales:**

```
SELECT value FROM v$parameter
WHERE name = 'control_files';
```
- Si el **Backup** será **consistente** entonces cerrar la BD. Si será **inconsistente** puede hacerse un *backup* de un único *tablespace*, o de todos:
  - Si el *tablespace* no es de sólo lectura, hay que **bloquear** las escrituras:
    - Marcar el comienzo de un *backup online*, estableciendo el estado de sus ficheros como **NOT ACTIVE** (columna `status` de `V$BACKUP`):

```
ALTER TABLESPACE users BEGIN BACKUP;
```
    - Tras hacer el *backup*, desmarcar los tablespaces (estado **ACTIVE**):

```
ALTER TABLESPACE users END BACKUP;
```

### 3. Backup usando Data Pump Export/Import

- Aunque la utilidad real es mover datos, puede usarse para *backup*. Recupera los datos que se guardaron y se pierden los cambios posteriores.
- **Export:** obtiene una copia consistente de la BD en un momento concreto.
  - Lo ideal es ejecutar la BD en modo restringido, para evitar modificaciones de la BD mientras Data Pump Export está ejecutándose.
  - **Modos de Export:**
    - **Schemas:** Exporta todos los objetos de un usuario.
    - **Tables:** Exporta todas o algunas de las tablas de un usuario.
    - **Full:** Exporta la BD completa, con todos sus objetos. El rol **EXP\_FULL\_DATABASE** permite exportar la BD completa.
    - **Tablespaces:** Exporta los espacios de tablas
    - **Transport Tablespaces:** Exporta metadatos de los espacios de tablas
  - **Ejemplos:**

```
expdp HELP=Y -- Obtiene ayuda
expdp pepito/pepito directory=TEST_DIR
content=METADATA_ONLY dumpfile=pepito.dmp -- Export de
todo el esquema (modo por defecto) del usuario, sin
los datos en el fichero indicado
```

### 3. Backup usando Data Pump Export/Import

- Es una herramienta en sí misma de Oracle
- Algunos Parámetros:
  - **HELP=[Y/N]:** Muestra una ayuda con una descripción de los parámetros.
  - **usuario/contraseña:** Indica la conexión a realizar para realizar el export.
  - **TABLES=(lista\_tablas):** Export en modo Table para exportar dichas tablas.
  - **FULL=Y:** Indica que el export será completo. Combinado con el parámetro `TRANSPORTABLE` permite hacer una copia completa de la BD con algunas excepciones (ver documentación)
  - **DIRECTORY=Directorio** especifica el **objeto directorio** sobre el que trabajará (debe estar previamente creado).
  - **DUMPFILE=Fichero:** Fichero que la utilidad generará.
  - **CONTENT=[DATA\_ONLY/METADADTA\_ONLY/ALL]:** Si queremos que export incluya o no las filas.
  - **PARFILE=Fichero:** Especifica un fichero con los parámetros de Export.
- Hay muchas más opciones. Consultar la guía de Oracle 18

# Data Pump Export

Oracle SQL Developer : DATAPUMP\_EXPORT\_JOBS SYSTEM.null@ORCL Local System

File Edit View Navigate Run Team Tools Window Help

Connections

- Indexes
- Packages
- Procedures
- Functions
- Operators
- Queues
- Queues Tables
- Triggers
- Types
- Sequences
- Materialized Views
- Materialized View Logs
- Synonyms
- Public Synonyms
- Database Links
- Public Database Links
- Directories
- Editions
- Java
- XML Schemas
- XML DB Repository
- OLAP Option
- Analytic Views
- Scheduler
- Jobs

DBA

- Connections
- Apollo
- ORCL Local System
- Database Configuration
- Database Status
- Data Pump
- Export Jobs
- Import Jobs
- Performance
- RMAN Backup/Recovery
- Resource Manager
- SQL Translator Framework
- Scheduler
- Security
- Storage
- Tuning

Export Wizard - Step 2 of 8

Source

Connection: ORCL Local System

Data or DDL: Data and DDL

Types to be Exported in Data Pump

- ☐ Database
- ☒ Tablespaces
- ☐ Schemas
- ☐ Tables

Help < Back Next > Finish Cancel

LOG FILES

Refresh: 0

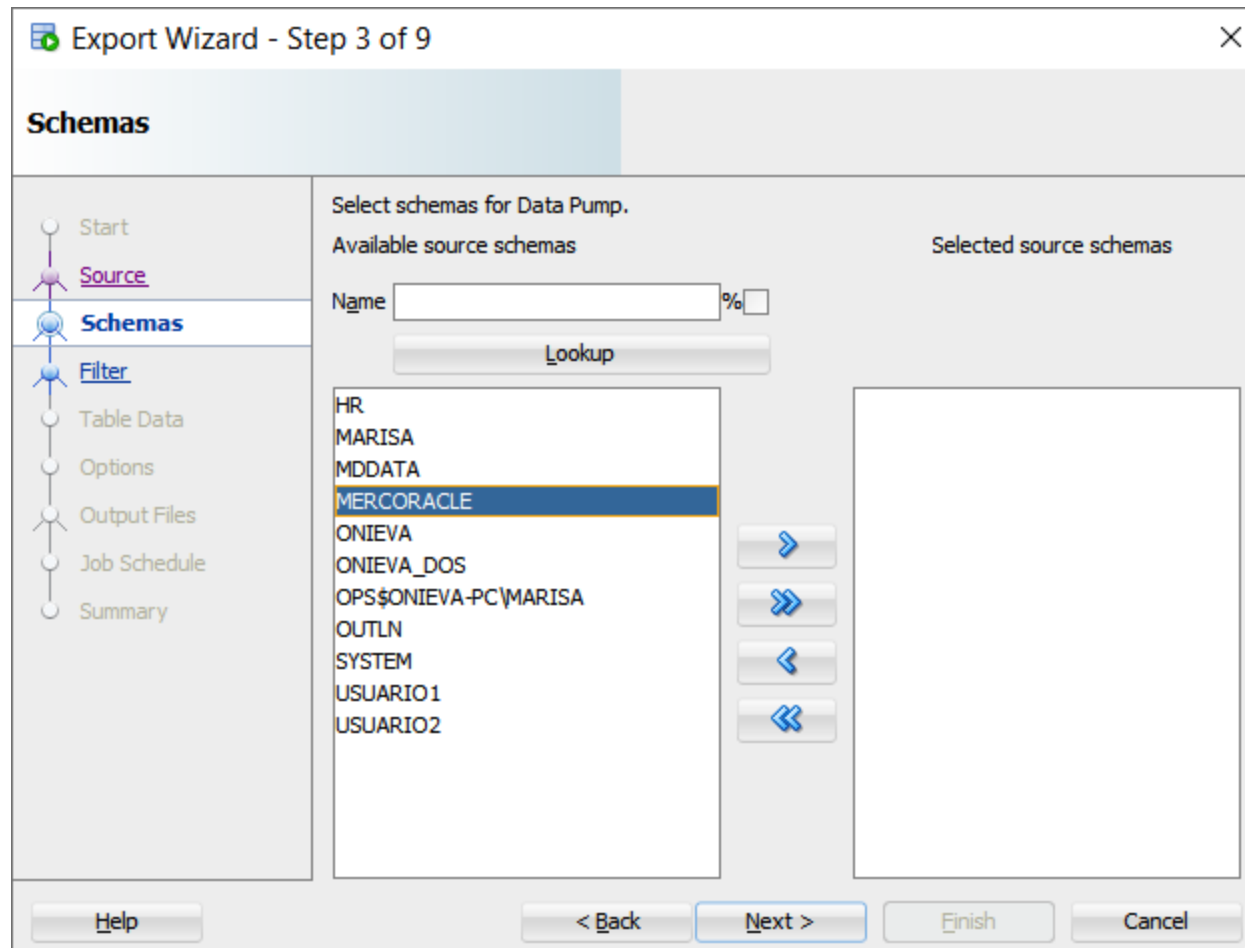
CHECKING MASTER TABLE

PL/SQL procedure successfully completed.

Error starting at line : 31 in command -

```
BEGIN
  IF (:UTIL_ERROR IS NOT NULL) THEN
```

# Data Pump Export



# Data Pump Export

Export Wizard - Step 5 of 9

## Table Data

Start  
Source  
Schemas  
Filter  
**Table Data**  
Options  
Output Files  
Job Schedule  
Summary

Name:  % ☒

| Database Object      | Columns | Object Where |
|----------------------|---------|--------------|
| MERCORACLE.CATEG...  | *       |              |
| MERCORACLE.CAT_E...  | *       |              |
| MERCORACLE.CLIENTE   | *       |              |
| MERCORACLE.DETALLE   | *       |              |
| MERCORACLE.EMPLEA... | *       |              |
| MERCORACLE.ENTREGA   | *       |              |
| MERCORACLE.EQUIPO    | *       |              |

Global Where:



# Data Pump Export

Export Wizard - Step 6 of 9

## Options

Start  
Source  
Schemas  
Filter  
Table Data  
**Options**  
Output Files  
Job Schedule  
Summary

Thread Number: 1

Estimate: Statistics Calculate 193KBytes

☒ Enable Logging

Log File: DATA\_PUMP\_DIR EXPDAT.LOG

☐ Export read-consistent view of data

By SCN ☒  
By Date ☐

VERSION: COMPATIBLE

☐ Delete Master table

Help < Back Next > Finish Cancel

# Data Pump Export

Export Wizard - Step 7 of 9

## Output Files

Start  
Source  
Schemas  
Filter  
Table Data  
**Options**  
**Output Files**  
Job Schedule  
Summary

Choose Output Files

| Directories   | File Names   | Size (M) |
|---------------|--------------|----------|
| DATA_PUMP_DIR | EXPDAT%U.DMP |          |

Add Row Remove Row

☒ Delete Existing Dump Files

☒ Append Timestamp to Dump, Log and Job names

☒ Compression (Requires At least 11g + Advanced Compression Option)

☐ Copy files to Oracle Object Storage Service (OSS)

OSS Transfer

☐ Oracle Directory path override:

Data Pump Export Job Name:

☐ Proceed to summary.

Help < Back Next > Finish Cancel

# Data Pump Export

Export Wizard - Step 8 of 9

## Job Schedule

[Start](#)  
[Source](#)  
[Schemas](#)  
[Filter](#)  
[Table Data](#)  
[Options](#)  
[Output Files](#)  
**[Job Schedule](#)**  
[Summary](#)

**Job Parameters**

Job Name:

Job Description:

**Job Schedule**

**Start**

☒ Immediately

☐ Later

Date

**Repeat**

☒ One Time Only

☐ Interval

Frequency

☐ Monthly

☐ Yearly

**Repeat Until**

☒ Indefinite

☐ Custom

Date

### 3. Backup usando Data Pump Export/Import

- **Es una herramienta en sí misma de Oracle**
- **Import:** El orden en el que se introducen los datos es el siguiente:
  - 1. Se introducen las definiciones de los tipos y de las tablas.
  - 2. Se introducen las filas correspondientes a cada tabla.
  - 3. Se crean los índices y se introduce su información.
  - 4. Finalmente, las restricciones de integridad, las vistas y los *triggers*.
- Si existían los objetos antes de lanzar el Import podemos tener problemas al utilizarlo: Las restricciones de integridad pueden rechazar la inserción de filas existentes en el fichero del Export.
  - **Soluciones:** Deshabilitar las restricciones y los *triggers*, o hacer que Import se salte alguno de esos pasos.
    - EXCLUDE: Permite excluir objetos y datos en la importación
    - INCLUDE: Permite incluir específicamente objetos y datos en la importación
    - REMAP\_DATA: Permite hacer un *mapeado* de datos al insertarlos. Un uso común es para la generación de claves primarias para evitar conflictos en la importación.
    - TABLE\_EXISTS\_ACTION=[SKIP/APPEND/TRUNCATE/REPLACE] nos dice como proceder cuando la tabla importada ya existe.
- **Ejemplo:**

```
impdp pepito/pepito REMAP_SCHEMA=pepito:nuevo
dumpfile=pepito.dmp -- Import de todo el esquema del usuario, sin
los datos, en otro esquema. El usuario debe tener permiso de acceso
al objeto de directorio en el que se encuentra el dumpfile.
```

# Tecnología de Flashback:

## Ventajas

> [Visión General](#)  
Base de Datos  
Tabla  
Borrado  
Consulta  
Versiones  
Transacción

- Las técnicas de recuperación tradicionales son lentas.
  - Se debe restaurar toda la base de datos o un archivo (no sólo los datos incorrectos).
  - Se deben examinar todos los cambios del log de la base de datos.
- La operación de flashback es *rápida*.
  - Los cambios se indexan por fila y por transacción.
  - Sólo se restauran los datos cambiados.
- Los comandos de flashback son *sencillos*.
  - No se incluyen procedimientos complejos de varios pasos.

## ***Cuándo Utilizar la Tecnología de Flashback***

| <b>Nivel de Objeto</b> | <b>Ejemplos de Supuestos</b>  | <b>Tecnología de Flashback</b> | <b>Usos</b>                  | <b>Afecta a los Datos</b> |
|------------------------|---|--------------------------------|------------------------------|---------------------------|
| <b>Base de datos</b>   | <b>Truncar una tabla; cambios no deseados realizados en varias tablas</b> | <b>Base de datos</b>           | <b>Logs de flashback</b>     | <b>Verdadero</b>          |
| <b>Tabla</b>           | <b>Borrar una tabla</b>   | <b>Borrado</b>                 | <b>Papelera de reciclaje</b> | <b>Verdadero</b>          |
|                        | <b>Actualizar con la cláusula WHERE incorrecta</b>                        | <b>Tabla</b>                   | <b>Datos de deshacer</b>     | <b>Verdadero</b>          |
|                        | <b>Comparar datos actuales con datos anteriores</b>                       | <b>Consulta</b>                | <b>Datos de deshacer</b>     | <b>Falso</b>              |
|                        | <b>Comparar versiones de una fila</b>                                     | <b>Versión</b>                 | <b>Datos de deshacer</b>     | <b>Falso</b>              |
| <b>Transacción</b>     | <b>Investigar varios estados históricos de los datos</b>                  | <b>Transacción</b>             | <b>Datos de deshacer</b>     | <b>Falso</b>              |

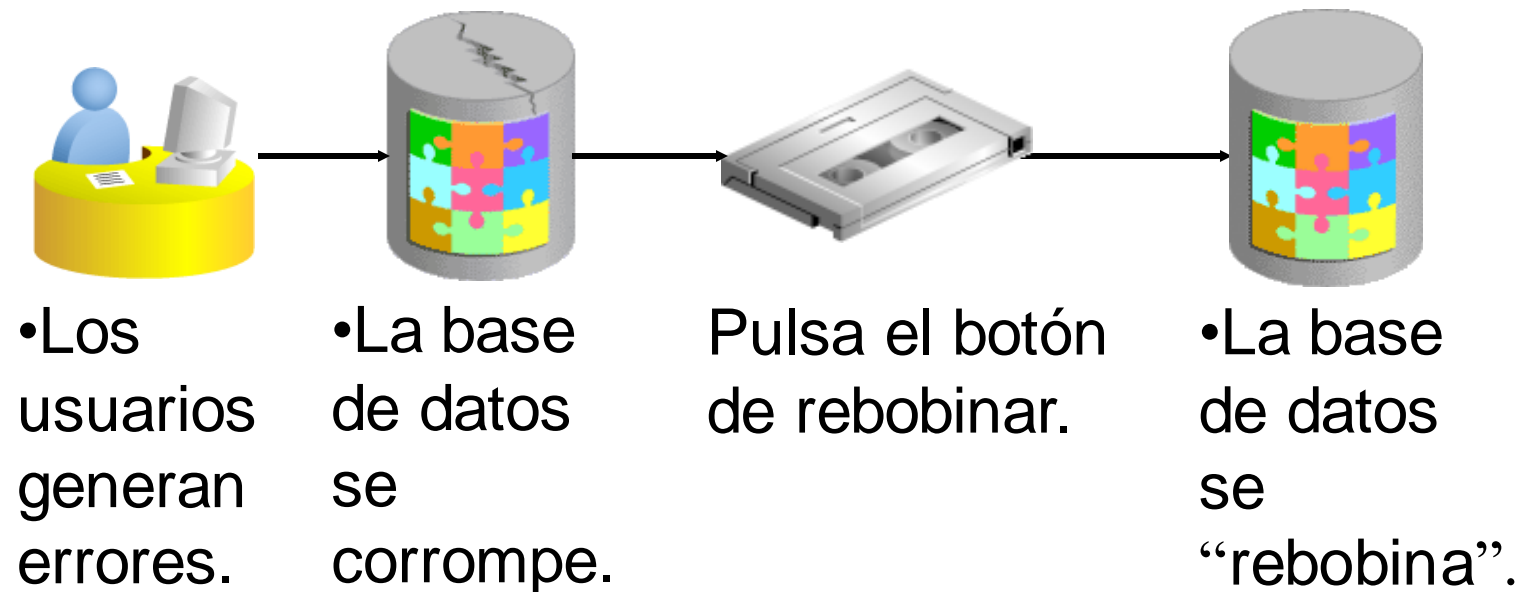
## *Realización de Flashback en Cualquier Error*

- La función de flashback de **base de datos** devuelve la base de datos a un punto anterior en el tiempo al deshacer todos los cambios realizados desde ese momento.
- La función de flashback de **tabla** recupera una tabla a un punto en el tiempo pasado sin tener que restaurar una copia de seguridad.
- La función de flashback de **borrado** restaura las tablas borradas por accidente.

# Flashback de **Base de Datos**: Visión General

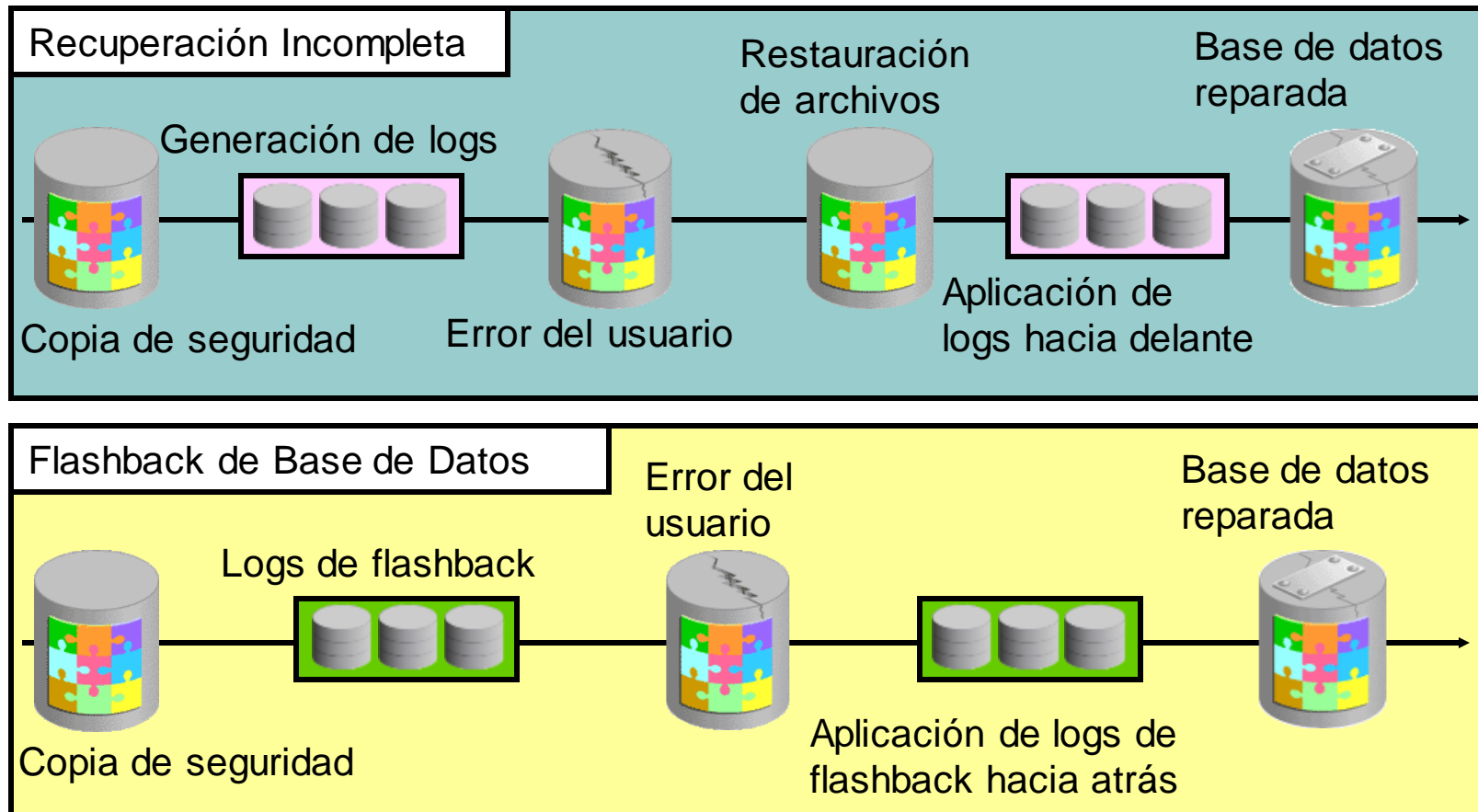
Visión General  
> Base de Datos  
Tabla  
Borrado  
Consulta  
Versiones  
Transacción

- La operación de flashback de base de datos:
  - Funciona como un botón de rebobinado en la base de datos
  - Se puede utilizar en casos de corrupción de datos lógicos provocada por los usuarios





# Flashback de Base de Datos: Reducción del Tiempo de Restauración



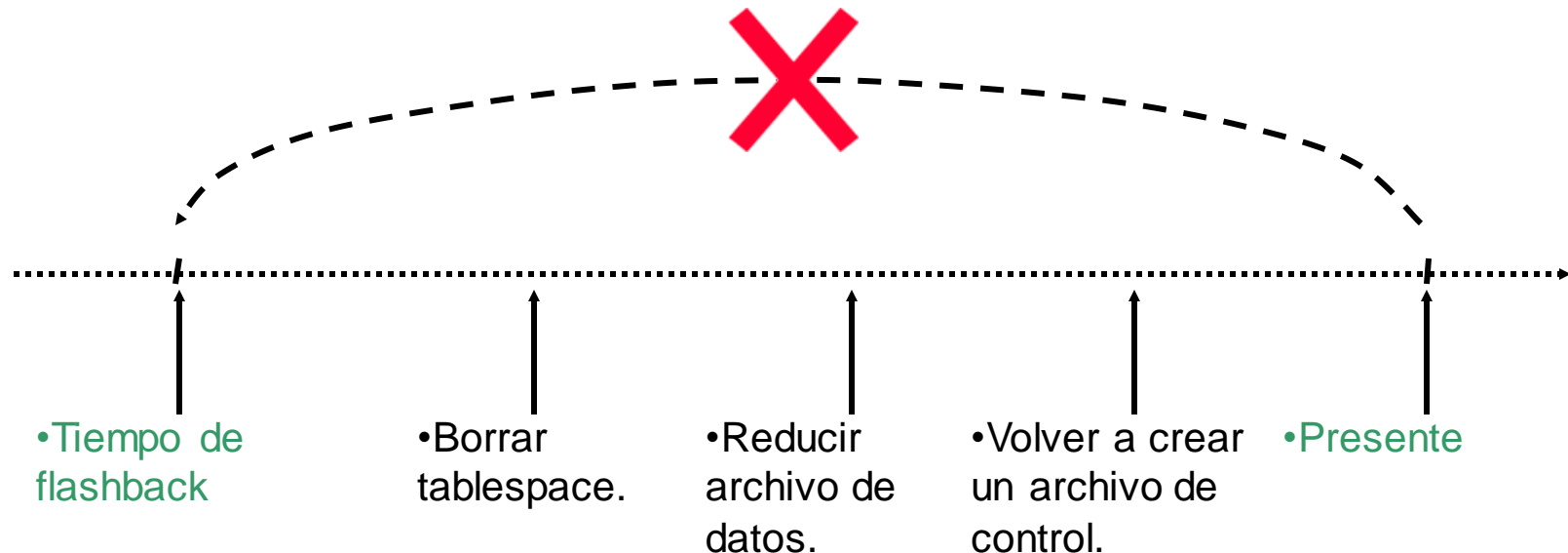
Con el flashback de base de datos, el tiempo para recuperar una base de datos es ahora proporcional al número de cambios que es necesario deshacer (y no al tamaño de la base de datos) porque no tiene que restaurar los archivos de datos.

## ***Flashback de Base de Datos: Consideraciones***

- Una vez terminada la operación de flashback de base de datos, la base de datos se debe abrir mediante uno de los siguientes métodos:
  - En modo de sólo lectura para verificar que se han utilizado la hora o el SCN correctos
  - Con el parámetro `RESETLOGS` para permitir las actualizaciones
- Lo contrario de “flashback” es “recuperar”.

## Flashback de Base de Datos: Limitaciones

- No se puede utilizar la función de flashback de base de datos en las siguientes situaciones:
  - El archivo de control se ha restaurado o se ha vuelto a crear.
  - Se ha borrado un tablespace.
  - Se ha reducido un archivo de datos.



## ***Configurar Flashback Database***

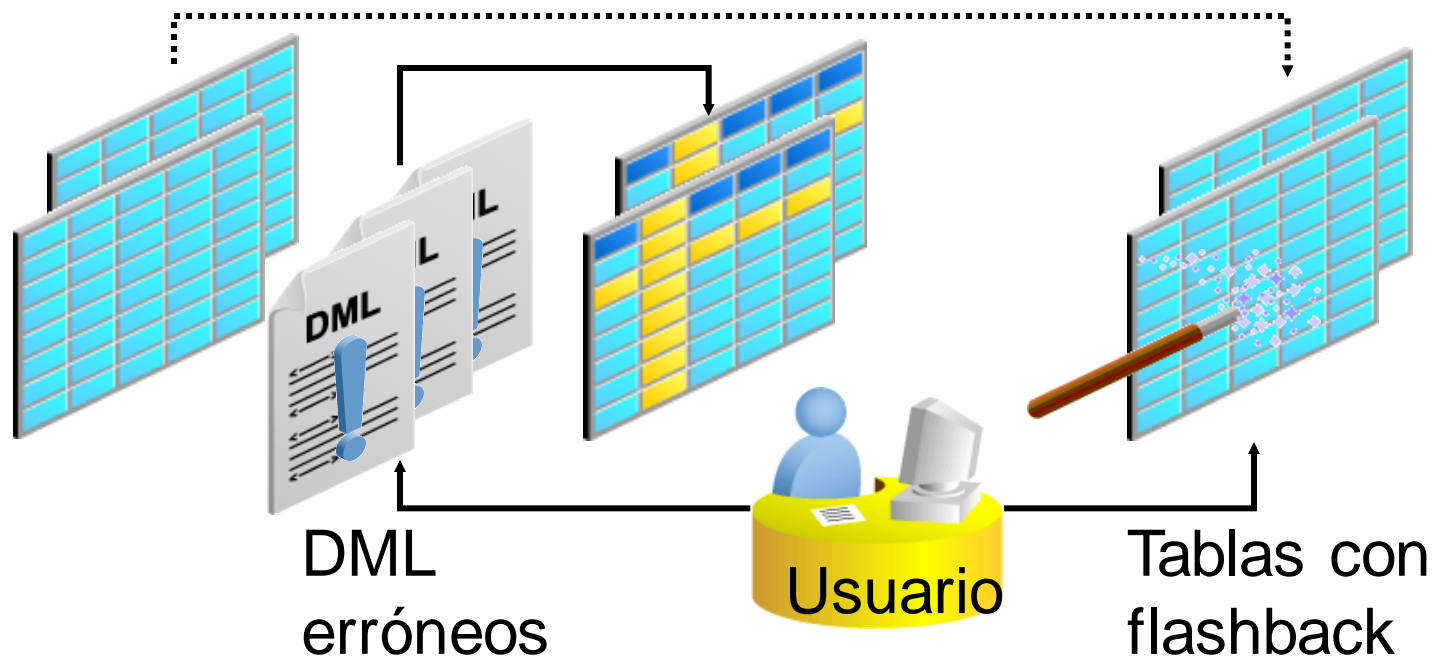
- Consultar si ya está disponible
  - `SELECT FLASHBACK_ON FROM V$DATABASE;`
- Configurar un fast recovery area y poner la BD en modo ARCHIVELOG.
  - Ver la documentación
- Activa Flashback Database
  - `ALTER DATABASE FLASHBACK ON;`

# Flashback de **Tabla**:

## Visión General

Visión General  
Base de Datos  
> **Tabla**  
Borrado  
Consulta  
Versiones  
Transacción

- El flashback de tabla recupera tablas a un punto en el tiempo específico.
- El flashback de tabla se produce in situ.
- La base de datos permanece online.



## ***Flashback de Tabla***

- Con el flashback de tabla, puede devolver una o más tablas a un punto en el tiempo específico sin restaurar ninguna copia de seguridad.
- Los datos se recuperan de un tablespace de deshacer para realizar una operación de flashback de tabla.
- El movimiento de filas debe estar activado en la tabla en la que se vaya a realizar la operación de flashback.

- ALTER TABLE pieza ENABLE ROW MOVEMENT;

## ***Realización de Flashback de Tabla***

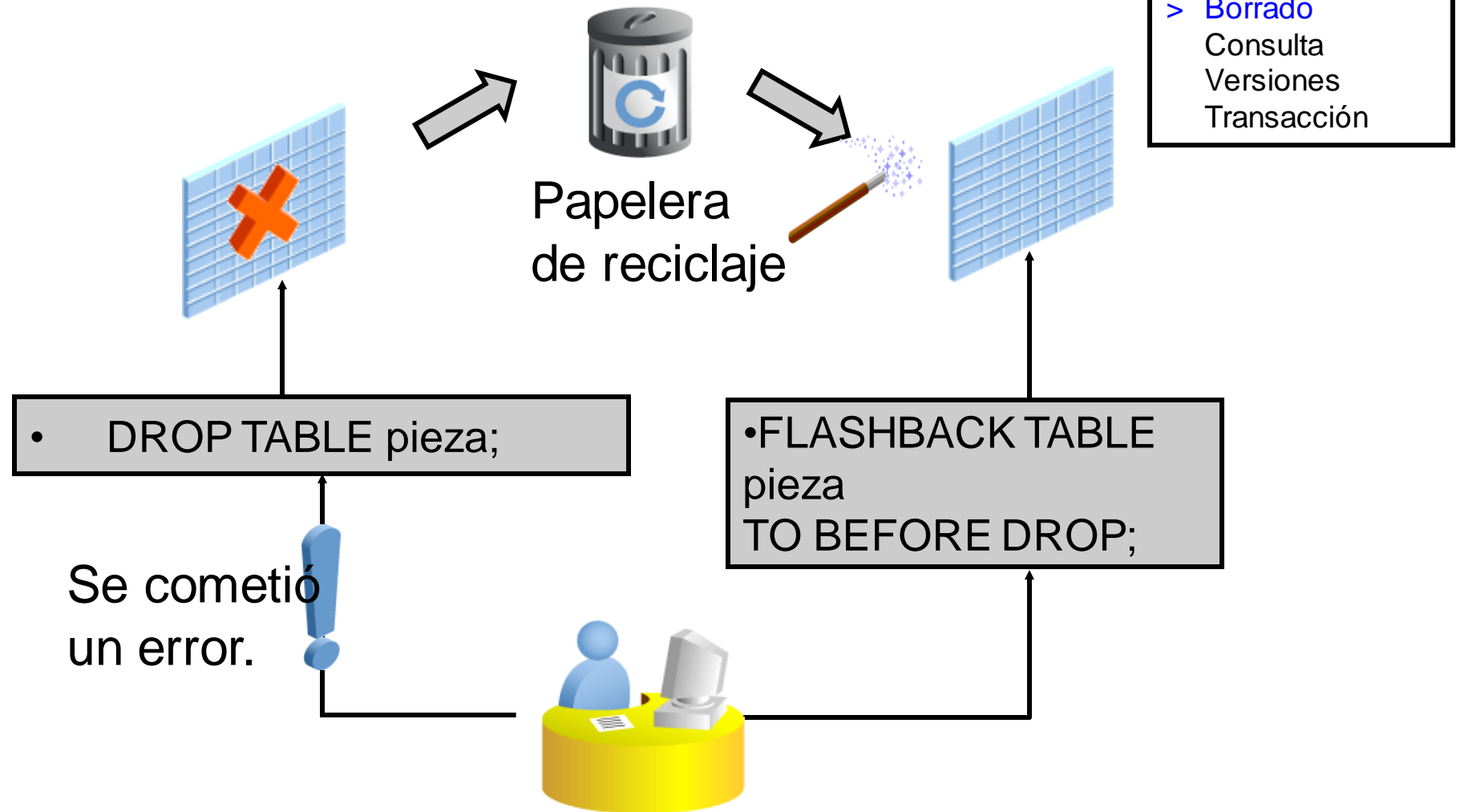
- FLASHBACK TABLE esc.pieza TO TIMESTAMP
- TO\_TIMESTAMP('2010-05-05 05:32:00',
- 'YYYY-MM-DD HH24:MI:SS');

## ***Flashback de Tabla: Consideraciones***

- El comando `FLASHBACK TABLE` se ejecuta como una sola transacción y adquiere bloqueos de DML exclusivos.
- El flashback no afecta a las estadísticas.
- Se mantienen los índices y los objetos dependientes actuales.
- Operaciones de flashback de tabla:
  - No se pueden realizar en tablas del sistema
  - **Sólo abarca algunas operaciones de DDL**
  - Se escriben en el archivo log de alertas
  - Generan datos de deshacer y de redo



# Flashback de **Borrado**: Visión General



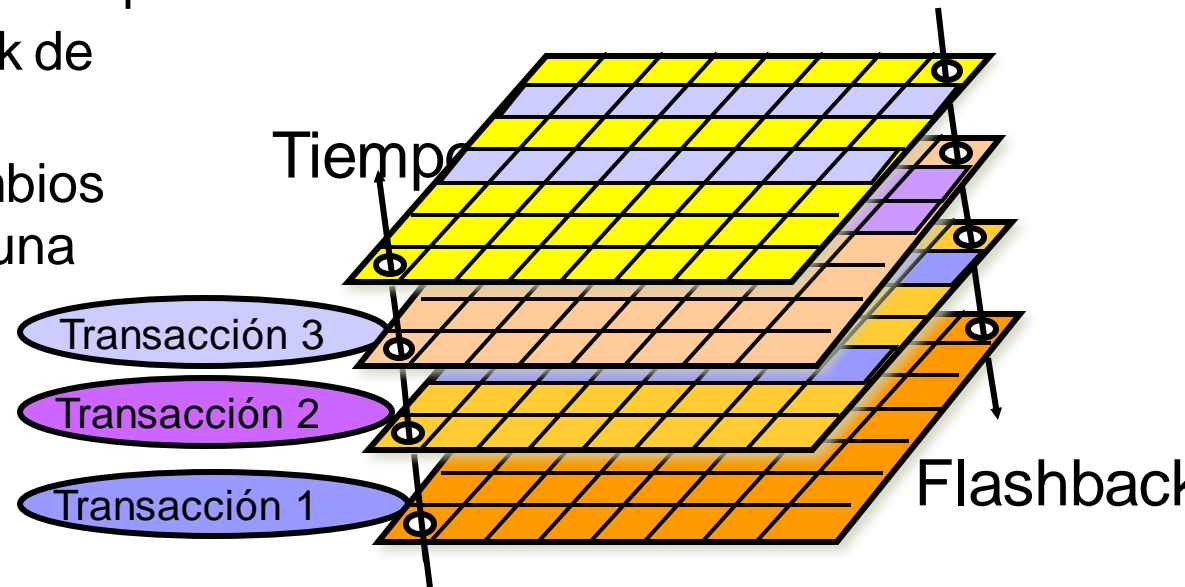
## ***Flashback de Borrado: Consideraciones***

- La función de flashback de borrado no funciona con las tablas que:
  - Residen en el tablespace `SYSTEM`
  - Utilizan una política de auditoría fine-grained o base de datos privada virtual
  - Residen en un tablespace gestionado por diccionario
  - Han sido depuradas, ya sea de forma manual o automática por problemas de espacio
- Las siguientes dependencias no se protegen:
  - Índices de unión de bitmap
  - Logs de vistas materializadas dependientes
  - Restricciones de integridad referencial
  - Índices borrados antes que las tablas
- Los índices asociados se recuperan si no se han borrado manualmente de manera previa al borrado de table.

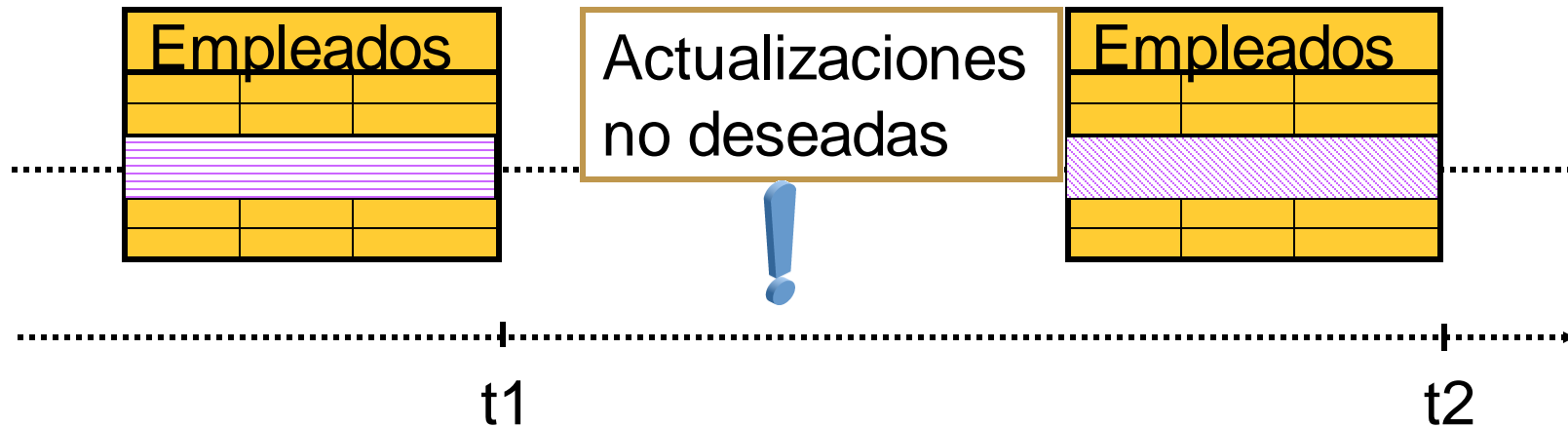
# Acceso al Tiempo de Flashback

Visión General  
Base de Datos  
Tabla  
Borrado  
> Consulta  
Versiones  
Transacción

- Consulta de flashback
  - Consultar todos los datos de un punto en el tiempo determinado.
- Consulta de flashback de versiones
  - Ver todas las versiones de una fila entre dos horas.
  - Ver las transacciones que cambiaron la fila.
- Consulta de flashback de transacciones
  - Ver todos los cambios que ha realizado una transacción.

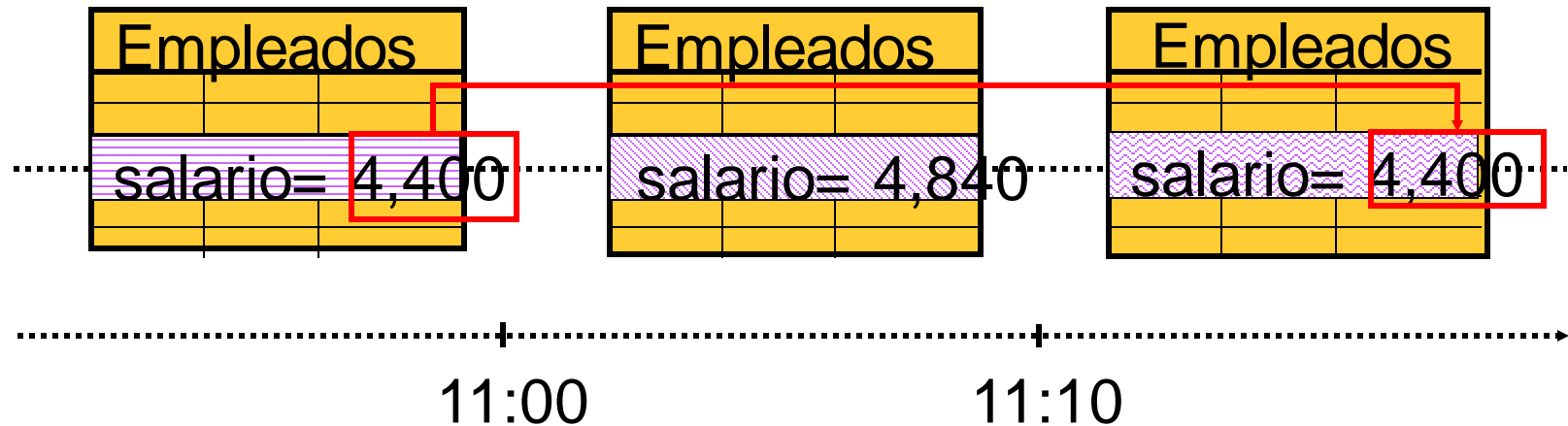


## Consulta de Flashback: Visión General



- SELECT numero, salario FROM empleados
- AS OF TIMESTAMP  $t_1$
- WHERE numero = 200

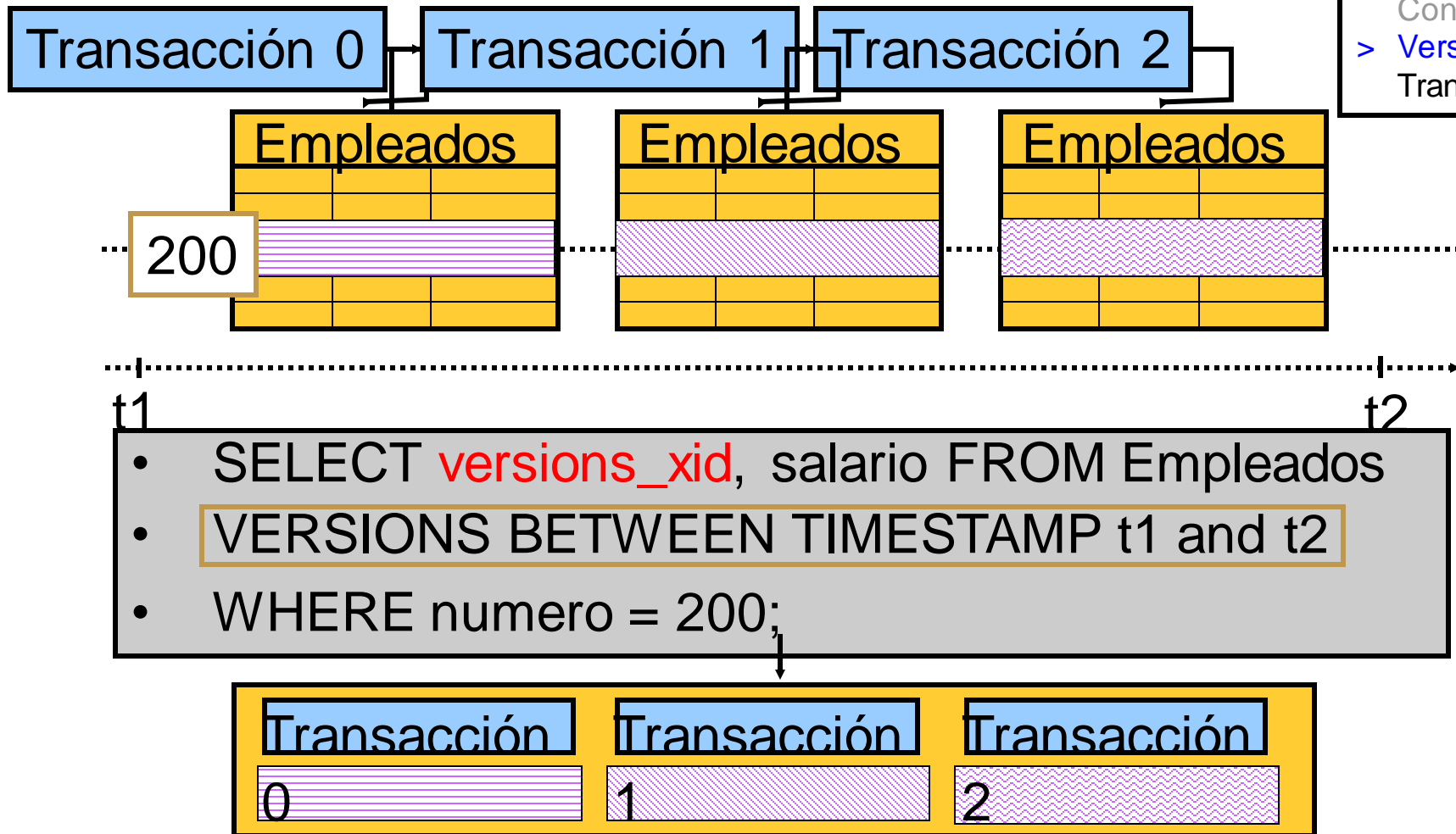
## Consulta de Flashback: Ejemplo



```
UPDATE Empleados SET salario =  
  (SELECT salario FROM Empleados  
   AS OF TIMESTAMP TO_TIMESTAMP  
   ('2010-05-04 11:00:00', 'yyyy-mm-dd hh24:mi:ss')  
   WHERE numero = 200)  
WHERE numero = 200
```

# Consulta de Flashback de Versiones: Visión General

Visión General  
Base de Datos  
Tabla  
Borrado  
Consulta  
> Versiones  
Transacción



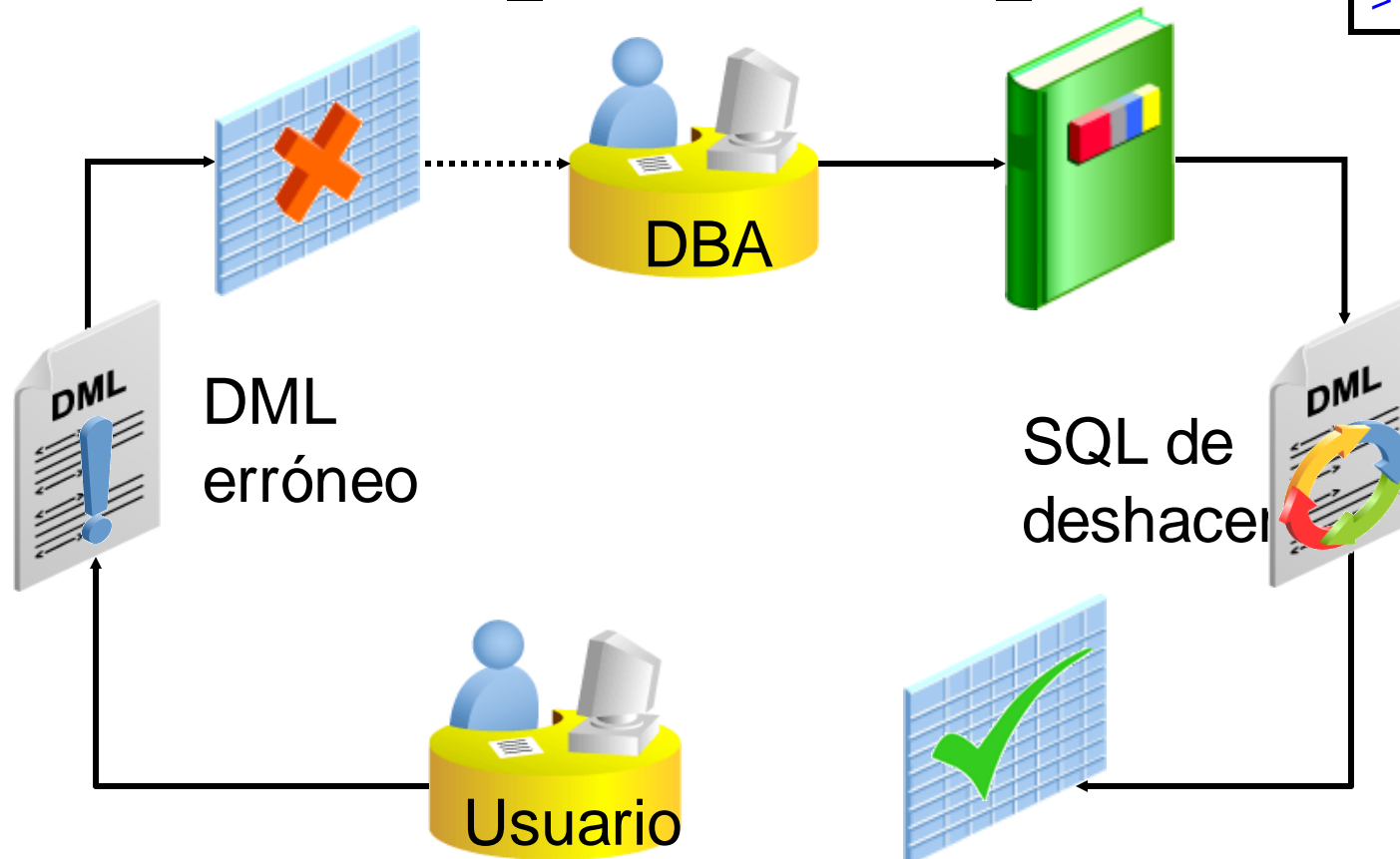
## ***Consulta de Flashback de Versiones: Consideraciones***

- La cláusula `VERSIONS` no se puede utilizar para realizar consultas en:
  - Tablas externas
  - Tablas temporales
  - Tablas fijas
  - Vistas
- La cláusula `VERSIONS` no puede abarcar comandos DDL.
- Las operaciones de reducción de segmentos se filtran.

# Consulta de Flashback de Transacciones: Visión General

Visión General  
Base de Datos  
Tabla  
Borrado  
Consulta  
Versiones  
> Transacción

## FLASHBACK\_TRANSACTION\_QUERY





## ***Consulta de Flashback de Transacciones: Consideraciones***

- Es necesario que previamente se haya activado supplemental logging
  - ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
- Los procedimientos DDL se consideran como actualizaciones de diccionario.
- Los objetos borrados aparecen como números de objeto.
- Los usuarios eliminados aparecen como identificadores de usuario.