

# El component de gestió de les dades

Restauració i reconstrucció

Marta Oliva Solé

P01/11031/00004





# Índex

Introducció .....	5
Objectius .....	6
1. Pèrdua de dades per errors o fallides .....	7
2. Arquitectura del component de gestió de les dades .....	9
2.1. Memòria intermèdia i el seu gestor .....	9
2.1.1. Estructura de dades de la memòria intermèdia .....	9
2.1.2. Operacions del gestor de memòria intermèdia.....	10
2.2. Gestor de recuperació .....	12
2.3. Diari d'operacions.....	13
2.3.1. Contingut del diari.....	13
2.3.2. Protocol d'escriptura avançada al diari .....	14
2.3.3. Implementació del diari .....	14
2.3.4. Punts de control del diari .....	15
3. Restauració .....	17
3.1. Modalitats de gestor de restauració .....	17
3.2. Modalitat prendre/no obligar .....	18
3.3. Modalitat no prendre/no obligar .....	27
3.4. Modalitat prendre/obligar .....	28
3.5. Modalitat no prendre/obligar .....	29
3.6. Restauració aprofitant l'estratègia d'ombreig.....	30
4. Reconstrucció.....	31
Resum .....	33
Activitats .....	35
Exercicis d'autoavaluació .....	35
Solucionari .....	36
Glossari.....	36
Bibliografia.....	37



## Introducció

Una de les funcions dels sistemes de gestió de bases de dades és que l'accés a la base de dades resulti transparent per a l'usuari. En aquest mòdul estudiem el component de gestió de dades com a component encarregat de l'accés físic a la base de dades.

El propòsit del mòdul és descriure els diferents subcomponents del component de gestió de dades i el seu funcionament. Veurem de quina manera la memòria intermèdia i el seu gestor, juntament amb el gestor de recuperació, treballen per a garantir un estat consistent de la base de dades sense pèrdua d'informació. En aquest sentit, treballarem les tècniques de restauració tenint en compte la modalitat del gestor de restauració, i també les tècniques de reconstrucció.

## Objectius

Els materials didàctics associats a aquest mòdul pretenen que l'estudiant assolixi els objectius següents:

1. Prendre consciència dels motius pels quals es podria arribar a tenir una pèrdua total o parcial de les dades emmagatzemades en una base de dades.
2. Conèixer quin component del sistema de gestió de bases de dades és l'encarregat de dur a terme els accessos a la base de dades física, i com es relaciona amb els altres components.
3. Entendre el comportament del component de gestió de les dades, i també dels seus subcomponents.
4. Comprendre en què consisteix el diari d'operacions d'un sistema de gestió de bases de dades.
5. Saber com afecta el comportament del component de gestió de les dades als mecanismes de recuperació (restauració i reconstrucció).
6. Distingir les diferents modalitats de gestor de restauració.
7. Adquirir l'habilitat per a decidir quina modalitat de gestor de recuperació pot resultar més idònia en unes circumstàncies o en unes altres.

## 1. Pèrdua de dades per errors o fallides

Durant la utilització habitual d'una base de dades (BD) es poden donar un conjunt de situacions que provoquen pèrdua de dades emmagatzemades. Les més comunes estan directament relacionades amb el procés d'execució de les transaccions, però altres són conseqüència de fallides de sistema o també d'errors en els dispositius d'emmagatzematge.

Exemples on es produeixen pèrdues d'informació

L'exemple més comú de fallida de sistema és quan l'ordinador es queda sense llum. Malauradament, segur que en més d'una ocasió ens hem quedat sense llum mentre generàvem un document. El primer que fem és pensar en la darrera vegada en què vam guardar la feina i tot seguit fem una estimació de la quantitat d'informació que hem pogut perdre. El mateix passa si ens quedem sense llum mentre s'executa una transacció que fa un ingrés en un dels nostres comptes bancaris. Evidentment la transacció no pot acabar, i ens queda el dubte de quin és el saldo del nostre compte bancari que ha quedat emmagatzemat a la base de dades de l'entitat bancària.

En altres ocasions ens trobem, per exemple, que no hi ha manera de poder llegir un document que havíem guardat prèviament en un disquet perquè està danyat. En aquesta ocasió podem assegurar que hi ha un error de dispositiu d'emmagatzematge. Això mateix pot passar amb qualsevol dels dispositius d'emmagatzematge, siguin del tipus que siguin, que es fan servir per a emmagatzemar una base de dades.

Ja sabeu què és una transacció i també quines són les seves propietats. En aquest apartat veurem com la propietat d'atomicitat i la propietat de definitivitat/durabilitat obliguen que en determinades circumstàncies calgui dur a terme una sèrie d'accions per a garantir que la BD contingui informació correcta:

1) Recordem que el fet que una transacció hagi de ser atòmica implica que tots els canvis que conté hagin de romandre a la BD com una unitat o bé, en cas contrari, no n'hi ha de romandre cap. Sempre que una transacció no pugui acabar, independentment de quin en sigui el motiu (perquè s'hagi avortat voluntàriament, per caiguda de sistema, etc.), caldrà desfer tots els canvis que hagués fet.

2) D'altra banda, la propietat de durabilitat obliga que tots els canvis produïts per una transacció confirmada hagin de ser permanents. Tenint en compte que per qüestions d'eficiència sempre es fa servir memòria intermèdia, i que cada canvi produït a la memòria intermèdia no té per què reproduir-se immediatament també en els dispositius d'emmagatzematge, cal assegurar-se que els canvis produïts per transaccions que han confirmat també s'hagin reflectit físicament. Per aquest motiu, en algunes circumstàncies caldrà refer canvis que només s'havien enregistrat en la memòria intermèdia i que, per diferents causes (per exemple, per una caiguda de sistema), s'han perdut.

Recordeu els conceptes introduïts en els apartats 1 i 3 del mòdul "Transaccions en les bases de dades" de l'assignatura Bases de dades II.

Vegeu què són les transaccions en l'apartat 1 del mòdul "Transaccions en bases de dades" de l'assignatura Bases de dades II.

### Propietats ACID de les transaccions

Les propietats ACID de les transaccions són:

- atomicitat
- consistència
- aïllament
- definitivitat/durabilitat

Les tècniques de desfer i refer reben conjuntament el nom de tècniques de restauració. En canvi, sempre que es produeixin errors en els dispositius d'emmagatzematge caldrà utilitzar tècniques de reconstrucció per a resoldre'ls. Així, doncs, mitjançant les tècniques de restauració es resolen errors derivats de problemes en el programari, mentre que amb les tècniques de reconstrucció se superen errors derivats de problemes en components del maquinari. Per a parlar de manera genèrica tant de les tècniques de restauració com de les tècniques de reconstrucció farem servir el nom de tècniques de recuperació.

En anglès, desfer es coneix per undo i refer per redo.

El component de gestió de les dades, que tenen tots els sistemes de gestió de bases de dades (SGBD), ha de servir qualsevol petició de lectura i/o escriptura a la BD. A més, s'ha d'encarregar d'executar les accions que assegurin que la informació de la BD sigui correcta (restauracions i reconstruccions), de la manera més simple, ràpida i automàtica possible, i evitar, en la mesura que es pugui, la intervenció humana.

Perquè el component de gestió de les dades pugui fer, de manera correcta i automàtica, les accions que assegurin que la BD sigui correcta, cal que les dades s'emmagatzemin de manera redundat. Els elements que ens permeten de guardar la informació de manera que sigui recuperable són, d'una banda, el diari i les còpies de seguretat, i, de l'altra, mantenir una còpia o duplicat idèntic de la BD de manera que quan no sigui possible accedir a la còpia original, es pugui passar a treballar amb el duplicat.

En anglès, diari és log, còpia de seguretat és backup i duplicat és mirroring.

Tot i que ja sabeu què són un diari i una còpia de seguretat, podrem ampliar-ne els nostres coneixements en els apartats següents d'aquest mòdul.

Vegeu què són un diari i una còpia de seguretat a l'apartat 3 del mòdul "Transaccions en les bases de dades" de l'assignatura Bases de dades II.



## 2. Arquitectura del component de gestió de les dades

Ja se us ha introduït una arquitectura funcional que a la vegada inclou una arquitectura per al component de gestió de les dades o, abreujat, gestor de dades. Podeu veure que el gestor de dades consta de tres subcomponents rellevants: la memòria intermèdia (MI), el gestor de memòria intermèdia i el gestor de recuperació. A més, apareix un component indispensable per al funcionament del gestor de dades, el diari d'operacions. En cadascun dels subapartats següents veurem amb més detall què són i com funcionen aquests subcomponents.

Vegeu la representació de l'arquitectura funcional en la figura 6 del mòdul didàctic "Evolució dels models de bases de dades i arquitectura funcional d'un SGBD" d'aquesta assignatura.

En anglès, memòria intermèdia és buffer pool.

### 2.1. Memòria intermèdia i el seu gestor

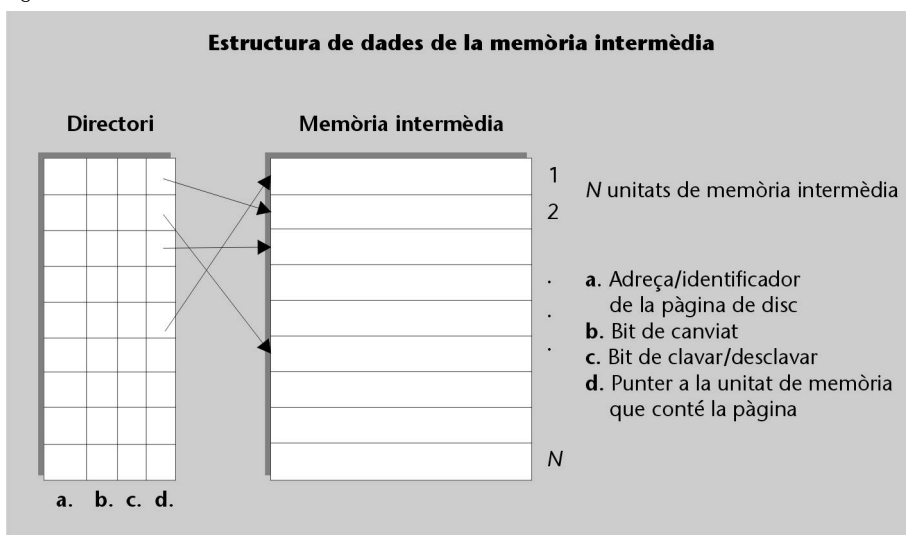
Tots els gestors de dades fan servir memòria intermèdia perquè, com sabeu, permet d'augmentar la velocitat de procés, per tant, es guanya en eficiència. Habitualment, l'encarregat de llegir i escriure a l'MI és el sistema operatiu. Atesa la importància que té, en aquest cas, per a l'eficiència de les tècniques de recuperació, serà l'SGBD mateix, i més concretament, el component per a la gestió de les dades de l'SGBD, que la manejarà mitjançant el gestor d'MI.

Vegeu l'entrada/sortida en un SGBD en el subapartat 3.4 del mòdul "Components d'emmagatzematge d'una base de dades" de l'assignatura Bases de dades II.

#### 2.1.1. Estructura de dades de la memòria intermèdia

La memòria intermèdia que utilitza el gestor de dades està pensada de manera convenient per a la recuperació. Per aquest motiu, la unitat mínima d'informació que hi podem emmagatzemar és una pàgina de disc de la base de dades. Gràcies al fet que es disposa d'un conjunt d'unitats d'MI podem mantenir una col·lecció de pàgines de disc a la memòria principal.

Figura 1



Per a saber quines pàgines de disc hi ha a l'MI disposarem d'un directori. A la figura 1 podem veure com cada entrada del directori conté l'adreça de la pàgina de disc i un apuntador a la posició de la unitat d'MI en què la pàgina de disc està guardada. A més, associats a cada unitat d'MI hi tindrem:

a) Un bit que ens servirà per a indicar si el contingut de la unitat d'MI s'ha modificat. L'anomenarem bit de canviat. Si no es produeix cap modificació el bit restarà a 0, i el posarem a 1 en cas que hi hagi hagut canvis.

b) Un altre bit que ens permetrà de mantenir la pàgina a l'MI, com si la clavéssim per a subjectar-la, sense que es descarregui fins que nosaltres ho vulguem. Aquest bit rep el nom de bit de clavar/desclavar, pel nom anglès pin/unpin. Posarem el bit a 1 sempre que vulguem clavar la pàgina a MI i el posarem a 0 quan la vulguem desclavar.

#### Dirty bit

En anglès, el bit de canviat s'anomena dirty bit perquè es considera que la pàgina física és bruta (no vàlida) i cal canviar-la, atès que s'ha fet una modificació.

### 2.1.2. Operacions del gestor de memòria intermèdia

El gestor de l'MI ofereix les dues operacions necessàries, d'una banda, per a llegir una pàgina de la BD i carregar-la a la memòria intermèdia, i de l'altra, per a emmagatzemar el contingut d'una pàgina, que està a la memòria intermèdia, a la BD. Ambdues operacions reben el nom, respectivament, de recollir, o portar, i descarregar.

Els mots recollir, o portar, i descarregar vénen dels termes anglesos fetch i flush, respectivament.

En el procediment corresponent a l'operació descarregar, que podem veure a continuació, cal indicar l'identificador de la pàgina que es vol treure de la memòria intermèdia.

```

procediment descarregar(p: id_pagina)
var w: valor_pagina;

    si canviat(p) llavors
        w := contingut(p);
        escriure_pagina(p, w);
        canviat(p) := 'no';
    fsi
    actualitzar_directori(p, 0);
fprocediment

```

El primer que fa és assegurar-se que la pàgina resti emmagatzemada correctament en el dispositiu on està emmagatzemada la BD. Així, doncs, en cas que el bit de canviat indiqui que la pàgina ha estat modificada (posat a 1), es fa una escriptura de pàgina indicant que a partir d'aquell moment la pàgina ja no és

canviada (coincideixen tant la còpia a l'MI com la còpia en el dispositiu d'emmagatzematge). Per acabar el procediment, s'actualitza l'entrada del directori, corresponent a la unitat d'MI que contenia la pàgina que s'ha descarregat, de manera que indiqui que en aquella unitat d'MI no hi ha cap pàgina de la BD (l'identificador de la pàgina es posa a 0). A partir d'aquest moment, la unitat d'MI està disponible per a recollir-hi qualsevol altra pàgina.


L'operació recollir procedeix de la manera següent:

```

procediment recollir(p: id_pagina)
var w: valor_pagina

    si buscar(0) = 'no' llavors
        <escollir una unitat d'MI segons LRU o bé FIFO>;
        descarregar(p);
    fsi
    llegir_pagina(p, w);
    actualitzar_directori(0, p);
    contingut(p) := w;
fprocediment
  
```

En primer lloc, es busca una unitat d'MI que estigui lliure, tenint en compte que les unitats d'MI que estan lliures tenen un 0 com a identificador de pàgina. En cas que no n'hi hagi cap de disponible, s'escull, d'entre les unitats d'MI que no contenen pàgines clavades, una unitat que es pugui descarregar (es pot aplicar la tècnica LRU o bé la tècnica FIFO). Un cop es disposa d'una unitat d'MI lliure, es passa a llegir físicament la pàgina corresponent a l'identificador de pàgina que s'ha indicat. A continuació, s'actualitza l'entrada del directori corresponent a una unitat d'MI lliure perquè hi consti l'identificador de la pàgina que s'acaba de llegir, i finalment, el contingut de la pàgina s'emmagatzema a la unitat d'MI.

Fixeu-vos que sempre que es recull una pàgina, cal fer una lectura a la BD; en canvi, quan es descarrega una pàgina, no sempre cal fer una escriptura a la BD. 

També heu de tenir en compte que quan es descarrega una pàgina, la descàrrega es pot fer seguint dues estratègies diferents depenent del gestor de dades. Hi ha gestors de dades que mantenen més d'una versió de cada pàgina en el dispositiu d'emmagatzematge, mentre altres només en guarden una.

Si el gestor de dades disposa d'una única versió de cada pàgina, cada actualització sobrescriu la pàgina, que s'emmagatzema sempre a la mateixa ubicació del disc original, de manera que es perd la informació que contenia abans del canvi. Aquesta manera d'emmagatzemar les actualitzacions rep el nom

#### Les tècniques d'LRU i FIFO

La sigla LRU prové de l'expressió anglesa Least Recently Used (la menys utilitzada recentment).

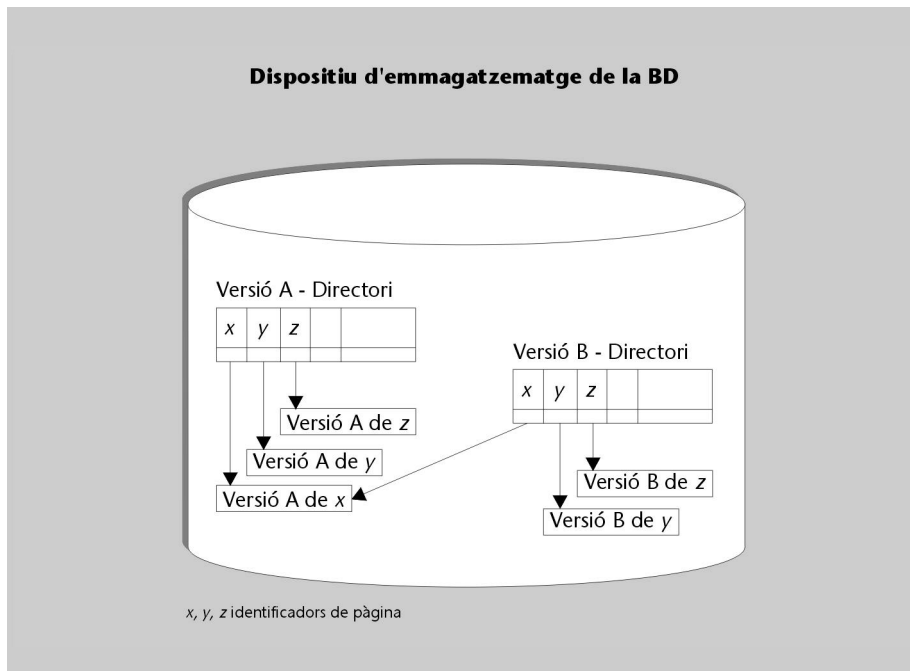
La sigla FIFO prové de l'anglès First-In-First-Out (la primera que entra és la primera que surt).

d'actualització al lloc. Si el gestor de dades manté diferents versions d'una mateixa pàgina, aleshores no cal que per cada actualització se sobrescrigui sobre la versió antiga de la pàgina; es pot buscar una posició nova en el dispositiu d'emmagatzematge. Aquesta tècnica rep el nom de tècnica d'ombreig, atès que a cada versió antiga d'una pàgina se l'anomena ombra. La majoria de gestors actuals utilitzen l'estratègia d'actualització al lloc.

L'estratègia d'actualització al lloc en anglès és in-place updating.

La tècnica d'ombreig en anglès és shadowing i ombra és shadow.

Figura 2



Per a l'estratègia d'ombreig cal disposar d'un directori amb tantes posicions com pàgines tingui la BD. Cada entrada del directori conté l'identificador de la pàgina i la seva ubicació física en el dispositiu d'emmagatzematge. Amb cadascun d'aquests directoris es defineix un estat de la BD. Com podeu veure a la figura 2, generalment es mantenen diversos d'aquests directoris, de manera que es disposa de diferents estats de la BD. Més endavant veurem com el fet d'utilitzar aquesta estratègia pot facilitar les tasques del gestor de recuperació. ⚠


## 2.2. Gestor de recuperació

El gestor de recuperació és el subcomponent principal del component de gestió de les dades de qualsevol SGBD. Inclou tant el gestor de restauració com el gestor de reconstrucció. El gestor de restauració és el responsable de les tècniques de restauració, i el gestor de reconstrucció és el responsable de les tècniques de reconstrucció.

En anglès, gestor de recuperació s'anomena recovery manager.

Atesa la importància de cadascun d'aquests, dedicarem l'apartat 3 i 4 d'aquest mòdul a la restauració per part del gestor de restauració, i a la reconstrucció de què és responsable el gestor de reconstrucció, respectivament. ⚠

## 2.3. Diari d'operacions

El diari d'operacions és un component que tot i que no forma part del gestor de dades, és indispensable per al seu funcionament, sempre i quan el gestor de dades utilitzi l'estratègia d'actualització al lloc per a fer la descàrrega de pàgines. 

En anglès, diari s'anomena log o journal.

El diari d'operacions té com a objectiu emmagatzemar la informació dels canvis de manera convenient perquè el gestor de recuperació pugui recuperar un estat consistent de la BD.


### 2.3.1. Contingut del diari

Conceptualment, el diari és una representació de la història d'execució de les transaccions. És imprescindible que s'emmagatzemi una anotació al diari, com a mínim, sempre que una transacció confirmi, avorti o modifiqui alguna dada.

Cada anotació, independentment del tipus d'operació a què correspongui, ha de contenir la informació següent: l'identificador de la transacció a què pertany l'operació, el tipus d'operació i un punter a l'anotació, guardada anteriorment, corresponent a l'operació anterior de la mateixa transacció. A més, com veurem a continuació, cal guardar informació addicional per a l'operació de modificació.

Consultant el diari hem de ser capaços de determinar quin és el darrer valor d'una dada que s'ha confirmat. Hi ha dues maneres possibles d'emmagatzemar aquesta informació al diari:

- 1) Es fa constar el valor que s'ha escrit a la BD en cada anotació corresponent a una modificació; aleshores es disposa d'un diari físic.
- 2) En lloc de guardar el valor, es guarda la descripció de les operacions d'alt nivell que s'han dut a terme sobre les dades; per tant, es passa a tenir un diari lògic.

El diari lògic requereix menys anotacions per a enregistrar els canvis, i això implica que el gestor de recuperació pot millorar el seu rendiment. Per contra, la recuperació des d'un diari lògic acostuma a ser més complexa. Per aquest motiu, i atès que resulta més usual que els gestors utilitzin diaris físics, a partir d'ara només ens ocuparem dels diaris físics. 

Tot diari físic ha de disposar de la informació necessària per a desfer canvis provocats per transaccions que no han confirmat. Aquesta informació correspon al valor que tenia la dada abans que la transacció la modifiqués, d'aquí el nom d'estat d'abans del canvi.

Exemple d'anotació en un diari lògic

Una anotació en un diari lògic pot ser:  
"inserció del registre R al fitxer F i actualització de l'índex de F per a reflectir la inserció".

El diari físic també ha de contenir la informació necessària per a refer els canvis de transaccions confirmades que no s'han arribat a guardar a la BD. En aquest cas la informació correspon al valor que la dada ha adquirit amb la modificació i s'anomena estat de després del canvi.

L'estat d'abans i l'estat de després del canvi es coneixen en anglès com *before image* i *after image*.

Així, doncs, a part de la informació comuna a la resta d'anotacions que es guarden al diari, tota anotació corresponent a una operació de modificació també inclou l'identificador de la pàgina que s'ha modificat, l'estat d'abans i l'estat de després.

### 2.3.2. Protocol d'escriptura avançada al diari

Perquè les anotacions al diari siguin correctes, el gestor de recuperació segueix el protocol d'escriptura avançada al diari.

El protocol d'escriptura avançada al diari assegura que l'estat d'abans del canvi d'una pàgina s'enregistri al diari abans que se sobrescrigui la pàgina amb el valor que serà l'estat de després.

En anglès, el protocol d'escriptura avançada es coneix per *Write-Ahead Log Protocol (WAL)*.

El mateix protocol no deixa que es completi l'operació de confirmació d'una transacció fins que el diari contingui totes les anotacions corresponents a totes les operacions de la transacció.

### 2.3.3. Implementació del diari

Físicament, el diari és un fitxer auxiliar on es va anotant la informació de tots els canvis produïts per les transaccions que s'han efectuat a la BD.

Atesa la importància que representa el fet de mantenir l'ordre en què es produeixen els canvis, la millor opció d'emmagatzematge és utilitzar un fitxer seqüencial. Aparentment, el fitxer seqüencial que implementa el diari disposa de capacitat il·limitada. Atès que cap dispositiu d'emmagatzematge disposa de capacitat il·limitada, s'assigna al fitxer una capacitat limitada però s'afegeix un enllaç des del final del fitxer al principi, de manera que un cop s'arribi al final del fitxer, es continuïn afegint les anotacions pel principi.

Recordeu els fitxers seqüencials en el mòdul didàctic "Fitxers seqüencials" de l'assignatura Bases de dades I.

A més, per a assegurar la disponibilitat del diari (tots els fitxers són susceptibles de tenir errors), se n'acostumen a emmagatzemar dues còpies, a la vegada que se'n fan còpies de seguretat periòdicament.

### 2.3.4. Punts de control del diari

El fet d'escriure anotacions al diari de manera continuada, i que el fitxer seqüencial que implementa el diari se sobrescriu periòdicament, ens porten a plantejar-nos les reflexions següents:

- a) Si les anotacions se sobrescriuen quan s'ha superat la capacitat que s'ha destinat al fitxer que implementa el diari, què passa amb la informació que es perd?
- b) Si s'emmagatzemen les anotacions al diari des de l'inici de funcionament de la BD, en cas que calgui recuperar un estat correcte o consistent de la BD, quanta feina haurà de fer el gestor de dades i quant de temps li portarà fer-la?

Els problemes que s'apunten amb aquestes preguntes se solucionen gràcies als mètodes de punts de control del diari.

El punt de control en anglès s'anomena checkpoint.

Una aplicació d'un punt de control (PC) consisteix en un mètode que obliga que un estat consistent de la BD s'emmagatzemi físicament, i a la vegada permet de desestimar les anotacions del diari fetes amb anterioritat al punt de control.

L'aplicació dels PC és automàtica i periòdica –aproximadament cada quinze minuts–, però la periodicitat d'aplicació es pot establir mitjançant el gestor de recuperació.

En aplicar un PC s'esdevenen les actuacions següents:

- 1) Suspensió temporal de l'execució de les transaccions. Amb aquesta actuació no es deixa que s'executin transaccions noves tot i que s'espera a que s'acabin d'executar les transaccions actives.
- 2) Descàrrega forçada de totes les unitats d'MI que s'han modificat (bit de canviat a 1) cap als dispositius d'emmagatzematge que emmagatzemen la BD.
- 3) Escripura d'un registre de PC al diari perquè quedi constància que s'ha guardat un estat consistent a la BD; per tant, es poden desestimar les anotacions del diari prèvies a aquest registre de PC.
- 4) Represa de l'execució de transaccions noves.

Fixeu-vos que gràcies a aquest procediment d'aplicació de PC, s'aconsegueix que la BD contingui el darrer valor confirmat de cada element afectat per les transaccions que s'executaven quan s'ha aplicat el PC. Per aquest motiu, el procediment rep el nom d'aplicació de PC consistent a la confirmació. També es

En anglès, l'aplicació de PC consistent a la confirmació es coneix per commit consistent checkpointing.

possible de donar al mateix procediment el nom de PC a transaccions acabades, atès que s'espera que les transaccions actives s'acabin d'executar per a fer la descàrrega de l'MI.

L'aplicació de PC consistent a la confirmació té l'inconvenient que en alguns sistemes, sobretot si es tracta de sistemes molt carregats, el temps d'espera (durant el qual el sistema queda inoperatiu) des que es deixa d'acceptar l'execució de transaccions noves fins que aquesta es reprèn, és massa llarg. Per aquest motiu, hi ha sistemes que en lloc d'aplicar els PC a transaccions acabades apliquen PC a accions acabades.

L'aplicació de PC a accions acabades consisteix a executar el mateix conjunt d'actuacions que l'aplicació de PC a transaccions acabades, però en lloc d'esperar que totes les transaccions actives s'acabin d'executar per a poder emmagatzemar un estat consistent de la BD, s'espera que s'acabin d'executar les accions actives de les transaccions que s'executaven a l'inici de l'aplicació del PC. Aquest nou procediment també rep el nom d'aplicació de PC consistent amb l'MI.

Cal que tingueu en compte que el fet d'aplicar un PC a accions acabades implica que es puguin descarregar unitats d'MI que hagin estat modificades per transaccions que encara no havien confirmat. En aquest cas, no es pot desestimar la informació emmagatzemada al diari prèviament a l'aplicació del PC, sinó que en determinades situacions caldrà recórrer a les anotacions anteriors a un PC per a poder desfer tots els canvis produïts per una transacció que, en lloc de confirmar, hagi avortat.

El registre de PC que s'emmagatzema al diari en cas de PC a accions acabades, inclou una llista de totes les transaccions que hi havia actives en el moment d'aplicar el PC. Aquesta informació permet de determinar el punt del diari a partir del qual les anotacions ja no són rellevants.

En els apartats següents d'aquest mòdul didàctic podrem aprendre com el gestor de restauració i el gestor de reconstrucció utilitzen el diari per a poder restaurar o bé reconstruir la BD quan calgui.

#### Acció

Recordeu que anomenem acció l'operació elemental sobre la BD que podem trobar en una transacció.

En anglès, l'aplicació de PC consistent amb l'MI es coneix per *buffer pool consistent checkpointing*.



### 3. Restauració

El gestor de restauració, inclòs en el gestor de recuperació, és l'encarregat de rebre i d'executar les operacions enviades pel gestor de concurrència. L'execució de cada operació es fa de manera convenient perquè siguin aplicables, quan calgui, les tècniques de restauració.

En concret, les úniques operacions que el gestor de restauració ofereix al gestor de concurrència són: llegir, escriure, confirmar i avortar. A part, disposa de l'operació reiniciar, que s'executa de manera automàtica sempre que es reinicia el sistema, per exemple, després d'una fallida.

El gestor de restauració executa cadascuna de les operacions de manera atòmica. Utilitza l'MI, mitjançant el gestor d'MI, per aconseguir una eficiència més gran tant en les lectures com en les escriptures, però no tots els gestors de restauració utilitzen l'MI de la mateixa manera.

#### 3.1. Modalitats de gestor de restauració

Segons la manera com les unitats de l'MI s'assignen a les transaccions, i depenent del moment en què les pàgines emmagatzemades es descarreguen a les unitats d'MI, es distingeixen diferents modalitats de gestor de restauració:

1) En el cas que el gestor de restauració comparteixi l'MI entre totes les transaccions actives, de manera que cada unitat d'MI es vagi assignant a qualsevol transacció just en el moment que la necessiti, el gestor de restauració haurà de treballar segons la modalitat prendre. Aquesta modalitat de gestor rep aquest nom perquè qualsevol transacció es deixa prendre qualsevol de les unitats d'MI que té assignades quan una altra transacció les necessita, tot i que la transacció encara estigui activa.

2) D'altra banda, el gestor de restauració pot mantenir (clavar) totes les pàgines que una transacció determinada necessita a l'MI sense necessitat de recollir-les i/o descarregar-les des de l'inici de la transacció fins al final. En aquest cas, la modalitat del gestor passa a ser la modalitat no prendre, ja que cap altra transacció pot disposar de les unitats d'MI ocupades per una altra transacció activa.

3) Si es dona el cas que el gestor de restauració força una descàrrega de totes les unitats d'MI assignades a una transacció determinada just quan aquesta confirma, aleshores la modalitat del gestor es la modalitat obligar.

! Vegeu la ubicació del gestor de concurrència en la representació de l'arquitectura funcional en la figura 6 del mòdul "Evolució dels models de bases de dades..." d'aquesta assignatura.

Enviament d'operacions al gestor de dades

Recordeu que el gestor de concurrència envia les operacions al gestor de dades de manera que es preservi, com a mínim, la seriabilitat de les transaccions que s'executen concurrentment.

Terminologia anglesa

En anglès, les operacions llegir, escriure, confirmar, avortar i reiniciar s'anomenen read, write, commit, abort i restart, respectivament.

Assignació de les unitats d'MI a les transaccions

Les unitats d'MI assignades a una transacció es poden conèixer, si es recorre en ordre el diari. Una altra opció consisteix a fer que el gestor de restauració mantingui una llista per cada transacció d'unitats assignades.

Els noms de les modalitats prendre/no prendre venen dels noms anglesos steal/no steal.

4) Quan el gestor de restauració no força la descàrrega de les unitats d'MI assignades a una transacció que confirma, la modalitat del gestor és no obligar. En aquesta modalitat, la descàrrega de cada unitat d'MI es fa en el moment en què aquesta es necessita per a recollir-hi una pàgina nova.

En anglès, obligar/no obligar  
són force/no force.

Cal que tingueu en compte les consideracions següents: 

a) En la modalitat no prendre és impossible que es descarregui una unitat d'MI que alguna transacció hagués modificat però que encara no hagi confirmat. Aquesta característica assegura que en cap moment no serà necessari desfer canvis.

b) El fet de compartir les unitats d'MI entre diferents transaccions, com es fa en la modalitat prendre, implica que abans que una transacció reculli una pàgina nova sobre una unitat d'MI, s'hagi de descarregar la pàgina que hi estava recollida. Sempre que un gestor de restauració utilitzi la modalitat prendre, cal tenir previst que en qualsevol moment sigui necessari desfer canvis produïts per transaccions que al final no han pogut confirmar i que s'hagin emmagatzemat físicament a la BD.

c) En obligar que es descarreguin les unitats d'MI, ens assurem que els canvis produïts per les transaccions confirmades estiguin reflectits físicament en el moment de la confirmació. Per aquest motiu podem assegurar que mai no serà necessari refer cap canvi si el gestor utilitza la modalitat obligar.

d) Si el gestor de restauració no obliga la descàrrega de les unitats d'MI modificades per una transacció confirmada, es poden perdre canvis si, per exemple, es produeix una fallida de sistema. En la modalitat no obligar de gestor cal preveure la necessitat d'haver de refer canvis que s'hagin perdut.


e) En la modalitat no prendre es requereix un nombre d'unitats d'MI superior que en la modalitat prendre. Cal tenir en compte que en la modalitat no prendre cal que l'MI disposi de tantes unitats com pàgines diferents pugui arribar a modificar una transacció, com a mínim.

f) La modalitat no obligar deixa que la descàrrega de les unitats d'MI es faci segons les necessitats que hi hagi de recollida de pàgines noves. Per tant, s'eliminen les descàrregues innecessàries que poden arribar a ser provocades per la modalitat obligar, tenint en compte que a la vegada s'aconsegueix de minimitzar el nombre d'entrades/sortides. Recordeu que es pot donar la circumstància que en començar-se a executar una transacció, necessiti les pàgines que ha modificat una transacció que acaba de confirmar.

### 3.2. Modalitat prendre/no obligar

Els gestors de restauració que funcionen segons la modalitat prendre/no obligar, malgrat que requereixen tant dels mecanismes per desfer com dels mecanismes

per refer, són els que s'utilitzen amb més freqüència en els SGBD. Tot i la complexitat que representa amb vista al gestor de restauració, aquesta modalitat, resulta ser la que minimitza més bé els recursos, tant pel que fa a la quantitat de memòria destinada a MI, com al nombre d'entrades/sortides que genera en recollir i descarregar les pàgines físiques.

Com ja hem anat anotant al llarg d'aquest mòdul, donem per fet que el gestor d'MI fa la descàrrega de pàgines segons l'estratègia d'actualitzar al lloc. També cal tenir en compte que utilitzarem diari físic. A més, considerarem que els PC s'aplicaran al diari de manera consistent amb l'MI, és a dir, a accions acabades. 

A continuació veurem com han d'estar implementades les operacions que el gestor de restauració ofereix al gestor de concurrència. La implementació ha de ser coherent amb la modalitat perquè així es puguin aplicar les tècniques de restauració sense problemes. En concret, veurem les operacions llegir, escriure, confirmar, avortar i reiniciar.

Com podem veure a continuació en el procediment corresponent a l'operació llegir, només cal obtenir el contingut de la pàgina després d'assegurar-nos que està recollida en una unitat d'MI.

```
procediment llegir(t: id_transaccio, p: id_pagina, v: valor_pagina)
  si buscar(p) = 'no' llavors
    recollir(p);
  fsi
  v := contingut(p);
fprocediment
```

En el procediment per a l'operació escriure podem observar que, després d'assegurar-nos que la pàgina estigui recollida a l'MI, i abans que es modifiqui la unitat d'MI corresponent a aquesta pàgina, al diari s'escriu l'anotació corresponent a una modificació, d'acord amb el protocol d'escriptura avançada al diari. Fixeu-vos que l'anotació al diari inclou el tipus d'anotació ('m'), l'identificador de la transacció a què correspon l'operació (t), l'identificador de la pàgina que es modifica (p), l'estat d'abans del canvi (w) i, finalment, l'estat de després del canvi (v). Recordeu que el gestor també hi afegeix el punter a l'anotació anterior de la mateixa transacció.

```
procediment escriure (t: id_transaccio,
                    p: id_pagina, v: valor_pagina)
  var w: valor_pagina
```

```

si buscar(p) = 'no' llavors
    recollir(p);
fsi
w := contingut(p);
escriure_diari('m', t, p, w, v);
contingut(p) := v;
canviat(p) := 'si';
fprocediment

```

Fixeu-vos que després de modificar el contingut d'una unitat d'MI sempre s'actualitza el bit de canviat corresponent per a aconseguir emmagatzemar físicament la pàgina en el moment en què aquesta es descarregui.

Atès que en modalitat no obligar no s'han de descarregar les pàgines modificades per la transacció que confirma, l'operació confirmar només ha d'anotar en el diari que la transacció (t) ha confirmat ('c').

```

procediment confirmar(t: id_transaccio)
    escriure_diari('c', t);
fprocediment

```

Quan una transacció s'ha d'avortar perquè el gestor de restauració rep l'operació avortar des del gestor de concurrència, cal desfer tots els canvis que hagués generat, ja que per la modalitat prendre, alguns podrien estar reflectits físicament.

Així, el procediment avortar ha de recórrer el diari en sentit contrari, cap enrere fins a la darrera anotació de la transacció (que és la primera que es va anotar), de manera que es puguin restaurar tots els estats d'abans del canvi de les pàgines modificades per la transacció avortada. Per cada anotació de modificació ('m') cal assegurar-se que la pàgina (p) estigui recollida a l'MI i després, cal modificar el contingut de la unitat d'MI que la recull amb l'estat d'abans del canvi ('w') llegit del diari. És indispensable posar el bit de canviat corresponent a la unitat d'MI que conté la pàgina (p) a 1, per tal que en el moment que es descarregui la pàgina el seu valor s'emmagatzemi físicament. Finalment, el procediment anota en el diari que la transacció (t) ha avortat ('a').

```

procediment avortar(t: id_transaccio)
var p: id_pagina;
    w: valor_pagina;

    llegir_diari_enrere('m', t, p, w);
    mentre quedin_registres_al_diari('m', t) fer
        si buscar(p) = 'no' llavors recollir(p) fsi

```

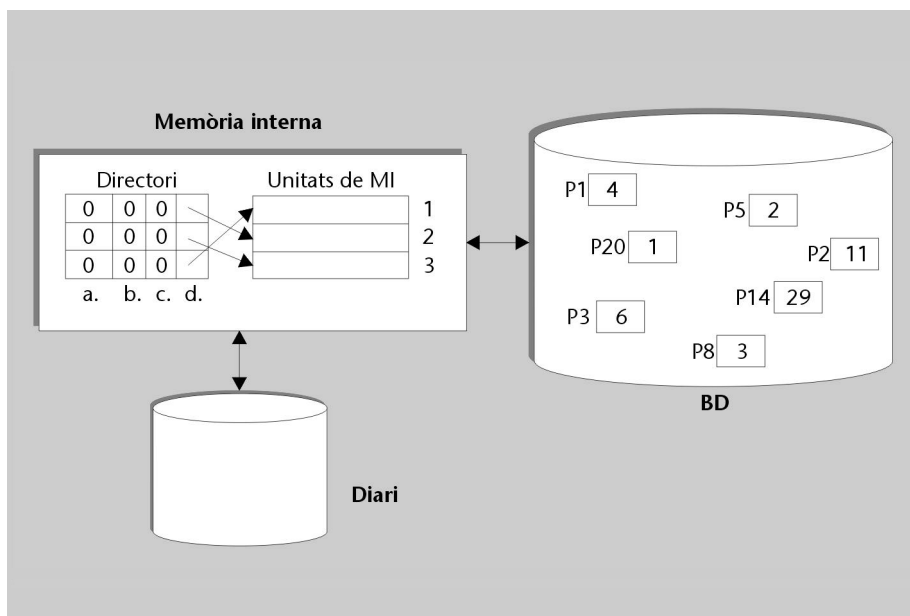
```

contingut(p) := w;
canviat(p) := 'si'
llegir_diari_enrere('m', t, p, w);
fmentre
  escriure_diari('a', t);
fprocediment

```

Vegem mitjançant un exemple que implica l'execució de cadascuna de les operacions que el gestor de restauració ofereix al gestor de concurrència en la modalitat prendre/no obligar. A la figura 3 podem observar un estat de la BD, de l'MI i del diari. Atès que partim d'un estat inicial, suposem que l'MI i el diari estan buits.

Figura 3

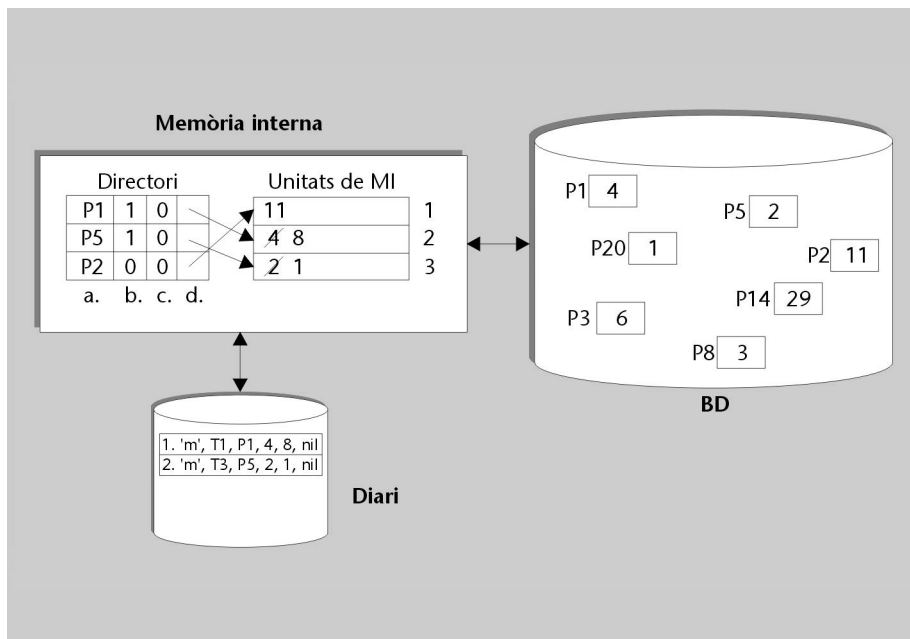


El gestor de restauració rep les operacions següents consecutivament:

- |                       |                        |                       |
|-----------------------|------------------------|-----------------------|
| a. llegir(T1,P1,v1),  | i. llegir(T3,P8,v3)    | q. llegir(T5,P14,v5)  |
| b. escriure(T1,P1,8)  | j. escriure(T3,P8,17)  | r. escriure(T5,P14,7) |
| c. llegir(T3,P5, v3)  | k. llegir(T2,P2,v2)    | s. llegir(T3,P3,v3)   |
| d. escriure(T3,P5,1)  | l. escriure(T2,P2,6)   | t. escriure(T3,P3,18) |
| e. llegir(T1,P2,v1)   | m. llegir(T4,P20,v4)   | u. llegir(T4,P1,v4)   |
| f. llegir(T2,P3,v2)   | n. escriure(T4,P20,50) | v. llegir(T6,P2,v6)   |
| g. escriure(T2,P3,29) | o. confirmar(T2)       | w. escriure(T6,P2,32) |
| h. avortar(T1)        | p. aplic_autom_PC      | x. confirmar(T6)      |

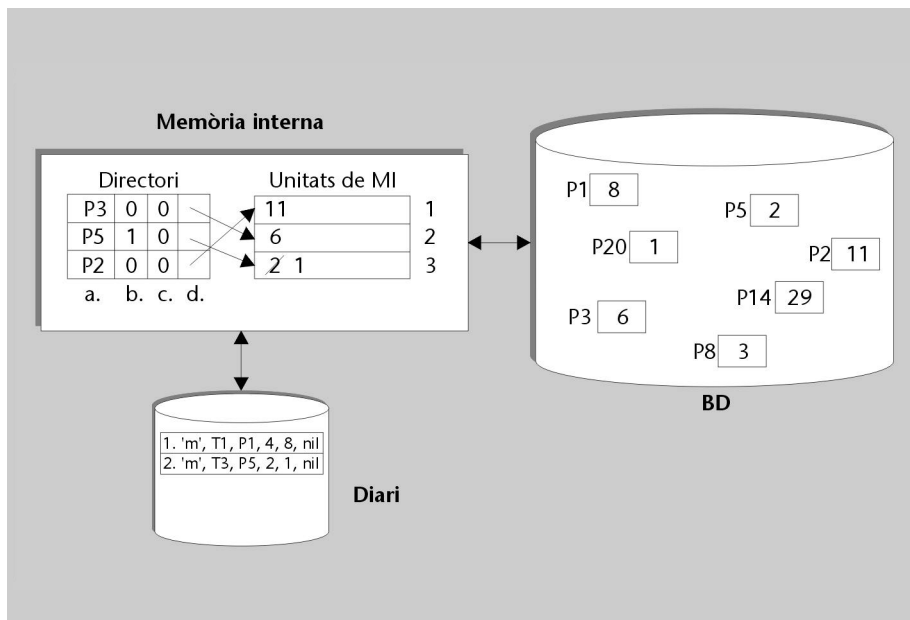
A la figura 4 podem veure com s'han anat ocupant les diferents unitats d'MI i quines han estat les anotacions al directori, tenint en compte que el gestor de restauració ha executat des de l'operació a fins l'operació e.

Figura 4



Arribat aquest punt, per a poder executar l'operació f, que requereix recollir la pàgina P3, cal descarregar una de les unitats d'MI. Si tenim en compte que la modalitat del gestor és prendre i apliquem la tècnica FIFO, la unitat que cal descarregar és la que conté la pàgina P1. Atès que la pàgina ha estat modificada, s'emmagatzemarà físicament al lloc que té assignat. Podem observar el resultat de tot el conjunt després d'haver executat l'operació f a la figura 5.

Figura 5

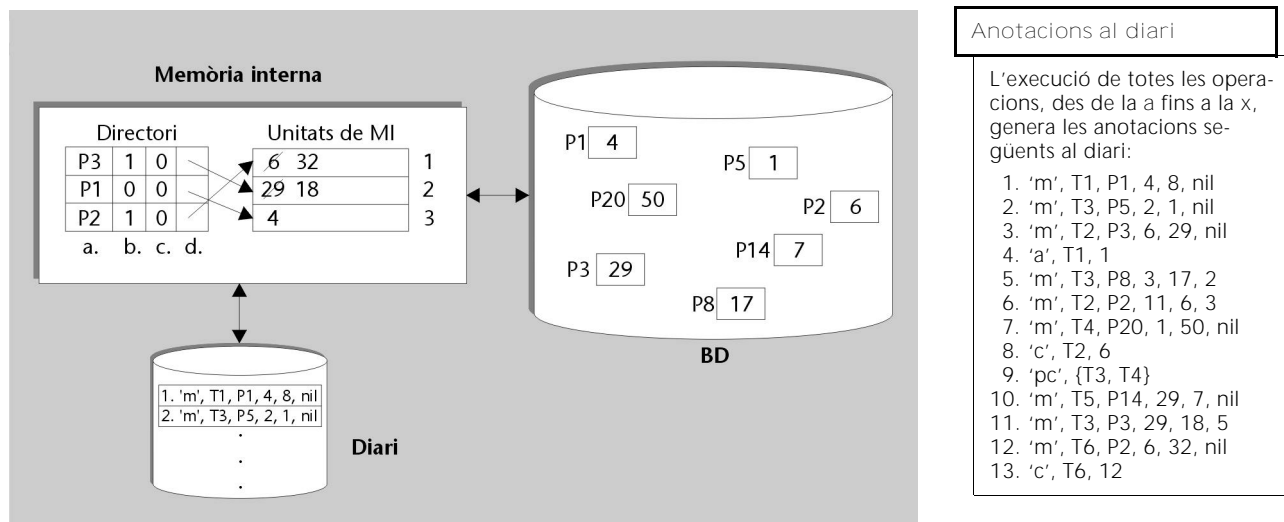


### Activitat

Simuleu l'execució de la resta d'operacions, des de l'operació g fins a l'operació x, tenint en compte l'operació p, tal com s'indica, s'aplica de manera automàtica un punt de control a accions acabades. Recordeu que quan una transacció confirma, no s'ha de descarregar les unitats d'MI que tingui assignades (modalitat no obligar).

A la figura 6 podeu comprovar l'estat de tot el conjunt després d'haver aplicat totes les operacions. Totes les anotacions que ha de contenir el diari estan llistades al marge.

Figura 6



L'operació reiniciar s'encarrega de restaurar un estat consistent (correcte) de la BD després d'una fallida de sistema. Cal que desfaci els canvis produïts per transaccions que no han confirmat i que s'havien emmagatzemat a la BD, i cal que refaci els canvis produïts per transaccions que s'han perdut tot i haver confirmat.

Atesa la seva complexitat, veurem el procediment reiniciar en dos nivells: el primer nivell ens descriu les accions principals que s'han de fer, i el segon nivell aprofundeix en el detall de cadascuna de les accions.

```

procediment reiniciar()
var LC: llista_txs_confirmades; LA: llista_txs_avortades; p: id_pagina;
w: estat_abans_canvi_pagina, v: estat_despres_canvi_pagina;
t: id_transaccio; r: tipus_anotacio; d: posicio_diari;

inicialitzar_MI_i_llistes_transaccions();
fase_desfer_fins_darrer_PC();
fase_desfer_complementaria();
fase_refer();
anotar_PC();
fprocediment

```

Quant al primer nivell, podem veure que el procediment reiniciar consta de cinc accions:

1) Acció inicialitzar\_MI\_i\_llistes\_transaccions: s'inicialitza l'MI per a aconseguir que en cap unitat d'MI hi resti informació d'abans de la fallida. També s'inicien

a buit les llistes de transaccions que han confirmat i transaccions que han avortat, atès que són essencials per al procediment reiniciar.

2) Acció fase\_desfer\_fins\_darrer\_PC: s'encarrega de desfer tots els canvis produïts per transaccions que encara no havien confirmat en el moment en què s'ha produït la fallida, o bé per transaccions que s'han avortat voluntàriament. L'acció només tracta les anotacions des del final del diari fins al darrer PC. Durant el recorregut del diari, l'acció dedueix les transaccions que havien confirmat i les que no ho havien pogut fer.

3) Acció fase\_desfer\_complementaria: es complementa la fase desfer de l'acció anterior. En llegir l'anotació corresponent al PC s'obté la llista de les transaccions que hi havia actives en el moment de l'aplicació del PC. A partir d'aquesta llista i de la llista de transaccions avortades que s'ha anat construint, es dedueix el conjunt de transaccions que no s'han pogut acabar d'executar. Aquest conjunt de transaccions és el que es desfà gràcies a aquesta fase complementària.

4) Acció fase\_refer: l'acció corresponent a la fase refer ha de refer des de l'últim PC aplicat, tots els canvis produïts per les transaccions que havien confirmat abans que es produís la fallida.

5) Acció anotar\_PC: la darrera tasca en el procediment reiniciar consisteix a registrar una anotació de PC al diari, de manera que quedi constància que en aquell punt la BD és consistent.

A continuació fem l'estudi en el segon nivell:

1) L'acció inicialitzar\_MI\_i\_llistes\_transaccions actualitza cada unitat d'MI per aconseguir que l'MI estigui buida. També inicialitza a buit la llista de transaccions que han confirmat (LC) i la llista de transaccions que han avortat (LA).

```
accio
inicialitzar_MI_i_llistes_transaccions()
  per cada unitat_MI(p) fer
    canviat(p) := 'no';
    actualitzar_directori(p, 0);
  fper
  LC := Ø; LA := Ø;
faccio
```

2) L'acció fase\_desfer\_fins\_darrer\_PC llegeix enrere en el diari fins que arriba a l'últim PC. Per cada anotació, analitza si correspon a una anotació de confirmació o d'avortament, i construeix les llistes LC i LA. Si l'anotació corres-



pon a una modificació, aleshores desfà el canvi en el cas que pertanyi a una transacció no confirmada.

```

accio fase_desfer_fins_darrer_PC()
  llegir_diari_enrere(r, t, p, w);
  mentre r ≠ 'pc' fer
    opcio
      cas r = 'c' fer LC := LC ∪ t fcas
      cas r = 'a' fer LA := LA ∪ t fcas
      cas r = 'm' fer
        si (t ∉ LC ∧ t ∉ LA) llavors
          si (t ∉ LA) llavors LA := LA ∪ t fsi;
          escriure_bd(p, w);
          si ultima_annotacio(t) llavors LA := LA - t fsi
        fsi
      fcas
    fopcio
  llegir_diari_enrere(r, t, p, w);
  fmentre
  d := posicio_actual_diari;
faccio

```

Al final de l'acció fase\_desfer\_fins\_darrer\_PC s'enregistra en una variable (d) la posició de l'últim PC en el diari per a ser utilitzada en la fase refer.

3) En l'acció fase\_desfer\_complementaria, en primer lloc es completa la llista de transaccions avortades amb l'identificador de les transaccions actives en el moment d'aplicació del PC que no figurin ni a LC ni a LA. A partir d'aquí es llegeix el diari enrere i, en cas que l'annotació de modificació llegida correspongui a alguna de les transaccions que encara resten a LA, es desfà el canvi reflectit en l'annotació.

```

accio fase_desfer_complementaria()
var lta: llista_transaccions_actives;

  llegir_diari_PC('pc', lta);
  per cada (t ∈ lta ∧ t ∉ LC ∧ t ∉ LA) fer LA := LA ∪ t fper
  mentre LA ≠ ∅ fer
    llegir_diari_enrere('m', t, p, w);
    si t ∈ LA llavors
      escriure_bd(p, w);
      si ultima_annotacio(t) llavors LA := LA - t fsi
    fsi
  fmentre
faccio

```

Tinguem present que si l'aplicació de PC fos a transaccions acabades, en lloc d'accions acabades, l'acció fase\_desfer\_complementaria no seria necessària.

4) L'acció fase\_refer, després de situar-se en la posició de l'últim PC, llegeix endavant el diari i refà tots els canvis enregistrats en les anotacions de modificació corresponents a transaccions incloses en LC.

```

accio fase_refer()
  situar-se_al_darrer_PC(d);
  llegir_diari_endavant('m',t,p,v);
  mentre no_final_de_fitxer fer
    si t ∈ LC llavors escriure_bd(p,v) fsi;
    llegir_diari_endavant('m',t,p,v);
  fmentre
faccio

```

5) L'acció escriure\_bd s'utilitza en la resta d'accions del procediment reiniciar. S'encarrega d'emmagatzemar físicament una pàgina determinada a la BD.

```

accio escriure_bd(p: id_pagina, w: valor_pagina)
  si buscar(p) = 'no' llavors recollir(p) fsi;
  contingut(p) := w;
  canviat(p) := 'si';
  descarregar(p);
faccio

```

6) L'acció anotar\_PC només enregistra una anotació de PC al diari. La llista de transaccions actives és buida, atès que just després de reiniciar no hi ha cap transacció activa en el sistema.

```

accio anotar_PC()
  lta := ∅;
  escriure_diari('pc', lta);
faccio

```

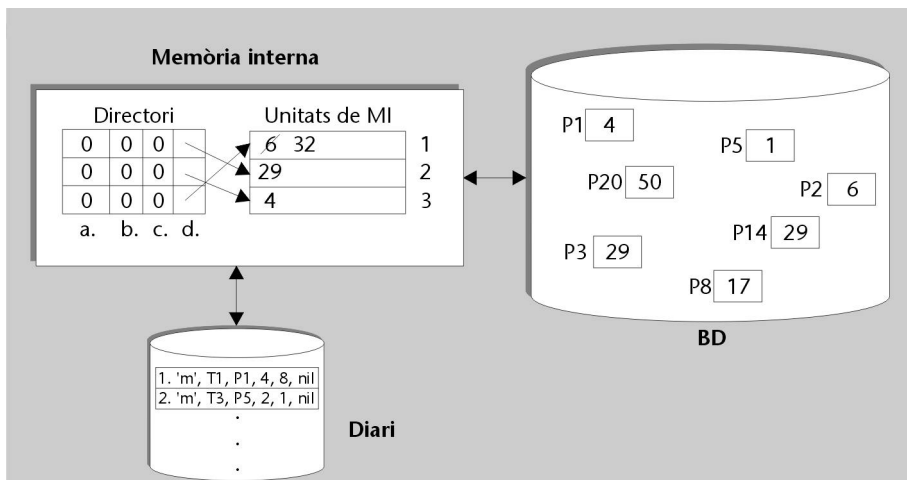
### Activitat

Imagineu que després de l'operació x de l'exemple proposat abans, s'ha produït una fallida de sistema. Simuleu l'execució de l'operació reiniciar, que hauria de fer el gestor de restauració, per a poder restablir el sistema i assegurar una BD consistent.

En les figures 7, 8 i 9 podeu observar el resultat després de l'execució de les accions fase\_desfer\_fins\_darrer\_PC, fase\_desfer\_complementaria i fase\_refer, respectivament.

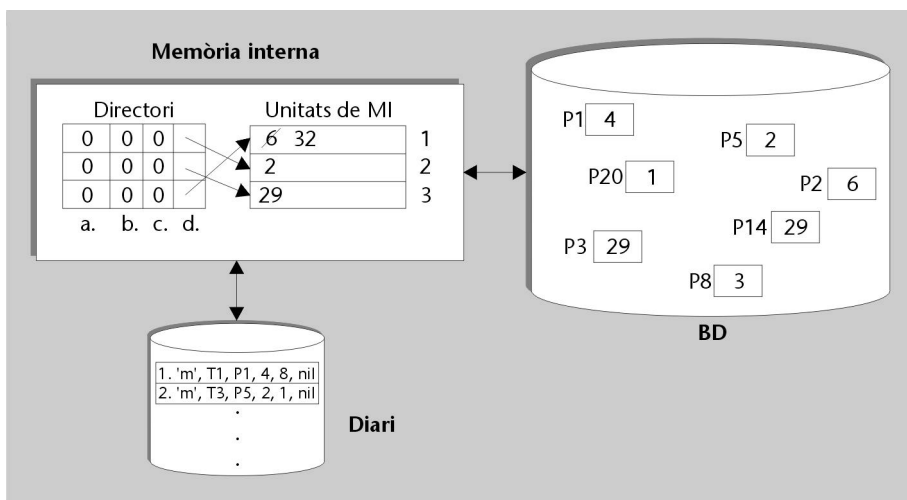
a) Execució de l'acció fase\_desfer\_fins\_darrer\_PC:

Figura 7



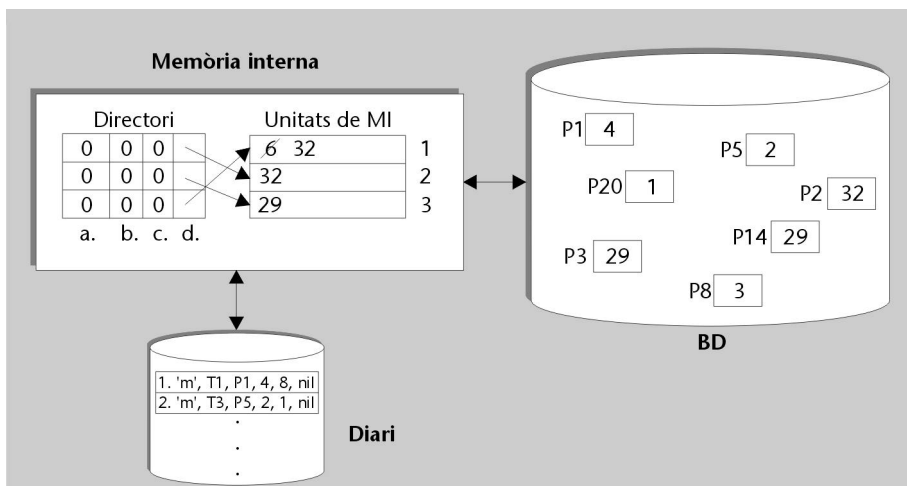
b) Execució de l'acció fase\_desfer\_complementaria:

Figura 8



c) Execució de l'acció fase\_refer:

Figura 9



### 3.3. Modalitat no prendre/no obligar

Els gestors de restauració que funcionen segons la modalitat no prendre/no obligar només han de preveure la necessitat de refer canvis de transaccions confirmades que s'hagin perdut. Gràcies al fet d'utilitzar la modalitat no prendre, en cap circumstància caldrà desfer cap modificació.

Els procediments de les operacions que ofereix el gestor de restauració al gestor de concurrència variaran respecte les que hem estudiat per a la modalitat prendre/no obligar en els aspectes següents:

1) Operació llegir: la part bàsica coincideix amb el mateix procediment que en la modalitat prendre/no obligar, però després de recollir la pàgina cal clavar-la (el bit de clavar/desclavar es posa a 1), atès que es treballa amb modalitat no prendre.

2) Operació escriure: en cas que la pàgina s'hagi de recollir, també caldrà clavar-la igual que en l'operació llegir. L'altra diferència és que en l'anotació al diari no farà falta enregistrar l'estat d'abans del canvi de la pàgina modificada, ja que en cap moment no serà necessari desfer-la.

3) Operació confirmar: tampoc no cal descarregar les unitats d'MI, però sí que cal desclavar-les (posar a 0 el bit de clavar/desclavar corresponent a totes les unitats d'MI assignades a la transacció que confirma). Igual que en la modalitat anterior, cal enregistrar l'anotació al diari.

4) Operació avortar: per la modalitat no prendre no s'ha pogut enregistrar físicament cap canvi; per tant, no cal desfer res. Com en l'operació confirmar, cal desclavar les unitats d'MI assignades a la transacció que ha avortat i fer l'anotació corresponent al diari.

5) Operació reiniciar: com ja hem vist, en aquesta modalitat només cal preveure la necessitat de refer, però no la de desfer. Així, doncs, de les cinc accions que hem après per a la modalitat prendre/no obligar, l'acció inicialitzar\_MI\_i\_llistes\_transaccions serà igual, de l'acció fase\_desfer\_fins\_darrer\_PC només es llegirà el diari enrere per a localitzar el darrer PC i per a identificar les transaccions que havien confirmat abans de la fallida. L'acció fase\_desfer\_complementaria és innecessària, i les altres dues accions seran idèntiques.

### 3.4. Modalitat prendre/obligar

Els gestors de restauració que funcionen segons la modalitat prendre/obligar només han de preveure la necessitat de desfer els canvis de transaccions que no han confirmat però que s'han pogut reflectir físicament. Gràcies al fet d'utilitzar la modalitat obligar, en cap circumstància farà falta refer cap modificació.

Els procediments de les operacions que ofereix el gestor de restauració al gestor de concurrència variaran respecte les que hem estudiat en els aspectes següents:

- 1) Operació llegir: coincideix amb el mateix procediment que en la modalitat prendre/no obligar.
- 2) Operació escriure: coincideix amb el mateix procediment que en la modalitat prendre/no obligar, però en l'anotació al diari no caldrà enregistrar l'estat de després del canvi de la pàgina modificada, ja que en cap moment serà necessari refer-la.
- 3) Operació confirmar: cal descarregar les unitats d'MI assignades a la transacció que confirma i després enregistrar l'anotació corresponent al diari. Per la modalitat prendre, és possible que el nombre d'unitats d'MI assignades a la transacció que s'hagin de descarregar sigui petit.
- 4) Operació avortar: coincideix amb el mateix procediment que en la modalitat prendre/no obligar.
- 5) Operació reiniciar: com ja hem vist a l'inici del subapartat, en aquesta modalitat només cal preveure la necessitat de desfer, però no de refer. Així, doncs, de les cinc accions que hem après per la modalitat prendre/no obligar, tot serà idèntic excepte l'acció fase\_refer, que no és necessària.

### 3.5. Modalitat no prendre/obligar

Els gestors de restauració que funcionen segons la modalitat no prendre/obligar, en cap circumstància cal que desfacin ni refacin cap modificació.

Els procediments de les operacions variaran, respecte les que ja hem estudiat, en els aspectes següents:

- 1) Operació llegir: coincideix amb el mateix procediment que en la modalitat no prendre/no obligar; per tant, després de recollir la pàgina, caldrà clavar-la (es posarà a 1 el bit de clavar/desclavar).
- 2) Operació escriure: en cas que faci falta recollir la pàgina també caldrà clavar-la igual que en l'operació llegir. En l'anotació al diari no caldrà enregistrar l'estat d'abans del canvi de la pàgina modificada, ja que en cap moment no serà necessari desfer-la; ni l'estat de després del canvi, atès que en cap moment no serà necessari refer-la.
- 3) Operació confirmar: cal descarregar les unitats d'MI assignades a la transacció que confirma i després enregistrar l'anotació corresponent al diari, igual

Enregistrament de l'estat de després del canvi

Tot i que per la restauració la informació de l'estat de després del canvi és innecessària, s'acostuma a enregistrar igualment atesa la seva importància per la reconstrucció.

Enregistrament de l'estat de després del canvi

Novament, tot i que la informació de l'estat de després del canvi és innecessària per a la restauració, també s'acostuma a enregistrar, atesa la importància que té per a la reconstrucció.

que en la modalitat prendre/obligar. Hem de tenir present que per a descarregar les unitats d'MI, primer cal desclavar-les.

Atès que s'ha de garantir que la descàrrega sigui atòmica, cal aplicar la tècnica d'ombreig; per tant, aquesta modalitat de gestor és inviable si la descàrrega de les unitats d'MI es fa segons la tècnica d'actualitzar al lloc.

Vegeu la restauració amb la tècnica d'ombreig en el subapartat 3.6 d'aquest mòdul.

4) Operació avortar: coincideix amb el procediment avortar de la modalitat no prendre/no obligar.

5) Operació reiniciar: només cal inicialitzar l'MI. Les fases desfer i refer són innecessàries.

### 3.6. Restauració aprofitant l'estratègia d'ombreig

Els mecanismes de restauració pel procediment reiniciar que hem estudiat en els subapartats anteriors utilitzaven el diari d'operacions, atès que és indispensable en sistemes en què l'estratègia de descàrrega per part del gestor d'MI és actualitzar al lloc.

Hem vist que l'altra estratègia de descàrrega de pàgines que pot aplicar el gestor d'MI és l'estratègia d'ombreig, i també hem apuntat que aquesta estratègia facilita les tasques del gestor de recuperació.

Recordeu el funcionament de l'estratègia d'ombreig vist en el subapartat 2.1.2 d'aquest mòdul.

En l'estratègia d'ombreig sempre s'utilitza un espai físic diferent per a emmagatzemar la pàgina després d'haver-la modificat (no se sobrescriu), i el valor que la pàgina tenia abans de modificar-la es manté. Aleshores, la utilització de diversos directoris permet de determinar els diferents estats de la BD.

#### Utilitat dels directoris

Gràcies a un directori podem saber quin era l'estat de la BD abans que s'iniciés l'execució d'una transacció, i mitjançant un altre directori es pot reflectir l'estat nou de la BD després que la transacció faci les modificacions.

Si tenim en compte que els canvis sempre s'emmagatzemen físicament, no cal preveure la necessitat d'haver de refer cap modificació perduda com a conseqüència d'una fallida. D'altra banda, desestimar els canvis produïts per una transacció que no ha pogut confirmar resulta tan senzill com desestimar el directori nou i les pàgines noves generats com a conseqüència de l'execució de la transacció i tornar a considerar com a vàlids el directori i les pàgines ombra.

Així, doncs, en aquest cas podem garantir que el mecanisme de restauració no ha de desfer ni refer. L'única complicació de fer servir aquesta tècnica està relacionada amb la localització d'espai lliure per a poder emmagatzemar les pàgines noves i el directori nou sempre que calgui, i també en l'alliberació de l'espai que aquests ocupen en cas que es desestimïn. Sempre que una transacció confirmi, les pàgines velles adreçades pel directori ombra es poden desestimar.

## 4. Reconstrucció

El gestor de reconstrucció, inclòs en el gestor de recuperació, és l'encarregat d'aplicar les tècniques de reconstrucció sempre que es produeixi una pèrdua parcial o total de la BD per errors en algun dels components del maquinari.

Recordeu el concepte de reconstrucció introduït en el subapartat 3.2 del mòdul "Transaccions en les bases de dades" de l'assignatura Bases de dades II.

En altres assignatures s'ha après que tant el diari d'operacions com les còpies de seguretat són essencials per a poder reconstruir la BD.

En el moment en què es detecta una pèrdua total o parcial per error en algun component del maquinari cal seguir els passos següents per a reconstruir la pèrdua:

- 1) Substituir o reparar el component del maquinari que té problemes.
- 2) Localitzar la còpia de seguretat just anterior a la data en què s'ha produït l'error, atès que és la còpia que conté més quantitat d'informació de l'estat de la BD en el moment de la pèrdua.
- 3) Copiar el contingut de la còpia de seguretat per a reconstruir la BD.
- 4) Per a finalitzar la reconstrucció, refer des del diari tots els canvis que s'han produït des de la data de la pèrdua fins a l'actualitat.

### Tipus de còpies de seguretat

Ja sabeu que existeixen diferents tipus de còpies de seguretat:

- estàtiques
- dinàmiques
- completes o totals
- incrementals

Aquest mateix procediment es pot fer servir per a corregir canvis que s'havien produït a la BD i que posteriorment s'ha vist que eren incorrectes.

D'altra banda, les anotacions prèvies a un PC que es poden desestimar no es perden, atès que es guarden sempre en còpies de seguretat del diari per a assegurar la disponibilitat de totes les anotacions del diari necessàries per a fer la reconstrucció.

Igualment, per a poder garantir la disponibilitat de les dades necessàries per a la reconstrucció si l'SGBD no disposa de diari perquè utilitza la tècnica d'ombra en fer la descàrrega de les unitats d'MI, es fan còpies de seguretat de les pàgines i directoris ombra.

Tot i així, hi ha una altra possibilitat a l'hora de superar un problema de pèrdua total o parcial de la BD, que consisteix a disposar d'un duplicat de la BD.

Quan un sistema de BD disposa de duplicat o còpia secundària, a part de la còpia primària de la BD, en cas que es produeixi algun error en el dispositiu

d'emmagatzematge de la còpia primària, es pot passar a fer servir el duplicat immediatament sense que els usuaris es vegin afectats pel canvi. Podem reconstruir la part perduda de la còpia primària a partir de la còpia secundària.

Tot i que si es disposa de duplicat, les còpies de seguretat semblen innecessàries, és habitual fer-ne per si la còpia secundària també tingués problemes.

Per a assegurar que el duplicat no té el mateix problema que la còpia primària, cal que estiguin emmagatzemats en dispositius diferents. Per cada còpia de la BD també s'emmagatzema un diari en dispositius diferents.

Durant el funcionament habitual de la BD, totes les lectures s'adrecen a la còpia primària. Quan arriba una escriptura, cal actualitzar les dues còpies perquè siguin idèntiques. Per a no disminuir el temps de resposta de la BD, en lloc d'actualitzar les dues còpies a la vegada, primer s'actualitza la còpia primària i posteriorment, el duplicat. Així, doncs, el fet de mantenir un duplicat incrementa la capacitat per a les lectures però no per les escriptures.



## Resum

En aquest mòdul hem après com la propietat d'atomicitat i la propietat de definitivitat de les transaccions posen de manifest la necessitat de les tècniques de restauració per a garantir la consistència de la BD. També hem vist com altres problemes, com els errors en components del maquinari, poden provocar pèrdua de dades, fet pel qual també es fan indispensables les tècniques de reconstrucció.

Tant de les tècniques de restauració com de les tècniques de reconstrucció se n'ocupa el gestor de recuperació, que és el subcomponent més important del component de gestió de les dades de l'SGBD. L'objectiu del component de gestió de les dades és atendre les peticions d'accés a la BD física alhora que es preserva la consistència de les dades. Per a fer-ho, utilitza l'MI mitjançant el gestor d'MI.

També hem ampliat els coneixements respecte al diari d'operacions. El diari és un component que, tot i que no està integrat en el component de gestió de les dades, és imprescindible tant per a la restauració com per a la reconstrucció.

Pel que fa a la restauració, hem vist els diferents mecanismes tenint en compte la modalitat del gestor de restauració (prendre o no prendre, i obligar o no obligar), la més habitual de les quals és la modalitat prendre/no obligar.

Finalment, després de recordar la possibilitat de reconstruir la BD a partir de còpies de seguretat i del diari, també hem après la possibilitat de fer-ho a partir d'un duplicat.



## Activitats

1. Atès que esteu acostumats a treballar amb Informix, intenteu esbrinar si aquest sistema de gestió de bases de dades disposa de diari o no. En cas afirmatiu, expliqueu de quin tipus és, i en cas negatiu, descriuiu quines eines o mecanismes utilitza en lloc del diari per a ajudar en les tècniques de recuperació.
2. Esclariu en quina modalitat de gestor de restauració treballa Informix.

## Exercicis d'autoavaluació

1. En quines situacions es pot produir pèrdua total o parcial de les dades emmagatzemades en una BD?
2. Descriuiu les operacions que ofereix el gestor d'MI.
3. Per a què s'utilitza el diari d'operacions?
4. Quina modalitat de gestor de restauració minimitza el nombre d'entrades/sortides?
5. Escriviu un algorisme per a l'operació confirmar per a un gestor de restauració que treballa segons la modalitat no prendre/no obligar.
6. En què es diferencia el procediment reiniciar si els punts de control al diari s'apliquen a transaccions acabades en lloc d'aplicar-se a accions acabades?
7. Les còpies de seguretat són útils només per a recuperar dades que s'han perdut?

## Solucionari

1. La pèrdua de dades emmagatzemades en una BD es pot produir bàsicament com a conseqüència d'una fallida de sistema, atès que impedeix una correcta execució de les transaccions, i també per errors en els dispositius d'emmagatzematge.
2. El gestor d'MI ofereix dues operacions. L'operació recollir serveix per a portar una pàgina física de la BD cap a una unitat de l'MI. Amb l'operació descarregar es treu una pàgina d'una unitat de l'MI a la vegada que s'emmagatzema físicament en el cas que s'hi hagi fet algun canvi.
3. El diari d'operacions s'utilitza per a emmagatzemar la informació dels canvis produïts durant l'execució de les transaccions i així s'aconsegueix disposar de la informació necessària per a recuperar un estat consistent de la BD.
4. Qualsevol gestor basat en la modalitat no obligar aconseguirà de minimitzar el nombre d'entrades/sortides, atès que només descarrega les unitats d'MI quan ho necessita.
5. Recordeu que les unitats d'MI assignades a una transacció es poden conèixer si es recorre enrere el diari. Aleshores, l'algorisme per a l'operació confirmar es podria escriure d'aquesta manera:

```
procediment confirmar(t: id_transaccio)
var p: id_pagina;
    lu: llista_d'unitats_MI;

    lu := unitats_MI_assignades(t);
    per cada p ∈ lu fer
        clavar(p) := 'no';
    fper
    escriure_diari('c', t);
fprocediment
```

Una altra opció consisteix a fer que el gestor de restauració mantingui una llista d'unitats assignades per cada transacció.

6. El fet d'aplicar els punts de control a transaccions acabades implica que no hi havia cap transacció activa en el moment de l'aplicació del PC; per tant, podem estar segurs que en reiniciar el sistema, en cap moment no caldrà desfer canvis anotats abans d'aplicar el darrer PC (l'acció fase\_desfer\_complementaria no cal).
7. No. Les còpies de seguretat són útils per a recuperar un estat de la BD determinat, ja sigui per pèrdua o bé perquè a partir d'un instant de temps concret, els canvis han generat dades incorrectes. Gràcies a les còpies de seguretat podem recuperar un estat correcte anterior a l'instant en què s'ha produït l'error, i així no cal corregir-lo.

## Glossari

actualitzar al lloc

Tècnica aplicada en la descàrrega d'unitats de memòria intermèdia cap a la base de dades física que sobrescriu la pàgina, atès que sempre es guarda en el mateix lloc.

component de gestió de les dades

Component del sistema de gestió de bases de dades que serveix les peticions d'accés a la base de dades de manera convenient per a assegurar-ne la correctesa de les dades.

descarregar

Operació del gestor de la memòria intermèdia que treu una pàgina de la memòria intermèdia i la guarda a la base de dades física en cas que s'hagi modificat.

diari d'operacions

Fitxer seqüencial que enregistra la història de les transaccions amb els canvis que provoquen.

duplicat

Còpia secundària d'una base de dades utilitzada per a afavorir les lectures i per a reconstruir la base de dades en cas de pèrdua total o parcial de les dades.

#### gestor de recuperació

Subcomponent del component de gestió de les dades encarregat dels mecanismes de recuperació de la base de dades.

#### memòria intermèdia

Part de la memòria interna dedicada a l'ús del sistema de gestió de bases de dades per a aconseguir eficiència en servir els accessos a la base de dades física.

sigla: MI

#### MI

Vegeu memòria intermèdia

#### modalitat no obligar

Modalitat de funcionament del gestor de restauració en què, en confirmar, no es força la descàrrega de les unitats de memòria intermèdia que estaven assignades a la transacció.

#### modalitat obligar

Modalitat de funcionament del gestor de restauració en què, en confirmar, el gestor de restauració força la descàrrega de les unitats de memòria intermèdia que estaven assignades a la transacció.

#### modalitat no prendre

Modalitat de funcionament del gestor de restauració en què el gestor de restauració no permet que es descarregui una unitat de memòria intermèdia mentre la transacció que l'utilitza romangui activa.

#### modalitat prendre

Modalitat de funcionament del gestor de restauració en què el gestor de restauració permet que es descarregui una unitat de memòria intermèdia tot i que la transacció que l'utilitza romangui activa.

#### ombreig

Tècnica aplicada en la descàrrega d'unitats de memòria intermèdia cap a la base de dades física que cerca un espai nou cada cop que s'ha de guardar una pàgina, i així manté la versió antiga.

#### portar

Sin: recollir

#### protocol d'escriptura avançada al diari

Protocol que estableix com s'han de fer les anotacions al diari.

#### punt de control del diari

Mètode que força l'emmagatzematge d'un estat consistent de la base de dades anotant-ho al diari.

#### recollir

Operació del gestor de memòria intermèdia que llegeix una pàgina i la guarda en una unitat de la memòria intermèdia.

sin: portar

#### reconstrucció

Terme que inclou tots els mecanismes o tècniques per a reconstruir una base de dades en cas de pèrdua total o parcial de les dades per errors en components del maquinari.

#### restauració

Terme que inclou tots els mecanismes o tècniques per a restaurar una base de dades, i així assegurar el compliment de la propietat d'atomicitat i la propietat de definitivitat de les transaccions.

#### SGBD

Sigla de sistema de gestió de bases de dades

## Bibliografia

Bernstein, P.A.; HadziIacos, V.; Goodman, N. (1987). *Concurrency Control and Recovery in Database Systems*. Reading (Massachusetts): Addison-Wesley.

Connolly, T.; Begg, C.; Strachan, A. (1999). *Database Systems. A Practical Approach to Design, Implementation, and Management*. Harlow (England): Addison-Wesley.

Elmasri, R.; Navathe, S.B. (2000). *Fundamentals of Database Systems*. Reading (Massachusetts): Addison-Wesley.

McFadden, F.R; Hoffer, J.A.; Prescott, M.B. (1999). *Modern Database Management*. Reading (Massachusetts): Addison-Wesley.

Ramakrishnan, R. (1998). *Database Management Systems*. Boston (Massachusetts): McGraw-Hill.