TEMA III

Sistemes de Gestió de Bases de Dades

Sistemes de gestió de bases de dades (SGBD)

- **Objectius:**
  - conéixer l'arquitectura ANSI/SPARC per a sistemes de gestió de bases de dades (SGBD).
  - aprendre el concepte d'independència de dades.
  - aprendre mecanismes i estratègies per al control de la integritat (qualitat) i seguretat (privacitat) en bases de dades (BD).
  - conéixer la implementació d'una base de dades relacional

Sistemes de gestió de bases de dades (SGBD)

Temari

3.1. Sistema de gestió de bases de dades: components i funcions

3.2. Independència de dades

3.3. Integritat

3.3.1. Concepte de transacció. Processament per transaccions

3.3.2. Integritat semàntica

3.3.3. Accessos concurrents

3.3.4. Reconstrucció de la base de dades

3.4. Seguretat

3.4.1. Control d'usuaris

3.4.2. Control d'accessos permesos

3.5 Implementació d'una Base de Dades Relacional.

3.1. Sistema de gestió de bases de dades

SGBD: programari que permet la creació i manipulació de bases de dades.

SGBD

Es basa

model de dades (model relacional)

Es compon

estructures de dades i operadors associats

3.1.1. Components i funcions del SGBD

Els SGBD permeten:

- descripció unificada de les dades i independent de les aplicacions
- independència de les aplicacions respecte de la representació física de les dades
- definició de vistes parcials de les dades per a distints usuaris
- gestió de la informació
- integritat i seguretat de les dades

3.1.1. Components i funcions del SGBD

Objectius de tècniques BD

- descripció unificada i independent de les aplicacions
- independència de les aplicacions
- definició de vistes parcials

Funcions SGBD

Definició de dades a diferents nivells:

- esquema lògic
- esquema intern
- esquemes externs

Components SGBD

Llenguatges de definició d'esquemes i traductors associats

3.1.1. Components i funcions del SGBD

Objectius de tècniques BD

Gestió de la informació

Funcions SGBD

Manipulació de les dades:

- consulta
- actualització

Gestió i administració de la base de dades

Components SGBD

Llenguatges de manipulació i traductors associats

Eines per a:

- reestructuració
- simulació
- estadístiques
- impressió

3.1.1. Components i funcions del SGBD

Objectius de tècniques BD

Integritat i seguretat de les dades

Funcions SGBD

Control de:

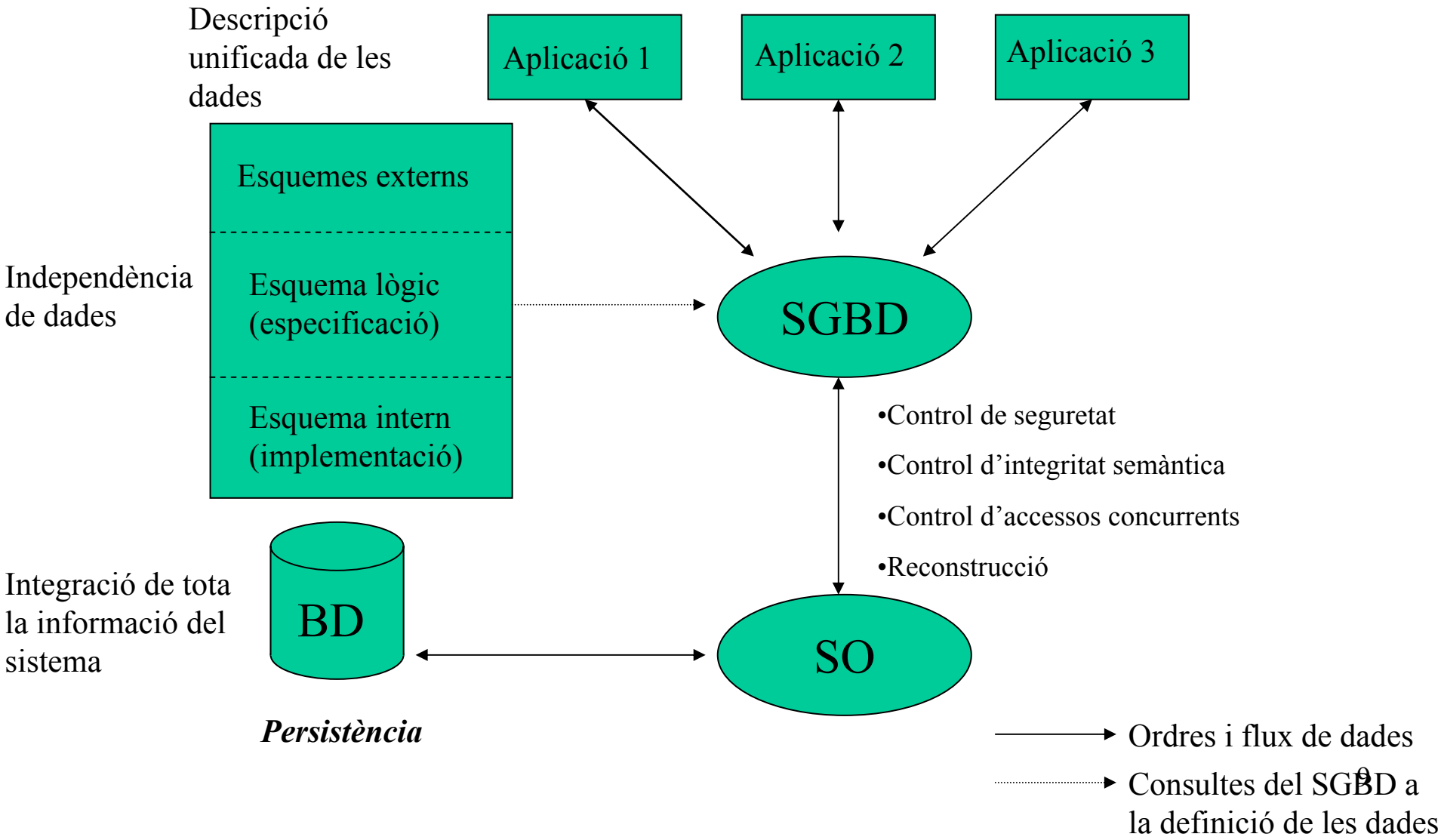
- integritat semàntica
- accessos concurrents
- reconstrucció en cas d'error
- seguretat (privacitat)

Components SGBD

Eines per a:

- control integritat
- reconstrucció
- control seguretat

3.1.2. Esquema d'accés del SGBD a les dades



3.1.2. Esquema d'accés del SGBD a les dades

Esquema extern aplicació 1:

```
CREATE VIEW Administratiu (dni, nom, salari_men)
    AS SELECT dni, nom, salari/14
        FROM Empleat
        WHERE tipus='AD'
```

Esquema lògic:

Empleat(dni, nom, adreça, salari, tipus)
CP: {dni}

Esquema intern:

Fitxer ordenat *Empleat* amb índex primari sobre el camp *dni* en el camí *h:/disc1/gerència*

3.1.2. Esquema d'accés del SGBD a les dades

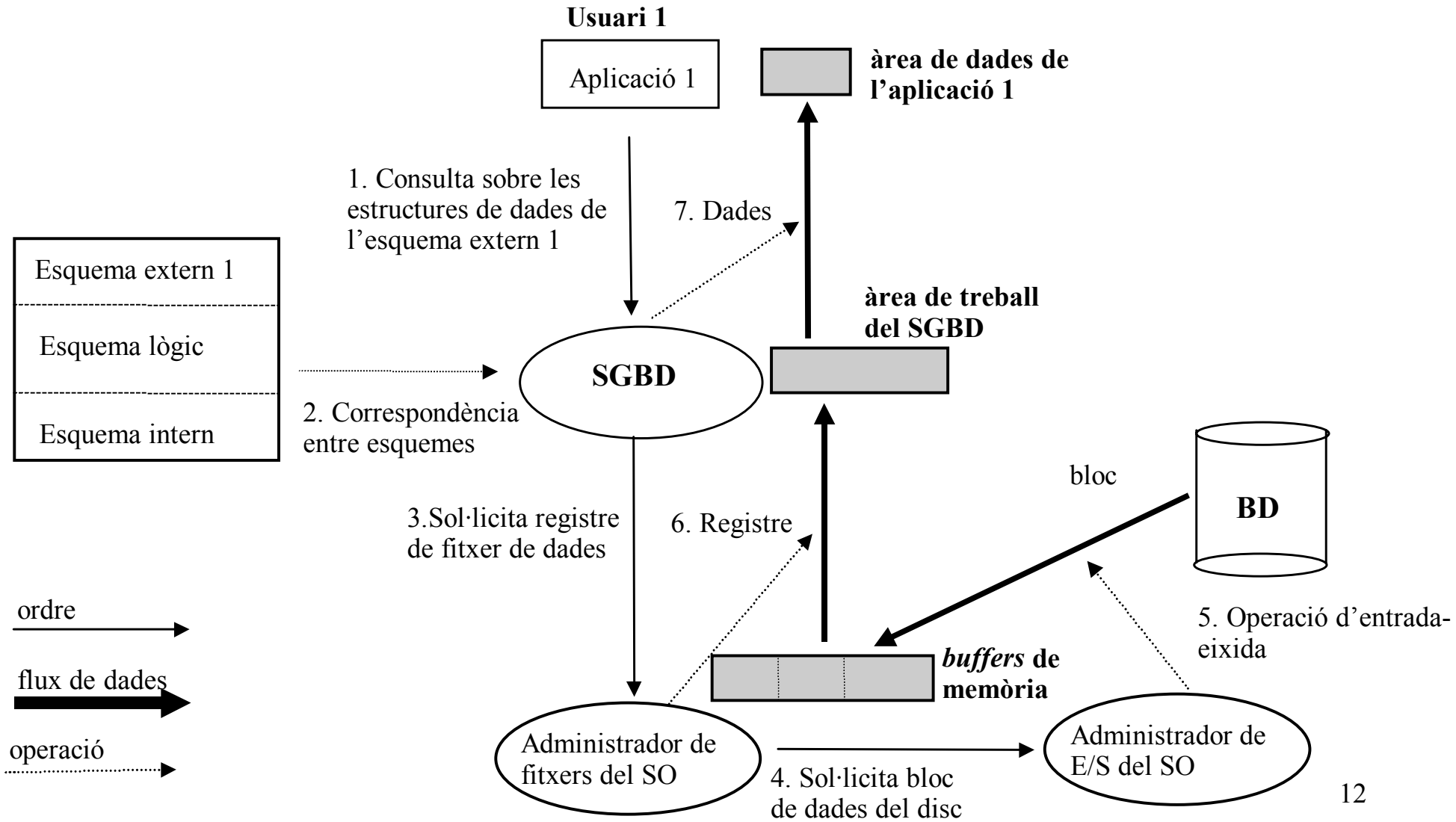
Aplicació 1: accedeix a la informació per l'esquema extern 1

```
SELECT nom, salari_men
    FROM Administratiu
    WHERE dni = paràmetre
```

SGBD: control de l'accés i resolució de l'operació demanada

SO: manipulació dels controladors dels dispositius de memòria secundària

3.1.2. Esquema d'accés del SGBD a les dades



3.1.2. EXEMPLE. Especificació

Una xicoteta immobiliària desitja mantenir informació sobre la venda dels edificis que gestiona. Es vol saber:

- De cada edifici, el codi, la ubicació, el districte, el propietari, el preu sol·licitat per aquest i l'agent encarregat de la venda si ja està assignat.
- De cada propietari, el codi, el nom i el telèfon.
- De cada agent, el DNI, el nom, la comissió per cada venda, els anys d'antiguitat i el telèfon.

Les restriccions que s'han de complir són les següents:

- La comissió d'un agent no pot excedir el 3% si la seua antiguitat és menor de 3 anys.
- No es vol tenir informació de propietaris si no es té almenys un edifici per a la venda.

Grups de treball:

- El personal d'administració té accés a tota la informació comentada.
- El cap de la immobiliària només desitja tenir informació referent als edificis amb preu sol·licitat superior a 5 milions. De cadascun desitja el codi, la ubicació i el districte.
- El cap és l'únic que pot modificar la informació dels agents.

3.1.2. EXEMPLE. Esquema lògic (SQL)

```
CREATE SCHEMA Immobiliària
CREATE TABLE Edifici
    (Codi d_cod PRIMARY KEY,            Ubicació d_ubi NOT NULL,
     Districte d_dis NOT NULL,          Preu d_pre NOT NULL,
     Dni_age d_dni REFERENCES Agent
          ON UPDATE CASCADE, ON DELETE NO ACTION
     Amo d_cod NOT NULL,  FOREIGN KEY(Amo) REFERENCES Propietari (cod)
          ON UPDATE CASCADE ON DELETE CASCADE)
CREATE TABLE Propietari
    (Cod d_cod PRIMARY KEY, Nom d_nom NOT NULL, Telèfon d_tel NOT NULL)
CREATE TABLE Agent
    (Dni_age d_dni PRIMARY KEY,  Comissió d_com, Anys d_anys NOT NULL,
     Tel d_tel NOT NULL, CHECK (NOT (anys < 3 AND comissió > 3)))
CREATE ASSERTION no_propet_sense_edificis CHECK NOT EXISTS
    (SELECT * FROM Propietari WHERE cod NOT IN (SELECT Amo FROM Edifici))
```

3.1.2. EXEMPLE. Esquemes externs (SQL)

GRANT ALL ON Edifici TO PUBLIC;

GRANT ALL ON Propietari TO PUBLIC;

GRANT SELECT ON Agent TO PUBLIC;

ESQUEMA EXTERN DEL CAP:

```
CREATE VIEW més_de_5 AS
        SELECT codi, ubicació, districte
        FROM Edifici
        WHERE E.preu >= 5000000;
GRANT ALL ON més_de_5 TO Cap;
GRANT ALL ON Agent TO Cap;
```

+ La resta de taules de l'esquema lògic (excepte edificis)

ESQUEMA EXTERN DEL PERSONAL ADMINISTRACIÓ:

Totes les taules de l'esquema lògic

3.1.2. EXEMPLE. Esquema físic

Edificis

- Fitxer dispers per dni_age
- Índex B+ sobre (districte + preu)

Propietaris

- Fitxer dispers per cod
- Índex B+ sobre nom

Agents

- Fitxer desordenat (se suposen pocs agents)

3.1.2. EXEMPLE. Procés d'accés

El cap es pregunta:

codi i ubicació dels edificis del districte 05?

1. L'aplicació interpreta la selecció del cap com a:

SELECT codi, ubicació

FROM més_de_5 WHERE districte = '05';

2. El SGBD converteix la consulta de l'esquema extern a l'esquema lògic:

SELECT codi, ubicació

FROM Edificis E

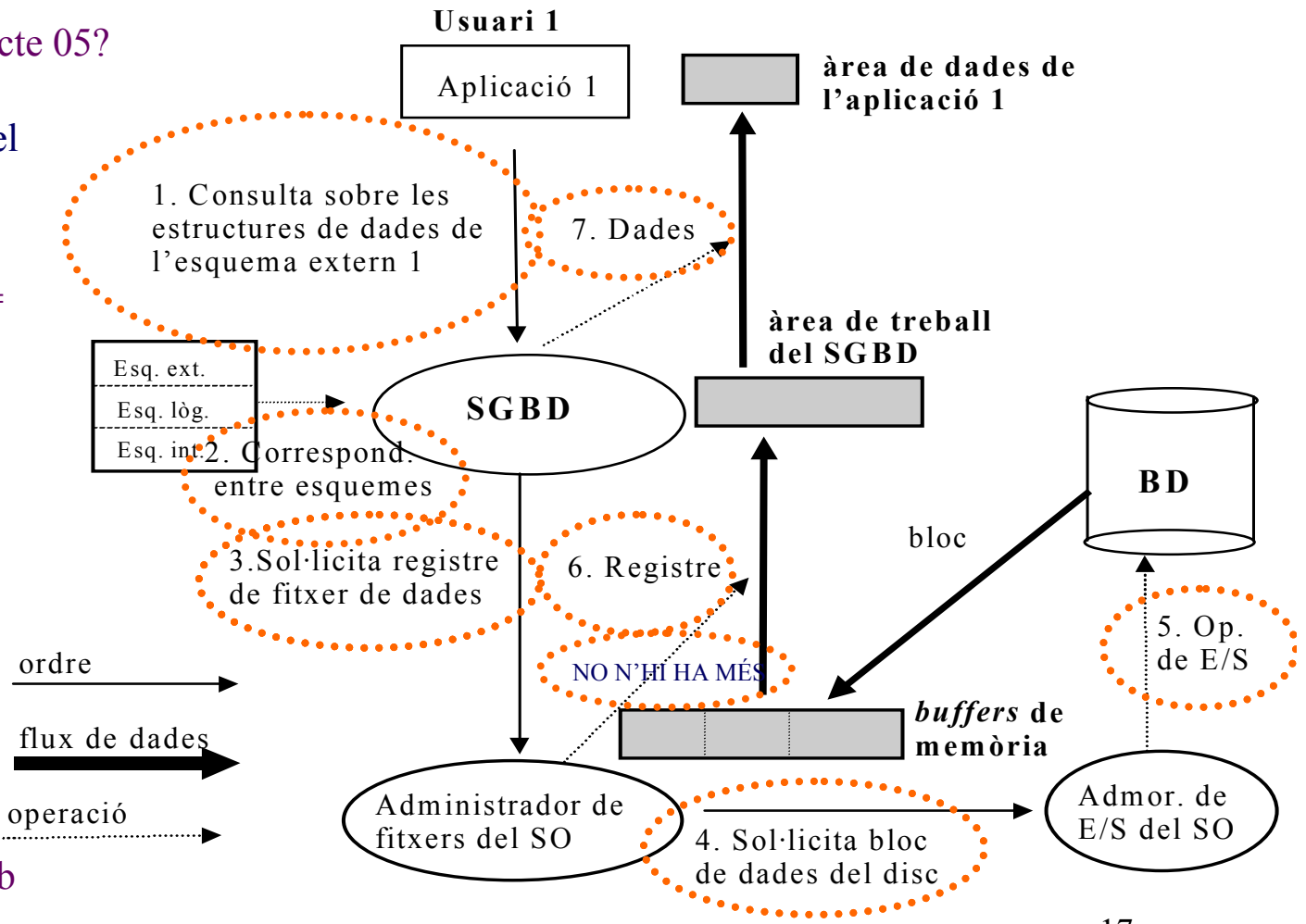
WHERE E.preu >= 5000000

AND districte = '05';

3,4,5,6. ES REPETEIX:

"Es llig usant l'índex B+ sobre (districte + preu) el primer registre amb districte = '05' i preu >= 5000000:

FINS QUE NO HI HAJA MÉS REGISTRES



7. S'eliminen els atributs que no s'han sol·licitat

3.2. Independència de dades

Propietat que assegura que els programes d'aplicació siguen independents dels canvis realitzats en **dades que no usen** o en **detalls de representació física de les dades a les quals accedeixen.**

3.2. Independència de dades

Proposta d'arquitectura del grup d'estudi ANSI/SPARC (1977) per als SGBD: planteja la definició de la base de dades a tres nivells d'abstracció:

- **Nivell conceptual** $\Rightarrow$ Esquema conceptual
  - descripció de la BD amb independència del SGBD
- **Nivell intern** $\Rightarrow$ Esquema intern
  - descripció de la BD en termes de la seua representació física
- **Nivell extern** $\Rightarrow$ Esquema extern
  - descripció de les vistes parcials de la BD que posseïsquen els distints usuaris

3.2. Independència de dades

Ja que no hi ha un model conceptual generalitzat i accessible als distints tipus de SGBD, es prefereix distingir quatre nivells:

- **Nivell conceptual** $\Rightarrow$ Esquema conceptual
  descripció organitzativa de la BD
- **Nivell lògic** $\Rightarrow$ Esquema lògic
  descripció de la BD en termes del model de dades del SGBD
- **Nivell intern** $\Rightarrow$ Esquema intern
  descripció de la BD en termes de la seua representació física
- **Nivell extern** $\Rightarrow$ Esquema extern
  descripció de les vistes parcials de la BD que posseeixen els distints usuaris

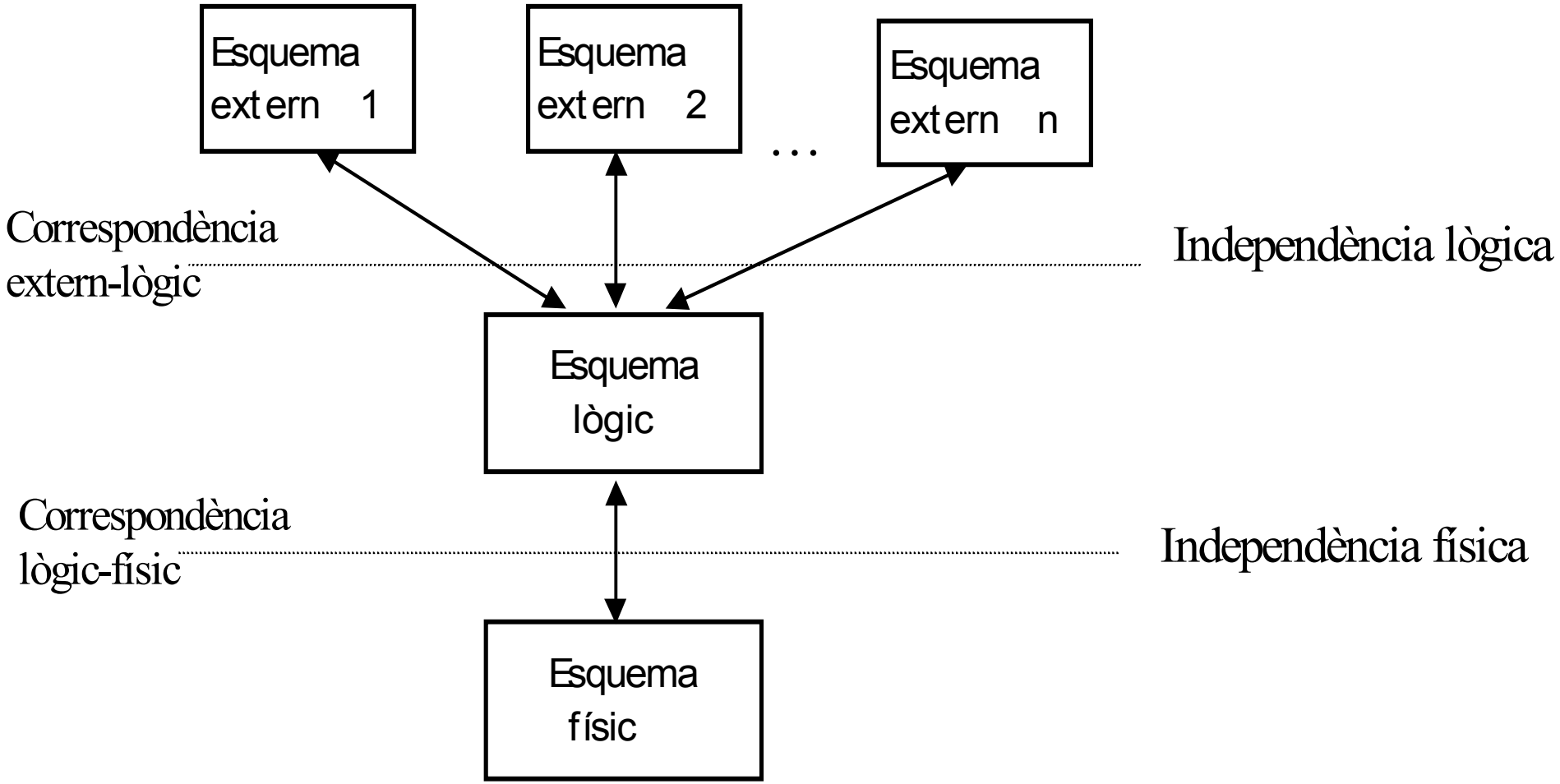
3.2. Independència de dades

Un SGBD que suporte l'arquitectura de nivells ha de:

- permetre definir els distints esquemes de la base de dades (a excepció de l'esquema conceptual)
- establir les correspondències entre els esquemes
- aïllar els esquemes: els canvis en un esquema no han d'afectar els esquemes de nivell superior i, per tant, tampoc els programes d'aplicació

INDEPENDÈNCIA DE DADES

3.2. Independència de dades



3.2. Independència de dades

- *Independència lògica* entre l'esquema lògic i els externs:
  - Els esquemes externs i els programes d'aplicació no s'han de veure afectats per modificacions de l'esquema lògic sobre dades que no usen.

- *Independència física* entre l'esquema intern i el lògic:
  - l'esquema lògic no s'ha de veure afectat per canvis en l'esquema intern referents a la implementació de les estructures de dades, maneres d'accés, grandàries de pàgines, camins d'accés, etc.

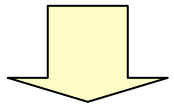
3.2. Independència de dades

ENLLAÇ:

– Transformació de l'esquema extern en l'esquema intern.

Tipus
- – Enllaç lògic (passos 2 i 7).
- – Enllaç físic (passos 3 i 6).

– Quan es produeix l'enllaç desapareix la independència.



És important determinar aquest moment

3.2. Independència de dades

Programa d'aplicació:

- Enllaç en temps de compilació:
  - ◊ Transformació de l'esquema extern que usa el programa en termes de l'esquema intern.
  - ◊ Qualsevol canvi de l'esquema lògic i/o intern requereix una recompilació.
- Enllaç en executar el programa:
  - ◊ No requereix cap acció sobre el programa.

3.2. Independència de dades

Moment de l'enllaç:

- en compilació o en la precompilació
- en el muntatge
- en iniciar-se l'execució o en el moment de connectar-se
- en cada accés a la base de dades

Major independència com més tardà siga l'enllaç

Menor cost com més primerenc siga l'enllaç

3.3. Integritat

- Objectiu de la tecnologia de bases de dades
- Qualitat de la informació:

  "*les dades han d'estar estructurades perquè reflectisquen adequadament els objectes, les relacions i les restriccions existents en la parcel·la del món real que modela la base de dades*"

- Representació dels objectes, de les relacions i de les restriccions en l'esquema de la base de dades.
- Canvis en la realitat → Actualitzacions dels usuaris.
- La informació continguda en la base de dades ha de preservar la definició de l'esquema.

3.3. Integritat

- Qualitat de la informació (perspectiva de la integritat):
  - SGBD ha d'assegurar que les dades s'emmagatzemen correctament
  - SGBD ha d'assegurar que les actualitzacions dels usuaris sobre la base de dades s'executen correctament i que es fan permanents

3.3. Integritat

Eines del SGBD orientades a la integritat:

- Comprovar (enfront d'actualitzacions) les restriccions d'integritat de l'esquema
- Controlar l'execució correcta de les actualitzacions (entorn concurrent)
- Recuperar (reconstruir) la base de dades en cas de pèrdues o accidents

3.3. Integritat: accessos concurrents

Comptes

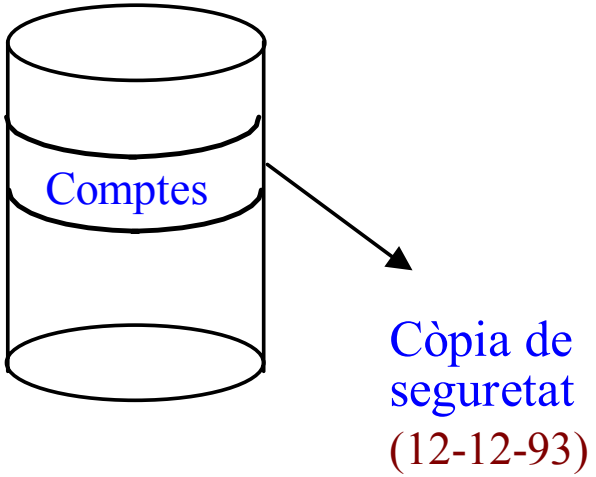
Núm.	Saldo
123	1000
555	2000

Comptes

Núm.	Saldo
123	800
555	2000

Temps	P1	P2
t1	llegir(123, saldo)	
t2		llegir(123, saldo)
t3	saldo←saldo-100	
t4		saldo←saldo-200
t5	escriure(123, saldo)	
t6		escriure(123, saldo)

3.3. Integritat: recuperació



Procediment de recuperació:

- substituïu el fitxer de *Comptes* per la seua còpia de seguretat

Efecte negatiu:

- s'han perdut les actualitzacions de 50 transaccions

3.3. Integritat: transaccions

- La integritat de la base de dades es veu en perill generalment per les operacions d'accés de les aplicacions.
- Les operacions d'accés a una base de dades s'organitzen en transaccions.

TRANSACCIÓ → Seqüència d'operacions d'accés a la base de dades que constitueixen una unitat lògica d'execució

3.3. Integritat: transaccions

Emp(dni, nom, dir, dept)
CP: {dni}
CAl: {dept} $\rightarrow$ Dep

Dep(cod, nom, ubicació)
CP: {cod}

R1: $\forall$Dx (Dep(Dx) $\rightarrow$ $\exists$Ex (Emp(Ex) $\wedge$ Dx.cod = Ex.dept))

Inserció d'un nou departament:

<d2, "Personal", "Planta 3a">

amb el primer empleat que és el de *dni* 20

3.3. Integritat: transaccions

1a idea

1) Inserció en *Dep*: <d2, “Personal”, “Planta 3a>

ERROR: la restricció *R1* no es compleix

2) Modificació de *Emp* en la tupla amb *dni* 20

2a idea

1) Modificació de *Emp* en la tupla amb *dni* 20

ERROR: la clau aliena sobre *dept* en *Emp* no es compleix

2) Inserció en *Dep*: <d2, “Personal”, “Planta 3a>

3.3. Integritat: transaccions

Operacions de les transaccions rellevants per al SGBD:

- **llegir(X)**: lectura o consulta de la dada X de la base de dades sobre la variable del programa del mateix nom
- **escriure(X)**: actualització (inserció, esborrament o modificació) de la dada X de la base de dades usant la variable del mateix nom del programa

3.3. Integritat: transaccions

Operacions de les transaccions rellevants per al SGBD:

– llegir(X):

1. buscar l'adreça del bloc que conté la dada X
2. copiar el bloc a un *buffer* de memòria principal
3. copiar la dada X del *buffer* a la variable X del programa

3.3. Integritat: transaccions

Operacions de les transaccions rellevants per al SGBD:

– escriure(X):

Si no s'ha llegit abans:

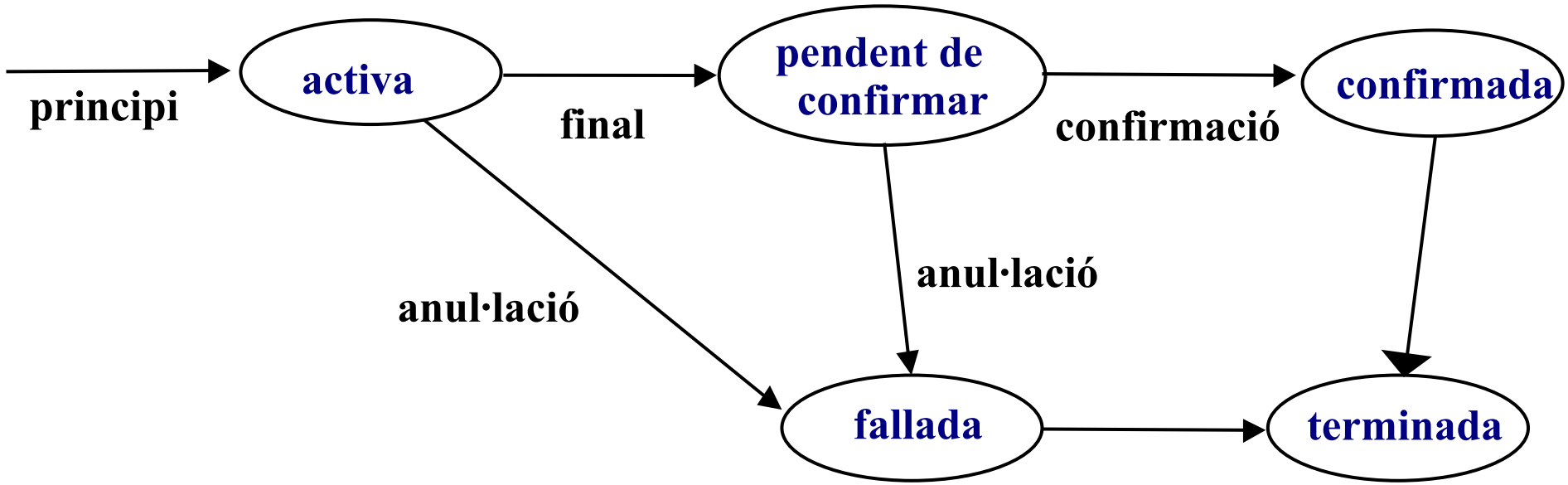
1. cercar l'adreça del bloc que conté la dada X
2. copiar el bloc a un *buffer* de memòria principal
3. copiar la dada X de la variable del programa a la posició adequada en el *buffer*
4. copiar el bloc actualitzat del *buffer* al disc

3.3. Integritat: transaccions

Operacions de definició de les transaccions:

- **principi**: indica el començament de la transacció
- **final**: indica que s'han acabat totes les operacions de la transacció
- **confirmació**: indica l'èxit de la transacció, que permet que el SGBD guarde els canvis efectuats en la base de dades
- **anul·lació**: indica el fracàs de la transacció a causa d'algun motiu. El SGBD desfà tots els possibles canvis efectuats per la transacció

3.3.- Integritat: transaccions



3.3. Integritat: transaccions

Propietats que han de complir les transaccions:

- **atomicitat**: una transacció és una unitat atòmica d'execució (o s'executen totes les seues operacions o cap)
- **consistència**: la transacció ha de donar lloc a un estat de la base de dades consistent (es compleixen totes les restriccions d'integritat)
- **aïllament**: les modificacions introduïdes per una transacció no confirmada no són visibles a la resta de transaccions
- **persistència**: la confirmació implica la gravació dels canvis introduïts en la base de dades, de forma que no es poden perdre per fallada del sistema o d'altres transaccions

3.3. Integritat: transaccions

Dos tipus de funcionament de les transaccions (segons SGBD):

- **Actualització immediata**: les actualitzacions tenen efecte immediat en la memòria secundària i en cas d'anul·lació s'han de desfer.
- **Actualització diferida**: les actualitzacions només tenen efecte immediat en la memòria principal i es transfereixen a la memòria secundària quan es confirmen.

3.3. Integritat: integritat semàntica

- Restricció d'integritat:

  Propietat del món real que modela la base de dades

- Les restriccions es defineixen en l'esquema lògic i el SGBD ha de vetlar pel seu compliment

- La comprovació es realitza quan la base de dades canvia (s'executa una operació d'actualització)

- Les restriccions que no s'inclouen en l'esquema de la base de dades s'han de mantenir en els programes d'aplicació

3.3. Integritat: integritat semàntica

- Tipus de restriccions d'integritat:

  – estàtiques: s'han de complir en cada estat de la base de dades (representable en CRT)

  – de transició: s'han de complir en dos estats consecutius

3.3. Integritat: integritat semàntica

- Restriccions en el SQL/92:
  - estàtiques:
    - ◊ sobre dominis: de valor
    - ◊ sobre atributs: valor no nul, de rang, etc.
    - ◊ sobre relacions: clau primària, unicitat i claus alienes
    - ◊ sobre la base de dades: condicions de cerca generals* (Assertions)
    - quan es comprova: després de cada operació (IMMEDIATE) / al final de la transacció (DEFERRED)
    - accions compensatòries:
  - de transició: s'han de complir en dos estats consecutius*

* no solen mantenir-les els sistemes comercials (es fan amb Triggers).

3.3. Integritat: integritat semàntica

- Procediments de comprovació de la integritat (regles d'activitat, *triggers*...) :
  - programació de la comprovació per part del dissenyador
  - permeten incloure en l'esquema de la base de dades les restriccions complexes
  - en els procediments s'ha d'incloure:
    - ◊ operacions que els activen (esdeveniment i condició)
    - ◊ codi a executar que inclou operacions sobre la base de dades
    - ◊ accions de rebuig o compensació en cas de violació

3.3. Integritat: control d'accessos concurrents

- El SGBD ha de controlar els accessos concurrents de les aplicacions.

- Problemes per interferència d'accessos concurrents:

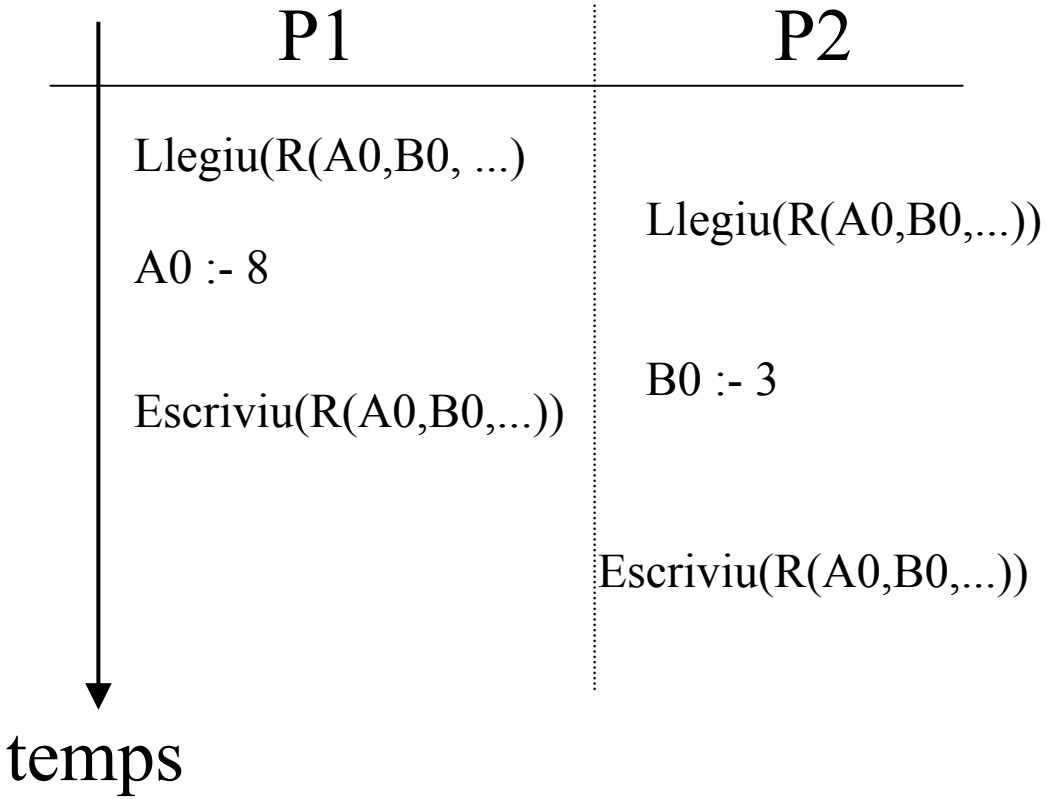
  a) pèrdua d'actualitzacions,

  b) obtenció d'informació incoherent corresponent a diferents estats vàlids de la base de dades, i

  c) lectura de dades actualitzades (no confirmades) que han sigut sotmeses a canvis que encara poden ser anul·lats.

3.3. Integritat: control d'accessos concurrents

a) Pèrdua de les actualitzacions



R

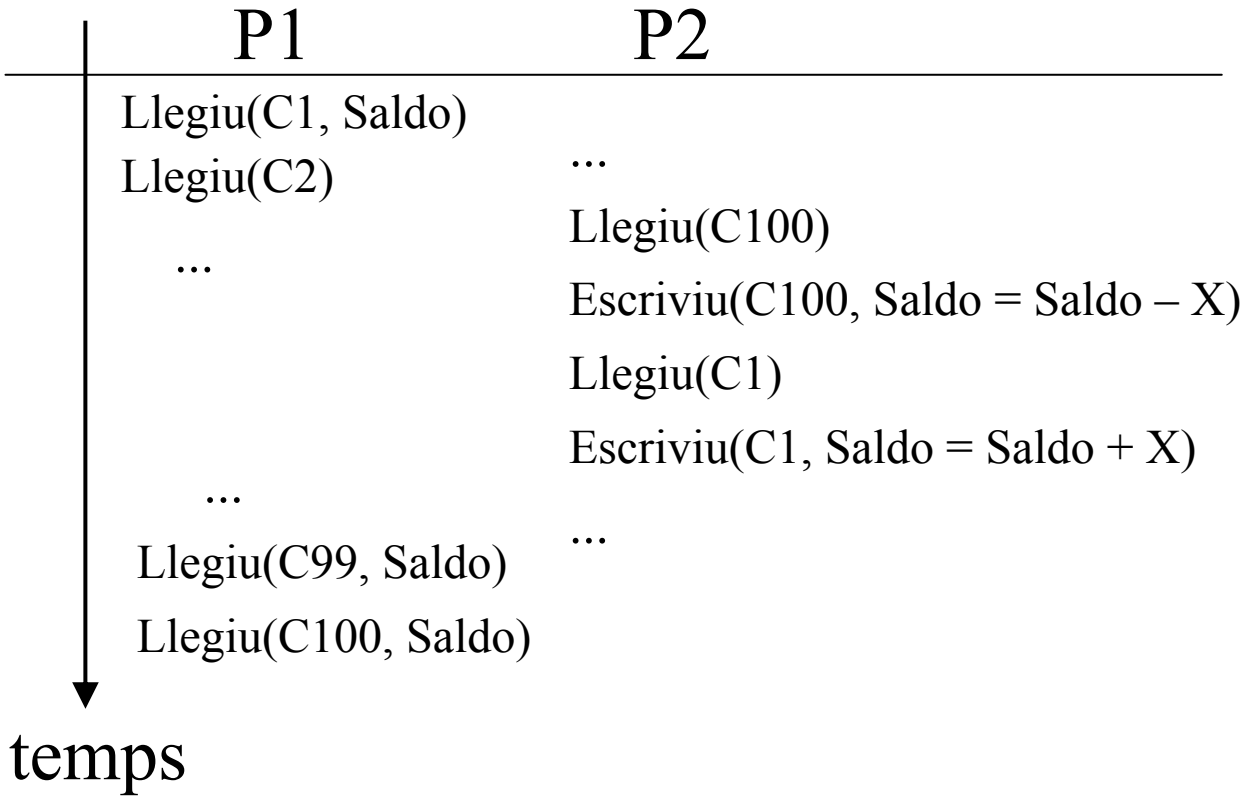
...				
A0	B0	...	...	
...				

3.3. Integritat: control d'accessos concurrents

b) Obtenció d'informació incoherent

P1: Obtenció del total de saldos.

P2: Transferència del compte 100 a l'1.

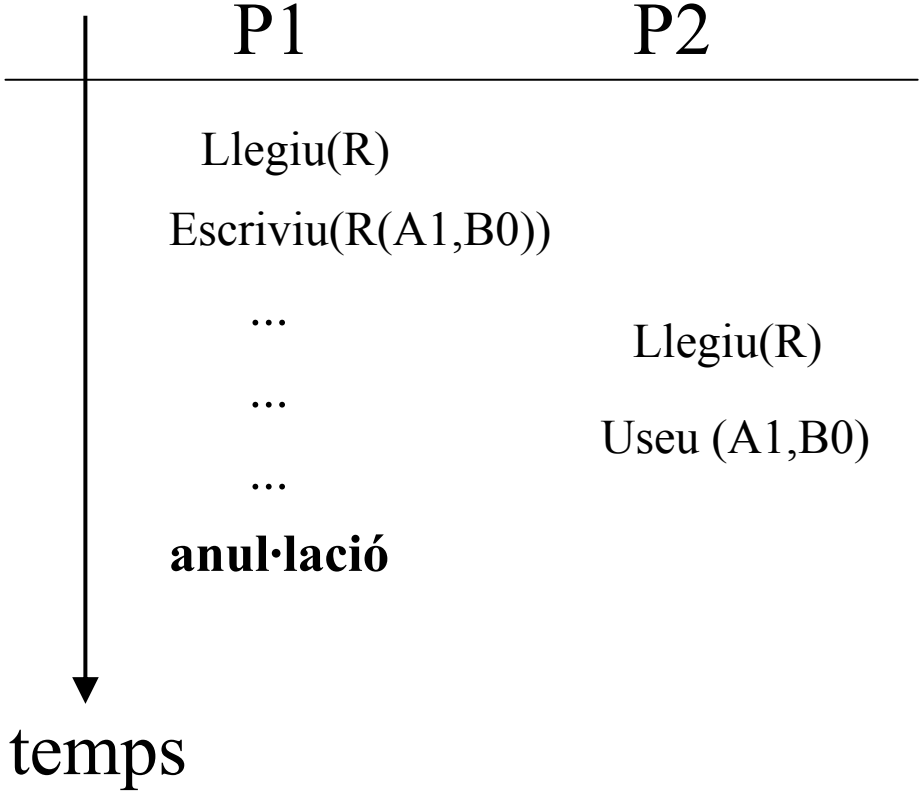


Comptes Corrents

C1	€200000	
C2	...	..
C100	€200000	..

3.3. Integritat: control d'accessos concurrents

c) Lectura de dades actualitzades sense confirmar



R

...				
A0	B0	...	...	
...				

3.3. Integritat: control d'accessos concurrents

Tècniques:

- Reserva d'ocurrències de dades (*locks*)
  - Exemples *a* i *c* es reserva un registre.
  - Exemple *b* es reserven tots.
  - Necessitat de controlar blocatges (*deadlocks*)

- Altres solucions: anul·lació en cascada (per a l'exemple *c*) o aïllament de transaccions (per als exemples *b* i *c*).

3.3.- Integritat: control d'accessos concurrents

Protocols de reserva (locks):

Nou!

Any 2005/2006

- La tècnica més comuna per a tractar l'accés concurrent és l'ús de reserves (*locks*). Si el protocol de reserva està actiu, aleshores:

Cap usuari no pot llegir o escriure si l'usuari no ha reservat la informació prèviament

- Un tros d'informació pot tindre tres estats possibles:
  - Lliure: ningú no pot llegir-lo, però qualsevol pot reservar-lo.
  - Read_lock (o reserva compartida): algú l'ha reservat per a lectura I pot llegir-lo. Alters usuaris poden també reservar-lo per a lectura. Cap altre no pot escriure-lo o reservar-lo per a escriptura.
  - Write_lock (o reserva exclusiva): algú l'ha reservat per a escriptura. Només este usuari pot llegir-lo i escriure-lo. Cap altre no pot reservar-lo.

3.3.- Integritat: control d'accessos concurrents

Protocols de reserva (locks):

Nou!

Any 2005/2006

- Les reserves han de utilitzar-se de manera convenient per a assegurar que les transaccions funcionen adequadament.
- Una manera simple d'aconseguir açò és el 2PL (two phase lock).
  - Totes les reserves en una transacció es fan al principi de la transacció.
  - Tots els alliberaments es fan al final de la transacció.
- Açò assegura que les transaccions funcionen adequadament (en qualsevol situació possible).
- No obstant això, aquesta tècnica encara pot generar bloquejos (deadlocks).

3.3. Integritat: reconstrucció de la base de dades

Les propietats d'atomicitat i persistència d'una transacció obliguen el SGBD a assegurar que:

- si es confirma, els canvis efectuats es graven en la base de dades i no es perden.
- Si s'anul·la, els canvis efectuats sobre la base de dades es desfan.

3.3. Integritat: reconstrucció de la base de dades

Causes de la fallada d'una transacció

- Locals a la transacció (funcionament del sistema normal)
  - errades en la transacció (accés a la base de dades incorrecte, càlculs fallats, etc.)
  - excepcions (violació de la integritat, de la seguretat, etc.)
  - control de la concurrència (estat de blocatge entre dues transaccions)
  - decisions humanes (per programa o explícites)

3.3. Integritat: reconstrucció de la base de dades

Causes de la fallada d'una transacció

- Externes a la transacció (errades del sistema)
  - fallades del sistema amb pèrdua de la memòria principal.
  - fallades del sistema d'emmagatzematge amb pèrdua de la memòria secundària.

3.3. Integritat: reconstrucció de la base de dades

Pèrdues de memòria principal

- En l'espai de temps entre la confirmació d'una transacció i la gravació dels seus camps en memòria secundària.
- La transacció està confirmada i els seus canvis estan en els blocs dels *buffers*.
- En aquest interval es produeix una fallada amb pèrdua de memòria principal i els blocs dels *buffers* es perden.

3.3. Integritat: reconstrucció de la base de dades

Pèrdues de memòria secundària

- Transacció confirmada amb els canvis que estan gravats en la base de dades.
- Fallada en la memòria secundària i aquests canvis es perden.

3.3. Integritat: reconstrucció de la base de dades

Reconstrucció enfront de fallades del sistema

Funcions
- Recuperar transaccions confirmades que no han sigut gravades.
- Anul·lar transaccions que han fallat.

- Mòdul de reconstrucció.
  - Tècnica més estesa: ús del ***fitxer diari*** *(log* o *journal)*.

3.3. Integritat: reconstrucció de la base de dades

Activitats sobre el fitxer diari

- Registrar les operacions d'actualització de les transaccions.
- S'emmagatzema en el disc per a evitar la desaparició per una fallada del sistema.
- Es grava periòdicament en una unitat d'emmagatzematge massiva.

3.3. Integritat: reconstrucció de la base de dades

Tipus d'entrades que es graven en el fitxer diari

- [inici, T]: s'ha iniciat la transacció d'identificador *T*.
- [escriviu, T, X, valor_abans, valor_després]: la transacció *T* ha realitzat una operació d'actualització sobre la dada *X*.
- [llegiu, T, X]: la transacció *T* ha llegit la dada *X*.
- [confirmeu, T]: la transacció *T* ha sigut confirmada.
- [anul·leu, T]: la transacció *T* ha sigut anul·lada.

3.3. Integritat: reconstrucció de la base de dades

Suposarem ACTUALITZACIÓ IMMEDIATA

Fallada d'una transacció $T \rightarrow$ Desfer canvis de T

- Actualitzeu les dades modificades per T amb el seu valor original (*valor_abans*).
- Busqueu les entrades en el diari

  [escriviu, T, X, valor_abans, valor_després]

Fallada del sistema $\rightarrow$ Apliqueu el procés anterior a totes les transaccions sense confirmar

3.3. Integritat: reconstrucció de la base de dades

Fallada del sistema → • Transaccions sense confirmar

[inici, T] en el diari sense [confirmar, T]

• Procés anterior

→ • Transaccions confirmades

[confirmar, T]

• Torneu a executar-les:

[escriviu, T, X, valor_abans, valor_després]

3.3. Integritat: reconstrucció de la base de dades

PROBLEMES:

- Grandària del fitxer diari pot créixer molt ràpidament.
- Recuperació en cas de fallada molt costosa (s'han de refer moltes operacions).

SOLUCIÓ:

Punts de verificació (*checkpoints*)

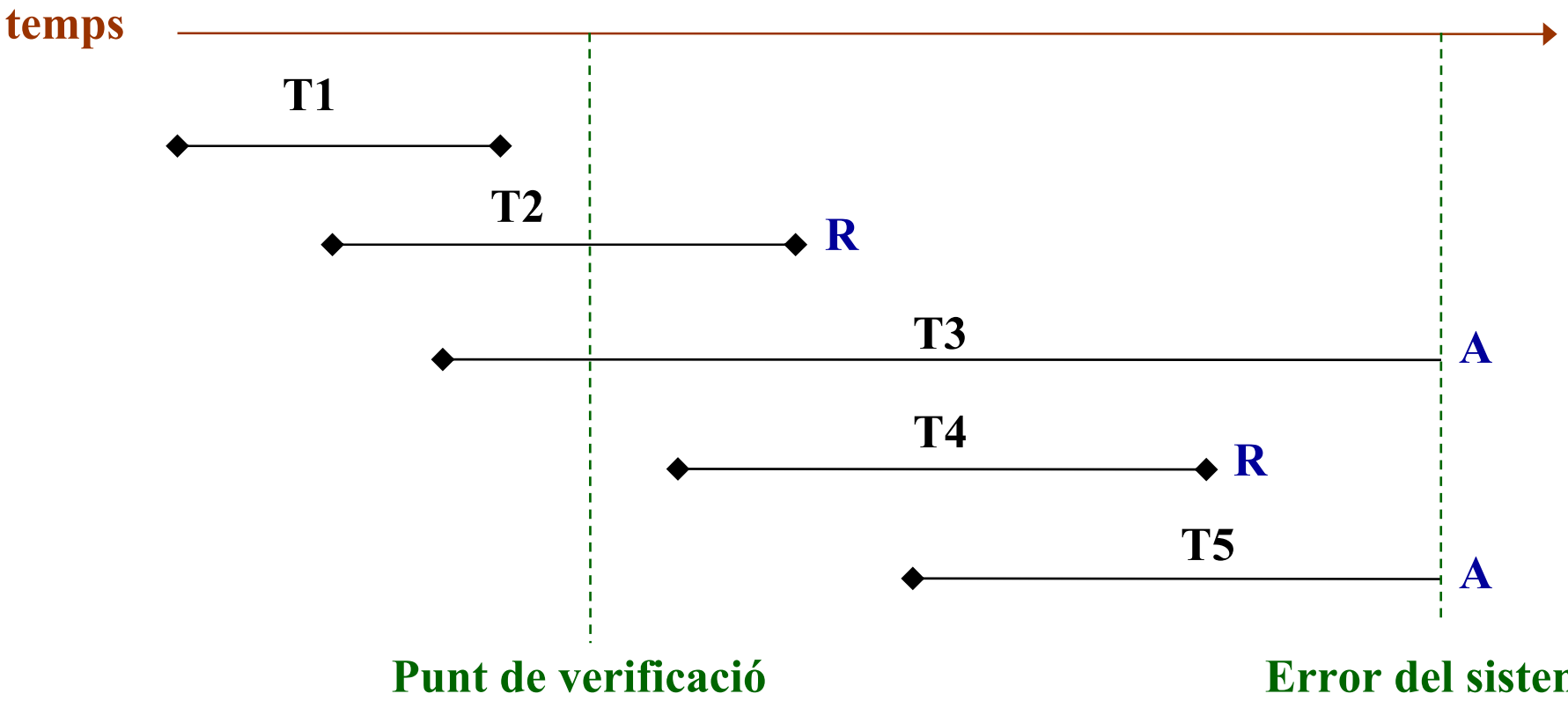
3.3. Integritat: reconstrucció de la base de dades

Punts de verificació → Es graven en el diari periòdicament

- Suspenen temporalment l'execució de transaccions.
- Graven en el diari el punt de verificació.
- Forcen la gravació de totes les actualitzacions de les transaccions confirmades (copieu els *buffers* al disc).
- Reprenen l'execució de les transaccions suspeses.

3.3. Integritat: reconstrucció de la base de dades

Punts de verificació → Reconstrucció a partir de l'últim



3.3. Integritat: reconstrucció de la base de dades

Reconstrucció enfront d'errors del sistema d'emmagatzematge

- Pèrdua de memòria secundària.
- Base de dades pot estar danyada totalment o parcialment.
- Tècnica: reconstruir la base de dades a partir de
  - La còpia de seguretat més recent.
  - A partir de l'instant de la còpia, s'utilitza el diari per a refer les operacions realitzades per les transaccions confirmades.

3.3. Integritat: reconstrucció de la base de dades

Cas ACTUALITZACIÓ DIFERIDA

El mecanisme de reconstrucció és el mateix (les confirmades s'han de repetir), exceptuant-ne:

- Les no confirmades no han de ser desfetes.

3.4. Seguretat

Objectiu:

Només poden accedir a la informació les persones i processos autoritzats i en la forma autoritzada.

3.4. Seguretat

Tècniques:

- Identificació de l'usuari.
- Determinació dels accessos permesos:

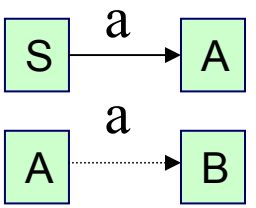
Modes:
  - Llista d'autoritzacions (objecte i operacions permeses) per usuari.
  - Nivells d'autorització (menys flexible).

- Gestió d'autoritzacions transferibles: traspàs d'autoritzacions d'un usuari a un altre.

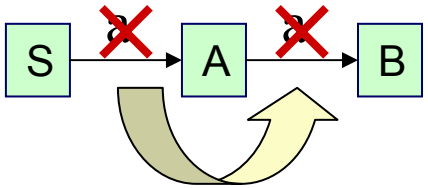
3.4. Seguretat

Requeriments per a realitzar la gestió d'autoritzacions transferibles:

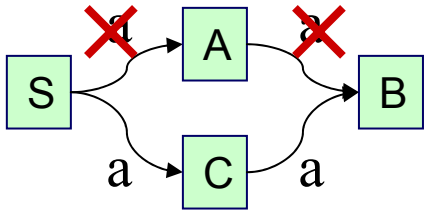
- Coneixement de les autoritzacions d'accés de cada usuari (quines són transferibles a tercers i quines no).
- Transferència d'una autorització d'un usuari a un altre (en mode transferible o no).



- Revocació posterior d'una autorització d'accés:
  - Si es va atorgar en mode transferible, revocació de les autoritzacions que partiren d'aquesta.



- Revocació independent d'una autorització d'accés atorgada de forma múltiple.



B ha de conservar el permís 'a'

3.5.- Implementació de BD relacionals.

ESQUEMA FÍSIC:

- Descripció de la BD en termes de la seua representació física (sobre dispositiu d'emmagatzematge secundari).

- L'arquitectura de tres nivells de un SGBD permet treballar amb una base de dades sense saber res dels seus detalls d'implementació. No obstant això, els detalls físics són importants per al comportament de la base dades en termes de temps de resposta i espai d'emmagatzematge.

3.5.- Implementació de BD Relacionals.

Tècniques de Bases de Dades:

- ✓Grans volums de dades
- ✓Persistència

⇓ Tecnologia actual

DB s'emmagatzemen en memòria secundària **(discs)**

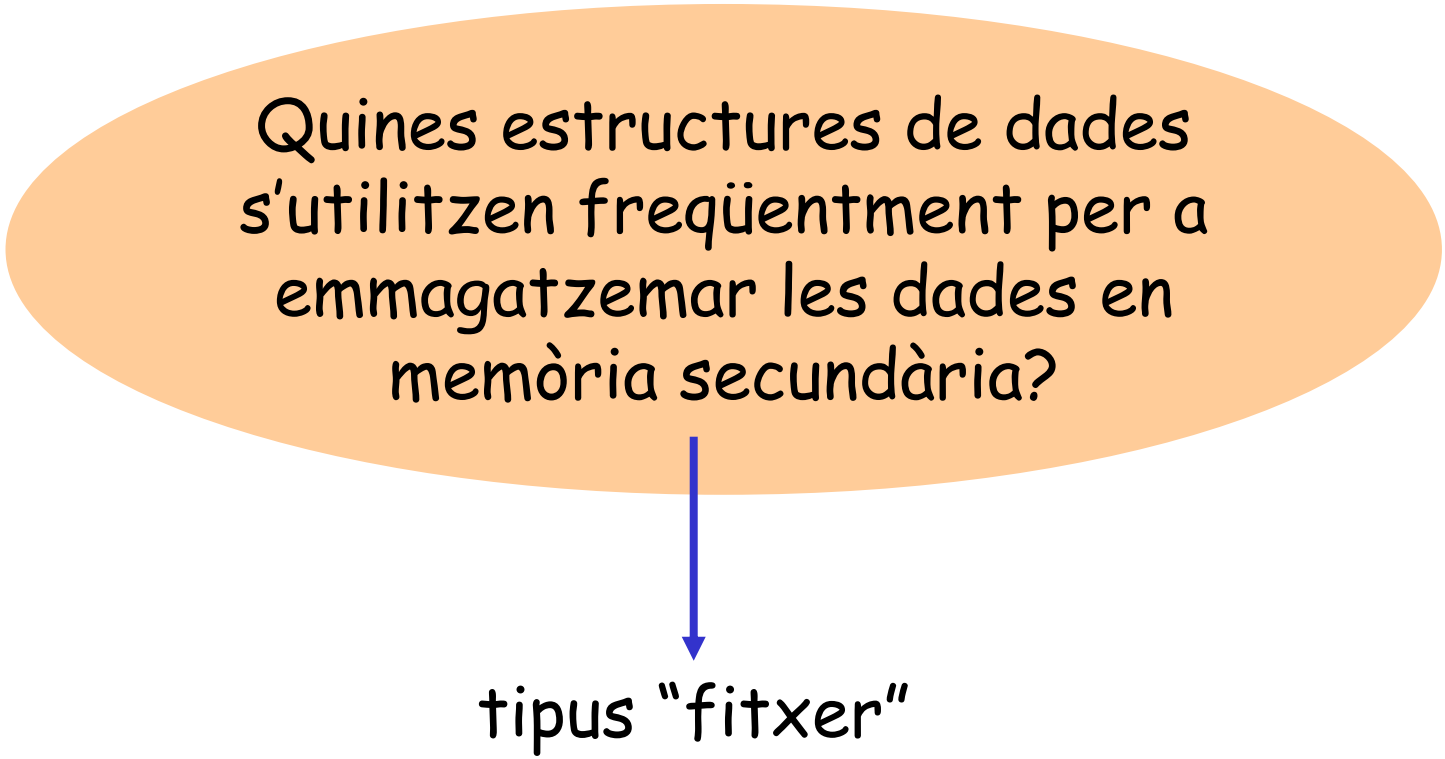
⇓ L'accés a memòria secundària és molt més lent que l'accés a memòria principal

Tria una implementació de les taules que **reduïsca el nombre d'accessos a disc**

anàlisi

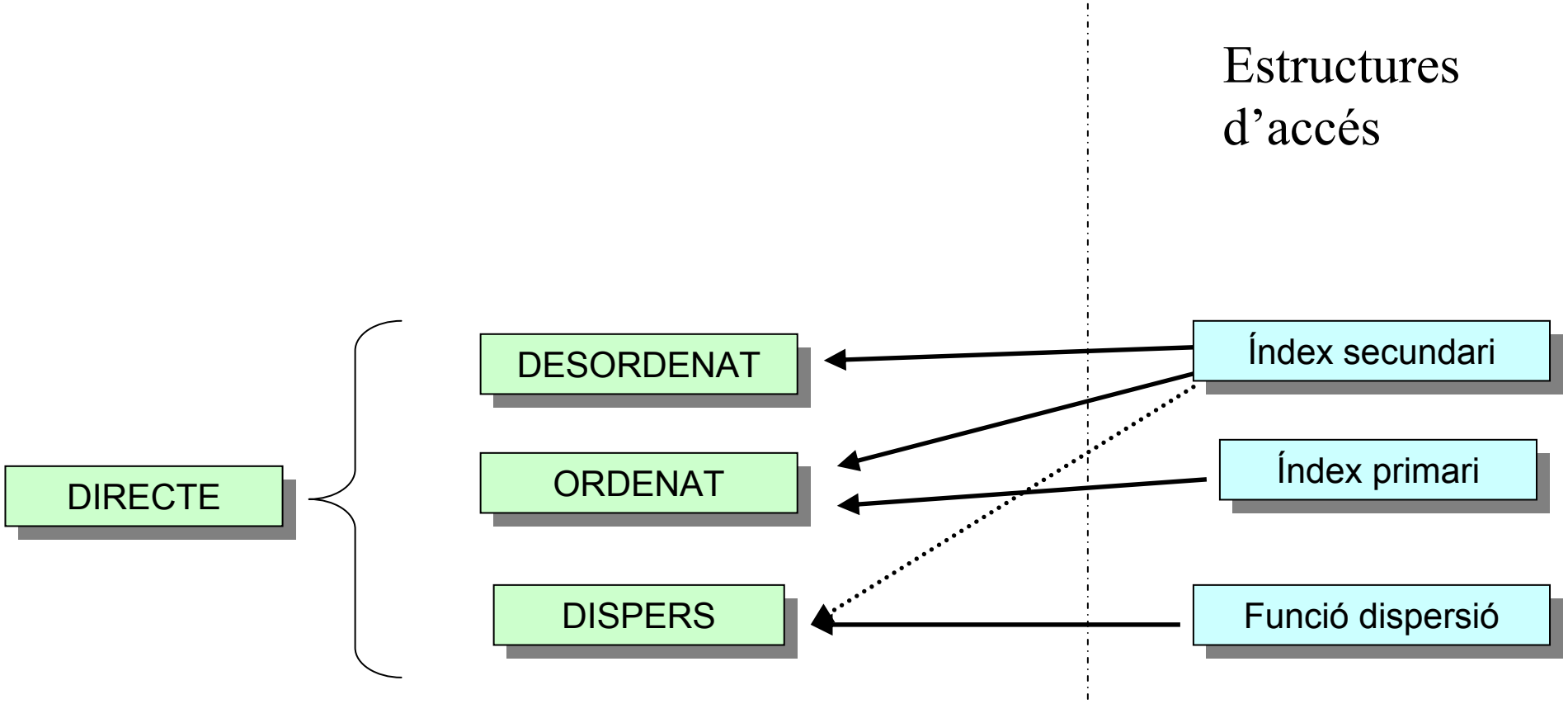
✓ característiques de les estructures de dades que s'utilitzen per a la memòria secundària (fitxers)

3.5.- Implementació de BD relacionals.



3.5.1. Conceptes previs

CLASSIFICACIÓ D'ORGANITZACIÓ DE FITXERS



3.5.2. Fitxers directes

- Inclou aquells models d'organització de fitxers que permeten accedir a la localització exacta del registre buscat.

- L'accés directe es pot aconseguir per diferents mètodes:
  - encaminament relatiu
  - dispersió
  - ús d'índexs

3.5.2.1. Fitxers directes ordenats

Fitxer ordenat amb encaminament relatiu:

- és un fitxer en què els registres s'emmagatzemen ordenats d'acord amb el valor d'un dels seus camps i que permet encaminament relatiu.

- Avantatges:
  - lectura en ordre del camp d'ordenació molt eficient.
  - trobar el següent en ordre del camp d'ordenació no requereix accessos addicionals (excepte en l'últim registre d'un bloc).
  - la recerca basada en el camp d'ordenació pot ser **binària** sobre els blocs.

3.5.2.1 Fitxers directes ordenats

- Inconvenients:
  - L'accés basat en un camp distint al d'ordenació obliga a buscar en tot el fitxer fins que es trobe.
  - La inserció és molt costosa, perquè s'ha de localitzar el lloc on s'ha de situar el registre, i desplaçar els registres posteriors per a fer lloc. Sol $\rightarrow$ buits o desbordament.
  - L'esborrament no és tan problemàtic si es marca el registre, però sense recuperar-ne l'espai.
  - La modificació del valor del camp d'ordenació pot canviar la seua posició en el fitxer, cosa que suposa un esborrament i una inserció.

3.5.2.1. Fitxers directes ordenats

- El cost de la modificació d'un camp distint del d'ordenació depén només de la condició de recerca del registre a modificar.
- Per a disminuir el gran cost en el cas d'inserció o modificació del valor del camp d'ordenació, hi ha dos solucions:
  - ús de **buits** en els blocs per a només haver de reorganitzar com a màxim la grandària del bloc. Problemes: espai perdut i reorganitzacions periòdiques quan s'omplin.
  - ús d'un fitxer temporal, anomenat de **desbordament** o de **transacció**, per a anar afegint els registres nous i que es mescla periòdicament amb el **principal** en el procés de **reorganització.**
- Els fitxers directes amb encaminament relatiu (desordenats o ordenats) només se solen usar en BD amb índexs.

3.5.2.2. Fitxers directes dispersos

Fitxer dispers (hash file):

Caracterització:

- Aquesta tècnica proporciona un accés molt ràpid quan la condició de recerca és el valor del **camp de dispersió**, que generalment és una clau.
- Hi ha una **funció** anomenada **de dispersió** o aleatorització que s'aplica al valor del camp de dispersió d'un registre i retorna l'adreça del bloc del disc on es guardarà el registre.
- Per a la recuperació de la majoria de registres es necessita un únic accés al disc.

3.5.2.2. Fitxers directes dispersos

Manera de funcionament:

- L'espai d'adreces assignat al fitxer es compon de ***poals*** (*buckets*), en cadascun dels quals caben molts registres.
- Suposem que tenim M poals, amb unes adreces relatives que oscil·len entre 0 i M–1. Hem d'escollir una funció que transforme el valor del camp de dispersió en un enter entre 0 i M–1.
- Una funció de dispersió comuna és $d(K) = K \bmod M$, que retorna la resta de dividir el valor K del camp de dispersió per M.

3.5.2.2. Fitxers directes dispersos

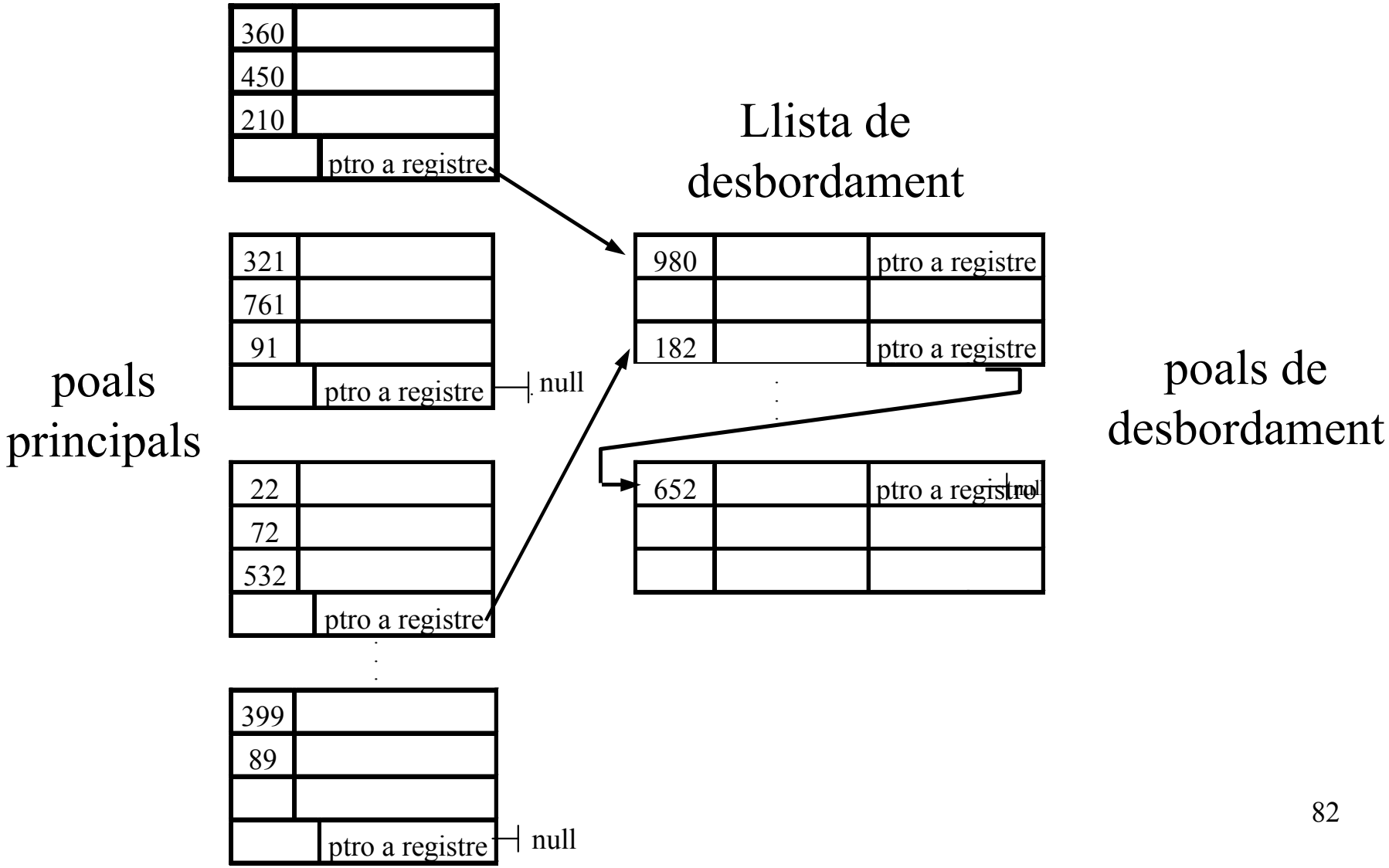
PROBLEMA:

- La majoria de les funcions de dispersió no poden garantir que per a distints valors s'obtindran adreces diferents, perquè el nombre de valors distints que pot prendre el camp de dispersió és molt major que el nombre d'adreces disponible per als registres.

SOLUCIÓ:

- Aplicar tècniques de resolució de col·lisions:
  - encaminament obert: a blocs següents
  - encadenament: llista de desbordament
  - dispersió múltiple: s'aplica una 2a f. de dispersió

3.5.2.2. Fitxers directes dispersos



3.5.2.2. Fitxers directes dispersos

- Avantatges:
  - Proporciona un accés molt ràpid per a localitzar un registre arbitrari atés el seu valor del camp de dispersió.

- Inconvenients:
  - No és molt útil quan es requereixen altres aplicacions per al mateix fitxer, si no és que es construeixen camins d'accés addicionals.
  - L'espai reservat per als fitxers és fix (es desaprofita molt d'espai al principi i sol estar desbordat amb el temps).

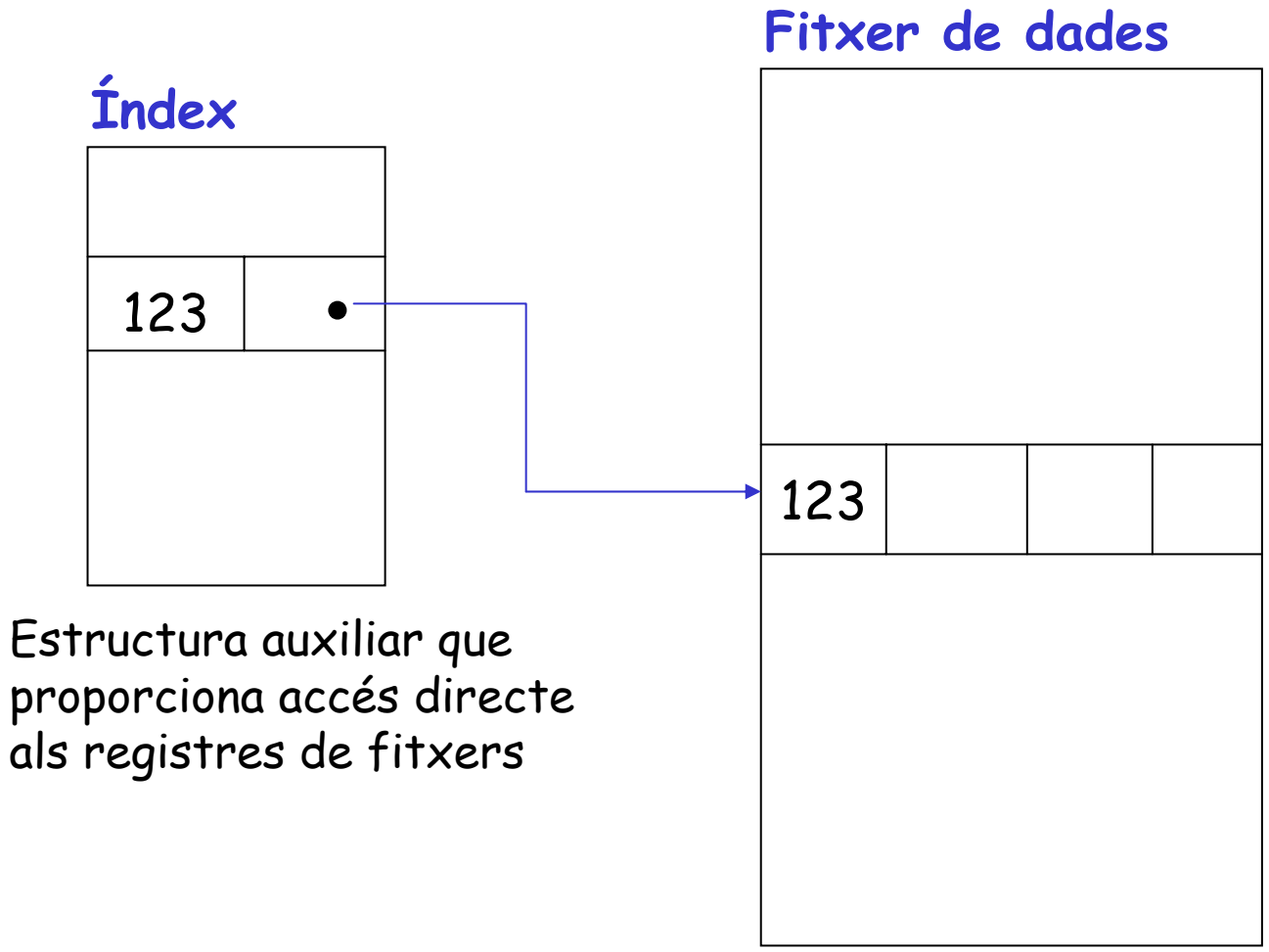
3.5.2.3. Fitxers directes indexats

ÍNDEXS:

- Un índex és una estructura d'accés definida sobre un o diferents **camps d'indexació**.
- Consisteix en un fitxer addicional amb registres (o entrades) que estan constituïts per dos camps: clau (camp d'indexació) i adreça.
- No afecten el fitxer original.
- Un fitxer pot tenir diferents índexs per a distints camps.

Els índexs proporcionen accés directe i accés ordenat als registres de fitxers que no estan ordenats pel camp indexat.

3.5.2.3.- Fitxers indexats



Estructura auxiliar que proporciona accés directe als registres de fitxers

Fitxers de dades amb una organisation seqüencial o dispersa.

3.5.3.- Elecció de l'esquema físic

- Opcions que es proporcionen pel SGBD per a implementar una BD:
  - Diferents tipus d'organitzacions: ordenada o desordenada, seqüencial, dispersa..
  - Diferents tipus de fitxers dispersos: funció de dispersió, blocs de desbordament,...
  - Diferent classes d'índexs.
  - Agrupament (clustering) de taules relacionades
  - Enllaços a través de punters a registres relacionats

3.5.3.- Elecció de l'esquema físic

- Quan es defineix un esquema (amb CREATE TABLE) podem expressar:

La definició de dades de la taula

Organització de fitxers

+

Índexs

3.5.3. Elecció de l'Esquema Físic

- Cada SGBD ofereix una varietat d'opcions per a l'organització dels fitxers.

- El dissenyador de la base de dades ha de tenir en compte:
  - factors de temps de resposta necessari,
  - utilització d'espai pels fitxers i les seues estructures d'accés,
  - freqüència d'execució de determinades consultes i transaccions,
  - altres requisits especificats per a la base de dades.

3.5.3. Elecció de l'Esquema Físic

- Els atributs que s'espera que siguen utilitzats freqüentment per a recuperar registres han de tenir definits sobre ells camins d'accés primari o índexs secundaris.
- A vegades es fa necessari reorganitzar alguns fitxers construint índexs nous o canviant els mètodes d'accés primaris.
- Una opció molt popular per a organitzar un fitxer en un sistema relacional és mantenir els registres del fitxer desordenats i crear tants índexs secundaris com es necessiten.

3.5.3. Elecció de l'Esquema Físic

- Si els registres es recuperaran freqüentment en ordre d'un atribut: **ordenació** sobre aquest atribut, amb l'índex primari corresponent, si no variarà molt.
- Si el fitxer sofrirà moltes insercions i esborrats, s'ha d'intentar **minimitzar el nombre d'índexs**.
- En molts sistemes l'**índex** no és una part integral del fitxer, sinó que es crea o destrueix **dinàmicament**.
- Si un fitxer **no** s'usarà sovint per a **recuperar registres en ordre**, es pot usar **dispersió** (que ha de ser dinàmica si variarà amb freqüència la grandària del fitxer).

3.5.3. Elecció de l'Esquema Físic

- Quan dues relacions tenen sengles atributs mitjançant els quals és habitual realitzar concatenacions, se solen utilitzar **agrupacions** o *clusters* (grup de sectors) de relacions.
- Una agrupació consisteix a guardar físicament en el mateix bloc les tuples de dos relacions habitualment concatenades.

EXEMPLE:

- Siguen **R1** i **R2** dues relacions que es concatenen habitualment i amb esquemes de relació:

R1(a1:dom1, a2:dom2)

CP:{a1}

R2(b1:dom3, b2:dom4, b3:dom1)

CP:{b1}

Caj:{b3} $\rightarrow$ R1

3.5.3. Elecció de l'Esquema Físic

EXEMPLE (cont.):

- Extensió de R1 i R2:

R1

a1	a2
12	Dotze
51	Cinquanta-u
84	Vuitanta-quatre

R2

b1	b2	b3
9A	ASDF	84
0B	QWER	51
1L	ZXCV	12
2X	QAZ	12
3P	POIU	84
4K	MNBV	51
5T	TTTT	51
6M	MMM	12

3.5.3. Elecció de l'Esquema Físic

EXEMPLE (cont.):

- Agrupació de R1 i R2:

BLOC 1

a1	**a2**	
12	Dotze	
	b1	**b2**
	1L	ZXCB
	2X	QAZ
	6M	MMM

BLOC 2

a1	**a2**	
51	Cinquanta-u	
	b1	**b2**
	0B	QWER
	4K	MNBV
	5T	TTTT

BLOC 3

a1	**a2**	
84	Vuitanta-quatre	
	b1	**b2**
	9A	ASDF
	3P	POIU

3.5.3. Elecció de l'Esquema Físic

AVANTATGES:

- Les agrupacions permeten:
    - Reduir el temps d'accés en concatenacions.
    - Estalvi d'espai d'emmagatzematge: la clau de l'agrupació només s'emmagatzema una vegada.

INCONVENIENTS:

- Les agrupacions redueixen el rendiment en insercions o modificacions.