

Disseny de la Capa de Gestió de Dades

- Introducció
- Tecnologia de les Bases de Dades Relacionals
- Disseny Lògic:
 - Traducció diagrama de classes
 - Restriccions d'Integritat
 - Informació Derivada
- Disseny Operacions
- Bibliografia

Què és i com s'aconsegueix la persistència?

- **Persistència:**
 - és la capacitat que molts sistemes software requereixen per emmagatzemar i obtenir dades usant un sistema d'emmagatzematge permanent.
- **Els objectes es poden fer persistents en:**
 - Bases de dades orientades a objectes
 - Bases de dades relacionals
 - Bases de dades objecte-relacional
 - etc.

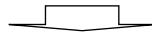
Com hi influeix la tecnologia de les Bases de Dades?

Dependència tecnològica:

- Propietats que es volen assolir (requisits no funcionals)
- Recursos tecnològics disponibles
 - família del llenguatge de programació
 - **família del sistema gestor de bases de dades (SGBD)**

Determina:

- L'arquitectura del sistema software i els patrons que s'usaran



El disseny que fem depèn del SGBD utilitzat

Ens centrem la persistència usant SGBD relacionals

Tecnologia de les Bases de Dades Relacionals

Components de dades

- Esquema de base de dades:
Unitat administrativa per agrupar components.
- Relacions (taules):
Permeten emmagatzemar informació físicament.
Un esquema de relació es denota per: $R(Atr_1, Atr_2, \dots, Atr_n)$
- Restriccions:
Estableixen condicions que la base de dades ha de satisfer.
Són gestionades pel propi SGBD.
- Vistes:
Són relacions derivades (calculades), no emmagatzemades físicament.
El seu esquema i contingut es deriven a partir d'una consulta relacional.

Tecnologia de les Bases de Dades Relacionals

Components de procés

- Procediments emmagatzemats (*stored procedures*):
Funcions que s'emmagatzemen a la BD i es tracten com uns objectes més de la BD.
- Disparadors (*triggers*):
Són regles ECA (Esdeveniment, Condició, Acció) que permeten executar una Acció quan es produeix un Esdeveniment i se satisfà la Condició.

Els exemples que posarem estan fets en el SGBD Informix

Tecnologia de les Bases de Dades Relacionals

Disseny lògic de la base de dades

- **Traducció del diagrama de classes**
 - Cas Domain Model: partim del diagrama normalitzat (sense les classes provinents de l'aplicació dels patrons)
 - Cas Transaction Script: partim del diagrama d'especificació o el normalitzat
- **Tractament de les restriccions d'integritat i informació derivada**
 - Fem ús dels elements proporcionats per la tecnologia relacional (Claus, Triggers, Procediments, ...)
- **Assignació de comportament**
 - Les responsabilitats assignades a la capa de gestió de dades, s'implementen fent ús dels elements proporcionats per la tecnologia relacional (Sql, Procediments, Triggers, ...)

Eventualment, es pot decidir implementar íntegrament a la capa de dades alguna responsabilitat assignada conjuntament a les capes de domini + dades

Tecnologia de les Bases de Dades Relacionals

Disseny lògic de la BD

- L' **esquema conceptual** conté **classes d' objectes**
- Els **SGBDs relacionals** implementen **taules relacionals** del tipus:
taula1 (atr1, atr2, ..., atrn), on atr1 = clau primària i atrn = clau forana



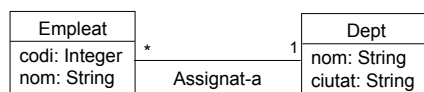
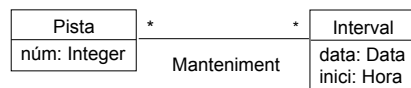
Disseny lògic

- Traducció dels elements de l' esquema conceptual a components implementables pels SGBDs relacionals.
 - Classes → taules
 - Associacions entre classes d' objectes → taules, atributs, vistes
 - Aspectes no contemplats per les taules relacionals:
 - associacions n -àries, amb $n > 2$
 - classes associatives
 - associacions binàries * - *
 - jerarquies d'especialització (i, per tant, herència)
 - identificadors interns
 - operacions associades a les classes d' objectes

} Tractament ja conegut; transformacions ja realitzades si hem normalitzat

Tecnologia de les Bases de Dades Relacionals

Disseny lògic de la BD, associacions binàries

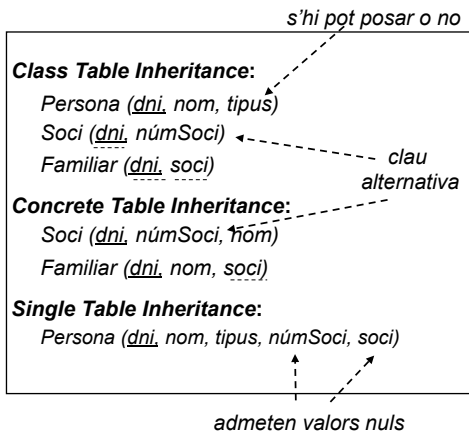
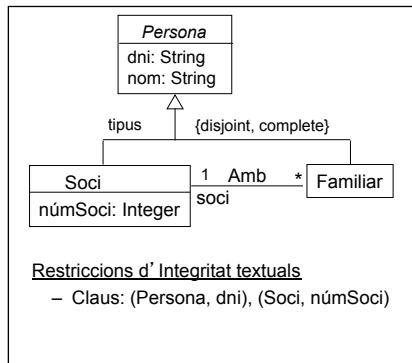
**Es tradueix a:***Dept (nom-d, ciutat)**Empleat (codi, nom-emp, nom-d)***Es tradueix a:***Interval (data, inici)**Pista (número)**Manteniment (núm-pista, data, inici)*

Nota: Es consideren intervals d'una hora

Tecnologia de les Bases de Dades Relacionals

Disseny lògic de la BD, jerarquies d'especialització (1)

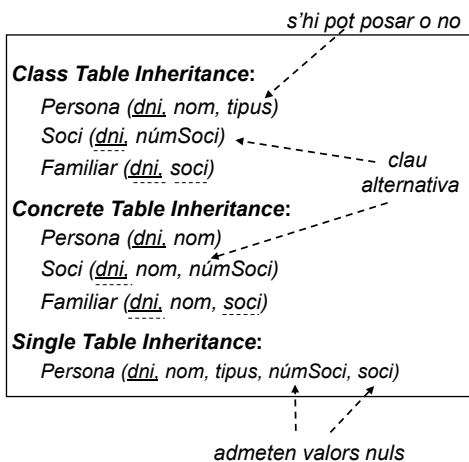
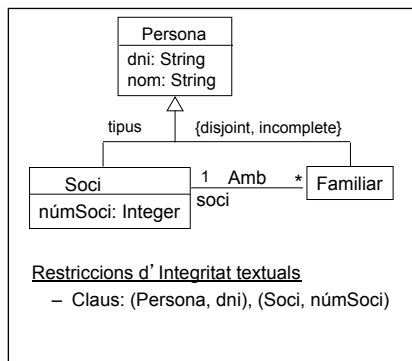
La traducció depèn de fins a quin nivell es vol col·lapsar o no la jerarquia



Tecnologia de les Bases de Dades Relacionals

Disseny lògic de la BD, jerarquies d'especialització (1, variant)

La traducció depèn de fins a quin nivell es vol col·lapsar o no la jerarquia



Tecnologia de les Bases de Dades Relacionals

Disseny lògic de la BD, jerarquies d'especialització (2)

Estratègia	Avantatges	Desavantatges
<i>Class Table Inheritance</i>	Simple Canviable	Poc eficient (múltiples accessos per objecte)
<i>Concrete Table Inheritance</i>	Eficient (un accés per objecte)	Poc canviable (propagació de canvis fets a les superclasses)
<i>Single Table Inheritance</i>	Eficient (un accés per objecte) Canviable	Pèrdua d'espai (però la BD pot ajudar)

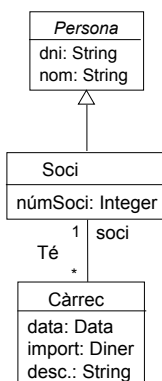
No hi ha una estratègia clarament millor

- consulteu l' administrador de Bases de Dades!

Tecnologia de les Bases de Dades Relacionals

Disseny lògic de la BD, classes sense clau externa

- Cal afegir alguna clau externa "artificial"
 - Normalment acostuma a ser una clau mantinguda pel propi sistema



Restriccions d'Integritat textuals

- Claus: (Persona, dni), (Soci, numSoci)

La taula Càrrec es tradueix a:

Càrrec(id-càrrec, data, import, desc., soci)

Identificador "artificial"

Tecnologia de les Bases de Dades Relacionals

Tractament de les restriccions d'integritat

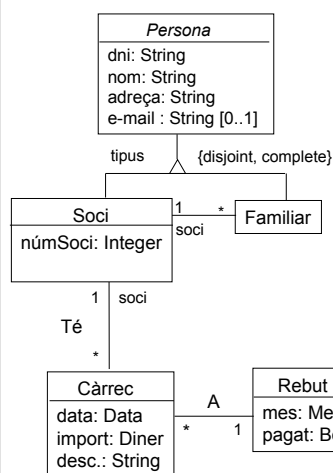
- Els SGBD relacionals proporcionen diverses funcionalitats per tractar les restriccions d'integritat:
 - restriccions de columna
 - not null
 - distinct
 - unique
 - primary key
 - references taula (foreign key)
 - check (condició)
 - etc.



Podem assignar directament al SGBD relacional les responsabilitats corresponents

Tecnologia de les Bases de Dades Relacionals

Tractament de les restriccions d'integritat, exemple



Persona (dni, nom, adreça, e-mail)

Familiar (dni, soci)

Soci (dni, númeroSoci)

Rebut (dniSoci, mes, pagat)

Càrrec (id-càrrec, data, import, descripció, soci, mes)

Restriccions d'Integritat textuals

- Clau: (Persona, dni)
=> Clau primària de *Persona*: dni
- Clau: (Soci, númeroSoci)
=> Clau primària de *Soci*: dni i
=> clau alternativa númeroSoci: unique(númeroSoci)
- No hi pot haver dos rebuts amb el mateix soci i mes
=> Clau primària de *Rebut*: (dniSoci, mes)
- Tots els càrrecs d'un rebut tenen una data dins de mes
=> Check de *Càrrec*: check(month(data) = mes)

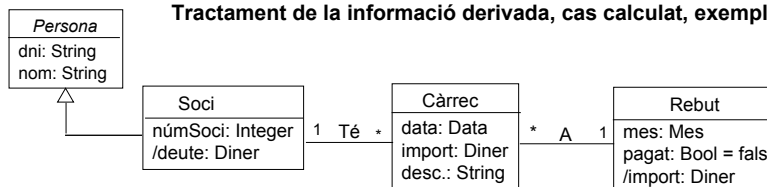
Tecnologia de les Bases de Dades Relacionals

Tractament de la informació derivada

- Els atributs i les associacions derivats es poden **calcular** o **materialitzar**
- Si es *calcula*:
 - Cal assignar a algú la responsabilitat de calcular el valor d'aquesta informació.
 - A les operacions de la capa de domini
 - A la capa de gestió de dades (al propi SGBD), mitjançant **vistes** que, quan es consulten, proporcionen automàticament la informació especificada.
- Si es *materialitza*:
 - Cal assignar a algú la responsabilitat de mantenir consistent el valor d'aquesta informació.
 - A les operacions de la capa de domini
 - A la capa de gestió de dades (al propi SGBD), normalment mitjançant **disparadors** (*triggers*)
- En la decisió hi influeix el temps de càlcul, la freqüència d'accés i l'espai ocupat.

Tecnologia de les Bases de Dades Relacionals

Tractament de la informació derivada, cas calculat, exemple



Deute = suma import de rebuts no pagats
Import = suma import dels càrrecs

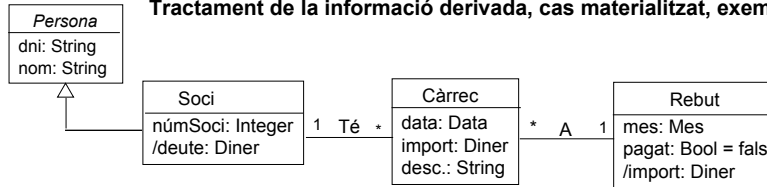
Soci (*dni*, *númSoci*)
Rebut (*dniSoci*, *mes*, *pagat*)
Càrrec (*id-car*, *data*, *import*, *desc.*, *soci*, *mes*)

```
CREATE VIEW import-rebut [dniSoci, mes, imp] AS
SELECT  r.dniSoci, r.mes, sum(c.import)
FROM    Càrrec c, Rebut r
WHERE   c.soci = r.dniSoci AND
        c.mes = r.mes
GROUP BY dniSoci, mes
```

```
CREATE VIEW deute-soci [dniSoci, deute] AS
SELECT  s.dni, sum(c.imp)
FROM    Soci s, Càrrec c, Rebut r
WHERE   s.dniSoci = c.soci AND
        c.soci = r.dniSoci AND
        c.mes = r.mes AND
        r.pagat = 'F'
GROUP BY dniSoci
```


Tecnologia de les Bases de Dades Relacionals

Tractament de la informació derivada, cas materialitzat, exemple



Deute = suma import de rebuts no pagats
Import = suma import dels càrrecs

Soci (dni, númSoci, deute)
Rebut (dniSoci, mes, pagat, import)
Càrrec (id-car, data, import, desc., soci, mes)

Dues alternatives:

- Assignar la responsabilitat a la capa de domini
- Assignant la responsabilitat de materialitzar al SGBD:
 - *Mitjançant disparadors*

```

CREATE TRIGGER inc-import INSERT ON Càrrec
REFERENCING new AS nou
FOR EACH ROW
  (UPDATE Rebut r SET r.import = r.import + nou.import
   WHERE nou.dniSoci = r.dniSoci AND nou.mes = r.mes)
  
```

```

CREATE TRIGGER dec-import DELETE ON Càrrec
REFERENCING old AS vell
FOR EACH ROW
  (UPDATE Rebut r SET r.import = r.import - vell.import
   WHERE vell.dniSoci = r.dniSoci AND vell.mes = r.mes)
  
```

- *Alguns SGBD poden permetre materialitzar la informació directament*

Tecnologia de les Bases de Dades Relacionals

Disseny d'operacions, assignació de responsabilitats al SGBD

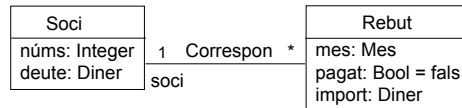
- Els llenguatges actuals d'especificació no permeten definir un comportament actiu de l'esquema conceptual.
- En canvi, els SGBD relacionals sí que ho permeten.
- Per tant, en alguns casos els contractes d'especificació defineixen aspectes que el SGBD pot gestionar directament.



A disseny es pot assignar aquesta responsabilitat al SGBD

Tecnologia de les Bases de Dades Relacionals

Disseny d'operacions, assignació de responsabilitats al SGBD, exemple



Soci (dni, númSoci, deute)

Rebut (soci, mes, pagat, import)

context CapaDeDomini::BaixaSoci(dniSoci: Integer)
exc soci-no-existeix: no existeix un soci amb dniSoci
post 2.1: S'esborra el soci dniSoci
2.2 - S'esborren tots els rebuts del soci

*Es passa
al SGBD*

CREATE TABLE Soci (
 dni STRING PRIMARY KEY,
 númSoci INTEGER,
 deute IMPORT,
 UNIQUE númSoci)

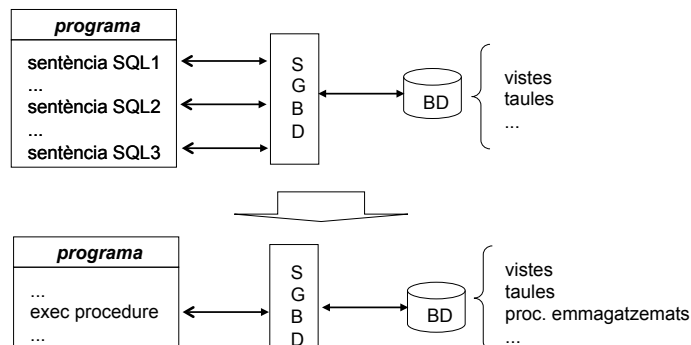
**En esborrar soci, el SGBD
 esborra automàticament els
 seus rebuts**

CREATE TABLE Rebut (
 soci STRING,
 mes MES,
 pagat BOOLEAN,
 import IMPORT,
 PRIMARY KEY (soci, mes),
 FOREIGN KEY (soci) references(Soci) ON DELETE CASCADE)

Tecnologia de les Bases de Dades Relacionals

Disseny d'operacions, procediments emmagatzemats

- Serveixen per:
 - Simplificar el desenvolupament d'aplicacions
 - Millorar el rendiment de la base de dades
 - Controlar les operacions que els usuaris realitzen contra la base de dades
- Poden comprometre la portabilitat



Tecnologia de les Bases de Dades Relacionals

Disseny d'operacions, procediments emmagatzemats

Soci (dni, númSoci, deute, núm-cc)

Rebut (dniSoci, mes, pagat, import)

CompteCorrent (núm-cc, saldo)

```
CREATE PROCEDURE PagarRebut (dniS, mesEmissió)
RETURNING INTEGER, CHAR(50);
DEFINE      codi-error INTEGER, ....
ON EXCEPTION SET codi-error, miss-error; RETURN codi-error, miss-error END EXCEPTION;

IF ((SELECT COUNT(*) FROM rebut WHERE mes=mesEmissió AND dniSoci=dniS)=1) THEN
  LET import,l-pagat = (SELECT import, pagat FROM rebut WHERE mes=mesEmissió AND dniSoci=dniS);
  IF ( 'Y' = l-pagat ) THEN RAISE EXCEPTION 2, 'El rebut ja està pagat' ;
  ELIF LET saldo, núm-cc = (SELECT c.saldo,c.núm-cc FROM soci s, comptecorrent c WHERE s.núm-cc=c.núm-cc
                           and s.dni=dniS);
    IF saldo < import THEN RAISE EXCEPTION 3, 'El soci no té prou saldo' ;
    ELSE UPDATE rebut SET pagat = 'Y' WHERE mes=mesEmissió AND dniSoci=dniS;
          UPDATE soci s SET s.deute = s.deute-import WHERE númSoci = s.dni;
          UPDATE comptecorrent c SET c.import = c.import-import WHERE núm-cc =c.núm-cc ENDIF
ELSE RAISE EXCEPTION 1, 'El soci no té rebut aquest mes' ; END IF;
RETURN 0,'Rebut Pagat' ;
END PROCEDURE;
```

Bibliografia

- M. Fowler
Patterns of Enterprise Application Architecture
Addison-Wesley, 2003
- H. Garcia-Molina, J. Ullman, J. Widom
Database Systems Implementation
Prentice-Hall, 2000.