

Capítol 4. Persistència i bases de dades

Autors: Xavier Franch, Lidia López



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

Índex

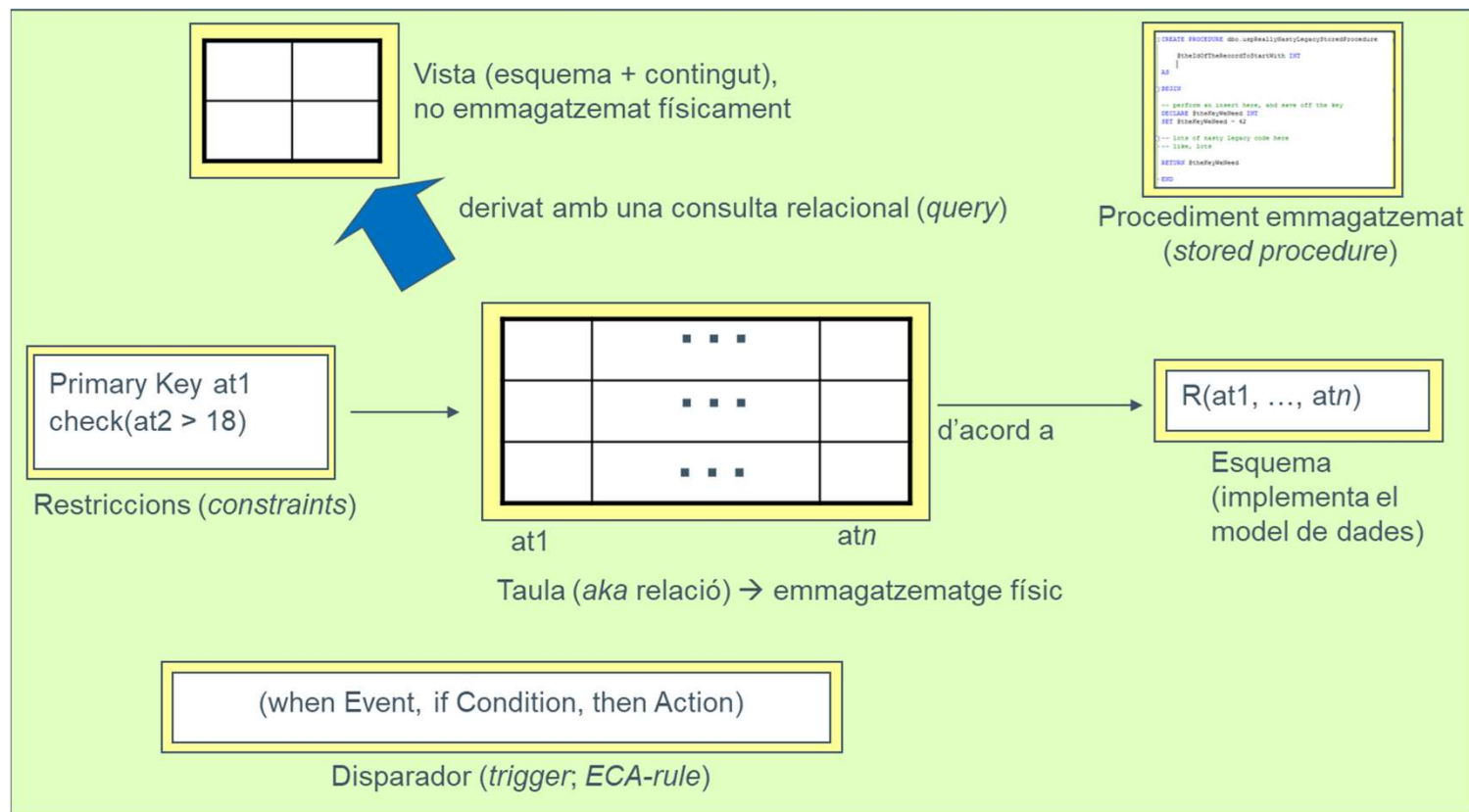
- 4.1 La tecnologia relacional
- 4.2 El procés de disseny, revisitat
- 4.3 Assignació de responsabilitats
- 4.4 Transformació del model
- 4.5 Disseny de les operacions
- 4.6 SQL

4.1 La tecnologia relacional

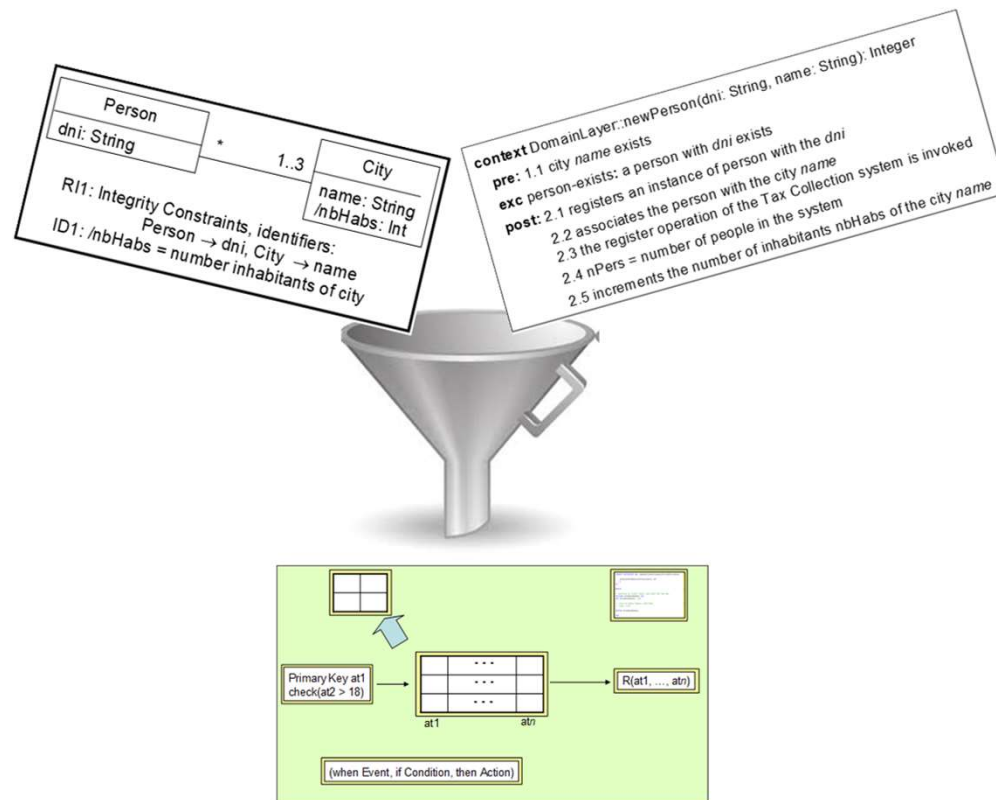
Què és una base de dades relacional?

- Col·lecció de dades que permet representar classes, els seus atributs i les seves associacions
- Aquestes dades les gestionen els **Sistemes de Gestió de Bases de Dades** (SGBD) que proporciona eines per:
 - Estructurar les dades
 - Manipular les dades (actualitzar i consultar)
 - Regles per mantenir la integritat de les dades

Vista resum



Salt conceptual: del món 0.0. al món relacional



Estructurar les dades

- Tota la informació es guarda en **taules**
 - Les taules es deriven de les classes (després de certes transformacions)
- Cada fila de la taula representa un **individu**
- Els **atributs** que representen de l'individu es guarden a les **columnes**
- Els atributs es representen fent servir tipus bàsic de dades (enters, caràcters, reals, ...)
 - Quan no tenim valor per un atribut es fa servir el valor **NULL**

Propietats d'una taula

- Els valors dels atributs són atòmics
 - Atributs multivaluats → taula adicional
 - Enum → taula adicional
- No hi ha files repetides
- No hi ha ordre entre files
- No hi ha ordre entre atributs
- Representació:
 - `nom_taula(atribut1, atribut2, atribut3, ...)`

Claus d'una taula

- **Clau primària:** atribut que identifica les files (subratllat)
 - Músic(nom,)
 - Pot ser multi-atribut
- **Clau alternativa:** atribut que pot identificar les files però no s'ha escollit com a clau primària
 - S'indica que el seu valor ha de ser únic
- **Clau forana:** subconjunt d'atributs que es fa servir per relacionar amb un altra taula (es correspon a la seva clau)
 - Permeten representar associacions i especialitzacions

Manipular les dades

- Actualització
 - Insertar fila a una taula
 - Esborrar fila d'una taula
 - Modificar atributs d'una fila
- Consulta
 - Obtenir les dades d'una taula i de totes les taules relacionades

Regles d'integritat

- **Integritat d'entitat:** els atributs que formen part de la clau primària han de tenir valors únics a tota la taula (no repetits)
- **Integritat referencial:** els valors de les claus foranes han de ser valors de la clau primària referenciada o valors NULL

Estratègies per mantenir la integritat referencial (claus forana)

- **Restricció:** No es pot esborrar una fila si es referenciada des d'un altra taula
- **Anul·lació:** S'esborra la fila i es posa NULL a totes les files que li feien referència
- **Cascada:** S'esborra la fila i totes les files que li feien referència
 - cal declarar-ho explícitament com a propietat de l'atribut

Extensions (en SQL)

- Procediments emmagatzemats (*stored procedures*)
 - permeten implementar qualsevol funcionalitat
 - en aquesta assignatura, no els farem servir
- Desencadenadors (*triggers*)
 - s'activen automàticament abans o després d'una actualització
 - en aquesta assignatura, només per a atributs derivats materialitzats
- Vistes (*views*)
 - permeten veure una projecció del contingut de la base de dades
 - no existeixen físicament
 - en aquesta assignatura, només per a atributs derivats calculats

4.2 El procés de disseny, revisitat

Context

Existeixen dos grans patrons dominants a la capa de domini i cadascun requereix patrons diferents a la capa de dades:

- *Domain Model* requereix *Data Mapper*
 - és la combinació que hem fet fins ara (capítol 3)
- *Transaction Script* requereix altres patrons
 - en aquesta assignatura, estudiarem *Row Data Gateway*
 - hi ha d'altres opcions: *Table Data Gateway*, *Active Record*, ...

El disseny de les capes de domini i de dades és radicalment diferent segons elegim una combinació de patrons o altra

- Degut a la relació amb la tecnologia relacional de la base de dades

Comparativa

Domain Model + Data Mapper *Transaction Script + Row Data Gateway*

Capa de domini

- Model de dades representa els conceptes del problema
- Diagrama de seqüència mostra interacció entre molts objectes

Capa de dades

- Esquema de la base de dades transparent
- Ofereix tres operacions de consulta per classe
- Actualitzacions automàtiques

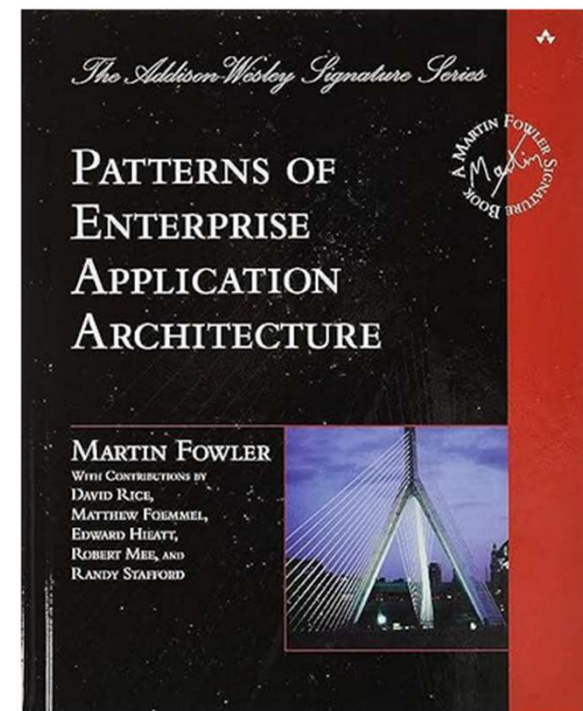
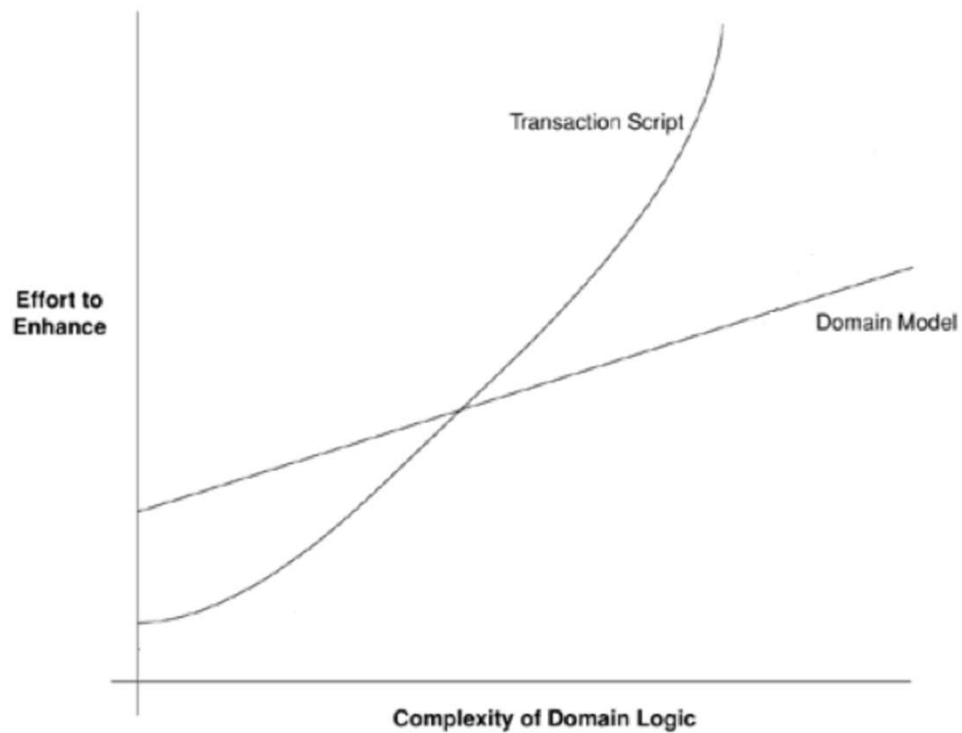
Capa de domini

- Model de dades només conté classes controladores
- Diagrama de seqüència només coreo-grafia operacions de la capa de dades

Capa de dades

- Esquema de la base de dades visible amb objectes mediadors (passarel·les)
- Les operacions de consulta es poden definir segons necessitats
- Actualitzacions explícites

Quin és millor?



Impacte en el procés de disseny

Casos d'ús	Diagrama de classes	Diagrames de seq.	Contractes	Disseny interfície
Essencials	Sense limitacions. No hi ha operacions	Un únic objecte per al sistema	De sistema. Amb possibles redundàncies. Només prec.	No existeix
1. Disseny de la interfície d'usuari				
<u>Concrets</u>	Sense limitacions. No hi ha operacions	<u>Model del comportament adaptat a la interfície</u> Un únic objecte per al sistema	De sistema. Amb possibles redundàncies. Només prec.	<u>Existeix</u>
2. Assignació de responsabilitats a capes				
Concrets	Sense limitacions. <u>Operacions de capa</u>	<u>Un objecte per capa</u>	<u>De capes. Auto-continguts. Precs. i excs.</u>	Existeix
3. Simplificació del model				
Concrets	<u>Sense elements no suportats.</u> Operacions de capa	Un objecte per capa	De capes (<u>en la capa de domini, adaptats</u>). Auto-continguts. Precs. i excs.	Existeix
4. Disseny de les operacions				
Concrets	Sense elements no suportats. <u>Operacions de classe</u>	<u>Tants objectes com calgui</u>	<u>De classe, incloent-ne operacions auxiliars.</u> Auto-continguts. Precs. i excs.	Existeix



la capa de dades rep més responsabilitats



la simplificació ha de tractar més elements



el model del comportament de la capa de domini és conceptualment més simple

4.3 Assignació de responsabilitats

Assignació de responsabilitats a la capa de dades

La tecnologia relacional ofereix alguns elements que poden rebre assignades certes responsabilitats:

Element relacional	Responsabilitat
KEY	Atribut clau
UNIQUE	Atribut que sense ser clau, no pot repetir valor
NOT NULL	Atribut que no pot prendre valor nul
CHECK	Propietats d'atributs individuals (e.g., enter positiu)
ON DELETE CASCADE	Integritat referencial en claus foranes
VIEW	Càlcul d'atributs derivats
TRIGGER	Actualització d'atributs materialitzats
STORED PROCEDURE	Responsabilitat arbitrària (<u>en aquesta assignatura</u> , no els usarem)

4.4 Transformació del model

Transformació del model 0.0. a tecnologia relacional

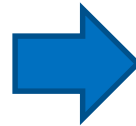
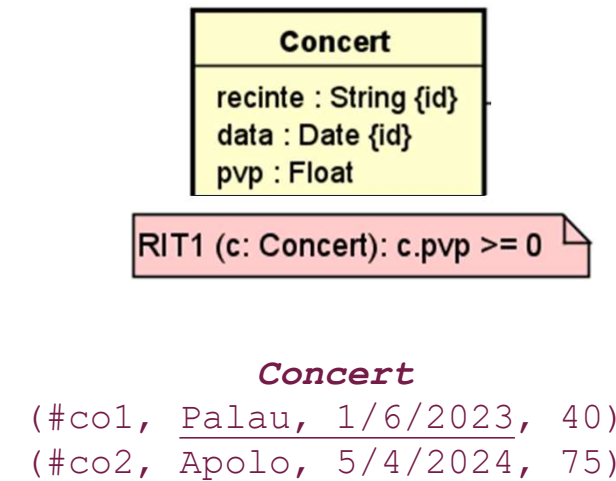
L'accés directe a la tecnologia relacional requereix la transformació d'elements addicionals del model de dades orientat a objectes:

- associacions de multiplicitat $m:n$, on $m > 1$ i $n > 1$
- jerarquies d'especialització

A banda, la resta de transformacions ja conegudes es poden definir directament sobre l'esquema de la base de dades

El resultat d'aquesta fase serà un esquema de base de dades, on també farem explícites les responsabilitats assignades a les diferents taules

Transformació del model – classes



Les claus sempre apareixen subratllades

Concert(recinte, data, pvp).on:

- pvp: CHECK(pvp >= 0) – RIT1

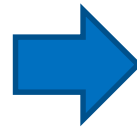
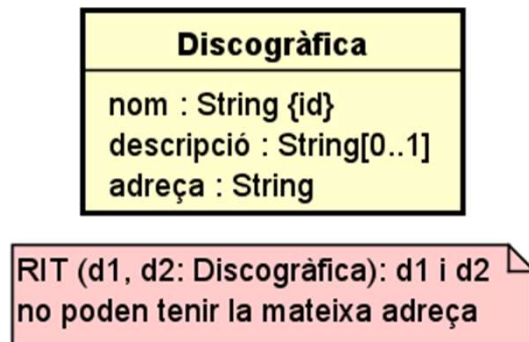
Les restriccions assignades a la capa de dades es declaren a la definició de la taula

Concert

Palau	1/6/2023	40
Apolo	5/4/2024	75

Els oids desapareixen

Transformació del model – classes amb restriccions

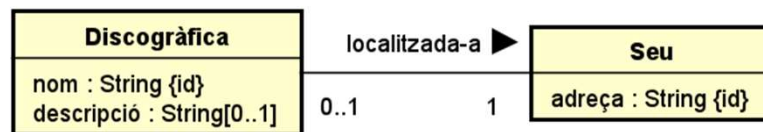


Discogràfica(nom, descripció, adreça), on:

- descripció: NULL
- adreça: UNIQUE

per motius pràctics,
declarem els que
poden ser nuls (que
són la majoria) en lloc
dels que no poden ser
nuls

Transformació del model – associacions binàries, cas un a un



Discogràfica
 (#d1, EMI, "La del gos")
 (#d2, Capitol, --)

Seu
 (#s1, Charing Cross)
 (#s2, LA)

localitzada-a
 (#d1, #s1)
 (#d2, #s2)



Discogràfica(nom, descripció, adSeu) on:

- descripció: NULL
- adSeu: REFERENCIA(Seu)

Seu(adreça)

s'elegeix una de
les taules per
lligar amb l'altra

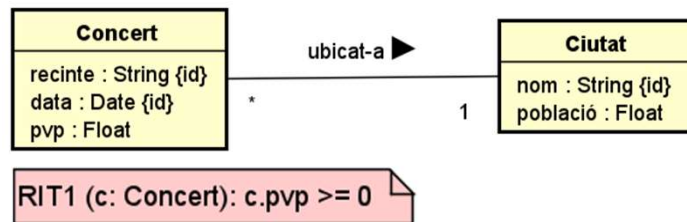
Discogràfica

EMI	La del gos	Charing Cross
Capitol	---	LA

Seu

Charing Cross
LA

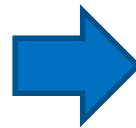
Transformació del model – associacions binàries, cas 1 a molts



Concert
 (#col, Palau, 1/6/2023, 40)
 (#co2, Apolo, 5/4/2024, 75)

Ciutat
 (#ci1, València, 1M)
 (#ci2, Barcelona, 3M)

ubicat-a
 (#col, #ci1)
 (#co2, #c12)



Concert(recinte, data, pvp, nomCiutat), on:

- nomCiutat: REFERENCIA(Ciutat)
- pvp: CHECK(pvp >= 0)

Ciutat(nom, població)

Concert

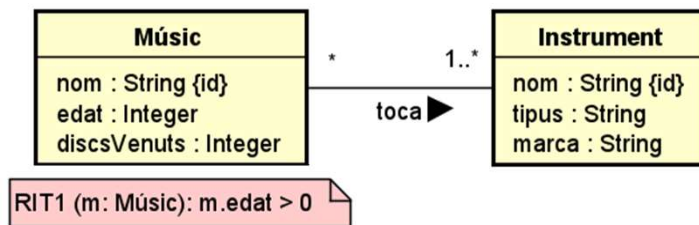
Palau	1/6/2023	40	València
Apolo	5/4/2024	75	Barcelona

Ciutat

València	1M
Barcelona	3M



Transformació del model – associacions binàries, cas molts a molts



Música

(#m1, Lennon, 40, 22M)
 (#m2, Clapton, 75, 9M)

Instrument

(#i1, guitarra, corda, Fender)
 (#i2, piano, percussió, Petrof)

toca

(#m1, #i1)
 (#m1, #i2)
 (#m2, #i1)

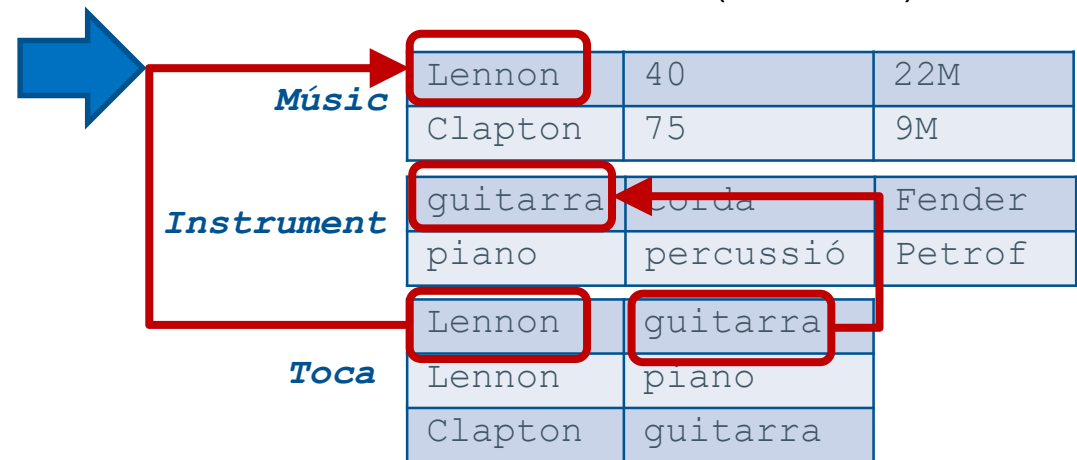
Música(nom, edat, discsVenuts), on:

- edat: CHECK(edat >= 0)

Instrument(nom, tipus, marca)

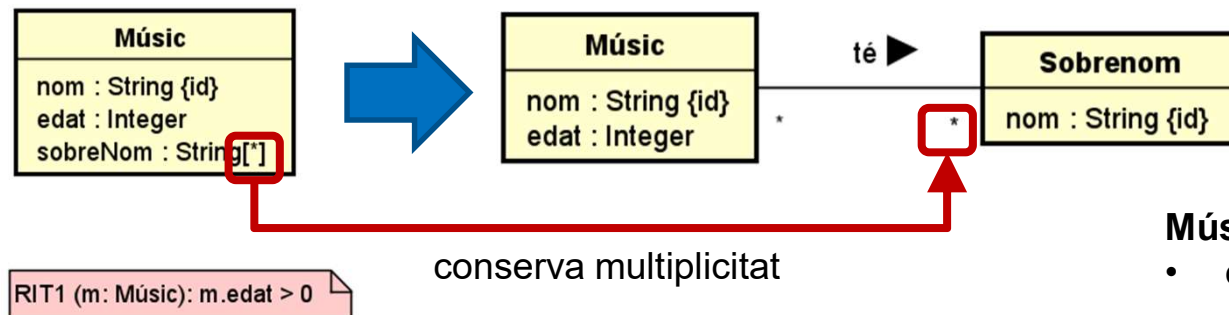
Toca(nomM, nomI), on:

- nomM: REFERENCIA(Música)
- nomI: REFERENCIA(Instrument)



Transformació del model – cas d'atributs multi-valuats

Cal fer una transformació prèvia del model de dades introduïnt una nova classe i associació, i aleshores ens trobem al cas anterior (associació binària molts a molts)



conserva multiplicitat

Músic(nom, edat), on:

- edat: CHECK(edat >= 0)

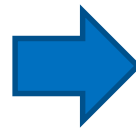
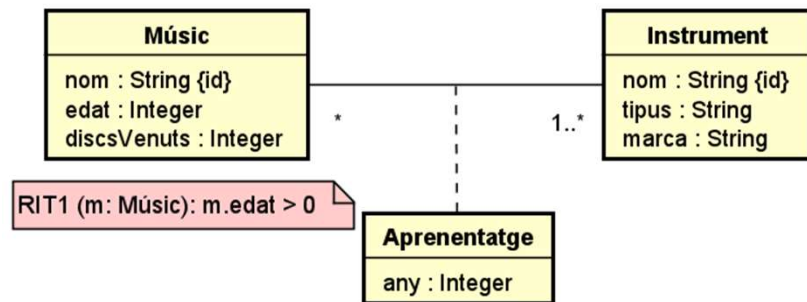
Sobrenom(nom)

Té(nomM, sobreM), on:

- nomM: REFERENCIA(Músic)
- sobreM: REFERENCIA(Sobrenom)

Transformació del model – classes associatives

Bàsicament el mateix que el cas anterior



Música(nom, edat, discsVenuts), on:

- edat: CHECK(edat>0)

Instrument(nom, tipus, marca)

Aprenentatge(nomM, nomI, any), on:

- nomM: REFERENCIA(Música)
- nomI: REFERENCIA(Instrument)

Música
 (#m1, Lennon, ...)
 (#m2, Clapton, ...)

Instrument
 (#i1, guitarra, ...)
 (#i2, piano, ...)

Aprenentatge
 (#a1, #m1, #i1, 1960)
 (#a2, #m1, #i2, 1959)
 (#a3, #m2, #i1, 1965)

Música

Lennon	40	22M
Clapton	75	9M

Instrument

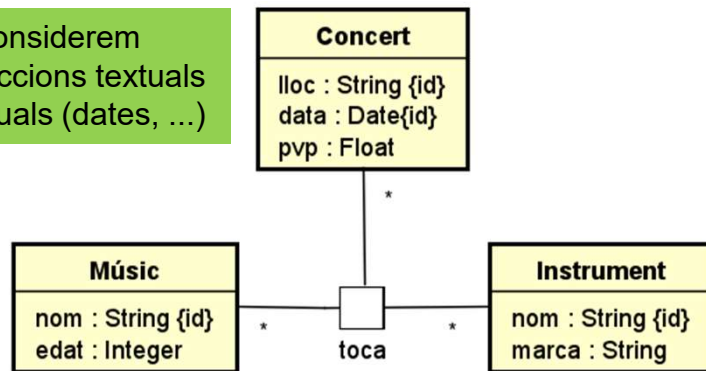
guitarra	corda	Fender
piano	percussió	Petrof

Aprenentatge

Lennon	guitarra	1960
Lennon	piano	1959
Clapton	guitarra	1965

Transformació del model – associacions ternàries, cas tots molts

No considerem restriccions textuais habituals (dates, ...)



Concert

(#c1, Porta Ferrada, 5/7/2009, 80)

Instrument

(#i1, Guitarra, Fender)

Música

(#m1, Lennon, 40)

tocat-a

(#c01, #gn1, #c1)

Concert(lloc, data, pvp)

Instrument(nom, marca)

Música(nom, edat)

Toca(llocC, dataC, nomI, nomM), on:

- llocC+dataC: REFERENCIA(Concert)
- nomI: REFERENCIA(Instrument)
- nomM: REFERENCIA(Música)

Concert

Porta Ferrada	5/7/2009	80
---------------	----------	----

Instrument

Guitarra	Fender
----------	--------

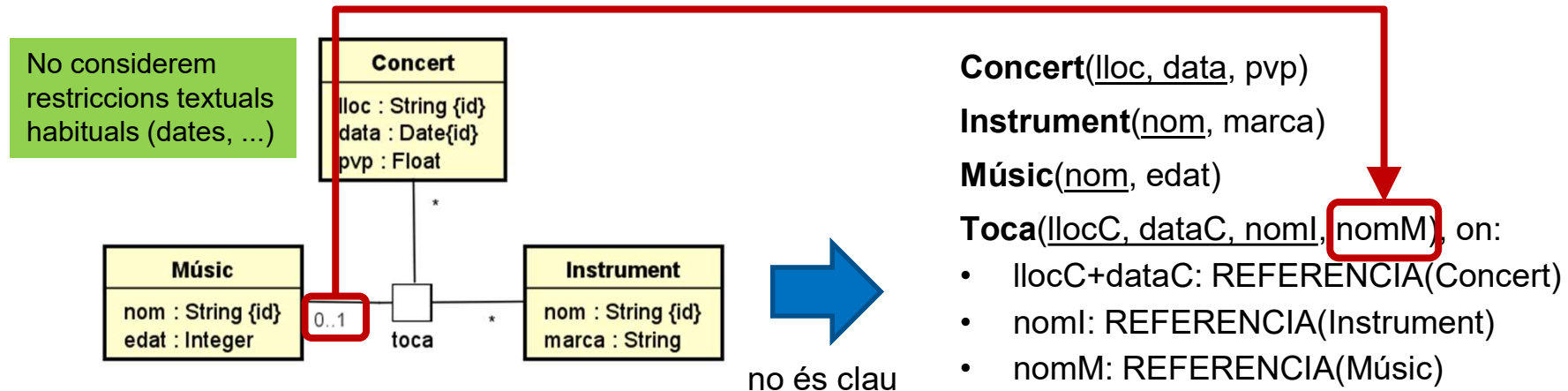
Música

Lennon	40
--------	----

Toca

Porta Ferrada	5/7/2009	Guitarra	Lennon
---------------	----------	----------	--------

Transformació del model – associacions ternàries, cas un 0..1



Concert

(#co1, Porta Ferrada, 5/7/2009, 40)
 (#co2, Apolo, 5/4/2024, 75)

Grup Musical

(#gn1, The Pretenders, 1978, Pop)

Cançó

(#c1, Brass in Pocket, 189, Pop)

tocat-a

(#co1, #gn1, #c1)

Concert

Porta Ferrada	5/7/2009	40
Apolo	5/4/2024	75

Grup Musical

The Pretenders	1978	Pop
----------------	------	-----

Cançó

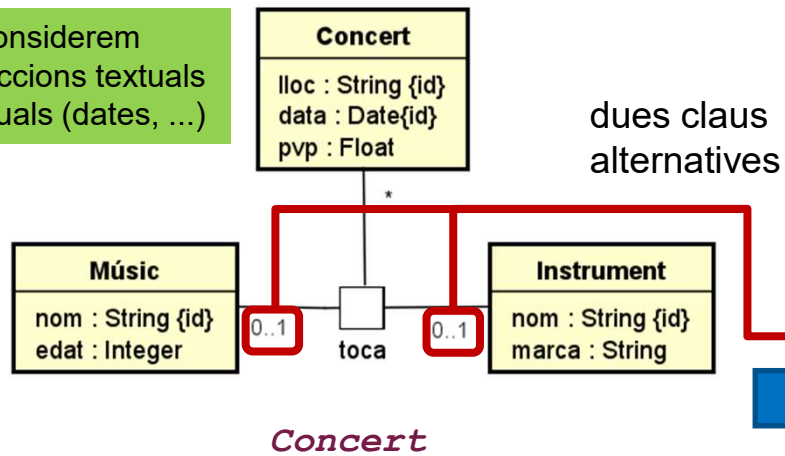
Brass in Pocket	189	Pop
-----------------	-----	-----

Tocat-a

Porta Ferrada	5/7/2009	The Pretenders	Brass in Pocket
---------------	----------	----------------	-----------------

Transformació del model – associacions ternàries, cas dos 0..1

No considerem restriccions textuais habituals (dates, ...)



dues claus alternatives

Concert(lloc, data, pvp)

Instrument(nom, marca)

Música(nom, edat)

Toca(llocC, dataC, nomI, nomM), on:

- llocC+dataC: REFERENCIA(Concert)

- nomI: REFERENCIA(Instrument)

- llocC+dataC+nomM: UNIQUE

- nomM: REFERENCIA(Música)

podria ser al revés

Concert
(#c1, Porta Ferrada, 5/7/2009, 80)

Instrument
(#i1, Guitarra, Fender)

Música
(#m1, Lennon, 40)

tocat-a
(#c01, #gn1, #c1)

Concert

Porta Ferrada	5/7/2009	80
---------------	----------	----

Instrument

Guitarra	Fender
----------	--------

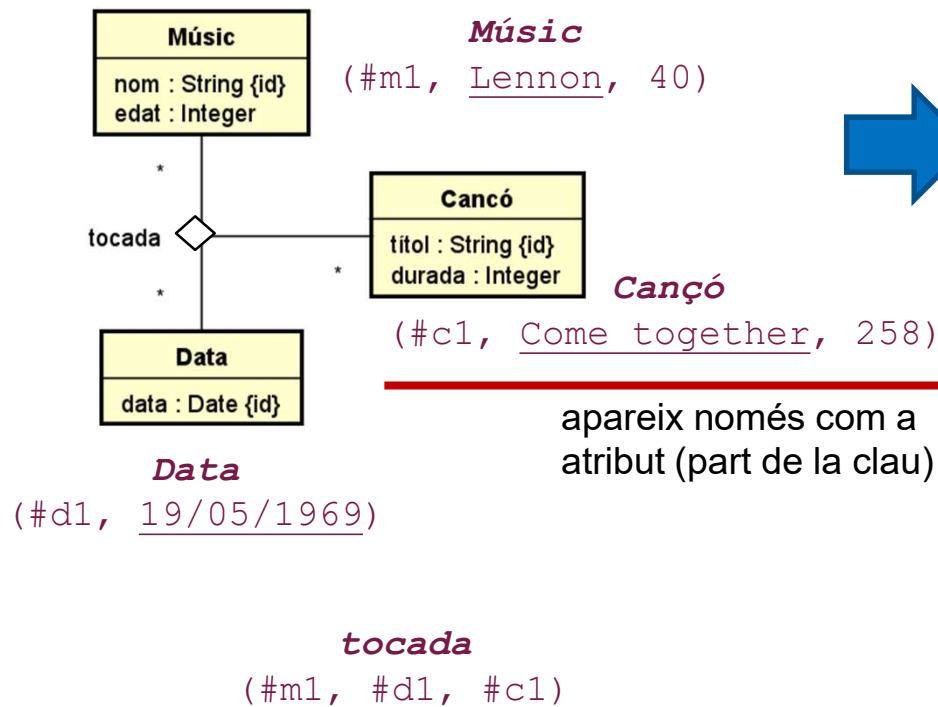
Música

Lennon	40
--------	----

Toca

Porta Ferrada	5/7/2009	Guitarra	Lennon
---------------	----------	----------	--------

Transformació del model – associacions ternàries amb data



Músic(nom, edat)
Cançó(títol, durada)
Tocada(nomM, títolC, dataT) on:

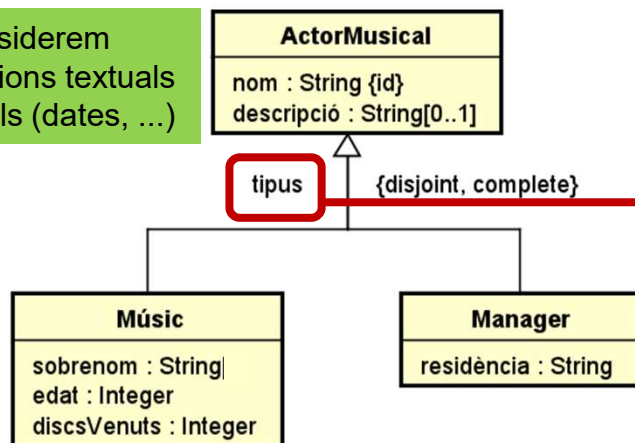
- nomM: REFERENCIA(Músic)
- títolC: REFERENCIA(Cançó)

Músic	Lennon	40	22M
Cançó	Come together	258	
Tocada	Lennon	Come tog.	19/05/69

Transformació del model – especialitzacions

Apliquem una estratègia anomenada *Class Table Inheritance* (n'hi ha d'altres)

No considerem restriccions textuais habituals (dates, ...)



Música

(#m1, Lennon, Escarbat, -, 40, 22M)
 (#m2, Clapton, -, Slow Hand, 75, 9M)

Manager

(#s1, Epstein, 5th Beatle, London)

ActorMusical(nom, descripció, tipus)

Música(nom, sobrenom, edat, discsVenuts)

- nom: REFERENCIA(ActorMusical)

Aprenentatge(nom, residència)

- nom: REFERENCIA(ActorMusical)

si fos *incomplete*, podria prendre el valor *actorMusical*

ActorMusical

Lennon	Escarbat	música
Clapton	-	música
Epstein	5th Beatle	manager

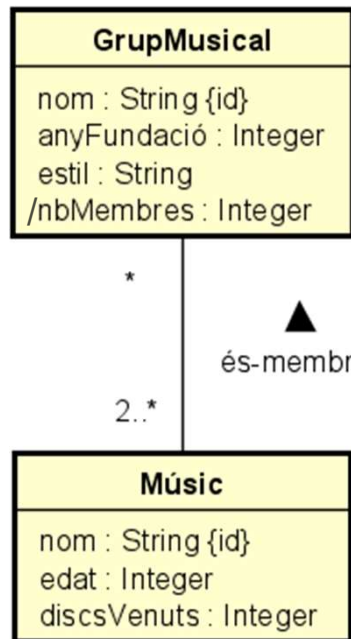
Música

Lennon	-	40	22M
Clapton	Slow Hand	75	9M

Manager

Epstein	London
---------	--------

Transformació del model – atributs derivats



materialitzat

calculat

capa domini

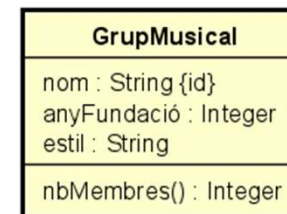
capa dades

capa domini

capa dades

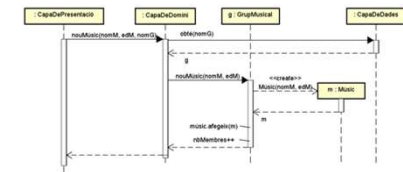


GrupMusical(nom, anyF, estil, nbMembres)



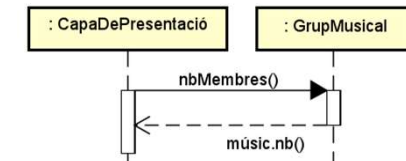
GrupMusical(nom, anyF, estil)

NbMembres(grup, nbmembers)



```

CREATE TRIGGER nbMembers AFTER INSERT ON Music
REFERENCING new AS nou_music
FOR EACH ROW
    if nou_music.grup IS NOT NULL
        UPDATE GrupMusical
        SET nbMembers=nbMembers+1
        WHERE nom = nou_music.grup
    )
    
```



```

SELECT Music.grup,
COUNT(*) AS nbmembers
FROM Music
GROUP BY Music.grup;
    
```

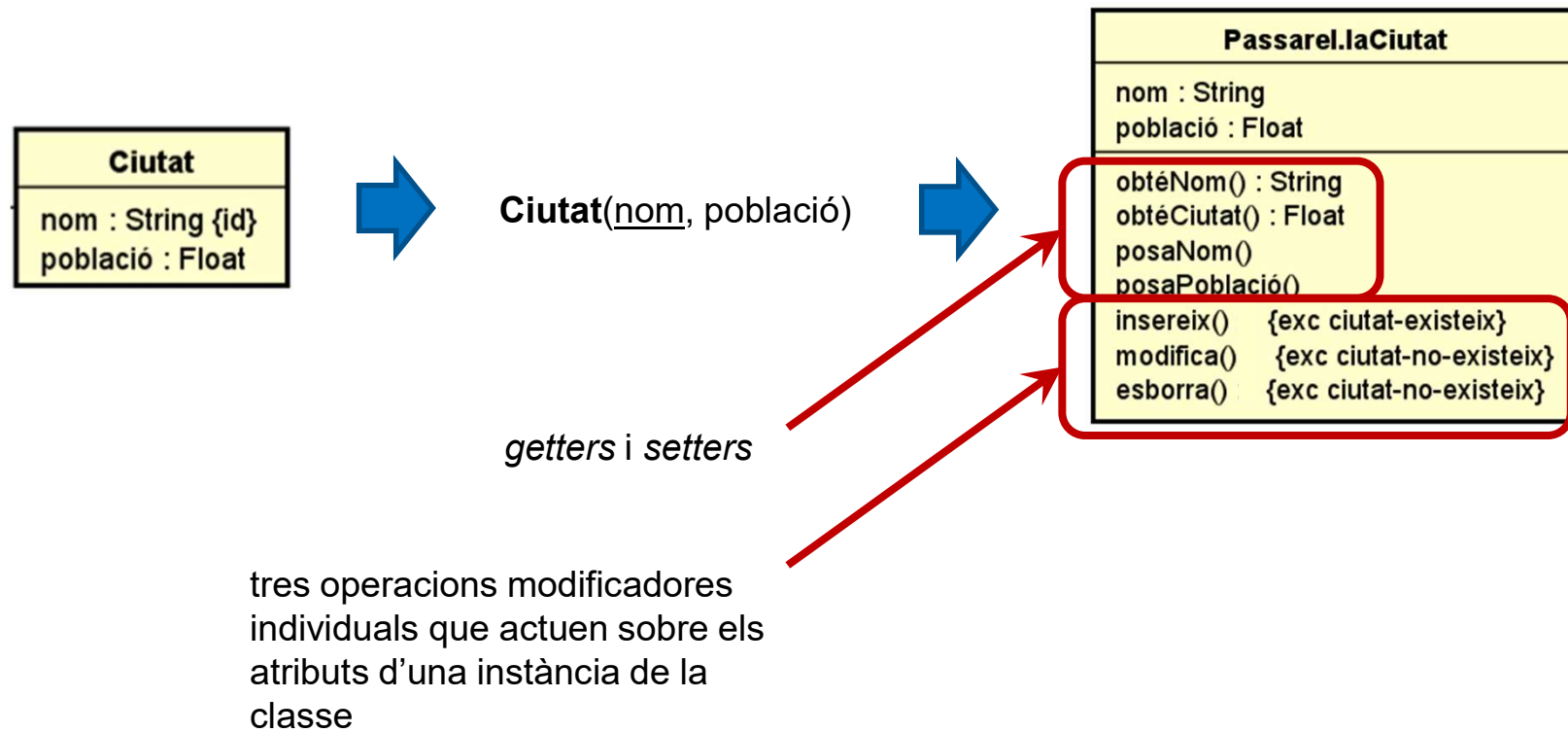
4.5 Disseny de les operacions

Patró Passarel.la Fila (*Row Data Gateway*)

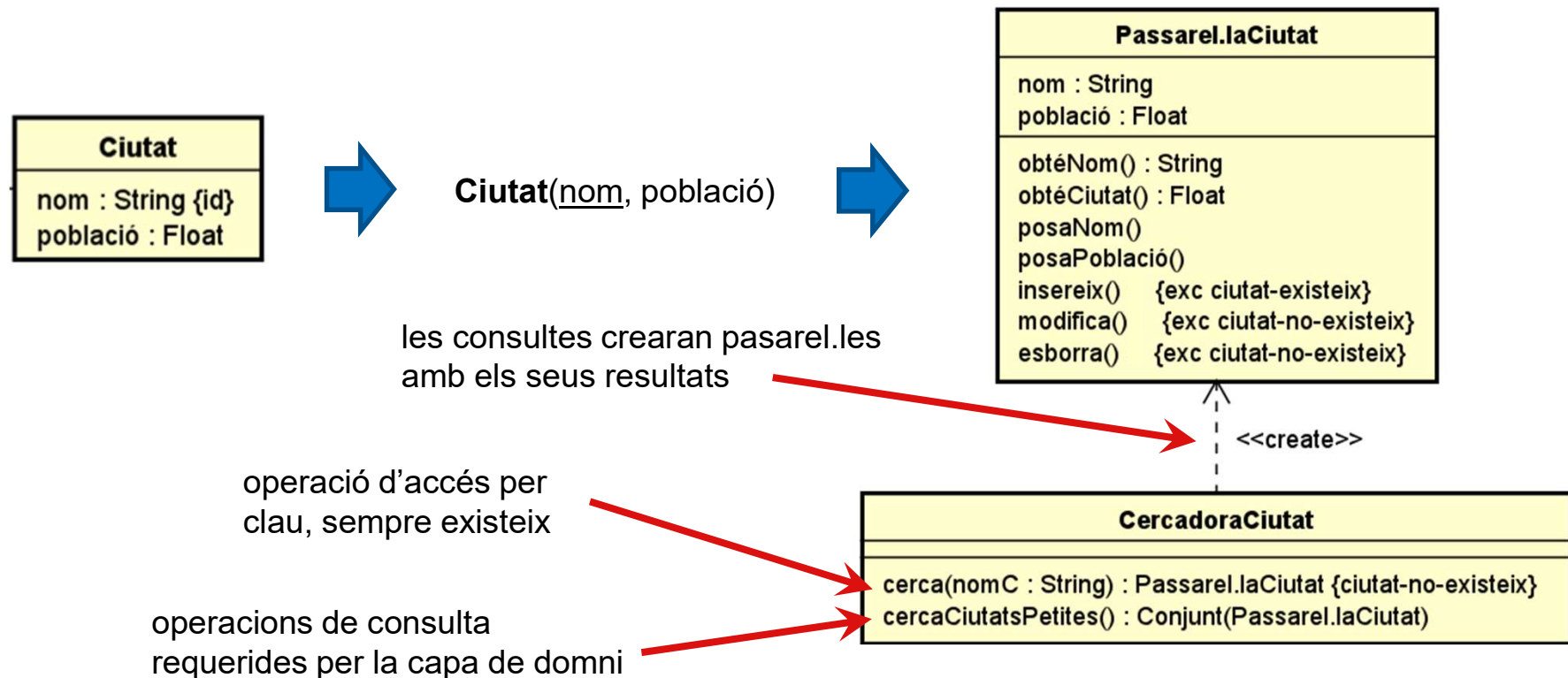
Permet definir controladors de tipus transacció a la capa de dades que regulen l'accés a les dades des de la capa de domini:

- Dues classes para cada taula de la capa de dades
 - Una classe base amb operacions modificadores individuals
 - Una instància (*passarel.la*) correspon a una fila de la taula
 - Una classe cercadora per obtenir passarel.las
- Les operacions de les classes passarel.la s'implementen directament sobre la base de dades
 - Poden aprofitar la capacitat expressiva del llenguatge SQL

Patró Passarel·la Fila – classe base



Patró Pasarel.la Fila – classe cercadora

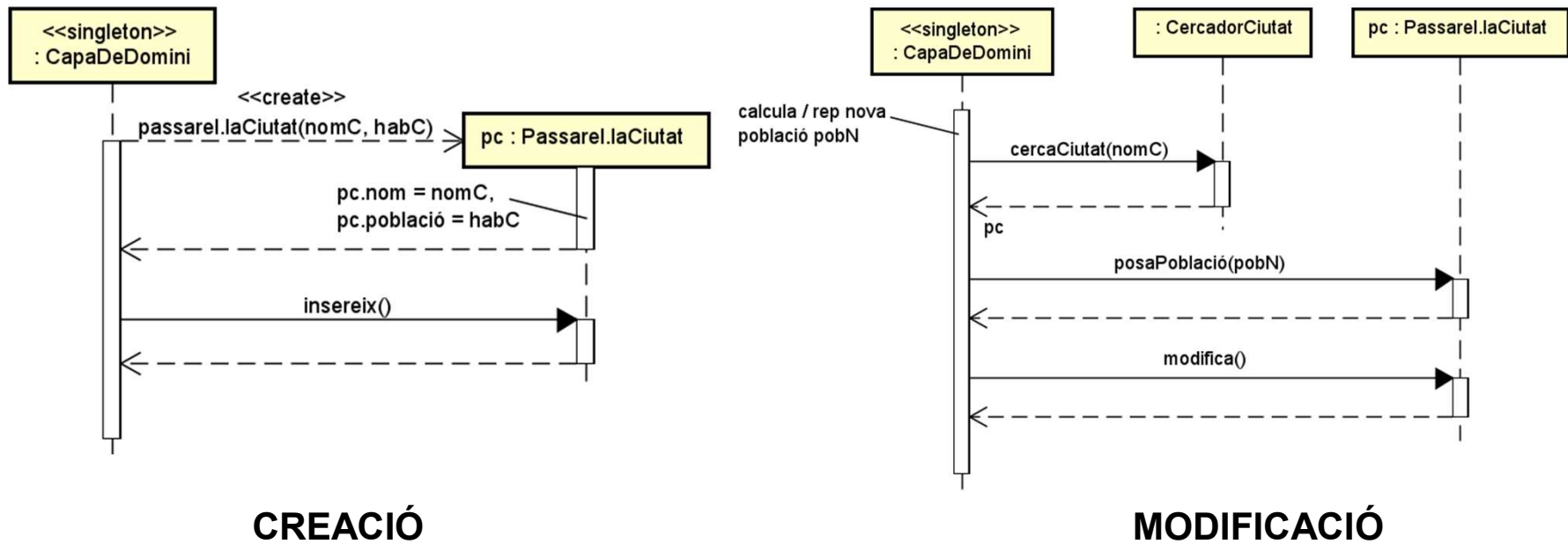


Patró Passarel.la Fila – utilització

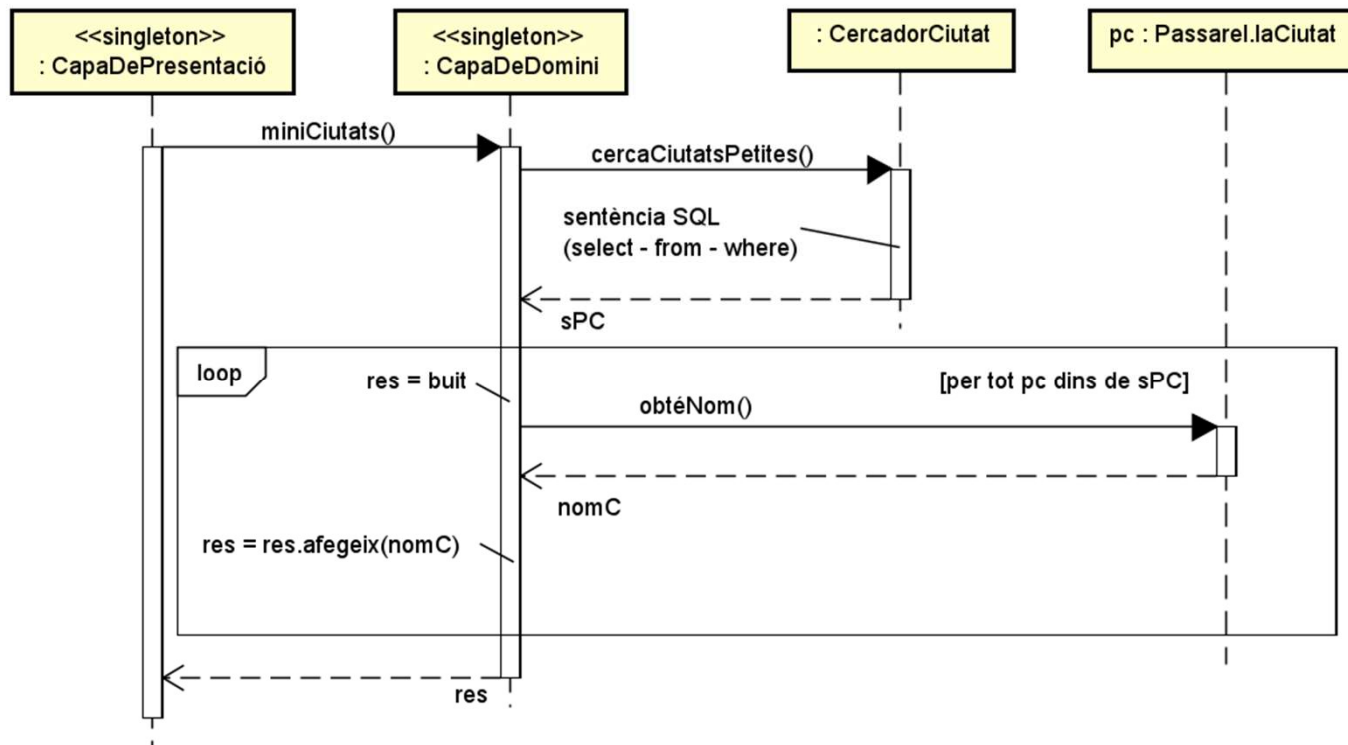
Els controladors de la capa de domini interaccionen directament amb les classes corresponents a les passarel.las:

- creen passarel.las per inserir objectes
- es comuniquen amb els cercadors per obtenir objectes (en forma de passarel.las)
- manipulen els objectes a la mateixa capa de domini
- retornen els objectes a la capa de dades usant les passarel.las

Patró Passarel.la Fila – actualitzacions



Patró Passarel·la Fila – ús en consultes



4.6 SQL

Structured Query Language (SQL)

- Llenguatge estructurat de definició, actualització i consulta de bases de dades
- Proposat per un departament d'investigació d'IBM (SEQUEL, Donald D. Chamberlin and Raymond F. Boyce, 1970s)
- Adoptat com a estàndard per al Model Relacional de bases de dades en els anys 1986 (ANSI) i 1987 (ISO)

Utilització del SQL

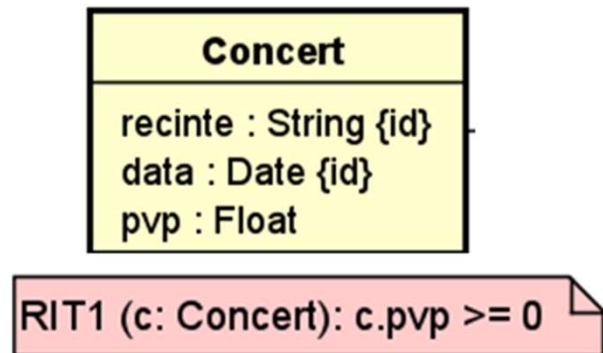
- Des d'un programa (C++, ...)
 - Fent servir alguna llibreria que permeti executar comandes SQL
 - Per implementar (part de) la capa de dades
- Des d'un programa client del SGBD
 - Com a mínim, per a crear i destruir taules
 - També per a jugar o depurar el programa

En tots dos casos, la sintaxi és la mateixa

(Per a més detalls, v. document de la pràctica)

Creació de taules

- Mitjançant la sentència **CREATE TABLE**



```
CREATE TABLE Concert (  
    recinte text NOT NULL,  
    data date NOT NULL,  
    pvp double precision NOT NULL,  
    PRIMARY KEY (recinte, data),  
    CHECK (pvp >= 0)  
)
```

Concert(recinte, data, pvp), on:

- pvp: CHECK(pvp >= 0) – RIT1

A tenir en compte

- Les claus primàries s'han de declarar explícitament amb l'etiqueta **PRIMARY KEY**
 - Si la clau primària és un atribut, al costat de l'atribut
 - Si és multi-atribut, al final de la taula
- Per claredat, els **CHECK** també els declarem al final de la taula
- La resta d'etiquetes, al costat de l'atribut
 - Recordeu que cal declarar explícitament els atributs que no poden prendre valor nul (per defecte, el poden prendre)
 - Excepció: claus primàries (que no poden ser nul·les)
- També cal usar els tipus predefinits de SQL
 - **text, integer, date, double precision, ...**

Exemples

Concert
recinte : String {id}
data : Date {id}
pvp : Float

RIT1 (c: Concert): c.pvp >= 0

```
CREATE TABLE Concert (  
  recinte text NOT NULL,  
  data date NOT NULL,  
  pvp double precision NOT NULL,  
  PRIMARY KEY (recinte, data),  
  CHECK (pvp >= 0)  
)
```

Discogràfica
nom : String {id}
descripció : String[0..1]
adreça : String

RIT (d1, d2: Discogràfica): d1 i d2
no poden tenir la mateixa adreça

```
CREATE TABLE Discogràfica(  
  nom text PRIMARY KEY,  
  descripció text,  
  adreça text NOT NULL UNIQUE  
)
```


Exemples



RIT1 (c: Concert): c.pvp >= 0

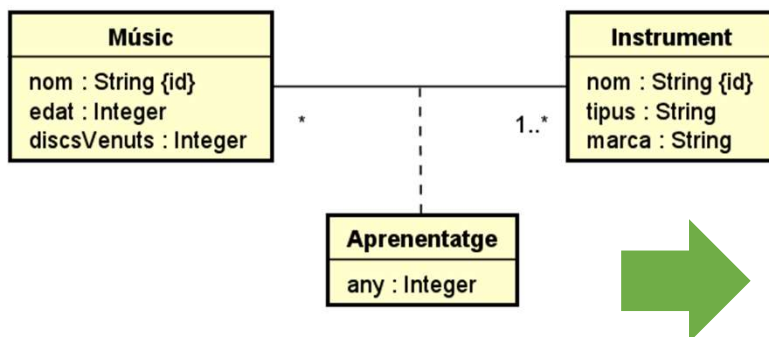
Concert(recinte, data, pvp, nomCiutat), on:

- nomCiutat: REFERENCIA(Ciutat)
- pvp: CHECK(pvp >= 0)

Ciutat(nom, població)

```
CREATE TABLE Concert(  
    recinte text NOT NULL,  
    data date NOT NULL,  
    pvp double NOT NULL,  
    nomCiutat text NOT NULL,  
    PRIMARY KEY (recinte, data),  
    FOREIGN KEY (nomCiutat),  
        REFERENCES Ciutat(nom),  
    CHECK (pvp >= 0)  
)  
  
CREATE TABLE Ciutat(  
    nom text PRIMARY KEY,  
    població double NOT NULL  
)
```

Exemples – esborrat en cascada



Música(nom, edat, discsVenuts):

Instrument(nom, tipus, marca)

Aprenentatge(nomM, nomI, any), on:

- nomM: REFERENCIA(Música) ON DELETE CASCADE
- nomI: REFERENCIA(Instrument) ON DELETE CASCADE

```
CREATE TABLE Música (
    nom text PRIMARY KEY,
    edat integer NOT NULL
)
```

```
CREATE TABLE Instrument (
    nom text PRIMARY KEY,
    tipus text NOT NULL,
    marca text NOT NULL
)
```

```
CREATE TABLE Aprenentatge (
    nomM text NOT NULL,
    nomI text NOT NULL,
    any integer NOT NULL,
```

```
PRIMARY KEY (nomM, nomI),
```

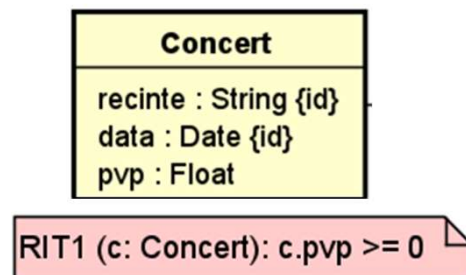
```
FOREIGN KEY (nomM) REFERENCES Música (nom)
```

```
FOREIGN KEY (nomI) REFERENCES Instrument (nom)
```

Si s'esborra un *Música* o un *Instrument*, s'esborra la fila de la taula *Aprenentatge*

ON DELETE CASCADE,
(nom) ON DELETE CASCADE

Poden haver-hi diferències entre SGBDs



Concert(recinte, data, pvp), on:

- pvp: CHECK(pvp >= 0) – RIT1

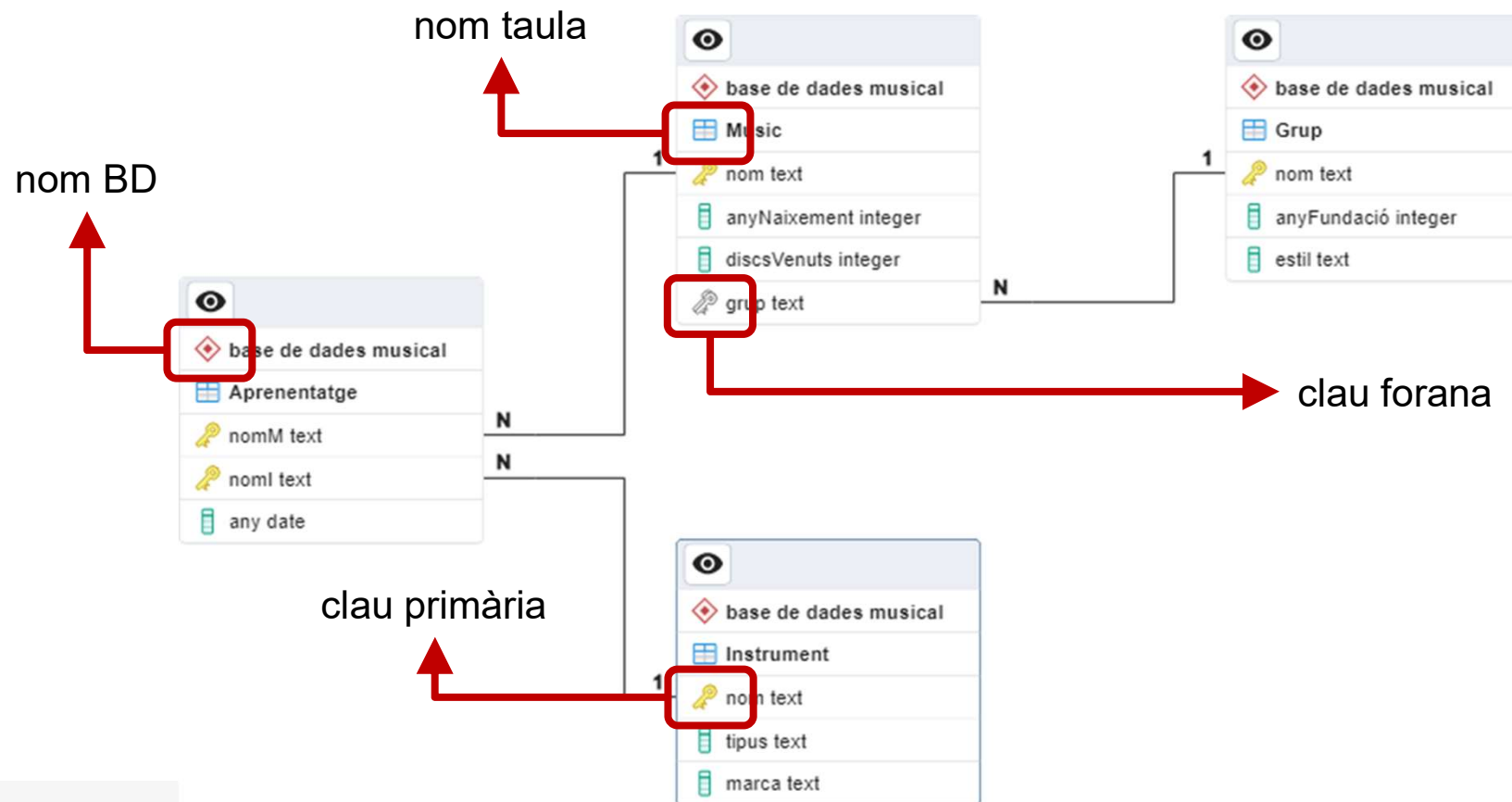
```
CREATE TABLE Concert (  
  recinte text NOT NULL,  
  data date NOT NULL,  
  pvp double precision NOT NULL,  
  PRIMARY KEY (recinte, data),  
  CHECK (pvp >= 0)  
)
```

Estàndard SQL

```
CREATE TABLE Concert (  
  recinte text NOT NULL,  
  data date NOT NULL,  
  pvp double precision NOT NULL,  
  CONSTRAINT concert_pkey PRIMARY KEY (recinte, data),  
  CONSTRAINT pvp CHECK (pvp >= 0)  
)
```

postgresql

Representació gràfica de l'esquema (diagrama ER)



Supressió de taules

- Mitjançant la sentència **DROP TABLE**
 - Normalment invocada des del client del SGBD

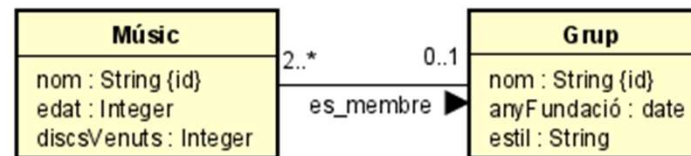
Música
nom : String {id}
edat : Integer
discsVenuts : Integer



DROP TABLE Música;

Inserció de files (i.e., creació)

1 fila



- Indicant el nom de les columnes (poden no estar totes)

```
INSERT INTO Músic (nom, discsVenuts, grup)
VALUES ("John Lennon", 150 "The Beatles");
```

- Sense indicar-los (cal donar tots els valors)

```
INSERT INTO Músic
VALUES ("John Lennon", 45 ,150, "The Beatles");
```

Supressió de files

N files

```
DELETE FROM Música WHERE nom = "John Lennon";
```

→ Esborra un músic en concret (PRIMARY KEY)

```
DELETE FROM Música WHERE grup = "The Beatles";
```

→ Esborra tots els músics d'un grup

```
DELETE FROM Música;
```

→ Esborra totes les files

Modificació de files

N files

```
UPDATE Música  
  SET discsVenuts = 183  
  WHERE grup = "The Beatles";
```

→ Modifica el valor de la columna `disc_venuts` de totes les files que corresponen a membres del grup "The Beatles"

Consultes sobre una taula

N files

```
SELECT * FROM Música;
```

→ Retorna totes les columnes de tots els músics (sense **WHERE**)

```
SELECT * FROM Música WHERE grup = "The Beatles";
```

→ Retorna totes les columnes de tots els músics del grup "The Beatles"

```
SELECT * FROM Música WHERE discs_venuts > 100;
```

→ Retorna totes les columnes de tots els músics que tenem més de 100 discs venuts

```
SELECT nom, edat FROM Música;
```

→ Retorna les columnes nom i edat de tots els músics

Operadors per a les condicions (where/check)

- aritmètics: *, +, -, /
- de comparació: =, <, >, <=, >=, <>
- lògics: NOT, AND, OR
- altres:
 - <columna> BETWEEN <límit 1> AND <límit 2>
 - <columna> IN (<valor1>, <valor2> [....., <valorN>])
 - <columna> IS [NOT] NULL

Ordenació en les cerques

```
SELECT nom, discsVenuts FROM Música  
    ORDER BY nom ASC;
```

→ Retorna el nom i els discs venuts de tots els músics ordenats ascendentment per la columna nom

```
SELECT nom, discsVenuts FROM Músics  
    ORDER BY discsVenuts DESC;
```

→ Retorna el nom i els discs venuts de tots els músics ordenats descendentment per la columna discVenuts

Funcions d'agregació

```
SELECT COUNT(*) FROM Música WHERE grup = "The Beatles";
```

→ Retorna el nombre de membres del grup anomenat "The Beatles"

```
SELECT COUNT(grup) FROM Música;
```

→ Retorna el nombre de músics que estan en un grup (no tots els músics estan a un grup)

```
SELECT COUNT(DISTINCT grup) FROM Música;
```

→ Retorna el nombre de grups (conta la columna grup sense repetits) dels músics que estan en un grup (no tots els músics estan a un grup)

Funcions d'agregació aritmètiques

```
SELECT SUM(discsVenuts) FROM Música;
```

→ Retorna el nombre total de discs venuts de tots els músics

```
SELECT AVG(edat) FROM Música WHERE grup = "The Beatles";
```

→ Retorna l'edat mitjana dels músics del grup "The Beatles"

```
SELECT MIN(edat), MAX(edat) FROM Música;
```

→ Retorna l'edat mínima i la màxima entre tots els músics

Funcions d'agrupació de files

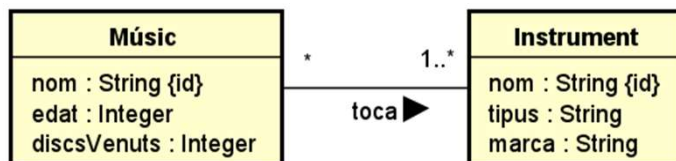
```
SELECT grup, COUNT(*) AS nbMembers  
  FROM Música  
  GROUP BY grup;
```

→ Retorna una fila per cada grup amb el nom del grup i el número de membres del grup (**COUNT**) a l'atribut `nbMembers`

```
SELECT grup, SUM(discsVenuts) AS totalDiscs  
  FROM Música  
  GROUP BY grup;
```

→ Retorna una fila per cada grup amb el nom del grup i la suma dels discos de tots els membris del grup (**SUM**) a l'atribut `totalDiscs`

Consultes files combinades (JOIN)



Músic

Lennon	40	22M
Clapton	75	9M

Toca

Lennon	guitarra
Lennon	piano
Clapton	guitarra

Instrument

guitarra	corda	Fender
piano	percussió	Petrof

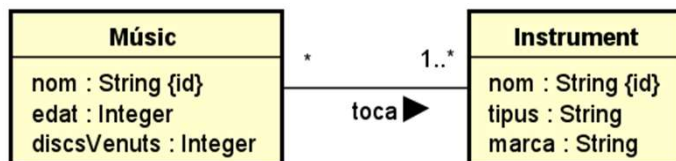
```

SELECT *
FROM Músics m, Toca t, Instrument i
WHERE t.nomM = m.nom AND t.nomI = i.nom;
    
```

Millor escollir els atributs al
SELECT fent servir AS per
decidir el nom si es necessari

nom	edat	discsVenuts	nomM	nomI	nom	Tipus	marca
Lennon	40	22M	Lennon	guitarra	guitarra	corda	Fender
Lennon	40	22M	Lennon	Piano	Piano	Percussió	Petrof
Clapton	75	9M	Clapton	Guitarra	Guitarra	Corda	Fender

Consultes files combinades (JOIN)



Músic

Lennon	40	22M
Clapton	75	9M

Toca

Lennon	guitarra
Lennon	piano
Clapton	guitarra

Instrument

guitarra	corda	Fender
piano	percussió	Petrof

```

SELECT t.nomM, m.edat, m.discsVenuts, t.nomI, i.tipus, i.marca
FROM Músics m, Toca t, Instrument i
WHERE t.nomM = m.nom AND t.nomI = i.nom;
    
```

nomM	edat	discsVenuts	nomI	tipus	marca
Lennon	40	22M	guitarra	corda	Fender
Lennon	40	22M	Piano	percussió	Petrof
Clapton	75	9M	Guitarra	Corda	Fender

Referències

Bibliografia

- H. García-Molina, J.D. Ullman, J. Widom. Database Systems: The Complete Book. 2nd ed. Pearson Education Limited, 2013
- M. Fowler, D. Rice, M. Foemmel, E. Heatt, R. Mee, R. Stafford: Patterns of Enterprise Application Architecture. Addison-Wesley, 2003

Recursos

- <https://www.w3schools.com/sql/default.asp>