

Disseny i Administració de Bases de Dades

Tema 2. Modelat conceptual

José L. Balcázar, Jordi Esteve

Departament de Ciències de la Computació (CS). EPSEVG. UPC

Primavera 2024

Introducció al modelat conceptual

A què ens referim per models conceptuais?

Models: representacions **simbòliques** de la realitat.

Conceptuals: basats en conceptes; esforç d'**abstracció**.

Introducció al modelat conceptual

A què ens referim per models conceptuais?

Models: representacions **simbòliques** de la realitat.

Conceptuals: basats en conceptes; esforç d'**abstracció**.

Dificultats generals:

- ▶ No caure en la incoherència: la solució no és única.

Introducció al modelat conceptual

A què ens referim per models conceptuais?

Models: representacions **simbòliques** de la realitat.

Conceptuals: basats en conceptes; esforç d'**abstracció**.

Dificultats generals:

- ▶ No caure en la incoherència: la solució no és única.
- ▶ Discernir els fets rellevants en el nostre context.

Introducció al modelat conceptual

A què ens referim per models conceptuals?

Models: representacions **simbòliques** de la realitat.

Conceptuals: basats en conceptes; esforç d'**abstracció**.

Dificultats generals:

- ▶ No caure en la incoherència: la solució no és única.
- ▶ Discernir els fets rellevants en el nostre context.
- ▶ Encertar el grau adequat de fidelitat de la representació:
 - ▶ un nivell de detall insuficient pot ser dolent,
 - ▶ un nivell de detall excessiu també.

Introducció al modelat conceptual

A què ens referim per models conceptuais?

Models: representacions **simbòliques** de la realitat.

Conceptuals: basats en conceptes; esforç d'**abstracció**.

Dificultats generals:

- ▶ No caure en la incoherència: la solució no és única.
- ▶ Discernir els fets rellevants en el nostre context.
- ▶ Encertar el grau adequat de fidelitat de la representació:
 - ▶ un nivell de detall insuficient pot ser dolent,
 - ▶ un nivell de detall excessiu també.
- ▶ El nivell adequat de detall varia amb l'objectiu del model:
 - ▶ comunicació amb usuaris o responsables,
 - ▶ comunicació entre desenvolupadors,
 - ▶ guia del procés de desenvolupament...

Motivació del modelat conceptual

Per què construir un model?

Abans de començar a modelar, és **imprescindible** saber per què estem modelant. Convé preguntar-se:

- ▶ Quines dades de la realitat anem a processar?
- ▶ Qui les vol processar? Per què?
- ▶ Quins graus d'eficàcia i detall volem?

Motivació del modelat conceptual

Per què construir un model?

Abans de començar a modelar, és **imprescindible** saber per què estem modelant. Convé preguntar-se:

- ▶ Quines dades de la realitat anem a processar?
- ▶ Qui les vol processar? Per què?
- ▶ Quins graus d'eficàcia i detall volem?

És crucial:

- ▶ que el model suposi un **progrès** en la nostra comprensió de la realitat a modelar;
- ▶ que, per aconseguir-ho, el model aportí **simplicitat**.

Modelat conceptual: Enfocs

Varis enfocs disponibles

Opcions de modelat:

- ▶ Amb **esquemes relacionals**:
 - ▶ Els que, gràcies a SQL, podem traduir directament per crear la base de dades (relacions, noms i tipus dels atributs, etc.);

Modelat conceptual: Enfocs

Varis enfocs disponibles

Opcions de modelat:

- ▶ Amb **esquemes relacionals**:
 - ▶ Els que, gràcies a SQL, podem traduir directament per crear la base de dades (relacions, noms i tipus dels atributs, etc.);
- ▶ Amb **diagrames E/R** (Entitat/Relació), o bé

Modelat conceptual: Enfocs

Varis enfocs disponibles

Opcions de modelat:

- ▶ Amb **esquemes relacionals**:
 - ▶ Els que, gràcies a SQL, podem traduir directament per crear la base de dades (relacions, noms i tipus dels atributs, etc.);
- ▶ Amb **diagrames E/R** (Entitat/Relació), o bé
- ▶ Amb **diagrames de classes** de UML:
 - ▶ si bé estan pensats per l'especificació de classes en un entorn OOP, els diagrames de classes de UML pràcticament representen el mateix que els diagrames E/R,
 - ▶ excepte ajustos notacionals i una única diferència important: doble caixa per representar una entitat dèbil;

Modelat conceptual: Enfocs

Varis enfocs disponibles

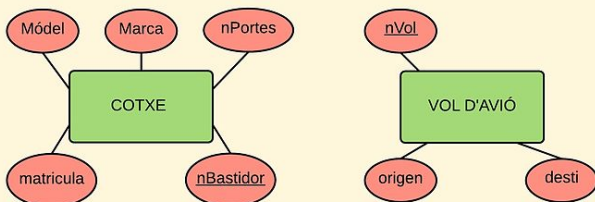
Opcions de modelat:

- ▶ Amb **esquemes relacionals**:
 - ▶ Els que, gràcies a SQL, podem traduir directament per crear la base de dades (relacions, noms i tipus dels atributs, etc.);
- ▶ Amb **diagrames E/R** (Entitat/Relació), o bé
- ▶ Amb **diagrames de classes** de UML:
 - ▶ si bé estan pensats per l'especificació de classes en un entorn OOP, els diagrames de classes de UML pràcticament representen el mateix que els diagrames E/R,
 - ▶ excepte ajustos notacionals i una única diferència important: doble caixa per representar una entitat dèbil;
- ▶ Els esquemes relacionals estan més propers al nivell concret de la pròpia base de dades; els altres estan més propers al nivell humà de raonar i modelar.
- ▶ Ens centrarem en els diagrames UML, dels que deduirem els esquemes relacionals.

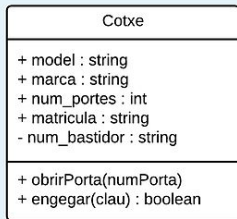
Modelat conceptual: Enfocs

Comparativa entre diagrama E/R i diagrama de classes de UML

Representació Entitat - Relació



Modelatge en UML



Modelat conceptual: elements

El model s'obté combinant elements de modelat

Elements de modelat:

- ▶ Cada opció consta de la seva pròpia tipologia d'elements;
- ▶ Taules i claus en models relacionals, entitats i relacions en E/R, classes i associacions en UML...

Modelat conceptual: elements

El model s'obté combinant elements de modelat

Elements de modelat:

- ▶ Cada opció consta de la seva pròpia tipologia d'elements;
- ▶ Taules i claus en models relacionals, entitats i relacions en E/R, classes i associacions en UML...
- ▶ Però, **sempre**:
 - ▶ Cada element d'un model ha de ser la contrapartida **clara** d'un fet o percepció de la realitat que estem modelant;
 - ▶ ha de ser un fet o percepció **rellevant**;
 - ▶ **tots** els fets o percepcions rellevants han d'estar recollits en el model;
 - ▶ i han d'estar-hi **un sol cop** (**DRY**: *Don't Repeat Yourself*).

Cap a models útils

Té bona pinta el nostre model?

Criteris:

- ▶ Cada element ha de tenir un significat clarament diferent del significat de la resta (**DRY**).
- ▶ Un model gran és probablement millorable si s'estructura millor.
- ▶ Un model complicat és probablement millorable si s'analitza millor.
- ▶ És desitjable que el significat de cada element del model es pugui explicar a un usuari no informàtic, i que ho entengui.
- ▶ Els casos en que un element pot no correspondre a res (per ex. atributs NULL) han de ser molt excepcionals.

Modelat conceptual desitjat

Per la construcció d'una Base de Dades Relacional

Són necessaris **esquemes relacionals**.

Para cada relació de la Base de Dades:

- ▶ nom,
- ▶ noms dels atributs,
- ▶ documentació addicional (connexions entre les relacions i altres restriccions d'integritat).

Dues maneres d'arribar-hi:

- ▶ amb diagrames E/R o diagrames de classes
- ▶ amb la teoria de la normalització (anàlisi de les dependències funcionals).

Hem de ser capaços d'aplicar **ambdós**.

Modelat conceptual per diagrames de classes

Models UML estàtics

Intermedis en abstracció entre la visió humana i els esquemes relacionals:

- ▶ classes,
- ▶ atributs,
- ▶ mètodes,
- ▶ associacions entre classes,
- ▶ multiplicitats,
- ▶ classes associatives,
- ▶ especialitzacions.

Modelat conceptual per diagrames de classes

Models UML estàtics

Intermedis en abstracció entre la visió humana i els esquemes relacionals:

- ▶ classes,
- ▶ atributs,
- ▶ mètodes,
- ▶ associacions entre classes,
- ▶ multiplicitats,
- ▶ classes associatives,
- ▶ especialitzacions.

En general, els **verbs** corresponen a mètodes i els **noms comuns** corresponen a classes o a atributs.

Tanmateix, verbs com **“té”** o **“consta de”** es poden referir a atributs respecte a classes o a associacions entre classes.

Els mètodes no els usarem doncs no desenvoluparem programari.

Multiplicitats

Quantitat d'objectes relacionats

Multiplicitats: Quants objectes de la classe llunyana es relaciona amb la classe propera.

Aquests grups de multiplicitats són equivalents:

- ▶ 0 (no l'usarem, no té utilitat)
- ▶ 0..1
- ▶ 1
- ▶ * 0..* n 0..n (no usarem * perquè presta a confusió)
- ▶ 1..* 1..n.
- ▶ 7 (es pot generalitzar amb 1..*).

Multiplicitats

Quantitat d'objectes relacionats

Multiplicitats: Quants objectes de la classe llunyana es relaciona amb la classe propera.

Aquests grups de multiplicitats són equivalents:

- ▶ 0 (no l'usarem, no té utilitat)
- ▶ 0..1
- ▶ 1
- ▶ * 0..* n 0..n (no usarem * perquè presta a confusió)
- ▶ 1..* 1..n.
- ▶ 7 (es pot generalitzar amb 1..*).

Per tant usarem aquestes multiplicitats:

- ▶ 0..1
- ▶ 1
- ▶ 0..*
- ▶ 1..*

Pista fonamental: substantius de l'enunciat

Però és una classe o un atribut?

El mantenim com atribut si la seva estructura interna és irrellevant (indivisibilitat).

Pista fonamental: substantius de l'enunciat

Però és una classe o un atribut?

El mantenim com atribut si la seva estructura interna és irrellevant (indivisibilitat).

Passa a ser una classe si:

- ▶ cada valor de l'atribut consta òbviamment de vàries parts rellevants,
- ▶ potser ens manca el valor
(recordem la discussió sobre NULL),
- ▶ ens agradaria que tingués varis valors simultàniament,
- ▶ trobem difícil concretar el domini,
- ▶ el mateix nom d'atribut apareix vàries vegades i amb la mateixa semàntica al llarg del procés de modelat...

Pista fonamental: substantius de l'enunciat

Però és una classe o un atribut?

El mantenim com atribut si la seva estructura interna és irrellevant (indivisibilitat).

Passa a ser una classe si:

- ▶ cada valor de l'atribut consta òbviament de varies parts rellevants,
- ▶ potser ens manca el valor
(recordem la discussió sobre NULL),
- ▶ ens agradaria que tingués varis valors simultàniament,
- ▶ trobem difícil concretar el domini,
- ▶ el mateix nom d'atribut apareix varies vegades i amb la mateixa semàntica al llarg del procés de modelat...

Dificultat crucial:

Posar-hi **suficient** imaginació, però **no massa**!

Màquines de la factoria, I

Un exemple inicial

Una factoria immensa té una elevada quantitat de màquines. De cadascuna tenim el número amb que està inventariada i el seu fabricant.

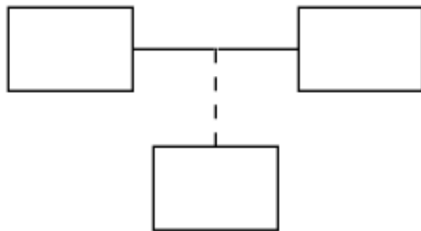
Periòdicament, un tècnic supervisa individualment cada màquina. Cada tècnic, de qui coneixem el DNI i les seves dades personals, coneix la distribució de la seva setmana laboral en franjes horàries amb la indicació de quina màquina ha de supervisar en cadascuna d'aquestes franges.

Màquines de la factoria, I

Un exemple initial

Una factoria immensa té una elevada quantitat de màquines. De cadascuna tenim el número amb que està inventariada i el seu fabricant.

Periòdicament, un tècnic supervisa individualment cada màquina. Cada tècnic, de qui coneixem el DNI i les seves dades personals, coneix la distribució de la seva setmana laboral en franjes horàries amb la indicació de quina màquina ha de supervisar en cadascuna d'aquestes franges.

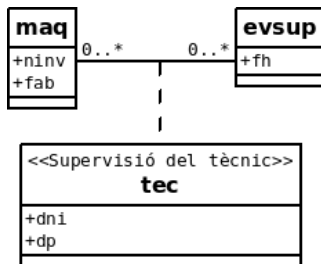


Màquines de la factoria, I

Un exemple inicial

Una factoria immensa té una elevada quantitat de màquines. De cadascuna tenim el número amb que està inventariada i el seu fabricant.

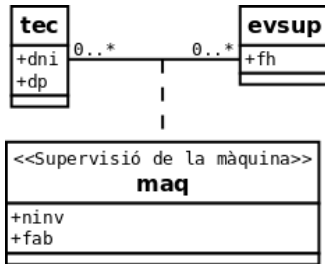
Periòdicament, un tècnic supervisa individualment cada màquina. Cada tècnic, de qui coneixem el DNI i les seves dades personals, coneix la distribució de la seva setmana laboral en franjes horàries amb la indicació de quina màquina ha de supervisar en cadascuna d'aquestes franges.



Màquines de la factoria, I

La solució no és única

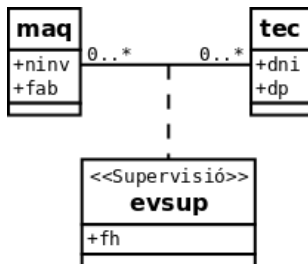
Una altra possible solució. Un tècnic en concret en una franja horària determinada només revisa una màquina.



Màquines de la factoria, I

Cal estar atents a com usem les classes associatives

Aquesta solució és **INCORRECTA**. Està indicant que un tècnic en concret i una màquina determinada només poden tenir una única supervisió. O sigui que un tècnic només podrà supervisar una màquina concreta un cop a la seva vida, supòsit que no està descrit a l'enunciat del problema.



Esquemes relacionals

Informació que explica com s'organitza la informació en una Base de Dades Relacional

Una Base de Dades Relacional consta de **relacions** (també anomenades **taules**), junt amb algunes **restriccions** sobre elles.

L'Esquema Relacional d'una Base de Dades és el resum que permet:

- ▶ programar les instruccions de creació de les relacions,
- ▶ garantir que les dades que es vagin inserint compleixen determinats requisits,
- ▶ accelerar determinades operacions (en especial els **join**),
- ▶ tenir criteris per accelerar altres operacions amb la creació d'índexs.

Esquemes relacionals

Informació que explica com s'organitza la informació en una Base de Dades Relacional

Una Base de Dades Relacional consta de **relacions** (també anomenades **taules**), junt amb algunes **restriccions** sobre elles.

L'Esquema Relacional d'una Base de Dades és el resum que permet:

- ▶ programar les instruccions de creació de les relacions,
- ▶ garantir que les dades que es vagin inserint compleixen determinats requisits,
- ▶ accelerar determinades operacions (en especial els **join**),
- ▶ tenir criteris per accelerar altres operacions amb la creació d'índexs.

Consta de cadascun dels **esquemes de relació** de cadascuna de les seves relacions.

Els esquemes de relació

Estudi de cadascuna de les taules, per separat i conjuntament

L'esquema de relació

(la informació del CREATE TABLE, essencialment) consta de:

- ▶ nom de la relació,
- ▶ nom i domini de cadascun dels atributs, amb les seves possibles condicions NOT NULL,
- ▶ claus:
 - ▶ clau primària (PRIMARY KEY),
 - ▶ claus alternatives (UNIQUE),
 - ▶ claus foranes (REFERENCES) i el seu ús.

Els esquemes de relació

Estudi de cadascuna de les taules, per separat i conjuntament

L'esquema de relació

(la informació del CREATE TABLE, essencialment) consta de:

- ▶ nom de la relació,
- ▶ nom i domini de cadascun dels atributs, amb les seves possibles condicions NOT NULL,
- ▶ claus:
 - ▶ clau primària (PRIMARY KEY),
 - ▶ claus alternatives (UNIQUE),
 - ▶ claus foranes (REFERENCES) i el seu ús.

(A SQLite: .schema, a MySQL: desc, a PostgreSQL: \d)

Claus

Primàries, alternatives i foranes

Claus:

Són conjunts formats per un o més atributs, que

- ▶ determinen tota la relació, i
- ▶ són el més petits possible sota aquesta condició.
- ▶ Una clau s'escull com **primària**, les demés són **alternatives**.

(Un atribut que pertany a una clau primària o alternativa **no hauria de poder ser nul**, però SQL ho permet.)

Claus

Primàries, alternatives i foranes

Claus:

Són conjunts formats per un o més atributs, que

- ▶ determinen tota la relació, i
- ▶ són el més petits possible sota aquesta condició.
- ▶ Una clau s'escull com **primària**, les demés són **alternatives**.

(Un atribut que pertany a una clau primària o alternativa **no hauria de poder ser nul**, però SQL ho permet.)

Un o més atributs d'una relació es pot “referir” a una altre relació guardant un valor de la seva clau primària (**clau forana**).

Els atributs de la clau forana apunten a una tupla existent en la taula referida (“restricció d'integritat referencial”).

Claus foranes, I

Regla d'or de les claus foranes

La clau forana apunta a la clau primària de la taula destí.

Per tant la clau forana ha de tenir els mateixos atributs que la clau primària a la que apunta. Si la clau primària està formada per varis atributs, la clau forana que “apunti” a aquesta clau primària ha de tenir els mateixos atributs.

```
CREATE TABLE ciutat(  
    nom_ciutat VARCHAR(50) NOT NULL,  
    nom_pais VARCHAR(20) NOT NULL,  
    PRIMARY KEY(nom_ciutat, nom_pais));
```

```
CREATE TABLE client (  
    codi CHAR(8) NOT NULL PRIMARY KEY,  
    nom_ciutat VARCHAR(50) NOT NULL,  
    nom_pais VARCHAR(20) NOT NULL,  
    FOREIGN KEY (nom_ciutat, nom_pais) REFERENCES  
ciutat);
```

Claus foranes, II

Modificacions possibles

Les claus foranes (en SQL: REFERENCES) permeten associar a una tupla informació procedent d'altres taules.

Amb cada clau forana:

- ▶ Què fer quan la clau primària externa referida **canvia** el seu valor?

Clàusula de propagació "ON UPDATE".

- ▶ Què fer quan **s'elimina** la tupla identificada per la clau primària externa referida?

Clàusula de propagació "ON DELETE".

Claus foranes, II

Modificacions possibles

Les claus foranes (en SQL: REFERENCES) permeten associar a una tupla informació procedent d'altres taules.

Amb cada clau forana:

- ▶ Què fer quan la clau primària externa referida **canvia** el seu valor?

Clàusula de propagació "ON UPDATE".

- ▶ Què fer quan **s'elimina** la tupla identificada per la clau primària externa referida?

Clàusula de propagació "ON DELETE".

En SQLite, per a que estiguin operatives s'han d'activar explícitament en cada sessió; si no es fa, per defecte estan *disabled*:

```
PRAGMA foreign_keys = ON;
```

Claus foranes, III

Clàusules de propagació de modificacions

Possibles accions:

(No totes disponibles en tots els SGBD!)

- ▶ NO ACTION
- ▶ RESTRICT
- ▶ CASCADE
- ▶ SET NULL
- ▶ SET DEFAULT

Transformació

Què fem ara amb el diagrama de classes UML?

Un cop tenim el diagrama

de classes i associacions, amb totes les multiplicitats,

- ▶ **cada classe** dona lloc a una **relació** (taula), amb els atributs com camps, marcats amb NOT NULL sempre que sigui factible,

Transformació

Què fem ara amb el diagrama de classes UML?

Un cop tenim el diagrama

de classes i associacions, amb totes les multiplicitats,

- ▶ **cada classe** dona lloc a una **relació** (taula), amb els atributs com camps, marcats amb NOT NULL sempre que sigui factible,
- ▶ a cada relació anotem la llista de **dependències funcionals** que compleix: clau primària i claus alternatives,

Transformació

Què fem ara amb el diagrama de classes UML?

Un cop tenim el diagrama

de classes i associacions, amb totes les multiplicitats,

- ▶ **cada classe** dona lloc a una **relació** (taula), amb els atributs com camps, marcats amb NOT NULL sempre que sigui factible,
- ▶ a cada relació anotem la llista de **dependències funcionals** que compleix: clau primària i claus alternatives,
- ▶ i les associacions s'implementen amb taules, amb claus foranes a les relacions que participen en l'associació. La clau primària sol estar formada pel **parell de claus foranes**.

Transformació

Què fem ara amb el diagrama de classes UML?

Un cop tenim el diagrama

de classes i associacions, amb totes les multiplicitats,

- ▶ **cada classe** dona lloc a una **relació** (taula), amb els atributs com camps, marcats amb NOT NULL sempre que sigui factible,
- ▶ a cada relació anotem la llista de **dependències funcionals** que compleix: clau primària i claus alternatives,
- ▶ i les associacions s'implementen amb taules, amb claus foranes a les relacions que participen en l'associació. La clau primària sol estar formada pel **parell de claus foranes**.
- ▶ Les classes associatives simplement afegeixen els seus atributs a la taula que implementa la associació.

Màquines de la factoria, I

Una possible solució

Classes: maq (màquina), tec (tècnic), evsup (event de supervisió)

```
CREATE TABLE maq
  (ninv int NOT NULL, fab text, PRIMARY KEY(ninv));
```

```
CREATE TABLE evsup
  (fh text NOT NULL, PRIMARY KEY(fh));
```

```
CREATE TABLE tec (
  dni int NOT NULL, dp text,
  fh text NOT NULL,
  ninv int NOT NULL,
  FOREIGN KEY (fh) REFERENCES evsup
    ON UPDATE CASCADE,
  FOREIGN KEY (ninv) REFERENCES maq
    ON UPDATE CASCADE,
  PRIMARY KEY(fh,ninv), UNIQUE(fh,dni));
```

Màquines de la factoria, I

Mateixa solució amb SQL simplificat

```
CREATE TABLE maq
  (ninv int PRIMARY KEY NOT NULL, fab text);

CREATE TABLE evsup
  (fh text PRIMARY KEY NOT NULL);

CREATE TABLE tec (
  dni int NOT NULL, dp text,
  fh text NOT NULL
    REFERENCES evsup ON UPDATE CASCADE,
  ninv int NOT NULL
    REFERENCES maq ON UPDATE CASCADE,
  PRIMARY KEY(fh,ninv), UNIQUE(fh,dni));
```

Ús de claus foranes

Algunes qüestions a tenir presents

Excepció: és convenient (encara que no obligatori), quan apareixen multiplicitats 1 o 0..1, implementar les associacions amb claus foranes.

Si usem una clau forana

per implementar una associació amb multiplicitat 1 o 0..1:

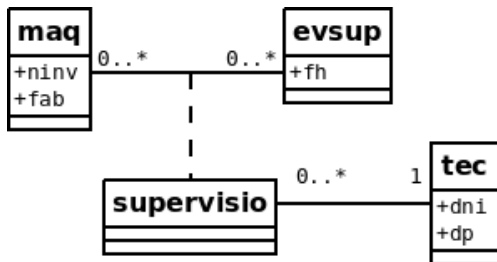
- ▶ Aquesta multiplicitat indica una dependència funcional.
- ▶ Per tant, simplement, l'altra classe conté un o varis camps que actúa com **clau forana** a la classe amb multiplicitat 1 o 0..1.
- ▶ Si la multiplicitat és 1 correspon a una condició NOT NULL.
- ▶ Si l'associació té una **classe associativa** (cosa poc habitual), tots els seus camps s'afegeixen a la classe que conté la clau forana.
- ▶ Ull amb les clàusules de propagació de les modificacions en claus foranes. Per exemple, si s'ha definit un ON DELETE CASCADE, es pot perdre tota la informació de la relació associada.

Màquines de la factoria, II

(Enunciat repetit! Una altra possible solució més elegant)

Una factoria immensa té una elevada quantitat de màquines. De cadascuna tenim el número amb que està inventariada i el seu fabricant.

Periòdicament, un tècnic supervisa individualment cada màquina. Cada tècnic, de qui coneixem el DNI i les seves dades personals, coneix la distribució de la seva setmana laboral en franjes horàries amb la indicació de quina màquina ha de supervisar en cadascuna d'aquestes franges.



Màquines de la factoria, II

Una altra possible solució més elegant

Ara usarem una clau forana per implementar l'associació amb multiplicitat 1 entre tècnic i supervisió.

```
CREATE TABLE tec
    (dni int PRIMARY KEY NOT NULL, dp text);

CREATE TABLE maq
    (ninv int PRIMARY KEY NOT NULL, fab text);

CREATE TABLE evsup
    (fh text PRIMARY KEY NOT NULL);

CREATE TABLE supervisio (
    fh text NOT NULL
        REFERENCES evsup ON UPDATE CASCADE,
    ninv int NOT NULL
        REFERENCES maq ON UPDATE CASCADE,
    dni int NOT NULL
        REFERENCES tec ON UPDATE CASCADE,
    PRIMARY KEY(fh,dni), UNIQUE(fh,ninv));
```


Especialitzacions

És suficient mantenir la mateixa clau

Una especialització

es pot representar simplement assegurant que les tuples de la **classe especialitzada** mantinguin com **clau primària** la mateixa que ja s'usa en la **classe general**.

- ▶ En aquest cas, aquesta clau és, simultàniament, clau primària i clau forana.
- ▶ Si l'especialització no afegeix nous atributs o associacions, es pot escollir representar-la amb un atribut booleà en la classe general
(però aquesta opció és infreqüent).

Claus dèbils

(“entitats dèbils” en esquemes E-R):

- ▶ Classes que la seva clau primària conté una clau forana.
- ▶ Les diferenciem amb una “doble caixa” en el diagrama de classes.

Anomalies

Entenem per **anomalies** situacions en les que el disseny escollit per la Base de Dades ens dificulta mantenir la informació correctament.

Exemple:

Considerem un esquema de relació

`Funcionari (NRP, Nom, Destí, Estat, DataEstat)`

a on rebem la indicació de que el mateix `Funcionari` pot tenir simultàniament varis `Destíns`, en concret quan `Estat` té el valor `Excedència`.

Tipus d'anomalies

Exemples sobre l'esquema de relació donat

Funcionari (NRP, Nom, Destí, Estat, DataEstat)

Exemples d'anomalies

► d'inserció:

com anotes el Nom i el NRP d'un funcionari si resulta que encara no saps el seu Destí?

Tipus d'anomalies

Exemples sobre l'esquema de relació donat

Funcionari (NRP, Nom, Destí, Estat, DataEstat)

Exemples d'anomalies

- ▶ d'inserció:

com anotes el Nom i el NRP d'un funcionari si resulta que encara no saps el seu Destí?

- ▶ d'eliminació:

si elimines tota la informació dels Destíns d'un funcionari, perds la relació entre NRP i Nom;

Tipus d'anomalies

Exemples sobre l'esquema de relació donat

Funcionari (NRP, Nom, Destí, Estat, DataEstat)

Exemples d'anomalies

- ▶ d'inserció:
com anotes el Nom i el NRP d'un funcionari si resulta que encara no saps el seu Destí?
- ▶ d'eliminació:
si elimines tota la informació dels Destíns d'un funcionari, perds la relació entre NRP i Nom;
- ▶ d'actualització:
si canvia el Nom associat a un NRP, s'ha d'actualitzar tants cops com valors diferents de Destí apareguin.

Dependències funcionals, I

Definició

El problema radica en que el NRP **determina** el Nom.

Per tant, si són necessàries vàries tuples, existeix **redundància** en la informació.

Sigui \mathcal{R} el conjunt d'atributs, $P \subseteq \mathcal{R}$ i $Q \subseteq \mathcal{R}$.

Existeix una **dependència funcional** $P \rightarrow Q$

(llegit: “ P determina Q ”)

si succeeix el següent:

Sempre que dos tuples tinguin els mateixos valors en els atributs de P , forçosament hauran de tenir també els mateixos valors en els atributs de Q .

Per exemple, si $P \subseteq \mathcal{R}$ conté una clau primària o alternativa de R , $P \rightarrow \mathcal{R}$.

Dependències funcionals, II

Per què requereixen atenció?

Les dependències funcionals mal gestionades implica tenir esquemes relacionals mal dissenyats que comporten:

- ▶ tenir relacions amb dades redundants i usar amb freqüència clàusules `DISTINCT` (amb consultes més lentes) en els `SELECT`, per exemple:

```
SELECT DISTINCT NRP, Nom FROM Funcionari;
```

- ▶ i dificulten els *joins*, creant oportunitats de *lossy-join* que “perden informació” per excés de tuples.

Dependències funcionals, III

Apareixen quan els valors d'alguns atributs poden **determinar** els valors d'altres atributs.

Exemples:

- ▶ El Número de Registre de Personal determina el nom del funcionari.
- ▶ Els atributs de direcció postal (ciutat, carrer i número) determinen conjuntament el codi postal.
- ▶ El títol de la pel·lícula, el seu director i l'any de la seva estrena determinen conjuntament la llista d'actors.
- ▶ Habitualment s'espera que el número de DNI determini totes les dades de la persona.

En una aplicació real no t'en pots fiar del tot.

- ▶ Una clau primària determina tots els valors de la tupla.

Màquines de la factoria, III

(Enunciat repetit!)

Una factoria immensa té una elevada quantitat de màquines. De cadascuna tenim el número amb que està inventariada i el seu fabricant.

Periòdicament, un tècnic supervisa individualment cada màquina. Cada tècnic, de qui coneixem el DNI i les seves dades personals, coneix la distribució de la seva setmana laboral en franges horàries amb la indicació de quina màquina ha de supervisar en cadascuna d'aquestes franges.

Dependències funcionals:

Màquines de la factoria, III

(Enunciat repetit!)

Una factoria immensa té una elevada quantitat de màquines. De cadascuna tenim el número amb que està inventariada i el seu fabricant.

Periòdicament, un tècnic supervisa individualment cada màquina. Cada tècnic, de qui coneixem el DNI i les seves dades personals, coneix la distribució de la seva setmana laboral en franges horàries amb la indicació de quina màquina ha de supervisar en cadascuna d'aquestes franges.

Dependències funcionals:

- ▶ $\text{DNI} \rightarrow \text{D.P.}$
- ▶ $\text{N.Inv.} \rightarrow \text{Fab}$
- ▶ $\text{DNI F.H.} \rightarrow \text{N.Inv.}$
- ▶ $\text{N.Inv. F.H.} \rightarrow \text{DNI}$

Formes normals, I

Es basen en analitzar dependències

Existeixen dependències inevitables (i desitjables): en concret, les **claus** causen dependències funcionals inevitables.

Per evitar les anomalies vistes anteriorment cal normalitzar la base de dades per a que la majoria (o totes) les dependències funcionals es converteixin en claus de les relacions.

Tenim diferents graus de normalització: Formes normals (NF).

Les formes normals van ser introduïdes per Codd el 1970. Cada forma normal implica que està normalitzada per les formes normals inferiors, per exemple la 3NF implica que també és 1NF i 2NF.

Informalment, una base de dades es diu que està "normalitzada" quan compleix que és 3NF. La majoria de relacions 3NF són lliures d'anomalies en les insercions, actualitzacions o eliminacions.

Descomposició d'una relació

Objectius

Si una relació R no està en la forma normal desitjada, la podem descompondre en vàries relacions $R_1, R_2 \dots R_n$ que estiguin en aquesta forma normal.

- ▶ Cada nova relació R_i conté un subconjunt d'atributs de R
- ▶ Cada atribut de R apareix com un atribut en almenys una de les noves relacions R_i

Descomposició d'una relació

Objectius

Si una relació R no està en la forma normal desitjada, la podem descompondre en diverses relacions $R_1, R_2 \dots R_n$ que estiguin en aquesta forma normal.

- ▶ Cada nova relació R_i conté un subconjunt d'atributs de R
- ▶ Cada atribut de R apareix com un atribut en almenys una de les noves relacions R_i

Cal assegurar-nos que:

- ▶ Descomposició que sigui *lossless-join* (sempre)
- ▶ No hi hagi redundàncies (BCNF, 3NF no sempre totes)
- ▶ Preservació de DF (3NF, BCNF no sempre totes)

Descomposició *lossless-join*, I

La descomposició de la relació R en relacions més petites $R_1, R_2 \dots R_n$ és *lossless-join* si:

- ▶ Tots els atributs de R apareixen en la descomposició:

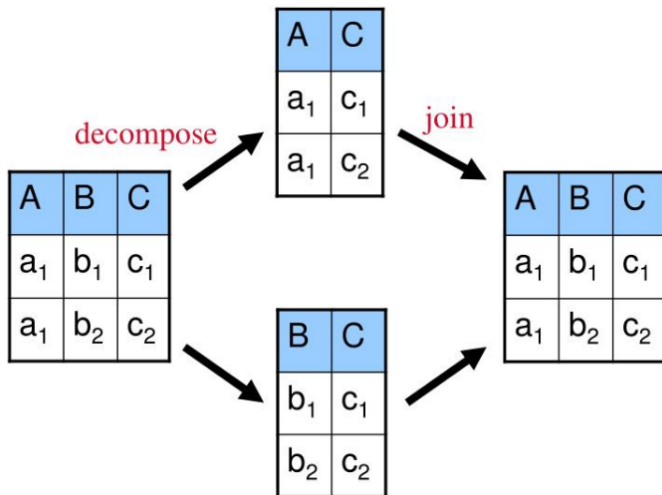
$$Atributs(R) = Atributs(R_1) \cup Atributs(R_2) \cup \dots \cup Atributs(R_n)$$

- ▶ Som capaços de reconstruir R fent natural joins de les relacions petites:

$$R = R_1 \bowtie R_2 \bowtie \dots \bowtie R_n$$

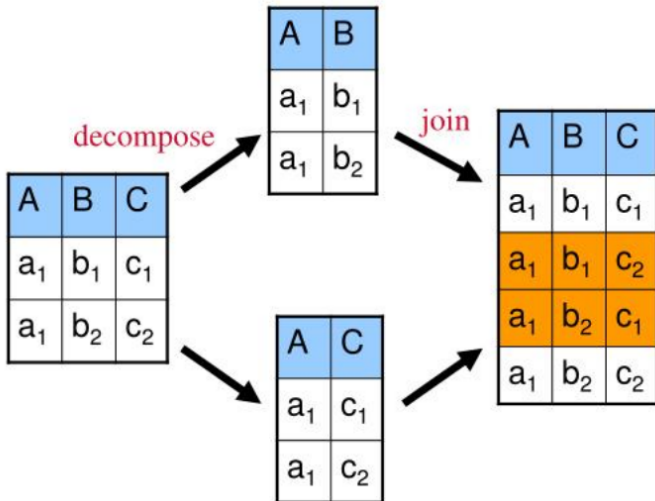
Descomposició *lossless-join*, II

Exemple



Descomposició *lossy-join*

Exemple



Descomposició *lossless-join*, III

Verificació

Una descomposició de la relació R en relacions $R_1, R_2 \dots R_n$ és *lossless-join* si per qualsevol parella de relacions R_i, R_j :

- ▶ $Atributs(R_i) \cap Atributs(R_j) \rightarrow Atributs(R_i)$ o
- ▶ $Atributs(R_i) \cap Atributs(R_j) \rightarrow Atributs(R_j)$

sempre que la intersecció no sigui buida.

Descomposició *lossless-join*, III

Verificació

Una descomposició de la relació R en relacions $R_1, R_2 \dots R_n$ és *lossless-join* si per qualsevol parella de relacions R_i, R_j :

- ▶ $Atributs(R_i) \cap Atributs(R_j) \rightarrow Atributs(R_i)$ o
- ▶ $Atributs(R_i) \cap Atributs(R_j) \rightarrow Atributs(R_j)$

sempre que la intersecció no sigui buida.

Per exemple:

$$R = \{A, B, C, D, E\}$$

$$DF = \{A \rightarrow BC, B \rightarrow D, CD \rightarrow E\}$$

- ▶ $R_1 = \{A, B, C\}$ $R_2 = \{A, D, E\}$ és una descomposició *lossless-join*?
- ▶ $R_1 = \{A, B, C\}$ $R_2 = \{C, D, E\}$ és una descomposició *lossless-join*?

1NF

1a Forma Normal

Per satisfer la 1NF cal que els valors de cada columna d'una relació siguin atòmics (no tinguin múltiples valors). Tampoc és desitjable que hi hagin atributs repetits (per ex. Assig1, Assig2, Assig3, ...).

Per exemple aquesta relació no està en 1NF doncs algun valor de **Assig** no és atòmic:

CodiEst	NomEst	Ciutat	Comarca	Assig	DataAprovat
E101	Marta	Sitges	Garraf	DABD	01-07-2019
E101	Marta	Sitges	Garraf	PMUD	30-01-2020
E102	Pol	Piera	Anoia	DABD,AMEP	28-06-2019
E103	Berta	Vilanova	Garraf	ESIN	01-02-2020
E104	Pau	Sitges	Garraf	DABD	01-07-2019

1NF

Exemple

La relació anterior la podem convertir a 1NF així:

CodiEst	NomEst	Ciutat	Comarca	Assig	DataAprovat
E101	Marta	Sitges	Garraf	DABD	01-07-2019
E101	Marta	Sitges	Garraf	PMUD	30-01-2020
E102	Pol	Piera	Anoia	DABD	28-06-2019
E102	Pol	Piera	Anoia	AMEP	28-06-2019
E103	Berta	Vilanova	Garraf	ESIN	01-02-2020
E104	Pau	Sitges	Garraf	DABD	01-07-2019

2NF

2a Forma Normal

Per satisfer la 2NF cal que la relació sigui 1NF i que qualsevol atribut no clau depengui de la clau sencera, no d'una part.

La darrera relació no estava en 2NF doncs la clau és **CodiEst, Assig** però atributs com **NomEst**, **Ciutat** o **Comarca** només depenen de **CodiEst**. Per satisfer 2NF cal partir-la en aquestes 2 relacions:

CodiEst	NomEst	Ciutat	Comarca
E101	Marta	Sitges	Garraf
E102	Pol	Piera	Anoia
E103	Berta	Vilanova	Garraf
E104	Pau	Sitges	Garraf

CodiEst	Assig	DataAprovat
E101	DABD	01-07-2019
E101	PMUD	30-01-2020
E102	DABD	28-06-2019
E102	AMEP	28-06-2019
E103	ESIN	01-02-2020
E104	DABD	01-07-2019

3NF

3a Forma Normal

Per satisfer la 3NF cal que la relació sigui 2NF i que no tingui dependències transitives dins seu.

La darrera relació dels estudiants no estava en 3NF doncs **Comarca** depen de **Ciutat** i **Ciutat** depen de **CodiEst**. Cal partir-la en aquestes 2 relacions per satisfer 3NF:

CodiEst	NomEst	Ciutat
E101	Marta	Sitges
E102	Pol	Piera
E103	Berta	Vilanova
E104	Pau	Sitges

Ciutat	Comarca
Sitges	Garraf
Piera	Anoia
Vilanova	Garraf

Formes normals, II

Les formes normals més habituals

Opcions usals de normalització:

- ▶ tercera forma normal (**3NF**), que és la més freqüent amb diferència, o
- ▶ forma normal de Boyce-Codd (**BCNF**), també anomenada 3.5NF.

Es diferencien en el grau en que eviten dependències evitables.

- ▶ En BCNF no existeixen dependències funcionals, excepte les inevitables (degudes a les claus);
- ▶ en 3NF es permeten algunes que són evitables, si quan les evitem perdem informació que volem conservar sobre altres dependències.

Formes normals, III

La definició precisa de les formes normals

\mathcal{R} està en **forma normal de Boyce-Codd** (BCNF) si:

sempre que $X \rightarrow A$, per $X \subseteq \mathcal{R}$ i $A \in \mathcal{R}$,

$A \in X$ o bé X conté una clau de \mathcal{R} ;

\mathcal{R} està en **tercera forma normal** (3NF) si:

sempre que $X \rightarrow A$, per $X \subseteq \mathcal{R}$ i $A \in \mathcal{R}$,

$A \in X$ o bé X conté una clau de \mathcal{R} ,

o bé A pertany a una clau (primària o alternativa).

S'aconsegueix “normalitzant” els esquemes.

Es a dir, descomposant la relació en relacions més petites en que les dependències corresponguin a claus.

Màquines de la factoria, III

Una factoria immensa té una elevada quantitat de màquines. De cadascuna tenim el número amb que està inventariada i el seu fabricant.

Periòdicament, un tècnic supervisa individualment cada màquina. Cada tècnic, de qui coneixem el DNI i les seves dades personals, coneix la distribució de la seva setmana laboral en franges horàries amb la indicació de quina màquina ha de supervisar en cadascuna d'aquestes franges.

Dependències funcionals:

- ▶ $\text{DNI} \rightarrow \text{D.P.}$
- ▶ $\text{N.Inv.} \rightarrow \text{Fab}$
- ▶ $\text{DNI F.H.} \rightarrow \text{N.Inv.}$
- ▶ $\text{N.Inv. F.H.} \rightarrow \text{DNI};$

El primer esquema relacional que havíem fet no està ni en **BCNF** ni en **3NF** doncs la dependència $\text{DNI} \rightarrow \text{D.P.}$ no correspon a cap clau.

Màquines de la factoria, III

Una possible solució normalitzada

Aquest esquema, on hem creat una nova relació tec on DNI és una clau primària, ja està en **BCNF** i, per tant, també en **3NF** doncs **totes** les dependències corresponen a claus (primàries/alternatives).

```
CREATE TABLE tec
```

```
(dni int PRIMARY KEY NOT NULL, dp text);
```

```
CREATE TABLE maq
```

```
(ninv int PRIMARY KEY NOT NULL, fab text);
```

```
CREATE TABLE evsup (
```

```
  fh text NOT NULL,
```

```
  dni int NOT NULL
```

```
    REFERENCES tec ON UPDATE CASCADE,
```

```
  ninv int NOT NULL
```

```
    REFERENCES maq ON UPDATE CASCADE,
```

```
  PRIMARY KEY(fh,dni), UNIQUE(fh,ninv));
```

Màquines de la factoria, IV

Variacions sobre un tema fabril

Una factoria immensa té una elevada quantitat de màquines. De cadascuna tenim el número amb que està inventariada i el seu fabricant.

Periòdicament, tècnics competents supervisen cada màquina, **possiblement en equip**.

Cada tècnic, de qui coneixem el DNI i les seves dades personals, coneix la distribució de la seva setmana laboral en franges horàries amb la indicació de quina màquina ha de supervisar en cadascuna d'aquestes franges.

Dependències funcionals:

Màquines de la factoria, IV

Variacions sobre un tema fabril

Una factoria immensa té una elevada quantitat de màquines. De cadascuna tenim el número amb que està inventariada i el seu fabricant.

Periòdicament, tècnics competents supervisen cada màquina, **possiblement en equip**.

Cada tècnic, de qui coneixem el DNI i les seves dades personals, coneix la distribució de la seva setmana laboral en franges horàries amb la indicació de quina màquina ha de supervisar en cadascuna d'aquestes franges.

Dependències funcionals:

- ▶ $\text{DNI} \rightarrow \text{D.P.}$
- ▶ $\text{N.Inv.} \rightarrow \text{Fab}$
- ▶ $\text{DNI F.H.} \rightarrow \text{N.Inv.}$

Màquines de la factoria, IV

Variacions sobre un tema fabril

Una factoria immensa té una elevada quantitat de màquines. De cadascuna tenim el número amb que està inventariada i el seu fabricant.

Periòdicament, tècnics competents supervisen cada màquina, **possiblement en equip**.

Cada tècnic, de qui coneixem el DNI i les seves dades personals, coneix la distribució de la seva setmana laboral en franges horàries amb la indicació de quina màquina ha de supervisar en cadascuna d'aquestes franges.

Dependències funcionals:

- ▶ $\text{DNI} \rightarrow \text{D.P.}$
- ▶ $\text{N.Inv.} \rightarrow \text{Fab}$
- ▶ $\text{DNI F.H.} \rightarrow \text{N.Inv.}$

L'anterior esquema, on només cal treure la clau alternativa $\text{UNIQUE}(\text{fh}, \text{ninv})$, també està en **BCNF** i, per tant, en **3NF**.

Màquines de la factoria, V

Una factoria immensa té una elevada quantitat de màquines. De cadascuna tenim el número amb que està inventariada i el seu fabricant.

Cada màquina té assignat un tècnic, **sempre el mateix**, que, de manera **individual**, la supervisa.

Cada tècnic, de qui coneixem el DNI i les seves dades personals, coneix la distribució de la seva setmana laboral en franges horàries amb la indicació de quina màquina ha de supervisar en cadascuna d'aquestes franges.

Dependències funcionals:

- ▶ $\text{DNI} \rightarrow \text{D.P.}$
- ▶ $\text{N.Inv.} \rightarrow \text{Fab}$
- ▶ $\text{DNI F.H.} \rightarrow \text{N.Inv.}$
- ▶ $\text{N.Inv.} \rightarrow \text{DNI}$

Màquines de la factoria, V

Una factoria immensa té una elevada quantitat de màquines. De cadascuna tenim el número amb que està inventariada i el seu fabricant.

Cada màquina té assignat un tècnic, **sempre el mateix**, que, de manera **individual**, la supervisa **setmanalment**.

Cada tècnic, de qui coneixem el DNI i les seves dades personals, coneix la distribució de la seva setmana laboral en franges horàries amb la indicació de quina màquina ha de supervisar en cadascuna d'aquestes franges.

Dependències funcionals:

- ▶ $\text{DNI} \rightarrow \text{D.P.}$
- ▶ $\text{N.Inv.} \rightarrow \text{Fab}$
- ▶ $\text{DNI F.H.} \rightarrow \text{N.Inv.}$
- ▶ $\text{N.Inv.} \rightarrow \text{DNI}$
- ▶ $\text{N.Inv.} \rightarrow \text{FH}$

L'anterior esquema també està en **3NF**, però no en **BCNF**.

Dificultats en el procés de normalització

Tres inconvenients

Una base de dades amb un disseny normalitzat ofereix **molts avantatges** un cop està en producció.

L'anàlisi de dependències funcionals

(que es va dissenyar per a bases de dades en les que no existia NULL...) presenta, tanmateix, tres **inconvenients**:

- ▶ Ofereix limitada **escalabilitat**,
- ▶ és **poc intuïtiu**, basat en un desenvolupament principalment simbòlic, i
- ▶ és **propens a errors**: convé complementar-lo amb algun procés de validació.

Bona estratègia: combinar les intuïcions que obtenim amb un model de classes i la seva traducció a relacional, amb l'anàlisi de dependències a petita escala sobre les taules resultants.