

Bases de Dades  
Departament d'Enginyeria Informàtica i Matemàtiques  
Universitat Rovira i Virgili

# Pràctica 3<sup>a</sup> Part: Disseny físic de bases de dades

Josep Bello Curto  
Leandro Favio Gomez Racca



06-06-2020

# Índex

<b>1</b>	<b>Creació de les taules</b>	<b>1</b>
1.1	Claus foranes . . . . .	1
1.2	Codis . . . . .	1
<b>2</b>	<b>Comprovacions</b>	<b>2</b>
2.1	Triggers . . . . .	2
2.2	Checks . . . . .	2
2.3	Stored Procedures . . . . .	2
<b>3</b>	<b>Operacions sobre la base de dades</b>	<b>3</b>
3.1	Tasca F . . . . .	3
3.2	Tasca G . . . . .	3
3.3	Tasca H . . . . .	3

# 1. Creació de les taules

## 1.1 Claus foranes

En la creació de les taules hem decidit que totes les claus foranes associades a un client s'eliminïn en cascada quan aquest client sigui eliminat de la base de dades, d'aquesta forma evitem tenir una factura, per exemple, guardada a nom d'un client que no existeix.

## 1.2 Codis

Per evitar possibles codis repetits tan en les joguines com en els clients hem decidit que aquests camps sigui de tipus `AUTO_INCREMENT`, d'aquesta forma ambdues comencen amb un 1 es van incrementant a mesura que es fiquen més tuples a la taula.

També així evitem que al fer els inserts es passi el mateix codi dues vegades sense voler.

## 2. Comprovacions

*Per evitar incongruències en les dades o complir certes restriccions establertes en l'enunciat hem utilitzat triggers, checks o stored procedures en funció del que hem cregut més oportu.*

### 2.1 Triggers

Per evitar que en la taula Equivalent tinguem les tuples A equivalent B i B equivalent A hem decidit fer un trigger abans d'un insert que ficarà com a joguina aquella que tingui un ID més petit i com a joguina\_equivalent aquella que tingui un ID més gran.

### 2.2 Checks

Per evitar encàrrecs de menys de 10 joguines o de més de 1000 hem ficat un check a la taula que s'encarregarà de comprovar-ho.

I per complir que el tipus de parentesc sigui P, S, T, Q, C hem ficat un check a la taula Parentesc que s'assegura que això sigui així.

### 2.3 Stored Procedures

Per inserir un client, sigui particular o empresa, a la base de dades hem decidit un stored procedure que s'utilitzarà quan vulguem realitzar aquesta operació al qual li passem com a arguments les dades del client, sigui particular o empresa, i en funció de l'argument que sigui NULL inserirà un client particular o un client empresa.

## 3. Operacions sobre la base de dades

### 3.1 Tasca F

Per aquesta tasca es demana fer una vista on tinguem tot l'històric d'encàrrecs de la tenda.

Es demana que si un client ha encarregat la mateixa joguina només aparegui una vegada, això està garantit ja que la clau primària de la taula Encàrrecs és l'ID del client i l'ID de la joguina, per tant un mateix client no pot encarregar la mateixa joguina dues vegades.

### 3.2 Tasca G

Per aquesta tasca hem introduït un nou camp a la taula de Factura que és una cadena de caràcters, un string, on fem el text: No pagat, a totes aquelles factures que fa menys de 5 dies que s'han fet i hem decidit deixar com a NULL la resta ja que no podem assegurar que estiguin pagades.

### 3.3 Tasca H

Per aquesta tasca s'han d'eliminar tots els clients que fa més de 5 anys que no fa una comanda. A part de fer això s'han d'eliminar també totes les referències a aquest client, ja que sinó hi hauria incongruències en les dades. Per garantir que això passi hem fet, com s'ha descrit anteriorment, que totes les claus foranes que facin referència a clients siguin eliminades en cascada quan el client s'elimina.