

# Cas pràctic sobre bases de dades relacionals

Pedro Cardona Vilaplana

PID\_00170164



Universitat Oberta  
de Catalunya

[www.uoc.edu](http://www.uoc.edu)



# Índex

<b>Introducció</b> .....	5
<b>1. Conceptes necessaris</b> .....	7
1.1. El model entitat-relació .....	7
1.2. Entitats i atributs .....	7
1.3. Relacions .....	7
1.4. Claus .....	8
1.5. Classificació de les entitats en funció de les claus .....	8
1.6. Diagrames entitat-relació .....	8
1.7. Cardinalitat de les relacions .....	9
1.8. Generació del model relacional a partir de l'entitat-relació .....	10
1.9. Exemple de conversió d'un model entitat-relació a un model relacional .....	11
1.10. Integritat referencial .....	12
1.11. <i>Structured query language</i> .....	13
1.12. <i>Query by example</i> .....	15
<b>2. Cas pràctic</b> .....	18
2.1. Consulta 1: obtenir els noms dels nostres clients ordenats alfabèticament .....	18
2.2. Consulta 2: obtenir la informació de comptes amb NIF 11111111-A .....	19
2.3. Consulta 3: calcular l'import d'interessos dels clients .....	20
2.4. Consulta 4: calcular la suma del total dels saldos dels comptes .....	22
2.5. Consulta 5: incrementar en un punt el tipus d'interès als clients que es diguin Joan .....	24
2.6. Consulta 6: eliminar el Joan com a client de l'entitat bancària .....	25
2.7. Consulta 7: duplicar la taula "Clients" sobre una de nova anomenada "CLIENTS2" .....	27
2.8. Consulta 8: incorporar un nou client a la taula CLIENTS2 .....	28
<b>Bibliografia</b> .....	31



## Introducció

Les bases de dades són unes eines molt potents que ens permetran gestionar grans quantitats de dades de manera ordenada, i convertir-les en informació mitjançant el seu tractament, i també en coneixement mitjançant la difusió d'aquesta. En aquest mòdul posarem en pràctica conceptes tractats en el mòdul teòric "El model relacional i l'àlgebra relacional", és a dir, el que té a veure amb les bases de dades relacionals i el llenguatge SQL.

Hem enfocat aquest cas pràctic a partir del refrescament (d'una manera molt pràctica i amb exemples) d'alguns dels conceptes clau estudiats en el mòdul i, principalment, amb exercicis sobre la posada en pràctica d'aquests conceptes mitjançant una base de dades d'exemple que nosaltres mateixos construirem.

Atès que es tracta de posar en pràctica la teoria apresada anteriorment és del tot recomanable que feu els exercicis que aquí proposem en una base de dades del vostre propi ordinador personal.

Si bé com a base de dades suport per als exemples i el cas pràctic utilitzarem una base de dades d'àmplia difusió com l'Access 2007, com que hem utilitzat exemples i sentències SQL estàndard podreu fer, amb facilitat, els diferents exercicis sobre qualsevol altra base de dades relacional com, per exemple, l'OpenOffice. Els resultats que obtindreu seran els mateixos.



## 1. Conceptes necessaris

### 1.1. El model entitat-relació

El model entitat-relació (E-R), també anomenat *model conceptual de dades*, és una tècnica de representació de les relacions que tenen les dades de la base de dades, que permet recrear una representació de la realitat que tenim l'objectiu de tractar en la nostra base de dades.

L'elaboració d'un model E-R sempre és un pas previ al disseny que finalment s'implementarà en una base de dades, i comprèn exclusivament una representació del disseny de les dades, i no el que es vol fer amb aquestes.

### 1.2. Entitats i atributs

Una entitat és una cosa o objecte concret o abstracte que existeix i que es pot diferenciar d'altres, com ara persones, mesos de l'any, etc. El primer exemple correspondria a un objecte concret, i el segon a un d'abstracte.

El conjunt d'entitats correspon amb un grup d'entitats del mateix tipus.

En una empresa, el conjunt d'empleats es podria denominar "Empleats".

Una entitat sempre està representada per un conjunt d'atributs que en representen les característiques.

En l'entitat "Empleats", alguns atributs possibles serien els següents: DNI, nom, cognoms, etc.

### 1.3. Relacions

Una relació és una associació que es dona entre diferents entitats.

En una empresa, a més de l'entitat "Empleats" que hem comentat també en tindrem una d'anomenada "Direccions", que fa referència a la direcció organitzativa (de l'organigrama de l'empresa) està a què adscrit l'empleat. Sobre això podríem definir una relació que associés l'empleat Pere amb la direcció "Comptabilitat".

A més, una relació pot tenir també els seus propis atributs.

En el cas anterior, l'atribut de la relació del Pere amb la direcció "Comptabilitat" podria ser la data d'adscripció en la direcció.

## 1.4. Claus

Una clau és el conjunt mínim compost per un o més atributs que permet identificar de manera unívoca una entitat dins d'un conjunt. Per tant, cap subconjunt d'atributs no podrà funcionar també com a clau.

En l'entitat "Empleats", el NIF seria la clau, ja que cap conjunt més petit que aquest no podria identificar de manera unívoca cada un dels empleats.

És possible que en un conjunt d'entitats hi hagi més d'una clau. Totes les claus possibles es denominen **claus candidates**, mentre que la clau escollida pel dissenyador de la base de dades per a identificar cada entitat es denomina **clau primària**.

## 1.5. Classificació de les entitats en funció de les claus

- **Entitats fortes:** són les que tenen una clau primària.
- **Entitats febles:** són les que no tenen entre els seus atributs una clau primària, per la qual cosa depenen d'una entitat forta que els permet identificar, mitjançant una relació, cada un dels seus atributs.

Encara que un conjunt d'entitats febles no té clau primària, es necessita conèixer un mitjà per a distingir totes les entitats del conjunt d'entitats que depenen d'una entitat forta particular. El conjunt mínim d'un o diversos atributs que ho permet és el que anomenem **discriminant** d'un conjunt d'entitats. La clau primària d'una entitat feble està formada pel discriminant més la clau primària de la relació forta amb la qual està associada.

Seguint amb el nostre cas d'empleats d'una empresa, tindriem una entitat anomenada "Nomines" que té els atributs NumNomina, Data i Import. NumNomina és el número de nòmina que ha cobrat l'empleat (la primera nòmina que cobra té NumNomina = 1; la següent, el valor 2, i així successivament). "Nomines" per si mateixa no té una clau que identifiqui una entitat de manera unívoca (podríem tenir dos empleats que en la seva primera nòmina, pagada el mateix dia, haguessin cobrat el mateix). En aquest cas "Nomines" depèn d'"Empleats" per a existir, per la qual cosa el discriminant seria NumNomina (per a un mateix "Empleats" sí que es pot identificar de manera unívoca cada nòmina). Per tant, la clau primària de "Nomines" seria NIF (clau primària de l'entitat forta amb què s'associa, "Empleats") més NumNomina (discriminador, és a dir, que permet la identificació unívoca de cada entitat "Nomines" per a cada empleat).

Les claus foranes són el conjunt d'atributs d'una entitat que al seu torn són clau primària d'una altra entitat amb què està relacionada.

## 1.6. Diagrames entitat-relació

El diagrama entitat-relació permet representar gràficament l'estructura lògica d'una base de dades mitjançant els elements següents:

- Rectangles: representen les entitats.
- El·lipses: representen els atributs de les entitats.



- Rombes: representen les relacions entre les entitats.
- Línies: enllacen atributs a entitats, atributs a relacions, i entitats a relacions.

Suposem que volem modelitzar les notes de les assignatures d'una titulació que han obtingut els alumnes:

- Alumnes dels que coneixem el DNI, nom i cognoms.
- Assignatures de les que sabem l'identificador d'assignatura, nom de l'assignatura i curs a què pertany.

El model entitat-relació corresponent seria:

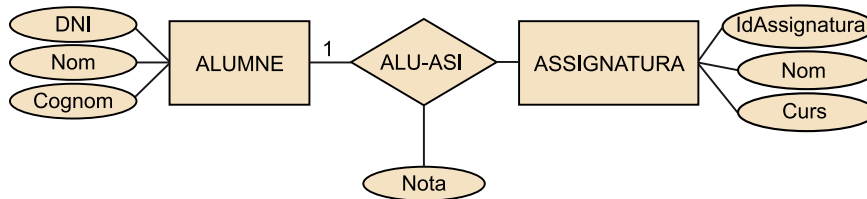


Figura 1. Model entitat-relació que representa la relació entre "Alumne" i "Assignatura".

### 1.7. Cardinalitat de les relacions

La correspondència de cardinalitats, o raó de cardinalitat, expressa el nombre d'entitats a què una altra entitat pot estar associada per mitjà d'una relació o, dit d'una altra manera, expressa el nombre d'entitats amb què pot associar-se una altra entitat.

D'acord amb l'anterior tenim les relacions possibles següents entre dos conjunts d'entitats "A" i "B":

- **Una a una (1:1):** cada instància o ocurrència de l'entitat "A" es relaciona, com a màxim, amb una instància o ocurrència de l'entitat "B" i viceversa.
- **Una a moltes (1:N):** una instància o ocurrència de l'entitat "A" es relaciona amb un nombre qualsevol d'instàncies o ocurrències de l'entitat "B", mentre que una instància o ocurrència de l'entitat "B" es relaciona, com a màxim, amb una instància o ocurrència de l'entitat "A".
- **Moltes a una (N:1):** una instància o ocurrència de l'entitat "A" es relaciona, com a màxim, amb una instància o ocurrència de l'entitat "B", mentre que una instància o ocurrència de l'entitat "B" es relaciona amb un nombre qualsevol d'instàncies o ocurrències de l'entitat "A".
- **Moltes a moltes (N:M):** una instància o ocurrència de l'entitat "A" es relaciona amb un nombre qualsevol d'instàncies o ocurrències de l'entitat "B" i viceversa.

La manera de representar la cardinalitat és:

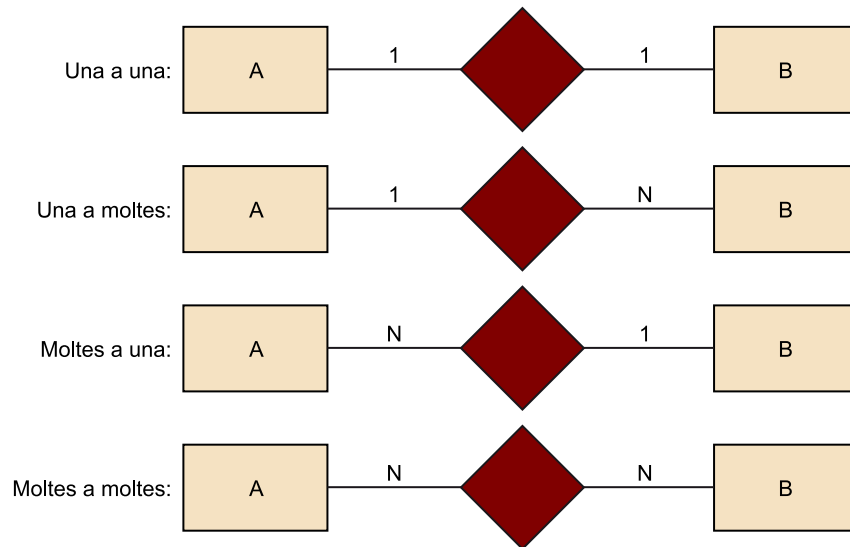


Figura 2. Representacions dels tipus de cardinalitats

### Exemples de cardinalitats

Un préstec pot ser, com a màxim, d'un client i, al seu torn, un mateix client pot tenir diversos préstecs, llavors la relació client a préstec és d'un a diversos.

Si, com a variació de l'exemple anterior, tenim que un préstec no sols pot pertànyer a un client sinó també a més (els titulars del contracte de préstec són diversos), llavors estarem davant d'una relació de diversos a diversos.

## 1.8. Generació del model relacional a partir de l'entitat-relació

Una vegada realitzat el model entitat-relació d'una base de dades, es genera el model relacional. A tall de resum, les regles de conversió per a poder derivar el model relacional a partir de l'entitat-relació són:

- Cada conjunt d'entitats fortes del model entitat-relació es converteix en una taula en el model relacional, i els atributs són els camps de la taula.
- Cada conjunt d'entitats febles es transforma en una taula, i els seus camps són els atributs de l'entitat feble més la clau primària de l'entitat forta amb què es relaciona.
- Cada conjunt de relacions es transforma en una taula els camps de la qual són les claus principals de cada una de les entitats amb què es relaciona més els atributs propis de la relació.
- Tota relació  $N:M$  es transforma en una taula intermèdia, i es relaciona amb les dues anteriors mitjançant relacions de cardinalitat 1 a  $N$ .

## 1.9. Exemple de conversió d'un model entitat-relació a un model relacional

Es vol dissenyar un sistema relacional per a una entitat financera que contingui informació sobre els clients, els contractes dels clients i les operacions fetes sobre cada un d'aquests. Per a això cal considerar les restriccions següents:

- Una operació s'identifica per un número d'operació, data d'operació i import.
- Un client pot tenir molts comptes.
- Un compte pot ser de diversos clients.
- Un compte només pot ser en una sucursal.

Partint de l'anterior identifiquem les entitats següents amb els atributs corresponents del model entitat-relació:

- "Clients" (NIF, nom, cognom)
- "Sucursals" (numsucu, direcció, població, telèfon)
- "Comptes" (numCTE, tipus, interès, saldo)
- "Operacions" (numop, data, import)

Gràficament es representaria de la manera següent:

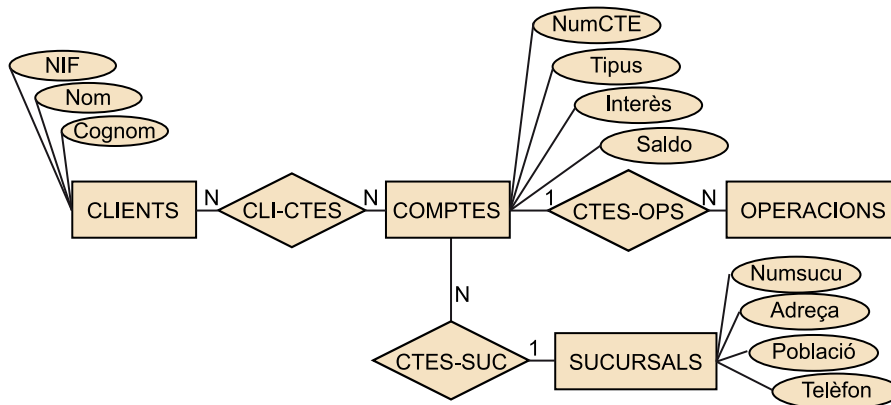


Figura 3. Model entitat-relació de l'entitat financera d'exemple

Aplicant les regles de transformació es converteix en l'estructura de taules següent que hem fet amb l'Access:

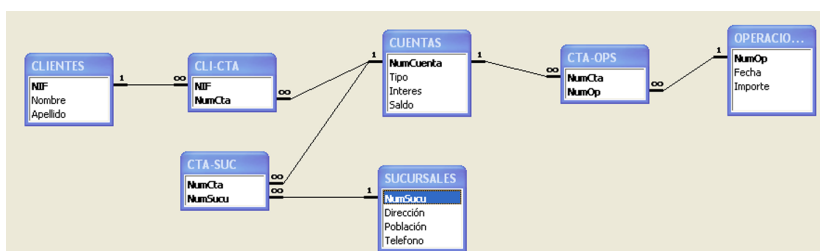


Figura 4. Estructura de taules corresponent al model entitat-relació de la figura 3

Podem veure que les entitats i les relacions s'han transformat en taula; els camps de les primeres són els atributs de les entitats i, en el cas de les relacions, les claus primàries de cada un amb què es relacionen.

### 1.10. Integritat referencial

A l'hora d'implementar en una base de dades el nostre model hem de tenir clar com volem que es comporti davant de modificacions i esborraments de claus amb l'objectiu d'assegurar que es manté la integritat en les relacions de la nostra base de dades.

No es manté la integritat quan en una relació 1:N s'elimina el registre en la taula amb cardinalitat 1, que és clau forana de la taula amb cardinalitat N (quedarien registres en la taula de cardinalitat N que tindrien com a clau forana una clau principal de la taula amb cardinalitat N que no existiria).

Tampoc no es manté la integritat si fem canvis sobre un camp que sigui clau principal de la taula amb cardinalitat 1 i aquests no es veuen reflectits sobre els registres de la taula de cardinalitat N la clau forana de la qual és la clau principal de la taula de cardinalitat 1.

Per això a l'hora de definir les relacions, alguns sistemes gestors de bases de dades com, per exemple, l'Access, permeten establir diferents comportaments del sistema quant a la integritat referencial (si és que volem que la tingui). Amb caràcter general, les dues possibilitats són:

1) **Quan duem a terme un esborrament.** A fi de mantenir la integritat referencial és possible que el sistema gestor de bases de dades dugui a terme la propagació dels esborraments a les taules relacionades. Si esborrem un registre d'una taula que es relaciona 1:N amb una altra, també s'esborraran els registres de la taula amb cardinalitat N en la relació.

Partint del cas que estem abordant (l'entitat financera), si esborrem el client amb NIF 21.212.121-A i exigim integritat referencial, també s'esborraran els registres amb aquest NIF de la taula CLI-CTE.

2) **Quan duem a terme canvis en els camps que són clau d'una taula.** De manera similar al cas anterior dels esborraments, és possible propagar els mateixos canvis als camps clau forana de les taules amb què es relaciona. Si modifiquem un registre d'un atribut clau d'una taula que es relaciona 1:N amb una altra, també s'actualitzaran al mateix valor els camps clau forana de la taula amb cardinalitat N en la relació.

Si ens hem adonat que tenim un NIF d'un client mal gravat i el modifiquem, canviant el 20.200.200-B pel valor 21.212.121-A, tenim l'actualització reflectida tant en la taula "Clients" com en la relació CLI-CTE (els comptes que abans es relacionaven amb el NIF 20.200.200-B ara ho fan amb el 21.212.121-A).

En l'Access trobem les possibilitats d'integritat referencial quan definim les relacions entre taules:

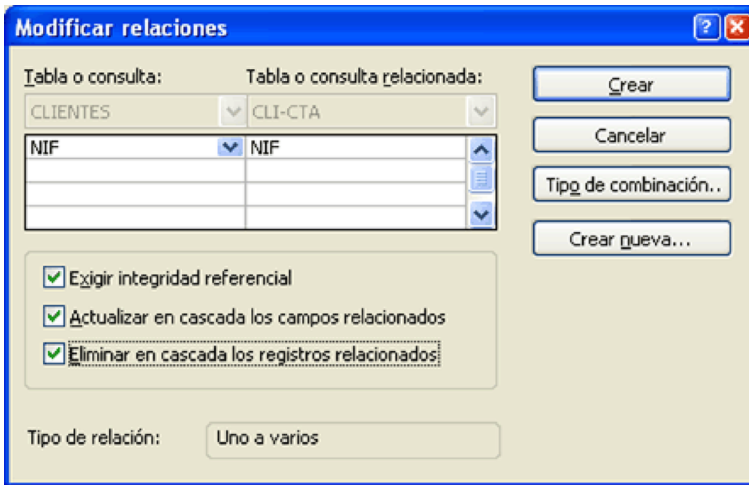


Figura 5. Mostra de l'exigència d'integritat referencial en l'Access

### 1.11. Structured query language

SQL són les inicials de *structured query language* o *llenguatge d'interrogació estructurat*. És un llenguatge no procedimental inventat per IBM en els anys setanta per a implementar el model relacional definit per Codd.

Realment SQL és una abreviatura que s'ha format amb el temps, ja que el nom inicial era SEQUEL (*structured english query language*). Actualment aquest és el llenguatge més àmpliament utilitzat pels sistemes gestors de bases de dades, si bé tots presenten les seves petites adaptacions d'aquest llenguatge, de manera que trobem algunes variacions del llenguatge SQL en funció de l'SGBD que utilitzem, si bé les ordres més habituals són les mateixes en tots.

Encara que el llenguatge SQL es consideri un llenguatge d'interrogació, conté moltes altres capacitats a més de la consulta en bases de dades, incloent-hi característiques per a definir l'estructura de les dades, inserció de dades, modificació de dades en la base de dades, etc.

Sobre la nostra base de dades de comptes d'una entitat bancària, considerarem que tenim carregada la informació següent:

Sobre l'entitat "Sucursals":

SUCURSALES : Tabla				
	NumSucu	Dirección	Población	Telefono
▶ +	1	Av. Pearson, 123	Barcelona	933333333
+	2	Plaza Catalunya,	Barcelona	934444444
+	3	Rambla, 16	Barcelona	935555555
+	4	Plaza Neptuno, 1	Madrid	911111111
+	5	Gran Vía, 11	Madrid	916666666
+	6	Colon, 9	Valencia	963333333
*	0			0

Figura 6. Detall de les dades carregades en la taula "Sucursals".

Sobre l'entitat "Clients":

CLIENTES : Tabla			
	NIF	Nombre	Apellido
▶ +	11111111-A	Juan	García
+	22222222-B	Pedro	Pérez
+	33333333-C	Helena	Martínez
+	44444444-D	Marta	González
+	55555555-E	Daniel	Fernández
+	66666666-F	Pablo	López
*			

Figura 7. Detall de les dades carregades en la taula "Clients".

Sobre l'entitat "Operacions":

OPERACIONES : Tabla			
	NumOp	Fecha	Importe
▶ +	1	10/02/2010	5
+	2	11/02/2010	7
+	3	11/02/2010	6
+	4	12/02/2010	6
+	5	13/02/2010	8
+	6	15/02/2010	9
+	7	16/02/2010	10
+	8	17/02/2010	12
+	9	20/02/2010	4
+	10	21/02/2010	5
+	11	23/02/2010	6
+	12	24/02/2010	11
*	0		0

Figura 8. Detall de les dades carregades en la taula "Operacions".

Sobre l'entitat "Comptes":

CUENTAS : Tabla				
	NumCuenta	Tipo	Interes	Saldo
▶ +	1	Cuenta corriente	1	100
+	2	Préstamo	5	50
+	3	Plazo Fijo	4	125
+	4	Cuenta corriente	1	154
+	5	Préstamo	4	95
+	6	Cuenta Corriente	2	67
*	0		0	0

Figura 9. Detall de les dades carregades en la taula "Comptes".

Sobre l'entitat "CLI-CTE" (derivada de la relació "Clients"- "Comptes"):

CLI-CTA : Tabla		
	NIF	NumCta
▶	11111111-A	1
	11111111-A	2
	22222222-B	3
	22222222-B	5
	33333333-C	1
	33333333-C	4
	44444444-D	1
	55555555-E	5
	66666666-F	6
*		0

Figura 10. Detall de les dades carregades en la taula CLI-CTE.

Sobre l'entitat "CTE-OPS" (derivada de la relació "Comptes"- "Operacions"):

CTA-OPS : Taula		
	NumCta	NumOp
▶	1	1
	1	2
	2	3
	2	4
	3	5
	3	6
	4	7
	5	8
	5	9
	5	10
	6	11
	6	12
*	0	0

Figura 11. Detall de les dades carregades en la taula "CTE-OPS"

Sobre l'entitat "CTE-SUC" (derivada de la relació "Comptes"- "Sucursals"):

CTA-SUC : Taula		
	NumCta	NumSucu
▶	1	1
	2	2
	3	3
	4	4
	5	5
	6	6
*	0	0

Figura 12. Detall de les dades carregades en la taula "CTE-SUC"

### 1.12. Query by example

Habitualment, els sistemes gestors de bases de dades permeten generar consultes utilitzant exemples (QBE, *query by example*), la qual cosa permet fer consultes a la base de dades d'una manera més intuïtiva i gràfica. Concretament l'Access també té un QBE que resulta relativament senzill de manejar.

Mitjançant el QBE evitem conèixer la sintaxi concreta SQL que utilitza el nostre sistema gestor de bases de dades, la qual cosa ens permet explotar la base de dades d'una manera més senzilla.

Per a accedir al QBE de l'Access hem de seleccionar l'entrada Consultes, del menú vertical de l'esquerra i, a continuació, seleccionar Crear una nova consulta en vista Disseny.

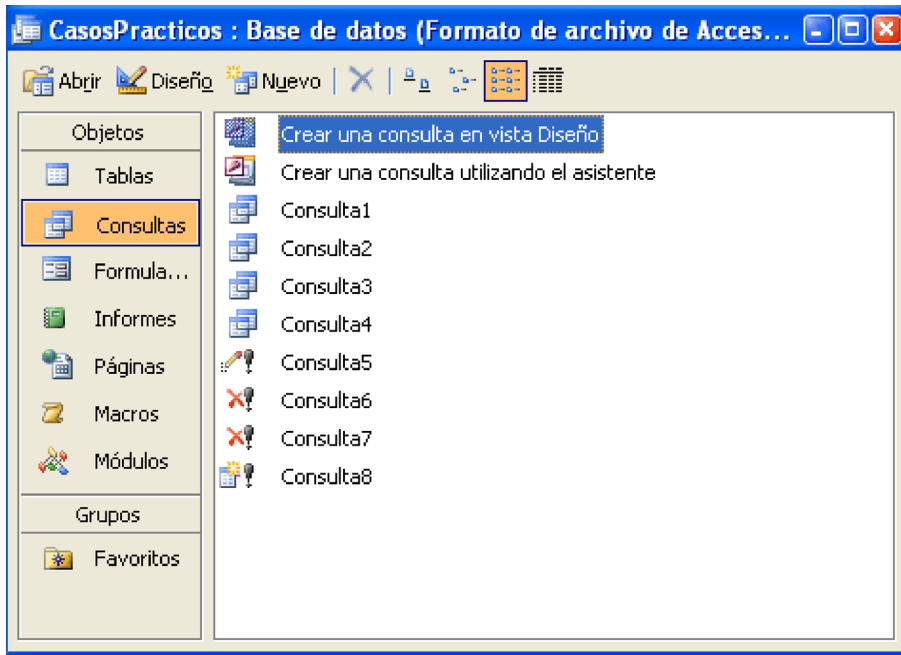


Figura 13. Detall de l'accés al QBE de l'Access

El que ens apareix a continuació és una finestra emergent des de la qual podem seleccionar la taula o taules que utilitzarem per a fer les nostres consultes, i una reixeta (*grid*) sobre la qual podrem arrossegat i deixar anar els camps que vulguem de les taules que hàgim seleccionat.

D'aquesta manera, amb accions molt simples podrem construir sentències SQL, que en alguns casos són sensiblement complexes. Addicionalment podem visualitzar la sentència en format SQL.

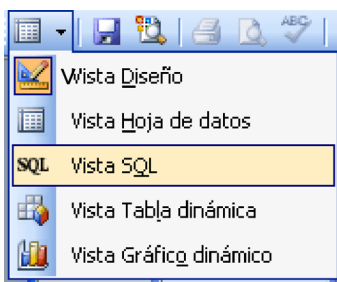


Figura 14. Accés a la redacció i visualització de sentències SQL en l'Access

Per a simplificar la representació dels camps s'utilitza, igual com en l'SQL, el símbol \* per a significar que volem tenir en compte tots els atributs de la taula des de la qual arrosseguem i deixem anar.

La consulta generada es pot desar i, el que és molt útil, es pot utilitzar posteriorment com a punt de partida per a altres consultes com si es tractés d'una taula. El contingut d'aquesta "taula" es calcula en el moment de llançar la consulta que la utilitza, és a dir, treballarem amb dades actualitzades i no estàtiques.



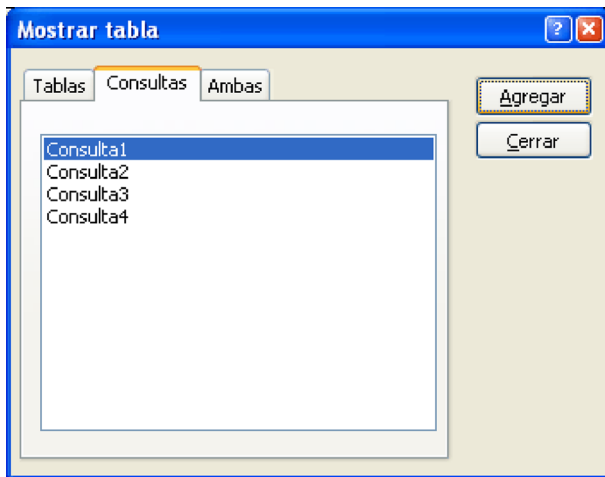


Figura 15. Mostra de la finestra Access que ens permet utilitzar les consultes anteriors com si fossin taules.

## 2. Cas pràctic

Per al desenvolupament d'aquest cas pràctic partirem de la base de dades de Clients, Comptes, Operacions i Sucursals que hem utilitzat com a exemple en aquest document.

Plantejarem una sèrie de consultes sobre aquesta que haurem de resoldre mitjançant l'SQL i l'editor QBE de l'Access.

### 2.1. Consulta 1: obtenir els noms dels nostres clients ordenats alfabèticament

La sentència SQL que haurem d'executar és la següent:

```
SELECT "Clients".Nom  
FROM "Clients"  
ORDER BY "Clients".Nom
```

Fer-ho des de l'editor QBE és, igualment, molt senzill:

- 1) En la finestra "Mostrar Taules" hem de seleccionar la taula sobre la qual volem fer la consulta, en aquest cas, "Clients". A continuació tanquem aquesta finestra.
- 2) Arrosseguem el camp "Nom" de la taula "Clients" que apareix en la part superior de la finestra i el deixem anar en la primera columna del *grid* (la reixeta que apareix en la part inferior de la finestra).
- 3) Fins aquí ens hauria seleccionat tots els clients, però, com que volem que estigui ordenat alfabèticament, hem de seleccionar en la fila "Ordre" del *grid* Ascendent.

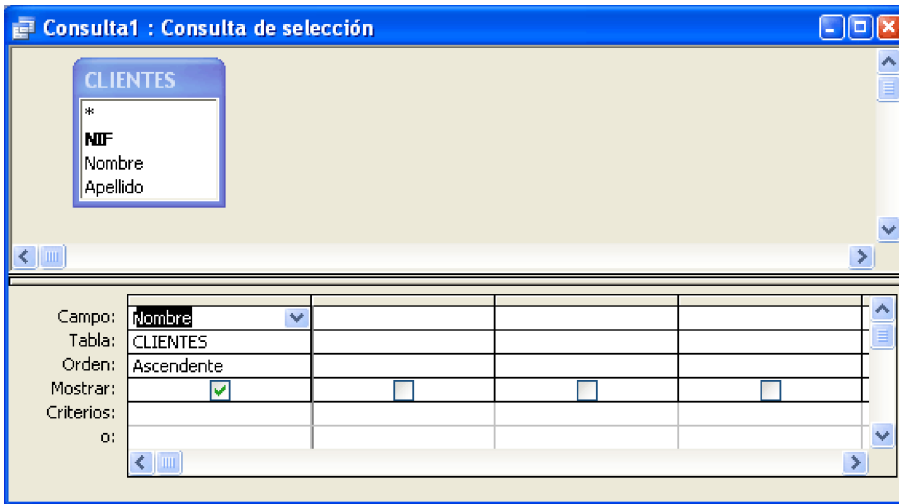


Figura 16. Finestra de l'editor QBE d'Access que representa la Consulta 1.

## 2.2. Consulta 2: obtenir la informació de comptes amb NIF 11111111-A

Per a obtenir la informació de comptes amb un NIF determinat el que hem d'aplicar és l'operació de consulta SQL següent:

```
SELECT "Comptes".*, "Clients".NIF
FROM "Comptes" INNER JOIN ("Clients" INNER JOIN [CLI-CTE] ON
"Clients".NIF = [CLI-CTE].NIF) ON "Comptes".NumCompte = [CLI-
CTE].NumCTE
WHERE ((("Clients".NIF)="11111111-A"));
```

La funció de combinació interna INNER JOIN selecciona les files que tenen valors idèntics en els camps sobre els quals compara. En aquest cas hem de fer la consulta sobre tres taules ("Clients", "CLI-CTE" i "Comptes"), de manera que hem d'imbricar dos INNER JOIN per a poder obtenir l'equivalent a una taula a partir de la combinació de les tres inicials. La sintaxi general d'implantació de sentències INNER JOIN és:

```
FROM (...(taula1 JOIN taula2 ON condicio1) JOIN taula3 ON condicio3)
JOIN...)
```

Fer-ho des de l'editor QBE també és molt senzill, i en aquest cas té l'avantatge que ens podem despreocupar de la sintaxi de les INNER JOIN imbricades que necessitem. Els passos a seguir són:

1) Des de la finestra Mostrar Taules hem de seleccionar les taules que necessitem per a aquesta consulta i que són "Clients", "Comptes" i "CLI-CTE". A continuació tanquem la finestra.

2) Des de la taula "Comptes" n'arrosseguem tots els camps, és a dir, arrosseguem el \*, i el deixem anar en la primera columna del *grid* inferior.

### Vegeu també

La funció de combinació interna INNER JOIN es descriu en el mòdul "El model relacional i l'àlgebra relacional".

3) Des de la taula "Clients" arrosseguem el camp NIF a la segona columna del *grid*, i traiem la marca "Mostrar", ja que només volem presentar la informació dels comptes.

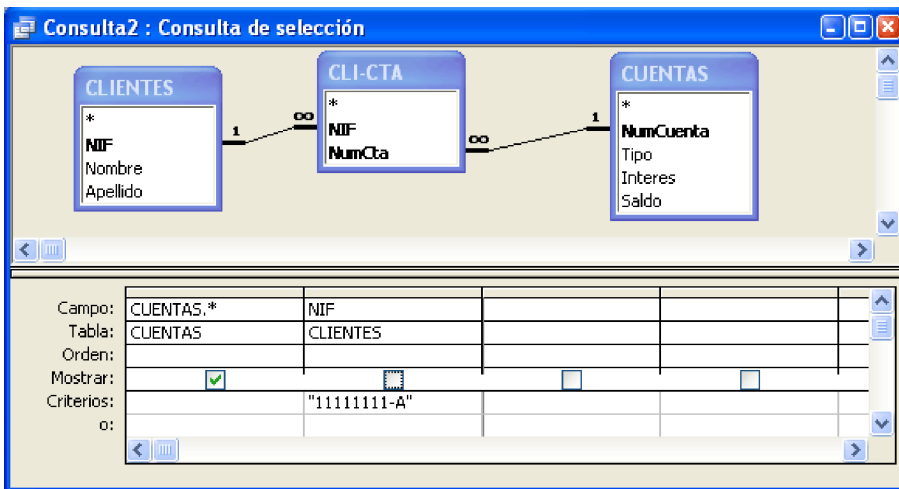


Figura 17. Finestra de l'editor QBE de l'Access que representa la Consulta 2.

Tenim una manera més senzilla de fer aquesta consulta? La resposta és sí. La podem fer simplement amb dues taules en comptes de tres, utilitzant exclusivament les taules "Comptes" i "CLI-CTE".

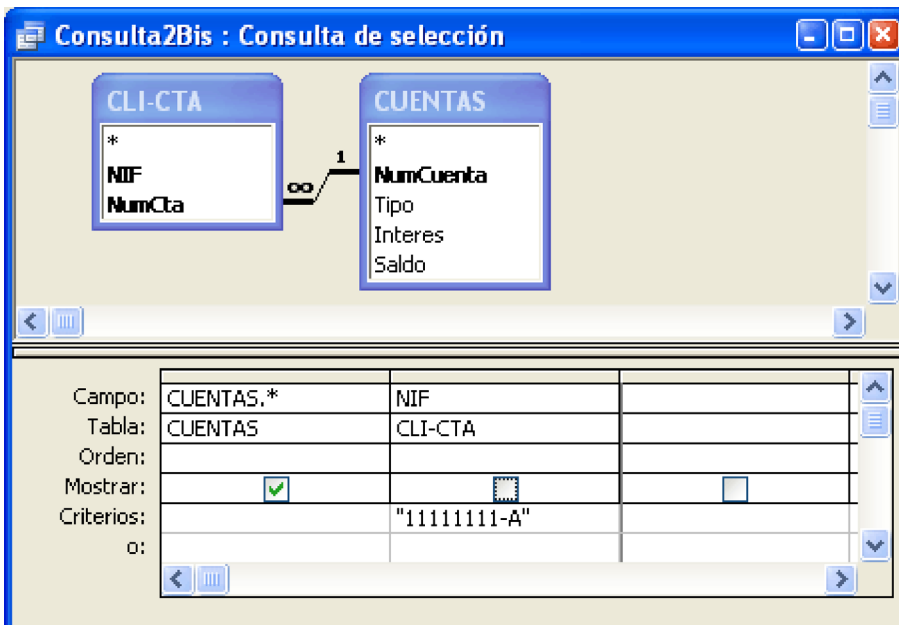


Figura 18. Finestra de l'editor QBE de l'Access que representa la Consulta 2 amb només dues taules.

### 2.3. Consulta 3: calcular l'import d'interessos dels clients

Ens demanen que calculem l'import en concepte d'interessos de cada client de l'entitat bancària del nostre exemple tenint en compte el següent:

- La fórmula del càlcul d'interessos per a tots els productes de l'entitat és  $\text{Saldo} * \text{Interès}$ .

- No apliquem tractaments diferenciats perquè hi hagi més d'un client en un mateix compte, és a dir, els interessos es multipliquen en funció del nombre de clients associats al compte.

El resultat és el següent:

```
SELECT "Clients".Nombre, Sum(([Interes]*[Saldo])) AS subtotal
FROM "Comptes" INNER JOIN ("Clients" INNER JOIN [CLI-CTE] ON
"Clients".NIF = [CLI-CTE].NIF) ON "Comptes".NumCompte = [CLI-
CTE].NumCTE
GROUP BY "Clients".Nombre;
```

Fixeu-vos que hem donat un àlies al camp calculat i l'hem anomenat *subtotal*.

Com sempre, fer-ho des del *grid* és intuïtiu, encara que en aquest cas també cal escriure una mica, no és solament arrossegar i deixar anar. Els passos a seguir són:

1) Des de la finestra Mostrar Taules seleccionem les taules "Clients", "CLI-CTE" i "Comptes". Tanquem la finestra Mostrar Taules.

2) Des de la taula "Clients" arrosseguem el camp Nom i el deixem anar en la primera columna del *grid*.

3) En la segona columna del *grid* hem d'escriure l'àlies que volem donar al resultat del càlcul seguit de dos punts i la fórmula que volem aplicar, en aquest cas Interès multiplicat per Saldo. Atès que només hi ha camps que s'anomenin igual en la taula "Comptes", podem evitar escriure la taula de la qual provenen, altrament les hauríem d'haver referenciat precedint els camps del nom de la taula i un punt. Us convido a fer la prova amb aquest exemple, de manera que us sigui d'utilitat quan hàgiu de tractar amb diferents taules que continguin alguns camps amb nom idèntic.

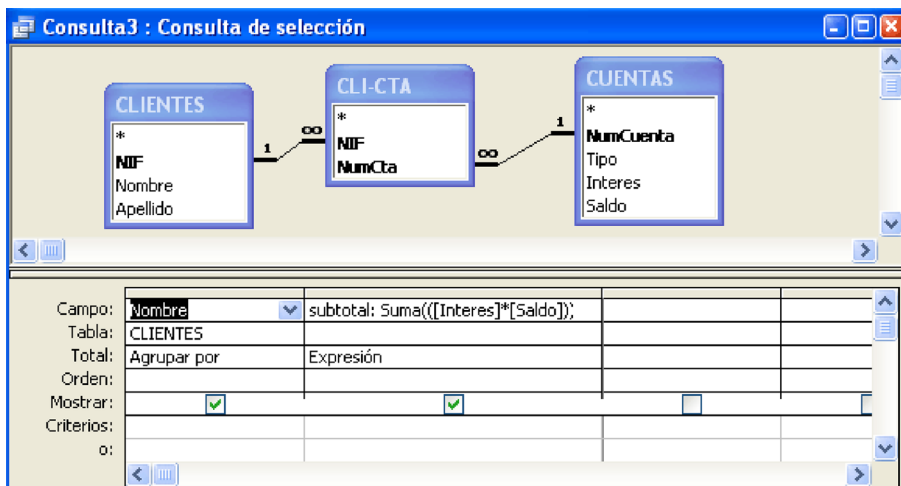


Figura 19. Finestra de l'editor QBE de l'Access que representa la Consulta 3.

Per a què hem volgut donar un àlies al resultat del càlcul? Recordeu que hem dit que una utilitat força significativa de l'Access és que les consultes es poden utilitzar com si fossin taules (per exemple, per a generar altres consultes a partir d'aquestes). En aquest cas, si posteriorment necessitem fer alguna cosa més amb aquest càlcul ho podem fer fàcilment, ja que en tenir l'àlies, és com si es tractés d'un camp qualsevol.

#### **2.4. Consulta 4: calcular la suma del total dels saldos dels comptes**

El resultat de la sentència SQL que haurem d'executar és el següent:

```
SELECT SUM ("Comptes".Saldo)
FROM "Comptes";
```

L'ús del QBE per a aquest cas també resulta molt senzill. Vegem els passos que hem de fer:

- 1) Com sempre, des de la finestra *Mostrar Taules* seleccionem les taules que necessitem per a la nostra consulta, en aquest cas "Comptes".
- 2) Des de la barra d'eines punxem sobre el símbol  $\Sigma$ , la qual cosa fa que en el *grid* aparegui una nova fila anomenada "Total". Veurem que el símbol  $\Sigma$  queda amb fons de diferent color, la qual cosa vol dir que l'opció està activada.
- 3) Arrosseguem el camp Saldo des de la taula "Comptes" a la primera columna del *grid*.
- 4) Per defecte, en la fila "Total" ens apareix l'opció *Agrupar per*, i si despleguem el menú veurem que hi ha diverses opcions. Tal com imaginàvem, hem de seleccionar *Suma*.

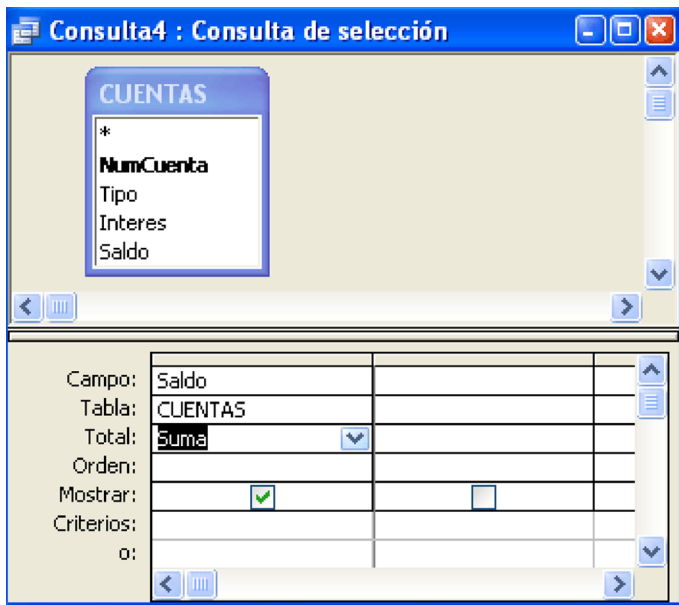


Figura 20. Finestra de l'editor QBE de l'Access que representa la Consulta 4.

És senzill reutilitzar aquest càlcul en altres consultes? La veritat és que sí, perquè podem reutilitzar la consulta com si fos una taula més, si bé és cert que, a diferència del cas anterior, no hem donat un àlies a aquest càlcul.

El *grid* també ens permet fer aquest càlcul donant-li un àlies. Simplement hem de desactivar el símbol  $\Sigma$  i escriure el càlcul en la primera fila del *grid*. En aquest cas el càlcul és Suma (Saldo).

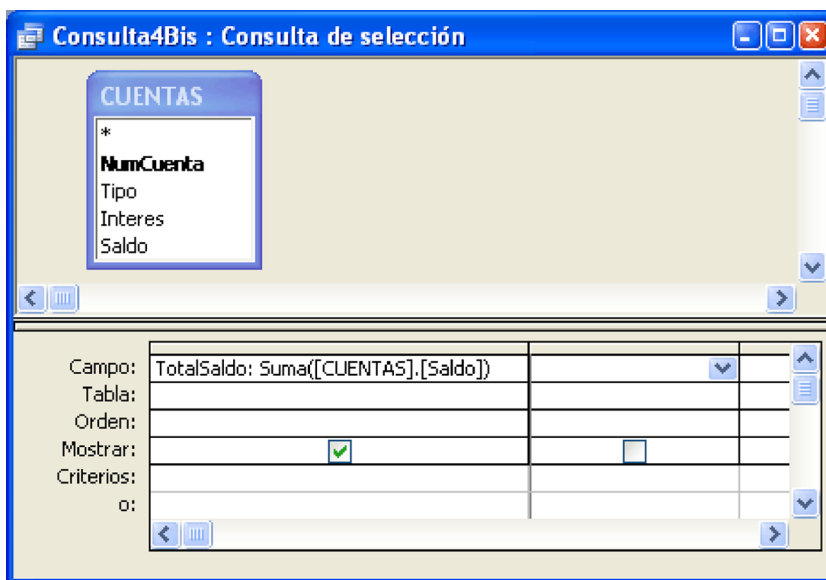


Figura 21. Finestra de l'editor QBE de l'Access que representa la Consulta 4 sense utilitzar un àlies.

Perquè serveixi de guia a l'hora de referenciar a camps de les taules, per a aquest exemple s'ha utilitzat la ruta completa del camp, és a dir "[Comptes].[Saldo]", encara que simplement amb "Saldo" hauria estat suficient, ja que no hi ha cap altre camp que es digui de la mateixa manera i menys en aquest cas, en què únicament utilitzem una taula per a la consulta.

## 2.5. Consulta 5: incrementar en un punt el tipus d'interès als clients que es diguin Joan

Per a això hem de desenvolupar una consulta que a tots registres de la taula "Comptes" sumi al valor del seu camp "Interes" una unitat.

La consulta en SQL seria:

```
UPDATE "Comptes" INNER JOIN ("Clients" INNER JOIN [CLI-CTE]
ON "Clients".NIF = [CLI-CTE].NIF) ON "Comptes".NumCompte = [CLI-
CTE].NumCTE
SET "Comptes".Interes = [Interes]+1
WHERE ((("Clients".Nombre)="Joan"));
```

Com ja sabeu, som davant d'una consulta d'actualització.

En l'Access, per defecte les sentències SQL són de selecció, és a dir, no alteren el contingut de la base de dades. Per a dur a terme un altre tipus de consultes (actualització, inserció, esborrament, etc.) haurem d'anar al menú de consultes.

### Vegeu també

En el mòdul 2 estudiem les consultes d'actualització.

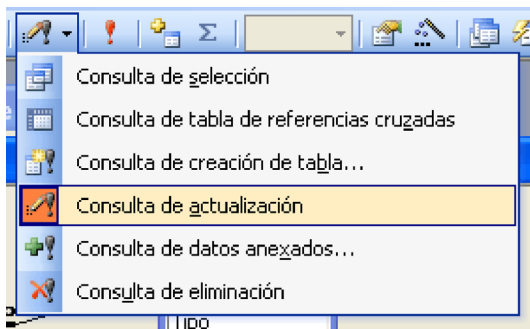


Figura 22. Menú en l'Access per a la selecció del tipus de consulta, seleccionant Consulta d'actualització

Quan fem aquest canvi, observem certes variacions sobre les dades que ens sol·licita el QBE de l'Access en funció del tipus de consulta de què es tracti.

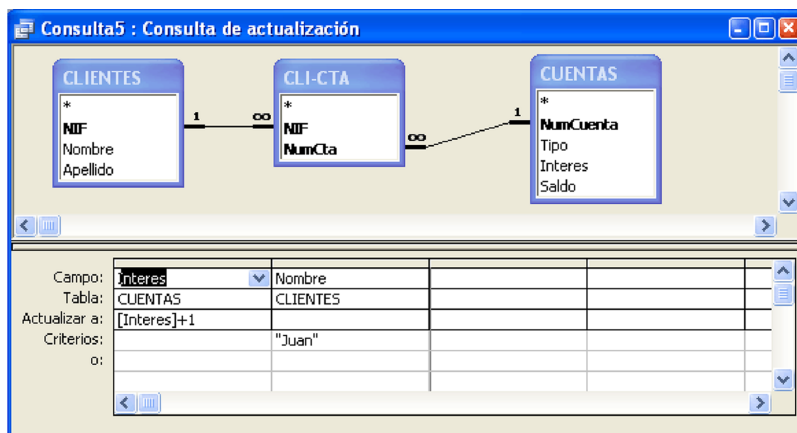


Figura 23. Finestra de l'editor QBE de l'Access que representa la Consulta 5.



## 2.6. Consulta 6: eliminar el Joan com a client de l'entitat bancària

El client de nom Joan ha decidit deixar de ser client de l'entitat financera, per la qual cosa l'hem d'eliminar de la taula "Clients".

La sentència SQL que ens permetrà dur a terme aquesta acció és:

```
DELETE "Clients".Nom
FROM "Clients"
WHERE (("Clients".Nom)="Joan");
```

Fixem-nos que l'especificació dels camps a eliminar a continuació de la paraula *DELETE* és opcional, ja que sempre s'esborren registres complets i mai, camps aïllats. Per tant, haurien tingut el mateix resultat les sentències SQL següents:

```
DELETE "Clients".NIF
FROM "Clients"
WHERE (("Clients".Nom)="Joan");
```

O bé:

```
DELETE "Clients".Cognom
FROM "Clients"
WHERE (("Clients".Nom)="Joan");
```

Què ha passat en la taula "CLI-CTE"? Fixeu-vos que també s'han esborrat les ocurrències que hi hagués relatives al client Joan, és a dir, en la taula "CLI-CTE" ja no apareixen registres que tinguin "11111111-A" com a valor del camp NIF (que és el NIF del Joan). Els continguts de les taules "Clients" i "CLI-CTE" després de l'execució de la sentència SQL són, respectivament:

CLIENTES : Tabla			
	NIF	Nombre	Apellido
▶ +	22222222-B	Pedro	Pérez
+	33333333-C	Helena	Martínez
+	44444444-D	Marta	González
+	55555555-E	Daniel	Fernández
+	66666666-F	Pablo	López
*			

Figura 24. Contingut de la taula "Clients" després de l'execució de la Consulta 6

CLI-CTA : Taula		
	NIF	NumCta
▶	22222222-B	3
	22222222-B	5
	33333333-C	1
	33333333-C	4
	44444444-D	1
	55555555-E	5
	66666666-F	6
*		0

Figura 25. Contingut de la taula "CLI-CTE" després de l'execució de la Consulta 6

Cal anar especialment amb compte amb aquest tipus de sentències que actualitzen la base de dades Access, ja que un cop executada ja no és possible recuperar la situació immediatament anterior a la seva execució (no existeix una opció de desfer).

D'altra banda, també és necessari que entenguem el disseny que hem fet de la base de dades respecte a l'exigència d'integritat referencial, ja que això implicarà, com hem vist, l'actualització o esborrament de registres addicionals als de la taula sobre la qual els estem fent directament. Aquest punt és especialment important si tenim en compte que l'Access no ens advertirà d'aquestes implicacions, és a dir, no ens avisarà dels esborraments o canvis que farà sobre altres taules de la base de dades.

Concretament per a aquesta consulta que hem executat, els dos missatges d'avís que apareixen abans de l'execució de la sentència són:

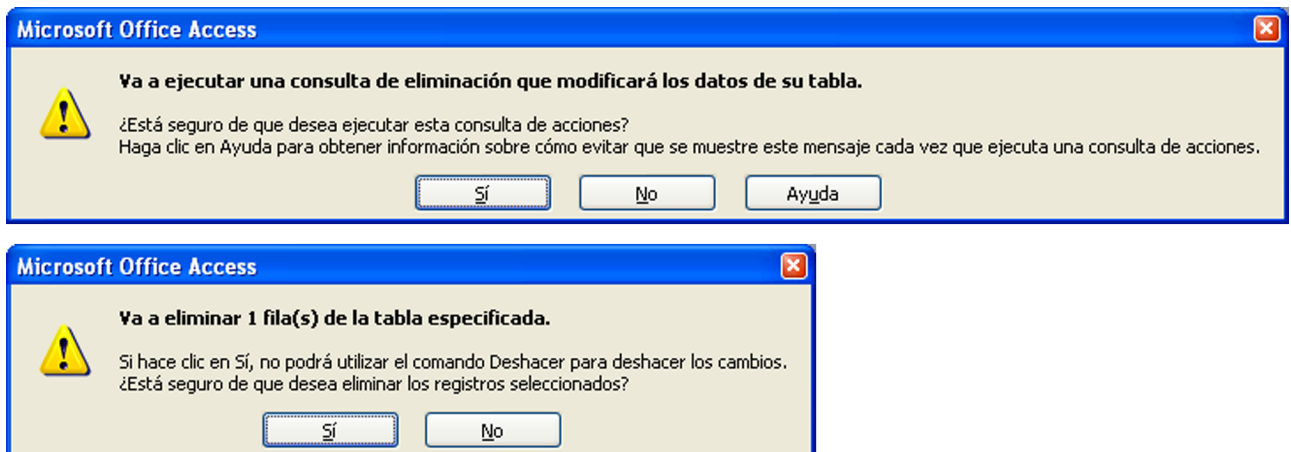


Figura 26. Finestres d'avís d'actualització de taules de l'Access

Com podeu veure només ens adverteix de l'esborrament d'una fila (registre) d'una taula, quan en realitat eliminarem un registre de la taula "Clients" i dos de la taula "CLI-CTE" (aquests últims per haver exigint integritat referencial en la relació "Clients" i "CLI-CTE").

## 2.7. Consulta 7: duplicar la taula "Clients" sobre una de nova anomenada "CLIENTS2"

Si no hi ha una taula amb el nom de la taula de destinació de la consulta, l'Access la crea, la qual cosa ens permet fer ús de la funció SELECT INTO directament sense necessitat d'haver creat prèviament la taula, és a dir, sense utilitzar CREATE TABLE. Això resulta molt còmode quan les taules tenen molts camps, ja que ens estalviem haver d'especificar-los.

Per tant, la sentència que necessitem és:

```
SELECT "Clients".* INTO CLIENTS2  
FROM "Clients";
```

Igual com per a la resta de sentències d'actualització, l'Access ens adverteix de la impossibilitat de desfer els canvis.

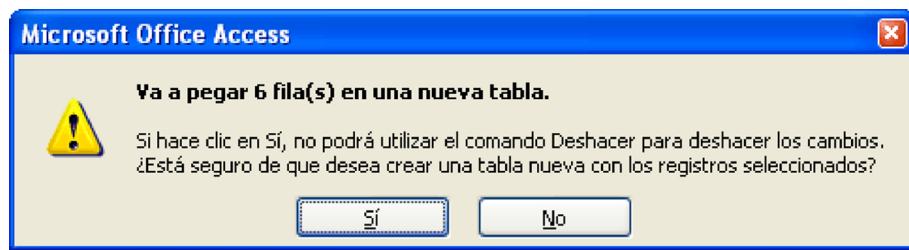


Figura 27. Finestres de creació d'una nova taula amb incorporació de dades en l'Access

Utilitzant la interfície QBE de l'Access hauríem de seguir aquests passos:

- 1) Escollir l'opció Nova consulta, Vista disseny i seleccionar la taula "Clients" en la finestra Mostrar Taula.
- 2) Seleccionar Consulta de creació de taula..., dels tipus de consulta.

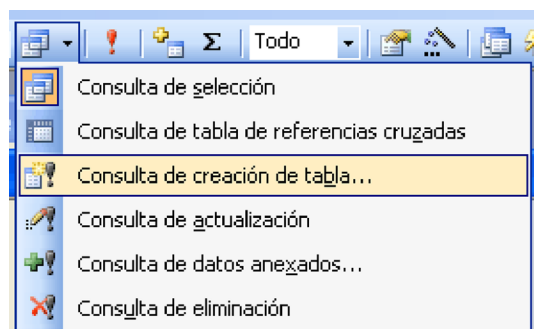


Figura 28. Menú per a la selecció d'una consulta de creació de taula en l'Access

3) Indicar la taula nova que volem crear. En aquest punt cal tenir en compte que l'Access ens presenta la possibilitat d'incorporar els registres sobre taules ja existents, que fins i tot poden ser d'altres bases de dades (us animo a provar aquestes possibilitats, que són molt interessants per a fer còpies de taules sobre altres bases de dades).

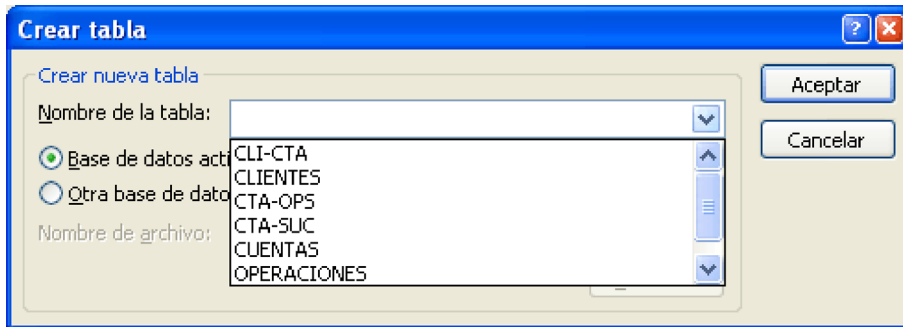


Figura 29. Finestra de selecció del nom de la nova taula a crear

4) A continuació hem d'indicar a la reixeta (*grid*) els camps des dels quals es nodrirà la informació sobre la nova taula "Clients2", és a dir, des de "Clients", per a la qual cosa haurem d'indicar que són tots els camps de "Clients", el que podem fer, com ja sabem, mitjançant el \*.

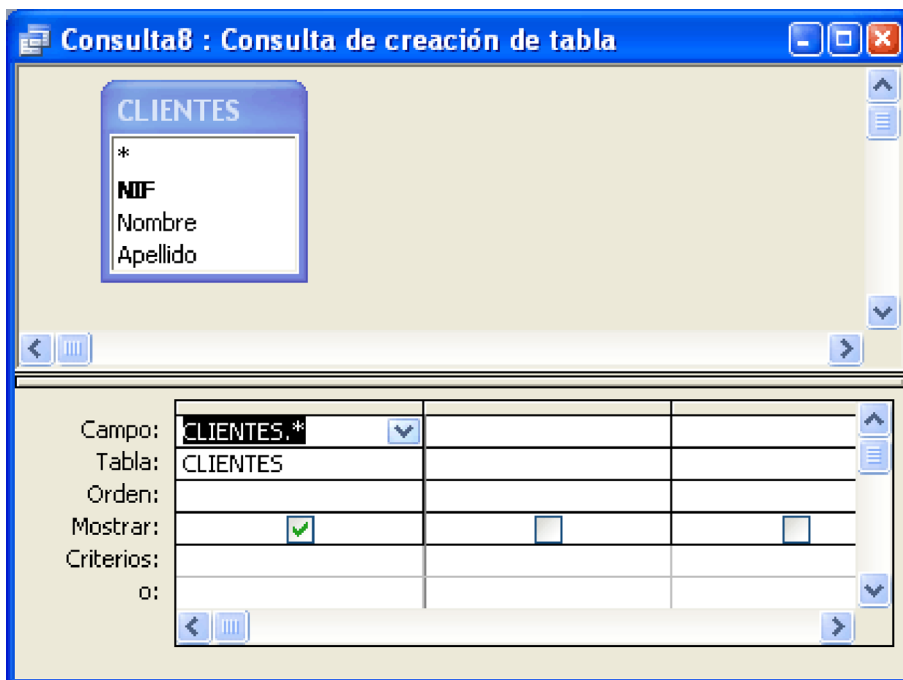


Figura 30. Finestra de l'editor QBE de l'Access que representa la Consulta 7.

## 2.8. Consulta 8: incorporar un nou client a la taula CLIENTS2

També és possible que vulguem incorporar nous registres sobre taules. En aquest cas se'ns demana que incorporem a la taula "CLIENTS2" (que acabem de crear al pas anterior) un nou client anomenat Mario García i amb NIF 77777777-G.

Utilitzant l'SQL, ho fariem de la manera següent:

```
INSERT INTO Clients2  
VALUES ("77777777-G", "Mario", "García");
```

En aquest cas, l'editor QBE de l'Access no permet fer ús d'aquesta funcionalitat, per la qual cosa si intentem mostrar-ho, ens apareixerà l'avís:

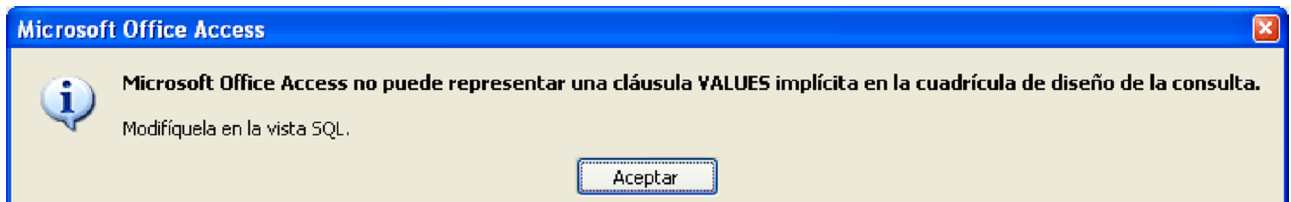


Figura 31. Avís d'error de l'Access per ús de sentència VALUES en editor QBE



## Bibliografia

**Date, C. J.** (2001). *Introducción a los sistemas de bases de datos* (7a. ed.). Prentice Hall.

**Silberschatz, A.; Korth, H. F.; Sudarshan, S.** (1998). *Fundamentos de bases de datos* (3a. ed.). McGraw-Hill.

**Celma, M.; Casamayor, J. C.; Mota, L.** (2003). *Bases de datos relacionales*. Pearson / Prentice Hall.

