

# Cas pràctic sobre bases de dades relacionals

Pedro Cardona Vilaplana

PID\_00201447



# Índex

<b>Introducció</b> .....	5
<b>1. Conceptes necessaris</b> .....	7
1.1. El model entitat-interrelació .....	7
1.2. Entitats i atributs .....	7
1.3. Interrelacions .....	8
1.4. Claus .....	8
1.5. Classificació de les entitats en funció de les claus .....	9
1.6. Diagrames entitat-interrelació .....	9
1.7. Cardinalitat de les interrelacions .....	10
1.8. Generació del model relacional a partir del model entitat-interrelació .....	11
1.9. Exemple de conversió d'un model entitat-interrelació a un model relacional .....	12
1.10. Integritat referencial .....	14
1.11. Llenguatge SQL ( <i>Structured query language</i> ) .....	15
1.12. Contingut de la base de dades de l'exemple .....	15
1.13. Mètodes de consultes QBE ( <i>Query by example</i> ) .....	17
<b>2. Cas pràctic</b> .....	20
2.1. Consulta 1: obtenir els noms dels clients ordenats alfabèticament .....	20
2.2. Consulta 2: obtenir la informació dels comptes amb NIF 11111111-A .....	21
2.3. Consulta 3: calcular l'import dels interessos dels clients .....	22
2.4. Consulta 4: calcular la suma del total dels saldos dels comptes .....	24
2.5. Consulta 5: incrementar en un punt el tipus d'interès als clients que es diguin Joan .....	26
2.6. Consulta 6: eliminar el Joan com a client de l'entitat bancària .....	27
2.7. Consulta 7: duplicar la taula CLIENT sobre una de nova anomenada CLIENT2 .....	29
2.8. Consulta 8: incorporar un nou client a la taula CLIENT2 .....	31
<b>Bibliografia</b> .....	33



## Introducció

Les bases de dades són eines molt potents que permeten gestionar grans quantitats de dades de manera ordenada, i convertir-les en informació mitjançant el seu tractament, i en coneixement mitjançant la difusió d'aquesta. En aquest document posarem en pràctica conceptes tractats en els tres primers mòduls didàctics de l'assignatura de Bases de dades: "Sistemes de base de dades", "El model relacional i l'àlgebra relacional" i "El llenguatge SQL". Especialment 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 conceptes clau estudiats en el mòdul "El model relacional i l'àlgebra relacional" i, principalment, amb exercicis per a posar en pràctica d'aquests conceptes mitjançant una base de dades d'exemple que nosaltres mateixos construïrem.

Atès que volem posar en pràctica la teoria apresada és del tot recomanable que feu els exercicis proposats aquí creant una base de dades en el vostre propi ordinador.

Per a fer els exercicis d'exemple i implementar la base de dades del cas pràctic, utilitzarem Microsoft Access, un sistema gestor de bases de dades relacional d'àmplia difusió. De tota manera, com que hem utilitzat exemples i sentències SQL estàndard, podreu fer fàcilment els diferents exercicis sobre qualsevol altra sistema gestor de bases de dades relacional com, per exemple, l'OpenOffice. Els resultats que obtindreu seran els mateixos.



# 1. Conceptes necessaris

## 1.1. El model entitat-interrelació

El model entitat-interrelació (E/R, de l'anglès *entity-relationship*), proposat per Peter Chen l'any 1976, també anomenat *model conceptual de dades*, és una tècnica de representació de les relacions que tenen les dades i que permet recrear la realitat que volem modelitzar en una 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 (utilitzant símbols gràfics) del disseny de les dades, i no del que es vol fer amb elles.

## 1.2. Entitats i atributs

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

Una entitat (o tipus d'entitat) està formada per un conjunt d'ocurrències d'entitat del mateix tipus.

### Exemple d'entitat

En una empresa, l'entitat formada pel conjunt d'empleats es podria anomenar EMPLEAT.

### Entitat tipus i ocurrència

- Un tipus d'entitat (o entitat tipus) és una categoria generalitzada que defineix un conjunt d'entitats més específiques que tenen els mateixos atributs. Normalment, s'abreuja i s'usa únicament el terme *entitat*.
- Una ocurrència d'entitat és un exemplar d'un tipus d'entitat que comparteix atributs amb altres ocurrències, cada una de les quals té el seu propi valor per a cada atribut. També es fa servir el terme *instància*.

### Exemples d'ocurrència d'entitat

Joan i Pere són ocurrències del tipus d'entitat PERSONA.  
Gener i febrer són ocurrències del tipus d'entitat MES.

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

### Vegeu també

El concepte de *model de dades* es descriu en l'apartat "Model de dades" del mòdul "Sistemes de base de dades".

### Vegeu també

Trobareu definicions i exemples d'ocurrències i tipus d'entitats en els apartats "Ocurrència i tipus" i "Classificació (i instanciació)" del mòdul "Sistemes de base de dades".

### Exemple d'atributs d'una entitat

En l'entitat EMPLEAT, alguns atributs possibles serien els següents: *DNI, nom, cognoms, data de naixement*, etc.

## 1.3. Interrelacions

Una interrelació és una associació<sup>1</sup> que es dona entre diferents entitats.

### Exemple d'interrelació

En una empresa, a més de l'entitat EMPLEAT, també en tindrem una anomenada DEPARTAMENT, que és la divisió organitzativa o funcional de l'empresa a què està adscrit l'empleat. Sobre aquestes entitats podríem definir una interrelació que associés l'empleat Pere amb el departament de Comptabilitat. (Aquesta interrelació es pot anomenar EMP-DEP o PERTANY.)

### El terme *relació*

El terme *relació* es refereix a una taula en el model relacional, que representa una entitat o una interrelació en el model conceptual E/R. L'usen alguns autors per a referir-se de forma poc ortodoxa al concepte d'*interrelació* (en anglès, *relationship*) entre entitats i, alguns SGBD per a fer referència al vincle entre taules.

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

### Exemple d'atributs d'una interrelació

En el cas anterior, un atribut de la interrelació del Pere amb el departament de Comptabilitat podria ser la *data d'adscripció* a aquest departament.

## 1.4. Claus

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

### Exemples de clau

En l'entitat EMPLEAT, el camp *NIF* seria la clau, ja que no existeix cap conjunt més petit que aquest per a identificar de manera unívoca cada un dels empleats.

Assumint que un país no té ciutats amb el mateix nom, el conjunt d'atributs de l'entitat CIUTAT format per *país* i *nom de ciutat* seria la clau, ja que cap conjunt més petit pot identificar de manera unívoca cada una de les ciutats del món. Per exemple, el subconjunt format únicament per l'atribut *nom de ciutat* no pot ser clau, ja que arreu del món hi ha ciutats que comparteixen el nom. És el cas de Barcelona, Guadalajara, Santiago, Sydney, etc.

### Conveni de nomenclatura, ortografia i format

El nom dels objectes de la base de dades s'escriu amb caràcters alfanumèrics, habitualment sense accents. Pot incloure determinats caràcters especials (com guionet "-" o guió baix "\_"). Diccionari que conté paraules buides de significat, que tenen únicament valor gramatical (articles, preposicions, pronoms, etc.). En anglès s'anomena *stoplist* o *stopword list*. Denotarem les entitats i les interrelacions en lletra majúscula i en singular; i els atributs en lletra minúscula i cursiva.

<sup>(1)</sup>El concepte d'*associació* en el model entitat-interrelació fa referència a la interrelació entre entitats, i es descriu en l'apartat "Associació (i dissociació)" del mòdul "Sistemes de base de dades".

### El nom de les interrelacions

El nom de les interrelacions es pot formar separant amb un guionet l'abreviació (tres lletres) de les entitats associades. En algunes ocasions, es pot usar el verb (en infinitiu o en tercera persona del singular) que descriu l'associació entre les entitats associades.



En una entitat és possible que 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 (o instància) s'anomena **clau primària**.

La **clau forana** és el conjunt d'atributs d'una entitat que al seu torn és clau primària d'una altra entitat amb què està interrelacionada.

#### Especificitat terminològica de l'Access

- La clau primària s'anomena *clau principal*.
- La interrelació, associació o vincle entre taules s'anomena *relació*. Insistim a no confondre aquest terme amb el concepte de *relació* (taula) en el model relacional.

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

- **Entitats fortes:** són les que tenen una clau primària. Tenen existència per si mateixes.
- **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 cada un dels seus atributs mitjançant una interrelació. Les seves ocurrences són identificables només per estar associades a una altra entitat (entitat forta). La seva existència depèn de l'existència d'una altra entitat.

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




#### Exemple d'entitat forta, entitat feble i discriminant

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

### 1.6. Diagrames entitat-interrelació

El diagrama entitat-interrelació (E/R) permet representar gràficament l'estructura lògica d'una base de dades mitjançant els elements següents:

- Rectangles: representen les entitats.
- Rombes: representen les interrelacions entre les entitats.
- El·lipsis: representen els atributs de les entitats i de les interrelacions.
- Línies: enllacen atributs a entitats, atributs a interrelacions, i entitats a interrelacions. Mai no enllacen entitats amb entitats.

Concepte relacional	Representació gràfica
entitat	
interrelació	
atribut	

### Exemple de diagrama E/R

Si volem modelitzar les notes de les assignatures d'una titulació que han obtingut els alumnes, definirem les entitats següents:

- **ALUMNE**: conté les dades dels alumnes, dels quals coneixem *DNI*, *nom* i *cognoms*.
- **ASSIGNATURA**: conté les dades de les assignatures, de les quals sabem l'*identificador d'assignatura*, *nom de l'assignatura* i *curs* a què pertany.

El model entitat-interrelació corresponent es representaria amb el diagrama E/R següent:

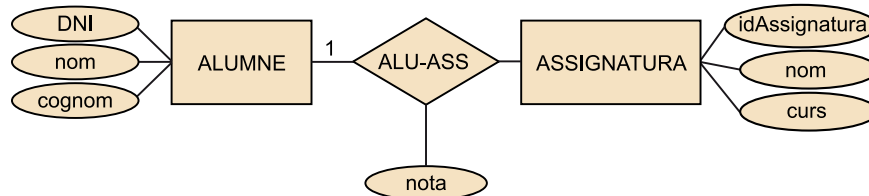


Figura 1. Diagrama E/R que representa la interrelació ALU-ASS entre les entitats ALUMNE i ASSIGNATURA.

### 1.7. Cardinalitat de les interrelacions

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 interrelació. Dit d'una altra manera, expressa el nombre d'entitats amb què pot associar-se una entitat.

D'acord amb l'anterior, entre dues entitats A i B es poden establir les correspondències següents:

- **Una a una (1:1)**: cada instància o ocurrència de l'entitat A s'associa, 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 s'associa amb un nombre qualsevol d'instàncies o ocurrències de l'entitat B, mentre que una instància o ocurrència de l'entitat B s'associa, 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 s'associa, com a màxim, amb una instància o ocurrència de l'entitat B, mentre que una instància o ocurrència de l'entitat B s'associa 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 s'associa amb un nombre qualsevol d'instàncies o ocurrències de l'entitat B i viceversa.

La manera de representar la cardinalitat és la següent:

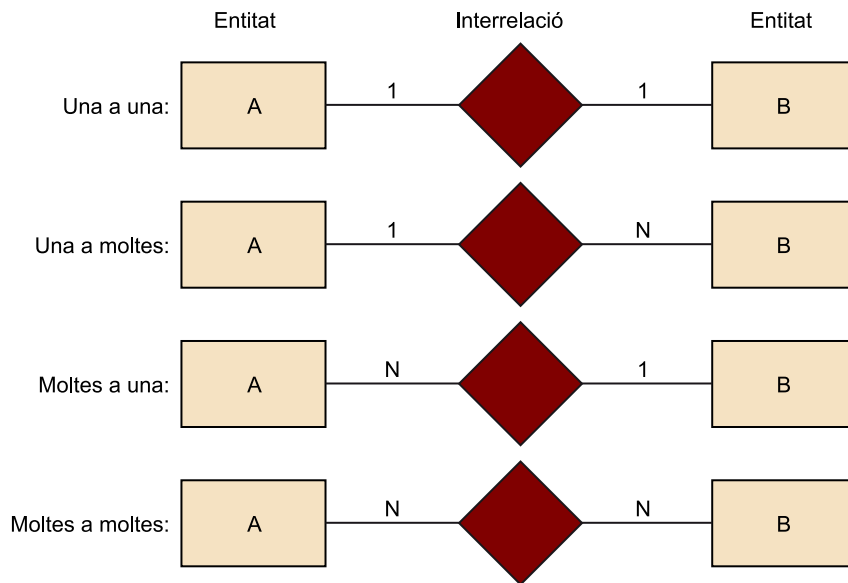


Figura 2. Representació de les diferents correspondències o raons de cardinalitat de les interrelacions entre entitats.

### Exemples de raó de cardinalitat

Si considerem que un préstec bancari pot ser, com a màxim, d'un client i, al seu torn, un mateix client pot tenir diversos préstecs, llavors la raó de cardinalitat de la interrelació entre les entitats CLIENT i PRESTEC és d'una a moltes (1:N).

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 interrelació amb raó de cardinalitat de moltes a moltes (N:M).

## 1.8. Generació del model relacional a partir del model entitat-interrelació

Una vegada elaborat el model entitat-interrelació 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 del model entitat-interrelació són les següents:

- Cada entitat forta es converteix en una taula, i els seus atributs en camps de la taula.
- Cada entitat feble es transforma en una taula, els camps de la qual són els atributs de l'entitat feble més la clau primària de l'entitat forta amb què s'associa.
- Cada interrelació es transforma en una taula els camps de la qual són les claus primàries de cada una de les entitats amb què s'associa més els atributs propis de la interrelació.
- Tota interrelació amb raó de cardinalitat  $N:M$  entre dues entitats es transforma en una taula intermèdia, i s'associa amb les taules de les dues entitats anteriors mitjançant raons de cardinalitat  $1:N$ .

### El model relacional

El model relacional postulat per E. F. Codd el 1970 es basa en la lògica de predicats i la teoria de conjunts. La idea fonamental és l'ús de "relacions". Una relació es conceptualitza com si fos una taula composta per conjunts de dades en **registres** (files o *tuples*) i **camps** (columnes).

### Vegeu també

El concepte de *taula* o *relació* es presenta en l'apartat "Visió informal d'una relació" del mòdul "El model relacional i l'àlgebra relacional".

Recordeu que els conceptes d'entitat i interrelació en el model conceptual E/R corresponen a un únic concepte en el model relacional: el de *relació* (o *taula*).

**Taula = relació**

En el model relacional, el concepte de *relació* és sinònim de *taula*.

### 1.9. Exemple de conversió d'un model entitat-interrelació 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 bancària s'identifica per un número d'operació, una data d'operació i un import.
  - Un client pot tenir molts comptes.
  - Un compte pot tenir diversos clients.
  - Un compte només pot pertànyer a una sucursal.
- Aquestes tres restriccions es poden esquematitzar de la manera següent:
- 1 client : N comptes
  - 1 compte : N clients
  - 1 sucursal : N comptes
- N clients : N comptes

Partint de les restriccions anteriors identifiquem les entitats següents amb els atributs corresponents del model entitat-interrelació:

- CLIENT (*NIF, nom, cognom*)
- SUCURSAL (*numSuc, adreça, poblacio, telefon*)
- COMPTE (*numCte, tipus, interes, saldo*)
- OPERACIO (*numOpe, data, import*)

Aquest model E/R es pot representar gràficament a partir del diagrama E/R següent:

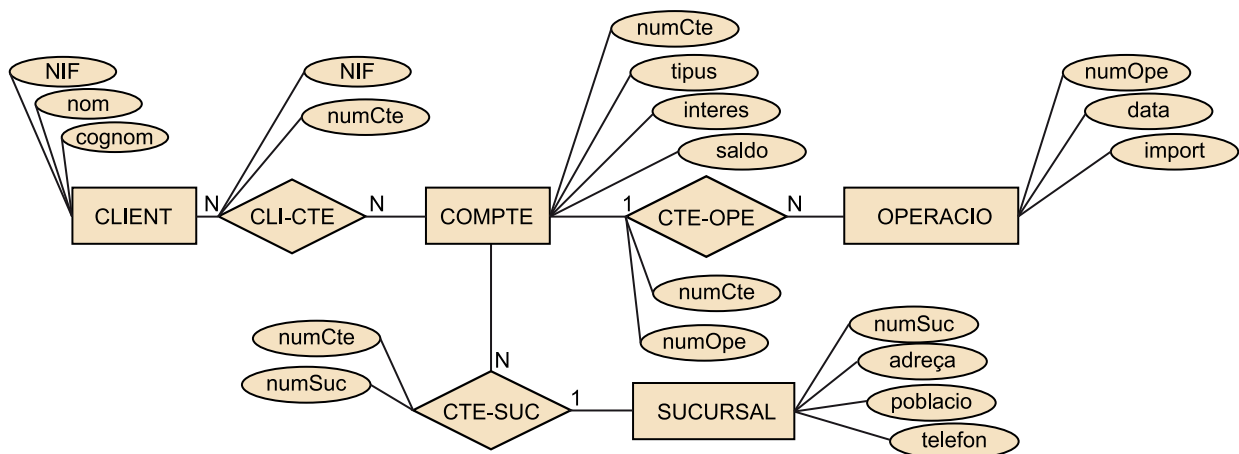


Figura 3. Model E/R de la base de dades de l'entitat financera d'exemple.

Les entitats s'associen per mitjà de tres interrelacions que també hem representat.

## Noms de taules i camps

De forma similar al conveni exposat a l'apartat 1.3 d'aquest document, escrivim les taules en majúscula i en singular; i els camps en minúscula i cursiva. Si el nom està format per diverses paraules, és convenient escriure'l sense espais, segons la notació següent:

- **Taules:** separant les paraules amb un guió baix (per exemple, COMANDA\_PENDENT), o inusualment seguint la sintaxi Pascal Case (lletra inicial de cada paraula en majúscula; per exemple: ComandaPendent).
- **Camps:** segons la sintaxi Camel Case (lletra inicial de cada paraula en majúscula, excepte la primera en minúscula; per exemple, *codiPostal*, *dataDeNaixement*). En algunes ocasions, també s'usa la notació "Underscore separated"<sup>2</sup>.

<sup>(2)</sup>Sintaxi per a escriure noms que consisteix a separar les paraules amb un guió baix.

Aplicant les regles de transformació d'entitats, interrelacions i atributs, el diagrama E/R es converteix en l'estructura de taules següent (anomenada *esquema de la base de dades*):

Taules procedents d'entitats	Taules procedents d'interrelacions
CLIENT ( <i>NIF</i> , <i>nom</i> , <i>cognom</i> )	CLI-CTE ( <i>NIF</i> , <i>numCte</i> )
SUCURSAL ( <i>numSuc</i> , <i>adreça</i> , <i>poblacio</i> , <i>telefon</i> )	CTE-OPE ( <i>numCte</i> , <i>numOpe</i> )
COMPTE ( <i>numCte</i> , <i>tipus</i> , <i>interes</i> , <i>saldo</i> )	CTE-SUC ( <i>numCte</i> , <i>numSuc</i> )
OPERACIO ( <i>numOpe</i> , <i>data</i> , <i>import</i> )	

### Vegeu també

El concepte d'*esquema* es descriu en l'apartat "Esquema de la base de dades" del mòdul "Sistemes de base de dades". En els apartats "L'arquitectura de tres nivells dels ANSI/SPARC" i "El nivell conceptual" del mateix mòdul s'explica la implicació de l'*esquema conceptual* de la base de dades en els SGBD.

Podem veure el següent:

- Les entitats i les interrelacions s'han transformat en taula.
- Els camps de les taules procedents d'entitats (CLIENT, SUCURSAL, COMPTE i OPERACIO) són els atributs d'aquestes entitats.
- Els camps de les taules procedents d'interrelacions (CLI-CTE, CTE-SUC i CTE-OPE) són les claus primàries de cada una de les entitats que s'associen.
- La clau primària es denota subratllant el nom dels camps que la formen.

L'estructura de taules implementada amb l'Access és la següent:

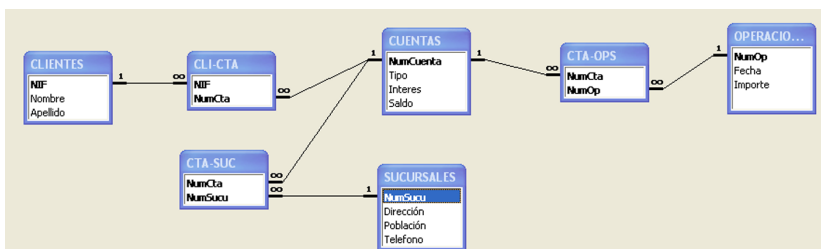


Figura 4. Estructura de taules corresponent al model E/R de la figura 3.

### Nota

La clau primària de cada taula es destaca en negreta. La raó de cardinalitat entre taules s'indica sobre la línia que les uneix mitjançant associacions 1 a ∞ (una a moltes).

## 1.10. Integritat referencial

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

No es manté la integritat quan en una interrelació de raó de cardinalitat 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 interrelacions, 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 les següents:

1) **Quan duem a terme una eliminació.** A fi de mantenir la integritat referencial és possible que el sistema gestor de bases de dades dugui a terme la propagació de les eliminacions a les taules associades. Si eliminem un registre d'una taula amb cardinalitat 1 associada mitjançant una raó de cardinalitat 1:N amb una altra, també s'eliminaran els registres de la taula associada amb cardinalitat N.

### Exemple d'eliminació de registres per propagació

Partint del cas que estem abordant (l'entitat financera), si eliminem el client amb NIF 21.212.121-A (de la taula CLIENT, amb cardinalitat 1) i exigim integritat referencial, també s'eliminaran els registres amb aquest NIF de la taula CLI-CTE (amb cardinalitat N).

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

### Exemple de modificació de registres per propagació

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

#### Nota

Recordeu que en l'Access la clau primària s'anomena *clau principal*.

#### Nota

Recordeu que l'associació o vincle entre taules s'anomena *relació*. Per tant, en aquest SGBD, *relació* no és sinònim de *taula*.

En l'Access trobem l'opció d'exigir integritat referencial quan definim les inter-relacions entre taules, tal com mostra el quadre de diàleg de la imatge següent:

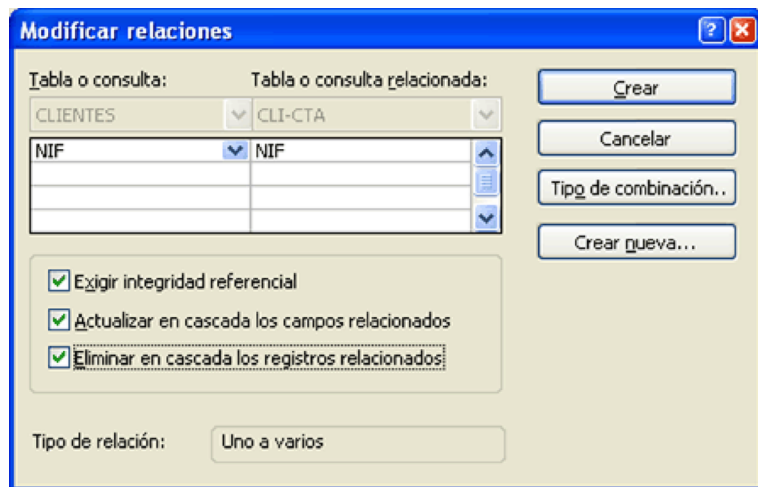


Figura 5. Quadre de diàleg que mostra l'exigència d'integritat referencial d'una interrelació en l'Access.

### 1.11. Llenguatge SQL (*Structured query language*)

SQL són les inicials de *structured query language* o *llenguatge d'interrogació estructurada*. 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. Tots ells presenten les seves petites adaptacions d'aquest llenguatge, de manera que trobem algunes variacions del llenguatge SQL en funció de l'SGBD que utilitzem, per bé que 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, inserir dades, modificar dades en la base de dades, etc.

### 1.12. Contingut de la base de dades de l'exemple

Sobre la base de dades de comptes i operacions de clients d'una entitat bancària exposada en l'apartat 1.9, considerarem que tenim carregada la informació següent:

a) Sobre l'entitat SUCURSAL:

SUCURSALES : Taula				
	NumSucu	Direcció	Població	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 SUCURSAL.

## b) Sobre l'entitat CLIENT:

CLIENTES : Taula			
	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 CLIENT.

## c) Sobre l'entitat OPERACIO:

OPERACIONES : Taula			
	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 OPERACIO.

## d) Sobre l'entitat COMPTE:

CUENTAS : Taula				
	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 COMPTE.



e) Sobre la interrelació CLI-CTE (entre les entitats CLIENT i COMPTE):

CLI-CTA : Taula		
	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.

f) Sobre la interrelació CTE-OPE (entre les entitats COMPTE i OPERACIO):

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-OPE.

g) Sobre la interrelació CTE-SUC (entre les entitats COMPTE i SUCURSAL):

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.

### 1.13. Mètodes de consultes QBE (*Query by example*)

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

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

Per a accedir a l'editor QBE de l'Access hem de seleccionar l'opció de crear una nova consulta en la vista Disseny.

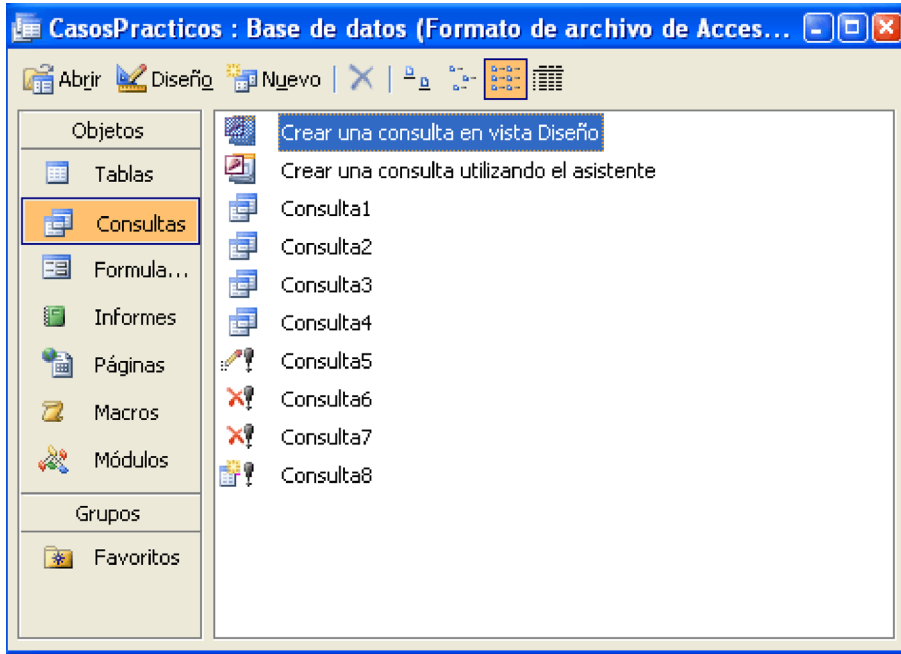


Figura 13. Finestra d'aplicació amb el detall de l'accés al QBE de l'Access.

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

#### Grid o graella

Quadricula de files i columnes amb cel·les sobre les quals es poden arrossegat els camps de les taules de la base de dades.

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.

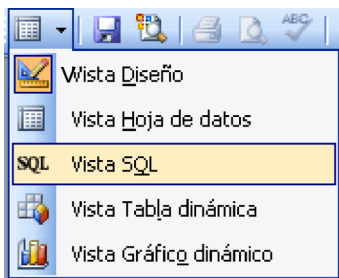


Figura 14. Menú de vistes amb l'opció d'accés a la redacció i visualització de sentències SQL en l'Access.

Per a simplificar la representació dels camps s'utilitza, igual com en l'SQL, el símbol asterisc (\*) 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", és a dir, el resultat de la consulta, es calcula en el moment de llançar la consulta que la utilitza. És a dir, treballarem amb dades actualitzades i no estàtiques.

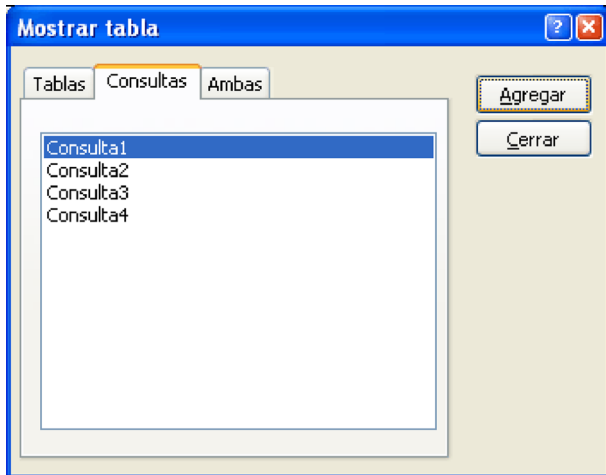


Figura 15. Quadre de diàleg "Mostrar taula" d'Access que permet utilitzar les consultes anteriors com si fossin taules.

## 2. Cas pràctic

Per al desenvolupament d'aquest cas pràctic partirem de la base de dades de l'entitat financera formada per les taules CLIENT, COMPTE, OPERACIO i SU-CURSAL que hem creat en l'exemple de l'apartat 1.9 d'aquest document.

Sobre la base de dades plantejarem una sèrie de consultes que haurem de resoldre mitjançant el llenguatge SQL i l'editor QBE de l'Access.

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

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

```
SELECT CLIENT.nom  
FROM CLIENT  
ORDER BY CLIENT.nom
```

Fer-ho des de l'editor QBE és, igualment, molt senzill. Vegem els passos que hem de fer:

- 1) En el quadre de diàleg "Mostrar taula" seleccionem la taula sobre la qual volem fer la consulta; en aquest cas, CLIENT. A continuació, tanquem aquest quadre de diàleg.
- 2) Arrosseguem el camp *nom* de la taula CLIENT que apareix en la part superior de la finestra i el deixem anar en la primera columna del *grid* (la graella que apareix en la part inferior de la finestra). D'aquesta manera se seleccionen tots els clients.
- 3) Com que volem que el resultat estigui ordenat alfabèticament, seleccionem l'opció "Ascendent" en la fila "Ordre" del *grid*.

#### Remissions al material didàctic

Totes les referències facilitades en aquest cas pràctic remeten a apartats del mòdul "El llenguatge SQL" de l'assignatura *Bases de dades*.

#### Vegeu també

La instrucció SELECT FROM i l'opció ORDER BY es descriuen respectivament en els apartats "Consultes a una base de dades relacional" i "Ordenació de les dades obtingudes en respostes a consultes" del mòdul "El llenguatge SQL".

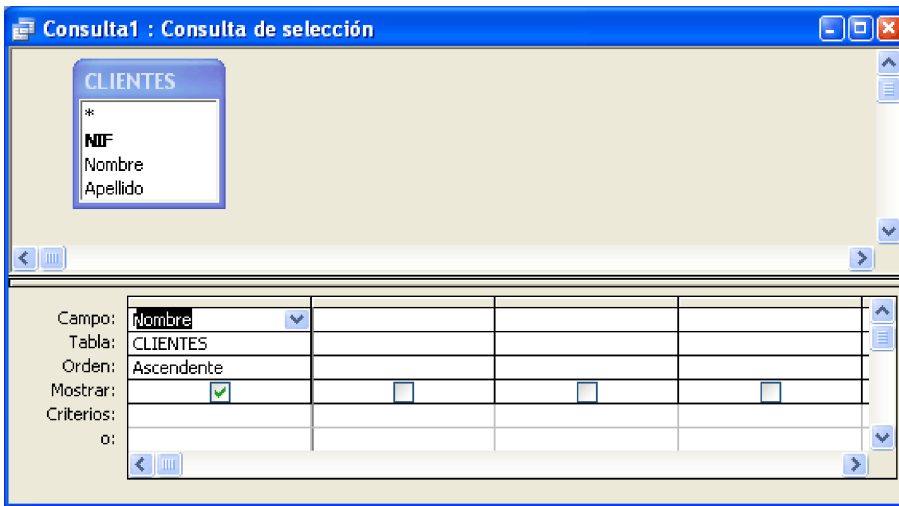


Figura 16. Finestra de l'editor QBE d'Access que representa la Consulta 1.

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

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

```
SELECT COMPTE.*, CLIENT.NIF
FROM COMPTE INNER JOIN (CLIENT INNER JOIN
[CLI-CTE] ON CLIENT.NIF = [CLI-CTE].NIF) ON
COMPTE.numCte = [CLI-CTE].numCte
WHERE ((CLIENT.NIF) = "11111111-A");
```

### La clàusula WHERE

La clàusula WHERE permet seleccionar les files que compleixen una determinada condició.

### Vegeu també

La instrucció de combinació interna INNER JOIN amb l'opció ON es descriu en l'apartat "Consultes a més d'una taula".

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 (CLIENT, CLI-CTE i COMPTE), 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 la següent:

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

### L'opció ON

L'opció ON permet expressar les condicions de combinació amb operadors de comparació (=, <, >, <=, >=, o <>).

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

1) Des del quadre de diàleg "Mostrar taula" seleccionem les taules que necessitem per a aquesta consulta, que són CLIENT, COMPTE i CLI-CTE. A continuació tanquem el quadre de diàleg.

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

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

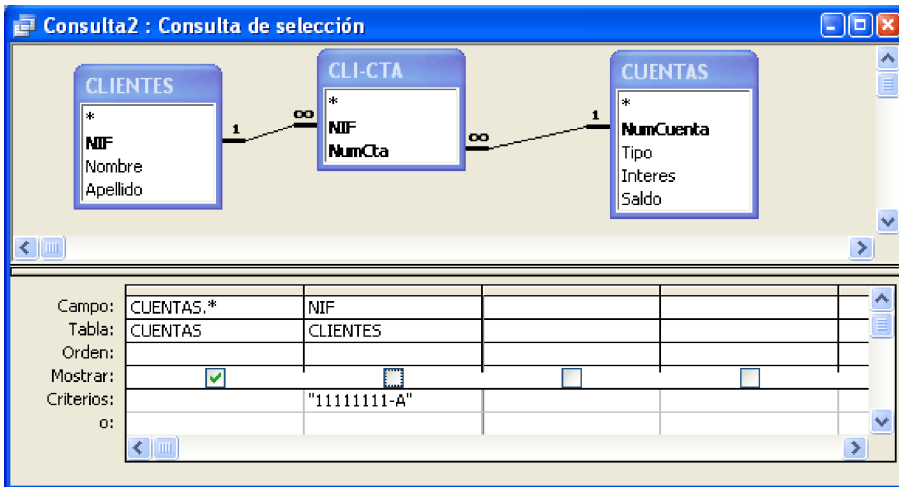


Figura 17. Finestra de l'editor QBE de l'Access que representa la Consulta 2.

Hi ha una manera més senzilla de fer aquesta consulta: simplement amb dues taules en comptes de tres, utilitzant exclusivament les taules COMPTE i CLI-CTE.

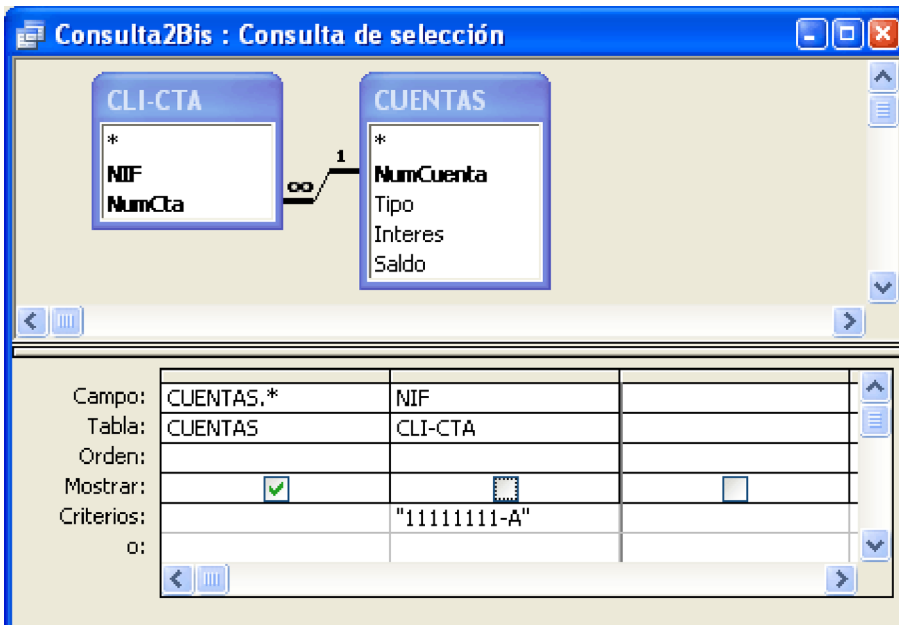


Figura 18. Finestra de l'editor QBE de l'Access que representa la Consulta 2 amb només dues taules.

### 2.3. Consulta 3: calcular l'import dels 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:

- El càlcul d'interessos per a tots els productes de l'entitat es calcula multiplicant el saldo per l'interès. Això ho expressem de la manera següent: *saldo \* interes*.
- 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 CLIENT.nom, Sum(([interes]*[saldo])) AS subtotal
FROM COMPTE INNER JOIN (CLIENT INNER JOIN
[CLI-CTE] ON CLIENT.NIF = [CLI-CTE].NIF) ON
COMPTE.numCte = [CLI-CTE].numCte
GROUP BY CLIENT.nom;
```

L'opció AS permet la definició d'àlies, és a dir, reanomenar els camps que volem seleccionar o les taules que volem consultar. Fixeu-vos que, en aquest cas, hem donat un àlies al camp calculat i l'hem anomenat *subtotal*.

D'altra banda, la clàusula GROUP BY permet agrupar registres segons el camp indicat. En aquest cas, agrupem els clients per *nom*.

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 els següents:

- 1) Des del quadre de diàleg "Mostrar taula" seleccionem les taules CLIENT, CLI-CTE i COMPTE. Tanquem el quadre de diàleg "Mostrar taula".
- 2) Des de la taula CLIENT 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 *interes* multiplicat per *saldo*. Atès que en aquesta consulta no hi ha camps que s'anomenin igual en la taula COMPTE i la taula CLIENT, podem evitar escriure la taula de la qual provenen. Si no fos així, les hauríem de referenciar precedint els camps amb el nom de la taula i un punt.

#### Vegeu també

Podeu trobar la sintaxi de la sentència SELECT FROM amb l'opció AS en l'apartat "Consultes a una BD relacional".

#### Vegeu també

La sintaxi de la clàusula GROUP BY s'indica en l'apartat "Consultes amb agrupació de files d'una taula".

#### Nota

Recordeu que, en una consulta, no sempre cal indicar per a cada camp el nom de la taula separat amb un punt. Això només és necessari quan seleccionem dades de taules diferents però amb el mateix nom de camp.

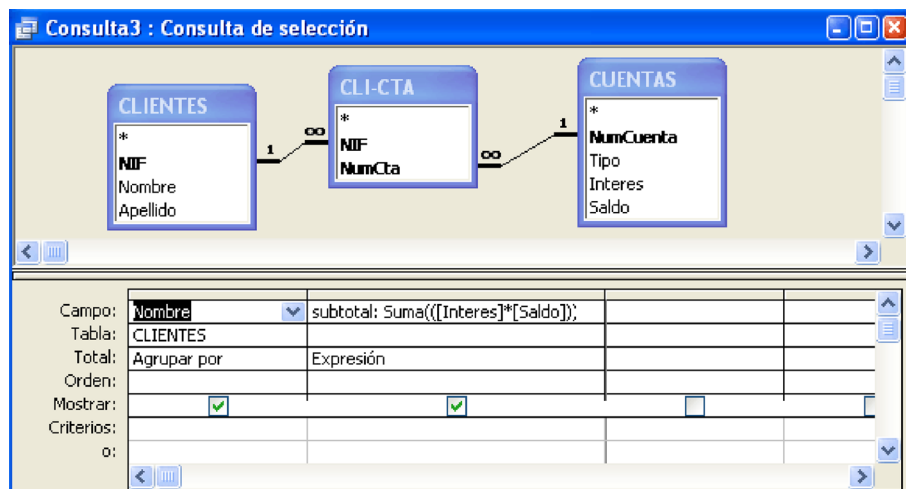


Figura 19. Finestra de l'editor QBE de l'Access que representa la Consulta 3.

Vegem quina és la utilitat de 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.

## Activitat

### Taules diferents amb camps del mateix nom

Amb aquesta consulta 3 podeu provar d'indicar el nom de la taula separada amb un punt del nom del camp. Això és útil quan cal tractar amb diferents taules que continguin alguns camps amb nom idèntic.

## 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 (COMPTE.saldo)
FROM COMPTE;
```

La funció d'agregació SUM suma els valors d'una columna. En aquest cas, la columna *saldo* de la taula COMPTE.

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

1) Com sempre, des del quadre de diàleg "Mostrar taula" seleccionem les taules que necessitem per a la nostra consulta; en aquest cas, només la taula COMPTE.

### Vegeu també

Podeu trobar un exemple d'ús de la funció SUM al final de l'apartat "Consultes amb agrupació de files d'una taula".



2) Des de la barra d'eines punxem sobre el botó amb 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 COMPTE a la primera columna del *grid*.

4) Per defecte, en la fila "Total" apareix l'opció "Agrupar per", i si despleguem el menú veurem que hi ha diverses opcions. Com és lògic, seleccionem l'opció "Suma".

### El símbol $\Sigma$

El símbol  $\Sigma$  és la lletra sigma majúscula de l'alfabet grec i s'utilitza molt sovint com a notació de *sumatori* (operador matemàtic que representa de forma compacta la suma d'un conjunt de nombres).

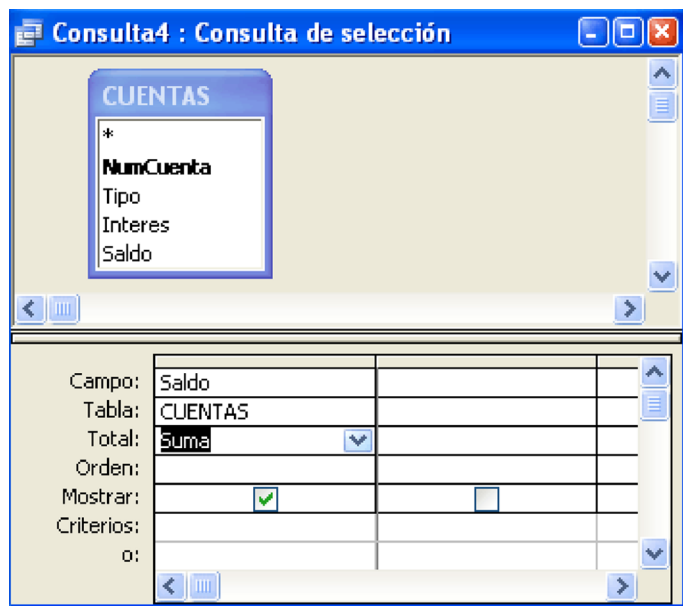


Figura 20. Finestra de l'editor QBE de l'Access que representa la Consulta 4.

Sovint és útil aprofitar aquest càlcul per a altres consultes reutilitzant la consulta com si fos una taula més. En aquest cas, però, a diferència de l'anterior, no hem donat un àlies a aquest càlcul.

De nou, el *grid* també permet fer aquest càlcul donant-li un àlies. Simplement hem de desactivar el símbol  $\Sigma$  i escriure la fórmula de càlcul en la primera fila del *grid*. En aquest cas, el càlcul és la suma de saldos, la qual expressem de la manera següent: Suma (*saldo*).

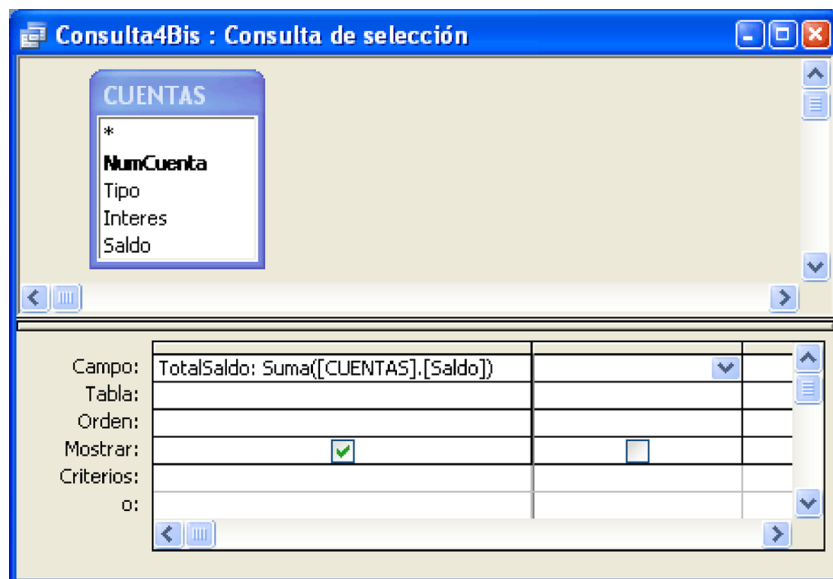


Figura 21. Finestra de l'editor QBE de l'Access que representa la Consulta 4 sense utilitzar un àlies.

Perquè serveixi de guia a l'hora de referenciar camps de les taules, per a aquest exemple s'ha utilitzat la ruta completa del camp, és a dir "[COMPTE].[saldo]", encara que simplement amb *saldo* hauria estat suficient, ja que no hi ha cap altre camp amb el mateix nom 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 els registres de la taula COMPTE sumi una unitat al valor del seu camp *interes*.

Per a modificar els valors d'alguns registres d'una taula cal usar la sentència UPDATE SET WHERE. La consulta en SQL que necessitem seria la següent:

```
UPDATE COMPTE INNER JOIN (CLIENT INNER JOIN [CLI-CTE]
ON CLIENT.NIF = [CLI-CTE].NIF)
ON COMPTE.numCte = [CLI-CTE].numCte

SET COMPTE.interes = [interes]+1

WHERE (((CLIENT.nom)="Joan"));
```

### Vegeu també

La sintaxi de la sentència UPDATE SET WHERE s'indica en l'apartat "Modificació de files d'una taula".

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 consulta haurem d'escollir l'opció corresponent (creació de taula, annexió de dades, actualització, eliminació, etc.).

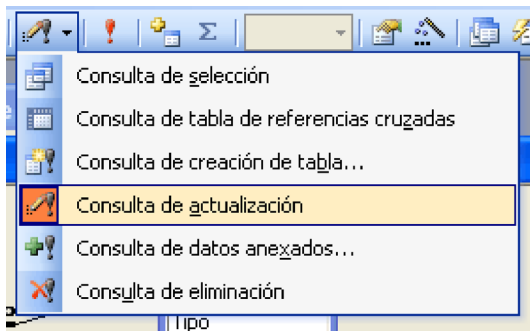


Figura 22. Menú en l'Access per a la selecció del tipus de consulta, seleccionant l'opció corresponent a una consulta d'actualització.

En aquest cas, escollim l'opció corresponent a una consulta d'actualització. Quan fem aquest canvi, observem certes variacions sobre les dades que sol·licita el QBE de l'Access en funció del tipus de consulta de què es tracti.

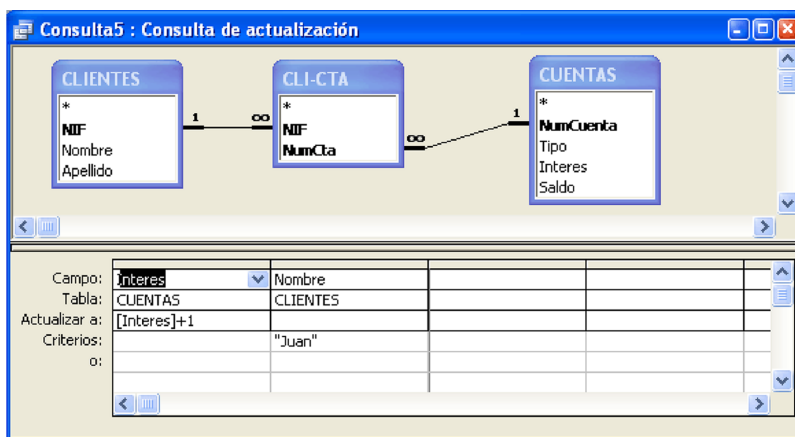


Figura 23. Finestra de l'editor QBE de l'Access que representa la Consulta 5.

## 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 CLIENT.

Per a esborrar valors d'algunes files d'una taula disposem de la sentència DELETE FROM WHERE.

La sentència SQL que permet eliminar totes les dades del Joan és la següent:

```
DELETE CLIENT.nom
FROM CLIENT
WHERE ((CLIENT.nom)="Joan");
```

### Vegeu també

El format de la sentència DELETE FROM WHERE s'indica en l'apartat "Esborrament de files d'una taula".

Fixem-nos que l'especificació dels camps a eliminar darrere 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 CLIENT.NIF
FROM CLIENT
WHERE ( ((CLIENT.nom)="Joan") );
```

O bé:

```
DELETE CLIENT.cognom
FROM CLIENT
WHERE ( ((CLIENT.nom)="Joan") );
```

Vegem què ha passat en la taula CLI-CTE. Fixem-nos que també s'han esborrat les ocurrencies que hi hagués relatives al client Joan. Dit d'una altra manera, en la taula CLI-CTE ja no apareixen registres que tinguin "1111111-A" com a valor del camp *NIF* (que és el *NIF* del Joan). Els continguts de les taules CLIENT i CLI-CTE després de l'execució de la sentència SQL són, respectivament, els següents:

CLIENTES : Taula			
	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 CLIENT 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, ja que un cop executada ja no és possible recuperar la situació immediatament anterior. 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 l'estem aplicant directament. Aquest punt és especialment important si tenim en compte que l'Access no informa d'aquestes implicacions, és a dir, no avisa dels esborraments o canvis que farà sobre altres taules de la base de dades.

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

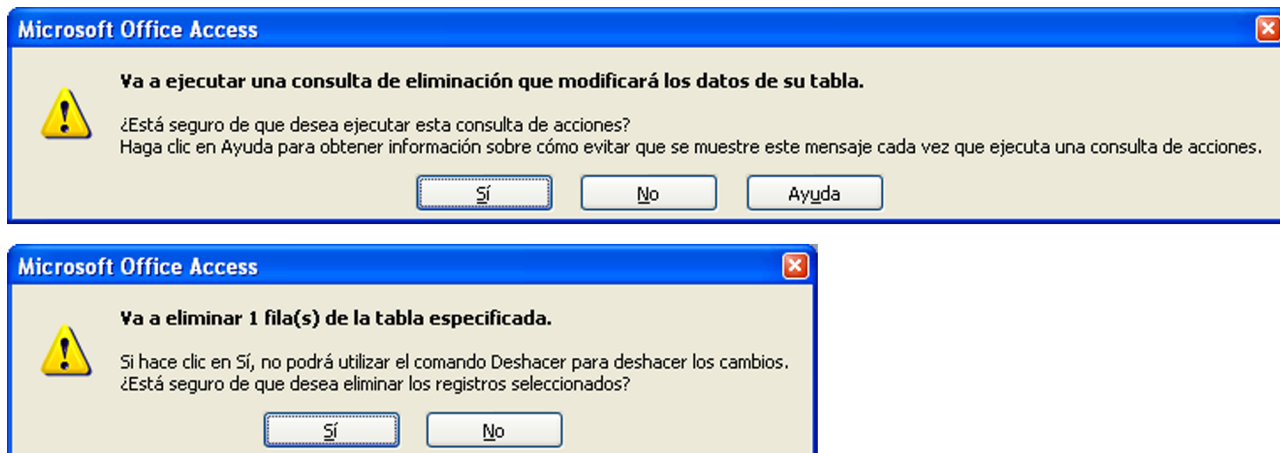


Figura 26. Quadres de diàleg amb missatges d'avís d'actualització i/o esborrament de taules de l'Access.

Com podeu veure, l'Access només avisa de l'esborrament d'una fila (registre) d'una taula, quan en realitat estarem eliminant un registre de la taula CLIENT i dos de la taula CLI-CTE (aquests últims pel fet d'haver exigit integritat referencial en l'associació entre CLIENT i CLI-CTE).

## 2.7. Consulta 7: duplicar la taula CLIENT sobre una de nova anomenada CLIENT2

Si no hi ha una taula amb el nom de la taula de destinació de la consulta, l'Access la crea, la qual cosa permet usar 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 estalvia haver d'especificar-los.

Per tant, la sentència que necessitem és la següent:

```
SELECT CLIENT.* INTO CLIENT2
FROM CLIENT;
```

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

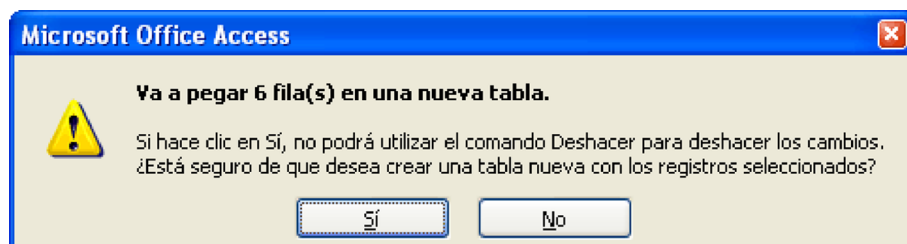


Figura 27. Quadres de diàleg amb missatge de creació d'una nova taula amb incorporació de dades en l'Access.

### La sentència SELECT-INTO

La instrucció SELECT amb la clàusula INTO permet combinar dades d'una o més taules en una taula nova. Se sol utilitzar per a fer una còpia de seguretat de les dades d'una taula.

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

1) Escollim l'opció de creació de nova Consulta en la vista Disseny i afegim la taula CLIENT del quadre de diàleg "Mostrar taula". Tanquem el quadre de diàleg.

2) Dels tipus de consulta, seleccionem la consulta de creació de taula (botó Crear taula).

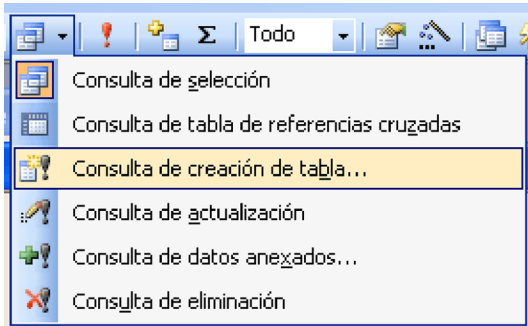


Figura 28. Menú per a la selecció d'una consulta de creació de taula en l'Access.

3) Especifiquem el nom de la taula nova que volem crear. En aquest punt cal tenir en compte que l'Access presenta la possibilitat d'incorporar els registres sobre taules ja existents, que fins i tot poden ser d'altres bases de dades.

### Activitat

Copieu taules sobre altres bases de dades.

Com a variant d'aquesta consulta 7 podeu provar d'incorporar registres sobre taules d'altres bases de dades.

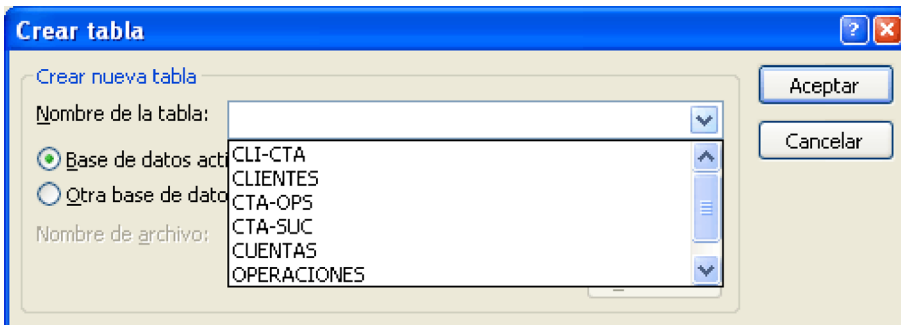


Figura 29. Quadre de diàleg de creació de taula amb la selecció del seu nom.

4) A continuació indiquem a la graella (*grid*) els camps dels quals es nodrirà la informació de la nova taula CLIENT2. Per aquest motiu, haurem d'indicar que són tots els camps de la taula CLIENT, el que podem fer, com ja sabem, mitjançant el símbol asterisc (\*).

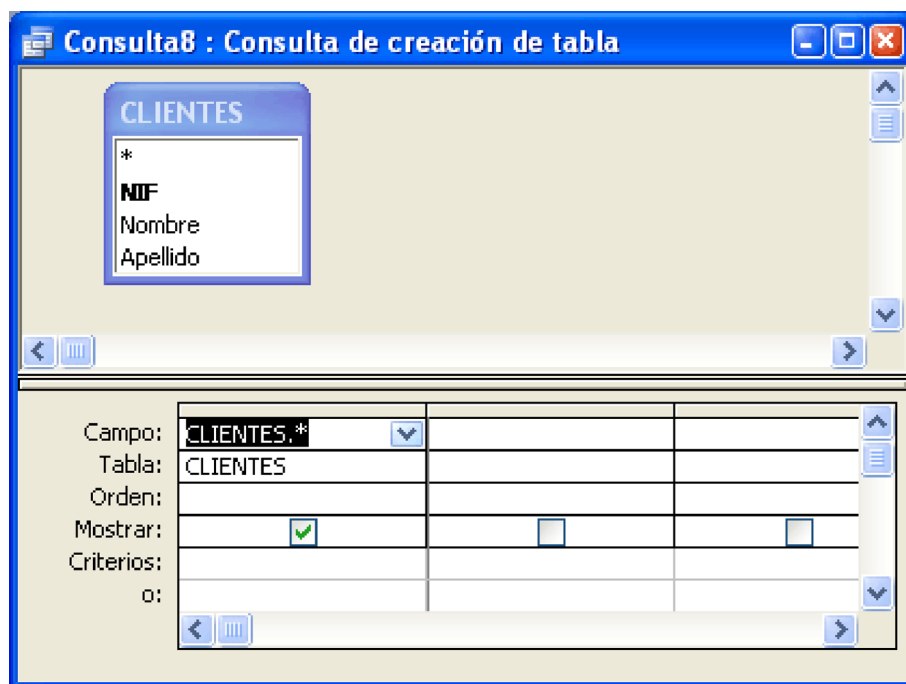


Figura 30. Finestra de l'editor QBE de l'Access que representa la Consulta 7.

## 2.8. Consulta 8: incorporar un nou client a la taula CLIENT2

També és possible que vulguem incorporar nous registres a les taules. En aquest cas incorporem a la taula CLIENT2 (que acabem de crear en la consulta anterior) un nou client anomenat Mario García i amb *NIF* 77777777-G.

Per introduir noves files en una taula utilitzant l'SQL, ho faríem amb la sentència INSERT INTO VALUES de la manera següent:

```
INSERT INTO CLIENT2
VALUES ("77777777-G", "Mario", "García");
```

### Vegeu també

El format de la sentència INSERT INTO amb la clàusula VALUES s'especifica en l'apartat "Inserció de files en una taula".

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

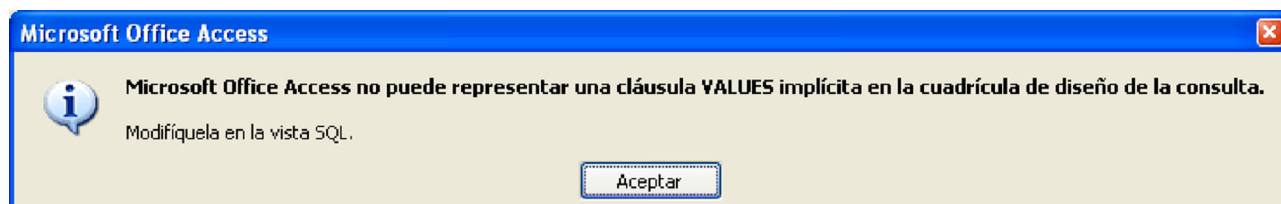


Figura 31. Quadre de diàleg d'avís d'error per haver utilitzat la clàusula VALUES en l'editor QBE de l'Access.





## Bibliografia

**Beaulieu, A.** (2010). *Aprende SQL* (2a. ed.). Madrid: Anaya Multimedia.

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

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

**Date, C. J.; Darwen, H.** (1997). *A guide to the SQL Standard* (4a. ed.). Reading, Massachussets: Addison-Wesley.

**Melton, J.; Simon, A. R.** (2002). *SQL: 1999. Understanding Relational Language Components*. San Francisco: Morgan Kaufmann.

**Silberschatz, A.; Korth, H. F.; Sudarshan, S.** (2006). *Fundamentos de bases de datos* (5a. ed.). Madrid: McGraw-Hill.

*Database Language SQL. Document ANSI/X3.135*. American National Standards Institute (ANSI).

*Database Language SQL. Document ISO/IEC 9075*. International Organization for Standardization (ISO).

