

# Apunts de Microsoft Access

Ramón López, 2004

<http://www.proferamon.com>  
[info@proferamon.com](mailto:info@proferamon.com)

# Índex

Introducció i objectiu .....	4
Objectiu .....	4
Introducció a Access .....	4
Què és una base de dades per a Access? .....	4
Creació d'una base de dades .....	5
Bases de dades relacionals .....	7
Característiques .....	7
Disseny .....	7
Qüestions que cal tenir presents a l'hora de definir claus primàries i secundàries .....	7
Abans de crear una base de dades .....	7
Disseny de bases de dades amb Microsoft Access .....	8
Procés detallat del disseny d'una base de dades .....	8
Exercici 1: habitants i municipis .....	10
Exercici 2: cursos de formació .....	10
Exercici 3: campionat d'escacs .....	10
Exercici 4: energia elèctrica .....	11
Taules .....	12
Introducció .....	12
Creació de taules .....	12
Definició de camps .....	12
Clau principal .....	15
Guardar el disseny .....	15
Introducció de dades .....	16
Interrelacions i integritat .....	18
Exercici 5: creació de taules .....	20
Exercici 6: relacions .....	22
Exercici 7: introducció de dades .....	22
Exercici 8: creació de taules .....	23
Exercici 9: introducció de dades .....	23
Annex 1: formats de presentació de dades .....	24
Camps de tipus data/hora .....	24
Camps de tipus numèric i moneda .....	25
Camps de tipus text i memo .....	25
Camps de tipus sí /no .....	26
Annex 2: format de màscares d'entrada .....	26
Consultes .....	27
Introducció a les consultes .....	27
Introducció al SQL .....	27
Consultes de selecció simple .....	27
Addició de camps .....	28
Operadors i expressions .....	29
Valors repetits .....	29
Ordenació de registres .....	30
Agrupament de dades .....	30
Filtrat de registres de sortida .....	31
Consultes sobre múltiples taules .....	32
Producte cartesià .....	32
Unió ( <i>join</i> ) .....	34
Consultes d'inserció .....	35
Consultes de creació de taula .....	35
Consultes d'actualització .....	36
Consultes d'eliminació .....	36
Consultes imbricades .....	36
Consultes de taula de referències creuades .....	38
Consultes específiques de SQL .....	38
Consultes d'unió de taules .....	38
Consultes de definició de dades .....	39
Modificació i accés a les dades d'una consulta. Vistes. ....	39
Exercici 10: consultes .....	40
Exercici 11: consultes .....	41
Exercici 12: consultes SQL .....	41
Formularis .....	43
Introducció .....	43
Creació de formularis .....	43
L'objecte formulari .....	44
Esdeveniments .....	46
Esdeveniments per als controls .....	47
Controls en un formulari .....	47
Propietats comunes .....	48
Propietats dels controls de llistes .....	48

Subformularis .....	49
Exercici 13: formularis .....	50
Creació d'un formulari per a Botigues .....	50
Creació d'un formulari per a BExistències .....	50
Exercici 14.....	51
Formularis imbricats .....	51
Exercici 15.....	52
Informes .....	53
Introducció .....	53
Controls en informes.....	53
Agrupament de dades en informes .....	53
Format de pàgina i columnes .....	54
Funcions interessants en informes .....	55
Exercici 16: informes .....	55
Macros .....	57
Operacions amb macros .....	57
Construcció de macros .....	59
Referències als objectes dels formularis o informes.....	60
Execució condicional .....	61
Bucles.....	61
Exemple de construcció i ús de macros .....	61
Organització de macros.....	63
Macros autoexecutables .....	63
Exercici 17: macros .....	64
Altres.....	65
Com protegir la base de dades mitjançant comptes d'usuari i grup .....	65

# Introducció i objectiu

## Objectiu

L'objectiu d'aquest manual és l'aprenentatge del SGBD Microsoft Access. Coneixerem l'entorn de treball que ofereix Access: els objectes que permet utilitzar, el seu ús i possibilitats. Els exercicis que inclou donaran solució un problema en totes les seves fases: es partirà d'un enunciat que descriu un problema i es pretén obtenir com resultat final una base de dades amb les especificacions descrites en l'enunciat. Per a això s'haurà de realitzar prèviament una anàlisi del problema utilitzant les eines estudiades en les classes de teoria.

## Introducció a Access

Access és un programa comercial de l'empresa *Microsoft Corporation*. Es tracta d'un sistema gestor de bases de dades dissenyat per ser utilitzat en computadores personals tipus PC. Encara que no està considerat com un sistema "seriós", la veritat és que permet realitzar les mateixes tasques que els tradicionalment considerats sistemes "seriosos", com pot ser Oracle. A diferència d'aquest, Access pot funcionar en sistemes de baix cost, i és molt més assequible. A més, disposa d'un entorn molt amigable. Per aquesta raó, cada vegada més empreses aposten per aquest sistema. Tant és així que els fabricants dels sistemes "seriosos", inclòs Oracle, tendeixen a fer que els seus productes se semblin, quant a interfície d'usuari es refereix, al que l'usuari demana, que no és altra cosa que l'entorn visual. Així apareix *Developer 2000*, un entorn visual per crear aplicacions per a Oracle.

Per totes aquestes raons, estudiar un sistema com Access és una bona aposta, ja que obre les portes a un mercat cada vegada més important, sense que això suposi deixar de conèixer altres sistemes molt estesos.

La instal·lació completa d'Access proporciona una sèrie d'eines de suport a l'usuari i al programador que poden facilitar la creació i administració d'una base de dades. No obstant això, ja que l'objectiu d'aquest curs és formar a l'alumne per enfrontar-se a qualsevol sistema, l'alumne haurà d'ignorar l'existència d'aquestes eines i aprendre a treballar sense recolzar-se en elles.

## Què és una base de dades per a Access?

En Access una base de dades és **un arxiu** que conté dades (estructurades i interrelacionades) i els objectes que defineixen i manegen aquestes dades: taules, consultes, formularis, informes, macros i mòduls. A més, es poden definir estratègies de seguretat i d'integritat. Poden coexistir en un disc diverses bases de dades, en diversos fitxers, absolutament independents entre si (encara que és possible referenciar des d'una base de dades objectes presents en altres bases de dades, és a dir, en altres fitxers).

**NOTA:** En alguns altres sistemes, com dBase, una base de dades és **un directori** del disc en el qual es troben multitud d'arxius que contenen les dades i els objectes.

Començarem a estudiar els diferents objectes que permetran crear una base de dades:

- **Taules:** són els magatzems de dades. Es tracta de llistes de registres. El disseny de taules consisteix en la definició dels camps necessaris per a emmagatzemar correctament la informació de la base de dades.
- **Consultes:** operacions que es poden realitzar sobre les dades de les taules: limitacions d'accés, selecció de dades, inserció, modificació i eliminació, càlculs...
- **Formularis:** pantalles que es presenten als usuaris d'una base de dades perquè tinguin un accés amigable a les dades i operacions.
- **Informes:** formats de presentació de les dades per generar còpies impreses de la informació registrada en la base de dades.
- **Macros i mòduls:** segments de codi en llenguatge d'alt nivell (Access Basic, Visual Basic, llenguatge de macros,...) que permeten realitzar operacions complexes amb les dades i objectes de la base de dades.

Els objectes en Access han d'estar identificats mitjançant un nom. Dintre de cada grup d'objectes (taules, consultes, formularis,...) els objectes s'organitzen com fitxers dintre d'un directori. Referències posteriors a aquests objectes es faran a través del seu nom identificatiu.

## Creació d'una base de dades

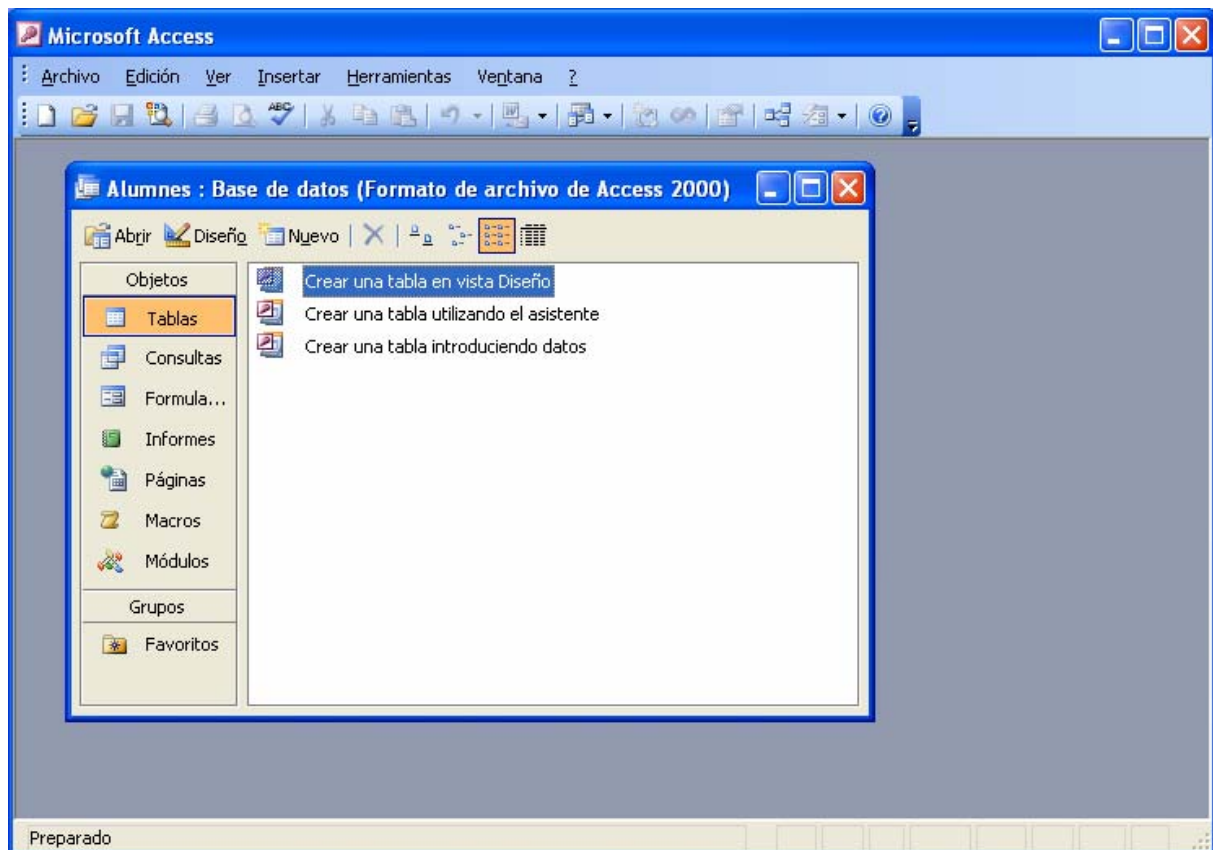
Començarem a treballar amb Access. El primer pas serà crear una base de dades buida. No es tracta d'un fitxer buit: al crear una base de dades buida s'emmagatzemen en ella algunes dades administratives i objectes del sistema, inicialment ocults a l'usuari i programador. Per tant, abans de tenir accés a una base de dades "buida", hem d'escollir un nom de fitxer per a emmagatzemar-la en disc.

Arribats a aquest punt és important advertir que, al contrari del que ocorre en la majoria dels programes d'aplicació del mercat, les dades d'una base de dades no necessiten ser explícitament guardats. Estem acostumats al fet d'haver de seleccionar l'opció de guardat del menú arxiu perquè els nostres documents i programes s'emmagatzemin en disc, mentre que la resta del temps treballem amb una còpia en RAM. En les bases de dades això no és així. Les dades estan sempre emmagatzemats en disc. Únicament es treballa amb una còpia en RAM durant el temps que un registre d'una taula està sent accedit, i **no és necessari utilitzar una opció de guardat per fer permanents els canvis en el registre**. Per tant, quan s'introdueixin o esborrin dades de les taules d'una base de dades, Access (ni cap base de dades) preguntarà si es desitgen guardar els canvis abans de tancar, perquè ja estaran guardats. Això té com avantatge que no es perdran les dades en cas que ocorri algun problema; com contrapartida sabem que no serà possible "desfer" l'esborrat accidental de dades.

No obstant, Access té un entorn suficientment amigable com per advertir a l'usuari que pot realitzar operacions destructives irreversibles abans de realitzar-les.

Això és únicament aplicable a les dades. La resta d'objectes en Access (definicions de taules, definicions de consultes, formularis, informes,...) han de ser emmagatzemats en disc dintre del fitxer de la base de dades de forma explícita.

Per aquestes raons, Access requereix crear el fitxer d'una base de dades buida abans de permetre treballar amb ella.



*Figura 1. Aspecte de la finestra principal d'una base de dades buida.*

El fitxer d'una base de dades Access té extensió MDB (*Microsoft Data Base*). A més d'aquest fitxer, pot existir un fitxer amb extensió LDB. Aquest fitxer conté informació de bloqueig de registres, que permet l'accés segur de múltiples usuaris de forma concurrent. El fitxer de la base de dades roman obert mentre es treballa amb les dades, i únicament es tanca quan es tanca la base de dades de

forma explícita, amb l'opció corresponent. Si per qualsevol problema, no es realitza un tancament ordenat, la base de dades pot danyar-se. Quan vulguem tornar a treballar amb una base de dades, l'única cosa que cal fer és obrir de nou el fitxer corresponent, com si es tractés d'un document d'un processador de text.

# Bases de dades relacionals

## Característiques

Tots els registres de la taula han de tenir la mateixa quantitat de camps, encara que algun d'ells estigui buit, han de ser registres de longitud fixa.

Cada camp té un nom o etiqueta que cal definir prèviament a la seva utilització. No obstant, una vegada creada la taula es podrà ampliar o disminuir el nombre de camps, mitjançant el SGBD (Sistema Gestor de Bases de Dades).

La base de dades podrà estar formada per moltes taules, una per cada tipus d'entitat.

Dintre d'una taula cada nom de camp ha de ser diferent.

Els registres d'una mateixa taula han de diferenciar-se, almenys, en el contingut d'algun dels seus camps, no pot haver dos registres "idèntics".

Els registres d'una taula poden estar disposats en qualsevol ordre.

El contingut de cada camp està delimitat per un rang de valors possibles.

Permet la creació de noves consultes a partir de les taules ja existents, relacionant camps de distintes taules anteriors. Aquesta condició és l'essència de les bases de dades relacionals, formant el que es diu un "taula virtual" (consulta).

## Disseny

Cada taula té un nom que la identifica unívocament.

Cada taula té un o més camps amb un nom, que estan disposats en un ordre específic d'esquerra a dreta.

Cada taula té zero o més registres, contenint cadascun un únic valor en cada camp. Els registres estan desordenats.

Tots els valors d'un camp determinat tenen el mateix tipus de dades, i aquests estan extrets d'un conjunt de valors legals que anomenen el domini del camp.

## Qüestions que cal tenir presents a l'hora de definir claus primàries i secundàries

Una clau primària és un camp o combinació de camps dintre d'una taula, el valor de la qual identifica unívocament a cada registre de la taula. Cada taula té una única clau primària.

Una clau secundària és un camp o combinació de camps en una taula, el valor de la qual és un valor de clau primària per a alguna altra taula. Una taula pot contenir més d'una clau secundària, enllaçant-la a una o diverses taules.

Una combinació clau primària / clau secundària crea una relació pare / fill entre les taules que les contenen.

## Abans de crear una base de dades

A diferència de quan treballem amb un processador de textos o un full de càlcul, en els quals anem treballant i, sobre la marxa, corregint les equivocacions quan se'ns fan evidents, millorant i ampliant els plantejaments a mesura que anem avançant en el treball, en el disseny de les bases de dades aquesta manera d'operar no és gens aconsellable. Encara que ens costi és molt important dedicar temps a pensar bé què és el que volem obtenir i esbossar sobre el paper el disseny previ abans de posar-nos a treballar amb l'ordinador. Per això l'ordre de treball hauria de ser:

1. Determinar la finalitat de la base de dades.
2. Determinar les taules que es necessiten en la base de dades.
3. Determinar els camps que es necessiten en les taules.
4. Identificar els camps amb valors exclusius en cada registre (claus primàries).
5. Determinar les relacions entre les taules.
6. Perfeccionar el disseny.

7. Introduir dades i crear altres objectes de la base de dades.
8. Utilitzar les eines d'anàlisi de Microsoft Access.

## ***Disseny de bases de dades amb Microsoft Access***

L'element central d'una base de dades és la taula. Una taula té un nom, un conjunt d'atributs (campus) que representen les seves propietats i un conjunt de registres que inclouen els valors que cadascun dels atributs pren per a cada element de la taula. Una taula es representa en dues dimensions: les columnes són els camps i les files els registres. Els camps de la taula es defineixen sobre dominis formats per un conjunt de valors homogenis. Els dominis es poden definir per extensió, indicant un rang de valors o indicant quina condició han de complir. (Exemple: els imparells).

Existeixen una sèrie de restriccions inherents:

- En una taula no pot haver dos registres iguals (obligatorietat de clau primària).
- L'ordre dels registres i el dels camps no és rellevant.
- Cada camp només pot prendre un únic valor del domini sobre el qual està definit en un registre.
- Cap camp que formi part de la clau primària d'una taula pot prendre un valor nul (regla d'integritat d'entitat).

Mecanismes del model relacional:

- La restricció de clau primària permet declarar un camp o conjunt de camps com la clau primària d'una taula (identifica unívocament cada registre de la taula).
- La restricció d'unicitat ens permet definir claus alternatives (els valors d'un o diversos camps no poden repetir-se en diferents registres d'una taula).
- La restricció d'obligatorietat permet declarar si un o diversos atributs d'una taula han de prendre sempre un valor, és a dir, no poden prendre valors nuls. La restricció d'integritat referencial, s'utilitza per relacionar taules d'una base de dades mitjançant claus alienes (conjunt de camps d'una taula que és clau primària en una altra o la mateixa taula). La integritat referencial ens indica que els valors de la clau aliena en la taula filla han de correspondre amb els valors de la clau primària en la taula pare. Els camps que són clau aliena en una taula no necessiten tenir els mateixos noms que els camps de la clau primària amb la qual ells es corresponen, però sí el mateix tipus de dades.
- A més de la integritat referencial que ens permet relacionar taules entre si donant lloc a l'estructura de la base de dades, el model relacional permet també definir les opcions d'esborrat o modificació de les claus alienes. Aquestes opcions indiquen les accions que cal portar a terme quan es produeix un esborrat o modificació d'un registre en la taula pare relacionada amb una taula filla. Les possibilitats per a una operació d'actualització (esborrat o modificació) són:
  - Actualitzar en cascada. La modificació d'un registre en la taula pare ocasiona una modificació de tots els registres relacionats en la taula filla. (Registres que la seva clau aliena coincideixi amb el valor de la clau primària del registre modificat en la taula pare).
  - Eliminar en cascada. L'eliminació d'un registre en la taula pare ocasiona una eliminació de tots els registres relacionats en la taula filla. (Registres que la seva clau aliena coincideixi amb el valor de la clau primària del registre eliminat en la taula pare).

## ***Procés detallat del disseny d'una base de dades***

1. Identificar i fitar el problema que ens ocupa.
2. Elaborar de forma el més detallada possible l'enunciat del problema.
3. Discutir l'enunciat per parts, anotant possibles entitats (taules), relacions entre taules (que normalment vindran marcades per la presència d'un verb) i els elements o conceptes que no sabem com catalogar.
4. Estudiar les cardinalitats de les relacions. Recordeu que hi ha tres tipus de relacions:
  - a) d'un a un (1:1) ( $1 - 1$ )
  - b) d'un a varis (1:N) ( $1 - \infty$ )



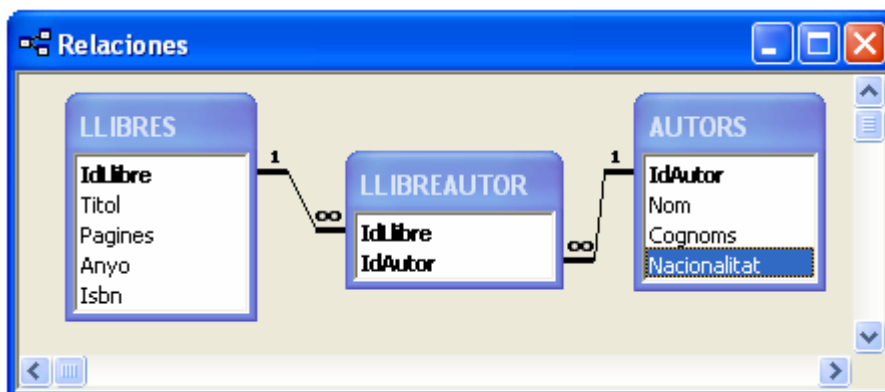
c) de varis a varis (N:M) ( $\infty - \infty$ )

5. D'acord amb les cardinalitats estudiar si són necessàries noves entitats.
6. Estudiar quin camp o camps són susceptibles de ser considerats clau principal en cadascuna de les entitats. Hem de recordar que les entitats fruit d'una relació varis a varis haurà de contenir les claus principals de cadascuna de les taules que la generen.
7. Assignar a cada entitat els atributs (camp) que es volen recollir.
8. Elaborar un esquema que contingui les taules, els seus camps, les seves claus principals, les relacions i les cardinalitats de les relacions.
9. Establir el tipus de camp, les seves dimensions i altres propietats per a tots els camps de totes les taules.

#### EXEMPLE:

1. Vull fer una base de dades dels llibres que tinc a casa.
2. En la base de dades vull recollir de cada llibre el títol, l'autor, la quantitat de pàgines, l'any de publicació i l'isbn. Dels autors, els seus noms, cognoms i la seva nacionalitat.
3. Entenc que hi ha dues entitats: LLIBRES i AUTORS. I una relació: els autors **escriuen** llibres.
4. Un autor pot escriure molts llibres. Un llibre pot haver estat escrit per molts autors. Com la relació entre les dues entitats és d'un a varis en les dues direccions, diem que la relació és de varis a varis. (N:M)
5. Ja que la relació és N:M serà necessària una tercera entitat LLIBREAUTOR.
6. En l'entitat LLIBRES pot ser clau principal l'isbn o bé un codi que se li assigni a cada llibre. En l'entitat AUTOR obligatòriament haurà de ser un codi qui identifiqui de forma unívoca a cada registre de la taula. En l'entitat LLIBREAUTOR seran dos les claus principals la de LLIBRES i la d'AUTOR.
7. De l'entitat LLIBRES guardaré: idllibre, títol, nombre de pàgines, any de publicació i isbn. De l'entitat AUTOR: idautor, nom, cognoms i nacionalitat. De l'entitat LLIBREAUTOR: idllibre i idautor.

8.



9.

TAULA                      CAMP                      TIPUS                      MIDA                      ALTRES

LLIBRES				
	<b>IdLibre</b>	Autonumèric		
	Títol	Text	50	
	Pàgines	Numèric	Enter	
	Anyo	Numèric	Enter	
	Isbn	Text	10	Indexat: Sí (sense duplicats)
LLIBREAUTOR				
	<b>IdLibre</b>	Numèric	Enter llarg	
	<b>IdAutor</b>	Numèric	Enter llarg	
AUTORS				
	<b>IdAutor</b>	Autonumèric		
	Nom	Text	25	
	Cognoms	Text	50	
	Nacionalitat	Text	15	

(Els camps clau principal estan marcats en negreta)

## ***Exercici 1: habitants i municipis***

Suposem el següent univers del discurs sobre municipis, habitatges i persones.

- Cada persona només pot habitar en un habitatge, però pot ser propietària de diversos habitatges.
- Un habitatge pot estar habitat per una o diverses persones, siguin o no propietàries de l'habitatge.
- Un habitatge pot pertànyer a una o diverses persones.
- No tots els habitatges han d'estar habitats.
- En cada habitatge habitat haurà un cap de família, la resta de les persones que convisquin amb el cap de família no podran ser al seu torn caps de família.
- De cada habitatge volem emmagatzemar el seu valor.

## ***Exercici 2: cursos de formació***

El departament de formació d'una empresa desitja construir una base de dades per planificar i gestionar la formació dels seus empleats.

- L'empresa organitza cursos interns de formació dels quals es desitja conèixer el codi de curs, el nom, una descripció, el nombre d'hores de durada i el cost del curs.
- Un curs pot tenir com prerrequisit haver realitzat un altre o uns altres prèviament, i, al seu torn la realització d'un curs pot ser prerrequisit d'uns altres. Un curs que és un prerrequisit d'un altre pot ser-ho de forma obligatòria o només recomanable.
- Un mateix curs té diferents edicions, és a dir, s'imparteix en diferents llocs, dates i amb diferents horaris (intensiu, de matí o tarda). En una mateixa data d'inici només pot impartir-se una edició d'un curs.
- Els cursos s'imparteixen per personal de la pròpia empresa.
- Dels empleats es desitja emmagatzemar el seu codi d'empleat, nom i cognoms, adreça, telèfon, NIF, data de naixement, nacionalitat, sexe, signatura i salari.
- Un mateix empleat pot ser docent en una edició d'un curs i alumne en altra edició, però mai pot ser ambdues coses alhora (en una mateixa edició d'un curs o ho imparteix o ho rep).

## ***Exercici 3: campionat d'escacs***

El club d'Escacs de Linares ha estat encarregat per la Federació Internacional d'Escacs de l'organització dels pròxims campionats mundials que se celebraran en l'esmentada localitat. Per aquest motiu, desitja dur a una base de dades tota la gestió relativa a participants, allotjaments i partides. Tenint en compte que:

- En el campionat participen jugadors i àrbitres; d'ambdós es vol conèixer el nombre d'associat, nom, adreça, telèfon de contacte i campionats en els quals ha participat (com jugador o com àrbitre). Dels jugadors es precisa a més el nivell de joc en una escala 1 a 10.
- Cap àrbitre pot participar com jugador.
- Els països envien al campionat un conjunt de jugadors i àrbitres, encara que no tots els països envien participants. Tot jugador i àrbitre són enviats per un únic país.
- Cada país s'identifica per un nombre i interessa conèixer a més del seu nom, el nombre de clubs d'escacs existents en el mateix.
- Cada partida s'identifica per un nombre, la juguen dos jugadors i l'arbitra un àrbitre. Interessa enregistrar les partides que juga cada jugador i el color (blanques o negres) amb el qual juga.
- Tant jugadors com àrbitres s'allotgen en un dels hotels en els quals es desenvolupen les partides, es desitja conèixer en quin hotel i en quines dates s'ha allotjat cadascun dels

participants. Els participants poden no romandre en Linares tot el campionat, sinó acudir quan han de jugar alguna partida allotjant-se en el mateix hotel o no. De cada hotel, es desitja conèixer el nom, l'adreça i el número de telèfon.

- El campionat es desenvolupa al llarg d'una sèrie de jornades (any, mes, dia) i cada partida té lloc en una de les jornades encara que no tinguin lloc partides totes les jornades.
- Cada partida es celebra en una de les sales de les quals poden disposar els hotels, es desitja conèixer la quantitat d'entrades venudes en la sala per a cada partida. De cada sala, es desitja conèixer la capacitat i mitjans que disposa (ràdio, televisió, vídeo...) per facilitar la retransmissió de les trobades. Una sala pot disposar de diversos mitjans.
- De cada partida es pretén enregistrar tots els moviments que la componen, la identificació de moviment s'estableix sobre la base d'un nombre d'ordre dintre de cada partida: per a cada moviment es guarden la jugada (5 posicions) i un comentari realitzat per un expert.

### **Exercici 4: energia elèctrica**

Es pretén portar a terme un control sobre l'energia elèctrica que es produeix i consumeix en un determinat país. Es parteix de les següents hipòtesis:

- Existeixen productors bàsics d'electricitat que s'identifiquen per un nom, dels quals interessa la seva producció mitja, producció màxima i data d'entrada en funcionament. Aquests productors bàsics el són d'una de les següents categories: hidroelèctrica, solar, nuclear o tèrmica. D'una central hidroelèctrica ens interessa saber la seva ocupació, capacitat màxima i quantitat de turbines. D'una central solar ens interessa saber la superfície total de panells solars, la mitjana anual d'hores de sol i tipus (fotovoltaica o termodinàmica). D'una central nuclear, ens interessa saber la quantitat de reactors que posseeix, el volum de plutoni consumit i el de residus nuclears que produeix. D'una central tèrmica, ens interessa saber la quantitat de forns que posseeix, el volum de carbó consumit i el volum de la seva emissió de gasos.
- Per motius de seguretat nacional interessa controlar el plutoni del que es proveeix una central nuclear, aquest control es refereix a la quantitat de plutoni que compra a cadascun dels seus possibles subministradors, (nom i país), i que porta un determinat transportista (nom i matrícula), ha de tenir-se en compte que el mateix subministrador pot vendre plutoni a diverses centrals nuclears i que cada port, (un únic port per compra), pot realitzar-lo un transportista diferent.
- Cada dia, els productors lliuren l'energia produïda a una o diverses estacions primàries, les quals poden rebre diàriament una quantitat distinta d'energia de cadascun d'aquests productors. Els productors lliuren sempre el total de la seva producció. Les estacions primàries s'identifiquen pel seu nom i tenen una quantitat de transformadors de baixa i alta tensió i són capçalera d'una o diverses xarxes de distribució.
- Una xarxa de distribució s'identifica per un número de xarxa i només pot tenir una estació primària com capçalera. La propietat d'una xarxa pot ser compartida per diverses companyies elèctriques, a cada companyia elèctrica se li identifica pel seu nom.
- L'energia sobrant en una de les xarxes pot enviar-se a altra xarxa. S'enregistra el volum total d'energia intercanviada entre dues xarxes.
- Una xarxa està composta per una sèrie de línies, cada línia s'identifica per un nombre seqüencial dintre del nombre de xarxa i té una determinada longitud. La menor de les línies possibles abastirà almenys a dos subestacions.
- Una subestació és abastida només per una línia i distribueix a una o diverses zones de servei. Les províncies (codi i nom), es troben dividides en aquestes zones de servei, encara que no pot haver zones de servei que pertanyin a més d'una província. Cada zona de servei pot ser atesa per més d'una subestació.
- En cada zona de servei es desitja enregistrar el consum mig i el nombre de consumidors finals de cadascuna de les següents categories: particulars, empreses i institucions.

# Taules

## Introducció

En aquest apartat estudiarem la forma de definir taules i treballar amb elles. Les taules són les estructures que permeten emmagatzemar les dades de la base de dades.

Una taula és un conjunt de registres. En aquest sentit podem considerar-la com un fitxer que conté *registres*, tants com es vulgui. Cada registre estarà compost per una col·lecció de *camp*s. Cada camp tindrà un *tipus* que indica la classe de dades que pot emmagatzemar. I cada tipus tindrà uns *atributs de tipus* que limiten el domini dels valors permesos, així com una sèrie de restriccions i normes.

Per exemple, podem tenir una taula per a emmagatzemar dades sobre els alumnes matriculats en una certa assignatura. Aquesta taula pot denominar-se "Alumnes" i contindrà registres amb, per exemple, els següents camps.

Camp	Descripció
NIF	NIF
Nom	Nom complet
Cognom1	Primer cognom
Cognom2	Segon cognom
Edat	Edat
Parcial1	Nota obtinguda en el primer parcial
Parcial2	Nota obtinguda en el segon parcial
Practiques	Nota obtinguda en les pràctiques

Observi's que no s'ha definit un camp *NotaMitja* per a emmagatzemar la nota final de l'assignatura, potser la més important. Això és degut al fet que és possible calcular-la d'alguna manera a través de la resta de camps del registre. En una taula no ha emmagatzemar-se informació innecessària. A més, si s'emmagatzemés, caldria recalcularla cada vegada que es modifiqués algun dels camps dels quals depèn, la qual cosa representa una font d'inconsistència. La forma adequada d'obtenir la nota mitja és a través de consultes, que estudiarem més endavant.

## Creació de taules

### Definició de camps

Una vegada decidits els camps que necessitem emmagatzemar en la taula, passarem a crear la taula. Per a això seleccionarem l'opció corresponent en el menú o en la finestra principal de la base de dades. Access ens mostrarà un formulari per a la definició dels camps. Aquesta finestra es denomina "vista disseny" enfront de la "vista de fulla de dades" que permetrà més endavant introduir les dades.

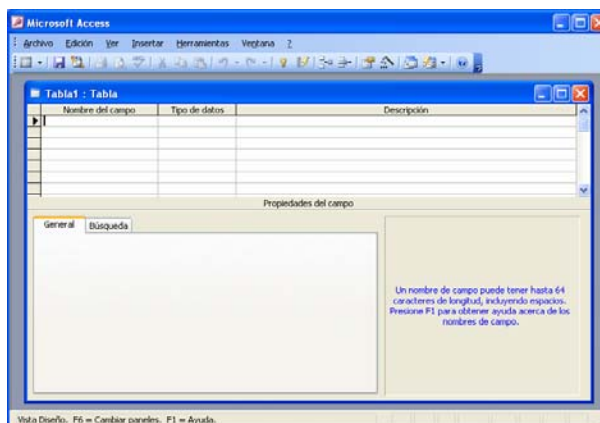


Figura 2. Formulari de definició de camps (vista disseny)

Per a cada camp que definim hem d'assignar-li un nom, un tipus i una descripció. A més, per a cada tipus existeixen una sèrie d'atributs de tipus que permeten descriure amb precisió el domini de valors acceptats pel camp.

Els tipus disponibles són els següents:

- **Text:** Una cadena de caràcters de longitud limitada. La longitud màxima es defineix com un dels atributs, i no pot ser major de 255 caràcters. L'espai ocupat en disc per aquest camp correspon a la longitud màxima definida. Un atribut important del tipus text és si es permet la longitud zero.
- **Memo:** Una cadena de caràcters de longitud il·limitada. L'espai ocupat en disc depèn del text emmagatzemat. Té com desavantatge respecte al tipus Text que no permet algunes operacions (ordre, agrupació...)
- **Numèric:** Un nombre sencer o real. Existeixen diverses mides i precisions: byte, enter, enter llarg, real de precisió simple, real de precisió doble, identificador global. Dintre de cadascun d'aquests subtipus es poden definir atributs de rang i precisió decimal.
- **Data/hora:** Data, hora o ambdós alhora. Existeixen diversos formats de data i hora que poden ser definits com atribut.
- **Moneda:** Un nombre amb format monetari. Es tracta d'un valor numèric real amb format especialitzat per a la manipulació de valors monetaris: abreviatura de la moneda local, separació amb punts i comes a l'estil local,...
- **Autonumèric:** Un comptador autoincrementat. Es tracta d'un valor numèric que el sistema genera automàticament per a cada registre de manera que mai existiran dos registres en la mateixa taula amb el mateix valor en el camp autonumèric.
- **Si/no:** Un valor lògic.
- **Objecte OLE:** Un objecte d'altra aplicació, vinculat mitjançant OLE: so, imatge, vídeo, gràfic...
- **Hipervincle:** Un vincle a un document de la Internet.
- **Assistent per a recerques:** No és pròpiament un tipus de camp. Serveix per a escollir valors per als registres d'altra taula o valors que escriguem en una llista de valors.

Cadascun d'aquests tipus disposa d'atributs propis, per exemple, la precisió decimal i el rang de valors en els camps numèrics, enfront de la longitud de cadena dels camps de tipus text. Però a més existeixen una sèrie d'atributs comuns, que es descriuen a continuació:

- **Format:** defineix mitjançant una expressió el format de les dades emmagatzemades. Per exemple en camps numèrics defineix si s'utilitzen separadors de milers, o si el valor s'interpreta com un percentatge (%); en camps de text pot forçar que els caràcters s'emmagatzemin en majúscules; en camps lògics si s'utilitza SI/NO o VERITABLE/FALS; etc. Al final del capítol s'inclou un annex amb els formats de presentació de dades.
- **Màscara d'entrada:** Força que les dades s'introdueixin en un format adequat. Per exemple per a un telèfon ens pot ser interessant mostrar a l'usuari alguna cosa com "\_\_\_\_ \_", de manera que no pugui introduir caràcters estranys. La màscara d'entrada és un conjunt de tres dades que defineixen els caràcters admesos, el format presentat a l'usuari i el format emmagatzemat. Al final del capítol s'inclou un annex amb informació sobre formats de màscara d'entrada. L'atribut Format preval sobre la màscara d'entrada, de manera que, quan editem les dades del camp, una vegada introduïts seguint una determinada màscara, el contingut del camp es mostrarà amb l'aspecte definit en el Format i no amb el de la màscara.
- **Títol:** És el títol de la columna que apareixerà en la manera de veure dades (quan editem les dades emmagatzemades) per al camp.
- **Valor predeterminat:** Un valor que automàticament introduirà Access en el camp si l'usuari no indica un altre.
- **Regla de validació:** Una condició que ha de complir la dada introduïda perquè sigui acceptat. Per exemple, per a un camp Edat es pot forçar que el valor introduït estigui comprès entre 18 i 65 anys amb una expressió com ">=18 AND <=65".
- **Text de validació:** Un missatge que Access mostrarà a l'usuari quan intenti introduir un valor no permès per una regla de validació. Per a l'exemple anterior podria ser alguna cosa com "L'edat ha de estar compresa entre 18 i 65 anys."
- **Requerit:** Indica si el camp ha de tenir algun valor o es permet deixar-lo en blanc. Per defecte un camp no conté cap valor. No cal confondre un camp buit amb un camp de text de longitud zero. Per això la importància de l'atribut de permetre longitud zero dels camps de tipus text.

- **Indexat:** Estableix si sobre el camp ha de construir-se un índex. Un índex és una estructura manejada per Access que fa que determinades operacions amb el camp siguin més ràpides (ordenacions i recerques), a costa d'un major espai en disc i memòria (per a emmagatzemar l'índex) i que altres operacions siguin més lentes i costoses (insercions i esborrats). Existeixen dues formes de fer índexs: amb i sense duplicats; això permet configurar la possibilitat o no que es repeteixin valors en els registres per a un camp.

Els atributs no comuns són:

- **Mida del camp:** en els camps de text permet indicar la quantitat de caràcters màxim que podrà contenir el camp (entre 0 i 255) el predeterminat és 50. En els camps numèrics indica el tipus de nombre que pot allotjar:
  - **Byte:** Emmagatzema nombres entre 0 i 255 (no admet nombres decimals).
  - **Enter:** Emmagatzema nombres entre -32.768 i 32.767 (no admet nombres decimals).
  - **Enter llarg** (Predeterminat): Emmagatzema nombres entre -2.147.483.648 i 2.147.483.647 (no admet nombres decimals).
  - **Simple:** Emmagatzema nombres entre  $-3,4 \cdot 10^{38}$  i  $3,4 \cdot 10^{38}$ . Admet nombres decimals amb una precisió de 7 dígit.
  - **Doble:** Emmagatzema nombres entre  $-1,79 \cdot 10^{308}$  i  $1,79 \cdot 10^{308}$ . Admet nombres decimals amb una precisió de 15 dígit.
- **Permetre longitud zero:** és propi dels camps de text. Permet cadenes de text de longitud zero ("").
- **Llocs decimals:** és propi dels camps numèrics. Serveix per indicar el nombre màxim de decimals. L'opció predeterminada és Automàtic.
- **Nous valors:** és propi dels camps autonumèrics. Indiquen com s'han de generar els nous valors. Les opcions són incrementalment (1,2,3...) o aleatòriament (a l'atzar).

Per generar les expressions que es poden introduir per a aquestes propietats, Access proporciona un assistent que facilita la recerca d'operadors i funcions. Per accedir a aquest assistent n'hi ha prou amb prémer sobre el botó "... " que apareix a la dreta de l'editor (quant estigui disponible). Potser el més important del generador d'expressions és la possibilitat d'explorar visualment el conjunt de funcions incorporades, entre les quals destaquen:

- Funcions de conversió entre tipus de dades.
- Funcions de data/hora. (Una funció molt útil per a la seva utilització com valor predeterminat és *Fecha()*, que proporciona la data actual.)
- Funcions matemàtiques.
- Funcions de tractament de cadenes.



**Figura 3. Editor visual d'expressions.**

A l'hora d'editar les dades en la vista fulla de dades, Access, per defecte, utilitza un editor de text en el qual es poden escriure les dades. De vegades és útil proporcionar a l'usuari la possibilitat de triar elements d'una llista, en comptes d'haver d'escriure'ls de nou. Per a això, Access permet seleccionar el tipus d'editor que s'utilitzarà per a un camp (atribut *Recerca: mostrar control*):

- **Quadre de text:** un editor normal.
- **Quadre de llista:** una llista de valors permesos.

- **Quadre combinat:** igual que l'anterior, però amb més opcions.
- **Casella de verificació** (només camps sí/no): un requadre per a activar i desactivar.

En el cas del quadre de llista i del quadre combinat, el conjunt de valors proposats pot obtenir-se de tres formes:

- Indicant explícitament una llista de valors separats per símbols de punt i coma.
- A través d'una taula/consulta, i llavors els valors proposats s'obtenen dels emmagatzemats en una taula (aquesta forma l'estudiarem més endavant).
- Indicant una taula, i llavors els valors proposats són el conjunt de camps d'aquesta taula.

## Clau principal

En tota taula sol existir una clau principal (també cridada clau primària o clau mestra). Una clau principal és un camp o conjunt de camps els valors dels quals no es repeteixen i a través dels quals s'identifica de forma única al registre complet. És a dir, que no hi ha dos registres en una taula amb la mateixa clau. En l'exemple dels alumnes, el camp *NIF* pot ser una clau principal, ja que no haurà dos alumnes amb el mateix *NIF*, i a més és possible identificar a un alumne a través del seu *NIF*. El camp *Cognom1* no pot ser clau principal perquè pot haver més d'un alumne amb el primer cognom igual. El conjunt format per (*Cognom1*, *Cognom2*, *Nom*) podria constituir una clau principal (suposant que no existeixen dues persones que es diguin exactament igual), ja que a través del conjunt podem identificar de forma única i inequívoca al registre (a l'alumne).

Sobre un camp que s'empri com clau principal, forçosament deu generar-se un índex sense duplicats, i no s'admetran valors nuls per al camp.

Si no tenim cap camp o conjunt de camps candidat a clau principal, abans d'acabar el disseny Access ens permetrà afegir un camp nou que serveixi com clau. Aquest camp es denominarà *Id.* i serà de tipus autonumèric.

En general, si en una taula no podem trobar un camp o conjunt de camps per establir-los com clau principal, el més probable és que el disseny de la taula no sigui apropiat, i potser hauríem de tornar a plantejar-nos l'estructura de taules que necessita la nostra base de dades.

La declaració d'índexs i claus principals afecta a l'ordre en que es mostren els registres a l'hora d'editar el contingut.

## Guardar el disseny

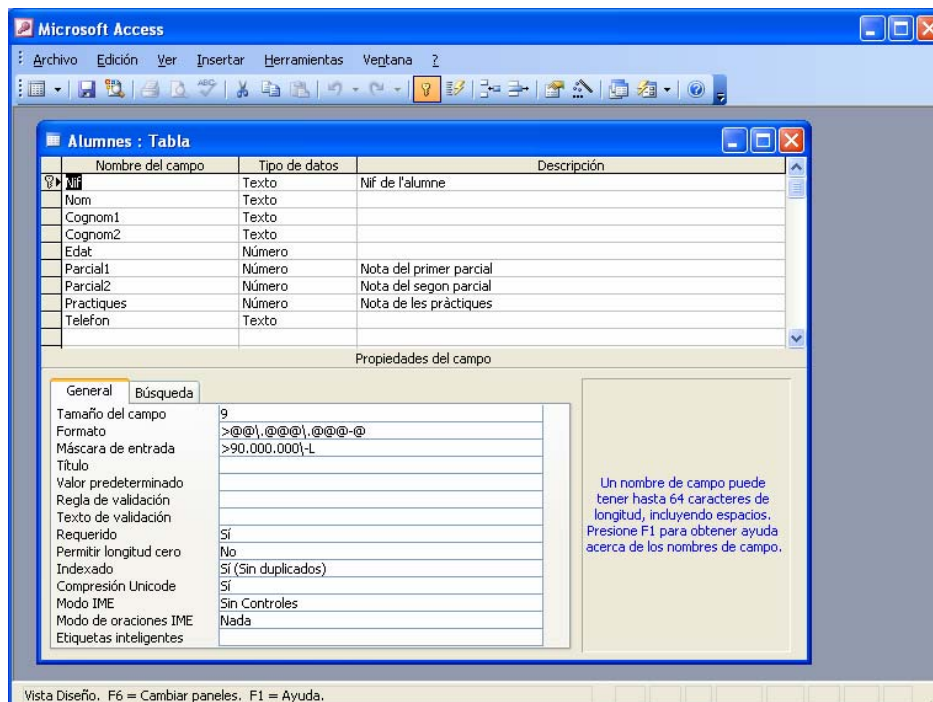
Després d'introduir tots els camps i definir els seus tipus i atributs, podem passar a utilitzar la taula, introduint, modificant i utilitzant les dades que pot emmagatzemar. Abans haurem de guardar la definició de la taula, utilitzant l'opció adequada. Access demanarà un nom per a la nova taula.

Després d'això podem passar a la vista de fulla de dades. Podrem tornar a modificar l'estructura de la taula sempre que vulguem amb la vista disseny. Llavors, abans de tornar a la vista de fulla de dades haurem de guardar la definició. Si ja hi ha dades introduïdes en la taula, Access no permetrà modificar la definició de la taula, o imposarà restriccions, en el cas que els canvis produeixin conflictes entre les noves definicions i les dades ja emmagatzemades (per exemple si intentem canviar el tipus d'un camp de text a tipus numèric i existeixen registres emmagatzemant cadenes no numèriques).

Per a l'exemple de la taula d'alumnes, els camps definits poden tenir les següents característiques:

Camp	Tipus	Descripció	Altres atributs
NIF	Text (9)	NIF	requerit; indexat sense repeticions; màscara d'entrada: <90.000.000\~L; 0; "*">; clau principal.
Nom	Text (40)	Nom complet	requerit
Cognom1	Text (25)	Primer cognom	requerit
Cognom 2	Text (25)	Segon cognom	requerit
Edat	Numèric byte	Edat	>=18 AND <65;
Parcial1	Numèric simple	Nota primer parcial	1 decimal; predeterminat 0; >=0 AND <=10;
Parcial2	Numèric simple	Nota segon parcial	1 decimal; predeterminat 0; >=0 AND <=10;

Practiques	Numèric simple	Nota pràctiques	1 decimal; predeterminat 0; >=0 AND <=2;
------------	----------------	-----------------	--

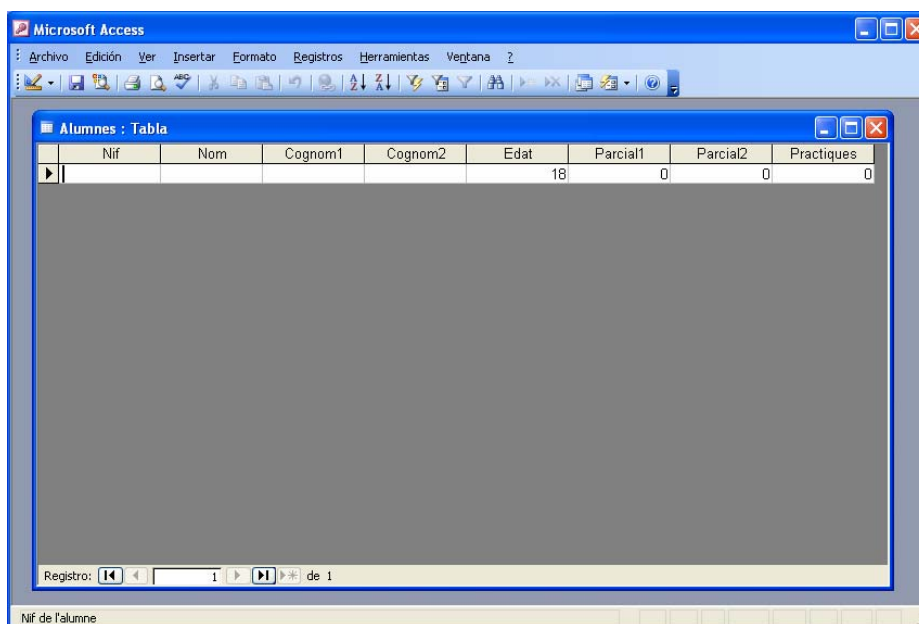


**Figura 4. Definició de camps per a la taula Alumnes.**

## Introducció de dades

Encara que la vista de fulla de dades d'una taula permet a l'usuari la inserció, eliminació i modificació de les dades contingudes en ella, no és la millor forma de fer-lo. El més correcte és utilitzar un formulari sobre la taula: un diàleg més atractiu i còmode que una simple taula, i que estudiarem en el seu moment.

La vista de fulla de dades d'una taula presenta una taula formada per files i columnes. Una fila correspon a un registre i cada columna a un camp. Inicialment la taula estarà buida. Al final de la taula sempre apareix una fila en blanc, destinada a la inserció de nous registres. Part dels camps d'aquesta fila poden aparèixer plens amb els valors per defecte programats en la definició de la taula. En qualsevol cas la modificació de qualsevol camp d'aquesta fila farà que automàticament aparegui una nova.



**Figura 5. Taula d'alumnes buida.**



Com hem vist, els canvis en els registres s'emmagatzemen en disc automàticament, sense que sigui necessari guardar-los explícitament. Els canvis que es realitzen en els camps d'un registre no s'emmagatzemen fins que s'abandona el registre, és a dir fins que es comença a editar altre registre o es tanca la taula. Així, podem cancel·lar els canvis fets en un registre sempre que no hàgim canviat de registre. N'hi ha prou amb prémer la tecla *Esc*. Una pulsació farà que es cancel·li la modificació en el camp actual (si la hi ha) al valor anterior. Llavors, una segona pulsació recuperarà el registre complet (si no s'ha modificat el camp actual, bastarà amb una pulsació per recuperar el registre complet).

Abans d'abandonar un camp, es verifiquen les regles de validació d'aquest camp, de manera que no s'accepta el canvi fins que totes les regles associades al camp es compleixin. A més, abans d'abandonar un registre es comprovaran les regles associades al registre, i no es farà l'escriptura en disc mentre no es compleixi alguna.

És possible utilitzar els comandos tallar, copiar i pegar habituals de Windows, així com seleccionar files i columnes completes, amb les següents pautes:

- Es poden seleccionar columnes completes, rangs de columnes, files completes i rangs de files, així com taules completes; no es poden seleccionar rectangles de diverses files i columnes alhora ni columnes o files saltejades. Per a seleccionar files existeix un polsador de selecció de registre, i per a les columnes cal prémer sobre el requadre de títol.
- Qualsevol selecció pot ser copiada al portapapers, però no esborrada ni tallada: no es poden esborrar columnes completes.
- Es poden eliminar, tallar, copiar i pegar files completes. És possible fer-ho entre taules distintes amb camps diferents, fins i tot des d'altres aplicacions. Llavors Access tractarà d'adaptar el contingut del portapapers als camps de destinació a través de la concordança de noms de camps i els seus tipus. Si no és possible quadrar totes les dades es crearà una taula "Errores de pegado" on aniran a parar les dades que no han pogut ser allotjades.
- Si al pegar diverses files es trenca alguna regla d'integritat, les files prohibides aniran a parar a una taula "Errores de pegado".

Per facilitar l'edició de dades, Access permet personalitzar parcialment la taula d'edició: és possible modificar la grandària de les cel·les i el tipus de lletra utilitzat, així com l'ordre dels registres mostrats i fins i tot fer que no apareguin determinats registres, encara que el més correcte per crear un entorn amigable a l'usuari és crear un formulari, que permet presentar diàlegs molt més còmodes i adequats.

Nif	Nom	Cognom1	Cognom2	Edat	Parcial1	Parcial2	Practiques
1.892.365-L	Ángel	Hormigo	Nieves	18	4	4	2
12.289.462-O	José Antonio	García	Calle	22	5,5	8,5	0
23.489.512-Y	Javier	García	García	21	6	8	1,6
23.784.528-U	Ángel	Calderón	Atanasio	18	0	0	0
23.784.534-J	Patricia	Dominguez	Tello	22	6	4	1,6
23.894.623-B	Carlos	Ramos	Orozco	22	5,2	8	1
23.895.623-K	Germán	Galeano		20	3	4	0
24.578.236-O	Rocío	Mariscal	Muñoz	24	6	6	1,6
28.365.365-J	Pablo	Díaz	Márquez	23	9	6,5	2
34.785.237-I	Isabel	Sánchez	Pérez	25	6	5	0,7
*				18	0	0	0

**Figura 6. Taula d'alumnes amb dades.**

Per finalitzar l'edició en una taula hem de tancar-la. No serà necessari guardar les dades modificades, doncs ja està emmagatzemats en disc. Però si hem modificat el format de presentació (grandària de cel·les, tipus de lletra, ordre de presentació o filtres), serà necessari tornar a guardar la taula, ja que la personalització de les opcions de visualització es considera com part de la definició de la taula. Si no guardem aquests canvis, la pròxima vegada que s'obri la taula en la vista de fulla de dades, aquests es presentaran en el format anterior.

## Interrelacions i integritat.

En una base de dades el més habitual és que existeixin diverses taules relacionades entre si. Per exemple, si tenim una taula de productes d'un magatzem i una taula de proveïdors de productes, és probable que en la de productes existeixi un camp que indiqui quin és el proveïdor que va subministrar aquest producte. En tal cas, hem d'establir unes restriccions sobre ambdues taules de manera que no existeixin productes que es refereixin a proveïdors que no existeixen, i no se'ns permeti eliminar cap proveïdor al que facin referència els productes.

La *integritat referencial* d'una base de dades pretén assegurar que no es donin situacions d'inconsistència com l'anteriorment descrita. Per establir les regles d'integritat referencial en Access hem d'indicar quins camps de quines taules estan relacionats.

L'opció *Relacions* del menú d'eines permet dissenyar un esquema de dependències de forma visual. Disposem d'un escriptori sobre el qual situem les taules de la base de dades, i mitjançant línies, unim els camps dependents. Seguint amb l'exemple dels productes i els proveïdors, suposem que tenim les següents taules:

PRODUCTES	PROVEÏDORS
IdProducte	IdProveïdor
Nom	Nom
Proveïdor	Adreça
Quantitat	NIF
	FAX

La taula de productes no ha d'emmagatzemar la informació completa dels proveïdors, ja que a l'haver més d'un producte del mateix proveïdor estaríem emmagatzemant informació innecessària (redundant). Així, només s'emmagatzema un petit codi *Proveïdor*, que es refereix a un dels registres de la taula *PROVEÏDORS*: aquell que el seu *IdProveïdor* coincideixi. És a dir, que el camp *PRODUCTES.Proveïdor* fa referència a *PROVEÏDORS.IdProveïdor*. Per a no perdre informació, ara hauríem d'assegurar-nos de no eliminar entrades de *PROVEÏDORS* que estiguin referenciades per entrades de *PRODUCTES*.

Existeixen diverses possibilitats per mantenir la integritat referencial, corresponents a les diverses respostes que podem fer a la pregunta: què passa si intentem esborrar un proveïdor?

- a) **Ho ignorem.** Com hem vist no és recomanable.
- b) **No ho permetem** (a menys que abans esborrem tots els registres associats).
- c) **Esborrem** també tots els productes relacionats.

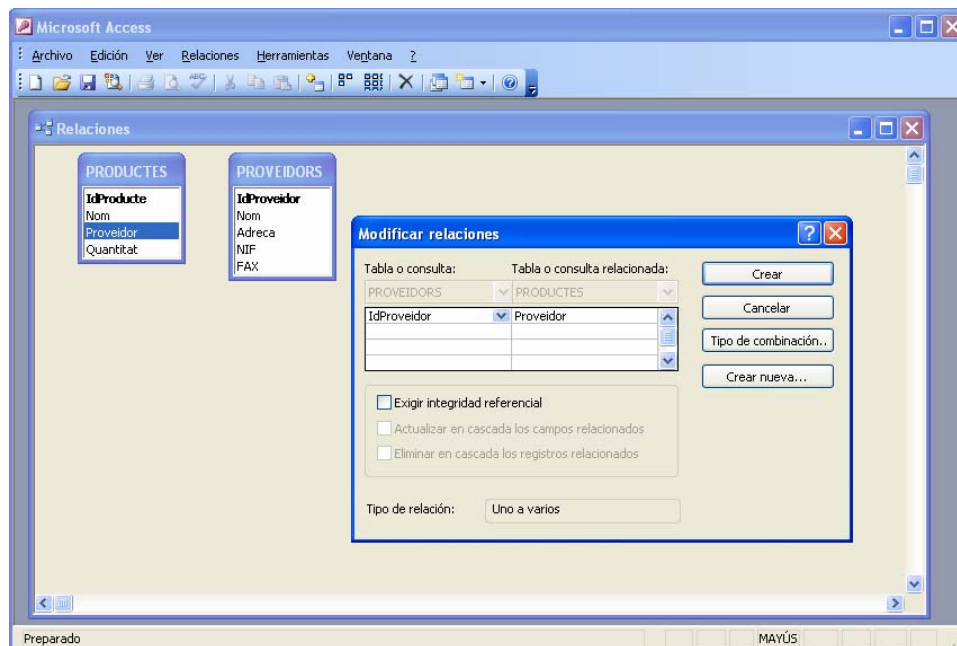
En aquest cas l'opció recomanable és la b), ja que per a l'exemple no té sentit que perdem la informació dels nostres productes a l'eliminar el proveïdor; encara que en molts altres casos l'adequada pot ser la c). En general, mai l'a).

També podem fer-nos la pregunta: què passa si intentem canviar l'identificador d'un proveïdor?

- a) **Ho ignorem** i permetem el canvi.
- b) **No ho permetem** (a menys que abans modifiquem tots els registres associats).
- c) **Modifiquem** també tots els productes relacionats.

En aquest cas l'opció recomanable és la c), ja que és la que resulta transparent a l'usuari.

A l'establir una relació de dependència en Access entre dues taules és necessari especificar quina de les possibilitats anteriors és la que volem aplicar.



**Figura 7. Assignació de regles d'integritat referencial.**

Existeixen tres opcions per a definir la integritat referencial:

- **Exigir integritat referencial:** activa o desactiva la integritat referencial entre els camps de les taules seleccionades. Les altres opcions no poden activar-se si no s'activa aquesta.
- **Actualitzar en cascada:** si s'activa, fa actualitzacions en cascada; si no, no permet les actualitzacions.
- **Eliminar en cascada:** si s'activa, fa eliminacions en cascada; si no, no permet les eliminacions.


## Exercici 5: creació de taules

Suposem que volem dissenyar una base de dades per gestionar una empresa de venda al públic.


L'empresa disposa de diverses botigues distribuïdes per diferents ciutats i es nodreix d'uns magatzems que també estan a distintes ciutats. A més, cal conèixer les existències dels productes tant en les botigues com als magatzems. Per últim, hem de desar informació sobre les comandes que les botigues fan als magatzems. Per això cal una sèrie de taules que continguin la informació.

Crear les següents taules:


### 1. BOTIGUES: Taula que guardarà informació de les distintes botigues

Camp	Tipus	Descripció	Altres atributs
Bid 	Text (3)	Identificador de botiga	El primer caràcter ha de ser una B i els altres dos seran números (B02, B12, etc.). Per introduir un identificador, només cal teclejar el número. Màscara: \B00 Títol: Identificador Requerit Clau principal
Bdir	Text (50)	Adreça de la botiga	Títol: Adreça Requerit
Bloc	Text (15)	Ciutat de la botiga	Títol: Localitat Requerit



### 2. MAGATZEMS: Taula que desarà informació dels distints magatzems

Camp	Tipus	Descripció	Altres atributs
Mid 	Text (3)	Identificador de magatzem	Màscara: \M00 Títol: Identificador Requerit Clau principal
Mdir	Text (50)	Adreça del magatzem	Títol: Adreça Requerit
Mloc	Text (15)	Ciutat del magatzem	Títol: Localitat Requerit



### 3. PRODUCTES: Taula que desarà informació sobre els productes que es venen

Camp	Tipus	Descripció	Altres atributs
Pid 	Text (3)	Identificador de producte	Màscara: \P00 Títol: Identificador Requerit Clau principal
Descripcio	Text (20)	Descripció del producte	Títol: Descripció Requerit
Pes	Numèric Enter	Pes del producte	Format: # " Kg" Regla de validació: >=0 Y <=1000 Text de validació: Entre 0 i 1.000 Requerit
Preu	Numèric Enter llarg	Preu del producte	Format: Moneda Regla de validació: >=0 Y <=6000 Text de validació: Entre 0 i 6.000 Requerit


4. MEXISTENCIES: Taula que desarà informació sobre les existències dels productes en els magatzems

Camp	Tipus	Descripció	Altres atributs
Mid 	Text (3)	Identificador de magatzem	Màscara: \M00 Títol: Id. de Magatzem Requerit Clau principal
Pid 	Text (3)	Identificador de producte	Màscara: \P00 Títol: Id. de Producte Requerit Clau principal
Mquant	Numèric Enter	Quantitat d'unitats en stock	Títol: Quantitat Regla de validació: >=0 Text de validació: El valor ha de ser positiu Requerit

5. BEXISTENCIES: Taula que desarà informació sobre les existències dels productes en les botigues

Camp	Tipus	Descripció	Altres atributs
Bid 	Text (3)	Identificador de botiga	Màscara: \B00 Títol: Id. de Botiga Requerit Clau principal
Pid 	Text (3)	Identificador de producte	Màscara: \P00 Títol: Id. de Producte Requerit Clau principal
Bquant	Numèric Enter	Quantitat d'unitats en stock	Títol: Quantitat Regla de validació: >=0 Text de validació: El valor ha de ser positiu Requerit

6. COMANDES: Taula que desarà informació sobre les comandes que les botigues realitzen als magatzems

Camp	Tipus	Descripció	Altres atributs
Cid 	Text (3)	Identificador de comanda	Màscara: \C00 Títol: Id. de Comanda Requerit Clau principal
Mid	Text (3)	Identificador de magatzem	Màscara: \M00 Títol: Id. de Magatzem Requerit
Bid	Text (3)	Identificador de botiga	Màscara: \B00 Títol: Id. de Botiga Requerit
Pid	Text (3)	Identificador de producte	Màscara: \P00 Títol: Id. de Producte Requerit
Quant	Numèric Enter	Quantitat d'unitats sol·licitades	Títol: Quantitat Regla de validació: >=0 Text de validació: El valor ha de ser positiu Requerit
Dcom	Data/Hora	Data de la comanda	Format: Data curta Títol: Data Valor predeterminat: =Fecha() Regla de validació: <=Fecha() Text de validació: No pot ser una data futura Requerit

## Exercici 6: relacions

Establir les relacions existents entre les taules.

## Exercici 7: introducció de dades

Introduir les dades corresponents a les taules creades en l'exercici anterior. Comprovar que les regles de validació definides funcionen correctament procurant introduir valors il·legals, deixar camps buits, etc.

### BOTIGUES

Bid	Bdir	Bloc
B01	Paseo de Gracia, 1	Barcelona
B02	Gran Vía, 9	Bilbao
B03	C/ Sagasta, 25	Madrid
B04	Av. República Argentina, 25	Sevilla
B05	Paseo Recoletos, 3	Madrid
B06	Av. Andalucía, 36	Huelva

### MAGATZEMS

Mid	Mdir	Mloc
M02	Polígono Sur s/n	Huelva
M03	Carretera Murcia, 124	Murcia
M05	Polígono Alhambra, 322	Granada

### PRODUCTES

Pid	Descripció	Pes	Preu
P03	Televisor M1	27 Kg	336 €
P04	Televisor M2	35 Kg	570 €
P10	Microondas mc1	12 Kg	102 €
P12	Secador sx	1 Kg	15 €
P13	Lavadora m100	95 Kg	276 €
P25	Vídeo v33	8 Kg	270 €
P26	Vídeo v45	10 Kg	222 €

### COMANDES

Cid	Mid	Bid	Pid	Quant	Dcom
C10	M02	B01	P03	100	17/02/02
C11	M02	B03	P04	25	17/02/02
C12	M03	B03	P10	50	18/02/02
C13	M03	B03	P25	10	18/02/02
C14	M02	B04	P04	15	20/02/02
C15	M05	B05	P26	30	21/02/02

### MEXISTÈNCIES

Mid	Pid	Mquant
M02	P03	250
M02	P04	125
M02	P10	710
M02	P25	25
M02	P26	100
M02	P12	25
M02	P13	75
M03	P04	210
M03	P10	14
M03	P25	50
M05	P03	40
M05	P25	100
M05	P26	12

### BEXISTÈNCIES

Bid	Pid	Bquant
B01	P03	17
B01	P04	130
B01	P10	10
B01	P25	22
B01	P26	42
B02	P03	4
B02	P10	80
B02	P25	12
B02	P26	23
B03	P10	125
B04	P03	20
B04	P04	89
B05	P10	9
B05	P25	12
B05	P26	16

## Exercici 8: creació de taules

1. ALUMNES: Taula que desarà informació dels alumnes:

Camp	Tipus	Descripció	Altres atributs
Nif	Text (9)	Nif de l'alumne	Format: >@@\.\@@@\.\@@@-@ Màscara: >90.000.000\L Clau principal. Requerit.
Nom	Text (40)		Requerit
Cognom1	Text (25)		Requerit Permetre longitud zero: sí
Cognom2	Text (25)		Requerit Permetre longitud zero: sí
Edat	Numèric Byte		Valor predeterminat: 18 Regla de validació: >=18 Y <=65 Text de validació: Entre 18 i 65
Parcial1	Numèric Simple	Nota del primer parcial	Posicions decimals: 1 Regla de validació: >=0 Y <=10 Text de validació: Entre 0 i 10
Parcial2	Numèric Simple	Nota del segon parcial	Posicions decimals: 1 Regla de validació: >=0 Y <=10 Text de validació: Entre 0 i 10
Practiques	Numèric Simple	Nota de les pràctiques	Posicions decimals: 1 Regla de validació: >=0 Y <=2 Text de validació: Entre 0 i 2

2. PERSONES: Taula que desarà informació de les persones del centre:

Camp	Tipus	Descripció	Altres atributs
Id	Autonumèric	Número identificador	Clau principal
Nom	Text (40)		Requerit
Cognoms	Text (31)		Requerit
Carrec	Text (15)	Càrrec que ocupa en el centre	Requerit

3. HISTORIA: Taula que desarà informació de la quantitat d'alumnes matriculats cada any:

Camp	Tipus	Descripció	Altres atributs
Anyo	Text (4)		
Numero	Numèric Enter	Quantitat d'alumnes matriculats cada any	

## Exercici 9: introducció de dades

Introduir les dades corresponents a la taula creada en l'exercici anterior. Comprovar que les regles de validació definides funcionen correctament intentant introduir valors il·legals, deixar camps buits, etc.

### ALUMNES

Nif	Nom	Cognom1	Cognom2	Edat	Parcial1	Parcial2	Practiques
28.365.365-J	Pablo	Díaz	Márquez	23	9	6,5	2
23.895.623-K	Germán	Galeano		20	3	4	0
23.489.512-Y	Javier	García	García	21	6	8	1,6
23.894.623-B	Carlos	Ramos	Orozco	22	5,2	8	1
23.784.534-J	Patricia	Domínguez	Tello	22	6	4	1,6
1.892.365-L	Ángel	Hormigo	Nieves	18	4	4	2
23.784.528-U	Ángel	Calderón	Atanasio	18	0	0	0
34.785.237-I	Isabel	Sánchez	Pérez	25	6	5	0,7
24.578.236-O	Rocío	Mariscal	Muñoz	24	6	6	1,6
12.289.462-O	José Antonio	García	Calle	22	5,5	8,5	0

## Annex 1: formats de presentació de dades

En tots els casos existeixen diversos formats predefinits d'ús habitual; a més, si el format desitjat no coincideix amb cap dels predefinits, és possible especificar un format personalitzat mitjançant l'ús d'una sèrie de símbols. Les següents taules mostren els identificadors de formats predefinits i els símbols emprats en camps de distints tipus.

### Camps de tipus data/hora

Valor	Descripció
Data general	(Predeterminat) Si el valor és només una data, no es mostra cap hora; si el valor és només una hora, no es mostra cap data. Aquest valor és una combinació dels valors de Data curta i Hora llarga. Exemples: 3/4/93, 05:34:00 PM i 3/4/93 05:34:00 PM.
Data llarga	Igual que el valor de Data llarga del quadre de diàleg Propietats de Configuració regional del Tauler de control de Windows. Exemple: Dissabte, 3 d'Abril de 1993.
Data mitjana	Exemple: 3-Abr-93.
Data curta	Igual que el valor de Data curta del quadre de diàleg Propietats de Configuració regional del Tauler de control de Windows. Exemple: 3/4/93. Atenció: El format Data curta assumeix que les dates compreses entre el 1/1/00 i el 31/12/29 són dates del segle XXI (és a dir, que els anys estan entre el 2000 i el 2029). Les dates compreses entre el 1/1/30 i el 31/12/99 s'assumeix que són dates del segle XX (és a dir, que els anys estan entre 1930 i 1999).
Hora llarga	Igual que el valor de la fitxa Hora del quadre de diàleg Propietats de Configuració regional del Tauler de control de Windows. Exemple: 5:34:23 PM.
Hora mitjana	Exemple: 5:34 PM.
Hora curta	Exemple: 17:34.

Símbol	Descripció
:	Separador d'hora. Els separadors s'estableixen en el quadre de diàleg Propietats de Configuració regional del Tauler de control de Windows.
/	Separador de data.
c	Igual que el format predefinit Data general.
d	Dia del mes en un o dos dígits numèrics, segons sigui necessari (1 a 31).
dd	Dia del mes en dos dígits numèrics (01 a 31).
ddd	Tres primeres lletres del dia de la setmana (Lun a Dom)
dddd	Nom complet del dia de la setmana (Lunes a Domingo).
dddddd	Igual que el format predefinit Hora curta.
dddddd	Igual que el format predefinit Data llarga.
s	Dia de la setmana (1 a 7).
ss	Setmana de l'any (1 a 53).
m	Mes de l'any en un o dos dígits numèrics, segons sigui necessari (1 a 12).
mm	Mes de l'any en dos dígits numèrics (01 a 12).
mmm	Tres primeres lletres del mes (Ene a Dic).
mmmm	Nom complet del mes (Enero a Diciembre).
t	Data mostrada com trimestre de l'any (1 a 4).
a	Nombre de dia de l'any (1 a 366).
aa	Dos últims dígits de l'any (01 a 99).
aaaa	Any complet (0100 a 9999).
h	Hora en un o dos dígits, segons sigui necessari (0 a 23).
hh	Hora en dos dígits (00 a 23).
n	Minut en un o dos dígits, segons sigui necessari (0 a 59).
nn	Minut en dos dígits (00 a 59).
s	Segon en un o dos dígits, segons sigui necessari (0 a 59).
ss	Segon en dos dígits (00 a 59).
tttt	Igual que el format predefinit Hora llarga.
AM/PM	Relotge de 12 hores amb les lletres majúscules "AM" o "PM", segons sigui apropiat.
am/pm	Relotge de dotze hores amb les lletres minúscules "am" o "pm", segons sigui apropiat.
A/P	Relotge de dotze hores amb la lletra majúscula "A" o "P", segons sigui apropiat.
a/p	Relotge de dotze hores amb la lletra minúscula "a" o "p", segons sigui apropiat.
AMPM	Relotge de dotze hores amb l'indicador de matí/tarda apropiat establert en el quadre de diàleg Propietats de Configuració regional del Tauler de control de Windows.



**NOTA:** Els formats personalitzats es mostren segons el valor especificat en el quadre de diàleg Propietats de Configuració regional del Tauler de control de Windows. Els formats personalitzats incoherents amb els valors especificats en el quadre de diàleg Propietats de Configuració regional s'ignoren. Per a agregar una coma o altre separador al format personalitzat s'ha d'incloure el separador entre cometes. Per exemple: d mmm ", " aaaa.

## Camps de tipus numèric i moneda

Valor	Descripció
Número general	(Predeterminat) Mostra el número tal com s'introdueix.
Moneda	Utilitza el separador de milers; mostra els números negatius entre parèntesi; el valor predeterminat de la propietat LugaresDecimales és 2.
Fix	Mostra almenys un dígit; el valor predeterminat de la propietat LugaresDecimales és 2.
Estàndard	Utilitza el separador de milers; el valor predeterminat de la propietat LugaresDecimales és 2.
Percentatge	Multiplica el valor per 100 i annexa un signe de percentatge; el valor predeterminat de la propietat LugaresDecimales és 2.
Científic	Utilitza la notació científica estàndard.

Els formats numèrics personalitzats poden tenir entre una i quatre seccions amb signes de punt i coma (;) com separador de llista. Cada secció conté l'especificació de format per a un tipus de número diferent:

1. El format per als números positius.
2. El format per als números negatius.
3. El format per als valors zero.
4. El format per als valors nuls.

Per exemple, el següent format Moneda personalitzat conté quatre seccions separades per signes de punt i coma i utilitza un format diferent per a cada secció:

\$,00[Verde];(\$,00)[Rojo];"Cero";"Null"

Si s'utilitzen diverses seccions però no s'especifica un format per a cada secció, les entrades per a les quals no hi ha cap format no mostraran gens o prendran com valor predeterminat el format de la primera secció.

Símbol	Descripció
, (coma)	Separador decimal. Els separadors s'estableixen fent doble clic en Configuració regional en el Tauler de control de Windows.
. (punt)	Separador de milers.
0	Marcador de posició de dígit. Mostra un dígit o 0.
#	Marcador de posició de dígit. Mostra un dígit o res.
\$	Mostra el caràcter literal "\$".
%	Percentatge. El valor es multiplica per 100 i se li annexa un signe de percentatge.
E- o e-	Notació científica amb un signe menys (-) al costat dels exponents negatius i gens al costat dels exponents positius. Aquest símbol ha d'utilitzar-se amb altres símbols, com en l'exemple 0,00E-00 o 0,00E00.
E+ o e+	Notació científica amb un signe menys al costat dels exponents negatius i un signe més (+) al costat dels exponents positius. Aquest símbol ha d'utilitzar-se amb altres símbols, com en l'exemple 0,00E+00.

## Camps de tipus text i memo

Símbol	Descripció
@	Es necessita un caràcter de text (ja sigui un caràcter o un espai).
&	No es necessita un caràcter de text.
<	Convertir tots els caràcters a minúscules.
>	Convertir tots els caràcters a majúscules.

Els formats personalitzats per als camps Text i Memo poden tenir fins dues seccions: la primera conté el format per als camps amb text i la segona el format per als camps amb cadenes de longitud zero i valors nuls.

Per exemple, per a obtenir que aparegui la paraula "Ninguno" quan no existeixi cap cadena en el camp, pot utilitzar-se el format personalitzat @;" Ninguno".

## Camps de tipus sí/no

En les últimes versions d'Access, els camps lògics es mostren a l'usuari de forma gràfica com una casella de verificació, encara que també pot mostrar-se com un text. En el primer cas, qualsevol format especificat és ignorat.

Els formats predefinits són Sí/No, Veritable/Fals i Activat/Desactivat.

Si, Veritable i Activat són equivalents entre si, igual que ho són No, Fals i Desactivat. Si s'especifica un format predefinit i, a l'editar les dades, s'introdueix un valor equivalent, es mostrarà el format predefinit del valor equivalent.

Per a formats personalitzats es poden utilitzar expressions que continguin fins tres seccions:

1. Aquesta secció no té cap efecte en el tipus de dades Sí/No. No obstant, es requereix un signe de punt i coma (;) com marcador de posició.
2. El text per a mostrar en lloc dels valors Sí, Veritable o Activat.
3. El text per a mostrar en lloc dels valors No, Fals o Desactivat.

## Annex 2: format de màscares d'entrada

L'expressió per al format de màscara d'entrada pot contenir fins tres seccions separades per signes de punt i coma (;).

1. Especifica la màscara d'entrada pròpiament aquesta, per exemple, !(999) 999-9999. Els caràcters que es poden utilitzar per a definir la màscara d'entrada es mostren més endavant.
2. Especifica si Access emmagatzema els caràcters de visualització literals en la taula a l'introduir dades. Si s'utilitza 0 per a aquesta secció, tots els caràcters de visualització literals (per exemple, el parèntesi d'una màscara d'entrada d'un nombre de telèfon) s'emmagatzemen amb el valor; si s'introdueix 1 o es deixa aquesta secció en blanc, només s'emmagatzemen els caràcters introduïts en el control.
3. Especifica el caràcter que Access mostra per a l'espai en el qual l'usuari ha d'escriure un caràcter en la màscara d'entrada. Per a aquesta secció pot utilitzar qualsevol caràcter. Per exemple per mostrar un asterisc, s'escriu entre cometes ("\*").

Es poden definir màscares d'entrada mitjançant els següents caràcters:

Caràcter	Descripció
0	Dígit (0 a 9, entrada obligatòria, signes més [+] i menys [-] no permesos).
9	Dígit o espai (entrada no obligatòria, signes més i menys no permesos).
#	Dígit o espai (entrada no obligatòria; els espais es mostren en blanc en la manera Edició, però s'eliminen quan es guarden les dades; signes més i menys permesos).
L	Lletra (A a Z, entrada obligatòria).
?	Lletra (A a Z, entrada opcional).
A	Lletra o dígit (entrada obligatòria).
a	Lletra o dígit (entrada opcional).
&	Qualsevol caràcter o un espai (entrada obligatòria).
C	Qualsevol caràcter o un espai (entrada opcional).
. , ; - /	Marcador de posició decimal i separadors de milers, hora i data (el caràcter depèn del valor del quadre de diàleg Propietats de Configuració regional en el Tauler de control de Windows).
<	Fa que tots els caràcters es converteixin a minúscules.
>	Fa que tots els caràcters es converteixin a majúscules.
!	Fa que la màscara d'entrada es mostri de dreta a esquerra, en lloc de mostrar-se d'esquerra a dreta. Els caràcters introduïts en la màscara sempre s'emplenen d'esquerra a dreta. Pot incloure el signe d'exclamació en qualsevol lloc de la màscara d'entrada.
\	Fa que el caràcter següent es mostri com un caràcter literal (per exemple, \A es mostra només com A).
Contrasenya	Quan establim la propietat <b>Màscara d'entrada (InputMask)</b> a la paraula Contrasenya, es crea un quadre de text d'entrada de contrasenya. Qualsevol caràcter escrit en aquest quadre de text s'emmagatzema com a tal, però es mostra com un asterisc (*).

# Consultes

## Introducció a les consultes

Les consultes són operacions que es realitzen sobre les dades d'una base de dades. Aquestes operacions poden ser de diversos tipus:

- **Consultes de selecció de dades:** permeten recuperar les dades emmagatzemades en les taules en el format i ordre adequats. A més permeten filtrar i agrupar la informació. El resultat d'aquestes consultes consisteix en una taula "virtual": una taula que físicament no ocupa espai (perquè treballa sobre les dades originals de les taules sobre les quals es defineix), però que permet ser usada de la mateixa forma que una taula real.

Dintre de les consultes de selecció podem ressaltar alguns grups importants:

- **Consultes de selecció simple:** permet filtrar registres i afegir o descartar camps dels registres. S'utilitzen per crear "vistes": consultes que s'utilitzen com taules per restringir l'accés a les dades a determinats usuaris.
- **Consultes d'unió:** permeten relacionar les dades de distintes taules a través de camps clau.
- **Consultes d'agrupament:** permeten obtenir resultats estadístics de conjunts de registres, com mitjanes d'un camp, totals, etc.
- **Consultes d'inserció de dades:** permeten afegir registres a una taula.
- **Consultes de modificació:** permeten modificar els valors dels camps dels registres d'una taula.
- **Consultes d'eliminació:** permeten eliminar registres d'una taula.
- **Consultes de creació de taula:** permeten crear noves taules els camps i registres de les quals s'obtenen a partir dels emmagatzemats en altres taules.

Les consultes a les dades de les bases de dades es fan a través dels denominats "llenguatges de consulta". El més utilitzat d'aquesta mena de llenguatges és el SQL (*Structured Query Language*). Access permet realitzar consultes en SQL. Però, donada la filosofia visual seguida en Access, també es permet la definició de consultes a través d'una interfície visual QBE (*Query By Example*). Ambdós mètodes són compatibles i intercanviables: es pot passar d'un a altre sense problemes. No obstant això, SQL és un estàndard que apareix en multitud de sistemes sense variacions, mentre que el desenvolupament visual depèn del fabricant.

## Introducció al SQL

Una consulta SQL està composta per una instrucció SQL que defineix aquesta consulta. Es tracta d'un comando que pot ocupar quantes línies de text es desitgi, acabat en punt i coma (;). SQL és un llenguatge senzill i intuïtiu: les consultes s'assemblen al llenguatge natural.

Existeixen algunes paraules reservades, com en qualsevol llenguatge: SELECT, INSERT, DELETE, UPDATE, SET, WHERE, IN, DISTICT, GROUP, ORDER, BY, etc.

## Consultes de selecció simple

La consulta més simple possible consisteix en la selecció de camps i registres d'una taula. S'identifiquen els camps que ens interessen i una condició que han de complir els registres seleccionats. El resultat és una taula que és un subconjunt de l'original.

El format genèric d'aquest tipus de consultes és:

```
SELECT <llista de camps> FROM <taula> WHERE <condició>;
```

Aquesta instrucció recupera certs camps dels registres d'una taula que verifiquen una condició. La clàusula **WHERE** és opcional. Si s'omet, se seleccionen tots els registres (se suposa que la condició és sempre veritable).

```
SELECT <llista de camps> FROM <taula>;
```

Si ens interessen tots els camps podem utilitzar el símbol \* per identificar a la llista completa:

```
SELECT * FROM <taula> WHERE <condició>;
```

Si no, podem especificar diversos camps identificant-los pels seus noms i separant-los per comes (,).

```
SELECT camp1, camp2, ..., campN FROM <taula> WHERE <condició>;
```

Suposem la taula d'Alumnes definida en el capítol "Taules", la qual hem guardat amb el nom "Alumnes".

Camp	Descripció
NIF	NIF
Nom	Nom complet
Cognom1	Primer cognom
Cognom2	Segon cognom
Edat	Edat
Parcial1	Nota obtinguda en el primer parcial
Parcial2	Nota obtinguda en el segon parcial
Practiques	Nota obtinguda en les pràctiques

Podem definir les següents consultes d'exemple:

Consulta	SQL
Seleccionar els alumnes majors de 25 anys.	SELECT * FROM Alumnes WHERE Edat>=25;
Seleccionar els cognoms i nom dels que hagin aprovat els dos parcials.	SELECT Cognom1, Cognom2, Nom FROM Alumnes WHERE (Parcial1>=5) AND (Parcial2>=5);
Generar un llistat amb els noms i cognoms dels alumnes i les seves notes.	SELECT Nom, Cognom1, Cognom2, Parcial1, Parcial2, Practiques FROM Alumnes;

És possible consultar, des d'una base de dades, una taula que pertanyi a altra base de dades. En aquest cas utilitzarem la sintaxi:

```
SELECT <llista de camps> FROM <taula> IN <base de dades>;
```

La clàusula **IN** permet especificar altra base de dades com origen de la taula. Exemple:

Consulta	SQL
Seleccionar els alumnes majors de 25 anys, suposant que la taula <i>Alumnes</i> està en altra base de dades que es diu <i>C:\ESCOLA\REGISTRE.MDB</i>	SELECT * FROM Alumnes IN "C:\ESCOLA\REGISTRE.MDB" WHERE Edat>=25;

Aquesta estructura permet també accedir a dades que es trobin emmagatzemats en altres bases de dades que no siguin Microsoft Access, sempre que Access es trobi correctament instal·lat i configurat.

## Addició de camps

Podem generar consultes en les quals apareguin nous camps. Per exemple ens pot interessar una consulta en la qual es mostri la nota mitja obtinguda pels alumnes. En tal cas podem utilitzar la sintaxi "<expressió> AS " per a cada columna afegida com si es tractés d'un camp més de la taula:

```
SELECT <llista de camps>, <expressió> AS <nom camp> FROM <taula> WHERE <condició>;
```

Alguns exemples:

Consulta	SQL
Obtenir els cognoms junt a la nota mitja, suposant que la mitja dels parcials és el 80% i la resta s'obté amb les pràctiques.	SELECT Cognom1, Cognom2, ((Parcial1 + Parcial2) /2) * 0.8 + Practiques AS NotaMitja FROM Alumnes;
Obtenir els noms complets dels alumnes junt al seu NIF.	SELECT Nom & " " & Cognom1 & " " & Cognom2 AS NomCompleto, NIF FROM Alumnes;

## Operadors i expressions

Les expressions en SQL són semblants a les utilitzades en la majoria dels llenguatges.

Operador	Significat	Operador	Significat
+	Suma aritmètica	"	Delimitador de cadenes
-	Resta aritmètica	&	Concatenació de cadenes
*	Producte aritmètic	=	Comparador igual
/	Divisió aritmètica	<>	Comparador distint
mod	Mòdul	>	Comparador major
AND	AND lògic	<	Comparador menor
OR	OR lògic	>=	Comparador major o igual
NOT	Negació lògica	<=	Comparador menor o igual
XOR	OR exclusiu lògic	( )	Delimitadores de precedència

No obstant això mereixen destacar els següents:

Operador	Significat
IS NULL	Comparador amb valor nul. Indica si un camp s'ha deixat en blanc. Exemple: Alumnes dels quals es desconeix la seva edat: <code>SELECT * FROM Alumnes WHERE Edat IS NULL;</code>
IS NOT NULL	Comparador amb valor no nul. Indica si un camp conté un valor, i no s'ha deixat en blanc. Exemple: Alumnes dels quals no es desconeix la seva edat: <code>SELECT * FROM Alumnes WHERE Edat IS NOT NULL;</code>
LIKE	Comparador de semblança. Permet realitzar una comparació de cadenes utilitzant caràcters comodins: ? = Un caràcter qualsevol * = Qualsevol combinació de caràcters (inclòs cap caràcter) Exemple: Alumnes el cognom dels qual comença per "A": <code>SELECT * FROM Alumnes WHERE Nombre LIKE "A*";</code>
BETWEEN...AND	Comparador de pertinença a rang. Per exemple, alumnes l'edat dels quals estigui compresa entre 18 i 20: <code>SELECT * FROM Alumnes WHERE Edat BETWEEN 18 AND 20;</code>
[ ]	Delimitadors d'identificadors. Serveixen per delimitar els noms dels objectes (camp, taules, etc.) quan aquests incloguin espais. Exemple: suposem una taula anomenada "Alumnes nous": <code>SELECT * FROM [Alumnes nous];</code>

## Valors repetits

Una consulta de selecció pot recuperar registres idèntics. Suposem la següent taula Alumnes:

NIF	Nom	Cognom1	Cognom2	Edat	Parcial1	Parcial2	Practiques
41.486.691-W	Juan	Gómez	Jurado	23	5	8	2
8.353.842-C	Alberto	García	Díaz	22	7	7	2
23.786.354-H	Juan	Izquierdo	Donoso	23	5	5	1
34.753.485-J	José	López	López	19	9	9	2
...							

La següent consulta de selecció:

```
SELECT Nom, Edat, Parcial1 FROM Alumnes;
```

Generarà dos registres amb les dades: Juan, 23, 5. Per evitar obtenir registres repetits, podem utilitzar el modificador **DISTINCT**:

```
SELECT DISTINCT Nom, Edat, Parcial1 FROM Alumnes;
```

Ara la consulta no retornarà registres repetits. Existeix altre modificador, **DISTINCTROW**. A diferència de l'anterior, **DISTINCTROW** no té en compte registres que estiguin completament duplicats en la taula d'origen (i no només per als camps seleccionats).

## Ordenació de registres

SQL permet especificar que els registres seleccionats es mostrin ordenats per algun o alguns dels camps seleccionats, ascendent o descendent. Per a això es disposa de la paraula reservada **ORDER BY**, amb el següent format:

```
SELECT <llista de camps> FROM <taula> WHERE <condició> ORDER BY <llista de camps per ordenar>;
```

La llista de camps per ordenar ha de ser un subconjunt de la llista de camps seleccionats. Per especificar un ordre invers (decreixent) s'emptra la clàusula **DESC** que pot ser inclosa després del nom del camp pel qual s'ordena de forma descendent. De la mateixa forma la clàusula **ASC** ordena de forma ascendent, encara que no és necessari especificar-la, ja que és l'opció per defecte. Exemples:

Consulta	SQL
Obtenir un llistat d'alumnes ordenats per cognoms.	SELECT * FROM Alumnes ORDER BY Cognom1, Cognom2, Nombre;
Obtenir els alumnes amb el primer parcial aprovat, començant per les millors notes.	SELECT * FROM Alumnes WHERE Parcial1 >= 5 ORDER BY Parcial1 DESC;
Obtenir els cognoms i les notes dels parcials dels alumnes que han tret millor nota en el primer parcial que en el segon, ordenant segons la diferència entre ambdues notes (les majors diferències primer). En cas d'empat, ordenar per cognoms de forma ascendent.	SELECT Cognom1, Cognom2, Parcial1, Parcial2 FROM Alumnes WHERE Parcial1 > Parcial2 ORDER BY (Parcial1-Parcial2) DESC, Apellido1, Apellido2;

## Agrupament de dades

SQL permet definir consultes en la qual s'ofereixen registres que s'obtinguin com resultat de l'agrupament de diverses registres. Per exemple, valor terme mitjà d'un camp, màxim, mínim, compte, etc.

Per a aquest tipus de consultes es proporcionen els següents operadors, que es denominen **funcions d'agregat**:

Operador	Significat
COUNT(<campo>)	Quantitat de registres seleccionats (excepte els que contenen valor nul per al camp). Si <camp> es una llista de camps (separats per &) o *, el registre es conte si algun dels camps que intervenen és no nul.
SUM(<campo>)	Suma del conjunt de valors continguts en el camp especificat. Els registres amb valor nul no es contenen.
AVG(<campo>)	Mitjana aritmètica del conjunt de valors continguts en el camp especificat. Els registres amb valor nul no es contenen.
MAX(<campo>)	Valor màxim del conjunt de valors continguts en el camp especificat. Els registres amb valor nul no es contenen.
MIN(<campo>)	Valor mínim del conjunt de valors continguts en el camp especificat. Els registres amb valor nul no es contenen.

El format d'aquest tipus de consultes és:

```
SELECT COUNT/SUM/AVG/MAX/MIN (<camp>) AS <nom> FROM <taula> WHERE <condició>;
```

Es poden incloure diverses funcions d'agregat en la mateixa consulta. Exemples:

Consulta	SQL
Obtenir la nota mitja de la classe per al primer parcial.	SELECT AVG(Parcial1) AS MitjaP1 FROM Alumnes;

Consulta	SQL
Obtenir la màxima i mínima nota mitja dels 2 parcials.	SELECT MAX(Parcial1+Parcial2)/2 AS MitjaMax, MIN(Parcial1+Parcial2)/2 AS MitjaMin FROM Alumnos;
Obtenir la màxima nota del primer parcial d'entre els alumnes que no tinguin 0 en les pràctiques.	SELECT MAX(Parcial1) AS MaxP1 FROM Alumnos WHERE Practiques <> 0;
Obtenir el nombre d'alumnes que han aprovat el primer parcial.	SELECT COUNT(*) AS Numero FROM Alumnos WHERE Parcial1 >= 5;

En totes les consultes vistes fins ara, les funcions d'agregat s'apliquen sobre el conjunt total de registres d'una taula (excepte els que no compleixen la clàusula WHERE, que són descartats), i el resultat de tals consultes és un únic valor. SQL permet crear grups de registres sobre els quals aplicar les funcions d'agregat, de manera que el resultat és un conjunt de registres per a cadascuna de les quals s'ha calculat el valor agregat. Els grups es componen de diversos registres que contenen el mateix valor per a un camp o conjunt de camps. El format és:

```
SELECT <agregat> AS <nom> FROM <taula> WHERE <condició> GROUP BY <llista de camps>;
```

D'aquesta forma, per a cada valor distint de la subministrada, es calcula la funció d'agregat corresponent, només amb el conjunt de registres amb aquest valor en els camps (els registres que no verifiquin la condició WHERE no es tenen en compte). Exemples:

Consulta	SQL
Obtenir el nombre d'alumnes que hi ha amb el mateix nom (sense cognoms) per a cada nom diferent (Quants Joans, Peres,... hi ha?)	SELECT Nom, COUNT(*) AS Quants FROM Alumnos GROUP BY Nom;
Obtenir el nombre de persones que han obtingut 0, 1, 2...10 en el primer parcial (menyspreant la part decimal de les notes <sup>1</sup> ). Ordenar el resultat pel nombre d'alumnes de forma descendent.	SELECT INT(Parcial1) AS Nota, COUNT(*) AS Quants FROM Alumnos GROUP BY INT(Parcial1) ORDER BY COUNT(*) DESC;

L'agrupament de files imposa limitacions òbvies sobre els camps que poden ser seleccionats, de manera que només poden obtenir-se **camps resultat d'una funció d'agregat o la combinació de camps que aparegui en la clàusula GROUP BY**, i mai altres camps de la taula d'origen. Per exemple la següent consulta seria incorrecta:

```
SELECT Nom FROM Alumnos GROUP BY Cognom1;
```

La raó que sigui incorrecta és trivial: quin Nom (dels diversos possibles) es seleccionaria per a cada grup de Cognom1? (Recordem que per a cada grup generat amb GROUP BY només es mostra una fila com resultat de la consulta.)

## Filtrat de registres de sortida

En aquestes consultes pot aparèixer una condició WHERE que permet descartar els registres que no han de ser tinguts en compte a l'hora de calcular les funcions d'agregat. No obstant això WHERE no permet descartar registres utilitzant com condició el resultat de la funció d'agregat. Per exemple, suposem la següent consulta: "seleccionar els noms d'alumnes per als quals hagi més de 2 alumnes amb el mateix nom (3 Peres, 4 Joans,...)". Intuïtivament podríem fer:

```
SELECT Nom, COUNT(*) FROM Alumnos WHERE COUNT(*)>2 GROUP BY Nom;
```

No obstant això no és correcte. La clàusula WHERE no pot contenir funcions d'agregat. Per a aquesta comesa existeix altra clàusula semblant a WHERE, **HAVING**, que té el següent format:

```
SELECT <agregat> AS <nom> FROM <taula> WHERE <condició> GROUP BY <llista de camps> HAVING <condició d'agregat>;
```

Per a l'exemple anterior la instrucció SQL adequada és:

```
SELECT Nom, COUNT(*) FROM Alumnos GROUP BY Nom HAVING COUNT(*)>2;
```

En resum: **WHERE** selecciona els registres que intervenen per calcular les funcions d'agregat i **HAVING** selecciona els registres que es mostren tenint en compte els resultats de les funcions d'agregat.

<sup>1</sup> La funció de Visual Basic INT proporciona la part entera d'un número.

En tots els casos, la clàusula ORDER BY pot ser inclosa. Evidentment aquesta clàusula afectarà únicament a l'ordre que es mostren els registres resultat, i no al càlcul de les funcions d'agregat. Els camps pels quals pot efectuar-se l'ordenació només poden ser aquells susceptibles de ser també mostrats, és a dir, que els camps admissibles en la clàusula ORDER BY són els mateixos que siguin admissibles en la clàusula SELECT: funcions d'agregat i la combinació de camps que aparegui en GROUP BY.

Recordem el format d'una instrucció SQL de selecció amb totes les opcions vistes fins ara:

```
SELECT <llista de camps per seleccionar>
FROM <taula>
WHERE <condicions>
GROUP BY <llista de camps>
HAVING <condició d'agregat>
ORDER BY <llista de camps per ordenar>;
```

## Consultes sobre múltiples taules

Totes les consultes estudiades fins el moment es basen a seleccionar registres i camps sobre les dades emmagatzemades en una única taula. SQL també permet obtenir resultats a través de la combinació de múltiples taules. La forma de fer-lo és a través de l'enllaç o unió (*join*) de diverses taules a través de claus externes (claus alienes, *foreign keys*). **Una clau externa és un camp o conjunt de camps que fan referència a altre camp o conjunt de camps d'altra taula.** Aquesta relació habitualment s'estableix entre un o diversos camps d'una taula i la clau principal d'altra taula, i la majoria de les vegades guardarà relació directa amb les polítiques d'integritat referencial definides.

## Producte cartesià

L'origen de les consultes basades en múltiples taules és l'operació de producte cartesià, que consisteix en una consulta per a la qual es generen registres resultat de totes les combinacions dels registres de les taules implicades. Suposem les taules següents: *Magatzems* registra els diversos magatzems d'una empresa; *Existències* emmagatzema l'estoc de cada tipus de peça en cada magatzem; *Peces* emmagatzema informació sobre cada tipus de peça:

Magatzems		Existències			Peces	
Id	Ciutat	Magatzem	Tipus	Quantitat	Tipus	Nom
M	Madrid	M	1	100	1	Circuiteria
B	Barcelona	M	2	30	2	TRC
Se	Sevilla	Se	4	200	3	Altaveu
V	Valencia	B	1	50	4	Carcassa
		V	1	10		
		V	2	200		
		V	3	100		

El producte cartesià de les taules Magatzems i Existències seria la següent taula:

Magatzems. Id	Magatzems. Ciutat	Existències. Magatzem	Existències. Tipus	Existències. Quantitat
M	Madrid	M	1	100
M	Madrid	M	2	30
M	Madrid	Se	4	200
M	Madrid	B	1	50
M	Madrid	V	1	10
M	Madrid	V	2	200
M	Madrid	V	3	100
B	Barcelona	M	1	100
B	Barcelona	M	2	30
B	Barcelona	Se	4	200



Magatzems. Id	Magatzems. Ciutat	Existències. Magatzem	Existències. Tipus	Existències. Quantitat
B	Barcelona	B	1	50
B	Barcelona	V	1	10
B	Barcelona	V	2	200
B	Barcelona	V	3	100
Se	Sevilla	M	1	100
Se	Sevilla	M	2	30
Se	Sevilla	Se	4	200
Se	Sevilla	B	1	50
Se	Sevilla	V	1	10
Se	Sevilla	V	2	200
Se	Sevilla	V	3	100
V	Valencia	M	1	100
V	Valencia	M	2	30
V	Valencia	Se	4	200
V	Valencia	B	1	50
V	Valencia	V	1	10
V	Valencia	V	2	200
V	Valencia	V	3	100

En la taula apareixen totes les combinacions de registres de les taules implicades. La forma d'obtenir una consulta de producte cartesià és especificant el nom de les taules implicades en la clàusula FROM:

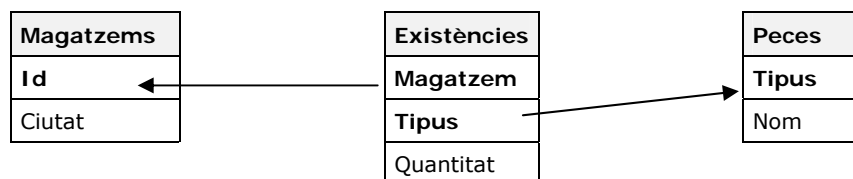
```
SELECT <llista de camps> FROM <taula1>, <taula2>, ... <taulaN>
```

La resta de clàusules estudiades fins ara (WHERE, ORDER BY, GROUP BY, HAVING...) segueixen sent vàlides i utilitzen el mateix format. Les llistes de camps vàlids són ara qualsevol dels de les taules utilitzades, com si es tractés d'una única taula en la qual existeixen tots els camps de totes les taules. Ja que és possible que existeixin camps amb el mateix nom en les diferents taules, a l'hora de nomenar els camps serà necessari especificar a quina taula pertanyen amb el format "<taula>.<camp>".

Així, la taula generada en l'exemple anterior (producte cartesià) s'obté amb la següent instrucció SQL:

```
SELECT * FROM Magatzems, Existències;
```

Les consultes de producte cartesià com a finalitat última són poc habituals. En general el producte cartesià s'utilitza com mitjà per obtenir consultes que relacionen diverses taules a partir de claus externes. En les taules d'exemple s'observa clarament la relació existent entre els camps:



*Existències.Magatzem* conté un identificador del magatzem al que es refereixen les existències (requereix integritat referencial d'algun tipus amb el camp *Magatzems.Id*), i *Existències.Tipus* conté un identificador del tipus al que es refereixen el registre d'existències (requereix integritat referencial amb el camp *Peces.Tipus*).

A més es pot intuir la següent informació:

- *Magatzems.Id* és la clau principal de Magatzems.
- *Peces.Tipus* és la clau principal de Peces.
- (*Existències.Magatzem*, *Existències.Tipus*) és la clau principal d'Existències.
- *Existències.Magatzem* és una clau aliena d'Existències sobre Magatzems.

- *Existències.Tipus* és una clau aliena d'Existències sobre Peces.

Les claus alienes permeten enllaçar la informació relacionada entre diferents taules. Per exemple, si volem relacionar les existències en un magatzem amb el nom de la ciutat on se situa aquest magatzem, podem calcular el producte cartesià de Magatzems i Existències, i descartar aquells registres per als quals no coincideixin els camps *Magatzems.Id* i *Existències.Magatzem*.

```
SELECT Magatzems.Ciutat, Existències.Tipus, Existències.Quantitat FROM
Magatzems, Existències WHERE Magatzems.Id = Existències.Magatzem;
```

Aquesta consulta ens proporcionaria la següent taula:

Magatzems.Ciutat	Existències.Tipus	Existències.Quantitat
Madrid	1	100
Madrid	2	30
Barcelona	1	50
Sevilla	4	200
Valencia	1	10
Valencia	2	200
Valencia	3	100

De la mateixa forma podríem enllaçar les tres taules mitjançant la següent consulta:

```
SELECT Magatzems.Ciutat, Peces.Nom, Existències.Quantitat FROM Magatzems,
Existències, Peces WHERE (Magatzems.Id = Existències.Magatzem) AND
(Existències.Tipus = Peces.Tipus);
```

El resultat de tal consulta és la següent taula:

Magatzems.Ciutat	Peces.Tipus	Existències.Quantitat
Madrid	1	Circuiteria
Madrid	2	TRC
Barcelona	1	Circuiteria
Sevilla	4	Carcassa
Valencia	1	Circuiteria
Valencia	2	TRC
Valencia	3	Altaveu

## Unió (join)

Encara que aquesta forma d'enllaçar taules és correcta, existeix altre mecanisme més adequat per a enllaçar taules a través de les seves claus externes. Es tracta de l'operació d'unió (*join*).

L'operació d'unió bàsicament obté el mateix resultat que un producte cartesià filtrat perquè només es mostrin els registres en els quals coincideixi la clau aliena (condició de *join*). La diferència és que es utilitzarà una clàusula específica per definir l'operació, en lloc de la genèrica WHERE, el que permetrà al SGDB identificar el tipus d'operació i proporcionar alguns avantatges sobre el resultat (que veurem més endavant).

La sintaxi per a una operació d'unió és:

```
SELECT <llista de camps>
FROM <taula1>
INNER JOIN <taula2> ON <taula1>.<camp1> = <taula2>.<camp2>;
```

Aquesta és la unió equiparable al producte cartesià filtrat com:

```
SELECT <llista de camps>
FROM <taula1>, <taula2>
WHERE <taula1>.<camp1> = <taula2>.<camp2>;
```

En general per a qualsevol nombre de taules, la unió es realitza mitjançant imbricacions d'unions. La sintaxi per a tres taules és:

```
SELECT <llista de camps>
FROM <taula1> INNER JOIN (
```

```
<taula2> INNER JOIN <taula3> ON <taula2>.<camp2> = <taula3>.<camp3>
    ) ON <taula1>.<camp1> = <taula2>.<camp2>;
```

I per a N taules:

```
SELECT <llista de camps>
FROM <taula1> INNER JOIN (
    <taula2> INNER JOIN ( ...
    <taulaN-1> INNER JOIN <taulaN> ON <taulaN-1>.<campoN-1>=<taulaN>.<campoN>
    ... ) ON <taula2>.<camp2> = <taula3>.<camp3>
    ) ON <taula1>.<camp1> = <taula2>.<camp2>;
```

## Consultes d'inserció

Les consultes d'inserció permeten afegir registres a una taula. Per a aquest tipus de consultes es requereix:

- 1) Una taula a la qual afegir les dades.
- 2) Una consulta de selecció de la qual obtenir les dades que s'afegeixen, o bé una llista dels valors a inserir.

El format SQL d'una consulta d'inserció de dades utilitzant una consulta de selecció com origen de les dades és:

```
INSERT INTO <taula destinació> (<llista camps destinació>) SELECT <llista
camps origen> FROM <taula origen>;
```

La llista de camps destinació és una llista de camps separats per comes; la llista de camps origen és una llista a l'estil de la utilitzada en una consulta de selecció qualsevol. Cada camp de la llista d'origen deu correspondre's amb un altre en la llista de destinació, en el mateix ordre, de manera que els registres obtinguts en la consulta s'afegeixen a la taula de destinació. Els camps no especificats seran omplerts amb els valors per defecte, a menys que no tinguin cap valor predeterminat, en aquest cas quedaran buits (amb valors nuls).

La part de la consulta de selecció pot contenir totes les opcions estudiades: agrupament, funcions d'agregat, ordenament de registres, condicions de filtrat, etc.

Per afegir dades a una taula sense utilitzar altra taula o consulta com origen de dades, es pot utilitzar la següent sintaxi:

```
INSERT INTO <taula destinació> (<llista camps destinació>) VALUES <llista
camps origen>;
```

Com en el cas anterior, ha d'existir una correspondència i compatibilitat exacta entre la llista de camps d'origen i la llista de camps de destinació

Exemples de consultes d'inserció:

Consulta	SQL
Suposem una taula Persones en la qual s'emmagatzema informació sobre el nom, cognoms i càrrec (en camps Nom, Cognoms, Carrec) de totes les persones de la universitat. Afegir a aquesta taula tots els alumnes de la taula Alumnes.	INSERT INTO Persones ( Nom, Cognoms, Carrec ) SELECT Nom, Cognom1 & " " & Cognom2 AS CognomsA, "Alumne" AS CarrecA FROM Alumnes;
Suposem una taula Historia en la qual s'emmagatzema informació sobre el nombre d'alumnes matriculats cada any. Aquesta taula té els camps: Anyo (tipus data) i Numero (Enter llarg). Afegir a aquesta taula el nombre d'alumnes actual amb la data d'aquest any.	INSERT INTO Historia ( Anyo, Numero ) SELECT Year(Date()) AS AquestAny, COUNT (*) As Total FROM Alumnes;
Afegir l'alumne "Francisco Pérez Solís", amb NIF 23.123.234-F a la llista d'alumnes.	INSERT INTO Alumnes ( Nom, Cognom1, Cognom2, NIF ) VALUES ("Francisco", "Pérez", "Pérez", "23.123.234-F");

## Consultes de creació de taula

Aquest tipus de consultes són idèntiques a les d'inserció excepte per que la taula de destinació de les dades especificada no existeix, i es crea en el moment d'executar la consulta.

## Consultes d'actualització

Les consultes d'actualització de dades permeten modificar les dades emmagatzemades en una taula. Es tracta de modificar els valors de determinats camps en els registres que compleixin una determinada condició. La sintaxi d'aquest tipus de consultes és:

```
UPDATE <taula> SET <camp> = <nou valor>, <camp> = <nou valor>, <camp> = <nou valor> WHERE <condició>;
```

Vegem alguns exemples:

Consulta	SQL
Aprovar el primer parcial a tots els alumnes que tinguin una nota entre 4,5 i 5.	UPDATE Alumnes SET Parcial1 = 5 WHERE (Parcial1 >= 4.5) AND (Parcial1 < 5);
Posar un 1 en les pràctiques a tots els alumnes que no tinguin cap nota en pràctiques i tinguin els dos parcials aprovats i amb una nota mitja entre ambdós major que 7.	UPDATE Alumnes SET Practiques = 1 WHERE (Parcial1 >= 5) AND (Parcial2 >= 5) AND (Parcial1+Parcial2 > 7) AND (Practiques IS NULL);
Arrodonir les notes dels alumnes llevant els decimals.	UPDATE Alumnes SET Parcial1 = INT (Parcial1), Parcial2 = INT (Parcial2), Practiques = INT (Practiques);
Posar un 0 en pràctiques als alumnes amb DNI "34.753.485-J"	UPDATE Alumnes SET Practiques = 0 WHERE DNI="34.753.485-J";
Oblidar el DNI dels alumnes que s'han presentat al segon parcial.	UPDATE Alumnes SET DNI = NULL WHERE Parcial2 IS NOT NULL;

## Consultes d'eliminació

Les consultes d'eliminació de dades permeten eliminar registres d'una taula de forma selectiva: els registres que compleixin una determinada condició. La sintaxi d'aquest tipus de consultes és:

```
DELETE [<taula>.*] FROM <taula> WHERE <condició>;
```

Les consultes d'eliminació no permeten esborrar camps; només registres complets. Per això la part <taula>.\* és opcional. Per eliminar el valor dels camps ha d'utilitzar-se una consulta d'actualització, canviant el valor dels camps a NULL.

Si no s'especifica cap condició, s'eliminen tots els registres. **No s'elimina la taula**, ja que l'estructura segueix existint, encara que no contingui cap registre.

Alguns exemples de consultes d'eliminació:

Consulta	SQL
Eliminar als alumnes que ho hagin aprovat tot.	DELETE FROM Alumnes WHERE (Parcial1 >= 5) AND (Parcial2 >= 5) AND (Practiques >=1);
Eliminar als alumnes dels quals es desconeixi el primer o el segon cognom.	DELETE FROM Alumnes WHERE (Cognom1 IS NULL) OR (Cognom2 IS NULL);
Eliminar a tots els alumnes.	DELETE FROM Alumnes;

## Consultes imbricades

Access permet la imbricació de consultes. La forma habitual d'utilitzar aquest mecanisme és emprar el resultat d'una consulta per seleccionar valors d'una altra. Per exemple, la consulta "prendre de la taula d'alumnes els aquells primers cognoms que també s'utilitzin com segons cognoms" s'escriuria en SQL com:

```
SELECT DISTINCT Cognom1 FROM Alumnes WHERE Cognom1 IN (SELECT Cognom2 FROM Alumnes);
```

La subconsulta es tanca entre parèntesi. Es poden imbricar tantes consultes com es vulgui. Les clàusules que permeten enllaçar la consulta principal i la subconsulta són les següents:

- **Qualsevol comparador** (>, <, =, etc...). En aquest cas, la subconsulta ha de proporcionar un resultat únic amb el qual realitzar la comparació.

- **Qualsevol comparador seguit de ALL, ANY o SOME.** En aquest cas, la subconsulta pot proporcionar múltiples registres com resultat.
  - ALL: es seleccionen en la consulta principal només els registres que verifiquin la comparació amb tots els registres seleccionats en la subconsulta.
  - ANY: es seleccionen en la consulta principal només els registres que verifiquin la comparació amb algun dels registres seleccionats en la subconsulta.
  - SOME és idèntic a ANY.
- **El nom un camp + IN.** En aquest cas la subconsulta pot proporcionar múltiples registres com resultat, i se seleccionen en la consulta principal els registres per als quals el valor del camp aparegui també en el resultat de la subconsulta. És equivalent a utilitzar "= ANY". Es pot utilitzar **NOT IN** per aconseguir l'efecte contrari, equivalent a "<>ALL".
- **La clàusula EXISTS.** El resultat de la consulta pot proporcionar múltiples registres. La condició avaluada és que en la subconsulta es recuperi algun registre (EXISTS) o no es recuperi cap registre (NOT EXISTS).

Exemples de consultes imbricades.

Consulta	SQL
Seleccionar els alumnes la nota dels quals en el primer parcial sigui major o igual que la mitjana de tots els alumnes en aquest parcial.	SELECT * FROM Alumnes WHERE Parcial1 >= (SELECT AVG(Parcial1) FROM Alumnes);
Seleccionar els alumnes majors que l'alumne amb millor nota en pràctiques (suposant que només hi ha un amb la màxima nota).	SELECT * FROM Alumnes WHERE Edat >= (SELECT Edat FROM Alumnes WHERE Practiques = (SELECT Max(Practiques) AS MaxPract FROM Alumnes));
Seleccionar els alumnes el nom dels quals també ho tinguin professors.	SELECT * FROM Alumnes WHERE Nom IN (SELECT Nom FROM Professors);
Seleccionar noms d'alumnes que també apareguin com cognoms.	SELECT Nom FROM Alumnes WHERE (Nom IN (SELECT Cognom1 FROM Alumnes)) OR (Nom IN (SELECT Cognom2 FROM Alumnes));
Indicar quants alumnes tenen la nota del primer parcial major que la màxima nota del segon parcial d'entre els alumnes que en les pràctiques no han aprovat.	SELECT Count(*) AS Numero FROM Alumnes WHERE Parcial1 > (SELECT MAX(Parcial2) FROM Alumnes WHERE Practiques<1);
Suposant que el registre de cada alumne conté el DNI del seu professor tutor en un camp DNIPprof, seleccionar en ordre alfabètic per cognoms els alumnes el tutor dels quals és "Carles".	SELECT * FROM Alumnes WHERE DNIPprof = (SELECT DNI FROM Professors WHERE Nom="Carles") ORDER BY Cognom1, Cognom2;
Suposant que el registre de cada alumne conté el DNI del seu professor en un camp DNIPprof, seleccionar els alumnes el professor dels quals és algun dels quals han suspès la pràctiques a tots els seus alumnes.	SELECT * FROM Alumnes WHERE DNIPprof NOT IN (SELECT DNIPprof FROM Alumnes WHERE Practiques >= 1);
Suposant que el registre de cada alumne conté el DNI del seu professor en un camp DNIPprof, el nom del professor que té tants alumnes amb el mateix nom com alumnes amb el mateix primer cognom (i més de 1), i que no té cap alumne menor de 18 anys.	SELECT Nom FROM Professors WHERE DNI IN (SELECT Alumnes.DNIPprof FROM Alumnes GROUP BY Alumnes.DNIPprof, Alumnes.Nom HAVING Count(*) IN (SELECT Count(*) AS Cap1 FROM Alumnes GROUP BY Alumnes.DNIPprof, Alumnes.Cognom1 HAVING Count(*)>1) ) AND DNI NOT IN (SELECT DNIPprof FROM Alumnes WHERE Edat<18);

De vegades és necessari utilitzar els valors dels camps de la consulta principal en la subconsulta. En tal cas és necessari identificar la taula del camp consultat utilitzant un nom i AS. Suposem la següent consulta: *Seleccionar els alumnes el professor dels quals és "Carles"*. Aquesta consulta pot escriure's en SQL de la següent forma (encara que no és l'única forma, ni la millor):

```
SELECT * FROM Alumnes AS Alu WHERE EXISTS (SELECT * FROM Professors WHERE (Alu.DNIPprof=Professors.DNI) AND (Professors.Nom="Carles"));
```

La raó de donar un nom a la taula dintre de la consulta és permetre a Access identificar correctament a la taula d'Alumnes de la qual s'obté el camp *DNIPprof*, ja que la subconsulta podria contenir també a la taula Alumnes.

## Consultes de taula de referències creuades

Les consultes de taula de referències creuades permeten crear una mena de taula en la qual tant els títols de fila com els de columna s'obtenen a partir de les dades d'una taula.

Suposem una taula Components:

Nom	Pes	Color
Torre 1	6	A
Torre 2	6	B
Torre 3	7	A
Unitat disc	1	A
CD ROM Hitachi	2	A
CD ROM SONY	3	B
CD ROM Creative	3	C
Teclat clònic	4	A

Nom	Pes	Color
Teclat Microsoft	5	C
Monitor SONY 15"	10	A
Monitor SONY 17"	15	A
Monitor SONY 21"	21	A
Monitor Hitachi 15"	9	B
Ratolí Genius	1	A
Ratolí IBM	2	B
Ratolí HP	2	B

Amb una consulta de taula de referències creuades podem aconseguir construir una taula que ens resumeixi el nombre de components de cada pes i color:

Color	1	2	3	4	5	6	7	9	10	15	21
A	2	1		1		1	1		1	1	1
B		2	1			1		1			
C			1		1						

La consulta que genera aquesta taula és la següent:

```
TRANSFORM Count(*) AS NumeroDeComponents SELECT Components.Color FROM
Components GROUP BY Components.Color PIVOT Components.Pes;
```

## Consultes específiques de SQL

Aquest tipus de consultes no es pot definir de forma visual en Access, pel que han de obligatòriament ser definides utilitzant comandos SQL. Veurem dos tipus de consultes específiques de SQL: d'unió de dades i de definició de dades.

### Consultes d'unió de taules

En Access aquest tipus de consultes es denominen "d'unió", però no les hem de confondre-les amb les de JOIN (que ja hem denominat "d'unió").

Les consultes d'unió permeten obtenir una taula a partir de les dades de vàries, però no com es fa en el producte cartesià, sinó al final de la taula, com si es afegissin les dades de les altres taules als quals ja hi ha en la primera. Per exemple, si tenim una taula de Professors i altra d'Alumnes, podem generar una consulta que ens doni els noms i cognoms de tots ells.

La sintaxi és:

```
SELECT <llista de camps> FROM <taula1> UNION [ALL] SELECT <llista de camps>
FROM <taula2>;
```

La clàusula opcional ALL permet obtenir registres duplicats: si s'omet no apareixen i si s'especifica, es mostraran només valors únics. Cada consulta de concatenació ha de retornar el mateix nombre de camps, i en el mateix ordre. Es necessita que els camps corresponents tinguin tipus de dades compatibles (que es puguin convertir entre si). Si els noms dels camps corresponents no coincideixen o han de ser canviats, cal utilitzar la clàusula AS de forma similar en tots els SELECT. Per a l'exemple d'alumnes i professors, la consulta seria:

```
SELECT Cognom1, Cognom2, Nom FROM Alumnes UNION SELECT Cognom1, Cognom2, Nom
FROM Professors;
```

Si la taula professors només tingués camps Cognoms (amb els dos cognoms junts) i Nom, podríem fer el següent:

```
SELECT Cognom1 & Cognom2 AS Cognoms, Nom FROM Alumnes UNION SELECT Cognoms,
Nom FROM Professors;
```

La clàusula ORDER BY ha d'especificar-se al final de la consulta, afecta a la consulta completa i només pot aplicar-se sobre camps mostrats en la selecció. La resta de clàusules (WHERE, GROUP BY, etc.) pertanyen a cada SELECT i s'especifiquen com en qualsevol altra consulta.

Si existeixen més de dues taules concatenades, el criteri UNION o UNION ALL utilitzat serà l'últim especificat.

## Consultes de definició de dades

Les consultes de definició de dades s'utilitzen per **crear taules, modificar definicions de taules, eliminar taules, crear índexs i esborrar índexs**. Aquest tipus de consultes permet crear taules buides, fent una especificació precisa de les característiques de la taula.

Vegem diversos exemples de consultes de definició de dades.

Consulta	SQL
Crear la taula d'Alumnes. Observar que DNI no pot ser nul i que la clau principal és DNI.	CREATE TABLE Alumnes ( [DNI] Text NOT NULL, [Cognom1] Text, [Cognom2] Text, [Nom] Text, [Edat] Integer, [Parcial1] Single, [Parcial2] Single, [Practiques] Single, CONSTRAINT [UnIndice] PRIMARY KEY ([DNI]) );
Modificar la taula d'Alumnes. Eliminar el camp Practiques.	ALTER TABLE Alumnes DROP COLUMN Practiques;
Modificar la taula d'Alumnes. Afegir el camp Telèfon de tipus text.	ALTER TABLE Alumnes ADD COLUMN Telefon Text;
Modificar la taula d'Alumnes. Afegir la restricció que DNI és una clau aliena sobre la taula <i>ExpedientsDisciplinaris</i> en el camp <i>DNIexpedient</i> .	ALTER TABLE Alumnes ADD CONSTRAINT UnaClaveExterna FOREIGN KEY (DNI) REFERENCES ExpedientsDisciplinaris (DNIexpedient);
Modificar la taula d'Alumnes. Eliminar la restricció definida en l'anterior consulta.	ALTER TABLE Alumnes DROP CONSTRAINT UnaClaveExterna;
Esborrar la taula <i>ExpedientsDisciplinaris</i>	DROP TABLE ExpedientsDisciplinaris;
Crear un índex en la taula Alumnes, sobre el camp DNI.	CREATE UNIQUE INDEX Indice1 ON Alumnes (DNI);
Eliminar l'índex definit per al DNI (el de la consulta anterior).	DROP INDEX Indice1 ON Alumnes;
Crear un índex en la taula Alumnes, sobre els camps Cognom1 i Cognom 2. Permetre valors repetits.	CREATE INDEX Indice2 ON Alumnes (Apellido1, Apellido2);

## Modificació i accés a les dades d'una consulta. Vistes.

La utilització de consultes en bases de dades persegueix dos objectius:

- **L'obtenció de resultats** sobre les dades emmagatzemades. És el que hem vist fins ara.
- **La generació de vistes.**

Les vistes són consultes de selecció que s'utilitzen com si es tractés de taules. De forma transparent a l'usuari, les vistes mostren el contingut d'una taula amb un format, ordre i contingut adequat a les necessitats de l'usuari. Per exemple, si no volem que un usuari tingui accés a les dades DNI de la taula d'alumnes, podem crear una consulta que proporcioni tots les dades (excepte el DNI) de tots els alumnes, i presentar-li la consulta com si fos la pròpia taula. De la mateixa forma, podem ocultar-li l'existència de determinats alumnes, etc.

Una consulta es pot presentar a gairebé tots els efectes de la mateixa forma que una taula. Es poden fer consultes sobre consultes, afegir, modificar o eliminar dades sobre les presentació del resultat d'una consulta, crear formularis i informes sobre consultes (en comptes de taules), etc.

No obstant existeix una limitació: determinades operacions no es permeten sobre determinades consultes emprades com vistes. Aquesta limitació està imposada per la possibilitat o impossibilitat que Access insereixi de forma automàticament els valors als quals ho es té accés mitjançant la vista.

Suposem una consulta com:

```
SELECT * FROM Alumnes WHERE (Edat > 21) ORDER BY NIF;
```

Aquesta consulta de selecció pot utilitzar-se com una vista, ja que és trivial esbrinar a quina fila i camp de la taula d'alumnes correspon cada fila o camp de la vista: qualsevol modificació, inserció o esborrat pot transmetre's a la taula base. És important ressenyar que els registres que s'afegeixin poden no verificar la condició WHERE o la clàusula ORDER BY. No obstant això no representa una falta a cap regla d'integritat i és perfectament legal. La pròxima vegada que es reconsulti la vista, els registres que no verifiquin la condició no tornaran a aparèixer, però estaran en la taula original.

No ocorre el mateix amb una consulta com:

```
SELECT AVG(Parcial1) AS MitjaP1 FROM Alumnes;
```

Si intentéssim utilitzar aquesta taula com una vista sobre la taula Alumnes, com s'introduirien les dades en Alumnes a l'afegir un nou registre en aquesta vista? El mateix ocorre amb moltes altres consultes. Els següents tipus de consultes **poden funcionar com vistes**:

- **Consultes de selecció simple**, amb o sense condició WHERE, que afegeixin o filtrin camps, amb qualsevol tipus d'ordre.
- **Consultes d'unió utilitzant INNER JOIN.**

Els següents tipus de consultes **no poden funcionar com vistes** a causa de la impossibilitat de reconèixer la ubicació de les dades en la taula original a partir de les de la vista:

- **Consultes d'unió basades en producte cartesià** (i filtrades amb WHERE) que no utilitzen INNER JOIN.
- **Consultes amb funcions d'agregat** (utilitzant GROUP BY).
- **Consultes de concatenació de taules.**
- **Consultes que no siguin de selecció**: inserció, eliminació, modificació, definició de dades, etc.

## Exercici 10: consultes

1. Codi i nom dels productes que valen més de 270 €.
2. Codi de les botigues on hi ha unitats del producte P25. (Ja que quan vam introduir el codi de producte es va fer amb una màscara, el valor que es va introduir fou 25, no P25)
3. Codi i data de comanda de les comandes de més de 15 unitats que s'hagin fet als magatzems M02 i M05.
4. Llistat de noms dels productes i preu, afegint una columna amb el preu més IVA.
5. Llistat de noms dels productes de tipus Televisor ordenats per preu de forma ascendent.
6. Llistat de noms dels productes que pesen entre 10 i 30 kg. ordenats per preu de forma descendent.
7. Codis de les botigues on hi ha existències (sense repetir).
8. Quantitat total i mitjana de productes per botiga.
9. Comptar el número de botigues.
10. Obtenir la millor relació preu / pes dels productes.
11. Obtenir els codis de les botigues on hi ha més de dos tipus diferents de productes.
12. Noms dels productes que tenen més de 20 unitats en la botiga B01.
13. Llistat de totes les dates i quantitats en les que s'han fet comandes de productes més cars de 300 €, agrupats per dates.
14. Noms dels productes que hi ha en existència en les botigues de Madrid, agrupats per nom de producte.
15. Llistat de les quantitats de productes dels que s'han fet comandes, per dia i per magatzem.
16. Adreça i localitat dels magatzems que tenen en existència més de 50 unitats del producte "Vídeo v33".
17. Quantitat d'unitats de "Secador sx" que té la botiga de Huelva.



18. Quantitat de productes en existència en les botigues de Madrid que pesen més de 10 kg.
19. Quantitat total d'unitats dels productes que hi ha en existència en les botigues.
20. Mostrar codi i nom dels productes existents en els magatzems, indicant adreça i població, mitjançant una taula de referències creuades.

## **Exercici 11: consultes**

1. Seleccionar els alumnes majors de 23 anys.
2. Seleccionar els cognoms i nom dels que han aprovat els dos parcials.
3. Generar un llistat amb els noms i cognoms dels alumnes i les seves notes. En un camp únic el NomComplet serà de la forma "Cognom1 Cognom2, Nom". El llistat estarà ordenat per aquest camp de forma ascendent.
4. Obtenir els cognoms junt a la nota mitja (1 decimal), suposant que la suma dels parcials és el 80% i la resta s'obté de les pràctiques.
5. Obtenir els alumnes amb el primer parcial aprovat, començant per les millors notes.
6. Obtenir els cognoms i les notes dels parcials dels alumnes que han tret millor nota en el primer parcial que en el segon, ordenat segons la diferència entre ambdues notes (les majors diferències primer). En cas d'empat, ordenar per cognoms de forma ascendent.
7. Obtenir la nota mitja de la classe pel primer parcial amb dos decimals de precisió.
8. Obtenir la màxima i mínima nota mitja dels dos parcials.
9. Obtenir la màxima nota del primer parcial d'entre els alumnes que no tinguin 0 en pràctiques.
10. Obtenir la quantitat d'alumnes que han aprovat el primer parcial.
11. Obtenir la quantitat d'alumnes que hi ha amb el mateix nom (sense cognoms) per a cada nom diferent.
12. Obtenir la quantitat de persones que han obtingut 0, 1, 2, ... 10 en el primer parcial (només la part entera de les notes, sense decimals). Ordenar el resultat per la quantitat d'alumnes de forma descendent.
13. Afegir a la taula Persones tots els alumnes de la taula Alumnes amb el càrrec "Alumne".
14. Afegir a la taula Història la quantitat d'alumnes actual amb l'any en curs.
15. Eliminar tots els alumnes.
16. Recuperar tots els alumnes des de la taula Alumnes (còpia).
17. Aprovar el primer parcial a tots els alumnes que tinguin una nota entre 4,5 i 5
18. Posar un 1 en les pràctiques a tots els alumnes que tinguin un 0 en pràctiques i tinguin els dos parcials aprovats amb una nota mitja entre ambdós major o igual que 7
19. Arrodonir les notes dels alumnes traient els decimals.
20. Eliminar als alumnes que ho han aprovat tot.
21. Eliminar als alumnes dels que es desconeixi el primer o el segon cognom.
22. Seleccionar als alumnes que tinguin una nota en el primer parcial major o igual que la mitja de tots els alumnes en aquest parcial.
23. Indicar quants alumnes, dels alumnes que no han aprovat les pràctiques, tenen la nota del primer parcial major que la màxima nota del segon parcial.
24. Afegir el camp Telèfon de tipus Text(9) a la taula Alumnes.
25. Crear una consulta d'unió entre Alumnes i Persones
26. Mitjançant una consulta de definició de dades, crear una nova taula amb la mateixa estructura que la taula Alumnes.
27. Mitjançant una consulta de definició de dades, eliminar la taula creada en l'exercici anterior.

## **Exercici 12: consultes SQL**

Escriure i provar en Access les següents consultes utilitzant el llenguatge SQL:

1. Codi i nom dels productes que valen més de 270€.
2. Codi de les botigues on hi ha unitats del producte P25.
3. Codi i data de comanda de les comandes de més de 11 unitats que hagin fet als magatzems M02 i M05.
4. Obtenir tota la informació dels magatzems.

5. Llistat de noms dels productes i el seu preu, afegint-li una columna amb el preu amb IVA.
6. Llistat de noms dels productes de tipus Televisor ordenats per preu de forma ascendent.
7. Llistat de noms dels productes que pesin entre 10 i 30 kg. ordenats per preu de forma descendent.
8. Codis de les botigues on hi ha existències (sense repetir).
9. Quantitat total i mitjana de productes per botiga.
10. Contar el nombre de botigues.
11. Obtenir la millor relació preu/pes dels productes.
12. Obtenir els codis de les botigues on hi ha més de dos tipus distints de productes.
13. Noms dels productes que tenen més de 20 unitats en la botiga B01.
14. Noms de les ciutats on hagi botiga o magatzem.
15. Noms de les ciutats on hagi botiga i magatzem.
16. Llistat de totes les dates i quantitats en les quals s'han demanat productes més cars de 300€.
17. Codi de productes dels quals hagi més productes en una botiga que en un magatzem.
18. Noms dels productes que hi ha en existència en les botigues de Madrid.
19. Llistat del nombre de productes que costen el mateix.
20. Llistat de les quantitats de productes demanats per dia i per magatzem.
21. Adreça i localitat dels magatzems que tenen en existència més de 50 unitats del producte "Vídeo v33".
22. Nombre d'unitats de "Secador sx" que té la botiga d'Huelva.
23. Nombre de productes en existència en les botigues de Madrid que pesin més de 10 kg.
24. Existències totals en les botigues, dividits per productes.

# Formularis

## Introducció

Els formularis són formats de pantalla destinats que l'usuari els utilitzi com interfície per a la manipulació de les dades. Són equivalents a la manera de vista de dades de les taules, però permeten major llibertat a l'hora de personalitzar els controls que es mostren en pantalla: col·locació dels controls, aspecte dels mateixos i de la pantalla en general, presentacions imbricades, gràfics i imatges, presentació d'objectes multimèdia, etc.

En general, un formulari està associat a una taula, de la qual pren la informació. Inicialment el formulari conté un conjunt de controls, cadascun d'ells associat a un camp de la taula subjacent. La modificació de les dades d'aquests controls es reflecteix automàticament en les dades de la taula; de fet, totes les consideracions fetes sobre l'edició de dades sobre les taules són aplicables als formularis.

La creació i definició de formularis es fa de la mateixa forma que es programa sobre un entorn visual. Els controls se situen sobre l'espai del formulari amb ajuda del ratolí, i les propietats de tots els objectes implicats es mostren i modifiquen mitjançant diàlegs, de manera que el codi de programació del formulari queda ocult.

En les seccions següents es comentarà el significat de les propietats dels principals objectes disponibles.

En Access les propietats dels objectes s'agrupen en quatre conjunts, segons la seva funcionalitat. Un diàleg flotant mostra i permet modificar l'estat de les propietats dels controls d'un formulari i del propi formulari. Aquest diàleg es mostra fent doble clic sobre qualsevol control. Conté cinc subcarpetes, que permeten accedir als quatre conjunts de propietats, i un cinquè conjunt que conté totes les propietats barrejades. Els quatre conjunts són:

- **Format:** Conté propietats referides a l'aspecte visual dels controls: posició, grandària, color, fonts, format de presentació de dades, etc.
- **Dades:** Conté propietats referides a l'associació amb els camps de la taula relacionada: camps associats, màscares d'entrada, valors predeterminats, regles de validació, etc.
- **Esdeveniments:** Conté propietats referides a les respostes que els controls ofereixen davant determinades situacions. Els esdeveniments es coneixen en altres SGBD com triggers o disparadors, i consisteixen en fragments de codi en un determinat llenguatge que s'executen quan es produeixen determinades situacions.
- **Altres:** Altres propietats. Destaca el nom del control, que és un identificador que permet referir-se al control per a accedir a les seves propietats.

Algunes propietats dels controls, especialment les del conjunt de Dades, poden contradir les propietats que es van establir per als camps als quals associem dits controls. Per exemple, regles de validació i màscares d'entrada. En tal cas, prevalen els definits per al formulari, sense perjudici dels definits per als camps de la taula. Per exemple:

- Una màscara d'entrada definida per al control en un formulari preval sobre la definida per al camp, ja que l'edició de les dades es realitza sobre el control del formulari.
- Una regla de validació definida per al control en un formulari deu verificar-se perquè el contingut del control sigui acceptat pel control, però a més deu verificar-se la regla definida per al camp de la taula, perquè en cas contrari la taula no acceptarà el contingut que li passa el control (encara que sigui vàlid per al control).





















Per defecte, quan associem un camp a un control, el control hereta automàticament les propietats compatibles del camp, com les regles de validació, els formats d'entrada i sortida, etc.

## Creació de formularis

Els formularis, igual que les taules i la resta d'objectes en Access, té dues maneres d'operació: un per a modificar la definició (vista de disseny) i altre per a la seva utilització (vista de formulari). I igual que la resta d'objectes, els formularis necessiten un nom identificatiu.

Existeixen assistents que permeten crear alguns tipus habituals de formularis. No obstant això, rares vegades aconseguen obtenir el formulari tal com es necessita.

Al crear un nou formulari, apareix una finestra buida (el formulari) sobre la qual podem situar els controls, i una taujana d'eines, que conté els controls disponibles.

	Permet <b>seleccionar els controls del formulari</b> per a moure'ls, canviar-los la grandària, accedir a les seves propietats, etc...
	Inicia l'assistent per a la creació de controls.
	Insereix un control de tipus <b>Etiqueta</b> : un text que apareix en el formulari.
	Insereix un control de tipus <b>Quadre de text</b> : un editor que permet escriure text.
	Insereix un control de tipus <b>Grup d'opcions</b> . Tots els controls de tipus Botó d'alternar, Botó d'opció i Casella de verificació que quedin dintre del mateix Grup d'opcions funcionen de forma autoexcloent, de forma que només un pot estar activat.
	Insereix un control de tipus <b>Botó d'alternar</b> , que pot estar pressionat o solt (valor lògic).
	Insereix un control de tipus <b>Botó d'opció</b> , que pot estar activat o desactivat (valor lògic).
	Insereix un control de tipus <b>Casella de verificació</b> , que pot estar activat o desactivat (valor lògic).
	Insereix un control de tipus <b>Quadre combinat</b> , que consisteix en un Quadre de text que pot desplegar-se i presentar una llista de valors permesos.
	Insereix un control de tipus <b>Quadre de llista</b> , que consisteix en una llista de valors permesos.
	Insereix un control de tipus <b>Botó de comando</b> , que permet executar una macro quan és pressionat.
	Insereix un control de tipus <b>Imatge</b> , que permet introduir un dibuix en el formulari. L'origen de la imatge és un arxiu gràfic.
	Insereix un control de tipus <b>Marc d'objecte independent</b> , que permet introduir un objecte multimèdia en el formulari, bé des d'un arxiu, o bé mitjançant vinculació OLE.
	Insereix un control de tipus <b>Marc d'objecte dependent</b> , que permet introduir un objecte multimèdia en el formulari, associat a un camp de la taula subjacent de tipus Objecte OLE.
	Insereix un <b>Salt de pàgina</b> (només útil en Informes, no el formularis).
	Insereix un control de tipus <b>Control fitxa</b> , que permet agrupar controls dintre de subcarpetes. Els controls de tipus Botó d'alternar, Botó d'opció i Casella de verificació que quedin dintre del mateix Control fitxa funcionen de forma independent, encara que quedin dintre de la mateixa subcarpeta.
	Insereix un control de tipus <b>Subformulari/Subinforme</b> , que permet imbricar altres formularis dintre de l'actual, vinculant el contingut d'alguns camps.
	Insereix un control de tipus <b>Línia</b> , consistent en una línia dibuixada en el formulari.
	Insereix un control de tipus <b>Rectangle</b> , consistent en una rectangle dibuixat en el formulari.
	Insereix controls avançats (ActiveX), com exploradors web, altres versions dels controls anteriors, objectes gràfics i multimèdia avançats, etc.

## L'objecte formulari

El formulari en si és un objecte que posseeix propietats com si fos un control. Abans de començar a afegir controls a un formulari deuriem establir-se les propietats del propi formulari, ja que permetrà que Access ens faciliti després la tasca d'afegir controls.

Abans de començar a estudiar les propietats d'un formulari, cal distingir dos tipus fonamentals de formularis i assenyalar que el formulari en si està dividit en diverses parts.

Existeixen **dos tipus de formularis**:

- **Continus**: cada registre ocupa una línia, com en una taula; sota cada registre hi ha altre registre.
- **Simples**: cada registre es presenta en una pàgina completa. Sota un registre no hi ha altre registre. L'organització dels registres s'assembla a un llibre de fitxes: una pàgina per registre.

Un tercer tipus, **Fulla de dades**, permet utilitzar el formulari com si es tractés d'una taula.

Tant els formularis simples com els continus estan dividits en diverses parts. Aquestes parts poden o no estar presents en un formulari, i funcionen com objectes (controls) en el sentit que tenen les seves pròpies propietats. Els controls del formularis poden estar situats indistintament en cadascuna d'aquestes parts, depenent de la funcionalitat que es vulgui donar al control. Aquestes parts són:

- **Encapçalat del formulari**. Apareix sempre en la part superior del formulari.
- **Encapçalat de pàgina**. Apareix sempre en la part superior del formulari, entre l'Encapçalat del formulari i el Detall.
- **Detall**. Conté els registres. Si el formulari és de tipus continu, apareixen múltiples línies, una amb cada registre de la taula subjacent. Si és de tipus simple, només apareix un registre, que va canviant a l'avançar i retrocedir entre les pàgines (registres). Si és massa gran per a la grandària de la finestra, o hi ha més registres dels que hi caben (en manera contínua), el Detall es desplaçarà per a visualitzar les parts que no càpiguen, mentre que la resta de les parts del formulari no sofriran canvis.
- **Peu de pàgina**. Apareix sempre en la part inferior del formulari, entre el Detall i el Peu del formulari.
- **Peu del formulari**. Apareix sempre en la part inferior del formulari.

Les principals propietats d'un formulari són les següents:

- **Títol**: text que apareix en la part superior de la finestra.
- **Vista predeterminada**: formulari simple, continu o taula.
- **Vistes permeses**: es pot canviar entre simple/continu i taula durant l'execució?
- **Barres de desplaçament**: quines barres de desplaçament es mostraran, en cas que sigui necessari.
- **Selectors de registre**: indica si deu mostrar-se el requadre que permet seleccionar el registre complet.
- **Botons de desplaçament**: indica si els botons de la part inferior del formulari que serveixen per navegar entre els registres deuen mostrar-se o no.
- **Separadors de registres**: indica si deu mostrar-se una línia de separació entre els registres (formularis continus).
- **Ajustament de grandària automàtica**: indica si el formulari deu adaptar-se al seu contingut la pròxima vegada que s'obri.
- **Centrat automàtic**: indica si el formulari deu centrar-se en pantalla la pròxima vegada que s'obri.
- **Estil de les vores**: indica si es permet canviar la grandària de la finestra durant l'execució.
- **Quadre de control**: indica si deu existir el menú de finestra.
- **Botons Minimitzar Maximitzar, Botó tancar, Botó què és això**: indica si deuen aparèixer els botons de maximitzar, minimitzar, tancar i ajuda.
- **Ample**: indica l'ample del formulari.
- **Imatge, Tipus d'imatge, Manera de canviar la grandària de la imatge, Distribució de la imatge, Mosaic d'imatges**: permeten posar una imatge de fons en el formulari i la forma de col·locar la imatge.
- **Línia X, Línia Y**: grandària de les reixetes horitzontal i vertical: nombre de subdivisions horitzontals i verticals per unitat de mesura de la quadrícula d'alineació.
- **Origen del registre**: taula o consulta de la qual s'obtenen les dades.
- **Filtre**: expressió WHERE que deuen complir els registres perquè siguin visualitzats.
- **Ordenar per**: llista de camps pels quals s'ordenen els registres.
- **Permetre filtres**: indica si s'activen les opcions que permeten a l'usuari afegir filtres durant l'execució.
- **Permetre edicions**: indica si es permet modificar el contingut dels registres existents.
- **Permetre eliminació**: indica si es permet eliminar registres existents.
- **Permetre afegir**: indica si es permet afegir nous registres.
- **Entrada de dades**: indica si el formulari només permet afegir dades.

- **Bloquejos de registre:** indica la forma que deuen bloquejar-se els registres utilitzats per a permetre l'accés concurrent de diversos usuaris.
- **Emergent:** indica si el formulari deu quedar davant de les altres finestres d'Access.
- **Modal:** indica si es permet activar altres formularis mentre el formulari estigui executant-se.
- **Cicle:** indica si després de l'últim control, la tecla de tabulació deu saltar al primer control, al primer control visualitzat o al següent registre.
- **Barra de menús, Barra d'eines, Menú contextual, Barra de menús contextuais:** permeten canviar el menú i les barres d'eines d'Access mentre s'executa el formulari.
- **Arxiu d'ajuda, Id. del context d'ajuda:** permeten especificar un arxiu d'ajuda de Windows i l'índex del tema per al formulari.

En la llista anterior s'han omès les propietats del conjunt d'esdeveniments. Abans de tractar-les devem estudiar els esdeveniments.

## Esdeveniments

Un esdeveniment és una acció determinada que es produeix en o amb un objecte particular. Access pot respondre a una varietat d'esdeveniments: clics del ratolí, canvis en les dades, formularis que s'obren o es tanquen, i molts altres. Els esdeveniments són normalment el resultat d'una acció de l'usuari.

Cada objecte o control està preparat per a respondre a un conjunt d'esdeveniments, coherents amb el tipus de control i la seva utilitat. Es pot establir una macro per a cada esdeveniment de cada objecte, de manera que quan es produeix aquest esdeveniment en aquest objecte, s'executa el codi de la macro. Un exemple bàsic consisteix a associar una macro que obri un formulari a l'esdeveniment de pulsació d'un botó, de manera que quan el premi el botó, s'obri el formulari. D'aquesta forma es poden fer programes tan complexos com es vulgui amb els formularis d'Access.

Per a moltes situacions existeixen dos esdeveniments: un denominats "Abans de XXX" i un altre "Després de XXX". Ambdós esdeveniments es produeixen quan es dona la situació XXX, amb la diferència que un dels esdeveniments té lloc just abans que la situació XXX faci el seu efecte, i l'altre just a continuació.

A continuació es llisten els principals esdeveniments per a formularis i controls. Esdeveniments per als formularis

- **A l'activar registre:** quan l'enfocament passa a un nou registre.
- **Abans d'inserir, Després d'inserir:** quan s'afegeix un nou registre.
- **Abans d'actualitzar, Després d'actualitzar:** quan es modifica un registre.
- **A l'eliminar, Abans de confirmar l'eliminació, Després de confirmar l'eliminació:** a l'eliminar un registre existent.
- **A l'obrir:** a l'obrir el formulari.
- **Al carregar:** al carregar el formulari.
- **Al canviar la grandària:** quan es canvia la grandària de la finestra del formulari.
- **Al descarregar:** al descarregar el formulari.
- **Al tancar:** al tancar el formulari.
- **A l'activar:** quan el formulari és activat, és a dir, passa al capdavant de la pantalla i comença a funcionar.
- **Al desactivar:** quan el formulari és desactivat, és a dir, deixa d'estar al capdavant de la pantalla i s'activa altre.
- **Al rebre enfocament:** quan l'enfocament passa al formulari.
- **Al perdre enfocament:** quan l'enfocament passa del formulari a altre objecte.
- **Al fer clic:** quan es pressiona i s'allibera el botó esquerre del ratolí sobre la superfície del formulari.
- **Al fer doble clic:** quan fa doble clic amb el botó esquerre del ratolí sobre la superfície del formulari.
- **Al baixar el ratolí:** en el moment que es pressiona el botó esquerre del ratolí.
- **Al pujar el ratolí:** en el moment que s'allibera el botó esquerre del ratolí.
- **Al moure el ratolí:** en el moment que es desplaça el punter ratolí sobre la superfície del formulari.
- **Al baixar una tecla:** en el moment que es pressiona una tecla.
- **Al pujar una tecla:** en el moment que s'allibera una tecla.
- **Al pressionar una tecla:** quan es produeix una pulsació completa (pressionar i alliberar) d'una tecla.

- **Tecla de vista prèvia:** indica si els esdeveniments de pulsacions de teclat deuen afectar abans als controls del formulari o al propi formulari.
- **A l'ocórrer un error:** quan es produeix un error en temps d'execució.
- **Al filtrar:** quan es modifica el filtre sobre els registres de la taula associada.
- **A l'aplicar el filtre:** quan s'aplica o elimina un filtre sobre els registres de la taula associada.
- **Al cronòmetre:** cada cert temps.
- **Interval de cronòmetre:** nombre de mil·lisegons entre dos esdeveniments *Al cronòmetre* consecutius.

## Esdeveniments per als controls

Per a controls independents ("passius", no associats a camps de taules): textos, dibuixos, etc.:

- **Al fer clic:** quan es pressiona i s'allibera el botó esquerre del ratolí sobre la superfície del control.
- **Al fer doble clic:** quan fa doble clic amb el botó esquerre del ratolí sobre la superfície del control.
- **Al baixar el ratolí:** en el moment que es pressiona el botó esquerre del ratolí sobre el control.
- **Al pujar el ratolí:** en el moment que s'allibera el botó esquerre del ratolí sobre el control.
- **Al moure el ratolí:** en el moment que es desplaça el punter ratolí sobre la superfície del control.

Per a controls depenents ("actius", que contenen informació associada a camps de taules): editors, quadres de llista, etc. (a més dels esdeveniments anteriors):

- **Abans d'actualitzar, Després d'actualitzar:** quan es modifica el contingut del control (en el moment que es realitza el canvi definitiu sobre el camp o registre).
- **Al canviar:** al canviar el contingut del control, sense que sigui necessari que es canviï definitivament sobre el registre o camp.
- **A l'entrar, Al rebre l'enfocament:** quan el control rep l'enfocament des d'altre control.
- **Al sortir, Al perdre l'enfocament:** quan el control perd l'enfocament a favor d'altre control.
- **Al baixar una tecla:** en el moment que es pressiona una tecla.
- **Al pujar una tecla:** en el moment que s'allibera una tecla.
- **Al pressionar una tecla:** quan es produeix una pulsació completa (pressionar i alliberar) d'una tecla.

## Controls en un formulari

En un formulari es poden incloure tants controls com es desitgi. Cada control deu anar identificat per un nom, que pot ser diferent del nom del camp al que estigui associat (encara que sovint solen coincidir, per regularitat). Si s'ha definit la taula associada a un formulari, Access ens permet afegir un control per a cada camp de forma fàcil, simplement arrossegant-los amb el ratolí des d'una finestra que conté la llista de camps disponibles. Els controls així afegits adopten les propietats que es deriven dels atributs assignats en la definició dels camps de les taules d'origen.

Quan s'afegeixen controls a un formulari, aquests adopten una determinada aparença per defecte en quan a color, grandària, fonts, i altres propietats. Els valors per defecte d'aquests atributs poden ser canviats a través d'un control d'exemple (amb l'opció Definir valors predeterminats dels controls en el menú Format).

Els controls depenents, per defecte afegeixen automàticament una etiqueta al ser afegits. Aquesta etiqueta es mou o esborra al moure o esborrar el control, però no al revés.

Cada control té un nom. Es pot accedir al contingut d'un control a través del seu nom. Per exemple, podem establir una regla de validació que verifiqui que el valor del control actual no sigui superior al d'un control el nom del qual és "ValorMàxim". L'expressió que caldria introduir en la propietat "Regla de validació" del control seria " $\leq$ ValorMàxim".

Cada control dependent sol anar associat a un camp, però no és necessari. En tal cas el control actua com una mera variable el valor de la qual pot ser modificat per l'usuari. No cal confondre per tant el nom d'un control amb el nom del camp al que està associat, encara que sovint siguin iguals. Altra possibilitat és que els controls mostrin el resultat d'algun càlcul (ho veurem més endavant).

Al canviar de registre actiu, els controls associats als camps de la taula es carreguen automàticament amb els valors del nou registre. De la mateixa forma, les modificacions que es realitzin sobre les dades dels controls es reflecteixen automàticament en la taula.

## Propietats comunes

A continuació es llisten les principals propietats que tenen gairebé tots els controls:

- **Visible:** indica si el control deu mostrar-se o romandre ocult.
- **Mostrar quan:** indica si el control és visible en pantalla o a l'imprimir el formulari.
- **Esquerra:** indica la posició horitzontal, és a dir, la distància des de la vora esquerra del formulari, en cm.)
- **Superior:** indica la posició vertical, és a dir, la distància des de la vora superior de la secció actual del formulari (encapçalat, detall, peu...) en cm.
- **Ample, Alt:** dimensions del control, en cm.
- **Color del fons, Efecte especial, Estil de les vores, Ample de les vores, Color de les vores, Color del text, Nom de la font, Grandària de la font, Font en cursiva:** defineixen l'aspecte general del control. Els colors es defineixen com color real RGB, codificat com un nombre de 24 bits, 8 per a cada color. L'efecte especial permet donar un aspecte tridimensional al formulari.
- **Origen del control:** camp associat al control.
- **Valor predeterminat:** igual que en la definició dels camps de les taules.
- **Regla de validació:** igual que en la definició dels camps de les taules.
- **Text de validació:** igual que en la definició dels camps de les taules.
- **Activat:** indica si el control es pot utilitzar o està desactivat.
- **Bloquejat:** indica si el control es pot utilitzar o està inutilitzat.
- **Nom:** nom identificatiu del control, per a referències des d'altres controls.
- **Text de la barra d'estat:** text d'ajuda que apareix en la barra d'estat d'Access quan el control rep l'enfocament.
- **Punt de tabulació:** indica si es pot accedir al control prement la tecla de tabulació.
- **Índex de tabulació:** indica en quin nombre d'ordre s'accedeix al control prement la tecla de tabulació.
- **Text d'ajuda del control:** text flotant que apareix sobre el control al situar-se el punter del ratolí sobre el control.
- **Id. del context d'ajuda:** índex del tema d'ajuda relacionat amb el control dintre de l'arxiu d'ajuda establert pel formulari.
- **Informació addicional:** un comentari que pot emmagatzemar-se sobre l'objecte.

Com s'ha vist, l'associació d'un control a un camp es fa a través de la propietat **Origen del control**. Si aquesta propietat s'estableix al nom d'un camp, el control queda associat a aquest camp, i les modificacions en el control es reflectirà en les dades del camp. Si la propietat es deixa en blanc, el control és independent i actua com una variable. Finalment, si en la propietat **Origen del control** s'introdueix una expressió començant per un símbol igual (=), llavors el control conté un valor calculat: el resultat d'avaluar l'expressió.

Per exemple, si tenim un camp "Preu" amb el seu control corresponent amb el mateix nom, podem afegir un control cridat "PreuAmbIva" que ens mostri el resultat d'afegir-li l'IVA, sense que necessitem emmagatzemar-lo en cap camp. Això ho fariem escrivint la següent expressió en la propietat Origen del control del control PreuAmbIva: "=Preu \* 1,16". Cada vegada que canviï el contingut de Preu automàticament es recalcula el contingut de PreuAmbIva.

## Propietats dels controls de llistes

Aquestes propietats són comunes als controls de tipus *Quadre combinat* i *Quadre de llista*. Ambdós tenen en comú el fet que poden contenir una llista de valors permesos per a seleccionar el valor del control (i per tant del camp associat, si ho hagués).

Aquests controls tenen algunes propietats específiques que permeten manejar la informació referent al contingut de la llista de valors:

- **Tipus d'origen de la fila:** indica si el contingut de la llista es prendrà de les dades contingudes en alguna taula, d'una llista de valors explícitament especificada, o d'una llista de camps pertanyents a alguna taula.
- **Origen de la fila:** depenent del valor de la propietat anterior, especifica, bé la taula d'origen per a obtenir les dades o camps a mostrar, o bé la llista de valors permesos, separats per punt i coma (;).



- **Nombre de columnes:** especifica el nombre de columnes que es mostraran en la llista. Té especial significat quan els valors permesos s'obtenen d'una taula, la qual, pot contenir diversos camps. Aquesta propietat indica quants camps de la taula es mostraran en la llista. L'ordre d'aparició dels camps és el mateix que es trobin definits en la taula.
- **Encapçalats de columna:** indica si en la part superior de la llista deu aparèixer el títol de la columna, indicant el nom del camps visualitzat.
- **Ample de columnes:** especifica l'ample, en centímetres, de cadascuna de les columnes mostrades amb els valors permesos. Es tracta d'una llista amb l'ample de cada columna, separat per punt i coma (;). Es permet ample 0.
- **Columna dependent:** quan una llista de valors permesos es mostra amb més d'una columna, només una d'aquestes columnes és qui proporciona el valor per al camp. Aquesta propietat indica quin és el nombre d'ordre de la columna.
- **Files en llista:** indica quantes línies amb valors permesos deuen mostrar-se en pantalla com a màxim. Si hi ha més, es mostrarà una barra de desplaçament.
- **Ample de la llista:** indica l'ample total de la llista de valors. Per defecte (automàtic) l'ample de la llista és el mateix que la grandària en horitzontal que tingui el control de llista, i no la suma dels amplex de les columnes mostrades. Si no caben les columnes en l'ample especificat es mostra una barra de desplaçament horitzontal.
- **Limitar a llista:** indica si el valor acceptat pel control deu ser obligatòriament un dels visualitzats en la llista, o si, per contra, es permet que l'usuari introdueixi un altre.

## Subformularis

Algunes vegades resulta interessant imbricar formularis, és a dir, incloure formularis dintre d'altres formularis. Suposem com exemple una base de dades per a emmagatzemar les nostres cintes de música. Tenim dues taules: cançons i cintes:

Cançons		Cintes
IDcanco		IDcinta
Títol		Títol
Autor		Durada
Cinta		Propietari
Cara		

Suposem que volem presentar un formulari en el qual apareguin les dades d'una cinta, i totes les cançons que contingui. La forma més adequada de fer això és utilitzant subformularis. El procés és el següent:

1. **Construcció del subformulari.** En el nostre cas, es tracta del formulari que conté les dades de les cançons.
2. **Construcció del formulari principal.** En el nostre cas, el de cintes. Deu contenir un control de tipus subformulari, que es refereixi al formulari de cançons.
3. **Enllaç dels formularis.** Cal associar dos camps, un en el formulari principal i altre en el subformulari, de manera que en el subformulari només es mostrin les files que continguin un camp el valor del qual coincideixi amb el d'un determinat camp en el principal. Així, no es mostraran totes les cançons, sinó només aquelles que pertanyin a la cinta que es mostra en el principal. Es tracta dels camps que estableixen la relació entre ambdues taules: *Cançons.Cinta* i *Cintes.IDcinta*.

Per a la construcció del subformulari, hem de tenir en compte el següent:

- Probablement ens convingui utilitzar un formulari de tipus continu: moltes cançons per a la mateixa cinta; sembla lògic que sigui un formulari de tipus continu (com una taula).
- Probablement el camp que enllaçarà el subformulari amb el formulari principal (*Cançons.Cinta*) no es mostri en el subformulari: sembla que no té sentit que per a totes les cançons de la cinta es mostri en el formulari repetit el mateix valor de la clau aliena. A més, aquest camp serà omplert per Access de forma automàtica al valor adequat a l'afegir noves files a la taula de cançons.
- No cal establir filtres ni condicions sobre el camp dependent, ja que Access ho farà automàticament a l'establir la dependència.
- Probablement sigui adequat no introduir encapçalats i peus en el subformulari, ja que a l'anar integrat dintre d'altre formulari, aquestes parts podem introduir-les directament en el formulari principal.

En quan al formulari principal, cal tenir el compte que a l'introduir el control de subformulari, cal donar-li la grandària i forma adequat perquè càpiga en ell el subformulari complet. Si és de tipus continu, cal donar-li l'ample adequat perquè no sobri ni falti espai, i l'alt adequat perquè càpiguen el nombre de files desitjat.

Finalment, cal establir les següents propietats del control subformulari:

- **Objecte origen:** nom del formulari utilitzat com subformulari.
- **Vincular camps secundaris:** nom del camp dependent en el subformulari (que estableix la relació entre el formulari principal i el subformulari: *Cançons.Cinta*).
- **Vincular camps principals:** nom del camp o control dependent en el formulari principal (que estableix la relació entre el formulari principal i el subformulari: *Cintes.IDcinta*). En el subformulari només es mostraran les files per a les quals el valor del camp establert a *Vincular camps secundaris* coincideixi amb el del control establert per a aquesta propietat.

Els camps de vinculació no han d'estar inclosos amb controls en el formulari principal o en el subformulari. És suficient amb que estiguin en les taules o consultes base d'aquests.

## Exercici 13: formularis

### Creació d'un formulari per a Botigues

Anem a generar un formulari per a la taula de Botigues que permeti afegir, modificar o eliminar les botigues de la base de dades. Es desitja que tingui el següent aspecte:

Cal tenir en compte els següents punts:

- El primer és establir la taula d'origen, la qual cosa facilitarà la tasca.
- És un **formulari de tipus simple**.
- **Els editors de dades deuen ser com els utilitzats en les taules:** quadres de text amb màscara d'entrada per a l'identificador i quadre combinat per a la ciutat.
- Atenció a l'**ordre de tabulació**.

### Creació d'un formulari per a BExistències

Ara pretenem fer un formulari que ens permeti afegir, modificar i eliminar les entrades de la taula de *BExistències*. L'aspecte desitjat per a aquest formulari és el que es mostra en les figures següents. Caldrà tenir en compte:

- Per a cada registre es podran editar els codis de botiga i producte, així com la quantitat. A més, apareixeran **dades addicionals sobre la botiga i el producte seleccionat**.
- Per a introduir els codis de botigues i productes cal utilitzar **quadres combinats**, i **quadres de text** per a la quantitat. La informació addicional s'escriu mitjançant quadres de text, però que **han de ser desactivats** perquè no puguin ser editats. És necessari definir com origen de dades per als quadres combinats la taula de Botigues o de Productes, indicar el nombre de columnes que es mostraran, l'ample de cadascuna d'elles, l'ample total, el nombre de files mostrades i la columna dependent.
- El primer és establir la taula d'origen. En aquest cas, desitgem que sota l'editor del codi de botiga i del codi de producte es mostri la informació de la botiga i del producte corresponent. Per això, l'origen de les dades del formulari no serà la taula *BExistències*,

sinó que **ens caldrà una consulta**. La informació necessària s'obté mitjançant la unió (join) de les taules *BExistències*, *Botigues* i *Productes*. La consulta serà:

```
SELECT * FROM Productes INNER JOIN (Botigues INNER JOIN
BExistències ON Botigues.Bid = BExistències.Bid) ON
Productes.Pid = BExistències.Pid;
```

- Es tracta d'un **formulari continu**. Atenció a l'encapçalat de formulari.
- Per a obtenir " Kg." després del pes cal **utilitzar un quadre de text com resultat d'expressió**. A més aquest quadre de text no podrà tenir el mateix nom que cap altre camp del formulari, perquè en cas contrari s'obtidran resultats inesperats.

## Exercici 14

Es proposa crear un formulari similar a l'anterior (*BExistències*), però per a la taula de Comandes. El formulari deu tenir les següents característiques:

- Serà de **tipus simple** (en comptes de continu).
- Per a cada registre es podran modificar els camps de codi de comanda, codi de producte, codi de botiga, codi de magatzem, quantitat, i data.
- Per al codi de comanda, la quantitat i la data s'utilitzaran **quadres de text**.
- Per a la resta de codis s'utilitzaran **quadres combinats**, obtenint-se la informació de les taules corresponents. A més, sota aquests controls **s'indicarà la informació addicional** com s'ha fet en el formulari de *BExistències*: adreça i ciutat de botigues i magatzems; i nom, pes i preu de productes.

## Formularis imbricats

Ara pretenem tenir un formulari de botigues en el qual ens apareguin totes les existències de cada botiga al costat de les dades de la botiga, com en el següent formulari:

Per a això cal modificar el formulari creat per a la taula de botigues, i cal afegir-li un control de tipus subformulari. Però abans cal generar el formulari que s'introduirà com subformulari, el qual obtindrem modificant el d'existències en botigues generat anteriorment.

El nou aspecte del formulari d'existències serà el següent:

P26	Video v45	42
P25	Video v33	22
P10	Microondas mc1	10
P04	Televisor M2	130
P03	Televisor M1	17
P26	Video v45	23
P25	Video v33	12
P10	Microondas mc1	80
P03	Televisor M1	4
P10	Microondas mc1	125
P04	Televisor M2	89
P03	Televisor M1	20
P26	Video v45	16
P25	Video v33	12
P10	Microondas mc1	9
*		

Les modificacions realitzades són l'eliminació de capçaleres i botons de desplaçament de registres, així com de tot l'espai sobrant, amb la finalitat d'ocupar el menor espai possible i donar un aspecte ordenat. A més, s'ha eliminat el camp corresponent al codi de botiga. El formulari modificat deu guardar-se en disc abans de continuar.

En el formulari principal cal introduir un control de tipus subformulari. A continuació s'associa aquest control al formulari anteriorment creat i finalment cal indicar els camps vinculats: **Bid** en el formulari principal i **BExistències.Bid** en el subformulari. Així, quan es canviï de registre, al canviar el codi de botiga (**Bid**), en el subformulari només apareixeran els registres el camp dels quals **BExistències.Bid** coincideixi amb **Bid**. Aquesta és la raó per la qual es va eliminar el camp corresponent al definir el subformulari (Access ho emplena automàticament, i apareixerien en pantalla tots els registres amb el mateix valor).

## Exercici 15

Es proposa realitzar els mateixos formularis que els desenvolupats fins ara, però orientats a les taules de Magatzems i MExistències.

# Informes

## Introducció

Els informes són formats de presentació de dades per a la recuperació de dades per impressora. Són semblants als formularis quant a la forma de definir-los i manejar-los, amb la diferència que en els informes apareixen algunes característiques pròpies del mitjà físic al que van orientats, és a dir, el paper (i no la pantalla).

Tal és la similitud entre formularis i informes que Access utilitza les mateixes eines i controls per a ambdós. Fins i tot és possible convertir un formulari a un informe. Al cap i a la fi, els formularis també podien imprimir-se.

Tot l'estudiat en el capítol anterior sobre disseny i manipulació de formularis és aplicable als informes. A més, estudiarem les possibilitats addicionals dels informes, que no existien en els formularis per pròpia naturalesa.

## Controls en informes

No existeix cap diferència en l'ús de controls en informes respecte a l'estudiat per a formulari. Els controls són idèntics i conserven totes les propietats presents per als formularis, excepte aquelles que es refereixen a la modificació de dades (regles de validació, esdeveniments sobre la modificació, etc...), ja que els informes no permeten la modificació de les dades, sinó només mostrar el contingut de les taules o consultes.

Adquireix un especial significat el control de salt de pàgina, que no s'utilitzava en els formularis perquè no existia el concepte de pàgina (excepte a l'imprimir).

## Agrupament de dades en informes

Una característica de gran utilitat en el disseny d'informes és la possibilitat d'agrupar les dades segons el valor d'un camp o conjunt de camps. A diferència d'una consulta amb agrupament (que utilitzi GROUP BY), aquest tipus d'agrupament que ofereixen els informes no es limita a la utilització de funcions d'agregat, sinó que permet introduir separadors entre conjunts de registres amb una característica comuna (que serveix com clàusula d'agrupament).

Per exemple, suposem un gran llistat d'alumnes admesos per a la realització d'un curs. Per a generar aquest llistat sembla interessant ordenar-los alfabèticament i imprimir una lletra en gran al principi de cada grup per a facilitar la recerca, com en la figura 8.

Access permet agrupar les dades en múltiples conjunts imbricats (p.ex. en un llistat d'alumnes de la Universitat, es pot agrupar primer per escola, després per carrera i després per la primera lletra del cognom). Els grups es poden generar en ordre creixent o decreixent, i per a cadascun d'aquests grups pot existir o no un encapçalat i un peu de grup. L'encapçalat de grup apareix just abans dels registres que pertanyen al grup, i el peu just després. Típicament en els encapçalats i peus es mostren funcions d'agregat calculades sobre els registres que pertanyen al grup (p.ex. compte d'elements, totals...). Això es pot fer fàcilment amb quadres de text amb funcions del tipus "`=Suma([Preu] * [Quantitat])`" (que calcularia la suma total per a una secció del producte dels camps Preu i Quantitat de cada registre).

Els grups es poden definir sobre valors d'un determinat camp o sobre valors calculats sobre algun camp o conjunt de camps. En aquest sentit poden resultar molt útils funcions com les de la següent taula:

Funció	Significat
Izq(Cadena; Nombre)	Proporciona els <i>Nombre</i> primers caràcters d'una <i>Cadena</i> .
Der(Cadena; Nombre)	Proporciona els <i>Nombre</i> últims caràcters d'una <i>Cadena</i> .
Medio(Cadena; Posició; Nombre)	Proporciona els <i>Nombre</i> caràcters d'una <i>Cadena</i> , començant per <i>Posició</i> .
Año(Data)	Proporciona l'any contingut en una variable o camp de tipus data/hora.
Mes(Data)	Proporciona el mes contingut en una variable o camp de tipus data/hora.
Día(Data)	Proporciona el dia contingut en una variable o camp de tipus

Funció	Significat
	data/hora.
Hora (Data)	Proporciona l'hora continguda en una variable o camp de tipus data/hora.
Minuto (Data)	Proporciona els minuts continguts en una variable o camp de tipus data/hora.
Segundo (Data)	Proporciona els segons continguts en una variable o camp de tipus data/hora.

[illegible]

**Figura 8. Llistat agrupat per primera inicial.**

A l'accedir al diàleg de definició de grups, Access ens mostra les següents opcions:

- **Camp/Expressió:** permet identificar els camps o expressions que generaran els grups, per exemple el camp *Cognom1*. El camp o expressió deu estar referida a les dades de la taula o consulta que es basi l'informe.
- **Ordre:** per a cada grup, indica si els inicis de cada grup deuen anar apareixent de forma ascendent o descendent. Per exemple, si agrupem per dates, l'ordre pot ser de dates ascendents o descendents. No es refereix a l'ordre dels registres pertanyents al grup, sinó als encapçalats de grup, als grups en si.
- **Encapçalat del grup:** per a cada grup, indica si deu incloure's una secció d'encapçalat del grup, que apareixerà just abans que els registres del grup.
- **Peu del grup:** per a cada grup, indica si deu incloure's una secció de peu del grup, que apareixerà just després que els registres del grup.
- **Agrupar en:** permet especificar una forma precisa de generar els grups. Depenent del tipus del camp o expressió per la qual s'agrupi, podrem seleccionar:
  - **Camps de tipus text:** es pot seleccionar que s'agrupi pel camp/expressió completa o pels *n* primers caràcters.
  - **Camps de tipus data:** es pot seleccionar que s'agrupi pel camp/expressió completa o any, mes, dia, hora, minut, segon, trimestre, setmana, etc...
  - **Camps numèrics:** es pot seleccionar que s'agrupi pel número en concret o per intervals del número.
- **Interval del grup:** indica cada quants valors distints del criteri d'agrupació deu crear-se un grup. Per exemple, si es decideix agrupar per dates, i a més per setmanes, i aquesta propietat s'estableix a 2, es faran grups per a dates que es diferenciïn en dues setmanes.
- **Mantenir junts:** indica si Access deu disposar tots els registres d'un grup en la mateixa pàgina (sempre que no ocupin més d'una pàgina), començant si és necessari una nova pàgina al principi del grup.

## ***Format de pàgina i columnes***

Com ja sabem, els informes utilitzen com suport bàsic el paper i la impressora. Aquest és un mitjà amb moltes possibilitats de configuració: existeixen diversos tipus d'impressores i infinitat de

grandàries i formes de paper. Els informes deuen dissenyar-se, fins cert punt, tenint en compte el paper sobre el qual aniran impresos.

Aquesta configuració, juntament amb altres opcions de presentació es troben en l'opció *Configurar pàgina* del menú *Arxiu*, i deu establir-se durant el disseny de l'informe (encara que podrà modificar-se en el moment d'imprimir). Les opcions disponibles d'Access són:

- **Impressora** (la predeterminada de Windows o una en concret).
- **Grandària, orientació i origen** del paper.
- **Marges**.
- **Nombre de columnes**. Es refereix al nombre de columnes per pàgina que s'utilitzaran per a llistar les dades en la secció de *Detall*. A més, es pot especificar la separació entre columnes i el sentit d'escriptura al llarg de les columnes.

És important establir correctament aquestes opcions. L'àrea disponible per introduir els controls no es veu limitada per la grandària del paper; no obstant això, a l'hora d'imprimir, el nostre disseny pot quedar dividit en diverses fulles si hem excedit la superfície disponible (que es pot calcular fàcilment coneixent la grandària del paper i els marges, ja que Access situa els controls utilitzant centímetres com unitat de mesura).

Altres detalls importants de conèixer són l'ample de les columnes. Quan només es treballa a una columna, que és el més habitual, no existeix major problema: l'ample de la columna coincideix amb l'ample de totes les seccions. No obstant això, al treballar a dues o més columnes, existeixen dues possibilitats:

- **Les columnes tenen el mateix ample que totes les seccions**, encara que realment només s'escriu a dues columnes la secció de detall.
- **Les columnes tenen un ample fix determinat**, que no és el mateix que el de totes les seccions, ni tan sols el de la de detall. Així, els encapçalats i peus poden tenir major (o menor) ample que les columnes, la qual cosa pot donar un aspecte més elegant a l'informe. (Així s'ha fet en l'exemple de la figura 8)

## Funcions interessants en informes

Les següents funcions poden resultar interessants a l'hora de confeccionar informes:

Funció	Significat
Página	Nombre de pàgina actual (per numerar les pàgines).
Páginas	Nombre de pàgines total en l'informe.
Fecha()	Data que s'imprimeix.
Ahora()	Data i hora que s'imprimeix.

## Exercici 16: informes

Crear un informe que llisti tota la informació dels productes dels magatzems, agrupant-los per magatzems. Haurà de tenir un aspecte similar a l'informe de la figura 9.

## Existències en els magatzems

Llistat obtingut el 26/06/2004

### M02 Polígono Sur, s/n Huelva

Identificador	Descripció	Pe s	Preu	Quantitat
P03	Televisor M1	27 Kg	336,00 €	250
P04	Televisor M2	35 Kg	570,00 €	125
P10	Microondas mcl	12 Kg	102,00 €	710
P12	Secador sx	1 Kg	15,00 €	25
P13	Lavadora m100	95 Kg	276,00 €	75
P25	Vídeo v33	8 Kg	270,00 €	25
P26	Vídeo v45	10 Kg	222,00 €	100
Total inversió emmagatzemada			277.695,00 €	

### M03 Carretera Murcia, 124 Murcia

Identificador	Descripció	Pe s	Preu	Quantitat
P04	Televisor M2	35 Kg	570,00 €	210
P10	Microondas mcl	12 Kg	102,00 €	14
P25	Vídeo v33	8 Kg	270,00 €	50
Total Inversió emmagatzemada			134.626,00 €	

### M05 Polígono Alhambra, 322 Granada

Identificador	Descripció	Pe s	Preu	Quantitat
P03	Televisor M1	27 Kg	336,00 €	40
P25	Vídeo v33	8 Kg	270,00 €	100
P26	Vídeo v45	10 Kg	222,00 €	12
Total inversió emmagatzemada			43.104,00 €	

Pàgina 1 de 1

Figura 9. Llistat de productes agrupats per magatzems.



## Macros

Una macro és un **conjunt d'una o més accions que cadascuna realitza una operació** determinada, tal com obrir un formulari o imprimir un informe. Les macros poden ajudar a automatitzar les tasques comunes. Per exemple es pot executar una macro que imprimeixi un informe quan l'usuari pressioni un botó de comando.

**Una macro consta d'una seqüència d'accions**, cadascuna de les quals pot realitzar una operació d'un conjunt d'operacions possibles. De forma opcional, per a cada acció es pot indicar una condició lògica de manera que l'acció només s'executi si la condició s'avalua com veritable.

## Operacions amb macros

La següent taula conté les possibles accions, agrupades per tipus de tasca:

Categoria	Acció	Tasca
Dades de formularis i informes	<i>AplicarFiltro</i>	Restringeix dades
	<i>BuscarRegistro</i> <i>BuscarSiguiente</i> <i>IrAControl</i> <i>IrAPágina</i> <i>IrARegistro</i>	Es mou per les dades
Execució	<i>EjecutarComando</i>	Executa un comando
	<i>Salir</i>	Surt de Microsoft Access
	<i>AbrirConsulta</i> <i>EjecutarCódigo</i> <i>EjecutarMacro</i> <i>EjecutarSQL</i>	Executa una macro, procediment o consulta
	<i>EjecutarAplicación</i>	Executa altra aplicació
	<i>CancelarEvento</i> <i>DetenerTodasMacros</i> <i>DetenerMacro</i> <i>Salir</i>	Atura l'execució
Importar/exportar	<i>EnviarObjeto</i> <i>SalidaHacia</i>	Envia objectes Microsoft Access a altres aplicacions
	<i>TransferirBaseDatos</i> <i>TransferirHojaCálculo</i> <i>TransferirTexto</i>	Transfereix dades entre formats de dades de Microsoft Access i altres
Manipulació d'objectes	<i>CambiarNombre</i> <i>CopiarObjeto</i> <i>Guardar</i>	Copia, guarda o canvia el nom d'un objecte
	<i>EliminarObjeto</i>	Elimina un objecte
	<i>DesplazarTamaño</i> <i>Maximizar</i> <i>Minimizar</i> <i>Restaurar</i>	Mou o canvia la grandària d'una finestra
	<i>AbrirConsulta</i> <i>AbrirFormulario</i> <i>AbrirInforme</i> <i>AbrirMódulo</i> <i>AbrirTabla</i> <i>Cerrar</i>	Obre o tanca un objecte
	<i>AbrirConsulta</i> <i>AbrirFormulario</i> <i>AbrirInforme</i> <i>Imprimir</i>	Imprimeix un objecte
	<i>SeleccionarObjeto</i>	Selecciona un objecte

Categoria	Acció	Tasca
	<i>EstablecerValor</i>	Estableix el valor d'un camp, control o propietat
	<i>MostrarTodosRegistros</i> <i>NuevaConsulta</i> <i>RepintarObjeto</i>	Actualitza dades o la pantalla
Diversos	<i>AgregarMenú</i>	Crea una barra de menús personalitzada, un menú contextual personalitzat, una barra de menús global, o menú contextual global
	<i>EstablecerElementoDelMenú</i>	Estableix l'estat dels elements de menú en una barra de menús personalitzada o en una barra de menús global
	<i>CuadroMsj</i> <i>Eco</i> <i>EstablecerAdvertencias</i> <i>RelojDeArena</i>	Mostra informació per pantalla
	<i>EnviarTeclas</i>	Genera pulsacions de tecles
	<i>MostrarBarraDeHerramientas</i>	Mostra o oculta la barra de comandos incorporada o personalitzada
	<i>Bip</i>	Produeix un avis sonor

La següent taula conté totes les accions possibles en ordre alfabètic, juntament amb la seva descripció:

Acció	Descripció
<i>AbrirConsulta</i>	Obre un objecte de tipus consulta en qualsevol de les maneres possibles.
<i>AbrirFormulario</i>	Obre un objecte de tipus formulari en qualsevol de les maneres possibles.
<i>AbrirInforme</i>	Obre un objecte de tipus informe en qualsevol de les maneres possibles, o l'envia a la impressora.
<i>AbrirMódulo</i>	Obre un procediment o funció d'un mòdul en vista disseny.
<i>AbrirTabla</i>	Obre un objecte de tipus taula en qualsevol de les maneres possibles, amb opcions especials per a afegir, editar o només lectura.
<i>AgregarMenú</i>	Permet afegir una opció al menú Aplica una condició WHERE o una consulta de selecció sobre les dades del formulari actiu.
<i>AplicarFiltro</i>	Emet un so.
<i>Bip</i>	Busca un registre que contingui un valor en un camp, utilitzant les opcions habituals per a buscar en Windows.
<i>BuscarRegistro</i>	Continua-la última recerca efectuada.
<i>BuscarSiguiente</i>	Canvia el nom d'un objecte de tipus taula, consulta, formulari, informe, macro o mòdul.
<i>CambiarNombre</i>	Cancel·la l'esdeveniment que va activar la macro i l'acció que va elevar l'esdeveniment (p.ex. l'actualització d'un registre)
<i>CancelarEvento</i>	Tanca l'objecte (taula, consulta, formulari, informe, macro o mòdul) actual o un determinat, amb opcions de guardar els canvis.
<i>Cerrar</i>	Reprodueix l'objecte (taula, consulta, formulari, informe, macro o mòdul) indicat en la base de dades (amb altre nom) o en altra base de dades.
<i>CopiarObjeto</i>	Mostra un diàleg estàndard de Windows, indicant un missatge, amb una icona, so i botons estàndard.
<i>CuadroMsj</i>	Canvia la posició i/o grandària de la finestra (objecte) activa.
<i>DesplazarTamaño</i>	Acaba l'execució de la macro.
<i>DetenerMacro</i>	Acaba l'execució de la macro actual i de totes les quals s'estan executant (que van invocar a l'actual).
<i>DetenerTodasMacros</i>	Estableix si es mostren o no els missatges emesos per la macro que s'està executant, mostrant opcionalment un missatge en la barra inferior.
<i>Eco</i>	Els missatges d'error no es veuen afectats.

Acció	Descripció
<i>EjecutarAplicación</i>	Llança un programa Windows o MS-DOS.
<i>EjecutarCódigo</i>	Executa un procediment o funció d'un mòdul (en Visual Basic).
<i>EjecutarComando</i>	Executa un comando d'un menú de Access.
<i>EjecutarMacro</i>	Executa una macro. Quan acabi la nova macro es continua per l'actual.
<i>EjecutarSQL</i>	Executa una consulta SQL (d'inserció, eliminació, actualització...).
<i>EliminarObjeto</i>	Esborra un objecte determinat de la base de dades (taula, consulta, formulari, informe, macro o mòdul), o el seleccionat en la finestra de la base de dades.
<i>EnviarObjeto</i>	Envia un objecte de la base de dades per correu electrònic.
<i>EnviarTeclas</i>	Simula la pulsació d'una seqüència de tecles i executa les accions que es deriven de la pulsació.
<i>EstablecerAdvertencias</i>	Especifica si deuen mostrar-se missatges de confirmació abans de determinades accions, o s'assumirà que s'accepta.
<i>EstablecerElementoDelMenú</i>	Activa, desactiva, bloqueja o desbloqueja una opció d'un menú.
<i>EstablecerValor</i>	Estableix el valor contingut en un control d'un formulari o informe.
<i>Guardar</i>	Guarda l'objecte actiu (taula, consulta, formulari, informe, macro o mòdul) o un objecte determinat.
<i>Imprimir</i>	Imprimeix l'objecte de la base de dades (taula, consulta, formulari, informe, macro o mòdul) actiu, amb les opcions habituals de rang de pàgines, nombre de còpies, etc.
<i>IrAControl</i>	Duu l'enfocament a un control determinat del formulari actual.
<i>IrAPágina</i>	Duu l'enfocament al primer control de la pàgina especificada d'un formulari.
<i>IrARegistro</i>	Duu l'enfocament al registre indicat.
<i>Maximizar</i>	Maximitza la finestra activa.
<i>Minimizar</i>	Minimitza la finestra activa.
<i>MostrarBarraDeHerramientas</i>	Mostra o oculta una barra d'eines.
<i>MostrarTodosRegistros</i>	Elimina els filtres aplicats a les dades d'una taula, formulari o informe, de manera que tornen a aparèixer tots els registres existents.
<i>NuevaConsulta</i>	Recarrega les dades en un control del formulari actiu, o recalcula el seu valor.
<i>RelojDeArena</i>	Canvia l'aspecte del cursor del ratolí a un rellotge de sorra o el seu aspecte predeterminat.
<i>RepintarObjeto</i>	Torna a dibuixar en pantalla (refrescar) l'objecte (taula, consulta, formulari, informe, macro o mòdul) indicat.
<i>Restaurar</i>	Restaura a la seva grandària normal una finestra, després de maximitzar o minimitzar.
<i>SalidaHacia</i>	Exporta les dades d'un objecte de la base de dades (taula, consulta, formulari, informe, macro o mòdul) a format RTF (Rich Text Format), Microsoft Excel, HTML o text.
<i>Salir</i>	Surt de Microsoft Access, amb opcions per a guardar els últims canvis.
<i>SeleccionarObjeto</i>	Permet activar un objecte per a continuació executar altra acció que s'apliqui sobre l'objecte actiu.
<i>TransferirBaseDatos</i>	Permet importar/exportar objectes de la base de dades des de/a altres bases de dades, així com vincular taules amb bases de dades externes.
<i>TransferirHojaCálculo</i>	Importa, exporta o vincula les dades d'una full de càlcul Microsoft Excel.
<i>TransferirTexto</i>	Importa, exporta o vincula les dades d'un fitxer de text.

## Construcció de macros

A l'iniciar la construcció d'una macro, per defecte Access mostra una taula amb dues columnes, una per a **establir una acció** i altra per a **introduir un comentari** descriptiu de l'acció. A més, per a cada acció es mostra una llista de propietats en la part inferior de la pantalla que permeten establir els objectes i maneres sobre els quals opera l'acció. Per exemple, si es realitzarà una acció de *AbrirFormulario*, caldrà indicar quin formulari deu obrir-se, i de quina manera. Per a cada tipus d'acció existeix un conjunt de propietats, algunes d'elles opcionals i altres necessàries.

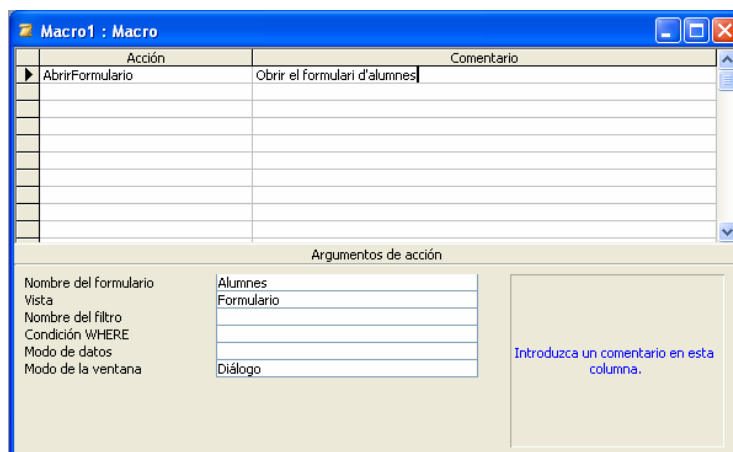


Figura 10. Taula d'accions per a la construcció de macros.

Adicionalment es poden afegir altres dues columnes a la taula d'accions: una per **establir expressions que condicionen l'execució de l'acció** (només s'executa si l'expressió s'avalua com veritable) i altra que permet **organitzar les macros** de forma imbricada. Ambdues característiques s'estudiaran més endavant.

Una vegada finalitzada la definició de la macro, abans de ser executada deu ser guardada en la base de dades amb un nom identificatiu, el qual serà emprat per a la seva referència des de formularis o informes per a establir-se com conductor d'esdeveniments. També és possible executar una macro de forma independent des de la finestra principal de la base de dades.

## Referències als objectes dels formularis o informes

En nombroses accions de macros és necessari consultar el valor o fer algun tipus de **referència als controls dels formularis o informes**. Recordem que cada control en un formulari o informe, així com els propis formularis/informes estan identificats mitjançant una propietat *nom*.

Cada control dintre d'un formulari o informe té un nom identificatiu únic. **Mitjançant aquest nom podem accedir al valor que té emmagatzemat**. Seguint una estratègia similar a la utilitzada en programació orientada a objectes, es pot identificar un control dintre d'un formulari o informe nomenant un **camí complet que permeti arribar al control**, partint de la base de dades i identificant cada pas amb el nom d'un objecte.

Inicialment existeixen dos grups de controls, els pertanyents a formularis i els pertanyents a informes. Ja que formularis i informes no comparteixen l'espai de noms, una identificació global d'un control en la base de dades deu reflectir aquest extrem, i deu començar per establir el grup al que pertany. A més, caldrà nomenar el formulari o informe concret i finalment el control. La següent taula mostra alguns exemples de referències al contingut de controls en una base de dades:

Control	Significat
Formularios![Magatzems]![Mid]	Contingut del control <i>Mid</i> del formulari <i>Magatzems</i> .
Informes![Magatzems]![Mid]	Contingut del control <i>Mid</i> de l'informe <i>Magatzems</i> .
Formularios![Notes Alumnes]![Cognom1]	Contingut del control <i>Cognom1</i> del formulari <i>Notes Alumnes</i> .

A més del contingut d'un control, es pot accedir a totes les seves propietats. Per a això n'hi ha prou amb concatenar el nom de la propietat. Per exemple:

Propietat de control	Significat
Formularios![Magatzems]![Mid].ColorDelFondo	Número del color de fons del control <i>Mid</i> del formulari <i>Magatzems</i> .
Informes![Magatzems]![Mid].NombreDeFuente	Nom de la font del text del control <i>Mid</i> de l'informe <i>Magatzems</i> .
Formularios![Notes Alumnes]![Cognom1].Altura	Altura (en centímetres ) del control <i>Cognom1</i> del


Propietat de control	Significat
	formulari <i>Notes Alumnes</i> .

Perquè una referència a un control d'un formulari o informe sigui vàlida, el formulari o informe concret **deu estar obert**, encara que no sigui el que estigui actiu en aquest moment. En un formulari de tipus continu existeixen múltiples "còpies" dels controls situats en la secció *Detall*. Llavors, una referència al contingut d'un d'aquests controls afectarà al control que contingui informació del **registre actiu** (en edició) en aquest moment.

En determinades situacions no és necessari utilitzar la identificació global dels controls. Per exemple, els controls d'un formulari es poden referenciar des del propi formulari només amb el nom del control, sense necessitat d'especificar el grup de formularis i el nom de formulari. També és possible referenciar controls de formularis o informes des de macros només amb el nom del control quan les macros s'executen com a conseqüència d'esdeveniments elevats des d'aquest formulari o informe.

La identificació de controls es pot realitzar des de i cap a qualsevol objecte de la base de dades: taules, consultes, formularis, informes, macros i mòduls.

## Execució condicional

Com ja es va introduir anteriorment, és possible afegir una columna a la taula d'accions d'una macro per **establir expressions condicionals que regulin l'execució de les accions** (cal prémer el botó  o l'opció corresponent del menú *Veure*). Per a cada acció es pot especificar una condició lògica. Aquesta expressió s'avalua abans d'executar l'acció, i si el resultat de l'expressió és veritable, l'acció s'efectua, i si no, s'ignora. Si per a una acció s'omet l'expressió condicional, se suposa que és una expressió veritable i l'acció corresponent s'executa sempre.

Si una seqüència d'accions depenen de la mateixa expressió no és necessari replicar l'expressió per a cada acció. L'expressió "..." (tres punts) indica a Access que la condició que regula a una acció és la mateixa que la de l'acció anterior.

## Bucles

L'única forma d'executar bucles dintre de les macros és mitjançant l'ús de l'acció **EjecutarMacro**. Les propietats d'aquesta acció són:

- **Nom de la macro:** macro que cal executar.
- **Nombre de repeticions:** nombre de vegades que cal executar la macro.
- **Expressió de repetició:** Expressió que indica continuació en el bucle: el bucle acaba quan l'expressió sigui falsa o acabi el nombre d'iteracions establert en *Numero de repeticions*.

## Exemple de construcció i ús de macros

Suposem que tenim un formulari per introduir característiques de clients. La nostra empresa ofereix descomptes especials a clients menors de 25 anys, a discapacitats i a posseïdors d'un carnet de soci segons el següent criteri:

Tipus	Descompte
Menors de 25 anys	5%
Discapacitats	10%
Socis	20%
Socis discapacitats	25%
Discapacitats menors de 25 anys	10%
Socis menors de 25 anys	NO PERMESOS

La taula que emmagatzema aquesta informació consta dels següents camps (el *Descompte*, encara que es pot calcular a partir dels altres camps, s'emmagatzema en la taula per raons d'eficiència):

Camp	Comentaris
Cognoms, Nom, DNI	Text.
Edat	Numèric.
Discapacitat	Text de longitud 1, amb dues úniques possibilitats: "S" o "N"
NumSoci	Numèric. Nul si no es soci.
Descompte	Numèric entre 0 i 100.

Tenim un formulari amb un control per a cada camp, amb el mateix nom que el camp. El formulari deu establir de forma automàtica el descompte a partir de les característiques del client. A més deu detectar les situacions prohibides, com un client menor de 25 anys.

**Figura 11. Aspecte del formulari descrit**

Vegem les macros necessàries per al formulari, així com els esdeveniments associats a cada macro. Suposem que el formulari es diu "Clients" i que existeixen regles de validació per a verificar que el contingut del camp minusvàlid és "S" o "N" i que el descompte està comprès entre 0 i 100.

Necessitarem les següents macros:

Nom de macro	Descripció	Esdeveniments als que s'associa
Calcular descompte	Calcula el descompte a partir de les dades del client, sempre i quan el descompte estigui en blanc.	Esdeveniment "Después de actualizar" del control <i>Edat</i> . Esdeveniment "Después de actualizar" del control <i>Discapacitat</i> . Esdeveniment "Después de actualizar" del control <i>NumSoci</i> .
Comprovar validesa	Comprova que no hagi situacions prohibides com socis menors de 25 anys.	Esdeveniment "Antes de actualizar" del control <i>Edat</i> . Esdeveniment "Antes de actualizar" del control <i>NumSoci</i> .

El contingut de la macro **Calcular descompte** serà el següent:

Condicció	Acció	Descripció	Propietats addicionals
([Edat]<25) Y ([Discapacitat]<>"S")	EstablecerValor	Posa descompte al 5% per a menors no discapacitats	Element: [Descompte] Expressió: 5
...	DetenerMacro	Terminar	
([Discapacitat]="S") Y ([NumSoci] Es Nulo)	EstablecerValor	Posa descompte al 10% per a discapacitats no socis	Element: [Descompte] Expressió: 10
...	DetenerMacro	Terminar	
([Discapacitat]="S") Y ([NumSoci] Es Negado Nulo)	EstablecerValor	Posa descompte al 25% per a socis discapacitats	Element: [Descompte] Expressió: 25
...	DetenerMacro	Terminar	

Condicció	Acció	Descripció	Propietats addicionals
(([Discapacitat]<>"S") Y ([NumSoci] Es Negado Nulo)	EstablecerValor	Posa descompte al 20% per a socis no discapacitats	Element: [Descompte] Expressió: 20
...	DetenerMacro	Terminar	
	EstablecerValor	Posa descompte al 0% si no és cap dels anteriors.	Element: [Descompte] Expressió: 0

El contingut de la macro **Comprovar validesa** serà el següent:

Condicció	Acció	Descripció	Propietats addicionals
(([Edat]<25) Y ([NumSoci] Es Negado Nulo)	CuadroMsj	Mostrar un missatge d'error.	Missatge: No pot ser soci Bip: Sí Tipus: Crític Títol: Error
...	CancelarEvento	No acceptar el canvi.	

## Organització de macros

És habitual que les bases de dades continguin un gran nombre de macros. En tal cas resulta incòmode treballar en la finestra de la base de dades amb tants noms de macros. Per això, Access permet organitzar les macros en grups de macros, encara que només permet un nivell d'agrupament.

L'estratègia és definir diferents parts dintre d'una mateixa macro. Cada part queda etiquetada amb un nom únic dintre de la macro, de forma que l'accés a cadascuna de les parts (que són macros independents) es fa mitjançant *[nom de macro].[nom de part]*.

Per fer tot això és necessari utilitzar la columna *Nom de macro* de la taula d'accions de l'editor de macros, la qual es mostra i oculta prement el botó  o l'opció corresponent del menú *Veure*.

En la columna *Nom de macro* deu especificar-se el nom del grup en la primera acció del grup, i per a la resta d'accions deu deixar-se en blanc. Així, les accions d'un grup seran aquelles que comencen en l'acció que conté el nom del grup, i totes les següents no identificades fins la pròxima acció identificada (que pertanyerà al següent grup).

En el següent exemple, hi ha tres grups dintre de la mateixa macro, amb 2, 3 i 5 accions:

Nom de macro	Acció	Comentari
SortirAccess	CuadroMsj	Missatge de comiat
	Salir	Sortir d'Access
Començar	CuadroMsj	Missatge de benvinguda
	Minimizar	Minimitzar la finestra de la base de dades
	AbrirFormulario	Obrir formulari principal
Informe	CuadroMsj	Missatge que s'imprimirà un informe
	EstablecerAdvertencias	No mostrar més missatges
	EjecutarConsulta	Fer els càlculs, primera part
	EjecutarConsulta	Fer els càlculs, segona part
	AbrirInforme	Imprimir l'informe

## Macros autoexecutables

És possible definir en tota base de dades una macro que s'executi de forma automàtica cada vegada que s'obre la base de dades. Aquesta macro deu tenir el nom "Autoexec". Habitualment s'utilitza per a obrir un formulari no vinculat a cap taula ni consulta, que conté una sèrie d'opcions

(botons de comando, menús, etc.) per obrir la resta de formularis de la base de dades, donant al sistema un aspecte de programa d'aplicació més que de base de dades.

## **Exercici 17: macros**

Crear una macro que automàticament (a l'obrir la base de dades) obri un formulari que presenti les següents opcions (mitjançant botons de comando):

- Obrir el formulari de botigues i les seves existències.
- Mostrar el llistat d'existències en els magatzems.
- Imprimir el llistat d'existències en els magatzems.
- Obrir el formulari Ordre
- Sortir d'Access.

El formulari Ordre deu crear-se primer. Es tracta d'un formulari que no està associat a cap taula. Conté tres controls de tipus *Quadre de text* anomenats C1, C2 i C3, que se suposa que contindran números. Deuen funcionar així:

- Sempre els valors introduïts deuen ser  $C1 < C2 < C3$ . Si s'intenta infringir aquesta regla, es deu avortar la modificació (utilitzar l'esdeveniment *Abans de modificar*).
- Si es modifica el valor de C1 o C3, C2 deu calcular-se automàticament C2 com  $(C1+C3)/2$ .
- Si C1 queda buit deu omplir-se a 0. Si C2 queda buit deu omplir-se a C1+1. Si C3 queda buit deu omplir-se a C2+1.



## Altres

### ***Com protegir la base de dades mitjançant comptes d'usuari i grup***

1. Crear un nou grup de treball. Per a això s'ha d'executar el programa c:\windows\system\wrkgadm.exe
2. Prémer sobre el botó crear. Introduir un nom, una organització i un identificador per al grup de treball. Acceptar.
3. **Important!** Al guardar l'arxiu d'informació del grup de treball NO fer-ho amb el nom que suggereix (c:\windows\system\system.mdw) perquè ja no hauria tornada endarrere, hauríem reemplaçat l'arxiu per defecte amb el nostre. És preferible fer-ho amb c:\windows\system\elnomquetuvulguis.mdw. Acceptar (3 vegades) i Sortir.
4. Entrar en Access, crear els usuaris i grups que necessitem prement en Eines / Seguretat / Comptes d'usuari i grup. Un dels usuaris haurà de ser l'administrador. Hem d'establir obligatòriament una contrasenya d'inici de sessió per a l'administrador, si no el fem no ens apareixerà la finestra de Connexió en la qual posar el nom d'usuari i contrasenya. A l'usuari Administrador (no confondre amb el nou administrador que hem creat) hem d'eliminar-ho del grup d'administradors.
5. Una vegada creats els usuaris, a través del menú Eines / Seguretat / Permisos d'usuari i de grup establim els permisos de cada usuari per a cada objecte, sigui taula (existent o nova), consultes, etc.
6. Ara ja estem preparats per protegir la base de dades a través del menú Eines / Seguretat / Assistent per a seguretat per usuaris. Elegim els objectes que volem protegir i desm la base de dades protegida amb un nom diferent al que tenia fins el moment.
7. Cada vegada que un usuari entri en la base de dades haurà d'introduir el seu nom d'usuari i contrasenya. Tindrà els permisos que l'administrador li hagi assignat.

**Atenció:** Si desaparegués el nostre arxiu mdw no podrem entrar en cap base de dades. Haurem d'unir-nos al grup estàndard (system.mdw) mitjançant c:\windows\system\wrkgadm.exe com he comentat al principi. Sempre podríem crear de nou el nostre arxiu mdw amb els usuaris i permisos procedents.