

A Microsoft Access

Access, prodotto dalla Microsoft, è il più diffuso sistema di gestione di basi di dati per l'ambiente Microsoft Windows. Access può essere utilizzato in due modalità: come gestore di basi di dati autonomo su personal computer e come interfaccia verso altri sistemi.

Come gestore di basi di dati autonomo risente dei limiti dell'architettura dei personal computer: offre un supporto transazionale limitato, con meccanismi di sicurezza, protezione dei dati e gestione della concorrenza piuttosto semplici e incompleti. D'altra parte ha un costo assai ridotto, che giustifica queste limitazioni, e le applicazioni cui è destinato non hanno tipicamente bisogno di fare un uso sofisticato di questi servizi. L'interfaccia del sistema sfrutta le potenzialità dell'ambiente grafico e offre un ambiente facile da usare, sia per l'utente applicativo sia per il progettista della base di dati.

Quando è usato come client di database server relazionali, Access mette a disposizione le proprie funzionalità di dialogo per l'interazione con questi sistemi. Access, in questo contesto, può essere visto come uno strumento che permette di evitare di scrivere codice SQL, in quanto acquisisce schemi e semplici interrogazioni tramite una rappresentazione grafica facilmente comprensibile; questi input vengono tradotti in opportuni comandi SQL in modo trasparente. Il protocollo ODBC, descritto nel Capitolo 6, viene usato per la comunicazione tra Access e il database server.

La descrizione di Access si concentrerà sulle funzioni di gestore di basi di dati, ponendo particolarmente l'accento sulle funzionalità di definizione di schemi e interrogazioni. La presentazione si baserà sulla versione italiana di Access 2007, disponibile come un prodotto a sé stante o come un componente di Microsoft Office. Per una descrizione completa del sistema, invitiamo a consultare i manuali forniti con il programma e la guida in linea raggiungibile dal menu di aiuto (l'ultimo a destra, caratterizzato dall'etichetta '?'). Sono anche facilmente reperibili un gran numero di testi dedicati alla descrizione di questo sistema (per esempio, [56]).

A.1 Caratteristiche del sistema

Il sistema viene attivato nel modo tradizionale in cui partono le applicazioni Windows, ovvero selezionando l'icona del programma in una finestra o in un menu.

All'avvio, il programma chiede se si vuole creare un nuovo database o aprire un database esistente, fornendo una lista dei database che sono stati utilizzati precedentemente dal programma. La creazione di un database può partire da un database vuoto o seguire una serie di modelli predefiniti (utili per utenti inesperti). A ogni database corrisponde un file; per aprire un database preesistente bisogna selezionare il corrispondente file. Per creare un nuovo database o aprirne uno

preesistente si possono anche utilizzare i comandi **Nuovo database** e **Apri database** del menu base dell'applicazione. La finestra principale è coerente con la struttura delle applicazioni Office 2007, presentando una serie di ambiti di lavoro, ciascuno caratterizzato da un opportuno insieme di comandi accessibili tramite icone poste sotto la classica barra dei menu. Nel caso di Access 2007, gli ambiti sono "Home", "Crea", "Dati esterni", "Strumenti database" e "Componenti aggiuntivi". La Figura A.1 mostra l'aspetto dell'applicazione quando è selezionato l'ambito "Home" sulla base di dati di esempio.

La finestra base dell'applicazione presenta nella configurazione standard sul lato sinistro tutti i nomi delle tabelle presenti nel database (se vi sono più tabelle di quante la finestra ne possa contenere, viene automaticamente aggiunta una barra di scorrimento con la quale è possibile navigare lungo la lista). In Figura A.1 si vede l'insieme di elementi della famiglia *Table* presenti nella base di dati di riferimento. Osserviamo che la base di dati contiene una tabella CITTÀ e una tabella PERSONA.

Descriveremo ora la definizione di tabelle e interrogazioni, mentre tratteremo sommariamente le funzionalità offerte per la gestione di maschere, report, macro e moduli.

A.2 La definizione delle tabelle

Per definire lo schema di una nuova tabella bisogna selezionare il contesto "Crea" dalla toolbar, che presenta tra le prime opzioni nel menu a icone sottostante proprio quelle relative alla creazione di nuove tabelle. Access propone la scelta tra

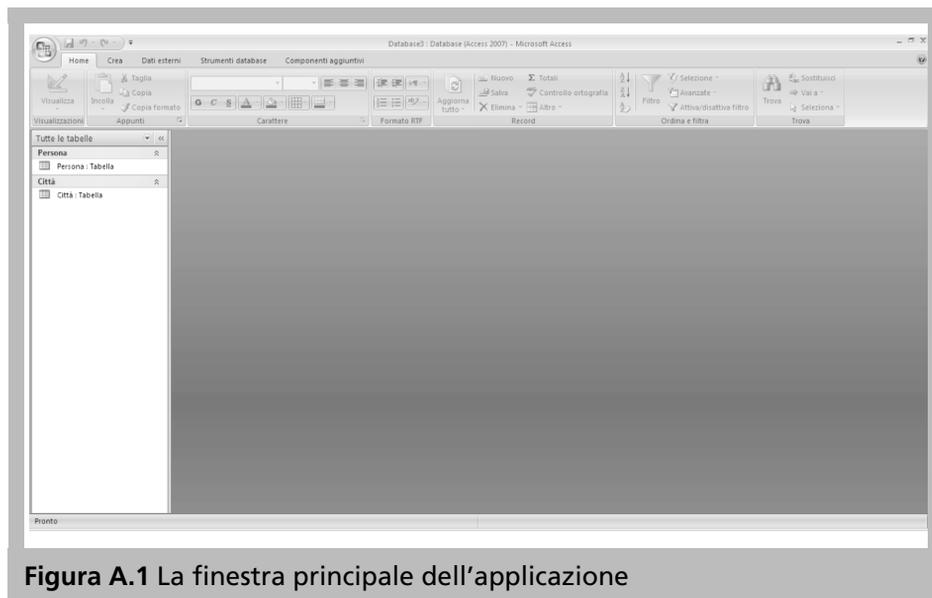


Figura A.1 La finestra principale dell'applicazione

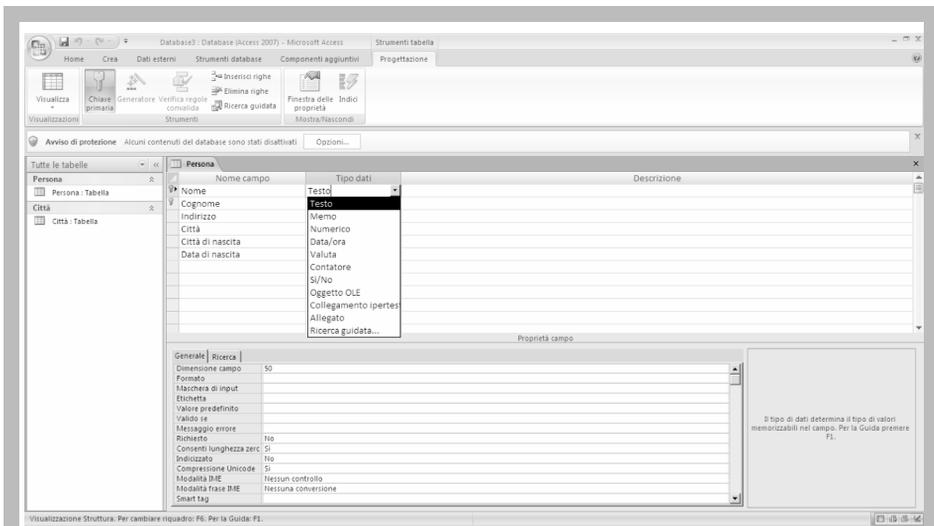


Figura A.2 Finestra di definizione dello schema delle tabelle

diverse modalità di definizione della tabella, tra cui **Tabella**, **Modelli di tabella** e **Struttura tabella**. Con l'opzione **Tabella** si definisce lo schema della tabella come se si definisse la struttura di un *foglio elettronico (spreadsheet)*, ossia gli oggetti gestiti da applicazioni come Microsoft Excel o Lotus 123. Questa interfaccia è dedicata a chi ha esperienza nell'uso di fogli elettronici e può rendere più facile per questa categoria d'utenti l'uso di un sistema relazionale. L'opzione **Modelli di tabella** consente l'uso di un "wizard", ovvero uno strumento di supporto che guida nella creazione di tabelle fornendo una collezione di esempi cui ispirarsi. Access propone l'uso dei wizard anche negli altri contesti di definizione dei vari altri componenti della base di dati. L'uso di wizard è consigliabile per la creazione di componenti dalla struttura regolare oppure nel caso di utenti poco esperti; per la gestione di situazioni particolari o nel caso di utenti sofisticati conviene invece fare uso delle funzionalità complete di definizione disponibili con la opzione **Struttura tabella**, su cui concentriamo l'analisi.

Per prima cosa si definiscono gli attributi, utilizzando la finestra che compare in Figura A.2. Per ciascun attributo bisogna specificare il nome e il dominio. Vi è poi un insieme di ulteriori informazioni definibili, che variano a seconda del dominio associato all'attributo (nel programma gli attributi vengono chiamati *campi*, i domini *tipi dati*). I domini che possono essere associati a un attributo ricordano i domini dello standard SQL, con qualche arricchimento. I domini definibili sono i seguenti.

- **Testo**: permette di rappresentare stringhe di caratteri (corrisponde al dominio `varchar` di SQL).
- **Memo**: permette di rappresentare stringhe di testo lunghe fino a 64.000 caratteri;

non è un dominio esplicitamente previsto da SQL, ma può essere inteso come un caso particolare di `varchar`.

- **Numerico**: rappresenta la famiglia di domini numerici, interi e reali. Le proprietà del dominio permettono di specificare se l'attributo rappresenta valori precisi o approssimati, e il grado di precisione della rappresentazione. Corrisponde alle famiglie di domini `numeric`, `decimal`, `integer`, `smallint`, `float`, `double` e `real`.
- **Data/Ora**: rappresenta istanti temporali e corrisponde ai domini SQL `date`, `time` e `timestamp`, a seconda che si decida di rappresentare rispettivamente solo la data, solo l'ora o entrambe.
- **Valuta**: serve per rappresentare valori monetari. È essenzialmente un caso particolare del dominio **Numerico**, caratterizzato da una rappresentazione numerica esatta su 8 byte con due cifre decimali. È rappresentato con un dominio apposito a causa della particolare rilevanza applicativa che ha la rappresentazione di valori monetari.
- **Contatore**: è un particolare dominio che associa a ogni riga della tabella un valore unico. Questo dominio permette di associare una chiave compatta alla tabella.
- **Sì/No**: corrisponde al dominio `boolean` di SQL-3.
- **Oggetto OLE**: rappresenta un generico oggetto che può essere gestito tramite OLE (*Object Linking and Embedding*). OLE è un protocollo che permette di specificare quale applicazione deve gestire un oggetto nell'ambiente Windows. In questo modo si possono inserire all'interno di una base di dati informazioni dei tipi più vari, come documenti di un word processor, fogli elettronici, immagini o informazioni multimediali, lasciando all'applicazione invocata tramite OLE il compito di gestirne adeguatamente il contenuto.
- **Collegamento ipertestuale**: permette la definizione di un riferimento. Il riferimento può essere interno alla base di dati o anche esterno (per esempio l'URL di una risorsa disponibile in Internet).
- **Allegato**: introdotto in Access a partire dalla versione 2007, permette di includere oggetti con la modalità *attachment*, incapsulando quindi interi oggetti dell'ambiente Windows in modo simile a quanto offerto dal dominio **Oggetto OLE**.
- **Ricerca guidata...**: questa opzione permette la definizione di un meccanismo che in fase di inserimento consente di scegliere un valore tra quelli presenti in un elenco predefinito o estratti tramite una query dal database.

Per ciascun attributo si possono poi specificare un certo numero di proprietà, che qualificano ulteriormente il suo dominio e la sua rappresentazione interna. Le proprietà compaiono nella parte inferiore della finestra di definizione dello schema.

- **Dimensione campo**: rappresenta la dimensione dell'attributo. Si può specificare solo per i domini **Testo** e **Numerico**. Per il tipo **Testo** la dimensione è un valore che rappresenta la lunghezza massima della stringa di caratteri. Per il tipo **Numerico**, si possono specificare le seguenti dimensioni:

- Byte: interi su 8 bit (valori tra 0 e 255);
 - Intero: interi su 16 bit (valori tra -32 768 e 32 767);
 - Intero lungo: interi su 32 bit;
 - Precisione singola: rappresentazione in virgola mobile su 32 bit;
 - Precisione doppia: rappresentazione in virgola mobile su 64 bit;
 - ID replica: identificatore su 128 bit, unico per ogni tupla, anche in un sistema distribuito;
 - Decimale: rappresentazione in virgola mobile, con la possibilità di specificare il numero di cifre decimali significative.
- *Formato*: descrive il formato di visualizzazione dei valori degli attributi. Access utilizza dove possibile i valori specificati nell'ambiente Windows (opzioni di internazionalizzazione del pannello di controllo). Per la rappresentazione di date, numeri e valori booleani, permette di scegliere tra vari formati predefiniti (sette formati predefiniti per le date, sette per i numeri, tre per i booleani). Si possono poi definire altri formati. È possibile inoltre distinguere la rappresentazione dei valori a seconda che siano positivi, negativi, pari a zero, o abbiano associato il valore nullo.
 - *Precisione*: questa informazione, definibile solo per attributi di dominio numerico Decimale specifica quante cifre decimali devono essere utilizzate nella rappresentazione.
 - *Scala*: questa informazione, definibile solo per attributi di dominio numerico Decimale specifica quante cifre decimali devono essere utilizzate al massimo a destra del separatore decimale.
 - *Posizioni decimali*: questa informazione, definibile solo per attributi di dominio numerico Precisione singola, Precisione doppia e Decimale, specifica il numero di cifre da utilizzare a destra del separatore decimale.
 - *Maschera di input*: questo parametro specifica il formato che deve essere utilizzato per l'immissione dei dati. Se per esempio un attributo registra un numero di telefono, composto da un prefisso di 3 cifre e da un numero di 7 cifre separati da un trattino, è possibile specificare una maschera di ingresso che individui le due parti e introduca immediatamente il trattino, permettendo all'utente di inserire solamente le cifre. Access mette a disposizione un wizard anche per la creazione della *Maschera di input*.
 - *Etichetta*: rappresenta il nome che può essere dato all'attributo quando compare in una maschera o in un report. Può capitare di utilizzare come nome dell'attributo un nome compatto con cui sia possibile scrivere interrogazioni concise, mentre per la visualizzazione dei risultati si preferisca usare un nome che rappresenti meglio il significato dell'attributo.
 - *Valore predefinito*: con questo parametro si specifica il valore di default per l'attributo. Corrisponde esattamente all'opzione default di SQL. Tutte le volte che si inserisce una nuova tupla, il valore di default comparirà come valore per l'attributo. Si può utilizzare come valore di default anche il risultato di un'espressione, come per esempio =Date() che assegna a un campo di dominio Data/Ora il giorno corrente.
 - *Valido se*: permette di specificare un vincolo sull'attributo. Access verifica auto-

maticamente che ogni valore inserito appartenga al dominio dell'attributo. Oltre a questo controllo, Access permette di specificare un vincolo generico per ogni attributo (in modo analogo alla clausola `check` di SQL). Questo vincolo viene espresso utilizzando la sintassi che è utilizzata per la specifica delle condizioni in QBE, che vedremo nei prossimi paragrafi.

- *Messaggio errore*: specifica il messaggio che deve essere visualizzato quando viene inserito un valore che non rispetta il vincolo d'integrità.
- *Richiesto*: specifica se deve essere sempre presente un valore per l'attributo. La proprietà può essere vera o falsa e corrisponde al vincolo `not null` di SQL.
- *Consenti lunghezza zero*: è una proprietà che vale solo per gli attributi di tipo Testo e Memo. Specifica se devono essere ammesse stringhe di lunghezza nulla, o se una stringa di lunghezza nulla deve essere considerata come un valore nullo. A seconda dei contesti applicativi, può essere utile gestire in modo diverso le stringhe vuote dal valore nullo. Si tenga presente che il valore nullo viene trattato in modo particolare da SQL, per cui un confronto di disuguaglianza su stringhe è soddisfatto da una stringa di lunghezza zero, ma non da una stringa nulla.
- *Indicizzato*: specifica se deve essere costruito un indice sull'attributo. Le opzioni possibili sono No, Sì (Duplicati ammessi) e Sì (Duplicati non ammessi). La terza opzione definisce sull'attributo un indice di tipo *unique*. Questo è anche il modo in cui si rappresentano i vincoli di tipo *unique*. Non si possono definire indici su attributi Oggetto OLE e Allegato. Con questa modalità si permette solo la definizione di indici su un solo attributo; per la definizione di indici più complicati bisogna operare al livello di tabella.
- *Compressione Unicode*: specifica per i campi di tipo testuale se la rappresentazione Unicode a 16 bit dei caratteri, che costituisce lo standard di Microsoft Access a partire dalla versione 2000, deve essere compressa nei casi in cui si rappresentano caratteri del normale alfabeto latino. La compressione avviene omettendo la rappresentazione del primo byte, il quale è pari a zero per tutti i caratteri del dominio ASCII.
- *Modalità IME*: specifica se sul campo deve essere utilizzato un *Input Method Extension*, ovvero una modalità particolare di acquisizione dell'input che consente di gestire gli alfabeti ideogrammatici che caratterizzano le lingue orientali. Non è un'opzione significativa se non si prevedono di utilizzare alfabeti di questo tipo.
- *Modalità frase IME*: specifica ulteriormente la modalità di input da adottare per l'acquisizione di frasi nei linguaggi ideogrammatici.
- *Smart tag*: specifica se all'attributo deve essere associata un'etichetta intelligente, che permette di invocare funzioni predefinite nella fase di input e consultazione del campo (per esempio, per attributi che rappresentano date è possibile accedere a un calendario o invocare l'applicazione di gestione dell'agenda).

Dopo aver definito i vari attributi, si completa la sessione di definizione indicando quali attributi devono essere considerati la chiave primaria della tabella. L'indicazione degli attributi di chiave avviene selezionando gli attributi e premendo il pulsante che raffigura la chiave nella barra degli strumenti associata all'opzione

“Progettazione”. Gli attributi che costituiscono la chiave vengono visualizzati con un’icona che rappresenta una chiave nella colonna che precede il nome. Automaticamente Access definirà un indice di tipo `unique` sugli attributi che costituiscono la chiave.

Si possono poi definire ulteriori proprietà a livello di tabella (cui si accede tramite l’icona **Finestra delle proprietà** associata all’opzione “Progettazione”. Le più significative sono le seguenti.

- *Descrizione*: è una descrizione testuale del contenuto della tabella.
- *Valido se*: specifica un vincolo che deve essere soddisfatto da ogni tupla della tabella. A livello di tabella si definiscono vincoli che coinvolgono diversi attributi. Pure in questo caso, la sintassi è quella usata per l’espressione delle condizioni sulle interrogazioni. Il controllo viene effettuato al termine dell’inserimento di ogni tupla.
- *Messaggio errore*: rappresenta il messaggio che viene visualizzato quando il sistema rileva una violazione del vincolo.
- *Filtro*: specifica le condizioni che devono essere soddisfatte dagli elementi che si vogliono visualizzare.
- *Ordina per*: descrive gli attributi rispetto ai quali si devono ordinare le tuple della tabella.

Per specificare indici su più attributi si deve aprire la finestra di definizione degli indici, premendo il pulsante **INDICI** sulla barra degli strumenti nell’opzione “Progettazione”. La finestra contiene una tabella (mostrata in Figura A.3) con colonne **Nome indice**, **Nome campo** e **Criterio ordinamento**. Per definire un indice su più attributi, si inserisce in una riga il nome dell’indice, il nome del primo attributo e la direzione di ordinamento. Nella riga successiva si lascia vuoto il nome dell’indice e si introduce il nome del secondo attributo e la corrispondente direzione di ordinamento, proseguendo allo stesso modo per tutti gli attributi che caratterizzano l’indice.

Prima di terminare la sessione di definizione bisogna salvare il risultato, chiudendo per esempio la finestra di definizione della struttura. Access a questo punto chiede quale nome deve essere associato alla tabella. Il nome può anche contenere al suo interno degli spazi.

A.2.1 Specifica dei cammini di join

Un cammino di join è un legame tra coppie di attributi di due tabelle e si usa per specificare che tra le due tabelle viene normalmente eseguito un join basato sull’uguaglianza di quegli attributi. Un cammino di join viene rappresentato graficamente da una linea che collega le due tabelle. Per esempio, nella base di dati “persone e città”, esiste un cammino di join tra l’attributo **Città di nascita** di **PERSONA** e **Nome** di **CITTÀ**. Access permette la definizione di cammini di join (chiamati *relazioni*); per ogni cammino di join è inoltre possibile specificare se vi è associato un vincolo di integrità referenziale. Tutto questo avviene senza che

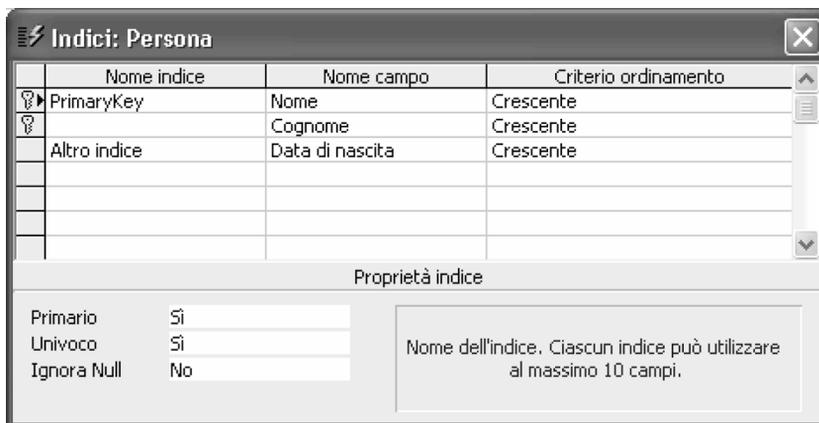


Figura A.3 La finestra di descrizione degli indici

venga immesso del testo, operando esclusivamente in modo grafico. Descriviamo brevemente la procedura.

Si inizia selezionando l'opzione **Relazioni** dalla toolbar associata all'opzione "Strumenti database". Si apre a questo punto una finestra (Figura A.4) in cui si possono inserire gli schemi delle tabelle create, selezionandole da una lista. È possibile definire i cammini di join tra gli schemi selezionando un attributo di uno schema e, tenendo premuto il pulsante del mouse, spostando il puntatore del mouse sull'attributo corrispondente dell'altra tabella. Una volta definito il legame tra gli attributi, Access apre una finestra che rappresenta gli attributi coinvolti nel join e che permette di estendere la condizione di join ad altri attributi o di modificare

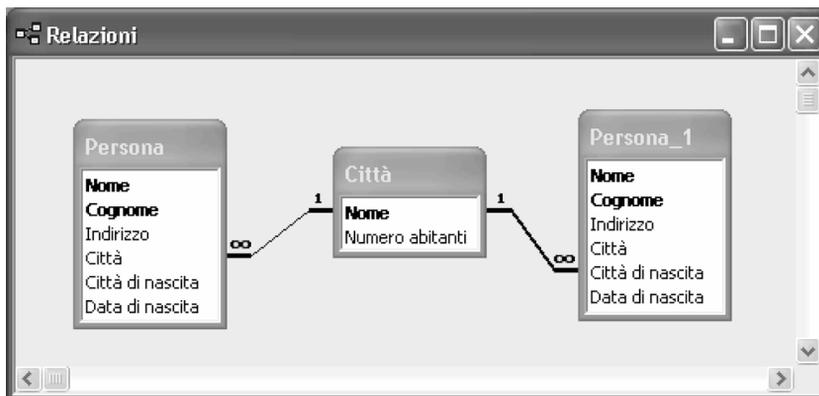


Figura A.4 La finestra di definizione dei cammini di join

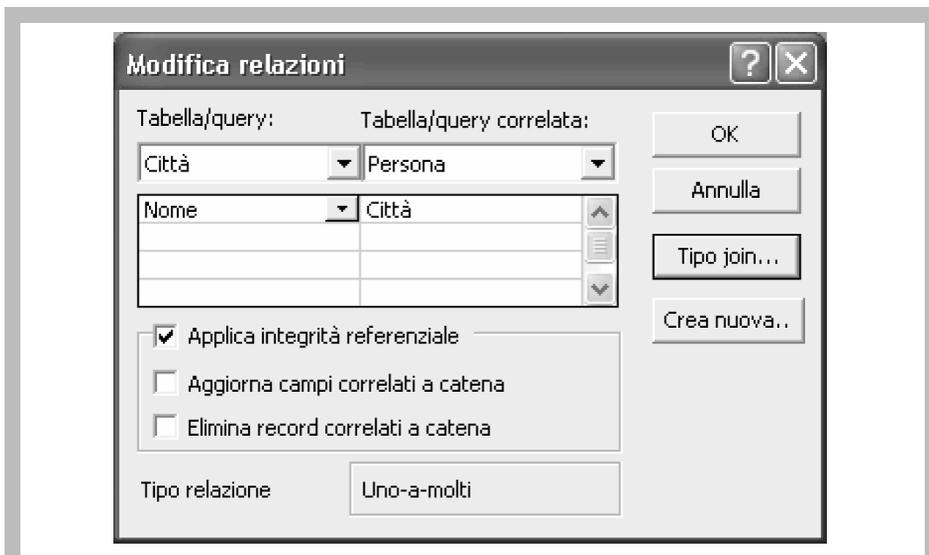


Figura A.5 La finestra di definizione delle proprietà dei join

la relazione così definita. Questa finestra è rappresentata in Figura A.5. Premendo il pulsante TIPO JOIN di questa finestra, si ha la possibilità di scegliere quale tipo di join, *inner*, *outer left* o *outer right*, deve essere utilizzato nel legame tra le due tabelle (a differenza di SQL, il *full outer join* non è offerto). In questo modo tutte le volte che si definisce una query che accede alle due tabelle, verranno assegnate di default le condizioni di join specificate.

Definito il cammino di join, è possibile specificare se a esso deve essere associato un vincolo di integrità referenziale. Nella stessa finestra, Access permette di definire una politica di reazione alle violazioni, per cui si può imporre che le modifiche o le cancellazioni siano seguite da corrispondenti modifiche e cancellazioni nelle altre tabelle. Se la politica non è specificata, ogni modifica che introduca una violazione viene semplicemente impedita.

Rispetto allo standard SQL-2, Access permette quindi un insieme di reazioni più limitato, corrispondente alle sole scelte di *cascade delete* e *cascade update*. Per specificare i vari vincoli di integrità referenziale, si devono seguire i criteri di progettazione descritti nel Capitolo 9. Access non permette che nel grafo compaia più di un cammino tra due tabelle; se si devono rappresentare più cammini di join tra due tabelle, bisogna introdurre più esemplari della stessa tabella del grafo (Figura A.4).

A.2.2 Popolamento delle tabelle

Dopo aver definito lo schema delle tabelle, si possono inserire tuple nella tabelle per popolare l'istanza della base di dati. Anche per questo compito Access fornisce un'interfaccia grafica facile da usare. Aprendo una tabella dalla finestra di

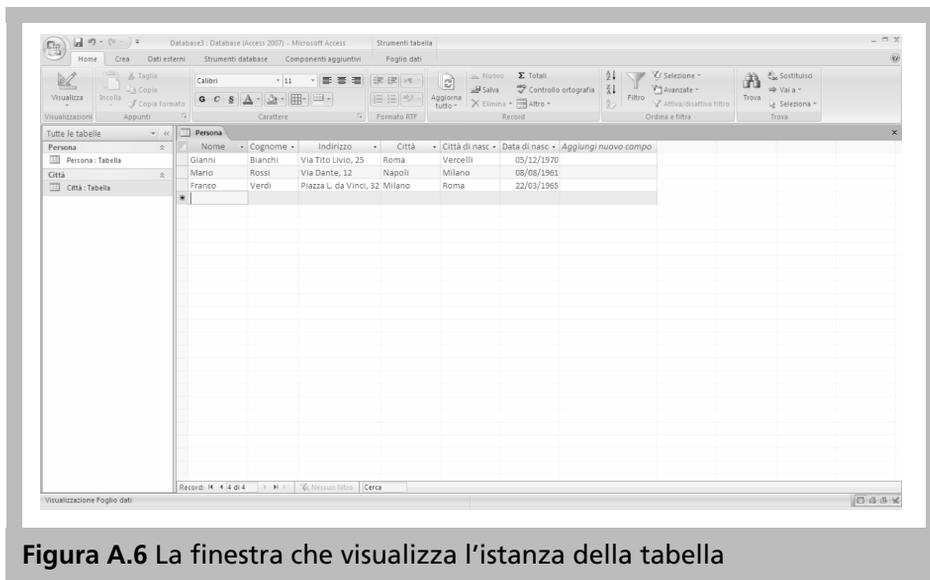


Figura A.6 La finestra che visualizza l'istanza della tabella

partenza selezionandola con doppio clic dall'elenco, compare una rappresentazione tabellare del contenuto della tabella. Questa consiste in una griglia con colonne con intestazione pari ai nomi degli attributi, e righe che descrivono le tuple della tabella, cui si somma una riga vuota che serve per inserire nuove tuple nella tabella. Access aggiunge anche una colonna aggiuntiva, che consente di modificare lo schema aggiungendo nuovi attributi. La Figura A.6 rappresenta questa finestra.

L'inserimento avviene ponendo il cursore nell'ultima riga e digitando un valore per ogni attributo. Se il valore inserito non rispetta tutti i vincoli definiti sull'attributo, l'inserimento viene immediatamente rifiutato. Spostando il cursore al di fuori della riga, si indica implicitamente che l'inserimento è terminato. A questo punto il sistema controlla che tutti i vincoli siano rispettati, ovvero che i campi per cui è necessario fornire un valore siano stati specificati e che i valori degli attributi rispettino le regole di validazione definite. Per effettuare delle modifiche di attributi basta andare a selezionare il valore da modificare con il cursore e inserire il nuovo valore. Terminata l'immissione del valore e spostato il cursore in una diversa posizione, i vincoli sono verificati e la modifica è completata.

A.3 La definizione di query

Per la definizione di interrogazioni, Access mette a disposizione due diversi strumenti: uno strumento grafico di formulazione di interrogazioni di tipo QBE (*Query By Example*) e un interprete SQL. Descriviamo dapprima le caratteristiche dell'interfaccia QBE, analizzando successivamente l'interprete SQL.

A.3.1 Query By Example

Il nome QBE fa riferimento a una famiglia molto vasta di linguaggi di interrogazione per basi di dati, che realizzano un'implementazione delle idee di base del calcolo relazionale dei domini (Paragrafo 3.2.1). Il punto fondamentale è che un'interrogazione venga formulata descrivendo le caratteristiche che devono essere possedute dalle righe del risultato. La definizione di una query avviene riempiendo uno schema di tabella con tutti gli attributi e le condizioni che caratterizzano una riga “esemplare” del risultato.

Per definire una nuova interrogazione si seleziona l'opzione “Crea” della finestra di base dell'applicazione e si preme il pulsante *Struttura query*, che apre la finestra di progettazione di query. Questa finestra è divisa in due metà (Figura A.7). La metà superiore è inizialmente vuota e viene riempita con una descrizione degli schemi delle tabelle selezionate da una lista. Le tabelle sono collegate dai cammini di join predefiniti (possiamo in effetti interpretare la metà superiore della finestra come la porzione del diagramma **Relazioni** rilevante per l'interrogazione che si deve definire). Nella metà inferiore della finestra compare una tabella inizialmente vuota, con un insieme di colonne senza nome e con righe etichettate **Campo**, **Ordinamento**, **Mostra** e **Criteri**.

Le celle della riga **Ordinamento** possono essere vuote o contenere una tra le opzioni **Crescente** o **Decrescente**. Quando la riga non è vuota viene imposto un ordinamento delle tuple del risultato, secondo i valori dell'attributo associato alla colonna in cui compare l'opzione. Se vi sono più colonne con la cella **Ordinamento** attivata, si applica per primo l'ordinamento che compare più a sinistra nelle colonne; a pari valori dell'attributo, si applicheranno man mano le successive condizioni di ordinamento, procedendo da sinistra a destra.

Le celle della riga **Mostra** contengono un quadratino che può o meno contenere un segno di spunta. Se il quadratino contiene tale segno, l'attributo che compare nella colonna dovrà far parte del risultato della interrogazione. Lo stato del quadratino commuta con un clic del mouse.

Le celle della riga **Criteri** contengono le condizioni che devono essere soddisfatte dalle tuple risultato della interrogazione. Le condizioni possono essere semplici confronti tra il valore dell'attributo che compare in cima alla colonna e una costante; in questo caso basta inserire il valore della costante nella cella. La condizione può anche essere più complicata e includere confronti più ricchi, espressioni e riferimenti ad altri attributi, come vedremo negli esempi successivi.

Per mettere i nomi degli attributi in cima alle colonne si possono utilizzare due modalità: si può scrivere direttamente il nome dell'attributo, eventualmente qualificato col nome della tabella di appartenenza, o si possono selezionare gli attributi che compaiono nella rappresentazione degli schemi della metà superiore della finestra, “trascinandoli” nelle relative colonne o facendo un doppio clic su di essi (a quel punto l'attributo viene riportato nella prima colonna libera). Dopo che la query è stata formulata, per eseguirla bisogna premere il pulsante **Esegui** della barra degli strumenti, associato a un punto esclamativo; dopo l'esecuzione, la tabella risultato della query compare al posto della finestra di definizione della query.

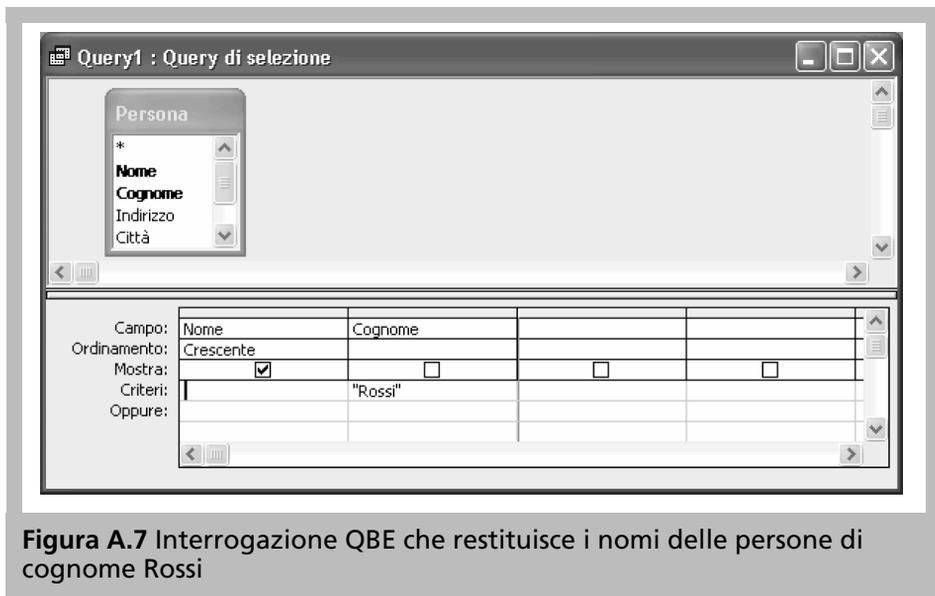


Figura A.7 Interrogazione QBE che restituisce i nomi delle persone di cognome Rossi

Vediamo alcuni esempi di definizione di interrogazioni. Supponiamo di avere una base di dati con una tabella PERSONA(Nome, Cognome, Indirizzo, Città, Città di nascita, Data di nascita) e una tabella CITTÀ(Nome, Numero abitanti). Per trovare i nomi, ordinati alfabeticamente, delle persone aventi cognome Rossi possiamo riempire lo schema nel modo descritto in Figura A.7.

Quando sono riempiti più campi della riga Criteri, l'interrogazione considera tutte le condizioni in congiunzione. Se bisogna selezionare le tuple che soddisfano più condizioni in disgiunzione, bisogna riempire più righe con i diversi criteri. Access etichetta automaticamente con Oppure le righe aggiuntive. Così, per trovare i nomi, cognomi e indirizzi delle persone di Milano aventi cognome Rossi o Bianchi, si creerà uno schema come quello in Figura A.8.

Nella lista degli attributi di una tabella compare anche il simbolo di asterisco che, con significato analogo a SQL, rappresenta tutti gli attributi. Per selezionare quindi tutti gli attributi delle tuple della tabella PERSONA che hanno la residenza nella città di nascita, potremo formulare la query QBE rappresentata in Figura A.9.



Figura A.8 Query che restituisce i milanesi chiamati "Rossi" o "Bianchi"

Campo:	Persona.*	Città		
Ordinamento:				
Mostra:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteri:		[Città di nascita]		
Oppure:				

Figura A.9 Query che restituisce le persone con città di nascita e di residenza uguali

In questa interrogazione, per imporre l'uguaglianza di due attributi, abbiamo imposto come valore dell'attributo **Città** il nome dell'attributo **Città di nascita** racchiuso tra parentesi quadre. Le parentesi quadre sono il costrutto sintattico che permette ad Access di distinguere le stringhe di caratteri costanti dai riferimenti ai componenti dello schema. Access consente poi di formulare delle condizioni utilizzando i normali operatori di confronto (<, ≤, >, ≥ e <>) e l'operatore **Like**, per il confronto di stringhe con espressioni regolari che usano i caratteri speciali * e ? (che corrispondono rispettivamente ai caratteri % e _ della sintassi standard SQL). Per trovare i nomi, i cognomi e le date di nascita delle persone che sono nate prima del 31 gennaio 1965 e che hanno un cognome che inizia con la lettera 'C', si potrà formulare la query mostrata in Figura A.10.

A ogni interrogazione è possibile associare delle proprietà, a diversi livelli. A livello di singola colonna, si può specificare un formato di visualizzazione diverso da quello immesso nella fase di definizione dello schema della tabella. Un'altra importante proprietà è l'eliminazione di eventuali duplicati presenti nel risultato. Per specificare che i duplicati devono essere rimossi bisogna accedere alla finestra di descrizione delle proprietà della query e assegnare alla proprietà **Valori univoci** il valore **Sì**.

Per formulare delle interrogazioni che richiedono più tabelle, è opportuno far comparire nella finestra superiore le tabelle richieste (selezionandole dalla finestra di dialogo che compare appena si crea una nuova interrogazione). Le tabelle

Campo:	Nome	Cognome	Data di nascita	
Ordinamento:				
Mostra:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteri:		Like "C*"	<#31/01/1965#	
Oppure:				

Figura A.10 Query che restituisce i nomi, i cognomi e le date di nascita delle persone con un cognome che inizia per 'C' nate prima del 31/1/65

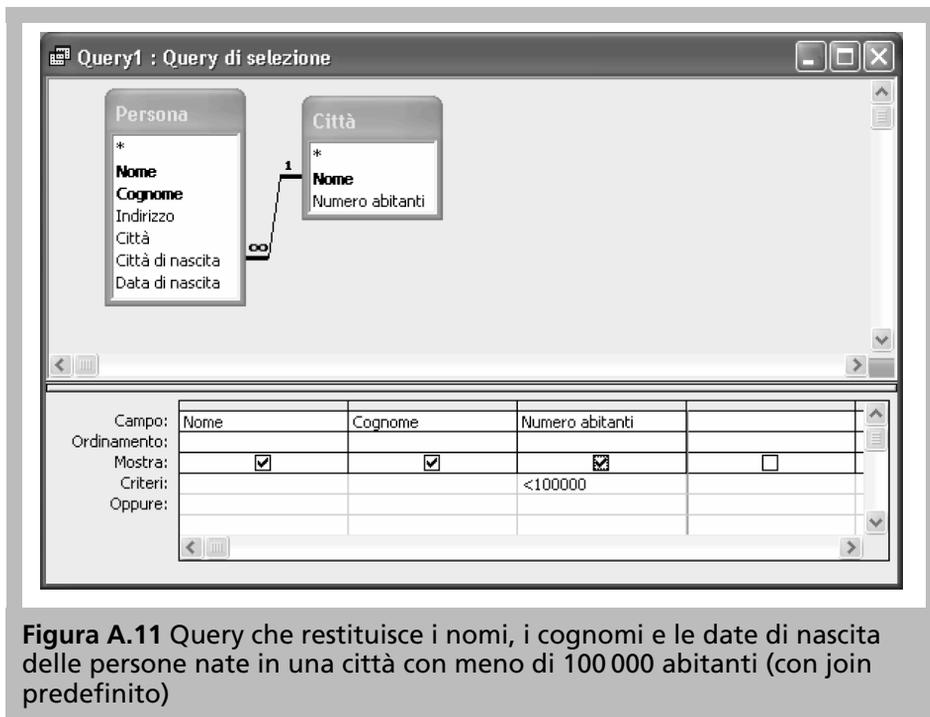


Figura A.11 Query che restituisce i nomi, i cognomi e le date di nascita delle persone nate in una città con meno di 100 000 abitanti (con join predefinito)

compariranno legate dai cammini di join che sono stati definiti al momento della definizione dello schema. Nella tabella nella metà inferiore della finestra di definizione della query è possibile aggiungere anche la riga **Tabella** (Figura A.12), selezionando l'opzione **Nomi tabelle** dalle icone nella toolbar. La riga rappresenta il nome della tabella da cui viene prelevato l'attributo. Le condizioni di join tra le tuple delle tabelle non dovranno essere specificate se queste sono state predefinite. Quando le condizioni di join predefinite non sono quelle richieste dalla particolare query, è possibile modificare i cammini che collegano le tabelle intervenendo sulla rappresentazione grafica della parte superiore della finestra.



Figura A.12 Query che restituisce i nomi e cognomi delle persone nate in una città con meno di 100.000 abitanti (senza join predefinito)

Volendo per esempio formulare la query che trova le persone nate in una città con meno di centomila abitanti, se è stato predefinito il legame di join tra le tabelle PERSONA e CITTÀ, si può formulare la query che appare in Figura A.11.

Qualora invece il legame di join non sia stato predefinito, si dovrà rendere esplicita la condizione nella tabella, formulando un'interrogazione come quella in Figura A.12, che usa i riferimenti agli attributi nelle righe **Criteri**.

Può essere necessario talvolta introdurre tra le tabelle che compaiono nella metà superiore anche tabelle di cui non vengono utilizzati attributi nella query. Un caso importante di questo tipo è quello in cui si devono estrarre informazioni tra due tabelle che non posseggono un cammino di join diretto, ma in cui il join è realizzato tramite una tabella intermedia. In questo caso, anche se l'interrogazione utilizzerà per la visualizzazione e l'applicazione di condizioni solo attributi delle due tabelle esterne, la tabella intermedia dovrà comparire nella metà superiore, con un approccio simile a quello utilizzato dall'algebra, dal calcolo relazionale (Capitolo 3) e da SQL (Capitolo 4). Infatti, senza un cammino di join che leghi le tabelle, il sistema esegue il prodotto cartesiano delle tabelle che compaiono nella metà superiore, applicando poi le condizioni che sono state definite per la query.

Vediamo ora la formulazione di interrogazioni QBE facenti uso di operatori aggregati. Gli operatori aggregati che Access mette a disposizione sono Somma, Media, Min, Max, Conteggio, che corrispondono agli operatori standard SQL sum, avg, min, max e count. A questi si aggiungono gli operatori DevSt (la deviazione standard), Var (la varianza), Primo e Ultimo (il valore dell'attributo rispettivamente per la prima e per l'ultima tupla). Per utilizzare questi operatori è necessario introdurre una nuova riga nello schema della query, la riga **Formula**. Questa riga viene introdotta selezionando l'opzione **Totali** dalla toolbar. Vediamo un semplice esempio d'uso degli operatori, definendo, come mostrato in Figura A.13, una query che permette di trovare il numero di tuple presenti nella tabella PERSONA.

Il fatto che nella riga **Formula** compaia il valore Conteggio fa sì che il risultato della interrogazione non sia l'elenco dei valori dell'attributo **Nome** della tabella PERSONA, ma appunto il numero di tuple. Nella riga **Campo** deve comparire uno qualsiasi degli attributi della tabella PERSONA; sarebbe stato meglio, per coerenza con SQL-2, che Access permettesse in questo contesto l'uso dell'asterisco (*), utilizzando invece il nome di un attributo per contare i diversi valori

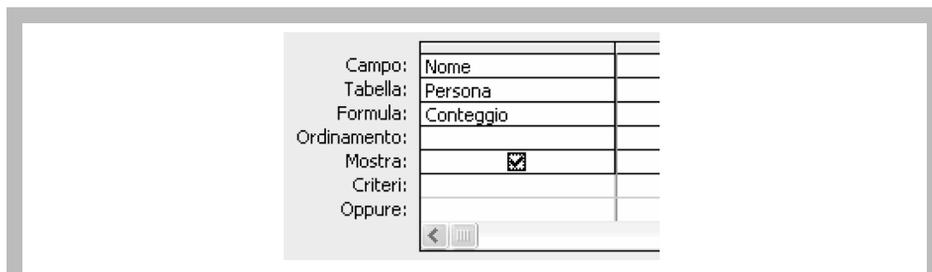


Figura A.13 Query che restituisce il numero di persone

Campo:	Cognome	Nome			
Tabella:	Persona	Persona			
Formula:	Raggruppamento	Conteggio			
Ordinamento:					
Mostra:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Criteri:					
Oppure:					

Figura A.14 Query che restituisce il numero di persone che possiedono ciascun cognome

dell'attributo; purtroppo nella valutazione degli altri operatori aggregati l'asterisco crea delle complicazioni, per cui in Access si è scelta la soluzione di impedire l'uso dell'asterisco quando è abilitata la riga **Formula**.

Nella interrogazione in Figura A.14 l'attributo **Cognome** viene caratterizzato dal valore **Raggruppamento** nella riga **Formula**, ovvero l'attributo viene utilizzato per raggruppare le tuple della tabella **PERSONA**. La seconda colonna rappresenta l'applicazione dell'operatore **Conteggio** a ogni singolo raggruppamento.

Access permette di esprimere delle condizioni sul risultato della valutazione degli operatori aggregati, in modo analogo alla clausola *having* in SQL. Per fare ciò, è sufficiente scrivere le condizioni nella riga **Criteri**, in modo analogo all'espressione di semplici condizioni sulle tuple. Così, per trovare i cognomi che sono posseduti da almeno due persone, si potrà scrivere l'interrogazione QBE in Figura A.15, che restituisce i cognomi posseduti da più di una persona, indicando per ogni cognome il numero di volte che questo appare nella tabella **PERSONA**.

Se in una interrogazione con raggruppamento le tuple devono essere selezionate preliminarmente in base ai valori di attributi che non vengono utilizzati nel raggruppamento, diventa necessario distinguere le condizioni che devono essere valutate prima del raggruppamento da quelle che devono essere eseguite nella fase successiva. Questa distinzione in SQL avviene ponendo le condizioni preliminari nella clausola *where* e le condizioni successive nella clausola *having*; nel linguaggio QBE la distinzione avviene ponendo il valore **Dove** nella riga **Formula**

Campo:	Cognome	Nome			
Tabella:	Persona	Persona			
Formula:	Raggruppamento	Conteggio			
Ordinamento:					
Mostra:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Criteri:		>1			
Oppure:					

Figura A.15 Query che restituisce il numero di persone che possiedono ciascun cognome, per i cognomi posseduti da più persone

Campo:	Cognome	Data di nascita	Cognome		
Tabella:	Persona	Persona	Persona		
Formula:	Raggruppamento	Dove	Conteggio		
Ordinamento:					
Mostra:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Criteri:		>#01/01/1975#	>1		
Oppure:					

Figura A.16 Query che restituisce gli omonimi tra le persone nate dopo l'1/1/75

per gli attributi che servono solo per la selezione delle tuple da raggruppare. La presenza del valore *Dove* è incompatibile con il flag di **Mostra**. Infatti, come in SQL, nel risultato delle interrogazioni che fanno uso di operatori aggregati possono comparire solo il risultato della valutazione degli operatori aggregati e gli attributi su cui viene effettuato il raggruppamento. Per esempio, per trovare le persone nate dopo l'1/1/75 che hanno lo stesso cognome di persone nate dopo la stessa data, si può utilizzare l'operatore *Count* e formulare l'interrogazione in Figura A.16.

Ci sono dei modi alternativi per formulare questa interrogazione che non fanno uso del termine *Dove*. Uno di questi consiste nel formulare e salvare nella base di dati un'interrogazione preliminare che estrae solo le persone nate dopo l'1/1/75; si può poi definire una seconda interrogazione, che parte dal risultato della prima andando a fare un raggruppamento in base al valore di **Cognome**, identificando così gli omonimi. A ogni interrogazione, infatti, Access associa un nome e un'interrogazione può estrarre le informazioni sia dalle tabelle del database, sia dalle interrogazioni già definite. Ogni interrogazione che viene salvata e resa persistente può quindi essere considerata come una definizione di una vista sul database.

A.3.2 L'interprete SQL

Oltre al linguaggio di interrogazione QBE, Access fornisce un interprete SQL, che può essere usato in alternativa a QBE. Access permette di passare rapidamente dal contesto QBE al contesto SQL e viceversa, scegliendo l'opzione **VISUALIZZAZIONE SQL** dalla toolbar. Il passaggio da un ambiente all'altro trasforma l'interrogazione corrente nella corrispondente interrogazione nell'altro ambiente.

Il passaggio da QBE a SQL è sempre possibile. In effetti, tutte le volte che una query QBE viene mandata in esecuzione, essa viene prima tradotta nella corrispondente forma SQL e quindi eseguita dall'interprete SQL. Il passaggio inverso non è invece sempre possibile, in quanto il linguaggio SQL è più potente di QBE, permettendo per esempio l'espressione di query con l'operatore di unione. Il linguaggio QBE è un linguaggio molto potente e facile da usare quando si devono formulare interrogazioni che fanno uso solo di selezioni, proiezioni e join: in questo caso la possibilità di formulare le interrogazioni senza bisogno di scrivere

Campo:	Cognome	Cognome	Numero abitanti	
Tabella:	Persona	Persona	Città	
Formula:	Raggruppamento	Conteggio	Dove	
Ordinamento:		Decrescente		
Mostra:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteri:		>1	>200000	
Oppure:				

Figura A.17 Query che restituisce il numero di omonimi in città sopra i 200.000 abitanti

del testo secondo una rigida sintassi rappresenta un notevole aiuto. D'altra parte QBE non mette a disposizione un meccanismo adeguato per la rappresentazione di interrogazioni complesse, come quelle che richiedono di formulare delle interrogazioni nidificate in SQL. Di fatto, quando una interrogazione SQL che fa uso di query nidificate viene tradotta in QBE, la traduzione riporta semplicemente il testo dell'intera query nidificata nell'opportuna cella della riga **Criteri**.

Per quanto riguarda la sintassi riconosciuta dall'interprete, si tratta di un'estensione della sintassi standard SQL, con un supporto per particolari funzionalità e determinate differenze sintattiche e semantiche. Alcune delle differenze sono le seguenti:

- la clausola `top` può essere usata per selezionare un certo numero di tuple dal risultato;
- vengono usate le parentesi quadre per racchiudere gli identificativi di tabelle e attributi (necessarie quando compaiono degli spazi o caratteri speciali all'interno degli identificatori);
- l'operatore `join` deve essere sempre qualificato con il termine `inner` od `outer`;
- la valutazione dell'operatore `count` è diversa: se si dà come argomento un attributo, non vengono restituiti i distinti valori dell'attributo, bensì il numero di valori non nulli (come se fosse specificata implicitamente l'opzione `all`); l'opzione `distinct` non è riconosciuta.

Per esempio, si consideri la query QBE in Figura A.17, per cui è specificata la proprietà che vengano restituiti solo i primi 10 elementi e per cui è stato predefinito un cammino di join. A essa corrisponde la seguente query nel dialetto SQL di Access:

```
select top 10 Cognome, count(Cognome) as NumeroOmonimi
from Persona inner join Città
    on Persona.[Città di nascita] = Città.Nome
where [Numero abitanti] > 200000
group by Cognome
having count(Cognome) > 1
order by count(Cognome) desc
```

A.4 Maschere e report

Le *maschere* permettono di rappresentare il contenuto del database in modo molto più chiaro e comprensibile, rispetto alla piatta rappresentazione delle righe delle tabelle.

Le maschere sono analoghe ai “moduli prestampati”, caratterizzati da un insieme di caselle in cui si devono inserire i dati, e un insieme di etichette che specificano quale dato deve essere inserito nella particolare casella. Le maschere possono servire per l’inserimento dei dati, realizzando una versione elettronica del modulo prestampato, e possono anche essere utilizzate per visualizzare e modificare il contenuto della base di dati.

Gli strumenti di generazione di maschere sono sempre stati uno degli strumenti di supporto più comuni tra quelli offerti dai DBMS commerciali. Access, al posto della semplice e tradizionale interfaccia a caratteri, sfrutta le prerogative dell’ambiente Microsoft Windows e permette la realizzazione di maschere grafiche. Lo strumento di definizione delle maschere permette di definire la posizione e il significato di ciascun componente della maschera, dando al progettista un’ampia libertà di personalizzazione a livello di fonti di caratteri, colori, disegni e simboli grafici.

Per creare una nuova maschera, bisogna selezionare il componente **Maschere** dalla toolbar associata all’opzione “Crea”. Il sistema offre sia uno strumento di progetto di base, sia uno strumento per la creazione guidata; mediante la creazione guidata si può creare in pochi istanti una semplice maschera associata a una tabella. Access offre in questo contesto la possibilità di scegliere tra vari strumenti per la creazione di maschere; essi differiscono in base allo schema di maschera che producono (per cui uno strumento produce una maschera con tutti gli attributi della tabella in sequenza uno per riga, un altro produce una rappresentazione tabellare arricchita da spazi e colori ecc.). Se non si utilizzano i servizi di uno strumento di creazione guidata, ci si trova davanti a una pagina bianca in cui si possono inserire man mano i componenti della maschera. La stessa interfaccia può essere usata per esplorare e modificare la struttura di una maschera preesistente, selezionando la maschera e premendo il pulsante **STRUTTURA**.

Una maschera è composta da diversi elementi, che devono essere definiti uno a uno. L’elemento di base di una maschera è il *controllo*, un oggetto cui corrisponde un’area rettangolare dello schermo e che può essere di tre tipi: *associato*, *non associato* e *calcolato*. Un controllo *associato* è un elemento della maschera cui viene associato un attributo di una tabella. Questo elemento rappresenterà il valore dell’attributo per la particolare tupla considerata. La rappresentazione del valore di un attributo in genere richiede di riprodurre una sequenza di caratteri; per attributi di tipo OLE, la rappresentazione viene affidata all’applicazione a esso associata. Un controllo *non associato* invece contiene un valore fisso, che non cambia al variare delle tuple; in questo modo si rappresentano tipicamente le etichette delle maschere (che anche in questo caso possono essere sequenze di caratteri, o altri oggetti di tipo generico; se si vuole per esempio inserire un logo nella maschera, bisognerà associare a un control-

lo non associato la figura che rappresenta il logo). Infine, i controlli *calcolati* permettono di visualizzare i risultati di espressioni valutate su combinazioni arbitrarie di parametri costanti e valori degli attributi delle tuple. I controlli *calcolati* non possono essere usati per l'immissione o la modifica dei valori degli attributi.

Con una maschera è anche possibile inserire nuove tuple, riportando direttamente i valori degli attributi nei controlli *associati*. Con le maschere è anche possibile interrogare la base di dati, utilizzando un semplice comando di ricerca. Per modificare il valore di un attributo, si seleziona la tupla e si apporta la modifica direttamente sulla casella che presenta il valore dell'attributo. Le modifiche vengono rese persistenti spostandosi su una diversa tupla, tipicamente premendo i tasti per l'avanzamento o l'arretramento di una pagina che permettono di scandire le tuple della tabella. La Figura A.18 mostra una maschera sulle tuple di PERSONA, estese con un attributo che contiene una foto.

Le maschere sono in genere costruite su una singola tabella. È possibile definire delle maschere che ne contengono altre al loro interno, specificando dei legami tra i valori visualizzati nelle maschere. In questo modo si possono per esempio definire delle maschere per la gestione di tabelle con relazioni uno a molti, in cui la maschera interna rappresenta tutte le tuple della relazione di dettaglio associate all'elemento esterno (si pensi a una coppia di tabelle che descrivono gli ordini; una maschera può illustrare al livello più esterno i dati dell'ordine e in una sottomaschera può comparire la distinta degli ordinativi, appartenenti a due tabelle distinte).

Un *report* si definisce in modo analogo a una maschera, usando gli stessi strumenti e concetti. La differenza principale consiste nel fatto che un report ha normalmente l'obiettivo di fornire una descrizione del contenuto della base dati al



Figura A.18 Una maschera che permette di accedere ad alcuni attributi di PERSONA

livello globale. Per questo i report tipicamente contengono dei controlli *calcolati* che forniscono consuntivi, e non permettono la visione di tuple particolari. Un'altra differenza con le maschere è che in genere i report vengono stampati, invece di essere utilizzati in modo interattivo.

A.5 La definizione di macro

Le macro costituiscono un modo per specificare un insieme di azioni che il sistema deve compiere. In questo modo è possibile automatizzare l'esecuzione di un insieme di compiti. Mediante le macro si possono svolgere le seguenti azioni.

- Far interagire una maschera con altre maschere e con i report. Per esempio, avendo una maschera che descrive i dati dei clienti e una che descrive gli ordini, è possibile aggiungere alla prima maschera un pulsante che attiva la seconda maschera visualizzando solo i dati degli ordini del cliente visualizzato. Si può anche aggiungere un pulsante che permette di stampare un report che elenca la situazione contabile nei confronti del cliente, con l'ammontare di merce rispettivamente ordinata, consegnata e pagata.
- Selezionare e raggruppare automaticamente le tuple. In una maschera si può inserire un pulsante che permette di selezionare immediatamente le tuple che rispettano particolari condizioni.
- Assegnare i valori agli attributi. Usando una macro, è possibile assegnare a un campo di una maschera un valore ottenuto da altri campi o da altre tabelle della base di dati.
- Garantire l'accuratezza dei dati. Le macro sono molto utili per manipolare e validare i dati sulle maschere. Per esempio, si può definire una macro che reagisce a diversi valori di un attributo con diversi messaggi, e garantire in modo sofisticato che i dati inseriti siano corretti.
- Impostare le proprietà delle maschere, dei report e dei campi. Con le macro si possono automatizzare i cambiamenti di qualsiasi proprietà di questi oggetti. Per esempio, è possibile rendere invisibile una maschera quando serve il suo contenuto ma non serve che questo venga visualizzato.
- Automatizzare i trasferimenti dei dati. Se si devono trasferire ripetutamente dati tra Access e altre applicazioni (sia in lettura sia in scrittura), si può automatizzare il compito.
- Creare un proprio ambiente di lavoro. Si può specificare una macro che apre tutto un insieme di tabelle, query, maschere e report tutte le volte che si apre un database, personalizzando eventualmente le barre degli strumenti.
- Specificare le reazioni a certi eventi. Per ogni campo di una maschera è possibile specificare quale macro debba essere eseguita in corrispondenza di ogni evento di accesso, selezione o modifica. Questa caratteristica costituisce la base per la definizione di comportamenti reattivi in Access, che però presentano diverse limitazioni rispetto a quanto è offerto dai trigger; infatti, tali comportamenti scattano in Access solo quando viene utilizzata una maschera particolare per la

manipolazione della base di dati, mentre non scattano se la stessa operazione è eseguita direttamente in SQL o tramite un'altra maschera.

Per la definizione di macro bisogna selezionare l'opzione "Crea" e poi attivare l'icona con etichetta **Macro**. La finestra di progettazione di macro contiene nella metà superiore una tabella con due colonne: **Azione** e **Commento**. La macro è composta da una sequenza di azioni descritte nella colonna **Azione**, ciascuna su una diversa riga, cui può essere associata una breve descrizione. Nella metà inferiore della tabella compaiono un insieme di attributi che, per ogni singola azione, specificano i parametri dell'azione. È possibile specificare delle condizioni che devono essere verificate affinché un comando venga eseguito, aggiungendo la colonna **Condizione** tramite l'opzione presente nella toolbar. Si può inoltre far uso di semplici strutture di controllo.

I comandi disponibili possono essere divisi in varie tipologie. Una prima tipologia è costituita dai comandi che permettono di chiedere l'esecuzione di altri servizi, che possono essere altre macro, query, generici comandi SQL, o anche applicazioni esterne. Un'altra tipologia sono i comandi che permettono di accedere ai dati, scandendo il contenuto di una tabella o di una maschera. Vi sono poi i comandi che permettono di manipolare il contenuto della base di dati. Infine, vi sono famiglie di comandi che permettono di trasferire dati tra Access e altre applicazioni, di modificare la dimensione delle finestre e di aprire delle finestre di dialogo con l'utente.

Una semplice macro è descritta in Figura A.19. La macro è associata alle modifiche sull'attributo **Numero abitanti** di una città; la macro assegna all'attributo **Popolosa** il valore *Sì* se il numero di abitanti è superiore a un milione, altrimenti assegna all'attributo il valore *No*.

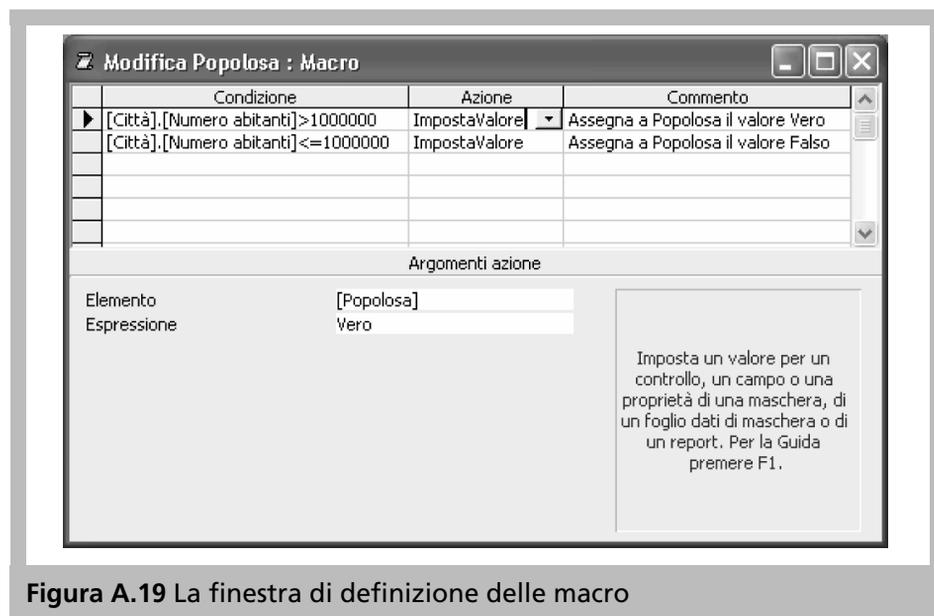


Figura A.19 La finestra di definizione delle macro