

## Le query di comando e di servizio in Access

In questa parte sono presentate alcune query di Access che permettono di: creare nuove tabelle, aggiungere o eliminare righe in una tabella, modificare il contenuto di alcuni record di una tabella. Vengono anche mostrate alcune query di servizio, utili per il controllo dei dati introdotti e le query a campi incrociati che visualizzano dati di sintesi in una tabella a doppia entrata.

L'uso del programma Access è importante anche per il confronto che si riesce a fare passando dalla query in *Visualizzazione Struttura* alla query in *Visualizzazione SQL* (scegliendo fra le proposte di *Visualizza* nella scheda *Progettazione*) lasciando che sia il sistema stesso a mostrare la sintassi per cancellare o modificare righe da una tabella, oltre a creare una nuova tabella o inserire righe in una tabella esistente usando un comando *Select* dentro un altro comando SQL.

Le query per cancellare o aggiornare righe, inserire record in una nuova tabella o in una tabella esistente prendono il nome di **query di comando** e si distinguono dalle normali query di selezione perché nella finestra delle query il loro nome è affiancato da un'icona speciale con un ! per evidenziare che si tratta di query che modificano il contenuto della base di dati. Nella terminologia di Access queste query prendono il nome di:



### Query di creazione tabella



### Query di accodamento



### Query di eliminazione



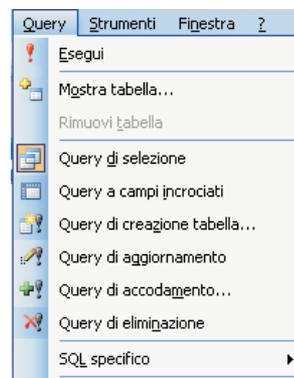
### Query di aggiornamento

Al momento dell'esecuzione di una query di comando il sistema avverte, con un messaggio in una finestra di dialogo, che la base di dati verrà modificata e nel corso dell'esecuzione informa quante righe saranno influenzate dalla query, permettendo, in entrambi i casi, di non procedere nell'esecuzione della query.



### Access 2003

Per creare una query di comando con Access 2003 bisogna entrare nella finestra delle query, scegliere l'opzione: *Crea una query in visualizzazione struttura* e, successivamente, scegliere **Query** nella barra dei menu. Viene presentato un menu che permette di scegliere tra le diverse *query di comando*. Si noti che nel medesimo menu è possibile accedere anche alla costruzione delle query a campi incrociati e a una finestra per scrivere query direttamente nel linguaggio SQL.



Per creare una query di comando bisogna scegliere l'opzione: *Crea una query in visualizzazione struttura* e, scegliere le tabelle necessarie alla costruzione della query. A questo punto la scheda *Progettazione* assume il seguente aspetto:



Nella scheda sono proposti i diversi tipi di query di comando ed è evidenziato il tipo *Selezione*, termine con il quale sono indicate le normali interrogazioni. Si noti che, nella stessa scheda, è possibile scegliere anche *query a campi incrociati* e (con le query *Pass-through*) accedere a una finestra per scrivere interrogazioni direttamente nel linguaggio SQL.



### Query di creazione tabella

Per creare una nuova tabella, estraendo i dati da tabelle esistenti, bisogna accedere alla query in visualizzazione struttura, definire nella griglia QBE i campi che formeranno la nuova tabella, impostare gli eventuali criteri di selezione per la righe da generare e nella barra dei menu, tra le opzioni della scheda *Progettazione*, scegliere la query di tipo **Creazione tabella**.

Il sistema richiede in un'apposita finestra di precisare il nome della tabella da creare e, nel corso dell'esecuzione, in un messaggio, viene indicato quanti record saranno inseriti nella nuova tabella.

### PROGETTO 1 - Creare una nuova tabella, di nome **DipendentiProduzione**, con le seguenti informazioni sui dipendenti del dipartimento **Produzione**: codice identificativo, cognome, nome, stipendio e descrizione del dipartimento di appartenenza.

La figura seguente mostra come sono stati costruiti i diversi campi nella griglia della query e la corrispondente codifica nel linguaggio SQL. Il codice della query *CreaDipendentiProduzione* si può esaminare passando alla sua *Visualizzazione SQL*. Il comando riportato in figura è stato ripulito dalle parentesi ridondanti. Si osservi che nella nuova tabella confluiscono campi che provengono da differenti tabelle.

Il codice mostra che se si vuole conservare la tabella risultante da una interrogazione, occorre aggiungere al comando *Select* la clausola **INTO** seguita dal nome da assegnare alla nuova tabella.

Le colonne della nuova tabella il cui contenuto è mostrato in figura hanno gli stessi nomi, formati e dimensioni degli attributi delle tabelle di origine.

ID	Nome	Cognome	Stipendio	Descrizione
1	Mario	Rossi	32000	Produzione
10	Margherita	Colombi	65000	Produzione
11	Fabrizio	Magenta	41000	Produzione
2	Giovanni	Bianchi	25500	Produzione

Tra i campi della tabella ci potrebbero essere anche campi ottenuti con espressioni di calcolo.



### Query di accodamento

In questo tipo di query i record prodotti dall'interrogazione vengono aggiunti, come nuovi record, ai record di una tabella esistente. Naturalmente i nuovi record devono essere di tipo compatibile con quelli della tabella di arrivo e i nuovi valori non devono violare i vincoli di integrità nei dati.

**PROGETTO 2** - Data una tabella di nome **NuoviDipendenti** con le informazioni dei dipendenti appena assunti, si desiderano aggiungere alle righe di **Impiegati** le informazioni di due soli nuovi assunti in corrispondenza degli ID di valore 2 e 3.

ID	Nome	Cognome	Residenza	Dipartimento	Stipendio
2	Giovanni	Bianchi	Milano	Prod	25500
3	Gianni	Verdi	Roma	Amm	26000
4	Giuseppe	Neri	Milano	Mkt	30000
* (toto)					0

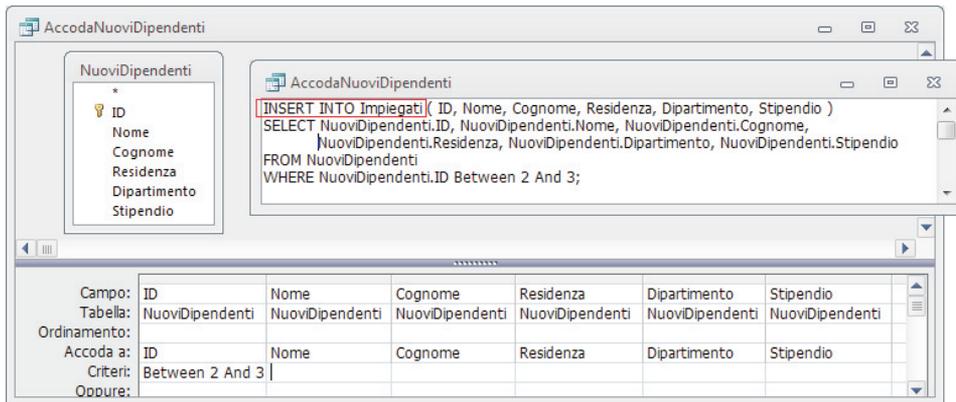
La tabella *NuoviDipendenti* ha il seguente schema:

**NuoviDipendenti** (ID, Nome, Cognome, Residenza, Dipartimento, Stipendio)

Si noti che lo schema della nuova tabella differisce da quello di *Impiegati* per l'inversione dell'ordine degli attributi *Dipartimento* e *Stipendio*.

Per accodare le righe desiderate bisogna trascinare tutti i campi di *NuoviDipendenti* nella griglia della query in visualizzazione struttura e impostare i criteri di selezione. Nella scheda *Progettazione* scegliere query di tipo **Accodamento**.

Il sistema richiede, con una finestra di dialogo, di precisare il nome della tabella alla quale accodare le righe selezionate dalla query da creare e, nel corso dell'esecuzione, in un messaggio, viene visualizzato il numero dei record che saranno accodati alla tabella *Impiegati*.



La figura mostra la query *AccodaNuoviDipendenti* in *Visualizzazione Struttura* e il corrispondente codice SQL ripulito dalle parentesi ridondanti.

Il codice mostra che, se si vogliono aggiungere le righe della tabella risultante da una query alle righe di una tabella già esistente, si deve usare il comando **INSERT INTO** seguito dal nome della tabella e dalla frase *Select* che determina le righe da aggiungere nella tabella. La precisazione dei nomi dei campi della vecchia tabella permette di attuare, se necessario, il corretto posizionamento dei valori prodotti dalla query nella tabella destinazione.

L'esecuzione della query modifica il numero di righe contenute nella tabella *Impiegati* come mostrato nella seguente figura.

ID	Nome	Cognome	Residenza	Stipendio	Dipartimento
1	Mario	Rossi	Torino	32000	Prod
2	Giovanni	Bianchi	Milano	25500	Prod
3	Gianni	Verdi	Roma	26000	Amm
5	Marco	Viola	Palermo	28300	
6	Enrico	Mori	Torino	25000	Mag
10	Margherita	Colombi	Roma	65000	Prod
11	Fabrizio	Magenta	Torino	41000	Prod
12	Franco	Volpi	Bari	61000	Amm
13	Ugo	Boss	Cagliari	85000	Direz
14	Mario	Gatti	Firenze	57000	R&S
16	Elisabetta	Gregis	Roma	29000	Amm
17	Laura	Moretti	Venezia	52600	Mkt
18	Erica	Bruni	Firenze	61500	Mag
19	Anita	Bianco	Perugia	39000	Direz



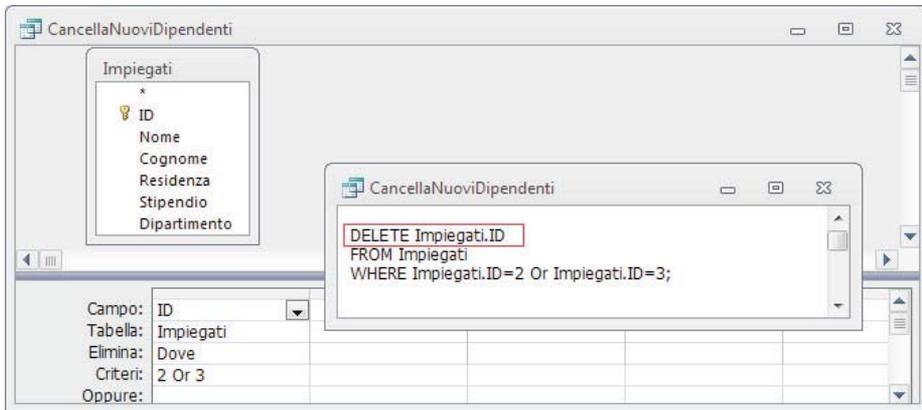
## Query di eliminazione

Una *query di eliminazione* permette di cancellare da una tabella le righe che soddisfano a certi criteri. Per costruire una *query di eliminazione* bisogna, come nelle precedenti query di comando, identificare mediante opportune condizioni le righe che saranno interessate all'operazione. Sino a questo punto, in pratica, ci si limita a costruire una normale query di selezione. Nel caso specifico delle query di eliminazione i campi da trascinare sulla griglia della QBE sono solo quelli necessari alla costruzione di una condizione nella riga *Criteri*. È opportuno, prima di procedere, eseguire la query per visualizzare i record che saranno cancellati.

Nella scheda *Progettazione* scegliere **Eliminazione** per il tipo della query. Quando la query è eseguita il sistema comunica, con un messaggio, il numero di record che saranno eliminati e chiede conferma prima di proseguire.

### PROGETTO 3 - Eliminare dalla tabella Impiegati i dipendenti aventi l'ID di valore 2 e 3.

La figura seguente mostra la query *CancellaNuoviDipendenti* in *Visualizzazione Struttura* e il corrispondente codice SQL.

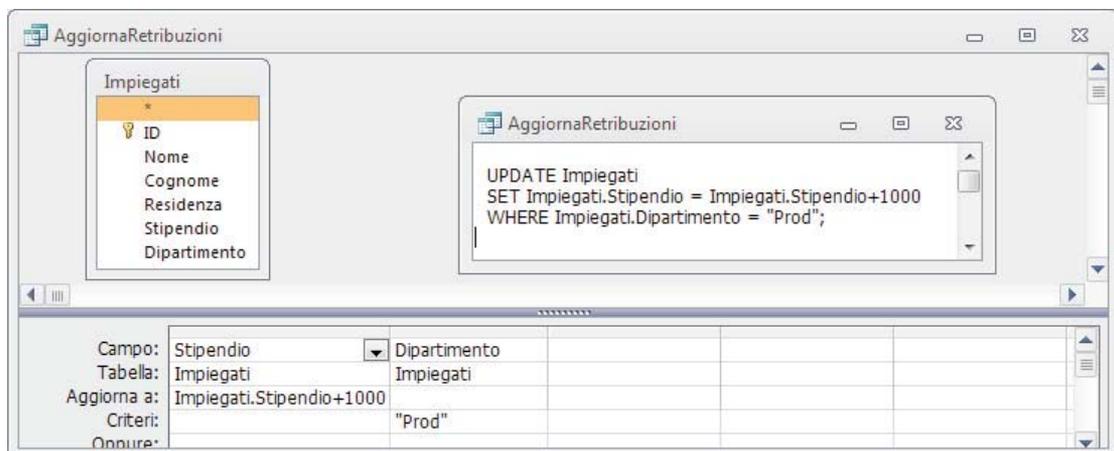


### Query di aggiornamento

Una *query di aggiornamento* ha lo scopo di modificare i valori di uno o più campi di una tabella in corrispondenza delle righe identificate da una condizione. La costruzione di una query di aggiornamento richiede di definire il nuovo valore nei campi da aggiornare e l'eventuale condizione che identifica i record da aggiornare. Pertanto nella griglia della QBE vanno trascinati i campi da aggiornare e i campi necessari per operare le selezioni. L'espressione che definisce i nuovi valori si inserisce in una riga delle griglia QBE etichettata con *Aggiorna a*: che appare quando nella scheda *Progettazione* si seleziona **Aggiornamento** per il tipo della query.

### PROGETTO 4 - Aumentare di 1000 euro le retribuzioni dei dipendenti del dipartimento Produzione.

Bisogna utilizzare i campi *Stipendio* e *Dipartimento* della tabella *Impiegati* come mostrato nella figura sottostante. Il codice SQL della query *AggiornaRetribuzioni* è mostrato nella finestra SQL della medesima figura.



## Query a campi incrociati e query di servizio



### Query a campi incrociati

Le query di tipo *A campi incrociati*, a differenza delle query di comando, non modificano i contenuti della base di dati. Il loro scopo consiste nel fornire rapporti di sintesi formati da raggruppamenti organizzati logicamente su due dimensioni. Per esempio, si consideri una tabella che raccoglie i voti degli studenti di una classe nelle diverse materie:

**VotiClasse** (NumRegistrazione, CodiceStudente, CodiceMateria, Data, Voto).

Si vuole produrre una tabella che riporta la media dei voti per studente (prima dimensione) e per materia (seconda dimensione).

La tabella a campi incrociati dispone le materie come intestazioni di colonna e gli identificativi degli studenti come intestazioni delle righe. All'incrocio tra righe e colonne si trova la media dei voti conseguiti nelle diverse discipline. In pratica bisogna chiedere al sistema di fare un doppio raggruppamento (per studente e per materia) e, all'interno di ogni raggruppamento, aggregare i voti con la funzione di raggruppamento AVG.

**PROGETTO 5 - Produrre una tabella a campi incrociati per conoscere dove risiedono i dipendenti descritti nella tabella *Impiegati*. La tabella deve avere come intestazioni di colonna i diversi dipartimenti, nelle righe devono essere riportate le città e, all'incrocio tra righe e colonne, il numero di dipendenti che risiedono in quella città per i diversi dipartimenti.**

Nella struttura della query bisogna trascinare i tre campi necessari per costruire la tabella (*ID*, *Residenza*, *Dipartimento*); nella scheda *Progettazione* occorre poi selezionare il tipo di query: *A campi incrociati* e specificare per i tre campi il ruolo che hanno nella costruzione della tabella: *Intestazione riga* oppure *Intestazione colonna* per i campi sui quali devono essere operati i raggruppamenti, *Valore* per il campo sul quale agisce la funzione di aggregazione (conteggio in questo esempio).

I ruoli dei tre campi sono indicati nella riga *Campi incrociati* dello schema QBE, mentre la funzione di aggregazione deve essere scelta nella riga *Formula*.

The screenshot shows a query design tool interface with the following components:

- Table List:** A list of fields from the 'Impiegati' table: ID, Nome, Cognome, Residenza, Stipendio, Dipartimento.
- SQL View:** A window displaying the SQL query:

```
TRANSFORM Count(Impiegati.ID) AS ConteggioDiID
SELECT Impiegati.Residenza
FROM Impiegati
GROUP BY Impiegati.Residenza
PIVOT Impiegati.Dipartimento;
```
- Design Grid:** A table defining the query structure:

Campo:	Residenza	Dipartimento	ID	
Tabella:	Impiegati	Impiegati	Impiegati	
Formula:	Raggruppamento	Raggruppamento	Conteggio	
Campi incrociati:	Intestazione riga	Intestaz. colonna	Valore	
Ordinamento:				
- Field Properties:** A dropdown menu for the 'ID' field, showing options: 'Intestazione riga', 'Intestaz. colonna', 'Valore', and '(Non visualizzato)'. 'Valore' is selected.

La figura mostra la query in *Visualizzazione struttura* e il codice SQL corrispondente. I commenti inseriti suggeriscono come utilizzare in Access le clausole **TRANSFORM** e **PIVOT**, in aggiunta alla clausola **GROUP BY**, per costruire tabelle a campi incrociati in linguaggio SQL. Nella tabella generata dalla query a campi incrociati, riportata in figura, si noti la prima colonna, con <> come intestazione del dipartimento, che raggruppa i dipendenti che mancano della corrispondente informazione nella tabella *Impiegati*.

Residen: <>	Amn	Dire	Mag	Mkt	Prod	R&S
Bari	1					
Cagliari		1				
Firenze			1			1
Palermo	1					
Perugia		1				
Roma	1					1
Torino			1			2
Venezia					1	

Access mette a disposizione strumenti per realizzare query che indichiamo con il nome di **query di servizio**:

- Query di ricerca dati non corrispondenti
- Presentazione dei valori più alti o più bassi di una query.
- Query di ricerca duplicati (che viene presentata al termine del Paragrafo 10).

#### • Query di ricerca dati non corrispondenti

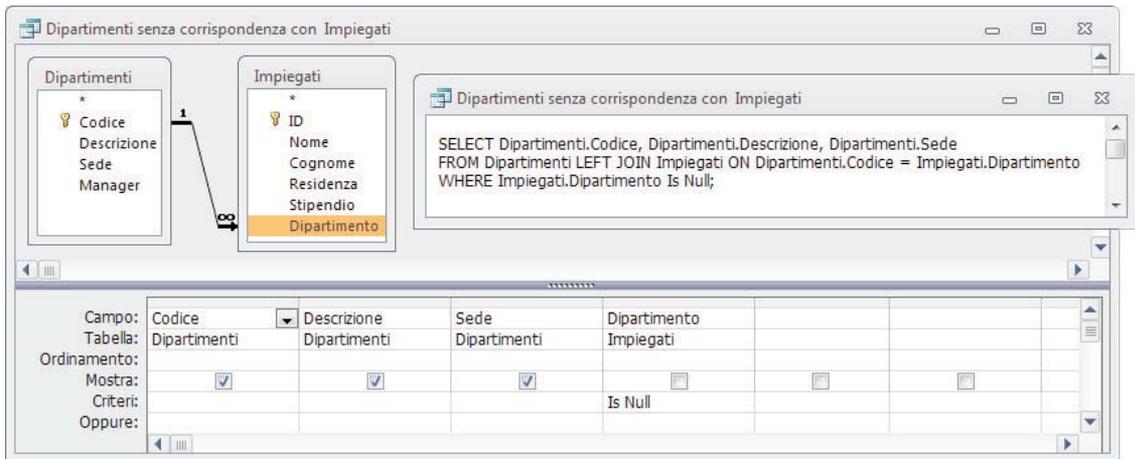
È una query utile per controllare se un dato presente in una tabella non ha corrispondenze nell'altra. Per esempio, si può verificare se ci sono dipartimenti senza impiegati o, viceversa, impiegati senza un corrispondente dipartimento di assegnazione oppure se nel campo *Manager* di *Dipartimenti* ci sono valori che non hanno corrispondenza nella tabella *Impiegati*. Alcune di queste situazioni sono controllate e impedito automaticamente con i vincoli di integrità referenziale, ma questo vale solo per alcune condizioni e solo se i vincoli sono stati introdotti.

La query di controllo viene prodotta automaticamente da un generatore. Bisogna scegliere **Creazione guidata Query ricerca dati non corrispondenti** nella finestra **Nuova query** attivata dalla scelta **Creazione guidata query** nella scheda **Crea**.

Il generatore richiede con successive finestre di dialogo di precisare: il nome delle tabelle interessate, dei campi che stabiliscono la corrispondenza e dei campi da visualizzare in output. Il generatore produce la query, le assegna un nome lungo, ma descrittivo del contenuto, e la esegue.

Nel caso delle tabelle *Impiegati* e *Dipartimenti* si vuole controllare, per esempio, se ci sono dipartimenti senza impiegati. Dopo aver avviato la creazione guidata, bisogna indicare come prima tabella *Dipartimenti*, quindi la tabella *Impiegati*, definire i campi corrispondenti nelle due tabelle e scegliere i valori da visualizzare, per esempio: *Codice*, *Descrizione* e *Sede* di *Dipartimento*. Access genera la query *Dipartimenti senza corrispondenza con Impiegati* la cui struttura e il relativo codice SQL sono mostrati nella figura seguente.

L'esecuzione della query produce un elenco con i dati del dipartimento *Personale* per il quale non ci sono dipendenti assegnati.



Il metodo adottato da Access per ricercare i valori non corrispondenti è quello indicato nel Capitolo 5 in un problema simile: le tabelle *Dipartimenti* e *Impiegati* sono congiunte con un *left join*, in modo che nella congiunzione compaiano tutte le righe di *Dipartimenti*. Le righe di *Dipartimenti* correlate a righe di *Impiegati* inesistenti sono completate con valori nulli che sono controllati nella clausola *Where*.

Si noti che volendo controllare la presenza di impiegati assegnati a dipartimenti inesistenti è necessario rispondere alla creazione guidata dichiarando come prima tabella *Impiegati* e, come seconda tabella, *Dipartimenti*. In questo modo viene costruita, con un *left join*, una congiunzione che comprende tutte le righe di *Impiegati* e gli eventuali impiegati senza corrispondenza con i dipartimenti possono essere riconosciuti per la presenza di valori nulli nella congiunzione.

### • Presentazione dei valori più alti o più bassi di una query

A volte la tabella prodotta da un'interrogazione contiene molte righe e solo alcune sono di interesse per chi esegue l'interrogazione. La tabella deve essere ordinata su un campo in base al quale si vogliono cercare i valori più alti o più bassi. L'ordinamento è decrescente per visualizzare i valori più alti e crescente per evidenziare i valori più bassi.

Nella query in *Visualizzazione Struttura*, facendo clic sulla piccola freccia nella casella di fianco a *Restituisci* nella scheda *Progettazione* viene data la possibilità di digitare il numero di righe che si vogliono visualizzare. Questa query corrisponde al comando **SELECT TOP** del linguaggio SQL. La figura mostra la struttura e il codice SQL di una query con la richiesta di elencare i 3 dipendenti della tabella *Impiegati* con lo stipendio più alto.

