



Le query in Access

Gabriele Russo Russo

Cos'è una query?

Una query è una interrogazione riguardo ai dati presenti in un database.

I dati che forniscono la risposta a questa “domanda”...

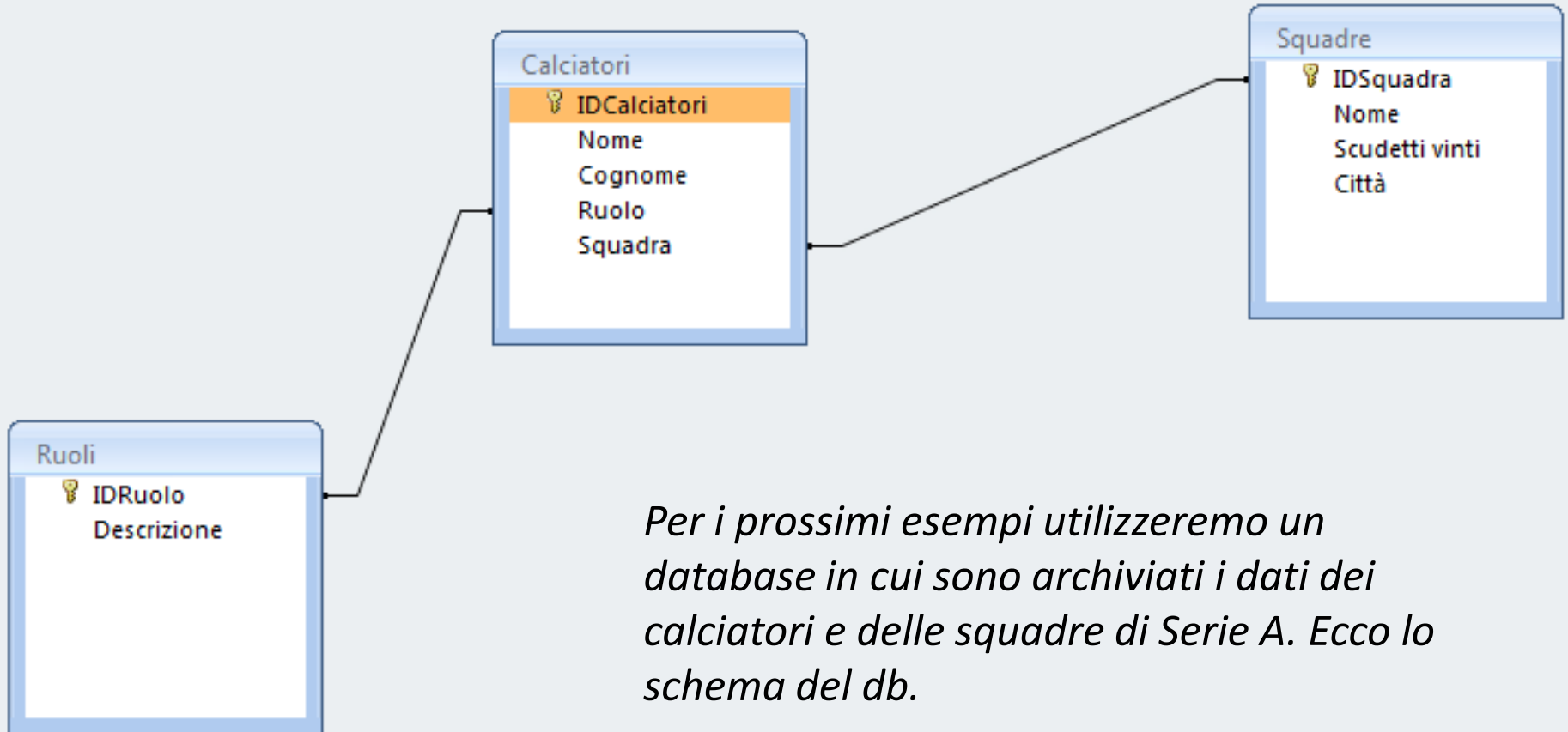
- provengono da una o più tabelle
- sono organizzati secondo la struttura di una tabella

Le query in Access

Abbiamo detto che la query è una specie di domanda. Il linguaggio con cui viene formulata la domanda al DBMS è **SQL** (*Structured Query Language*).

Access offre uno strumento che facilita il lavoro, il **QBE** (*Query By Example*), che scrive per noi la richiesta in SQL. Vediamo un esempio...

La nostra prima query



Calciatori

*

🔑 IDCalciatori
Nome
Cognome
Ruolo
Squadra

Mostra tabella

?

X

Tabelle

Query

Entrambe


Calciatori

Ruoli

Squadre

Aggiungi

Chiudi

Calciatori	
*	
	IDCalciatori
	Nome
	Cognome
	Ruolo
	Squadra

	Nome	Cognome
	Alessandro	Del Piero
	Gianluigi	Buffon
	Francesco	Totti
	Giampaolo	Pazzini
	Daniele	De Rossi
*		



Campo:
 Tabella:
 Ordinamento:
 Mostra:
 Criteri:
 Oppure:

Nome	Cognome		
Calciatori	Calciatori		
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

L'esempio che abbiamo appena visto è davvero molto semplice...ma anche inutile! Infatti abbiamo fatto solo una specie di "sintesi" dei dati della tabella "Calciatori".

Prima di passare ad un esempio più interessante, analizziamo il codice SQL di questa prima query:


```
SELECT Calciatori.Nome, Calciatori.Cognome  
FROM Calciatori;
```

Per capire questa sintassi, possiamo provare a interpretarla così:
"prendi i campi Nome e Cognome **dalla** tabella Calciatori"

Un altro esempio

Proviamo a creare una query che visualizzi solo le squadre che hanno vinto più di 10 scudetti nella propria storia.

La nostra query quindi avrà come base ancora solo una tabella (*Squadre* in questo caso), ma questa volta non dovrà visualizzare tutti i records, bensì solo quelli che rispondono ad un certo “criterio”.

Squadre	
*	
	IDSquadra
	Nome
	Scudetti vinti
	Città

	Nome	Città	Scudetti vin
	Juventus	Torino	27
	Inter	Milano	18
	Milan	Milano	17
*			



Campo:	Nome	Città	Scudetti vinti	
Tabella:	Squadre	Squadre	Squadre	
Ordinamento:				
Mostra:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Criteri:			>10	
Oppure:				

Anche questa volta diamo uno sguardo al codice SQL della query:

```
SELECT Squadre.Nome, Squadre.Città,  
Squadre.[Scudetti vinti]  
FROM Squadre  
WHERE (((Squadre.[Scudetti vinti])>10));
```

Abbiamo introdotto la parola **WHERE** che introduce una condizione. In questo caso “se il campo *Scudetti vinti* della tabella *Squadre* è maggiore di 10”. Notare che Access inserisce un paio di parentesi tonde di troppo, che potrebbero anche non esserci.

Terzo esempio

Fino ad ora abbiamo spesso parlato di relazioni e sappiamo come crearle. E' il momento di vedere come utilizzarle!

Vogliamo creare una query che visualizzi l'elenco dei calciatori, indicando per ognuno la squadra di appartenenza (non l'ID, ma il nome!). E' ovvio che dovremo basarci su due tabelle...



Mostra tabella

Tabelle Query Entrambe

Calciatori
Ruoli
Squadre

Aggiungi Chiudi



Campo:
 Tabella:
 Ordinamento:
 Mostra:
 Criteri:
 Oppure:

Nome	Cognome	Nome
Calciatori	Calciatori	Squadre
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>



IDCalciatori	Nome	Cognome	Ruolo	Squadra	Aggiungi nu
1	Alessandro	Del Piero	4	1	
2	Gianluigi	Buffon	1	1	
3	Francesco	Totti	4	2	
4	Giampaolo	Pazzini	4	3	
5	Daniele	De Rossi	3	2	
*	(Nuovo)				

IDSquadra	Nome	Scudetti vinti	Città	Ag
1	Juventus	27	Torino	
2	Roma	3	Roma	
3	Inter	18	Milano	
4	Milan	17	Milano	
5	Lazio	2	Roma	
*	(Nuovo)			

Calciatori.Nome	Cognome	Squadre.Nome
Alessandro	Del Piero	Juventus
Gianluigi	Buffon	Juventus
Francesco	Totti	Roma
Giampaolo	Pazzini	Inter
Daniele	De Rossi	Roma
*		

Ecco il risultato della nostra query di esempio e in alto le due tabelle di partenza. Access ha esplicitato la relazione che avevamo stabilito.

Anche questa volta diamo uno sguardo al codice SQL della query:

```
SELECT Calciatori.Nome, Calciatori.Cognome,  
Squadre.Nome  
FROM Squadre INNER JOIN Calciatori ON  
Squadre.IDSquadra = Calciatori.Squadra;
```

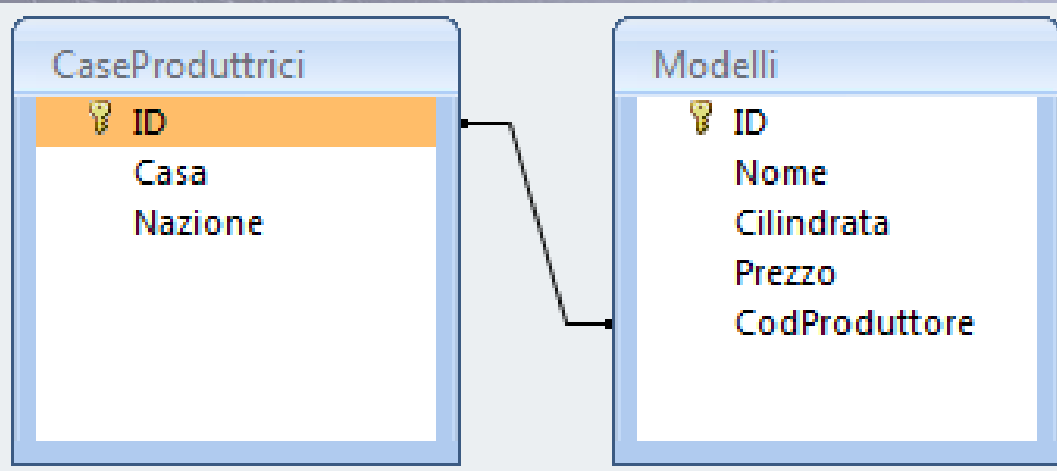
Questa volta abbiamo dovuto indicare la relazione. Per farlo usiamo **INNER JOIN...ON**. Cioè chiediamo di prendere i dati dalla tabella *Squadre* **unita** a *Calciatori*, secondo la corrispondenza dell'*IDSquadra* al valore del campo *Squadra* del calciatore.

Altri tipi di query

Fino ad ora abbiamo visto solo una piccola parte delle funzionalità delle query. Dobbiamo quindi analizzare le altre possibili soluzioni offerte da Access:

- Funzioni di aggregazione
- Query a campi incrociati
- Query di comando (creazione tabella, accodamento, etc...)

Per i prossimi esempi utilizzeremo un nuovo database: *Automobili*.



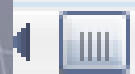
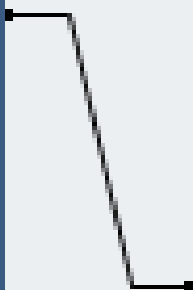
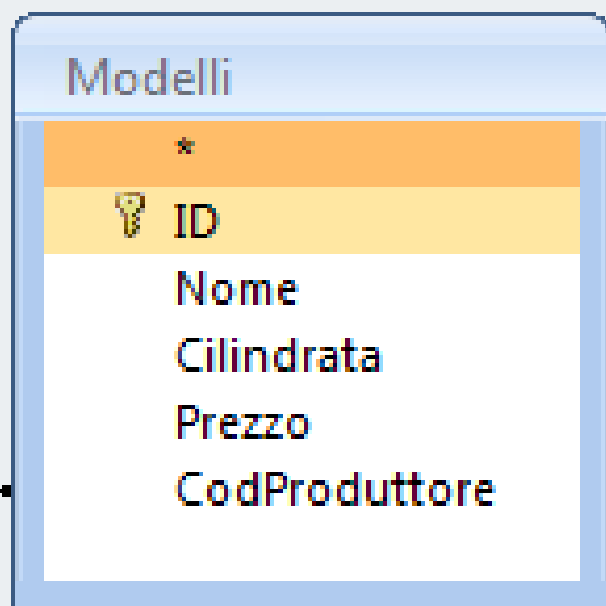
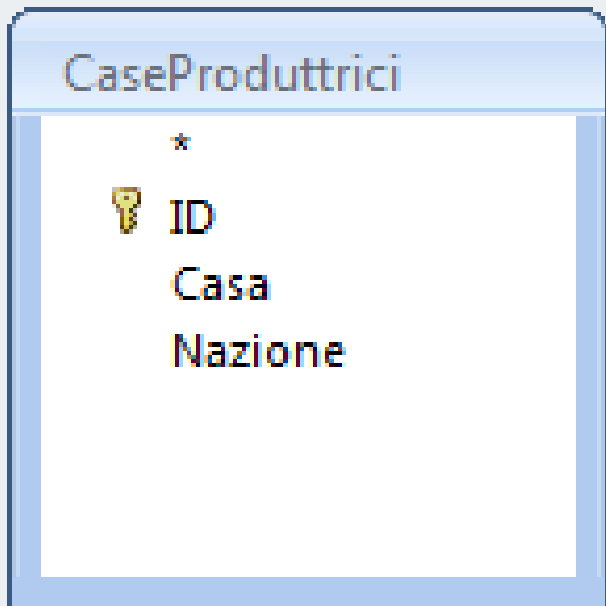
	CaseProduttrici	Modelli	Relazioni
	ID	Casa	Nazione
		1 Fiat	Italia
		2 Mercedes	Germania
		3 Renault	Francia
		4 Ferrari	Italia

	ID	Nome	Cilindrata	Prezzo	CodProdutt.
	1	Cinquecento	1300	€ 15.000,00	1
	2	Bravo	1900	€ 18.000,00	1
	3	Panda	1300	€ 14.000,00	1
	4	Classe A	1600	€ 22.000,00	2
	5	CLS	3500	€ 70.000,00	2
	6	Classe C	1900	€ 35.000,00	2
	7	Clio	1200	€ 12.000,00	3
	8	Megane	1500	€ 19.000,00	3
	9	599 GTB	6000	€ 250.000,00	4
	10	California	4200	€ 180.000,00	4

Funzione di aggregazione

Vogliamo ottenere una lista delle case costruttrici di automobili e il prezzo medio dei loro modelli.

Le colonne che ci interessano quindi sono la **Casa costruttrice**, raggruppando tutti i modelli della stessa casa, e il prezzo medio.



Campo:
 Tabella:
 Formula:
 Ordinamento:
 Mostra:
 Criteri:
 Oppure:

Casa	Prezzo
CaseProduttrici	Modelli
Raggruppamento	Media
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Casa	MediaDiPre
Ferrari	€ 215.000,00
Fiat	€ 15.666,67
Mercedes	€ 42.333,33
Renault	€ 15.500,00

Anche questa volta diamo uno sguardo al codice SQL della query:

```
SELECT CaseProduttrici.Casa, AVG (Modelli.Prezzo) AS PrezzoM  
FROM CaseProduttrici INNER JOIN Modelli ON CaseProduttrici.ID  
= Modelli.CodProduttore  
GROUP BY CaseProduttrici.Casa;
```

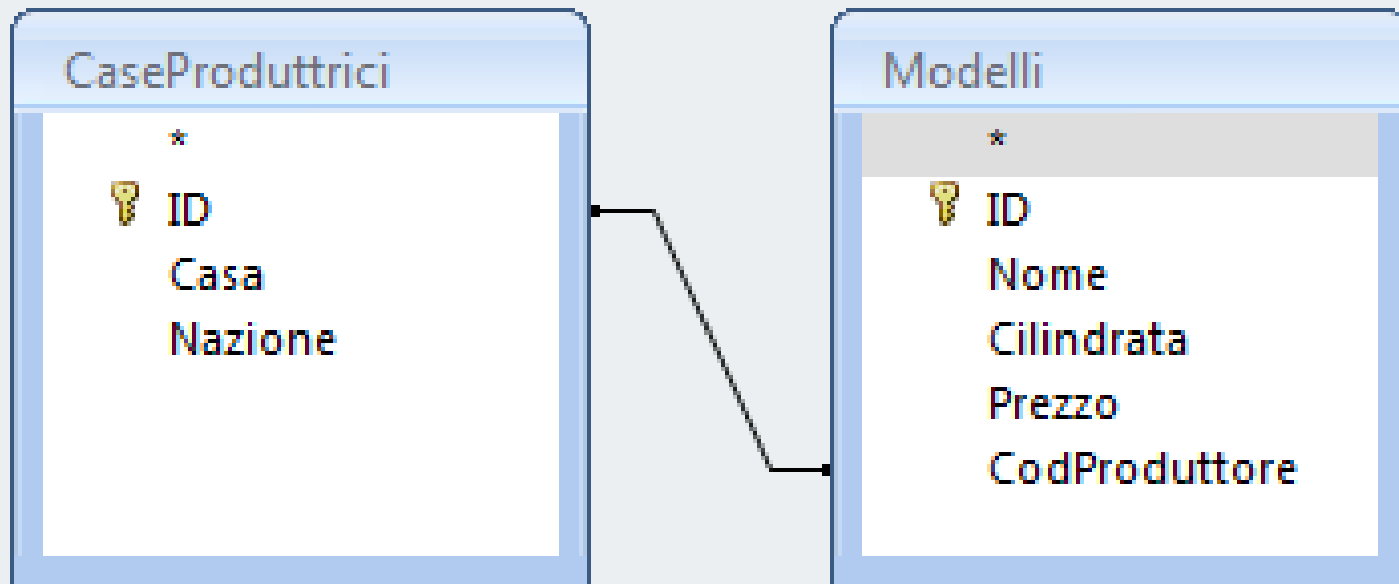
Le formule utilizzate (raggruppamento e media) si traducono in SQL con **GROUP BY** e **AVG** (da “average”).

Query a campi incrociati

Una **query a campi incrociati** estrae valori di riepilogo (somme, medie...) da un campo di una tabella e li raggruppa in una **tabella a doppia entrata**. Un esempio chiarisce la definizione...

Es.: vogliamo ottenere i prezzi medi delle varie case costruttrici relativamente alle diverse cilindrate.

L'intestazione delle colonne quindi sarà data dalle varie case, per ogni riga avremo invece una diversa cilindrata.



Campo:
 Tabella:
 Formula:
 Campi incrociati:
 Ordinamento:
 Criteri:
 Oppure:

Cilindrata	<input type="text" value="▼"/>	Casa	Prezzo
Modelli		CaseProduttrici	Modelli
Raggruppamento		Raggruppamento	Media
Intestazione riga		Intestaz. colonna	Valore

Ecco il risultato della nostra query a campi incrociati:

CampiIncorciati				
Cilindrata	Ferrari	Fiat	Mercedes	Renault
1200				€ 12.000,00
1300		€ 14.500,00		
1500				€ 19.000,00
1600			€ 22.000,00	
1900		€ 18.000,00	€ 35.000,00	
3500			€ 70.000,00	
4200	€ 180.000,00			
6000	€ 250.000,00			

Ed ecco il codice SQL:

```
TRANSFORM AVG(Modelli.Prezzo) AS MediaDiPrezzo  
SELECT Modelli.Cilindrata  
FROM CaseProduttrici INNER JOIN Modelli ON  
CaseProduttrici.ID = Modelli.CodProduttore  
GROUP BY Modelli.Cilindrata  
PIVOT CaseProduttrici.Casa;
```

Il codice deve sempre iniziare con **TRANSFORM** che introduce il criterio di raggruppamento (la media in questo caso...). Alla fine troviamo **PIVOT** che indica il campo da utilizzare per le intestazioni di colonna.

Query di comando

Le **query di comando** apportano modifiche a più records contemporaneamente.

Ne esistono quattro tipi:

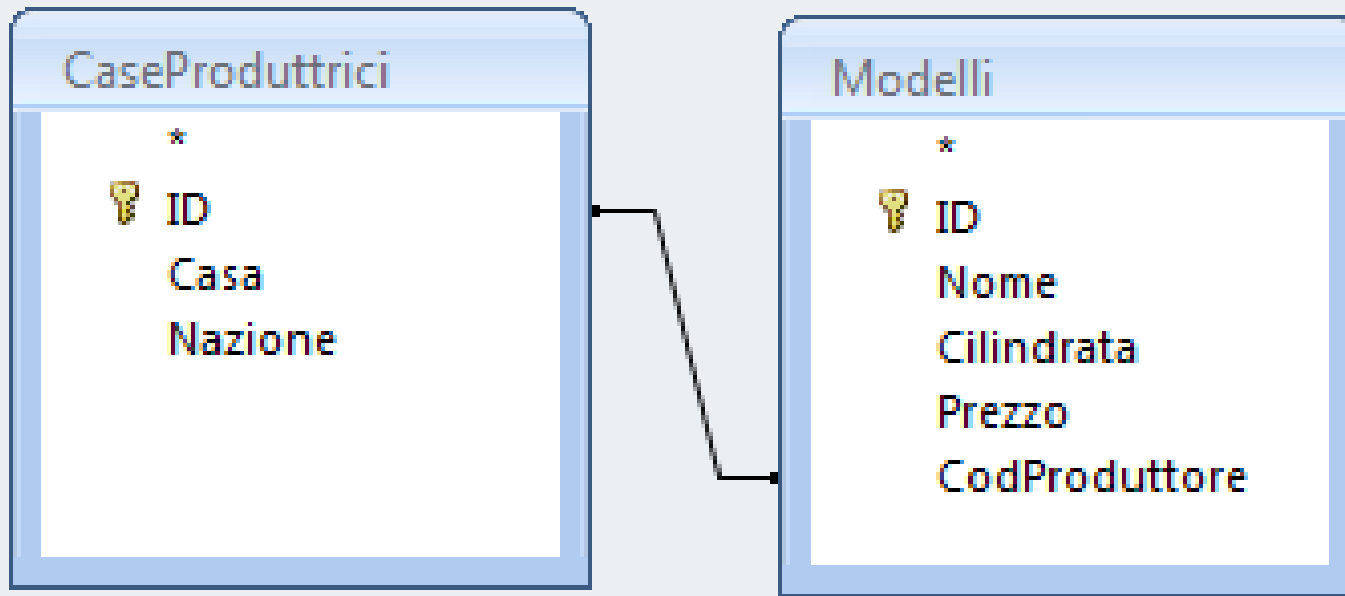
- Query di creazione tabella
- Query di aggiornamento
- Query di eliminazione
- Query di accodamento

*Per scegliere il tipo in Access, selezionata la scheda Query, dobbiamo aprire il menù **Tipo di query**.*

Query di creazione tabella

Questo tipo di query crea una nuova tabella, partendo dai dati di una o più tabelle. Tutte le query visualizzano i risultati in forma di tabelle, ma solo questo tipo crea fisicamente un nuovo oggetto-tabella.

Es.: vogliamo creare una tabella *ListinoPrezzi* che mostri tutti i modelli, la relativa casa produttrice e il prezzo. Selezioniamo il tipo query Creazione tabella e inseriamo il nome della nuova tabella da creare.



Campo:	Casa	Nome	Prezzo	
Tabella:	CaseProduttrici	Modelli	Modelli	
Ordinamento:				
Mostra:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Criteri:				
Oppure:				

Ed ecco il codice SQL:

```
SELECT CaseProduttrici.Casa, Modelli.Nome,  
Modelli.Prezzo INTO ListinoPrezzi  
FROM CaseProduttrici INNER JOIN Modelli ON  
CaseProduttrici.ID = Modelli.CodProduttore;
```

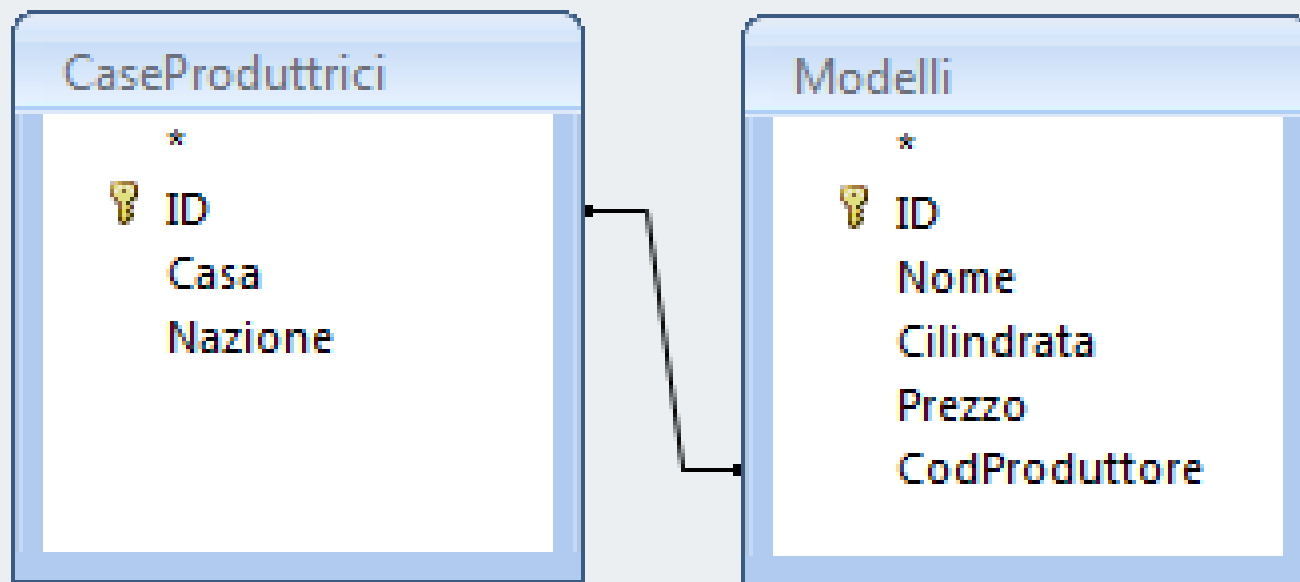
La parola **INTO** permette di indicare la nuova tabella in cui inserire i dati prodotti dalla query.

La tabella *ListinoPrezzi*, una volta creata, apparirà ovviamente non tra le query, ma tra le tabelle.

Query di aggiornamento

Con questa query possiamo modificare i dati presenti in una tabella. In particolare la **query di aggiornamento** modifica i dati di una colonna, in tutte le righe o solo in alcune (secondo un criterio).

Es.: vogliamo fare uno sconto del 10% su tutte le auto Fiat. Scegliamo il tipo Aggiornamento.



Campo:	Prezzo	Casa	
Tabella:	Modelli	CaseProduttrici	
Aggiorna a:	[Prezzo]*0,9		
Criteri:		"Fiat"	
Oppure:			

ID	Nome	Cilindrata	Prezzo	CodProdotto
1	Cinquecento	1300	€ 15.000,00	1
2	Bravo	1900	€ 18.000,00	1
3	Panda	1300	€ 14.000,00	1
4	Classe A	1600	€ 22.000,00	2
5	CLS	3500	€ 70.000,00	2
6	Classe C	1900	€ 35.000,00	2
7	Clio	1200	€ 12.000,00	3
8	Megane	1500	€ 19.000,00	3
9	599 GTB	6000	€ 250.000,00	4
10	California	4200	€ 180.000,00	4

ID	Nome	Cilindrata	Prezzo	CodProdotto
1	Cinquecento	1300	€ 13.500,00	1
2	Bravo	1900	€ 16.200,00	1
3	Panda	1300	€ 12.600,00	1
4	Classe A	1600	€ 22.000,00	2
5	CLS	3500	€ 70.000,00	2
6	Classe C	1900	€ 35.000,00	2
7	Clio	1200	€ 12.000,00	3
8	Megane	1500	€ 19.000,00	3
9	599 GTB	6000	€ 250.000,00	4

Ed ecco il codice SQL:

```
UPDATE CaseProduttrici INNER JOIN Modelli ON  
CaseProduttrici.ID = Modelli.CodProduttore  
SET Modelli.Prezzo = [Prezzo]*0.9  
WHERE (CaseProduttrici.Casa ="Fiat");
```

La sintassi usata è **UPDATE**...*campo*...**SET**...*nuovo-valore*

Query di eliminazione


Con questa query possiamo eliminare alcune righe di una tabella, che corrispondono ad un certo criterio. Se non viene specificato alcun criterio la tabella viene svuotata completamente.

N.B. questa query elimina interamente le righe (non si possono eliminare solo alcune colonne)

Es.: vogliamo eliminare dalla tabella *Modelli* tutte le auto con una cilindrata maggiore di 2000. Selezioniamo Query di eliminazione...

Modelli

*

 ID

Nome

Cilindrata

Prezzo

CodProduttore



Campo:	Modelli.*	Cilindrata	
Tabella:	Modelli	Modelli	
Elimina:	Da	Dove	
Criteri:		> "2000"	
Oppure:			

Ed ecco il codice SQL:

```
DELETE Modelli.*, Modelli.Cilindrata  
FROM Modelli  
WHERE ((Modelli.Cilindrata)>"2000");
```

La sintassi usata è **DELETE...FROM**. Da notare l'uso dell'asterisco per indicare tutti i campi della tabella.

Query di accodamento

E' molto simile alla query di creazione tabella, ma in questo caso vogliamo inserire i dati selezionati in una tabella esistente.

Se la tabella di origine e quella di destinazione hanno un numero di campi diverso possiamo indicare quali campi "copiare" oppure (se quella di origine ha meno campi dell'altra) i campi rimasti vuoti saranno NULL.

Potremmo usarla ad esempio nel caso avessimo una tabella *NuoviModelli*, i cui dati devono essere “accodati” a *Modelli*.

Ecco il codice SQL:

```
INSERT INTO Modelli  
SELECT NuoviModelli.*  
FROM NuoviModelli;
```

La sintassi usata è **INSERT INTO...** La seconda parte del codice è molto semplice.

A computer monitor and mouse are positioned on a desk in the foreground. The background is a light blue grid with several semi-transparent windows floating in the air. These windows contain text, including the words 'Video', 'Music', and 'Finance Groups'. The overall scene is a digital or virtual environment.

FINE

Gabriele Russo Russo