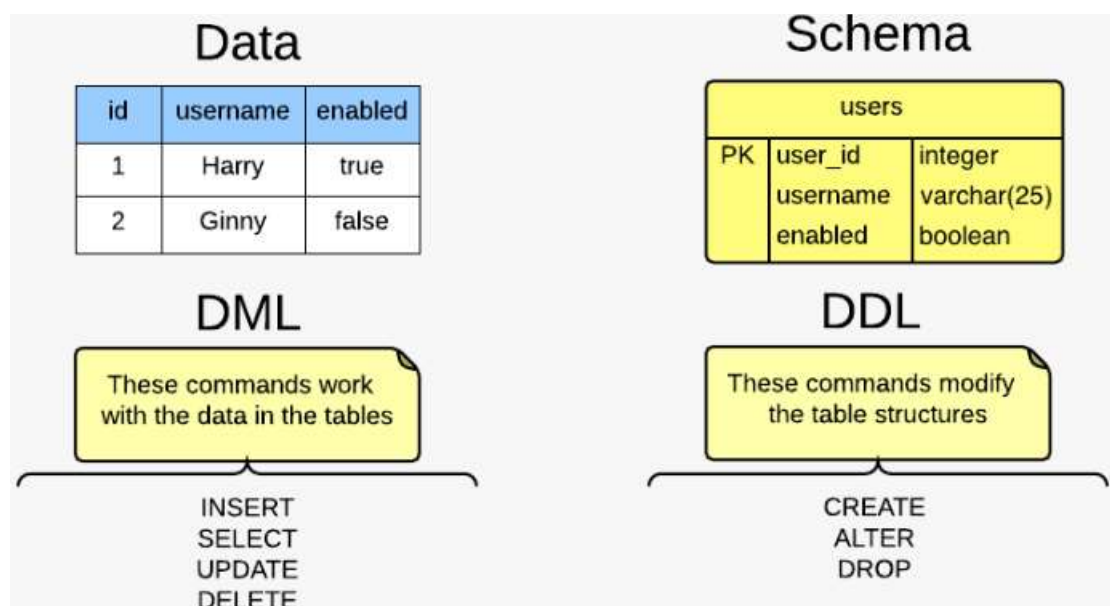




- **DDL (Data Definition Language):** consente di creare e modificare **schemi** di database;
- **DML (Data Manipulation Language):** consente di inserire, modificare e gestire i dati memorizzati;
- **DQL (Data Query Language):** consente di interrogare i dati memorizzati;
- **DCL (Data Control Language):** consente di creare e gestire strumenti di controllo ed accesso ai dati;
- **TCL (Transaction Control Language):** consente di avviare, concludere e gestire le **transazioni**.

>> anche su [wikipedia](https://en.wikipedia.org/wiki/SQL)

DDL e DML

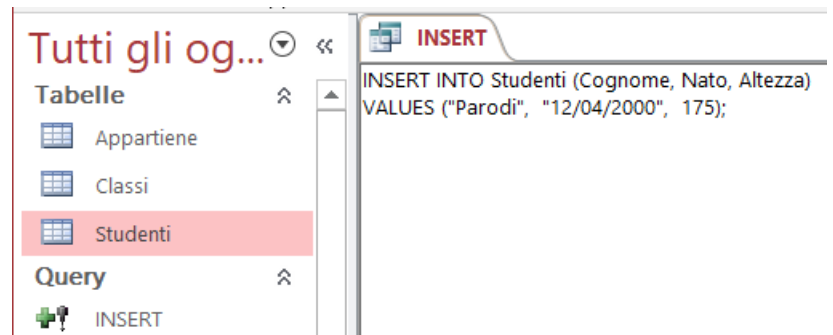
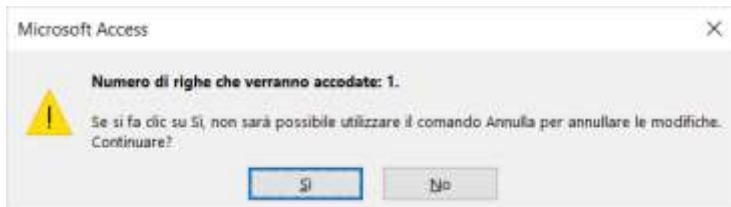
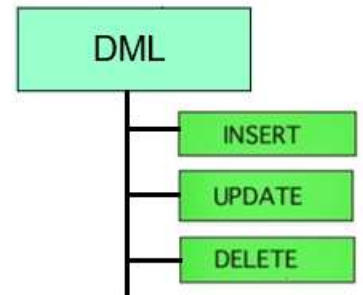


ACCESS: [Data Manipulation Language \(DML\)](#) per *manipolare* dati

(leggere, **inserire**, **aggiornare** o **eliminare** i dati in un database)

INSERT

INSERT INTO Studenti (Cognome, Nato, Altezza)
VALUES ("Parodi", "12/04/2000", 175);



ID inserito automaticamente:

	IdStudente	Cognome	Nato	Altezza
+	8	Giovane	15/02/2002	175
+	13	Vecchio	04/04/1998	165
+	14	Parodi	12/04/2000	175

Sintassi per [inserire](#)

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

se si inseriscono valori **in tutte le colonne** (in sequenza come da struttura):

```
INSERT INTO table_name  
VALUES (value1, value2, value3, ...);
```

[Aggiornare](#)

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

Se si omette la clausola WHERE, tutti i record nella tabella verranno aggiornati!

[>> Esempio](#)

[Cancellare](#)

```
DELETE FROM table_name WHERE condition;
```

Se si omette la clausola WHERE, tutti i record nella tabella verranno eliminati!

[>> Esempio](#)

ACCESS: [Data Definition Language \(DDL\)](#) per *definire*, agendo sullo schema

(creare, modificare o eliminare gli **oggetti** in un database)

CREATE TABLE

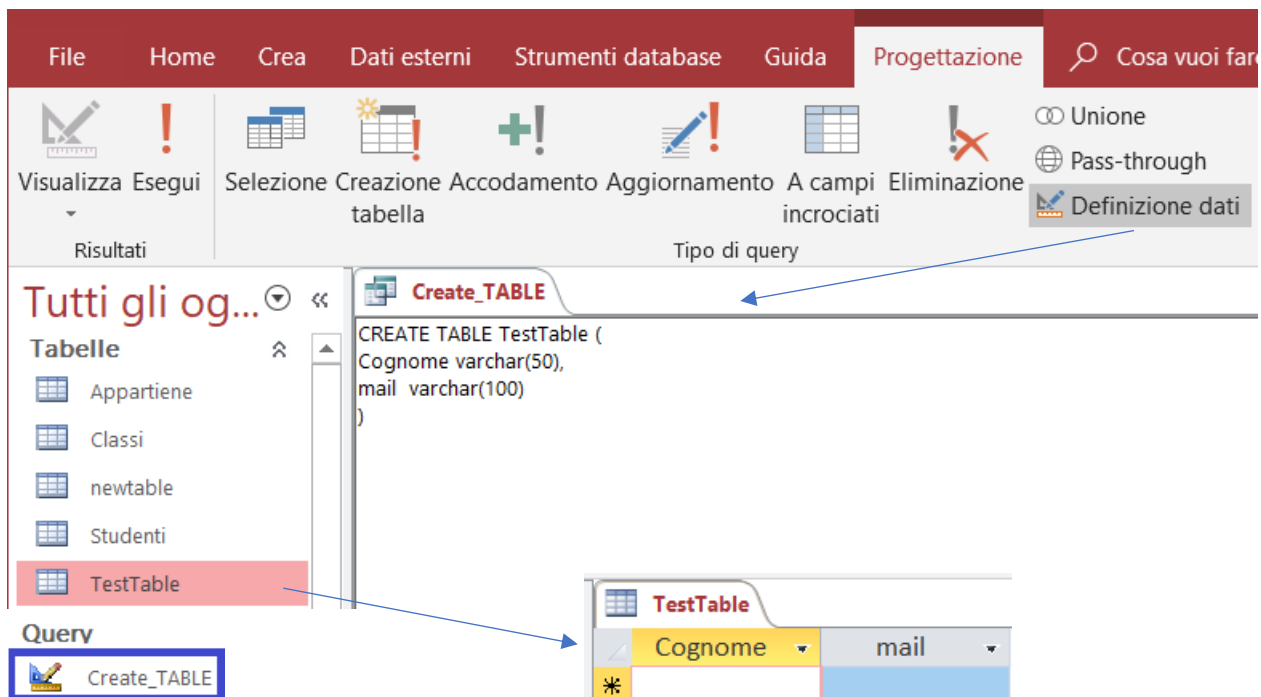
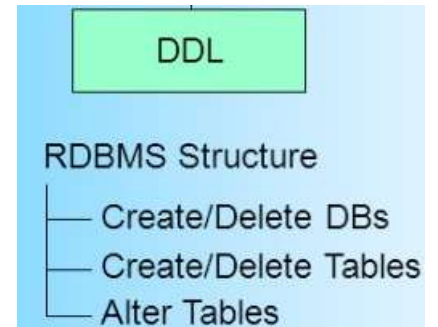
```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```

```
CREATE1 [TEMPORARY] TABLE tabella (  
    campo1 tipo [(dimensioni)] [NOT NULL] [WITH  
    COMPRESSION | WITH COMP] [indice1] [,  
    campo2 tipo [(dimensioni)] [NOT NULL] [indice2] [, ...] [,  
    CONSTRAINT indicemulticampo [, ...]  
)
```

Una clausola CONSTRAINT stabilisce diverse limitazioni per un campo (indici a campo singolo o multicampo) e può essere usata per stabilire la chiave primaria. È anche possibile usare l'istruzione CREATE INDEX per creare una chiave primaria oppure indici aggiuntivi nelle tabelle esistenti.

Semplificando:

```
CREATE TABLE NOMETABELLA (  
    NomeCampo1 Tipo1 [(dimensione1)] Vincoli1,  
    NomeCampo2 Tipo2[(dimensione2)] Vincoli2,  
    ...  
    NomeCampon TipoN[(dimensioneN)] VincoliN  
)
```

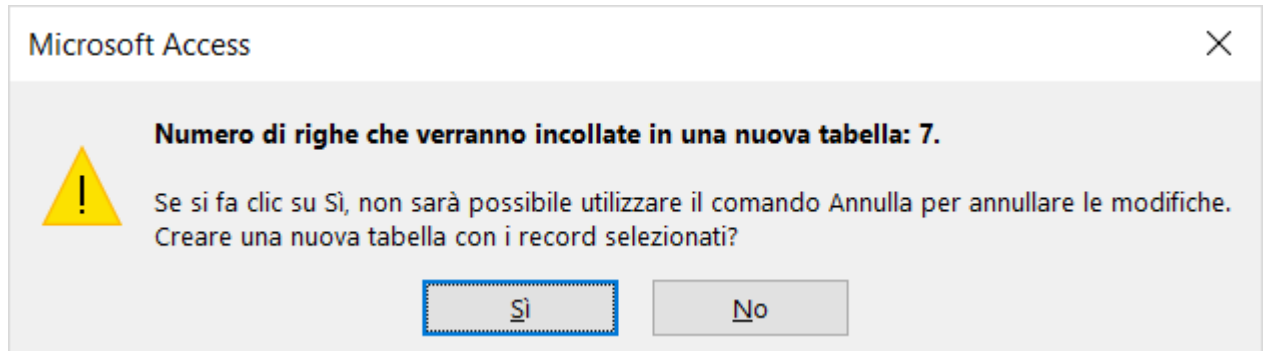


¹ **Nota:** Il motore di database di Microsoft Access non supporta l'uso di istruzioni CREATE TABLE o DDL con database di motori di database non Microsoft Access.

oppure usando **altra tabella** ... quindi **copiando**

Copiare tabelle

```
SELECT *  
INTO newtable  
FROM Studenti
```



Tutti gli og... <<

newtable

IdStudente	Cognome	Nato	Altezza
4	Rossi	10/09/1999	182
5	Verdi	15/03/1999	178
6	Bianchi	20/09/1999	178
7	Neri	12/01/2001	180
8	Giovane	15/02/2002	175
13	Vecchio	04/04/1998	165
14	Parodi	12/04/2000	175
*	(Nuovo)		

Tabella

- Appartiene
- Classi
- newtable
- Studenti

Query

- Copia_Table

Sintassi

```
SELECT *  
INTO newtable [IN externaldb]  
FROM oldtable  
WHERE condition;
```

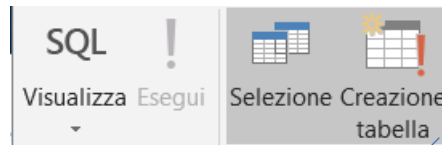
Solo alcune colonne

```
SELECT column1, column2, column3, ...  
INTO newtable [IN externaldb]  
FROM oldtable  
WHERE condition;
```

NB: in ACCES non è accettata la sintassi seguente:

```
CREATE TABLE new_table_name AS  
SELECT column1, column2, ...  
FROM existing_table_name  
WHERE ....;
```

Creazione guidata di tabella
è in realtà una **copia**:

A dialog box titled 'Creazione tabella' with a question mark and a close button. It contains a section 'Crea nuova tabella' with a text field for 'Nome tabella:'. Below it are two radio buttons: 'Database corrente' (selected) and 'Altro database:'. Under 'Altro database:' is a text field for 'Nome file:' and a 'Sfoglia...' button. On the right side, there are 'OK' and 'Annulla' buttons.

Si apre in visualizzazione struttura: potendo aggiungere tabella e selezionare campi per **copia**
analoga alla precedente:

Query
Copia_Table

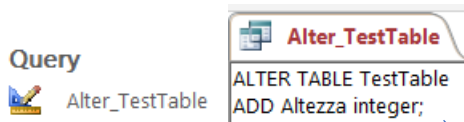
A screenshot of a SQL environment interface. At the top, there is a toolbar with icons for 'Creazione tabella', '+!', 'Aggiornamento', and 'A campi E incrociati'. Below the toolbar, a tab labeled 'Query6' is active. The main area shows a structure view for a table named 'Studenti'. The fields listed are 'IdStudente' (marked with a key icon), 'Cognome', 'Nato', and 'Altezza'. At the bottom, there is a table with the following content:

Campo:	Studenti.*	
Tabella:	Studenti	
Ordinamento:		
Mostra:	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Criteri:		

Modificare Tabelle (aggiungendo colonne):

ALTER Table

```
ALTER TABLE TestTable  
ADD Altezza integer;
```



Cognome	mail	Altezza
*		

Sintassi:

```
ALTER TABLE table_name  
ADD column_name datatype;
```

Nome campo	Tipo dati
Cognome	Testo breve
mail	Testo breve
Altezza	Numerico

Eliminare Tabelle

DROP Table

```
DROP TABLE TestTable;
```



Tabelle

- Appartiene
- Classi
- newtable
- Studenti

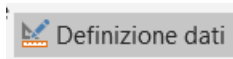


tabella cancellata

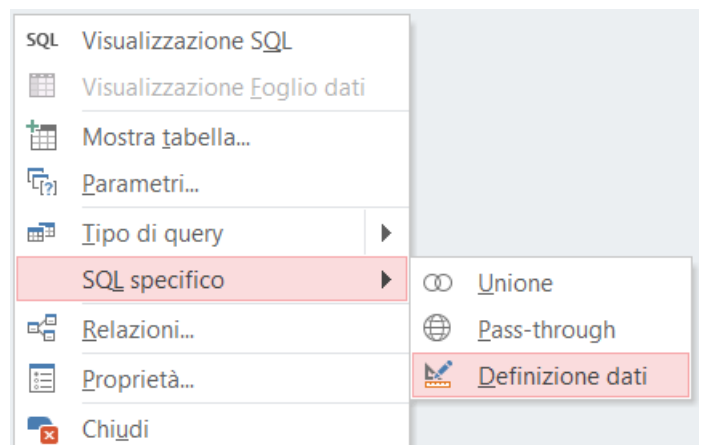
Sintassi

```
DROP TABLE table_name;
```

short cut

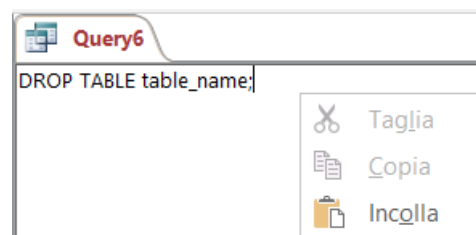
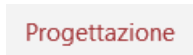


per visualizzare solo

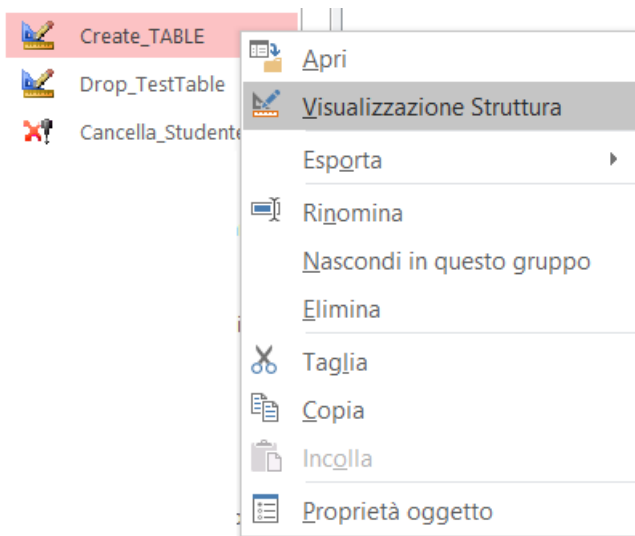


senza interfaccia visuale

in fase



Ricreiamo tabella cancellata:



Tabelle

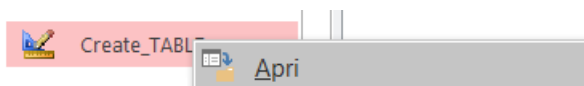
Appartiene
Classi
newtable
Studenti
TestTable

In **Visualizzazione Struttura** rivediamo il **codice SQL**: **CREATE TABLE TestTable (**
Cognome varchar(50),
mail varchar(100)
)

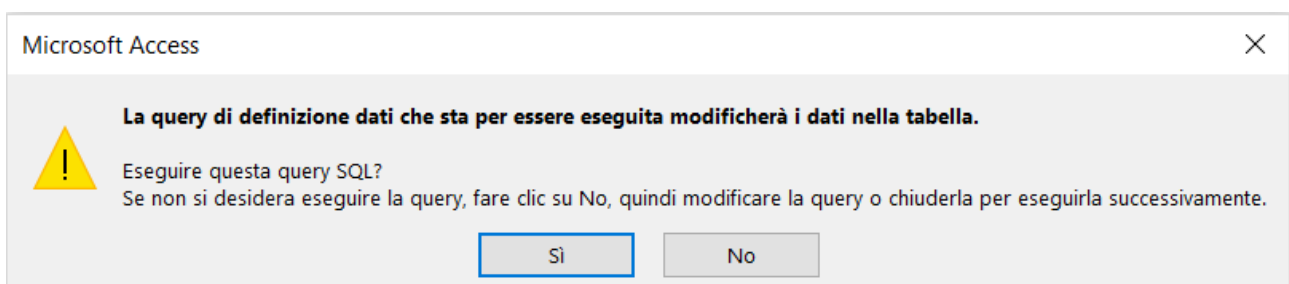
potendo aggiungere altri campi

```
CREATE TABLE TestTable (  
Cognome varchar(50),  
mail varchar(100),  
altezza integer  
)
```

Con **Apri**



prevede il controllo



Esempio di Aggiornamento

UPDATE Studenti **SET** Studenti.Altezza = [Quale altezza?]
WHERE Cognome=[Quale cognome?]

The screenshot illustrates the execution of an UPDATE query in Microsoft Access. The background shows a table with columns for name, date, and height. The query window displays the SQL statement: `UPDATE Studenti SET Studenti.Altezza = [Quale altezza?] WHERE Cognome=[Quale cognome?]`. Two dialog boxes prompt for input: one for the height (180) and one for the surname (Parodi). A final confirmation dialog box, titled "Microsoft Access", displays a warning icon and the message: "Numero di righe che verranno aggiornate: 1. Se si fa clic su Sì, non sarà possibile utilizzare il comando Annulla per annullare le modifiche. Continuare?". The "Sì" button is highlighted with a blue box, and an arrow points from it to the updated height value of 180 in the table.

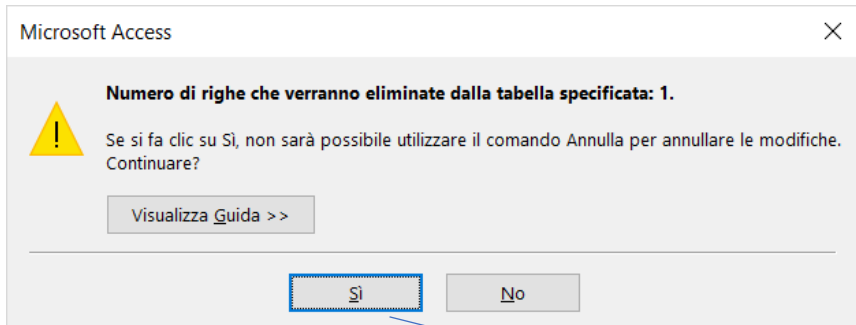
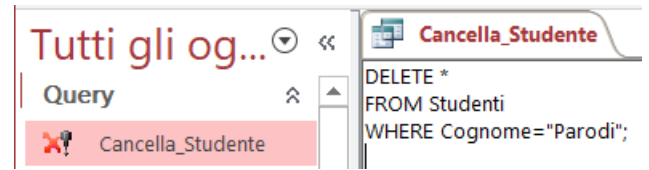
14	Parodi	12/04/2000	175
----	--------	------------	-----

14	Parodi	12/04/2000	180
----	--------	------------	-----

Esempio di Cancellazione



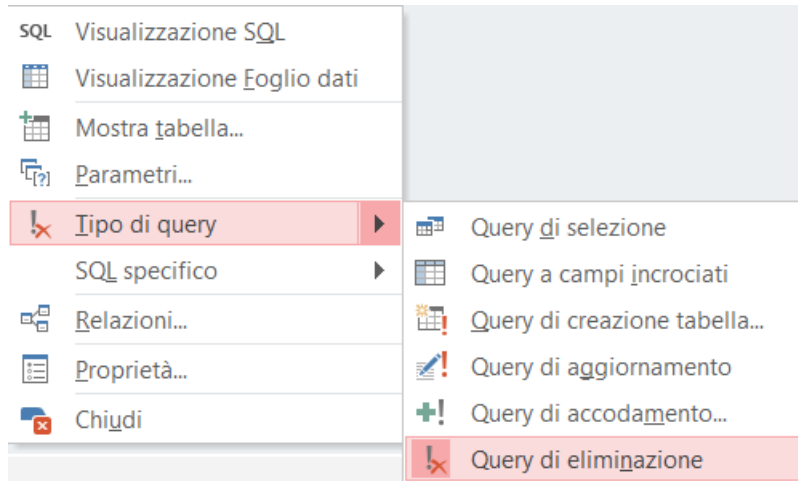
```
DELETE *  
FROM Studenti  
WHERE Cognome="Parodi"
```



+	13	Vecchio	04/04/1998	165
	(Nuovo)			

Equivalente senza specificare tutti i campi

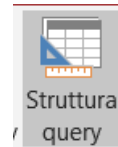
```
DELETE FROM Studenti  
WHERE Cognome="Parodi";
```



(in Access inserito simbolo * di default)

Modifiche (manipolazione dei dati) guidate

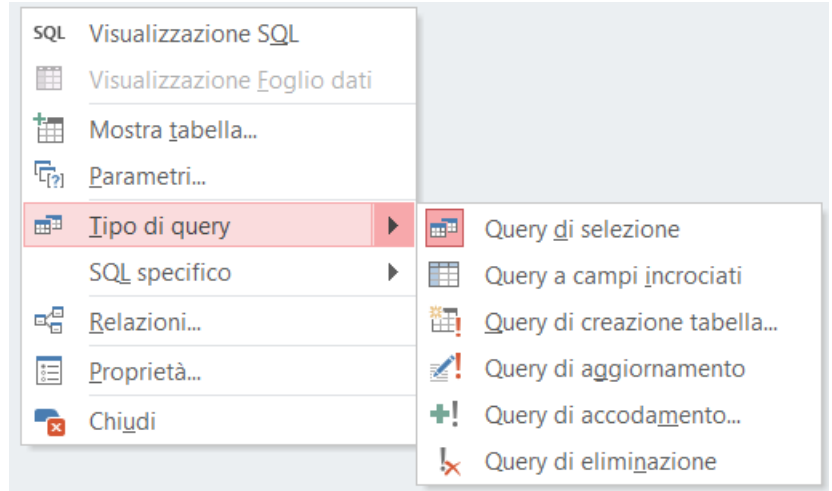
Di default, in Access, creando la **struttura di una query**



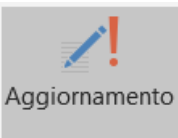
è prevista la realizzazione di query di **tipo selezione**



SQL **SELECT;**



Possibili **altri tipi:**

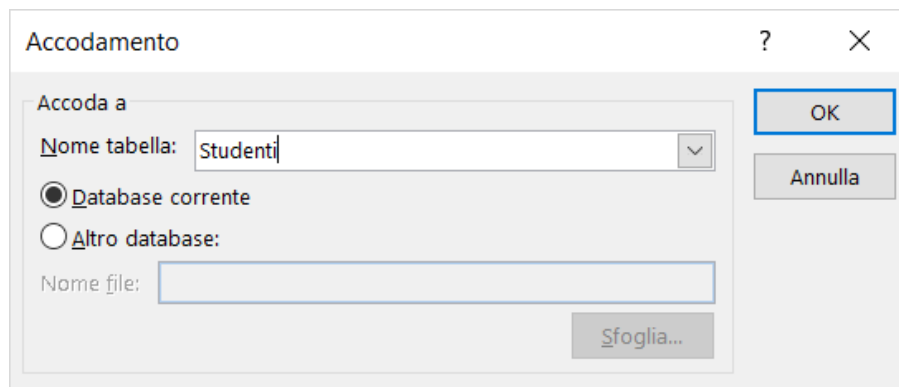


SQL **UPDATE SET;**



SQL **DELETE *;**

Modifiche guidate ... che richiedono il nome della tabella



automatico SQL di default: **INSERT INTO Studenti
SELECT;**

SQL **INSERT INTO Studenti
SELECT;**

**Reinserendo dopo la cancellazione,
si noti come non vengano riutilizzati
codici cancellati:**

+	13	Vecchio	04/04/1998	165
+	15	Parodi	12/04/2000	175

L' ID è infatti un contatore auto-incrementante