

## SQLite e Linguaggio SQL

### Cos'è SQLite

Un **DataBase** (*Base di Dati*) è un insieme di dati organizzati in una o più **Tabelle**, a loro volta strutturate in Righe (**Record**) e Colonne (**Campi**).

**SQLite** è una **Libreria Software** che consente di Gestire un DataBase.

Il sito ufficiale di SQLite è raggiungibile al seguente link: <https://www.sqlite.org/index.html>

SQLite, per consentire la gestione e l'accesso al DataBase, accetta comandi espressi in **Linguaggio SQL** (*Structured Query Language*).

La documentazione completa sui comandi e le funzionalità del *Linguaggio SQL per SQLite* è reperibile al seguente link: <https://www.sqlite.org/lang.html>

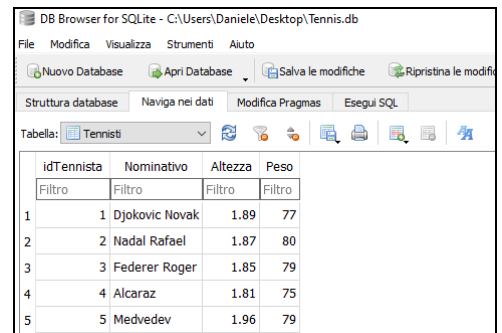
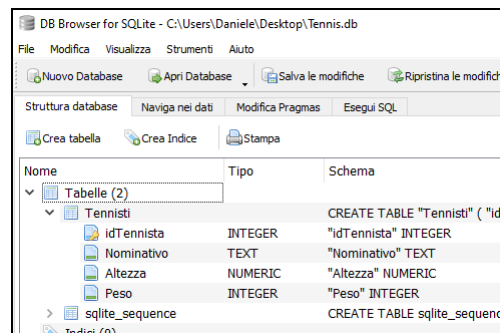
SQLite è di pubblico dominio (ossia è utilizzabile liberamente) ed è disponibile in diversi linguaggi/ambienti fra cui **C#** (Visual Studio), **Java** (Android Studio), **PHP**, ecc.

### DB Browser for SQLite

Per *Gestire*, in modo semplice e interattivo, un DataBase SQLite, useremo il software **DB Browser for SQLite**.

*DB Browser* consente di Creare DataBase e Tabelle, di Visionare, Inserire, Modificare i dati e, in generale, di inviare Comandi SQL a SQLite.

DB Browser è gratuito ed è scaricabile dalla sezione download del sito ufficiale <https://sqlitebrowser.org>



**Comando SQL**  
**CREATE TABLE**  
*(semplificato)*

**CREATE TABLE** consente di **Creare una Tabella** nel DataBase.

E' necessario indicare l'elenco dei Campi con i relativi Tipi.

```
CREATE TABLE <nome-tabella>
( <nome-campo> <tipo-campo>,
  <nome-campo> <tipo-campo>,
  ... )
```

<nome-tabella> è il nome della Tabella da creare.

<nome-campo> <tipo-campo> sono, rispettivamente, il **Nome** da attribuire al Campo e il suo **Tipo di Dato**.

Se il Campo è una **Chiave Primaria** è possibile specificarlo così:

```
<nome-campo> <tipo-campo> PRIMARY KEY
```

Se si desidera una **Chiave Primaria numerica Generata Automaticamente**:

```
<nome-campo> INTEGER PRIMARY KEY AUTOINCREMENT
```

I **Tipi di Dato** disponibili per i Campi delle Tabelle in SQLite sono:

<b>INTEGER</b>	numeri interi
<b>TEXT</b>	testo e/o stringhe
<b>NUMERIC</b>	numeri decimali (codificati in virgola fissa)
<b>REAL</b>	numeri decimali (codificati in virgola mobile)

Esempio:

```
CREATE TABLE Alunni
( idAlunno INTEGER PRIMARY KEY AUTOINCREMENT,
  Cognome TEXT,
  Nome TEXT,
  Classe INTEGER,
  Sezione TEXT,
  Altezza NUMERIC,
  Peso INTEGER )
```

Nome	Tipo	Schema
▼ Tabelle (2)		
▼ Alunni		CREATE TABLE Alunni ( idA
idAlunno	INTEGER	"idAlunno" INTEGER
Cognome	TEXT	"Cognome" TEXT
Nome	TEXT	"Nome" TEXT
Classe	INTEGER	"Classe" INTEGER
Sezione	TEXT	"Sezione" TEXT
Altezza	NUMERIC	"Altezza" NUMERIC
Peso	INTEGER	"Peso" INTEGER
sqlite_sequence		CREATE TABLE sqlite_sequid

**Comando SQL**  
**SELECT**  
*(semplificato)*

**SELECT** consente di **Ottenere Dati** dal DataBase (**Interrogazione** o **Query**).

Questo comando **Restituisce una Tabella** con i dati richiesti, nella forma richiesta.

E' possibile scegliere *quali campi includere*, *imporre un filtro* per decidere *quali record si desiderano* e anche *ordinare i dati* secondo il campo desiderato.

**SELECT** <elenco-campi>  
**FROM** <nome-tabella>  
**WHERE** <condizione>  
**ORDER BY** <campo>

<elenco-campi> è la lista dei campi desiderati, separati da virgola ( , ).

Se si specifica solo un asterisco ( \* ) si ottengono tutti i campi.

<nome-tabella> è il nome della Tabella da cui estrarre i dati.

<condizione> è una espressione booleana sui dati del record: saranno elencati solo i record per cui questa condizione è vera (True).

<campo> è il campo secondo il quale è necessario ordinare i record. E' possibile aggiungere la parola **DESC** per ottenere un ordinamento discendente.

Esempio:

idTennista	Nominativo	Altezza	Peso
Filtro	Filtro	Filtro	Filtro
1	Djokovic Novak	1.89	77
2	Nadal Rafael	1.87	80
3	Federer Roger	1.85	79
4	Alcaraz	1.81	75
5	Medvedev	1.96	79

```
SELECT Nominativo, Altezza
FROM Tennisti
WHERE Altezza > 1.86
ORDER BY Nominativo
```

Nominativo	Altezza
Djokovic Novak	1.89
Medvedev	1.96
Nadal Rafael	1.87

**Comando SQL**  
**INSERT**  
*(semplificato)*

**INSERT** consente di **Aggiungere un Record** a una Tabella e di memorizzare in esso i valori desiderati.

**INSERT INTO** *<nome-tabella>*  
**(** *<elenco-campi>* **) VALUES**  
**(** *<elenco-valori>* **)**

*<nome-tabella>* è il nome della Tabella in cui aggiungere il nuovo record.

*<elenco-campi>* è la lista dei campi ai quali si desidera attribuire un valore nel nuovo record. Nell'elenco non vanno inclusi i campi definiti con AUTOINCREMENT perché, in tal caso, è SQLite stesso ad attribuirvi automaticamente il valore.

*<elenco-valori>* è la lista dei valori da attribuire rispettivamente ai campi elencati precedentemente.

Esempio:

```
INSERT INTO Tennisti
( Nominativo, Altezza, Peso )
VALUES ( 'Zverev Alexander', 2.02, 85 )
```

nista	Nominativo	Altezza	Peso
	Filtro	Filtro	Filtro
1	Djokovic Novak	1.89	77
2	Nadal Rafael	1.87	80
3	Federer Roger	1.85	79
4	Alcaraz	1.81	75
5	Medvedev	1.96	79
6	Zverev Alexander	2.02	85

**Comando SQL**  
**DELETE**  
*(semplificato)*

**DELETE** consente di **Eliminare uno o più Record** da una Tabella.

E' necessario specificare una **clausola WHERE** (già vista nel comando *SELECT*) per specificare quali record si desidera rimuovere.

**DELETE FROM** <nome-tabella>

**WHERE** <condizione>

<nome-tabella> è il nome della Tabella in cui aggiungere il nuovo record.

<condizione> è una espressione booleana sui dati del record: saranno rimossi solo i record per cui questa condizione è vera (True).

Esempio:

prima:

nista	Nominativo	Altezza	Peso
	Filtro	Filtro	Filtro
1	Djokovic Novak	1.89	77
2	Nadal Rafael	1.87	80
3	Federer Roger	1.85	79
4	Alcaraz	1.81	75
5	Medvedev	1.96	79
6	Zverer Alexander	2.02	85

dopo:

DELETE FROM Tennisti			
WHERE Nominativo = 'Zverer Alexander'			
idTennista	Nominativo	Altezza	Peso
	Filtro	Filtro	Filtro
1	Djokovic Novak	1.89	77
2	Nadal Rafael	1.87	80
3	Federer Roger	1.85	79
4	Alcaraz	1.81	75
5	Medvedev	1.96	79

**Comando SQL**  
**UPDATE**  
*(semplificato)*

**UPDATE** consente di **Modificare i Dati in uno o più Record** di una Tabella.

Anche qui è necessario specificare una **clausola WHERE** (già vista nel comando **SELECT**) per specificare quali record si desidera rimuovere.

```
UPDATE <nome-tabella>
SET <campo> = <valore>,
    <campo> = <valore>,
    ...
WHERE <condizione>
```

<nome-tabella> è il nome della Tabella in cui aggiungere il nuovo record.

<campo> = <valore> specifica che si desidera modificare <campo> e impostarlo con il nuovo <valore>. Si possono modificare più campi specificando più volte la dicitura <campo> = <valore> e separando le varie diciture con la virgola.

<condizione> è una espressione booleana sui dati del record: saranno modificati solo i record per cui questa condizione è vera (True).

Esempio:

```
UPDATE Tennisti
SET Altezza = 1.93,
    Peso = 93
WHERE idTennista = 2
```

idTennista	Nominativo	Altezza	Peso
Filtro	Filtro	Filtro	Filtro
1	Djokovic Novak	1.89	77
2	Nadal Rafael	1.93	93
3	Federer Roger	1.85	79
4	Alcaraz	1.81	75
5	Medvedev	1.96	79