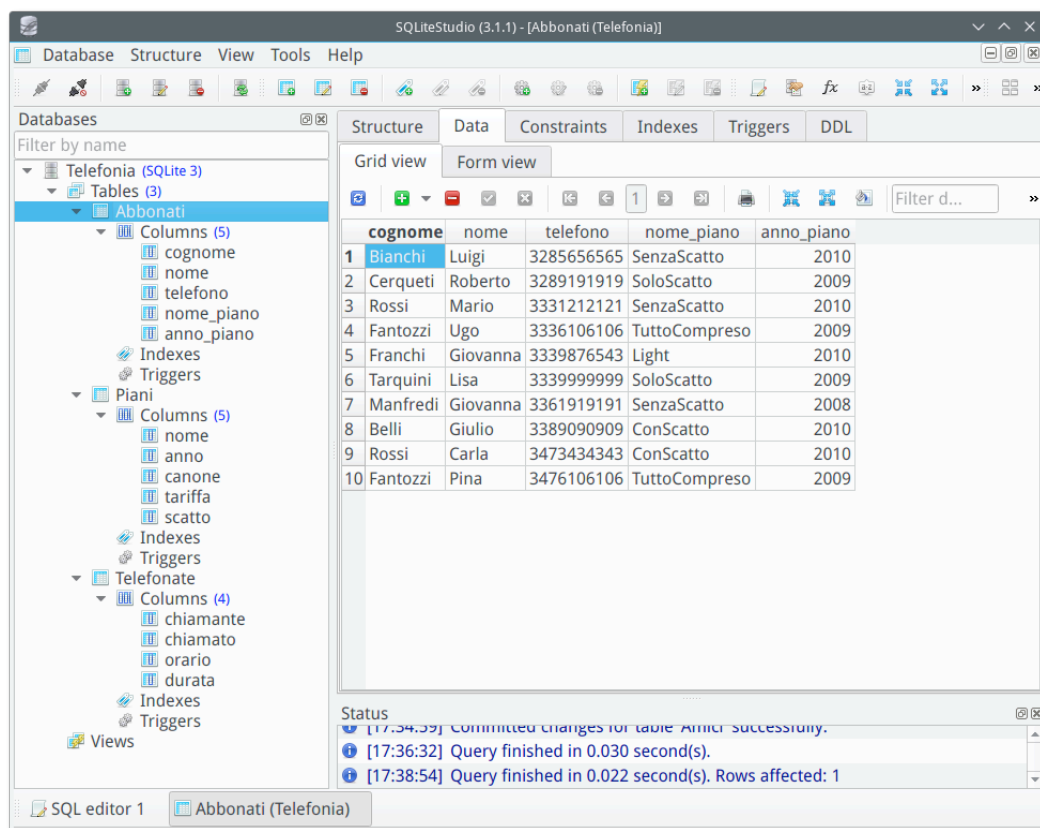


Fondamenti di Basi di Dati

Esercitazione 2 - Interrogazioni semplici con SQLiteStudio

Analizzare il database

Scarichiamo e apriamo il file **Telefonia.sqlite** con SQLiteStudio (ricordiamo che il file con il database deve essere prima salvato in una cartella, e poi aperto da SQLiteStudio tramite il comando **Database -> Add a database**, come descritto nella prima esercitazione). Analizziamo il database, esaminando gli attributi e i dati presenti nelle varie tabelle:



Sono presenti tre relazioni: **Abbonati**, **Piani** e **Telefonate**. Determinare lo schema delle relazioni completando lo schema riportato qui sotto (copiare i nomi degli attributi):

Abbonati	Piani	Telefonate
cognome	nome	chiamante
nome	anno	
telefono		
nome_piano		
anno_piano		

Osserviamo che la relazione **Abbonati** rappresenta gli abbonati di una compagnia telefonica; la relazione **Piani** rappresenta i piani tariffari disponibili. Ogni abbonato ha un piano tariffario. Quali sono le chiavi primarie delle relazioni? Indicarle graficamente sottolineando gli attributi che le costituiscono.

Interrogazioni

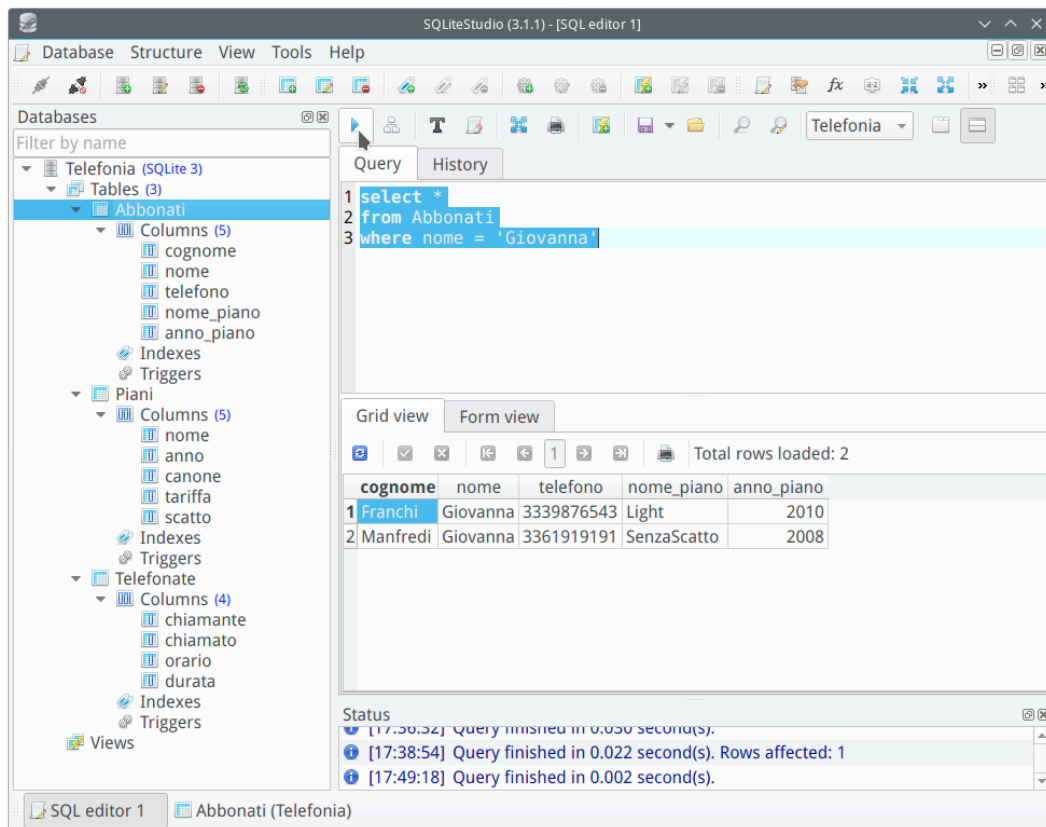
Dati degli abbonati di nome Giovanna

Cerchiamo tutti i dati degli abbonati di nome Giovanna. Prima ancora di scrivere la query, analizziamo visivamente i dati presenti nel database, per determinare il risultato atteso. In questo modo avremo la possibilità di verificare il buon funzionamento della query, una volta che l'avremo preparata. Apriamo dunque la tabella **Abbonati** e cerchiamo le righe relative agli abbonati di nome Giovanna, che sono le seguenti:

cognome	nome	telefono	nome_piano	anno_piano
Franchi	Giovanna	3339876543	Light	2010
Manfredi	Giovanna	3361919191	SenzaScatto	2008

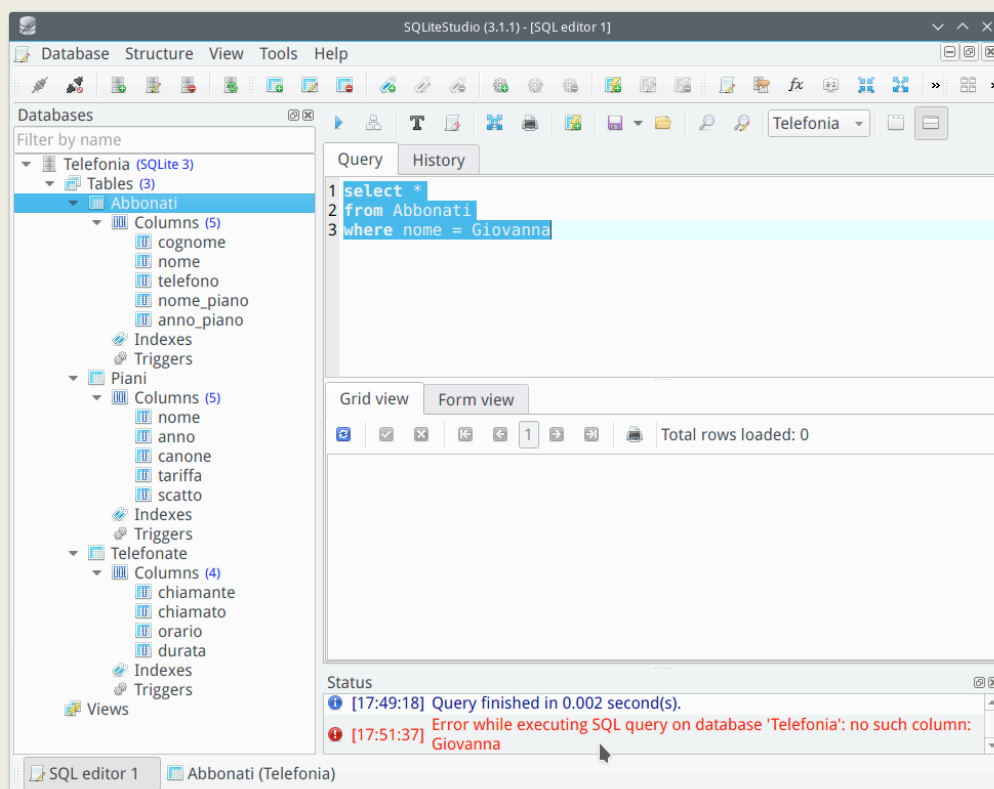
Questo è quindi il risultato che ci aspetteremo da una corretta esecuzione della query. Scriviamo ora la query ed eseguiamola nella scheda **Execute SQL**, avendo cura di verificare la correttezza del risultato:

```
select *
from Abbonati
where nome = 'Giovanna'
```



Nota: proviamo che cosa succede inserendo una query errata. Supponiamo di “dimenticare” gli apici che racchiudono la stringa **Giovanna**, ed inseriamo la seguente query:

```
select *
from Abbonati
where nome = Giovanna
```



Nell'area in basso contenente i messaggi di stato appare un messaggio di errore, in rosso. È importante leggere e interpretare il messaggio di errore, che in questo caso indica che non è presente una colonna di nome Giovanna. Mancando gli apici, SQLite ha considerato la parola **Giovanna** non un valore di tipo stringa, ma il nome di un attributo (al pari di **nome**)... ma un attributo di nome **Giovanna** non esiste nella nostra tabella **Abbonati**. Di qui l'errore.

Supponiamo invece che interessino soltanto il cognome e il numero di telefono degli abbonati di nome Giovanna. Modifica ed esegui la query in modo che restituisca soltanto tali dati, ovvero la seguente tabella:

cognome	telefono
Franchi	3339876543
Manfredi	3361919191

Abbonati il cui numero telefonico finisce per 6

Cerchiamo il nome e il numero di telefono degli abbonati il cui numero di telefono finisce con la cifra 6.

Ricordiamo che i numeri telefonici non sono numeri interi, ma stringhe (VARCHAR). Dunque abbiamo la possibilità di usare l'operatore **like** per cercare i “numeri” che finiscono con il simbolo 6.

Prova a scrivere la query, a eseguirla e a verificare che il risultato coincida con il risultato atteso (suggerimento: il risultato è composto da due tuple).

La soluzione è la seguente:

```
select nome, telefono
from Abbonati
where telefono like '%6'
```

Durata delle telefonate di gennaio

Determinare la durata delle telefonate effettuate a gennaio 2012 – e quindi dal 1 gennaio 2012 (incluso) al 1 febbraio 2012 (escluso).

Verificare che il risultato sia quello desiderato. Le tuple del risultato saranno le seguenti durate: {4, 8}.

Query con espressione booleana con parentesi

Cerca il cognome degli abbonati con un piano detto 'SenzaScatto' che non sia del 2010, oppure con un piano detto 'Light'.

La condizione da verificare è dunque:

- Il nome del piano è **SenzaScatto** e inoltre l'anno del piano è diverso da **2010**
- Oppure -
- Il nome del piano è **Light**

Componi il predicato di selezione combinando diversi sottopredicati con gli operatori booleani, inserendo opportunamente le parentesi. Esegui la query e verifica che il risultato coincida con quello atteso, cioè: {'Franchi', 'Manfredi'}.

Numeri di telefono da cui è partita o a cui è arrivata qualche telefonata

Cerchiamo i numeri di telefono da cui è partita o è arrivata qualche telefonata. Poiché tali informazioni si trovano in due attributi distinti della relazione **Telefonate**, ovvero **chiamante** e **chiamato**, non è possibile ottenere il risultato tramite un'unica query di tipo select.

Eseguiamo dunque due select: la prima per trovare i numeri da cui è partita qualche telefonata, la seconda per trovare i numeri a cui è arrivata qualche telefonata. Infine uniamo i risultati.

La query complessiva è:

```
select chiamante from Telefonate
union
select chiamato from Telefonate
```

Esegui la query e verifica che il risultato sia composto da 8 tuple.

Numeri di telefono da cui è partita una telefonata a marzo 2012, che non siano di Fantozzi

Cerca i numeri telefonici da cui è partita una telefonata nel mese di marzo 2012, che però non siano dell'abbonato Fantozzi.

Si tratta di cercare dalla relazione **Telefonate** i numeri chiamanti delle telefonate di marzo 2012 (A). Dalla relazione **Abbonati** invece cercheremo i numeri di telefono di abbonati con cognome Fantozzi (B). Infine da A togliamo insiemisticamente B con l'operatore **except**.

Verifica che il risultato sia il seguente: {'3361919191', '3331212121'}.

Cancellazione di tuple

Pina Fantozzi

Cancelliamo l'abbonato **Pina Fantozzi**. Prima di eliminare l'abbonato, scriviamo una query di tipo select per verificare la correttezza del predicato di selezione delle tuple da cancellare:

```
select *  
from Abbonati  
where nome='Pina' and cognome='Fantozzi'
```

Eseguiamo la query e verifichiamo che selezioni una sola tupla. A questo punto modifichiamo la query in modo da ottenere il comando di cancellazione desiderato:

```
delete from Abbonati  
where nome='Pina' and cognome='Fantozzi'
```

Eseguiamola e verifichiamo il buon esito della cancellazione.

Abbonati a piani del 2008

Cancellare tutti gli abbonati con un piano telefonico del 2008. Verificare quanto si sta per cancellare con una select.

Aggiornamento di tuple

Incremento della durata delle telefonate

Aumentiamo di un minuto la durata di tutte le telefonate, tramite il seguente comando:

```
update Telefonate  
set durata = durata + 1
```

Verificare il buon esito dell'aggiornamento e poi, con un opportuno comando, riportare le durate al loro valore originario.

Scambio di nome e cognome

Scrivere una query che, nella relazione **Abbonati**, scambi il cognome con il nome degli abbonati il cui cognome inizia con la lettera F. Ad esempio, dopo l'esecuzione del comando di aggiornamento "Fantozzi Ugo" sarà diventato "Ugo Fantozzi".

Verificare il buon esito dell'aggiornamento.

Suggerimento: per ogni tupla, in seguito all'aggiornamento, l'attributo cognome dovrà avere il valore che prima possedeva l'attributo nome; al contempo, l'attributo nome dovrà avere il valore che prima dell'aggiornamento possedeva l'attributo cognome.