

---

# **Esercitazione 2**

## **SQL 1**

# La famiglia

---

Base di dati contenete la seguente informazioni:

- Informazione sulle persone:
  - Nome
  - Sesso
  - Anno di nascita
  - Città
- Informazioni sui matrimoni:
  - Nomi dei coniugi
  - Data del matrimonio
- Informazioni sulla discendenza:
  - Relazione genitore-figlio
- Informazioni sui luoghi
  - Città
  - Stato di appartenenza
  - Periodo temporale

# **Schema della base di dati**

---

**Persone** (Nome, Sesso, Anno, Città)

**Discendenza** (Genitore, Figlio)

**Stato** (Città, Inizio, Fine, Stato)

**Matrimoni** (Marito, Moglie, Anno)

# Vincoli di integrità relazionale

- Tra Genitore in **Discendenza e Persone**
- Tra Figlio in **Discendenza e Persone**
- Tra Marito in **Matrimoni e Persone**
- Tra Moglie in **Matrimoni e Persone**

*Hanno senso valori nulli ?*

- Anno e Città in **Persone**
- Anno in **Matrimoni**

# Definizione della tabella Persona

---

```
create table Persone (  
    Nome          character(20) primary key,  
    Sesso         character(1) not null,  
    Città         character(20) ,  
    Anno          Numeric(4) ,  
    check (Sesso = 'M' OR sesso = 'F' )  
);
```

# Definizione della tabella Discendenza

```
create table Discendenza (  
    Genitore    character(20) ,  
    Figlio      character(20) ,  
    primary key (Genitore, Figlio) ,  
    foreign key(Genitore) references Persone(Nome) ,  
    foreign key(Figlio) references Persone(Nome)  
);
```

# Definizione della tabella Stato

---

```
create table Stato (  
    Città      character(20) ,  
    Inizio     numeric(4) ,  
    Fine       numeric(4) ,  
    Stato      character(30) ,  
    primary key (Città, Inizio, Fine)  
);
```

# Definizione della tabella Matrimoni

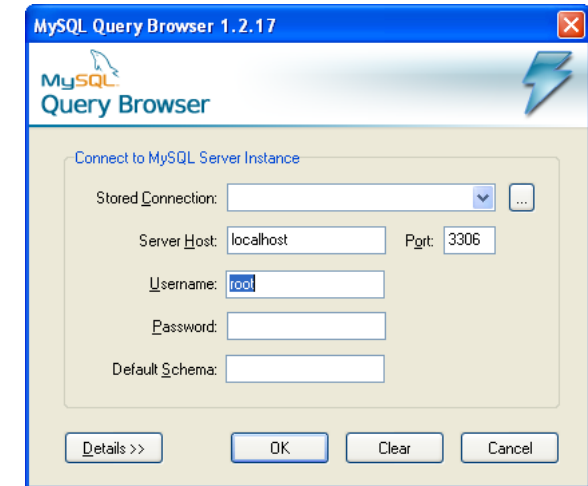
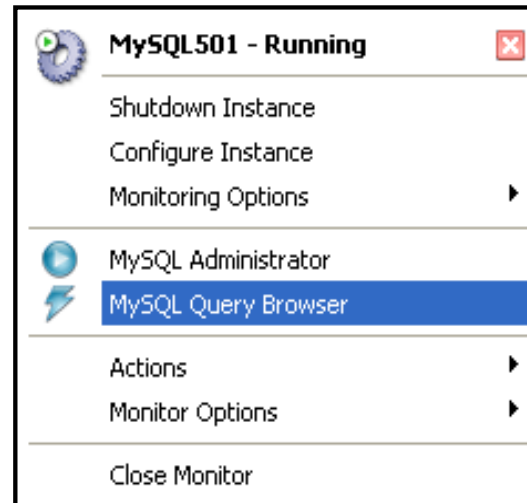
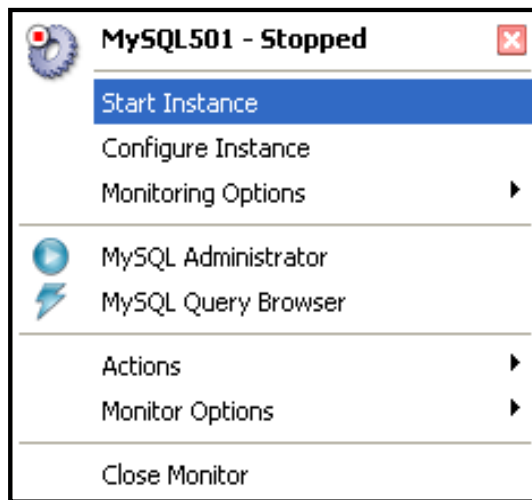
```
create table Matrimoni (  
    Marito    character(20) references Persone(Nome) ,  
    Moglie    character(20) references Persone(Nome) ,  
    Anno      numeric(4)  
    primary key (Marito, Moglie)  
);
```

# MySQL

---

- Effettuare i download dal sito:  
[www.mysql.org](http://www.mysql.org)
- Occorre installare:
  - DBMS server: [MySQL Community Server v. 5.1](#)
  - Le interfacce client del pacchetto: [MySQL GUI Tools](#)
- Una volta installato il sistema è costituito da diverse componenti:
  - Il DBMS server (che gira come servizio in Windows)
  - I due client con interfaccia grafica [MySQL Query Browser](#) e [MySQL Administrator](#)
  - La console di controllo [MySQL System Tray Monitor](#) che consente di controllare il server e di avviare i client

# Avvio



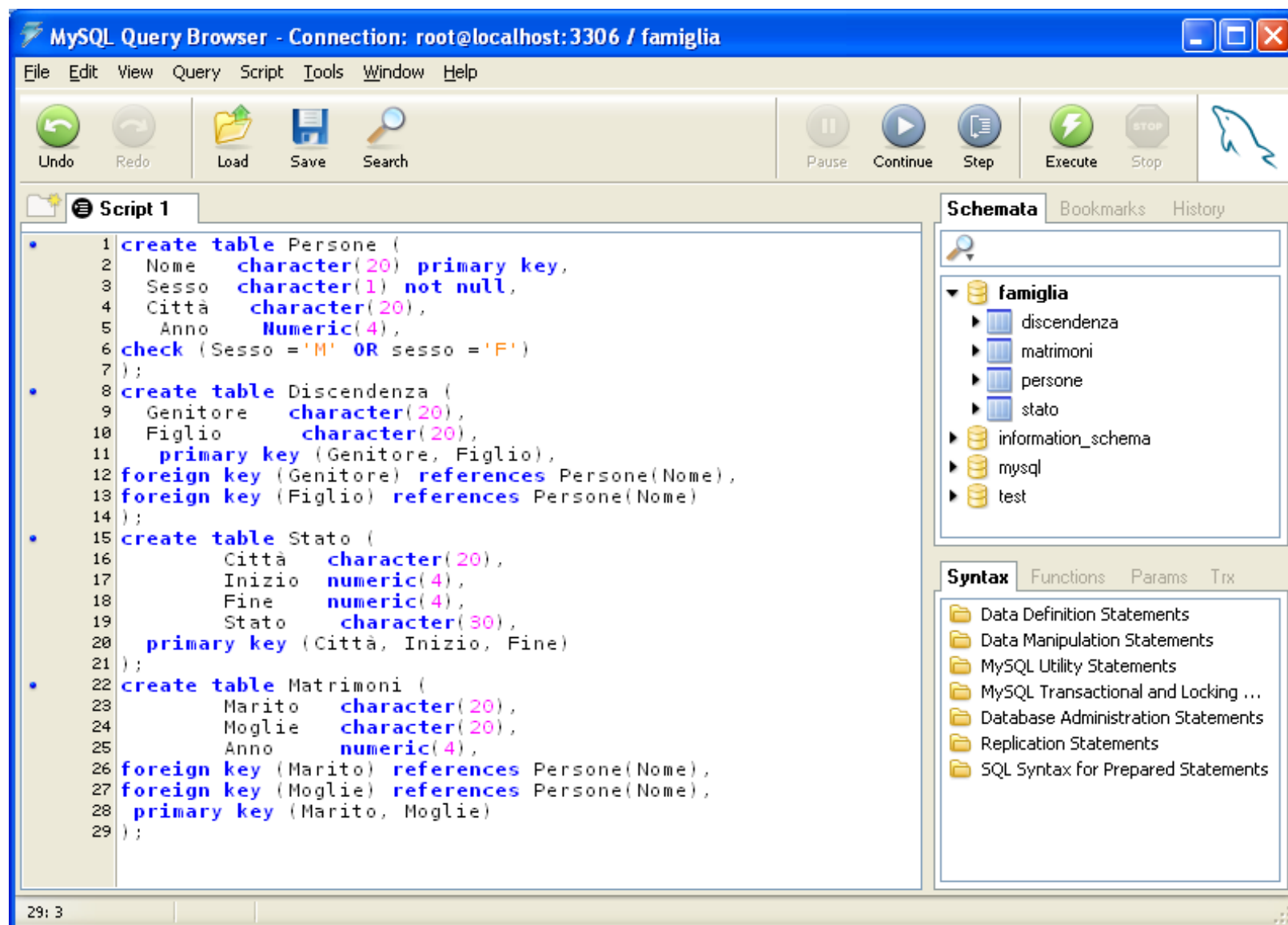
1. Lanciare **MySQL System Tray Monitor** dal menu di avvio
2. Attivare dal **Tray Monitor** il server MySQL
3. Attivare dal **Tray Monitor** l'interfaccia di interrogazione **MySQL Query Browser**
4. Effettuare il login come amministratore (root)

# Definizione dello schema

---

- Creare un nuovo schema: nel settore **Schemata** dal menu destro del mouse
- Selezionare il nuovo DB con doppio click del mouse
- Aprire un nuovo *Script Tab*
- Incollare scrivere o incollare nello Script Tab le definizioni SQL delle tabelle e dei relativi vincoli
- Avviare l'azione con il tasto execute
- Gli schemi delle tabelle vengono creati
- È possibile alterare le definizioni, sia graficamente tramite il comando **Edit Table** dal menu destro del mouse
- L'istanza del DB è a questo punto vuota: occorre popolarla

# Creazione dello schema tramite script

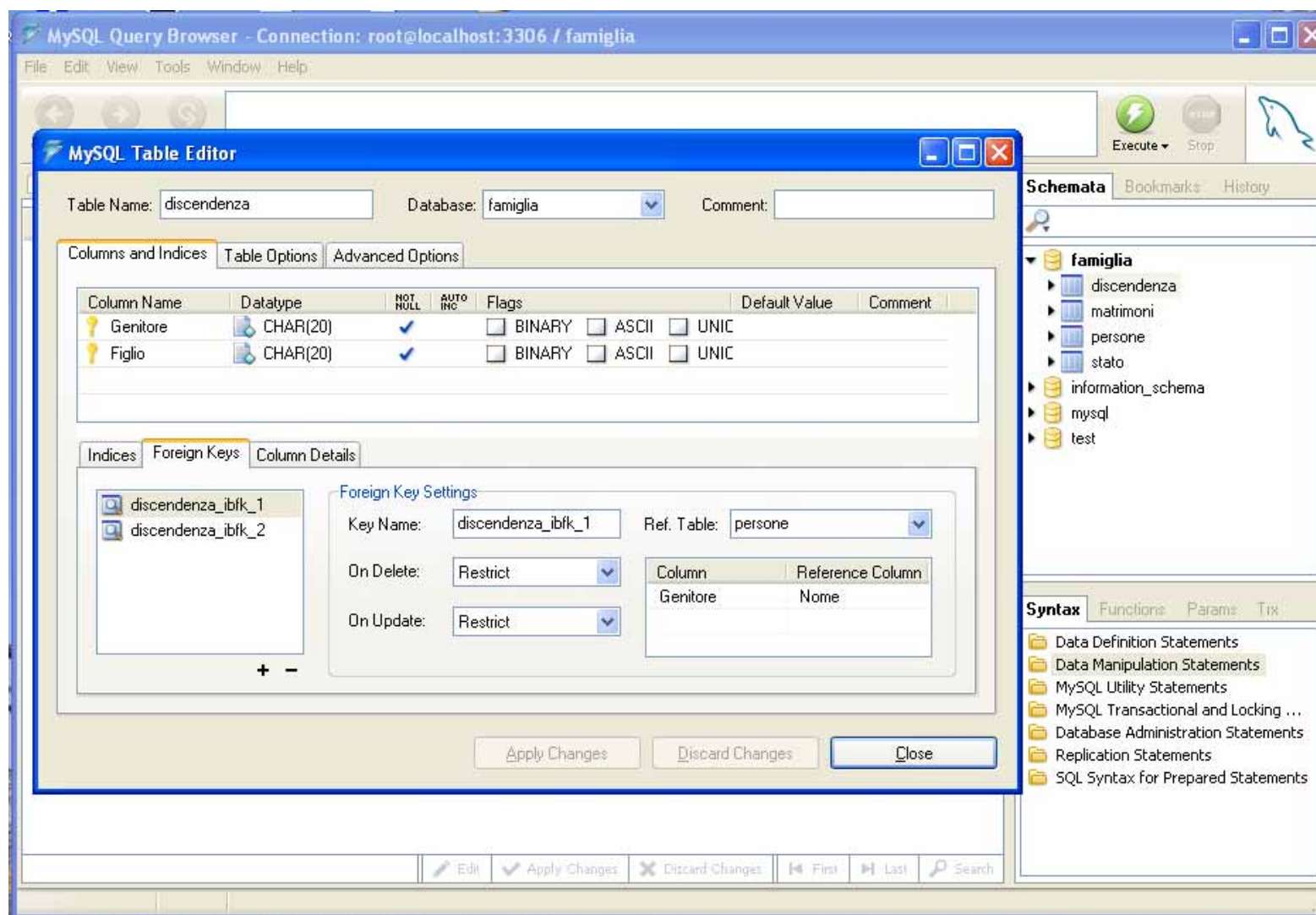


The screenshot shows the MySQL Query Browser interface. The main window displays a SQL script for creating a database schema. The script includes the following SQL statements:

```
1 create table Persone (  
2   Nome      character(20) primary key,  
3   Sesso    character(1) not null,  
4   Città    character(20),  
5   Anno     Numeric(4),  
6   check (Sesso = 'M' OR sesso = 'F')  
7 );  
8 create table Discendenza (  
9   Genitore  character(20),  
10  Figlio    character(20),  
11  primary key (Genitore, Figlio),  
12  foreign key (Genitore) references Persone(Nome),  
13  foreign key (Figlio) references Persone(Nome)  
14 );  
15 create table Stato (  
16   Città    character(20),  
17   Inizio   numeric(4),  
18   Fine     numeric(4),  
19   Stato    character(30),  
20   primary key (Città, Inizio, Fine)  
21 );  
22 create table Matrimoni (  
23   Marito    character(20),  
24   Moglie    character(20),  
25   Anno      numeric(4),  
26   foreign key (Marito) references Persone(Nome),  
27   foreign key (Moglie) references Persone(Nome),  
28   primary key (Marito, Moglie)  
29 );
```

The interface also shows a 'Schemata' panel on the right, which displays the database structure for the 'famiglia' database. The structure includes tables: discendenza, matrimoni, persone, and stato. Other databases listed are information\_schema, mysql, and test.

# Consultare lo schema delle tabelle



Comando **Edit Table** dal menu destro del mouse sul nome della tabella

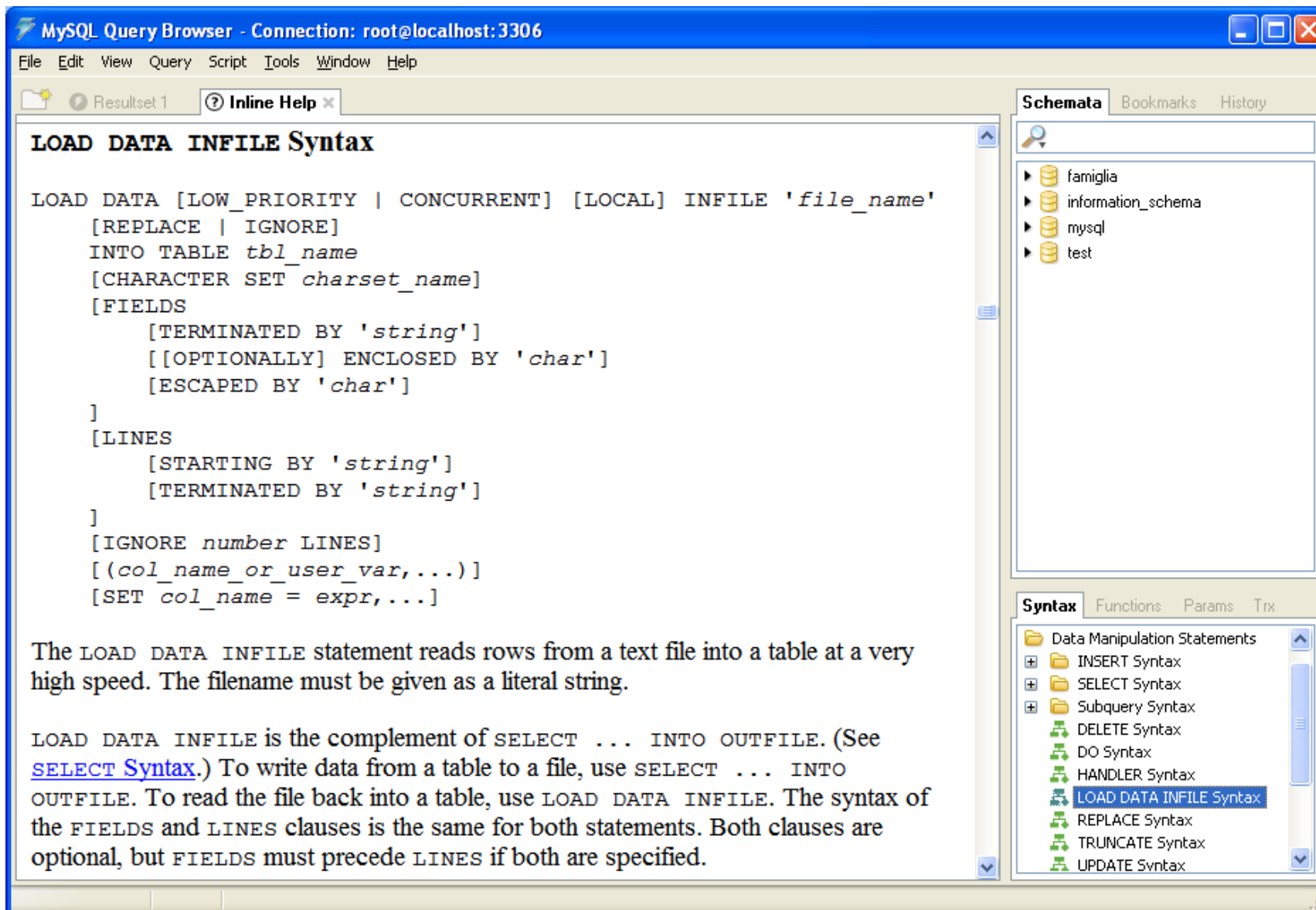
# Popolazione della base di dati

---

- Due possibilità:
  - Inserire le tuple singolarmente
  - Importare i valori da file
- Comando SQL:

```
load data infile '<nomefile con path>'
into table <nometabella>
fields terminated by '<separatore campi>'
lines terminated by '<carattere fine-linea>';
```
- Carichiamo ciascuna tabella con un comando singolo, cominciando da **Persone** (altre tabelle hanno vincoli verso di essa)

# Consultare la sintassi online



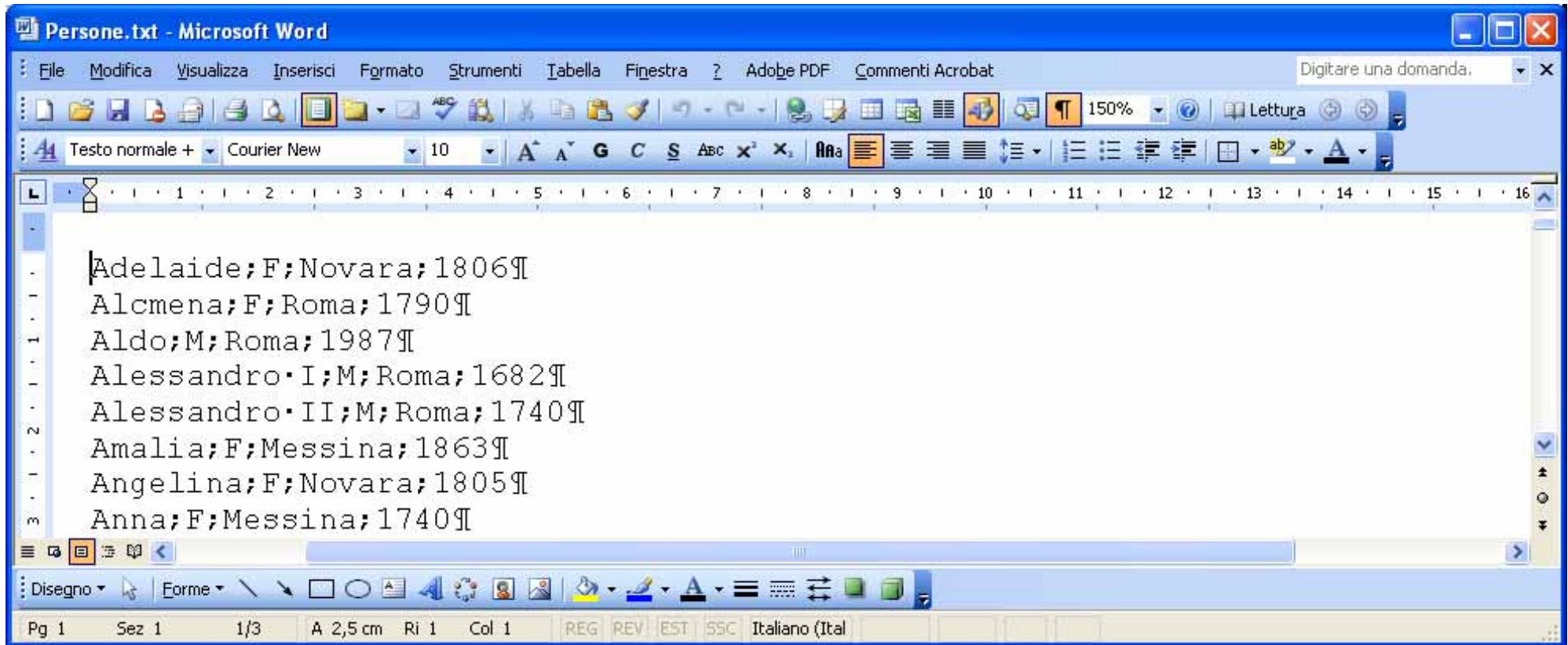
The screenshot shows the MySQL Query Browser interface. The main window displays the syntax for the `LOAD DATA INFILE` statement. The syntax is as follows:

```
LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE 'file_name'
  [REPLACE | IGNORE]
  INTO TABLE tbl_name
  [CHARACTER SET charset_name]
  [FIELDS
    [TERMINATED BY 'string']
    [[OPTIONALLY] ENCLOSED BY 'char']
    [ESCAPED BY 'char']
  ]
  [LINES
    [STARTING BY 'string']
    [TERMINATED BY 'string']
  ]
  [IGNORE number LINES]
  [(col_name_or_user_var,...)]
  [SET col_name = expr,...]
```

The text below the syntax explains that the `LOAD DATA INFILE` statement reads rows from a text file into a table at a very high speed. It notes that the filename must be given as a literal string. It also states that `LOAD DATA INFILE` is the complement of `SELECT ... INTO OUTFILE`. To write data from a table to a file, use `SELECT ... INTO OUTFILE`. To read the file back into a table, use `LOAD DATA INFILE`. The syntax of the `FIELDS` and `LINES` clauses is the same for both statements. Both clauses are optional, but `FIELDS` must precede `LINES` if both are specified.

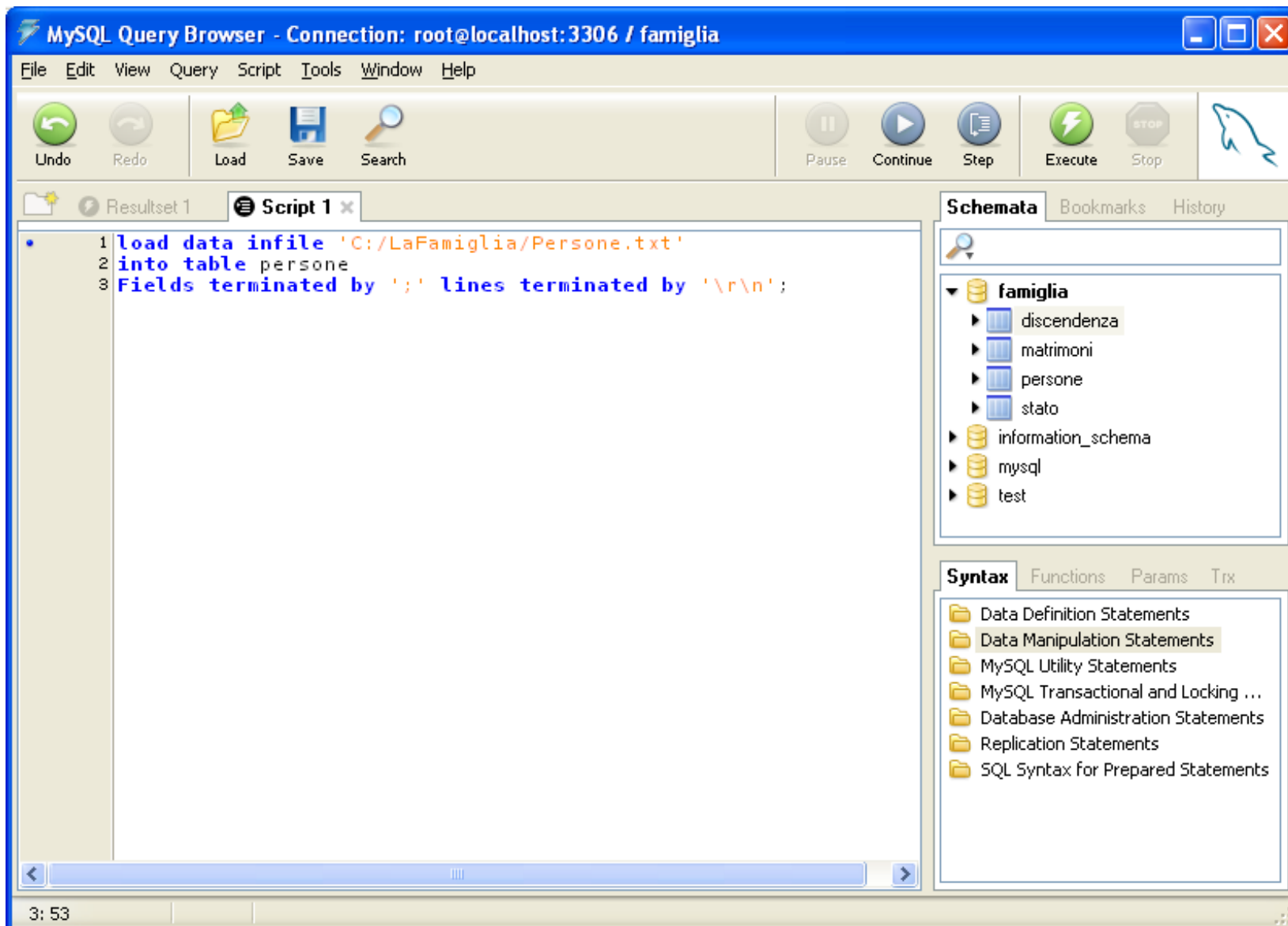
The interface also shows a 'Schemata' panel on the right with a tree view of databases: famiglia, information\_schema, mysql, and test. Below that is a 'Syntax' panel with a list of SQL syntax topics, where 'LOAD DATA INFILE Syntax' is highlighted.

# Il file contenente i dati



- Ciascuna riga contiene i valori degli attributi di una tupla (nell'ordine di definizione)
- Valori separati dal carattere ';' e linee separate da '\r\n'

# Popolazione della tabella persone



# Ispezionare il contenuto

The screenshot shows the MySQL Query Browser interface. The title bar reads "MySQL Query Browser - Connection: root@localhost:3306". The menu bar includes File, Edit, View, Query, Script, Tools, Window, and Help. The query editor contains the SQL statement: `SELECT * FROM famiglia.discendenza d;`. Below the query editor are buttons for "Go back", "Next", "Refresh", "Execute", and "Stop". The main area displays "Resultset 2" with a table of data. The table has two columns: "Genitore" and "Figlio". The data is as follows:

Genitore	Figlio
Francesca Bo.	Adelaide
Giuseppe B. Bo.	Adelaide
Camillo	Alcmena
Lucio II	Aldo
Maria D.	Aldo
Clemente	Alessandro I
Diego I	Antonino I
Giovanna S. C.	Antonino I
Anna	Antonino II
Diego II	Antonino II
Alcmena	Antonino III
Diego III	Antonino III
Carlotta	Antonino IV
Diego IV	Antonino IV
Adelaide	Benedetto Ca.
Carlo Ca.	Benedetto Ca.
Felice	Bernardino
Giuseppe Bt.	Bernardino
Alessandro II	Camillo
Felice	Candido
Giuseppe Bt.	Candido
Felice	Carlo B.
Giuseppe Bt.	Carlo B.

At the bottom of the window, it states "151 rows fetched in 0,0237s (0,0011s)". The status bar includes buttons for "Edit", "Apply Changes", "Discard Changes", "First", "Last", and "Search". On the right side, there is a "Schemata" panel showing a tree view of databases: "famiglia" (with sub-databases "discendenza", "matrimoni", "persone", "stato"), "information\_schema", "mysql", and "test". Below the schemata panel is a "Syntax" panel with a tree view of SQL statement categories: "Data Definition Statements", "Data Manipulation Statements", "MySQL Utility Statements", "MySQL Transactional and Locking ...", "Database Administration Statements", "Replication Statements", and "SQL Syntax for Prepared Statements".

# Esempi 1

---

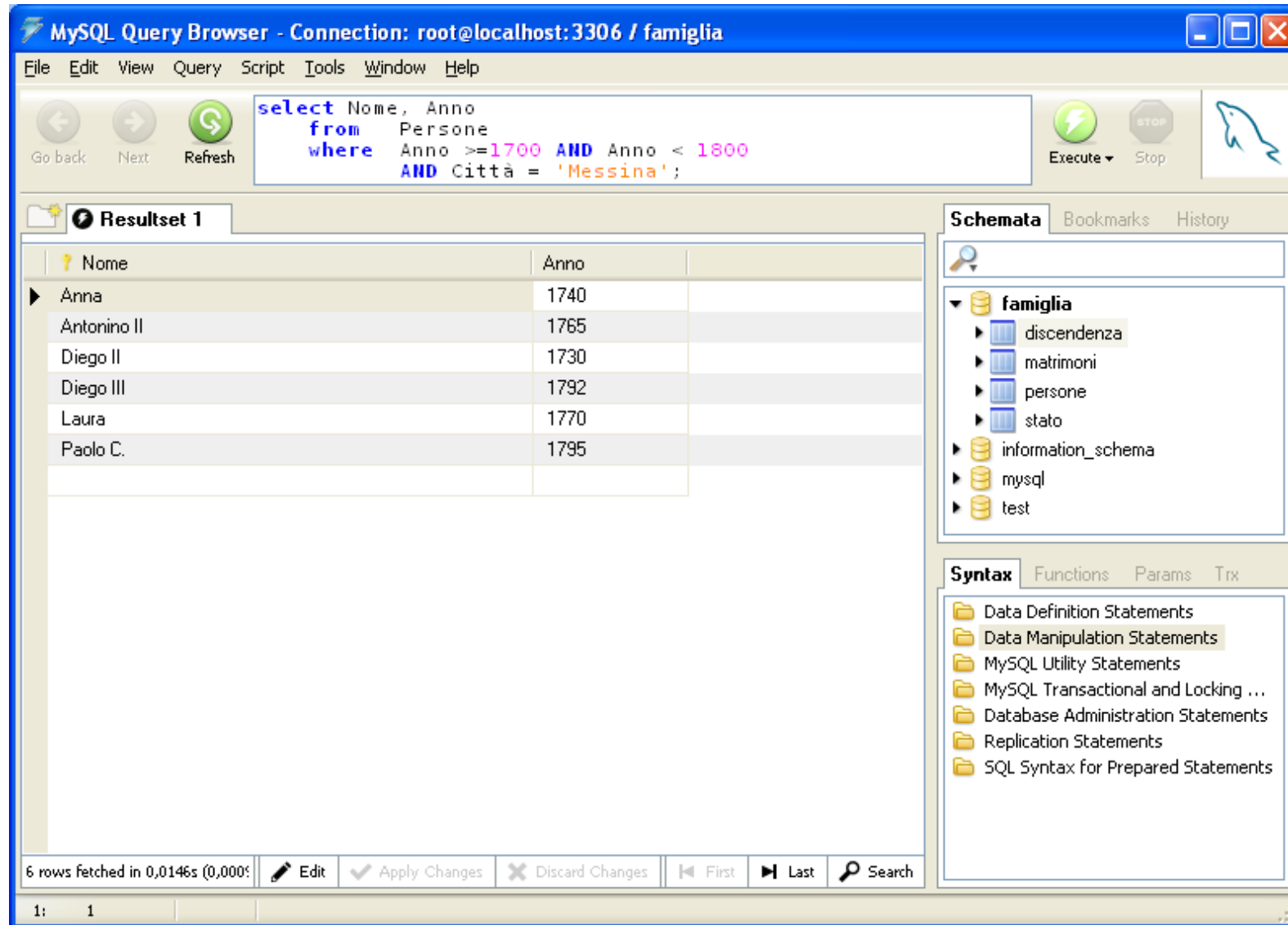
**Query 1:** *I nomi e gli anni di nascita di tutte le persone nate a Messina durante il '700*

```
select Nome, Anno
from Persone
where Anno >=1700 AND Anno < 1800
      AND Città = 'Messina'
```

**Query 2:** *Le città in cui sono nate Persone durante il '700*

```
select distinct Città
from Persone
where Anno >=1700 AND Anno < 1800
order by Città
```

# Esempio 1 su MySQL



The screenshot shows the MySQL Query Browser interface. The title bar indicates the connection is 'root@localhost:3306 / famiglia'. The menu bar includes File, Edit, View, Query, Script, Tools, Window, and Help. The query editor contains the following SQL query:

```
select Nome, Anno
from Persone
where Anno >=1700 AND Anno < 1800
AND Città = 'Messina';
```

The results are displayed in a table with two columns: 'Nome' and 'Anno'. The data is as follows:

Nome	Anno
Anna	1740
Antonino II	1765
Diego II	1730
Diego III	1792
Laura	1770
Paolo C.	1795

The interface also shows a 'Schemata' panel on the right, listing databases: famiglia (expanded to show tables: discendenza, matrimoni, persone, stato), information\_schema, mysql, and test. A 'Syntax' panel is also visible, listing various SQL statement categories.

At the bottom, the status bar indicates '6 rows fetched in 0,0146s (0,000%)' and provides options for Edit, Apply Changes, Discard Changes, First, Last, and Search.

# Esempi 2

---

**Query 3:** *Il nomi dei genitori di tutte le persone di sesso femminile nate a Roma dopo il 1950*

```
select Genitore, Figlio, Anno
from Discendenza D, Persone P
where D.Figlio = P.Nome
      AND P.Città = 'Roma'
      AND P.Anno >= 1950
      AND P.Sesso = 'F'
order by Genitore
```

# Esempi 3

---

**Query 4:** *Tutte le persone nate a Novara durante il Regno di Sardegna*

```
Select  Nome, Anno
From    Persone, Stato
where   Persone.Città = Stato.Città
        AND Anno >= Inizio AND Anno <= Fine
        AND Stato = 'Regno di Sardegna'
        AND Persone.Città = 'Novara'
order  by Anno
```

# Esempi 4

---

**Query 5:** *Il nomi dei padri di tutte le persone nate a Roma dopo il 1950*

```
select Genitore as Padre, Figlio, F.Anno
from   Discendenza D, Persone F, Persone P
where  D.Figlio = F.Nome
        AND D.Genitore = P.Nome
        AND F.Città = 'Roma'
        AND F.Anno >= 1950
        AND P.Sesso = 'M'
order by Genitore
```

# Esempi 5

---

**Query 6:** *I nonni di Eugenia II e le rispettive città e anni di nascita*

```
select N.Genitore as Nonno, Città, Anno
from   Discendenza G, Discendenza N,
       Persone P
where  G.Figlio = 'Eugenia II'
       AND G.Genitore = N.Figlio
       AND N.Genitore = Nome
order by N.Genitore
```

# Esempi 6

---

*Query 7: Nome del padre, della madre e del figlio, di tutti i nati a Roma dopo il 1950, ordinati per nascita del figlio*

```
select P.Genitore as Padre,  
       M.Genitore as Madre, P.Figlio  
from   Discendenza P, Discendenza M,  
       Persone PP, Persone PM, Persone PF  
where  M.Figlio = PF.Nome  
       AND P.Figlio = PF.Nome  
       AND P.Genitore = PP.Nome AND PP.Sesso = 'M'  
       AND M.Genitore = PM.Nome AND PM.Sesso = 'F'  
       AND PF.Città='Roma' AND PF.Anno >= 1950  
order by PF.Anno
```