

# **Basi di Dati**

MySQL & MySQL Workbench

Anna Monreale

# MySQL

- MySQL è un DBMS gratuito ed open-source
  - (<http://www.mysql.com>)
- MySQL può essere utilizzato in Windows, Linux, MacOS X.
- MySQL è un **RDBMS** (relational database management system) ovvero un DBMS basato sul modello relazione
- Una Relazione è la struttura di una tabella (entità) che è definita da **records** (righe di dati) e **attributi** (colonne della tabella di dati)

ID	Nome	Cognome	Sesso	Data Nascita
1	Mario	Rossi	M	15/01/1988
2	Agata	Coletta	F	12/12/1977
3	Giulio	Coro	M	10/10/1980

# MySQL

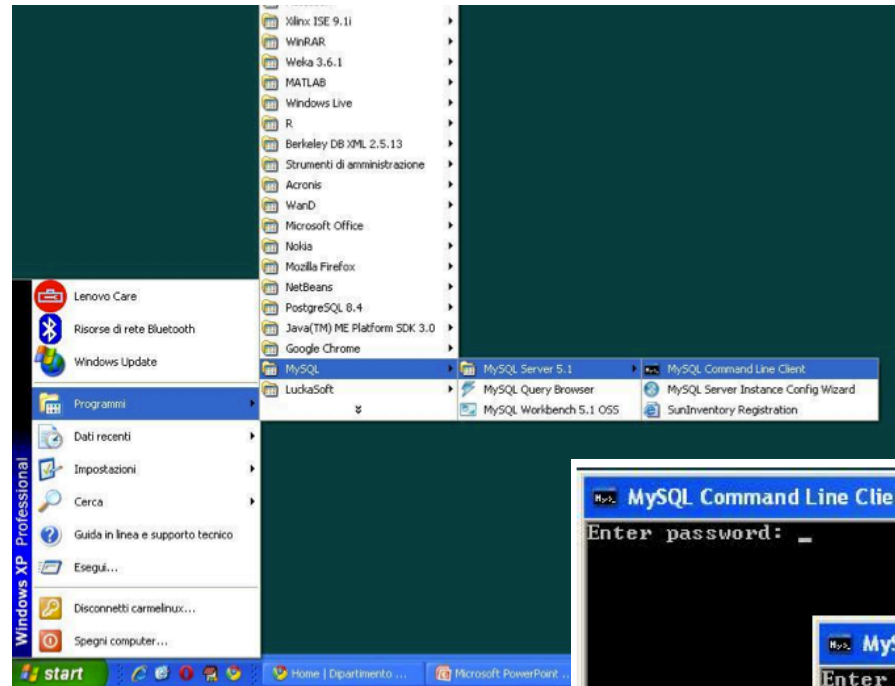
- MySQL è composta da un client e un server
- **MySQL Community Server** è scaricabile liberamente  
<http://dev.mysql.com/downloads/>
- Interazione con il server può avvenire a riga di comando oppure tramite interfaccia grafica
  - **MySQL Workbench:**  
<http://www.mysql.com/products/workbench/>
  - Permette la modellazione e la progettazione di un database
  - Permette l'interrogazione del database

# Download e Installazione di MySQL

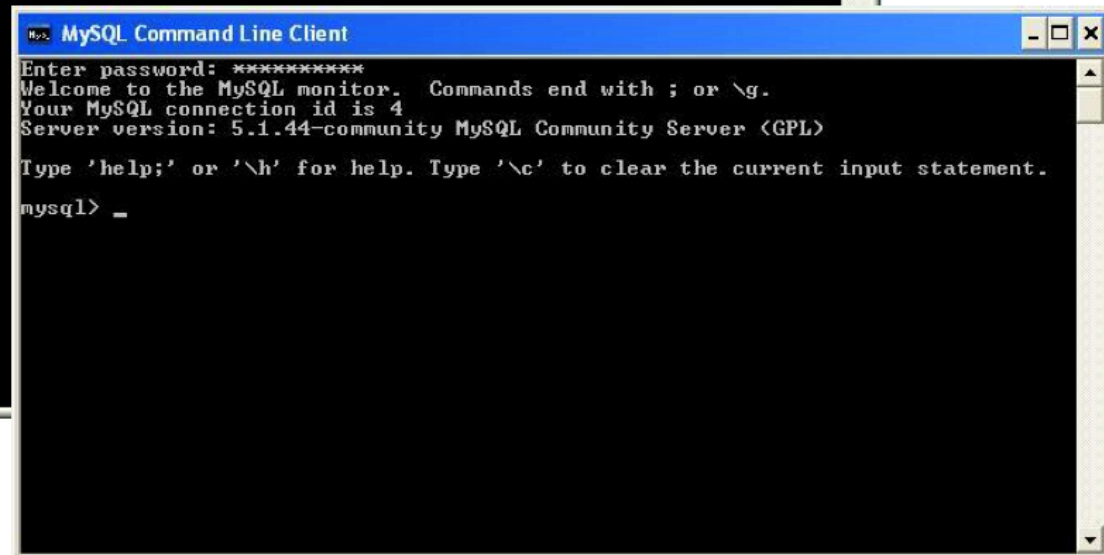
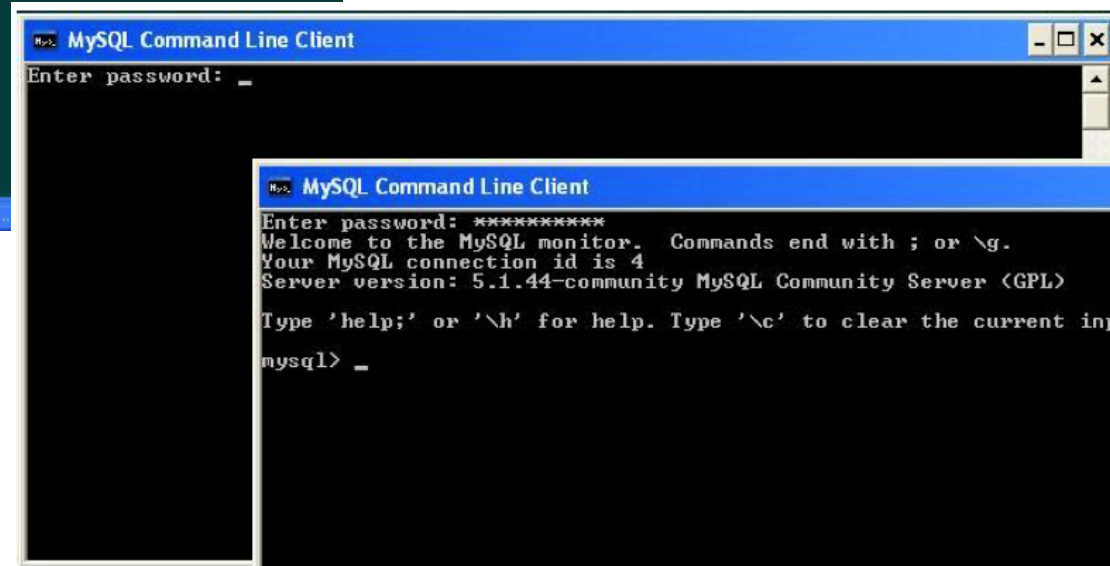
- Download e installa MySQL Community Server
- **Windows:** scaricare l'Installer e lanciarlo. Durante l'installazione sarà necessario settare una password per *“l'utente root”*
- **Mac OSX:** scaricare l'installer .dmg e seguire l'installazione guidata.
- **Linux Ubuntu e altre distribuzioni Debian:** è necessario eseguire il comando `“apt-get install mysql”` da shell con i privilegi di root

# MySQL a linea di comando: Windows

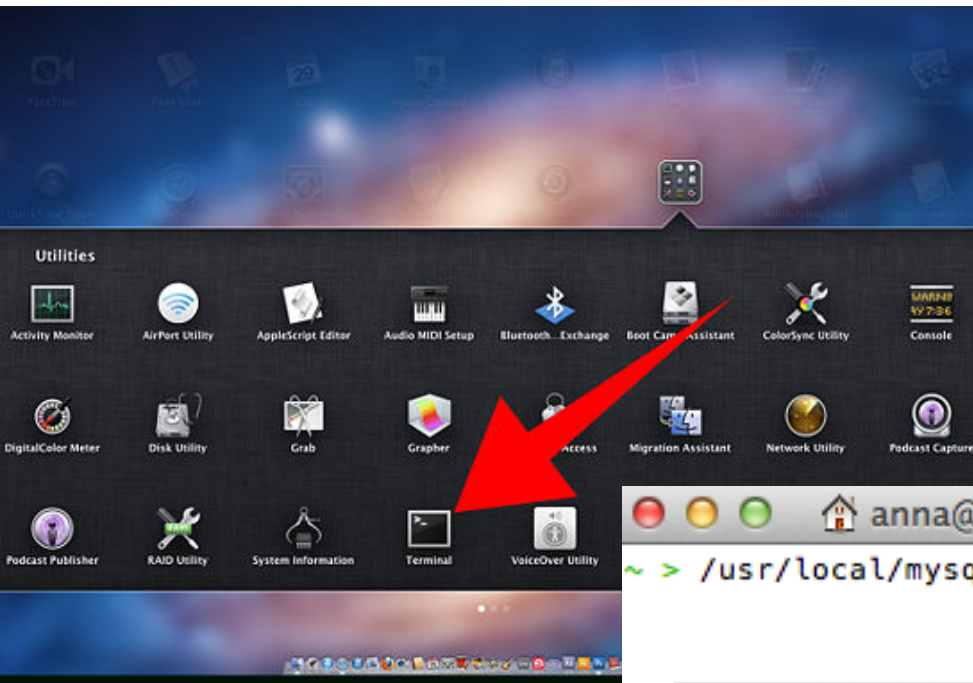
- Inserire la password che è settata al momento dell'installazione



Windows



# MySQL a linea di comando: Mac OSX



MAC OS X

- Digitare: `/usr/local/mysql/bin/mysql -u root -p`
- Inserire la password che è settata al momento dell'installazione

```
anna@scorpion /Users/anna — bash — 66x19
~ > /usr/local/mysql/bin/mysql -u root -p
```

```
anna@scorpion /Users/anna — mysql — 66x19
Last login: Mon Mar 17 17:41:00 on ttys000
Kernel Information: Darwin 11.4.2 x86_64
GNU bash, version 3.2.48(1)-release (x86_64-apple-darwin11)
Copyright (C) 2007 Free Software Foundation, Inc.
Uptime: 23:57 up 1 day, 9 mins, 2 users, load averages: 1.74 1.85 1.87
Server time is: Mon Mar 17 23:57:22 CET 2014
~ > /usr/local/mysql/bin/mysql -u root -p
Enter password:
```

# Comandi per gestire un DB

- help (lista dei comandi MySQL)
- SHOW DATABASES
- CREATE DATABASE
- DROP DATABASE
- USE

**N.B. all commands except “help” must be followed by “;”**

# SHOW DATABASES

```
mysql> show databases;
```

```
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| test |  
+-----+  
3 rows in set (0.05 sec)
```

```
mysql> use test;  
Database changed
```



# CREATE DATABASE

```
mysql> CREATE DATABASE esempio;
```

```
Query OK, 1 row effected (0.06 sec)
```

```
mysql> show databases;
```

```
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| test |
| esempio |
+-----+
4 rows in set (0.05 sec)
```

# Tipi di dati in MySQL: numerici

- **“Integer” data type:**
  - TINYINT-128 > 127
  - SMALLINT-32768 > 32767
  - MEDIUMINT-8388608 > 8388608
  - INT-2147483648 > 2147483647
  - BIGINT-9223372036854775808 > 9223372036854775807
- **“Not-Integer” data type:**
  - Float
  - Double
  - Decimal

# Tipi di dati in MySQL: stringa

- **CHAR(n)255 byte <--- lunghezza fissa**
- **VARCHAR(n)255 byte <--- lunghezza variabile**
- TINYTEXT255 byte
- TINYBLOB255 byte
- TEXT65535 byte
- BLOB65535 byte
- MEDIUMTEXT1.6 Mb
- MEDIUMBLOB1.6 Mb
- LONGTEXT4.2 Mb
- LONGBLOB4.2 Mb

# Tipo “Data”

- DATETIME: AAAA-MM-GG HH:MM:SS
- DATE: AAAA-MM-GG
- TIME: HH:MM:SS
- YEAR:AAAA
- TIMESTAMP(14):AAAAMMGGHHMMSS
- TIMESTAMP(12):AAMMGGHHMMSS
- TIMESTAMP(10):AAMMGGHHMM
- TIMESTAMP(8):AAAAMMGG
- TIMESTAMP(4):AAMM
- TIMESTAMP(2):AA

# Base di dati

- Base di dati è composta da insieme di tabelle
- Tabella è una collezione di ennuple
- Ennupla è un insieme di coppie (attributo, valore)

# MySQL Workbench

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree is expanded to show a database named 'FirstExample' with several tables, including 'Studenti'. A context menu is open over the 'Studenti' table, listing various actions such as 'Select Rows - Limit 1000', 'Copy to Clipboard', 'Create Table...', and 'Drop Table...'. A blue callout box labeled 'Databases' has a red arrow pointing to the 'FirstExample' database in the tree. Another blue callout box labeled 'Spazio dove scrivere le queries' has a red arrow pointing to the central SQL editor area. The bottom status bar indicates '1 objects dropped'.

**Databases**

**Spazio dove scrivere le queries**

Action	Response	Duration / Fetch Time
DROP DATABASE `Es...	OK	0.000 sec
DROP DATABASE `a...	OK	0.000 sec

1 objects dropped

# Esempio

## ▸ **Professori**

- codice, nome, qualifica, facoltà
- numeri di telefono

## ▸ **Studenti**

- matricola, nome, tipo di corso (ciclo): laurea tr., laurea spec.
- relatore della tesi

## ◆ **Corsi**

- codice, titolo, docente, ciclo

## ◆ **Esami**

- studente, voto, lode, corso

## ◆ **Tutorato Studentesco**

- studente tutore, studente tutorato

# Tabelle

```
TABLE Professori (  
  cod char(4) PRIMARY KEY,  
  cognome varchar(20) NOT NULL,  
  nome varchar(20) NOT NULL,  
  qualifica char(15),  
  facolta char(10) );
```

```
TABLE Studenti (  
  matr integer PRIMARY KEY,  
  cognome varchar(20) NOT NULL,  
  nome varchar(20) NOT NULL,  
  ciclo char(20),  
  anno integer,  
  relatore char(4)  
  REFERENCES Professori(cod)
```

```
);
```

```
TABLE Corsi (  
  cod char(3) PRIMARY KEY,  
  titolo varchar(20) NOT NULL,  
  ciclo char(20),  
  docente char(4)  
  REFERENCES Professori(cod)
```

```
);
```

```
TABLE Tutorato (  
  studente integer  
  REFERENCES Studenti(matr),  
  tutor integer  
  REFERENCES Studenti(matr),  
  PRIMARY KEY (studente,tutor));
```

```
TABLE Esami (  
  studente integer  
  REFERENCES Studenti(matr)  
  ON DELETE cascade  
  ON UPDATE cascade,  
  corso char(3)  
  REFERENCES Corsi(cod),  
  voto integer,  
  lode bool,  
  CHECK (voto>=18 and voto<=30),  
  CHECK (not lode or voto=30),  
  PRIMARY KEY (studente, corso));
```

```
TABLE Numeri (  
  professore char(4)  
  REFERENCES Professori(cod),  
  numero char(9),  
  PRIMARY KEY (professore,numero));
```



# Creiamo il Database e le Tabelle (1)

Untitled - MySQL Workbench

MySQL Model\* x EER Diagram x

CorsiInfo x SQL File 3\* x

```
1 CREATE database CorsiInfo;
2 USE CorsiInfo;
3
4 CREATE TABLE Professori (
5     cod char(4) PRIMARY KEY,
6     cognome varchar(20) NOT NULL,
7     nome varchar(20) NOT NULL,
8     qualifica char(15),
9     facolta char(10)
10 );
11 CREATE TABLE Studenti (
12     matr integer PRIMARY KEY,
13     cognome varchar(20) NOT NULL,
14     nome varchar(20) NOT NULL,
15     ciclo char(20),
16     anno integer,
17     relatore char(4),
18     FOREIGN KEY(relatore) REFERENCES Professori(cod)
19 );
20
21 CREATE TABLE Corsi (
22     cod char(3) PRIMARY KEY,
23     titolo varchar(20) NOT NULL,
24     ciclo char(20),
25     docente char(4),
26     FOREIGN KEY(docente) REFERENCES Professori(cod)
27 );
28
29 CREATE TABLE Tutorato (
30     studente integer,
31     tutor integer,
32     FOREIGN KEY(studente) REFERENCES Studenti(matr),
33     FOREIGN KEY(tutor) REFERENCES Studenti(matr),
34     PRIMARY KEY (studente,tutor)
35 );
36
```

Context Help Snippets

CREATE TABLE

**Topic: CREATE TABLE**  
Syntax:  
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl\_name (create\_definition,...) [table\_options] [partition\_options]

Or:  
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl\_name [(create\_definition,...)] [table\_options] [partition\_options] select\_statement

Or:  
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl\_name { LIKE old\_tbl\_name | (LIKE old\_tbl\_name) }

create\_definition:  
col\_name  
column\_definition  
| [CONSTRAINT [symbol]] PRIMARY KEY [index\_type] (index\_col\_name,...) [index\_option] ...  
| {INDEX|KEY} [index\_name] [index\_type] (index\_col\_name,...) [index\_option] ...  
| [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY] [index\_name] [index\_type] (index\_col\_name,...) [index\_option] ...  
| {FULLTEXT|SPATIAL} [INDEX|KEY] [index\_name] (index\_col\_name,...)

# Creiamo il Database e le Tabelle (1)

Untitled - MySQL Workbench

MySQL Model\* x EER Diagram x

```
1 CREATE database CorsiInfo;
2 USE CorsiInfo;
3
4 CREATE TABLE Professori (
5     cod char(4) PRIMARY KEY,
6     cognome varchar(20) NOT NULL,
7     nome varchar(20) NOT NULL,
8     qualifica char(15),
9     facolta char(10)
10 );
11 CREATE TABLE Studenti (
12     matr integer PRIMARY KEY,
13     cognome varchar(20) NOT NULL,
14     nome varchar(20) NOT NULL,
15     ciclo char(20),
16     anno integer,
17     relatore char(4),
18     FOREIGN KEY(relatore) REFERENCES Professori(cod)
19 );
20
21 CREATE TABLE Corsi (
22     cod char(3) PRIMARY KEY,
23     titolo varchar(20) NOT NULL,
24     ciclo char(20),
25     docente char(4),
26     FOREIGN KEY(docente) REFERENCES Professori(cod)
27 );
28
29 CREATE TABLE Tutorato (
30     studente integer,
31     tutor integer,
32     FOREIGN KEY(studente) REFERENCES Studenti(matr),
33     FOREIGN KEY(tutor) REFERENCES Studenti(matr),
34     PRIMARY KEY (studente,tutor)
35 );
```

vincolo di chiave

Context Help Snippets

CREATE TABLE

Topic: CREATE TABLE

Syntax:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name (create_definition,...) [table_options] [partition_options]
```

Or:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name [(create_definition,...)] [table_options] [partition_options] select_statement
```

Or:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name { LIKE old_tbl_name | (LIKE old_tbl_name) }
```

create\_definition:

```
col_name
column_definition
| [CONSTRAINT [symbol]]
PRIMARY KEY [index_type] (index_col_name,...)
[index_option] ...
| {INDEX|KEY} [index_name] [index_type] (index_col_name,...)
[index_option] ...
| [CONSTRAINT [symbol]]
UNIQUE [INDEX|KEY] [index_name] [index_type] (index_col_name,...)
[index_option] ...
| {FULLTEXT|SPATIAL} [INDEX|KEY] [index_name] (index_col_name,...)
```

# Creiamo il Database e le Tabelle (1)

Untitled - MySQL Workbench

MySQL Model\* x EER Diagram x

CorsiInfo x SQL File 3\* x

```
1 CREATE database CorsiInfo;
2 USE CorsiInfo;
3
4 CREATE TABLE Professori (
5     cod char(4) PRIMARY KEY,
6     cognome varchar(20) NOT NULL,
7     nome varchar(20) NOT NULL,
8     qualifica char(15),
9     facolta char(10)
10 );
11 CREATE TABLE Studenti (
12     matr integer PRIMARY KEY,
13     cognome varchar(20) NOT NULL,
14     nome varchar(20) NOT NULL,
15     ciclo char(20),
16     anno integer,
17     relatore char(4),
18     FOREIGN KEY(relatore) REFERENCES Professori(cod)
19 );
20
21 CREATE TABLE Corsi (
22     cod char(3) PRIMARY KEY,
23     titolo varchar(20) NOT NULL,
24     ciclo char(20),
25     docente char(4),
26     FOREIGN KEY(docente) REFERENCES Professori(cod)
27 );
28
29 CREATE TABLE Tutorato (
30     studente integer,
31     tutor integer,
32     FOREIGN KEY(studente) REFERENCES Studenti(matr),
33     FOREIGN KEY(tutor) REFERENCES Studenti(matr),
34     PRIMARY KEY (studente,tutor)
35 );
36
```

**vincolo di riferimento**

**vincolo di riferimento**

Context Help Snippets

CREATE TABLE

**Topic: CREATE TABLE**  
Syntax:  
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl\_name (create\_definition,...) [table\_options] [partition\_options]

Or:  
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl\_name [(create\_definition,...)] [table\_options] [partition\_options] select\_statement

Or:  
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl\_name { LIKE old\_tbl\_name | (LIKE old\_tbl\_name) }

create\_definition:  
col\_name  
column\_definition  
| [CONSTRAINT [symbol]] PRIMARY KEY [index\_type] (index\_col\_name,...) [index\_option] ...  
| {INDEX|KEY} [index\_name] [index\_type] (index\_col\_name,...) [index\_option] ...  
| [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY] [index\_name] [index\_type] (index\_col\_name,...) [index\_option] ...  
| {FULLTEXT|SPATIAL} [INDEX|KEY] [index\_name] (index\_col\_name,...)

# Creiamo il Database e le Tabelle (2)

The screenshot shows the MySQL Workbench interface. The main editor window displays the following SQL code:

```
1 CREATE TABLE Esami (  
2     studente integer,  
3     corso char(3),  
4     voto integer,  
5     lode bool,  
6     FOREIGN KEY(corso) REFERENCES Corsi(cod),  
7     FOREIGN KEY(srudente) REFERENCES Studenti(matr),  
8     CHECK (voto>=18 and voto<=30),  
9     CHECK (not lode or voto=30),  
10    PRIMARY KEY (studente, corso)  
11 );  
12  
13  
14 CREATE TABLE Numeri (  
15     professore char(4),  
16     numero char(9),  
17     PRIMARY KEY (professore,numero),  
18     FOREIGN KEY(professore) REFERENCES Professori(cod)  
19 );  
20
```

The right-hand pane shows the 'Context Help' for the 'CREATE TABLE' command. The 'Topic' is 'CREATE TABLE'. The 'Syntax' section shows the general syntax for creating a table, including options for temporary tables, if not exists, table options, and partition options. The 'Or:' section shows alternative syntaxes for creating a table, including the use of 'LIKE' or 'AS' to create a table based on an existing one. The 'create\_definition:' section lists the components of a column definition, such as column name, data type, constraints, and indexes.

# Creiamo il Database e le Tabelle (2)

Untitled - MySQL Workbench

MySQL Model\* x EER Diagram x

\* CorsilInfo x SQL File 3\* x

```
1
2 CREATE TABLE Esami (
3     studente integer,
4     corso char(3),
5     voto integer,
6     lode bool,
7     FOREIGN KEY(corso) REFERENCES Corsi(cod),
8     FOREIGN KEY(studente) REFERENCES Studenti(matr),
9     CHECK (voto>=18 and voto<=30),
10    CHECK (not lode or voto=30),
11    PRIMARY KEY (studente, corso);
12
13
14 CREATE TABLE Numeri (
15     professore char(4),
16     numero char(9),
17     PRIMARY KEY (professore, numero),
18     FOREIGN KEY(professore) REFERENCES Professori(cod)
19 );
20
```

Context Help Snippets

CREATE TABLE

**Topic: CREATE TABLE**  
Syntax:  
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl\_name (create\_definition,...) [table\_options] [partition\_options]

Or:  
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl\_name [(create\_definition,...)] [table\_options] [partition\_options] select\_statement

Or:  
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl\_name { LIKE old\_tbl\_name | (LIKE old\_tbl\_name) }

create\_definition:  
col\_name  
column\_definition  
| [CONSTRAINT [symbol]]  
PRIMARY KEY [index\_type] (index\_col\_name,...) [index\_option] ...  
| {INDEX|KEY} [index\_name] [index\_type] (index\_col\_name,...) [index\_option] ...  
| [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY] [index\_name] [index\_type] (index\_col\_name,...) [index\_option] ...  
| {FULLTEXT|SPATIAL}

Vincoli di ennupla: predicati sui valori delle ennupe

# Creiamo il Database e le Tabelle (2)

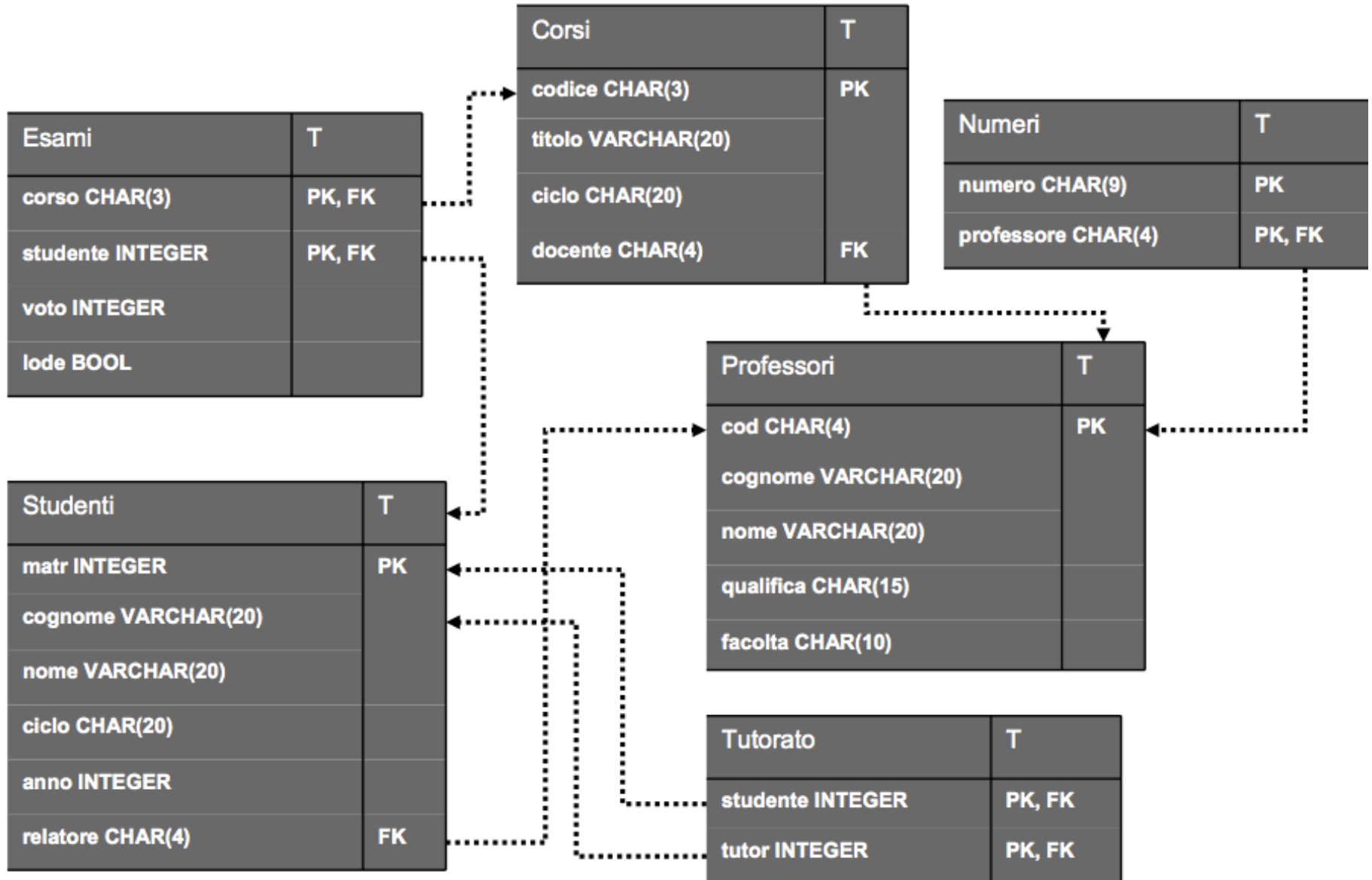
The screenshot shows the MySQL Workbench interface. The main window displays a SQL script for creating two tables: 'Esami' and 'Numeri'. The 'Esami' table has columns for 'studente' (integer), 'corso' (char(3)), 'voto' (integer), and 'lode' (bool). It includes foreign keys to 'Corsi(cod)' and 'Studenti(matr)', and check constraints for 'voto' (between 18 and 30) and 'lode' (not both 30). The 'Numeri' table has columns for 'professore' (char(4)) and 'numero' (char(9)), with a primary key on both and a foreign key to 'Professori(cod)'. The left sidebar shows the 'Schemas' tree with 'CorsInfo' expanded to show tables: Corsi, Esami, Numeri, Professori, Studenti, and Tutorato. A red arrow points from the 'Esami' table in the schema tree to the 'Esami' table definition in the SQL script. Another red arrow points from the 'Execute' button in the SQL editor toolbar to a text box. A third red arrow points from the 'Execute' button to another text box.

```
1
2 CREATE TABLE Esami (
3     studente integer,
4     corso char(3),
5     voto integer,
6     lode bool,
7     FOREIGN KEY(corso) REFERENCES Corsi(cod),
8     FOREIGN KEY(srudente) REFERENCES Studenti(matr),
9     CHECK (voto>=18 and voto<=30),
10    CHECK (not lode or voto=30),
11    PRIMARY KEY (studente, corso)
12 );
13
14 CREATE TABLE Numeri (
15     professore char(4),
16     numero char(9),
17     PRIMARY KEY (professore,numero),
18     FOREIGN KEY(professore) REFERENCES Professori(cod)
19 );
20
```

**Esequi lo script o la parte selezionata**

**Database con Tabelle**

# Vediamo lo Schema (1)



# Vediamo lo Schema (2)

The screenshot shows the MySQL Workbench interface. The 'File' menu is open, and the 'Import' submenu is visible. A red arrow points to the 'Reverse Engineer MySQL Create Script...' option. The interface includes a menu bar, a toolbar, a search bar, and several panels: 'Description', 'User Types', 'History', 'Views', 'Routines', 'Routine Groups', 'Schema Privileges', 'SQL Scripts', 'Model Notes', and 'Templates'.

**File** Edit View Arrange Model Database Tools Scripting Help

Untitled - MySQL Workbench

localhost x

Description

No Selection

Templates

timestamps  
create\_time, update\_time

user  
username, email, password, cre...

category  
category\_id, name

**Views** (0 items)  
Add View

**Routines** (0 items)  
Add Routine

**Routine Groups** (0 items)  
Add Group

► **Schema Privileges**

► **SQL Scripts**

► **Model Notes**

Type	Definition	Flags
BOOL	TINYINT(1)	
BOOLEAN	TINYINT(1)	
FIXED	DECIMAL(10,0)	
FLOAT4	FLOAT	
FLOAT8	DOUBLE	
INT1	TINYINT(4)	
INT2	SMALLINT(6)	
INT3	MEDIUMINT(9)	
INT4	INT(11)	
INT8	BIGINT(20)	
INTEGER	INT(11)	
LONG VARB...	MEDIUMBLOB	
LONG VARC...	MEDIUMTEXT	
LONG	MEDIUMTEXT	
MIDDLEINT	MEDIUMINT(9)	
NUMERIC	DECIMAL(10,0)	
DEC	DECIMAL(10,0)	
CHARACTER	CHAR(1)	

New document.



### Input and Options

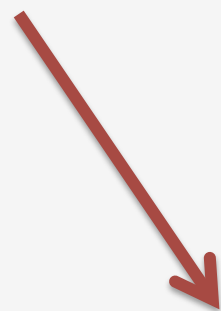
- Input and Options
- Reverse Engineer
- Results

Select the script containing the schemata to reverse engineer

Select SQL script file:

File encoding:

Place imported objects on a diagram



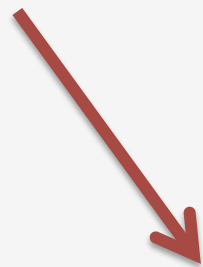
### SQL Import Finished Successfully

- Input and Options
- Reverse Engineer
- Results

Import of SQL script file '/Users/anna/Desktop/BDD/Esercizio/CorsiInfo.sql' has finished.

6 tables, 0 views and 0 stored procedures were imported in 1 schemas.

There were warnings during the import.  
Go Back to the previous page to review the logs.





**Navigator** Info

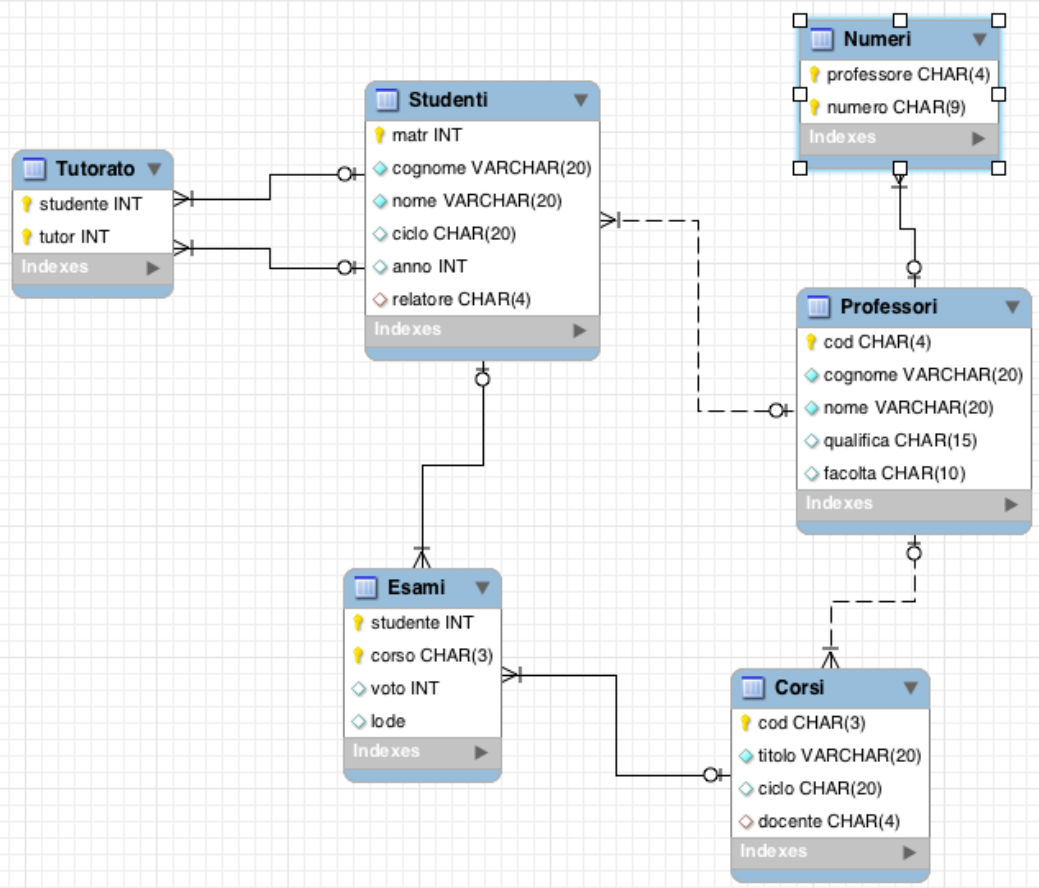
100

**Catalog** Layers User Types

- mydb
  - Tables
  - Views
  - Routine Groups
- CorsiInfo
  - Tables
  - Views
  - Routine Groups

Description Properties History

Numeri: Table

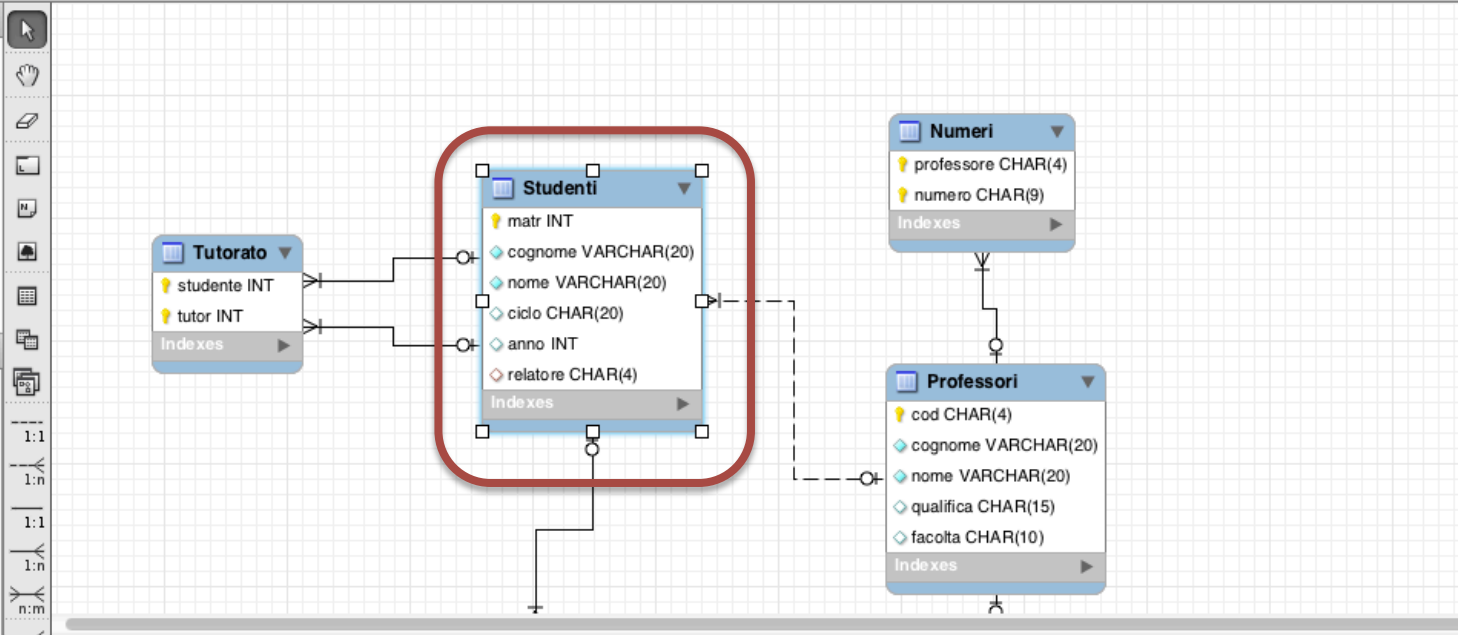


Navigator Info

100

Catalog Layers User Types

- mydb
  - Tables
  - Views
  - Routine Groups
- CorsiInfo
  - Tables
  - Views
  - Routine Groups



Studenti - Table

Name:  Schema: CorsiInfo

Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
matr	INT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
cognome	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
nome	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
ciclo	CHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
anno	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
relatore	CHAR(4)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
<click to edit>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column details 'matr'

Collation:

Comments:

Columns Indexes Foreign Keys Triggers Partitioning Options Inserts Privileges

# Inserimento istanze nelle tabelle

Professori

<u>cod</u>	cognome	nome	qualifica	facolta
FT	Totti	Francesco	ordinario	Ingegneria
CV	Vieri	Christian	associato	Scienze
ADP	Del Piero	Alessandro	supplente	null

Studenti

<u>matr</u>	cognome	nome	ciclo	anno	relatore
111	Rossi	Mario	laurea tr.	1	null
222	Neri	Paolo	laurea tr.	2	null
333	Rossi	Maria	laurea tr.	1	null
444	Pinco	Palla	laurea tr.	3	FT
77777	Bruno	Pasquale	laurea sp.	1	FT
88888	Pinco	Pietro	laurea sp.	1	CV

Corsi

<u>cod</u>	titolo	ciclo	docente
PR1	Programmazione I	laurea tr.	FT
ASD	Algoritmi e Str. Dati	laurea tr.	CV
INFT	Informatica Teorica	laurea sp.	ADP

# Inserimento istanze nelle tabelle

Tutorato

<u>studente</u>	<u>tutor</u>
111	77777
222	77777
333	88888
444	88888

Numeri

<u>professore</u>	<u>numero</u>
FT	0971205145
FT	347123456
VC	0971205227
ADP	0971205363
ADP	338123456

Esami

<u>studente</u>	<u>corso</u>	<u>voto</u>	<u>lode</u>
111	PR1	27	false
222	ASD	30	true
111	INFT	24	false
77777	PR1	21	false
77777	ASD	20	false
88888	ASD	28	false
88888	PR1	30	false
88888	INFT	30	true

# INSERT INTO

- Per inserire dei dati in una tabella si utilizza l'istruzione:

```
INSERT INTO nome_tabella [(nome_attributo1,nome_attributo2,...)]  
VALUES (valore1,valore2,...);
```

- L'ordinamento degli attributi (se presente) e dei valori è significativo.
- Le due liste di attributi e di valori devono avere lo stesso numero di elementi.
- Se la lista di attributi è omessa, si fa riferimento a tutti gli attributi della relazione secondo l'ordine con cui sono stati definiti.
- Se la lista di attributi non contiene tutti gli attributi della relazione, per gli altri viene inserito un valore nullo (che deve essere permesso) o un valore di default.

# Inserimento istanze nelle tabelle

Untitled - MySQL Workbench

localhost x MySQL Model\* x EER Diagram x

NT  
Status  
Connections  
and Privileges  
and System Variables  
Export  
Import/Restore  
Stop / Shutdown  
Logs  
Tools File

CorsiInfo\* x CorsilInfo x

```
1 INSERT INTO PROFESSORI (cod, cognome, nome, qualifica, facolta)
2 VALUES('FT', 'Totti', 'Francesco', 'ordinario', 'Ingegneria');
3
4 INSERT INTO PROFESSORI (cod, cognome, nome, qualifica, facolta)
5 VALUES('CV', 'Veri', 'Christian', 'associato', 'Scienze');
6
7 INSERT INTO PROFESSORI (cod, cognome, nome, qualifica, facolta)
8 VALUES('ADP', 'Del Piero', 'Alessandro', 'supplente', null);
9
10
```

100% 1:10

Action Output

	Time	Action	Response	Duration / Fetch Time
✓	6 22:47:34	CREATE TABLE Corsi ( cod char(3) PRIM...	0 row(s) affected	0.238 sec
✓	7 22:47:34	CREATE TABLE Tutorato ( studente int...	0 row(s) affected	0.327 sec
✓	8 22:47:34	CREATE TABLE Esami ( studente integ...	0 row(s) affected	0.277 sec
✓	9 22:47:35	CREATE TABLE Numeri ( professore char...	0 row(s) affected	0.207 sec
✗	10 22:59:04	Apply changes to CorsilInfo	Selected name conflicts with existing schema `Corsi...	
✗	11 22:59:12	Apply changes to CorsilInfo	Selected name conflicts with existing schema `Corsi...	
✗	12 23:00:46	Apply changes to prova		
✗	13 23:04:43	show schema CorsilInfo	Error Code: 1064. You have an error in your SQL syn...	0.000 sec

Context Help

INSERT

Topic: INSERT  
Syntax:  
INSERT [LOW\_PRIORITY] [DELAYED] [HIGH\_PRIORITY] [IGNORE] [INTO] tbl\_name [PARTITION (partition\_name [(col\_name, {VALUES | VALUE} (DEFAULT),...)] [ON DUPLICATE KEY UPDATE col\_name=expr])] col\_name=expr

Or:  
INSERT [LOW\_PRIORITY] [DELAYED] [HIGH\_PRIORITY] [IGNORE] [INTO] tbl\_name [PARTITION (partition\_name [(col\_name, {VALUES | VALUE} (DEFAULT),...)] [ON DUPLICATE KEY UPDATE col\_name=expr])] col\_name=expr

Or:  
INSERT [LOW\_PRIORITY] [HIGH\_PRIORITY] [INTO] tbl\_name [PARTITION (partition\_name [(col\_name, {VALUES | VALUE} (DEFAULT),...)] [ON DUPLICATE KEY UPDATE col\_name=expr])] col\_name=expr

Columns  
cod  
cognome  
nome

CorsiInfo Session

ssori



# Aggiungere un vincolo di chiave esterna

- Soluzione: Eseguo le seguenti istruzioni in quest'ordine:

1- Creo la tabella “**Persona**” SENZA vincoli di foreign key;

```
CREATE TABLE Persona(  
CodFiscale CHARACTER(20) primary key,  
LuogoNascita CHARACTER(1));
```

# Aggiungere un vincolo di chiave esterna

- Soluzione: Eseguo le seguenti istruzioni in quest'ordine:

1- Creo la tabella “**Persona**” SENZA vincoli di foreign key;

2- Creo la tabella “**Città**” CON i vincoli di foreign key verso la tabella “**Persona**” (che ora esiste);

```
CREATE TABLE Citta(  
  Nome CHARACTER(20) primary key,  
  Sindaco CHARACTER(20),  
  FOREIGN KEY (Sindaco) REFERENCES  
  Persona(CodFiscale));
```

# Aggiungere un vincolo di chiave esterna

- Soluzione: Eseguo le seguenti istruzioni in quest'ordine:
  - 1- Creo la tabella "**Persona**" SENZA vincoli di foreign key;
  - 2- Creo la tabella "**Città**" CON i vincoli di foreign key verso la tabella "**Persona**" (che ora esiste);
  - 3- Aggiungo il vincolo di foreign key alla tabella "**Persona**" verso la tabella "**Città**" (che ora esiste).

```
ALTER TABLE Persona  
ADD CONSTRAINT fk_citta  
FOREIGN KEY (LuogoNascita) REFERENCES Citta(Nome);
```

# Eliminazione di dati dalle tabelle

- ES: Eliminiamo tutte le persone con meno di 18 anni dalla tabella "Persone"

```
DELETE FROM persone WHERE eta<18;
```