

Riprendiamo a parlare di mysql considerando un esercizio e mettendo in pratica quanto visto nel file precedente.

Vogliamo creare un nuovo DB secondo le seguenti specifiche definite nel modello logico:

Corsi (CodiceCorso, Descrizione, NumeroOre, DataInizio, DataFine, Importo)

Utenti (Matricola, Nome, Cognome, Indirizzo, Città, Telefono)

Iscrizioni (CodiceCorso, Matricola, DataIscrizione)

Il db si compone di 3 tabelle, *Corsi* e *Utenti* sono collegate attraverso la tabella *Iscrizioni*.

Iniziamo con la creazione del database chiamato **Istruzione**.

Per prima cosa avviamo XAMPP e i server Apache e Mysql in esso contenuti

quindi stabiliamo la connessione da riga di comando seguendo il percorso che ci porta alla cartella in cui è contenuto **mysql.exe** (in windows potrebbe essere in C:\Programmi\XAMPP\mysql\bin).

Digitare **mysql -u root -p**

creiamo il DB con il comando CREATE

```
mysql> create database istruzione;
Query OK, 1 row affected (0.33 sec)

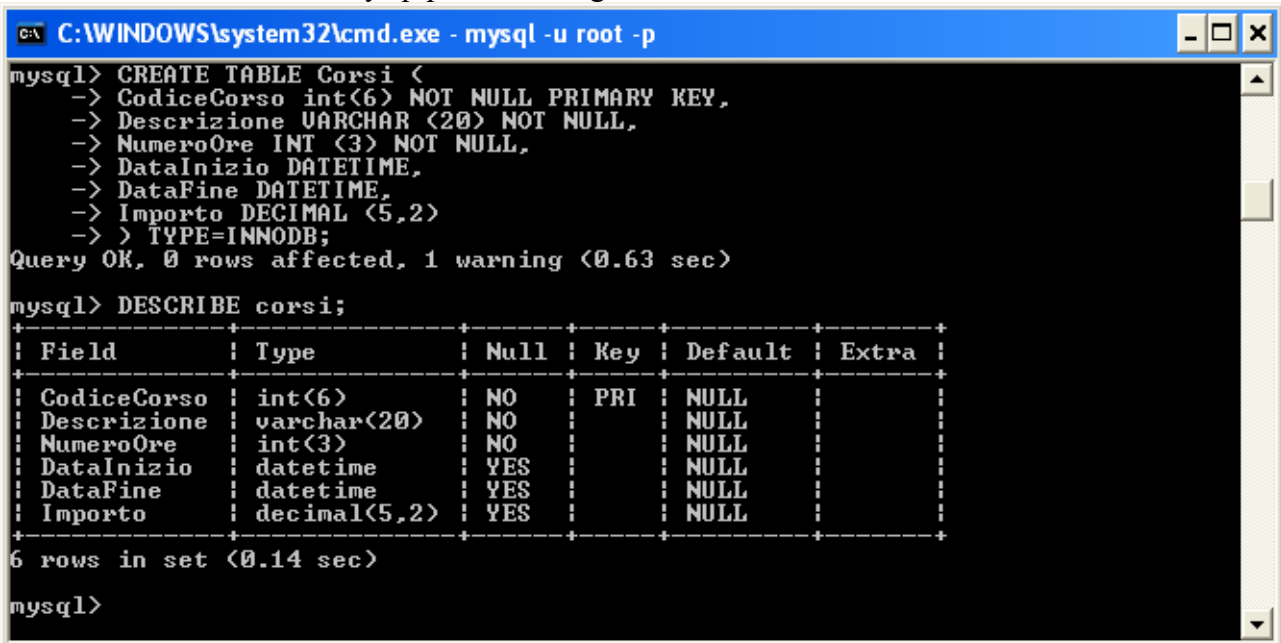
mysql>
```

Dobbiamo a questo punto definire su quale database intendiamo lavorare (use istruzione;)

Creiamo a questo punto le tabelle, iniziando dalla prima, **Corsi**

```
CREATE TABLE Corsi ( CodiceCorso INT(6) NOT NULL PRIMARY KEY, Descrizione
VARCHAR (20) NOT NULL, NumeroOre INT (3) NOT NULL, DataInizio DATETIME,
DataFine DATETIME, Importo DECIMAL (5,2));
```

che scritto direttamente in mysql produce il seguente risultato:



```
C:\WINDOWS\system32\cmd.exe - mysql -u root -p

mysql> CREATE TABLE Corsi (
->   CodiceCorso int(6) NOT NULL PRIMARY KEY,
->   Descrizione VARCHAR(20) NOT NULL,
->   NumeroOre INT(3) NOT NULL,
->   DataInizio DATETIME,
->   DataFine DATETIME,
->   Importo DECIMAL(5,2)
-> ) TYPE=INNODB;
Query OK, 0 rows affected, 1 warning (0.63 sec)

mysql> DESCRIBE corsi;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| CodiceCorso    | int(6)        | NO   | PRI | NULL    |       |
| Descrizione    | varchar(20)   | NO   |     | NULL    |       |
| NumeroOre      | int(3)        | NO   |     | NULL    |       |
| DataInizio     | datetime     | YES  |     | NULL    |       |
| DataFine       | datetime     | YES  |     | NULL    |       |
| Importo        | decimal(5,2) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.14 sec)

mysql>
```

Abbiamo anche inserito l'istruzione di visualizzazione (DESCRIBE) che mostra la struttura appena creata. Creiamo allo stesso modo la seconda tabella, **Utenti**

```
CREATE TABLE Utenti ( Matricola INT (6) NOT NULL PRIMARY KEY, Nome VARCHAR
(15) NOT NULL, Cognome VARCHAR (20) NOT NULL, Indirizzo VARCHAR (20), Citta
VARCHAR (15), Telefono VARCHAR (10));
```

che inserito nel db produce

```
mysql> describe utenti;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Matricola  | int(6)        | NO   | PRI | NULL    |       |
| Nome       | varchar(15)   | NO   |     | NULL    |       |
| Cognome    | varchar(20)   | NO   |     | NULL    |       |
| Indirizzo  | varchar(20)   | YES  |     | NULL    |       |
| Citta      | varchar(15)   | YES  |     | NULL    |       |
| Telefono   | varchar(10)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.05 sec)

mysql>
```

Bene, a questo punto non ci rimane che implementare la terza tabella, sulla quale dobbiamo fare qualche considerazione riguardo l'integrità referenziale che si ha considerando la relazione tra le tabelle.

In pratica la tabella *Iscrizioni* serve da collegamento tra le tabelle *Utenti e Corsi*, e rispettivamente si ha una relazione 1 a molti (1 a N) tra *Utenti e Iscrizioni*, e un'altra, sempre 1 a molti, tra *Corsi e Iscrizioni*. Con le considerazioni sulle regole di derivazione dal modello concettuale a quello logico sappiamo che una associazione 1 a N origina nella tabella che partecipa a N un nuovo campo che collegherà la tabella alla chiave primaria della tabella che partecipa a 1 all'associazione. Detto in altri termini devo avere che il campo *CodiceCorso* sarà chiave esterna verso *CodiceCorso* della tabella *Iscrizioni*, mentre il campo *Matricola* sarà chiave esterna verso *Matricola* della tabella *Utenti*.

Vediamo di mettere in pratica quanto visto con le istruzioni sql.

```
CREATE TABLE Iscrizioni ( CodiceCorso INT (6) NOT NULL, Matricola INT (6) NOT NULL,
DataIscrizione DATETIME, PRIMARY KEY (CodiceCorso, Matricola));
```

dove abbiamo indicato la chiave primaria costituita da due campi (codiceCorso e Matricola) ed è necessario definirla in questo modo. Dal momento che ci deve essere una corrispondenza tra i campi indicati delle due tabelle, abbiamo definito CodiceCorso e Matricola esattamente come compaiono nelle rispettive tabelle come chiavi principali.

Facciamo altre considerazioni.

1. come già visto un campo che viene usato come chiave esterna verso un'altra chiave principale viene indicizzato, per ottimizzare le prestazioni in termini di accesso del DB. Quindi aggiungiamo le righe

```
INDEX codice_key (CodiceCorso)
INDEX matricola_key(Matricola)
```
2. dobbiamo creare i riferimenti dei campi appena indicizzati con le chiavi primarie delle altre tabelle.

```
Foreign KEY (CodiceCorso) REFERENCE Corsi (CodiceCorso)
Foreign KEY (Matricola) REFERENCE Utenti (Matricola)
```

Per entrambi i casi mettiamo le condizioni che sulla cancellazione della riga principale (a cui la chiave esterna punta) comporta la cancellazione delle righe della tabella figlia, mentre per variazioni della tabella madre riportiamo le variazioni anche sul campo chiave esterna della tabella figlia.

A questo punto la creazione della tabella può essere completata, mettiamo assieme le righe viste

```
CREATE TABLE Iscrizioni ( CodiceCorso INT (6) NOT NULL, Matricola INT (6) NOT NULL,
DataIscrizione DATETIME, PRIMARY KEY (CodiceCorso, Matricola), INDEX codice_key
(CodiceCorso), INDEX matricola_key(Matricola), FOREIGN KEY (CodiceCorso) REFERENCES
Corsi (CodiceCorso) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY
(Matricola) REFERENCES Utenti (Matricola) ON DELETE CASCADE ON UPDATE
CASCADE);
```

```
mysql> describe iscrizioni;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| CodiceCorso | int(6)    | NO   | PRI | NULL    |       |
| Matricola   | int(6)    | NO   | PRI | NULL    |       |
| DataIscrizione | datetime | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql>
```

A questo punto possiamo popolare le tabelle, ovviamente iniziando da quelle madri, ovvero dalle tabelle che sono indipendenti. Se dovessimo iniziare dalla tabella iscrizioni, non potremmo collegare i record inseriti a nessun record di utenti e di corsi, quindi il sistema risponderebbe con un errore.

Iniziamo dalla tabella dei corsi, e ci accorgiamo che magari il campo CodiceCorso, chiave primaria della tabella, avremmo dovuto, per comodità, definirla di tipo auto_increment, vediamo di aggiungere questa caratteristica al campo.

Per farlo vediamo qualche riga di teoria e poi realizziamo quanto proposto. Dobbiamo introdurre il comando che consente la modifica la struttura di una tabella, ALTER TABLE secondo la sintassi:

ALTER TABLE nome_tabella elenco_azioni

che consente anche diverse modifiche contemporaneamente realizzando una lista di elementi separati dalla virgola. Vediamo alcune delle operazioni più comuni

Modificare una tabella

Per cambiare il nome di una tabella:

```
SINTASSI: ALTER TABLE Nome_Tabella RENAME Nuovo_Nome;
```

```
mysql> ALTER TABLE corsi RENAME corsi2;
Query OK, 0 rows affected (0.05 sec)

mysql> show tables;
+-----+
| Tables_in_istruzione |
+-----+
| corsi2                |
| iscrizioni            |
| utenti                |
+-----+
3 rows in set (0.30 sec)

mysql>
```

Modificare una colonna

Ci sono migliaia di motivi per cui potrete trovarvi nella necessità di modificare il nome o le proprietà di una colonna. Quando ne avrete bisogno eseguite:

```
SINTASSI: ALTER TABLE Nome_Tabella CHANGE Nome_Colonna_da_cambiare
Nuovo_Nome_Colonna Proprietà_della_Nuova_Colonna;
```

supponiamo di voler cambiare il nome della colonna Descrizione e il numero di caratteri per questo campo (da 20 a 25)

```
mysql> alter table corsi2 CHANGE Descrizione DescrizioneNew VARCHAR(25);
Query OK, 0 rows affected (0.33 sec)
Records: 0 Duplicati: 0 Avvertimenti: 0

mysql> describe corsi2;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| CodiceCorso    | int(6)        | NO   | PRI | NULL    | auto_increment |
| DescrizioneNew | varchar(25)   | YES  |     | NULL    |                |
| NumeroOre      | int(3)        | NO   |     | NULL    |                |
| DataInizio     | datetime     | YES  |     | NULL    |                |
| DataFine       | datetime     | YES  |     | NULL    |                |
| Importo        | decimal(5,2) | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

Posso modificare anche solo il tipo della colonna, in quel caso il nuovo nome sarà uguale al vecchio.

Aggiungere una colonna

Per aggiungere una colonna si esegue:

```
SINTASSI: ALTER TABLE Nome_Tabella ADD Nome_della_Nuova_Colonna
Proprietà_Colonna;
```

aggiungiamo una colonna AnnoNascita alla tabella utenti, in posizione successiva a Cognome.

```
mysql> ALTER TABLE utenti ADD AnnoNascita INT(4) AFTER Cognome;
Query OK, 0 rows affected (0.39 sec)
Records: 0 Duplicati: 0 Avvertimenti: 0

mysql> describe utenti;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| Matricola      | int(6)        | NO   | PRI | NULL    |                |
| Nome           | varchar(15)   | NO   |     | NULL    |                |
| Cognome        | varchar(20)   | NO   |     | NULL    |                |
| AnnoNascita    | int(4)        | YES  |     | NULL    |                |
| Indirizzo     | varchar(20)   | YES  |     | NULL    |                |
| Citta         | varchar(15)   | YES  |     | NULL    |                |
| Telefono      | varchar(10)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.01 sec)

mysql>
```

Eliminare una colonna

Per eliminare una colonna si userà il comando DROP, che abbiamo già visto nella lezione precedente. Come si può immaginare, tutti i dati contenuti nella colonna verranno persi dopo l'esecuzione del comando, quindi attenzione! Digitate:

```
SINTASSI: ALTER TABLE Nome_Tabella DROP Nome_Colonna;
```

eliminiamo la colonna appena inserita nella tabella "utenti"

```
mysql> describe utenti;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Matricola  | int(6)        | NO   | PRI | NULL     |       |
| Nome       | varchar(15)   | NO   |     | NULL     |       |
| Cognome    | varchar(20)   | NO   |     | NULL     |       |
| Indirizzo  | varchar(20)   | YES  |     | NULL     |       |
| Citta      | varchar(15)   | YES  |     | NULL     |       |
| Telefono   | varchar(10)   | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.03 sec)

mysql>
```

Bene, dopo aver visto questi esempi che mostrano come modificare la struttura di una tabella, torniamo al problema iniziale, quello di far diventare il campo CodiceCorso della tabella corsi come AUTO_INCREMENT. L'istruzione sarà la seguente, in analogia con gli esempi fatti in precedenza.

ALTER TABLE corsi MODIFY CodiceCorso INT (6) AUTO_INCREMENT;

a questo punto torniamo al popolamento delle tabelle.

Utenti

1	Pinco	Pallino	Via sassari	Sassari	079797979
2	Ciccio	Ciccio	Via cagliari	Cagliari	070222222
3	Pippo	Pippo	Via boh	Boh	099001100
4	Pluto	Pluto	Via lui	Lei	08888990

```
mysql> Insert INTO utenti
-> VALUES (1,'Pinco','Pallino','Via sassari','Sassari','079797979'),
->          (2,'Ciccio','Ciccio','Via cagliari','Cagliari','070222222'),
->          (3,'Pippo','Pippo','via boh','Boh','099001100'),
->          (4,'Pluto','Pluto','Via lui','Lei','08888990');
Query OK, 4 rows affected (0.14 sec)
Records: 4 Duplicati: 0 Avvertimenti: 0

mysql>
```

Attenzione all'uso dell'apice (se la query si scrive in word ci potrebbero essere problemi nella conversione del ' usate in quel caso notepad)

Corsi

1	Italiano	14	30/10/2009	30/12/2009	1000
2	Storia	20	30/8/2009	30/10/2009	1000
3	Informatica	18	22/1/2009	30/04/2009	1000
4	Inglese	20	1/10/2009	3/11/2009	1000

```

C:\WINDOWS\system32\cmd.exe - mysql -u root -p
mysql>
mysql> Insert INTO corsi2
-> VALUES (1,'Italiano',14,'2009-10-30','2009-12-20',1000),
-> (2,'Storia',20,'2009-8-30','2009-10-30',1000),
-> (3,'Informatica',18,'2009-1-22','2009-4-30',1000),
-> (4,'Inglese',20,'2009-10-1','2009-11-3',1000);
Query OK, 4 rows affected, 4 warnings (0.11 sec)
Records: 4 Duplicati: 0 Avvertimenti: 4

mysql> select * from corsi2;
+-----+-----+-----+-----+-----+-----+
| CodiceCorso | DescrizioneNew | NumeroOre | DataInizio | DataFine | Importo |
+-----+-----+-----+-----+-----+-----+
| 1 | Italiano | 14 | 2009-10-30 | 2009-12-20 | 999.99 |
| 2 | Storia | 20 | 2009-08-30 | 2009-10-30 | 999.99 |
| 3 | Informatica | 18 | 2009-01-22 | 2009-04-30 | 999.99 |
| 4 | Inglese | 20 | 2009-10-01 | 2009-11-03 | 999.99 |
+-----+-----+-----+-----+-----+-----+

4 rows in set (0.00 sec)

mysql>

```

Iscrizioni(CodiceCorso,Matricola,DataIscrizione)

Utenti(Matricola,Nome,Cognome,Indirizzo,Città,Telefono)

Corsi (CodiceCorso, Descrizione, NumeroOre, DataInizio, DataFine,Importo)

```

INSERT INTO utenti VALUES (1,'Pinco','Pallino','Via sassari','Sassari',079797979),
(2,'Ciccio','Ciccio','Via cagliari','Cagliari',070222222), (3,'Pippo','Pippo','via boh','Boh',099001100),
(4,'Pluto','Pluto','Via lui','Lei',0888990);

```

```

INSERT INTO corsi2 VALUES (1,'Italiano',14,'2009-10-30','2009-12-20',1000),
(2,'Storia',20,'2009-8-30','2009-10-30',1000), (3,'Informatica',18,'2009-1-22','2009-4-30',1000),
(4,'Inglese',20,'2009-10-1','2009-11-3',1000);

```