

MYSQL



IL DBMS MYSQL

- MySQL e' un Database Management System (DBMS) open source.
- Puo' essere scaricato da link <https://dev.mysql.com/downloads/mysql/>
- Mysql mette a disposizione un client a riga di comando che consente di collegarsi al server MySQL per sfruttarne le funzionalità.
- Esiste anche una versione dotata di interfaccia grafica
 - <https://dev.mysql.com/downloads/workbench/>



COLLEGARSI AL DBMS

```
shell> mysql --user=root --password=xxx
```

oppure

```
shell> mysql -uroot -p
```



CONNESSIONE

- Una volta connessi apparirà la shell di mysql

```
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 6  
Server version: 5.7.18 Homebrew  
  
Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> █
```

- Da qui si potrà procedere all'invio di comandi
- Ogni comando deve terminare con ;



SHOW DATABASES

- Il comando `show databases;` mostra i database che sono presenti

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| CORSO      |
| metastore  |
| mysql      |
| performance_schema |
| sys        |
+-----+
6 rows in set (0.00 sec)
```



CREARE ED ELIMINARE UN DATABASE

- Sintassi del comando per creare un database:
 - CREATE DATABASE [IF NOT EXISTS] nome_db
 - IF NOT EXISTS p evita la segnalazione di errori se il DB esiste
- Sintassi del comando per eliminare un database:
 - DROP DATABASE [IF EXISTS] nome_db

```
mysql> create database ESEMPIO;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| CORSO      |
| ESEMPIO    |
| metastore  |
| mysql      |
| performance_schema |
| sys        |
+-----+
7 rows in set (0.00 sec)
```



IMPORTARE COMANDI

- E' possibile costruire un file con all'interno una serie di comandi che poi possono essere importati
- La sintassi e':
 - **source** nomefile.estensione



CREARE TABELLE

- Come prima cosa bisogna selezionare uno dei database esistenti tramite il comando: `use nomeDB;`

```
mysql> use CORSO;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed
```

- Per creare una tabella la sintassi e':
 - `CREATE TABLE [IF NOT EXISTS] nome_tabella`
 - `[([definizione attributi] [opzioni di tabella])]`



CREARE TABELLE

- ***Reference_definition***
- Tramite le ***reference_definition*** è possibile definire vincoli di integrità referenziale, ovvero l'attributo su cui è definito può assumere solo valori specificati nell'attributo di un'altra tabella.
 - REFERENCES nome_tabella [(colonna_indice,...)]



CREARE TABELLE

```
CREATE TABLE [IF NOT EXISTS] nome_tabella  
[(  
  [definizione attributi]  
  [opzioni di tabella]  
)]
```

- ▶ Le **opzioni tabella** si riferiscono all'intera tabella e permettono di definire diverse proprietà di questa.

- ▶ Le più importanti sono:

PRIMARY KEY (nome_attributo1, nome_attributo2,...)

Permette di definire come chiave primaria della tabella un insieme di attributi di questa.

INDEX (nome_attributo1, nome_attributo2,...)

Permette di definire degli indici su uno o più attributi della tabella

FOREIGN KEY (nome_att1,nome_att2,...)

REFERENCE nome_tab(nome_att1,nome_att2,...)

Permette di definire vincoli di integrità referenziale su più attributi



ESEMPIO

- Creare la seguente tabella:
 - **Persona (nome, cognome, età, sesso)**
 - Nome character(20)
 - Cognome Character (20)
 - Codice fiscale character (20) primary key
 - Età numeric (2)
 - Sesso character

```
Database changed
mysql> CREATE TABLE Persone( Nome character(20) primary key, Cognome character (20), codiceFiscale character(20), eta numeric (2), sesso character);
Query OK, 0 rows affected (0.02 sec)

mysql> show tables
-> ;
+-----+
| Tables_in_corso |
+-----+
| Corso           |
| Persone         |
| Studenti        |
+-----+
3 rows in set (0.00 sec)
```



VISUALIZZARE LE COLONNE

- Si usa il comando DESCRIBE nomeTabella

```
mysql> describe Persone
```

```
  -> ;
```

Field	Type	Null	Key	Default	Extra
Nome	char(20)	NO	PRI	NULL	
Cognome	char(20)	YES		NULL	
codiceFiscale	char(20)	YES		NULL	
eta	decimal(2,0)	YES		NULL	
sesso	char(1)	YES		NULL	

```
5 rows in set (0.01 sec)
```



MODIFICARE LE TABELLE

- Si usa il comando:

```
ALTER TABLE nome_tabella
  ADD nome_attributo TIPO |
  ADD INDEX (nome_attributo,...) |
  ADD [CONSTRAINT [nome_vincolo]] PRIMARY KEY (nome_attributo,...) |
  ADD [CONSTRAINT [nome_vincolo]] UNIQUE (nome_attributo,...)
  ADD [CONSTRAINT [nome_vincolo]]
      FOREIGN KEY (colonna_indice,...) [reference_definizione] |
  CHANGE vecchio_attributo nuovo_attributo TIPO |
  DROP nome_attributo
  DROP PRIMARY KEY
  DROP INDEX nome_attributo
  .....
  .....
```



RINOMINARE TABELLE

- Rinominiamo la tabella Persone in Individui

```
mysql> Alter table Persone rename Individui;
Query OK, 0 rows affected (0.01 sec)

mysql> show tables
-> ;
+-----+
| Tables_in_corso |
+-----+
| Corso           |
| Individui       |
| Studenti        |
+-----+
3 rows in set (0.00 sec)
```



RINOMINARE UNA COLOLLA

- Rinominiamo la colonna età in anni

```
mysql> ALTER TABLE Individui CHANGE eta anni numeric(3);
Query OK, 0 rows affected (0.03 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe Individui;
```

Field	Type	Null	Key	Default	Extra
Nome	char(20)	NO	PRI	NULL	
Cognome	char(20)	YES		NULL	
codiceFiscale	char(20)	YES		NULL	
anni	decimal(3,0)	YES		NULL	
sesso	char(1)	YES		NULL	

```
5 rows in set (0.00 sec)
```



VINCOLI DI CHIAVA ESTERNA

- Si usa la sintassi

`ALTER TABLE Nome_Tabella`

`ADD CONSTRAINT [nome_vincolo]`

`FOREIGN KEY (nome_col_che_referenzia)`

`REFERENCE Nome_Tabella_Referenziata(nome_colonna_refe);`



INSERIRE DATI

- Si usa la sintassi:
- `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.



ESEMPIO

```
mysql> INSERT INTO Individui(Nome, Cognome, codiceFiscale, anni, sesso) VALUES('Giovanni', 'Rossi', 'GVNRSS78R34C616T',25,'M');  
Query OK, 1 row affected (0.01 sec)
```



INTERROGARE DATI

- Si usa il comando SELECT, che ha la seguente sintassi:

```
SELECT nome_attributo,...,nome_attributo  
FROM nome_tabella, ...,nome_tabella  
[WHERE condizione]
```

Le tre parti sono solitamente chiamate:

- ▶ target list
- ▶ clausola from
- ▶ clausola where



ESEMPIO

- Selezionare tutti gli Individui

```
mysql> select * from Individui;
```

Nome	Cognome	codiceFiscale	anni	Sesso
Giovanni	Rossi	GVNRSS78R34C616T	25	M

1 row in set (0.00 sec)



ELIMINARE RECORD

- Si usa la sintassi: `DELETE FROM nome_tabella [WHERE condizione]`
- Eliminiamo tutte le persone che hanno meno di 30 anni

```
[mysql> DELETE from Individui WHERE anni<30;  
Query OK, 1 row affected (0.00 sec)
```

```
[mysql> select * from Individui;  
Empty set (0.00 sec)
```

