

Databases 101

<https://www.w3schools.com/sql/>

I database sono strutture ordinate di dati che vengono interrogati attraverso il linguaggio *SQL*. Le operazioni sui database sono di tipo *CRUD* (creation, reading, updating, deleting).

Il vantaggio principale dell'utilizzo di un database è la *persistenza* dei dati.

altri vantaggi sono:

1. la capacità di immagazzinare grandi quantità di dati;
2. offrire funzionalità per gestione e sviluppo di applicazioni;
3. Garantire una buona velocità di elaborazione;
4. Consentire a più utenti di accedere ai dati;
5. Assicurare l'affidabilità dei dati.

La *Modellazione dei dati* è un processo di progettazione del database attraverso la creazione di *tables* (tabelle). Queste contengono dati contrassegnati da un *tipo* (intero, string boolean, etc). I dati sono salvati all'interno del Database per garantirne l'accesso anche tra esecuzioni diverse.

Ci sono 2 modelli principali:

1. Modello *Entità-Relazione*;
2. Modello a *Oggetti*.

Sintassi

Per creare ed entrare in un database:

```
CREATE [database_name];  
  
USE [database_name];
```

Per creare una tabella:

```
CREATE TABLE [table_name] ([nome_attributo] [tipo_attributo], ...);
```

si possono aggiungere delle caratteristiche agli attributi, come:

```
CREATE TABLE [table_name] ([nome_attributo] [tipo_attributo] NOT NULL AUTO_INCREMENT  
PRIMARY KEY ....., ...);
```

Per aggiungere una Foreign Key, dopo aver dichiarato gli attributi della tabella:

```
CREATE TABLE [table_name] (attributo1 int,  
    ...  
    ...  
    FOREIGN KEY (attributo1) REFERENCES [table_2](attributo2));
```

Per selezionare dei *record* da una tabella si usa un SELECT statement:

```
SELECT [attributo_1],[attributo_2],...,[attributo_n] FROM [table_name];
```

Per ottenere tutti gli attributi di un record si usa *:

```
SELECT * FROM [table_name];
```

L'operatore SELECT è modulare, si possono aggiungere, in maniera opzionale ma in questo ordine WHERE, ORDER BY, ASC/DESC: con la sintassi:

```
SELECT * FROM [table_name] WHERE [attributo_1] = [qualcosa] ORDER BY  
[attributo_per_cui_ordinare] ASC/DESC;
```

Per aggiungere delle entrate in una tabella:

```
INSERT INTO [table_name] ([attributo_1],[attributo_2],...[attributo_n]) VALUES  
([valore_attributo1],...[valore_attributo_n]);
```

Gli attributi sono assegnati nell'ordine specificato.

Si possono aggiungere più record per volta aggiungendo un'altra tonda dopo la prima con la stessa sintassi.

Se non vengono inseriti gli attributi di partenza, SQL assumerà che stiano venendo assegnati tutti.

Per aggiornare gli elementi di una tabella:

```
UPDATE [table_name] SET [attributo_da_cambiare] = [qualcosa] WHERE [condizione];
```

IMPORTANTE usare sempre WHERE sia per quest'operazione che per la prossima, pena cancellazione dei dati, senza possibilità di recupero.

Per eliminare un record da una tabella:

```
DELETE FROM [table_name] WHERE [condizione];
```

WHERE (condizione)

Gli operatori utilizzabili sono:

Operatore	Esempio
uguale: =	nome = "Giggino"

Operatore	Esempio
maggiore: >	età > 15
minore: <	età < 15
maggiore e uguale: >=	anno >= 1990
minore e uguale: <=	anno <= 1990
diverso: <>	città <> "Livorno"
valori tra: <i>BETWEEN</i>	codice BETWEEN 10 AND 20
cercare un pattern: <i>LIKE</i>	
cercare diversi possibili valori: <i>IN</i>	genere IN ("Horror","Commedia")

Le condizioni di un WHERE possono essere concatenate con gli operatori *AND* e *OR*, ad esempio:

```
SELECT * FROM tabella WHERE nome = "franco" AND cognome "Novaga";

--oppure

SELECT * FROM tabella WHERE nome = "franco" OR id = 1;
```

Le condizioni possono essere negate con l'operatore *NOT*:

```
SELECT * FROM tabella WHERE NOT città = "Livorno";
--che schifo Livorno
```

Modello Entità-Relazione

Un'*entità* è qualsiasi cosa sul quale sono raccolte informazioni.

Gli *attributi* sono quegli oggetti che descrivono un'entità.

Le *Relazioni* sono oggetti che collegano le entità di un database.

Tutti gli oggetti devono avere nomi unici e significativi.

Il primo ci permette di usare entità (tabelle ad esempio) e relazioni, che sono i modi con cui le tabelle comunicano tra di loro, è quello che useremo principalmente.

I dati di due tabelle diverse potrebbero essere collegati attraverso delle relazioni. Un attributo di una tabella potrebbe essere segnato come *Primary Key*, una chiave che "indicizza" in maniera unica gli elementi di una tabella, come ad esempio il numero di matricola di studenti universitari.

Possiamo collegare due tabelle utilizzando una *Foreign Key*, che "punta" alla chiave primaria di un'altra tabella.

MySQL

è un DbMs è un Database Managment System, che permette di interagire sia con i dati stessi che con i vari protocolli di sicurezza.