Sommario

[Informazioni e dati 2](#_Toc514274815)

[Entità e attributi 3](#_Toc514274816)

[Caratteristiche degli attributi 4](#_Toc514274817)

[Le relazioni 6](#_Toc514274818)

[Cardinalità 8](#_Toc514274819)

[Attributi delle relazioni 9](#_Toc514274820)

[Strumenti per la progettazione : draw io 10](#_Toc514274821)

[Lo schema relazionale 12](#_Toc514274822)

[Normalizzazione 12](#_Toc514274823)

[Implementazione di un database (gli strumenti) 12](#_Toc514274824)

[Interrogazioni di una base di dati 12](#_Toc514274825)

[Select From e Where 12](#_Toc514274826)

[Operazioni sugli attributi 12](#_Toc514274827)

[Aliasing 12](#_Toc514274828)

[Order by 13](#_Toc514274829)

[Limit 13](#_Toc514274830)

[Raggruppamento 13](#_Toc514274831)

# Informazioni e dati

Le **informazioni** hanno lo scopo di aumentare la conoscenza. Al contrario di molti altri ambiti, nell’informatica l’informazione deve essere trasmessa in un unico modo, più precisamente l'informazione deve essere ben strutturata, chiara, deve esplicitare efficacemente quello che intende rappresentare.

L'informazione verbale, quella a cui siamo abituati e che otteniamo per mezzo di comunicazioni verbali, può essere talvolta fraintendibile o lacunosa, in alcuni casi incompleta o ambigua, in generale poco strutturata.

Per rendere un’informazione strutturata occorre andare oltre l'utilizzo del solo linguaggio naturale, trasformando l'informazione in un dato, ovvero un qualcosa di standard, inconfondibile, oggettivo e schematico.

I **dati** sono descrizioni elementari, spesso codificate, di un’entità, di un fenomeno o di altro. L’elaborazione dei dati può portare alla conoscenza di un’informazione, ogni tipo di dato dipende dal codice e dal formato impiegati. I dati devono seguire una regola, devono essere normalizzati, e, attraverso le

memorie di massa, possono essere salvati in modo permanente. Singoli dati o insiemi di dati possono essere catalogati in file o nei database. Inoltre un dato acquista valore solo se posto in relazione ad un contesto.

# Entità e attributi

L’entità rappresenta un concetto complesso e di rilievo che descrive classi di oggetti con esistenza autonoma. Ad esempio possibili entità di un campionato sportivo sono: Squadra, giornata di campionato, giocatore, partita. Un’entità può avere un solo nome nello schema concettuale e viene rappresentata nel diagramma ER con il nome all’interno di un rettangolo.

Un attributo è un concetto che ha una struttura semplice, esso non ha esistenza autonoma ma è associato ad una entità o ad una relazione. Un attributo ha un solo significato all’interno dell’insieme degli attributi della stessa entità o relazione. Dunque attributi di entità o relazioni diverse possono avere lo stesso nome. Esso si rappresenta graficamente con i rispettivi simboli indicati nel punto seguente collegato con una linea all’entità o relazione e con scritto accanto il nome dell’attributo.

Per spiegare meglio i concetti espressi ci serviamo di un esempio:

Si vuole realizzare un database per la memorizzazione di alcune informazioni riguardanti i comuni italiani.

**Ogni comune italiano (di cui vogliamo registrare il nome, l’indirizzo della sede e il nome del sindaco attuale) appartiene a una provincia (identificate da un ID e un nome). In ognuno di essi sono nati e hanno vissuto (in un determinato periodo e in una o più case) più cittadini (dei quali ci interessa il nome, il cognome, il sesso, le loro nazionalità, il codice fiscale e se sono morti).Inoltre i cittadini possono essere attualmente sposati tra di loro.**

Nel nostro caso, analizzando le informazioni che ci vengono date nel testo, possiamo capire che le entità sono tre: cittadino, comune e provincia.

Le entità si rappresentano come dei rettangoli con il proprio nome all’interno.



Gli attributi sono delle proprietà significative di un’entità che ne attribuiscono delle informazioni utili per descriverle meglio.

Essi hanno un nome, un tipo e delle caratteristiche.

## Caratteristiche degli attributi

Per attributo s’intende una caratteristica che specifica un determinato oggetto.

***Attributi semplici***

Un attributo si definisce semplice quando non è ulteriormente scomponibile, cioè rappresenta un’unità informativa di base che caratterizza un’entità e al quale è associato un singolo valore. (Esempio: numero progressivo in un elenco.)

***Attributi composti***

***(C)***

Un attributo si definisce composto quando è ulteriormente scomponibile in più attributi semplici, cioè rappresenta contemporaneamente più unità informative di base che caratterizzano un’entità. (Esempio: un attributo composto da sottoattributi come l’indirizzo.)

***Attributi multipli***

***(M)***

Un attributo si definisce multiplo quando a esso possono essere associati anche più valori dello stesso tipo contemporaneamente. (Esempio: un attributo come sport che può essere riempito da più di un valore.)

***Attributi not null o obbligatori***

***(NN)***

Sono attributi che devono contenere necessariamente qualcosa, cioè non possono essere nulli, in quanto essenziali.

***Attributi null o facoltativi***

Un attributo si definisce facoltativo quando il suo contenuto non è di fondamentale importanza, cioè può essere nullo.

***Chiave primaria***

***(PK)***

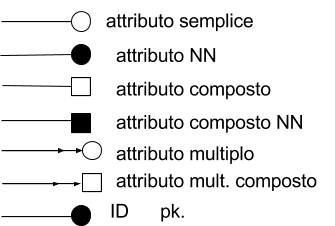
La chiave primaria è un insieme di attributi che permette di individuare univocamente un record in una tabella. È unica per ogni tabella e non può essere nulla. (Esempio: ID, codice fiscale..)

***Attributi unique***

***(U)***

È una chiave univoca, con le stesse caratteristiche della chiave primaria, ad eccezione che essa può essere nulla.

In forma grafica vengono rappresentati nel seguente modo:



È possibile fare un’altra distinzione in base alla tipologia degli attributi, che quindi si possono dividere, per il momento, in:

* Numerici, spesso abbreviati con la sigla num, composti esclusivamente da caratteri numerici
* String, composti da caratteri alfanumerici
* Char, composti da una sola lettera dell’alfabeto
* Boolean, che permettono di fare una scelta, ad esempio tra vero o falso, si o no, ecc.
* Datetime, che permettono di inserire come dato una data o un orario.

Ecco gli attributi delle entità ricavati dal testo, con il proprio tipo indicato tra parentesi:

|  |  |  |
| --- | --- | --- |
| ***Cittadino***  codice fiscale (string)  nome (string)  cognome (string)  sesso (char)  nazionalità (string)  morto (boolean) | ***Comune***  nome (string)  nome sindaco (string)  indirizzo sede (string)  id (numerico) | ***Provincia***  id (numerico)  nome (string) |

# Le relazioni

Le relazioni rappresentano legami logici tra due o più entità. Una relazione, normalmente,viene rappresentata graficamente nello schema “E-R” mediante un rombo con il suo nome, all’interno o al di sopra di esso,e da linee che connettono la relazione con le entità componenti.

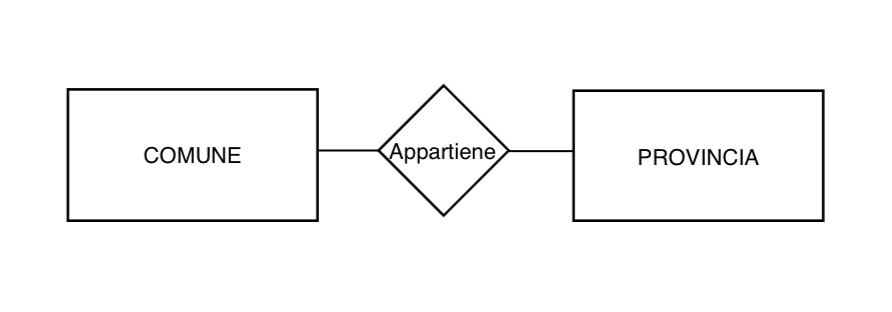
Il nome della relazione può essere un verbo in modo da fornire un’indicazione di lettura oppure può essere un sostantivo in modo da non dare una direzione di lettura.

Le relazioni oltre a collegare due entità diverse, possono legare una sola entità (relazioni su se stessi).

Continuiamo utilizzando il solito esempio:

Tra l’entità “comune” e l’entità “provincia” esiste una relazione individuata (nella traccia) dal verbo “appartiene”.

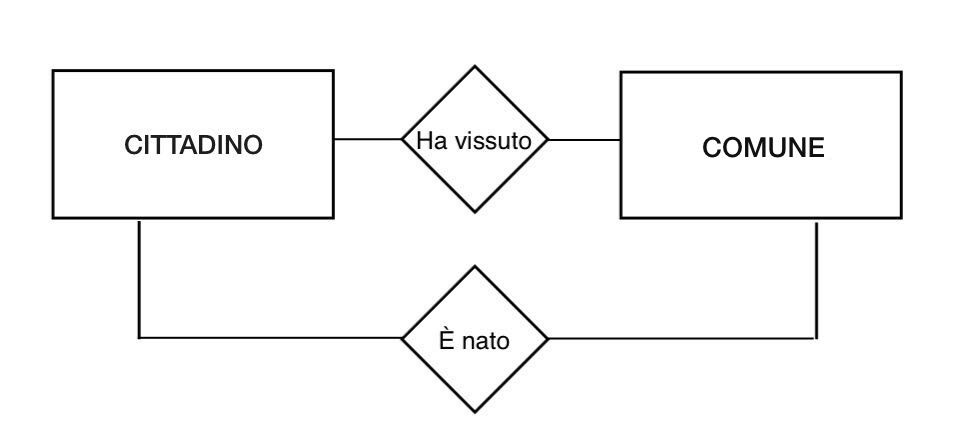
Questa relazione viene rappresentata collegando i due rettangoli (contenenti i nomi dell’entità) attraverso un segmento, con al centro un rombo contenente il nome della relazione.



È possibile ricavare altre relazioni dal testo.

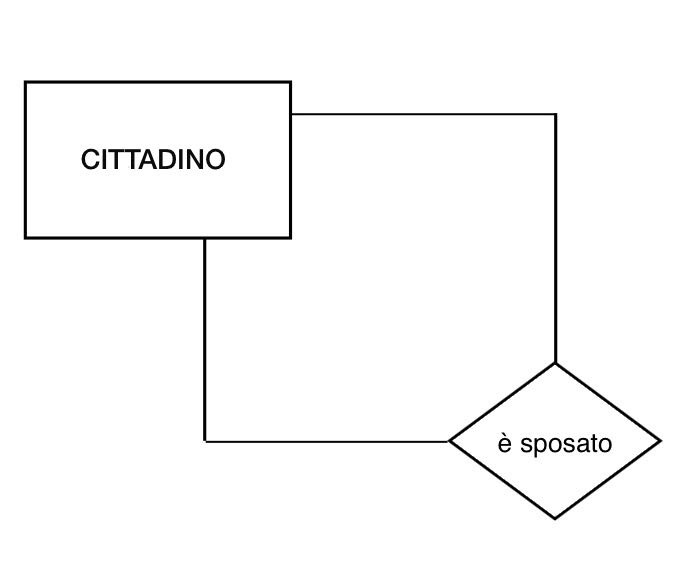
Tra l’entità “cittadino” e l’entità “comune” ce ne sono ben due, individuate dai verbi “ha vissuto” ed “è nato”.

Vengono rappresentate con lo stesso ragionamento:



L’ultima relazione ricavabile è differente dalle precedenti.

La traccia dell’esercizio ci dice, infatti, che alcuni cittadini possono essere sposati; cioè due cittadini possono avere una relazione tra di loro. È possibile legare un’entità con se stessa, e ciò viene rappresentato in questa maniera:



## Cardinalità

Per ciascuna relazione viene indicato quante volte un’entità può essere legata ad un’altra.

Vengono usate tre combinazioni:

* 1-1 (uno a uno)
* 1-N (uno a molti)
* N-N (molti a molti)

Esaminiamo tutte le relazioni trovate e attribuiamo loro una cardinalità:

* “Comune appartiene a provincia”;

un comune può appartenere a una provincia, ma una provincia può avere più comuni, la cardinalità quindi è 1-N (uno a molti).

* “Cittadino ha vissuto in comune”;

un cittadino può aver vissuto in molti comuni e in un comune possono aver vissuto molti cittadini, in questo caso la cardinalità è N-N (molti a molti).

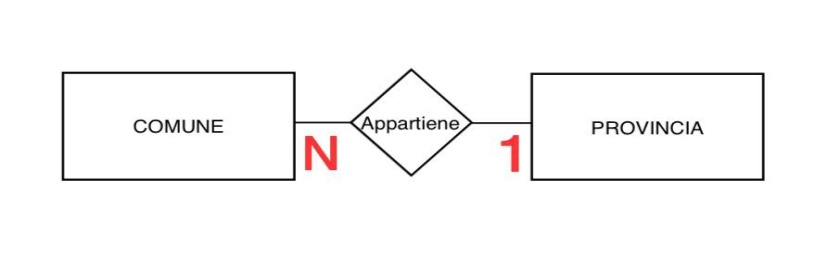
* “Cittadino è nato in comune”;

un cittadino è nato solamente in un comune, ma in un comune possono essere nati più cittadini. Anche questa è di tipo 1-N.

**N.B!** La cardinalità deve essere indicata anche nelle relazioni da un entità su se stessa.

* “Cittadino è sposato con cittadino”;

un cittadino può essere attualmente sposato con massimo un altro cittadino, quindi la relazione sarà 1-1.



I valori della cardinalità sono indicati vicino alla rappresentazione della relazione, seguendo il verso del ragionamento:

## Attributi delle relazioni

Abbiamo visto che le entità sono collegate tra loro attraverso delle relazioni, e che queste entità sono caratterizzate da attributi più o meno identificativi.

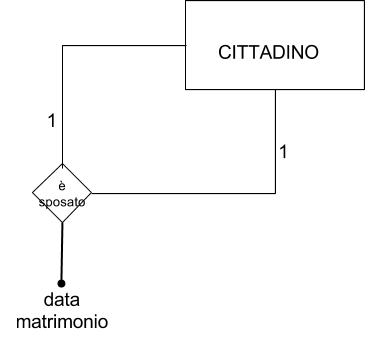
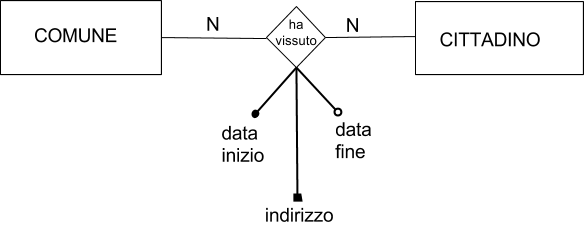
Dobbiamo però tener conto che anche le relazioni possono avere degli attributi; essi seguono i tipi e le caratteristiche di quelli delle entità.

Essi si verificano quando la cardinalità delle relazioni è N-N o 1-1, ma possono essere presenti, anche se raramente, nelle relazioni N-1.

Nel nostro esempio possiamo ricavare ben tre attributi della relazione tra “comune” e “cittadino” nominata “ha vissuto”.

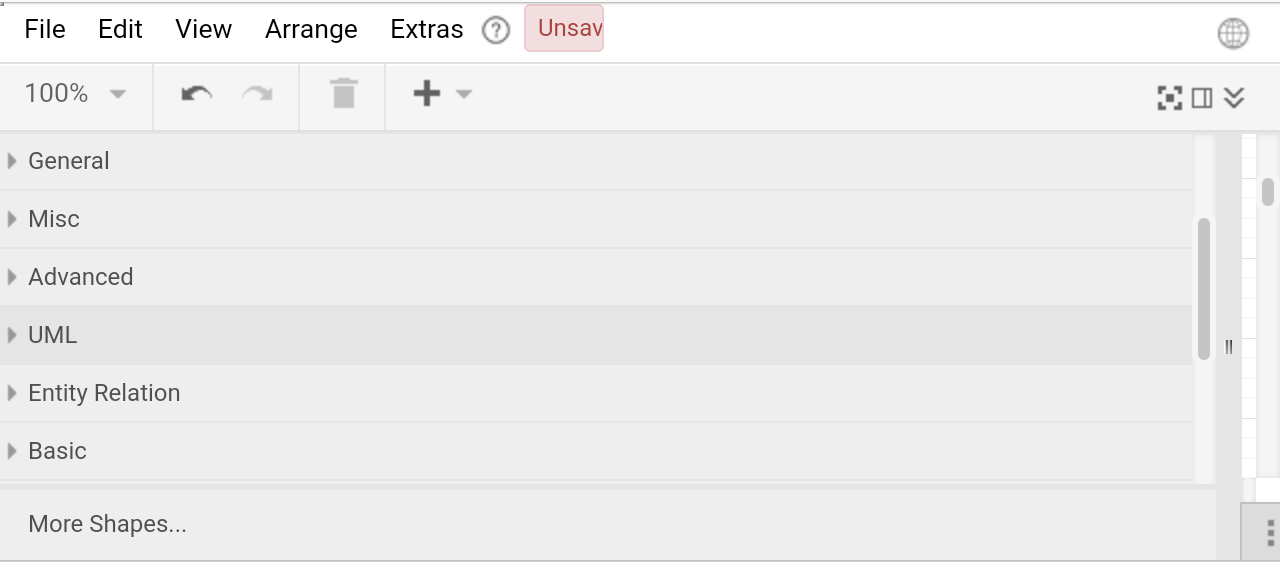
Infatti ci interessa sapere da quando un cittadino ha vissuto in quel determinato comune, e, nel caso non ci abitasse più, quando è avvenuto il cambiamento. Inoltre vogliamo anche sapere quali sono stati gli indirizzi di abitazione.

Tutto ciò è rappresentabile così:

  
  
  
  
Gli attributi esistono anche nelle relazioni tra la stessa identità, ci si comporta nella stessa maniera :

# Strumenti per la progettazione : draw io

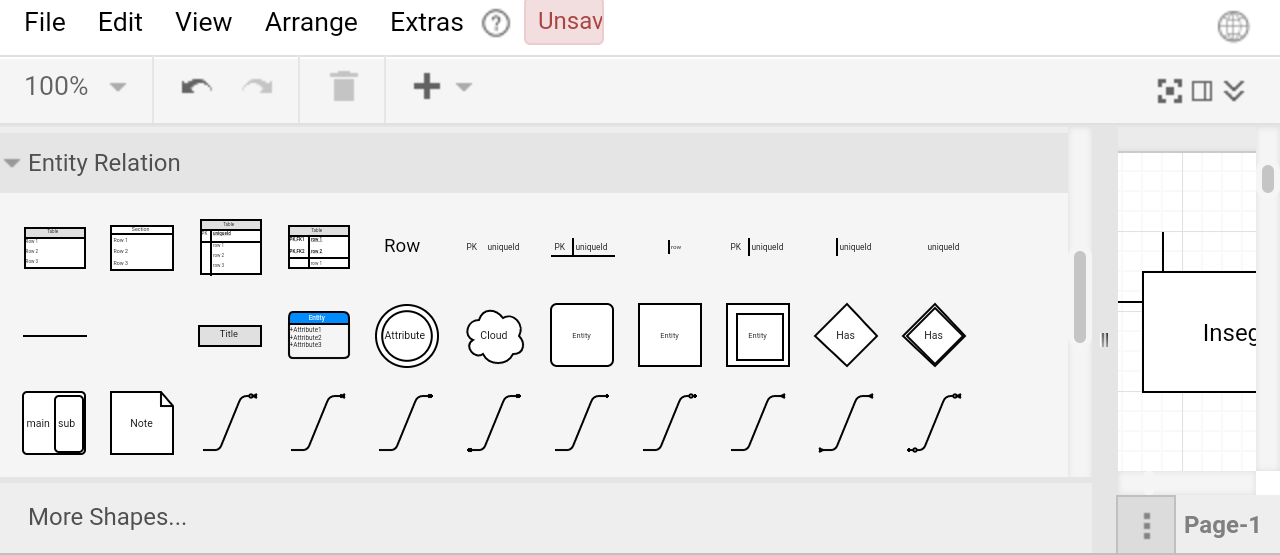
Draw.io è un'applicazione che permette di creare diagrammi, perciò è molto utile per creare diagrammi entità-relazione.



Il pulsante con il *100%* serve per fare lo zoom o ridurre il piano di lavoro.

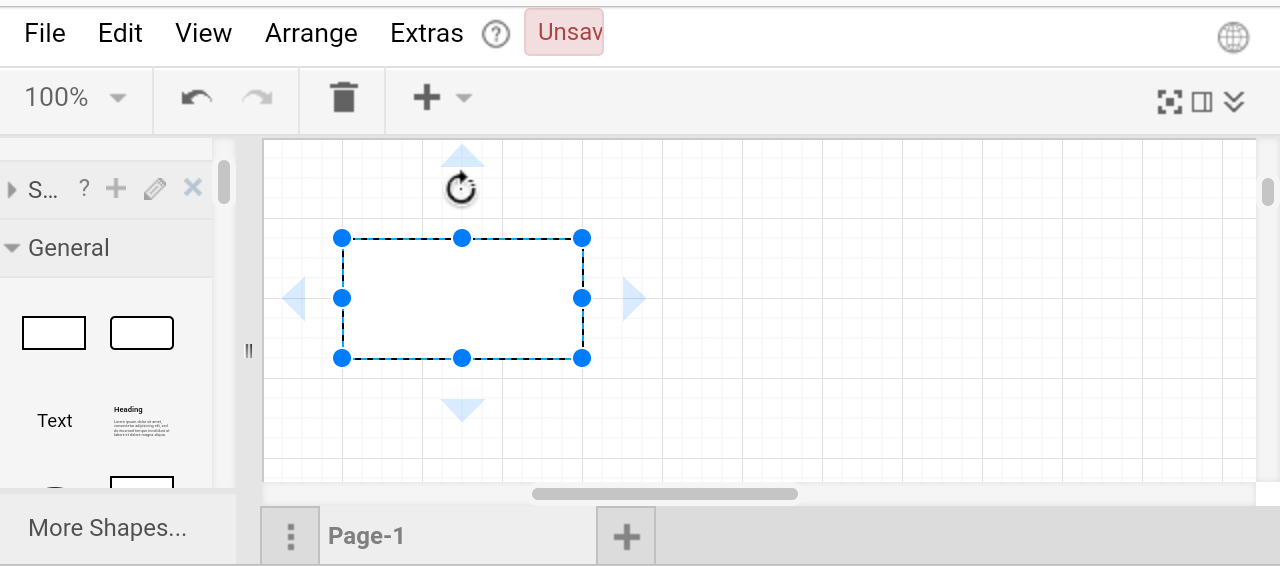
Le due frecce sono per tornare indietro, accanto c’è il cestino e infine il *+* serve per aggiungere al piano di lavoro immagini, testi, forme ecc.

Ci sono molte forme disponibili da usare per creare il proprio diagramma, per quello entità-relazione è molto utile la categoria *Entity Relation*.



Per creare il diagramma si trascinano le figure sul piano di lavoro.

Una volta fatto si possono modificare le dimensioni, far ruotare le figure e cambiarle di posizione trascinandole nel punto desiderato del piano di lavoro.



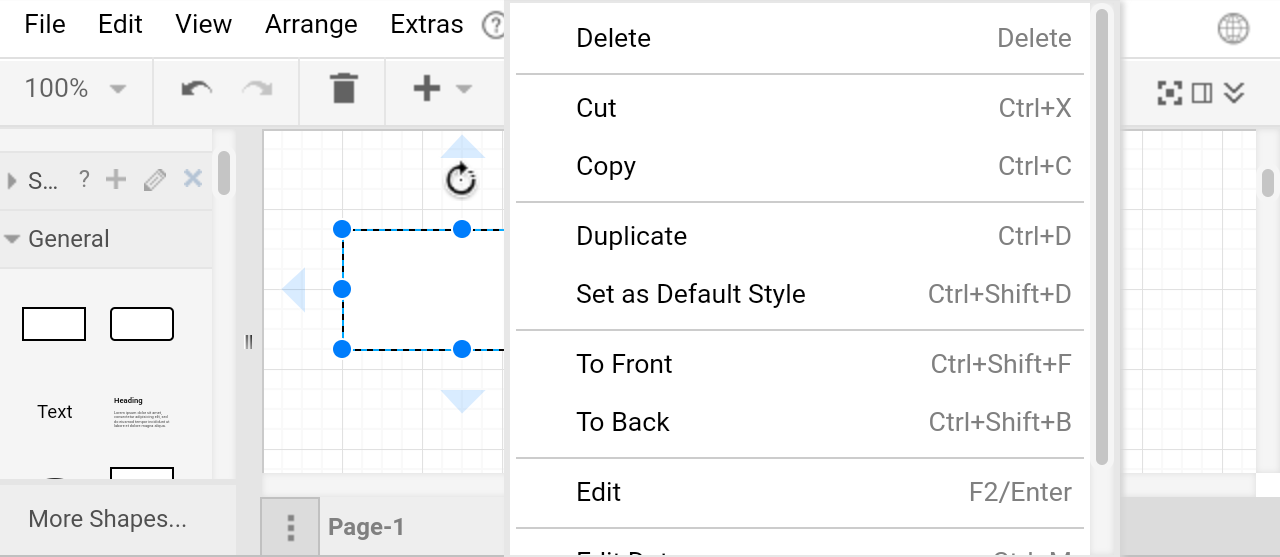
All’interno è possibile scrivere.

Per usare quest’opzione si clicca due volte sull’immagine.

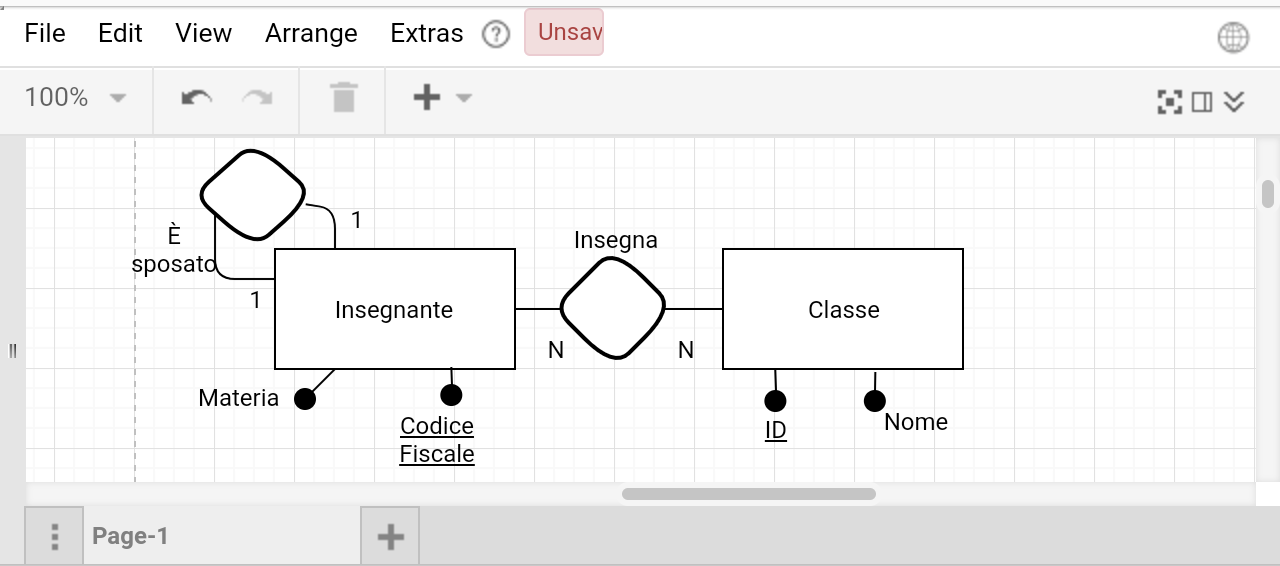
Si aprirà una barra con varie opzioni.

Per inserire il testo all’interno della figura si clicca l’opzione *Edit.*

Tra le opzioni della barre c’è eliminare, copiare, duplicare ecc.



*Esempio di esercizio inerente al paragrafo Le Relazioni.*

**

Questo esempio di database è stato creato usando i rettangoli, le linee e le caselle di testo della categoria *General.*

I rombi sono stati presi dalla categoria *Entity Relation*, mentre i simboli degli attributi in *UML.*

*Draw.io* è uno strumento semplice e molto funzionale che possono usare tutti.

# Lo schema relazionale

Chiavi esterne, attr. multipli che diventano tabelle

## Normalizzazione

# Implementazione di un database (gli strumenti)

# Interrogazioni di una base di dati

## Select From e Where

## Operazioni sugli attributi

## Aliasing

## Order by

## Limit

## Raggruppamento