Esercitazione 1: DB-MAIN e modello E/R

Sistemi Informativi L-B

Home Page del corso:
http://www-db.deis.unibo.it/courses/SIL-B/

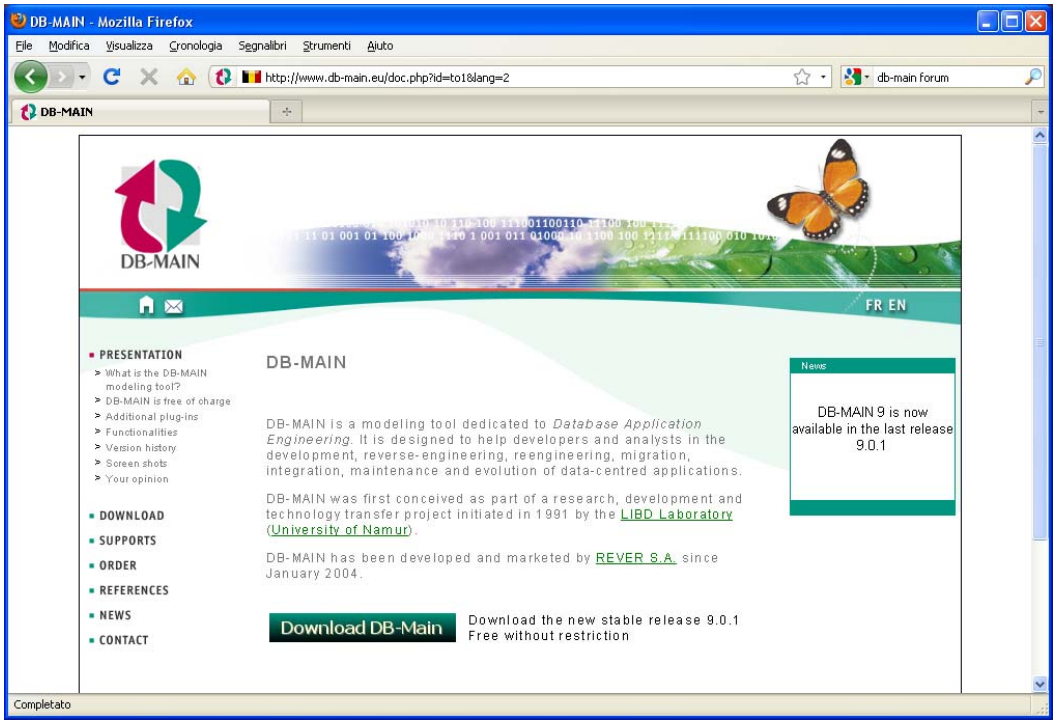
Versione elettronica: esercitazione1.pdf

DB-MAIN: uno strumento di progettazione

- DB-MAIN è un tool di modellazione orientato ai dati
- Supporta diversi processi legati all'ingegnerizzazione di DB, tra cui quello basato sulla metodologia del corso
- E' stato sviluppato a partire dal 1991 dal LIBD Laboratory della University of Namur (Belgio)
- Dal 2004 è stato commercializzato da REVER S.A.
- Dal 2009 è freeware, si pagano solo l'assistenza, la formazione e i plug-in aggiuntivi

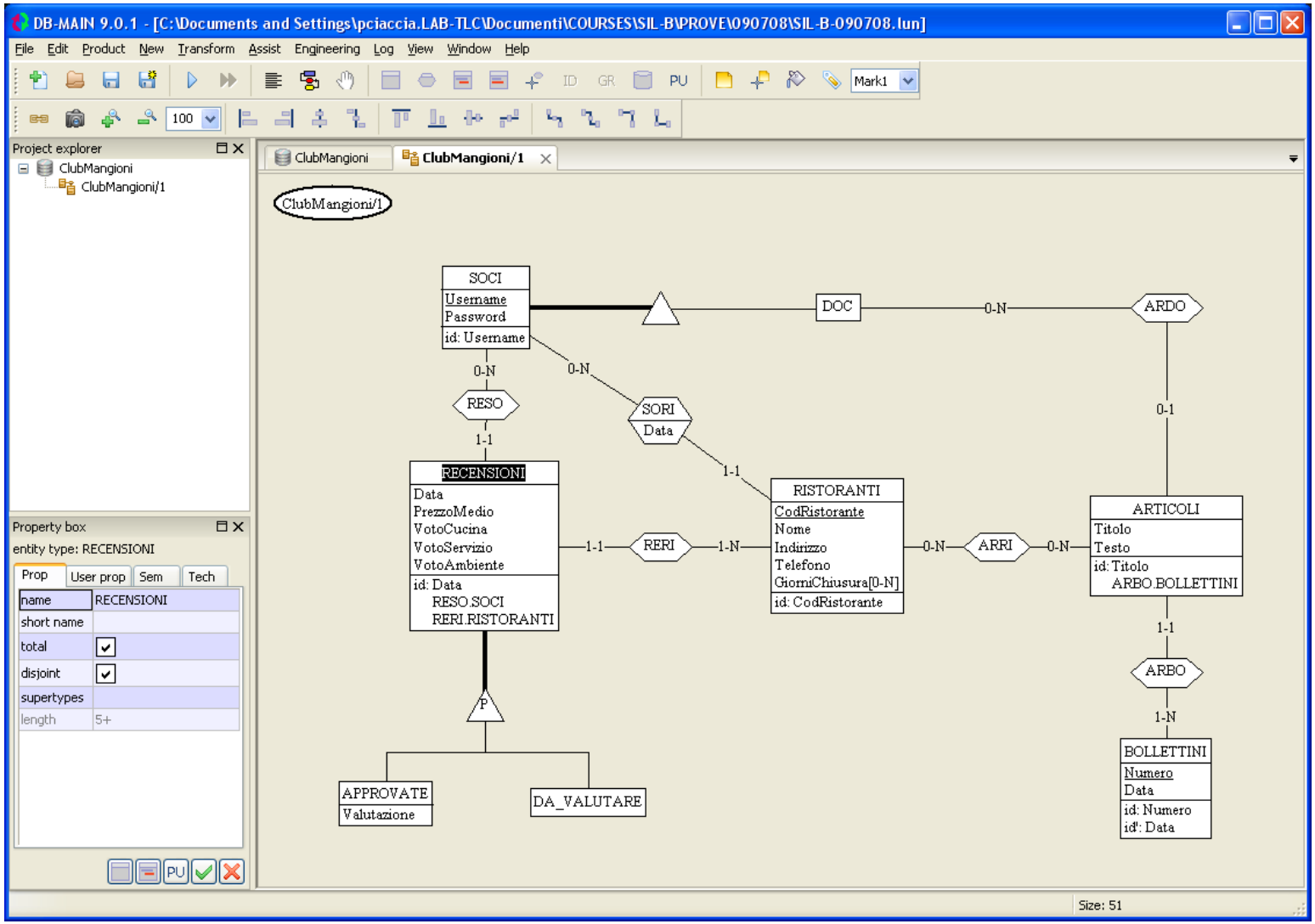
DB-MAIN: Documentazione

Home: http://www.db-main.eu

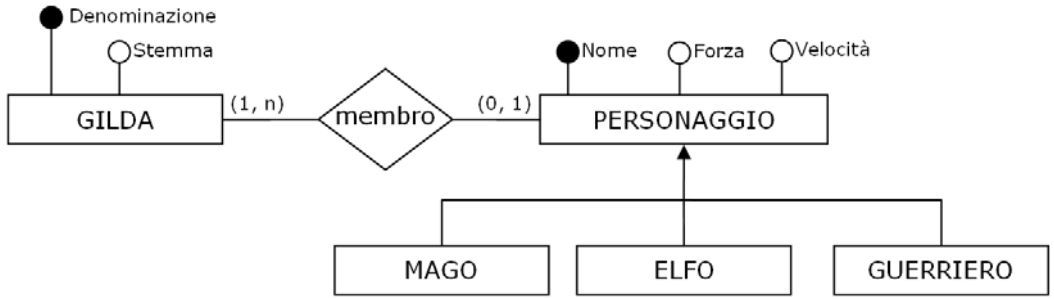


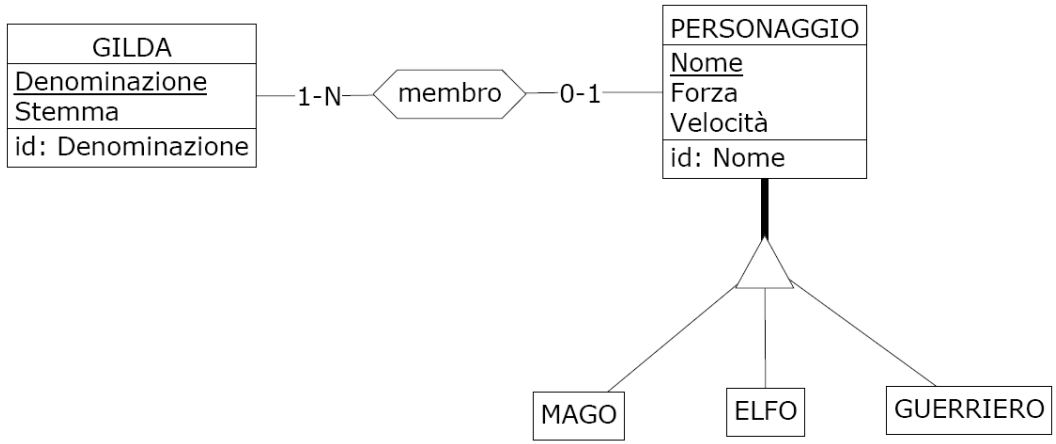
Reference Manual: http://www-db.deis.unibo.it/courses/SIL-B/DOCS/

DB-MAIN: uno strumento grafico



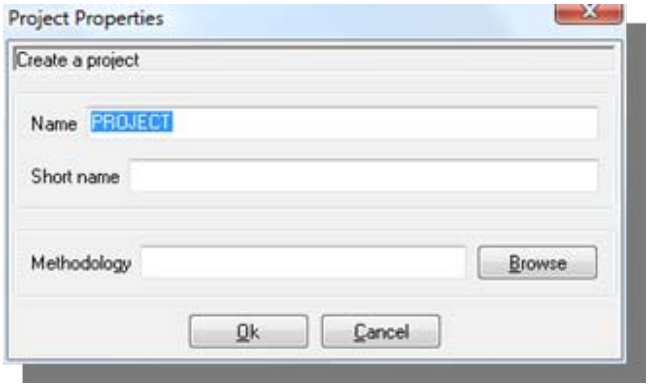
Diagrammi E/R in DB-MAIN

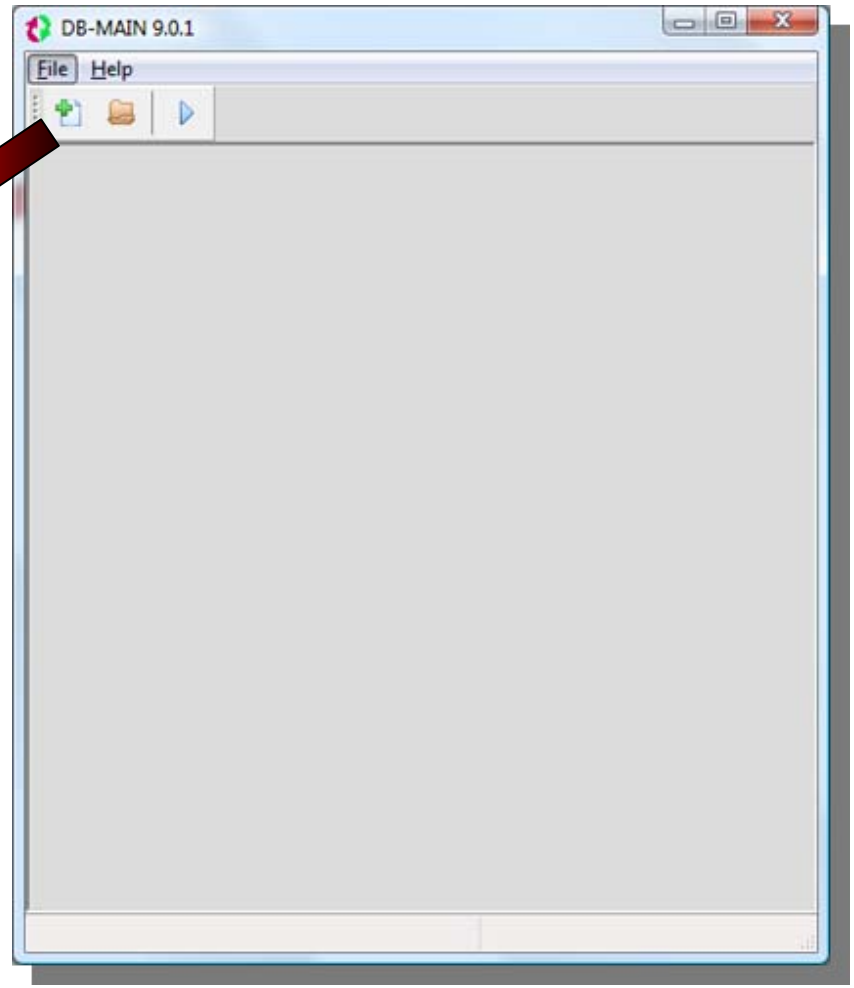




L’ambiente di lavoro

- Eseguire DB-main.exe
- Creare un nuovo progetto

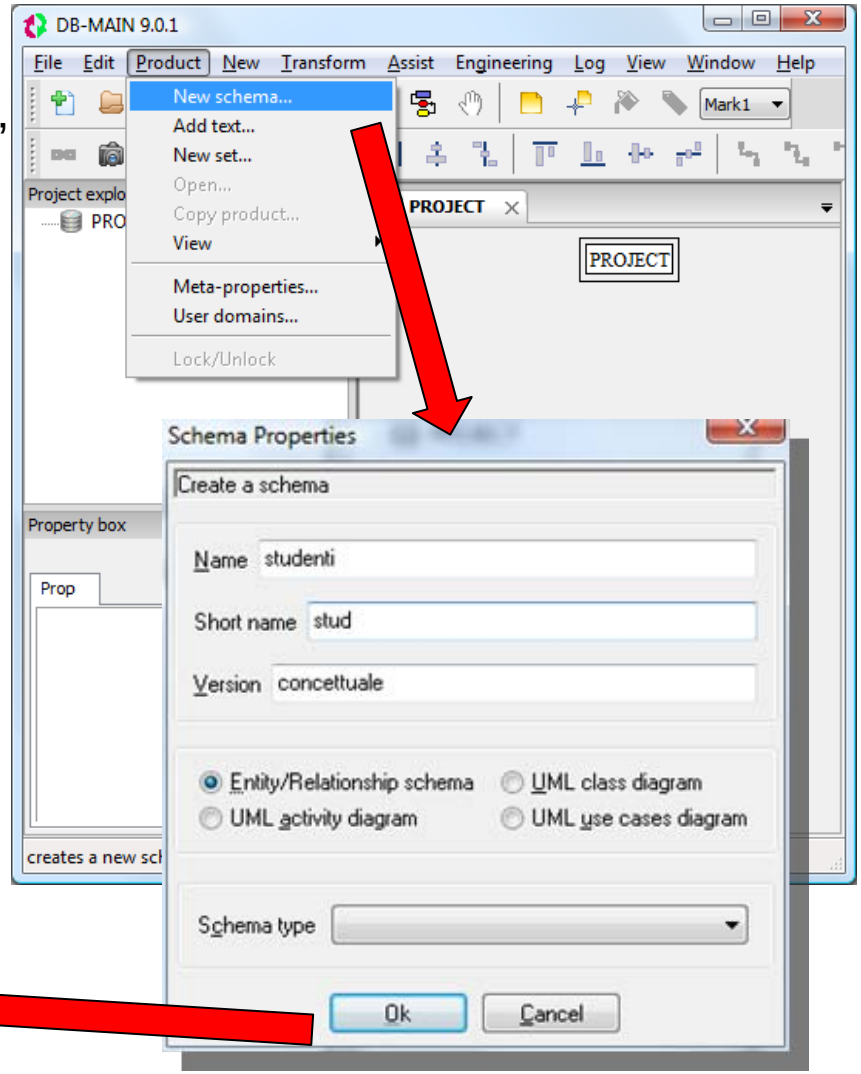


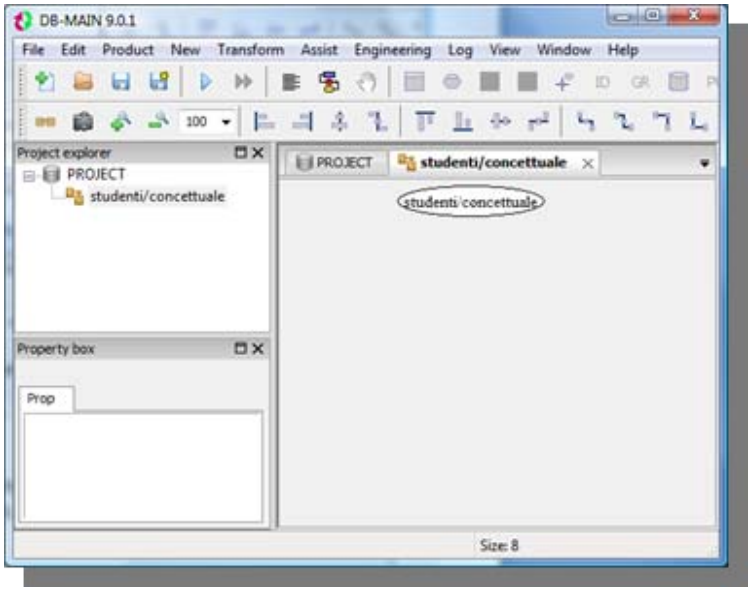


- o aprire un progetto esistente

Creazione schema concettuale

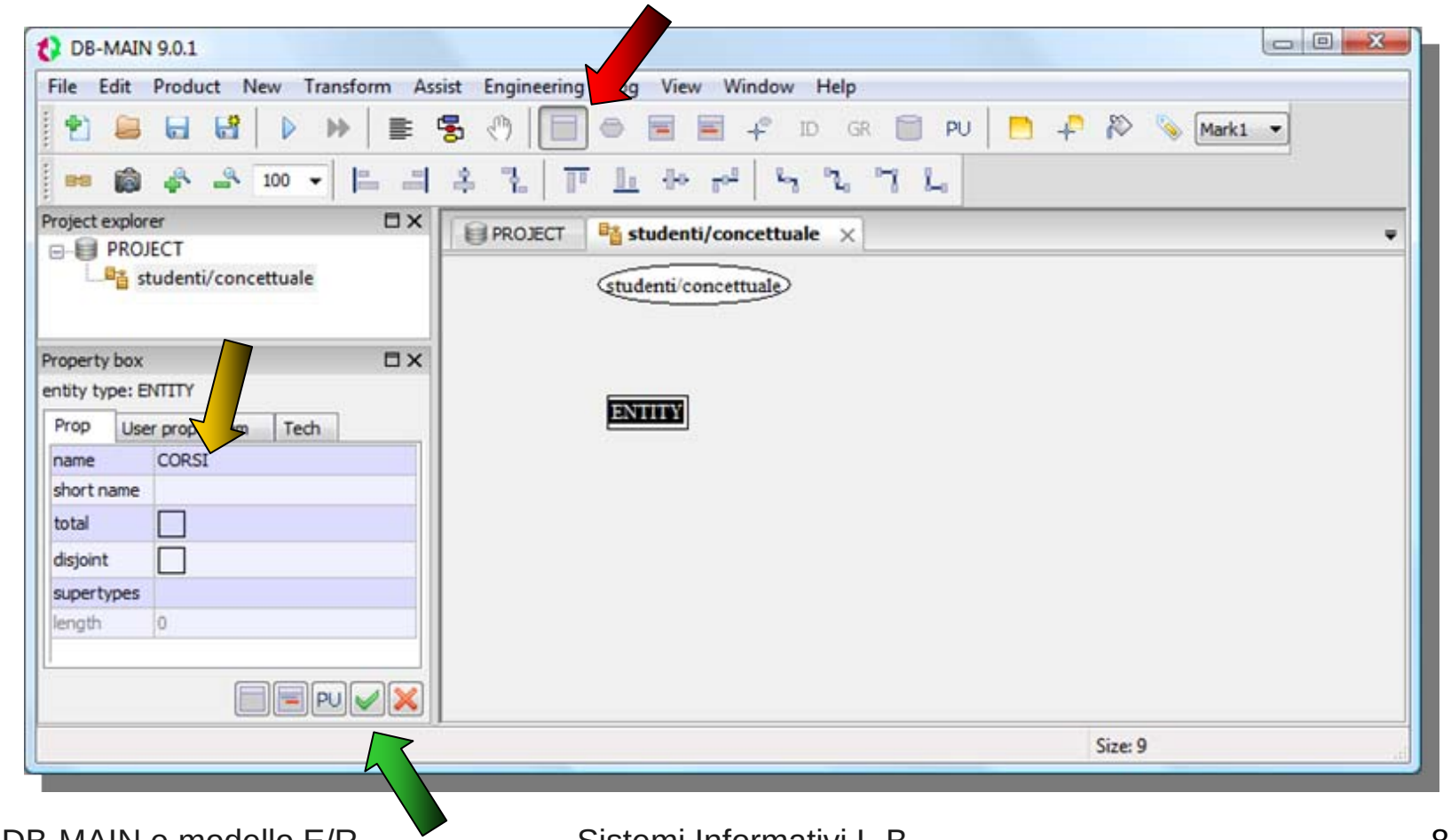
- Creazione di un nuovo schema:
  - Comando “Product/New Schema…”
  - Versione: Schema Concettuale
- Nel progetto viene aggiunta una linguetta in cui disegnare lo schema.





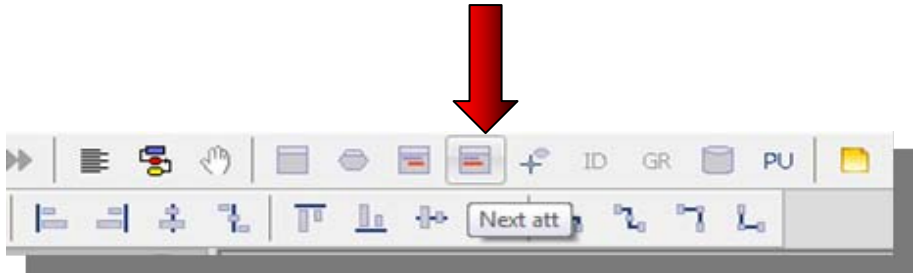
Definizione di entità

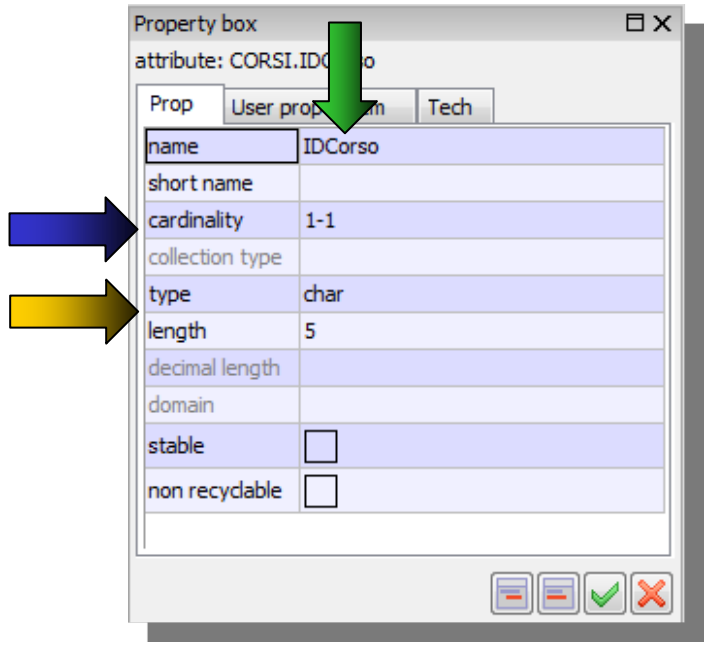
- Creazione di una nuova entità
- Definizione del nome
- Conferma



Definizione di attributi

- Selezionare l'entità a cui aggiungere l'attributo
- Sulla toolbar principale cliccare sul pulsante  ("Next att")
- Definizione del nome dell'attributo
- Definizione della cardinalità
  - La cardinalità di un attributo consente di esprimere se può assumere valore NULL
- Definizione del tipo

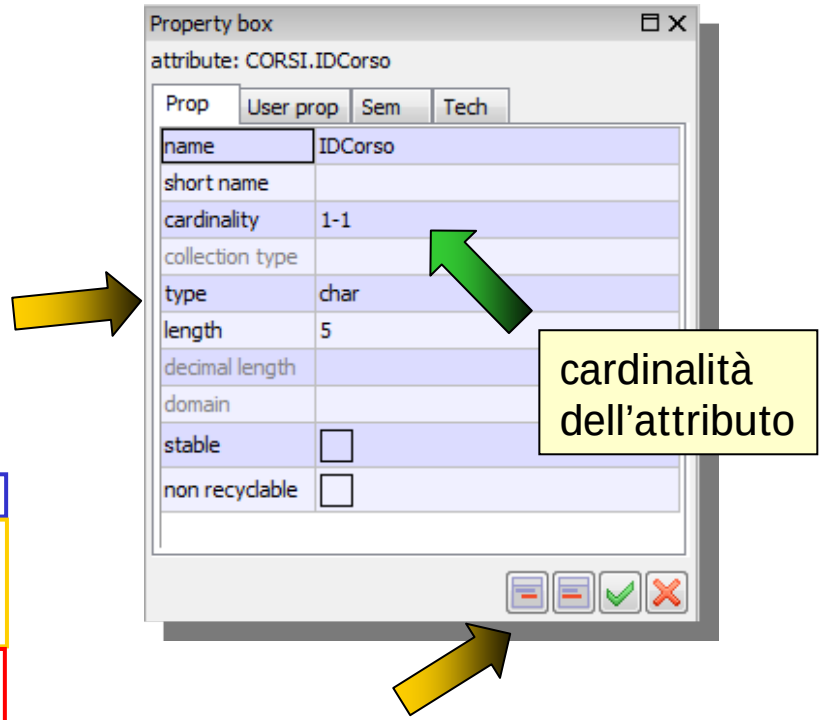




Definizione di attributi

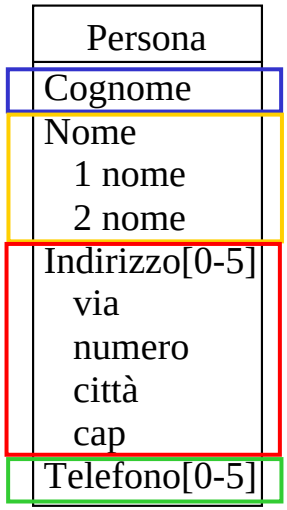
- Diversi tipi di attributi:
  - semplici
  - ripetuti
  - composti
  - composti e ripetuti

selezione del dominio su cui è definito l'attributo (type="compound" per attributi composti)



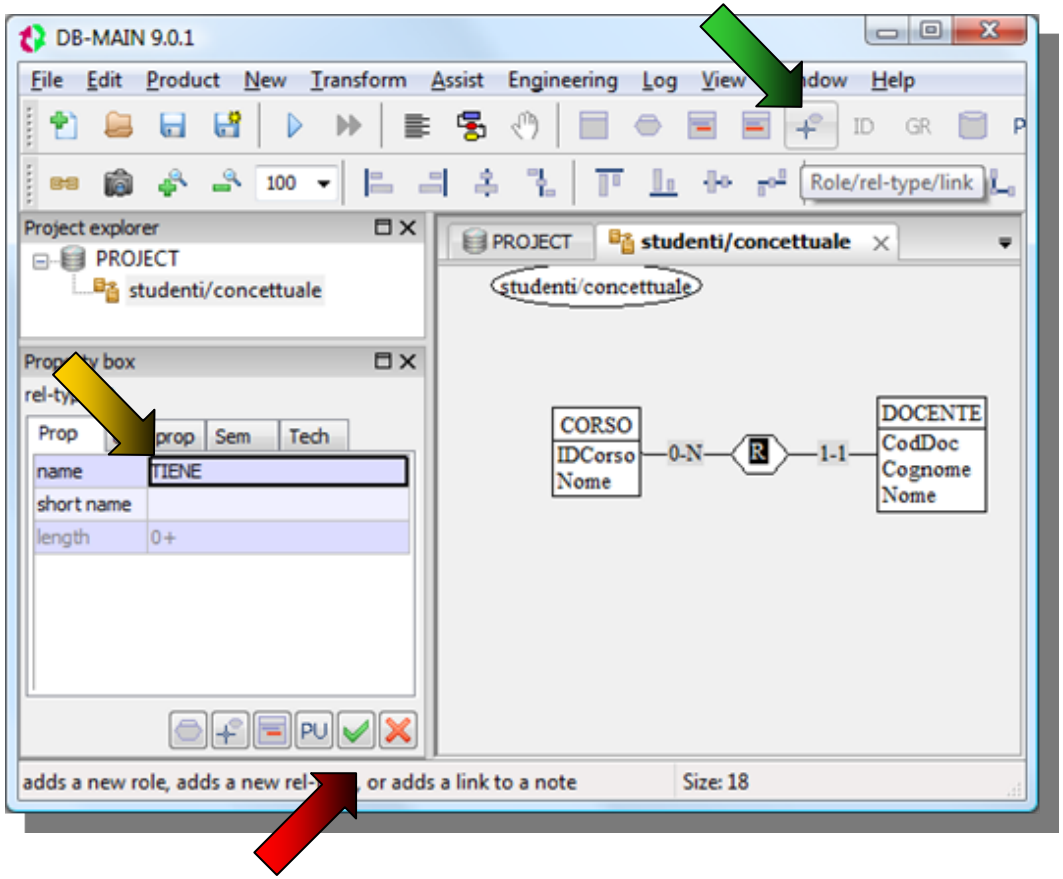
cardinalità dell'attributo

permette di aggiungere sotto-attributi agli attributi composti



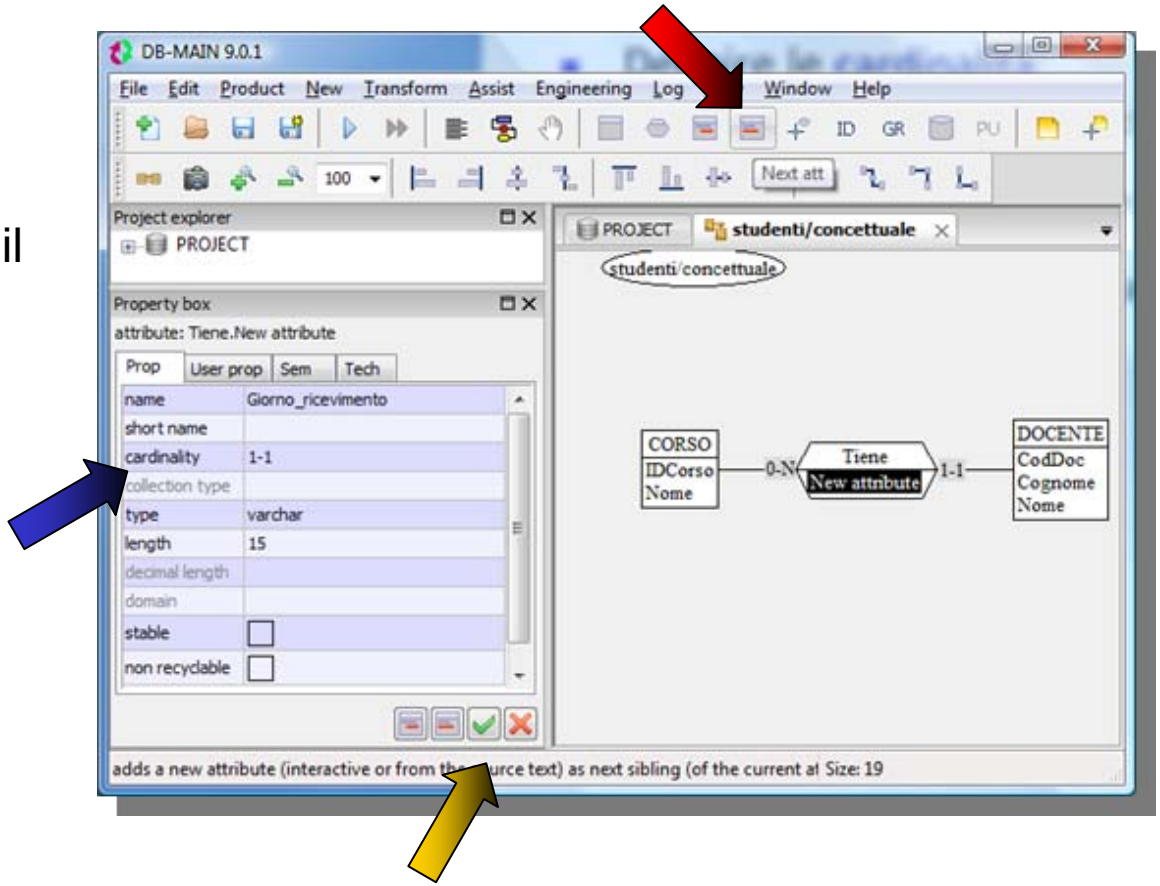
Definizione di associazioni (1)

- Selezionare lo strumento "*Role|rel-type/link*"
- Trascinare un collegamento tra le due entità coinvolte. Trascinando il collegamento dall'entità A all'entità B esse di default parteciperanno all'associazione rispettivamente con cardinalità (0, n) e (1, 1).
- Definire il nome dell'associazione



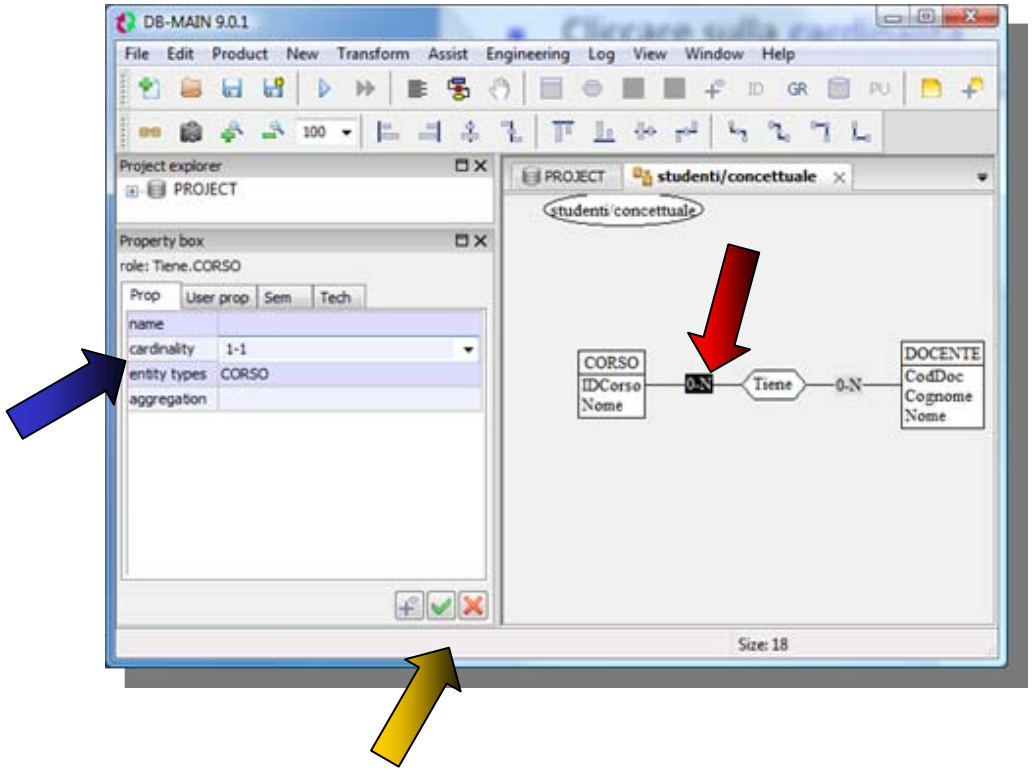
Definizione di associazioni (2)

- Per aggiungere eventuali attributi cliccare su 
- Definire le cardinalità degli attributi:
  - è sufficiente selezionare il valore desiderato dal menu a discesa nel relativo campo. Se il valore desiderato non è disponibile lo si può inserire da tastiera.



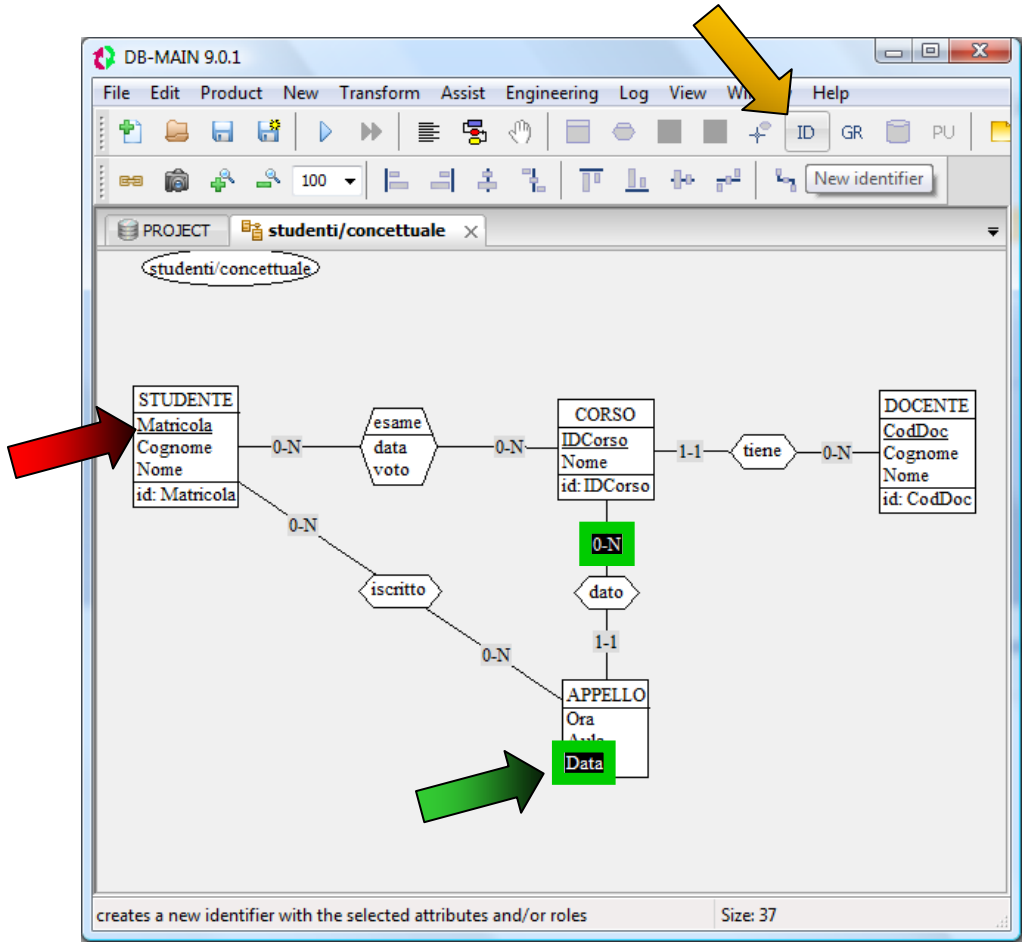
Definizione di associazioni (3)

- Per modificare la cardinalità delle partecipazioni:
- Cliccare sulla cardinalità dell'associazione e modificarla nel property box



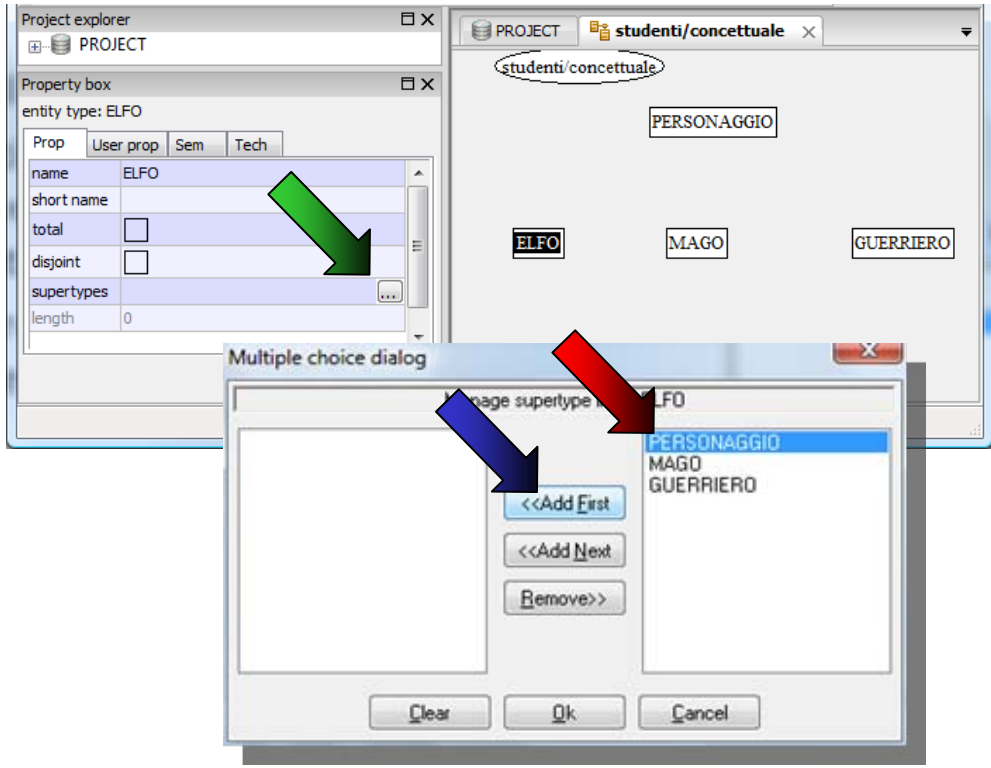
Definizione degli identificatori

- Identificatori:
  - Semplice: selezionare l'attributo
  - Composto interno: selezionare più attributi tenendo premuto il tasto shift
  - Esterno: selezionare gli attributi dell'entità e la cardinalità dell'associazione che partecipa all'identificatore
- Premere il pulsante 



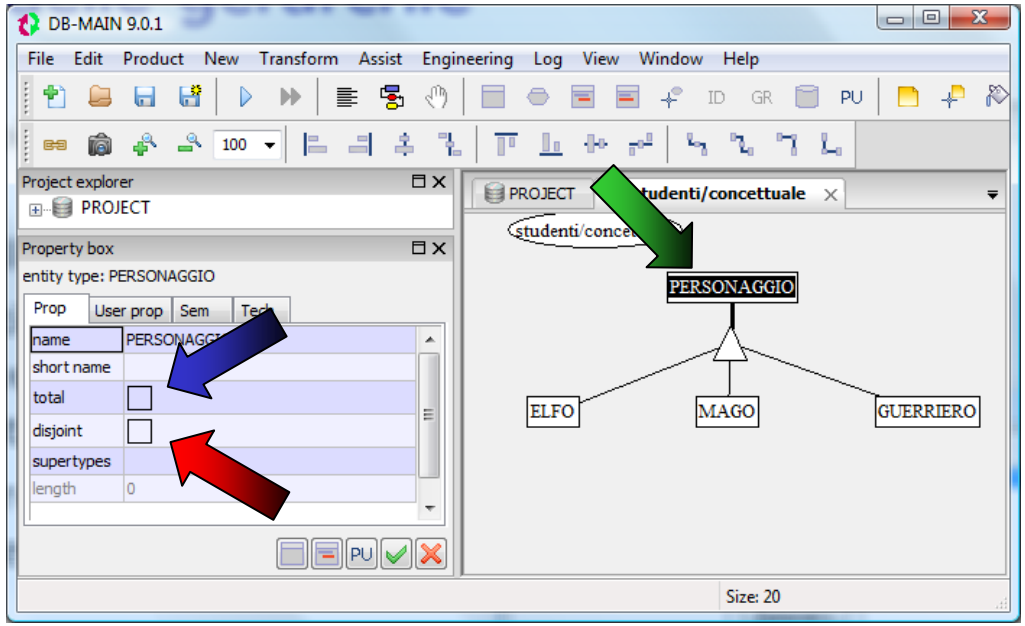
Definizione delle gerarchie

- Definire le entità coinvolte nella gerarchia
- Selezionare una delle entità figlie
- Nel property box utilizzare il pulsante  della voce Supertypes per aggiungere l'entità padre



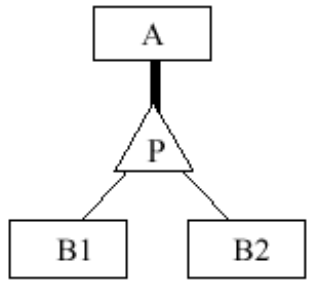
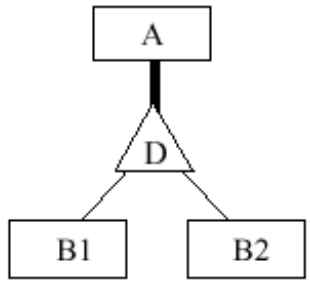
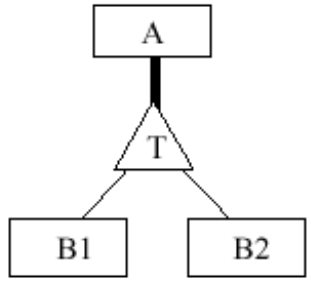
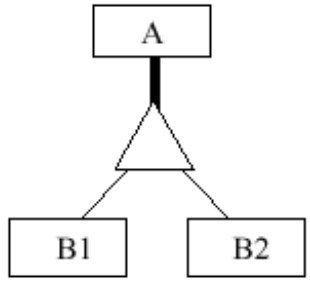
Definizione delle gerarchie

- È possibile esprimere la tipologia della gerarchia nelle proprietà dell'entità padre
  - totale-parziale (total)
  - esclusiva-sovrapposta (disjoint)



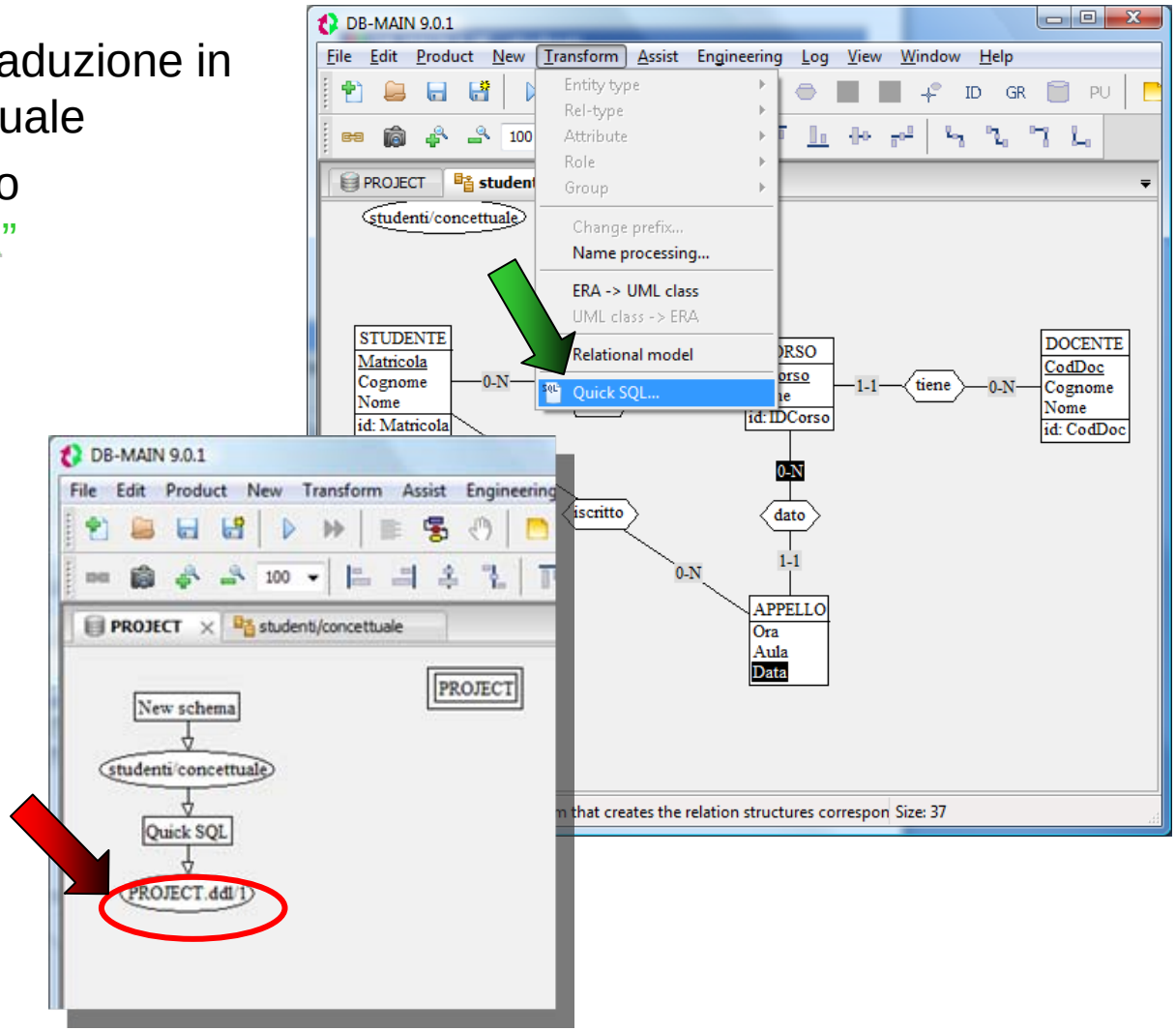
Tipi di gerarchie

- T = Total (totale e sovrapposta)
- P = Partition (totale ed esclusiva)
- D = Disjoint (parziale ed esclusiva)
- △ = (parziale e sovrapposta)

	Total (T)	**Partial (¬T)**
Disjoint (D)		
Overlapping (¬D)		

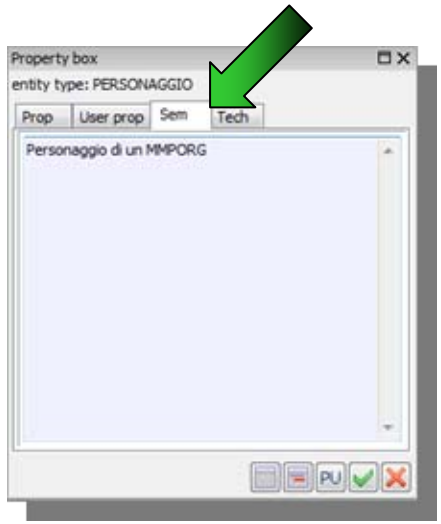
Creare il DB associato

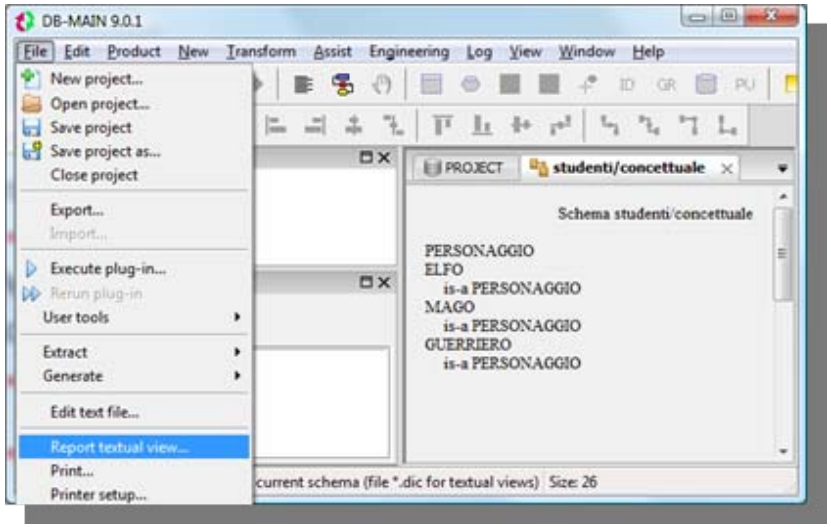
- È possibile ottenere una traduzione in SQL dello schema concettuale
  - Selezionare il comando "Transform/Quick SQL"
  - Salvare il file di testo
  - Il progetto contiene nel nuovo file le istruzioni SQL DDL per generare il DB

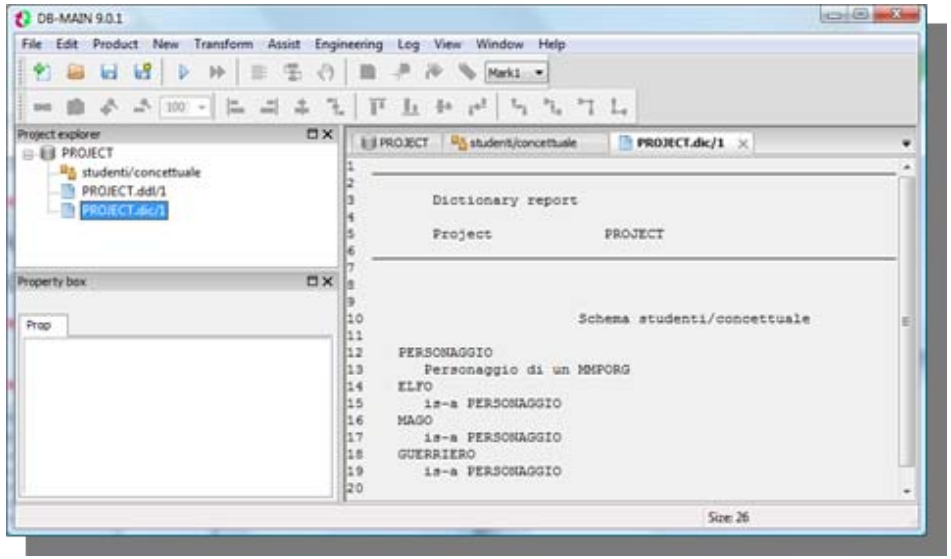


Documentare lo schema

- È possibile documentare lo schema inserendo un commento nella finestra Semantic Description associata a ciascun oggetto
  - Nella property box selezionare la linguetta “Sem”
- Per visualizzare il report dello schema comprendente le descrizioni:
  - Passare alla visualizzazione testuale con il menu View/Text Standard
  - Selezionare File/Report Textual view





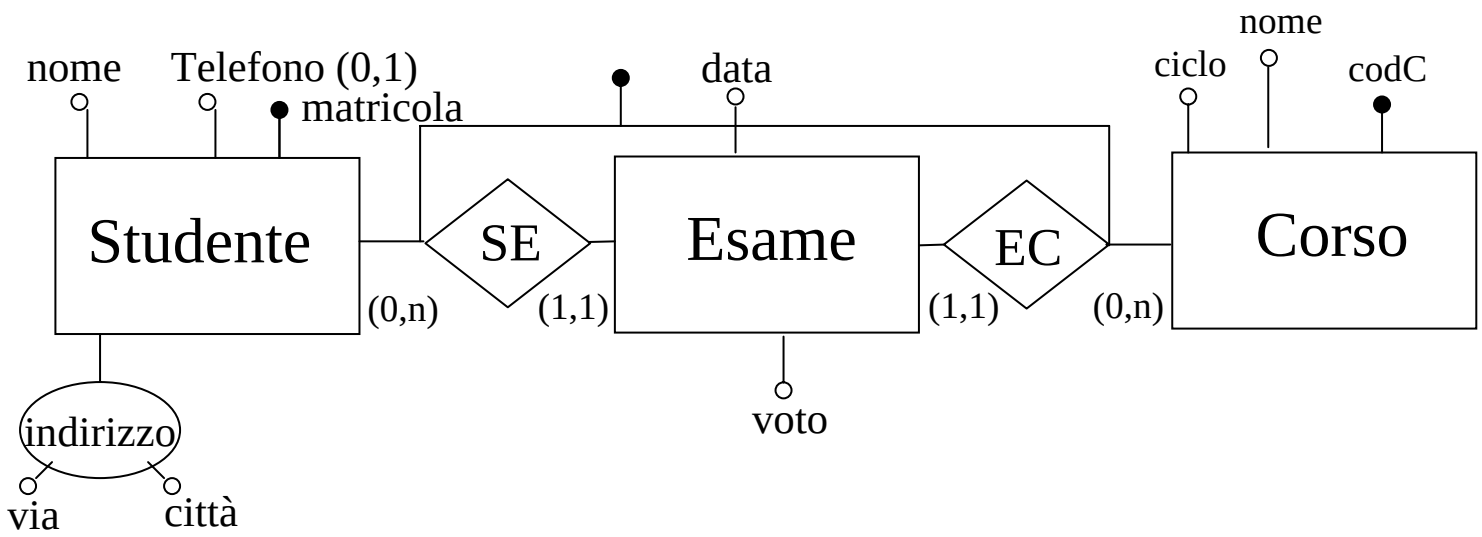


Tool Grafici

- Comandi grafici disponibili (“Window/Graphical Tools”)
  - Gli oggetti attualmente selezionati nella vista principale possono essere copiati negli appunti di Windows come immagine utilizzando il pulsante 
  - Allineamento degli oggetti grafici: 
  - Spostamento indipendente degli oggetti (voce “Independent” in “View/Graphical settings”)
  - Gerarchie “squadrate” (check-box “IS-A square” in “View/Graphical settings”)
- Utile da sapere:
  - Cambiamento ordine degli attributi: tasti ALT+↑
  - Tasto destro del mouse
    - su associazione: centratura rispetto alle entità
    - su cardinalità di un ramo dell’associazione: centratura rispetto ad associazione ed entità

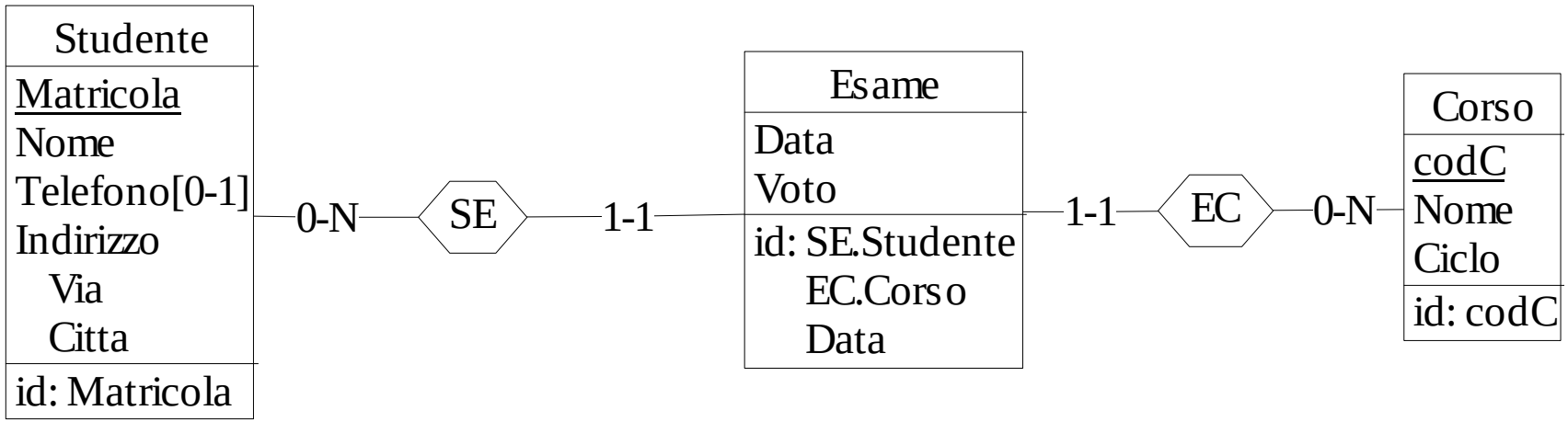
Generazione automatica del DDL (1)

Si consideri il seguente diagramma E/R…



Generazione automatica del DDL (2)

… e la relativa rappresentazione con DB-MAIN.



Generiamo automaticamente il codice SQL con il comando "Transform/Quick SQL" e controlliamo il codice generato…

Generazione automatica del DDL (4)

- Controlliamo il codice generato da DB-MAIN:

```
create database Studenti;

create table Studente (
     Matricola char(9) not null, Nome varchar(50) not null, Telefono char(10),
     Ind_Via varchar(50) not null, Ind_Citta varchar(20) not null,
     constraint ID_Studente_ID primary key (Matricola));

create table Esame (
     Matricola char(9) not null, codC char(4) not null, Data date not null,
      Voto numeric(2) not null, constraint ID_Esame_ID primary key (Matricola, codC, Data));

create table Corso (
     codC char(4) not null, Nome varchar(30) not null, Ciclo numeric(1) not null,
     constraint ID_Corso_ID primary key (codC));
```

Si noti come:

- La cardinalità 1-1 degli attributi sia tradotta come “not null” mentre gli attributi con cardinalità 0-1 permettono null (ad es. l’attributo Telefono di Studente);
- Gli attributi composti (ad es. l’attributo indirizzo) sono tradotti come diversi attributi aventi nome con prefisso comune;
- Il tipo per i valori numerici sia un generico tipo “numeric”;
- Non sia ancora specificata alcuna foreign key.

Generazione automatica del DDL (5)

- Dopo la definizione delle tabelle DB-MAIN aggiunge il codice per definire i constraint, tra cui le foreign key e i check:

```
alter table Esame add constraint FKEC_FK
     foreign key (codC)
     references Corso;

alter table Esame add constraint FKSE
     foreign key (Matricola)
     references Studente;
```

DB-MAIN aggiunge tutti i vincoli dopo aver creato le tabelle per evitare il problema delle foreign key “circolari”

Generazione automatica del DDL (6)

- Cosa accade cambiando il vincolo di cardinalità dell'attributo Telefono per l'entità studente trasformandolo in (1,3)? Il codice generato da DB-MAIN diventa:

```
create table Studente (Matricola char(9) not null, Nome varchar(50) not null,
   Ind_Via varchar(50) not null, Ind_Citta varchar(20) not null, constraint
   ID_Studente_ID primary key (Matricola));


create table Telefono (Matricola char(9) not null, Telefono char(10) not
   null, constraint ID_Telefono_ID primary key (Matricola, Telefono));


alter table Telefono add constraint FKStu_Tel foreign key (Matricola)
   references Studente;
alter table Studente add constraint ID_Studente_CHK check(exists(select *
   from Telefono where Telefono.Matricola = Matricola));
```

Attenzione! Il codice generato da DB-MAIN ha diversi problemi:

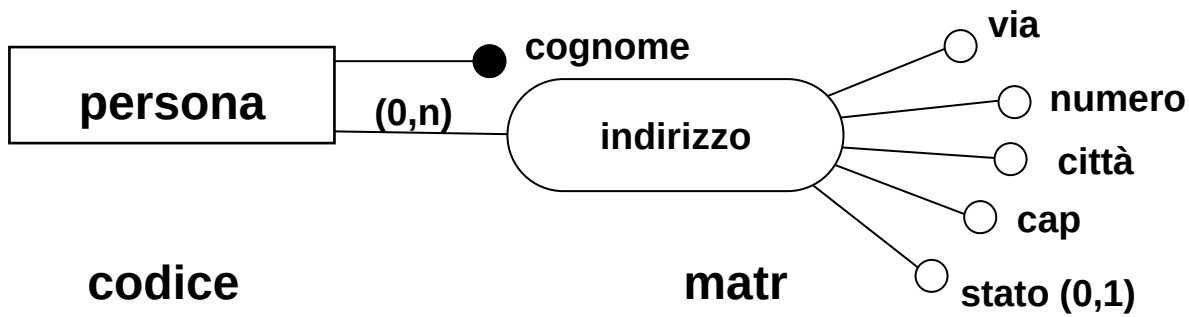
- non è valido per DB2 a causa del check complesso;
- introduce una dipendenza circolare tra Studente e Telefono senza possibilità di foreign key nulla;
- non esprime il vincolo di cardinalità massima dell'attributo Telefono.

Morale: non fidarsi ciecamente del codice di DB-MAIN!

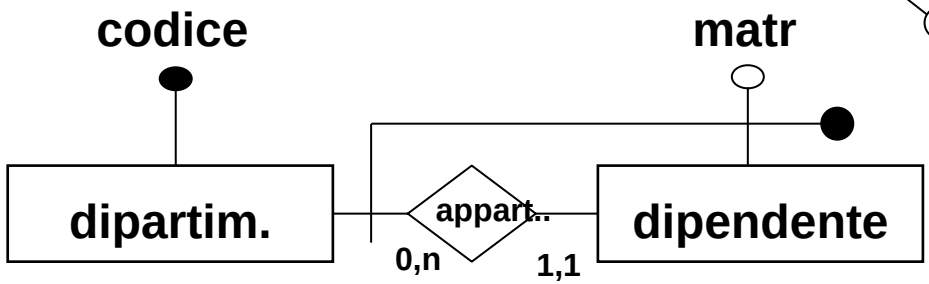
Primi esercizi con DB-MAIN

- Rappresentare con DB-MAIN i seguenti schemi E/R:

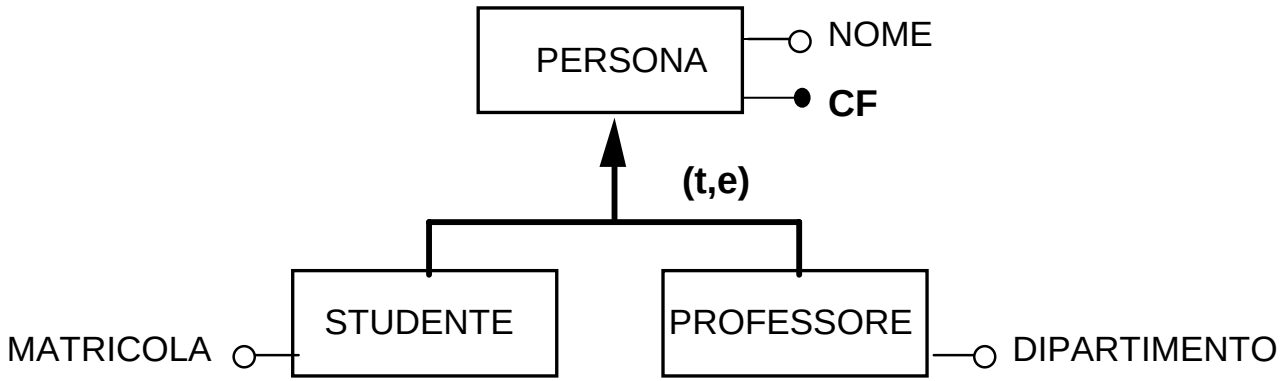
1.



2.

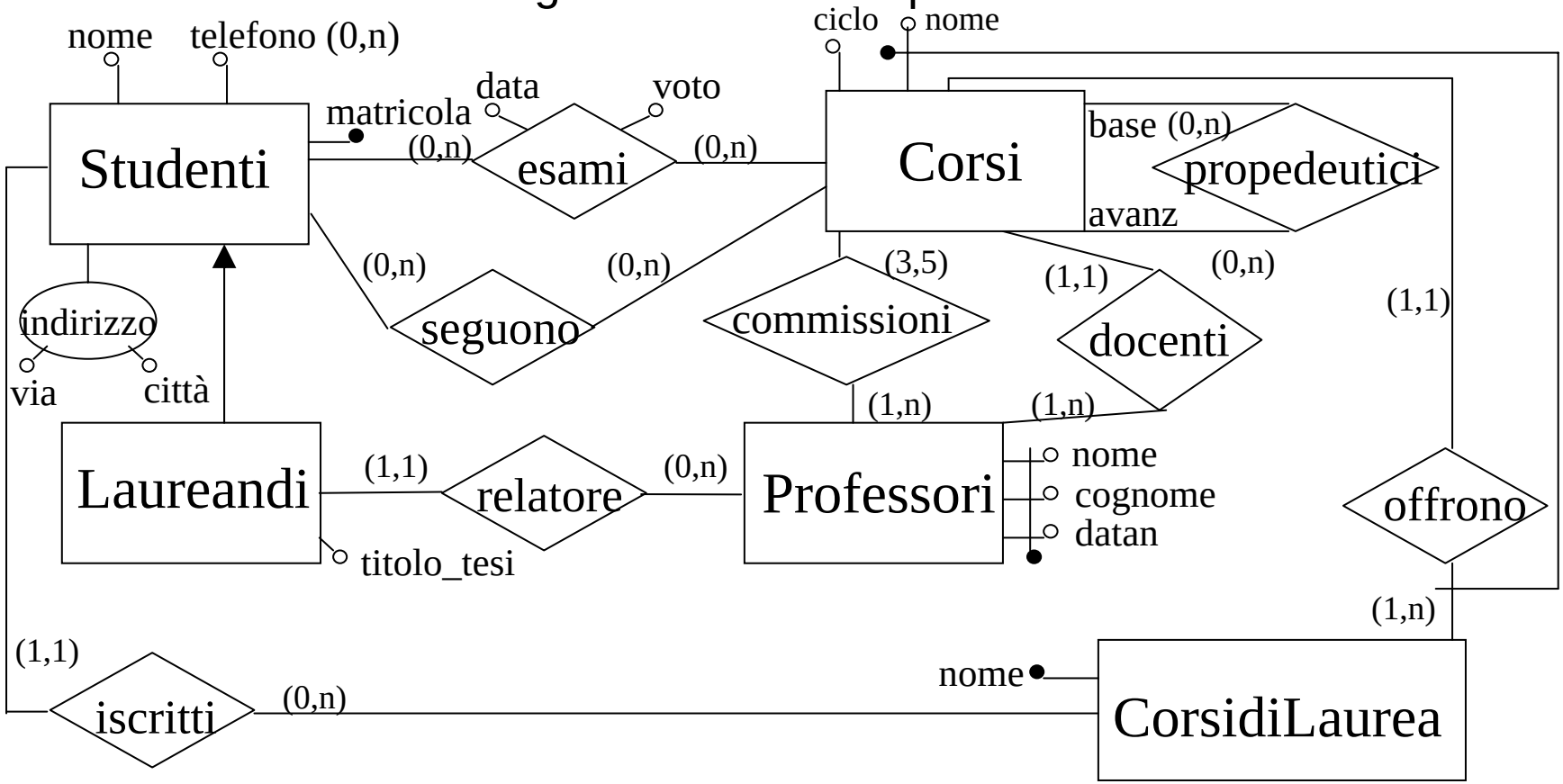


3.



Lo schema “Facoltà”

4. Creare lo schema che segue (progetto Facolta.lun)
   - generare il codice SQL e ragionare sul DB ottenuto
   - modificare il file .ddl e generare le corrispondenti table in DB2



Lo schema “Persone”

5. Rappresentare con DB-MAIN (progetto Persone.lun) lo schema derivante dalle seguenti specifiche:

Le persone hanno CF, cognome ed età; gli uomini anche la posizione militare; gli impiegati hanno lo stipendio e possono essere segretari, direttori o progettisti (un progettista può essere anche responsabile di progetto); gli studenti (che non possono essere impiegati) un numero di matricola; esistono persone che non sono né impiegati né studenti (ma i dettagli non ci interessano)

NB: In DB-MAIN non è possibile definire due o più gerarchie distinte con la stessa entità padre. Trovare una rappresentazione che permetta comunque di rappresentare le entità descritte e le loro relazioni di tipo gerarchico