

# Progetto d'Esame di Basi di Dati

Simone Alessandro Casciaro

[simonealessandro.casciaro@studenti.unimi.it](mailto:simonealessandro.casciaro@studenti.unimi.it)

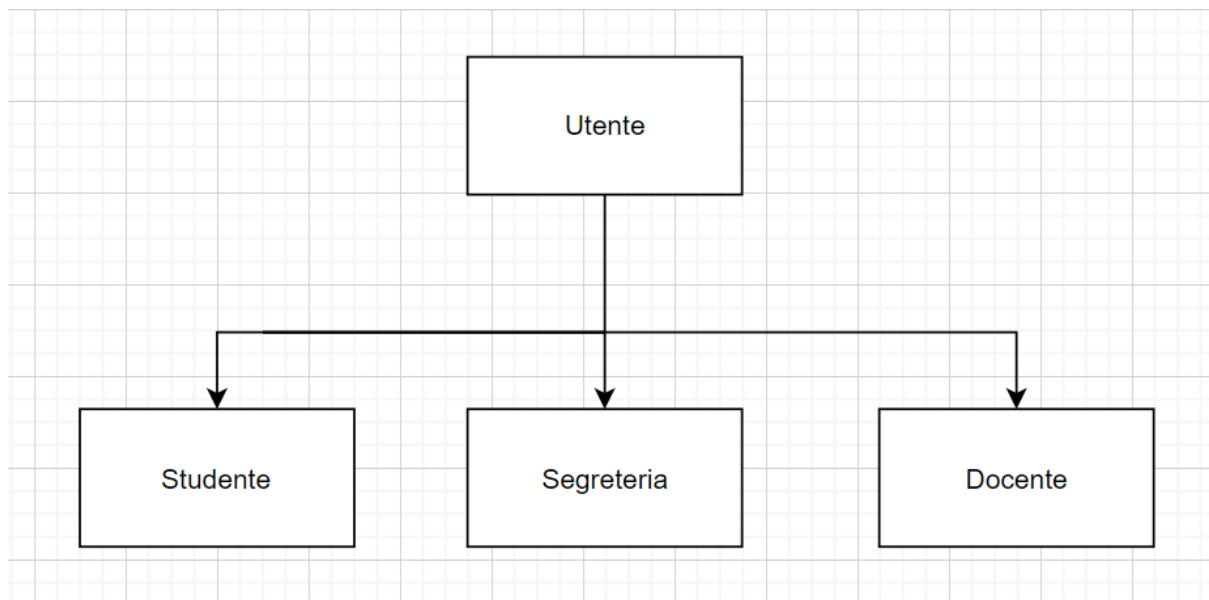
## Analisi dei Requisiti

L'analisi dei requisiti della base di dati parte dall'analisi dei 4 scenari principali posti dal progetto.

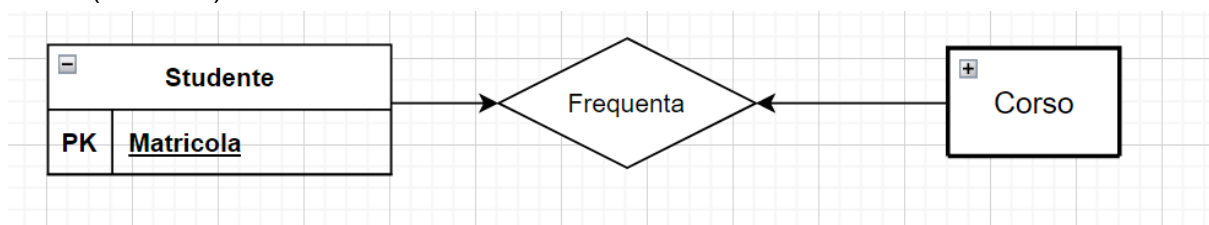
### Primo Macro-Scenario

Il primo macro-scenario prevede la categorizzazione di 3 utenti all'interno del database: Segreteria, Docente e Studente.

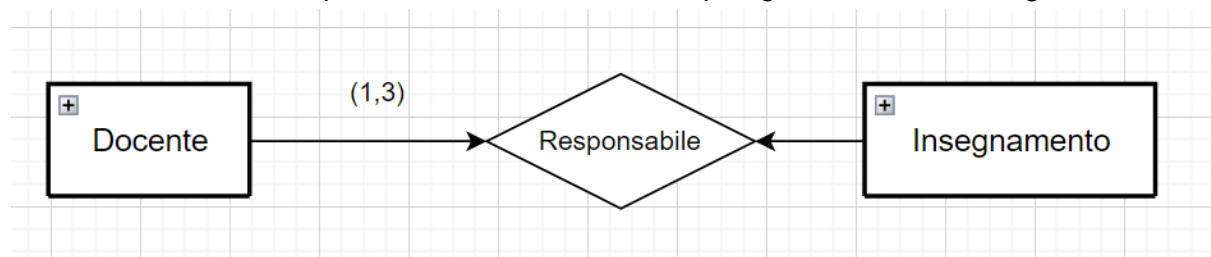
Possiamo strutturare questa richiesta con una gerarchia nel modello concettuale.



Nello stesso macro-scenario troviamo la necessità di assegnare ad ogni studente una matricola identificativa e, in base al corso di laurea, un insieme di esami da superare. Possiamo dunque stabilire una relazione tra l'entità Studente e una nuova entità chiamata Corso (di laurea).



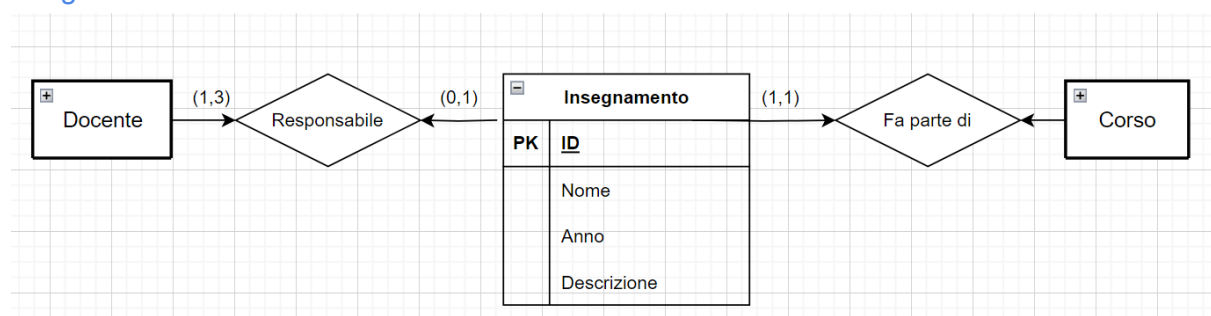
Ogni utente Docente, invece, deve avere almeno un insegnamento di cui è responsabile sino a un massimo di 3. Similmente, possiamo stabilire una relazione tra l'entità Docente e una nuova entità chiamata Insegnamento. Possiamo inoltre segnare le cardinalità, almeno dal lato del Docente, in quanto come detto il Docente può gestire da 1 a 3 insegnamenti.



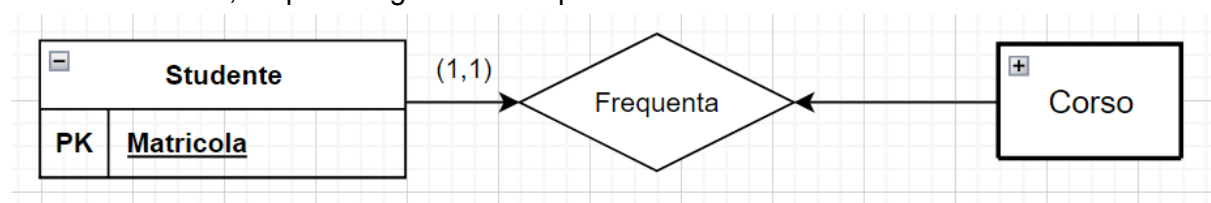
## Secondo Macro-Scenario

Nel secondo macro-scenario scopriamo che esiste una relazione tra l'entità Corso e l'entità Insegnamento, in quanto ogni insegnamento è caratterizzato da un codice univoco all'interno del corso di laurea, oltre che da un nome, una descrizione testuale e un anno previsto che dipende anche dalla durata del corso di laurea stesso (2 se il corso è magistrale, 3 se è triennale). Inoltre, ogni insegnamento dispone dell'indicazione del docente responsabile, il che ci permette di stabilire le cardinalità della relazione Docente - Insegnamento anche dal lato dell'entità Insegnamento.

**Nota:** per motivi di inserimento dati, si permette la possibilità di avere insegnamenti privi di docente responsabile. Sarà compito della segreteria assegnare un docente a tali insegnamenti.

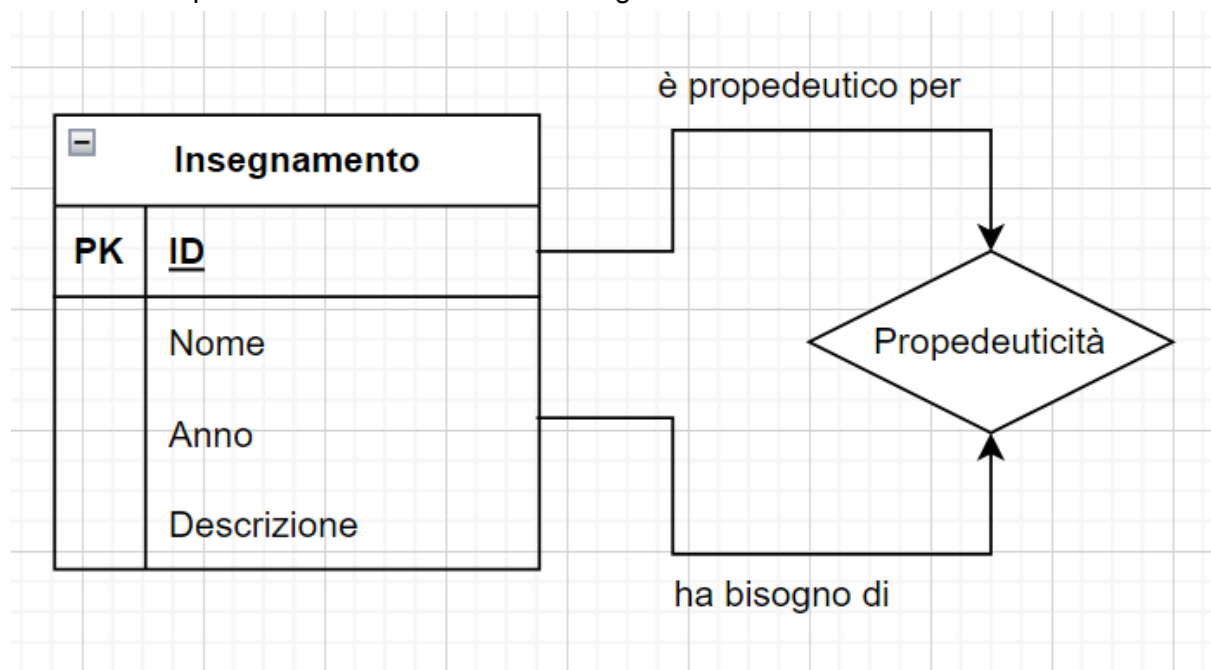


Nel secondo macro-scenario vengono inoltre specificate le cardinalità della relazione Studente - Corso, in quanto ogni studente può essere iscritto solo ad un corso di laurea.



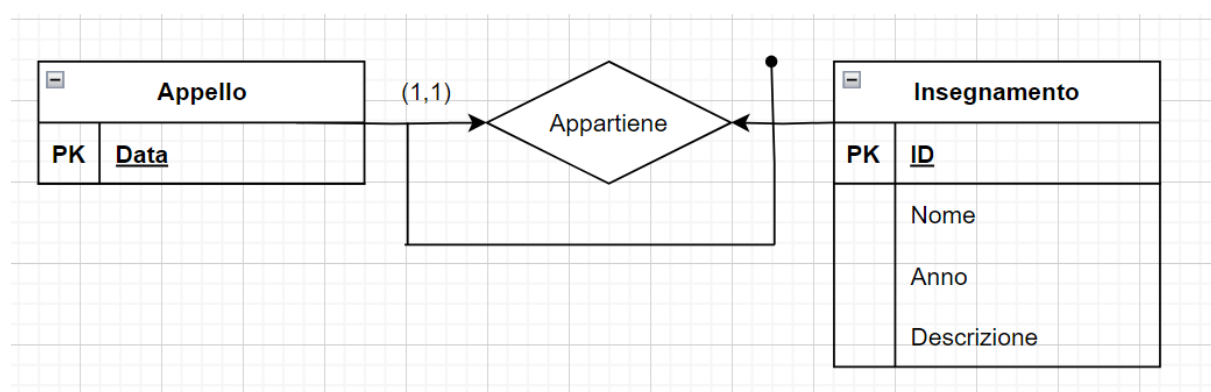
Per ogni insegnamento si possono definire delle propedeuticità, ovvero degli insegnamenti che devono essere superati tramite esami prima di potersi iscrivere all'esame di un altro insegnamento.

Questa è dunque una relazione tra l'entità Insegnamento e se stessa.



## Terzo Macro-Scenario

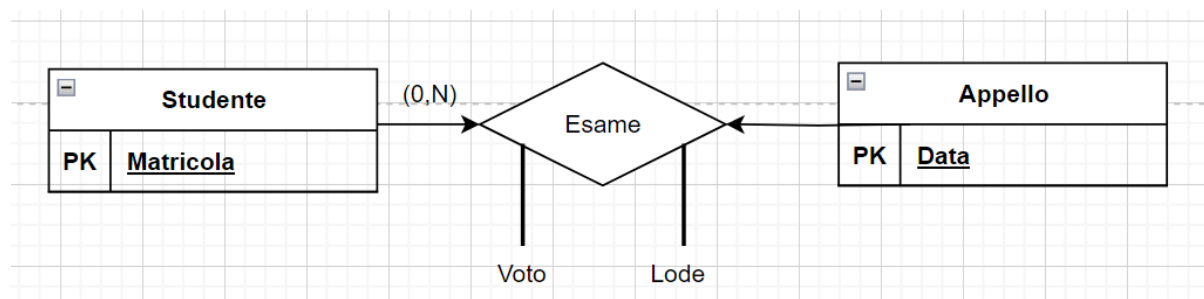
Il terzo macro-scenario prevede la gestione del calendario degli esami per gli insegnamenti. È possibile gestire tale richiesta attraverso una relazione tra l'entità Insegnamento e una nuova entità chiamata Appello, dove per "Appello" si intende la data in cui un docente intende collocare un esame.



**Nota:** Dal momento che la data, da sola, non è sufficiente ad identificare un appello, l'entità Appello sarà debole rispetto all'entità Insegnamento. Tale specificazione sarà importante nel momento in cui trasformeremo lo schema concettuale nel modello logico.

## Quarto Macro-Scenario

Il quarto macro-scenario si concentra sull'iscrizione agli esami da parte degli studenti. Usando le entità già introdotte in precedenza, tale richiesta può essere soddisfatta attraverso una relazione tra l'entità *Studente* e l'entità *Appello*.



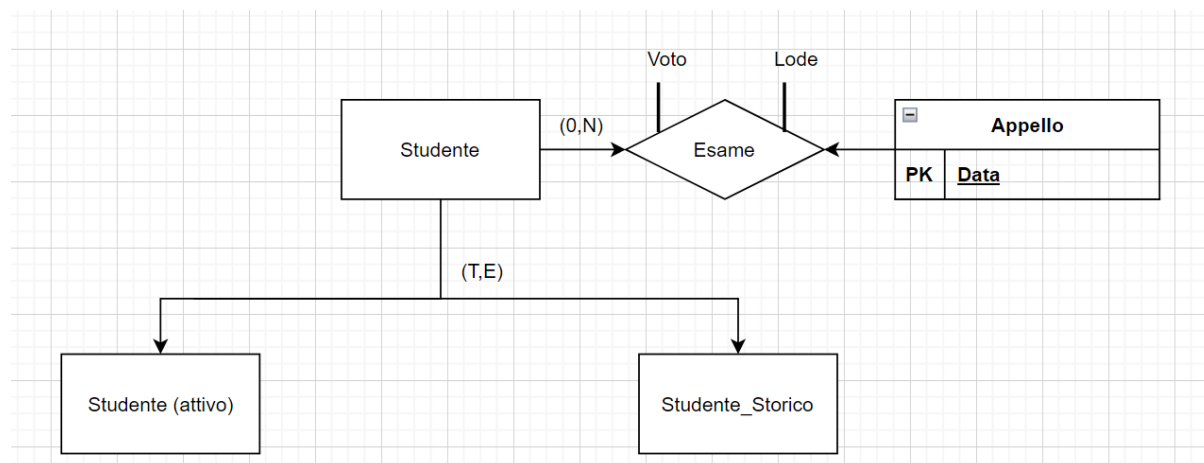
In questo modo, i voti che i docenti registrano diventano attributi della relazione *Esame*.

## Requisiti della Base di Dati

### Mantenimento delle informazioni e delle carriere di studenti rimossi

Tale richiesta rende necessaria la presenza di una nuova entità: *Studente\_Storico*, che terrà traccia di tutti gli studenti inattivi.

Dal momento che gli studenti possono essere attivi o inattivi, possiamo racchiudere le entità *Studente* e *Studente\_Storico* in una nuova gerarchia. La gerarchia sarà totale ed esclusiva, in quanto tutti gli studenti sono o attivi o inattivi e non possono essere entrambi nello stesso momento.



### Correttezza delle iscrizioni agli esami

Questa richiesta non aggiunge dettagli allo schema concettuale che si sta costruendo, tuttavia sarà necessario un Trigger che controllerà che uno studente si iscriva solo agli appelli degli insegnamenti previsti dal proprio corso di laurea.

## Correttezza del calendario d'esame

Questa richiesta non aggiunge dettagli allo schema concettuale che si sta costruendo, tuttavia sarà necessario un Trigger che controllerà che un docente non possa inserire appelli per un esame nello stesso giorno in cui un altro esame dello stesso anno e dello stesso corso è presente.

## Produzione della carriera completa di uno studente

Questa richiesta non aggiunge dettagli allo schema concettuale che si sta costruendo, tuttavia sarà necessaria una funzione che, presa in analisi la matricola di uno studente, possa riprodurre tutti gli esami a cui si è iscritto, con associati i voti e le date.

## Produzione della carriera valida di uno studente

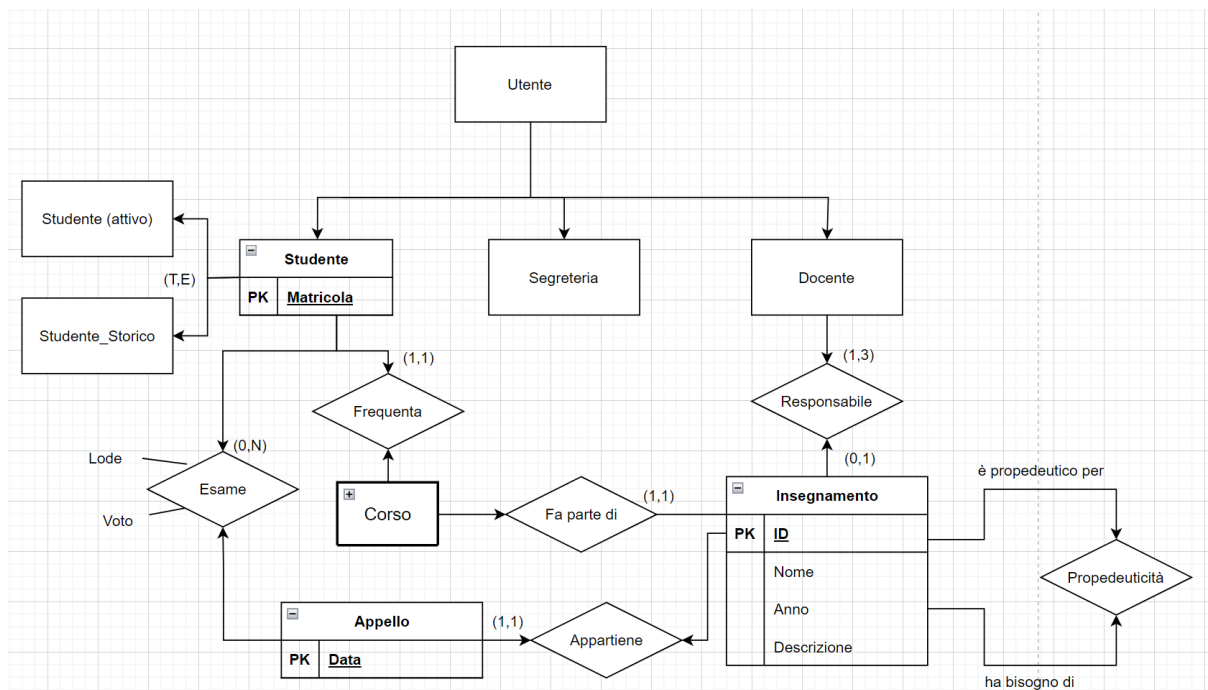
Questa richiesta non aggiunge dettagli allo schema concettuale che si sta costruendo, tuttavia sarà necessaria una funzione che, presa in analisi la matricola di uno studente, possa riprodurre tutti gli esami che lo studente ha superato, contando solo gli appelli più recenti per ogni insegnamento e tali per cui i voti siano maggiori o uguali a 18.

## Produzione delle informazioni su un corso di laurea

Questa richiesta non aggiunge dettagli allo schema concettuale che si sta costruendo, tuttavia sarà necessaria una funzione che, preso in analisi il nome di un Corso di laurea, possa riprodurre tutte le informazioni di ogni insegnamento che lo compone, tra cui le descrizioni e il docente responsabile.

# Ulteriori Informazioni

Alla luce dei requisiti sopracitati, lo schema concettuale assume questo aspetto:



Reputo opportuno, tuttavia, aggiungere qualche informazione alle entità presenti. Ad esempio:

- Per ogni utente (e di conseguenza ogni Studente, ogni Segretario e ogni Docente) possiamo salvare informazioni quali username, la password (necessarie poi per il login alla pagina web), il nome, il cognome, la data di nascita, il sesso e l'indirizzo di casa. Per gli studenti, inoltre, possiamo memorizzare anche la data di iscrizione all'università. In questo modo, possiamo scegliere l'username come chiave primaria dell'entità Utente. Gli studenti, tuttavia, invece dell'username avranno la matricola come chiave primaria.
- Per gli studenti chiamati storici, possiamo memorizzare nel database anche la data di inattività (ovvero il giorno in cui lo studente smette di essere iscritto all'università) e la motivazione per cui è divenuto inattivo, che può essere conseguimento della laurea o rinuncia agli studi.
- Per l'entità Corso, possiamo memorizzare il nome del corso, la durata (2 se il corso è Magistrale, 3 se è Triennale) e una sua descrizione testuale. Dal momento che non possono esistere due corsi di laurea con lo stesso nome all'interno dell'università, possiamo considerare il nome come chiave primaria.
- All'entità Appello possiamo inserire anche il luogo, in modo tale da rendere disponibile agli studenti che si iscrivono agli esami tale informazione.
- In fine, all'entità Insegnamento, oltre ai già presenti ID, nome, anno e descrizione, possiamo aggiungere il numero di CFU di cui l'insegnamento è composto (utile poi per calcolare il totale dei CFU conseguiti dagli studenti e la media ponderata per le loro carriere).

## **Considerazioni sulle cardinalità delle relazioni**

### **Relazione “Frequenta” (Studente - Corso)**

Ogni studente, come esplicitato nelle specifiche, può (e deve) essere iscritto ad un solo corso di laurea. Dunque (1,1)

Ogni corso può avere numerosi studenti iscritti. Può averne anche 0, nel caso di un corso appena istituito. Dunque (0,N)

### **Relazione “Esame” (Studente - Appello)**

Ogni studente può iscriversi a tutti gli appelli che vuole (purché chiaramente appartenenti al suo corso di laurea). Può anche decidere di non iscriversi a nessuno. Dunque (0,N)

Ad ogni appello possono iscriversi numerosi studenti, così come nessuno. Dunque (0,N)

Anche gli attributi “Voto” e “Lode” sono opzionali. In quanto, nel momento in cui uno studente si iscrive all'esame e il docente non ha ancora corretto la sua prova, il voto (e l'eventuale lode) non sono ancora assegnati.

### **Relazione “Fa parte di” (Corso - Insegnamento)**

Ogni insegnamento può (e deve) appartenere ad uno e uno solo corso di laurea. Dunque (1,1)

Ogni corso può essere composto da più insegnamenti. Anche da nessuno, in quanto nel database si dovrà prima inserire il corso e poi tutti gli insegnamenti che ne fanno parte.

Dunque (0,N)

### **Relazione “Appartiene” (Appello - Insegnamento)**

Ogni appello può (e deve) appartenere ad un solo insegnamento, soprattutto vista la decisione di rendere l'appello una entità debole rispetto all'insegnamento. Dunque (1,1)

Ogni insegnamento può avere più appelli, così come nessuno. Dunque (0,N)

### Relazione “Responsabile” (Docente - Insegnamento)

Da specifiche, ogni docente deve avere da 1 a 3 insegnamenti di cui è responsabile.

Dunque (1,3)

Ogni insegnamento può essere associato ad al massimo un docente. Come espresso nelle note del secondo macro-scenario, si permette all’insegnamento di essere privo di docente.

Dunque (0,1)

### Relazione “Propedeuticità” (Insegnamento - Insegnamento)

Ogni insegnamento può essere propedeutico per molti insegnamenti, così come nessuno.

Dunque (0,N)

Allo stesso modo, ogni insegnamento può “aver bisogno” di molti insegnamenti, così come nessuno. Dunque (0,N)

Alla luce di queste considerazioni, lo schema concettuale appare in questo modo:

