

# PROGETTO BASI DI DATI

Luca Carone Poletti

Mat:05099A

Mail: [luca.caronepoletti@studenti.unimi.it](mailto:luca.caronepoletti@studenti.unimi.it)

## Sommario

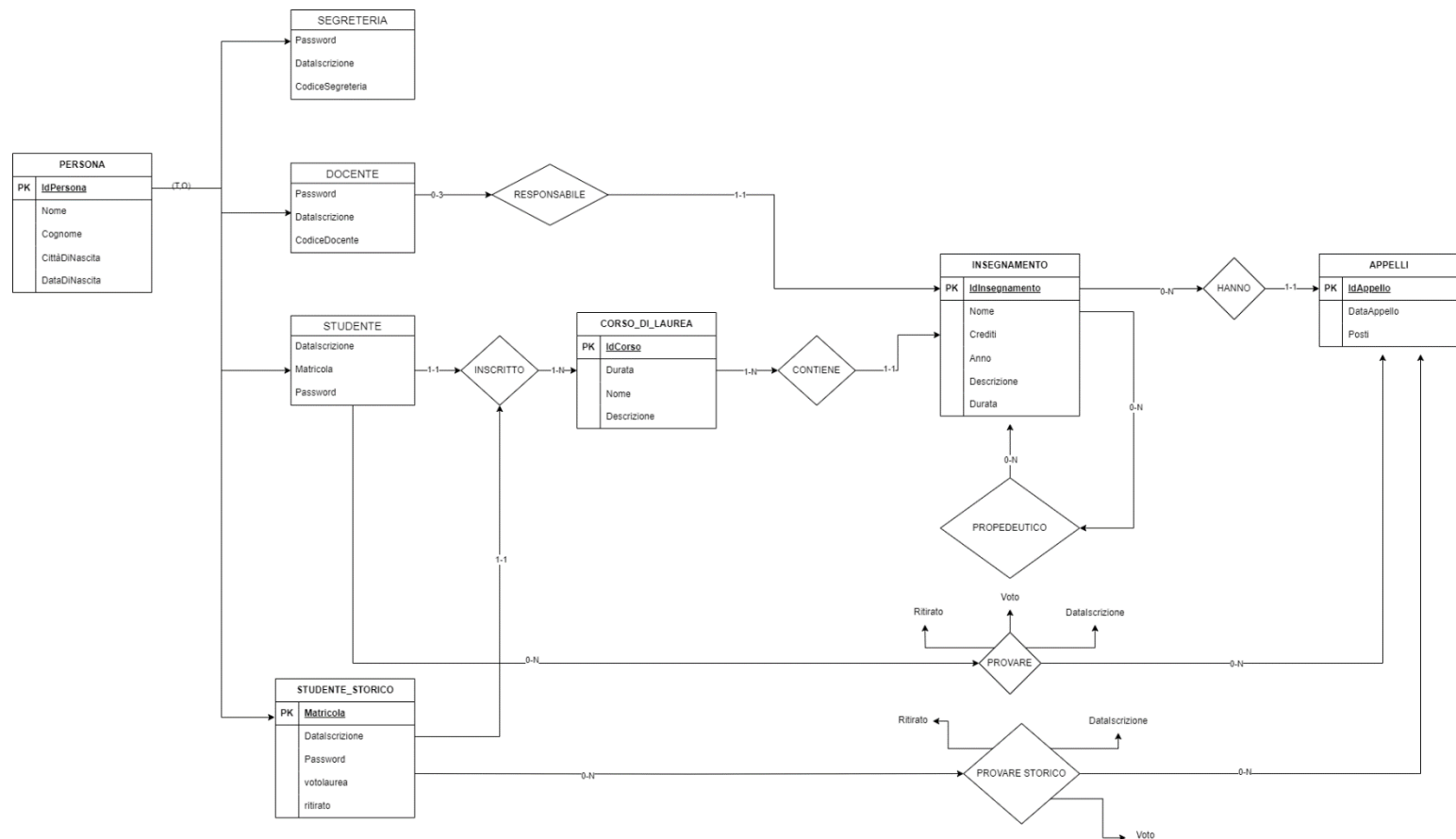
INTRODUZIONE.....	2
MODELLO ER.....	2
MODELLO ER-V2 .....	3
MODELLO LOGICO: .....	3
TRIGGER - CHECK - FUNCTION.....	4

## INTRODUZIONE

Il progetto si pone l'obiettivo di sviluppare un'applicazione di basi di dati per la gestione degli esami universitari dal punto di vista di:

- DOCENTI
- STUDENTI
- SEGRETERIA

## MODELLO ER



Andiamo ad analizzare in specifico i vari componenti:

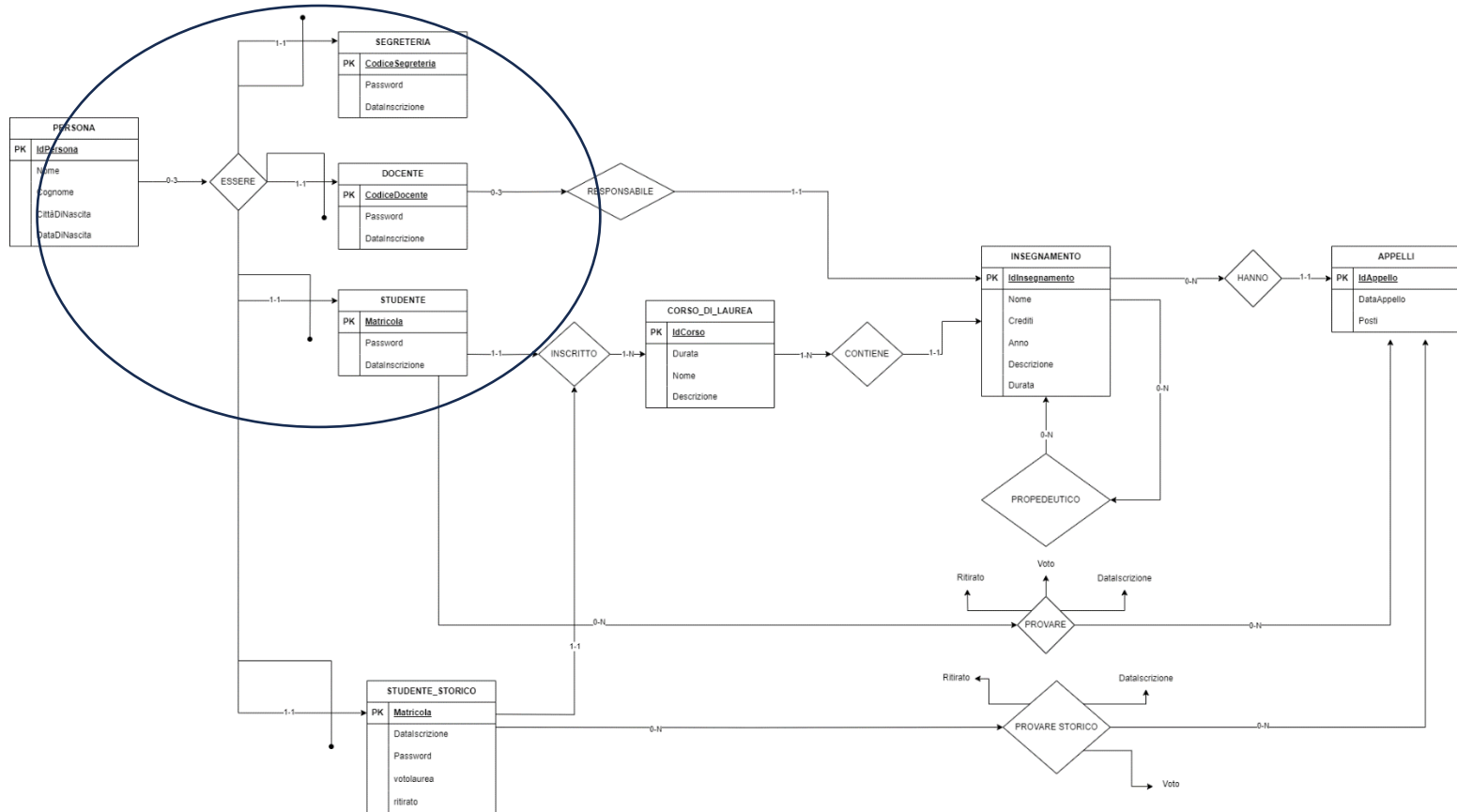
- Persona: individuo generico che si iscrive al dB. dell'università.
- Segreteria: utente responsabile della gestione della segreteria.
- Studente: studente iscritto all'università.
- Studente storico: registro che conserva i dati degli studenti eliminati o laureati.
- Docente: insegnante universitario.

Una persona può essere contemporaneamente segretario, docente e studente, a condizione che non insegni nello stesso corso in cui è iscritta (per evitare che ciò avvenga è stato predisposto un trigger).

- Corso di laurea: corso al quale è possibile iscriversi.
- Insegnamenti: materie del corso di laurea, ciascuna gestita da un professore.
- Insegnamenti propedeutici: possibilità di inserire propedeuticità tra insegnamenti dello stesso corso ma di anni differenti.
- Appelli: lista appelli creati dai professori per ciascuna materia.
- Provare: registro delle iscrizioni agli appelli da parte degli studenti.

- Provare Storico: registro in cui vengono conservati i dati delle prove ai vari appelli una volta che uno studente viene eliminato o si laurea.

## MODELLO ER-V2



SOLUZIONE PARENTELA: per risolvere la parentela ho introdotto la relazione “essere”.

- Una persona può essere nessuno o tutti e tre gli enti (SEGRETERIA, STUDENTE, DOCENTE).
- Uno STUDENTE, DOCENTE, SEGRETERIA è sempre e per forza una persona.

INFO AGGIUNTIVE:

- Tutti i codici identificativi sono di lunghezza dieci e generati dal sistema tramite una funzione php.
- Le date di iscrizione non richieste esplicitamente sono generate automaticamente tramite il parametro default= CURRENT\_DATE dal database.

## MODELLO LOGICO:

- Persona (idpersona, nome, cognome, cittadinascita, datadinascita)
- Segreteria (codicesegreteria, password, dataiscrizione, idpersona)
- Studente (matricola, password, dataiscrizione, idpersona, idcorso)
- Studente storico (matricola, password, dataiscrizione, idpersona, idcorso, votolaurea\*, ritirato)
- Docente (codicedocente, password, dataiscrizione, idpersona)
- Corso di laurea (idcorso, durata, descrizione, nome)
- Insegnamenti (idpersona, codicedocente, idcorso, idinsegnamento, nome, crediti, descrizione, durata, anno)

- Propedeutico (idinsegnamento, idinsegnamento\_dipendente)
- Appelli (idappello, dataappello, idinsegnamento, posti)
- Provare (idappello, idpersona, matricola, voto\*, dataiscrizione, ritirato)
- Provare storico (idappello, idpersona, matricola, voto\*, dataiscrizione, ritirato)

## TRIGGER- CHECK- FUNCTION

### CHECK:

- APPELLI:
  - appelli\_posti\_check: controlla che il numero dei posti inseriti per un appello sia > 0
  - data\_check: controlla che la data inserita per un appello sia >= alla data odierna
- CORSO DI LAUREA:
  - corso\_di\_laurea\_durata\_check: la durata del corso deve essere 3 o 5 anni
- INSEGNAMENTO:
  - insegnamento\_anno\_check: controlla che l'anno dell'insegnamento inserito sia >0 e <=3
  - insegnamento\_durata\_check: controlla che la durata inserita sia >0
- PROPEDEUTICO:
  - propedeutico\_check: controlla che un insegnamento non sia propedeutico a sé stesso
- PROVARE:
  - provare\_voto\_check: controlla che il voto inserito sia >= 0 e <=30
- STUDENTE STORICO:
  - votolaurea: controlla che il voto della laurea inserito sia compreso tra 0 e 111 (1 per la lode)

### TRIGGER:

- calendario (): impedisce di inserire un appello se la data in questione è già occupata da un insegnamento dello stesso corso e anno
- check\_prof (): impedisce l'inserimento di uno studente se quest'ultimo insegna nel corso selezionato
- creazione\_insegnamento (): impedisce l'inserimento di un insegnamento se il docente ha già 3 corsi attivi
- insert\_prova ():
  - impedisce l'iscrizione di uno studente ad un appello se:
    - non è iscritto al corso in questione
    - i posti sono finiti
    - ci sono delle periodicità non rispettate
  - diminuisce i posti di 1 per l'appello se l'iscrizione ha successo
- propedeutico (): impedisce l'inserimento di una propedeuticità se:
  - i due insegnamento sono di corsi differenti
  - l'insegnamento dipendente è di un anno superiore all' insegnamento preso in considerazione
- sposta\_studente (): quando uno studente viene eliminato (caso ritirato) i suoi dati vengono spostati in studente storico (il parametro ritirato viene settato a true di default) e i dati dei suoi esami vengono spostati in provare storico

### FUNCTION:

- laurea (mat varchar, voto integer): elimina uno studente e imposta:
  - votolaurea=voto
  - ritirato=false