UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL’INFORMAZIONE



CORSO DI LAUREA IN INFORMATICA

INSEGNAMENTO DI BASI DI DATI I E OBJECT ORIENTATION

ANNO ACCADEMICO 2021/2022

Progettazione e sviluppo di una base di dati relazionale per la descrizione e memorizzazione di Class Diagram UML con supporto a tutte le relative caratteristiche

*Autori: Docenti:*

Paolo Dezio Prof. Adriano PERON

MATRICOLA: N86/3612 Prof. Sergio DI MARTINO

p.dezio@studenti.unina.it

Stefano M. SABIA

MATRICOLA: N86/3214

st.sabia@studenti.unina.it

**Capitolo 1**

**Descrizione del progetto**

**1.1 Descrizione del problema**

Si progetterà e svilupperà una base di dati relazionale per la gestione di corsi di formazione. Il sistema permetterà agli operatori di gestire i corsi, questi ultimi saranno organizzate in diverse aree tematiche definibili dagli operatori stesso. È possibile inoltre, iscrivere studenti ai corsi, tenere traccia delle presenze/assenze degli studenti iscritti ed effettuare interrogazioni avanzate sui corsi erogati come ad esempio: ricerca per categoria, data o parola chiave, statistiche sul tasso di frequenza e visualizzazione degli studenti che hanno ottenuto un tasso minimo di presenze e che quindi sono idonei al superamento del corso.

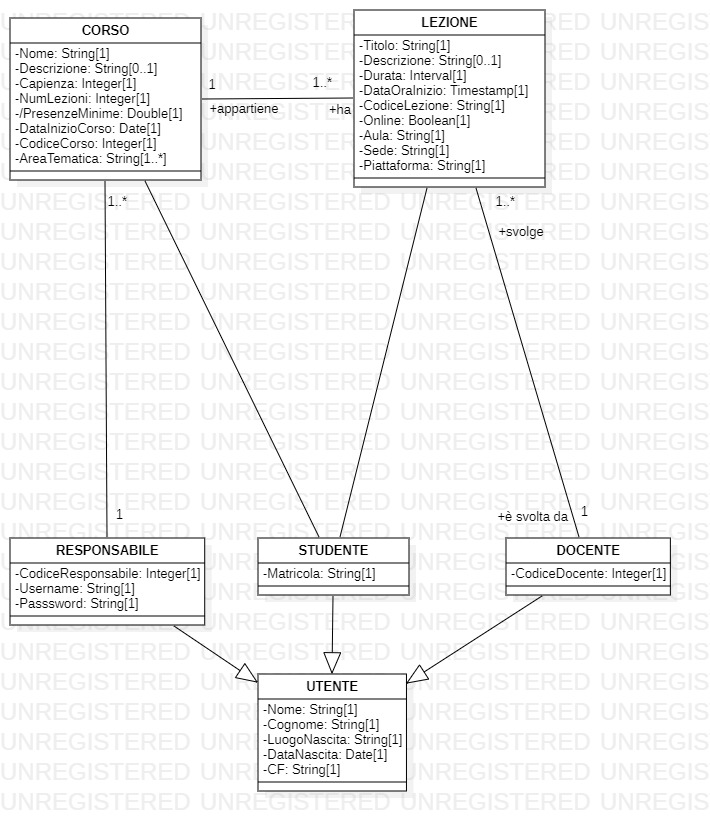
**Capitolo 2**

**Progettazione concettuale**

**2.1 Introduzione**

In questo capitolo inizia la progettazione della base di dati tramite un Class Diagram UML. Si procederà successivamente all’analisi del Class Diagram al fine di valutarne entità, associazioni e vincoli per poi passare a quello ristrutturato.

**2.2 Class Diagram**

****

**2.3 Ristrutturazione del Class Diagram**

Allo scopo di rendere il Class Diagram idoneo alla trasposizione in schemi relazionali, si procederà con la ristrutturazione dello stesso, eliminando quindi attributi strutturati, multipli ed eventuali gerarchie.

**2.4 Rimozione degli attributi multipli**

È presente all’interno della classe **CORSO** l’attributo multiplo **AreaTematica**. Si procede quindi alla creazione di una nuova classe denominata **AREA\_TEMATICA** con attributo **Categoria** per ovviare al problema.

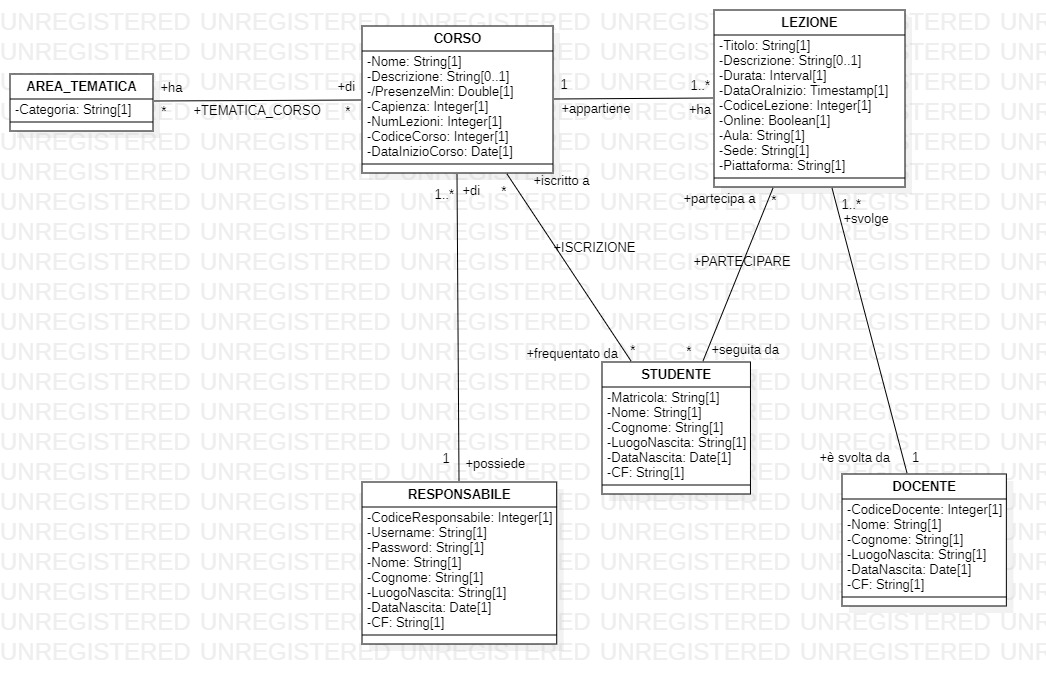
**2.4 Rimozione degli attributi strutturati**

Non sono presenti attributi strutturati.

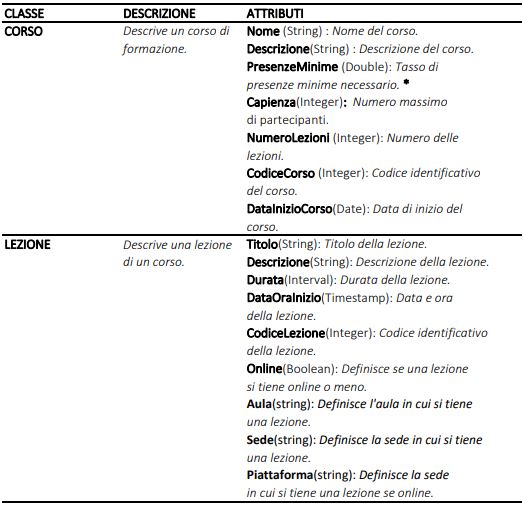
**2.6 Rimozione delle gerarchie di specializzazione**

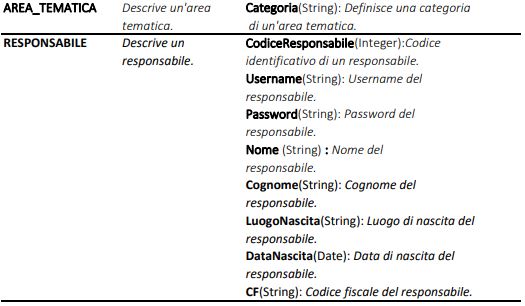
All’interno del diagramma è presente una generalizzazione **UTENTE** con tre specializzazioni: **RESPONSABILE**, **STUDENTE** e **DOCENTE**. Al fine di rimuovere le gerarchie di specializzazione. La soluzione adottata è stata quella di inserire gli attributi della classe **UTENTE** in ognuna delle classi di specializzazione.

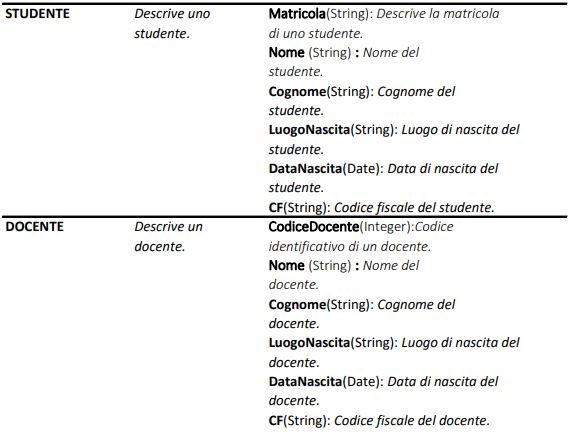
**2.7 Class Diagram ristrutturato**

****

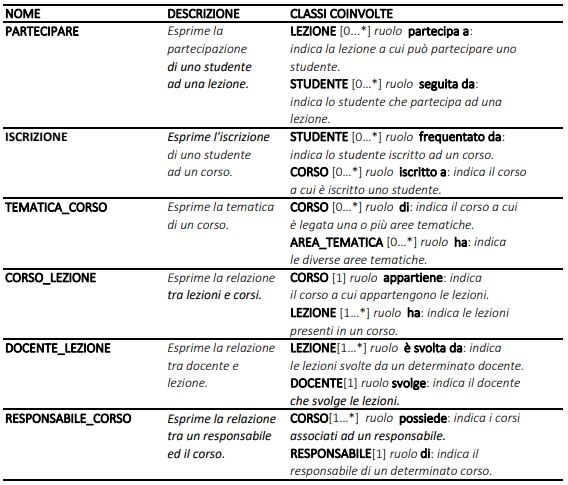
**2.8 Dizionario delle classi**

****

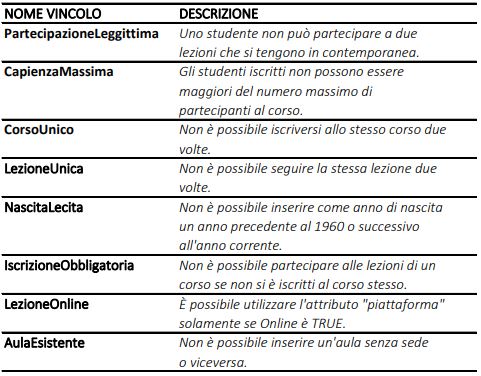
****

****

**2.9 Dizionario delle associazioni**

****

**2.10 Dizionario dei vincoli**

****

**Capitolo 3**

**Progettazione Logica**

**3.2 Introduzione**

In questo capitolo ci occupiamo di codificare il Class Diagram ristrutturato in uno schema logico in modo da rendere esplicita anche la codifica delle associazioni e le eventuali chiavi primarie ed esterne presenti. In particolare, le chiavi primarie presenteranno una singola sottolineatura mentre quelle esterne una doppia sottolineatura.

**3.2 Schema logico**

**CORSO** (CodiceCorso, CodiceResponsabile, Nome, Descrizione, Capienza, NumLezioni, DataInizioCorso)

**LEZIONE** (CodiceLezione, CodiceDocente, Titolo, Descrizione, Durata, DataOraInizio, Online, Aula, Sede, Piattaforma)

**AREA\_TEMATICA** (Categoria)

**STUDENTE** (Matricola, Nome, Cognome, LuogoNascita, DataNascita, CF)

**RESPONSABILE** (CodiceResponsabile, Username, Password, Nome, Cognome, LuogoNascita, DataNascita, CF)

**DOCENTE** (CodiceDocente, Nome, Cognome, LuogoNascita, DataNascita, CF)

**TEMATICA\_CORSO** (CodiceCorso, Categoria)

**ISCRIZIONE** (Matricola, CodiceCorso)

**PARTECIPARE** (Codice\_Lezione, Matricola)

**Capitolo 4**

**Progettazione fisica**