Federico Brandini
mat. 317271

Registro elettronico

1. Descrizione del problema e specifiche

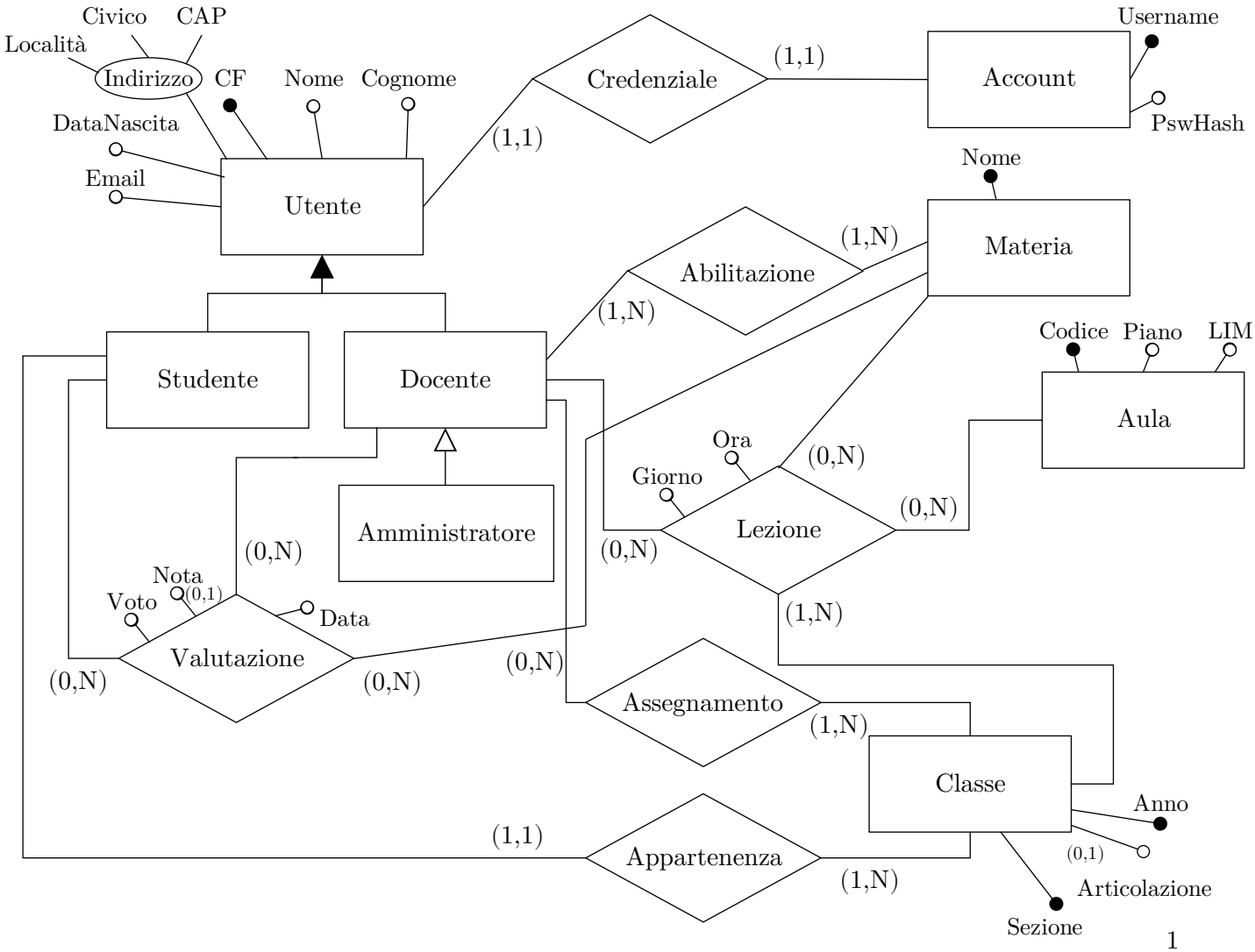
Si desidera realizzare un registro elettronico per una scuola, per la gestione di studenti e insegnanti (più in generale, utenti). Ogni studente fa parte di una classe, alla quale sono assegnati degli insegnanti, che svolgono lezioni in specifiche aule in una certa ora (le lezioni durano tutte un'ora) e in un certo giorno della settimana. Tutti gli utenti accedono al registro tramite credenziali fornite dalla scuola; le funzionalità disponibili variano a seconda del tipo di account:

- gli studenti possono esclusivamente visionare i propri voti e l'orario settimanale delle lezioni;
- gli insegnanti oltre a visionare, inserire e cancellare i voti degli studenti delle loro classi, possono visualizzare il proprio orario lavorativo nella scuola;
- gli amministratori possono inserire e cancellare studenti e insegnanti.

Un amministratore è un docente (con più funzionalità); comunque, possono esistere docenti non amministratori.

2. Progettazione concettuale

Partiamo dallo **schema concettuale** realizzato secondo il modello entità-relazioni (E/R).



Federico Brandini
mat. 317271

Si presume che una scuola intenda anche memorizzare per quali materie gli insegnanti sono abilitati; essendo che uno stesso docente può insegnare più materie e che per una stessa materia possono esistere più docenti che la insegnano, allora l'associazione *Abilitazione* è molti a molti.

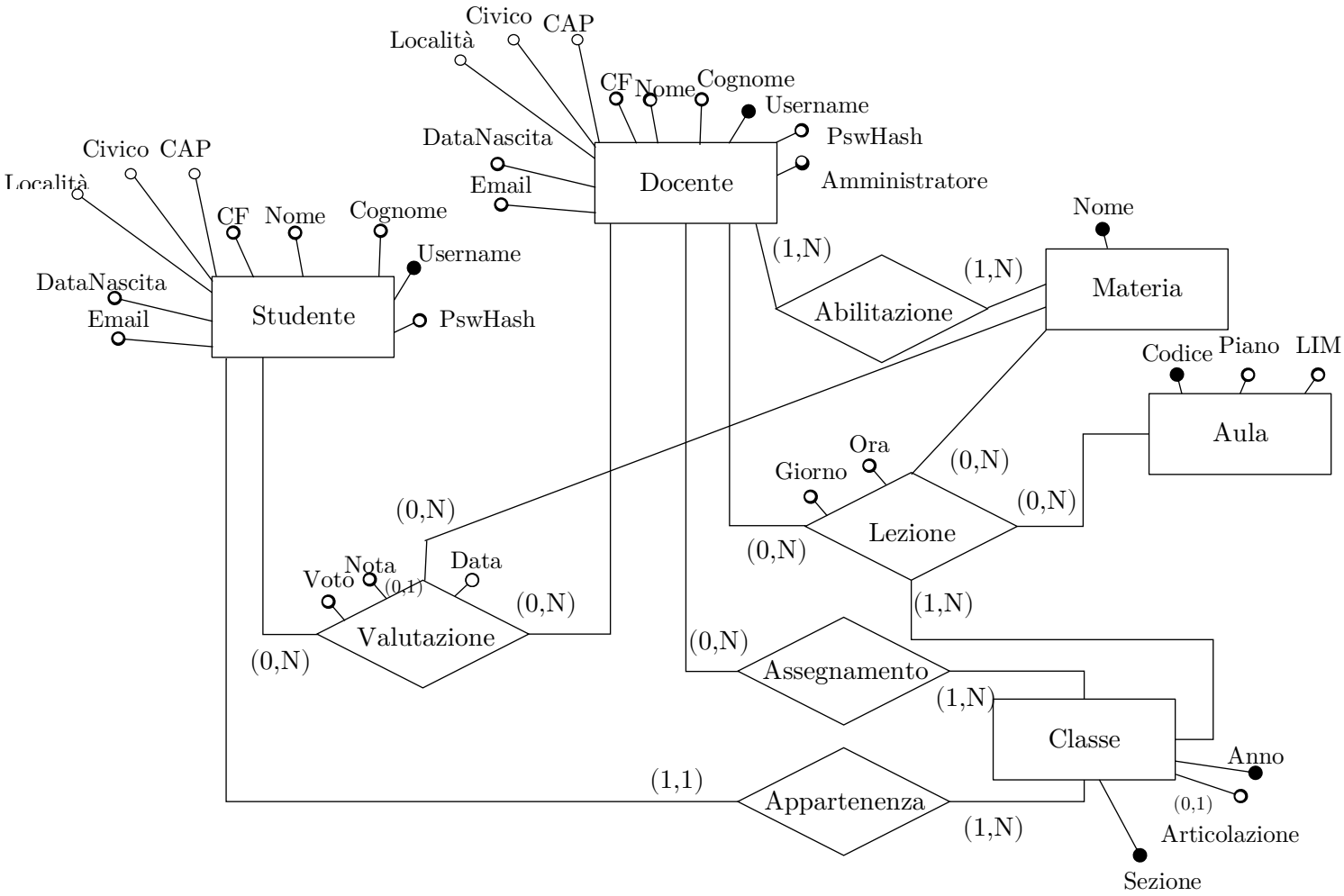
Nell'associazione *Lezione*, l'attributo *Giorno* indica un giorno della settimana (eg. lunedì, martedì, ...), mentre *Ora* indica il numero progressivo associato alla determinata lezione il tal giorno.

Nell'entità *Account*, l'attributo *PswHash* indica l'hash della password. Per ragioni di sicurezza all'interno di una base di dati non bisognerebbe memorizzare password in chiaro, ma una loro sintesi tramite, appunto, le funzioni di Hash.

Nell'entità *Aula*, l'attributo *LIM* indica se in quell'aula è presente una lavagna interattiva multimediale oppure no.

Nell'entità *Classe*, l'attributo *Articolazione* indica, se presente, l'indirizzo e/o l'articolazione di tale indirizzo di studio in quella scuola.

Passiamo ora alla fase di **ristrutturazione dello schema**, per prepararlo alla fase di progettazione logica. Sviluppiamo l'attributo composto *Indirizzo* nelle sue determinanti ed eliminiamo tutte le gerarchie effettuando prima un accorpamento di *Amministratore* in *Docente* e, in seguito, accorpando *Utente* nelle due entità figlie. Per evitare di creare due relazioni (*CredenzialeStudente* e *CredenzialeDocente*) ed essendo che le informazioni relative all'account vengono sempre utilizzate insieme all'utente, decidiamo di inglobare gli attributi di *Account* in *Utente* (quindi in *Studente* e *Docente*). Scegliamo come identificatore di queste ultime due entità il solo username. Realizziamo, quindi, lo schema concettuale ristrutturato.



Federico Brandini
mat. 317271

3. Progettazione logica

Partendo dallo schema precedente, procediamo con la traduzione nello schema logico seguendo il modello Relazionale.

- Studente(Username, PswHash, CF, Nome, Cognome, DataNascita, Email, Località, Civico, CAP, AnnoClasse, SezClasse)
- Docente(Username, PswHash, Amministratore, CF, Nome, Cognome, DataNascita, Email, Località, Civico, CAP)
- Materia(Nome)
- Aula(Codice, Piano, LIM)
- Classe(Anno, Sezione, Articolazione)
- Abilitazione(UserDocente, Materia)
- Lezione(UserDocente, Materia, Aula, AnnoClasse, SezClasse, Giorno, Ora)
- Valutazione(ID, UserDocente, Materia, UserStudente, Data, Voto, Nota)
- Assegnamento(UserDocente, AnnoClasse, SezClasse)

Siccome un docente non può tenere due lezioni diverse lo stesso giorno alla stessa ora, possiamo ridurre la superchiave di *Lezione* con la chiave primaria identificata dalla terna $(UserDocente, Giorno, Ora)$. Contestualmente, $(Aula, Giorno, Ora)$ e $(AnnoClasse, SezClasse, Giorno, Ora)$ sono comunque superchiavi della relazione.

Nell'eventualità, un docente potrebbe assegnare più valutazioni allo stesso studente, per la stessa materia e nello stesso giorno; di conseguenza è necessario introdurre una chiave primaria "artificiale" (un codice identificativo *ID*) nella relazione *Valutazione*.

Aggiungiamo i vincoli di integrità referenziale.

- Studente[AnnoClasse, SezClasse] → Classe[Anno, Sezione]
- Abilitazione.UserDocente → Docente.Username
- Abilitazione.Materia → Materia.Nome
- Lezione.UserDocente → Docente.Username
- Lezione.Materia → Materia.Nome
- Lezione[AnnoClasse, SezClasse] → Classe[Anno, Sezione]
- Valutazione.UserDocente → Docente.Username
- Valutazione.Materia → Materia.Nome
- Valutazione.UserStudente → Studente.Username
- Assegnamento.UserDocente → Docente.Username
- Assegnamento[AnnoClasse, SezClasse] → Classe[Anno, Sezione]

4. Query utilizzate

Elenchiamo le principali interrogazioni utilizzate nella parte di back-end del sito web (i simboli '?' costituiscono dei segnaposto per dei valori costanti).

```
SELECT Username, Nome, Cognome
FROM Studente
WHERE (Username, PswHash) = (?, ?);
```

SELECT Username, Nome, Cognome, Amministratore
FROM Docente
WHERE (Username, PswHash) = (?, ?);
(Le prime due query che vediamo servono per la fase di login, per autenticare l'utente che desidera accedere al registro elettronico e per estrarre le sue informazioni generali; questa prima fase permette di capire anche il tipo di account connesso (Studente o Docente); la prima fra le due interrogazioni seguenti è relativa agli account Studente, mentre la seconda agli account Docente)

SELECT Materia
FROM Abilitazione
WHERE UserDocente = ?
ORDER BY Materia;
(Restituisce le materie, in ordine alfabetico crescente, alle quali un certo docente è abilitato)

SELECT AnnoClasse, SezClasse
FROM Assegnamento
WHERE UserDocente = ?
ORDER BY AnnoClasse, SezClasse;
(Restituisce le classi, in ordine di anno e di sezione, assegnate ad un certo docente)

SELECT Username, Cognome, Nome
FROM Studente
WHERE (AnnoClasse, SezClasse) = (?, ?)
ORDER BY Cognome, Nome;
(Restituisce gli studenti di una certa classe ordinati per cognome e nome)

SELECT ID, UserStudente, Data, Voto, Nota
FROM Valutazione
WHERE (UserStudente, Materia) = (?, ?)
ORDER BY Data;
(Restituisce le valutazioni di un certo studente in una determinata materia ordinate per data)

SELECT DISTINCT Materia
FROM Valutazione
WHERE UserStudente = ?
ORDER BY Materia;
(Restituisce le materie ordinate alfabeticamente, nelle quali un determinato studente ha ricevuto almeno una valutazione)

SELECT Giorno, Ora, Materia, AnnoClasse, SezClasse, Aula
FROM Docente, Lezione
WHERE UserDocente = ?
ORDER BY Ora;
(Restituisce le lezioni tenute da un determinato docente ordinate per ora)

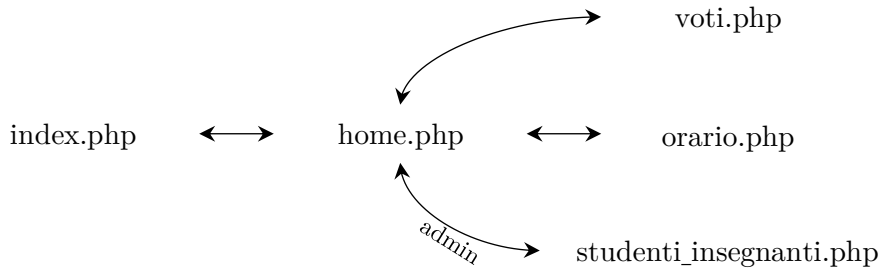
SELECT Lezione.Giorno, Lezione.Ora, Lezione.Materia, Docente.Cognome, Lezione.Aula
FROM Docente, Lezione
WHERE Docente.Username = Lezione.UserDocente AND (Lezione.AnnoClasse, Lezione.SezClasse) = ANY (SELECT AnnoClasse, SezClasse FROM Studente WHERE Username = ?)
ORDER BY Lezione.Ora;
(Restituisce le lezioni, ordinate per ora, alle quali deve partecipare un determinato studente)

SELECT COUNT(*) AS NumMaterie
FROM Materia;
(Restituisce il numero di materie presenti nella scuola)

SELECT COUNT(*) AS NumClassi
FROM Classe;
(Restituisce il numero di classi presenti nella scuola)

5. Descrizione del sito e funzionamento

Partiamo introducendo la struttura del sito; per farlo possiamo utilizzare un grafo, dove ogni nodo corrisponde ad una pagina web, gli archi entranti costituiscono gli inlinks e quelli uscenti gli outlinks. Assunto ciò, la struttura del sito web può essere schematizzata come segue.



Il sito web inizia dal documento web *index.php*, con la schermata di login, il quale, una volta effettuato, porta a *home.php*, dove sarà possibile scegliere fra più funzionalità (due nel caso di studenti e semplici docenti e tre per docenti amministratori) attraverso un menù:

- **voti** (*voti.php*), che permette agli studenti di visualizzare i propri voti e ai docenti di inserirne di nuovi (oltre a visualizzarli e cancellarli);
- **orario** (*orario.php*), che consente a studenti e docenti di visualizzare il loro orario settimanale rispettivamente di lezione e di lavoro;
- **studenti e insegnanti** (*studenti_insegnanti.php*), che consente ai soli docenti amministratori di visualizzare tutti gli studenti e i docenti della scuola (oltre a eliminarli, inserirne di nuovi e promuovere ad amministratore i semplici docenti).