

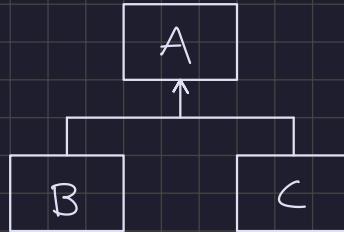
Basi di dati: progettazione

B

Prova in itinere del 29 novembre 2024

a) Si illustri il costrutto di generalizzazione del modello Entità-Relazioni (1 punto)

La generalizzazione è un legame logico tra un'entità padre e più entità figlie. Questo legame crea una classe di oggetti che generalizza le classi di oggetti rappresentate dalle entità figlie. È rappresentata nel seguente modo:



Un esempio è il seguente:



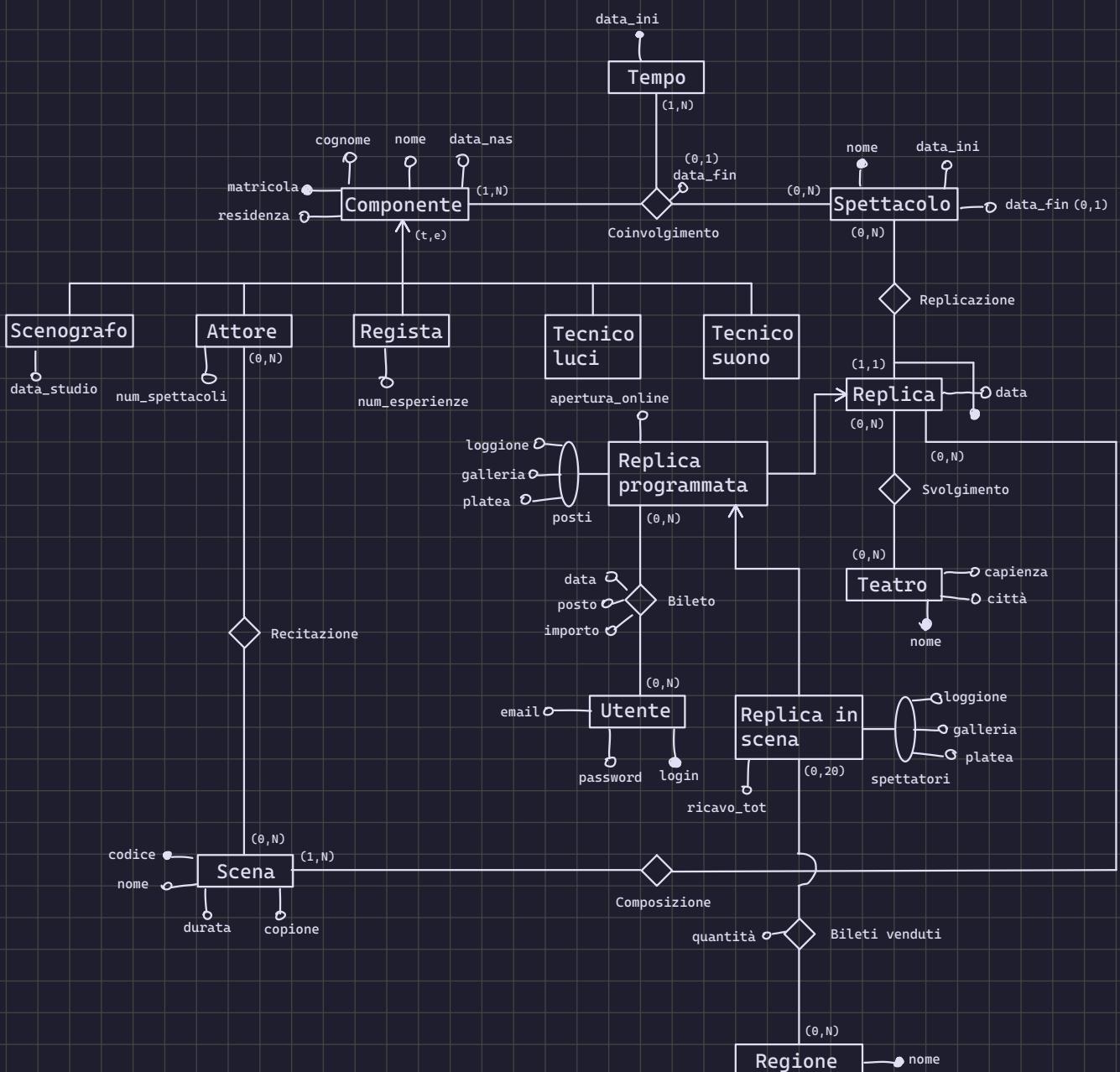
La generalizzazione si divide in:

- Totale esclusiva (t,e)
- Totale sovrapposta (t,s)
- Parziale esclusiva (p,e)
- Parziale sovrapposta (p,s)

PROGETTAZIONE DI UNA BASE DI DATI (è obbligatorio svolgere l'esercizio in modo completo, vale a dire occorre produrre uno schema concettuale e il corrispondente schema logico)

1. Si vuole progettare un sistema informativo per la gestione delle attività di una compagnia teatrale che tiene spettacoli nei teatri italiani.
 - a) Il sistema registra gli spettacoli che la compagnia offre o ha offerto. Ogni spettacolo è caratterizzato da: un nome univoco, una data di inizio delle repliche e una eventuale data di fine. Ogni componente della compagnia viene registrato nel sistema assegnando: una matricola univoca, il cognome, il nome, la data di nascita e la città di residenza. I componenti vengono classificati in: scenografo, attore/attrice, tecnico suono, tecnico luci e regista. Per lo scenografo si registra in aggiunta la data del titolo di studio, per l'attore/attrice il numero di spettacoli in cui hanno recitato, per il regista il numero di esperienze di regia svolte. Infine, ogni componente viene inserito in uno o più spettacoli indicando la data di inizio e l'eventuale data di fine del coinvolgimento del componente nello spettacolo. Il sistema deve essere in grado di ricostruire i vari spettacoli in cui un componente è o è stato coinvolto.
 - b) Per ogni spettacolo si programmano le repliche che si svolgeranno presso i teatri italiani. Per ogni replica si registra la data della replica (univoca nell'ambito dello spettacolo) e il teatro presso cui si svolge. Per ogni teatro si registra: il nome univoco, la città e la capienza. Per ogni replica programmata si registra il numero di biglietti (quantità) messi in vendita divisi per categoria (platea, loggione e galleria). Inoltre, si indica la data di apertura delle vendite dei biglietti che avviene solo on line. Gli utenti che desiderano acquistare i biglietti si registrano indicando: login univoco, password e indirizzo email. Al momento dell'acquisto il sistema registra: l'utente, la replica, la data dell'acquisto, il posto assegnato e l'importo pagato. Ogni utente può acquistare uno e un solo biglietto per ogni replica.
 - c) Per ogni replica di uno spettacolo si registrano inoltre nel sistema tutte le scene riportando: un codice univoco, il nome della scena, la sua durata media e il copione. Si indicano inoltre gli attori che sono coinvolti nelle varie scene. Dopo che una replica è andata in scena si registra il numero di spettatori presenti nelle varie categorie (platea, loggione e galleria) e il ricavo totale. Il sistema memorizza anche per ogni replica andata in scena e per ogni regione italiana il numero di biglietti venduti nella regione.

Progettare lo schema concettuale utilizzando il modello entità-relazione (ER). Non aggiungere attributi non esplicitamente indicati nel testo, indicare almeno un identificatore per ogni entità.



2. Generare lo schema relazionale a partire dallo schema concettuale ottenuto all'esercizio 1. Indicare esplicitamente per ogni relazione dello schema relazionale: le chiavi primarie, gli attributi che possono contenere valori nulli e i vincoli di integrità referenziale.

```
Componente(matricola, nome, cognome, residenza, data_nas, tipo: {scenografo, regista, tecnico_luci, tecnico_suono}, data_studio*, num Esperienze*)
```

Attore(componente, num_spettacoli) Attore.componente → Componente

Tempo(data ini)

Spettacolo(nome, data ini, data fin*)

Coinvolgimento([componente], [spettacolo], [tempo], data_fin*)

Coinvolgimento.componente → Componente
Coinvolgimento.spettacolo → Spettacolo
Coinvolgimento.tempo → Tempo

`Replica(data, spettacolo)` `Replica.spettacolo` → `Spettacolo`

Teatro(nome, città, capienza)

Svolgimento(replica, spettacolo, teatro)

Svolgimento.(replica, spettacolo) → Replica
Svolgimento.teatro → Teatro

Scena(codice, nome, durata, copione)

Recitazione(attore, scena)

Recitazione.attore → Attore
Recitazione.scena → Scena

Composizione(replica, spettacolo, scena)

Composizione.(replica, spettacolo) → Replica
Composizione.scena → Scena

ReplicaProgrammata(replica, spettacolo, apertura_online, loggione, galleria, platea)

ReplicaProgrammata.(replica, spettacolo) → Replica

Utente(login, password, email)

Bileto(utente, replica_programmata, spettacolo, data, posto, importo)

Bileto.utente → Utente
Bileto.(replica_programmata, spettacolo) → ReplicaProgrammata

ReplicaInScena(replica_programmata, spettacolo, ricavo_tot, loggione, galleria, platea)

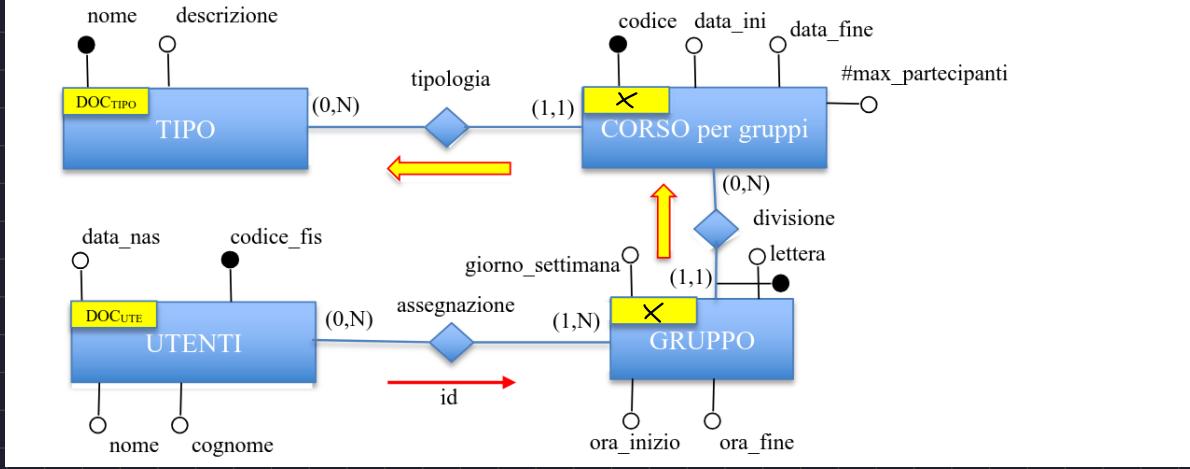
ReplicaInScena.(replica_programmata, spettacolo) → ReplicaProgrammata

Regione(nome)

BiletiVenduti(replica_in_scena, spettacolo, regione, quantità)

BiletiVenduti.(replica_in_scena, spettacolo) → ReplicaInScena
BiletiVenduti.regione → Regione

3. Dato il seguente schema ER etichettato relativo all'erogazione di corsi di formazione a gruppi di utenti, inserire le etichette corrette per le entità CORSO e GRUPPO e generare lo schema dei documenti corrispondente.



```

DOC_tipo: {
  _id: autogen,
  nome: string,
  descrizione: string,
  tipologia: [{{
    codice: integer,
    data_ini: date,
    data_fine: date,
    #max_partecipanti: integer,
    divisione: [{{
      lettera: string,
      ora_fine: timestamp,
      ora_inizio: timestamp,
      giorno_settimana: integer,
      id_utente: [integer]
    }}, ... ]
  }}, ... ]
}
  
```

```

DOC_ute: {
  _id: autogen,
  codice_fis: string,
  data_nas: date,
  nome: string,
  cognome: string,
}
  
```