

Basi di Dati
 Prof. L. Tanca e F. A. Schreiber
APPELLO DEL 6 MARZO 2015
Tempo: 2h30m

Si consideri il seguente schema di base di dati, che vuole memorizzare alcune informazioni relative a Twitter.

TWEET (TweetId, Testo, Utente, DataPubblicazione, NazionePubblicazione, Retweet)
 UTENTE (IdUtente, Nome, Nazione, DataIscrizione)
 SEGUE (Utente, UtenteSeguito, DataInizio)
 PREFERITO (Utente, Tweet)

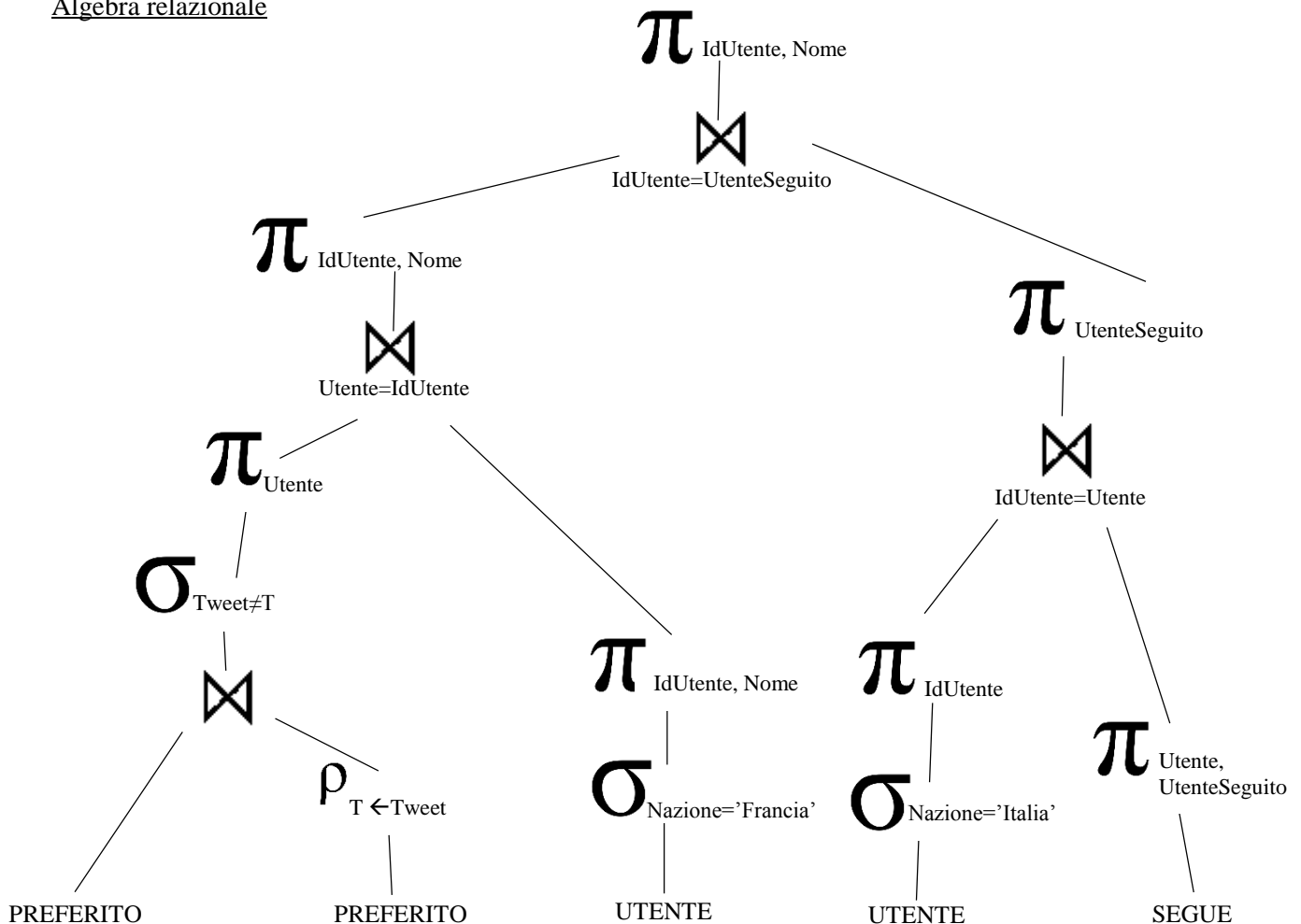
Gli utenti possono pubblicare tweet, eventualmente facendo il retweet di un tweet già esistente (cioè pubblicandolo nuovamente). Se un tweet è il retweet di un tweet già esistente, allora l'attributo Retweet della tabella TWEET contiene l'id del tweet originale; se il tweet non è un retweet l'attributo Retweet vale NULL. Gli identificatori di tweet e utenti sono valori interi.

A. LINGUAGGI FORMALI (6 punti)

Esprimere in algebra relazionale (ottimizzata) e calcolo relazionale la seguente interrogazione:

Trovare gli id e i nomi di tutti gli utenti francesi che hanno indicato almeno due diversi tweet come preferiti e che sono seguiti da utenti italiani. (4 punti)

Algebra relazionale



Calcolo relazionale

$\{t \mid (\exists t1 \in \text{PREFERITO}, \exists t2 \in \text{PREFERITO}, \exists t3 \in \text{UTENTE}, \exists t4 \in \text{UTENTE}, \exists t5 \in \text{SEGUE}$
 $(t1[\text{Tweet}] \neq t2[\text{Tweet}] \wedge t3[\text{Nazione}] = \text{'Francia'} \wedge t4[\text{Nazione}] = \text{'Italia'} \wedge t1[\text{Utente}] = t2[\text{Utente}] \wedge$
 $t1[\text{Utente}] = t3[\text{IdUtente}] \wedge t4[\text{IdUtente}] = t5[\text{Utente}] \wedge t3[\text{IdUtente}] = t5[\text{UtenteSeguito}])\}$

Esprimere in Datalog la seguente interrogazione:

Trovare gli id degli utenti iscritti a partire dal 01/06/2014 che non hanno mai pubblicato retweet di tweet pubblicati da utenti che seguono. (2 punti)

UtenteRetweetSeguito(Utente) :- Tweet(_, _, Utente, _, _, Retweet), Tweet(Retweet, _, Utente2, _,
_, _), Segue(Utente, Utente2, _)
Risultato(IdUtente) :- Utente(IdUtente, _, _, DataIscrizione), DataIscrizione >= "01/06/2014",
¬ UtenteRetweetSeguito(IdUtente)
?- Risultato(x)

B. SQL: INTERROGAZIONI (7 punti)

Scrivere in SQL le seguenti interrogazioni:

1. *Trovare le nazioni da cui sono stati pubblicati meno di 10000 tweet da parte di utenti i cui tweet non sono mai stati retwittati. (3 punti)*

```
SELECT NazionePubblicazione
FROM Tweet
WHERE Utente NOT IN (
    SELECT T2.Utente
    FROM Tweet AS T1, Tweet AS T2
    WHERE T1.Retweet=T2.TweetId
)
GROUP BY NazionePubblicazione
HAVING COUNT(*)<10000
```

NOTA:

Questa soluzione non restituisce le nazioni in cui tutti gli utenti sono stati retwittati almeno una volta. Includere queste nazioni rende l'interrogazione molto complessa – più di quanto si volesse intendere nella fase di scrittura del tema d'esame – e quindi la soluzione qui riportata è stata considerata corretta.

2. *Trovare gli id degli utenti che seguono il maggior numero di persone provenienti da una nazione diversa dalla propria. (4 punti)*

```
CREATE VIEW NumeroSeguitiNazioneDiversa(Utente, Numero) AS (
    SELECT U1.IdUtente, COUNT(*)
    FROM Utente AS U1, Segue AS S, Utente AS U2
    WHERE U1.IdUtente=S.Utente AND S.UtenteSeguito=U2.IdUtente AND
        U1.Nazione <> U2.Nazione
    GROUP BY U1.IdUtente
)
```

```

SELECT Utente
FROM NumeroSeguitiNazioneDiversa
WHERE Numero = ( SELECT MAX(Numero)
                   FROM NumeroSeguitiNazioneDiversa
                   )

```

C. DDL: DEFINIZIONE DI SCHEMI E VINCOLI (3 punti)

1. Specificare in SQL la creazione delle tabelle TWEET e SEGUE, definendo i vincoli di tupla e di dominio ritenuti opportuni ed esprimendo eventuali vincoli di integrità referenziale relativi a tutte le tabelle dello schema. (1 punto)

```

CREATE TABLE Tweet (
    TweetId INTEGER PRIMARY KEY,
    Testo VARCHAR(255),
    Utente INTEGER REFERENCES Utente(IdUtente) ON UPDATE CASCADE ON DELETE CASCADE,
    DataPubblicazione DATE,
    NazionePubblicazione VARCHAR(255),
    Retweet INTEGER DEFAULT NULL REFERENCES Tweet(TweetId) ON UPDATE CASCADE ON DELETE SET NULL
)

```

```

CREATE TABLE Segue (
    Utente INTEGER REFERENCES Utente(IdUtente) ON UPDATE CASCADE ON DELETE CASCADE,
    UtenteSeguito REFERENCES Utente(IdUtente) ON UPDATE CASCADE ON DELETE CASCADE,
    DataInizio DATE,
    PRIMARY KEY (Utente, UtenteSeguito)
)

```

2. Specificare in SQL il vincolo che controlla che nessun tweet contenuto nella base di dati abbia data di pubblicazione precedente alla data di iscrizione dell'utente che lo ha pubblicato. (2 punti)

```

CREATE ASSERTION VerificaDate CHECK (
    NOT EXISTS (
        SELECT *
        FROM Tweet AS T, Utente AS U
        WHERE T.Utente=U.IdUtente AND T.DataPubblicazione<U.DataIscrizione
    )
)

```

D. TEORIA (4 punti)

Illustrare il modello ANSI/SPARC e spiegare la funzione e le caratteristiche di ciascun livello.

C. PROGETTAZIONE DI UNA BASE DI DATI (11 punti)

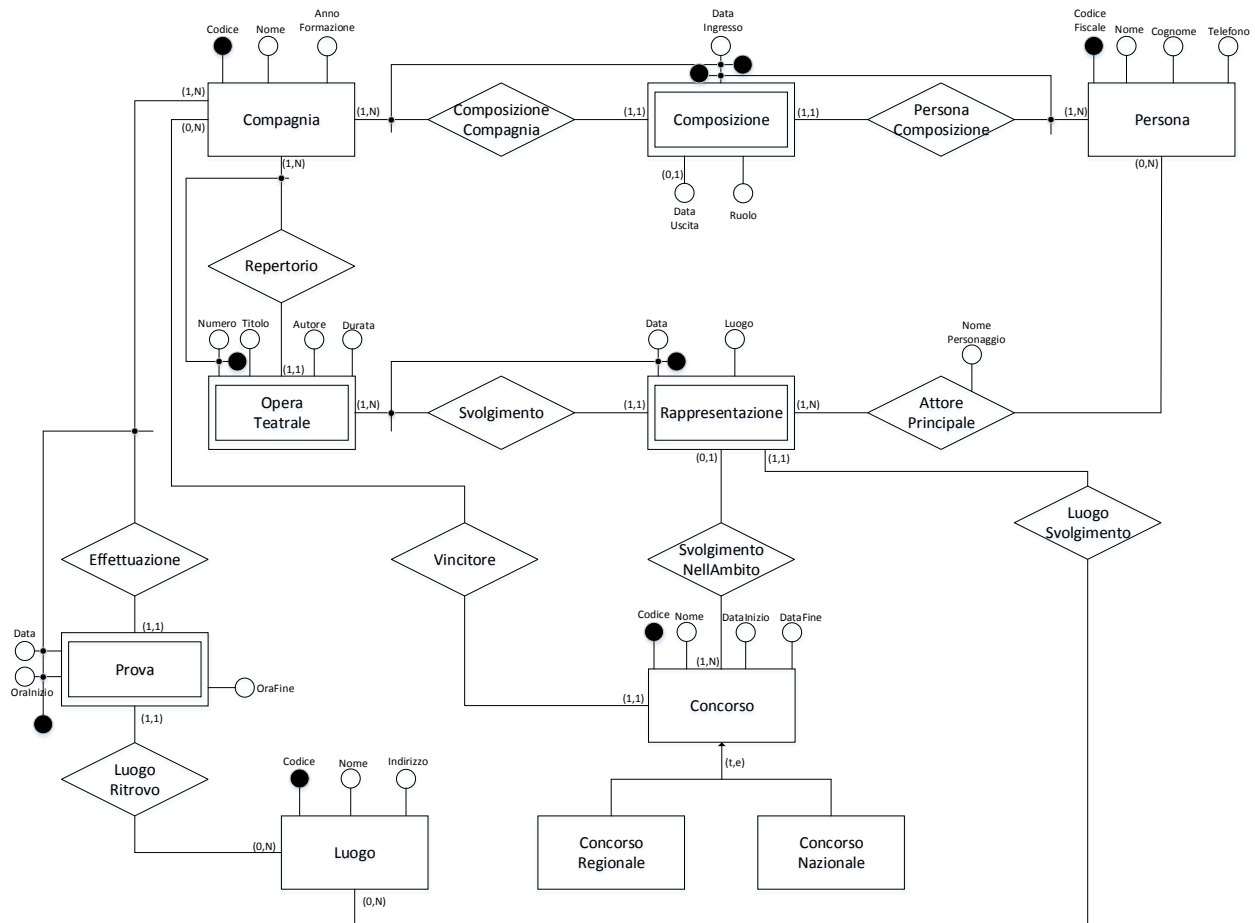
Progettare la base di dati relativa alla situazione descritta nel seguito. Svolgere il progetto concettuale usando il modello ER, e il progetto logico con il modello relazionale. Specificare le chiavi e gli eventuali vincoli d'integrità.

Si vuole tenere traccia delle attività delle compagnie teatrali dislocate sul territorio lombardo. Ogni compagnia è contraddistinta da un codice univoco, e si registra anche il nome e l'anno in cui si è formata. Ogni compagnia è di solito composta da attori, da direttori artistici e dal responsabile della comunicazione. Ogni persona è identificata dal codice fiscale, nome, cognome, numero di telefono ruolo all'interno della compagnia (es. attore, direttore artistico), data di ingresso nella compagnia e eventuale data di uscita. Ogni persona appartiene ad un sola compagnia alla volta e il ruolo può cambiare a seconda della compagnia in cui si trova. Ogni compagnia ha poi un repertorio di opere teatrali di cui si conoscono il numero progressivo unico per ogni compagnia, il titolo, l'autore e la durata. Ogni stagione si decide quali opere rappresentare e si definisce il calendario. Per ogni rappresentazione si vogliono anche memorizzare il luogo in cui si tiene la lista degli attori principali e il nome del personaggio che interpretano.

Alcune rappresentazioni si tengono nell'ambito di concorsi nazionali o regionali. Ogni concorso è identificato da un codice identificativo, nome, data inizio, data fine. Per ogni concorso si vuole tenere traccia della compagnia teatrale vincitrice.

La compagnia si incontra periodicamente per provare. Si vuole memorizzare la lista delle prove identificate dalla data, ora inizio, ora fine e riferimento al luogo di ritrovo. Di tutti i luoghi si memorizza un codice identificativo, nome, indirizzo.

Progetto concettuale



Progetto logico

Compagnia (Codice, Nome, AnnoFondazione)

Persona (CodiceFiscale, Nome, Cognome, Telefono)

Composizione (Compagnia, Persona, DataIngresso, DataUscita*, Ruolo)

OperaTeatrale (Compagnia, Numero, Titolo, Autore, Durata)

Rappresentazione (Compagnia, NumeroOpera, Data, Luogo, Concorso*)

AttorePrincipale (Compagnia, NumeroOpera, Data Attore, NomePersonaggio)

Concorso (Codice, Nome, DataInizio, DataFine, Tipo, Vincitore)

Luogo (Codice, Nome, Indirizzo)

Prova (Compagnia, Data, OraInizio, OraFine, LuogoRitrovo)

Chiavi esterne

Composizione.Compagnia → Compagnia.Codice

Composizione.Persona → Persona.CodiceFiscale

OperaTeatrale.Compagnia → Compagnia.Codice

Rappresentazione.(Compagnia, NumeroOpera) → OperaTeatrale.(Compagnia, Numero)

Rappresentazione.Concorso → Concorso.Codice

Rappresentazione.Luogo → Luogo.Codice

AttorePrincipale.(Compagnia, NumeroOpera, Data) → Rappresentazione.(Compagnia, NumeroOpera, Data)

AttorePrincipale.Attore → Persona.CodiceFiscale

Concorso.Vincitore → Compagnia.Codice
Prova.Compagnia → Compagnia.Codice
Prova.LuogoRitrovo → Luogo.Codice