PROVA COMPLETA - Basi di Dati e Lab
11/06/2018
Corso di Laurea in Ingegneria Informatica

Prof. Sonia Bergamaschi

Esercizio 1 (punti 4)

a. Definizione di terza forma normale ed esempi di violazione.

Esercizio 2 (punti 11)

Si vuole creare un sistema informativo per la gestione di progetti software. All'interno del sistema possono registrarsi degli utenti, ogni utente ha un indirizzo email, una password; un username univoco.

Un utente può creare dei repository. Un repository ha un nome identificativo, una data di creazione e una descrizione. Su un repository, oltre a chi lo ha creato, possono agire anche altri utenti in base ai privilegi a loro assegnati. I privilegi sono standard, definiti dalla piattaforma, ogni privilegio ha un nome univoco e una descrizione che indica che cosa consente di fare.

Un repository può contenere diversi tipi di elementi; un elemento ha un nome ed è identificato da un codice all'interno del repository. Un elemento può essere o un file o una directory, le directory possono contenere a loro volta altri file o altre directory; un file può essere contenuto in una sola directory. Un utente può modificare un repository tramite il comando commit. Un commit ha un messaggio che lo descrive; è identificato dal repository, dall'utente che lo ha effettuato e da un timestamp, e può agire su più elementi. Quando un elemento viene modificato tramite commit, se ne registrano le modifiche effettuate.

Con la funzionalità forum gli utenti possono aprire delle discussioni su un repository per fare domande, segnalare errori, etc. Una discussione ha: un titolo, un autore, una data di apertura ed un codice univoco. Ad una discussione possono essere inviate delle risposte da parte di altri utenti, ogni risposta ha un autore, un testo, una data ed è identificata da un numero crescente all'interno della discussione. Il sistema chiude automaticamente le discussioni che non ricevono risposte per più di 90 giorni, riportando la data di chiusura; quando la discussione è chiusa non è più possibile inserire risposte.

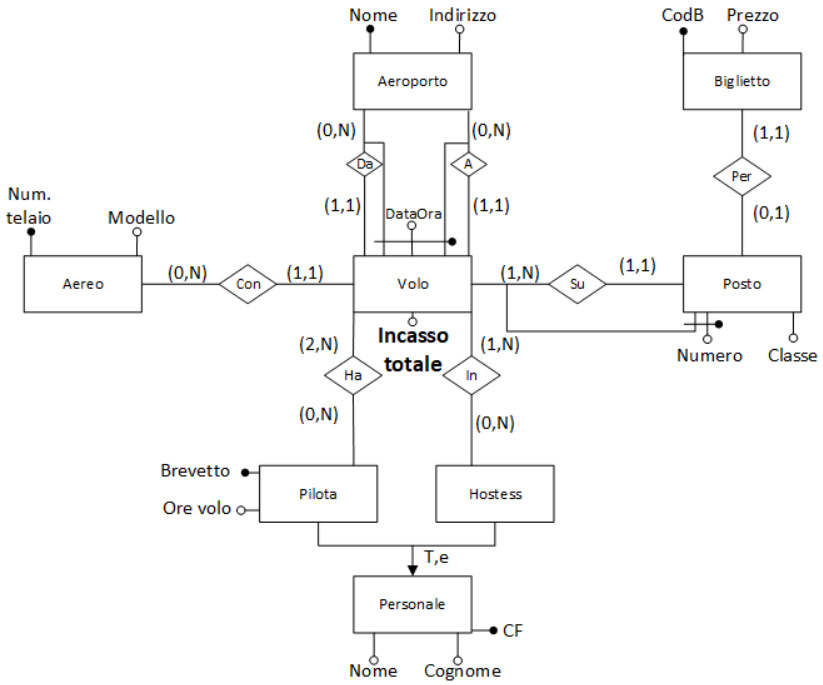
Viene richiesto di:

- **a)** Progettare lo schema E/R, evidenziando gli eventuali vincoli non esprimibili nel modello E/R. **(9 punti)**
- **b)** Modificare lo schema precedente aggiungendo il seguente vincolo: **(2 punti)**

Per motivi di memorizzazione si vuole limitare la proliferazione di repository limitandone la creazione ad un massimo di 3 all'anno per utente.

Esercizio 3 (punti 7)

Sia dato il seguente schema E/R



A partire dallo schema, dalle informazioni contenute nella tabella dei volumi dei dati e dalla frequenza delle operazioni, viene richiesto di:

1. Determinare se conviene o meno mantenere l'attributo derivato "Incasso totale" che rappresenta la somma dei prezzi dei biglietti venduti per un volo, considerando le seguenti operazioni (**4 punti**)

   Operazione 1: dato un volo ottenere l'incasso totale (somma dei prezzi dei biglietti venduti)
   Operazione 2: inserimento di un nuovo biglietto supponendo noti il posto e il volo (posto e volo esistono già, non vanno inseriti).

   Frequenza: operazione 1 50/giorno, operazione 2 100/giorno.

2. Produrre lo schema logico relazionale evidenziando eventuali vincoli non esprimibili. (**3 punti**)

Tabella dei volumi dei dati

CONCETTO	TIPO	VOL.
Aeroporto	E	3
Biglietto	E	1200
Aereo	E	10
Volo	E	80
Posto	E	2400

CONCETTO	TIPO	VOL.
Ha	R	160
In	R	320
Personale	R	1000

Il 20% del personale sono piloti, l'80% sono hostess.

Esercizio 4 (punti 11)

Sia dato il seguente schema relazionale di un sistema di gestione dei treni.

```
Treno (id treno, modello, anno, num_posti)

Stazione (cod stazione, nome, città)
AK (nome)

Tratta (id tratta, id_treno, partenza, arrivo, data_partenza,
ora_partenza, durata)
FK: id_treno References Treno NOT NULL
FK: partenza References Stazione NOT NULL
FK: arrivo References Stazione NOT NULL
```

Si richiede di scrivere in algebra relazionale e in SQL le seguenti interrogazioni:

a) Selezionare i nomi delle stazioni da cui nel 2010 sono partiti tutti i modelli di treni presenti. **(punti 2+2)**.

b) Selezionare i dati dei treni che sono passati per la città di Modena (partenza o arrivo) nel mese di gennaio del 2012. **(punti 2+2)**

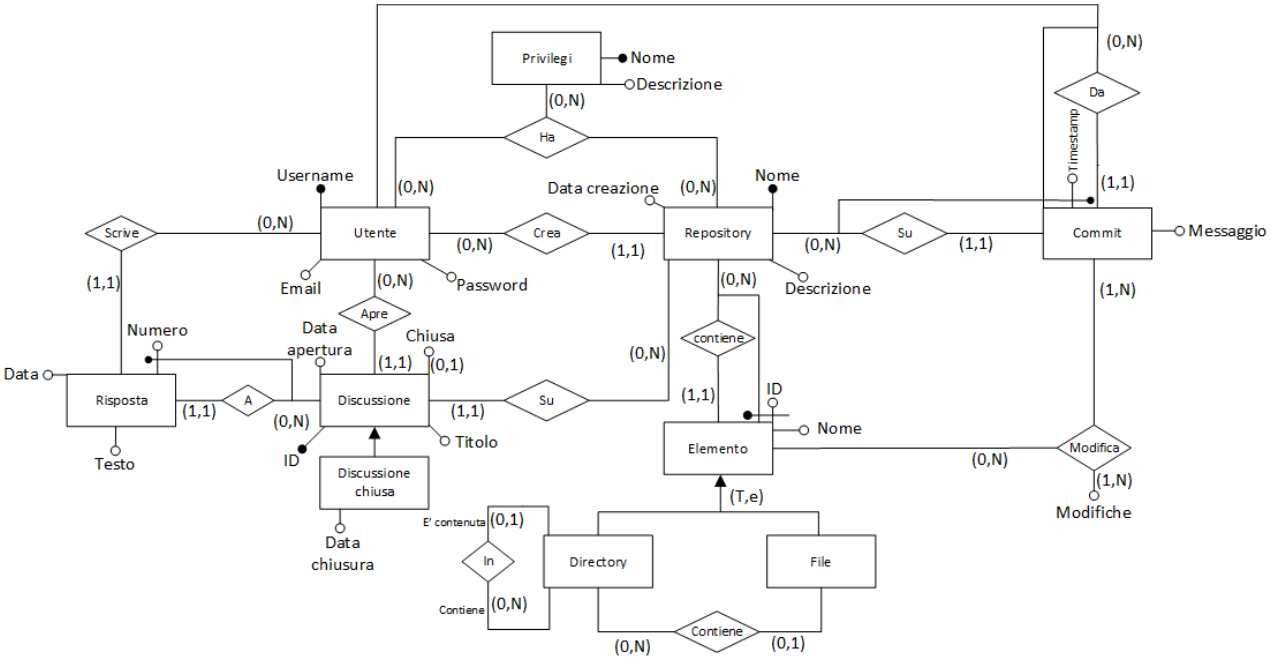
Si richiede di scrivere in SQL le seguenti interrogazioni:

c) Selezionare i dati dei treni che nel 2016 hanno effettuato più ore di lavoro (somma delle durate delle tratte). **(punti 3)**

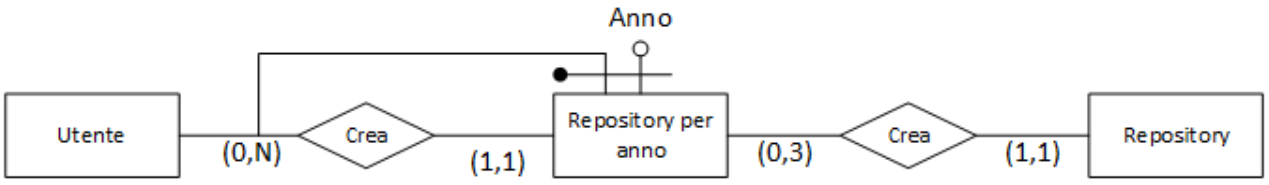
SOLUZIONE

Esercizio 2

Vincoli non esprimibili: "Il sistema chiude automaticamente le discussioni che non ricevono risposte per più di 90 giorni".



Vincolo aggiuntivo



Esercizio 3

Operazione 1

Con dato derivato

CONCETTO	ACCESSI	TIPO	DESCRIZIONE / MOTIVAZIONE DELL'OPERAZIONE
Volo	1	L	Leggo il totale

Senza dato derivato

CONCETTO	ACCESSI	TIPO	DESCRIZIONE / MOTIVAZIONE DELL'OPERAZIONE
Su	30	L	Leggo associazione con posti. Ci sono 2400 posti e 80 voli, ogni volo in media ha 2400/80 = 30 posti.
Posto	30	L	Leggo i posti
Per	15	L	Leggo associazione con biglietto. Ci sono 1200 biglietti e 2400 posti, quindi viene venduto il 1200/2400 = 0,5 (50%) dei posti. Abbiamo 30 posti, ci saranno 30*0,5 = 15 biglietti da leggere.
Biglietto	15	L	Leggo i prezzi dei biglietti

Accessi totali 90

Operazione 2

Con dato derivato

CONCETTO	ACCESSI	TIPO	DESCRIZIONE / MOTIVAZIONE DELL'OPERAZIONE
Biglietto	1	S	Inserisco su biglietto
Per	1	S	Associo biglietto a posto
Volo	1	L	Leggo il valore di "incasso totale"
Volo	1	S	Aggiorno il valore di "incasso totale"

Accessi totali 7

Senza dato derivato

CONCETTO	ACCESSI	TIPO	DESCRIZIONE / MOTIVAZIONE DELL'OPERAZIONE
Biglietto	1	S	Inserisco su biglietto
Per	1	S	Associo biglietto a posto

Accessi totali 4

Accessi totali con dato derivato (1*50+7*100) = 750
Accessi totali senza dato derivato (90*50+4*100) = 4900

Conviene mantenere il dato derivato!

Schema logico

Aereo (Num telaio, modello)

Aeroporto (Nome, indirizzo)

Pilota (CF, Brevetto, Ore volo, Nome, Cognome)
AK(Brevetto)

Hostess (CF, Nome, Cognome)

Volo (Partenza, Destinazione, DataOra, Num telaio, incasso totale)
FK Num telaio **REFERENCES** Aereo
FK Partenza **REFERENCES** Aeroporto
FK Destinazione **REFERENCES** Aeroporto

Ha (Partenza, Destinazione, DataOra, CF pilota)
FK CF_pilota **REFERENCES** Pilota
FK Partenza, Destinazione, DataOra **REFERENCES** Volo
Nota: almeno 2 piloti per volo

In (Partenza, Destinazione, DataOra, CF hostess)
FK CF_hostess **REFERENCES** Hostess
FK Partenza, Destinazione, DataOra **REFERENCES** Volo

Posto (Partenza, Destinazione, DataOra, Numero, Classe)
FK Partenza, Destinazione, DataOra **REFERENCES** Volo

Biglietto (CodB, Prezzo, Partenza, Destinazione, DataOra, Numero)
AK Partenza, Destinazione, DataOra, Numero
FK Partenza, Destinazione, DataOra, Numero **REFERENCES** Posto

Esercizio 4

A. Selezionare i nomi delle stazioni da cui nel 2010 sono partiti tutti i modelli di treni presenti. **(punti 2+2)**

Si prendono le tratte del 2010

$$S1 = \sigma_{data\_partenza \geq 2010-1-1 \; AND \; data\_partenza \leq 2010-12-31}(Tratta)$$

Si estraggono le stazioni di partenza e modelli dei treni

$$S2 = \pi_{partenza,modello}(Treno \bowtie S1)$$

Si divide S2 per i modelli di treni esistenti ottenendo così le stazioni da cui sono partiti tutti i modelli e si mette in join con le stazioni visto che si vogliono i nomi

$$\pi_{nome}\left(Stazione \bowtie \left(S2 \div \pi_{modello}(Treno)\right)\right)$$

```
SELECT DISTINCT nome
FROM stazione s
WHERE NOT EXISTS (
      SELECT *
      FROM treno t
      WHERE NOT EXISTS (
            SELECT *
            FROM tratta tr
            JOIN treno t1 ON tr.id_treno = t1.id_treno
            WHERE t1.modello = t.modello
            AND tr.partenza = s.cod_stazione
            AND YEAR(data_partenza) = 2010
      )
)
```

B. Selezionare i dati dei treni che sono passati per la città di Modena (partenza o arrivo) nel mese di gennaio del 2012. **(punti 2+2)**

Si prendono le stazioni di Modena

$$S1 = \sigma_{città='Modena'}(Stazione)$$

Si prendono le tratte di gennaio 2012

$$S2 = \sigma_{data\_partenza \geq '2012-01-01' \; AND \; data\_partenza \leq '2012-12-31'}(Tratta)$$

Infine si estraggono i treni che sono partiti/arrivati a Modena

$$Treno \ltimes \pi_{id\_treno}\left(S1 \bowtie_{S2.partenza=S1.codstazione} (S2)\right) \cup \left(S1 \bowtie_{S2.arrivo=S1.codstazione} (S2)\right)$$

```
SELECT DISTINCT t.*
FROM treno t
JOIN tratta tr ON t.id_treno = tr.id_treno
JOIN stazione sp ON tr.partenza = sp.cod_stazione
JOIN stazione sa ON tr.arrivo = sa.cod_stazione
WHERE tr.data_partenza >= '2012-01-01'
AND tr.data_partenza < '2012-01-31'
AND (sp.città = 'Modena' OR sa.città = 'Modena')
```

C. Selezionare i dati dei treni che nel 2016 hanno effettuato più ore di lavoro (somma delle durate delle tratte) **(punti 3)**

```
SELECT t.*, SUM(durata) AS ore_lavoro
FROM treno t
JOIN tratta tr ON t.id_treno = tr.id_treno
WHERE tr2.data_partenza >= ‘1-1-2016’
AND tr2.data_partenza <= ‘31-12-2016’
GROUP BY t.id_treno
HAVING SUM(durata) >= ALL (
      SELECT SUM(durata)
      FROM tratta tr2
      WHERE tr2.data_partenza >= ‘1-1-2016’
      AND tr2.data_partenza <= ‘31-12-2016’
      GROUP BY tr2.id_treno
)
```