

SECONDO PARZIALE - Basi di Dati e Lab - Basi di Dati A
10/06/2016

Corso di Laurea in Ingegneria Informatica
Prof. Sonia Bergamaschi
Ing. Laura Po

Esercizio 1 (6 punti)

- 1) Istruzione di grant: definizione ed esempio. (3 punti)
- 2) Operazione di divisione in algebra relazionale: definizione ed esempio. (3 punti)

Esercizio 2 (punti 27)

Sia dato il seguente schema relazionale di un software gestionale per una biblioteca.

Libro (cod_libro, titolo, autore, abstract, anno_publicazione, lingua)
AK: titolo, autore
Genere (cod_gen, genere, descrizione)
Libro_Genere (cod_libro, cod_gen)
FK: cod_libro **references** Libro
FK: cod_gen **references** Genere
Utente (num_tessera, cognome, nome, data_nascita)
Prestito(cod_libro, num_tessera, data_prestito, data_scadenza)
FK: cod_libro **references** Libro
FK: num_tessera **references** Utente
Prenotazione(cod_libro, num_tessera, data_prenotazione, data_scadenza, fl_valida)
FK: cod_libro **references** Libro
FK: num_tessera **references** Utente

Prenotazione in corso di validità fl_valida = 1, altrimenti fl_valida = 0.

Ci può essere solo 1 prenotazione valida in corso per libro.

- A. Si richiede di scrivere in algebra relazionale e in SQL le seguenti interrogazioni:
- A1) Mostrare gli utenti, che sono nati dopo l'anno 1999 e che hanno prenotato o si sono fatti prestare almeno un libro pubblicato prima del 1980. **(punti 2+2)**
 - A2) Selezionare gli utenti che non hanno preso in prestito alcun libro che ha preso in prestito Mario Rossi. **(punti 2+2)**
 - A3) Selezionare il nome e cognome degli utenti, i quali hanno preso in prestito tutti i libri scritti da Alessandro Manzoni. **(punti 2+2)**
- B. Si richiede di scrivere la seguente interrogazione B1 secondo le seguenti modalità:
- per gli studenti dall'A.A. 2011/2012 in poi in Calcolo Relazionale su tuple
 - per gli studenti di anni accademici precedenti in algebra relazionale e in SQL
- B1) Selezionare i nomi e cognomi degli utenti che hanno prenotato libri di genere "giallo" dal 01/05/2016 al 01/06/2016. **(3 punti)**

C. Si richiede di scrivere in SQL le seguenti interrogazioni:

- C1) Creare una vista, che mostri nome e cognome degli utenti che hanno preso in prestito almeno 5 libri di autori diversi. **(punti 4)**
- C2) Creare una vista, che per ogni utente mostri il titolo dell'ultimo libro che ha prenotato. **(punti 4)**
- C3) Trovare per ogni genere, il titolo del libro più prenotato, e a parità del numero di prenotazioni, mostrare il titolo del libro che è stato prenotato più recentemente. **(punti 4)**

Soluzione compito

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

A1 (() Mostrare gli utenti, che sono nati dopo l'anno 1999, i quali hanno fatto almeno una prenotazione o un prestito di un libro pubblicato prima del 1980. **(punti 2+2)**

Algebra relazionale:

$R1 : \pi_{num_tessera}(Prestito \bowtie \sigma_{anno_pubblicazione < 1990}(Libro))$

$R2 : \pi_{num_tessera}(Prenotazione \bowtie \sigma_{anno_pubblicazione < 1990}(Libro))$

$\sigma_{data_nascita > '31/12/1999'}(Utente) \bowtie (R1 \cup R2)$

Query SQL:

```
select u.*
from utenti u
where u.data_nascita > '31/12/1999'
and num_tessera in
(
    select p1.num_tessera
    from Prestito p1
    join Libro l on p1.cod_libro = l.cod_libro
    where anno_publicazione < 1980

    union

    select p1.num_tessera
    from Prenotazione p2
    join Libro l on p2.cod_libro = l.cod_libro
    where anno_publicazione < 1980
)
```

A2) Selezionare gli utenti, i quali non hanno preso in prestito alcun libro che ha preso in prestito Mario Rossi. **(punti 2+2)**

Algebra relazionale:

$R1 : Prestito \bowtie \sigma_{nome = 'Mario' \text{ AND } cognome = 'Rossi'}(Utente)$

$R2 : \pi_{num_tessera}(Prestito \bowtie \sigma_{cod_libro = .cod_libro} R1)$

$Utente \bowtie (\pi_{num_tessera}(Utente) - R2)$

Query SQL:

```
select *
from Utente u
where u.num_tessera not in (
    select
        p2.num_tessera
    from Utente ui
    join Prestito p1 on p1.num_tessera = ui.num_tessera
    join Prestito p2 on p2.cod_libro = p1.cod_libro
    where ui.nome = 'Mario'
    and ui.cognome = 'Rossi'
);
```

A3) Selezionare il nome e cognome degli utenti, i quali hanno preso in prestito tutti i libri scritti da Alessandro Manzoni. **(punti 2+2)**

Algebra relazionale:

$\pi_{nome, cognome}(Utente) \bowtie (\pi_{num_tessera, cod_libro}(Prestito) \div \pi_{cod_libro}(\sigma_{autore = 'Alessandro Manzoni'}(Libro)))$

Query SQL:

```
select
    u.nome,
    u.cognome
from Utente u
where not exists (
    select *
    from Libro l
    where l.autore = 'Alessandro Manzoni'
    and not exists (
        select *
        from Prestito p
        where p.cod_libro = l.cod_libro
        and p.num_tessera = u.num_tessera
    )
)
```

B. Si richiede di scrivere la seguente interrogazione B1 secondo le seguenti modalità:

- per gli studenti dall'A.A. 2011/2012 in poi in Calcolo Relazionale su tuple
- per gli studenti di anni accademici precedenti in algebra relazionale e in SQL

B1) Selezionare i nomi e cognomi degli utenti che hanno prenotato libri col genere “giallo” dal 01/05/2016 al 01/06/2016. **(3 punti)**

Calcolo Relazionale:

$\{u.(nome, cognome) \mid$
 $p(Prenotato), u(Utente), lg(Libro_Genere), g(Genere) \mid$
 $u.num_tessera = p.num_tessera \wedge p.cod_libro = lg.cod_libro \wedge$
 $g.cod_gen = lg.cod_gen \wedge g.genere = 'giallo' \wedge$
 $p.data_prenotazione \geq '01/05/2016' \wedge p.data_prenotazione \leq '01/06/2016'\}$

Algebra relazionale:

$R1 : Libro_Genere \bowtie \sigma_{genere = 'giallo'}(Genere)$

$R2 : \sigma_{data_prenotazione \geq '01/05/2016' \wedge data_prenotazione \leq '01/06/2016'}(Prenotazione) \bowtie R1$

$\pi_{nome, cognome}(Utente \bowtie R2)$

Query SQL:

```
select u.nome, u.cognome
from Prenotato p
join Utente u on u.num_tessera = p.num_tessera
join Libro_Genere lg on lg.cod_libro = p.cod_libro
join Genere g on g.cod_gen = lg.cod_gen
where g.genere = 'giallo'
and p.data_prenotazione between '01/05/2016' and '01/06/2016'.
```

C. Si richiede di scrivere in SQL le seguenti interrogazioni:

C1) Creare una vista, che mostri nome e cognome degli utenti che hanno preso in prestito almeno 5 libri di autori diversi. **(punti 4)**

```
create view c1 as
select u.nome, u.cognome
from Libro l
join Prestito p on p.cod_libro = l.cod_libro
join Utente u on u.num_tessera = p.num_tessera
group by u.num_tessera, u.nome, u.cognome
having count(distinct l.autore) >= 5
```

C2) Creare una vista, che per ogni utente mostri il titolo dell'ultimo libro che ha prenotato **(punti 4)**

```
create view c2 as
select p.num_tessera, l.titolo
from Libro l
join prenotato p on p.cod_libro = l.cod_libro
where data_prenotazione = (
    select max(data_prenotazione)
    from prenotazione pi
    where pi.num_tessera = p.num_tessera)
```

C3) Trovare per ogni genere, il titolo del libro più prenotato, a parità del numero di prenotazioni, mostrare il titolo del libro che è stato prenotato più recentemente. **(punti 4)**

```
select *
from Prenotazione p
join Libro_Genere lg on p.cod_libro = lg.cod_libro
join Genere g on g.cod_gen = lg.cod_gen
group by g.cod_gen, p.cod_libro
having count(*) >= all (
    select count(*)
    from Prenotazione p1
    join Libro_Genere lg1 on p1.cod_libro = lg1.cod_libro
    where p1.cod_gen = p.cod_gen
    group by p1.cod_libro
)
and max(p.data_prenotazione) >= all (
    select max(p2.data_prenotazione)
    from Prenotazione p2
    join Libro_Genere lg2 on p2.cod_libro = lg2.cod_libro
    group by g2.cod_gen, p2.cod_libro
    having count(*) >= all (
        select count(*)
        from Prenotazione p3
        join Libro_Genere lg3 on p1.cod_libro = lg3.cod_libro
        where p2.cod_gen = p3.cod_gen
        group by p3.cod_libro
    )
)
```