

SECONDO PARZIALE - Basi di Dati e Lab - Basi di Dati A
11/06/2015

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

Esercizio 1 (6 punti)

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

Esercizio 2 (punti 27)

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

Cliente(cod_cliente, nome, cognome, data_nascita, CF)

AK: CF

Prodotto(id_prodotto, nome_prodotto, tipologia, produttore)

Acquisto(cod_cliente, id_prodotto, quantita, data)

FK: cod_cliente **references** Cliente

FK: id_prodotto **references** Prodotto

Prezzo(id_prodotto, data_inizio_validita, data_fine_validita, prezzo)

FK: id_prodotto **references** Prodotto

Sul DBMS è definito un vincolo che impedisce che nello stesso intervallo di tempo ci siano 2 prezzi validi per lo stesso prodotto

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

- A1) Selezionare i nomi dei prodotti acquistati da Mario Rossi che hanno tipologia hi-tech o che hanno un prezzo d'acquisto nell'anno 2015 di almeno 500€.
(punti 2+2)
- A2) Selezionare i nomi e i cognomi dei clienti che hanno acquistato solo prodotti hi-tech (tipologia prodotto 'hi-tech'). **(punti 2+2)**
- A3) Selezionare il codice fiscale dei clienti che hanno acquistato tutti i prodotti della tipologia 'yogurt'. **(punti 2+2)**

B. Si richiede di scrivere in Calcolo Relazionale su tuple la seguente interrogazione **(3 punti)**

B1) Selezionare i nomi e cognomi dei clienti che hanno acquistato i prodotti del produttore 'Sony'.

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

C1) Creare una vista che riporti per ogni prodotto (visualizzare id e nome del prodotto) il prezzo massimo, la data d'inizio e fine validità. **(punti 4)**

C2) Selezionare tutti i dati dei clienti che hanno fatto acquisti di almeno 1000€ in una singola giornata. **(punti 4)**

C3) Selezionare per ogni prodotto (visualizzare id e nome del prodotto), la quantità venduta e il totale ricavato nell'anno 2014. **(punti 4)**

Soluzione

Esercizio 2

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

A1) Selezionare i nomi dei prodotti acquistati da Mario Rossi che hanno tipologia hi-tech o che hanno un prezzo d'acquisto nell'anno 2015 di almeno 500€. **(punti 2+2)**

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

R2: $\sigma_{\text{tipologia}='hi-tech'}(\text{Prodotto}) \bowtie \text{R1}$

R3: $(\text{Prodotto} \bowtie \sigma_{\text{prezzo}>500 \text{ AND } \text{data_inizio_validita} < '01/01/2016' \text{ AND } \text{data_fine_validita} > '31/12/2014'}(\text{Prezzo})) \bowtie \text{R1}$

$\pi_{\text{id_prodotto}, \text{nome_prodotto}}(\text{R2} \cup \text{R3})$

```
SELECT DISTINCT
    p.id_prodotto,
    p.nome_prodotto
FROM
    Prodotto p,
    Cliente c,
    Acquisto a
WHERE p.id_prodotto = a.id_prodotto
AND c.cod_cliente = a.cod_cliente
AND c.nome = 'Mario'
AND c.cognome = 'Rossi'
AND p.tipologia = 'hi-tech'

UNION

SELECT DISTINCT
    p.id_prodotto,
    p.nome_prodotto
FROM
    Prodotto p,
    Cliente c,
    Acquisto a,
    Prezzo pr
WHERE p.id_prodotto = a.id_prodotto
AND c.cod_cliente = a.cod_cliente
AND pr.id_prodotto = a.id_prodotto
AND c.nome = 'Mario'
AND c.cognome = 'Rossi'
AND pr.prezzo > 500
AND pr.data_inizio_validita < '01/01/2016'
AND pr.data_fine_validita > '31/12/2014'
```

Soluzione SQL alternativa:

```
SELECT DISTINCT
    p.id_prodotto,
```

```

        p.nome_prodotto
FROM Prodotto p
JOIN Acquisto a ON p.id_prodotto = a.id_prodotto
JOIN Cliente c ON c.cod_cliente = a.cod_cliente
LEFT OUTER JOIN Prezzo pr ON pr.id_prodotto = a.id_prodotto
WHERE c.nome = 'Mario'
AND c.cognome = 'Rossi'
AND (p.tipologia = 'hi-tech'
      OR
      (pr.prezzo > 500
       AND pr.data_inizio_validita < '01/01/2016' AND pr.data_fine_validita > '31/12/2014'))

```

A2) Selezionare i nomi e i cognomi dei clienti che hanno acquistato solo prodotti hi-tech (tipologia prodotto 'hi-tech'). **(punti 2+2)**

R1: $\pi_{\text{cod_cliente}}(\text{Acquisto})$

R2: $\pi_{\text{cod_cliente}}(\text{Acquisto} \bowtie \sigma_{\text{tipologia} \neq \text{'hi-tech'}}(\text{Prodotto}))$

$\pi_{\text{nome, cognome}}(\text{Cliente} \bowtie (R1 - R2))$

```

SELECT DISTINCT
    c.nome,
    c.cognome
FROM
    Cliente c,
    Acquisto a
WHERE c.cod_cliente = a.cod_cliente
AND c.cod_cliente not in (
    SELECT DISTINCT a1.cod_cliente
    FROM
        Acquisto a1,
        Prodotto p2
    Where a1.id_prodotto = p2.id_prodotto
    AND p2.tipologia <> 'hi-tech')

```

A3) Selezionare il codice fiscale dei clienti che hanno acquistato tutti i prodotti della tipologia 'yogurt'. **(punti 2+2)**

R1: $\pi_{\text{id_prodotto}}(\sigma_{\text{tipologia}=\text{'yogurt'}}(\text{Prodotto}))$

R2: $\pi_{\text{cod_cliente, id_prodotto}}(\text{Acquisto})$

$\pi_{\text{CF}}(\pi_{\text{CF, cod_cliente}}(\text{Cliente}) \bowtie (R2 \div R1))$

Selezionare il codice fiscale dei clienti per i quali non esistono prodotti della tipologia yogurt che non abbiano acquistato.

```

SELECT DISTINCT c.CF
FROM Cliente c
WHERE NOT EXISTS (

```

```

SELECT *
FROM Prodotto p
WHERE p.tipologia = 'yogurt'
AND NOT EXISTS (
    SELECT *
    FROM Acquisto a
    WHERE a.id_prodotto = p.id_prodotto
    AND a.cod_cliente = c.cod_cliente)

```

B. Si richiede di scrivere in Calcolo Relazionale su tuple la seguente interrogazione **(3 punti)**

B1) Selezionare i nomi e cognomi dei clienti che hanno acquistato i prodotti del produttore 'Sony'.

```

{c.(nome, cognome) |
c(Cliente), a(Acquisto), p(Prodotto) |
c.cod_cliente = a.cod_cliente ∧ p.id_prodotto = a.id_prodotto ∧ p.produttore = 'Sony'}

```

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

C1) Creare una vista che riporti per ogni prodotto (visualizzare id e nome del prodotto) il prezzo massimo, la data d'inizio e fine validità. **(punti 4)**

```

CREATE VIEW max_price AS
SELECT
    p.id_prodotto,
    p.nome_prodotto,
    pr.prezzo AS prezzo_massimo,
    pr.data_inizio_validita,
    pr.data_fine_validita,
FROM
    Prodotto p,
    Prezzo pr
WHERE p.id_prodotto = pr.id_prodotto
AND pr.prezzo = (
    SELECT MAX(pr1.prezzo)
    FROM Prezzo pr1
    WHERE pr.id_prodotto = pr1.id_prodotto)
-- nel caso in cui per un determinato prodotto ci siano più prezzi uguali li prenderei tutti
-- Se aggiungo la seguente condizione (opzionale), prendo il più recente dei prezzi massimi
AND pr.data_inizio_validita = (
    SELECT MAX(pr2.data_inizio_validita)
    FROM Prezzo pr2
    WHERE pr2.id_prodotto = pr.id_prodotto
    AND pr.prezzo = pr2.prezzo)

```

C2) Selezionare tutti i dati dei clienti che hanno fatto acquisti di almeno 1000€ in una singola giornata. **(punti 4)**

```

SELECT c.*
FROM
    Cliente c
WHERE c.cod_cliente IN (
    SELECT DISTINCT
        a.cod_cliente
    FROM
        Acquisto a,
        Prezzo p
    WHERE p.id_prodotto = a.id_prodotto
    AND a.data BETWEEN pr.data_inizio_validita AND pr.data_fine_validita
    GROUP BY a.cod_cliente, a.data
    HAVING SUM(a.quantita * p.prezzo) >= 1000)

```

C3) Selezionare per ogni prodotto (visualizzare id e nome del prodotto), la quantità venduta e il totale ricavato nell'anno 2014. **(punti 4)**

```

SELECT
    p.id_prodotto,
    p.nome_prodotto,
    SUM(a.quantita),
    SUM(a.quantita * pr.prezzo)
FROM
    Prodotto p,
    Acquisto a,
    Prezzo pr
WHERE p.id_prodotto = a.id_prodotto
AND p.id_prodotto = pr.id_prodotto
AND a.data BETWEEN pr.data_inizio_validita AND pr.data_fine_validita
AND a.data BETWEEN '01/01/2014' AND '31/12/2014'
GROUP BY p.id_prodotto, p.nome_prodotto

```