

SECONDO PARZIALE – Basi di Dati e Lab

07/06/2019

Corso di Laurea in Ingegneria Informatica

Prof. Sonia Bergamaschi

Esercizio 1 (punti 8)

È necessario totalizzare almeno 4 punti in questo esercizio affinché il resto della prova venga valutato

- Sintassi di interrogazioni innestate. (4 punti)
- Definizione dell'operatore di divisione nell'algebra relazionale. (4 punti)

Esercizio 2 (punti 25)

Sia dato il seguente schema relazionale di un sistema di gestione ordini.

Fornitore (nomeF, indirizzo, città, telefono)

Prodotto (codP, nome, prezzo)

Fornisce (codP, nomeF)

FK codP **REFERENCES** Prodotto

FK nomeF **REFERENCES** Fornitore

Cliente (idC, nome, cognome, indirizzo, città)

Ordine (codO, data, idC, nomeF)

FK idC **REFERENCES** Cliente **NOT NULL**

FK nomeF **REFERENCES** Fornitore **NOT NULL**

Riga_ordine (codO, codP, quantità)

FK codO **REFERENCES** Ordine

FK codP **REFERENCES** Prodotto

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

- Selezionare i clienti di Milano che hanno acquistato prodotti con un prezzo superiore a 100€. (punti 2+2)
- Selezionare i prodotti che sono forniti da tutti i fornitori di Modena. (punti 2+2)
- Selezionare i clienti di Bologna che non hanno mai effettuato ordini da fornitori di Modena. (punti 2+2)

B) Si richiede di scrivere in SQL le seguenti interrogazioni:

- Selezionare nome e cognome dei clienti che nel 2018 hanno effettuato almeno 10 ordini. (punti 3)
- Selezionare nome e cognome del cliente che ha effettuato l'ordine più costoso. (punti 3)
- Creare una vista che mostri per ogni cliente ed ordine, il numero di prodotti distinti che ha acquistato e l'importo complessivo speso. (punti 3)
- Selezionare i clienti che nel 2018 hanno effettuato ordini da fornitori di Modena per un importo complessivo superiore agli ordini effettuati da fornitori di Modena nel 2017. **Suggerimento:** è possibile aiutarsi con una vista che selezioni per ogni cliente l'importo speso da fornitori di Modena per ogni anno. (punti 4)

- 1) Selezionare i clienti di Milano che hanno acquistato prodotti con un prezzo superiore a 100€.
(punti 2+2)

$$S1 = \pi_{idC} (Ordine \bowtie Riga_ordine \bowtie \sigma_{prezzo > 100}(Prodotto)) \\ \sigma_{città='Milano'}(Cliente) \bowtie S1$$

```
SELECT DISTINCT c.*
FROM cliente c
JOIN ordine o ON c.idC = o.idC
JOIN riga_ordine r ON r.codO = o.codO
JOIN prodotto p ON r.codP = p.codP
WHERE p.prezzo > 100 AND c.città = 'Milano'
```

- 2) Selezionare i prodotti che sono forniti da tutti i fornitori di Modena. (punti 2+2)

$$S1 = \pi_{nomeF} (\sigma_{città='Modena'}(Fornitore)) \\ Prodotto \bowtie (Fornisce \div S1)$$

Selezionare i prodotti per cui NON ESISTE alcun fornitore di Modena che NON li fornisca.

```
SELECT *
FROM prodotto p
WHERE NOT EXISTS (
    SELECT *
    FROM fornitore f
    WHERE città = 'Modena'
    AND NOT EXISTS (
        SELECT *
        FROM fornisce fo
        WHERE fo.codP = p.codP
        AND fo.nomeF = f.nomeF
    )
)
```

- 3) Selezionare i clienti di Bologna che non hanno mai effettuato ordini da fornitori di Modena.
(punti 2+2)

$$S1 = \pi_{idC} (Ordine \bowtie \sigma_{città='Modena'}(Fornitore)) \\ S2 = \sigma_{città='Bologna'}(Cliente) \\ S2 \bowtie (\pi_{idC}(S2) - S1)$$

```
SELECT *
FROM cliente
WHERE città = 'Bologna'
AND idC NOT IN (
    SELECT idC
    FROM ordine o
    JOIN fornitore f ON o.nomeF = f.nomeF
    WHERE f.città = 'Modena'
)
```

4) Selezionare nome e cognome dei clienti che nel 2018 hanno effettuato almeno 10 ordini. (punti 3)

```
SELECT nome, cognome
FROM cliente c
JOIN ordine o ON c.idC = o.idC
WHERE YEAR(o.data) = 2018
GROUP BY c.idC, c.nome, c.cognome
HAVING COUNT(*) >= 10
```

5) Selezionare nome e cognome del cliente che ha effettuato l'ordine più costoso. (punti 3)

```
SELECT nome, cognome
FROM cliente c
JOIN ordine o ON c.idC = o.idC
JOIN riga_ordine r ON r.codO = o.codO
JOIN prodotto p ON r.codP = p.codP
GROUP BY c.nome, c.cognome, o.codO
HAVING SUM(r.quantità*p.prezzo) >= ALL(
    SELECT SUM(r1.quantità*p1.prezzo)
    FROM riga_ordine r1
    JOIN prodotto p1 ON p1.codP = r.codP
    GROUP BY r1.codO
)
```

Oppure con una vista

```
CREATE VIEW costo_ordine_cliente AS
    SELECT o.codO, o.idC, SUM(r.quantità*p.prezzo) AS totale_ordine
    FROM ordine o
    JOIN riga_ordine r ON r.codO = o.codO
    JOIN prodotto p ON p.codP = r.codP
    GROUP BY o.codO, o.idC;
```

```
SELECT nome, cognome
FROM cliente c
JOIN costo_ordine_cliente co ON co.idC = c.idC
WHERE totale_ordine >= ALL(
    SELECT totale_ordine
    FROM costo_ordine_cliente
)
```

- 6) Creare una vista che mostri per ogni cliente ed ordine, il numero di prodotti distinti che ha acquistato e l'importo complessivo speso. **(punti 3)**

```
CREATE VIEW stats AS
    SELECT c.idC, nome, cognome, codO, COUNT(*codP) AS num_prodotti,
SUM(quantità*prezzo) AS spesa_totale
    FROM cliente c
    JOIN ordine o ON c.idC = o.idC
    JOIN riga_ordine r ON o.codO = r.codO
    JOIN prodotto p ON r.codP = p.codP
    GROUP BY idC, nome, cognome, codO
```

** si noti che DISTINCT non è necessario in quanto in ogni ordine un prodotto può comparire una volta sola (chiave riga_ordine è codO, codP)*

- 7) Selezionare i clienti che nel 2018 hanno effettuato ordini da fornitori di Modena per un importo complessivo superiore agli ordini effettuati da fornitori di Modena nel 2017.
Suggerimento: è possibile aiutarsi con una vista che selezioni per ogni cliente l'importo speso con fornitori di Modena per ogni anno. **(punti 4)**

```
CREATE VIEW spesa_annua AS
    SELECT c.idC, nome, cognome, YEAR(o.data_ordine) AS anno,
SUM(r.quantità*p.prezzo) AS importo
    FROM cliente c
    JOIN ordine o ON c.idC = o.idC
    JOIN fornitore f ON o.nomeF = f.nomeF
    JOIN riga_ordine r ON r.codO = p.codO
    JOIN prodotto p ON r.codP = p.codP
    WHERE f.città = 'Modena'
    GROUP BY c.idC, nome, cognome, YEAR(o.data_ordine)
```

```
SELECT idC, nome, cognome
FROM spesa_annua s1
WHERE anno = 2018
AND importo > (
    SELECT importo
    FROM spesa_annua s2
    WHERE s2.anno = 2017
    AND s2.idC = s1.idC
)
```