

SECONDO PARZIALE - Basi di Dati e Lab

27/05/2020

Corso di Laurea in Ingegneria Informatica

Prof. Sonia Bergamaschi

Sia dato il seguente schema relazionale relativo alla gestione di un gruppo di parchi di divertimento.

Parco (idP, città, tipo)

AK: città

Attrazione(nomeA, livello, num_posti)

Contiene(idP, nomeA)

FK: idP **REFERENCES** Parco

FK: nomeA **REFERENCES** Attrazione

Persona (CF, nome, cognome, nazionalità, età)

Ticket (CF, idP, data, tipo, prezzo)

FK: CF **REFERENCES** Persona

FK: idP **REFERENCES** Parco

Visita (CF, idP, data, nomeA, ora)

FK: CF, idP, data, **REFERENCES** Ticket

FK: nomeA **REFERENCES** Attrazione

Esercizio 1 (punti 18)

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

1. Selezionare i dati delle persone di età superiore a 20 anni che hanno acquistato un ticket per un parco di tipo "acquatico". **(punti 3+3)**
2. Selezionare le città dei parchi che non possiedono un'attrazione di nome "torri gemelle". **(punti 3+3)**
3. Selezionare il nome e il cognome delle persone che hanno visitato tutte le attrazioni di livello "8". **(punti 3+3)**

Esercizio 2 (punti 15)

Si richiede di scrivere in SQL la seguente interrogazione (può essere risolta anche con più viste):

1. Selezionare i parchi che hanno venduto ticket a persone di nazionalità italiana oppure che contengono un'attrazione con nome "ottovolante". **(punti 3)**
2. Selezionare le persone che hanno visitato solo parchi che contengono solo attrazioni di livello superiore a "5". **(punti 3)**
3. Per ogni parco riportare il numero totale di biglietti venduti, l'incasso totale e il numero distinto di persone che hanno visitato il parco. I valori devono essere calcolati e riportati separatamente per ogni anno. **(punti 4)**
4. Creare una vista che mostri per ogni nazionalità il numero distinto di parchi e di attrazioni visitati dal 2017 al 2019 da persone di quella nazionalità. Devono essere riportate solamente le nazionalità che hanno più di 100 persone di età uguale o maggiore di 18. **(punti 5)**

Esercizio 1

1. Selezionare i dati delle persone di età superiore a 20 anni che hanno acquistato un ticket per un parco di tipo "acquatico".

$$\sigma_{età>20}(Persona) \bowtie Ticket \bowtie \sigma_{tipo='acquatico'}(Parco)$$

```
SELECT DISTINCT P.*
FROM Persona P
JOIN Ticket T ON (P.CF = T.TF)
JOIN Parco P ON (T.idP = P.idP)
WHERE P.età > 20 AND P.tipo = 'acquatico'
```

2. Selezionare le città dei parchi che non possiedono un'attrazione di nome "torri gemelle".

$$S1 = \pi_{città}(Parco \bowtie \sigma_{nomeA='torri\ gemelle'}(Contiene)) \\ \pi_{città}(Parco) - S1$$

oppure

$$S1 = \pi_{idP}(\sigma_{nomeA='torri\ gemelle'}(Contiene)) \\ \pi_{città}(Parco \bowtie (\pi_{idP}(Parco) - S1))$$

```
SELECT P.città
FROM Parco P
WHERE P.idP NOT IN (
    SELECT C.idP
    FROM Contiene C
    WHERE C.nomeA = 'torri gemelle'
)
```

3. Selezionare il nome e il cognome delle persone che hanno visitato tutte le attrazioni di livello "8".

$$S1 = \pi_{CF,nomeA}(Visita) \div \pi_{nomeA}(\sigma_{livello=8}(Attrazione)) \\ \pi_{nome,cognome}(Persona \bowtie S1)$$

Selezionare il nome e il cognome delle persone per le quali non esiste un'attrazione di livello "8" che non abbiano visitato.

```
SELECT P.nome, P.cognome
FROM Persona P
WHERE NOT EXISTS (
    SELECT *
    FROM Attrazione A
    WHERE A.livello = 8
    AND NOT EXISTS (
        SELECT *
        FROM Visita V
        WHERE V.CF = P.CF AND V.nomeA = A.nomeA
    )
)
```

)
)

Esercizio 2

1. Selezionare i parchi che hanno venduto ticket a persone di nazionalità italiana oppure che contengono un'attrazione con nome "ottovolante".

```
SELECT P.*  
FROM Parco P  
JOIN Ticket T ON V.idP = P.idP  
JOIN Persona PE ON PE.CF = T.CF  
WHERE PE.nazionalità = 'italiana'
```

UNION

```
SELECT P1.*  
FROM Parco P1  
JOIN Contiene C ON C.idP = P1.idP  
WHERE C.nomeA = 'ottovolante'
```

Non si può risolvere con diversi JOIN mettendo in OR le due condizioni perché se un parco non ha l'ottovolante ma è stato visitato da italiani non esce nel risultato, idem se non ha l'ottovolante ma è stato visitato da italiani.
Si può, eventualmente, fare con un OUTER JOIN, es:

```
SELECT DISTINCT P.*  
FROM PARCO P  
LEFT JOIN Contiene C ON C.idP = P.idP  
LEFT JOIN Attrazione A ON A.nomeA = C.nomeA  
LEFT JOIN Visita V ON V.idP = P.idP  
LEFT JOIN Persona PE ON PE.CF = V.CF  
WHERE A.nomeA = 'ottovolante' OR PE.nazionalità = 'italiana'
```

2. Selezionare le persone che hanno visitato solo parchi che contengono solo attrazioni di livello superiore a "5"

```
SELECT P.*  
FROM Persona P  
WHERE CF NOT IN (  
    SELECT CF  
    FROM Visita V  
    JOIN Contiene C ON C.idP = V.idP  
    JOIN Attrazione A ON A.nomeA = C1.nomeA  
    WHERE A.livello <= 5  
)
```

3. Per ogni parco riportare il numero totale di biglietti venduti, l'incasso totale e il numero distinto di persone che hanno visitato il parco. I valori devono essere calcolati e riportati separatamente per ogni anno.

```
SELECT idP, YEAR(data) AS anno, COUNT(*) AS biglietti_venduti,  
SUM(Prezzo) AS incasso, COUNT(DISTINCT CF) AS visitatori  
FROM ticket  
GROUP BY idP, YEAR(data)
```

4. Creare una vista che mostri per ogni nazionalità di una persona il numero distinto di parchi e di attrazioni visitati dal 2017 al 2019 da persone di quella nazionalità. Devono essere riportate solamente le nazionalità che ha più di 100 persone di età uguale o maggiore di 18.

```
CREATE VIEW Nazionalità_100 AS  
SELECT nazionalità  
FROM Persona  
WHERE età >= 18  
GROUP BY nazionalità  
HAVING COUNT(*) > 100
```

```
CREATE VIEW Parchi_Nazionalità_100 AS  
SELECT P.nazionalità, COUNT(DISTINCT V.idP) AS num_parchi,  
COUNT(DISTINCT V.nomeA) AS num_attrazioni  
FROM Persona P  
JOIN Nazionalità_100 N ON P.nazionalità = N.nazionalità  
JOIN Visita V ON P.CF = V.CF  
WHERE V.data >= '01-01.2017' AND V.data <= '31-12-2019'  
GROUP BY P.nazionalità
```

Oppure:

```
CREATE VIEW Parchi_Nazionalità_100 AS  
SELECT P.nazionalità, COUNT(DISTINCT V.idP) AS num_parchi,  
COUNT(DISTINCT V.nomeA) AS num_attrazioni  
FROM Persona P  
JOIN Nazionalità_100 N ON P.nazionalità = N.nazionalità  
JOIN Visita V ON P.CF = V.CF  
WHERE V.data >= '01-01.2017'  
AND V.data <= '31-12-2019'  
AND P.nazionalità IN (  
    SELECT nazionalità  
    FROM Persona  
    WHERE età >= 18  
    GROUP BY nazionalità  
    HAVING COUNT(*) > 100  
)  
GROUP BY P.nazionalità
```