

Fondamenti di Basi di Dati

Esercitazione 6a - Operatori di aggregazione

Database

Il database **Telefonia.sqlite** è lo stesso usato per le esercitazioni precedenti. Come sempre, è consigliabile riportare su carta lo schema logico del DB per rendere più agevole la formulazione delle interrogazioni.

Interrogazioni

Canone più alto

Determinare il canone del piano telefonico avente il canone più alto.

I canoni sono rappresentati nella tabella **Piani**. In essa occorre cercare la tupla con il valore massimo dell'attributo **canone**:

```
select max(canone)
from Piani
```

Eeguire l'interrogazione e verificare che il risultato corrisponda con il risultato atteso.

Prima telefonata fatta nel mese di febbraio 2012

Cercare l'orario in cui è stata effettuata la prima telefonata di febbraio 2012.

La query è simile alla precedente: bisogna estrarre l'orario minimo dalla tabella **Telefonate**. Però occorre selezionare solo le tuple che rappresentano telefonate effettuate nel mese di febbraio, attraverso un'opportuna clausola **where**.

Eeguire l'interrogazione e verificare che il risultato sia quello atteso, ossia {'2012-02-10 15:20:00'}.

Statistiche sulle telefonate di Mario Rossi

Determinare la durata minima, media e massima, e la durata complessiva, delle telefonate di Mario Rossi.

Anche in questo caso occorre selezionare alcune tuple della tabella **Telefonate**, e calcolare minimo, media e massimo delle loro durate. Però cognome e nome del chiamante si trovano nella tabella **Abbonati**: dunque occorre effettuare un join tra le due tabelle.

Eeguire l'interrogazione e verificare che il risultato sia quello atteso, ossia {(0, 1, 2, 3)}.

Statistiche sulle telefonate di ciascun abbonato

Per ciascun abbonato, determinare il cognome, il nome, e la durata minima, media e massima delle telefonate che ha effettuato, e il numero di telefonate effettuate.

La query si presenta simile alla precedente; però, invece che selezionare un solo abbonato, li consideriamo tutti, e raggruppiamo le tuple per abbonato.

Procediamo per passi. Per il momento consideriamo solo la tabella **Telefonate** e calcoliamo durata minima, media e massima di ciascun numero di telefono chiamante (ossia raggruppiamo per chiamante):

```
select chiamante, min(durata), avg(durata), max(durata), count(*)
from Telefonate
group by chiamante
```

Dopo aver eseguito la query, ne scriviamo una versione arricchita, che mostri, invece del numero del chiamante, il suo nome, ricorrendo a un join con la tabella Abbonati:

```
select
  cognome, nome,
  min(durata), avg(durata), max(durata), count(*)
from Telefonate, Abbonati
where chiamante = telefono
group by chiamante
```

Abbiamo potuto inserire nella clausola select gli attributi **cognome** e **nome** perché essi dipendono funzionalmente dall'attributo del raggruppamento **chiamante**: dato il numero di telefono del chiamante, ne sono determinati univocamente il cognome e il nome.

Osserviamo però che non tutti gli abbonati compaiono nel risultato. Per esempio Luigi Bianchi non è presente. Perché? Come ovviare a questo inconveniente?

...

Perché il join è incompleto: solo gli abbonati che hanno effettuato almeno una telefonata sono considerati. Per far comparire tutti gli abbonati occorre effettuare un join esterno tra le due tabelle. Siccome non tutti gli abbonati hanno effettuato una telefonata, non tutti i numeri di telefono degli abbonati (**Abbonati.telefono**) trovano corrispondenza in almeno un numero chiamante di una telefonata (**Telefonate.chiamante**). Perciò raggruppiamo per **Abbonati.telefono**, attributo che è senz'altro definito in tutte le tuple, e facciamo un join esterno:

```
select
  cognome, nome,
  min(durata), avg(durata), max(durata), count(chiamante)
from Abbonati left join Telefonate
  on chiamante = telefono
group by telefono
```

Osserviamo che abbiamo sostituito **count(*)** con **count(chiamante)** per evitare di contare le tuple in cui **chiamante** vale NULL, cioè le tuple che nascono dal join esterno quando a un abbonato non è possibile associare alcuna telefonata.

Statistiche sulle telefonate degli abbonati chiacchieroni

Per ciascun abbonato che abbia fatto almeno 2 telefonate, determinare il cognome, il nome, e la durata minima, media e massima delle telefonate che ha effettuato, e il numero di telefonate effettuate.

La query è del tutto analoga alla precedente; occorre aggiungere una condizione sull'attributo aggregato count(chiamante) che faccia sì che il DBMS selezioni solo le tuple aggregate in cui il conteggio delle telefonate sia maggiore o uguale a 2. Quindi:

```
select  cognome,  nome,  chiamante,  min(durata),  avg(durata),
max(durata), count(chiamante)
from Abbonati left join Telefonate
    on chiamante = telefono
group by telefono
having count(chiamante) >= 2
```

(In realtà in questo caso il join esterno è superfluo, in quanto ci interessano solo abbonati che hanno fatto qualche telefonata.)

Calcolo delle bollette di marzo 2012

Per ciascun abbonato determinare cognome, nome, numero di telefonate fatte nel mese di marzo 2012, e importo da pagare. Per semplicità, limitarsi agli abbonati che hanno fatto almeno una telefonata a marzo. Considerare solo le telefonate andate a buon fine (cioè con durata maggiore di 0).

Per eseguire questa query occorrono due join: Abbonati – Telefonate e Abbonati – Piani. Quest'ultima tabella si rende necessaria per leggere le caratteristiche del piano tariffario degli abbonati, e poter calcolare l'importo da pagare secondo la formula:

$$\text{canone} + \text{numero_telefonate} * \text{scatto} + \text{durata_totale_telefonate} * \text{tariffa}$$

Il numero di telefonate si calcola con l'operatore aggregato count(*); la durata totale con sum(durata).

Provare a scrivere ed eseguire la query.

Una soluzione è la seguente:

```
select
    A.cognome,
    A.nome,
    canone + count(*) * scatto + sum(durata) * tariffa as importo
from
    Abbonati A,
    Telefonate T,
    Piani P
where
    A.nome_piano = P.nome
    and A.anno_piano = P.anno
    and A.telefono = T.chiamante
    and orario >= '2012-03-01'
    and orario < '2012-04-01'
    and T.durata > 0
group by
    A.telefono
```