

Infomatica (SGI) - Esercizi SQL – Esercizio 6 (Esercizi vari su Query)

Prova del 29/01/2016. Testo domanda. Date le tabelle seguenti (tra parentesi i nomi degli attributi, le CHIAVI PRIMARIE sono scritte interamente in maiuscolo, le Foreign Key hanno solo la prima lettera maiuscola):

- Volo: (ID_VOLO, Id_citta_partenza, Id_citta_arrivo, Id_pilota)
- Passeggero: (ID_PASSEGGERO, nome, cognome)
- Prenotazione: (ID_PRENOTAZIONE, Id_passeggero, Id_volo, data, prezzo_biglietto)
- Pilota: (ID_PILOTA, nome, cognome, stipendio_annuo)
- Città: (ID_CITTA, nome)

Assumete che nella base di dati siano contenuti i dati raccolti in un anno di attività. Nello stesso anno, ogni volo è sempre stato condotto da uno e un solo pilota (sempre lo stesso). Con un'unica query, visualizzate, per ogni volo, l'Id_volo, la città di partenza, la città di arrivo, il rapporto tra la somma incassata con i biglietti e lo stipendio annuo speso per il pilota. Non mostrate i dati per quei voli che hanno un rapporto superiore a 100. Mostrate solamente una riga per ogni volo nella tabella risultato.

Prova del 19/02/2016. Testo domanda. Date le tabelle seguenti di un database (tra parentesi i nomi degli attributi, le CHIAVI PRIMARIE sono scritte interamente in maiuscolo, le Foreign Key hanno solo la prima lettera maiuscola):

- Opera: (ID_OPERA, titolo)
- Rappresentazione: (ID_RAPPRESENTAZIONE, giorno, Id_opera)
- Biglietto: (ID_BIGLIETTO, prezzo, Id_rappresentazione, Id_spettatore)
- Spettatore: (ID_SPETTATORE, nome, cognome)
- Attore: (ID_ATTORE, nome, cognome)
- Recitazionein: (ID_ATTORE, ID_OPERA)
- Il database contiene i dati di un anno delle opere messe in scena in un teatro, dei biglietti venduti e degli attori che hanno recitato.

Scrivete una query SQL che mostri per ogni opera, il rapporto tra il totale del prezzo dei biglietti incassati da un'opera e il numero di attori che recitano nell'opera stessa.

Nella tabella risultato ogni opera dovrà apparire una sola volta, inoltre dovete visualizzare solo i dati delle opere che hanno più di 10 attori.

Fate attenzione alla granularità delle diverse tabelle.

Prova del 13/07/2016. Testo domanda. Date le tabelle seguenti di un database (tra parentesi i nomi degli attributi, le CHIAVI PRIMARIE sono scritte interamente in maiuscolo, le Foreign Key hanno solo la prima lettera maiuscola):

- AnagraficaProdotto: (ID_PRODOTTO, descrizione)
- AnagraficaCliente: (ID_CLIENTE, nome, cognome)
- Fattorino: (ID_FATTORINO, nome, cognome)
- Vendita: (ID_VENDITA, quantita, prezzo_unitario, Id_prodotto, Id_cliente, Id_consegna)
- Consegna: (ID_CONSEGNA, indirizzo, giorno, Id_fattorino)

Il database contiene i dati delle vendite a domicilio effettuate da un sito di e-commerce. Ogni cliente è registrato nell'AnagraficaCliente.

Scrivete una query SQL che mostri per ogni fattorino, il rapporto tra il totale del prezzo incassato con le vendite consegnate dal fattorino e il numero di consegne effettuate dal fattorino stesso.

Nella tabella risultato dovranno apparire anche il nome e il cognome del fattorino, inoltre ogni fattorino dovrà apparire una e una sola volta nel risultato. Prestate attenzione al fatto che una consegna può raggruppare più prodotti e anche più ordini (es., un utente può acquistare diverse volte e ricevere tutto con un'unica consegna, oppure acquisti di utenti diversi che abitano nello stesso condominio possono venire consegnati attraverso un'unica consegna).

Prova del 14/09/2016. Testo domanda. Date le tabelle seguenti di un database (tra parentesi i nomi degli attributi, le CHIAVI PRIMARIE sono scritte interamente in maiuscolo, le Foreign Key hanno solo la prima lettera maiuscola):

- CasaFarmaceutica: (ID_CASA, nome)
- Farmaco: (ID_FARMACO, nome, Id_casa)
- Farmacia: (ID_NEGOZIO, nome, indirizzo)
- Vendita: (Id_farmaco, Id_farmacia, n_confezioni, prezzo_totale, giorno)

Il database contiene i dati delle vendite di medicinali effettuate da un gruppo di farmacie. Ogni record della tabella vendita riporta l'acquisto di un farmaco effettuato da un cliente. L'attributo n_confezioni memorizza il numero di confezioni acquistate dal cliente, l'attributo prezzo_totale rappresenta il prezzo totale pagato dal cliente per acquistare le confezioni.

Scrivete una query SQL che mostri per ogni farmaco, il rapporto tra soldi incassati e numero di confezioni vendute. Per ogni farmaco dovete visualizzare il nome del farmaco stesso e della casa farmaceutica che lo vende. Nel risultato devono essere visualizzati solo quei farmaci il cui rapporto è maggiore o uguale a 50.

Prova del 24/11/2016 – Domanda aperta 1

Date le tabelle seguenti di un database (tra parentesi i nomi degli attributi, le CHIAVI PRIMARIE sono scritte interamente in maiuscolo, le Foreign Key hanno la prima lettera maiuscola, gli attributi id_... presenti nelle chiavi primarie, presi singolarmente fungono da foreign key):

- Albergo: (ID_ALBERGO, nome, indirizzo)
- Cliente: (ID_CLIENTE, nome, cognome)
- Camera: (ID_ALBERGO, ID_CAMERA, n_letti)
- Prenotazione: (Id_cliente, Id_albergo, Id_camera, da_data, a_data, n_personone)
- Pulizia: (ID_ALBERGO, ID_CAMERA, DATA, ore)

Il database contiene i dati delle prenotazioni di camere effettuate dai clienti di una catena di alberghi e delle ore impiegate per pulire le camere. La tabella camere ha una chiave formata da due attributi perché in ogni albergo id_camera ha sempre valori compresi tra 1 e il numero di camere dello specifico albergo. Dato che diverse camere (in diversi alberghi) possono avere id_camera uguale a 1, allora per identificare univocamente una camera occorre indicare anche l'id_albergo. Gli attributi da_data, a_data della tabella prenotazioni contengono le date rispettivamente del primo e dell'ultimo giorno di prenotazione, l'attributo n_personone della stessa tabella contiene il numero di persone per cui viene effettuata la prenotazione (es., una camera a due letti potrebbe essere prenotata da 1 sola persona). La tabella pulizia indica, per ogni camera e per ogni giorno, il numero di ore impiegate per pulirla (le ore sono valori interi, possibili valori sono 1, 2, ...). Gli id nelle diverse tabelle sono sempre valori interi.

Scrivete una query SQL che mostri per ogni albergo il rapporto tra il numero di ore impiegate per pulire le camere e il numero di prenotazioni effettuate nell'albergo. Nel risultato i dati di un albergo devono essere riportati una sola volta, per ogni albergo devono essere mostrati anche id_albergo, nome e indirizzo. Dal risultato devono essere esclusi quegli alberghi che hanno un rapporto inferiore a 2.

Soluzioni Prova del 29/01/2016

```
select v.id_volo, v.id_citta_partenza, v.id_citta_arrivo, i.tot/p.stipendio_annuo as rap
from (
  select id_volo, sum(prezzo_biglietto) as tot
  from prenotazione as pr
  group by id_volo
) as i, volo as v, pilota as p
where i.id_volo = v.id_volo and v.id_pilota=p.id_pilota and i.tot/stipendio_annuo>100;
```

Altra possibile soluzione:

```
select v.id_volo, v.id_citta_partenza, v.id_citta_arrivo, sum(prezzo_biglietto)/p.stipendio_annuo as rap
from prenotazione as pr, volo as v, pilota as p
where pr.id_volo=v.id_volo and v.id_pilota=p.id_pilota
group by id_volo
having rap>100;
```

Soluzione prova del 19/02/2016

```
select o.id_opera, o.titolo, totale / num
from
(select b.id_opera, sum(prezzo) as totale
 from Biglietto as b, Rappresentazione as r
 where b.id_opera=r.id_opera
 group by id_opera
) as incasso,
(select id_opera, count(*) as num
 from recitazionein
 group by id_opera
) as rec, opera as o
where incasso.id_opera=rec.id_opera and rec.id_opera=o.id_opera and rec.num>10;
```

Soluzione prova del 13/07/2016

```
SELECT f.id_fattorino, f.nome, f.cognome, incassi.tot/cons.n AS rapporto
FROM fattorino AS F,
(
  SELECT c.id_fattorino, SUM(v.quantita*v.prezzo_unitario) as tot
  FROM vendita AS v, consegna AS c
  WHERE v.id_consegna=c.id_consegna
  GROUP BY c.id_fattorino) AS incassi,
(
  SELECT id_fattorino, COUNT(*) AS n
  FROM consegna
  GROUP BY id_fattorino
) AS cons
WHERE f.id_fattorino=incassi.id_fattorino and f.id_fattorino=cons.id_fattorino;
```

Soluzione prova del 14/09/2016

```
SELECT f.id_farmaco, f.nome, cf.nome, SUM(v.prezzo_totale)/SUM(n_confezioni) AS rapporto
FROM vendita AS v, farmaco AS f, casafarmaceutica AS cf
WHERE v.id_farmaco=f.id_farmaco AND f.id_casa=cf.id
GROUP BY f.id_farmaco
HAVING rapporto >=50;
```

Soluzione prova del 24/11/2016 – Domanda aperta 1

```
select a.id_albergo, a.nome, a.indirizzo, pul_ag.so / pren_ag as n_pren as rap
from (
  select id_albergo, sum(ore) as so
  from pulizia
  group by id_albergo
) as pul_ag,
(
  select id_albergo, count(*) as n_pren
  from prenotazione
  group by id_albergo
) as pren_ag,
albergo as a
where a.id_albergo=pul_ag.id_albergo and a.id_albergo=pren_ag.id_albergo
having rap>=2;
```