

Prova del 22/03/05 - Compito A - Query con operatori aggregati

Si consideri la tabella SQL, che descrive informazioni sugli incassi dei film, definita nel modo seguente:

```
create table film
{
  titolo          varchar(20),
  regista         varchar(20),
  genere         varchar(20) not null,
  durata         internal hour to minute,
  incasso        integer,
  primary key(titolo, regista)
}
```

elenicare i registi in base agli incassi complessivamente riportati, in ordine discendente, e a parità di incasso, alfabeticamente in base al regista in ordine ascendente

SOLUZIONI:

Ricordiamo che la clausola `group by` permette di specificare come dividere la tabella in sottoinsiemi. Ammette come argomento un insieme di attributi, e raggruppa le righe della tabella che possiedono lo stesso valore per questo insieme di attributi.

Nel nostro caso dobbiamo partizionare la tabella `film` in base al nome del regista, per poi applicare l'operatore aggregato `sum` su ogni sottoinsieme di righe (dobbiamo trovare l'incasso totale per ogni regista).

```
SELECT regista, sum(incasso) as incassoTotale
FROM film
GROUP BY regista
ORDER BY incassoTotale desc, regista asc
```

Si noti che `sum(incasso)` viene ridenominato come `incassoTotale` per poterlo usare nella clausola `ORDER BY`

Prova del 22/03/05 - Compito B - Query con operatori aggregati

Si consideri la tabella SQL definita nel modo seguente:

```
create table traffico_telefonico
{
  nr_utente          varchar(15),
  cod_fiscale       char(16) not null,
  nr_chiamato       varchar(15),
  data_ora           timestamp,
  durata            internal hour to minute,
  primary key(nr_utente, data_ora)
}
```

che descrive (in modo semplificato) il traffico telefonico di un particolare gestore (nr_utente è il numero telefonico dell'utente del gestore, mentre nr_chiamato, data_ora e durata sono le informazioni sulle chiamate effettuate).

Elencare i numeri degli utenti in base al numero totale di chiamate effettuate in modo discendente, e, a parità di questo valore, alfabeticamente per codice fiscale

SOLUZIONE:

Ricordiamo che la clausola `group by` permette di specificare come dividere la tabella in sottoinsiemi. Ammette come argomento un insieme di attributi, e raggruppa le righe della tabella che possiedono lo stesso valore per questo insieme di attributi.

Nel nostro caso dobbiamo partizionare la tabella `traffico_telefonico` in base al numero dell'utente, per poi applicare l'operatore aggregato `count` su ogni sottoinsieme di righe (trovare il numero totale di chiamate effettuate da parte di ogni utente equivale a contare quante righe ci sono per ogni diverso nr_utente).

```
SELECT nr_utente,cod_fiscale, count(*) as nrTotChiamate
FROM traffico_telefonico
GROUP BY nr_utente
ORDER BY nrTotChiamate desc, cod_fiscale asc
```

Si noti che `count(*)` viene ridenominato come `nrTotChiamate` per poterlo usare nella clausola `ORDER BY`

Prova del 22/03/05 - Compito A - Query con join

Dato il seguente schema di basi di dati (gli attributi sottolineati sono le chiavi delle relazioni):

```
calciatori(Codice, Nome, Cognome, Squadra, Ruolo)
squadre(Nome, ScudettiVinti, Serie)
```

elenicare in ordine alfabetico per cognome i centrocampisti che giocano in squadre che hanno vinto qualche scudetto

SOLUZIONE:

Si noti che:

- le squadre che hanno vinto qualche scudetto sono quelle per cui il valore assunto dall'attributo ScudettiVinti è maggiore di 0 (0, equivalentemente, diverso da 0)

- la condizione di join è calciatori.Squadra = squadre.Nome

```
SELECT Nome, Squadra
FROM calciatori, squadre
WHERE calciatori.Ruolo = 'centrocampista'
AND squadre.ScudettiVinti > 0
AND calciatori.Squadra = squadre.Nome
ORDER BY Nome
```

La specifica dei nomi delle tabelle nella clausola WHERE non era necessaria non essendoci omonimie tra attributi. E' stata inserita per maggiore chiarezza.

Prova del 22/03/05 - Compito B - Query con join

Dato il seguente schema di basi di dati (gli attributi sottolineati sono le chiavi delle relazioni):

```
pittori(Codice, Nome, Cognome, Data_di_nascita)  
quadri(CodicePittore, Titolo, Data, Museo, Citta)
```

elenicare in ordine cronologico (dal meno recente al più recente) i titoli dei quadri di Picasso esposti al museo MOMA di New York

SOLUZIONE:

Si noti che la condizione di join è

```
pittori.Codice = quadri.CodicePittore
```

```
SELECT Titolo, Data  
FROM pittori, quadri  
WHERE pittori.Nome = 'Picasso'  
AND quadri.Museo = 'Moma'  
AND quadri.Citt = 'New York'  
AND pittori.Codice = quadri.CodicePittore  
ORDER BY Data asc
```

La specifica dei nomi delle tabelle nella clausola WHERE non era necessaria non essendoci omonimie tra attributi. E' stata inserita per maggiore chiarezza.