

**Basi di Dati**  
**Corso di Laurea in Informatica Umanistica**  
*Appello del 21.04.2006 (modificato)*  
**Parte 2 – SQL (aggregati), progettazione concettuale e logica**  
*Docente: Dino Pedreschi*

Si considerino i seguenti fatti riguardanti la medicina di base:

- ogni medico è caratterizzato da un codice (che lo identifica univocamente), dalle proprie generalità (nome, cognome, ecc.) e recapiti, da un orario di ambulatorio, dai pazienti che segue;
- ogni paziente è caratterizzato dal proprio codice fiscale, dalle generalità (nome, cognome, ecc.) e recapiti, e dal proprio medico curante;
- i medici prescrivono ricette ai pazienti, ciascuna caratterizzata dalla data di prescrizione, dal costo del ticket e da una serie di prestazioni sanitarie, identificate da un codice, ciascuna delle quali può essere:
  - un farmaco (caratterizzato dal nome, dalla marca e dal tipo – compresse, fiale, etc.),
  - un esame (es., analisi delle urine, emocromo, etc.)
  - una visita specialistica (caratterizzata da una descrizione, es. visita ortopedica, e da un motivo

1) Si definisca uno schema concettuale che rappresenti le informazioni richieste.

2) Si traduca lo schema concettuale del punto 1) in uno schema logico di base di dati nel modello relazionale.

3) Si risolvano, in riferimento allo schema del punto 2), le seguenti interrogazioni utilizzando SQL:

1. Elenco che riporta, per ciascun medico di base, il numero di ricette in cui ha prescritto un farmaco antidepressivo;

```
Select R.codMedico, count(*) AS num
from ricette R JOIN Farmaco F on R.codice=F.codice
where F.tipo = 'antidepressivo'
group by R.codMedico
```

2. Elenco che riporta, per ciascun paziente di medici di base della città di Pisa, il numero di ricette in cui è stata prescritta una visita specialistica;

```
Select R.codicePaziente as codicePaziente, count(*) AS
numVisiteSpec
from Medici M JOIN Ricette R ON R.codMedico = M.codice
JOIN Visite V ON V.codice = R.codice
where M.citta = 'Pisa'
group by R.codicePaziente
```

3. Creare una vista che elenca il numero di pazienti di ciascun medico di base, e limitatamente ai medici che seguono almeno 150 pazienti. Visualizzare il contenuto della vista ordinando in senso decrescente rispetto al numero di pazienti;

```
Create view VistaNumPazienti (codMedico, numPazienti) AS
select medicocurante, count(*)
from Pazienti
```

```
group by medicoCurante
having count(*) >= 150
```

```
Select *
from vistanumPazienti
order by numPazienti
```

Variante query 3:

Creare una vista che elenca il numero di pazienti di ciascun medico di base, e limitatamente ai medici che seguono almeno 150 pazienti. La vista deve contenere anche il nome e il cognome del medico.

```
Create view VistaNumPazientiConNome (Cognome, Nome,
codMedico, numPazienti) AS
select M.cognome, M.nome, M.codice, count(*)
from Pazienti P JOIN Medici M on P.codMedico = M.codice
group by M.codice, M.cognome, M.nome
having count(*) >= 150
```

4. Elencare il codice e il nome dei medici che hanno prescritto ricette che non contengono visite specialistiche;

```
Select codice, nome
from medici
where codice IN (select codiceMedico
                 from ricette
                 where codiceRicette not in
                 (select codice from visite))
```

oppure

```
Select codice, nome
from medici
where codice NOT IN (select codiceMedico
                    from ricette JOIN visite ON codice=codiceRicetta)
```

oppure

```
Select codice, nome
from medici M JOIN ricette R on R.codiceMedico = M.codice
where R.codice not in
      (select codice from visite)
```

5. Elenco che riporta i dettagli della ricetta con il prezzo massimo;

```
Select Codice, data, costo
from ricette
where costo = (select max(costo) from ricette)
```

6. Elenco che, per ogni medico il cui cognome inizia per 'A' che ha scritto minimo dieci ricette, riporta il codice del medico e il numero di ricette prescritte;

```
Select M.codice, count(*)
from Medico M JOIN ricette R on M.codice = R.codMedico
where M.cognome LIKE 'A%'
group by M.codice
having count(*) >= 10
```

7. Elenco delle ricette il cui ticket ha un costo inferiore a tutti i ticket delle ricette corrispondenti a visite specialistiche.

```
Select codice, data, costo
from ricette
where costo < all (
    select costo
    from ricetta R join visita V on R.codice=V.codice)
```

oppure

```
Select codice, data, costo
from ricette
where costo < (
    select min(costo)
    from ricetta R join visita V on R.codice=V.codice)
```

oppure

```
Select codice, data, costo
from ricette
where costo < all (
    select costo
    from ricetta
    where codice IN (Select codice
                    from visite)
    )
```