

# Basi di dati, nuovo ordinamento

## Sistemi informativi, primo modulo

### Homework 24 ottobre 2002

#### Soluzioni

**Domanda 1** (60%) Si consideri una base di dati che contiene informazioni sugli impiegati, i progetti e le sedi di una azienda, con le partecipazioni degli impiegati ai progetti e le sedi di svolgimento dei progetti stessi; essa contiene le seguenti relazioni:

- IMPIEGATI(Matricola,Cognome,Nome,Progetto), con vincolo di integrità referenziale fra Progetto e la relazione PROGETTI
- PROGETTI(Codice,Titolo)
- SEDI(Nome,Città,Indirizzo)
- SVOLGIMENTO(Progetto,Sede), con vincoli di integrità referenziale fra Progetto e la relazione PROGETTI fra Sede e la relazione SEDI

Formulare:

1. in SQL, l'interrogazione che trova i progetti che si svolgono presso sedi situate a Roma, mostrando, per ciascuno, il codice e il titolo del progetto e il nome della sede

Possibile soluzione

```
select Codice, Titolo, nome
from Progetti, Svolgimento, Sedi
where Codice = Progetto
      and Sede = Nome
      and Citta = 'Roma'
```

2. in SQL, l'interrogazione che conta, per ciascun progetto, gli impiegati che lavorano ad esso

Possibile soluzione

Se interessano solo i codici dei progetti:

```
select progetto, count (*)
from impiegati
group by progetto
```

Se invece interessano anche i titoli:

```
select progetto, titolo, count (*)
from impiegati, progetti
where progetto = codice
group by progetto, titolo
```

Se ci sono progetti cui non lavora nessun impiegato:

```
select codice, titolo, count(matricola)
from progetti left join impiegati on codice = progetto
group by codice,
```

Se ci sono progetti cui non lavora nessun impiegato, ma il sistema non supporta il join esterno:

```
select codice, titolo, count(matricola)
from progetti join impiegati on codice = progetto
group by codice, titolo
union
select codice, titolo, 0
from progetti
where codice not in (select progetto
                    from impiegati)
```

3. in Java-JDBC (o con SQL immerso in un linguaggio o pseudolinguaggio di programmazione), una classe (o un frammento di programma) che stampa tutti i progetti (con codice e titolo) e, per ciascuno di essi, gli impiegati coinvolti (mostrando matricola e cognome); in sostanza, va prodotto un prospetto del tipo seguente:

```
CodiceProgetto  TitoloProgetto
MatricolaImpiegato  CognomeImpiegato
MatricolaImpiegato  CognomeImpiegato
...
CodiceProgetto  TitoloProgetto
MatricolaImpiegato  CognomeImpiegato
...
```

Possibile soluzione

È necessario far eseguire l'interrogazione che trova progetti e impiegati, stampando una volta i dati su un progetto e poi i rispettivi impiegati. Allo scopo, il risultato deve essere ordinato sulla base del progetto (con la `order by` e non con la `group by`), ad esempio (si noti il join esterno, che serve a produrre anche i progetti senza impiegati)

```
select Codice, Titolo, Matricola, Cognome
from Progetti left join Impiegati on Codice = Progetto
order by Codice
```

Una possibile soluzione con JDBC è nel file `Query1.3.java` allegato.

4. estendere la risposta al quesito precedente mostrando anche, per ciascun progetto, la lista delle sedi di svolgimento, costruendo quindi un prospetto come il seguente:

```
CodiceProgetto  TitoloProgetto
MatricolaImpiegato  CognomeImpiegato
MatricolaImpiegato  CognomeImpiegato
...
NomeSede        Città
NomeSede        Città
...
CodiceProgetto  TitoloProgetto
...
```

Possibile soluzione

In questo caso è necessario scandire in parallelo la risposta all'interrogazione sopra citata e la risposta all'interrogazione che trova le sedi dei progetti (sempre ordinata sul codice del progetto): per ogni progetto, si stampa prima il nome, poi la lista degli impiegati ed infine la lista delle sedi. L'interrogazione potrebbe essere la seguente:

```
select Codice, Nome, Citta
from Progetti join Svolgimento on Codice = Progetto
join Sedi on Sede = Nome
order by Codice
```

Una possibile soluzione con JDBC è nel file `Query1.4.java` allegato.

**Domanda 2** (10%) Con riferimento ad una relazione SOCI(Codice, Cognome, Nome, Categoria, Età), scrivere le interrogazioni SQL che calcolano l'età media dei soci di ciascuna categoria, nei due casi seguenti:

1. se l'età non è nota si usa per essa il valore nullo;

Possibile soluzione

```
select Categoria, AVG(Eta) AS EtaMedia
from Soci
group by Categoria
```

2. se l'età non è nota si usa per essa il valore 0.

Possibile soluzione

```
select Categoria, AVG(Eta) AS EtaMedia
from Soci
where Eta <> 0
group by Categoria
```

**Domanda 3** (15%) Mostrare come in SQL si possa formulare, senza usare il costrutto **except**, un'interrogazione che calcola la differenza di due relazioni *R* e *S* definite entrambe sugli attributi *A* e *B*.

Possibile soluzione

```
select *
from R
where not exists (select *
                  from S
                  where S.A = R.A
                  and S.B = R.B)
```

**Domanda 4** (15%) Considerare le relazioni DIPARTIMENTI(Codice, Direttore) e IMPIEGATI(Matricola, Nome, Stipendio, Direttore) e le due interrogazioni seguenti, specificare se e in quali casi esse possono produrre risultati diversi:

SELECT AVG(Stipendio)	SELECT AVG(Stipendio)
FROM Impiegati	FROM Impiegati I, Dipartimenti D
WHERE Direttore IN (SELECT Direttore	WHERE I.Direttore = D.Direttore
FROM Dipartimenti)	

Possibile soluzione

Le due interrogazioni calcolano rispettivamente:

1. la media degli stipendi degli impiegati il cui direttore dirige un dipartimento
2. la media dei valori degli stipendi per le ennuple ottenute concatenando quelle degli impiegati con quelle dei dipartimenti con uguale direttore: un impiegato il cui direttore dirige più dipartimenti compare più volte, modificando potenzialmente la media

In altre parole, le due interrogazioni calcolano la media dei valori di stipendio nelle ennuple ottenute in risposta alle due interrogazioni seguenti. Nella seconda, un impiegato il cui il cui direttore dirige più dipartimenti compare più volte e quindi il suo stipendio "pesa di più" nel calcolo della media.

SELECT *	SELECT *
FROM Impiegati	FROM Impiegati I, Dipartimenti D
WHERE Direttore IN (SELECT Direttore	WHERE I.Direttore = D.Direttore
FROM Dipartimenti)	

Quindi i risultati delle due interrogazioni possono differire quando vi sono direttori che dirigono più dipartimenti.