

# Basi di Dati - Corso di Laura in Informatica Umanistica

Appello del 28/06/2010

## Parte 1: Algebra Relazionale e linguaggio SQL

Docente: Giuseppe Amato

Sia dato il seguente schema di base di dati per la gestione dei progetti di un'impresa:

```
TABLE Reparti (  
  idReparti INT NOT NULL primary key,  
  Nome VARCHAR(15) NOT NULL,  
  ...  
)
```

```
TABLE Persone(  
  CodiceFiscale CHAR(16) NOT NULL primary key,  
  Cognome VARCHAR(15) NOT NULL,  
  Nome VARCHAR(15) NOT NULL,  
  DataDiNascita DATE,  
  Reparto INT NOT NULL REFERENCES Reparti (idReparti ),  
  ...  
)
```

```
TABLE Progetti (  
  idProgetti INT NOT NULL primary key,  
  Nome VARCHAR(15) NOT NULL,  
  ...Costo Integer Not NULL  
)
```

```
TABLE PersonaSegueProgetto (  
  CodiceFiscale CHAR(16) NOT NULL REFERENCES Persone (CodiceFiscale ),  
  idProgetti INT NOT NULL REFERENCES Progetti (idProgetti ),  
  GiorniLavorati INT,  
  PRIMARY KEY (CodiceFiscale, idProgetti) , ...  
)
```

Risolvere le seguenti interrogazioni utilizzando la sintassi SQL oppure la forma standard per le interrogazioni dell'algebra relazionale:

- 1) Trovare il costo del progetto con costo minore seguito da Mario Bianchi.
- 2) Trovare il nome delle persone che seguono progetti il cui costo è superiore a 20000 Euro.
- 3) Trovare il nome dei progetti seguiti dal reparto 'Ingegneria'.
- 4) Trovare i progetti seguiti dal reparto di appartenenza di Marco Bianchi.
- 5) Trovare i reparti che non seguono il progetto 'autostrade'.
- 6) Trovare i progetti seguiti sia da Bianchi che da Neri

### Soluzioni:

1)

```
select min(costo)
from   persone join PersonaSegueProgettoon persone.CodiceFiscale=persone_has_progetti.CodiceFiscale
       join progetti on persone_has_progetti.idProgetti=progetti.idProgetti
where  Cognome='Bianchi' and persone.Nome='Marco'
```

2)

```
select distinct Persone.nome, Persone.cognome
from persone natural join PersonaSegueProgettojoin progetti on
persone_has_progetti.idProgetti=progetti.idProgetti
where Costo>20000
```

3)

```
select distinct reparti.nome
from reparti join persone on idReparti=reparto
             natural join persone_has_progetti
             join progetti on persone_has_progetti.idProgetti=progetti.idProgetti
where reparto.nome='Ingegneria'
```

4)

```
select distinct progetti.nome
from reparti join persone as Bianchi on reparti.idReparti=Bianchi.reparto
             join persone on reparti.idReparti=persone.reparto
             join PersonaSegueProgettoon persone.CodiceFiscale=persone_has_progetti.CodiceFiscale
             join progetti on progetti.idProgetti=persone_has_progetti.idProgetti
where Bianchi.Cognome='Bianchi' and Bianchi.Nome='Marco'
```

5)

```
select reparti.idreparti, reparti.nome
from reparti
except
```

```
(select reparti.idreparti, reparti.nome
from reparti join persone on idReparti=reparto
  join PersonaSegueProgettoon persone_has_progetti.CodiceFiscale=persone.CodiceFiscale
  join progetti on progetti.idProgetti=persone_has_progetti.idProgetti
  where progetti.nome="autostrade")
```

6)

```
select progetti.nome
from persone join PersonaSegueProgettoon persone_has_progetti.CodiceFiscale=persone.CodiceFiscale
  join progetti on progetti.idProgetti=persone_has_progetti.idProgetti
  where persone.cognome='Neri'
```

intersect

```
(select progetti.nome
from persone join PersonaSegueProgettoon persone_has_progetti.CodiceFiscale=persone.CodiceFiscale
  join progetti on progetti.idProgetti=persone_has_progetti.idProgetti
  where persone.cognome='Bianchi')
```

# Basi di Dati - Corso di Laura in Informatica Umanistica

Appello del 28/06/2010

## Parte 2: SQL (Aggregati) e Progettazione concettuale e logica

Docente: Giuseppe Amato

Si consideri la seguente realtà riguardante la gestione dei negozi di una catena di supermercati:

Ogni *negozio* è caratterizzata da un nome da una città ed un indirizzo.

Ogni negozio offre vari prodotti. I *prodotti* hanno un nome una marca ed un modello (ad esempio, Televisore LCD, Samsung, LE42F).

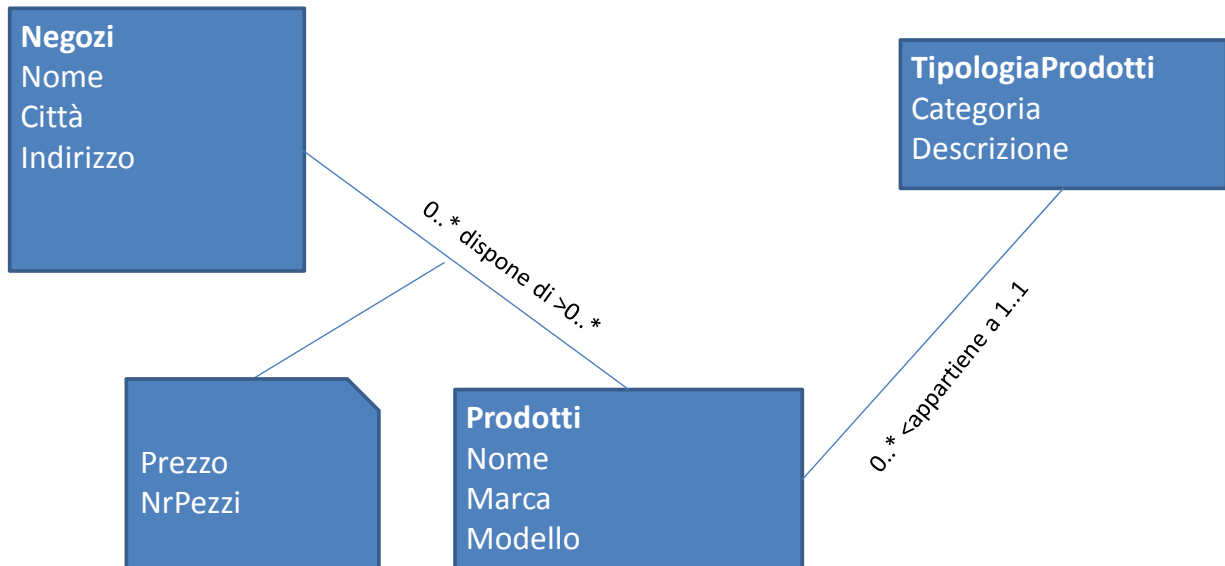
Per ogni *prodotto offerto da un negozio* si vuole sapere il numero di pezzi disponibili del prodotto ed il prezzo di vendita.

I prodotti, inoltre, sono raggruppati in *tipologie di prodotto*. Ogni tipologia di prodotto, è caratterizzata da un Categoria (ad esempio Elettronica, Giardinaggio, Casalinghi, etc) e da una descrizione.

1. Si definisca uno schema concettuale che rappresenta le informazioni sopra descritte.
2. Si traduca lo schema concettuale del punto 1) in uno schema logico di base di dati nel modello relazionale
3. Si risolvano le seguenti query facendo riferimento allo schema logico del punto 2)
  - a. Trovare il numero di prodotti esistenti per le varie categorie disponibili.
  - b. Trovare per ogni negozio che possiede prodotti di categoria elettronica il numero di prodotti diversi disponibili di tale categoria.
  - c. Trovare i negozi che con i prodotti disponibili coprono più di 20 categorie di prodotti (suggerimento: usare le viste).

## Soluzioni:

1)



2)

```
CREATE TABLE Negozi (  
  idNegozi INT NOT NULL ,  
  Nome VARCHAR(20) NOT NULL ,  
  Via VARCHAR(20) NOT NULL ,  
  Citta VARCHAR(10) NOT NULL ,  
  PRIMARY KEY (idNegozi) )
```

```
CREATE TABLE TipologieProdotti (  
  idTipologieProdotti INT NOT NULL ,  
  Descrizione VARCHAR(45) NOT NULL ,  
  Categoria VARCHAR(15) NOT NULL ,  
  PRIMARY KEY (idTipologieProdotti) )
```

```
CREATE TABLE Prodotti (  
  idProdotti INT NOT NULL ,  
  Nome VARCHAR(45) NOT NULL ,  
  Marca VARCHAR(45) NOT NULL ,  
  Modello VARCHAR(45) NOT NULL ,  
  Tipologia INT NOT NULL ,  
  PRIMARY KEY (idProdotti) ,  
  FOREIGN KEY (Tipologia )  
  REFERENCES TipologieProdotti (idTipologieProdotti ))
```

```
CREATE TABLE ProdottiNelNegozio (  
  Negozio INT NOT NULL ,  
  Prodotto INT NOT NULL ,  
  Prezzo DECIMAL(9,2) NOT NULL ,  
  NumeroPezzi INT NOT NULL ,  
  PRIMARY KEY (Prodotto, Negozio) ,  
  FOREIGN KEY (Negozio )  
  REFERENCES Negozi (idNegozi )  
  FOREIGN KEY (Prodotto )  
  REFERENCES appello-28-06-2010.Prodotti (idProdotti )  
)
```

3)

a)

```
select count(*), categoria  
  
from tipologieprodotti join prodotti on tipologia=idTipologieprodotti  
  
group by categoria
```

b)

```
select idNegozi, negozi.nome, count(*)  
  
from   negozi join prodottinelnegozio on idNegozi=negozi  
       join prodotti on prodotto = idProdotti  
       join tipologieprodotti on tipologia=idTipologieprodotti  
  
where categoria='Elettronica'  
  
group by idNegozi, negozi.nome
```

c)

```
create view negozicategorie as
```

```
select distinct idNegozzi, negozi.nome, idTipologieProdotti  
from negozi join prodottinelnegozio on idNegozzi=negozio  
      join prodotti on prodotto = idProdotti  
      join tipologieprodotti on tipologia=idTipologieprodotti
```

```
select idNegozzi, nome  
from negozi  
group by idNegozzi, nome  
having count(*)>=20
```