

# Basi di Dati - Corso di Laura in Informatica Umanistica

Appello del 10/09/2010

## Parte 1: Algebra Relazionale e linguaggio SQL

Docente: Giuseppe Amato

Sia dato il seguente schema di base di dati per la gestione di una biblioteca:

```
CREATE TABLE Alberghi (  
  idAlberghi INT NOT NULL PRIMARY KEY,  
  Nome VARCHAR(15) NOT NULL ,  
  Città VARCHAR(15) NULL )
```

```
CREATE TABLE Servizi (  
  idServizi INT NOT NULL PRIMARY KEY,  
  Nome VARCHAR(10) NOT NULL ,  
  Descrizione VARCHAR(45) NULL )
```

```
CREATE TABLE Stanze (  
  Piano VARCHAR(45) NOT NULL PRIMARY KEY,  
  Numero VARCHAR(45) NOT NULL PRIMARY KEY,  
  Albergo INT NOT NULL PRIMARY KEY REFERENCES Alberghi (idAlberghi ),  
  PostiLetto INT(11) NOT NULL ,  
  Fumatori Boolean NOT NULL))
```

```
CREATE TABLE AlbergoOffreServizi (  
  Servizio INT NOT NULL PRIMARY KEY REFERENCES Servizi (idServizi ),  
  Albergo INT NOT NULL PRIMARY KEY REFERENCES Alberghi (idAlberghi ),  
  Prezzo DECIMAL(9,2) NULL)
```

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

- 1) Trovare il numero di stanze doppie offerte dall'hotel Continental di Roma.
- 2) Trovare il numero di alberghi che offrono il servizio Sauna a Firenze
- 3) Trovare gli alberghi che hanno stanze con 4 posti letto e che offrono il servizio Sauna a meno di 100 Euro.
- 4) Trovare gli alberghi che non hanno il servizio idromassaggio.
- 5) Trovare gli alberghi che hanno solo stanze non fumatori
- 6) Trovare gli alberghi che hanno sia la sauna che la palestra.

### Soluzioni:

1)

```
select count(*)  
from alberghi join stanze on albergo=idAlberghi  
where nome="Continental" and PostiLetto=2 and città="Roma"
```

2)

```
select count(*)  
from alberghi join AlbergoOffreServizi on albergo=idAlberghi join Servizi on servizio=idServizi  
where Città="Firenze" and Servizi.nome="Sauna"
```

3)

```
select distinct Alberghi.Nome, Alberghi.Città  
from alberghi join AlbergoOffreServizi on AlbergoOffreServizi.albergo=idAlberghi join Servizi on  
servizio=idServizi join Stanze on Stanze.Albergo=idAlberghi  
where Servizi.nome="Sauna" and PostiLetto =4 and Prezzo < 100
```

4)

```
select Nome, Città, idAlberghi  
from Alberghi  
except  
select Alberghi.Nome, Alberghi.Nome, Città, idAlberghi  
from alberghi join AlbergoOffreServizi on AlbergoOffreServizi.albergo=idAlberghi join Servizi on  
servizio=idServizi  
where Servizi.nome = "Idromassaggio"
```

5)

```
Select Alberghi.Nome, Alberghi.Nome, Città, idAlberghi  
from Alberghi  
except  
select Alberghi.Nome, Alberghi.Nome, Città, idAlberghi  
from alberghi join stanze on albergo=idAlberghi  
where Fumatori=1
```

6)

```
select Alberghi.Nome, Alberghi.Nome, Città, idAlberghi
```

```
from alberghi join AlbergoOffreServizi on AlbergoOffreServizi.albergo=idAlberghi join Servizi on  
servizo=idServizi
```

```
where Servizi.nome = "Sauna"
```

```
intersect
```

```
select Alberghi.Nome, Alberghi.Nome, Città, idAlberghi
```

```
from alberghi join AlbergoOffreServizi on AlbergoOffreServizi.albergo=idAlberghi join Servizi on  
servizo=idServizi
```

```
where Servizi.nome = "Palestra"
```

# Basi di Dati - Corso di Laura in Informatica Umanistica

Appello del 10/09/2010

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

Docente: Giuseppe Amato

Si consideri la seguente realtà riguardante una società di gestione dei parcheggi:

La società gestisce vari parcheggi. Ogni *parcheggio* è caratterizzato da un nome, da una tipologia (es.: Multipiano, Sotterraneo, all'aperto, Custoditi, etc.), e da un indirizzo.

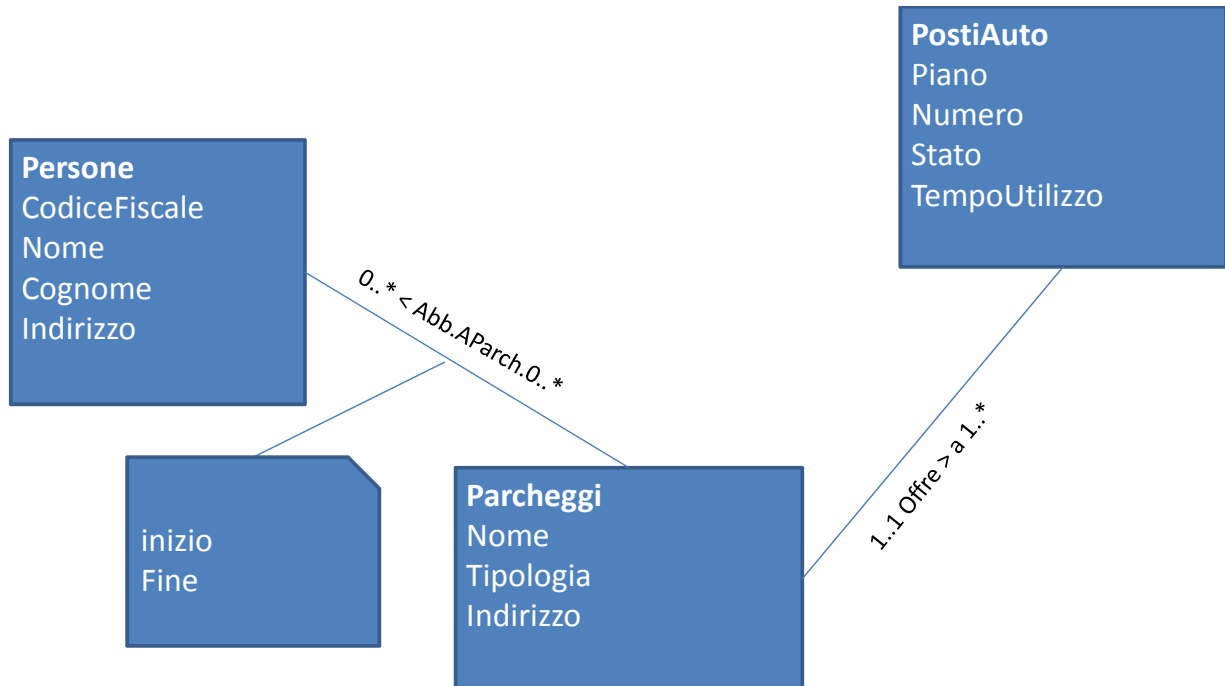
Ogni parcheggio contiene vari posti auto. Ogni *posto auto* ha una posizione (ad esempio il piano ed il numero), può essere nello stato occupato, libero, o indisponibile. A seconda che sia occupato da una automobile, sia libero e disponibile, o indisponibile (ad esempio per motivi tecnici). Per ogni posto auto si vuole anche memorizzare il tempo di utilizzo, ossia il tempo globale in cui il posto auto si è trovato in stato occupato.

I *cittadini* possono comprare l'abbonamento ad un parcheggio. Ogni cittadino può comprare abbonamenti diversi per parcheggi diversi. Di ogni cittadino si vuole memorizzare il Nome il Cognome, il codice fiscale, l'indirizzo. L'abbonamento ha un inizio ed una fine.

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 i parcheggi a cui è abbonato Mario Rossi, ed il corrispondente numero totale di posti auto.
  - b. Trovare per ogni parcheggio, il posto auto meno utilizzato.
  - c. Trovare i parcheggi con più di 40 posti disponibili.

## Soluzioni:

1)



2)

```
CREATE TABLE Parcheggio (
  idParcheggio INT NOT NULL ,
  Nome VARCHAR(10) NOT NULL ,
  Tipologia VARCHAR(15) NULL ,
  Indirizzo VARCHAR(45) NULL ,
  PRIMARY KEY (idParcheggio) )
```

```
CREATE TABLE PostiAuto (
  Piano INT NOT NULL ,
  Numero INT NOT NULL ,
  Stato VARCHAR(10) NOT NULL ,
  TempoUtilizzo INT NOT NULL ,
  idParcheggio INT NOT NULL ,
  PRIMARY KEY (Piano, Numero, idParcheggio) ,
```

```
FOREIGN KEY (idParcheggio )REFERENCES Parcheggi (idParcheggio ))
```

```
CREATE TABLE persone (  
CodiceFiscale CHAR(16) NOT NULL ,  
Nome VARCHAR(15) NOT NULL ,  
Cognome VARCHAR(15) NOT NULL ,  
Indirizzo VARCHAR(45) NULL ,  
PRIMARY KEY (CodiceFiscale) )
```

```
CREATE TABLE PersonaAbbonataAParcheggio (  
CodiceFiscale CHAR(16) NOT NULL ,  
idParcheggio INT NOT NULL ,  
Inizio DATE NOT NULL ,  
Fine DATE NOT NULL ,  
PRIMARY KEY (CodiceFiscale, idParcheggio) ,  
FOREIGN KEY (CodiceFiscale ) REFERENCES persone (CodiceFiscale )  
FOREIGN KEY (idParcheggio ) REFERENCES Parcheggi (idParcheggio ))
```

3)

a)

```
select parcheggi.nome, count(*)  
  
from persone natural join personaabbonataaparcheggio join parcheggi on  
personaabbonataaparcheggio.idParcheggio=parcheggi.idParcheggio join postiauto on  
postiauto.idParcheggio=parcheggi.idParcheggio  
  
where persone.nome="Mario" and cognome="Rossi"  
  
group by parcheggi.idParcheggio, parcheggi.nome
```

b)

```
select nome, piano, numero, TempoUtilizzo  
  
from parcheggi natural join postiauto  
  
where (nome,TempoUtilizzo)in  
  
(select nome, min(TempoUtilizzo)  
  
from parcheggi natural join postiauto  
  
group by parcheggi.idParcheggio, nome)
```

c)

```
select nome
```

```
from parcheggi natural join postiauto
where stato="libero"
group by parcheggi.idParcheggio, nome
having count(*)>40
```