

# Basi di Dati - Corso di Laura in Informatica Umanistica

Verifica del 16/04/2010

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

Docente: Giuseppe Amato

### Compito A

Si consideri la seguente realtà riguardante l'organizzazione interna di un'azienda:

L'azienda è suddivisa in *reparti*. Ogni reparto ha un codice identificativo, un nome, un responsabile, ed una descrizione delle mansioni svolte nel reparto

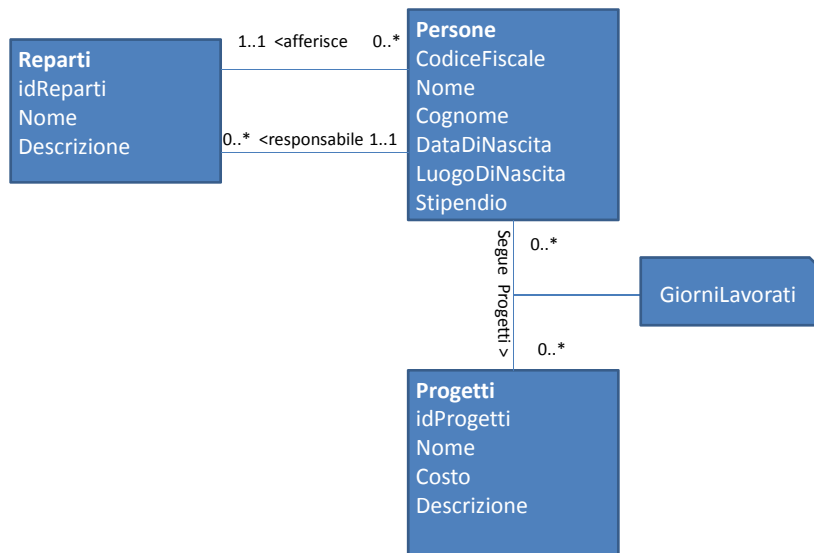
Nell'azienda lavorano un certo numero di *persone*, ed ogni persona afferisce ad un reparto. Ogni persona è identificata dal suo codice fiscale, ed ha un nome, un cognome un anno di nascita, ed uno stipendio.

L'azienda gestisce un certo numero di *progetti*. Ogni progetto è identificato da un codice univoco, ed ha un nome, un costo, ed una descrizione. Ad un progetto lavorano un certo numero di persone, e una stessa persona può lavorare a più progetti. Per ogni progetto a cui lavora una persona è indicato il numero di giorni che ha lavorato al progetto.

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 persone che lavorano in ogni reparto
  - b. Trovare per ogni persona il numero di progetti dove hanno lavorato più di 20 giorni
  - c. Trovare per ogni progetto, in cui lavorano più di cinque persone, il numero totale di giorni lavorati da tutte le persone che vi lavorano

Soluzioni:

1)



2)

```
CREATE TABLE Reparti (  
  idReparti INT NOT NULL ,  
  Nome VARCHAR(15) NOT NULL,  
  Descrizione VARCHAR(45),  
  Responsabile CHAR(16) NOT NULL,  
  PRIMARY KEY (idReparti) ,  
  FOREIGN KEY (Responsabile )  
    REFERENCES Persone (CodiceFiscale )  
)
```

```
CREATE TABLE Persone(  
  CodiceFiscale CHAR(16) NOT NULL ,  
  Cognome VARCHAR(15) NOT NULL,  
  Nome VARCHAR(15) NOT NULL,  
  DataDiNascita DATE,  
  LuogoDiNascita DATE,  
  Stipendio INT,  
  Reparto INT NOT NULL ,  
  PRIMARY KEY (CodiceFiscale) ,  
  FOREIGN KEY (Reparto )  
    REFERENCES Reparti (idReparti )  
)
```

```
CREATE TABLE Progetti (  
  idProgetti INT NOT NULL ,  
  Nome VARCHAR(15) NOT NULL,  
  Costo INT,  
  Descrizione VARCHAR(45),  
  PRIMARY KEY (idProgetti) )
```

```
CREATE TABLE Persone_has_Progetti (  
  CodiceFiscale CHAR(16) NOT NULL ,  
  idProgetti INT NOT NULL ,  
  GiorniLavorati INT,  
  PRIMARY KEY (CodiceFiscale, idProgetti) ,  
  FOREIGN KEY (CodiceFiscale )  
    REFERENCES Persone (CodiceFiscale )  
  FOREIGN KEY (idProgetti )  
    REFERENCES Progetti (idProgetti )  
 )
```

3.a)  
select Reparti.Nome, count(\*)  
from Reparti join persone on reparto=idReparti  
group by idReparti, Reparti.Nome

3.b)  
select Cognome, Nome, count(\*)  
from persone natural join persone\_has\_progetti  
where GiorniLavorati > 20  
group by CodiceFiscale, Nome, Cognome

3.c)  
select Nome, sum(GiorniLavorati)  
from progetti natural join persone\_has\_progetti  
group by idProgetti, Nome  
having count(\*)>5

# Basi di Dati - Corso di Laura in Informatica Umanistica

Verifica del 16/04/2010

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

Docente: Giuseppe Amato

### Compito B

Si consideri la seguente realtà riguardante l'organizzazione interna di un'impresa:

L'impresa è suddivisa in *sezioni*. Ogni sezione ha un codice identificativo, un nome, un responsabile, ed una descrizione delle mansioni svolte nella sezione

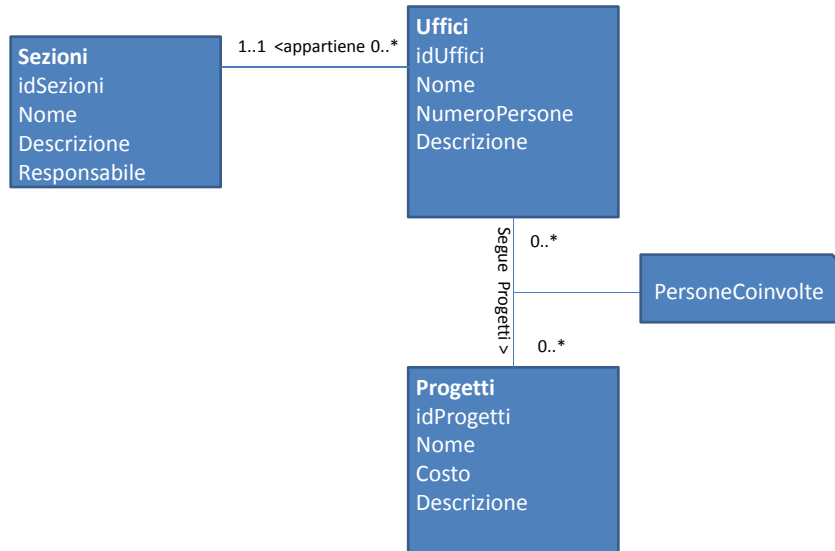
Ogni sezione è suddivisa in vari *uffici*. Ogni ufficio è identificato da un identificatore unico, ed ha un nome, un numero di persone che vi lavorano, ed una descrizione.

L'impresa gestisce un certo numero di *progetti*. Ogni progetto è identificato da un codice univoco, ed ha un nome, un costo, ed una descrizione. Un progetto è seguito da più uffici, e uno stesso ufficio può seguire più progetti. Per ogni progetto seguito da un ufficio è indicato il numero di persone che lavora al progetto.

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 uffici che afferiscono ad ogni sezione
  - b. Trovare per ogni ufficio il numero di progetti seguiti dove vi lavorano più di 20 persone di quell'ufficio.
  - c. Trovare per ogni progetto, di cui si occupano più di 7 uffici, il numero totale di persone che lo seguono

Soluzione:

1)



2)

```
CREATE TABLE Sezioni (  
  idSezioni INT(11) NOT NULL ,  
  Nome VARCHAR(15) NOT NULL ,  
  Responsabile VARCHAR(30),  
  Descrizione VARCHAR(45),  
  PRIMARY KEY (idSezioni) )
```

```
CREATE TABLE Uffici (  
  idUffici INT(11) NOT NULL ,  
  Nome VARCHAR(15) NOT NULL ,  
  NumeroPersone INT(11) NOT NULL ,  
  Descrizione VARCHAR(45),  
  Sezione INT(11) NOT NULL ,  
  PRIMARY KEY (idUffici) ,  
  FOREIGN KEY (Sezione )  
  REFERENCES Sezioni (idSezioni) )
```

```
CREATE TABLE Progetti (  
  idProgetti INT NOT NULL ,  
  Nome VARCHAR(15) NOT NULL ,  
  Costo INT ,  
  Descrizione VARCHAR(45),  
  PRIMARY KEY (idProgetti) )
```

```
CREATE TABLE Uffici_has_Progetti (  
  idUffici INT(11) NOT NULL ,  
  idProgetti INT NOT NULL ,  
  PRIMARY KEY (idUffici, idProgetti) )
```

idUffici INT(11) NOT NULL ,  
idProgetti INT(11) NOT NULL ,  
PersoneCoinvolte INT,  
PRIMARY KEY (idUffici, idProgetti) ,  
FOREIGN KEY (idUffici )  
REFERENCES Uffici (idUffici )  
FOREIGN KEY (idProgetti )  
REFERENCES Progetti (idProgetti )

3.a)

```
select Sezioni.Nome, count(*)  
from Sezioni join Uffici on idSezioni=Sezione  
group by idSezioni, Sezioni.Nome
```

3.b)

```
select Nome, count(*)  
from uffici natural join uffici_has_progetti  
where PersoneCoinvolte>20  
group by idUffici, Nome
```

3.c)

```
select Nome, sum(PersoneCoinvolte)  
from progetti natural join uffici_has_progetti  
group by idProgetti, Nome  
having count(*) > 7
```