

Laurea in Ingegneria Informatica
Nuovo Ordinamento
A.A. 2004/2005

Esame di Basi di Dati
G. De Giacomo – M. Lenzerini

Appello del 16/12/2004
Soluzione del compito B

Specifica

Problema 1

Si richiede di progettare lo schema concettuale Entità-Relazione di un'applicazione relativa ai pellegrinaggi ai Santuari. Ogni pellegrinaggio prevede il pernottamento in camera singola per una ed una sola data. Di ogni iscrizione ad un pellegrinaggio interessa il codice (identificativo), la persona che l'ha effettuata, la data in cui è stata effettuata, e la somma pagata. Ogni iscrizione è o individuale o di gruppo. Di ogni iscrizione individuale interessa il comune in cui è stata effettuata, la data di pernottamento, e la camera d'albergo assegnata per il pernottamento. Ovviamente, le iscrizioni individuali assegnano ogni camera d'albergo al massimo una volta in ogni data. Di ogni iscrizione di gruppo interessa lo sconto ottenuto sul prezzo, il numero di persone iscritte, la data richiesta per il pernottamento, e l'albergo scelto per il pernottamento stesso. Di ogni persona interessa il codice fiscale (identificativo), la data di nascita, il sesso, ed il comune di residenza, con la data dalla quale la persona vi risiede. Di ogni albergo interessa il nome (identificativo), l'anno di fondazione, il comune in cui è situato, ed il direttore (uno ed uno solo). Di ogni comune interessa il nome (unico nella regione), la regione, e la persona che funge da sindaco. Ogni camera d'albergo è identificata da un numero unico nell'ambito dell'albergo, ed è caratterizzata da un numero intero che indica la categoria (base, lusso, ecc.).

Specifica (cont.)

Problema 2

Si richiede di effettuare la progettazione logica dell'applicazione, producendo (in qualunque forma) lo schema relazionale completo di vincoli, e seguendo queste indicazioni:

- alle iscrizioni individuali si accede spesso separatamente rispetto alle iscrizioni di gruppo,
- quando si accede ad una iscrizione individuale, si vogliono spesso conoscere le seguenti proprietà: il codice, il comune in cui è stata effettuata, la somma pagata, la camera d'albergo assegnata, e la data di pernottamento.

Si noti che le interrogazioni menzionate nel problema 3 non sono da considerarsi significative per le indicazioni di progetto.

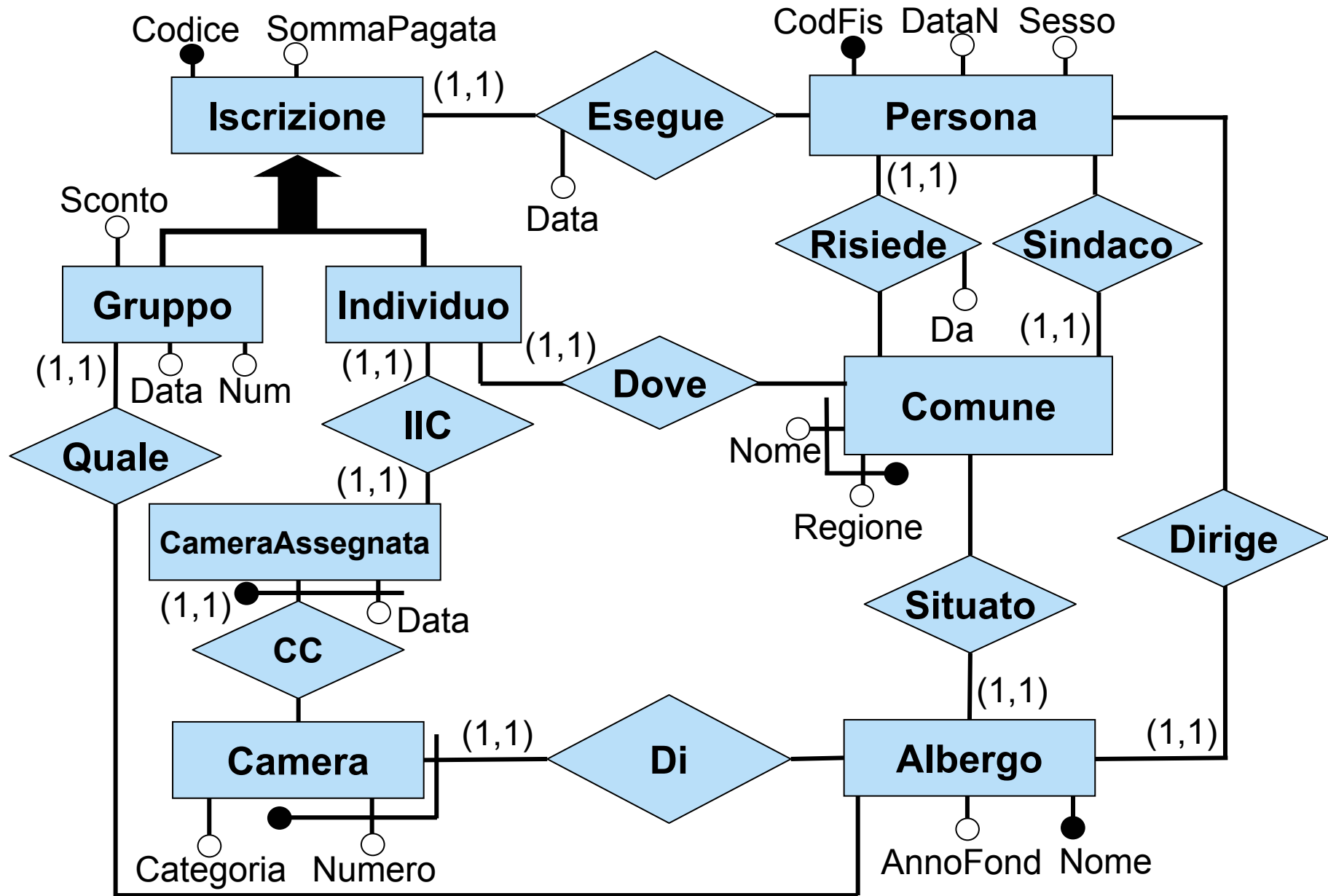
Specifica (cont.)

Problema 3

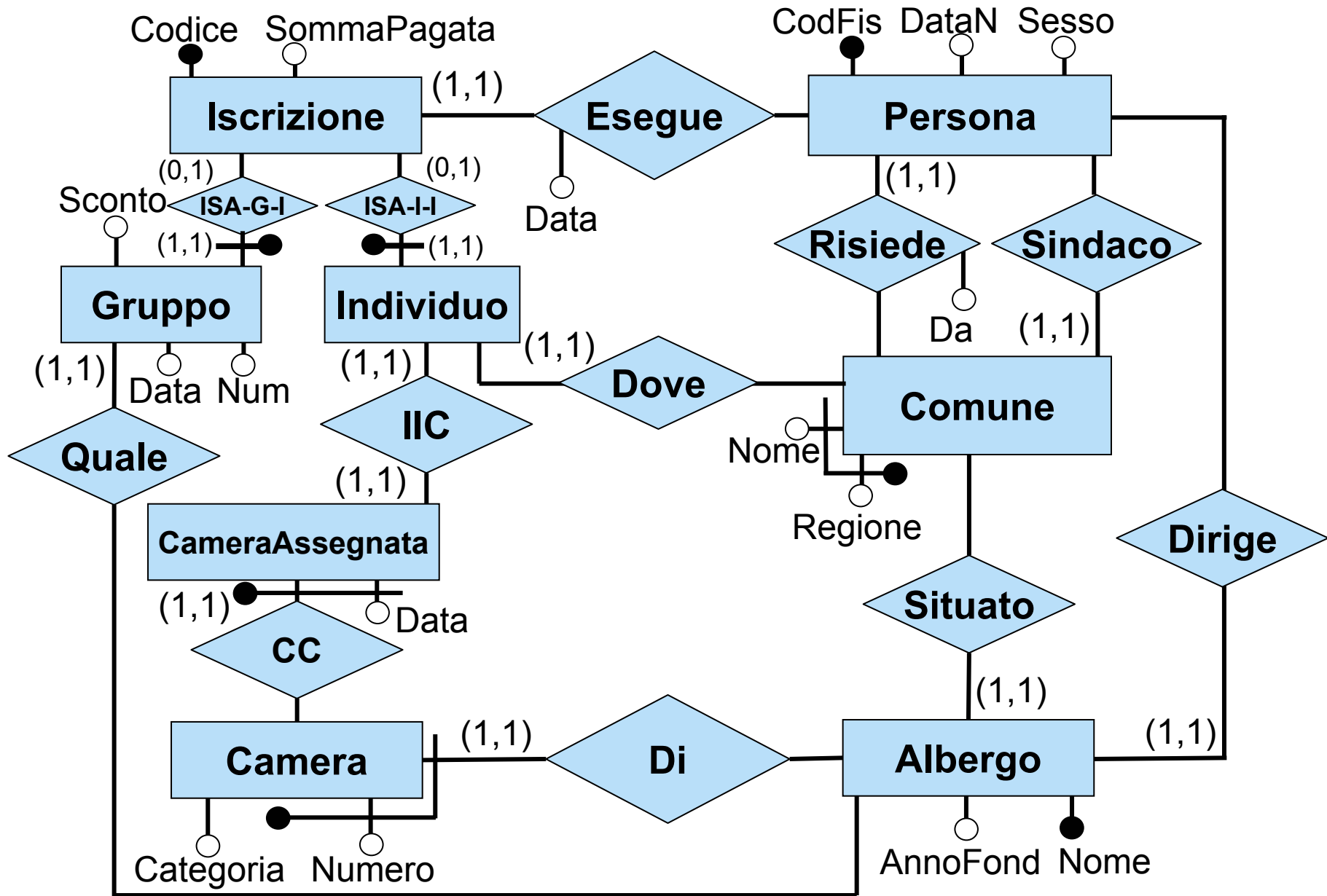
Sulla base dello schema relazionale definito per il problema 2, esprimere in SQL le seguenti interrogazioni:

1. Calcolare il codice fiscale ed il sesso delle persone che hanno effettuato almeno una iscrizione nel 2003.
2. Calcolare il codice fiscale e la data di nascita delle persone che hanno effettuato almeno una iscrizione di gruppo per un pernottamento in gennaio, e per la quale hanno pagato una somma di almeno 500 Euro.
3. Calcolare il comune di residenza e la data di nascita delle donne che hanno effettuato almeno una iscrizione individuale per la quale è stata assegnata una camera d'albergo situato in un comune diverso da quello in cui risiedono.
4. Per ogni persona che ha effettuato almeno una iscrizione individuale per la quale è stata assegnata una camera di categoria pari a 3, contare il numero di iscrizioni (individuali o di gruppo) effettuate.

Problema 1 – Schema concettuale



Problema 1 – Schema concettuale ristrutturato



Problema 2 – Vincoli sullo schema concettuale ristrutturato

- **Ogni istanza di Iscrizione partecipa ad ISA-G-I oppure ad ISA-I-I ma non ad entrambe** (dovuto all'eliminazione della generalizzazione).

Problema 2 – Schema logico

Iscrizione(Codice, SommaPagata)

foreign key: Iscrizione[Codice] \subseteq Esegue[Prenotazione]

Esegue(Iscrizione, Persona, Data)

foreign key: Esegue[Iscrizione] \subseteq Iscrizione[Codice]

foreign key: Esegue[Persona] \subseteq Persona[CodFis]

Gruppo(Codice, Sconto, Data, Num)

foreign key: Gruppo[Codice] \subseteq Iscrizione[Codice]

inclusione: Gruppo[Codice] \subseteq Quale[Gruppo]

Individuo(Codice)

foreign key: Individuo[Codice] \subseteq Iscrizione[Codice]

foreign key: Individuo[Codice] \subseteq IIC[Individuo]

foreign key: Individuo[Codice] \subseteq Dove[Individuo]

Dove(Individuo, NomeComune, RegioneComune)

foreign key: Dove[Individuo] \subseteq Individuo[Codice]

foreign key: Dove[NomeComune, RegioneComune] \subseteq Comune[Nome, Regione]

Persona(CodFis, DataN, Sesso)

foreign key: Persona[CodFis] \subseteq Risiede[Persona]

Risiede(Persona, NomeComune, RegioneComune, Da)

foreign key: Risiede[Persona] \subseteq Persona[CodFis]

foreign key: Risiede[NomeComune, RegioneComune] \subseteq Comune[Nome, Regione]

Sindaco(NomeComune, RegioneComune, Persona)

foreign key: Sindaco[Persona] \subseteq Persona[CodFis]

foreign key: Sindaco[NomeComune, RegioneComune] \subseteq Comune[Nome, Regione]

Problema 2 – Schema logico (cont.)

Comune(Nome, Regione)

foreign key: Comune[Nome,Regione] \subseteq Sindaco[NomeComune,RegioneComune]

Situato(Albergo, NomeComune, RegioneComune)

foreign key: Situato[Albergo] \subseteq Albergo[Nome]

foreign key: Situato[NomeComune,RegioneComune] \subseteq Comune[Nome,Regione]

Albergo(Nome, AnnoFond)

foreign key: Albergo[Nome] \subseteq Situato[Albergo]

foreign key: Albergo[Nome] \subseteq Dirige[Albergo]

Dirige(Albergo, Persona)

foreign key: Dirige[Persona] \subseteq Persona[Codice]

foreign key: Dirige[Albergo] \subseteq Albergo[Nome]

Camera(Numero, Albergo, Categoria)

foreign key: Camera[Albergo] \subseteq Albergo[Nome]

CameraAssegnata(Numero, Albergo, Data)

foreign key: CameraAssegnata[Numero,Albergo] \subseteq Camera[Numero,Albergo]

foreign key: CameraAssegnata[Numero,Albergo,Data] \subseteq IIC[Numero,Albergo,Data]

Quale(Gruppo, Albergo)

foreign key: Quale[Albergo] \subseteq Albergo[Nome]

foreign key: Quale[Gruppo] \subseteq Gruppo[Codice]

IIC(Numero, Albergo, Data, Individuo)

foreign key: IIC[Numero,Albergo,Data] \subseteq CameraAssegnata[Numero,Albergo,Data]

foreign key: IIC[Individuo] \subseteq Individuo[Codice]

chiave: Numero, Albergo, Data

Vincoli di generalizzazione:

$\text{Gruppo}[\text{Codice}] \cap \text{Individuo}[\text{Codice}] = \emptyset$

$\text{Iscrizione}[\text{Codice}] \subseteq \text{Gruppo}[\text{Codice}] \cup \text{Individuo}[\text{Codice}]$

Problema 2 – Ristrutturazione dello schema logico

- Poiché alle iscrizioni individuali si accede spesso separatamente rispetto alle iscrizioni di gruppo, effettuiamo la decomposizione orizzontale di **Iscrizione** in:
IscrizioneGruppo(Codice, SommaPagata)
IscrizioneIndividuo(Codice, SommaPagata)
- Poiché quando si accede ad una iscrizione individuale si vogliono spesso conoscere il codice, il comune in cui è stata effettuata, la somma pagata, la camera d'albergo assegnata, e la data in cui tale camera verrà occupata, effettuiamo l'accorpamento di **IscrizioneIndividuo**, **Individuo**, e **IIC**, ottenendo la relazione:
Individuo(Codice, SommaPagata, Numero, Albergo, Data, NomeComune, RegioneComune)
- Per eliminare la relazione inutile **CameraAssegnata**, effettuiamo l'accorpamento di **CameraAssegnata** con **Individuo**.

Ovviamente, in tutti i casi dobbiamo aggiustare di conseguenza i vincoli di foreign key.

Problema 2 – Schema logico ristrutturato

IscrizioneGruppo(Codice, SommaPagata)

foreign key: IscrizioneGruppo[Codice] \subseteq Effettua[Iscrizione]

foreign key: IscrizioneGruppo[Codice] \subseteq Gruppo[Codice]

Esegue(Iscrizione, Persona, Data)

foreign key: Esegue[Iscrizione] \subseteq IscrizioneGruppo[Codice]

foreign key: Esegue[Persona] \subseteq Persona[CodFis]

Gruppo(Codice, Sconto, Data, Num)

foreign key: Gruppo[Codice] \subseteq IscrizioneGruppo[Codice]

inclusione: Gruppo[Codice] \subseteq Quale[Gruppo]

Individuo(Codice, SommaPagata, Numero, Albergo, Data, NomeComune, RegioneComune)

foreign key: Individuo[Numero,Albergo] \subseteq Camera [Numero,Albergo]

foreign key: Individuo[NomeComune,RegioneComune] \subseteq Comune[Nome,Regione]

chiave: Numero, Albergo, Data

Persona(CodFis, DataN, Sesso)

foreign key: Persona[CodFis] \subseteq Risiede[Persona]

Risiede(Persona, NomeComune, RegioneComune, Da)

foreign key: Risiede[Persona] \subseteq Persona[CodFis]

foreign key: Risiede[NomeComune,RegioneComune] \subseteq Comune[Nome,Regione]

Sindaco(NomeComune, RegioneComune, Persona)

foreign key: Sindaco[Persona] \subseteq Persona[CodFis]

foreign key: Sindaco[NomeComune,RegioneComune] \subseteq Comune[Nome,Regione]

Problema 2 – Schema logico ristrutturato (cont.)

Comune(Nome, Regione, SalarioMedio)

foreign key: Comune[Nome,Regione] \subseteq Sindaco[NomeComune,RegioneComune]

Situato(Albergo, NomeComune, RegioneComune)

foreign key: Situato[Albergo] \subseteq Albergo[Nome]

foreign key: Situato[NomeComune,RegioneComune] \subseteq Comune[Nome,Regione]

Albergo(Nome, AnnoFond)

foreign key: Albergo[Nome] \subseteq Situato[Albergo]

Dirige(Albergo, Persona)

foreign key: Dirige[Albergo] \subseteq Albergo[Nome]

foreign key: Dirige[Persona] \subseteq Persona[Codice]

Camera(Numero, Albergo, Categoria)

foreign key: Camera[Albergo] \subseteq Albergo[Nome]

Quale(Gruppo, Albergo)

foreign key: Quale[Albergo] \subseteq Albergo[Nome]

foreign key: Quale[Gruppo] \subseteq Gruppo[Codice]

Vincoli di generalizzazione:

Gruppo[Codice] \cap Individuo[Codice] = \emptyset

Esegue[Iscrizione] \subseteq Gruppo[Codice] \cup Individuo[Codice]

Problema 3 – Interrogazioni SQL

1. Calcolare il codice fiscale ed il sesso delle persone che hanno effettuato almeno una iscrizione nel 2003.

```
select P.CodFis, P.Sesso
from Persona P, Esegue E
where P.CodFis = E.Persona and
      E.Data.Anno = 2003
```

2. Calcolare il codice fiscale e la data di nascita delle persone che hanno effettuato almeno una iscrizione di gruppo per un pernottamento in gennaio, e per la quale hanno pagato una somma di almeno 500 Euro.

```
select P.CodFis, P.DataN
from Persona P, Esegue E, IscrizioneGruppo R, Gruppo G
where P.CodFis = E.Persona and
      E.Iscrizione = R.Codice and
      R.SommaPagata >= 500 and
      R.Codice = G.Codice and
      G.Data.Mese = "Gennaio"
```

Problema 3 – Interrogazioni SQL (cont.)

3. Calcolare il comune di residenza e la data di nascita delle donne che hanno effettuato almeno una iscrizione individuale per la quale è stata assegnata una camera d'albergo situato in un comune diverso da quello in cui risiedono.

```
select S.NomeComune, S.RegioneComune, P.DataN
from Persona P, Risiede S, Esegue E, Individuo G, Situato T
where P.Sesso = 'f' and P.NomeComune = S.NomeComune and
      P.RegioneComune = S.RegioneComune and
      P.CodFis = E.Persona and E.Iscrizione = G.Codice and
      G.Albergo = T.Albergo and
      (T.NomeComune <> P.NomeComune or
       T.NomeRegione <> P.NomeRegione)
```

4. Per ogni persona che ha effettuato almeno una iscrizione individuale per la quale è stata assegnata una camera di categoria pari a 3, contare il numero di iscrizioni (individuali o di gruppo) effettuate.

```
select E.Persona, count(*)
from Esegue E
where E.Persona in (select Persona
                   from Esegue F, Individuo S
                   where F.Iscrizione = S.Codice)
group by E.Persona
```