

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 A

Specifica

Problema 1

Si richiede di progettare lo schema concettuale Entità-Relazione di un'applicazione relativa alle prenotazioni di posti in teatro. Di ogni teatro interessa il nome (identificativo), l'anno di fondazione, ed il comune in cui è situato. Di ogni comune interessa il nome (unico nella regione), la regione, ed il salario annuo medio dei suoi abitanti. Ogni posto di un teatro è identificato dal numero di sedia unico nell'ambito della fila del teatro stesso, ed è caratterizzato da un numero intero che indica la categoria (platea, galleria, ecc.). Di ogni prenotazione di posti interessa il codice (identificativo), la persona che l'ha effettuata, la data in cui è stata effettuata, e la somma pagata. Ogni prenotazione è o singola o collettiva. Ogni prenotazione singola è relativa ad una data per un teatro, e dà diritto all'assegnazione di esattamente un posto di quel teatro per quella data. Di ogni prenotazione singola interessa il posto di teatro assegnato, la data per la quale è stato assegnato il posto, ed anche la persona (una ed una sola) che occuperà il posto assegnato. Ovviamente un posto di teatro in una certa data può essere assegnato al massimo una volta. Ogni prenotazione collettiva è relativa invece ad un certo numero di posti, tutti per la stessa data e per lo stesso teatro. Di ogni prenotazione collettiva interessa quindi il numero di posti richiesti (non quali, o per quali persone), la data ed il teatro per i quali sono stati richiesti, ed anche lo sconto ottenuto sul prezzo della prenotazione. Di ogni persona interessa il codice fiscale (identificativo), la data di nascita, il sesso, il comune di nascita, ed il comune di residenza, con la data dalla quale la persona vi risiede.

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 prenotazioni singole si accede spesso separatamente rispetto alle prenotazioni collettive,
- quando si accede ad una prenotazione singola, si vogliono spesso conoscere le seguenti proprietà: il codice, la somma pagata, il posto assegnato, la data per la quale è stata assegnato, e la persona che occuperà il posto prenotato.

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

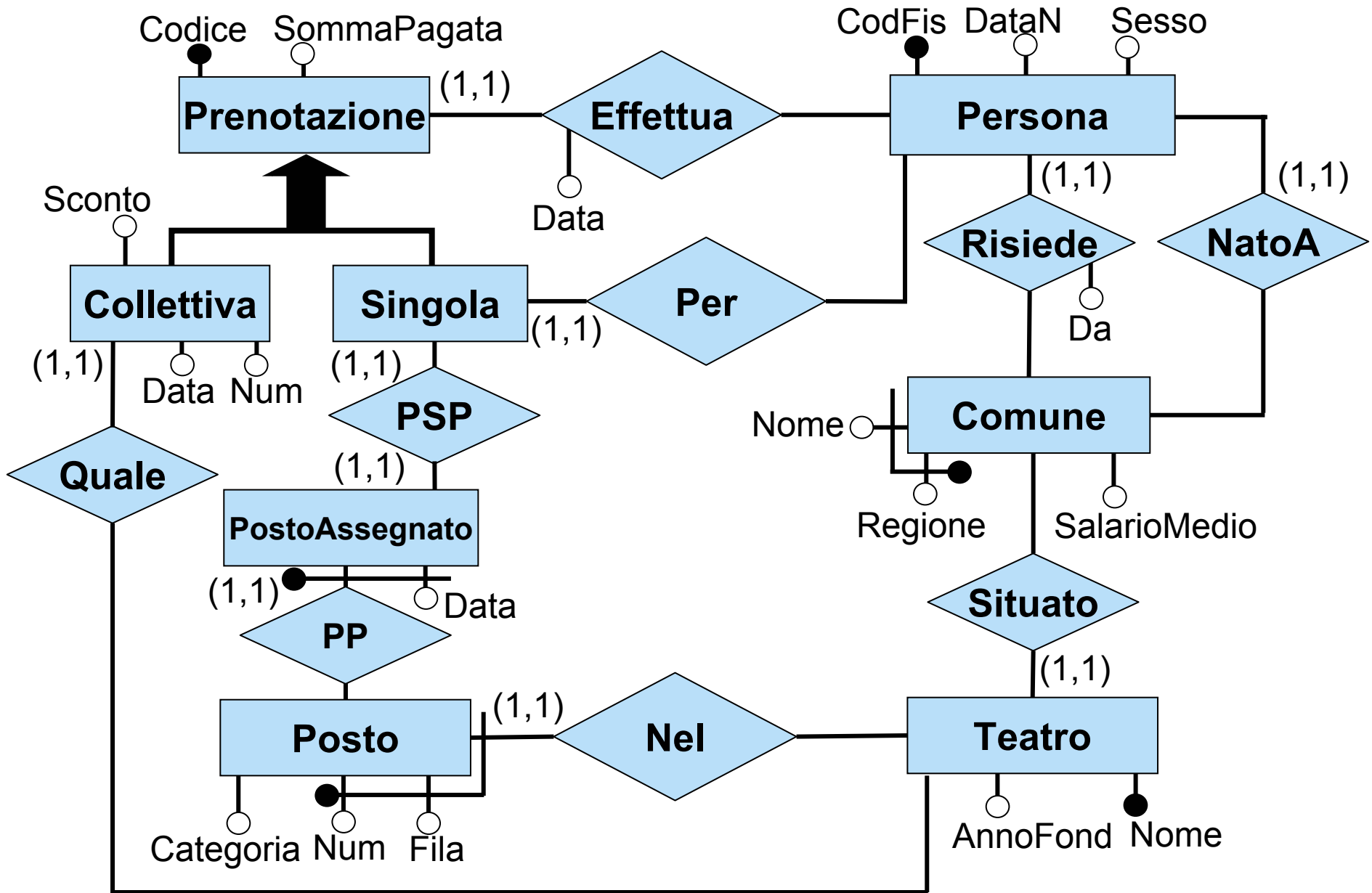
Specifica (cont.)

Problema 3

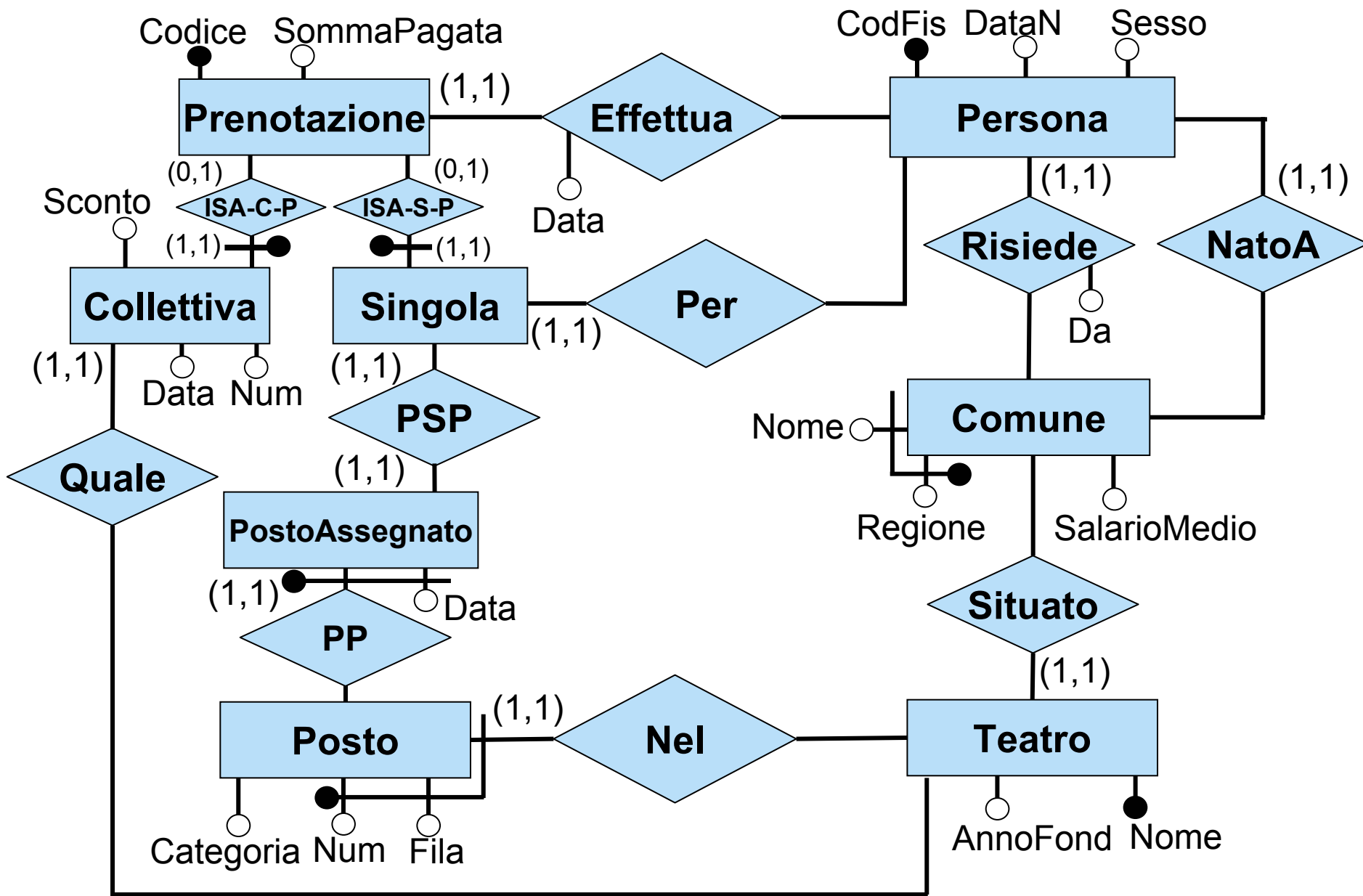
Sulla base dello schema relazionale definito per il problema 2, esprimere in SQL le seguenti interrogazioni:

1. Calcolare il codice fiscale ed il sesso delle persone che hanno effettuato almeno una prenotazione nel 2003.
2. Calcolare il codice fiscale e la data di nascita delle persone che hanno effettuato almeno una prenotazione collettiva per una data di agosto, e per la quale hanno pagato una somma di almeno 200 Euro.
3. Calcolare il comune di residenza e la data di nascita delle donne che hanno effettuato almeno una prenotazione singola per un teatro situato in un comune diverso da quello in cui risiedono.
4. Per ogni persona che ha effettuato almeno una prenotazione singola per la quale è stato assegnato un posto di categoria 1, contare il numero di prenotazioni (singole o collettive) effettuate.

Problema 1 – Schema concettuale



Problema 1 – Schema concettuale ristrutturato



Problema 2 – Vincoli sullo schema concettuale ristrutturato

- **Ogni istanza di Prenotazione partecipa ad ISA-C-P oppure ad ISA-S-P ma non ad entrambe** (dovuto all'eliminazione della generalizzazione).

Problema 2 – Schema logico

Prenotazione(Codice, SommaPagata)

foreign key: Prenotazione[Codice] \subseteq Effettua[Prenotazione]

Effettua(Prenotazione, Persona, Data)

foreign key: Effettua[Prenotazione] \subseteq Prenotazione[Codice]

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

Collettiva(Codice, Sconto, Data, Num)

foreign key: Collettiva[Codice] \subseteq Prenotazione[Codice]

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

Singola(Codice)

foreign key: Singola[Codice] \subseteq Prenotazione[Codice]

foreign key: Singola[Codice] \subseteq PSP[Singola]

foreign key: Singola[Codice] \subseteq Per[Singola]

Per(Singola, Persona)

foreign key: Per[Singola] \subseteq Singola[Codice]

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

Persona(CodFis, DataN, Sesso)

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

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

Risiede(Persona, NomeComune, RegioneComune, Da)

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

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

NatoA(Persona, NomeComune, RegioneComune)

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

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

Problema 2 – Schema logico (cont.)

Comune(Nome, Regione, SalarioMedio)

Situato(Teatro, NomeComune, RegioneComune)

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

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

Teatro(Nome, AnnoFond)

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

Posto(Num, Fila, Teatro, Categoria)

foreign key: Posto[Teatro] \subseteq Teatro[Nome]

PostoAssegnato(Num, Fila, Teatro, Data)

foreign key: PostoAssegnato[Num, Fila, Teatro] \subseteq Posto[Num, Fila, Teatro]

foreign key: PostoAssegnato[Num, Fila, Teatro, Data] \subseteq PSP[Num, Fila, Teatro, Data]

PSP(Num, Fila, Teatro, Data, Singola)

foreign key: PSP[Num, Fila, Teatro, Data] \subseteq PostoAssegnato[Num, Fila, Teatro, Data]

foreign key: PSP[Singola] \subseteq Singola[Codice]

chiave: Num, Fila, Teatro, Data

Quale(Collettiva, Teatro)

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

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

Vincoli esterni (di generalizzazione):

Collettiva[Codice] \cap Singola[Codice] = \emptyset

Prenotazione[Codice] \subseteq Collettiva[Codice] \cup Singola[Codice]

Problema 2 – Ristrutturazione dello schema logico

- Poiché alle prenotazioni singole si accede spesso separatamente rispetto alle prenotazioni collettive, effettuiamo la decomposizione orizzontale di **Prenotazione** in:

PrenotazioneCollettiva(Codice, SommaPagata)

PrenotazioneSingola(Codice, SommaPagata)

- Poiché quando si accede ad una prenotazione singola, si vogliono spesso conoscere il codice, la somma pagata, il posto assegnato, la data per la quale è stata assegnato, e la persona che occuperà il posto prenotato, effettuiamo l'accorpamento di **PrenotazioneSingola**, **Singola**, **PSP**, e **Per** ottenendo la relazione:

Singola(Codice, SommaPagata, Num, Fila, Teatro, Data, Persona)

- Per eliminare la relazione inutile **PostoAssegnato**, effettuiamo l'accorpamento di **PostoAssegnato** con **Singola**.

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

Problema 2 – Schema logico ristrutturato

PrenotazioneCollettiva(Codice, SommaPagata)

foreign key: PrenotazioneCollettiva[Codice] \subseteq Effettua[Prenotazione]

foreign key: PrenotazioneCollettiva[Codice] \subseteq Collettiva[Codice]

Effettua(Prenotazione, Persona, Data)

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

Collettiva(Codice, Sconto, Data , Num)

foreign key: Collettiva[Codice] \subseteq PrenotazioneCollettiva[Codice]

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

Singola(Codice, SommaPagata, Num, Fila, Teatro, Data, Persona)

foreign key: Singola[Num,Fila,Teatro] \subseteq Posto[Num,Fila,Teatro]

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

chiave: Num, Fila, Teatro, Data

Persona(CodFis, DataN, Sesso)

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

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

Risiede(Persona, NomeComune, RegioneComune, Da)

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

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

Problema 2 – Schema logico ristrutturato (cont.)

NatoA(Persona, NomeComune, RegioneComune)

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

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

Comune(Nome, Regione, SalarioMedio)

Situato(Teatro, NomeComune, RegioneComune)

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

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

Teatro(Nome, AnnoFond)

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

Posto(Num, Fila, Teatro, Categoria)

foreign key: Posto[Teatro] \subseteq Teatro[Nome]

Quale(Collettiva, Teatro)

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

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

Vincoli esterni:

Collettiva[Codice] \cap Singola[Codice] = \emptyset

Effettua[Prenotazione] \subseteq Collettiva[Codice] \cup Singola[Codice]

Problema 3 – Interrogazioni SQL

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

```
select P.CodFis, P.Sesso
from Persona P, Effettua 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 prenotazione collettiva per una data di agosto, e per la quale hanno pagato una somma di almeno 200 Euro.

```
select P.CodFis, P.DataN
from Persona P, Effettua E, PrenotazioneCollettiva R,
      Collettiva C
where P.CodFis = E.Persona and
      E.Prenotazione = R.Codice and
      R.SommaPagata >= 200 and
      R.Codice = C.Codice and
      C.Data.Mese = "Agosto"
```

Problema 3 – Interrogazioni SQL (cont.)

3. Calcolare il comune di residenza e la data di nascita delle donne che hanno effettuato almeno una prenotazione singola per un teatro situato in un comune diverso da quello in cui risiedono.

```
select S.NomeComune, S.RegioneComune, P.DataN
from Persona P, Risiede R, Effettua E, Singola G, Situato S
where P.Sesso = 'f' and P.CodFis = R.Persona and
      P.CodFis = E.Persona and E.Prenotazione = G.Codice and
      G.Teatro = S.Teatro and
      (S.NomeComune <> R.NomeComune or
       S.NomeRegione <> R.NomeRegione)
```

4. Per ogni persona che ha effettuato almeno una prenotazione singola per la quale è stato assegnato un posto di categoria 1, contare il numero di prenotazioni (singole o collettive) effettuate.

```
select E.Persona, count(*)
from Effettua E
where E.Persona in (select Persona
                   from Effettua F, Singola S
                   where F.Prenotazione = S.Codice)
group by E.Persona
```