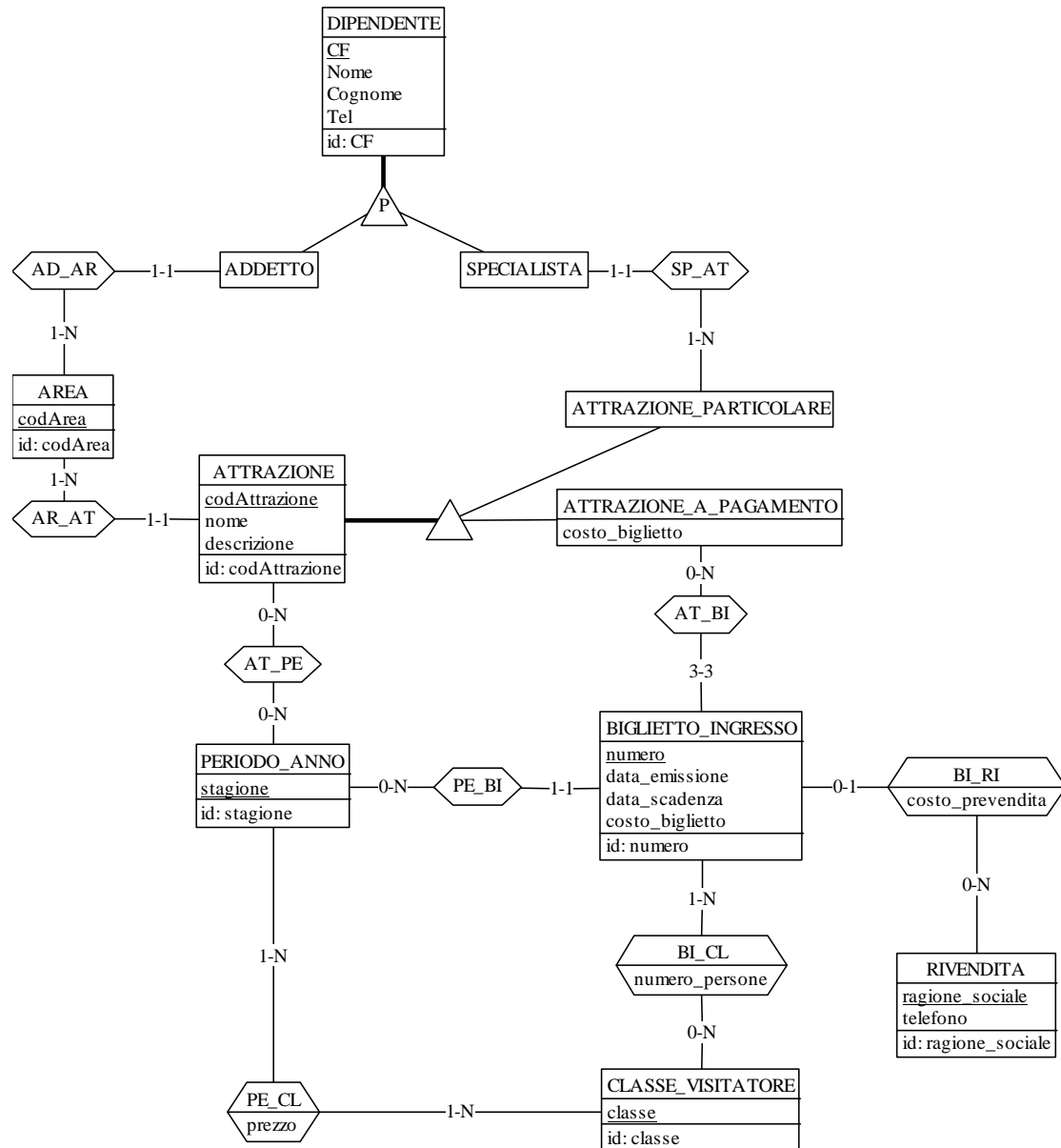


Sistemi Informativi L-B - Prova integrata
16 gennaio 2007
Risoluzione

Tempo a disposizione: 2 ore

1) Progettazione concettuale (5 punti)



Commenti:

- Il costo di prevendita viene specificato nell'associazione BI_RI perché si assume che dipenda dal costo complessivo del biglietto
- Si è supposto che uno specialista non sia anche un addetto

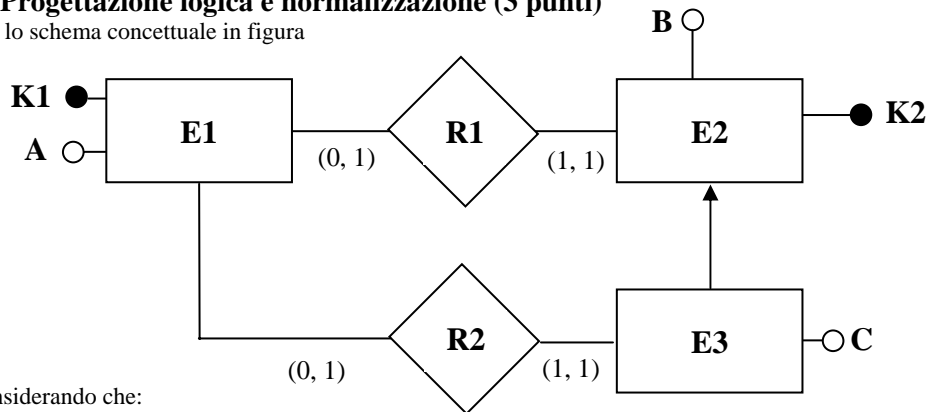
Sistemi Informativi L-B - Prova integrata

16 gennaio 2007

Risoluzione

2) Progettazione logica e normalizzazione (3 punti)

Dato lo schema concettuale in figura



e considerando che:

- a) tutti gli attributi sono di tipo INT;
- b) le entità E2 e E3 vengono accorpate, inglobando le associazioni R1 e R2;
- c) istanze di E3 non sono mai associate, tramite il cammino R2-E1-R1, a istanze di E2 aventi B = C

si progettino gli opportuni schemi relazionali e si definiscano **nel database STUDENTI** tali schemi facendo uso dell'SQL di DB2; per gli eventuali vincoli non esprimibili a livello di schema si predispongano opportune **query di verifica da eseguire prima di effettuare inserimenti di tuple**, allo scopo di evitare che tali inserimenti violino i vincoli stessi.

```
CREATE TABLE E1 (  
  K1 INT NOT NULL PRIMARY KEY,  
  A INT NOT NULL);
```

```
CREATE TABLE E2 (  
  K2 INT NOT NULL PRIMARY KEY,  
  B INT NOT NULL,  
  K1R1 INT NOT NULL REFERENCES E1,  
  C INT,  
  K1R2 INT REFERENCES E1,  
  SEL SMALLINT CHECK (SEL IN (2,3)),  
  CONSTRAINT CHECK_E3  
    CHECK ( (SEL=2 AND C IS NULL AND K1R2 IS NULL) OR  
            (SEL=3 AND C IS NOT NULL AND K1R2 IS NOT NULL AND  
              (B<>C OR K1R1<>K1R2)) ));
```

La violazione del vincolo di cui al punto c) può avvenire o inserendo una tupla in E3 (SEL=3), e quindi un legame in R2 e uno in R1 (2 possibili violazioni), o una tupla con SEL=2 (1 sola possibile violazione per l'aggiunta di un legame in R1). Il check SQL evita solo il caso banale in cui il cammino R2-E1-R1 associa una tupla con se stessa. Primo caso: è data la tupla (k2,b,k1r1,c,k1r2,3) che si intende inserire in E2:

```
SELECT * FROM E2  
WHERE (E2.K1R1 = k1r2 AND E2.B = c) -- ok se non restituisce una tupla  
OR     (E2.K1R2 = k1r1 AND E2.C = b) -- violazione per aggiunta di un legame in R2  
-- violazione per aggiunta di un legame in R1
```

Secondo caso: è data la tupla (k2,b,k1r1,null,null,2) che si intende inserire in E2:

```
SELECT * FROM E2  
WHERE (E2.K1R2 = k1r1 AND E2.C = b) -- ok se non restituisce una tupla  
-- violazione per aggiunta di un legame in R1
```

3) Elaborazione di interrogazioni (2 punti)

a) Dati gli schemi R1(AB), R2(BC), e R3(CD), il join tra R2 e R3 restituisce un numero di tuple pari a $0.8 * NR(R2)$, in quanto R2.C è una foreign key non nulla nell'80% dei casi. Per ognuna di queste tuple (ognuna, in quanto R2.B è chiave primaria, quindi non ha valori nulli) nel secondo join è necessario scandire tutta la relazione interna, e pertanto il costo del secondo join è pari a

$$0.8 * NR(R2) * NP(R1)$$

b) Per quanto visto, il primo join mantiene solo l'80% dei valori di R2.B. Pertanto, se R1.B non avesse valori nulli, il risultato si potrebbe stimare pari a $0.8 * NR(R1)$ tuple, assumendo i valori della foreign key R1.B uniformemente distribuiti. Questo valore, con la stessa ipotesi, si riduce a $0.7 * 0.8 * NR(R1)$ quando R1.B ha nel 30% dei casi valore nullo