

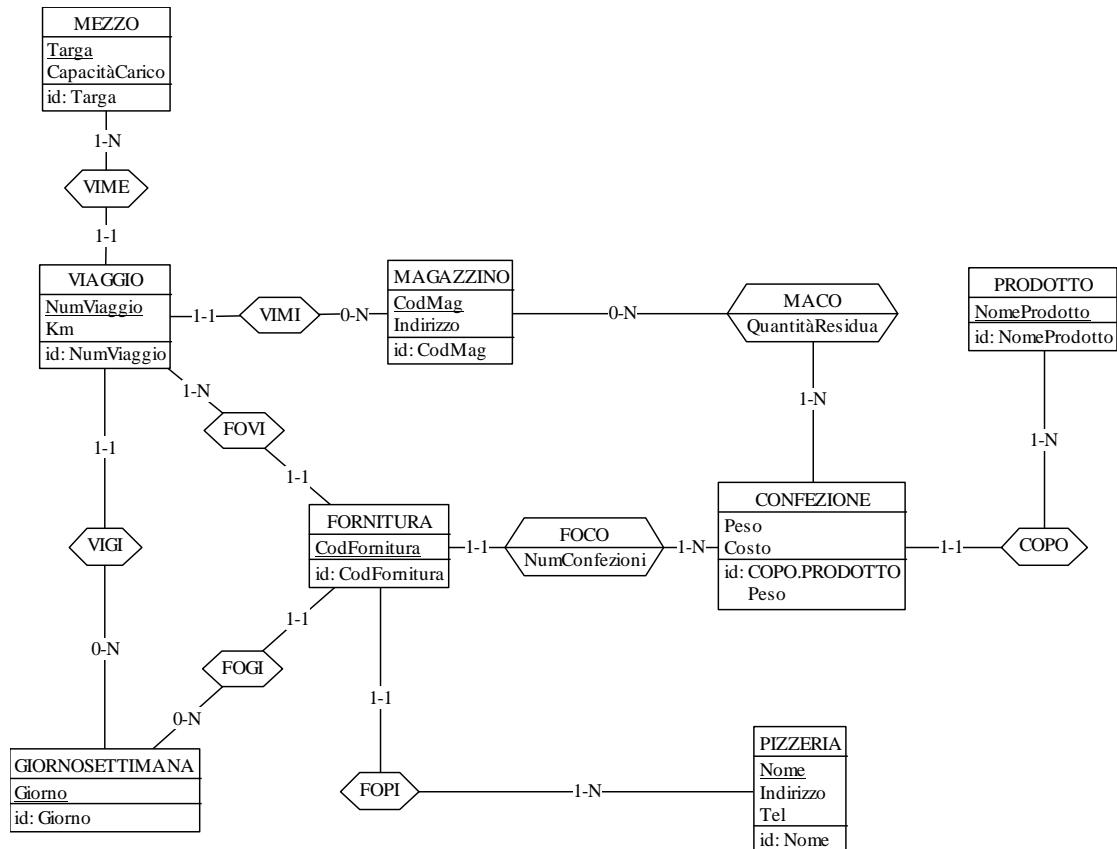
Sistemi Informativi L-B

9 luglio 2008

Risoluzione

Tempo a disposizione: 2 ore

1) Progettazione concettuale (5 punti)



Commenti:

- La difficoltà principale dell'esercizio è probabilmente nella modellazione delle forniture delle confezioni. La soluzione proposta considera che una **FORNITURA** riguardi un singolo tipo di confezione e un singolo giorno della settimana. In questo modo si ovvia alla difficoltà di dover considerare che in giorni diversi una stessa confezione può essere richiesta in quantità diverse e diventa immediato specificare, tramite l'associazione **FOVI**, la composizione del carico di ogni **VIAGGIO**.
- L'associazione **FOGI**, che specifica il giorno della settimana in cui eseguire una fornitura è ridondante, in quanto derivabile da **FOVI** e **VIGI** (ovvero, se una fornitura è associata a un viaggio che ha luogo, ad esempio, il martedì, allora la fornitura è associata a martedì). Si è tuttavia ritenuto opportuno mantenere tale ridondanza in modo da separare chiaramente la parte che riguarda le richieste delle pizzerie (che quindi includono anche i dati di **FOGI**) dalla parte che riguarda più propriamente l'organizzazione dei viaggi (che include i dati di **FOVI** e **VIGI**).

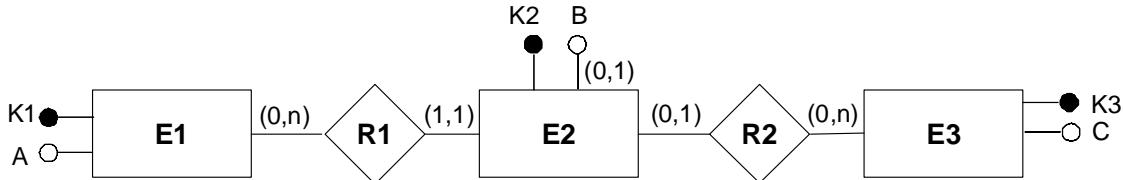
Sistemi Informativi L-B

9 luglio 2008

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 associazioni R1 e R2 non vengono tradotte separatamente;
- c) l'attributo B è definito per un'istanza di E2 se e solo se tale istanza non partecipa a R2;
- d) Se un'istanza di E2 partecipa a R2, allora le istanze di E1 ed E3 a cui essa è associata sono tali per cui è $A < C$;

si progettino gli opportuni schemi relazionali e si definiscano 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 E3 (
    K3 INT NOT NULL PRIMARY KEY,
    C INT NOT NULL
);
```

```
CREATE TABLE E2 (
    K2 INT NOT NULL PRIMARY KEY,
    B INT,
    K1 INT NOT NULL REFERENCES E1,
    K3 INT REFERENCES E3,
    CONSTRAINT BR2 CHECK ((B IS NULL AND K3 IS NOT NULL) OR
                           (B IS NOT NULL AND K3 IS NULL)) );
```

L'inserimento di una tupla (k2,NULL,k1,k3) in E2 deve rispettare il vincolo d), pertanto va eseguita la query:

```
SELECT * FROM E1, E3          -- ok se restituisce una tupla
WHERE K1 = k1
AND K3 = k3
AND A < C;
```

3) Data Base fisico (2 punti)

Il Buffer Manager mette a disposizione i seguenti metodi:

getAndPinPage:	per richiedere il caricamento di una pagina in memoria e porvi un pin
unPinPage:	per rilasciare la pagina e eliminare un pin
setDirty:	per indicare che la pagina è stata modificata, ovvero è dirty
flushPage:	per forzare la scrittura della pagina su disco, rendendola così non-dirty