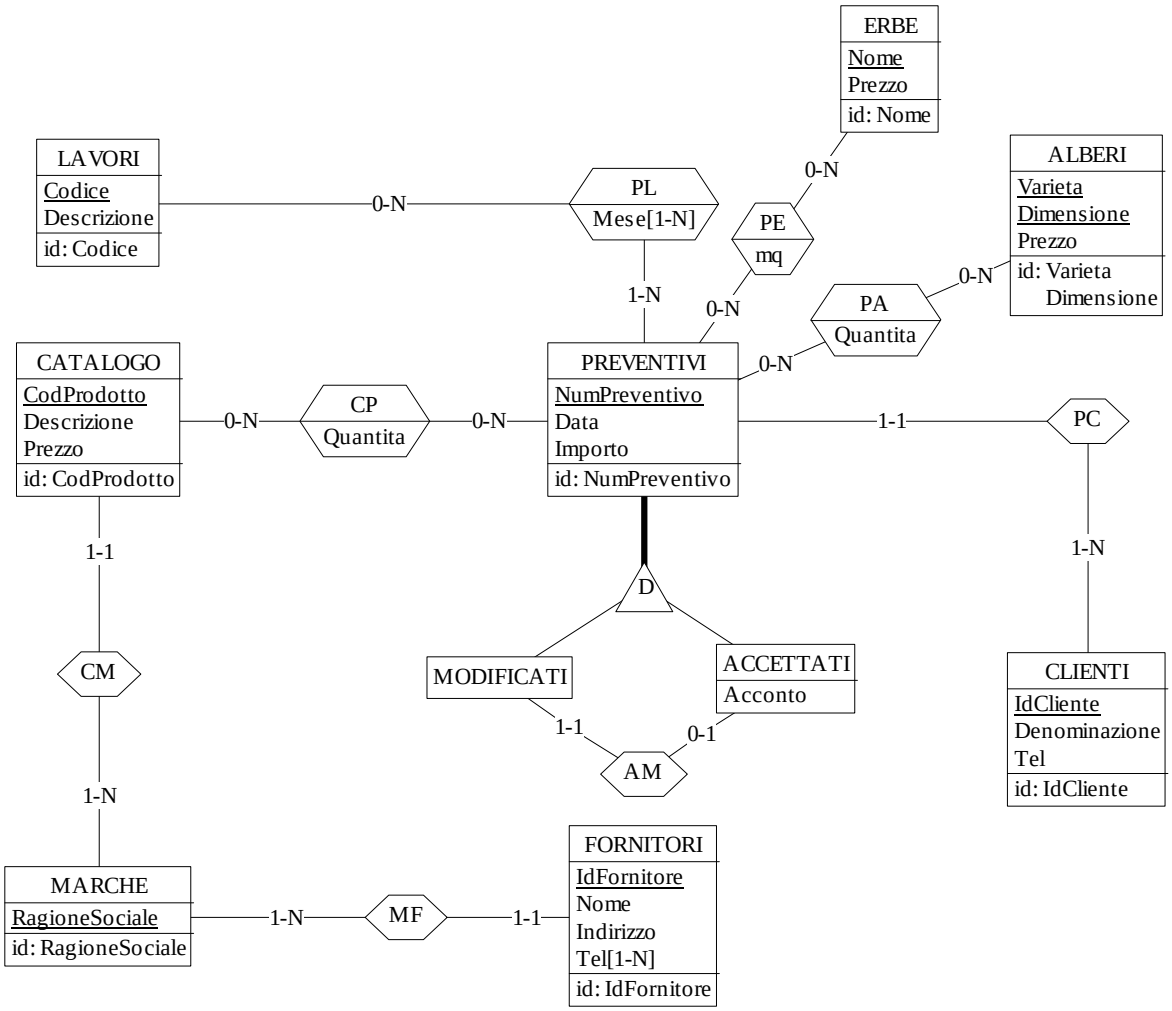
Sistemi Informativi L-B
14 maggio 2010
Risoluzione

Tempo a disposizione: 2 ore

1) Progettazione concettuale (5 punti)



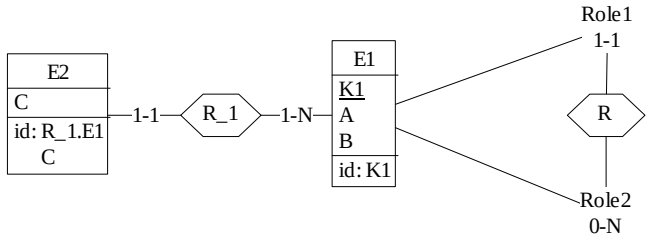
Commenti:

- L'entità PREVENTIVI mantiene tutti i preventivi presenti in un certo istante nel DB. Tra questi vi sono quelli ACCETTATI i quali, se frutto di una modifica, fanno riferimento al preventivo originale presente in MODIFICATI. La gerarchia è parziale in quanto vi sono anche dei preventivi in sospeso, ovvero né accettati né modificati e per i quali non sono ancora passati 30 giorni dalla Data in cui sono stati predisposti.
- Un preventivo deve includere almeno un albero o un tipo di erba da seminare (per la quale l'associazione PE specifica i metri quadri)

Sistemi Informativi L-B
14 maggio 2010
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) viene generata **una sola tabella (non normalizzata)** per tutto lo schema;

si progetti lo schema relazionale e si definisca tale schema 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 E12(
K1 INT NOT NULL,
C INT NOT NULL,
A INT NOT NULL,
B INT NOT NULL,
K1R INT NOT NULL,
PRIMARY KEY (K1,C) );
```

Poiché E12 non è normalizzata, vi sono 3 FD che devono essere fatte rispettare (K1 $\rightarrow$ A, K1 $\rightarrow$ B e K1 $\rightarrow$ K1R). Pertanto, in fase di inserimento di una tupla del tipo (k1,c,a, b,k1r), è necessario eseguire la query:

```
SELECT * FROM E12  -- ok se non  restituisce nessuna tupla
WHERE (K1 = k1 AND A <> a)
 OR       (K1 = k1 AND B <> b)
 OR       (K1 = k1 AND K1R <> k1r);
```

Inoltre va verificato il vincolo che k1r si riferisca ad un valore di K1 esistente:

```
SELECT * FROM E12  -- ok se  restituisce almeno una tupla
WHERE K1 = k1r;
```

3) DB Fisico (2 punti)

a) SELECT * FROM R WHERE A = 5
E' necessario determinare quante pagine, in media, occupa un singolo valore di A. In media un valore di A è ripetuto NT/NK(A) volte. Poiché in ogni pagina vi sono NT/NP tuple si ha:
(NT/NK(A)) / (NT/NP) = NP/NK(A)
arrotondando poi all'intero superiore

b) SELECT * FROM R WHERE B = 10
In questo caso il costo si può stimare pari a NP, in quanto le tuple con B = 10 sono sparse su tutto il file

c) SELECT * FROM R WHERE C BETWEEN 30 AND 50
Il costo è ancora NP, perché non vi è alcun ordinamento su C