

# Sistemi Informativi L-B

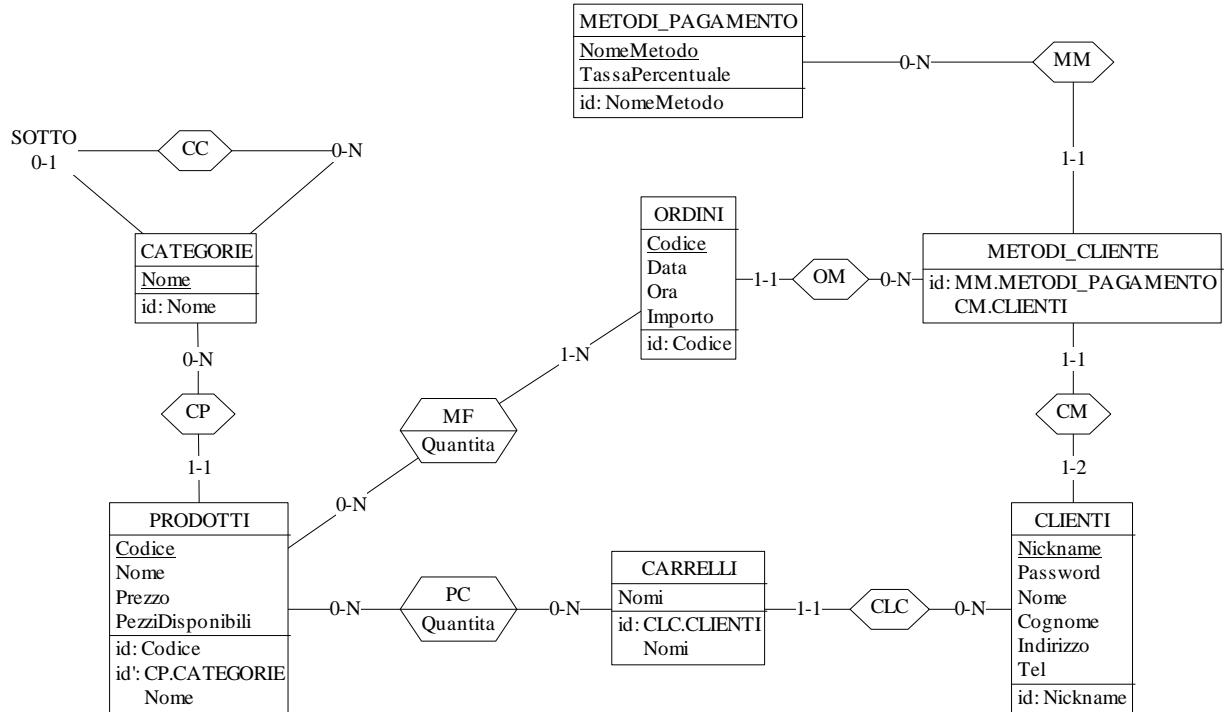
15 giugno 2010

Risoluzione

**Tempo a disposizione: 2 ore**

---

## 1) Progettazione concettuale (5 punti)



### Commenti:

- L'entità **METODI\_CLIENTE** è ottenuta mediante reificazione, e serve a vincolare il metodo di pagamento di un ordine
- Si noti che **non** esiste alcuna relazione tra **ORDINI** e **CARRELLI**, in quanto quando un ordine viene generato (inserito nel DB) il corrispondente carrello viene eliminato (in pratica il tutto avviene all'interno di un'unica transazione)
- Un carrello è inizialmente vuoto; ciò giustifica la cardinalità minima 0 con cui **CARRELLI** partecipa nell'associazione **PC**. Viceversa, un ordine consiste di almeno 1 prodotto
- Le **CATEGORIE** formano un albero, la cui radice non è sottocategoria di nessun'altra categoria

# Sistemi Informativi L-B

15 giugno 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) l'associazione R1 non viene tradotta separatamente;
- c) le entità E1, E2 ed E3 vengono tradotte assieme;
- d) l'entità E4 viene tradotta separatamente;

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 E4(
K1 INT NOT NULL PRIMARY KEY,
D INT NOT NULL);
```

```
CREATE TABLE E1(
K1 INT NOT NULL PRIMARY KEY,
A INT NOT NULL,
SEL23 SMALLINT NOT NULL CHECK (SEL2 IN (2,3)),
-- SEL23 = 2 (3) : la tupla e' un'istanza di E2 (E3)
K2 INT,
B INT,
C INT,
K1E4 INT REFERENCES E4,
CONSTRAINT E2_E3 CHECK (
(SEL23 = 2 AND K2 IS NOT NULL AND B IS NOT NULL AND C IS NULL AND K1E4 IS NULL) OR
(SEL23 = 3 AND K2 IS NULL AND B IS NULL AND C IS NOT NULL AND K1E4 IS NOT NULL)) );
```

```
ALTER TABLE E4
ADD CONSTRAINT FK FOREIGN KEY (K1) REFERENCES E1;
-- va dichiarato dopo la definizione di E1 per evitare riferimenti circolari!
```

In fase di inserimento in E4 di una tupla del tipo (k1,d), è necessario garantire che k1 referenzi un'istanza di E2:

```
SELECT * FROM E1 -- ok se restituisce una tupla
WHERE K1 = k1 AND SEL23 = 2;
```

Inoltre, inserendo in E1 una tupla del tipo (k1,a2,k2,b,NULL,NULL), bisogna verificare che k2 non sia un valore duplicato:

```
SELECT * FROM E1 -- ok se non restituisce nessuna tupla
WHERE K2 = k2;
```

## 3) Transazioni (2 punti)

Si vedano le dispense del corso

