

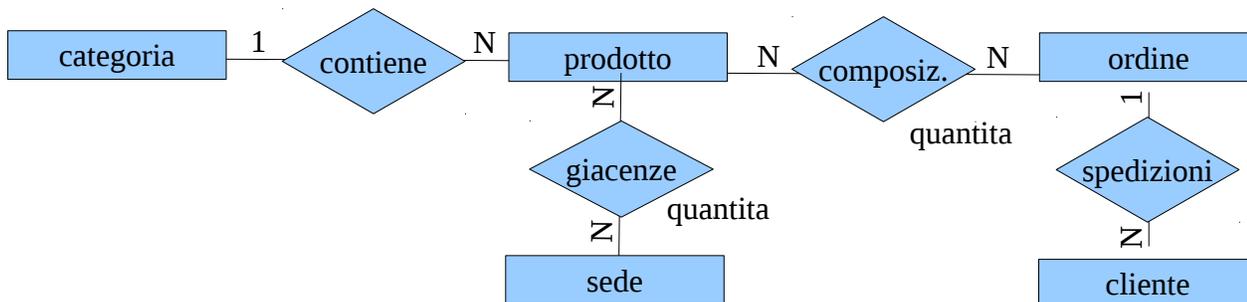
Tema d'esame 2001

Una grande società commerciale...

vocabolario dei termini

- Negozio: negozio in una città, con via, civico e telefono
- Magazzino: magazzino in una città, con via, civico e con telefono. Poiché negozio e magazzino hanno gli stessi attributi, si possono ricondurre ad una sola entità chiamata sede (con un nuovo attributo che permette di distinguerli tra loro).
- Sede: contiene diversi tipi di prodotti, può essere negozio oppure no, in tal caso è magazzino.
- disponibilità o quantità: quantità (*disponibile*) di ogni prodotto in ogni sede
- NOTA sul tempo di consegna: il testo dice letteralmente "si vuole che..." che è realisticamente impossibile a causa di imprevisti e ritardi nelle consegne. In un caso reale andrebbe registrato il tempo *previsto* e il tempo *effettivo* di consegna. Non essendo richiesto, si indicherà solo nella pagina web il tempo di consegna "ideale", che dipende solo dal fatto che il prodotto si trovi in un negozio oppure in un magazzino. Quindi non è necessario memorizzare il tempo, ma basta conoscere dove si trova il prodotto da consegnare, e poi applicare un'istruzione di condizione (if).
- (tipo di)Prodotto: insieme di oggetti dello stesso tipo, con stessa descrizione e stesso prezzo. Per quanto riguarda il (tipo di)prodotto si ricorda che: un prodotto appartiene ad una categoria, un prodotto giace in una sede, un prodotto può essere contenuto in un ordine.
- Categoria: bisogna raggruppare più tipi di prodotti in una categoria più ampia (ad esempio, DVD player marca Pony e DVD player marca Sonyo in una generica categoria "dvd player").
- Ordine: richiesta del cliente, dovrà contenere più o meno gli stessi dati che poi finiscono anche nella fattura: numero, data, tipi di prodotti e la quantità (*evasa*) per OGNI tipo do prodotto nell'ordine. Il totale può essere un campo calcolato...
- NOTA sul cliente: questo importante dato non è indicato nel testo del problema! Dati a piacere...
- La funzione di aggiornamento automatico

schema E-R



regole di lettura

Ogni ordine deve essere spedito ad un cliente

Ad ogni cliente possono essere spediti uno o più ordini

Ogni ordine deve essere composto da uno o più prodotti

Ogni (tipo di) prodotto può comporre uno o più ordini

Ogni (tipo di) prodotto può giacere in uno o più sedi

In ogni sede possono giacere uno o più prodotti

Ogni categoria contiene uno o più prodotti

Ogni prodotto è contenuto in una categoria

schema logico

categorie (idcategoria, descrizione)

clienti (cfcliente, nome, cognome, via, civico, città, provincia)

prodotti (idprodotto, descrizione, prezzo, immagine, idcategoria)

ordini (fatturaordine, dataordine, cfcliente, dataconsegna)

composizioni (fatturaordine, dataordine, idprodotto, quantitaevasa)

sedi (idsede, via, civico, città, provincia, telefono, negozio)

giacenze (idprodotto, cittasede, quantitadisponibile)

NOTA: giacenze e composizioni hanno una chiave primaria composta dalle due chiavi esterne

Osservazioni:

per come è definita la relazione "composizioni", non è possibile inserire due volte lo stesso prodotto nella stessa fattura, ma questa è una condizione realistica.

per come è definita la relazione "giacenze", non è possibile inserire due volte la quantità di un prodotto nella stessa sede, ma questa è una condizione realistica.

testing nella pagina seguente...

Testing

categorie

idcategoria	descrizione
44	accessori
45	dvd
11	tv

prodotti

idprodotto	descrizione	prezzo	idcategoria	immagine
a201	antenna	3.50	44	/img/1.jpg
b202	tv Pony 15"	199.99	11	
c900	tv Pony 20"	12.50	11	

clienti

cfcliente	nome	cognome	via	civico	citta	provincia
rss	Mario	Rossi				PR
bnc	Anna	Bianchi				RE

ordini

fattura ordine	dataordine	cfcliente	consegna
2300	2001-01-01	rss	2001-01-02
2301	2001-02-20	rss	2001-02-22
2302	2001-02-20	bnc	2001-02-21

composizioni

fattura ordine	dataordine	idprodotto	quantitaevasa
2300	2001-01-01	a201	1
2300	2001-01-01	c900	3
2301	2001-02-20	b202	3
2302	2001-02-20	a201	1
2302	2001-02-20	b202	1
2302	2001-02-20	c900	1

sedi

citta	negozio
Fidenza	true
Parma	true
Reggio Emilia	false

giacenze

idprodotto	cittasede	quantitadisponibile
a201	Fidenza	50
a201	Parma	41
a201	Reggio Emilia	75
b202	Fidenza	12
b202	Parma	22
b202	Reggio Emilia	17
c900	Fidenza	0
c900	Parma	23
c900	Reggio Emilia	29

Schema dei dati (documenta i tipi di dato e i vincoli)

archivio	campo	vincolo	tipo	dimensione	
categorie	idcategoria	c.primaria	NUMERIC	3,0	
	descrizione		VARCHAR	40	
clienti	cfcliente	c.primaria	CHAR	16	
	nome		VARCHAR	30	
	cognome		VARCHAR	30	
	via		VARCHAR	40	
	civico		VARCHAR	9	
	citta		VARCHAR	30	
	provincia		CHAR	2	
prodotti	idprodotto	c.primaria	CHAR	6	
	descrizione		VARCHAR	50	
	prezzo		NUMERIC	7,2	
	immagine		VARCHAR	200	
	idcategoria	c.esterna	NUMERIC	3,0	
ordini	fatturaordine	c.primaria	NUMERIC	10,0	
	dataordine	composta	DATE	-	
	cfcliente		CHAR	16	
	dataconsegna		DATE	-	
composizioni	fatturaordine	c.esterna	NUMERIC	10,0	} c.prim. composta
	dataordine	composta	DATE	-	
	idprodotto	c.esterna	CHAR	6	
	quantitaevasa		NUMERIC	5,0	
sedi	citta	c.primaria	VARCHAR	30	
	negozio		BOOLEAN	-	
giacenze	idprodotto	c.esterna	CHAR	6	} c.prim. composta
	cittasede	c.esterna	VARCHAR	30	
	quantitadisponibile		NUMERIC	6,0	

Interrogazioni

1. Visualizzare i prodotti di una stessa categoria in un intervallo di prezzo prescelto;

```
SELECT prodotti.*
FROM prodotti NATURAL JOIN categorie
WHERE categorie.descrizione=$varcategoria
AND prezzo>=$varminimo
AND prezzo<=$varmassimo;
```

2. Visualizzare l'elenco degli ordini evasi entro le 24 ore dai vari punti vendita con riferimento a un prescelto arco di tempo.

3. aggiornare la giacenza dei prodotti in funzione degli ordini evasi;

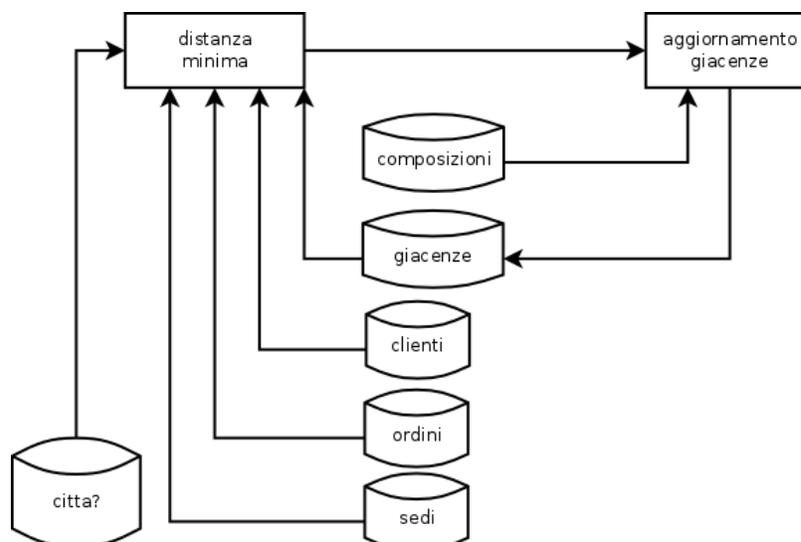
NOTA: quando viene evasa una certa quantità (ordinata) di un certo tipo di prodotto, la quantità (disponibile) in quella sede deve essere diminuita. Le funzioni che si attivano automaticamente vengono chiamate Trigger...

Per rispondere a queste domande, negli ordini dovrebbe essere contenuta la sede da dove viene evaso l'ordine. Questa informazione non è presente nella tabella ordini perché è troppo difficile considerare questo aspetto.

Immaginiamo che un cliente ordini una tv che è presente in tutti e venti i punti vendita: chi decide da dove prelevare il prodotto? Bisognerebbe considerare la distanza in km tra i magazzini e il cliente! Per risparmiare sui trasporti servirebbe un database delle città italiane e delle loro distanze e una funzione che *calcola la distanza* minima dal cliente, ma questo è troppo impegnativo per questo tipo di esercizio.

Ma finché la domanda "Da dove viene evaso l'ordine?" è senza risposta, non si può rispondere nemmeno alle ultime due richieste.

A titolo illustrativo, la formula matematica che aggiorna la quantità disponibile in una sede sarebbe:
 $\text{quantitadisponibile} = \text{quantitadisponibile} - \text{quantitaevasa}$
e questo sarebbe lo schema delle funzioni



Fabio Proietti (c) 2013 Licenza: <http://creativecommons.org/licenses/by-sa/3.0/>

```
CREATE TABLE societa.categorie (  
    idcategoria NUMERIC(3,0) PRIMARY KEY,  
    descrizione VARCHAR(40)  
);  
  
CREATE TABLE societa.clienti (  
    cfcliente CHAR(16) PRIMARY KEY,  
    nome VARCHAR(30),  
    cognome VARCHAR(30),  
    via VARCHAR(40),  
    civico VARCHAR(9),  
    citta VARCHAR(30),  
    provincia CHAR(2)  
);  
  
CREATE TABLE societa.prodotti (  
    idprodotto CHAR(6) PRIMARY KEY,  
    descrizione VARCHAR(50),  
    prezzo NUMERIC(7,2),  
    immagine VARCHAR(200),  
    idcategoria VARCHAR(40)  
);  
  
CREATE TABLE societa.ordini (  
    fatturaordine NUMERIC(10),  
    dataordine DATE,  
    cfcliente CHAR(16) REFERENCES societa.clienti(cfcliente),  
    consegna DATE,  
    CONSTRAINT chiavep_composta PRIMARY  
KEY(fatturaordine,dataordine)  
);  
  
CREATE TABLE societa.composizioni (  
    fatturaordine NUMERIC(10),  
    dataordine DATE,  
    idprodotto CHAR(6) REFERENCES societa.prodotti(idprodotto),  
    quantitaevasa NUMERIC(5,0),  
    CONSTRAINT chiavee_composta FOREIGN  
KEY(fatturaordine,dataordine) REFERENCES  
societa.ordini(fatturaordine,dataordine)  
);  
  
CREATE TABLE societa.sedi (  
    citta VARCHAR(30) PRIMARY KEY,  
    negozio BOOLEAN  
);  
  
CREATE TABLE societa.giacenze (  
    idprodotto CHAR(6) REFERENCES societa.prodotti(idprodotto),  
    cittasede VARCHAR(30) REFERENCES societa.sedi(citta),  
    quantitadisponibile NUMERIC(6)  
);
```

COMANDI ALTERNATIVI PER LIBREOFFICE

```
CREATE TABLE categorie (  
    idcategoria NUMERIC(3,0) PRIMARY KEY,  
    descrizione VARCHAR(40)  
);
```

```
CREATE TABLE clienti (  
    cfcliente CHAR(16) PRIMARY KEY,  
    nome VARCHAR(35),  
    cognome VARCHAR(35),  
    via VARCHAR(40),  
    civico VARCHAR(6),  
    citta VARCHAR(35),  
    provincia CHAR(2)  
);
```

```
CREATE TABLE prodotti (  
    idprodotto CHAR(4) PRIMARY KEY,  
    descrizione VARCHAR(35),  
    prezzo NUMERIC(6,2),  
    immagine VARCHAR(100),  
    idcategoria VARCHAR(40)  
);
```

```
CREATE TABLE ordini (  
    fatturaordine NUMERIC(10),  
    dataordine DATE,  
    cfcliente CHAR(16),  
    consegna DATE,  
    FOREIGN KEY (cfcliente) REFERENCES clienti(cfcliente),  
    PRIMARY KEY(fatturaordine,dataordine)  
);
```

```
CREATE TABLE composizioni (  
    fatturaordine NUMERIC(10),  
    dataordine DATE,  
    idprodotto CHAR(4),  
    quantitaevasa NUMERIC(3),  
    FOREIGN KEY(idprodotto) REFERENCES prodotti(idprodotto),  
    FOREIGN KEY(fatturaordine,dataordine) REFERENCES  
  
    ordini(fatturaordine,dataordine)  
);
```

```
CREATE TABLE sedi (  
    citta VARCHAR(35) PRIMARY KEY,  
    negozio BOOLEAN  
);
```

```
CREATE TABLE giacenze (  
    idprodotto CHAR(4) ,
```

Fabio Proietti (c) 2013 Licenza: <http://creativecommons.org/licenses/by-sa/3.0/>

```
cittasede VARCHAR(35),  
quantitadisponibile NUMERIC (4),  
FOREIGN KEY (idprodotto) REFERENCES prodotti(idprodotto),  
FOREIGN KEY (cittasede) REFERENCES sedi(citta)  
);
```

OVVIAMENTE VA SVOLTO ANCHE IL TEMA INIZIALE