

Considerazioni sulla complessità, flessibilità ed efficienza di una base dati

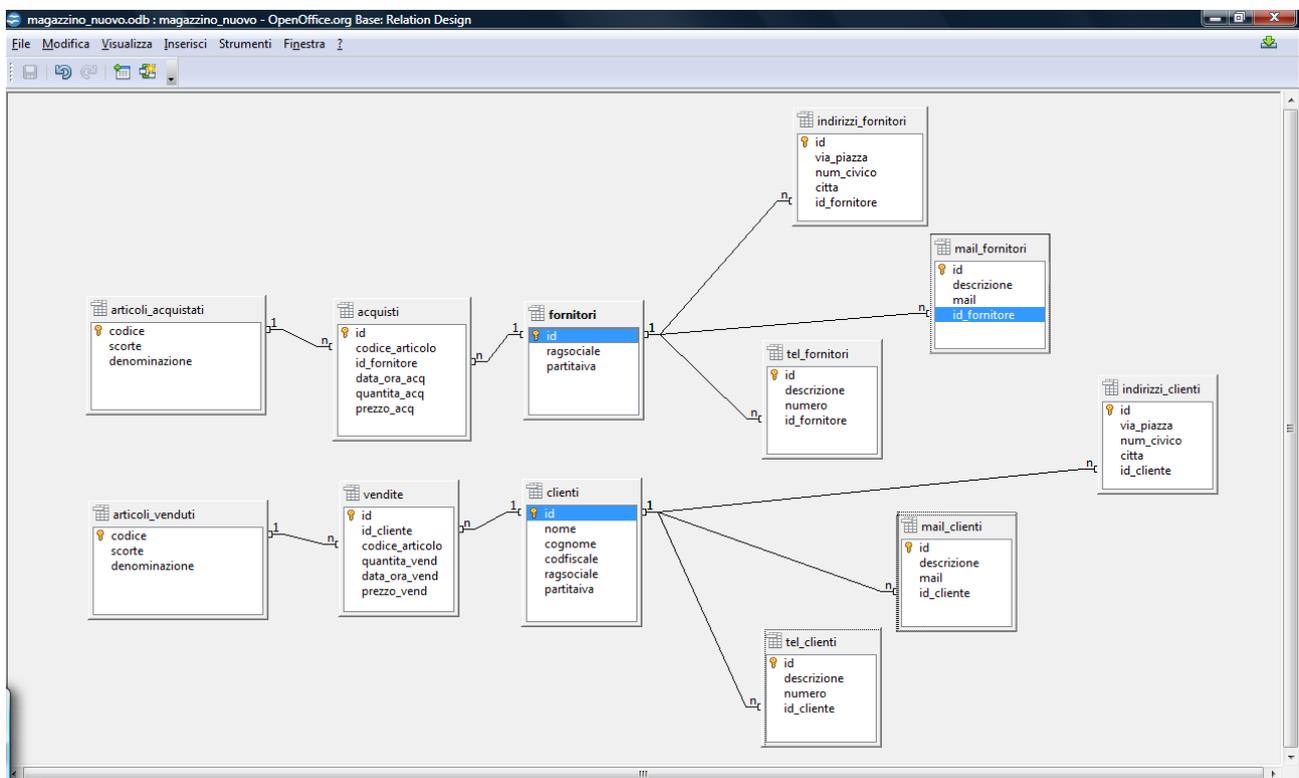
Esempio: GestioneMagazzino

Progetto 1)

Entità: vendite e acquisti, articoli _acquistati e venduti, fornitori e clienti, indirizzi,telefono, email di clienti e fornitori.

Complessità diagramma E/R : elevata (numero entità = 12)

Flessibilità: la possibilità di associare ad ogni cliente/fornitore un numero non definito a priori di recapiti telefonici, email e indirizzi rappresenta sicuramente un vantaggio pagato al prezzo di una maggiore complessità del diagramma E/R e delle interrogazioni e di una minore efficienza di queste ultime rispetto al caso in cui queste informazioni siano specificate direttamente nelle tabelle clienti/fornitori.

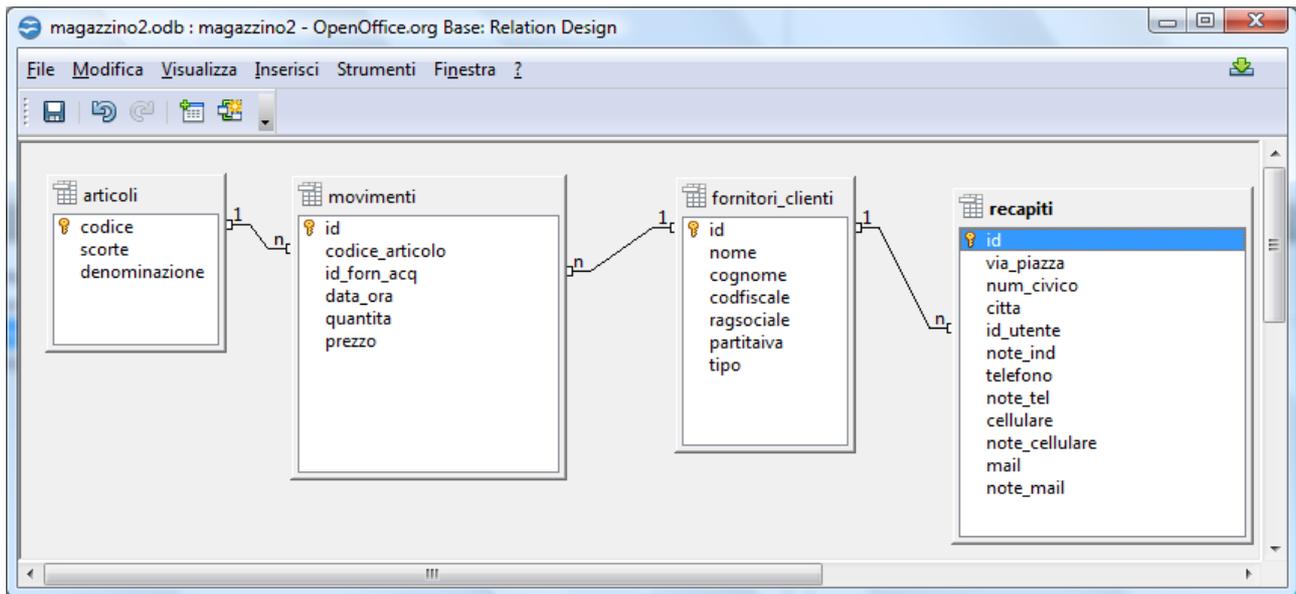


E' possibile ridurre il numero di entità (e dunque di tabelle) presenti, notando che vi sono molte entità che presentano gli stessi attributi. Di seguito vengono analizzati vantaggi e svantaggi di una simile operazione.

Progetto 2)

Tabelle :

- vendite e acquisti(**movimenti**),
- articoli _acquistati e venduti(**articoli**),
- fornitori e clienti(**fornitori_clienti** l'attributo tipo = 'F' per i fornitori 'C' per i clienti),
- indirizzi,telefono e mail di clienti e fornitori(**recapiti**).



Vantaggi: complessità minore

Complessità diagramma E/R e modello logico relazionale: bassa (numero entità = 4 = numero tabelle)

Svantaggi: rallentamento interrogazioni per vendite e acquisti

Efficienza in termini di tempo necessario per effettuare le interrogazioni: la tabella movimenti contiene i dati sia delle vendite che degli acquisti (Se ad es. acquisti contiene 2 ordini e vendite 5, movimenti conterrà 7 records), lo stesso avviene per articoli e movimenti.

Quando si effettua il prodotto cartesiano tra queste 3 tabelle il numero di records sarà molto maggiore rispetto al caso in cui effettuo il prodotto cartesiano tra articoli_acquistati, acquisti e fornitori o tra articoli_venduti, vendite e clienti.

Esempio:

Num. Righe Tabella	Num. Righe Tabella	Num. Righe Tabella	Num. Righe Prodotto Cartesiano
articoli_acquistati = 200	acquisti = 60	fornitori = 15	30*60*15 = 27.000
articoli_venduti = 4	vendite = 200	clienti = 50	4*200*50 = 40.000
articoli = 204	movimenti = 260	fornitori_clienti = 65	204*260*65 = 3.447.600

Come si vede il numero aumenta considerevolmente dalle decine di migliaia passiamo ai milioni di record e questo considerando valori di partenza molto bassi di partenza.

L'aumento del numero di records del prodotto cartesiano non rende solo la ricerca più lenta ma comporta un rallentamento soprattutto per la memoria richiesta per conservarne i risultati. Ovviamente ci sono delle ottimizzazioni che si possono fare attraverso l'utilizzo degli indici, tuttavia è evidente che in generale si avrà un rallentamento delle query.

Tuttavia...

Nel caso della tabella recapiti accorpendo differenti informazioni invece potrei ottenere dei vantaggi in termini di efficienza delle interrogazioni. Infatti se devo recuperare indirizzi, mail e numeri telefono il prodotto cartesiano coinvolgerà un numero inferiore di tabelle.

Supponiamo che vi siano 50 fornitori e 200 clienti, ognuno dei quali associato ad un indirizzo, una email, un numero di telefono fisso e un numero di cellulare.

Per ottenere queste informazioni insieme ai dati dei fornitori dovrei effettuare nel primo caso un'interrogazione simile alla seguente:

```
SELECT fornitori.*,mail, via_piazza, num_civico,citta, telefono FROM fornitori, mail_fornitori, indirizzi_fornitori, tel_fornitori WHERE (mail_fornitori.id_fornitore = fornitori.id) AND (indirizzi_fornitori.id_fornitore = fornitori.id) AND (tel_fornitori.id_fornitore = fornitori.id);
```

nel secondo:

```
SELECT fornitori_clienti.*, recapiti.* FROM fornitori_clienti, recapiti WHERE (recapiti.id_utente = fornitori_clienti.id) AND (fornitori_clienti.tipo = 'F');
```

nel primo caso il prodotto cartesiano tra le tabelle nella clausola FROM mi darebbe :

$$50*50*50*50 = 50^4 = 6.250.000$$

nel secondo:

$$250 * 250 = 250^2 = 62.500$$

Come si può comprendere facilmente sebbene il numero di elementi nella tabella fornitori_clienti possa essere molto più elevato di quello della tabella fornitori, l'esponente è quello che fa crescere più velocemente il numero di righe del prodotto cartesiano.

Per cui la seconda interrogazione sarà più efficiente della prima. Ovviamente potrei rendere più efficiente anche la prima base dati accorpendo i dati relativi ai vari recapiti.

Efficienza in termini di memoria utilizzata per immagazzinare i dati:

Lo spreco di memoria per la tabella recapiti è abbastanza modesto in quanto sebbene alcuni attributi possano rimanere non specificati(ad esempio se si inserisce solo il numero di telefono) questi sono di tipo VARCHAR per cui la memoria allocata per gli attributi non istanziati sarà trascurabile.