Associazioni con attributi

Gli attributi possono essere riferiti sia alle entità sia alle associazioni quando specificano una caratteristica legata alla relazione esistente fra due entità. Per maggior chiarezza analizziamo subito un esempio. Consideriamo le entità FILM e ATTORE: le istanze dell'entità FILM rappresentano tutti i film esistenti nel campione che stiamo analizzando, mentre nell'entità ATTORE sono presenti le informazioni relative agli attori che hanno interpretato i film. Fra le due entità è presente una relazione di tipo M:N (INTERPRETA) perché un film è interpretato da molti attori e un attore può interpretare più film. Vogliamo ora registrare il ruolo che ha avuto un attore in un film (attore protagonista, secondo attore, comparsa ecc.): risulta evidente che l'attributo Ruolo non può essere legato all'entità FILM (perché nel film sono presenti diversi ruoli) e neppure all'entità ATTORE (nella sua carriera l'attore avrà interpretato diversi ruoli). L'attributo Ruolo è un attributo della relazione INTERPRETA perché dipende sia da FILM sia da ATTORE specificando così il ruolo che ha avuto un attore in un determinato film.

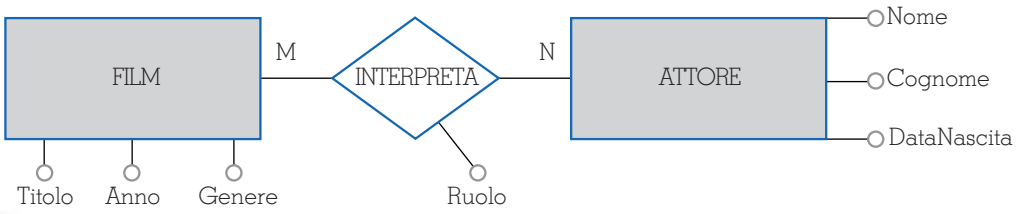


figura 1 Associazione con un attributo

esercizio guidato

Testo Nell'ambito della gestione degli esami universitari considera i dati relativi alle facoltà (Medicina, Informatica, Fisica, Lettere), ai corsi, agli studenti e ai corsi superati dagli studenti mediante una prova d'esame. La base di dati deve essere in grado di fornire l'elenco dei corsi presenti in una facoltà e la stampa delle informazioni (corso, voto e data) relative a tutti gli esami sostenuti da uno studente.

Analisi Le entità che si possono individuare sono:

- FACOLTÀ (per rappresentare le facoltà universitarie) caratterizzata dagli attributi Codice (chiave primaria), Nome, Sede;
- CORSO con attributi Codice (chiave primaria), Titolo, Docente;
- STUDENTE con attributi Matricola (chiave primaria), Nome, Cognome, Data di Nascita.

Tra l'entità FACOLTÀ e l'entità CORSO esiste la relazione uno a molti (APPARTIENE) poiché a una facoltà universitaria possono appartenere corsi ma un corso è relativo a una sola facoltà (figura 2).

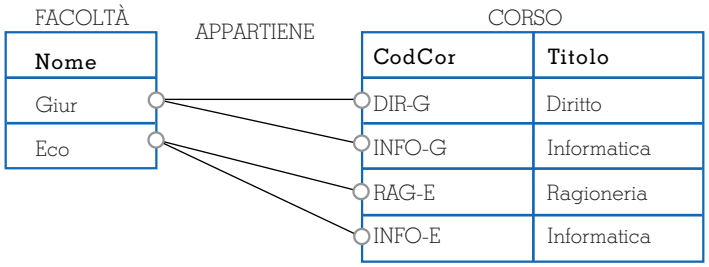


figura 2 Esempio di associazione 1:N FACOLTÀ e CORSO

Tra l'entità CORSO e l'entità STUDENTE esiste la relazione molti a molti (SOSTIENE) poiché per un corso molti studenti sostengono l'esame e uno studente sostiene esami relativi a molti corsi. I dati relativi al voto e alla data di un esame sono attributi della relazione *Sostiene* perché dipendono sia dallo studente che dal corso (figura 3).

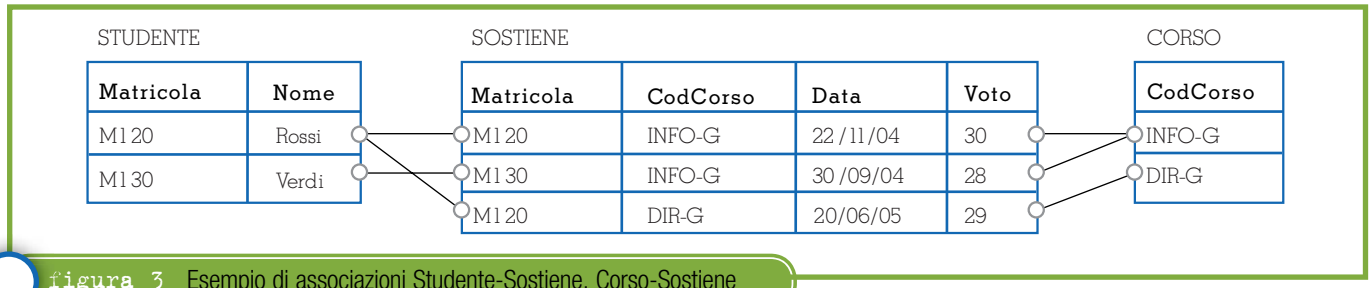


figura 3 Esempio di associazioni Studente-Sostiene, Corso-Sostiene

Lo schema E/R risultate è raffigurato nella figura 4.

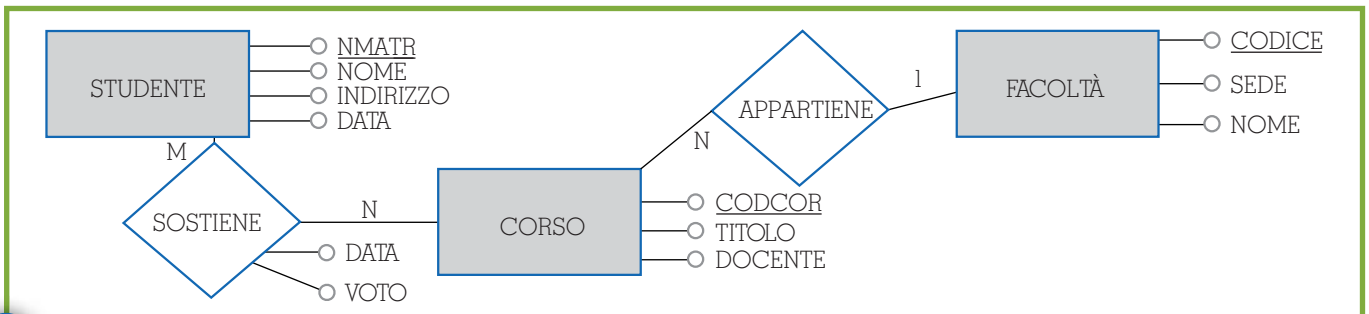


figura 4 Schema E/R per l'esercizio

Si possono riassumere le informazioni presenti nello schema E/R con le seguenti tabelle:

STUDENTE

nome	descrizione	tipo	lunghezza	chiave
Nmatr	numero di matricola dello studente	stringa	8 byte	primaria
Nome	nome e cognome	stringa	40 byte	
Indirizzo	indirizzo	stringa	40 byte	
DataN	data di nascita	data	8 byte	

FACOLTÀ

nome	descrizione	tipo	lunghezza	chiave
Codice	codice identificativo della facoltà	stringa	8 byte	primaria
Sede	indirizzo e città dove ha sede la facoltà	stringa	50 byte	
Nome	denominazione della facoltà	stringa	20 byte	

CORSO

nome	descrizione	tipo	lunghezza	chiave
CodCor	codice identificativo del corso	stringa	6 byte	primaria
Titolo	titolo del corso	stringa	40 byte	
Docente	nome del docente del corso	stringa	20 byte	

L'associazione SOSTIENE ha i seguenti attributi:

nome	descrizione	tipo	lunghezza
Data	data in cui uno studente ha sostenuto	data	8 byte
Voto	voto dell'esame	numerico	2 byte

Schemi e sottoschemi

L'insieme delle entità, delle relazioni e degli attributi individuati per rappresentare quella parte di realtà che interessa tutte le applicazioni determina lo *schema*.
La figura 1 mostra lo schema di una base di dati descritta con il modello E/R. In essa si evidenziano tre tipi di entità: IMPIEGATO, FILIALE e PROGETTO, ciascuna caratterizzata dai propri attributi.
Sono inoltre presenti due associazioni, Lavora e Partecipa, che permettono di realizzare i collegamenti fra le entità. Poiché un impiegato può occuparsi di più progetti e a un progetto possono partecipare più impiegati, la relazione Partecipa esistente fra IMPIEGATO e PROGETTO è di tipo M:N; l'attributo della relazione (Data inizio) permette di conoscere la data di inizio attività degli impiegati rispetto ai diversi progetti.

La relazione Lavora esistente fra FILIALE e IMPIEGATO è di tipo 1:N poiché presso una filiale possono lavorare più impiegati, ma un impiegato lavora presso una sola filiale della ditta. La sua presenza permette di conoscere presso quale filiale lavorano gli impiegati.

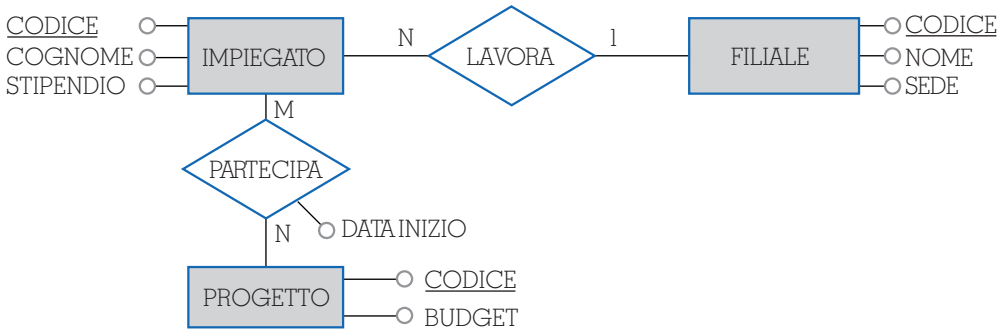


figura 1 Esempio di schema E/R

Per *sottoschema* o *vista* (view) si intende una visione particolare dei dati relativa alla singola applicazione. Il sottoschema deriva dallo schema globale e semplifica la visione dei dati dei singoli utenti.

Se per esempio si vogliono conoscere quali impiegati lavorano in una filiale, si può definire un sottoschema dallo schema di figura 1 che '*veda*' solo le entità IMPIEGATO e FILIALE e la relazione Lavora come illustrato in figura 2.

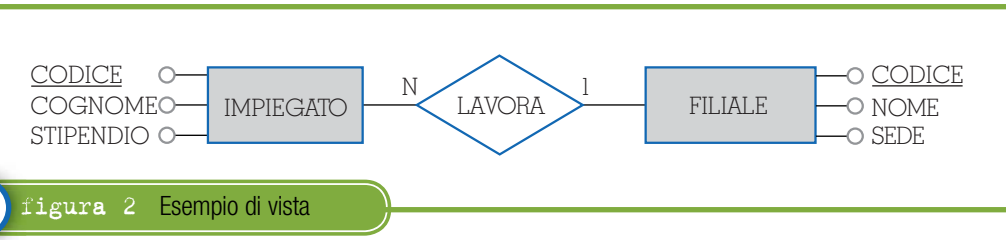


figura 2 Esempio di vista

Su uno schema possono essere definiti diversi sottoschemi, ognuno dei quali descrive la porzione di dati utilizzata dall'utente che è così autorizzato a interagire con la base di dati solo per quei dati sui quali ha la competenza. In questo modo si eliminano errori per modifiche e cancellazioni improprie da parte di utenti non autorizzati e si garantisce che dati *sensibili* come per esempio lo stipendio di un impiegato siano visti solo dagli utenti consentiti.

L'applicazione che si occupa degli stipendi del personale sarà ovviamente interessata solo alla parte dello schema che contiene l'entità IMPIEGATO (figura 3), mentre quella riguardante la partecipazione degli impiegati ai vari progetti utilizzerà le entità IMPIEGATO (della quale non vede l'attributo Stipendio), PROGETTO e la relazione Partecipa come illustrato in figura 4.

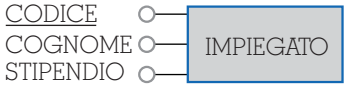


figura 3 Esempio di sottoschema

figura 4 Esempio di sottoschema

Gli esempi visti finora sono molto semplici perché il sottoschema consiste in sostanza di un sottoinsieme dello schema di partenza. In realtà le possibilità offerte da alcuni sistemi per la definizione di sottoschemi sono molto più vaste. È possibile, per esempio, definire nuovi tipi di entità utilizzando attributi delle entità esistenti: dallo schema di figura 1 è possibile creare il sottoschema di figura 5 nel quale è definita la nuova entità SEDI, contenente le informazioni degli impiegati e della filiale in cui lavorano i dipendenti.

figura 5 Esempio di sottoschema

Grazie al fatto che ogni applicazione è legata, solitamente, non a tutto lo schema ma a un particolare sottoschema o vista, si può parlare di indipendenza logica dei dati che è, come abbiamo visto, la caratteristica del DBMS che permette di modificare una parte dello schema logico dei dati, senza che, per questo, si debbano rivedere le applicazioni che non fanno riferimento alla parte dello schema modificato.
Se si rende necessario modificare lo schema come per esempio aggiungere nuovi attributi a un'entità esistente, non tutti i sottoschemi sono influenzati dal cambiamento. Nel caso esaminato, volendo aggiungere l'attributo Data_Assunzione all'entità IMPIEGATI, le applicazioni che lavorano con i sottoschemi delle figura 4 e figura 5 non sono influenzate dal cambiamento. Ragionevolmente si può ipotizzare che la data di assunzione sia elaborata dall'applicazione che calcola gli stipendi degli impiegati: in questo caso il sottoschema di figura 3 dovrà essere modificato e di conseguenza anche i programmi applicativi che interfacciano quel sottoschema.

verifica le tue conoscenze

1 Che cos'è contenuto nello schema?
2 Che cos'è contenuto nel sottoschema?
3 Perché è utile lavorare sul sottoschema?
4 Perché i sottoschemi realizzano l'indipendenza logica?

Il modello relazionale

Le tabelle

Il modello relazionale è stato proposto nel 1970 dal matematico inglese Edgar Codd e grazie alla sua coerenza e usabilità è diventato dagli anni Ottanta quello più utilizzato per la produzione di DBMS.

La struttura fondamentale del modello relazionale è la "relazione", cioè una tabella bidimensionale costituita da righe e colonne. Le tabelle rappresentano le entità che si ritengono essere interessanti nel database: gli attributi delle entità rappresentano i *campi* delle relazioni, cioè le colonne delle tabelle, mentre per *ennuple* (N-ple) o *tuple* si intendono le occorrenze delle relazioni (cioè le righe, o *record*, delle tabelle). Il modello relazionale si basa sulla seguente definizione matematica di relazione: *dati due insiemi D_1 e D_2, non necessariamente disgiunti, una relazione R definita su D_1 e D_2 è un qualunque sottoinsieme del prodotto cartesiano D_1 D_2*. Chiariamo meglio con un esempio: siano D_1 = {Giovanni, Mario, Anna} e D_2 = {10, 25, 18}. Il prodotto cartesiano D_1 D_2 è formato dalle coppie {(Giovanni,10), (Giovanni, 25), (Giovanni, 18), (Mario, 10), (Mario, 25), (Mario, 18), (Anna, 10), (Anna, 25), (Anna, 18)}. Una relazione è un sottoinsieme del prodotto cartesiano e pertanto, dal prodotto cartesiano appena calcolato, estraiamo solo le coppie per noi significative che rappresentano, per esempio, l'età delle persone. Definiamo quindi la relazione PERSONE formata dalle coppie {(Giovanni, 10), (Mario, 25), (Anna, 18)} che si rappresenta in tabella come illustrato in figura 1.

Potrebbe nascere confusione tra relazione intesa come "insieme di dati nel modello relazionale" e relazione intesa come "associazione tra entità". In questa Unità useremo il termine tabella invece del termine relazione, ricordando che sono sinonimi, così come lo sono i termini "campo" e "attributo", e i termini "record", "occorrenza" ed "ennupla". In figura 2 sono mostrati altri esempi di tabelle relazionali.

PERSONE

Giovanni	10
Mario	25
Anna	18

figura 1 Rappresentazione tabellare della relazione Persone

MOVIMENTI

cognome	indirizzo	città	n. telefono
Rossi	Via Roma 4	Torino	011 3572943
Bianchi	Via Torino 12	Milano	02 4372049
Verdi	Via Genova 12	Torino	011 6993524

LOCALITÀ

città	regione
Torino	Piemonte
Milano	Lombardia
Roma	Lazio

figura 2 Tabelle relazionali

L'identificazione dei record

Chiavi

Per identificare i record all'interno della tabella si fa uso di campi chiave o di identificatori. Un *identificatore* è un insieme di uno o più attributi che identifica univocamente una data occorrenza. Al posto di identificatore si parla a volte di *chiave candidata* e in una tabella ne possono esistere più di una (per esempio nella tabella PERSONA possono esistere due identificatori, il Codice fiscale e l'insieme di Nome, Cognome e Data-Nascita). Si osservi che almeno una chiave candidata esiste sempre poiché al limite può essere costituita dall'insieme di tutti gli attributi (non possono infatti esistere record uguali, cioè le righe della tabella devono essere tutte diverse).

Tra tutte le chiavi candidate (gli identificatori) viene scelta una *chiave primaria* per la memorizzazione e per effettuare i collegamenti con le altre relazioni. Normalmente ne viene scelta una tra quelle con il minor numero di attributi. Gli attributi della chiave primaria non possono assumere il valore null (che significa un valore non determinato), in quanto non permetterebbero più di identificare una particolare tupla in una relazione. Una *chiave esterna* è un attributo (o insieme di attributi) che non è chiave nella tabella, ma lo è in un'altra e serve per realizzare il collegamento logico tra le entità.

Dominio

Due tabelle possono essere messe in relazione solo attraverso l'uguaglianza dei valori di un certo campo. Due valori sono confrontabili solo se appartengono allo stesso dominio.
Per *dominio* si intende l'insieme dei valori su cui è definito un certo attributo, cioè l'insieme di tutti i valori elementari che un attributo può assumere (per esempio il campo Età potrà essere definito sul dominio degli interi che va da 0 a 150).
Il concetto di dominio è fondamentale perché sono proprio gli attributi di diverse relazioni definite sullo stesso dominio che permettono di realizzare le associazioni tra i diversi tipi di entità. In particolare questo è possibile tramite l'uguaglianza tra chiave primaria di una tabella e chiave esterna di un'altra.
Nell'esempio della figura 3 si vede come la tabella LOCALITÀ sia collegata alla tabella Amici poiché in entrambe esiste l'attributo Città, chiave primaria nella tabella LOCALITÀ e chiave straniera nella tabella AMICI (figura 3).
Se il numero di attributi di una chiave straniera (o meglio la sua dimensione) è elevato, spesso è conveniente introdurre un nuovo campo nella tabella di partenza (per esempio un campo Codice) facendolo diventare la nuova chiave.

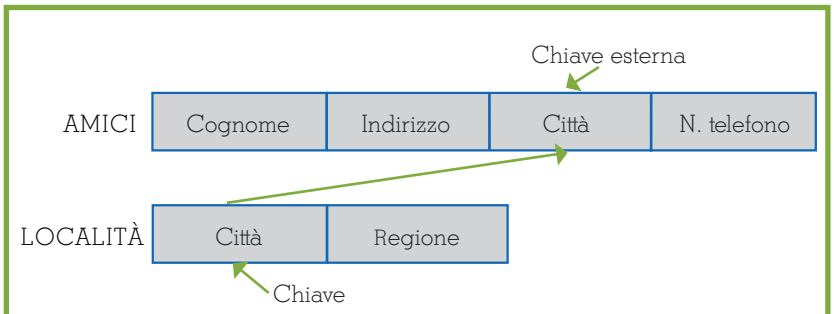


figura 3 Chiave primaria e chiave esterna

Altre caratteristiche

Altre caratteristiche risultano utili per definire le metriche (ossia esprimere le misure quantitative) sulle relazioni presenti in un database: per *grado* di una relazione si intende il numero di campi delle tabelle, mentre per *cardinalità* si intende il numero delle occorrenze presenti in un dato istante. Nelle tabelle non conta l'ordinamento delle N-ple (record), ma le N-ple non si possono ripetere e sono identificate univocamente dalla chiave. Nello schema della figura 4 sono riassunti e illustrati i concetti appena espressi sull'esempio della tabella AMICI.

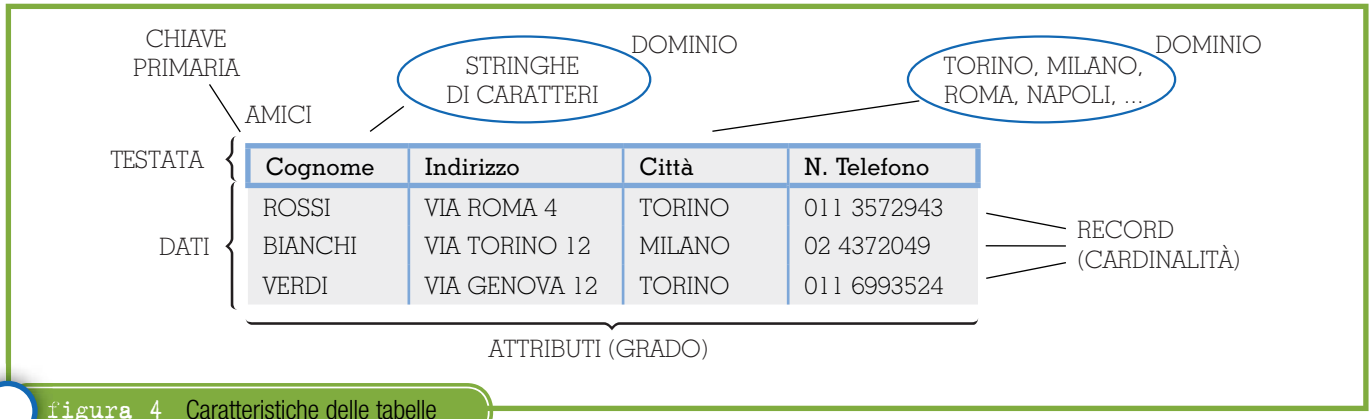


figura 4 Caratteristiche delle tabelle

verifica le tue conoscenze

1 Che cosa è una tabella?
2 Da dove deriva il termine modello relazionale?
3 Come si definisce una chiave primaria?
4 Che cosa indica il dominio di un attributo?
5 Che cosa indica il grado di una tabella? E la cardinalità?

Dallo schema E/R allo schema logico relazionale

Dallo schema concettuale dei dati (schema E/R) si ottiene lo schema logico dei dati, applicando alcune semplici regole di derivazione sulle entità e sulle relazioni individuate nello schema E/R. A differenza dello schema concettuale, quello logico dipende strettamente dal modello di rappresentazione dei dati utilizzato dal DBMS (gerarchico, reticolare o relazionale). In seguito si farà riferimento allo schema logico utilizzato nei DBMS relazionali, denominato schema logico relazionale.

La rappresentazione delle entità

Ogni entità dello schema E/R rappresenta una tabella, ogni istanza di entità rappresenta un record della tabella e ogni attributo diventa un campo del record. La chiave dell'entità identifica la chiave d'accesso per la tabella.
All'entità PRODOTTO, rappresentata in figura 1a, corrisponde la tabella relazionale di figura 1b:

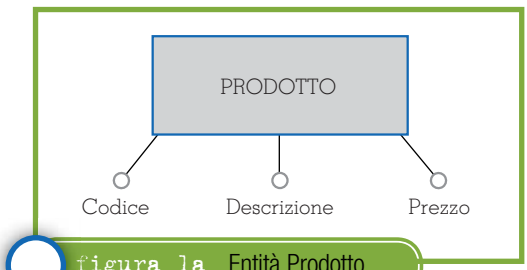


figura 1a Entità Prodotto

PRODOTTO

Codice	Descrizione	Prezzo

figura 1b Tabella relazionale

Per descrivere l'entità in forma più concisa si può scriverne il nome seguito dal nome dei suoi attributi racchiusi tra parentesi; l'attributo chiave è sottolineato. L'entità PRODOTTO viene in questo caso rappresentata così:

PRODOTTO (Codice, Descrizione, Prezzo)

PRODOTTO

nome campo	descrizione	tipo	lunghezza	chiave
Codice	codice del prodotto	stringa	3 byte	primaria
Descrizione	descrizione del prodotto	stringa	30 byte	
Prezzo	prezzo del prodotto	numerico	6 byte	

figura 2 Descrizione della tabella PRODOTTO

La rappresentazione dettagliata dell'entità spesso avviene mediante una tabella descrittiva nella quale sono evidenziati gli attributi e le loro caratteristiche (nome, descrizione, tipo, dimensione), come illustrato in figura 2.

La rappresentazione delle associazioni

Risulta essere più complessa la traduzione dallo schema concettuale a quello logico delle associazioni (si dice anche *risolvere l'associazione*), in cui si effettueranno scelte diverse in base al tipo di associazione e al previsto uso che se ne potrà fare.
Le associazioni uno a uno (1:1) si risolvono facendo migrare la chiave di una delle due tabelle create nell'altra, in base alle modalità con cui si pensa verranno utilizzate.

Se per esempio (figura 3) vi sono le entità MARITO e MOGLIE collegate tra loro, la loro traduzione porterà a creare due tabelle: MOGLIE e MARITO. Nella traduzione dell'associazione, la scelta potrà essere di far migrare il codice fiscale della MOGLIE nella tabella MARITO se le richieste di informazioni sulla moglie sono subordinate a quelle sul marito (cioè prima si accede alla tabella MARITO e poi alla tabella MOGLIE, partendo dal suo codice fiscale presente in MARITO). Ovviamente, se l'accesso ai dati del marito avviene sempre dopo l'accesso ai dati della moglie, si effettuerà la migrazione del codice fiscale del marito. Se invece non vi sono prevalenze nell'accesso, è possibile far migrare la chiave di ciascuna tabella nell'altra (figura 4).

MOGLIE (C_F, Cognome, Nome)
MARITO (C_F, Cognome, Nome, C_FMoglie)

figura 3 Accesso alla MOGLIE in base al codice del MARITO

A) Accesso al MARITO in base al codice della MOGLIE
MOGLIE(C_F, Cognome, Nome, C_FMarito)
MARITO(C_F, Cognome, Nome)

B) Possibilità di accesso da entrambe le relazioni
MARITO(C_F, Cognome, Nome, C_FMoglie)
MOGLIE(C_F, Cognome, Nome, C_FMarito)

figura 4 Altre possibili traduzioni dell'esempio

Nelle associazioni uno a molti la chiave primaria della tabella di partenza dell'associazione (quella con caratteristica 1) diventa chiave esterna (*foreign key*) dell'entità di arrivo associata, cioè l'attributo che è identificatore univoco diventa un campo nella tabella dell'entità di arrivo. Nell'esempio di figura 5, poiché una persona può possedere più auto, ma un'auto può essere posseduta da una sola persona, si farà migrare la chiave di PERSONA (in questo caso il codice fiscale) nella tabella AUTO. Questa diventerà chiave esterna permettendo così il collegamento logico tra AUTO e PERSONA (figura 5).

Schema E/R

Auto(Targa, Modello,Colore,C_F) Persona(C_F,Cognome,Indirizzo)

AUTO

nome campo	descrizione	tipo	lunghezza	chiave
Targa	targa dell'automobile	stringa	8 byte	primaria
Modello	modello dell'automobile	stringa	30 byte	
Colore	colore dell'automobile	stringa	10 byte	
C_F	codice fiscale del proprietario	stringa	16 byte	esterna

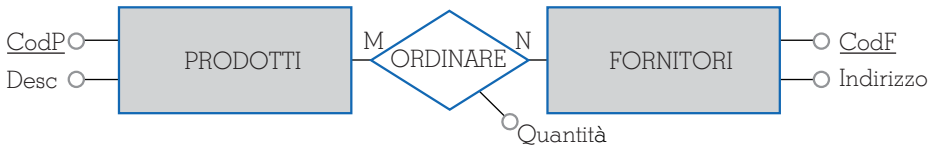
PERSONA

nome campo	descrizione	tipo	lunghezza	chiave
C_F	codice fiscale della persona	stringa	16 byte	primaria
Cognome	cognome della persona	stringa	15 byte	
Indirizzo	indirizzo della persona	stringa	30 byte	

figura 5 Soluzione delle associazioni 1:N

Dal modello concettuale vengono derivate le tabelle che rappresentano le entità e l'associazione uno a molti viene tradotta aggiungendo agli attributi dell'entità "a molti" la chiave dell'entità "a uno".

Le associazioni molti a molti (M:N) si risolvono creando una nuova tabella (in aggiunta alle tabelle derivate dalle due entità) in cui vengono inserite le chiavi delle due entità e gli eventuali attributi dell'associazione (figura 6).
Nell'esempio riportato, poiché un fornitore può fornire più prodotti e un prodotto può essere fornito da più fornitori, si crea una nuova tabella (ORDINARE) composta dalle chiavi primarie delle tabelle di partenza (CodF e CodP) e dall'attributo Quantità.
Si nota come né CodF né CodP possano essere chiave primaria della relazione ORDINARE, perché non individuerebbero in modo univoco una riga della tabella. In questo caso la chiave primaria è data dalla concatenazione delle due chiavi straniere.
Ogni record della tabella ORDINARE indica quindi la quantità di un certo prodotto venduta da un determinato fornitore.



Schema E/R

PRODOTTI

nome campo	descrizione	tipo	lunghezza	chiave
CodP	codice del prodotto	stringa	3 byte	primaria
Desc	descrizione del prodotto	stringa	30 byte	

FORNITORI

nome campo	descrizione	tipo	lunghezza	chiave
CodF	codice del fornitore	stringa	5 byte	primaria
Indirizzo	indirizzo del fornitore	stringa	50 byte	

ORDINARE

nome campo	descrizione	tipo	lunghezza	chiave
CodP	codice del prodotto	stringa	3 byte	esterna
CodF	codice del fornitore	stringa	5 byte	esterna
Quantità	quantità ordinata	intero	4 byte	

figura 6 Schema e soluzione delle associazioni M:N

Dal modello concettuale vengono derivate le tabelle che rappresentano le entità e l'associazione molti a molti viene tradotta introducendo una terza tabella contenente le chiavi delle due entità e gli eventuali attributi dell'associazione.

Conclusione

Dopo avere analizzato le regole che consentono il passaggio dallo schema E/R allo schema logico possiamo riassumere, ricordando che la progettazione delle basi dati si articola nelle seguenti fasi operative:

- progettazione concettuale;
- progettazione logica;
- progettazione fisica.