

# Asymetrix InfoModeler

di Francesco Petroni

**N**umerose sono le motivazioni che ci hanno indotto a presentarvi la prova dell'Asymetrix InfoModeler.

La prima è che si tratta di un prodotto che ricade in una categoria, quella dei CASE, poco sviluppata per il mondo dei PC, in pratica quindi una novità. E di cosa sia un CASE dovremo parlare.

La seconda è che si tratta di un prodotto che serve di supporto ai vari prodotti di tipo DBMS oggi operanti sotto Windows: Fox Pro e Access della Microsoft, Paradox della Borland, Oracle7, SQLServer e un SQL standard, usabile con parecchi altri prodotti software.

La categoria DBMS per Windows è una categoria emergente, che interessa gli utenti finali sia i professionisti

del software, che, dal loro canto, sono abituati ad approcciare i problemi con strumenti CASE, e che quindi sono i naturali destinatari di questa tipologia di prodotti.

La terza motivazione consiste nel fatto che InfoModeler è un prodotto della Asymetrix, casa che ha sempre dato alle sue scelte un carattere di innovazione se non addirittura di sperimentazione.

Ricordiamo della Asymetrix il famoso ToolBook (MC n.106), uno dei più diffusi prodotti di authoring multimediale, e l'eccellente Compel (MC n.131), prodotto di categoria Presentation, molto evoluto e particolarmente «dotato» in termini di strumentazione Multimediale. Ma precisiamo subito che

InfoModeler non ha nulla a che vedere con la multimedialità.

Il problema principale nel presentare InfoModeler sarà quello di far capire il suo ambito di applicabilità, poco noto o addirittura sconosciuto a buona parte degli utilizzatori, non professionisti, di prodotti DBMS.

## A che cosa serve InfoModeler

Lo sviluppo di una applicazione gestionale complessa comporta l'utilizzo di un prodotto di tipo DBMS. Con un prodotto di tipo DBMS si costruisce innanzitutto la struttura del Database, che è costituito da un certo numero di tabelle e di relazioni tra di esse. Ogni tabella contiene propri campi, alcuni dei

**InfoModeler**

quali rivestono una particolare importanza (campi chiave) in quanto è su di questi che si appoggiano le relazioni e quindi i collegamenti tra le tabelle.

Se il DB è stato ben costruito può essere visto dal prodotto come un unico oggetto facilmente gestibile, attraverso maschere di acquisizione/modifica dati, e facilmente manipolabile con procedure elaborative realizzate con il linguaggio interno al DBMS.

Tutti i prodotti DBMS più recenti posseggono funzionalità:

- per definire le strutture delle tabelle e le relazioni tra di esse,
- per realizzare, senza programmare, maschere per la gestione dei dati,
- per disegnare, senza programmare, report, anche di tipo complesso, immediatamente stampabili,
- per manipolare i dati, ad esempio aggiornamenti batch, oppure selezioni e/o ordinamenti di dati, oppure calcoli di aggregazione statistica, anche in questo caso senza programmare.

In caso non si riesca con i normali strumenti operativi ad ottenere quello che si vuole si può ricorrere al linguaggio di programmazione sempre presente in un DBMS.

I prodotti che oggi offre il mercato sono ottimi, permettono di creare in poco tempo applicazioni anche di tipo complesso. Questa fortunata realtà non deve però far perdere di vista il requisito fondamentale che sta alla base del successo di un'applicazione DBMS. Un'applicazione DBMS non può funzionare bene se non è stato ben progettato il Database.

Il progetto del Database è un'attività preventiva rispetto a quella che si fa con il PC e che si deve concludere con la formalizzazione della struttura del Database e quindi con la decisione di quali Tabelle questo debba prevedere e quali Relazioni. Se il progetto è sbagliato, ad esempio non si è prevista una Tabella o non si è capito il funzionamento di una Relazione, l'applicazione, per quanto si possa essere bravi ad usare il prodotto DBMS, non funzionerà bene, il che, per un'applicazione gestionale, vuol dire che non serve a nulla.

Come fare quindi ad ottimizzare questa fase criticissima che riguarda, è bene dirlo subito, numerosi aspetti, espressivi, logici, formali, ecc. che nulla hanno, ancora, a che vedere non solo con un DBMS, ma neanche con un PC?

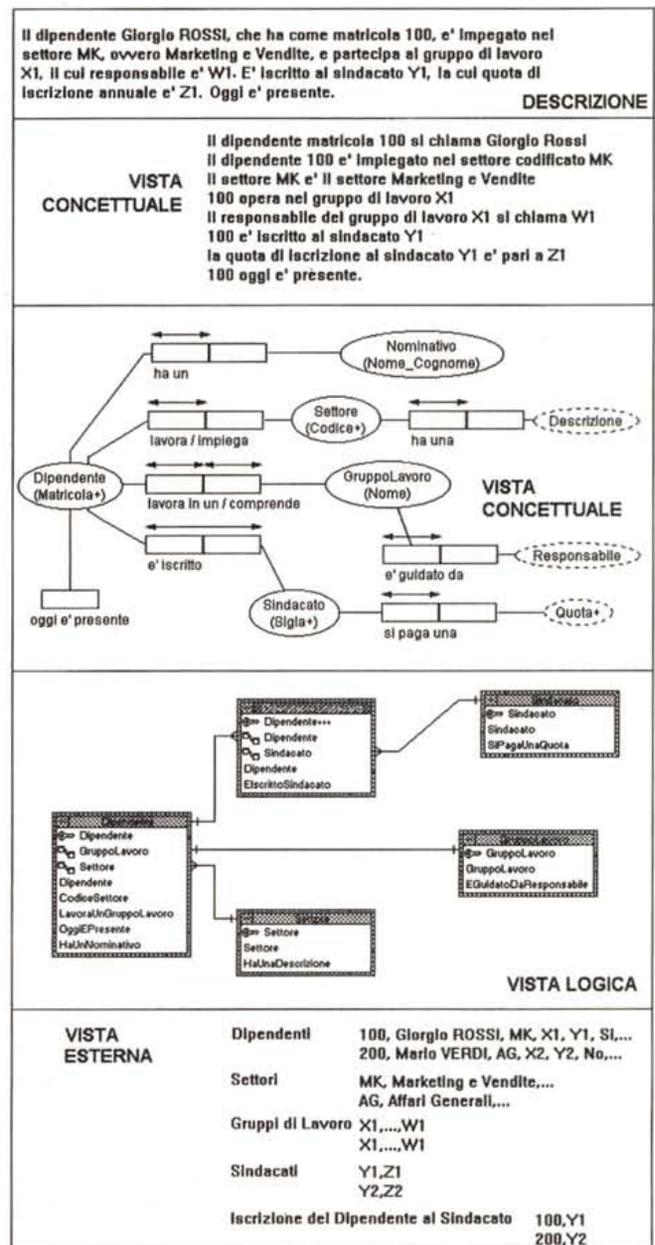
Ebbene i prodotti CASE, e il nostro InfoModeler, hanno come obiettivo quello di occupare questa «zona insidiosa» nella quale si compie quel delicatissimo passaggio da una vista esterna del problema alla sua esatta definizione tecnica.

**Produttore:**  
Asymetrix Europe  
Arnulfstrasse, 27  
D/8000 M2 Germany

**Distributori:**  
Mado S.r.l.  
Via Masaccio, 11 - 42100 Reggio Emilia  
Tel.: 0522/512828  
J. Soft S.r.l. - Via Cassanese, 224  
Palazzo Tintoretto - 20090 Centro Direz. Milano  
Oltre Segrate (MI) - Tel.: 02/26920700  
Ingram Micro Italia - Via Roma, 74  
20060 Cassina de' Pecchi (MI) - Tel.: 02/95343604

**Prezzi indicativi (IVA esclusa):**  
Asymetrix InfoModeler  
per Access Paradox FoxPro  
per SQL Server Oracle 7  
Lit. 1.694.000  
Lit. 3.182.000

Figura 1 - Asymetrix InfoModeler - Un esempio passo passo. Le varie viste.  
Per dare maggior concretezza alla nostra prova eseguiamo, alla fine, un piccolo esercizio pratico abbastanza semplice, in termini di numero di elementi in gioco, ma significativo in termini di aspetti relazionali. Il tema è descritto nel testo. Qui vediamo in un collage i passaggi dalla vista «umana» (una frase complessa detta in un certo modo) alla vista Concettuale (più frasi elementari, disegnabili con InfoModeler) alla vista Logica (in pratica la vista delle Tabelle) e infine alla vista esterna (quella che fa vedere i Dati).



«linguaggio» di comunicazione tra i vari tecnici (immaginate un team che si occupa di sviluppare un DB complessissimo) e inoltre permette al prodotto, ci riferiamo già ad InfoModeler, di costruire automaticamente la struttura del database fisico.

La necessità di un formalismo, che permetta in pratica di tradurre i concetti in disegni, esiste in tutte le attività umane che abbiano un minimo di complessità e che riguardino l'intervento di più individui. Si pensi alla progettazione edile, oppure a quella elettrica o a quella elettronica. Un progetto edile si concretizza in una serie di disegni, nei quali, attraverso un formalismo universalmente riconosciuto, si individuano (e soprattutto li individuano progettista ed esecutore) gli elementi in gioco.

**La varie viste di un problema**

È interessante classificare le varie possibili viste di un problema che si vuole affrontare e risolvere realizzando un Database (fig.1).

La vista NATURALE è quella DESCRIZIONE del problema capita da tutti, ed è quella che in genere fornisce il «committente» al tecnico «analista» incaricato di analizzare il problema e di sviluppare l'applicazione. L'utente sa che gli serve un'applicazione ma si esprime con normali frasi in italiano. Ad esempio:

– il dipendente Giorgio ROSSI, che ha come matricola 100, è impegnato nel settore MK, ovvero Marketing e Vendite, e partecipa al gruppo di lavoro X1, il cui responsabile è W1. È iscritto al sindacato Y1, la cui quota di iscrizione annuale è Z1. Oggi è presente.

La vista CONCETTUALE è quella che scompone in frasi semplici, e ben attribuibili a precise categorie di frasi standard, il concetto espresso prima:

- il dipendente matricola 100 si chiama Giorgio Rossi,
- il dipendente 100 è impiegato nel settore codificato con MK,
- il settore MK è il settore Marketing e Vendite,
- 100 opera nel gruppo di lavoro X1,

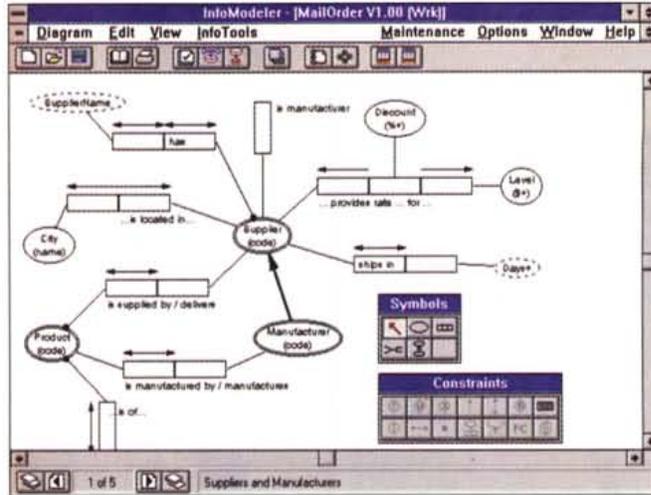
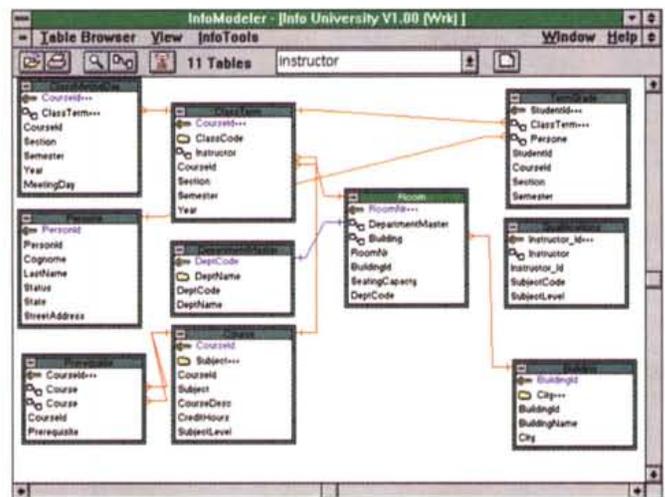


Figura 3 - Asymetrix InfoModeler - Ambiente in Vista Table Browser. Questa vista è prodotta automaticamente come output della vista precedente, quando quest'ultima abbia superato le varie fasi di Validazione. Mostra le varie Tabelle dell'Applicazione, i Campi di ciascuna Tabella, con le varie simbologie che indicano quelli chiave, quelli indice, che indicano i legami (relazionali) tra le tabelle, ecc. Nella lista in alto è possibile decidere quali tabelle vedere e quali no. Inoltre sono possibili altre varianti per indicare il dettaglio di cosa vedere e cosa no.



- il responsabile del gruppo di lavoro X1 si chiama W1,
- 100 è iscritto al sindacato Y1,
- la quota di iscrizione al sindacato Y1 è pari a Z1,
- la matricola 100 oggi è presente.

Una persona particolarmente sensibile potrebbe già individuare la necessità di costruire quattro tabelle, quella dei Dipendenti, quella dei Settori Aziendali, quella dei Gruppi di Lavoro e quella dei Sindacati. Potrebbe individuare i campi delle varie tabelle e i campi di collegamento tra le varie tabelle.

In realtà il nostro problema presenta un'insidia, che... porta le tabelle a cinque, questo nel caso che un dipendente si possa iscrivere a più sindacati (relazione Molti a Molti). Ma lo vedremo dopo.

Ogni frase può essere scomposta nei suoi componenti (soggetto, predicato e oggetto) e ogni componente può essere definito con maggior dettaglio. Ad ogni componente si può associare un Simbolo grafico.

È questo il momento in cui entra in gioco il formalismo grafico di InfoModeler che vede le frasi come FACTS e i componenti come OBJECTS, PREDICATES, RULES e CONSTRAINTS.

La vista CONCETTUALE diventa quindi un disegno, che assomiglia vagamente ad un diagramma di flusso, con pochi tipi di simboli e pochi tipi di linee. Se il problema è complesso il relativo disegno sul video può occupare più pagine (in InfoModeler scorribili con specifici pulsanti (fig.2)).

La successiva vista LOGICA, che viene prodotta automaticamente da

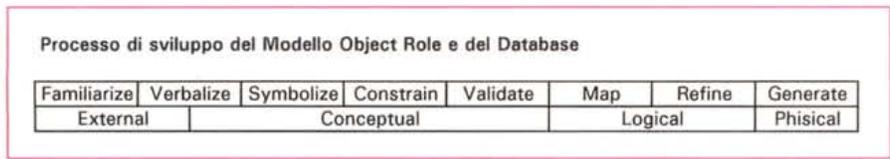


Figura 4 - Processo di sviluppo del Modello FORML. Il ciclo produttivo del modello ORM, dalla descrizione del problema fino alla costruzione del Database, si può scomporre in otto passi logici, definibili con parole abbastanza... poco tecniche. I passi logici possono essere ricomposti in quattro fasi operative.

Figura 2 - Asymetrix InfoModeler - Ambiente in Vista Diagrammer. Questo è il foglio di disegno in cui, prelevando gli elementi disponibili nelle due palette, quella con i Simboli e quella con i Vincoli, si realizza la vista grafica dell'applicazione. Anche se si tratta in pratica di disegnare, l'utilizzatore deve conoscere esattamente il significato concettuale dei vari simboli.

Figura 5 - Asymetrix InfoModeler - Anatomia di una frase.

La materia prima di un modello sviluppato con InfoModeler sono le frasi, frasi comuni scritte in lingua inglese, o anche in lingua italiana, in cui si possano individuare un soggetto, un predicato, un oggetto. Ogni frase che ognuno potrebbe dire a voce nel mondo reale va espressa in maniera simbolica usando una piccolo assortimento di elementi grafici che indicano il tipo di ciascun elemento coinvolto e le sue caratteristiche.

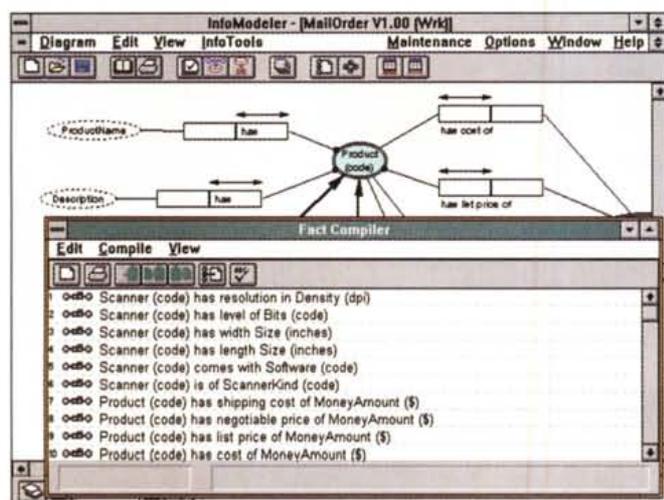
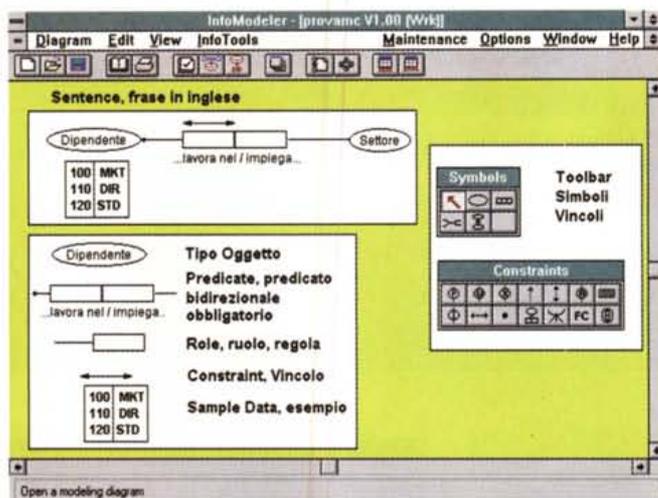


Figura 6 - Asymetrix InfoModeler - Fact Compiler.

Il problema da analizzare deve essere scomposto in frasi semplici. Ogni frase viene formalizzata in un FACT e simboleggiata usando gli oggetti grafici proposti dalle due Palette. Oltre alla vista Diagrammer, che è quella principale dal punto di vista operativo, c'è questa Fact Compiler che mostra l'elenco dei Fact dichiarati. Il Fact si può scrivere nella tabella Fact Compiler e trascinare sul foglio con il disegno.

InfoModeler, è quella che presenta i vari oggetti coinvolti nella frase lunga, poi scomposta in frasi corte, tramutati in simboli grafici, già organizzati in Tabelle collegate tra di loro. In figura 3 ne vediamo una particolarmente complessa, non riferita al nostro esempio.

Infine la vista ESTERNA, quella che mostra i dati già organizzati nelle tabelle. Es. Dipendenti

100, Giorgio ROSSI, MK, X1, Y1, Si,...

200, Mario VERDI,...

Settori  
MK, Marketing e Vendite,...

AG, Affari Generali,...

Gruppi di Lavoro

X1, ..., W1

X2, ..., W2

Sindacati

Y1, Z1

Y2, Z2

e poi come vedremo:

Iscrizione del Dipendente al Sindaca-

to

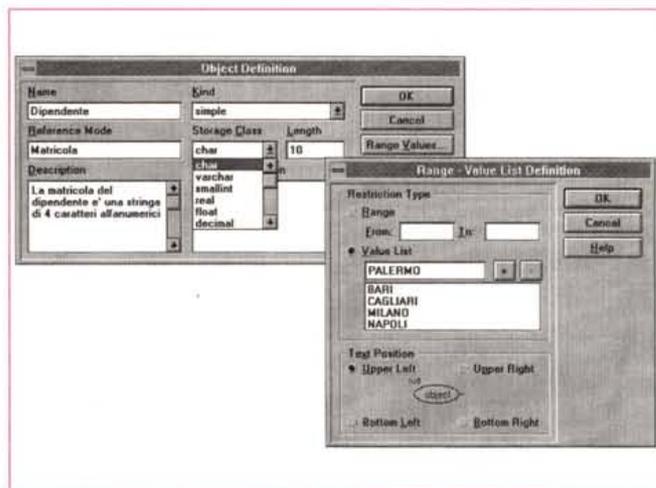
100,X1

200,X2

In definitiva InfoModeler accetta come Input semplici frasi standard espresse nel suo linguaggio grafico formale (vista FORMALE), esegue la validazione sintattica delle frasi, nelle quali deve riconoscere gli oggetti e la corret-

Figura 7 - Asymetrix InfoModeler - Specifiche dell'oggetto.

Un classico doppio click e si entra nella box che serve per la definizione fine del tipo di oggetto. Alcune definizioni servono per specificare le caratteristiche logiche dell'oggetto, ad esempio Kind, con le quali si indica se l'oggetto è un'Entità o un Valore, poi altre sue caratteristiche più descrittive, come la Storage Class, che serve poi per la costruzione delle tabelle del prodotto DBMS. Si arriva fino alla definizione di campi di tipo Lista.



tezza delle varie specifiche, costruisce le viste logiche, e infine costruisce la struttura del Database, nel formato scelto dall'utente.

Nelle pagine dei manuali che descrivono tale processo viene sempre riprodotto uno schema, che evidenzia i vari passaggi, molto chiaro che potete vedere in figura 4.

### Vantaggi derivanti dall'uso di un linguaggio formale

Asymetrix InfoModeler adotta il linguaggio FORML (Formal Object Role Modeling Language) creato per formalizzare graficamente modelli «object-role» ORM.

Il formalismo FORML costituisce una modalità di descrizione dei dati e delle funzionalità del database che si basa sull'uso del linguaggio comune.

Il formalismo più noto è comunque l'Entity-Relationship (in sigla ER) che potremo definire più tecnico in quanto parte direttamente con un formalismo grafico, al contrario dell'ORM, che invece parte più a monte direttamente dalla descrizione in lingua del problema.

Nei manuali viene opportunamente proposto un confronto tra i due linguaggi.

FORML può essere utilizzato dai membri di un team di tecnici incaricato di analizzare e di sviluppare un Database di tipo complesso.

Trattandosi di un formalismo grafico può essere usato anche indipendentemente dal PC, su fogli di carta a quadretti per intenderci. Usare un prodotto software come InfoModeler presenta, rispetto al semplice uso «manuale» della simbologia FORML, diversi grossi vantaggi:

- la correttezza formale degli elementi

inseriti nel grafico, in quanto è possibile usare solo elementi presi dalle palette, - il controllo della sintassi, in quanto per ogni elemento vanno definite, in apposite Dialog Box, una serie di specifiche, proprie di quel tipo di elemento, che vengono controllate e validate,

- la produzione automatica degli schemi grafici e della documentazione testuale,

- dulcis in fundo, la produzione automatica della struttura del Database.

Un altro vantaggio, particolare dell'Asymetrix InfoModeler, è la gestione delle versioni del progetto, che vengono memorizzate nei vari file prodotti da InfoModeler.

**Il materiale**

È evidente che di un prodotto del genere è molto più importante capire le implicazioni «concettuali» di quanto non sia importante imparare gli aspetti operativi, la cui conoscenza è del tutto inutile se prima non si padroneggiano i concetti sottostanti il disegno del database e il linguaggio FORML.

Va quindi verificato il fatto che la manualistica sia in grado di spiegare in maniera semplice e produttiva la filosofia FORML anche ai non addetti ai lavori.

Ebbene il primo manuale, A Guide to FORML, è ottimo, nel senso che l'approccio, un misto tra descrizioni semplici, introduzione agli aspetti concettuali e ai formalismi, con esempi molto comprensibili, risulta essere efficace.

Le pagine sono 140, distribuite su sei capitoli; i primi introduttivi e i secondi specifici sui singoli concetti base (Facts, Predicates, Roles, Object, types e subtypes, Constraint) e tre appendici.

Molto interessante l'appendice A che presenta un confronto tra il metodo Entity-Relationship, il linguaggio formale più diffuso, e il metodo ORM. La B riguarda la descrizione delle simbologie grafiche usate nel Table Browser e la C l'illustrazione di un Case Study, riguardante un modello complesso.

Il secondo manuale è l'User Manual (200 pagine).

A differenza del primo manuale, che in un certo senso prescinde dal prodotto InfoModeler e si concentra sul linguaggio, parla dei comandi di InfoModeler. I primi capitoli trattano i concetti base del FORML, che vengono esemplificati con esercizi, e che vengono visti direttamente nel prodotto. Vengono trattati gli aspetti operativi coinvolti nella costruzione del database, come ambiente di lavoro, strumenti per il disegno, finestre di dialogo per la definizione «fine» degli oggetti, e poi tutte le

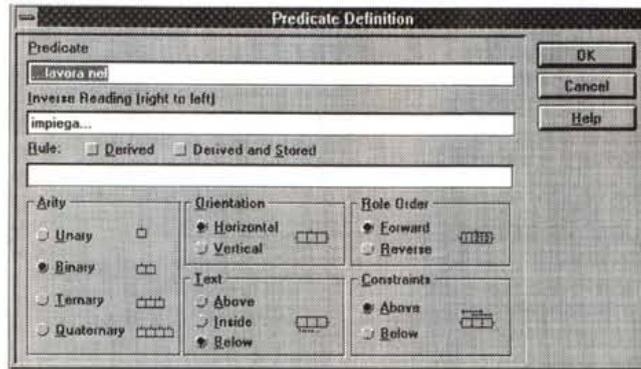


Figura 8 - Asymetrix InfoModeler - Definizione del predicato. Volendo semplificare potremo dire che il Predicato è il verbo che mette in relazione il soggetto di una frase con il complemento oggetto. Nello specificare nel dettaglio il Predicato ne va individuata l'Arity, ovvero il numero di Object coinvolti (da 1 a 4), la direzione, che può essere diretta e inversa, ecc.

Figura 9 - Asymetrix InfoModeler - Box Object Browser. Gli oggetti inseriti nelle vari «frasi» disegnate vengono accuratamente catalogati. Da questa Box è possibile facilmente risalire alle posizioni occupate dei vari oggetti nello schema. Ad esempio in questo caso vediamo in quali frasi è «coinvolto» un Fornitore (Supplier).

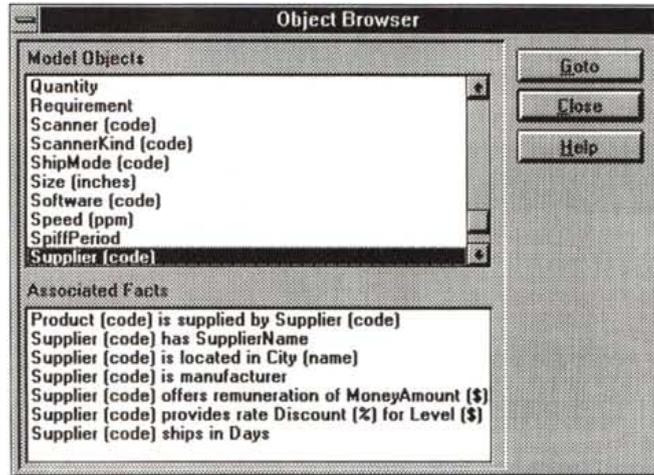
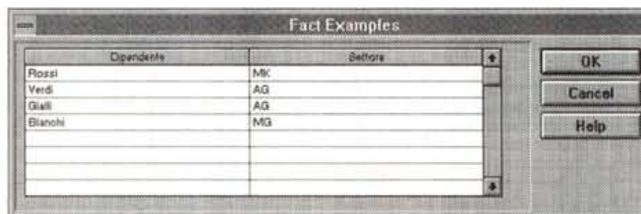


Figura 10 - Asymetrix InfoModeler - Inserimento di dati di esempio. È possibile inserire dati di esempio che non vengono usati da InfoModeler per costruire il modello ma servono per far capire meglio all'utente quali vincoli deve impostare, ad esempio quelli che indicano la duplicabilità delle chiavi.



funzionalità successive, come quelle che servono per validare il progetto, per produrre la vista Tabellare, per la costruzione del Database, nel formato DBMS desiderato, per produrre la documentazione cartacea.

L'ultimo manuale è il Database Drivers Guide, che approfondisce, in un totale di oltre 60 pagine, i problemi connessi con la produzione dei file dei database nei formati DBMS desiderati. I formati sono: Access, FoxPro, Paradox, SQL generico, ORACLE7, SQL Server (Microsoft o Sybase), ad ognuno è dedicato un suo specifico capitolo.

**Piccolo glossario degli elementi in gioco**

Innanzitutto gli elementi complessi, ovvero i FACTS, che costituiscono la

formalizzazione delle frasi (fig. 5), ricadenti in sei tipi standard, comprensive di Soggetto, Predicato e Oggetto. Esistono frasi di tipo «unario», con solo soggetto e predicato:

Filippo fuma,

frasi di tipo «binario», con soggetto, predicato ed oggetto:

Giorgio è nato il 2 febbraio,

nei tipi binari deve esistere comunque un vincolo di unicità e uno di obbligatorietà, che in questo caso significa che Giorgio ha e deve avere una sola data di nascita. Esistono poi tipi «binari ciclici» che legano due oggetti dello stesso tipo:

Giorgio ha come capo Filippo,

Esistono tipi ternari, che coinvolgono un soggetto e due oggetti:

Luigi ha preso 10 in matematica,

Esistono tipo quaternari, che coinvol-

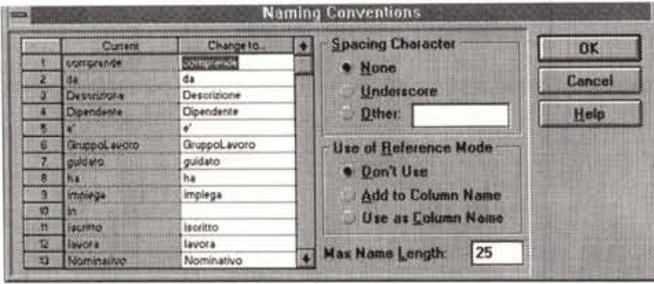


Figura 11 - Asymetrix InfoModeler - Naming Conventions. Quando InfoModeler crea il modello usa i nomi degli Oggetti inseriti nei vari Facts e inventa dei nomi «composti» quando viene creato, dalla procedura, un nuovo elemento. Questi nomi possono essere modificati per renderli più compatte ed omogenei con quelli decisi direttamente dall'utilizzatore.

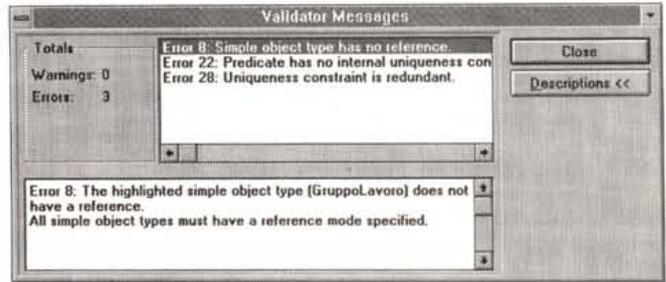
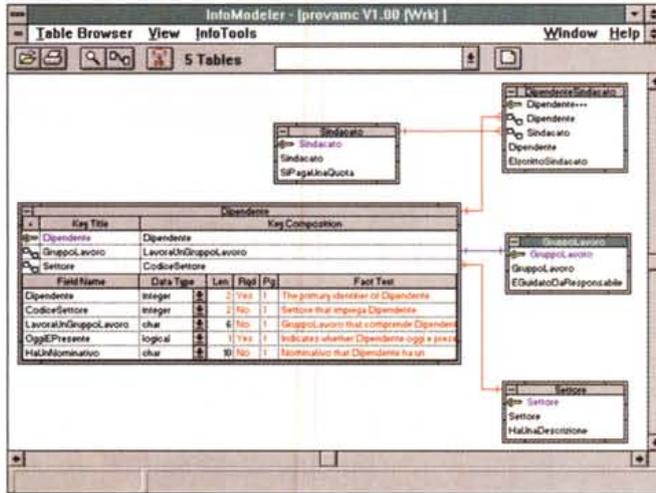


Figura 12 - Asymetrix InfoModeler - Una fase della Validazione.

Si lavora, e quindi si disegna, nell'ambiente Diagrammer. Alla fine, ma è opportuno farlo via via che si procede nel lavoro, occorre lanciare la prima funzionalità di Validazione che controlla la correttezza e la completezza degli elementi in gioco, segnalando e specificando le varie cose che non vanno. Alla fine del disegno e quando la validazione non segnala errori si può passare alla creazione del Modello vero e proprio.

Figura 13 - Asymetrix InfoModeler - Dettaglio della Tabella.

Direttamente nella vista Table Browser è possibile impostare il livello di dettaglio desiderato per ciascuna tabella. Questa si può vedere in forma icona, oppure in forma normale, con il semplice elenco dei campi, ed infine in forma espansa con tutte le informazioni di dettaglio, compresi i Facts in cui è coinvolto il campo. Si tratta di una vista operativa, in cui si può, ad esempio, modificare il tipo dei dati.



gono un soggetto e tre oggetti:

Mario ha comprato 5 quaderni per 6.500 lire da Eugenio,

Esistono tipi derivati che usano il risultato di calcoli fatti su insiemi di altri tipi:

Luigi compra i quaderni a 700  
Luigi vende i quaderni a 1.200  
Luigi guadagna ...

InfoModeler assegna una simbologia grafica a ciascun tipo di Facts. L'analista deve «disegnare», nell'ambiente Diagrammer, già visto in figura 2, individuando il tipo rappresentato dalla propria frase e impostando tutte le specifiche necessarie per quel tipo.

Esiste anche la vista Fact Compiler (fig. 6) che rappresenta l'elenco dei vari Facts impostati.

Si tratta di una vista operativa nel senso che ci si può scrivere direttamente la frase per poi trascinarla nel disegno.

Un Fact contiene soggetti, predicati ed oggetti. Soggetti e oggetti «giocano un ruolo» all'interno di un Fact. Quindi un Fact si può scomporre in OBJECT TYPE, tipi di oggetti coinvolti, in PREDICATES, che indicano quale relazione esiste tra gli oggetti coinvolti.

Un Predicate si scompone in Rules,

ruoli, che indicano un solo tipo di relazione tra il predicato e uno degli oggetti.

Gli OBJECT sono gli elementi di base della costruzione, corrispondono a collezioni di entità stesso tipo. Es. ARTICOLI, DIPENDENTI, ecc. Da un punto di vista grammaticale sono dei sostantivi, mentre i predicati sono i verbi.

Gli Object possono essere innanzitutto Entità, oppure Valori. Il primo sottotipo del tipo Entità è l'oggetto Semplice, il che significa che l'entità è identificata univocamente dall'oggetto:

il numero di matricola o la targa dell'automobile,

C'è poi il sottotipo Composto, quindi individuato da due entità differenti:

la stanza 12 del secondo piano,

Il formalismo che copre questa tipologia di Object Type richiede due Facts di tipo binario e ben cinque vincoli.

L'ultimo sottotipo di Object è il Nested ancora più complesso. Ad esempio:

attaccanti e difensori sono compresi nei calciatori,

Gli oggetti di tipo Entità vanno referenziati (ad esempio il dipendente ha un numero di matricola: Dipendente(matricola)).

Il tipo Valore si divide a sua volta in due sottotipi: Stringa e Numero, a cia-

scuno dei quali si può assegnare un ulteriore sottotipo.

Gli oggetti sono rappresentati graficamente da un'ellisse, che è tratteggiata quando si tratta di un tipo valore. Due ellissi concentriche, in InfoModeler, fanno in modo che l'oggetto serva da collegamento tra le pagine dello stesso progetto complesso.

Il PREDICATE è la parte del Fact che specifica il contesto entro il quale vive il rapporto tra i vari oggetti coinvolti. Un predicato è diviso in Roles (ruoli) ognuno dei quali connette il predicato stesso al singolo oggetto. Un Predicate, in un Fact binario, indica quindi un ruolo svolto dal primo oggetto rispetto al secondo:

Giorgio lavora nel CED,

Esiste la possibilità di definire il predicato inverso:

Il CED impiega Giorgio,

La Arity è il numero di Object coinvolti nel Fact e quindi rappresenta anche il tipo di Fact:

Filippo fuma, arity 1

Giorgio è nato il 2 febbraio, arity 2

Luigi ha preso 10 in matematica,

arity 3

I ROLE sono i ruoli svolti da un oggetto in una relazione. Per indicare più ROLES in un singolo PREDICATO vanno separati da puntini di sospensione:

...compra...da...

L'ultimo elemento in gioco sono i Vincoli, i CONSTRAINTS. Sono proposti dalla specifica Palette e sono ben 14, ciascuno dei quali si applica ad uno specifico gruppo di una specifica tipologia di Roles.

Ce ne sono alcuni di uso frequente come quello che stabilisce un rapporto

di univocità tra due oggetti (Filippo è nato il 3/4/54), simboleggiato da una freccia, oppure quello che serve per indicare l'obbligatorietà di un dato, stavolta una pallina (Rossi ha matricola 100) o quello che serve per indicare un indice secondario non univoco.

Altri sono poco diffusi ed in certi casi astrusi concettualmente. Alcuni vincoli di tipo complesso collegano tra di loro, con speciali simbolismi, due predicati.

Sono evidenti comunque due fatti sui quali torneremo in sede di conclusioni. InfoModeler ha come utente di destinazione l'Analista di professione, che ha già ben chiari in mente alcuni dei concetti logici per i quali InfoModeler propone dei simboli o dei vincoli.

Il secondo è che l'apprendimento del prodotto richiederà molto tempo, in quanto è improbabile che, anche facendo numerose applicazioni complesse, si riescano a toccare e quindi a sperimentare le varianti più particolari ed infrequenti.

### Gli ambienti operativi

Al lancio del prodotto appare l'ambiente operativo Diagrammer, sul quale occorre subito piazzare le due palette. Si può cominciare a tracciare gli oggetti, i predicati e i legami tra i vari oggetti e i singoli role dei predicati. In questa fase si usa la Palette dei Simboli (che contiene selettore, oggetti, predicati, connettore dei ruoli e connettore dei sottotipi). Per specificare meglio ciascun oggetto doppio click sull'oggetto stesso (fig. 7), così come per specificare meglio i predicati (fig. 8). Come detto si può anche usare il Fact Compiler.

Si passa poi alla definizione dei vincoli usando l'apposita Palette.

In caso di modelli molto complessi si può sfruttare l'Object Browser che permette di localizzare ciascun oggetto (fig. 9). Inoltre è possibile «popolare» i concetti inserendo dati di esempio (fig. 10). È anche possibile modificare i nomi «generati» dalla procedura (fig. 11).

Si disegna, se si sceglie la strada del disegno, in un ambiente non molto aiutato. Mancano e sarebbero comode le funzioni di allineamento degli oggetti, di raggruppamento, di selezione multipla. Insomma l'impaginazione va fatta a mano, con l'unico vantaggio che occorre spostare solo gli oggetti e i predicati in quanto i connettori, le linee insomma, si riposizionano automaticamente.

Finita la parte grafica occorre validare il disegno fatto, per poi passare oltre. Il menu file contiene i tre momenti logici seguenti il momento del disegno: - Validate, che esegue la convalida del disegno e soprattutto segnala ed evi-

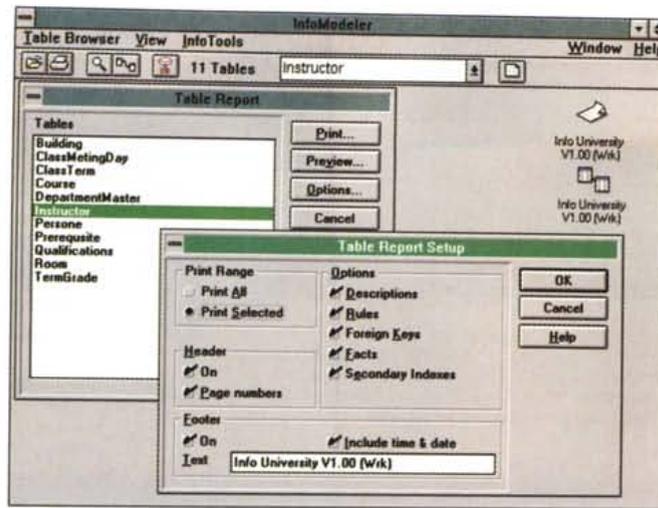


Figura 15 - Asymetrix InfoModeler - Vista Codice SQL. Il momento finale del lavoro è quello costituito da costruzione del Database, nel formato scelto dall'utente. A seconda del tipo scelto vengono creati dei programmi di generazione di tabelle (è il caso di FoxPro e di Paradox) oppure strutture dei file per i dati (è il caso di Access). Qui vediamo una videata del codice SQL nel quale si individuano variabili, istruzioni, commenti, ecc.



Figura 14 - Asymetrix InfoModeler - Report delle Tabelle. Un vantaggio non trascurabile nell'uso di un prodotto del genere è costituito dalle varie funzionalità di report, che permettono di stampare sia le viste grafiche dei vari modelli, sia la documentazione più tradizionale, quella testuale. In questo caso è possibile scegliere quali degli elementi della definizione delle Tabelle si vogliono produrre.

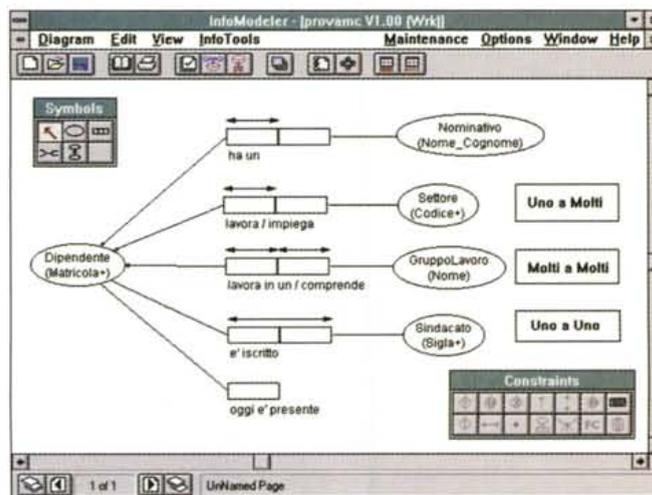
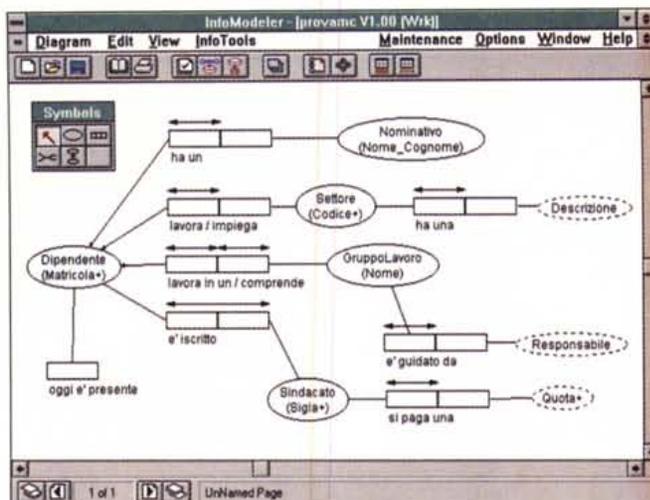


Figura 16 - Asymetrix InfoModeler - Un esempio passo passo. Le relazioni. Nei riquadri a destra (aggiunti a mano al di fuori di InfoModeler) abbiamo descritto il tipo di relazione richiamata. Uno a Molti, Molti a Molti, Uno a Uno. A ciascun tipo di relazione corrisponde uno specifico vincolo individuato dal tipo di freccia posta sopra il predicato.

denza gli errori (fig.12). A questo punto è anche possibile modificare i nomi. - Build Logical Model, può essere eseguito solo quando la validazione è andata a buon fine (in caso di necessità que-

sto comando provvede anche alla validazione). In pratica è il processo che traduce il nostro disegno nella vista tabellare, la Table Browser (fig.13). - Build Database.

Figura 17 - Asymerix InfoModeler - Un esempio passo passo. Vista Diagramma completo. Questa è la vista Diagramma completo. Si tratta di un esempio semplice in cui non abbiamo usato i tipi più particolari di Oggetti, di Predicati e di Vincoli. Da questa vista si può lanciare il comando di validazione e poi il comando di costruzione del Modello Formale. Se ambedue i passaggi non danno errori (gli errori comunque vengono evidenziati in modo che sia subito possibile correggerli) si può visualizzare la vista Table Browser.



delle tabelle, caratteristico del metodo più diffuso che è l'Entry Relationship.

InfoModeler ha come input i fact, scritti nel Fact Browser o disegnati nell'ambiente Diagrammer. Poi provvede ad eseguire il processo di validazione per controllare che i vari fact digitati non contengano errori sintattici e/o errori logici. Se tutto è a posto viene dapprima generato il modello logico comprendente le tabelle «normalizzate» (corrette dal punto di vista relazionale) e poi il database fisico.

La sua caratteristica principale è quindi quella di seguire e supportare tutte le fasi nelle quali si scompone la progettazione e la costruzione di un database.

Il suo linguaggio ci è parso dominabile, nel senso che i concetti generali così come le tipologie di elementi di più frequente utilizzo, sono assolutamente comprensibili e quasi immediatamente usabili. Alcuni concetti più complessi, ad esempio i tipi più particolari di vincoli, sono più difficili da capire ma anche meno probabili da incontrare.

I manuali aiutano molto, così come l'help in linea. Rimane il fatto che essendo l'argomento più filosofico che tecnico, in questo caso è necessaria una totale padronanza della lingua inglese. A tal proposito sarebbe auspicabile una traduzione in italiano, almeno del manuale «Guide to FORML». A suo tempo, la casa che produce InfoModeler, la Asymerix, molto nota specie nel settore Multimediale, con il quale peraltro InfoModeler non ha nulla a che vedere, tradusse il manuale di Tool-Book.

In conclusione si tratta di un prodotto destinato agli Analisti di professione, specie quelli responsabili sia del coordinamento di gruppi di persone, sia dello svolgimento delle successive fasi del lavoro, ai quali consigliamo sinceramente di provarlo, magari su applicazioni complesse per le quali già dispongono di esatti modelli logici, questo allo scopo anche di confrontare la validità dei risultati.

Per quanto abbiamo avuto noi possibilità di provarlo, ci è sembrato abbastanza facile da usare, nella definizione di fatti di complessità non eccessiva. Non abbiamo avuto invece la possibilità di costruire un esempio che «movimentasse» tutte le tipologie di oggetti e di vincoli, anche perché probabilmente, come detto, non esiste «in natura».

Riteniamo in definitiva che si tratti di uno strumento che richiede un certo «investimento» per impararlo e per renderlo produttivo nell'ambito della propria attività professionale.

MB

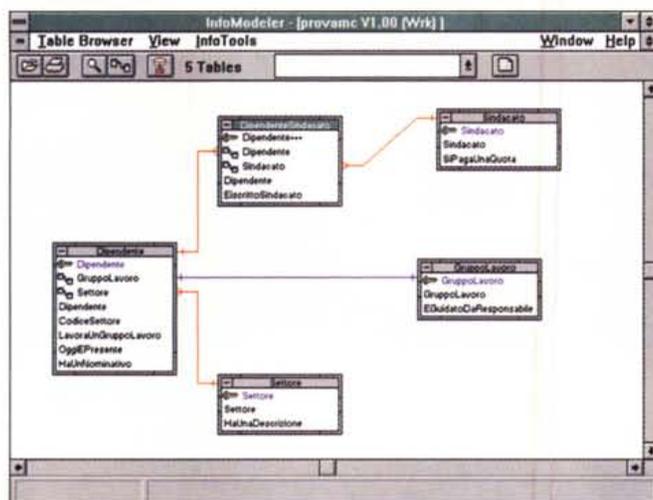


Figura 18 - Asymerix InfoModeler - Vista Table Browser.

La nostra semplice frase è stata tradotta in ben cinque tabelle, relazionate tra di loro. Le relazioni appaiono come linee che collegano campi delle tabelle. Questa vista non è operativa, nel senso che da qui non è possibile aggiungere campi, o relazioni, o quant'altro. È però possibile «entrare» nelle tabelle per modificare qualche cosa, come i nomi dei campi.

## La costruzione del Database

A questo punto, avendo generato il Modello, siamo pronti per la costruzione del Database. L'operazione può essere preceduta dalla stampa dei vari diagrammi e dalla stampa della documentazione, che si avvale di un box nel quale si inseriscono le varie specifiche (fig.14).

Occorre scegliere il tipo di database desiderato, del quale va preventivamente installato il Driver. A seconda del tipo scelto vengono prodotti dei programmi, scritti in formato testuale nel linguaggio del DBMS scelto (es. FoxPro oppure Paradox o SQL), che generano tabelle e relazioni, oppure direttamente tabelle e relazioni (es. Access).

In figura 15 vediamo uno stralcio del codice generato in SQL generico.

## Un esercizio passo passo

Nelle figure dalla 16 alla 18 presentiamo alcune fasi della costruzione del

modello relativo alla nostra prima frase, quella di figura 1. In figura 16 in particolare vediamo il momento in cui sono state scritte le varie frasi. Abbiamo aggiunto, sulla videata, delle scritte che fanno capire come vengano rappresentate le tre classiche tipologie di relazione, uno a molti, uno a uno, molti a molti. Gli esperti sanno che una relazione molti a molti (un dipendente può iscriversi a più sindacati, un sindacato ha più iscritti) genera una Tabella intermedia (matrice). Ed infatti il nostro modello presenta correttamente (in figura 18 è evidente) la tabella che contiene l'iscrizione di quel particolare dipendente a quel sindacato.

## Conclusioni

Ricapitolando: InfoModeler supporta l'approccio ORM (Object-Role Modeling) alla progettazione del database, che consiste nella generazione del modello concettuale mediante la scrittura di «fact» nei quali gli «object» giocano dei «rules», invece del disegno diretto