SISTEMI PER L'ARCHIVIAZIONE E RECUPERO DELLE INFORMAZIONI

Antonio Albano
Università di Pisa
Dipartimento di Informatica
Via F. Buonarroti 2, 56127 Pisa

INDICE

1 SISTEMI INFORMATIVI E INFORMATICI

Un recente spot pubblicitario televisivo presentava situazioni che mostravano come la conoscenza di informazioni appropriate semplifichi l'organizzazione pratica della vita quotidiana: lo slogan concludeva "l'informazione migliora la vita". Questo è un segno di quanto sia ormai patrimonio del senso comune la consapevolezza che il possesso di informazioni giuste, al momento opportuno, consente di affrontare con successo situazioni di incertezza in cui occorre prendere decisioni. La situazione più frequente però non è quella in cui mancano le informazioni, ma quella in cui le informazioni, sebbene a portata di mano, risultano difficilmente raggiungibili perché non si sa come ritrovarle. A tutti è capitato di cercare in casa propria un documento, che sappiamo contenere un'informazione utile, di cui però non ricordiamo il tipo di documento o il luogo ove è conservato.

Uno degli usi più comuni dei calcolatori è proprio per la memorizzazione e recupero di informazioni in base ad una specifica parziale del loro contenuto. Ad esempio, disponendo di un elenco telefonico gestito in modo automatico, si vuole trovare l'indirizzo di un abbonato di cui si conosce il telefono.

Molti altri esempi potrebbero essere fatti con riferimento ad attività della vita quotidiana, ma, per tenersi al tema del presente lavoro, si considera il caso del trattamento delle informazioni in un'organizzazione, al fine di migliorarne il funzionamento.

Per organizzazione si intende un insieme organizzato di uomini, risorse, strumenti e procedure, finalizzato al conseguimento di alcuni obiettivi o a fornire dei servizi: uno studio professionale, un'azienda, un ente pubblico o un ente di servizi (banca, assicurazione, ecc.) sono esempi di organizzazioni.

Ogni organizzazione, per il suo funzionamento, ha bisogno di disporre di informazioni accurate e di poterle altresì elaborare tempestivamente. Poichè questo aspetto è di grande importanza, è uso dare una visione di un'organizzazione che isoli il modo in cui la risorsa informazione viene trattata, in modo analogo a ciò che si fa per le altre risorse classiche, come il capitale e il lavoro. Quando l'attenzione è sulla risorsa informazione dell'organizzazione, si dice che di essa interessa il suo sistema informativo, del quale si propone la seguente definizione.

Un *sistema informativo* è una combinazione di risorse, umane e materiali, e di procedure organizzate per la raccolta, l'archiviazione, l'elaborazione e lo scambio delle informazioni necessarie alle attività operative (informazioni di servizio), alle attività di gestione (informazioni di gestione), e alle attività di programmazione, controllo e valutazione (informazioni di governo).

Con il termine *sistema* si evidenzia il fatto che esiste un insieme organizzato di elementi, di natura diversa, che interagiscono in modo coordinato, mentre con *informativo* si precisa che tutto ciò è finalizzato alla gestione delle informazioni e quindi le interazioni che preme evidenziare sono quelle dovute a scambi di informazioni (flussi informativi). Delle informazioni trattate da un'organizzazione, interessa qui prendere in considerazione solo quelle strutturate, con formato predeterminato, e di carattere prevalentemente globale, cioè di utilità a più reparti; si tralasciano invece informazioni non strutturate, tipo lettere, disegni, immagini, che riguardano un solo ufficio.

Vediamo alcuni esempi di organizzazioni, di diversa complessità, evidenziando le finalità del trattamento delle informazioni:

- uno studio medico mantiene, per ragioni fiscali, informazioni sui pazienti, sulle visite fatte e sulle parcelle richieste;
- una biblioteca gestisce informazioni sui materiali raccolti, sui prestiti, sulle persone che prendono in prestito materiali, per svolgere varie attività che si possono raggruppare in tre categorie:
  - attività rivolte alla raccolta dei documenti: gestione delle nuove accessioni; gestione dei periodici; descrizione dei documenti; organizzazione dei cataloghi e dell'inventario;
  - attività rivolte alla conservazione e consultazione dei documenti: gestione dei soggetti; produzione di cataloghi; recupero delle informazioni bibliografiche; distribuzione dei documenti agli utenti con il prestito e la consultazione;
  - attività rivolte alla gestione della biblioteca: controllo dell'arrivo dei periodici e restituzione dei prestiti; preparazione di statistiche per migliorare il servizio; gestione amministrativa;
- un'industria manufatturiera gestisce informazioni per svolgere attività che includono:
  - gestione degli ordini e dei pagamenti dei venditori dei prodotti;
  - gestione degli ordini e dei pagamenti ai fornitori di materiali per la produzione;
  - gestione del magazzino;
  - pianificazione della produzione;
  - controllo di gestione;
- un comune gestisce informazioni per svolgere le seguenti attività:
  - gestione dei servizi demografici (anagrafe, stato civile, servizio elettorale e vaccinale) e della rete viaria;
  - gestione dell'attività finanziaria secondo la normativa vigente;

- gestione del personale per il calcolo della retribuzione in base al tipo di normativa contrattuale;
- gestione dei servizi amministrativi e sanitari delle Unità Sanitarie Locali;
- gestione della cartografia generale e tematica del territorio.

Questi esempi riguardano situazioni molto diverse fra loro, eppure se si pensa ai tipi di informazioni gestite, non è difficile riconoscere che esse possono essere raggruppate, in base al tipo di elaborazioni a cui vengono sottoposte, come segue:

- elaborazioni rivolte alla gestione dei rapporti con l'esterno, dovuti a servizi offerti o a prodotti scambiati (processi produttivi o operativi). Per questo tipo di elaborazioni occorrono informazioni dettagliate da trattare secondo procedure standardizzate;
- elaborazioni rivolte alla gestione operativa dell'organizzazione: conoscenza, gestione e controllo delle risorse e delle loro modalità di utilizzo: personale, contabilità, bilancio, magazzini, beni mobili o immobili, ecc. (processi gestionali). Anche per questo tipo di elaborazioni occorrono informazioni dettagliate da trattare con procedure standardizzate, ma le elaborazioni sono in generale di natura più complessa;
- elaborazioni rivolte all'attività di programmazione per fissare priorità di interventi (processi decisionali o di governo). In questo caso si tratta di sintetizzare i dati provenienti dai processi operativi, e dall'ambiente esterno in cui opera l'organizzazione, per trasformarli in dati aggregati, statistiche, proiezioni e così via per fornire ai dirigenti gli elementi su cui basare le scelte, la pianificazione e gli interventi.

Le informazioni di un'organizzazione, una volta ridotte a dati con un processo di interpretazione, quantificazione e formalizzazione, possono essere trattate automaticamente con gli elaboratori elettronici. La riduzione dei costi della tecnologia informatica ha diffuso largamente questa possibilità, rendendo più accurate e rapide le procedure e potenziando i modi di elaborazione delle informazioni. Con il termine *sistema informatico* si indicano gli strumenti informatici impiegati per il trattamento automatico delle informazioni al fine di agevolare le funzioni del sistema informativo.

1.2 Evoluzione dei sistemi informatici

Agli inizi degli anni '60, l'esigenza più sentita era una strumentazione per

migliorare l'efficienza e la produttività di alcune parti dei processi operativi. Ciò ha determinato una diffusione della tecnologia informatica per applicazioni settoriali, soprattutto nell'amministrazione, con l'obiettivo di automatizzare quelle attività che richiedono l'elaborazione sistematica e ripetitiva di grandi quantità di dati. Questo nuovo modo di operare comportava indubbi vantaggi, ma presentava degli inconvenienti. I vantaggi riguardavano soprattutto la correttezza dei risultati, la riduzione dei costi e la maggiore produttività settoriale. L'inconveniente principale, invece, riguardava la scarsa integrazione dei dati in comune ai diversi settori, con duplicazioni di dati che comportava il rischio di copie incoerenti, e una limitata possibilità di correlare dati settoriali per generare informazioni di interesse globale per l'organizzazione.

A partire dagli anni '70, il progresso della tecnologia ha reso disponibili nuovi strumenti informatici che, rendendo possibile una gestione integrata dei dati, interessavano ogni livello delle organizzazioni: i dati trattati automaticamente non erano suddivisi per interessi settoriali, ma venivano trattati globalmente, in modo che ciascuna informazione, benché rappresentata una sola volta, era utilizzabile per attività diverse del sistema informativo. Si è passati quindi da sistemi informatici settoriali a sistemi informatici per l'organizzazione, con notevoli riflessi sulla struttura dell'organizzazione stessa, in quanto un impiego razionale della tecnologia informatica comporta necessariamente una revisione del modo di funzionare della struttura organizzativa che deve utilizzarla.

Con gli anni '80, infine, l'aumento delle velocità di elaborazione dei calcolatori, i progressi delle tradizionali memorie magnetiche e l'affermarsi delle memorie ottiche hanno consentito l'archiviazione e l'elaborazione dell'informazione nelle varie forme che essa può assumere – dato, testo, suono, documento, disegno, immagine, etc. – allargando ulteriormente lo spettro delle possibilità di applicazione di questa tecnologia.

1.3 Modelli informatici

Per comprendere come la tecnologia informatica viene usata nella gestione delle informazioni, è opportuno tener presente che l'informatica offre metodologie e strumenti per la costruzione di modelli di situazioni reali, con l'obiettivo di lavorare sul modello per riprodurre o predire l'evoluzione della situazione oggetto di studio.

I modelli ricorrono ampiamente nella tecnologia e in ogni campo che richiede un'attività di progettazione. Essi permettono di riprodurre le caratteristiche essenziali di fenomeni reali, omettendo quei dettagli che, ai fini

dello studio che ci si prefigge, costituirebbero un'inutile complicazione.

Nel campo delle scienze si impiegano modelli di natura diversa, che per semplicità qui si raggruppano in due categorie:

– modelli in scala, cioè la riproduzione in dimensioni ridotte di macchine, edifici, strutture che hanno grandi dimensioni e che vanno studiate, prima di passare alla loro realizzazione in scala reale, in condizioni il più possibile simili a quelle che esisteranno nella realtà;
– modelli astratti (matematici o simbolici), definiti come "la rappresentazione formale di idee e conoscenze relative a un fenomeno". Questa definizione evidenzia tre aspetti fondamentali di un modello astratto:

  a) è la rappresentazione di alcuni fatti di un fenomeno;
  b) la rappresentazione è data con un linguaggio formale;
  c) il modello è il risultato di un processo di interpretazione di un fenomeno, guidato dalle idee e conoscenze già possedute dal soggetto che interpreta.

L'informatica consente di costruire modelli astratti diversi:

– modelli per l'analisi del problema;
– modelli per la progettazione della soluzione;
– modelli per la realizzazione del progetto.

La differenza fra le varie categorie di modelli sta nel diverso livello di astrazione a cui si opera, e quindi nei tipi di fatti che si prendono in considerazione: per costruire il modello da utilizzare a regime si usano linguaggi formali che consentono di descrivere la soluzione del problema in tutti i dettagli richiesti da uno specifico sistema di elaborazione; per costruire invece il modello per la progettazione, e per l'analisi, si usano linguaggi che consentono di ignorare dei dettagli del sistema di elaborazione, per concentrarsi su alcune caratteristiche essenziali al fine di una più agevole comprensione degli aspetti cruciali del problema.

Un'analogia con l'atlante stradale aiuta a comprendere il ruolo dei diversi livelli di astrazione usati nella costruzone di un modello, in funzione del problema da risolvere: quando si pianifica un viaggio, uno sguardo al quadro d'unione delle singole tavole con la cartografia stradale consente di stabilire l'itinerario e le distanze da percorrere; un esame delle singole tavole consente di programmare deviazioni per strade panoramiche e le vie di accesso alle città da visitare; infine, l'esame delle piante delle città consente di fissare

dettagliati itinerari di visita.

L'automazione di un sistema informativo complesso comporta l'impiego di tutti i tipi di modelli, in modo da studiare il sistema gradualmente, per approssimazioni successive, fino ad arrivare alla realizzazione del sistema informatico di supporto alle attività del sistema informativo.

Nella prossima sezione si chiarirà innanzitutto quali fatti si modellano, cioè i fatti che si rappresentano in un modello, per passare poi a vedere nella sezione 3 come essi si possano modellare utilizzando un semplice formalismo grafico, adatto per costruire modelli di analisi. I concetti che introdurremo saranno utili per capire meglio le differenze fra i vari modi previsti dai sistemi commerciali per basi di dati per rappresentare le informazioni riconducibili a dati. Come esempio si considererà una gestione semplificata dei prestiti dei libri di una biblioteca universitaria.

I meccanismi presenti nel formalismo grafico potrebbero essere precisati ulteriormente con un linguaggio formale adatto a costruire modelli di progettazione e di realizzazione, ma, per il carattere divulgativo della presentazione, ciò non verrà fatto perché richiederebbe l'uso di nozioni specialistiche.

Nella sezione 4 si presenta una metodologia per la progettazione di basi di dati, soffermandosi sulla fase della progettazione concettuale. Nella sezione 5 si propongono alcuni casi per provare ad applicare la metodologia a problemi di complessità diversa.

Nella sezione 6 si presenta il modello relazionale dei dati come esempio di modello da usare per la realizzazione di basi di dati. Viene proposta una metodologia per trasformare un progetto in una realizzazione e si mostra l'SQL, un linguaggio praticamente universale per la definizione e l'uso interattivo di basi di dati relazionali. La possibilità di sperimentare l'uso del linguaggio consentirà di capire bene quali sono i vantaggi del suo uso per il recupero di informazioni da basi di dati strutturati. Vengono anche fatti dei cenni alla cosiddetta normalizzazione dei dati che fornisce dei criteri per stabilire quando una realizzazione soddisfi alcuni criteri di qualità.

Nella sezione 7 si passerà poi ad illustrare le caratteristiche principali dei sistemi per la gestione di basi di dati e nella sezione 8 quelle dei sistemi di archiviazione.

Infine nella sezione 9 si prenderà in considerazione il problema della gestione di informazioni rappresentate in forma testuale, chiamate talvolta banche di dati e chiariremo in che senso non vada confuso questo termine con basi di dati.

2 COSA SI MODELLA

Nella costruzione di un modello si supporrà di modellare la conoscenza concreta e la conoscenza astratta. In generale si considerano anche altri tipi di conoscenze, in particolare la conoscenza procedurale che riguarda il modo in cui si usa la conoscenza concreta per offrire determinati servizi. Ad esempio, quale procedura va seguita per la catalogazione di una nuova pubblicazione in una biblioteca oppure quale procedura va seguita quando si acquista un nuovo libro. Il trattamento di questi aspetti esula dai fini di questo lavoro.

2.1 La conoscenza concreta

La conoscenza concreta riguarda i fatti specifici che si vogliono rappresentare. Adottando un approccio semplificato, si suppone che la realtà consista di entità che hanno alcune proprietà. Si suppone inoltre che le entità omogenee siano raggruppabili in collezioni e che siano connesse fra di loro da associazioni che evolvano nel tempo. Precisiamo meglio questi concetti.

2.1.1 Le entità

Definizione
Le entità sono ciò che esiste e di cui interessa rappresentare alcuni fatti (o proprietà).

Ad esempio, sono entità i libri la “Divina Commedia” o il “Decamerone”, gli utenti “Caio” e “Sempronio” della biblioteca in esame.

Definizione
Le proprietà costituiscono i fatti che interessano soltanto perché descrivono caratteristiche di determinate entità.

Esempi di proprietà sono il cognome e il recapito di un utente. La differenza che esiste tra una proprietà ed una entità scaturisce dalla diversa interpretazione del loro ruolo nel modello: le proprietà sono fatti che non interessano di per sé, ma solo come caratterizzazione di altri fatti interpretati come entità.

Per meglio comprendere la complessità di una realtà è opportuno semplificarla e organizzarla, estraendo dalle entità interessanti, fra di loro distinte, i loro caratteri essenziali comuni, con un processo di astrazione che

individui *tipi* di entità. Ad esempio, persona è il tipo di Giovanna e Mario.

Definizione
Un tipo entità è una descrizione astratta di ciò che accomuna un insieme di entità omogenee (della stessa natura), esistenti o possibili.

Un tipo non è una specifica collezione di entità, ma descrive la struttura di tutte le entità "possibili" o "concepibili" di una certa natura. Ad esempio il tipo persona descrive non solo tutte le persone che esistono, ma anche quelle che esisteranno o che potrebbero esistere; quindi un tipo va pensato come una collezione infinita di entità possibili.

Ad un tipo sono associate le proprietà delle entità che appartengono a tale tipo, nonché le caratteristiche di tali proprietà. Ad esempio, il tipo utente ha le proprietà cognome e recapito, intendendo con questo che ogni utente ha un cognome e un recapito, ma con un valore in generale diverso da quello di tutti gli altri. Nell'esame di una realtà, tra tutte le possibili proprietà di entità omogenee, con il processo di astrazione che porta a definire il loro tipo si isolano solo quelle che sono interessanti per il fine che ci si prefigge. Ad esempio, per gli utenti della biblioteca si ritengono interessanti il cognome e il recapito, ma non il colore degli occhi o dei capelli.

Per ciò che riguarda le caratteristiche delle proprietà, ogni proprietà ha un nome, detto anche attributo, e un dominio, ovvero l'insieme dei possibili valori che tale proprietà può assumere, e può essere inoltre classificata come segue:

1. proprietà atomica, se il suo valore non è scomponibile (ad esempio, il cognome di una persona); altrimenti è detta strutturata (ad esempio, la proprietà residenza è scomponibile in indirizzo, CAP, città);
2. proprietà unione, se il suo valore può essere di tipi diversi (ad esempio, la proprietà titolare di un corso può essere un professore associato o un professore ordinario); altrimenti è detta semplice;
3. proprietà univoca, se il suo valore è unico (ad esempio, il cognome di un utente ha un unico valore); altrimenti è detta multivalore (ad esempio, la proprietà recapiti telefonici di una persona è multivalore se ammettiamo che alcune persone possano essere raggiungibili attraverso diversi numeri telefonici);
4. proprietà totale (obbligatoria), se ogni entità dell'universo del discorso ha per essa un valore specificato, altrimenti è detta parziale (o opzionale). Ad esempio, si può considerare il cognome di un utente una proprietà totale ed il suo recapito telefonico una proprietà parziale;

5. proprietà costante, se il suo valore non cambia nel tempo, altrimenti è detta variabile. Ad esempio, la data di nascita di una persona è una proprietà costante, mentre l'indirizzo è variabile;
6. proprietà calcolata, se il suo valore può essere determinato a partire dal valore di altre proprietà, altrimenti è detta non calcolata. Ad esempio l'età di una persona può essere calcolata a partire dalla data di nascita, mentre il suo cognome è una proprietà non calcolata.

2.1.2 Le collezioni di entità

Un altro aspetto molto utile nella costruzione di un modello è di raggruppare le entità dello stesso tipo in *collezioni* che chiameremo *classi*. Ad esempio la classe dei libri è l'insieme dei libri che la biblioteca possiede ad un certo istante. Il nome della classe si usa per riferirsi all'insieme degli elementi che la compongono.

Definizione
Una classe è una raccolta di entità omogenee.

In seguito useremo nomi al plurale per le classi proprio per sottolineare il fatto che esse si riferiscono a più entità.

Si noti che una classe ha due aspetti, uno intensionale, invariante nel tempo, e un altro estensionale, variabile nel tempo. L'aspetto intensionale riguarda il tipo degli elementi, mentre l'aspetto estensionale riguarda l'insieme dei suoi elementi. Come accade in generale per gli insiemi, gli elementi di una classe possono essere dati in due modi: elencandoli in modo esplicito (ad esempio, gli elementi della classe delle persone sono Mario, Giorgio ecc.), oppure caratterizzandoli mediante una condizione sui valori delle loro proprietà (ad esempio, i minorenni sono tutte le persone con età inferiore a 18 anni). Vedremo come nelle basi di dati le classi di solito si costruiscono dando in modo esplicito gli elementi, ma sarà anche molto utile definire classi dando condizioni che dovranno essere soddisfatte dai loro elementi.

2.1.3 Le gerarchie di collezioni

Un altro aspetto interessante da modellare è il fatto che spesso le classi di entità sono organizzate in una gerarchia di specializzazione (o di generalizzazione, a seconda del verso di percorrenza della gerarchia): le classi in gerarchia modellano insiemi di entità ad un diverso livello di dettaglio. Una classe della gerarchia minore di altre viene detta sottoclasse, mentre una classe della gerarchia maggiore di altre viene detta superclasse. La più importante proprietà delle gerarchie di classi è che gli elementi di una sottoclasse ereditano le caratteristiche degli elementi della superclasse, oltre ad averne altre proprie. L'uso delle gerarchie di classi è molto comune, ad esempio per classificare gli organismi animali e vegetali: quando diciamo che i marsupiali sono mammiferi intendiamo che le femmine sono dotate di ghiandole mammarie per l'allattamento dei piccoli, proprietà di ogni mammifero, ma hanno come proprietà specifica il marsupio.

Come altro esempio, la classe utenti può essere pensata come una generalizzazione delle classi studenti e docenti, per rappresentare il fatto che entità classificate come elementi di studenti e docenti possono essere classificate anche come elementi di utenti; si prescinde così dalle proprietà che le rendono semanticamente diverse, e si evidenzia invece che sono entità omogenee ad un particolare livello di astrazione, ad esempio con cognome, residenza e data di nascita come proprietà comuni. Si dice anche che studenti e docenti sono *sottoclassi*, o *specializzazioni*, della *superclasse* utenti. In generale è possibile anche definire una sottoclasse come specializzazione di più superclassi: ad esempio la sottoclasse degli studenti lavoratori a partire dalle classi studenti e impiegati.

Nel processo di modellazione è critico sia stabilire che cosa descrivere come una proprietà o come un'entità, sia stabilire una corretta gerarchia di classi per il problema in esame.

Come le classi anche le sottoclassi hanno un aspetto intensionale, riguardante il tipo dei loro elementi, e uno estensionale, riguardante gli elementi che le compongono.

Per quanto riguarda il primo aspetto, se C è una sottoclasse di D, allora il tipo degli elementi di C è un sottotipo del tipo degli elementi di D, ovvero gli elementi di C ereditano tutte le proprietà degli elementi di D, ma possono avere anche altre proprietà specifiche (*vincolo intensionale*). Ad esempio, se il tipo studente è definito come sottotipo del tipo utente, uno studente automaticamente eredita le proprietà delle persone, ma può anche avere altre proprietà come matricola, corso di laurea, etc. In generale una proprietà di un

supertipo potrebbe essere anche ridefinita nel sottotipo, ma per semplicità questa possibilità non verrà qui presa in considerazione (ad esempio, un pinguino è un uccello, ma non vola).

Per quanto riguarda l'aspetto estensionale, se C è una sottoclasse di D, allora ogni elemento di C è anche un elemento di D, ovvero gli elementi di C sono sempre un sottoinsieme degli elementi di D (*vincolo estensionale*).

E' utile distinguere almeno tre modalità di definizione di sottoclassi: per *sottoinsieme*, per *sottoinsiemi disgiunti*, e per *partizione*.

Le sottoclassi sottoinsieme, specializzazioni della stessa superclasse, non sono in generale fra loro disgiunte. Un elemento della classe utenti può essere contemporaneamente sia un elemento della sottoclasse studenti, che della sottoclasse impiegati.

Le sottoclassi sottoinsiemi disgiunti sono un gruppo di sottoclassi, specializzazioni della stessa superclasse, con elementi disgiunti, ma la cui unione è un sottoinsieme degli elementi della superclasse. Ad esempio, le sottoclassi matricole e laureandi, sono sottoinsiemi disgiunti della superclasse studenti.

Le sottoclassi partizione di una stessa classe sono fra loro disgiunte, ma l'unione dei loro elementi coincide con gli elementi della superclasse. Ad esempio, le sottoclassi Maschi e Femmine, partizione della superclasse Persone.

Una sottoclasse può essere definita anche a partire da un'altra sottoclasse, modellando così gerarchie a più livelli. Ad esempio, la sottoclasse Studenti potrebbe essere a sua volta specializzata introducendo le sottoclassi StudentiInCorso e StudentiFuoriCorso.

2.1.4 Le associazioni

E' facile convincersi che la realtà non consiste solo di collezioni di entità ognuna indipendente dalle altre, ma di entità collegate tra di loro da alcuni fatti che le arricchiscono di significato. Ad esempio, Dante ha scritto la "Divina Commedia", oppure Tizio ha in prestito il "Decamerone". Chiameremo questi collegamenti *istanze di associazioni*.

Definizione
Un'istanza di associazione è un fatto che correla due o più entità, stabilendo un legame logico fra di loro.

Come nel caso delle entità si è trovato utile la nozione di classe, così si trova utile la nozione di associazione fra classi. Il numero delle classi coinvolte

è chiamato il grado dell'associazione. Per semplicità, per ora limitiamoci al caso di associazioni fra due classi C1 e C2, dette binarie, che sono le più comuni. Vedremo più avanti qualche esempio di associazione ternaria, mentre quelle che coinvolgono più di tre classi sono rare e di non facile interpretazione.

Un'associazione è definita come un insieme di istanze di associazione fra elementi di C1 e C2. Il prodotto cartesiano delle estensioni di C1 e C2 (ovvero l'insieme delle possibili coppie con primo componente un elemento di C1 e secondo componente un elemento di C2) è detto dominio dell'associazione.

Ad esempio, se Dante e Boccaccio sono gli elementi della classe autori e la "Divina Commedia" e il "Decamerone" sono elementi della classe dei libri, l'associazione "ha scrittto" fra le classi autori e libri ha come istanze le coppie (Dante, "Divina Commedia") e (Boccacio, "Decamerone").

Anche le istanze di associazioni, come le entità, possono avere delle proprietà per descrivere fatti che non sono specifici delle entità coinvolte, ma del fatto che le correla. Ad esempio, consideriamo le collezioni delle persone e degli appartamenti e supponiamo che una persona possa possedere più appartamenti e un appartamento possa avere più proprietari. Supponiamo inoltre di essere interessati anche alla quota di possesso che ogni persona ha di un appartamento. Questo fatto non è una proprietà delle persone o degli appartamenti, ma dell'essere proprietari di appartamenti e quindi dell'associazione "possiede" fra le classi persone ed appartamenti.

Un'associazione binaria che correla una classe con sé stessa è detta ricorsiva. Ad esempio, la classe delle persone è collegata con sé stessa dall'associazione "ha antenati".

Proprietà strutturali delle associazioni

Un'associazione è caratterizzata, oltre che dal suo dominio e dalle caratteristiche delle eventuali proprietà, anche dalle seguenti proprietà strutturali: la molteplicità e la totalità.

Definizione

La molteplicità di un'associazione fra X e Y riguarda il numero massimo di elementi di Y che possono trovarsi in relazione con un elemento di X e viceversa. Si dice che l'associazione è univoca da X ad Y se ogni elemento di X può essere in relazione con al più un elemento di Y. Se non esiste tale vincolo si dice che l'associazione è multivalore da X ad Y. Allo stesso modo si definisce

il vincolo di univocità da Y ad X.

Si osservi che il vincolo di univocità da X ad Y è indipendente dal vincolo di univocità da Y ad X, dando luogo a quattro possibili combinazioni di presenza e assenza dei due vincoli.

Queste combinazioni si esprimono in modo compatto come segue. La cardinalità di un'associazione fra X e Y descrive contemporaneamente la molteplicità dell'associazione e della sua inversa. Si dice che la cardinalità è 1 a molti (1:N) se l'associazione è multivalore da X ad Y ed univoca da Y ad X. La cardinalità è molti ad 1 (N:1) se l'associazione è univoca da X ad Y e multivalore da Y ad X. La cardinalità è molti a molti (N:M) se l'associazione è multivalore in entrambe le direzioni, ed è uno ad uno (1:1) se l'associazione è univoca in entrambe le direzioni.

Pensando alle classi X ed Y come due insiemi di punti che corrispondono ai loro elementi, una linea che collega un punto di X con uno di Y rappresenta un'istanza dell'associazione fra X e Y. Se l'associazione è (1:N) vuol dire che un elemento di X può essere collegato a più di un elemento di Y. Se invece l'associazione è (1:1) vuol dire che un elemento di X può essere collegato ad un solo elemento di Y. In modo analogo si trattano gli altri casi (N:1) e (N:M) scambiando il ruolo di X ed Y.

Ad esempio, in un universo del discorso popolato da studenti, dipartimenti, corsi del piano di studi e professori, l'associazione Frequenta(Studenti, Corsi) (dove indichiamo con la notazione A(C1, C2) un'associazione A con dominio il prodotto cartesiano di C1 e C2) ha cardinalità (M:N), l'associazione Insegna(Professori, Corsi) ha cardinalità (1:N), l'associazione Dirige(Professori, Dipartimenti) ha cardinalità (1:1).

L'altra proprietà strutturale di un'associazione fra due collezioni X e Y, detta la totalità, riguarda il numero minimo di elementi di Y che sono associati ad ogni elemento di X. Se almeno un elemento di Y deve essere associata ad ogni elemento di X, si dice che l'associazione è totale su X, e viceversa sostituendo X con Y ed Y con X. Quando non sussiste il vincolo di totalità, si dice che l'associazione è parziale.

Pensando alla notazione insiemistica, nel caso di un'associazione parziale fra le classi X ed Y accade che esistono punti di X che non sono collegati con punti di Y.

Ad esempio, l'associazione Dirige(Professori, Dipartimenti) è totale su Dipartimenti, in quanto ogni dipartimento ha un direttore, ma non su Professori, in quanto non tutti i professori sono direttori di dipartimenti.

Per *struttura della conoscenza concreta* si intende l'insieme dei tipi di entità, l'insieme delle collezioni e le associazioni fra loro.

Immaginando di fotografare la realtà ad un certo istante, le entità interessanti, i valori delle loro proprietà e delle associazioni alle quali le entità partecipano, costituiscono uno *stato* della realtà. In generale, la realtà non è statica, ma dinamica, ovvero evolve, in quanto le entità, i valori delle loro proprietà e associazioni cambiano nel tempo per effetto di processi continui, dipendenti dal trascorrere del tempo, o di processi discreti, dipendenti dal verificarsi di eventi in certi istanti: ad esempio, il cambio della residenza di un utente, l'acquisizione di un nuovo libro, la concessione di un nuovo prestito ecc.

2.2 La conoscenza astratta

La conoscenza astratta riguarda i fatti generali che descrivono la (a) struttura della conoscenza concreta, (b) restrizioni sui valori possibili della conoscenza concreta e sui modi in cui essi possono evolvere nel tempo (vincoli d'integrità), e (c) regole per derivare nuovi fatti da altri noti.

E' utile classificare i vincoli d'integrità in statici e dinamici.

I vincoli d'integrità statici definiscono delle condizioni sui valori della conoscenza concreta che devono essere soddisfatte indipendentemente da come evolve l'universo del discorso. Le condizioni possono riguardare:

1. i valori di una proprietà. Ad esempio, (a) un utente ha le proprietà codice fiscale, cognome, residenza, con valori di tipo stringa di caratteri alfanumerici e anno di nascita, con valori di tipo intero; (b) uno studente universitario deve avere almeno diciassette anni; (c) lo stipendio di un impiegato è un numero positivo;
2. i valori di proprietà diverse di una stessa entità. Ad esempio, (a) per ogni impiegato le trattenute sulla paga devono essere inferiori ad un quinto dello stipendio; (b) se X è sposato con Y allora il suo stato civile è coniugato;
3. i valori di proprietà di entità diverse di uno stesso insieme. Ad esempio, (a) le matricole degli studenti sono tutte diverse; (b) se due persone hanno la stessa data di nascita, allora hanno anche la stessa età; (c) se una persona X è sposata con Y, allora Y è sposata con X. Un un attributo, o un insieme di attributi, è detto chiave rispetto ad una classe di elementi, se i suoi valori identificano univocamente un elemento della collezione, e se ogni attributo della chiave è necessario a questo fine. Un esempio di chiave in persone è il codice fiscale. In generale possono esistere più chiavi per una classe di entità, come accade per la classe degli studenti che hanno come

chiave non solo il codice fiscale ma anche la matricola. In questi casi se ne sceglie una come principale e viene chiamata chiave primaria.

4. i valori di proprietà di entità di insiemi diversi. Ad esempio: il presidente della commissione degli esami deve essere il titolare del corrispondente corso;
5. caratteristiche di insiemi di entità. Ad esempio, il numero degli studenti è inferiore ad un limite massimo prestabilito oppure un laureando in Scienze dell'informazione deve aver sostenuto almeno sedici esami annuali.

I vincoli d'integrità dinamici definiscono delle condizioni sul modo in cui la conoscenza concreta può evolvere nel tempo. Ad esempio, una persona coniugata non potrà cambiare stato civile ritornando ad essere scapolo o nubile; lo stipendio non decresce nel tempo; uno studente di un anno di corso non può iscriversi ad un anno precedente; una data di nascita non può essere modificata. In conclusione, mentre un vincolo statico riguarda ogni singolo stato dell'universo del discorso, un vincolo dinamico riguarda le transizioni da uno stato ad un altro.

Infine, esempi di fatti derivabili da altri sono l'età di una persona, ricavabile per differenza fra l'anno attuale e il suo anno di nascita, oppure la media dei voti degli esami superati da uno studente.

3 COME SI MODELLA

Nella costruzione di un modello informatico si supporrà di procedere in due stadi: prima si "definisce" il modello, descrivendo la struttura della conoscenza concreta e le altre parti della conoscenza astratta, poi si "costruisce" la rappresentazione di fatti specifici conformi alle definizioni date, ovvero la rappresentazione della conoscenza concreta.

Ad esempio, per costruire un modello informatico per la gestione di informazioni sui libri, prima si devono definire, tra le molteplici proprietà che caratterizzano un libro, quelle che interessano ai fini dell'applicazione: possono essere titolo, autore, editore ecc., come si usa in una biblioteca, oppure dettagliate informazioni qualitative su materiale e stato di conservazione come può interessare ad un laboratorio di restauro.

Una volta definite le proprietà interessanti comuni a tutti i possibili libri, si passa a costruire per ogni entità "libro" della realtà oggetto di studio una rappresentazione nel modello informatico, assegnando un valore per ogni proprietà definita.

Per la definizione del modello si possono usare diversi tipi di formalismi, che si differenziano innanzitutto per il "modello dei dati" che supportano, cioè per meccanismi di astrazione offerti per rappresentare la realtà.

Nel seguito si presentano due tipi di modelli dei dati: il modello a oggetti e il modello relazionale. Il primo verrà usato come esempio di formalismo per la progettazione di una basi di dati, mentre il secondo verrà usato come esempio di formalismo per la realizzazione di una base di dati. Il modello relazionale è adottato dagli attuali sistemi commerciali più diffusi, ma esistono anche sistemi che adottano il modello a oggetti

3.1 Il modello dei dati a oggetti

Per rappresentare in maniera naturale e diretta l'idea che il progettista si fa del mondo osservato, il modello dei dati a oggetti prevede i seguenti meccanismi d'astrazione: oggetto, tipo di oggetto, classe, gerarchie fra tipi e gerarchie fra classi.

Per rendere più chiara la presentazione, verranno dati esempi di utilizzo di questi meccanismi tramite un formalismo grafico che serva a definire, ad un primo livello di astrazione, lo schema di una base di dati, ovvero la struttura della conoscenza concreta, che chiameremo schema concettuale. Il formalismo grafico viene anche chiamato diagramma entità-relazioni. Si tenga però presente che, limitandoci ad un formalismo grafico, non saremo in grado di

mostrare un'altra caratteristica fondamentale di un modello a oggetti, che è stata proprio il motivo del suo successo: la possibilità di trattare non solo la struttura delle entità, ma anche le operazioni ad esse applicabili per modellare la conoscenza procedurale.

3.1.1 Oggetto e tipo di oggetto

Un oggetto è un'entità software con stato ed identità, che modella un'entità dell'universo del discorso.

Lo stato è costituito da un insieme di campi, o componenti, che sono valori costanti o variabili associati ad un nome, che possono assumere valori di qualsiasi complessità, e che modellano le proprietà dell'entità. I nomi dei componenti dello stato sono detti attributi degli oggetti. Negli esempi che seguono useremo sempre nomi semplici o composti che iniziano con una lettera maiuscola per riferirsi alle classi, associazioni o attributi.

Come accade per le proprietà delle entità, un attributo di un oggetto può avere valori di tipo atomico o strutturato, semplice o unione, univoco o multivalore, totale o parziale, costante o modificabile.

Ad esempio, nel caso della biblioteca, alcuni possibili attributi degli elementi della classe Utenti sono CodiceFiscale, Cognome, Residenza, AnnoDiNascita, con i primi due associati a valori di tipo stringa, sequenza di caratteri alfanumerici, il terzo strutturato, il quarto associato a valori di tipo intero.

Ogni oggetto è un valore di un tipo che specifica la struttura di un insieme di possibili oggetti, ovvero quali sono gli attributi e il tipo dei valori che possono assumere.

Come è stato detto in precedenza, in questo capitolo si vuole proporre anche un formalismo grafico per descrivere gli aspetti essenziali di uno schema concettuale di una base di dati. Per essere precisi senza appesantire la notazione, si preferisce non introdurre una notazione grafica per descrivere un tipo oggetto, ma solo una notazione per descrivere collezioni di oggetti e associazioni fra di loro che danno una visione immediata della struttura della base di dati.

3.1.2 Classi e associazioni

Nel formalismo grafico che si adotta, una classe si rappresenta con un rettangolo etichettato con il nome della classe (per convenzione un sostantivo maiuscolo al plurale). In alcuni esempi semplici, gli attributi del tipo degli elementi della classe si rappresentano associandoli direttamente alla classe usando la notazione grafica mostrata nelle Figure 1 e 2. Si usano le seguenti convenzioni:

1. attributi con valori atomici sono rappresentati con ovali collegati alla classe da un arco che termina con una freccia singola, se l'attributo è univoco, oppure con una freccia doppia, se l'attributo è multivalore;
2. se l'attributo ha valori strutturati, l'ovale si sostituisce con un quadratino dal quale escono archi verso ovali in numero pari ai campi del valore strutturato. L'arco dalla classe al quadratino e gli archi dal quadratino agli ovali possono avere frecce singole o doppie con lo stesso significato descritto al punto precedente;
3. se l'attributo ha valori unione, si procede come nel caso precedente sostituendo il quadratino con un cerchietto;
4. proprietà che possono avere valori non specificati si rappresentano con un taglio sull'arco;
5. gli attributi della chiave primaria si sottolineano.

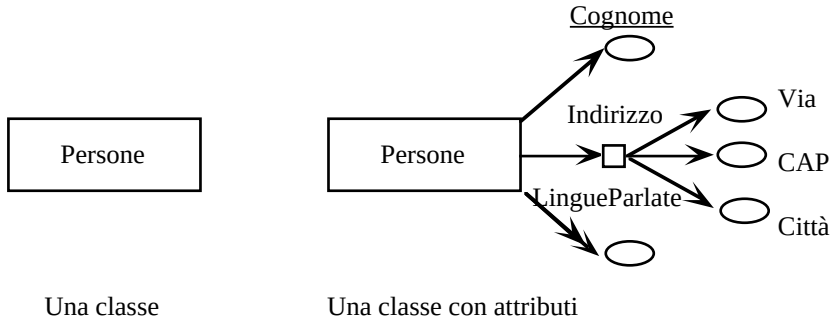

Figura 1 Rappresentazione grafica di una classe

Poiché di solito i tipi degli elementi hanno molti attributi, per non appesantire la rappresentazione grafica, si preferisce descriverli separatamente con un cosiddetto descrittore di classe, in cui si specificano il nome della classe, il nome del tipo degli elementi e poi si elencano gli attributi e i tipi dei loro valori.

Per descrivere i tipi dei valori useremo la seguente notazione:

1. i tipi primitivi integer, real, bool e string,
2. i tipi record, insieme di tante coppie “Attributo :Tipo del valore”, quanti sono i campi del record, separate da un punto e virgola e racchiuse fra parentesi quadre. Ad esempio, il tipo indirizzo è [Via:string; CAP:string; Città:string];
3. i tipi unione, insiemi di coppie “Attributo :Tipo del valore”, quante sono le alternative, separate da un punto e virgola e racchiuse fra parentesi tonde. Il tipo del valore può mancare quando non interessa (tipi enumerazione). Ad esempio, il tipo colore è (rosso; verde; bianco);
4. il tipo sequenza di valori di un tipo T, indicato come “seq T” (una sequenza si differenzia da un insieme perché gli elementi sono ordinati e possono essere uguali, pertanto {1; 2 3} è un insieme di tre interi, mentre {1; 2; 2; 3; 3; 3} è una sequenza di sei interi);
5. valori opzionali di tipo T si indicano con “optional T”.

Ad esempio, la classe delle Persone viene descritta come segue:

Classe Persone
Tipo oggetto Persona

Attributi Cognome :string;
Indirizzo :[Via :string; Numero :integer; Città :string];
LingueParlate :seq string;
Sesso :(M; F)

Un’associazione binaria tra classi si rappresenta con un rombo collegato con degli archi alle classi associate. Il rombo è etichettato con il nome dell’associazione scelto utilizzando un predicato che dia un significato alla frase con la struttura “soggetto predicato complemento” ottenuta leggendo da sinistra a destra (o dall’alto al basso) il diagramma, dove il soggetto è il generico elemento della prima collezione e il complemento il generico elemento della seconda collezione. Ad esempio, con riferimento alla Figura 2, l’associazione fra le classi Studenti ed Esami superati verrà chiamata HaSuperato.

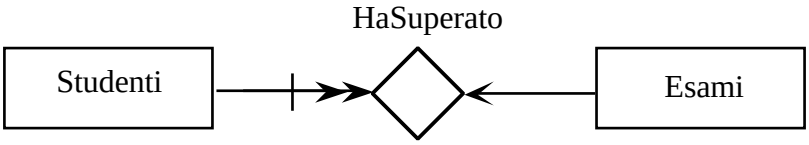

Figura 2 Rappresentazione di un'associazione

Quando però non si riesce a trovare un verbo specifico per l'associazione, oppure quando più associazioni in uno schema finirebbero per avere lo stesso nome, si può utilizzare come nome dell'associazione la concatenazione dei nomi delle classi coinvolte. Ad esempio, per l'associazione fra Studenti ed Esami si potrebbe usare il nome StudentiEsami.

L'univocità di un'associazione, rispetto ad una classe A, si rappresenta disegnando una freccia singola sull'arco tra A e il rombo; l'assenza di tale vincolo è indicata da una freccia doppia. Una freccia singola che esce dalla classe A ed entra nel rombo si può quindi leggere come "ogni elemento della classe A partecipa al più ad un'istanza dell'associazione". Similmente, la parzialità è rappresentata con un taglio sullo stesso arco, mentre il vincolo di totalità è rappresentato dall'assenza di tale taglio.

Ad esempio, in Figura 2 è rappresentata l'associazione fra studenti e gli esami da loro superati. L'arco singolo e non tagliato uscente dalla classe Esami specifica l'univocità e la totalità dell'associazione in questa direzione, ovvero il fatto che ad ogni esame corrisponde uno ed un solo studente.

Se l'associazione binaria ha delle proprietà, si aggiungono degli archi uscenti dal rombo etichettati con il nome della proprietà.

In Figura 3 è mostrato un esempio di un'associazione tra i libri di una biblioteca e gli utenti, che modella i prestiti, e che ha una proprietà "Data".

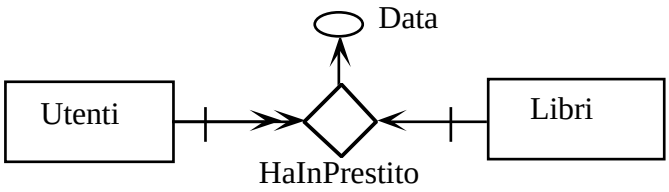

Figura 3 Rappresentazione di associazione con proprietà

Un'associazione con proprietà, come quella tra i libri di una biblioteca e gli utenti di Figura 3, può essere modellata interpretando un'istanza di associazione come un'entità e definendo così una classe Prestiti, associata in modo (1:1) ai Libri e in modo (N:1) agli Utenti, e aggiungendo un attributo

“Data” alla classe Prestiti stessa (si veda la Figura 4).

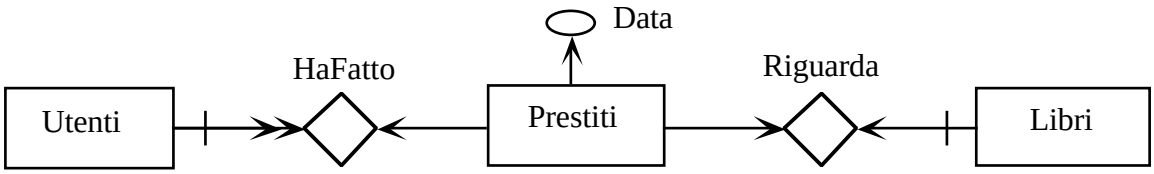

Figura 4 Trasformazione di associazione con proprietà

Gerarchia fra classi

Quando si definiscono più sottoclassi di una stessa classe, su questo insieme di sottoclassi possono essere definiti i seguenti vincoli:

1. un insieme di sottoclassi soddisfa il vincolo di disgiunzione se ogni coppia di sottoclassi in questo insieme è disgiunta, ovvero è priva di elementi comuni (sottoclassi disgiunte);
2. un insieme di sottoclassi soddisfa il vincolo di copertura se l’unione degli elementi delle sottoclassi coincide con l’insieme degli elementi della superclasse (sottoclassi copertura).

I due vincoli sono indipendenti fra loro; quando sono entrambi soddisfatti, l’insieme di sottoclassi costituisce una partizione della superclasse.

I quattro tipi di sottoclassi si descrivono come mostrato in Figura 5: il vincolo di disgiunzione viene rappresentato con il pallino nero, mentre il vincolo di copertura viene rappresentato con una freccia doppia verso la superclasse.

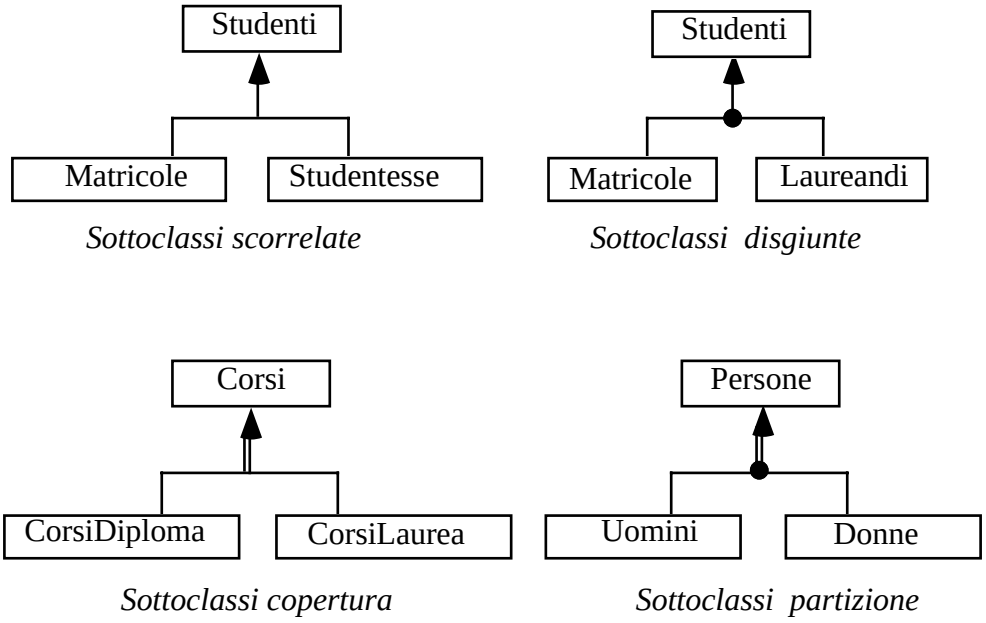

Figura 5 Esempi di sottoclassi

Esempio
A titolo di esempio, in Figura 6 si mostra, ad un primo livello di dettaglio, la rappresentazione con il formalismo grafico di alcuni fatti riguardanti una biblioteca universitaria: descrizioni bibliografiche, libri, autori, utenti e prestiti. Delle entità interessano le seguenti proprietà:

1. Di una descrizione bibliografica interessano il codice, il titolo dell'opera, l'editore, l'anno di pubblicazione e un insieme di termini usati per la classificazione del contenuto dell'opera.
2. Di un libro interessano la collocazione e il numero della copia.
3. Di un autore interessano il nome e cognome, la nazionalità, la data di nascita
4. Di un utente interessano il nome, il cognome, l'indirizzo e i recapiti telefonici.
5. Di un prestito interessano la data del prestito e la data di restituzione.

Le associazioni interessanti sono:

1. HaScritto (N:M) tra autori e descrizioni bibliografiche, che collega un

autore con le descrizioni bibliografiche delle opere che ha scritto. Ogni autore ha scritto almeno un libro e ogni descrizione bibliografica riguarda almeno un autore;

2. Descrive (N:1) tra descrizioni bibliografiche e libri, che collega una descrizione bibliografica alle copie dei libri presenti in biblioteca. Ogni libro ha una descrizione bibliografica e ogni descrizione bibliografica descrive una o più copie di libri, supporremo inoltre che possa descrivere anche un libro ordinato ma non ancora acquisito dalla biblioteca;
3. HaFatto (N:1) tra utenti e prestiti, che collega gli utenti ai prestiti che ha fatto e che non sono ancora scaduti. Ogni utente della biblioteca può avere nessuno, uno o più prestiti, ma un prestito ha sempre associato l'utente che lo ha fatto;
4. Riguarda (1:1) tra prestiti e libri, che collega i prestiti alle copie dei libri interessati. Una copia di un libro può essere coinvolta in al più un prestito e un prestito riguarda una copia di un libro.

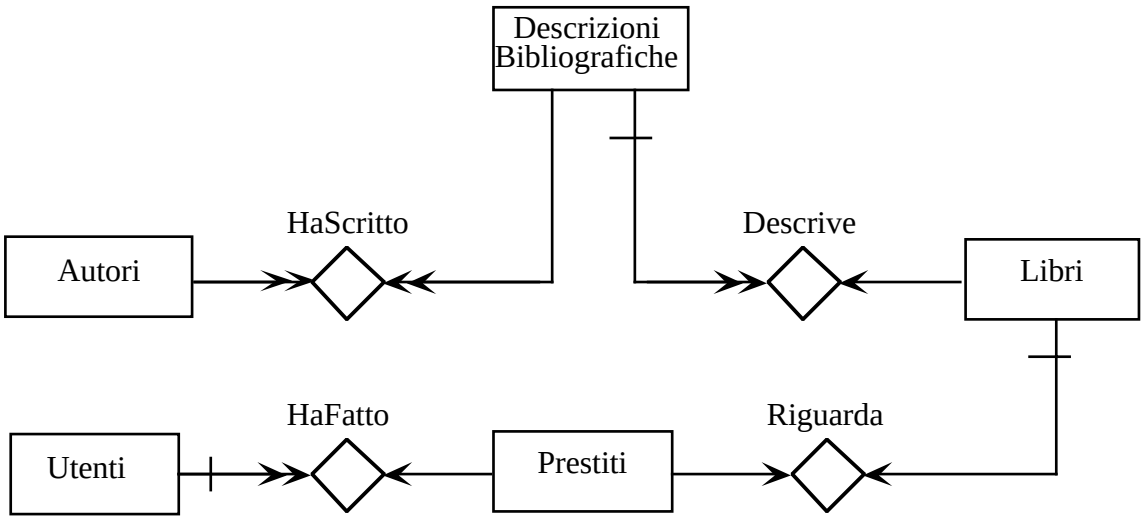

Figura 6 Prima soluzione del problema della biblioteca

Nella rappresentazione grafica si indicano anche le seguenti proprietà strutturali di un'associazione:

– la cardinalità: se l'associazione è univoca, l'arco ha una sola freccia, altrimenti la freccia è doppia; nell'esempio, un utente può avere più prestiti, ma un prestito riguarda un solo utente;
– il vincolo di dipendenza: se è parziale, si taglia l'arco; nell'esempio un libro

può non essere in prestito (fine esempio).

Passando ad un successivo livello di dettaglio si può rendere più fedele lo schema per la base di dati della biblioteca, distinguendo ad esempio le sottoclassi degli studenti e docenti della classe Utenti, nonché la sottoclasse dei libri per sola consultazione, che possono essere presi in prestito solo dagli utenti che sono docenti.

Questo approfondimento dell'analisi comporta l'introduzione di opportune sottoclassi che conducono ad una soluzione più completa, riportata in Figura 7.

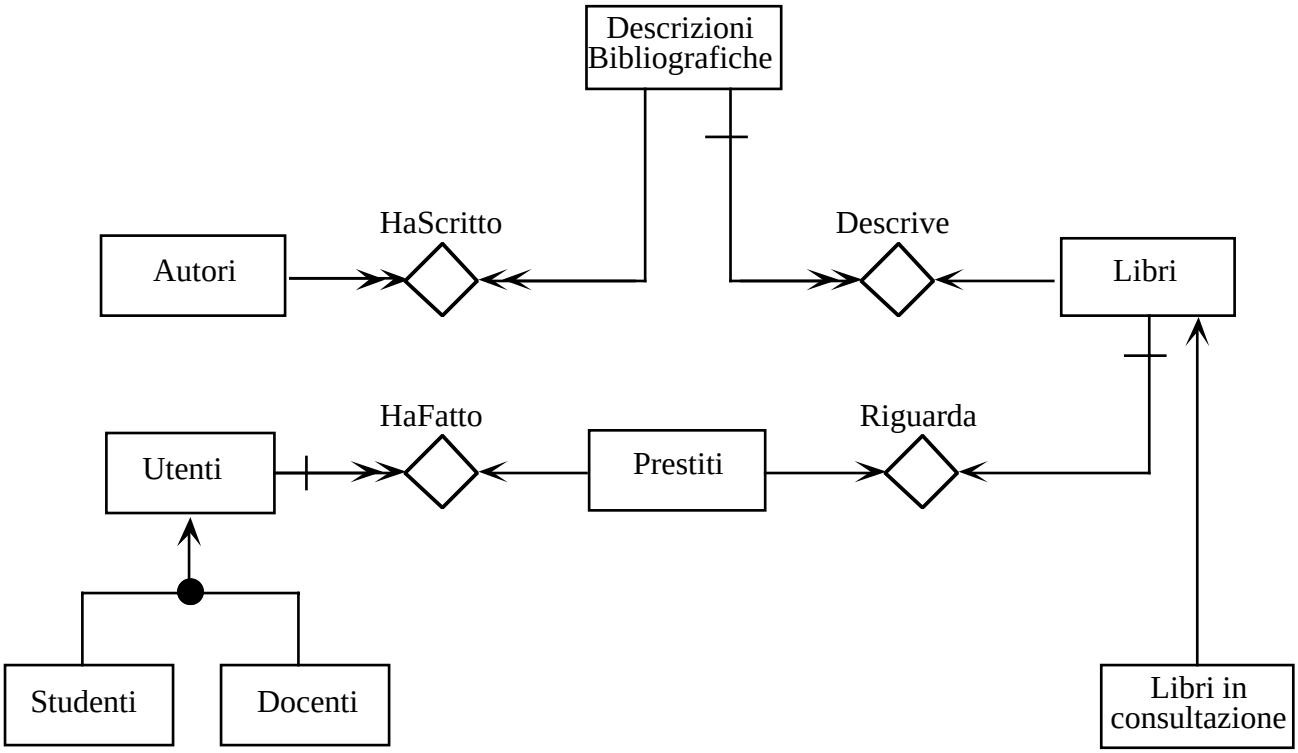

Figura 7 Soluzione finale del problema della biblioteca

4 COME SI PROCEDE

Finora si è visto come definire lo schema concettuale di una base di dati immaginando che sia chiaro quali entità sono in gioco, le loro proprietà e le associazioni a cui partecipano. Nella pratica le cose non sono così semplici ed occorre un lungo procedimento di analisi e studio della situazione da modellare per poter produrre una progettazione concettuale della base di dati e poi una sua realizzazione.

Per dare un'idea di come si procede, si considera una tipica metodologia a più fasi in cui gli aspetti del problema vengono considerati gradualmente per ottenere una realizzazione soddisfacente. Limitiamoci a considerare tre fasi in cui di solito si organizza la progettazione e per le quali avremo modo di fare degli esempi: analisi dei requisiti, progettazione concettuale e progettazione logica. Queste fasi verranno illustrate, per semplicità, considerando solo il problema del trattamento delle informazioni riconducibili a dati poiché il trattamento degli aspetti procedurali richiede competenze più ampie.

4.1 Analisi dei requisiti

Scopo dell'analisi dei requisiti è la definizione dei bisogni informativi del committente. La persona che deve progettare una base di dati, dopo alcune riunioni preliminari con un responsabile che ha il problema di voler archiviare e recuperare informazioni con un calcolatore per svolgere determinati compiti, deve innanzitutto acquisire conoscenza sul dominio del discorso per familiarizzarsi con la terminologia e la natura dei problemi. In altre parole il progettista deve innanzitutto capire di "cosa si parla". Poi si passa ad un'analisi dettagliata del problema per raccogliere una descrizione dei bisogni informativi e formulare la cosiddetta specifica dei requisiti in linguaggio naturale.

Quando il problema è di limitata complessità, e basta interagire con una sola persona, il procedimento che porta alla specifica dei requisiti può essere relativamente veloce, ma quando il problema è complesso e sono coinvolte persone diverse, che di solito hanno una visione personale dei problemi, il procedimento diventa lungo e comporta un faticoso lavoro di unificazione dei concetti coinvolti.

Più avanti vengono mostrati degli esempi relativamente semplici di specifica dei requisiti espressi in linguaggio naturale in una forma che con poco sforzo dovrebbe consentire di passare alla fase successiva di progettazione concettuale. Ma nella realtà di solito si parte da descrizioni più

confuse. Provando a risolvere i problemi proposti, si può constatare come l'interpretazione dei requisiti anche in questa forma non è del tutto banale e comporta un lavoro di riflessione e di interpretazione. Se poi il lavoro viene fatto da più di una persona si può anche verificare come ogni frase dei requisiti possa avere interpretazioni diverse. Lo scopo dell'analisi dei requisiti è proprio quello di chiarire la corretta interpretazione dei fatti descritti (di solito riparlando con le persone che li hanno espressi) riformulando la specifica in modo chiaro.

4.2 Progettazione concettuale

Scopo della seconda fase, la progettazione concettuale, è di tradurre la specifica dei requisiti in un progetto della struttura concettuale dei dati descritta utilizzando un formalismo grafico proposto nella sezione precedente. Il risultato principale di questa fase è lo schema concettuale che descrive in maniera formale le informazioni da rappresentare nella base di dati.

Lo schema concettuale si definisce procedendo con i seguenti passi, tenendo presente che ciascuno di questi passi può richiedere di modificare alcune scelte fatte nei passi precedenti (nella lettura di questi passi si pensi all'esempio della biblioteca descritto in precedenza):

1. identificazione delle classi;
2. descrizione delle associazioni fra le classi;
3. definizione di sottoclassi;
4. Definizione delle proprietà degli elementi delle classi.

Identificazione delle classi.

Si produce una lista preliminare delle classi di oggetti che interessa modellare e si assegna ad ognuna di esse un nome appropriato. Questo elenco iniziale ha un grado di completezza e di significatività che dipende dal grado di comprensione del problema e, in generale, sarà soggetto a modifiche mano a mano che si procede. Nella scelta iniziale delle classi si tengono presenti le entità di cui si vogliono ricordare alcuni fatti, senza nessuna pretesa di minimalità. Eventuali ridondanze verranno eliminate successivamente.

Descrizione delle associazioni fra le classi.

Si individuano le possibili associazioni fra le classi finora definite e le loro proprietà strutturali. L'analisi delle associazioni può portare ad eliminare una classe che può essere rappresentata da un'associazione, o ad aggiungere una nuova classe per rappresentare un'associazione, in particolare se interessa rappresentare alcuni attributi di quell'associazione.

Definizione di sottoclassi.

Per definire le sottoclassi si esaminano tutte le classi già definite per capire (a) se può essere utile definirne di nuove per caratterizzare particolari sottoinsiemi di alcune classi, (b) se esistono classi che sono un sottoinsieme di altre e quindi possono essere ridefinite, e (c) se esistono oggetti di classi che possono assumere nel tempo stati significativi per l'applicazione (ad es. gli stati di una pratica soggetta a diversi livelli di valutazione), e quindi suggeriscono l'opportunità di specializzare le relative classi per distinguere gli oggetti in base allo stato in cui si trovano.

Definizione delle proprietà degli elementi delle classi.

Per ogni tipo di oggetto si elencano le proprietà interessanti, specificando, per ognuna di esse, il nome e il tipo. In questo passo va prestata molta attenzione alla possibilità che i valori di alcune proprietà siano più significativi come oggetti a sé stanti e quindi convenga introdurre nuove classi, o viceversa al fatto che talune entità possano essere rappresentate come semplici attributi di altre. Inoltre, è frequente il caso in cui, tentando di elencare le proprietà di un oggetto, si scopre che la classe non era ben definita ed occorre rifarsi al significato di ciò che si sta descrivendo per decidere come procedere.

4.3 Progettazione logica

Scopo della terza fase della metodologia, la progettazione logica, è di tradurre lo schema concettuale nello schema logico espresso nel modello dei dati del sistema scelto per la realizzazione della base di dati. Mostreremo questa fase più avanti dopo aver discusso il modello relazionale dei dati che supporremo sia il modello dei dati da usare per la realizzazione.

5 ALCUNI CASI

Gestione di dati sui film

Si vogliono trattare informazioni su attori e registi di film. Di un attore o un regista interessano il nome, che lo identifica, l'anno di nascita e la nazionalità. Un attore può essere anche un regista. Di un film interessano il titolo, l'anno di produzione, gli attori, il regista e il produttore. Due film prodotti lo stesso anno hanno titolo diverso.

Gestione di dati su incidenti d'auto

Si vogliono gestire i dati di interesse di una compagnia di assicurazione ramo RCA. Interessano i dati sui clienti, auto e incidenti. Di un cliente interessano codice fiscale (che lo identifica), nome e indirizzo. Di un'auto interessano targa e modello. Di un incidente interessa l'auto assicurata coinvolta (si suppone che sia unica) l'ammontare del danno (in lire) e la percentuale di colpa. Un cliente può avere più automobili e un'automobile ha un solo proprietario. Un'automobile può essere stata coinvolta in più incidenti.

Gestione di dati anagrafici

Si vogliono trattare informazioni sulle persone che vivono o sono decedute in un comune italiano.

Di una persona interessano: nome, cognome, codice fiscale, data nascita (giorno, mese, anno), età, indirizzo (via, numero, cap, comune), sesso (m, f), stato civile (celibe (nubile), coniugato(a), vedovo(a), separato(a), divorziato(a), deceduto(a)), madre, padre e antenati.

Una persona può essere vivente o deceduta. Di una persona vivente interessano: indirizzo, numeri telefonici, comune di residenza, familiari conviventi, figli viventi e figli conviventi. Le persone di un nucleo familiare condividono lo stesso indirizzo, telefono e comune di residenza.

Di una persona deceduta interessano: data decesso, età, comune del decesso, comune dove è stata seppellita.

Di un matrimonio interessano: data, sposo, sposa e comune dove è stato celebrato. Non sono ammessi matrimoni fra consanguinei ovvero fra persone che hanno uno stesso antenato.

Di un comune interessano: nome, se capoluogo di provincia, prefisso telefonico, gli abitanti, il numero degli abitanti, le persone seppellite e decedute, il numero delle persone seppellite e decedute.

Gestione di dati di condomini

Si supponga di dover memorizzare in una base di dati le informazioni di interesse per un amministratore di condomini. Di un condominio interessano l'indirizzo e il numero del conto corrente dove vengono fatti i versamenti per le spese sostenute.

Un condominio si compone di appartamenti, dei quali interessano il numero dell'interno, il numero dei vani, la superficie, lo stato (libero o occupato).

Gli appartamenti possono essere locati, e dell'inquilino interessano il nome, il codice fiscale, il telefono e il saldo, cioé la somma che l'inquilino deve all'amministrazione condominiale per le spese sostenute. Alcuni appartamenti locati possono essere stati disdetti, ed in questo caso interessa la data della disdetta.

Un appartamento può avere più proprietari, e un proprietario può possedere più appartamenti. Di ogni proprietario interessano il nome, il codice fiscale, l'indirizzo, il telefono e il saldo, cioé la somma che il proprietario deve all'amministrazione condominiale per spese sostenute.

Le spese riguardano i condomini e di esse interessano il codice di identificazione, la natura (luce, pulizia, ascensore ecc.), la data e l'importo. Fra le spese si distinguono quelle straodinarie, a carico dei proprietari, e quelle ordinarie, a carico degli inquilini. Le spese ordinarie vengono pagate in un'unica rata, mentre le spese straordinarie possono essere pagate in più rate e di ognuna di esse occorre ricordare la data e l'importo.

Gestione di dati sul personale

Si vogliono gestire informazioni riguardanti gli impiegati, le loro competenze, i progetti a cui partecipano e i dipartimenti a cui appartengono.

Ogni impiegato ha una matricola che lo identifica, assegnata dalla società. Di ogni impiegato interessano il nome, la data di nascita e la data di assunzione. Se un impiegato è coniugato con un altro dipendente della stessa società, interessano la data del matrimonio e il coniuge. Ogni impiegato ha una qualifica (ad es., segretaria, impiegato, programmatore, analista, progettista ecc.). Dei laureati e delle segretarie interessano altre informazioni. Dei laureati interessa il tipo di laurea e delle segretarie la velocità di battitura a macchina.

Ogni impiegato svolge attività per un solo progetto alla volta e interessa conoscere i progetti in corso a cui partecipa.

La società è organizzata in dipartimenti, identificati da un nome e da un

numero di telefono. Un impiegato afferisce ad un solo dipartimento. Ogni dipartimento si approviggiona presso vari fornitori e un fornitore può rifornire più dipartimenti. Di ogni fornitore interessano il nome e l'indirizzo.

Interessano inoltre la data e il fornitore dell'ultimo acquisto fatto da un dipartimento.

Più impiegati partecipano ad un progetto e un impiegato può partecipare a più progetti, ma può essere assegnato ad un unico progetto per città. Di ogni città con un progetto in corso interessano la sua popolazione e la regione.

Un impiegato può avere più competenze, ma usarne solo alcune per un particolare progetto. Un impiegato usa ogni sua competenza in almeno un progetto.

Ad ogni competenza è assegnato un codice unico e una descrizione. I progetti in corso sono identificati da un numero e sono caratterizzati da una stima del loro costo.

6 MODELLO RELAZIONALE DEI DATI

Il modello relazionale dei dati, proposto nel 1970 ed adottato nei sistemi commerciali a partire dal 1978, si è diffuso rapidamente tanto sui sistemi centrali quanto sugli elaboratori personali.

I meccanismi per definire una base di dati con questo modello sono solo due: l'ennupla e la relazione. Un'ennupla, come un record, è un insieme finito di coppie (Attributo, valore atomico), mentre una relazione è un insieme finito (eventualmente vuoto) di ennuple con la stessa struttura. Un'ennupla si usa per rappresentare entità e la relazione si usa per rappresentare classi di entità. Si tenga però presente che non si possono rappresentare proprietà strutturate, o multivalore, perché i campi di un'ennupla sono atomici (numeri, stringhe o il valore NULL) e vedremo più avanti come risolvere questi problemi di rappresentazione. Prima di procedere, precisiamo il significato di alcuni termini che si usano in questo contesto.

Una relazione si definisce dandole un nome ed elencando fra parentesi tonde il tipo delle sue ennuple, un insieme di coppie (Attributo, tipo dei valori). La definizione di una relazione è detta "schema della relazione". Ad esempio, lo schema di una relazione per memorizzare dati su studenti è;

Studenti(Cognome :string,
Matricola :string,
Città :string,
AnnoNascita :integer)

In uno schema di relazione gli attributi della chiave primaria vengono sottolineati. Gli attributi delle ennuple vengono anche detti "attributi della relazione". Due relazioni hanno lo stesso tipo se hanno uguali il numero degli attributi, gli attributi e il tipo degli attributi con lo stesso nome.L'ordine degli attributi non è significativo. Ad esempio, consideriamo le seguente relazioni Impiegati e Docenti:

Impiegati(Matricola :string,
Cognome :string,
AnnoNascita :integer,
Città :string)

Docenti(Cognome :string,
Matricola :integer,

Città :string,
AnnoNascita :integer)

Le relazioni Impiegati e Studenti hanno lo stesso tipo sebbene gli attributi siano elencati in ordine diverso, mentre Docenti ha tipo diverso da Impiegati e Studenti pur avendo gli stessi attributi perché Matricola ha tipo intero e non stringa.

Una base di dati relazionale si definisce elencando gli schemi delle relazioni che ne fanno parte. L'insiemi degli schemi di relazione di una base di dati è detto "schema relazionale".

E' d'uso visualizzare una relazione come una tabella bidimensionale, con le colonne identificate dagli attributi e le righe contenenti i valori dei campi, nell'ordine indicato dall'intestazione delle colonne. Per brevità, nell'intestazione della tabella (che rappresenta lo schema della relazione) si omette di specificare il tipo dei valori degli attributi, supponendo che siano tutti stringhe di caratteri. Nel seguente esempio si mostra una tabella con alcuni dati sugli studenti.

Studenti

Nome	Matricola	Provincia	AnnoNascita
Isaia	071523	PI	1962
Rossi	067459	LU	1960
Bianchi	079856	LI	1061
Bonini	075649	PI	1962

Le associazioni tra i dati sono rappresentate attraverso i valori di opportuni campi, chiamati *chiavi esterne*, che assumono come valori quelli della chiave primaria di un'altra relazione. Ad esempio, per descrivere anche la classe degli esami superati dagli studenti della tabella precedente, il fatto che un esame è associato ad uno studente si modella prevedendo nello schema della relazione Esami un campo che assume come valori la chiave primaria degli Studenti, cioè la Matricola. Lo schema della relazione Esami è:

Esami(Materia :string,
Candidato :string,
Data :string,

Voto :integer)

Vediamo un esempio di tabella con dati sugli esami.

Esami			
Materia	Candidato	Data	Voto
DA	071523	12/01/85	28
DA	067459	15/09/84	30
MTI	079856	25/10/84	30
DA	075649	27/06/84	25
LFC	071523	10/10/83	18

Questo modo di procedere è molto intuitivo perché ha analogie con quello che si fa nella vita quotidiana quando si compilano moduli; ad esempio, quando si va in banca per versare dei soldi sul conto di un cliente, si compila un modulo in cui il cliente si identifica con il codice del suo conto corrente. L'aspetto originale del modello relazionale dei dati consiste nelle operazioni previste sulle tabelle: tali operazioni restituiscono sempre altre tabelle; inoltre fra le operazioni primitive ne è prevista una che consente di creare una nuova tabella come giunzione di due tabelle, per combinare ogni dato di una tabella con quelli che gli corrispondono nell'altra. Prima di vedere quali sono questi operatori, mostriamo come si può procedere per trasformare uno schema concettuale di una base di dati in uno schema relazionale.

6.1 Progettazione logica relazionale

Come per il modello a oggetti, anche per il modello relazionale è possibile definire un formalismo grafico in cui si rappresentano solo gli schemi di relazione e le loro associazioni, o più precisamente le chiavi esterne. In questo formalismo, una relazione è rappresentata da un rettangolo che ne contiene il nome. La presenza di una chiave esterna in R che riferisce la chiave primaria di S è rappresentata da una freccia che va da R ad S. Quando sia utile, la freccia può essere etichettata con il nome degli attributi che formano la chiave esterna, e ulteriori attributi della relazione possono essere rappresentati come visto nella sezione 2. Riportiamo, a titolo di esempio, in Figura 8 una

rappresentazione grafica dei due schemi sopra descritti.

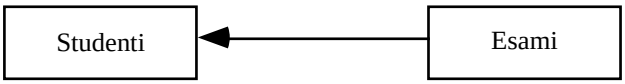

Figura 8 Rappresentazione grafica di uno schema relazionale

Data una descrizione di una base di dati con i meccanismi di astrazione del modello a oggetti, la sua trasformazione con il modello relazionale è alquanto laboriosa perché bisogna trattare non solo la descrizione delle associazioni, date con il meccanismo dell'aggregazione, e la descrizione delle gerarchie di classi, ma anche le eventuali proprietà strutturate e multivalore, per passare ad una loro descrizione con attributi definiti su domini atomici.

Inoltre, essendo il modello relazionale meno espressivo del modello a oggetti, in generale si può procedere in più modi nella trasformazione e la scelta fra possibili alternative va fatta cercando di ottimizzare lo spazio di memoria occupata dalla base di dati e le prestazioni delle applicazioni, prendendo in considerazione sia le operazioni che saranno eseguite più frequentemente (operazioni principali). Per queste ragioni i suggerimenti che si daranno vanno presi come indicazioni di massima da rivedere in una fase successiva di progettazione fisica, che però in questo testo non viene presa in considerazione.

Nella conversione di uno schema espresso con il modello a oggetti limitatamente agli aspetti riconducibili a dati, gli obiettivi da perseguire sono:

1. rappresentare le stesse informazioni;
2. minimizzare la ridondanza;
3. agevolare il recupero dei dati in relazione.

In generale nella conversione occorre duplicare delle informazioni e non si possono sempre rappresentare direttamente tutti i vincoli imposti dai mecanismi del modello a oggetti. Per garantire la coerenza dei dati duplicati, e il rispetto dei vincoli non esprimibili nel modello relazionale, occorre quindi definire opportunamente le operazioni che modificano la base di dati.

La trasformazione di uno schema a oggetti in uno schema relazionale avviene eseguendo i seguenti passi:

1. rappresentazione delle associazioni uno a uno e uno a molti;
2. rappresentazione delle associazioni molti a molti o non binarie;
3. rappresentazione delle gerarchie di inclusione;

4. rappresentazione degli attributi multivalore;
5. appiattimento gli attributi composti.

Rappresentazione delle associazioni binarie uno a molti e uno a uno

Come abbiamo già visto nell'esempio degli studenti e degli esami, le associazioni uno a molti si rappresentano aggiungendo agli attributi della relazione rispetto a cui l'associazione è univoca una chiave esterna che riferisce l'altra relazione. Ad esempio, la relazione tra esami e studenti, essendo univoca rispetto agli esami, si rappresenta aggiungendo agli esami una chiave esterna Candidato.

Quando l'associazione è uno ad uno la chiave esterna si aggiunge ad una qualunque delle due relazioni, preferendo quella rispetto a cui l'associazione è totale. Se l'associazione ha degli attributi, questi vanno aggiunti alla relazione a cui si aggiunge la chiave esterna.

Rappresentazione di associazioni molti a molti o non binarie

Un'associazione molti a molti tra due classi si rappresenta aggiungendo allo schema una nuova relazione che contiene due chiavi esterne che riferiscono le due relazioni coinvolte; la chiave candidata di questa relazione è costituita dall'insieme di tutti i suoi attributi. Questa relazione contiene un'ennupla per ogni istanza dell'associazione.

In modo analogo, un'associazione ternaria si modella aggiungendo una nuova relazione che contiene tre chiavi esterne. In entrambi i casi, se l'associazione ha degli attributi, questi attributi vengono aggiunti alla nuova relazione, e non vanno a far parte della chiave della nuova relazione.

Ad esempio, applicando questa trasformazione all'associazione HaScritto fra Autori e Descrizioni bibliografiche dello schema della biblioteca, si ottiene il disegno in Figura 9.

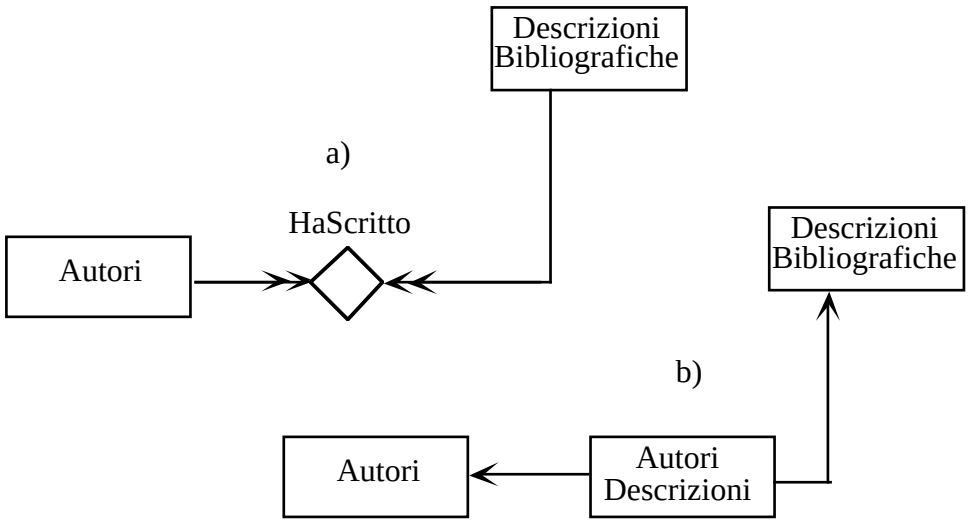

Figura 9 Trasformazione di associazioni (N:M)

Rappresentazione delle gerarchie fra classi

Sia data una classe A con due sottoclassi B e C, tali che i tipi associati alle tre classi abbiano, rispettivamente, attributi (XA), (XA XB) e (XA XC), e sia KA la chiave primaria di A. Nel modello relazionale vi sono almeno tre modi diversi di rappresentare questa situazione (ognuna con vantaggi e svantaggi che per semplicità non verranno discussi):

1. relazione unica: si definisce un'unica relazione con attributi (XA XB XC D) che raccoglie tutti gli elementi delle tre classi; gli attributi XB e XC possono assumere il valore nullo, e l'attributo D serve a indicare la classe a cui appartiene l'elemento;
2. partizionamento verticale: si definiscono tre relazioni RA(XA), RB(KA*, XB), RC(KA*, XC), dove RA contiene tutti gli elementi della classe A, anche se stanno in qualche sottoclasse, mentreRB ed RC contengono solo quegli attributi, degli elementi di B e di C, che non sono in XA (attributi propri delle sottoclassi), ed una chiave esterna KA* che permette di ritrovare in RA il valore degli altri attributi;
3. partizionamento orizzontale: si definiscono tre relazioni RA(XA), RB(XA, XB), RC(XA, XC), dove RA contiene solo gli elementi della classe A che non stanno in nessuna delle sottoclassi, mentre RB ed RC contengono tutti gli elementi di B e di C; se le sottoclassi costituiscono una copertura, la relazione RA(XA) non viene definita perché sarebbe sempre vuota.

Esempio

Consideriamo la classe Utenti con due attributi Codice e Cognome, e due sottoclassi di tipo partizione: Studenti, con attributo Matricola, e Docenti, con attributo Dipartimento.

Le tre tecniche precedenti darebbero i seguenti risultati:

1. relazione unica: un unica relazione Utenti con attributi Codice, Cognome, Matricola, Dipartimento e TipoUtente che raccoglie tutti gli utenti; gli attributi Matricola e Dipartimento sono opzionali, ed il discriminatore TipoUtente indica se l'utente è uno studente o un docente;
2. partizionamento verticale: si definiscono le relazioni Utenti, con attributi Codice e Cognome, Studenti con attributi Codice, Matricola e la relazione Docenti con attrubuti Codice e Dipartimento. La relazione Utenti contiene il codice ed il cognome di tutti gli utenti, mentre le altre due relazioni contengono gli attributi propri delle sottoclassi, nonché il codice, che permette di risalire al nome;
3. partizionamento orizzontale: trattandosi di sottoclassi che non soddisfano il vincolo di copertura si definiscono le relazioni Utenti, con attributi Codice e Cognome, Studenti con attributi Codice, Cognome, Matricola e la relazione Docenti con attributi Codice, Cognome e Dipartimento. La relazione Utenti contiene le informazioni degli utenti che non sono né studenti né docenti, la relazione Studenti contiene le informazioni degli studenti e la relazione Docenti contiene le informazioni dei docenti. (fine esempio)

Lo schema relazionale grafico per l'esempio della biblioteca è mostrato in Figura 10, nel'ipotesi di aver trasformato la gerarchia con la tecnica del partizionamento orizzontale.

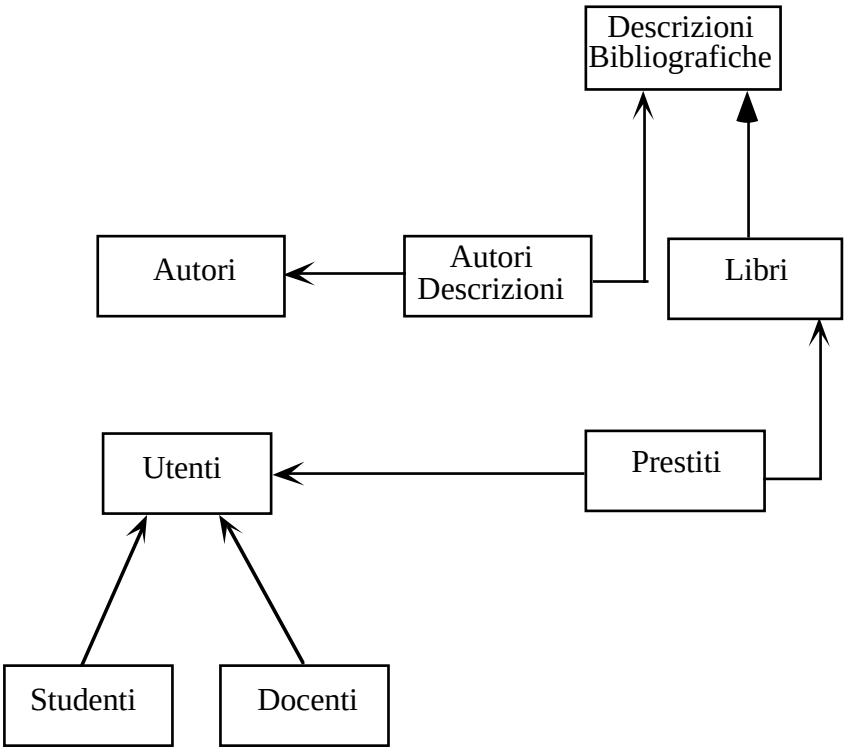

Figura10 Schema relazionale della base di dati della biblioteca

Rappresentazione delle proprietà multivalore

Una proprietà multivalore di una classeC si rappresenta eliminando il corrispondente attributo da C e creando una relazione con due attributi: una chiave esterna che fa riferimento alla chiave primaria di C ed un attributo che corrisponde all'attributo multivalore da trasformare. Un oggetto con chiave primaria k ed in cui l'attributo assume valore a1, ..., an si rappresenta poi inserendo nella nuova relazione n coppie (k, a1), ..., (k,an).

Ad esempio, si immagini che un utente abbia attributi Codice, Cognome e Telefoni, con Telefoni multivalore. Applicando la trasformazione, si ottengono le due seguenti relazioni:

Utenti(Codice, Cognome)

TelefoniUtenti(Codice, Telefono)

La chiave primaria della nuova relazione è costituita dalla concatenazione di tutti i suoi attributi.

Appiattimento degli attributi composti

Se un attributo A di uno schema di relazione è di tipo strutturato con campi Ai, si sostituisce A con gli attributi Ai. Se A faceva parte della chiave primaria dello schema di relazione, si sostituisce A con gli attributi Ai nella chiave, e poi si verifica che non esista un sottoinsieme degli attributi della nuova chiave primaria che è esso stesso una chiave.

Sullo schema relazionale ottenuto si ripetono questa trasformazione e la precedente finchè esistono schemi di relazioni con proprietà composte e proprietà multivalore.

Ad esempio, se gli utenti hanno un attributo strutturato Indirizzo con attributi Via, CAP e Città, applicando la trasformazione alla relazione

Utenti(Codice, Cognome, Indirizzo) si ottiene lo schema di relazione

Utenti(Codice, Cognome, Via, CAP, Città)

Al termine di questo passo, lo schema ottenuto rappresenta uno schema relazionale che equivale a quello a oggetti di partenza, se non per il fatto che alcuni vincoli sono andati perduti.

Si lascia al lettore come esercizio trovare gli schemi di relazione per lo schema concettuale della biblioteca.

6.2 L'algebra relazionale

Una base di dati può assere utilizzata con due modalità: interattivamente o da programmi. La prima modalità è la più semplice e può essere padroneggiata rapidamente anche da non esperti. I calcolatori personali hanno fatto grandi progressi in questa direzione, in particolare con l'impiego di terminali grafici, ed hanno liberato gli utenti dalla dipendenza dai linguaggi di programmazione per generare facilmente resoconti ('report generation') e semplici statistiche, a partire dai dati memorizzati. La seconda modalità, accesso ai dati da programmi scritti in un linguaggio di programmazione, è importante per l'esperto che automatizzi applicazioni che prevedono elaborazioni particolari sui dati oppure che debba disegnare interfacce specifiche per utenti non esperti.

Come esempio di linguaggio per l'uso interattivo di basi di dati, relazionali, vediamo prima gli operatori dell'algebra relazionale e poi il linguaggio SQL (Structured Query Language), che offre una sintassi per l'algebra relazionale che si è dimostrata molto semplice per non esperti. Il

termine algebra è dovuto al fatto che sono previsti operatori che agiscono su relazioni e producono altre relazioni come risultato.

6.2.1 Gli operatori fondamentali

Siano R e S due relazioni, Ai un generico attributo e S.Aj l'attributo Aj della relazione S:

R	
A	B
a1	b1
a2	b2
a3	b3

S	
A	B
a1	b1
a2	b2
a4	b4

Le operazioni fondamentali dell'algebra sono le seguenti:

R RINOMINA Ai IN Ai', …, Aj IN Aj'

restituisce la relazione ottenuta sostituendo in R gli attributi Ai , …, Aj con gli attributi Ai' , …, Aj'. Questo operatore si usa per cambiare il tipo di una relazione.

R UNIONE S

con R ed S relazioni con lo stesso tipo (attributi uguali e con lo stesso tipo). Restituisce una relazione dello stesso tipo di R con le ennuple che stanno in R o in S (o in entrambe). Ad esempio, l'unione delle due tabelle R ed S è:

A	B
a1	b1
a2	b2
a3	b3
a4	b4

R DIFFERENZA S

con R ed S relazioni con lo stesso tipo. Restituisce una relazione dello stesso tipo di R con le ennuple che stanno in R ma non in S. Ad esempio, la differenza delle due tabelle R ed S è:

A	B
a3	b3

PROIETTA R SU A1, .A2, .., An

con A1, A2, ..., An attributi di R. Restituisce una relazione di tipo {A1: T1, A2: T2, ..., An: Tn} con elementi la copia delle ennuple di R proiettate sugli attributi A1, A2, ..., An. Ad esempio, la proiezione di R sull'attributo A è:

A
a1
a2
a3

RESTRINGI R CON Condizione

restituisce una relazione dello stesso tipo di R con elementi la copia delle ennuple di R che soddisfano la condizione. Ad esempio, la restrizione di R con la condizione (A = a1) è:

A	B
a1	b1

R PRODOTTO S

dove R ed S sono relazioni con attributi diversi. Il prodotto di R e S restituisce una relazione con attributi queli di R e di S ed elementi la copia delle ennuple del prodotto cartesiano di R e S, ovvero ogni ennupla di R è concatenata con tutte le ennuple di S. Ad esempio, il prodotto delle due seguenti tabelle W e Z produce come risultato la terza tabella

W	
A	B
a1	b1
a2	b2

Z	
C	D
a1	b1
a3	b3

A	B	C	D
a1	b1	a1	b1
a1	b1	a3	b3
a2	b2	a1	b1
a2	b2	a3	b3

Il prodotto di solito si applica a relazioni che descrivono fatti correlati, ovvero relazioni in cui una contenga una chiave esterna per l'altra. In questi casi interessa creare una relazione in cui siano presenti solo la concatenazione delle ennuple con valori uguali della chiave e della chiave esterna, cioè siano concatenate solo le ennuple "in associazione", e si completa l'operazione con la restrizione "chiave primaria = chiave esterna". La combinazione dell'operatore prodotto e della restrizione "chiave primaria = chiave esterna" viene chiamata operazione di giunzione. Per le due relazioni W e Z, supponendo che A sia la chiave primaria e C la chiave esterna, la giunzione si esprime come:

RESTRINGI (W PRODOTTO Z) CON A = C

Si noti come al posto della relazione che è il primo argomento

dell'operatore "restringi" si può usare il risultato del prodotto delle tabelle W e Z.

6.3 Il linguaggio SQL

E' il linguaggio più diffuso per basi di dati relazionali, di ogni tipo di calcolatore, definito nei laboratori dell'IBM e commercializzato a partire dal 1982. Vediamo prima i comandi per definire basi di dati e poi quelli per interrogare basi di dati.

6.3.1 Definizione della base di dati

Per definire una relazione (detta tabella nella terminologia SQL), si usa il comando "create table", ad esempio

```
CREATE TABLE Studenti
   (Nome CHAR(20),
   Matricola CHAR(8) NOT NULL,
   Città CHAR(2),
   AnnoNascita SMALLINT)
PRIMARY KEY (Matricola)
```

La clausola NOT NULL è un esempio di vincolo d'integrità: il valore di un attributo dichiarato NOT NULL va obbligatoriamente specificato quando si aggiunge un'ennupla alla relazione. Un altro vincolo d'integrità è l'eventuale chiave primaria dichiarata con l'opzione "primary key". Gli attributi della chiave primaria non possono assumere valori NULL.

Quando nella definizione di una tabella vengono dichiarati dei vincoli, il sistema che gestisce la base di dati controlla che le operazioni che modificano la tabella inserendo nuove ennuple o modificando i valori di attributi non violino i vincoli dichiarati. Se un vincolo può essere violato, l'operazione non viene eseguita e viene segnalata una condizione di errore.

Vediamo come dichiarare la tabella degli esami con la dichiarazione della chiave esterna con l'opzione "foreign key, references Tabella on delete Azione".

```
CREATE TABLE Esami
   (Materia CHAR(20),
   Candidato CHAR(8) NOT NULL,
   Data CHAR(2),
   Voto SMALLINT)
PRIMARY KEY (Materia, Candidato)
FOREIGN KEY (Candidato)
   REFERENCES Studenti
   ON DELETE no action
```

Quando si dichiara un vincolo di chiave esterna, il sistema fa i seguenti controlli:

1. quando si inserisce un'ennupla nella tabella Esami, o quando si modifica il campo chiave esterna, il valore della chiave esterna deve essere presente in un'ennupla della tabella Studenti;
2. quando si elimina un'ennupla dalla tabella Studenti, se il valore della sua chiave primaria è usata come valore di una chiave esterna di un'ennupla della tabella Esami, allora sono possibili tre scelte:
   a. on delete no action, come mostrato nella dichiarazione della tabella, per proibire la cancellazione dell'ennupla da Studenti. Questa opzione vale anche quando si modifica il valore della chiave primaria di Studenti;
   b. on delete cascade, per eliminare sia l'ennupla da Studenti che tutte le ennuple di Esami che usano il valore della chiave primaria dell'ennupla che si elimina;
   c. on delete set null, per eliminare l'ennupla da Studenti e porre a null il valore della chiave esterna di tutte le ennuple di Esami che usano il valore della chiave primaria dell'ennupla che si elimina.

La definizione di una relazione può essere modificata in qualsiasi momento, anche dopo aver inserito dei dati, aggiungendo colonne, con il comando ALTER TABLE, oppure eliminata, con il comando DROP TABLE.

Definizione di tabelle virtuali

Una tabella virtuale (detta *vista* (*view*)) è una tabella calcolata con un'espressione SQL a partire da altre tabelle sia base che virtuali, con la sintassi che vedremo più avanti. Una tabella virtuale non corrisponde a dati fisicamente esistenti, ma denota dei dati ricavabili da altri secondo l'espressione usata per definire la tabella. L'espressione viene valutata ogni volta che si opera sulla tabella virtuale.

Su una tabella virtuale si può operare come sulle tabelle base, per quanto riguarda l'operazione di ricerca, mentre le operazioni di modifica sono soggette a restrizioni perché in generale non sono riconducibili a modifiche sulle tabelle base usate per definire la tabella virtuale.

Ad esempio, vediamo come definire la tabella calcolata degli studenti pisani:

```
CREATE VIEW StudentiPisani AS
   SELECT Nome, Matricola, AnnoNascita
   FROM Studenti
   WHERE Città = 'PI'
```

Le tabelle calcolate sono utili per ragioni diverse, in particolare per dare agli utenti visioni diverse dei dati memorizzati e per semplificare alcuni tipi di interrogazioni.

6.3.2 Modifica dei dati

Nuovi dati si inseriscono nella tabella con il comando INSERT. Ad esempio, per aggiungere una nuova ennupla alla relazione Studenti si dà il comando

```
INSERT INTO Studenti VALUES ("Tizio", "081575", "MI", 1965)
```

Per cambiare invece l'attributo Città da "MI" a "TO" per lo studente con matricola "081575", si dà il comando:

```
UPDATE Studenti
SET Città = "TO"
WHERE Matricola = "081575"
```

Per eliminare invece l'ennupla dello studente con matricola "081575", si dà

il comando:

```
DELETE Studenti
WHERE Matricola = "081575"
```

6.3.3 Recupero dei dati

Per estrarre dati da una tabella, si usa il comando SELECT che, nella sua forma più semplice, ha il seguente formato:

```
SELECT   Attributi
FROM     Relazioni
WHERE    Condizione
```

dove:

- *Attributi* è l'elenco degli attributi della relazione risultato (l'abbreviazione '*' sta per tutti gli attributi);
- *Relazioni* è l'elenco delle relazioni coinvolte dall'operazione;
- *Condizione* è un predicato per restringere le ennuple da prendere in considerazione per produrre il risultato.

Vediamo come esprimere in SQL alcune operazioni, mostrando anche il loro effetto.

Trovare il nome, la matricola e la città degli studenti

```
SELECT   Nome, Matricola, Città
FROM     Studenti
```

Questa operazione equivale all'operazione di proiezione dell'algebra relazionale.

Nome	Matricola	Città
Isaia	071523	PI
Rossi	067459	LU
Bianchi	079856	LI
Bonini	075649	PI
Tizio	081575	MI

Trovare tutti i dati degli studenti di Pisa

```
SELECT  *
FROM    Studenti
WHERE   Città = "PI"
```

Questa operazione equivale all'operazione di restrizione dell'algebra relazionale.

Nome	Matricola	Città	AnnoNascita
Isaia	071523	PI	1962
Bonini	075649	PI	1962

Le operazioni di restrizione e proiezione possono essere combinate, come mostrato dal seguente esempio.

Trovare la matricola, l'anno di nascita e il nome degli studenti di Pisa

```
SELECT  Nome, Matricola, AnnoNascita
FROM    Studenti
WHERE   Città = "PI"
```

Nome	Matricola	AnnoNascita
Isaia	071523	1962
Bonini	075649	1962

L’operazione mostrata nel prossimo esempio consente di estrarre informazioni da più relazioni tra le quali sono definite delle associazioni con il meccanismo della chiave esterna, prendendo in considerazione solo le ennuple correlate, ovvero le ennuple di una relazione che hanno il valore della chiave esterna uguale a quella della chiave primaria delle ennuple dell’altra relazione.

Trovare il nome e la data degli esami per gli studenti di Lucca che hanno superato l’esame di DA con 30

```
SELECT   Nome, Data
FROM     Studenti, Esami
WHERE    Materia = "DA"
   AND   Voto = 30
   AND   Candidato = Matricola
   AND   Città = "LU"
```

Nome	Data
Rossi	15/09/84

L’aspetto originale dei linguaggi per basi di dati relazionali è il fatto che le operazioni sulla base di dati vengono formulate senza imporre al sistema una specifica modalità di accesso ai dati, ma lasciando al sistema la scelta della migliore strategia da seguire, in base ad opportune informazioni sui dati che esso gestisce autonomamente. Ad esempio, per trovare “il nome e la data degli esami per gli studenti di Lucca che hanno superato l’esame di DA con 30” il sistema potrebbe procedere cercando prima gli studenti di Lucca e poi, per ognuno di essi, selezionare la matricola da usare per accedere agli esami, scartando quelli che non sono di “DA” e sono stati superati con meno di trenta. Un’altra possibilità sarebbe di cercare prima gli esami di “DA” superati con trenta, e poi trovare lo studente corrispondente e controllare che sia di Lucca. Questi due modi di procedere possono portare ad esecuzioni con tempi di risposta molto diversi a seconda della stima che il sistema può fare sul numero di esami di “DA” superati con trenta o sul numero di studenti di Lucca. I sistemi relazionali prevedono un modulo per l’ottimizzazione dell’esecuzione delle operazioni e quindi chi formula le interrogazioni può

ignorare questi problemi.

Le espressioni del linguaggio SQL possono essere date in modo interattivo nella forma vista, oppure in una forma grafica (come nel sistem Access della Microsoft), che per semplici richieste risulta essere più facile da usare. Il risultato può essere restituito al terminale oppure essere utilizzato per produrre tabulati con opportune intestazioni, oppure ancora visualizzato in forma grafica usando un opportuno generatore di grafici.

Per utenti specialisti, che ad esempio devono sviluppare applicazioni con una logica procedurale complessa, è prevista l'immersione di queste espressioni SQL in linguaggi di programmazione.

6.3.4 Altri operatori

Gli operatori di proiezione, restrizione e giunzione sono sufficienti per recuperare qualsiasi insieme di dati da una base di dati relazionale. Spesso però interessa applicare ai dati recuperati altre operazioni per produrre il risultato desiderato; ad esempio ordinare i dati in base al valore di alcuni attributi, contare il numero degli elementi dell'insieme ottenuto, oppure calcolare il valore minimo, massimo, medio dei valori di un attributo numerico degli elementi di un insieme. Per queste ragioni il linguaggio SQL prevede opportuni operatori per questi scopi, alcuni dei quali vengono illustrati con esempi di richieste sulla tabella Esami.

Ordinare gli esami per valori crescenti della materia.

```
SELECT   *
FROM     Esami
ORDER BY Materia
```

Esami

Materia	Candidato	Data	Voto
DA	071523	12/01/85	28
DA	067459	15/09/84	30
DA	075649	27/06/84	25
LFC	071523	10/10/83	18
MTI	079856	25/10/84	30

Partizionare l'insieme degli esami in sottoinsiemi ognuno dei quali contiene tutti gli esami relativi alla stessa materia. Di ogni sottoinsieme interessa il nome della materia, il numero degli esami, il valore minimo, massimo e medio dei voti. Il risultato va ordinato in base al nome della materia.

```
SELECT   Materia, count(*), min(Voto), max(Voto), avg(Voto)
FROM     Esami
GROUP BY Materia
ORDER BY Materia
```

Materia	count(*)	min(Voto)	max(Voto)	avg(Voto)
DA	3	25	30	27,66
LFC	1	18	18	18
MTI	1	30	30	30

6.4 Normalizzazione di schemi relazionali

Il modello relazionale dei dati ha consentito lo sviluppo di una teoria per studiare formalmente alcuni importanti problemi che si incontrano nel progetto e uso di basi di dati relazionali. Non è questa la sede per trattare questa teoria, alla quale sono dedicati volumi specifici, ma è utile accennare ad un aspetto che ormai fa parte del bagaglio minimo di nozioni di chiunque si avvicina alle basi di dati relazionali. Si tratta della nozione di schema ben formato, che nella terminologia relazionale si chiama schema in un'opportuna "forma normale": uno schema ben formato evita alcuni inconvenienti che si crerebbero nell'uso dei dati.

Per illustrare il problema, si consideri il solito schema relazionale per memorizzare fatti sugli studenti e gli esami da loro superati. Si è visto che una soluzione è di definire due schemi di relazioni come segue, con Matricola chiave primaria di Studenti, Materia e Candidato chiave di Esami e Candidato chiave esterna per Studenti (per comodità supponiamo che esista anche un'altra chiave Codice per Esami, che sceglieremo come chiave primaria):

Studenti(Matricola, Cognome, Città, AnnoNascita)
Esami(Codice, Materia, Candidato, Voto, Data)

Si ricorda inoltre che uno studente può aver superato più esami e che un

esame è associato ad un unico studente (l'associazione è quindi (1:N).

Supponiamo che qualcuno proponga di definire diversamente le relazioni come segue:

StudentiMalDefiniti(Matricola, Cognome, Città, AnnoNascita, Codice)

Esami(Codice, Materia, Voto, Data)

In altre parole, invece di rappresentare l'associazione fra studenti ed esami con la chiave esterna in Esami, si pone la chiave esterna per Esami in StudentiMalDefiniti. Basta un momento di riflessione e scoprire che questa scelta presenta dei problemi. Ad esempio, se uno studente fa tre esami, allora nella tabella StudentiMalDefiniti si troveranno tre ennuple con valori diversi del campo Codice dell'esame e tutti gli altri attributi con valori uguali. Altro problema è che la chiave di StudentiMalDefiniti non è più Matricola, ma Matricola e Codice. Poiché non si può inserire un'ennupla senza specificare il valore degli attributi della chiave primaria, ciò comporta che i dati sugli studenti si possono inserire solo dopo che lo studente ha fatto un esame. Questo fatto è scomodo perchè esistono studenti che danno un esame tardi e così i loro dati non possono far parte della tabella degli studenti dal momento in cui si iscrivono all'università.

Esistono altri problemi, chiamati "anomalie" nella terminologia relazionale, ma per ora quelli fin qui esposti possono bastare per convincersi che la seconda soluzione andrebbe sconsigliata. La teoria relazionale dei dati ha cercato di stabilire dei metodi formali per decidere che alcuni schemi non sono ben formati e come trasformarli in altri equivalenti ben formati. Questi metodi possono essere poi codificati in un programma che risolva automaticamente il problema di fare una buona progettazione relazionale.

Naturalmente per poter stabilire che uno schema è mal definito non basta elencare i suoi attributi, che sono stringhe senza significato, ma occorre dare delle informazioni su come questi attributi sono correlati fra loro per specificarne meglio il significato. Per essere più precisi, consideriamo come esempio di questo tipo di informazione le cosiddette "dipendenze funzionali fra dati".

Definizione

Sia R una relazione che contiene almeno due attributi A e B; si dice che A determina B (che si esprime in modo abbreviato come A –> B), se e solo se nelle intenzioni di chi ha definito la relazione, per ogni insieme di ennuple che possono esistere nella relazione, non possono esistere due ennuple che hanno lo stesso valore di A e valori diversi di B.

Quando “A –> B” si dice anche che “esiste una dipendenza funzionale di B da A” per analogia con la definizione di funzione in matematica: per ogni valore dell’attributo A che appare in una ennupla della relazione, esiste un unico valore di B che gli corrisponde nella stessa ennupla.

Considereremo completa la definizione di uno schema di una relazione se oltre agli attributi siano state anche specificate le dipendenze funzionali fra di loro, che hanno un ruolo analogo a quello dei vincoli d’integrità che si son visti finora.

Ad esempio, si consideri la relazione Persone così definita:

Persone(CodiceFiscale, Cognome, Città, AnnoNascita, Età)

Con CodiceFiscale la chiave. Cerchiamo le dipendenze funzionali fra gli attributi della relazione.

Essendo l’attributo codice fiscale una chiave, esso ovviamente determina ogni altro attributo della tabella perché non possono esistere due ennuple con lo stesso valore del codice fiscale e valori diversi degli altri attributi. Considerando invece l’attributo cognome, non è vero che esso determina gli altri attributi perché, ad esempio, possono esistere due persone diverse con lo stesso cognome che avranno quindi codici fiscali diversi. Considerando poi gli attributi anno di nascita ed età, è facile convincersi che vale la dipendenza funzionale “AnnoNascita –> Eta”; infatti, se esistono due persone con lo stesso anno di nascita, esse avranno anche la stessa età. Riepilogando, per la relazione Persone valgono le seguenti dipendenze funzionali:

CodiceFiscale –> Cognome
CodiceFiscale –> Città
CodiceFiscale –> AnnoNascita
CodiceFiscale –> Eta
AnnoNascita –> Eta

La definizione vista di dipendenza funzionale è un caso particolare di una definizione più generale che considera due generici insiemi di attributi X e Y di attributi di una relazione:

Definizione
Sia R una relazione che contiene almeno due insiemi di attributi X e Y, non necessariamente disgiunti; si dice che X determina Y (che si esprime in modo abbreviato come X –> Y), se e solo se nelle intenzioni di chi ha definito la

relazione, per ogni insieme di ennuple che possono esistere nella relazione, non possono esistere due ennuple che hanno lo stesso valore degli attributi X e valori diversi degli attributi Y.

Dalla definizione di dipendenza funzionale scaturiscono alcune interessanti proprietà:

1. se X –> Y, e X –> Z, allora X –> YZ e viceversa se X –> YZ, allora X –> Y, e X –> Z. Ad esempio, grazie a questa proprietà, per la tabella Persone tutte le dipendenze con CodiceFiscale a sinistra si possono sostituire con la seguente dipendenza (per migliorare la leggibilità gli attributi di un insieme si elencano separati da virgole):
   CodiceFiscale –> Cognome, Città, AnnoNascita, Eta
2. se X –> Y, e Y –> Z, allora X –> Z (transitività). Grazie a questa proprietà, la dipendenza "CodiceFiscale –> Eta" si può eliminare perché implicata dalle dipendenze "CodiceFiscale –> AnnoNascita" e " AnnoNascita –> Eta".
3. se X –> Y, e W è un altro insieme di attributi della relazione, allora XW –> YW. Ad esempio, se "CodiceFiscale –> AnnoNascita", allora
   "CodiceFiscale, Cognome –> AnnoNascita, Cognome"

   Oppure, se "CodiceFiscale –> Cognome, Città, AnnoNascita, Eta" allora
   "CodiceFiscale, Cognome –> Cognome, Città, AnnoNascita, Eta"

   Si noti che a destra Cognome non va ripetuto perché è già presente (gli X e Y sono insiemi di attributi)

4. se Y è un sottoinsieme di X, allora X –> Y e la dipendenza è detta banale.

Usando le dipendenze funzionali si può dare una definizione precisa di chiave di una relazione.

Definizione
Sia T l'insieme degli attributi di una relazione R e X un sottoinsieme di T. X è chiave di R se valgono le seguenti proprietà:
1. X –> T
2. non esiste un sottoinsieme W di X tale che W –> T.

Se vale solo la prima proprietà diremo che X è una superchiave, cioè contiene una chiave.

Dagli esempi visti sopra, risulta che CodiceFiscale è chiave per la relazione

Studenti, mentre CodiceFiscale e Cognome è una superchiave.

Riprendiamo ora in considerazione la relazione StudentiMalDefiniti vista in precedenza come esempio di relazione mal definita:

StudentiMalDefiniti(Matricola, Cognome, Città, AnnoNascita, Codice)

Proviamo a definire le dipendenze fra gli attributi. Certamente varrà la seguente:

Matricola –> Cognome, Città, AnnoNascita

Poiché uno studente può fare più esami, non varrà invece la dipendenza “Matricola –> Codice” e quindi la mtricola, non determinando tutti gli altri attributi della relazione non è più chiave, come avevamo già scoperto in precedenza. La chiave della relazione è invece Matricola e Codice.

Questo esempio ci consente di introdurre l’ultima nozione di cui si voleva parlare in questa breve introduzione alla cosiddetta teoria relazionale dei dati: gli schemi in forma normale. Questa nozione consente di dare un senso preciso alla frase "lo schema R è mal definito", riformulandola dicendo "lo schema R non è in forma normale".

Sono state definite varie forma normali, fra cui la prima che dice "uno schema è in prima forma normale se i suoi attributi sono di tipo atomico". Questa è poco utile perché abbiamo assunto finora che ogni schema è definibile solo con attributi di tipo atomico. Un’altra forma normale invece è più interessante per i nostri scopi e sarà l’unica che si prende in considerazione per concludere l’argomento:

Definizione

Uno schema di relazione R è in forma normale di Boyce-Codd (abbreviata in FNBC) se e solo se per ogni dipendenza non banale X –> Y, X è una superchiave.

In altre parole, questa forma normale richiede che uno schema, per essere privo di alcune anomalie, abbia attributi che dipendano soltanto da chiavi della relazione.

Tenendo presente questa definizione, si può dire che le relazioni Studenti ed Esami sono in FNBC, la relazione Persone non lo è per colpa della dipendenza “AnnoNascita –> Età”, e la relazione StudentiMalDefiniti non lo è per colpa della dipendenza “Matricola –> Cognome”. In entrambi i casi di schemi non in FNBC, si hanno delle anomalie dovute ad un’inutile duplicazione di dati.

Una volta scoperto che uno schema non è nella forma normale voluta, esso si può trasformare con un opportuno procedimento detto di normalizzazione. La regola principale che si segue è quella di decomporre lo schema in due schemi come segue: se uno schema R(X, Y, Z) non è nella FNBC a causa della dipendenza X –> Y, allora R si decompone nei due schemi R1(X, Y) e R2(X, Z). Se R1 e R2 non sono ancora in FNBC si ripete il procedimento.

Ad esempio, poiché nella relazione StudentiMalDefiniti valgono le dipendenze

Matricola –> Cognome
Matricola –> Città
Matricola –> AnnoNascita

e quindi la dipendenza

Matricola –> Cognome, Città, AnnoNascita che viola la FNBC, la relazione si decompone in:

Studenti(Matricola, Cognome, Città, AnnoNascita)
CodiciMatricole(Matricola, Codice)

Si noti che decomponendo in questo modo uno schema si definiscono due nuovi schemi fra i quali esiste un'associazione rappresentata dalla presenza in uno di essi della chiave esterna per l'altro. Ad esempio, nella relazione CodiciMatricole l'attributo Matricola è chiave esterna per Studenti.

Questa proprietà è del tutto generale ed è interessante perchè ci fa vedere come in uno schema mal definito esistono dipendenze fra gli attributi che non sono dovute a chiavi e ciò scaturisce dal fatto che non si è modellata correttamente un'entità come ennupla di una relazione. Normalizzando lo schema i fatti rappresentati si rappresentano in più relazioni collegate dal meccanismo delle chiavi esterne.

La normalizzazione degli schemi appare così come un altro approccio alla progettazione logica di basi di dati relazionali che ha poco in comune con l'approccio basato sulla trasformazione di schemi concettuali discusso in precedenza. In effetti se si elencano tutti i fatti elementari che si vogliono modellare (gli attributi) e le dipendenze che sussistono fra questi attributi, e si parte immaginando che tutti gli attributi appartengano ad un'unica relazione, con il processo di normalizzazione si suddividono gli attributi in relazioni più piccole ottenendo uno schema con relazioni in FNBC. Con la progettazione

concettuale, invece, si organizzano le informazioni in modo intuitivo utilizzando il modello a oggetti e poi si trasforma lo schema ottenuto in uno relazionale usando le regole di trasformazione viste. Si può dimostrare, proprio usando la teoria delle dipendenze funzionali, che le regole di trasformazione portano a schemi in FNBC, se le entità erano state ben definite (per essere sicuri di questo basta controllare che le relazioni che le rappresentano siano in FNBC) .

Le due soluzioni ottenute con questi procedimenti possono anche non coincidere, ma saranno certamente ben fatte. Pertanto i due approcci sono entrambi importanti per stabilire un procedimento che porta a schemi relazionali ben definiti. Entrambi i procedimenti sono quindi applicabili per raggiungere lo stesso scopo e la scelta fra i due dipende dal tipo di metodologia che il progettista intende seguire. Di solito le metodologie usate nella pratica in casi complessi preferiscono la trasformazione del progetto concettuale in relazionale con le regole viste.

7 SISTEMI PER LA GESTIONE DI BASI DI DATI

Dopo aver visto cosa si intende per progettare e realizzare una base di dati relazionale, in questa sezione chiariremo qual'è il significato tecnico del termine basi di dati e le funzionalità che offrono i sistemi che ne consentono la definizione e l'uso. Queste precisazioni sono necessarie per capire quali prodotti rientrano giustamente in questa categoria e quali funzionalità sono importanti per un corretto sviluppo di applicazioni che usano basi di dati.

Iniziamo col chiarire cosa si intende per "base di dati", visto che spesso questo termine viene usato per riferirsi ad un qualsiasi insieme di dati archiviati con un calcolatore.

Definizione

Una base di dati è una raccolta di dati permanenti suddivisi in due categorie:

1. i metadati, ovvero lo schema della base di dati (database schema), una raccolta di definizioni che descrivono la struttura di alcuni insiemi dati, le restrizioni sui valori ammissibili dei dati (vincoli d'integrità) e le relazioni esistenti fra gli insiemi. Lo schema va definito prima di creare i dati ed è indipendente dalle applicazioni che usano la base di dati;
2 i dati, le rappresentazioni dei fatti conformi alle definizioni dello schema, con le seguenti caratteristiche:
   a) sono organizzati in insiemi omogenei, fra i quali sono definite delle relazioni. La struttura dei dati e le relazioni sono descritte nello schema con opportuni meccanismi di astrazione che caratterizzano il cosiddetto modello dei dati;
   b). sono molti, in assoluto e rispetto ai metadati, e non possono essere gestiti in memoria temporanea;
   c) sono permanenti, cioè, una volta creati, continuano ad esistere finché non sono esplicitamente rimossi; la loro vita quindi non dipende dalla durata delle applicazioni che ne fanno uso;
   d) sono accessibili mediante transazioni (transactions), unità di lavoro atomiche che non possono avere effetti parziali;
   e) sono protetti sia da accesso da parte di utenti non autorizzati, sia da corruzione dovuta a malfunzionamenti hardware e software;
   f) sono utilizzabili contemporaneamente da utenti diversi. Il termine "utente" viene usato sia con il significato di persona che accede ai dati da un terminale in modo interattivo usando un opportuno linguaggio, sia con il significato di programma applicativo che contiene istruzioni per l'accesso ai dati.

Esempio
Per chiarire i concetti esposti si consideri la base di dati degli studenti ed esami superati definiti dagli schemi di relazioni:

Studenti(Matricola, Cognome, Città, AnnoNascita)
Esami(Materia, Candidato, Voto, Data)

Come già detto, è bene distinguere le definizioni delle relazioni dai dati che sono stati memorizzati in esse ad un certo istante. Mentre a questo punto dovrebbe essere chiaro che una base di dati contiene i dati immessi, meno ovvio è il fatto che in una base di dati si memorizzano anche informazioni sui dati definiti, chiamati metadati. Esempi di queste informazioni sono:

1. i nomi delle relazioni definite;
2. il tipo delle ennuple delle relazioni;
3. le chiavi primarie ed esterne definite;
4. i vincoli sui valori ammissibili degli attributi.

Nel caso dei sistemi relazionali, queste informazioni sono memorizzate in tabelle predefinte che sono gestite automaticamente dal sistema e sono interrogabili anche dagli utenti per avere informazioni sulle definizioni esistenti. (fine esempio)

Con la precedente definizione si mettono in evidenza i seguenti fatti:

– I dati sono strutturati, cioè hanno un formato predefinito, e il numero dei tipi di dati presenti è relativamente piccolo rispetto al numero degli esemplari di ognuno di essi.
– I dati sono raggruppati in insiemi omogenei, in relazione fra loro, e sono previsti operatori per estrarre elementi da un insieme e per conoscere quelli che, in altri insiemi, sono in relazione con essi.
– I dati sono molti (diciamo da milioni a miliardi di caratteri) e sono memorizzati in una memoria permanente, tipicamente a dischi magnetici.
– I dati sono una risorsa condivisa e disponibile per usi molteplici che spesso hanno un'importanza relativa variabile nel tempo.
– I dati sono protetti da usi non autorizzati e da malfunzionamenti hardware e software.

Queste caratteristiche delle basi di dati sono garantite da un sistema per la

gestione di basi di dati (DBMS, Data Base Management System), che ha il controllo dei dati e li rende accessibili agli utenti autorizzati.

Definizione
Un DBMS è un sistema centralizzato o distribuito che offre opportuni linguaggi per definire lo schema della base di dati, per scegliere le strutture dati per la memorizzazione dei dati, per usare la base di dati interattivamente o da programmi.

Si passa ora ad esaminare le funzionalità che caratterizzano un DBMS. Non tutti i sistemi offrono tutte le funzionalità che si prenderanno in considerazione, in particolare i DBMS previsti per calcolatori personali ne sacrificano alcune per ragioni di costo (tipicamente la gestione delle transazioni e l'accesso concorrente ai dati), ma l'elenco che segue include quelle funzionalità da considerarsi irrinunciabili per prodotti da usare nella gestione delle informazioni in organizzazioni medio-grandi.

7.1 Funzionalità dei DBMS

Un DBMS offre specifiche funzionalità per i seguenti scopi:
– definizione di basi di dati;
– uso dei dati;
– controllo dei dati;
– amministrazione della base di dati;
– distribuzione dei dati.

7.1.1 Definizione di basi di dati

Nei DBMS la base di dati è descritta separatamente dai programmi applicativi che ne fanno uso ed è utile distinguere tre diversi livelli di descrizione dei dati: il livello fisico, il livello logico e il livello di vista logica.

Al *livello fisico* viene descritto il modo in cui vanno organizzati fisicamente i dati nelle memorie permanenti e quali strutture dati ausiliarie prevedere per facilitarne l'uso. La descrizione di questi aspetti viene chiamata *schema fisico* o *interno*.

Al *livello logico* viene descritta la struttura degli insiemi di dati e delle relazioni fra loro, secondo un modello dei dati, senza nessun riferimento alla loro organizzazione fisica nella memoria permanente. La descrizione della struttura della base di dati viene chiamata lo *schema logico*.

Al *livello di vista logica* viene definito come deve apparire la struttura della base di dati ad alcune categorie di utenti. Questa descrizione viene anche chiamata *schema esterno* o *vista*, per evidenziare il fatto che essa si riferisce a ciò che un utente immagina che sia la base di dati. Le differenze fra una vista e lo schema logico della base di dati di solito riguardano diversi aspetti: gli insiemi di dati accessibili, la struttura dei dati o anche il modello dei dati. Normalmente esistono più schemi esterni, uno per ogni applicazione, in generale interdipendenti in quanto i dati in comune hanno una rappresentazione unica nella base di dati e, quindi, le modifiche loro apportate attraverso uno schema esterno si riflettono su tutte le applicazioni che li utilizzano.

Esempio

Per chiarire la differenza fra i tre livelli di descrizione dei dati, si consideri una base di dati per gestire informazioni sui docenti di un'università, di supporto alle attività dell'ufficio stipendi e della biblioteca.

Al livello di vista logica, l'ufficio stipendi richiede una vista dei dati sui docenti che include i seguenti campi: Nome e cognome, Codice fiscale, Parametro e Stipendio. La biblioteca richiede invece una vista dei dati sui docenti che include i seguenti campi: Nome e cognome, Recapito telefonico.

Al livello logico, i dati sui docenti sono descritti da un unico insieme di ennuple che includeranno i campi diversi che occorrono nelle due viste. Grazie al meccanismo degli schemi esterni, ogni applicazione vedrà poi solo i dati di sua competenza.

Ad esempio, nei sistemi relazionali, al livello logico, l'insieme dei dati è visto come una "tabella" con tante colonne quanti sono i campi di interesse, dichiarata come

```
CREATE TABLE Personale
   (Nome: char(30),
   CodiceFiscale: char(15),
   Stipendio: int,
   Parametro: char(6),
   Recapito: char(8))
```

All'ufficio stipendi e alla biblioteca viene consentito, invece, di accedere solo alle viste logiche di loro competenza, dichiarate come

```
CREATE VIEW Stipendi AS
   SELECT Nome, CodiceFiscale, Stipendio, Parametro
```

```
    FROM Personale

CREATE VIEW Biblioteca AS
    SELECT Nome, Recapito
    FROM Personale
```

Una *view* è una tabella calcolata da altre con un comando SQL. Nell'esempio è stato usato solo l'operatore che proietta una tabella su alcune colonne rendendo così inaccessibili le altre; pertanto quando si accede alla vista logica Biblioteca, sono accessibili solo i campi Nome e Recapito delle persone.

Infine, al livello fisico, il progettista della base di dati fisserà un'organizzazione fisica per l'insieme dei dati dei docenti descritto al livello logico, scegliendone una fra quelle previste dal DBMS. (fine esempio)

L'approccio con tre livelli di descrizione dei dati è stato proposto come un modo per garantire le proprietà di *indipendenza logica* e *fisica* dei DBMS, che sono un obiettivo importante di questi sistemi.

Per *indipendenza fisica*, da leggere 'indipendenza delle applicazioni dall'organizzazione fisica dei dati', si intende il fatto che i programmi applicativi non devono essere modificati in seguito a modifiche dell'organizzazione fisica dei dati. Il caso più frequente di modifica dell'organizzazione fisica dei dati si presenta quando occorre intervenire sulle strutture dati ausiliarie che agevolano il reperimento dei dati per migliorare le prestazioni di alcune applicazioni, oppure, nel caso di sistemi distribuiti, quando occorre cambiare il nodo della rete dove alcuni dati sono memorizzati per ridurre i costi di trasferimento.

Esempio

Si supponga che l'ufficio stipendi esegua con frequenza l'operazione di recupero dei dati riguardanti i docenti che abbiano un determinato parametro. Se il numero dei docenti è basso, l'operazione potrebbe essere eseguita con ritardi accettabili visitando serialmente tutti i dati presenti, ma se il numero dei docenti cresce nel tempo, ad un certo punto questo modo di procedere comporterebbe tempi di risposta intollerabili. Occorre allora modificare lo schema interno aggiungendo un indice sull'attributo Parametro con il comando

```
CREATE INDEX IndiceParametro ON Personale(Parametro)
```

Non è il caso di approfondire cosa sia un indice, basti tener presente l'analogia con gli indici analitici dei libri: per trovare rapidamente dove è trattato un argomento si consulta l'indice per trovare la pagina del libro utile a questo scopo. Una cosa analoga accade nei DBMS: per trovare rapidamente un dato in base ad una condizione sul valore di un attributo, si usa un indice che associa ad ogni valore dell'attributo un riferimento a dove il dato è memorizzato nella memoria permanente. (fine esempio)

Per garantire l'indipendenza fisica non è necessario che il DBMS abbia un'architettura con tre livelli di descrizione dei dati, ma è sufficiente che gli operatori sulla base di dati disponibili agli utenti non dipendano dall'organizzazione fisica dei dati. In questo modo cambiando il modo in cui è memorizzata una tabella (cosa possibile nei DBMS in ogni momento con un semplice comando) non si hanno conseguenze sui programmi che ne fanno uso.

Per *indipendenza logica*, da leggere 'indipendenza delle applicazioni dall'organizzazione logica dei dati', si intende il fatto che i programmi applicativi non devono essere modificati in seguito a modifiche dello schema logico. Le modifiche possono essere l'aggiunta di nuove definizioni, la modifica o l'eliminazione di alcune di quelle esistenti.

Quindi, mentre l'indipendenza fisica garantisce da modifiche dell'organizzazione fisica, l'indipendenza logica garantisce da modifiche delle esigenze informative. Quanto ampia sia questa indipendenza dipende dai meccanismi che offre il DBMS per definire la corrispondenza fra uno schema esterno, al quale fanno riferimento i programmi applicativi, e lo schema logico. Infatti, o la modifica non interessa lo schema esterno, oppure, perché non vengano modificati i programmi, occorre poter ridefinire lo schema esterno in modo da lasciare inalterata la visione della base di dati.

Nei sistemi relazionali l'indipendenza fisica e logica è offerta nella massima generalità.

Esempio

Supponiamo che si decida di cambiare l'organizzazione logica dei dati memorizzando i dati sul personale in due tabelle, Docenti e TecniciEAmministrativi. Per rendere le applicazioni che usano la tabella Personale indipendenti da questa modifica, la base di dati si ridefinisce come segue:

CREATE TABLE Docenti

```
    (Nome: char(30),
    CodiceFiscale: char(15),
    Stipendio: int,
    Parametro: char(6),
    Recapito: char(8))

CREATE TABLE TecniciEAmministrativi
    (Nome: char(30),
    CodiceFiscale: char(15),
    Stipendio: int,
    Parametro: char(6),
    Recapito: char(8))

CREATE VIEW Personale AS
    SELECT *
    FROM Docenti
UNION
    SELECT *
    FROM TecniciEAmministrativi
```

(fine esempio)

7.1.2 Uso dei dati

Come conseguenza dell'integrazione dei dati, un DBMS deve prevedere più modalità d'uso per soddisfare le esigenze delle diverse categorie di utenti che possono accedere alla base di dati. Il valore della base di dati, infatti, dipende dalla facilità con cui può essere utilizzata e quindi dalle possibilità di accesso offerte dal sistema. Prendiamo in considerazione le esigenze di tre categorie di utenti: i programmatori delle applicazioni, gli utenti non programmatori e gli utenti delle applicazioni.

Un programmatore delle applicazioni ha bisogno di accedere alla base di dati da programmi sviluppati con linguaggi di programmazione diversi: COBOL, C e Pascal sono gli esempi più diffusi. Gli operatori predefiniti del modello dei dati possono essere codificati nel linguaggio ospite secondo tre modalità:

– come procedure predefinite;
– come nuovi costrutti e il compilatore del linguaggio ospite viene esteso per trattare anche i nuovi costrutti, traducendoli in chiamate a procedure

predefinite;
– come costrutti da preelaborare, per convertire i nuovi costrutti in chiamate a procedure predefinite, prima di sottoporre un programma alla traduzione con il compilatore tradizionale del linguaggio ospite.

Per i sistemi relazionali sono stati proposti diversi linguaggi che prevedono una completa integrazione delle caratteristiche di un linguaggio di programmazione con i meccanismi di definizione ed uso della base di dati; sono i cosiddetti linguaggi della quarta generazione.

Nella categoria 'utenti non programmatori' rientrano coloro che richiedono un linguaggio interattivo a sè stante, di facile uso, per fare principalmente ricerche di dati. Utili sono anche strumenti per (a) definire in modo dichiarativo il formato in cui vanno stampati i risultati delle ricerche, (b) per visualizzare i risultati in forma grafica (istogrammi, diagrammi, torte ecc.), oppure (c) per definire in modo dichiarativo gli effetti di semplici applicazioni che prevedono il recupero e la visualizzazione di dati. I linguaggi più efficaci, in particolare quelli che prevedono l'uso della grafica e interfacce amichevoli, sono stati sviluppati con l'introduzione dei sistemi relazionali e la loro diffusione sui calcolatori personali.

Infine, gli utenti delle applicazioni sono coloro che richiedono delle modalità molto semplici per attivare un numero predefinito di operazioni, senza avere nessuna competenza informatica. Esempi sono gli impiegati addetti agli sportelli di una banca o alle prenotazioni di una compagnia aerea. In questi casi un'operazione è invocata interattivamente, con uso di un terminale video: l'utente agisce selezionando una delle possibili scelte proposte (menu), e fornisce i valori degli argomenti riempendo campi di opportune "maschere". Per programmare agevolmente queste applicazioni sono utili alcuni strumenti che facilitino la definizione delle interfacce.

7.1.3 Controllo dei dati

Una caratteristica molto importante dei DBMS è il tipo di meccanismi offerti per garantire le seguenti proprietà di una base di dati: integrità, affidabilità e sicurezza.

Integrità

I DBMS prevedono dei meccanismi per controllare che i dati inseriti, o modificati, siano conformi alle definizioni date nello schema, in modo da garantire che la base di dati si trovi sempre in uno stato che rispetti i vincoli

dichiarati. Esempi di vincoli discussi in precedenza sono le chiavi primarie ed esterne.

Affidabilità

I DBMS devono disporre di meccanismi per proteggere i dati da malfunzionamenti hardware o software e da interferenze indesiderate dovute all'accesso contemporaneo ai dati da parte di più utenti.

Per quanto riguarda la protezione da malfunzionamenti, un DBMS prevede che le interazioni con la base di dati avvengano per mezzo di transazioni, cioè con un meccanismo che garantisce il buon esito delle operazioni delle applicazioni nel caso di funzionamento normale, e che esclude effetti parziali dovuti all'interruzione delle applicazioni per una qualsiasi ragione. Più precisamente vale la seguente definizione.

Definizione
Una transazione è un programma sequenziale costituito da un insieme di azioni di lettura e scrittura in memoria permanente e di elaborazioni di dati in memoria temporanea, con le seguenti proprietà:

- Atomicità: solo le transazioni che terminano normalmente fanno transitare la base di dati in un nuovo stato. Le transazioni che terminano prematuramente sono trattate dal sistema come se non fossero mai iniziate; pertanto eventuali loro effetti sulla base di dati sono annullati.
- Serializzabilità: l'effetto sulla base di dati dell'esecuzione contemporanea di più transazioni è equivalente ad una esecuzione seriale delle transazioni, cioè ad una esecuzione in cui le transazioni vengono eseguite una dopo l'altra in un qualche ordine.
- Persistenza: le modifiche sulla base di dati di una transazione terminata normalmente sono permanenti, cioè non sono alterabili da eventuali malfunzionamenti.

Una transazione può essere una semplice espressione in un linguaggio di interrogazione, oppure un programma in un linguaggio di programmazione che opera sulla base di dati. Un malfunzionamento è un evento a causa del quale la base di dati può trovarsi in uno stato scorretto.

Si distinguono tre tipi di malfunzionamenti: *fallimenti di transazioni*, *fallimenti di sistema* e *disastri*.

I *fallimenti di transazioni* sono interruzioni di transazioni che non comportano perdite di dati in memoria temporanea o permanente.

I fallimenti di transazioni sono dovuti a situazioni già previste nei programmi, il cui verificarsi comporta la terminazione prematura della transazione; oppure a situazioni non previste nei programmi, il cui verificarsi causa la terminazione prematura della transazione da parte del sistema. Esempi sono la violazione di vincoli di integrità o il tentativo di accesso a dati protetti.

I *fallimenti di sistema* sono interruzioni del suo funzionamento dovuti ad un'anomalia hardware o software dell'unità centrale o di una periferica, con conseguente interruzione di tutte le transazioni attive. Si assume che il contenuto della memoria permanente sopravviva, mentre si considera perso il contenuto della memoria temporanea.

Esempi tipici di questo genere di malfunzionamento sono l'interruzione dell'alimentazione elettrica del sistema o un errore del sistema operativo o del software di base.

I *disastri* sono malfunzionamenti che danneggiano la memoria permanente contenente la base di dati (ad es. rottura dei dischi magnetici).

Quando si verifica un malfunzionamento vengono attivate, in modo utomatico o semi-automatico, opportune procedure per garantire che la base di dati contenga soltanto quelle modifiche apportate dalle transazioni terminate con successo prima dell'occorrenza del malfunzionamento.

Per poter eseguire queste procedure un DBMS mantiene una copia di sicurezza della base di dati e tiene traccia di tutte le modifiche fatte sulla base di dati dal momento in cui è stata eseguita l'ultima copia di sicurezza. Grazie a questi dati ausiliari, quando si verifica un malfunzionamento il DBMS può ricostruire una versione corretta dei dati utilizzando l'ultima copia e rieseguendo tutte le operazioni che hanno modificato i dati e di cui ha mantenuto traccia.

Altra funzionalità di un DBMS è di consentire l'esecuzione simultanea di più transazioni, risolvendo il problema di farle funzionare correttamente, senza interferenze indesiderate quando esse operano sugli stessi dati (controllo della concorrenza). Il classico esempio di interferenza è quello che porta alla perdita di modifiche.

Esempio
Si supponga che Antonio e Giovanna condividano un conto corrente e che contemporaneamente facciano un prelievo e un versamento da sportelli diversi. Sia 350 il saldo, 400 la somma che Giovanna versa e 50 la somma che Antonio preleva. Supponiamo che sulla base di dati si verifichino i seguenti eventi, nell'ordine mostrato:

- il cassiere di Giovanna legge il saldo 350,
- il cassiere di Antonio legge il saldo 350,
- il cassiere di Giovanna modifica il saldo in 750,
- il cassiere di Antonio modifica il saldo in 300.

L'effetto dell'operazione di Giovanna è annullato da quello dell'operazione di Antonio e il saldo finale è di 300 (fine esempio).

Per evitare interferenze indesiderate il DBMS deve, invece, coordinare opportunamente l'esecuzione concorrente di un insieme di transazioni T1, …, Tn, intercalando opportunamente nel tempo le azioni sulla base di dati di ogni transazione, in modo che l'effetto dell'esecuzione sia quello ottenibile eseguendo le transazioni isolatamente, in un qualche ordine.

Un modo molto semplice di risolvere il problema sarebbe quello di eseguire le transazioni isolatamente, cioè in modo tale che, per ogni coppia di transazioni Ti e Tj, tutte le azioni di Ti precedono quelle di Tj, o viceversa (esecuzione seriale). Questa soluzione impedirebbe però ogni forma di concorrenza riducendo il DBMS ad un elaboratore seriale di transazioni.

Sicurezza

I DBMS prevedono meccanismi sia per controllare che solo persone autorizzate accedano ai dati, sia per restringere i dati accessibili e le operazioni che si possono fare su di essi.

Esempi del primo tipo sono la "identificazione" degli utenti autorizzati, con parole di riconoscimento, oppure la possibilità di proteggere i dati da furti mediante crittografia. Per mostrare esempi specifici dell'area base di dati, immaginiamo di disporre di dati riguardanti cittadini, fra cui il codice fiscale, dati anagrafici e il reddito. Eventuali restrizioni che si potrebbero imporre a categorie diverse di utenti sono:

- alcuni utenti non possono accedere a questi dati, ma solo ad altri presenti nella base di dati;

– alcuni utenti possono accedere ai dati, ma non possono modificarli.;
– alcuni utenti possono accedere solo ai dati che li riguardano, senza modificarli;
– alcuni nel caso precedente, ma si possono modificare solo i dati anagrafici;
– alcuni utenti possono accedere solo ai dati anagrafici, da alcuni uffici e in alcune ore del giorno;
– alcuni utenti possono applicare solo operazioni statistiche sul reddito, ma non possono accedere a dati singoli né fare modifiche.

Questa lista potrebbe continuare e diventare ancora più significativa se si considerassero più insiemi di dati in relazione logica, ma è sufficiente per dare un'idea della complessità e flessibilità di un meccanismo per garantire la sicurezza dei dati.

7.1.4 Amministrazione della base di dati

L'amministratore della base di dati (Data Base Administrator (DBA)) è una persona (o un gruppo di persone) che dopo aver partecipato allo studio di fattibilità per decidere l'impiego di un DBMS, ne seleziona uno, lo mette in funzione, lo mantiene in esercizio e segue ogni applicazione dalla progettazione all'impiego. In particolare, quindi, ha bisogno di strumenti per svolgere le seguenti attività:
– analisi dei requisiti di nuove applicazioni e progettazione, sviluppo e manutenzione di basi di dati e delle applicazioni che ne fanno uso;
– definizione degli schemi di basi di dati (logici, fisici ed esterni), delle autorizzazioni e modalità di accesso ai dati per ogni classe di utenti e delle politiche per la sicurezza dei dati;
– definizione delle procedure per il caricamento dei dati, la creazione di copie di sicurezza, il ripristino dei dati dopo malfunzionamenti di sistema o dei dischi;
– controllo del funzionamento del sistema per decidere eventuali riorganizzazioni della struttura logica e fisica dei dati al fine di migliorare le prestazioni delle applicazioni.

Un importante strumento previsto dai DBMS per l'amministrazione di basi di dati è il cosiddetto dizionario o catalogo dei dati, che contiene informazioni su ciò che è definito nella base di dati, su quali definizioni operano le applicazioni e su come sono memorizzati i dati.

7.1.5 Distribuzione dei dati

Un DBMS moderno deve garantire la possibilità di distribuire i dati gestiti dal sistema su più elaboratori collegati tra di loro da una rete locale o geografica, eventualmente replicando alcuni dati. La distribuzione dei dati su reti locali è molto utile per sostituire i costosi calcolatori di grandi dimensioni, tradizionalmente usati per i DBMS, con reti di calcolatori di costo più ridotto. La distribuzione dei dati su reti geografiche è utile ad aziende con più sedi per poter gestire in ogni sede i dati di interesse locale, mantenendo la possibilità di accedere a tutti i dati dell'organizzazione. Infine, entrambi i tipi di distribuzione, se combinati con la replicazione di alcuni dati, permettono di continuare ad operare sui dati anche quando un elaboratore sia fuori servizio. La distribuzione dei dati aumenta la complessità dei DBMS che devono essere in grado di gestire transazioni eseguite su nodi diversi della rete e garantire la coerenza di eventuali dati duplicati.

Da questa rapida presentazione delle funzionalità dei DBMS si dovrebbe intuire come essi siano dei prodotti complessi e costosi. Attualmente i DBMS relazionali sono disponibili su tutte le categorie di calcolatori con differenze che riguardano:

- la quantità di dati memorizzabili;
- il numero di utenti che possono accedere contemporaneamente ai dati;
- i meccanismi offerti per il controllo dei dati, in particolare per l'integrità e l'affidabilità;
- le tecniche per la gestione dei dati nelle memorie permanenti;
- i tipi di linguaggi per l'uso dei dati da parte degli specialisti;
- i tipi di prodotti per l'utente finale;
- le prestazioni, in particolare la complessità dell'ottimizzatore, cioè il modulo per la traduzione dei comandi SQL in accessi ai dati nelle memorie permanenti. Più sofisticato è questo modulo, maggiore è la velocità di risposta del sistema;
- gli strumenti per l'amministrazione della base di dati.

8 SISTEMI PER LA GESTIONE DI ARCHIVI

Il modello dei dati dei sistemi di archiviazione è molto semplice: consente di trattare insiemi indipendenti di dati, detti archivi, ma non associazioni fra di essi. Gli elementi di un archivio sono registrazioni, cioè dati strutturati in campi, o caratteri nel caso di archivi di testo. Pertanto mentre nei modelli dei dati dei DBMS è sempre presente un operatore per trovare un elemento di un insieme in relazione con un elemento di un altro insieme, nel caso degli archivi esistono solo operatori per agire su insieme alla volta.

Si possono distinguere due tipi di sistemi di archiviazione, che chiameremo *procedurale* e *dichiarativo*.

I sistemi di archiviazione procedurale consentono di trattare archivi solo usando un linguaggio di programmazione. E' destinato quindi a utenti con competenze informatiche specifiche.

I sistemi di archiviazione dichiarativi sono stati sviluppati con l'avvento dei calcolatori personali per consentire una facile memorizzazione, modifica e recupero di dati da parte di utenti non esperti. Questi sistemi offrono funzionalità simili ai sistemi per basi di dati, ma rimangono sostanzialmente diversi perché le similitudini riguardano solo le modalità di definizione e recupero dei dati e non le altre funzionalità che qualificano un sistema per basi di dati, quali la gestione di transazioni, dell'affidabilità e della concorrenza.

Come nei sistemi per basi di dati, nei sistemi di archiviazione dichiarativi è previsto sia un linguaggio per la definizione di archivi di registrazioni , di solito con interfacce grafiche per agevolarne l'uso, sia un linguaggio per recuperare dati che soddisfano criteri complessi. Nei sistemi più recenti si possono trattare anche dati multimediali, cioè registrazioni con campi di tipo testo, disegno, immagini, suono e sequenze video. Sono inoltre forniti strumenti per stampare i dati recuperati in formati opportuni e, in alcuni casi, per sviluppare semplici interfacce a menu.

9 SISTEMI PER LA GESTIONE DI TESTI

Un concetto che può essere confuso con le basi di dati è quello delle banche di dati (o banca dati). Per alcuni anni le due espressioni sono state considerate equivalenti (e alcune volte lo sono ancora ora), ma in effetti esistono fra di loro delle differenze sostanziali che proviamo a chiarire con un esempio.

Oggi nel mondo esistono numerose raccolte di riassunti di articoli apparsi in riviste specializzate e gestite da organismi internazionali. L'esempio più famoso è quello che riguarda le riviste di chimica, ma ne esistono anche altre di interesse nazionale come la banche di dati giuridici gestite dalla Cassazione, quelle delle Camere per registrare l'iter parlamentare delle proposte di legge, quelle sulle aziende che svolgono attività commerciali, quelle sui brevetti, quelle della polizia ecc. Queste raccolte sono consultabili interattivamente, e recentemente anche con Internet, per fare ricerche in base al nome di un autore oppure in base al contenuto del riassunto (vedremo meglio cosa vuol dire fra poco). Le informazioni non sono però organizzate in più insiemi in relazione fra loro, come accade nelle basi di dati, ma sono rappresentate sostanzialmente come insiemi di testi.

Altri aspetti particolari delle banche dati sono il fatto che non sono modificabili in linea e non sono utilizzabili per la gestione del sistema informativo di un'organizzazione, ma solo per raccogliere informazioni in forma testuale. Ad esempio, è senz'altro molto utile una banca dati bibliografica per effettuare ricerche di pubblicazioni su alcuni argomenti, ma essa è del tutto inutile per la gestione di una biblioteca, dove invece interessa una base di dati aggiornabile per trattare mediante transazioni l'acquisto dei libri, gli abbonamenti delle riviste, i prestiti ecc.

Concludendo questa breve premessa, per banca di dati intenderemo una raccolta di informazioni rappresentate in forma testuale e messe a disposizione di un gran pubblico di utenti per essere reperite specificando in modo parziale il loro contenuto.

Il problema della gestione di testi per consentire il recupero di quelli che contengono alcune informazioni, è stato affrontato nel settore disciplinare noto attualmente con il nome di "recupero dell'informazione" e sono stati sviluppati sistemi adatti a tale scopo. Ne esistono diversi in commercio, ma oggi i DBMS più grandi prevedono la possibilità di gestire basi di dati con campi di tipo testo offrendo funzionalità analoghe. Per la ricerca di informazioni disponibili su calcolatori di tutto il mondo collegati in rete Internet è disponibile il sistema AltaVista, mentre per limitare la ricerca a dati presenti su calcolatori italiani è disponibile il sistema Arianna.

Altra possibilità prevista per il prossimo futuro è la disponibilità di banche dati multimediali, ad esempio di immagini, sulle quali si possono fare ricerche per contenuto con una filosofia analoga a quella usata per i testi ed esistono già DBMS che consentono di gestire anche basi di dati multimediali con questa logica.

In questa sezione, dopo un'introduzione al problema della gestione di testi, che mette in evidenza le differenze esistenti fra questo e il problema della gestione di dati strutturati, verranno trattati gli aspetti principali che caratterizzano i sistemi per il recupero dell'informazione: a) il modo in cui si rappresenta il contenuto dei documenti; b) il criterio adottato per stabilire quali documenti recuperare per soddisfare una richiesta.

9.1 Preliminari

La necessità di gestire automaticamente grandi quantità di informazioni memorizzate in forma di testo ha giustificato un vasto lavoro di ricerca motivato inizialmente dalla consultazione di materiale bibliografico (libri, giornali e riviste) e di sommari di pubblicazioni scientifiche, ed estesosi successivamente entro i confini dell'automazione del lavoro d'ufficio (lettere, studi, relazioni), dove, secondo recenti stime, soltanto un terzo delle informazioni eterogenee trattate sono dati strutturati, mentre il resto è costituito da testi, immagini, voce. Nel seguito si parlerà in generale di documenti per riferirsi ad entità caratterizzate dalla seguente definizione.

Definizione
Un documento è un'entità che possiede una parte strutturata, chiamata profilo, e una parte di testo.

Il profilo contiene informazioni quali il nome dell'autore, il titolo, l'editore, la data e il luogo di pubblicazione, nel caso di un libro; l'autore e la data, nel caso di un rapporto; il mittente, il destinatario, la data e l'oggetto, nel caso di una lettera. I documenti, una volta archiviati, possono venir recuperati sia in base alle informazioni presenti nel profilo, sia in base al contenuto del testo. Mentre nel primo caso si utilizzano le tradizionali tecniche sviluppate per i sistemi di gestione di basi di dati, nel secondo caso si usano tecniche sviluppate espressamente per la gestione di testi. Poiché in questa sede interessa il recupero dei documenti in base al contenuto del testo, parlando di documento ci si riferirà al testo in esso presente, ignorando l'esistenza di dati strutturati.

Definizione
Un sistema per il recupero dell'informazione (Information Retrieval System (IRS)) è un sistema che gestisce raccolte di documenti al fine di recuperare documenti giudicati rilevanti dal sistema stesso rispetto alle richieste effettuate dagli utenti.

Quando una persona desidera trovare i documenti di una raccolta che contengono alcune informazioni, formula una richiesta e di solito ottiene in risposta sia documenti effettivamente utili, o *rilevanti*, che documenti inutili. La rilevanza di un documento non può essere garantita dal sistema per il recupero dell'informazione, ma può essere stabilita solo da colui che ha formulato la richiesta. Pertanto può accadere che documenti che l'utente considererebbe rilevanti alla sua richiesta non facciano parte dei documenti recuperati dal sistema. Cioè può accadere sia che il sistema giudichi rilevanti documenti considerati inutili dall'utente, sia che il sistema giudichi irrilevanti, e quindi non includa nella risposta all'interrogazione, documenti ritenuti viceversa rilevanti da parte dell'utente. Un sistema per il recupero dell'informazione cerca di limitare questi due inconvenienti che, in generale, non possono essere eliminati.

Definizione
Un documento è rilevante se soddisfa il bisogno d'informazione che l'utente ha espresso con la sua richiesta.

La presenza di documenti non rilevanti fra quelli recuperati, e la contemporanea assenza di documenti potenzialmente rilevanti, è dovuta sia alla difficoltà da parte di chi formula la richiesta di caratterizzare in modo univoco, ma sintetico, il contenuto dei documenti che desidera, sia alla difficoltà di rappresentare in modo completo il contenuto dei documenti.

Esempio
Si supponga di avere un insieme Impiegati di dati strutturati, con attributi Nome, Indirizzo, Codice, AnnoAssunzione, e Stipendio; per conoscere il nome e l'indirizzo degli impiegati assunti dopo il 1970 che guadagnano più di due milioni al mese, una possibile formulazione della richiesta è la seguente:

```
SELECT Nome, Indirizzo
FROM Impiegati
WHERE AnnoAssunzione ≥ 1970 AND Stipendio > 2.000.000
```

In risposta si ottengono i dati con il valore degli attributi che soddisfa esattamente la condizione specificata e quindi certamente rilevanti.

Si supponga invece di avere un insieme di documenti; per recuperare i documenti relativi all'uso dei calcolatori per lo sviluppo di progetti architettonici, sapendo che il termine CAD è sinonimo di progetto assistito dal calcolatore, una possibile formulazione della richiesta è la seguente:

SEARCH architett* AND (CAD OR (progetto AND calcolatore))
IN SENTENCE

dove "architett*" sta per qualsiasi parola che inizia con i caratteri "architett".

In risposta si ottengono documenti in cui le parole specificate nella richiesta, usate in un diverso contesto, assumono anche un significato differente da quello ad essi attribuito da chi ha formulato la richiesta. Ad esempio, fra i documenti recuperati potrebbe esserci quello contenente la seguente frase "… l'impiego del calcolatore per lo sviluppo di progetti architettonici riguarda il campo di applicazione dell'informatica conosciuto con il nome di CAD (Computer Aided Design), ovvero progettazione assistita dal calcolatore …", ma anche quello contenente la frase "… nell'affrontare il progetto dell'architettura di un calcolatore bisogna tener conto del settore di applicazione in cui verrà utilizzato …". D'altra parte, documenti concettualmente pertinenti alla richiesta d'informazione dell'utente potrebbero essere ignorati, con questa formulazione dell'interrogazione. Fra i documenti non recuperati potrebbe esserci quello contenente la seguente frase "… l'uso di computer nel disegno di componenti VLSI è una delle aree di sicuro interesse per la progettazione assistita dal calcolatore …" (fine esempio)

In Figura 11 sono mostrate le attività principali svolte nella gestione di documenti.

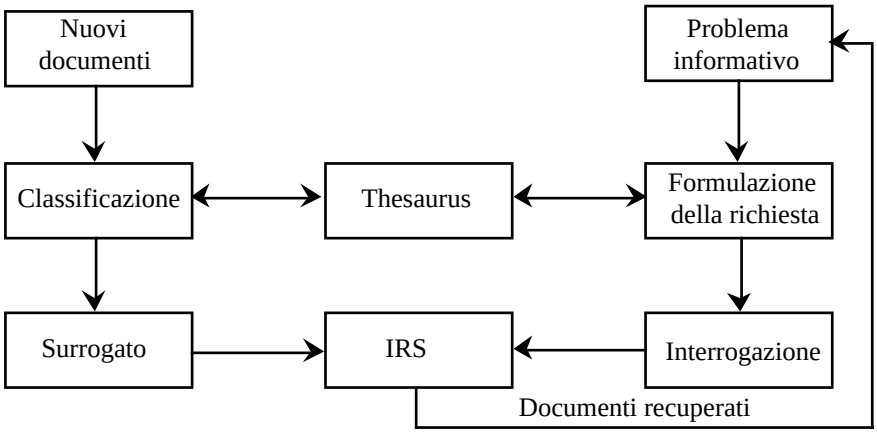

Figura 11 Le attività di gestione di documenti

Il *problema informativo* corrisponde ad un particolare bisogno di informazione dell'utente. Tramite un processo di rappresentazione, il problema informativo viene tradotto in una *richiesta* espressa nel linguaggio di interrogazione dell'IRS. Analogamente, dai documenti, tramite un altro processo di rappresentazione, spesso chiamato di *classificazione* o *indicizzazione*, si passa al *surrogato dei documenti*, cioè alla loro rappresentazione nell'IRS. Sia nella classificazione di un documento da parte di un esperto che nella formulazione della richiesta da parte di un utente può essere usato un vocabolario controllato organizzato in un *thesaurus*.

I metodi di rappresentazione dei documenti si possono separare in due categorie: quelli che danno una *rappresentazione diretta* del contenuto dei documenti e quelli che ne danno una *rappresentazione indiretta*. Nel primo caso il documento è rappresentato dalle parole in esso contenute; nel secondo caso, il documento è rappresentato da *termini di indicizzazione*, derivati manualmente o automaticamente, che ne descrivono in modo sintetico e completo il contenuto.

All'interno dell'IRS, l'esecuzione di una richiesta utente di una ricerca di documenti avviene confrontando la rappresentazione del contenuto dei documenti (surrogato) con la rappresentazione della richiesta utente (interrogazione). In questo processo di confronto, l'IRS adotta una particolare *tecnica di recupero* dei documenti, che serve per giudicare quali documenti sono rilevanti, e in che misura, rispetto all'interrogazione.

La presenza di documenti non rilevanti come risultato di una richiesta utente e, contemporaneamente, l'assenza di alcuni documenti rilevanti, è da imputare sia al processo di trasformazione dal problema informativo all'interrogazione (cioè, come un bisogno di informazione dell'utente viene

espresso nel linguaggio di interrogazione del sistema), sia al processo di trasformazione dal contenuto dei documenti al loro surrogato (cioè, il modo in cui viene effettuata la classificazione dei documenti).

Definizione
Si definisce tecnica di recupero (retrieval technique) di un IRS la tecnica adottata dal sistema per confrontare l'interrogazione utente con il surrogato dei documenti.

La tecnica di recupero adottata da un IRS, è il meccanismo interno del sistema che lo guida nel giudicare come rilevanti o non rilevanti i documenti di una raccolta, in rapporto ad una specifica interrogazione. Essa quindi determina l'efficacia di un sistema nel rispondere alle interrogazioni utente.

Come sarà mostrato più avanti, le tecniche di recupero sono di due tipi: per *corrispondenza esatta* (*exact match*) e per *similitudine* o *corrispondenza parziale* (*partial match*).

Le tecniche di recupero per corrispondenza esatta sono quelle basate sull'assunzione che le informazioni specificate nella richiesta siano esattamente contenute nella componente testuale del documento. Questa tecnica di recupero è usata in gran parte dei sistemi commerciali, ma presenta alcuni svantaggi:

a) molti documenti rilevanti sono ignorati, se il testo corrisponde solo parzialmente all'interrogazione;
b) i documenti ritrovati non sono ordinati per rilevanza rispetto all'interrogazione;
c) non è possibile tenere in considerazione l'importanza relativa di concetti sia nell'interrogazione che nei documenti;
d) la logica del linguaggio di interrogazione risulta spesso complicata; e) l'efficacia dipende dalla misura in cui le due rappresentazioni da confrontare siano basate o meno sullo stesso vocabolario.

Le tecniche di recupero per corrispondenza parziale sono invece basate sull'assunzione che le informazioni specificate nella richiesta possano essere contenute parzialmente nel documento e che i documenti ritrovati possano essere ordinati per valori decrescenti di rilevanza. Queste tecniche consentono una maggiore flessibilità, rispetto alle tecniche per corrispondenza esatta, e sono quelle su cui si concentra, attualmente, il maggiore sforzo di ricerca.

Per chiarire ulteriormente le caratteristiche generali dei sistemi per il

recupero dell'informazione, è utile confrontarle con quelle dei DBMS, esaminando i seguenti punti, riassunti in Tabella 1:

– modello dei dati: come si rappresentano le informazioni. Nei DBMS le informazioni si rappresentano come insiemi di dati strutturati e relazioni fra insiemi. Negli IRS le informazioni si rappresentano come insiemi di testi;
– richiesta: come si specifica ciò che si cerca. Nei DBMS l'utente descrive, in modo completo e preciso, ciò di cui ha bisogno; negli IRS non si specifica completamente il valore del testo di un documento, ma se ne specifica il contenuto mediante una "descrizione" abbreviata e pertanto soggettiva e incompleta;
– tecnica di recupero: come il sistema, in fase di ricerca, decide se un documento soddisfa la richiesta. Nei DBMS la scelta delle registrazioni da recuperare si basa sulla corrispondenza esatta fra quanto specificato nella richiesta e quanto in esse contenuto; negli IRS questa corrispondenza è in generale parziale perché basata su un criterio di similitudine che dipende da come si rappresenta il contenuto del documento, come verrà mostrato più avanti;
– risultato: cosa fornisce il sistema come risposta ad una richiesta. Nei DBMS vengono fornite solo le registrazioni che soddisfano la condizione di ricerca; negli IRS vengono forniti documenti probabilmente rilevanti: è compito dell'utente stabilire quali di essi siano davvero tali, sapendo che il sistema non garantisce che fra i documenti non recuperati non ne esistano di rilevanti. In questo contesto, "probabilmente rilevante" significa "rilevante", con un alto grado di probalilità, a giudizio del sistema.

	DBMS	*IRS*
richiesta	completa	incompleta
tecnica di recupero	corrispondenza esatta	corrispondenza parziale
risultato	registrazioni cercate	documenti probabilmente rilevanti

Tabella. 1. Confronto fra DBMS e IRS.

La presenza di documenti non rilevanti fra quelli recuperati viene detta *effetto rumore*, mentre il mancato recupero di documenti rilevanti viene chiamato *effetto silenzio*. Si tratta di due effetti negativi che caratterizzano un sistema per il recupero dell'informazione.

Dato un insieme di documenti e una richiesta, è possibile individuare quattro sottoinsiemi: l'insieme dei documenti correttamente recuperati in quanto rilevanti per la richiesta (A), quello dei documenti che pur non essendo rilevanti sono stati recuperati (B), l'insieme dei documenti giustamente omessi in quanto non rilevanti (C) e quello dei documenti non recuperati anche se rilevanti (D) (vedi Figura 12).

	Documenti Rilevanti	*Documenti Irrilevanti*
Documenti recuperati	A (corretti)	B (inesatti)
Documenti non recuperati	D (omessi)	C (da omettere)

Figura 12 Classificazione dei documenti

Per misurare l'efficacia di un sistema per il recupero dell'informazione si usano principalmente due parametri, chiamati *richiamo* (*recall*) e *precisione* (*precision*), i valori dei quali dipendono dal tipo di sistema, dalla raccolta dei documenti e dalla competenza di chi formula la richiesta.

Definizione
Il richiamo è il rapporto fra il numero di documenti rilevanti recuperati (A) e il totale dei documenti rilevanti archiviati (A + D).

Definizione
La precisione è il rapporto fra il numero di documenti rilevanti recuperati (A) e il totale dei documenti recuperati (A + B).

Il massimo valore sia per il richiamo che per la precisione è 1. Il richiamo misura la capacità del sistema di recuperare tutti i documenti rilevanti, mentre

la precisione misura la capacità del sistema di recuperare solo documenti rilevanti. Un sistema con precisione P < 1 ammette nelle risposte documenti non rilevanti, che l'utente provvederà semplicemente a scartare. Un sistema con richiamo R < 1 ammette che documenti rilevanti non siano reperiti.

Un sistema per il recupero dell'informazione è tanto più *efficace* quanto più alti sono il richiamo e la precisione.

Dopo queste precisazioni sulle caratteristiche generali dei sistemi per il recupero dell'informazione, si passa ora ad esaminare i principali aspetti che li distinguono:

– il modo in cui si rappresenta il contenuto dei documenti;
– il criterio adottato per stabilire quali documenti recuperare per soddisfare una richiesta.

9.2 Rappresentazione del contenuto dei documenti

La rappresentazione del contenuto dei documenti può essere diretta o indiretta. Con la rappresentazione diretta, un testo è rappresentato nella sua forma originaria come una sequenza di parole, eventualmente strutturate in frasi e in paragrafi. Ai fini della ricerca, vengono trascurate le parole contenute in una lista di parole da ignorare (lista di esclusione o stop list) — come articoli, preposizioni, congiunzioni, avverbi ecc. — ritenute poco rappresentative del contenuto di un documento.

La sequenza di parole di un testo, però, non sempre è una rappresentazione adeguata perché essa consente solo il recupero di testi con richieste che specificano una condizione sulle parole in essi presenti. Il tipo di richiesta più comune, invece, è quella in cui si tenta di caratterizzare con una condizione il contenuto concettuale del testo; ad esempio con la richiesta "trovare i documenti che trattano il problema dell'emigrazione", si vorrebbe avere fra i documenti rilevanti anche quello con titolo "Gli albanesi in Italia nel 1996", anche se ci sono poche parole in comune con quanto richiesto. Pertanto la rappresentazione diretta del contenuto di un documento non è in generale adeguata sebbene sia adottata dalla maggioranza dei sistemi commerciali, come vedremo più avanti.

Con la rappresentazione indiretta, ai fini delle ricerche, ad un testo è associato un insieme di *parole chiave* (*keywords*), semplici o composte, che ne descrivono in modo sintetico il contenuto. Ad esempio, a questa sezione potrebbero essere associate le seguenti parole chiavi: recupero dell'informazione e indicizzazione. L'operazione di attribuzione delle parole chiave ad un testo, denominata *classificazione* o *indicizzazione* (*indexing*), è di solito fatta manualmente da esperti, ma sono state studiate anche tecniche

automatiche basate su metodi statistici, come nel caso del sistema Smart.

9.2.1 Indicizzazione manuale

L'indicizzazione manuale può essere fatta usando parole estratte dal testo o termini controllati, o descrittori, estratti da un thesaurus preesistente.

Definizione
Un thesaurus è un insieme di termini, e di relazioni fra di essi, che costituiscono il lessico specialistico da usare per descrivere il contenuto dei documenti pubblicati in un ambito disciplinare.

Il thesaurus ha quindi un ruolo analogo a quello di un vocabolario di una lingua con la differenza che per i termini, oltre alla eventuale definizione, vengono indicate le relazioni che esistono fra di essi. Le relazioni possono essere di tre tipi: preferenza, gerarchia e affinità semantica.

Le relazioni di preferenza si usano per rimandi da termini non accettati a termini accettati e viceversa. Esse sono USA o VEDI e USATO PER. Ad esempio: Elaboratore VEDI Calcolatore; Calcolatore USATO PER Elaboratore, Calcolatrice, Stazione di lavoro.

Un semplice esempio di thesaurus è l'indice delle categorie merceologiche in cui sono classificati gli operatori economici riportati nelle pagine gialle. Per agevolare la ricerca, sono previsti i rimandi fra le categorie. Ad esempio in corrispondenza a "minicomputers" si trova un rimando ai termini "personal computers" e "elaboratori elettronici".

Le relazioni di gerarchia mettono in evidenza il rapporto specificità-generalità tra due termini; esse sono: termine più generale (broader term (BT)) e termine più specifico (narrower term (NT)). Ad esempio: Felini NT Gatti Leoni Tigri, Gatti BT Felini.

Le relazioni di affinità semantica si usano per collegare termini con significato affine o che esprimono concetti correlati; esse sono: termine correlato (related term (RT)) e sinonimi (synonymous term (ST)).

Ad esempio, In corrispondenza del termine "geometria" si potrebbe trovare:

- BT matematica
- NT geometria piana
  geometria solida
  geometria analitica
- RT algebra lineare

e in corrispondenza del termine CAD:

– BT applicazioni delcalcolatore
– NT CAD architettonico
  CAD di circuiti
– RT grafica al calcolatore
– ST computer aided design

L'indicizzazione manuale ha il vantaggio di permettere una rappresentazione indiretta del contenuto dei documeni con termini che evidenziano i concetti in essi trattati, ma può portare a rappresentazioni non accurate né consistenti se non è fatta da persone con una buona conoscenza sia dell'argomento trattato nel documento che delle caratteristiche della raccolta di documenti. Una rappresentazione è *accurata* quando viene fatta usando un numero adeguato di termini; in caso contrario si pregiudica il richiamo del sistema. Una rappresentazione è *consistente* se documenti che trattano lo stesso argomento vengono rappresentati, anche da persone diverse, con gli stessi termini; in caso contrario si pregiudica la precisione del sistema. In generale, comunque, con l'indicizzazione manuale è difficile garantire rappresentazioni accurate e consistenti.

9.2.2 Indicizzazione automatica

L'applicazione di tecniche di comprensione automatica del linguaggio naturale per l'estrazione dei concetti trattati in un testo, al fine di dare una rappresentazione indiretta del suo contenuto, sono ancora in uno stadio sperimentale. Per questa ragione l'indicizzazione automatica si basa su tecniche statistiche applicate alla rappresentazione diretta del contenuto dei documenti, partendo dal presupposto che la frequenza di occorrenza delle parole in un testo in linguaggio naturale sia correlata con l'importanza di queste parole nel rappresentare il suo contenuto. Se invece che un singolo documento si considera una raccolta di documenti, occorre che la rappresentazione di un documento sia tale da distinguerlo dagli altri. Pertanto per stabilire quali parole chiave scegliere nell'indicizzazione, si tiene conto anche di come esse siano distribuite nella raccolta; infatti se una parola appare con una frequenza alta in tutti i documenti, allora diminuisce la sua importanza ai fini dell'identificazione di un documento specifico. Si pensi alla parola "calcolatore" in una raccolta di testi di informatica.

Un semplice algoritmo per l'indicizzazione automatica di documenti consiste dei seguenti passi:

1. *Eliminazione delle parole comuni*. Il procedimento inizia con l'analisi delle parole che costituiscono i documenti per eliminare quelle più comuni contenute nella lista di esclusione, come preposizioni, articoli, congiunzioni, avverbi ecc. Ad esempio una lista di esclusione per testi in lingua italiana contiene parole tipo: a, abbastanza, ad, adesso, agli, al, alla, …, che, chi, ce, ci, ciò, cioè, …. Da esperimenti fatti su testi in lingua italiana risulta che circa il 50% delle parole usate sono particelle e quindi eliminandole si riduce mediamente la lunghezza di un testo alla metà.

   Quando i documenti sono brevi (notizie di agenzie, messaggi, leggi ecc.) e trattano temi molto specifici si considera il testo completo del documento, altrimenti si esamina il titolo unitamente ad un estratto del suo contenuto.

2. *Riduzione delle parole alla radice*. Si eliminano i suffissi delle parole riducendole alla radice per prescindere dalle diverse forme verbali di un verbo, dal singolare o plurale, ecc. I problemi che si presentano in questa fase dipendono dalla lingua usata nei documenti. Ad esempio per la lingua italiana i suffissi "are", "ere", "ire" devono essere eliminati dall'infinito di un verbo, ma non dalle parole "casolare" o "giardiniere" da cui bisogna togliere rispettivamente "e" e "iere".
   La scelta di rappresentare i documenti con solo le radici delle parole significative in essi contenute ha lo scopo di ridurre il numero di termini assegnati alla raccolta e di aumentare il richiamo, poiché più parole hanno la stessa radice. Infatti la stessa trasformazione viene eseguita sulle parole elencate nella richiesta e ciò porta a recuperare, oltre ai documenti contenenti parole uguali a quelle specificate, anche i documenti in cui si trovano una o più parole diverse dalle parole specificate, ma aventi la stessa radice e molto probabilmente significati affini.

3. *Scelta dei termini con alto potere discriminante*. Questo passo è suggerito da alcuni autori per eliminare quei termini con scarsa capacità di distinguere i documenti rilevanti da quelli che non lo sono. Di solito si eliminano, oltre alle parole comuni, anche quelle radici con alta o bassa frequenza di occorrenza nella raccolta. Infatti i termini con alti valori della frequenza di occorrenza non permettono di discriminare tra i documenti, per cui la loro eliminazione porta ad un aumento della precisione, mentre i termini con bassi valori della frequenza di occorrenza potrebbero essere utili a questo

scopo, tuttavia è probabile che non saranno utilizzati nella maggior parte delle interrogazioni.

I termini che si ottengono dopo l'eliminazione delle parole comuni, la riduzione delle restanti alla radice e l'esclusione dei termini con scarso "potere discriminante", sono assegnati ai documenti della raccolta.

Infine, l'uso del thesaurus è prevista anche in sistemi che adottano l'indicizzazione automatica, per sostituire termini estratti automaticamente con termini più specifici o più generali.

9.2.3 Indicizzazione con termini pesati

L'efficacia dell'indicizzazione aumenta se ai termini che caratterizzano un documento si assegna un peso che rifletta l'importanza del termine per il documento.

Se n sono i termini usati per l'indicizzazione, l'i-esimo documento della raccolta viene rappresentato dal vettore $Di = (T_{i1}, T_{i2}, \ldots, T_{in})$ dove $T_{ij} \geq 0$, con $1 \leq j \leq n$, è il peso del termine j nel documento i (il peso è zero se il termine non è presente).

Una raccolta di documenti si riduce così ad una matrice di termini, con tante righe quanti sono i documenti e tante colonne quanti sono sono i termini usati per l'indicizzazione.

Fra le funzioni proposte per il calcolo del peso di un termine, la più usata tiene conto sia della sua rappresentatività, considerando la sua frequenza di occorrenza in un documento, sia della capacità del termine di discriminare un documento dagli altri della raccolta, considerando la sua distribuzione nell'intera raccolta.

9.3 Memorizzazione dei surrogati dei documenti

Sia con la rappresentazione diretta che indiretta del contenuto dei documenti il surrogato di un documento è un insieme di parole estratte dal testo o da un vocabolario controllato, che chiameremo per semplicità termini della rappresentazione. L'insieme dei surrogati dei documenti di una raccolta D, usando l'insieme T di termini, può essere pensato come una relazione da D a T che chiameremo *relazione di rappresentazione*.

Per agevolare la ricerca dei documenti che soddisfano una richiesta è comodo memorizzare, oltre all'archivio dei documenti nella forma originale, non la relazione di rappresentazione ma una generalizzazione della sua

inversa, che chiameremo *indice* della raccolta, costituita da un insieme di coppie (T_i, I_i), dove I_i è un insieme di informazioni sui documenti descritti dal termine T_i. Cosa siano esattamente le coppie (T_i, I_i) dell'indice dipende da come si rappresenta il contenuto dei documenti e dal tipo di ricerca che si vuole agevolare.

Possibili valori per T_i, che costituiscono il cosiddetto *dizionario*, sono:

- le parole distinte presenti nei documenti (indicizzazione totale);
- le parole distinte presenti nei documenti che non fanno parte di una lista delle parole da ignorare, o le loro radici (indicizzazione incompleta);
- i termini controllati del thesaurus usato per l'indicizzazione dei documenti (indicizzazione controllata).

Le informazioni I_i associate ad un termine T_i possono essere di tre tipi:

1. *Informazioni sulle occorrenze delle parole nei documenti*, nel caso di testi nella rappresentazione diretta, oppure *informazioni sui termini di indicizzazione*, nel caso di testi nella rappresentazione indiretta.

   Nel primo caso, esempi di queste informazioni sono:

   - gli identificatori interni ID dei documenti contenenti almeno un'occorrenza del termine; gli identificatori sono assegnati automaticamente dal sistema ai documenti memorizzati sequenzialmente. Se un termine dell'indice è presente più volte in un documento, nell'indice si riporta una sola volta il riferimento a quel documento;
   - tante coppie (ID, zona) quante sono le zone del documento che contengono almeno un'occorrenza del termine. La dimensione di una zona può essere decisa al momento del caricamento dei dati, ma tipicamente si fa coincidere con un paragrafo. Se un termine dell'indice è presente più volte in una zona, nell'indice si riporta una sola volta il riferimento a quella zona. L'informazione sulla zona in cui è presente un termine semplifica la ricerca di documenti contenenti termini nella stessa zona;
   - tante coppie (ID, PT) quante sono le occorrenze dei termini nei documenti, dove PT è la posizione del termine nel documento ID, relativa all'inizio del testo. L'informazione sulla posizione del termine semplifica la ricerca di documenti contenenti termini in una relazione di

vicinanza (ad es., “presidente” e “repubblica” separate al più da una parola);
- tante quadruple (ID, NP, NF, PT) quante sono le occorrenze dei termini nei documenti, ma invece di riportare la posizione dell’occorrenza del termine nel documento, si riporta il numero NP del paragrafo, il numero NF della frase nel paragrafo e PT la posizione del termine nella frase. Queste informazioni semplificano ulteriormente la ricerca di documenti contenenti termini in una relazione più complessa di quella di vicinanza.

Nel caso di indice per termini di indicizzazione, le informazioni associate ad un termine sono tante coppie (ID, P) quanti sono i documenti caratterizzati dal termine, dove P è il peso del termine per il documento ID.

2. *Informazioni statistiche.* Tipicamente il numero dei riferimenti associati ad un termine, il numero di volte che un termine appare nel documento (frequenza del termine) e il numero dei documenti in cui appare il termine (frequenza nei documenti).

3. *Informazioni sui sinonimi.* Ad ogni valore dell’indice si associa l’insieme dei riferimenti ai suoi sinonimi.

Esempio
Si considerino i seguenti documenti dei quali si riporta la frase più significativa contenuta nel riassunto:

- DOC-1: … nei sistemi per la gestione di basi di dati fondati sul modello relazionale il linguaggio d’interrogazione …
- DOC-2: … per basi di dati distribuite si adotta il modello di dati relazionale …
- DOC-3: … i DBMS offrono linguaggi diversi per l’accesso ai dati …
- DOC-4: … il linguaggio di definizione dei dati dei DBMS …
- DOC-5: … l’elaborazione delle interrogazioni nei DBMS con hardware specializzato …

Si supponga che i surrogati siano definiti usando i seguenti termini estratti dai riassunti: basi di dati (considerato come un unico termine), modello, distribuito, dati, linguaggio, interrogazione, DBMS, hardware. Si supponga inoltre che i termini al plurale siano equivalenti a quelli al singolare e quelli al

maschile siano equivalenti a quelli al femminile. L'indice che si ottiene è riportato in Figura 13. (fine esempio)

Termine	Frequenza nei documenti	Riferimenti
base di dati	2	DOC-1, DOC-2
dato	3	DOC-2, DOC-3, DOC-4
dbms	3	DOC-3, DOC-4, DOC-5
distribuito	2	DOC-2, DOC-5
hardware	1	DOC-5
interrogazione	3	DOC-1, DOC-3, DOC-5
linguaggio	3	DOC-1, DOC-3, DOC-4
modello	2	DOC-1, DOC-2

Figura 13 Esempio di indice della raccolta.

La presenza dell'indice richiede però una notevole occupazione di memoria, che nei sistemi commerciali, con indicizzazione totale, varia tipicamente dal 60% (eliminando meno di 100 parole comuni) al 100% (senza l'eliminazione di parole comuni) della dimensione dell'archivio di testo. L'inserimento di nuovi documenti, o una modifica del testo, comporta inoltre un pesante aggiornamento dell'indice. L'efficienza del reperimento e la facilità di inserimento sono antagonisti nelle tecniche basate su indici, ma nonostante questi svantaggi, questa soluzione è adottata dai sistemi commerciali. Vediamone alcuni:

– ALEPH, distribuito dalla Aleph Yissum ltd, adotta due indici, uno per l'indicizzazione controllata, basata sui termini di un thesaurus da usare nel processo di indicizzazione manuale dei documenti, e l'altro per l'indicizzazione incompleta. In entrambi i casi ad un termine si associa solo l'identificatore interno dei documenti e quindi non sono consentite ricerche con parole in un contesto.
– FOLIO VIEWS 1.0, distribuito dalla Folio Corp., adotta l'indicizzazione totale e le zone del testo per costruire le liste dei riferimenti. Il sistema riesce ad garantire tempi di ricerca bassi con una particolare tecnica di

compressione dei dati applicata sia ai testi che all'indice.

- CDS/ISIS, distribuito dall'Unesco, adotta l'indicizzazione totale o incompleta, a scelta dell'utente, con le parole dell'indice estratte automaticamente dal testo oppure segnalate nel testo racchiuse fra simboli speciali; le informazioni associate alle parole sono le posizioni delle parole nel testo.
- STAIRS, distribuito dalla IBM, che è stato il primo esempio di sistema commerciale per il recupero dell'informazione, adotta l'indicizzazione incompleta e come posizione delle parole nel testo usa la combinazione del numero del paragrafo, numero di frase e posizione nella frase. Le parole dell'indice possono essere poste dall'utente in relazione di sinonimia ed hanno associate delle informazioni statistiche per consentire di ordinare i documenti che soddisfano una richiesta, secondo un criterio specificato dall'utente.
- FULCRUM FUL/TEXT, distribuito dalla DATAMAT, adotta l'indicizzazione incompleta e le posizioni delle parole nel testo per costruire le liste dei riferimenti.
- DBT, distribuito dall'Istituto di Linguistica Computazionale del CNR di Pisa, adotta l'indicizzazione totale.

9.4 Tipi di richieste e tecniche di recupero

9.4.1 Recupero per corrispondenza esatta

Sia che si adotti la rappresentazione diretta che quella indiretta del contenuto dei documenti, la richiesta è formulata specificando una condizione sui termini che devono essere presenti nel surrogato del testo da recuperare. Una condizione può essere semplice, cioè riguardare un solo termine, o composta, cioè una composizione di condizioni semplici con gli operatori logici AND, OR, NOT. Di solito le lettere minuscole e maiuscole vengono trattate alla stessa stregua; si possono specificare parti di parole — inizio (ad es. "inf*", dove `*' sta per una qualsiasi sequenza di caratteri, anche vuota), fine (ad es. "*atica") o parte mediana (ad es. "*mat*") — oppure parole da flettere (ad es. "andare@"), cioè una parola che sta per tutte le sue forme grammaticalmente derivate (ad es. andare, andavo, andato, andrei, vado, ma non andamento).

Quando si adotta la rappresentazione diretta del contenuto dei documenti, si possono formulare condizioni del tipo:

A??B

per richiedere tutti i testi nei quali si trova la stringa di caratteri A seguita da due caratteri qualsiasi e dalla stringa di caratteri B.

A * B

per richiedere tutti i testi nei quali si trova la stringa di caratteri A seguita da un numero non specificato di caratteri qualsiasi e dalla stringa di caratteri B.

AB

per richiedere tutti i testi nei quali la parola A e la parola B si trovano l'una di seguito all'altra.

A NEAR n B

per richiedere tutti i testi nei quali la parola A si trova prima della parola B e fra di esse sono presenti al più n parole intermedie.

(A, B) n WORD APART

per richiedere tutti i testi nei quali si trovano le due parole A e B, in qualunque ordine, separate da al più n parole intermedie.

(A, B) NEAR

per richiedere tutti i testi nei quali le due parole A e B si trovano nella stessa frase.

(A, B) IN PARAGRAPH oppure
(A, B) IN SENTENCE

per richiedere tutti i testi nei quali le due parole A e B si trovano nello stesso paragrafo o nella stessa frase.

(A, B, C, D) %n (threshold OR)

per richiedere tutti i testi nei quali si trovano almeno n fra le parole elencate.

La ricerca dei documenti che soddisfano questo tipo di condizioni è agevolata quando il sistema prevede la gestione dell'inversa della relazione di indicizzazione con un indice in cui sono memorizzate le informazioni sull'occorrenza delle parole nei testi.

Esempio
Supponendo di voler ricercare tutti i documenti che contengono la parola "gestione" seguita da "dati" separate al massimo da una parola, la condizione di ricerca si può formulare nel seguente modo:

"gestione" NEAR 1 "dati"

Documenti contenenti "gestione dei dati" saranno considerati corrispondenti all'interrogazione, mentre quelli contenenti "gestione automatica dei dati" non verranno recuperati.

Supponendo invece di voler ricercare i documenti che parlano di linguaggi d'interrogazione si può formulare la richiesta nel seguente modo:

("linguagg*", "interrogazione") IN SENTENCE

dove il simbolo "*" si usa per recuperare documenti contenenti la parola "linguaggio" sia al plurale che al singolare. Il testo dei documenti recuperati può contenere la frase "Il linguaggio d'interrogazione di un DBMS relazionale…", oppure la frase "… la formulazione dell'interrogazione nei diversi linguaggi esaminati …". (fine esempio)

Il recupero per corrispondenza esatta ha due svantaggi.

Il primo è che documenti attinenti all'argomento, ma privi di termini nella relazione specificate, non sono recuperati e, viceversa, è possibile che siano recuperati documenti contenenti i termini nella relazione specificata ma che in realtà non hanno niente in comune con l'argomento a cui ci si interessa. Questi problemi si presentano nei sistemi che usano sia una rappresentazione diretta che indiretta del contenuto dei documenti. Nel primo caso ciò è dovuto al fatto che non è ovvio stabilire quali parole devono essere contenute nel testo affinché esso possa essere considerato rilevante, perché esistono parole diverse per esprimere lo stesso concetto. Nel secondo caso, invece, i problemi sono dovuti a indicizzazioni non accurate o non consistenti.

Il secondo svantaggio è che il recupero basato sulla coincidenza fra quanto espresso nella richiesta e quanto contenuto nella rappresentazione del testo

trascura i documenti la cui rappresentazione corrisponde solo parzialmente alla richiesta, ma che trattano ugualmente l'argomento voluto. Dal momento che in un sistema per il recupero dell'informazione l'utente desidera conoscere quali testi si devono consultare se si è interessati ad un particolare argomento, il sistema deve essere in grado di recuperare anche documenti che trattano in misura diversa, secondo un qualche criterio, l'argomento descritto nella richiesta.

9.4.2 Recupero per similitudine

Le richieste vengono di solito formulate elencando alcuni termini che si ritiene descrivano il contenuto dei testi voluti, termini eventualmente scelti con l'aiuto di un thesaurus, se esso è una delle componenti del sistema. Se il sistema prevede un thesaurus, questo può essere anche usato per sostituire un termine con un suo sinonimo usato per indicizzare i documenti oppure per sostituire termini troppo specialistici, e quindi poco frequenti, con termini più generali, per ridurre il fenomeno del silenzio. Per decidere se un documento debba essere recuperato, il sistema fa una valutazione del grado di similitudine dei documenti presenti con la descrizione di quelli richiesti.

Il modo più semplice per valutare il grado di similitudine è di contare quanti termini della richiesta sono presenti nel documento. Un altro modo è di sommare il numero delle occorrenze nel documento di ogni termine della richiesta. In generale, come un documento è rappresentato dal vettore $D_i = (T_{i1}, T_{i2}, \ldots, T_{in})$, così la richiesta viene rappresentata con un vettore $Q = (q_1, q_2, \ldots, q_n)$ nel quale ogni q_i vale 1 se il termine corrispondente è stato specificato nell'interrogazione, zero altrimenti, oppure, se si usano termini pesati, q_i vale il peso del termine (si noti che la richiesta non contiene operatori logici come nel caso di recupero per corrispondenza esatta). Durante la ricerca, si calcola la similitudine fra il vettore dell'interrogazione e il vettore di ogni documento come il coseno dell'angolo fra i due vettori e si recuperano tutti i documenti con una similitudine superiore ad un valore di soglia stabilito dall'utente. Se i documenti che dovrebbero essere recuperati sono molti, è utile ordinarli secondo i valori di similitudine calcolati e restituire all'utente solo quelli che occupano i primi posti nell'ordinamento (ranking) (di solito i primi dieci o venti).

9.4.3 Retroazione sulla rilevanza

Il richiamo e la precisione possono essere migliorati se, in fase di recupero, si applica il processo noto come retroazione sulla rilevanza (relevance feedback), secondo il quale l'utente, con l'aiuto del sistema, riformula la richiesta in base ai documenti recuperati in precedenza. Esperimenti hanno dimostrato che il miglioramento nella precisione, rispetto ad un processo di ricerca senza retroazione sulla rilevanza, può arrivare fino al 90%.

9.5 Osservazioni

Sono stati effettuati numerosi esperimenti per valutare le prestazioni dei sistemi che adottano l'indicizzazione automatica. Esperimenti eseguiti su piccole collezioni (meno di 1.000 documenti) hanno mostrato che non sempre l'indicizzazione manuale, eseguita da esperti servendosi di un vocabolario controllato, porta a risultati migliori dell'indicizzazione automatica, totale o incompleta. Questi risultati non sono confermati da alcuni autori che ne riportano altri, di esperimenti eseguiti su collezioni di circa 40.000 documenti, dai quali risulta che le prestazioni dei sistemi con indicizzazione automatica, totale o incompleta, non sono soddisfacenti, avendo un richiamo di circa il 20%, nel caso di precisione molto alta (circa 75%).

Per quanto riguarda l'indicizzazione automatica, gli esperimenti finora eseguiti su informazioni bibliografiche non sono riusciti a dimostrare un sostanziale miglioramento nelle prestazioni con l'uso di un thesaurus in cui vengano specificate relazioni complesse. Il grado di efficienza raggiunto con l'estrazione di singoli termini, riducendo le parole significative alla radice, è soddisfacente, mentre l'uso del thesaurus si è dimostrato utile soprattutto per quanto riguarda le relazioni di sinonimia.

Le tecniche di recupero dei documenti per corrispondenza parziale, basate sul calcolo di un coefficiente di similitudine di un documento rispetto alla richiesta, sono da preferire alle tecniche per corrispondenza esatta. La migliore strategia di ordinamento dei documenti è risultata quella basata sul calcolo della correlazione del coseno, assegnando ai termini associati ad un documento un peso pari alla loro frequenza nel documento e ai termini nella richiesta un peso pari all'inversa della loro frequenza nella raccolta, in modo da tener conto sia dell'importanza dei termini nel distinguere fra loro i documenti, sia della loro importanza nel descrivere il contenuto di un particolare documento.

L'impiego di un indice, con indicizzazione incompleta, è la soluzione

preferita negli attuali sistemi commerciali, alcuni dei quali sono stati esaminati a titolo di esempio. Le richieste che si possono formulare sono di solito di tipo booleano, o contestuale, e la ricerca si basa quindi esclusivamente sulla corrispondenza esatta.

10 CONCLUSIONI

Sono state presentate le caratteristiche principali dei sistemi per l'archiviazione e il recupero dell'informazione, in particolare dei sistemi per la gestione di basi di dati, archivi e testi.

Attualmente questi sistemi sono disponibili con funzionalità diverse e con un grado diverso di integrazione di strumenti per la produttività individuale. I sistemi relazionali sono certamente quelli più interessanti quando interessa trattare basi di dati e possono essere usati per operazioni di ricerche anche da utenti non esperti. Quando tutte le funzionalità di questi sistemi non sono necessarie, sistemi di facile uso sono anche quelli di archiviazione di tipo dichiarativo, che hanno fatto grandi progressi nel fornire linguaggi semplici e intuitivi, diventando così accessibili a categorie sempre più ampie di utenti. L'interesse per questi prodotti diventerà ancora maggiore con il diffondersi di strumenti per lo sviluppo di applicazioni che trattano dati multimediali.

Per quanto riguarda i sistemi per il recupero dell'informazione rappresentata in forma testuale, sono stati esaminati due tipi di sistemi: quelli più comuni, basati sulla rappresentazione diretta del testo, e quelli più specialistici, basati sulla rappresentazione indiretta del testo, dove si cerca di estrarre dal testo i concetti fondamentali in esso contenuti. I due problemi principali che caratterizzano questo secondo tipo di sistemi sono come scegliere i termini da usare nell'indicizzazione dei documenti e come stabilire se un documento è rilevante per una richiesta.

Note bibliografiche

Gli argomenti trattati sono oggetto di numerosi testi specifici, scritti di solito in lingua inglese. In queste note si citano solo quelli relativi ad alcune opere in italiano, per andare incontro alle esigenze di coloro che vorrebbero approfondire gli argomenti senza dover prima approfondire le loro conoscenze sulla lingua inglese.

I modelli dei dati, la progettazione di basi di dati e i sistemi per la gestione di basi di dati, anche multimediali, sono trattati diffusamente in [Albano 97]. Il problema della progettazione concettuale di basi di dati è trattato in [Batini

85]. Per un'introduzione non specialistica ai sistemi per il recupero dell'informazione si veda anche [Baldacci 88], dove è presentato il sistema ATLAS per l'automazione delle biblioteche, realizzato con la tecnologia delle basi di dati. Per coloro interessati a sapere non solo cosa si può fare con i sistemi per l'archiviazione e recupero dell'informazione, ma anche come sono fatti questi sistemi, si segnala [Albano 92].

Bibliografia

Albano A., G. Ghelli e R. Orsini [97], *Basi di dati relazionali e ad oggetti: modelli dei dati, linguaggi e sistemi*, Zanichelli, Bologna, 1997.

Albano A. [92], *Basi di dati: strutture e algoritmi*, Addison-Wesley Masson, Milano, 1992.

Baldacci M.B. [88], *Rappresentazione e ricerca delle informazioni*, La Nuova Italia Scientifica, Roma, 1988.

Batini C., G. De Petra, M. Lenzerini e G. Santucci, *La progettazione concettuale dei dati*, Franco Angeli, Milano, 1985.