

Il modello Relazionale

1 ** RIPASSO

le 7 Virtù del DBMS

persistenza (dei dati nel tempo)

affidabilità (backup, recovery e meccanismi di sicurezza)

volume (grandi moli di dati)

condivisione (dei dati tra più applicazioni e quindi controllo della concorrenza)

riservatezza (controllo dell'accesso ai dati e permessi differenziati)

efficienza (rispondere in tempi ragionevoli)

efficacia (non usare risorse infinite)

(non riscrivere ogni volta gli stessi algoritmi più o meno ottimizzati)

Transazionalità

atomicità

concorrenza

persistenza

2 ** INTEGRAZIONE

* Livelli di astrazione ed indipendenza dei dati

Equivalenza espressiva di diversi modelli

I "banali dettagli implementativi"

Modi diversi di vedere la stessa struttura

Livelli di astrazione

Livello fisico (o interno)

Livello logico (aggiunta del significato al livello fisico, senza dettagli implementativi)
utilizzato dai programmi
indipendenti dalla struttura fisica

Livello esterno (altri modi per vedere un unico livello logico)

Livello concettuale

astrae dai dettagli del livello logico cercando di modellare la realtà

Esempio di livello fisico

Struct in C

Data Division in COBOL

Scheda per lettura ottica (tipo modulo INPS)

Esempi di livelli

Esempio di livello logico

Archivio anagrafico dell'ufficio del personale di un'azienda

Esempio di livello esterno

Organigramma di un'azienda partendo dall'archivio dell'ufficio del personale

Esempio di livello concettuale

Modello delle entità e delle loro relazioni all'interno di un'azienda astratta

Indipendenza tra livelli

Indipendenza fisica

Indipendenza logica

Modelli logici

Gerarchico (da approfondire il concetto)

Reticolare (da approfondire il concetto)

Relazionale

Ad Oggetti

Modello ad oggetti

Oggetto

Attributi

Metodi

Gerarchia ed ereditarietà

3 Modello relazionale

Storia

Proposto da E. F. Codd nel 1970 per favorire l'indipendenza dei dati

Disponibile in DBMS reali nel 1981 (non è stato facile implementativamente coniugare l'indipendenza con efficienza e affidabilità!)

Si basa sul concetto matematico di relazione (con una variante per renderle più "usabili")

Le relazioni hanno come naturale rappresentazione le tabelle

3.1 Relazioni e tabelle

Accezioni di Relazione

matematica: come nella teoria degli insiemi

informatica: relationship tradotto anche con associazione o correlazione

relazione secondo il modello relazionale dei dati

3.1.1 Relazione matematica

Esempio semplice di relazione a due domini

• $D1 = \{a, b\}$

• $D2 = \{x, y, z\}$

• **prodotto cartesiano** $D1 \times D2$

• **relazione** $r \subseteq D1 \times D2$

Definizione più generale di Relazione

Siano $D1, \dots, Dn$ (n insiemi anche non distinti)

Definiamo **prodotto cartesiano** $D1 \times \dots \times Dn$:

l'insieme di tutte le n -uple $(d1, \dots, dn)$ tali che $d1 \in D1, \dots, dn \in Dn$

Definiamo **relazione matematica** su $D1, \dots, Dn$:

un sottoinsieme di $D1 \times \dots \times Dn$.

$D1, \dots, Dn$ sono definiti come i **domini** della relazione

Una relazione matematica è un insieme di n -uple ordinate:

$(d1, \dots, dn)$ tali che $d1 \in D1, \dots, dn \in Dn$

Una relazione matematica è un insieme per il quale :

• non c'è ordinamento fra le n -uple

• le n -uple sono distinte

• ciascuna n -upla è ordinata: l' i -esimo valore proviene dall' i -esimo dominio

• la struttura è quindi detta **posizionale**

3.1.2 Relazione secondo il modello "relazionale"

Nel modello relazionale la struttura delle relazioni è non posizionale, ed ad ogni dominio della n -upla viene assegnato un nome che lo identifica in modo univoco all'interno della relazione.

Il dominio viene quindi chiamato "attributo".

3.1.3 Relazioni – Relationships

In informatica il termine Relationship indica una relazione tra due o più entità (da questo nasce poi un pericoloso rischio di confusione sul termine relazione in italiano)

La relationship è un concetto fondamentale in tutto ciò che è trattamento e rappresentazione della informazione in quanto la realtà è intrinsecamente composta da entità (oggetti) e dalle interazioni tra tali entità (come vedremo nel sistema di modellazione “entità-relazioni”)

3.1.4 Tabelle

Tabella è una struttura bidimensionale dove ogni colonna è caratterizzata da un nome univoco, ed ogni riga rappresenta una n-upla

Tutte le relazioni matematiche possono essere rappresentate come tabelle
Non tutte le tabelle sono rappresentazioni di relazioni matematiche

Una tabella rappresenta una relazione se
le righe sono diverse fra loro
le intestazioni delle colonne sono diverse tra loro
i valori di ogni colonna sono fra loro omogenei

3.1.5 Rappresentazione delle relationship

Nel modello gerarchico ed in quello reticolare la relationship viene espressa tramite i puntatori che da un “record” permettono di referenziarne un altro (della stessa o di un'altra tabella)

Nel modello relazionale la relationship viene espressa tramite l'uguaglianza dei valori tra un sottoinsieme definito degli attributi di due “record” di due relazioni (siano esse anche eventualmente la stessa)

3.2 Analisi di dettaglio del modello Relazionale

Il modello relazionale è basato su una teoria matematica rigorosa, mentre i modelli precedenti erano il risultato dell'evoluzione implementativa

Concetti centrali del modello Relazionale

- Relazione
- Attributo
- Record o Tupla (Ennupla)

3.2.1 Definizioni matematiche

Schema di una relazione:

$R(A_1, \dots, A_n)$

Schema di base di dati:

$$R = \{R_1(X_1), \dots, R_k(X_k)\}$$

Base di dati : insieme di relazioni

Ennupla (in inglese T-upla) :

funzione che associa ad un insieme di attributi X una serie di valori appartenenti ai singoli domini dei vari attributi

oppure anche insieme di attributi valorizzati secondo i vincoli dei domini corrispondenti

valore di un attributo della ennupla:

il valore dell'attributo A per la ennupla t si indica con $t[A]$

Relazione (o istanza di una relazione)

dato uno schema $R(X)$ si dice relazione un insieme r di ennuple appartenenti ad X

Base di Dati (o istanza di base di dati)

dato un insieme R di schemi $R_i(X_i)$ indicabile con $R = \{R_1(X_1), \dots, R_n(X_n)\}$ si definisce Base di Dati un insieme r composto da r_i relazioni su R_i , espresso matematicamente con $r = \{r_1, \dots, r_n\}$

3.2.2 Considerazioni sul modello

Vantaggi del modello relazionale:

- indipendenza dall'implementazione fisica
- vengono rappresentate solo le informazioni rilevanti dal punto di vista dell'applicazione
- i programmi e gli utenti ragionano sugli stessi dati
- i dati sono più facilmente portabili da un sistema ad un altro
- si supera la direzionalità dei puntatori dei modelli gerarchici e reticolari

Svantaggi:

- i puntatori sono più performanti nella navigazione attraverso la base di dati

3.2.3 Attributi

Tipizzazione degli attributi

Esempi di tipi

Numero

Intero

Decimal

Float

Caratteri

Singolo carattere

Array di caratteri

Stringa di caratteri

Tempo
Date
Time
Timestamp
Altri tipi
Booleani
CLOB / BLOB
Tipi composti ed "User Defined"

3.2.4 L'informazione incompleta - NULL

Cosa significa non avere il valore di un attributo

- non esiste quel valore
- il valore esiste ma non è noto
- il valore non è rappresentabile con quella struttura dati (pi-greco)
- etc..

Perché è necessario avere un valore NULL

Non conviene usare valori del dominio come "Jolly" non utilizzati perché:

- i valori Jolly potrebbero diventare non più non utilizzati
- è sempre necessario interpretare il valore secondo una codifica non esplicita

Non sempre ad un attributo A di una relazione R può essere concesso di assumere il valore NULL: è necessario definire dei "vincoli" sul modello

3.2.5 I Vincoli

Definizione di vincolo:

restrizione imposta alle possibili ennuple di uno schema di basi di dati affinché tali tuple modellino in modo corretto una realtà specifica

Cioè i vincoli rappresentano i confini imposti dalla realtà alle infinite combinazioni dei dati insiemistici e matematici.

Oppure anche un vincolo è una proprietà che deve essere soddisfatta dalle istanze che rappresentano informazioni corrette per l'applicazione

E' sempre quindi relativo ad un'applicazione, non ad un dominio del possibile matematico.

I vincoli sono di diversi tipi :

- intrarelazionali (relativi ai valori di una singola ennupla o del singolo record)
 - di dominio (che riguardano i valori ammissibili per un certo attributo)
 - di ennupla (che legano tra loro i valo i dei vari domini di una ennupla)
- relazionali (che coinvolgono una relazione nella sua interezza)
- interrelazionali (cioè tra ennuple di diverse relazioni)

I vincoli intrarelazionali hanno la caratteristica di poter essere valutati considerando soltanto la singola ennupla, mentre quelli interrelazionali richiedono la valutazione di più relazioni contemporaneamente.

I vincoli relazionali hanno invece la caratteristica di riguardare una singola relazione, ma

coinvolgono tutte le ennuple di tale relazione, possono essere visti come un caso degenero di interrelazionale.

Ma pur essendo un caso "degenero" hanno un'importanza particolare proprio nel modello relazionale per la definizione delle CHIAVI delle relazioni

3.2.6 Chiavi ed identificazione delle ennuple

Definizione intuitiva di chiave di una relazione:

insieme di attributi che identifica in modo univoco le ennuple di una relazione

Definizione matematica di chiave:

- sia data una relazione r
- un insieme K di attributi è detto superchiave per r se r non contiene due ennuple distinte t_1 e t_2 per le quali $t_1[K] = t_2[K]$
- K è detto chiave per r se non esiste un sottoinsieme K_i di K che sia a sua volta superchiave di r (cioè K è una superchiave minimale di r)

Spesso si rischia analizzando solo i dati di esempio di una relazione di identificare come chiave un set di attributi che in realtà (cioè nel mondo reale che quello schema di relazioni vuole modellare) non ha tale caratteristica.

ATTENZIONE: nei DBMS relazionali non è necessario che per ogni tabella (relazione) sia definita una chiave, ma in tal caso non sarà sempre possibile operare sulla singola ennupla in quanto potrà capitare di avere in una tabella una serie di ennuple tra loro indistinguibili. Questo non è di solito un bene, ma bisogna vedere cosa in realtà quello schema modella.

3.2.7 Esempi di rappresentazione di informazioni nel modello relazionale

Codifica delle informazioni (tabelle di trascodifica)

Strutture nidificate (o master-detail)

Strutture ricorsive (gerarchia, albero, grafo comunque connesso)