Modello Relazionale

Oreste Signore
(Oreste.Signore@cnuce.cnr.it)

Contenuto

- Definizione
- Il concetto di chiave
  - Chiave primaria
  - Chiave esterna
- L' algebra relazionale
  - Le operazioni
  - Le operazioni "speciali"
- La normalizzazione
- Trasformazione da concettuale a relazionale

Definizione di Relazione

- Dati gli insiemi D_1, D_2, …., D_n (non necessariamente distinti), R(D_1, D_2, …., D_n) è una relazione su questi insiemi se è un insieme di n-ple ordinate (d_1, d_2, …., d_n) tale che $d_1 \in D_1$, $d_2 \in D_2$ e così via.
- $R(D_1, D_2, ..., D_n) \subseteq D_1 \otimes D_2 \otimes ..... \otimes D_n$
- n è il grado della relazione
- D_1, D_2,….., D_n sono i domini della relazione
- il numero di n-ple è la cardinalità della relazione

Qualche esempio

- **Circonferenza di raggio R**
  - D_1=insieme dei numeri reali
  - D_2=insieme dei numeri reali
  - CircR(D_1,D_2)={(x_i,y_i)| $x_i^2+y_i^2=R^2$}
- **ElencoTelefonico**
  - Dipendente= {NomiDipendentiCNR}
  - Istituto= {IstitutiCNR}
  - Telefono={NumeriDiTelefonoCentralino}
  - ElencoTelefonico(Dipendente,Istituto,Telefono)
- **Organigramma**
  - Coordinatore= {NomiDipendentiCNR}
  - Dipendente= {NomiDipendentiCNR}
  - Organigramma(Coordinatore,Dipendente)

Visualizzazione delle relazioni

- **Tabella bidimensionale**
  - le colonne sono identificate dagli attributi
  - le righe contengono i valori delle n-ple, nell' ordine indicato dall' intestazione delle colonne
  - l' ordine delle righe è indifferente: una relazione è un insieme non ordinato
  - l'ordine delle colonne non e' importante poiché il riferimento agli attibuti avviene *per nome e non per posizione*
- **Un database relazionale è un insieme di relazioni, solitamente rappresentate sotto forma di tabelle bidimensionali**

Il concetto di chiave

- **Chiave primaria (Primary Key o PK)**
  - Una combinazione di attributi che con i loro valori identificano univocamente ogni riga della tabella.
  - E' normale imporre che nessun componente della chiave primaria abbia valore nullo.
- **Chiave candidata (Candidate Key o CK)**
  - Una qualsiasi altra combinazione di attributi che identifica univocamente ogni riga della tabella
- **Chiave esterna (Foreign Key o FK)**
  - Un attributo di una relazione R1 è chiave esterna se i suoi valori sono valori della chiave primaria di R2.
  - R1 e R2 non sono necessariamente distinte.

Qualche riflessione

- **Il concetto di chiave è compreso nel modello relazionale?**
  - **Secondo alcuni il concetto di chiave è estraneo alla definizione del modello relazionale**
  - **Però, ogni relazione è un insieme, e tutti gli elementi dell' insieme sono diversi tra loro e devono essere distinguibili**
  - **Quindi esiste sempre una combinazione di attributi (eventualmente anche di tutti gli attributi) che rende un elemento diverso dall' altro**
  - **Talvolta, per comodità, viene creata appositamente una proprietà chiave (il Codice Fiscale è un esempio tipico)**

L' algebra relazionale: le operazioni tradizionali

- **Unione**
- **Intersezione**
- **Differenza**
- **Prodotto**

Un esempio (1)

- Siano date due tabelle, Manager e Employee
  - Manager modella la percentuale di tempo impiegata in attività di coordinamento nello specifico progetto
  - Employee modella la percentuale di tempo impiegata in attività di sviluppo nello specifico progetto
- Entrambe sono definite sui domini:
  - matr=numeri di matricola
  - progetto= codici progetto
  - percentuale=numero intero compreso tra 0 e 100
- Le due relazioni sono union-compatible, cioè definite sugli stessi domini nello stesso ordine

Un esempio(2)

Manager : Table

matr	progetto	percentuale
1	P1	10
2	P3	5
3	P4	5
4	P5	15
5	P6	10
8	P1	10
0		0

Record: 1

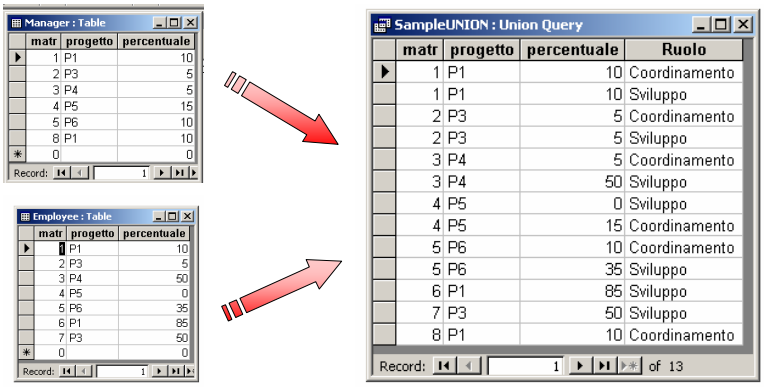
Employee : Table

matr	progetto	percentuale
1	P1	10
2	P3	5
3	P4	50
4	P5	0
5	P6	35
6	P1	85
7	P3	50
0		0

Record: 1

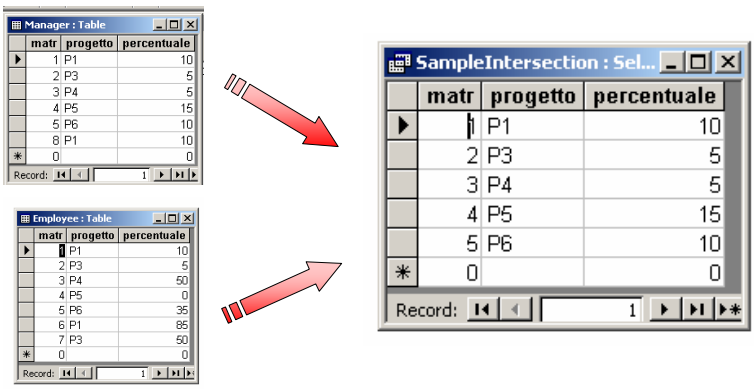
Unione

- Tutti gli elementi presenti nella prima o nella seconda tabella



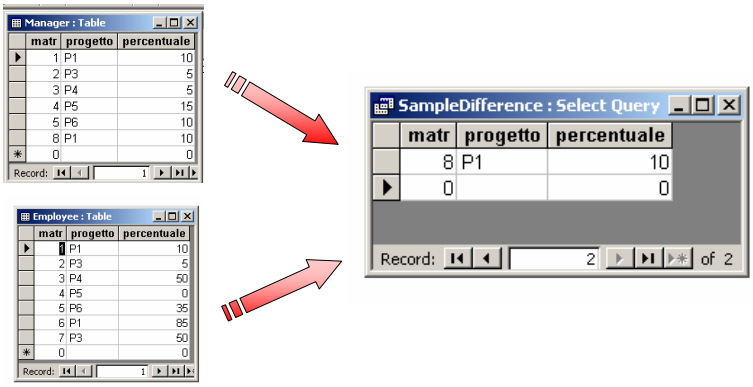
Intersezione

- Solo gli elementi presenti in entrambe le tabelle



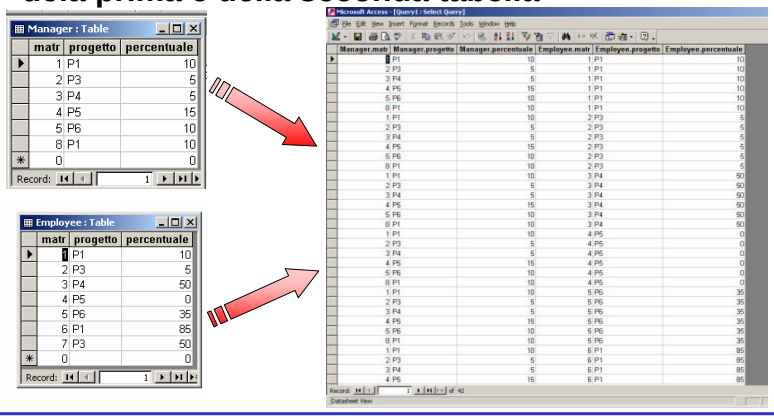
Differenza

- **Tutti gli elementi presenti nella prima ma non nella seconda tabella**



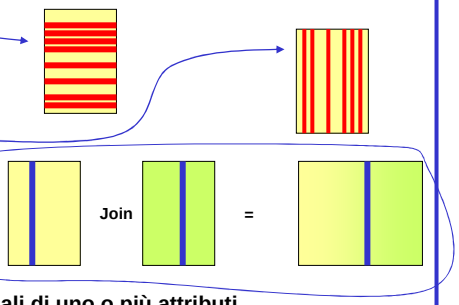
Prodotto

- **Tutte le possibili combinazioni tra gli elementi dela prima e della seconda tabella**



L' algebra relazionale: le operazioni particolari

- **SELECT**
  - seleziona righe
- **PROJECT**
  - seleziona colonne
- **JOIN**
  - Unisce tabelle per valori uguali di uno o più attributi
  - Vari tipi (natural, left,right)
- **Divisione**
  - Quoziente della divisione (raramente supportata)



Qualche riflessione

- Il risultato di una operazione dell' algebra relazionale è sempre una relazione
- In termini di visualizzazione, il risultato di una operazione su tabelle restituisce sempre una tabella
- Il risultato quindi è:
  - un insieme non ordinato
  - eventualmente composto anche da un solo elemento (tabella di una riga e una colonna)

La normalizzazione

- **Condizioni via via più stringenti**
  - Una relazione è in forma normale n+1 se è in forma normale n e inoltre soddisfa alcune altre condizioni
- **Perché normalizzare?**
  - Non per risparmiare spazio
  - Se le relazioni non sono normalizzate, si possono riscontrare effetti anomali a seguito di operazioni di:
    - inserimento
    - cancellazione
    - aggiornamento.

Un esempio

- **Una ditta di franchising rifornisce di vari Prodotti, identificati dal codice P#, vari Fornitori, ognuno dei quali è identificato da un codice F#.**
  Ogni Fornitore viene rifornito delle varie Parti in certe Quantità (QTA).
  Ogni Fornitore risiede in una città (Luogo), e acquista la merce con una certa valuta (Valuta).

Prima Forma Normale (1NF)

- **Una relazione R è in *prima forma normale* se tutte le istanze di un tipo record contengono lo *stesso numero di campi*.**

oppure:

- **Una relazione R è in *prima forma normale* se e solo se tutti i suoi domini contengono solo *valori atomici*.**

La tabella "StatoFornitori"

- **La tabella StatoFornitori è in 1NF**

F# e P# Costituiscono la PK

F#	P#	Luogo	QTA	Valuta
F01	P01	Roma	100	Lire
F01	P02	Roma	200	Lire
F03	P04	Parigi	30	Franchi
F04	P03	Londra	200	Sterline
F05	P01	Los Angeles	50	Dollari

Ma 1NF non basta...

- **Si potrebbero verificare anomalie in:**
  - **Inserimento**
    non è possibile inserire un nuovo fornitore (es. F02, con sede in Milano), se non gli viene fornita almeno una parte
  - **Cancellazione**
    cancellare la riga (F03,P04) fa sparire qualunque informazione sul fornitore F03
  - **Aggiornamento**
    se il fornitore F01 cambia Luogo, o Valuta, vanno effettuati più aggiornamenti, con potenziali errori

La soluzione

- **Le anomalie vengono risolte spezzando la relazione in due:**
  - **Una relazione F, con chiave F#**
  - **Una relazione FP, con chiave (F#,P#)**
- **Una semplice operazione di Join fornisce:**

F#	Luogo	Valuta
F01	Roma	Lire
F02	Milano	
F03	Parigi	Franchi
F04	Londra	Sterline
F05	Los Angeles	Dollari

F#	P#	QTA
F01	P01	100
F01	P02	200
F03	P04	30
F04	P03	200
F05	P01	50

F#	P#	Luogo	QTA	Valuta
F01	P01	Roma	100	Lire
F01	P02	Roma	200	Lire
F02		Milano		
F03	P04	Parigi	30	Franchi
F04	P03	Londra	200	Sterline
F05	P01	Los Angeles	50	Dollari

Seconda Forma Normale (2NF)

- La relazione F è in *seconda forma normale*
- Una relazione R è in *seconda forma normale* quando è in 1NF e un campo non chiave fornisce informazioni sull' *intera chiave*, e non su altri campi

oppure:

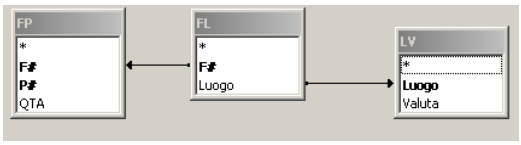
- Una relazione R è in *seconda forma normale* se è in 1NF e ogni attributo non chiave è *funzionalmente dipendente* dalla chiave.

Ma 2NF non basta...

- Si potrebbero verificare anomalie in:
  - **Inserimento**
    per i fornitori localizzati a Tokio le transazioni saranno in Yen
  - **Cancellazione**
    cancelliamo il fornitore F05 (l' unico con il quale le transazioni sono in Dollari)
  - **Aggiornamento**
    tutte le transazioni con i fornitori italiani (residenti a Roma o Milano) devono essere eseguite in Euro

La soluzione

- Le anomalie vengono risolte spezzando la relazione in due:
  - Una relazione FL, con chiave F#
  - Una relazione LV, con chiave Luogo
- I dati della tabella StatoFornitori vengono ricostruiti con una opportuna operazione di Join:



FL : Table

F#	Luogo
F01	Roma
F02	Milano
F03	Parigi
F04	Londra
F05	Los Angeles

LV : Table

Luogo	Valuta
Atene	Dollari
Londra	Sterline
Los Angeles	Dollari
Milano	Lire
Parigi	Franchi
Roma	Lire

Terza Forma Normale (3NF)

- Una relazione R è in *terza forma normale* se è in 2NF e ogni attributo non chiave è dipendente in modo *non transitivo* dalla chiave primaria.

Quarta Forma Normale (4NF)

- Una relazione del tipo:
  - IMPIEGATO (Matricola, Titolo-di-studio, Lingue-conosciute)

  è in terza forma normale (è "tutta chiave") ma contiene due fatti multivalore indipendenti: il Titolo-di-studio e le Lingue-conosciute.
- Una relazione R è in *quarta forma normale* se i suoi record non contengono due o più *fatti multivalore* indipendenti riguardo a un' entità.
- Nel caso in esempio, la relazione andrebbe spezzata in due relazioni:
  - IMP_Titolo (Matricola, Titolo-di-studio)

  e
  - IMP_Lingue (Matricola, Lingue-conosciute)

  entrambe in 4NF.

Progettare sempre in 3NF?

- In un database in 3NF le informazioni sono suddivise tra molte relazioni
- È necessario ricorrere spesso a operazioni di Join (costose in termini di prestazioni)
- Spesso si progetta in 3NF, e poi si denormalizza in fase di realizzazione

Da Concettuale a Relazionale

- Il modello relazionale è meno ricco del modello semantico di dati utilizzato nella fase di progettazione concettuale.
- La trasformazione di un modello concettuale in una base di dati relazionale comporta necessariamente l' introduzione di vincoli procedurali che devono sostituire quelli intrinseci al modello concettuale.
- Alcuni fatti:
  - Manca il meccanismo della *generalizzazione*, per cui le gerarchie di classi vanno tradotte in modo opportuno, mantenendo i vincoli impliciti nel concetto di sottoclasse, e modificando in modo adeguato le associazioni
  - *Le relazioni non coincidono con le classi* (mancano per esempio gli attributi multipli)
  - Non esiste il concetto di *aggregazione*, perché gli attributi (domini) sono elementari, e non possono essere n-ple di altre relazioni

Obiettivi della trasformazione

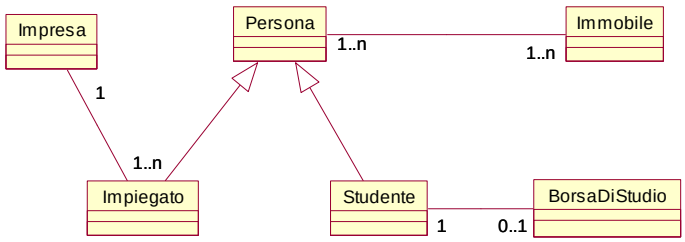
- La definizione di una base di dati relazionale passa attraverso una serie di fasi, che permettono di risolvere le differenze di rappresentazione, salvaguardando gli obiettivi della trasformazione, e cioè:
  - ottimizzare le prestazioni
  - rappresentare in modo completo i dati
  - mantenere i vincoli

I passi della trasformazione

- Trasformazione delle strutture non esistenti nel DBMS, con conseguente modifica dello schema concettuale
  - conversione delle gerarchie
- Traduzione nello schema relazionale
  - traduzione delle associazioni
- Ristrutturazione dello schema per garantire una buona efficienza
- Definizione dei parametri fisici
- Valutazione della bontà dello schema relazionale prodotto.

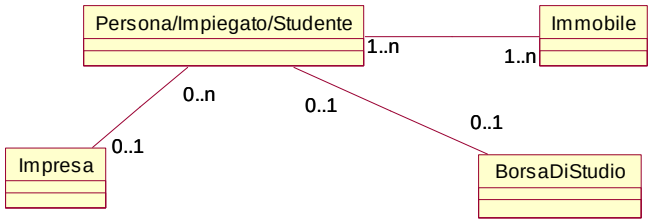
Trasformazione delle gerarchie

- Si abbia il seguente Class Diagram:



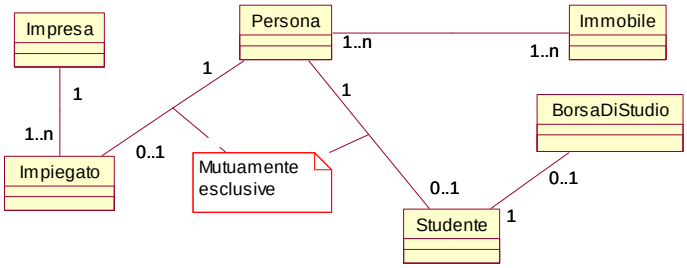
Trasformazione delle gerarchie per accorpamento

- riunificazione di tutti gli attributi specifici delle sottoclassi con quelli della classe di generalizzazione
- ristrutturazione dello schema e modifica delle associazioni
- introduzione di vincoli procedurali



Trasformazione delle gerarchie mediante associazioni

- introduzione di associazioni opzionali tra la classe di generalizzazione e le sottoclassi
- definizione delle cardinalità delle associazioni
- introduzione di vincoli procedurali



Trasformazione delle gerarchie (cont.)

- È possibile adottare soluzioni "miste":
  - alcune sottoclassi vengono accorpate alla classe di generalizzazione
  - altre vengono congiunte mediante associazioni opzionali
- Non è possibile dettare delle regole generali
- La scelta tra le varie alternative è dettata da considerazioni:
  - quantitative (numero di accessi prevedibili ai record, quantità di byte da trasferire a fronte di una query, ingombri fisici, etc.)
  - qualitative (eventuale maggiore o minore complessità delle procedure, difficoltà a garantire il rispetto dei vincoli, etc.)

Associazioni binarie 1:1

- Nella conversione delle associazioni, si può considerare l' opportunità di accorpare le entità tra cui esiste una associazione con diretta e inversa univoche.
- La scelta verrà fatta in base alle caratteristiche dell' associazione (per esempio la sua totalità, o la probabilità di esistenza nel caso in cui sia parziale).

Associazioni binarie 1:1

- Diretta univoca e totale, inversa univoca e totale
- Manager-Progetto
- Può essere tradotta come:
  - MANAGER (Matr, Cognome, Nome, ..., Prog#)
  - PROGETTO (Prog#, Nome_progetto, Data_inizio, Budget, ...)
  - Null Prog# not allowed in MANAGER

oppure in modo simmetrico

Associazioni binarie 1:1

- Diretta univoca e parziale, inversa univoca e totale
- Manager-Reparto
- Può essere tradotta come:
  - MANAGER (Matr, Cognome, Nome, ..., Rep#)
  - REPARTO (Rep#, Nome_reparto, ...)
  - Null Rep# allowed in MANAGER

oppure

  - MANAGER (Matr, Cognome, Nome, ...)
  - REPARTO (Rep#, Nome_reparto, ..., Matr)
  - Null Matr not allowed in Reparto

Associazioni binarie 1:N

- In questo caso non sono possibili soluzioni "simmetriche"
- Si utilizza il concetto di chiave esterna
- 

- Va tradotta come:
  - REPARTO (Rep#, Nome_reparto, ...)
  - PERSONA (Matr, Cognome, Nome, ..., Rep#)
  - Null Rep# not allowed in PERSONA
- Ovvi gli altri casi

Associazioni n-arie, N:M, con attributi

- Si risolvono tutte in modo analogo, creando una nuova classe con:
  - Tutte le chiavi primarie delle classi coinvolte nell' associazione
  - Gli attributi dell' associazione
- In tal modo, tutte le associazioni vengono ricondotte al tipo 1:N

Un esempio in MS Access (1)

- Come avviare Access
- Come definire le tabelle
  - le varie viste (design, datasheet)
  - caratteristiche dei campi
  - indici
- La Primary Key
  - un solo attributo
  - più attributi contigui
  - più attributi non contigui
- Importazione di dati da file esterni

Un esempio in MS Access (2)

- Le Query
  - la varie viste (design, SQL)
  - le condizioni di selezione
  - le clausole di ordinamento
  - il Join (natural, left, right)
  - il self-join
  - subquery SQL
  - Crosstab Query
  - Make-Table Query
  - Update Query
  - Append Query
  - Delete Query
  - Union

Un esempio in MS Access (3)

- Forms
- Reports

Un altro ambiente

- MySQL
- Interfaccia Web (con PHP)

Uno sguardo al futuro

- La definizione del modello relazionale è del 1970
- Molte modifiche ed evoluzioni dei prodotti
- Alcuni concetti erano e restano fondamentali in tutte le applicazioni
- I DBMS relazionali costituiscono ancora il cuore della maggior parte delle applicazioni esistenti
- Meglio investire su architetture e prodotti conformi agli standard

Conclusioni

Buon lavoro!