Il Modello Relazionale

- il modello relazionale, sebbene non sia stato il modello usato nei primi DBMS, e' divenuto lentamente il modello piu' importante al punto che e' oggi comunemente usato in tutti i DBMS disponibili a livello commerciale

- la ragione principale della popolarita' di questo modello e' che fornisce linguaggi semplici e di tipo dichiarativo, ma potenti al tempo stesso, con cui esprimere le operazioni per l'accesso e la manipolazione dei dati

- il modello relazionale e' basato sul concetto matematico di **relazione**;
  questo fornisce al modello una base teorica che permette di dimostrare formalmente proprieta' di dati e di operazioni

Il Modello Relazionale

- il concetto matematico alla base del modello relazionale e' la **relazione**

- una relazione e' formalmente definita come:
  un sottoinsieme finito del **prodotto Cartesiano** di una lista di **domini**

- un dominio e' un insieme (anche infinito) di valori;
  ad esempio
  - l'insieme dei numeri interi e' un dominio;
  - l'insieme delle stringhe di caratteri di lunghezza 20 e' un dominio
  - l'insieme {0,1} e' un dominio

- Siano $D_1, D_2, \ldots D_k$ domini. Il prodotto Cartesiano di tali domini, indicato con

$$D_1 \times D_2 \times \ldots \times D_k$$

  e' definito come l'insieme
  $\{(v_1, v_2, \ldots, v_k) \mid v_1 \in D_1, v_2 \in D_2, \ldots v_k \in D_k\}$

Il Modello Relazionale

- prodotto Cartesiano: esempio

  dati: $k=2$, $D_1=\{0,1\}$, e $D_2=\{a,b,c\}$

  $D_1 \times D_2 = \{(0,a), (0,b), (0,c), (1,a), (1,b), (1,c)\}$

- una **relazione** e' un qualsiasi sottoinsieme del prodotto Cartesiano di uno o piu' domini; per esempio:

  {(0,a), (0,c),(1,b)} e' una relazione

  {(1,b), (1,c)} e' una relazione

- gli elementi di una relazione sono detti **tuple**
  nell'esempio precedente

  (0,a), (0,c),(1,b), (1,c) sono tuple

- una relazione sottoinsieme del prodotto Cartesiano di k domini ha **grado** k

  nell'esempio precedente le relazioni hanno grado 2

Il Modello Relazionale

- ogni tupla di una relazione di grado k ha k componenti
  nell'esempio precedente le tuple hanno 2 componenti

  sia r una relazione di grado k
  sia t una tupla di r
  sia i un intero appartenente all'insieme {1,...,k}
  t[i] denota la i-sima componente di t

  esempio:
  sia r={(0,a), (0,c),(1,b)}
  sia t=(0,a) una tupla di r
  t[2] = a
  t[1] = 0

- la **cardinalita'** di una relazione e' il numero di tuple appartenenti alla relazione

  la relazione {(0,a), (0,c),(1,b)} ha cardinalita' 3

Il Modello Relazionale

Una definizione piu' semplice

- una relazione puo' essere vista, alternativamente, come una **tabella**, in cui ogni riga e' una tupla ed ogni colonna corrisponde da una componente
- alle colonne sono associati dei nomi, detti **nomi di attributo**
  la coppia (nome di attributo, dominio) e' detta **attributo**
- l'insieme degli attributi di una relazione ne costituisce lo **schema**
- se una relazione ha nome R ed attributi di nomi rispettivamente $A_1, A_2, \ldots, A_k$, lo schema e' spesso indicato con
  $R(A_1, A_2, \ldots, A_k)$
- inoltre $U_R = \{A_1, A_2, \ldots, A_k\}$ viene usato per denotare l'insieme di tutti i nomi di attributo della relazione R
- esempio: relazione di nome Info_Citta'

Citta'	Regione	Popolazione
Roma	Lazio	3000000
Milano	Lombardia	1500000
Genova	Liguria	800000
Pisa	Toscana	150000

schema Info_Citta'(Citta',Regione,Popolazione)

Il Modello Relazionale

Una definizione piu' semplice

- in questa definizione del modello relazionale, le componenti delle tuple sono denotate tramite i nomi di attributi
  (notazione per nome in contrasto con la notazione per posizione)
- dato uno schema di relazione $R(A_1, A_2, \ldots, A_k)$, una tupla t su tale schema puo' essere rappresentata tramite la notazione

  $[A_1{:}v_1, A_2{:}v_2, \ldots, A_k{:}v_k]$

  dove v_i (i=1,....,k) e' un valore appartenente al dominio di A_i (indicato con dom(A_i))

  inoltre t[A_i] denota il valore dell'attributo di nome A_i della tupla t

- esempio

  t=[Citta': Roma, Regione: Lazio, Popolazione:3000000]

  e' una tupla definita sullo schema Info_Citta'

  t[Citta'] = Roma

Il Modello Relazionale

Valori nulli

- non sempre sono disponibili informazioni sulle entita' del dominio applicativo rappresentate nella base di dati

  ==> alcune tuple possono non avere un valore per un qualche attributo

- si introduce un valore speciale (**valore nullo**) che denota la mancanza di valore

  [spesso denotato con '?']

Il Modello Relazionale

Il concetto di chiave

- data una relazione, la **chiave** della relazione e' un insieme di attributi che distingue tra loro le tuple della relazione
- piu' precisamente, un insieme X di attributi di una relazione R, e' *chiave* di R se verifica entrambe le seguenti proprieta':
  1. qualsiasi sia lo stato di R, non esistono due tuple distinte di R che abbiano lo stesso valore per tutti gli attributi in X;
  2. nessun sottoinsieme proprio (*) di X verifica la proprieta' (1).
- nell'esempio precedente:
  chiave(Info_Citta') = (Citta')
  se non esistono citta' con lo stesso nome in regioni diverse

  chiave(Info_Citta') = (Citta',Regione)
  se esistono citta' con lo stesso nome in regioni diverse

(*) dati due insiemi S and S', S' e' sottoinsieme proprio di S, se vale la relazione di inclusione stretta $S \supset S'$

Il concetto di chiave

- una chiave non puo' avere valori nulli
  (questa proprieta' puo' non essere verificata dagli attributi non chiave)
- una relazione puo' avere piu' di un insieme X che verifica le proprieta' viste
- in alcuni casi, puo' essere necessario scegliere una chiave se il sistema usato non supporta piu' chiavi
- in tal caso, il termine **chiavi candidate** viene usato per indicare le possibili chiavi
- il termine **chiave primaria** viene usato per indicare la chiave selezionata
- un criterio nella scelta della chiave primaria consiste nello scegliere tra le chiavi candidate quella piu' frequentemente usata nelle interrogazioni
- un altro criterio e' scegliere la chiave che contiene il minor numero di attributi

Il concetto di chiave esterna

- date due relazioni R ed R' tali che
  - R abbia un insieme di attributi X;
  - R' abbia come chiave un insieme Y di attributi;

  Y e' **chiave esterna** di R su R'
  se Y e' un sottoinsieme di X
- in altre parole, se una relazione R ha tra i suoi attributi un insieme di attributi che costituisce la chiave di una relazione R, allora tale insieme di attributi e' una chiave esterna di R su R'
- R' e' detta **relazione riferita**
- le chiavi esterne permettono di collegare tra loro tuple di relazioni diverse e costituiscono un meccanismo, detto <u>per valore</u>, per modellare le associazioni tra entita'
- una tupla che deve riferire un'altra tupla include tra i suoi attributi uno o piu' attributi il cui valore e' il valore della chiave della seconda tupla

Un esempio

definiamo due relazioni che contengono informazioni riguardanti i dipendenti di un'azienda ed i dipartimenti in cui l'azienda e' organizzata

le relazioni sono definite come segue

Impiegati (Imp#, Nome, Mansione, Data_A,Stipendio, Premio_P, Dip#)

chiave(Impiegati) = Imp#

chiave_esterna(Impiegati) = Dip#

(relazione riferita: Dipartimenti)

Dipartimenti(Dip#, Nome_Dip,Ufficio#, Divisione#, Dirigente)

chiave(Dipartimenti) = Dip#

Un esempio

Impiegati

Imp#	Nome	Mansione	Data_A	Stipendio	Premio_P	Dip#
7369	Rossi	ingegnere	17-Dic-80	1600,00	500,00	20
7499	Andrei	tecnico	20-Feb-81	800,00	?	30
7521	Bianchi	tecnico	20-Feb-81	800,00	100,00	30
7566	Rosi	dirigente	02-Apr-81	2975,00	?	20
7654	Martini	segretaria	28-Set-81	800,00	?	30
7698	Blacchi	dirigente	01-Mag-81	2850,00	?	30
7782	Neri	ingegnere	01-Giu-81	2450,00	200,00	10
7788	Scotti	segretaria	09-Nov-81	800,00	?	20
7839	Dare	ingegnere	17-Nov-81	2600,00	300,00	10
7844	Turni	tecnico	08-Set-81	1500,00	?	30
7876	Adami	ingegnere	28-Set-81	1100,00	500,00	20
7900	Gianni	ingegnere	03-Dic-81	1950,00	?	30
7902	Fordi	segretaria	03-Dic-81	1000,00	?	20
7934	Milli	ingegnere	23-Gen-82	1300,00	150,00	10
7977	Verdi	dirigente	10-Dic-80	3000,00	?	10

Dipartimenti

Dip#	Nome_Dip	Ufficio	Divisione	Dirigente
10	Edilizia Civile	1100	D1	7977
20	Ricerche	2200	D1	7566
30	Edilizia Stradale	5100	D2	7698

Il Modello Relazionale

Integrita' referenziale

- l'integrita' referenziale rappresenta un importante vincolo di integrita' semantica
- se una tupla t riferisce come valori di una chiave esterna i valori $v_1,.....,v_n$ allora deve esistere nella relazione riferita una tupla t' con valori di chiave $v_1,.....,v_n$
- le relazioni Impiegati e Dipartimenti verificano l'integrita' referenziale
- si consideri la seguente tupla e si assuma che sia inserita nella relazione Impiegati

  [Imp#: 7899, Nome: Smith, Mansione: tecnico,
  Data_A:03-Dic-81, Stipendio:2000,
  Premio_P: 100, Dip#: 50]

  tale tuple viola l'integrita' referenziale in quanto non esiste un dipartimento (nella relazione Dipartimenti) che abbia numero 50
- i linguaggi per basi di dati (SQL) permettono all'utente di specificare per quali relazioni e quali attributi e' necessario mantenere l'integrita' referenziale (e le azioni da eseguire in caso di violazione)

Operazioni nel Modello Relazionale

Le operazioni sulle relazioni possono essere espresse in due formalismi di base:

1) **Algebra relazionale:** le interrogazioni (queries) sono espresse applicando operatori specializzati alle relazioni

2) **Calcolo relazionale:** le interrogazioni (queries) sono espresse per mezzo di formule logiche che devono essere verificate dalle tuple ottenute come risposta all'interrogazione

Un importante risultato teorico stabilisce tuttavia che, sotto determinate assunzioni, i due formalismi hanno lo stesso potere espressivo: ognuno puo' esprimere qualsiasi query che l'altro puo' esprimere, ma non di piu'

Algebra Relazionale

- Esistono cinque operazioni di base:
  - *unione*
  - *differenza*
  - *prodotto Cartesiano*
  - *proiezione*
  - *selezione*
- queste operazioni definiscono completamente l'algebra relazionale
- ogni operazione restituisce come risultato una relazione; e' pertanto possibile applicare una operazione al risultato di un'altra operazione (proprieta' di chiusura)
- esistono operazioni addizionali, che possono essere espresse in termini delle cinque operazioni di base
- tali operazioni non aggiungono potere espressivo all'insieme delle operazioni di base, ma sono utili come abbreviazioni; di queste la piu' importante e' l'operazione di *join*
- rispetto alla notazione per nome del modello relazionale, puo' essere utile introdurre una ulteriore operazione di *renaming* che permette di modificare i nomi degli attributi

Algebra Relazionale - Unione

- l'unione di due relazioni R e S, indicata con

  $R \cup S$

  e' l'insieme delle tuple che sono in R, o in S, o in entrambe
- l'unione di due relazioni puo' essere eseguita solo se le relazioni hanno lo stesso grado; inoltre il primo attributo di R deve essere compatibile con il primo attributo di S, il secondo attributo di R deve essere compatibile con il secondo attributo di S, e cosi' via
- se le relazioni hanno nomi di attributo diversi, nella relazione risultato per convenzione si usano i nomi della prima relazione (in questo caso R), a meno di opportune ridenominazioni
- le tuple duplicate vengono eliminate
- il grado della relazione risultato e' uguale al grado delle relazioni operandi

Algebra Relazionale - Unione

esempio

A	B	C
a	b	c
d	a	f
c	b	d

relazione R

D	E	F
b	g	a
d	a	f

relazione S

A	B	C
a	b	c
d	a	f
c	b	d
b	g	a

$R \cup S$

Algebra Relazionale - Differenza

- la differenza di due relazioni R e S, indicata con

  $R - S$

  e' l'insieme delle tuple che sono in R, ma non in S
- la differenza (come l'unione) di due relazioni puo' essere eseguita solo se le relazioni hanno lo stesso grado e gli attributi sono compatibili
- se le relazioni hanno nomi di attributo diversi, nella relazione risultato per convenzione si usano i nomi della prima relazione (in questo caso R), a meno di opportune ridenominazioni
- il grado della relazione risultato e' uguale al grado delle relazioni operandi

Algebra Relazionale - Differenza

esempio

A	B	C
a	b	c
d	a	f
c	b	d

relazione R

D	E	F
b	g	a
d	a	f

relazione S

A	B	C
a	b	c
c	b	d

$R - S$

Algebra Relazionale - Prodotto Cartesiano

- il prodotto Cartesiano di due relazioni R e S, di grado rispettivamente k_1 e k_2, indicato con

  $R \times S$

  e' una relazione di grado $k_1 + k_2$ le cui tuple sono tutte le possibili tuple che hanno:

  - come prime k_1 componenti tuple di R, e
  - come seconde k_2 componenti tuple di S
- nella relazione risultato i nomi dei primi k_1 attributi sono i nomi degli attributi della relazione R e i nomi degli ultimi k_2 attributi sono i nomi degli attributi della relazione S
- se le due relazioni hanno attributi con lo stesso nome, e' necessario ridenominare gli attributi in una delle due relazioni

Algebra Relazionale - Prodotto Cartesiano

esempio

A	B	C
a	b	c
d	a	f
c	b	d

relazione R

D	E	F
b	g	a
d	a	f

relazione S

A	B	C	D	E	F
a	b	c	b	g	a
a	b	c	d	a	f
d	a	f	b	g	a
d	a	f	d	a	f
c	b	d	b	g	a
c	b	d	d	a	f

R x S

Algebra Relazionale - Proiezione

- la proiezione di una relazione R su un insieme $A=\{A_1, A_2,....,A_m\}$ di attributi, indicata con

  $$\Pi_{A1,\ A2,....,Am}(R)$$

  e' una relazione di grado *m* le cui tuple hanno come attributi solo gli attributi specificati in A

- pertanto la proiezione genera un insieme T di m-tuple
  (una m-tupla e' una tupla con m attributi)

  sia $t = [A_1:v_1, A_2:v_2,....,A_m:v_m]$ una m-tupla in T
  t e' tale che esiste una tupla t' in R tale che:
  $\forall A_i \in A \ \ t[A_i] = t'[A_i]$

- l'operazione di proiezione ha dunque l'effetto di generare, da una data relazione, una relazione che contiene solo gli attributi specificati nell'operazione

- nella relazione risultato gli attributi hanno l'ordine specificato in A

Algebra Relazionale - Proiezione

esempio

A	B	C
a	b	c
d	a	f
c	b	d

relazione R

A	C
a	c
d	f
c	d

$\Pi_{A,C}(R)$

B	A
b	a
a	d
b	c

$\Pi_{B,A}(R)$

Algebra Relazionale - Selezione

predicati

- un predicato F su una relazione ha una delle seguenti forme:

  - predicato semplice
  - combinazione Booleana di predicati semplici;
    le combinazioni Booleane sono ottenute mediante gli operatori logici

    $\wedge$ (AND), $\vee$ (OR), $\neg$ (NOT)

- un *predicato semplice* ha una delle seguenti forme

  (i) A op *costante*
  (ii) A op A'

  dove A e A' sono attributi di R;
  op e' un *operatore relazionale di confronto*:
  $<, >, \leq, \geq, =$, etc.
  costante e' un valore costante compatibile con il dominio di A

- esempi:

B=b	predicato semplice della forma (i)
A=C	predicato semplice della forma (ii)
$B=b \vee A=C$	combinazione Booleana
$B=b \wedge A=C$	combinazione Booleana
$\neg B=b$	combinazione Booleana

Algebra Relazionale - Selezione

- la selezione su una relazione R, dato un predicato F, indicata con

  σ_F (R)

  e' una relazione che contiene tutte le tuple che verificano il predicato F

- il grado della relazione risultato e' uguale al grado della relazione operando; i nomi degli attributi della relazione risultato sono gli stessi della relazione operando

- se nessuna tupla di R verifica il predicato F, allora il risultato e' una relazione vuota (indicata con 0)

- pertanto, se k e' il grado di R, la selezione genera un insieme T di k-tuple
  (una k-tupla e' una tupla con k attributi)

  sia t = $[A_1{:}v_1, A_2{:}v_2, ..., A_k{:}v_k]$ una k-tupla in T
  t e' tale che:

  $F A_1/t[A_1], A_2/t[A_2], ..., A_k/t[A_k]$

  e' vera, dove la notazione $A_i/t[A_i]$, i=1,..,k

  indica la sostituzione in F del nome di attributo A_i (se tale nome compare in F) con il valore dell'attributo di nome A_i in t

Algebra Relazionale - Selezione

esempio

A	B	C
a	b	c
d	a	f
c	b	d

relazione R

A	B	C
a	b	c
c	b	d

$\sigma_{B=b}$(R)

A	B	C
d	a	f

$\sigma_{\neg B=b}$(R)

A	B	C
a	b	c
c	b	d

$\sigma_{B=b \vee A=C}$ (R)

$\sigma_{B=b \wedge A=C}$ (R) =0

Algebra Relazionale

esempi dalla base di dati impiegati e dipartimenti

- **Q1:** selezionare i nomi degli impiegati che hanno uno stipendio maggiore di 2000

  $\Pi_{Nome}(\sigma_{Stipendio>2000}$(Impiegati))

Nome
Rosi
Blacchi
Neri
Dare
Verdi

- **Q2:** selezionare i nomi ed i numeri di dipartimento degli impiegati che hanno uno stipendio maggiore di 2000 e hanno mansione di ingegnere

  $\Pi_{Nome,Dip\#}(\sigma_{Stipendio>2000 \wedge Mansione= \text{'ingegnere'}}$ (Impiegati))

Nome	Dip#
Neri	10
Dare	10

- **Q3:** selezionare i numeri di impiegato degli impiegati che: (a) lavorano nel dipartimento 30 e (b) sono ingegneri o tecnici

  $\Pi_{Imp\#}(\sigma_{Dip\#=30 \wedge (Mansione= \text{'ingegnere'} \vee Mansione= \text{'tecnico'})}$ (Impiegati))

Imp#
7499
7521
7844
7900

Algebra Relazionale - Ridenominazione

La ridenominazione di una relazione R rispetto ad una lista di coppie di nomi di attributi

$(A_1, B_1), (A_2, B_2), ..., (A_m, B_m)$

tale che A_i (i=1,...,m) e' un nome di attributo di R, denotata con

$\rho_{A1,\ A2,\ ....,Am \leftarrow B1,\ B2,\ ....,Bm}$ (R)

ridenomina l'attributo di nome A_i (i=1,...,m) con il nome B_i

La ridenominazione e' corretta se il nuovo schema della relazione R ha attributi con nomi tutti distinti

Esempio:

R(A,B,C)

$\rho_{A,\ B,\ C \leftarrow AA,\ BB,\ CC}$ (R)

ha l'effetto di modificare lo schema della relazione R in R(AA,BB,CC)

Algebra Relazionale - Operazioni di Base

Sia R = (A1, ..., Ak) uno schema di relazione, con Ai nome di attributo con dominio Si, per i = 1 ...k. Indichiamo con $\Re(R)$ l'insieme di tutte le relazioni su tale schema.

- $\_ \cup \_ : \Re(R) \times \Re(R) \to \Re(R)$

  $r1 \cup r2 = \{t \mid t \in r1 \vee t \in r2\}$

- $\_ \setminus \_ : \Re(R) \times \Re(R) \to \Re(R)$

  $r1 \setminus r2 = \{t \mid t \in r1, t \notin r2\}$

- $\_ \times \_ : \Re(R1) \times \Re(R2) \to \Re(R1 \cdot R2)$

  $r1 \times r2 = \{t1 \cdot t2 \mid t1 \in r1, t2 \in r2\}$

  [nella versione "posizionale"]

  $\_ \times \_ : \Re(R1) \times \Re(R2) \to \Re(R1 \oplus R2)$

  [nella versione "con nome"]

- $\pi_{R'}\ \_ : \Re(R) \to \Re(R')$ con $R \supseteq R'$

  $\pi_{R'}(r) = \{t[R'] \mid t \in r\}$

- $\sigma_F\ \_ : \Re(R) \to \Re(R)$

  $\sigma_F(r) = \{t \mid t \in r, F(t)\}$

Algebra Relazionale - Join

- il join di due relazioni R e S sugli attributi A di R ed A' di S, indicato con

  $$R \underset{A\theta A'}{|X|} S$$

  e' definito come $\sigma_{A\theta A'}$ (R x S)
- il join e' pertanto un prodotto Cartesiano seguito da una selezione; il predicato AθA' e' detto *predicato di join*
- il grado della relazione risultato e' uguale alla somma dei gradi delle relazioni operandi
- spesso il join e' anche indicato con le seguenti notazioni

  R.A θ S.A'   R[A θ A']S
- il join prende il nome di equijoin quando l'operatore θ usato nel predicato di join e' l'operatore di uguaglianza (=)

Algebra Relazionale - Join

esempio

A	B	C
1	2	3
4	5	6
7	8	9

relazione R

D	E
3	1
6	2

relazione S

A	B	C	D	E
1	2	3	3	1

$$R \underset{A=E}{|X|} S$$

A	B	C	D	E
1	2	3	3	1
1	2	3	6	2
4	5	6	6	2

$$R \underset{B<D}{|X|} S$$

Algebra Relazionale - Join naturale

- l'operazione di join naturale rappresenta una "semplificazione" del join
- si consideri l'interrogazione

  "Ritrovare tutti gli impiegati e l'ufficio in cui lavorano"

  usando il join tale interrogazione e' espressa come

  $$\Pi_{Nome,\ Ufficio}\ (Impiegato \underset{Impiegato.Dip\# = Dipartimento.Dip\#}{|x|} Dipartimento)$$
- notare che questo particolare join impone l'eguaglianza su quegli attributi che appaiono in entrambe le relazioni
- i join basati sull'eguaglianza degli attributi sono frequenti nella pratica
- l'operazione di *join naturale* indica un tipo di join basato sull'eguaglianza degli attributi a comune tra due relazioni
- e' un'operazione che a differenza delle altre ha senso solo nella notazione con nome

Algebra Relazionale - Join naturale

- l'interrogazione precedente viene espressa come segue

  $\Pi_{\text{Nome, Ufficio}}$ (Impiegato |x| Dipartimento)

- Definizione

  siano R ed S due relazioni

  sia {A1,A2,....,Ak} = $U_R \cap U_S$ l'insieme degli attributi presenti sia nello schema di R che nello schema di S

  sia {I1,I2,....,Im} = $U_R \cup U_S$ l'unione degli attributi nello schema di R e nello schema di S

  l'espressione che definisce il join naturale e'

  R |x| S = $\Pi_{I1,I2,....,Im}$ (σ_C (R x ($\rho_{A1, A2, ...,Ak \leftarrow S.A1, S.A2,... S.Ak}$ (S))))

  dove C e' un predicato della forma

  A1=S.A1 AND A2=S.A2 AND .... Ak=S.Ak

- il join naturale esegue pertanto un join eguagliando gli attributi con lo stesso nome delle due relazioni e poi elimina gli attributi duplicati

Algebra Relazionale - Join naturale

R

A	B	C
a	b	c
d	b	c
b	b	f
c	a	d

S

B	C	D
b	c	d
b	c	e
a	d	b

R |X| S

A	B	C	D
a	b	c	d
a	b	c	e
d	b	c	d
d	b	c	e
c	a	d	b

Algebra Relazionale - Divisione

- si supponga che gli impiegati della base di dati di esempio siano assegnati a dei corsi di aggiornamento; ogni impiegato in genere partecipa a piu' corsi e viceversa ogni corso e' seguito da piu' impiegati; vengono inoltre rappresentate le informazioni relativamente ai corsi

- si supponga, pertanto, che siano definite le seguenti relazioni

  Segue (Imp#, Corso#)

  Corsi (Corso#, Argomento, Durata)

- un possibile contenuto delle due relazioni e' il seguente:

Segue

Imp#	Corso#
7369	10
7369	20
7369	30
7782	10
7782	40

Corsi

Corso#	Argomento	Durata
20	CAD	5
10	Basi di dati	3
30	Basi di dati	2
40	Sistemi oper.	4

Algebra Relazionale - Divisione

- consideriamo la seguente interrogazione

  "trovare il numero di impiegato degli impiegati che seguono tutti i corsi il cui argomento e' basi di dati"

- il numero di corso dei corsi il cui argomento e' basi di dati e' ottenuto come segue:

  R1 = $\Pi_{\text{Corso\#}}$ ($\sigma_{\text{Argomento = 'Basi di dati'}}$ (Corsi))

  il risultato dell'espressione R1 e' il seguente insieme di numeri di corso {10,30}

- il risultato della interrogazione e' dato, pertanto, da tutti quegli impiegati che appaiono nella relazione Segue con ognuno dei numeri di corso determinati da R1

  il risultato della nostra interrogazione e' pertanto solo l'impiegato il cui numero e' 7369

Algebra Relazionale - Divisione

- l'operazione che permette di eseguire l'interrogazione precedente e' l'operazione di divisione

- Definizione
  Date due relazioni R ed S con insiemi di attributi U_R ed U_S, rispettivamente, e tali che $U_R \supseteq U_S$, l'operazione di divisione di R per S e' denotata da
  $R \div S$
  ed e' espressa come segue:
  $\Pi_{(U_R - U_S)}(R) - \Pi_{(U_R - U_S)}((\Pi_{(U_R - U_S)}(R) \times S) - R)$
  l'espressione a destra del -
  determina tutte le tuple di R
  che non sono associate ad almeno
  una tupla di S

- l'interrogazione dell'esempio precedente e' espressa come segue

  $\text{Segue} \div \Pi_{Corso\#}(\sigma_{Argomento = \text{'Basi di dati'}}(\text{Corsi}))$

Algebra Relazionale - Divisione

- sviluppo:

  R = Segue

  $S = \Pi_{Corso\#}(\sigma_{Argomento = \text{'Basi di dati'}}(\text{Corsi}))$

  S = {10, 30}
  U_R = {Imp#, Corso#}
  U_S = {Corso#}

1) $\Pi_{(U_R - U_S)}(R) = \Pi_{Imp\#}(R) =$

Imp#
7369
7782

2) $\Pi_{(U_R - U_S)}(R) \times S = \Pi_{Imp\#}(R) \times S =$

Imp#
7369
7782

x

Corso#
10
30

=

Imp#	Corso#
7369	10
7369	30
7782	10
7782	30

3) $(\Pi_{(U_R - U_S)}(R) \times S) - R = (\Pi_{Imp\#}(R) \times S) - R =$

Imp#	Corso#
7369	10
7369	30
7782	10
7782	30

-

Imp#	Corso#
7369	10
7369	20
7369	30
7782	10
7782	40

=

Imp#	Corso#
7782	30

Algebra Relazionale - Divisione

4) $\Pi_{(U_R - U_S)}((\Pi_{(U_R - U_S)}(R) \times S) - R) =$
$\Pi_{Imp\#}((\Pi_{Imp\#}(R) \times S) - R) =$

Imp#
7782

5) $\Pi_{(U_R - U_S)}(R) -$ Calcolato al passo (1)
$\Pi_{(U_R - U_S)}((\Pi_{(U_R - U_S)}(R) \times S) - R)$
Calcolato al passo (4)

Imp#
7369
7782

-

Imp#
7782

=

Imp#
7369

Algebra Relazionale - Operazioni Derivate

Sia R = (A1, ..., Ak) uno schema di relazione, con Ai nome di attributo con dominio Si, per i = 1 ...k. Indichiamo con $\Re(R)$ l'insieme di tutte le relazioni su tale schema.

- $\_ \cap \_ : \Re(R) \times \Re(R) \to \Re(R)$
  $r1 \cap r2 = r1 \setminus (r1 \setminus r2) = \{t \mid t \in r1, t \in r2\}$

- $\_ |x|_F \_ : \Re(R1) \times \Re(R2) \to \Re(R1 \oplus R2)$
  $r1 |x|_F r2 = \sigma_F(r1 \times r2) =$
  $\{t1 \cdot t2 \mid t1 \in r1, t2 \in r2, F(t1,t2)\}$

- $\_ |x| \_ : \Re(R1) \times \Re(R2) \to \Re(R1 \cup R2)$
  $r1 |x| r2 =$
  $\{t \mid t[R1] \in r1, t[R2] \in r2\}$

  - se $R1 \cap R2 = \emptyset$ $r1 |x| r2 = r1 \times r2$
  - se $R1 = R2$ $r1 |x| r2 = r1 \cap r2$

- $\_ \div \_ : \Re(R1) \times \Re(R2) \to \Re(R1 \setminus R2)$ con $R1 \supseteq R2$
  $r1 \div r2 = \{t \mid \forall t2 \in r2\ \exists t1 \in r1$ t.c.
  $t1[R1 \setminus R2] = t, t1[R2] = t2\}$

Calcolo Relazionale

- l'algebra relazionale e' un linguaggio "procedurale" in quanto, nello specificare un'espressione algebrica, dobbiamo indicare le operazioni necessarie per generare il risultato della interrogazione

- nel calcolo relazionale viene data una descrizione formale del risultato senza specificare come ottenerlo

- due varianti del calcolo relazionale

  - tuple relational calculus (TRC)

    le variabili rappresentano tuple

  - domain relational calculus (DRC)

    le variabili rappresentano valori di domini

Calcolo Relazionale

In TRC una query e' un'espressione della forma

$\{t: U \mid P(t)\}$

e' cioe' definita come l'insieme di tutte le tuple t definite su un insieme di attributi U tali che il predicato P e' vero per t

<u>Notazione</u> t[A] indica il valore della tupla t per l'attributo A (ad es. t[Nome])

$t \in R$ indica che la tupla t e' nella relazione R

<u>Esempi</u>:

- determinare tutti gli impiegati il cui stipendio e' maggiore di 2000

  $\{t: U_{Impiegati} \mid t \in Impiegati \wedge t[Stipendio] > 2000\}$

Calcolo Relazionale

- trovare i nomi degli impiegati il cui stipendio e' maggiore di 2000

  $\{t: \{Nome\} \mid (\exists s)\ (s \in Impiegati \wedge s[Stipendio] > 2000 \wedge s[Nome] = t[Nome])\}$

  t rappresenta una variabile che indica tuple appartenenti ad una relazione che ha come schema {Nome}

  la notazione $(\exists t)(Q(t))$ indica che esiste una tupla t tale che Q(t) e' vera

- trovare i nomi e gli uffici degli impiegati che hanno uno stipendio e' maggiore di 2000

  $\{t: \{Nome, Ufficio\} \mid (\exists s)\ (s \in Impiegati \wedge s[Stipendio] > 2000 \wedge s[Nome] = t[Nome] \wedge (\exists u)\ (u \in Dipartimenti \wedge s[Dip\#] = u[Dip\#] \wedge u[Ufficio] = t[Ufficio]))\}$

- trovare i nomi degli impiegati che hanno uno stipendio maggiore di 2000 oppure lavorano in un dipartimento della divisione D1

  $\{t: \{Nome\} \mid (\exists s)\ (s \in Impiegati \wedge s[Nome] = t[Nome] \wedge (s[Stipendio] > 2000 \vee (\exists u)\ (u \in Dipartimenti \wedge s[Dip\#] = u[Dip\#] \wedge u[Divisione] = "D1")))\}$

Calcolo Relazionale

Atomi. Gli atomi hanno le seguenti forme:

1. $s \in R$ dove: R e' un nome di relazione; s e' una variabile.

   Questo atomo asserisce che la tupla denotata da s appartiene alla relazione R.

2. $s[A]\ \theta\ u[A']$ dove: s ed u sono variabili; θ e' un operatore relazionale di confronto; A ed A' sono nomi di attributi.

   Questo atomo asserisce che il valore dell'attributo di nome A della tupla denotata da s e' in relazione θ con il valore dell'attributo di nome A' della tupla denotata da u.

3. $s[A]\ \theta\ a$ dove: s e' una variabile; θ e' un operatore relazionale di confronto; A e' un nome di attributo; a e' una costante.

   Questo atomo asserisce che il valore dell'attributo di nome A della tupla denotata da s e' in relazione θ con il valore a.

Calcolo Relazionale

Variabili libere e legate.

Data una formula F e una variabile x, x e' *libera* se e solo se x non compare in un quantificatore

$\exists x$ (quantificatore esistenziale)

$\forall x$ (quantificatore universale)

Formule. Le formule legali del linguaggio sono tutte e sole le formule definibili in accordo a quanto segue

1. Ogni atomo e' una formula. Tutte le occorrenze delle variabili presenti nell'atomo sono libere
2. Se $\phi 1$ e $\phi 2$ sono formule, allora:
   $\phi 1 \wedge \phi 2$, $\phi 1 \vee \phi 2$, $\neg \phi 1$ sono formule
   Tutte le occorrenze delle variabili presenti nell'atomo sono libere o legate a seconda che siano libere o legate in $\phi 1$ o $\phi 2$
3. Se ϕ e' una formula, allora:
   $(\exists s)\ (\phi)$, $(\forall s)\ (\phi)$ sono formule
   Tutte le occorrenze libere di s in ϕ sono legate dal quantificatore $\exists$ ($\forall$)
4. Se ϕ e' una formula, allora: (ϕ) e' una formula

Calcolo Relazionale

Esempio:

- la formula

  $(\exists s)\ (s \in \text{Impiegati} \wedge s[\text{Stipendio}] > 2000)$

  e' una formula legale
  tutte le occorrenze di s sono legate

- la formula

  $(\exists x)\ (x \in \text{Impiegati} \wedge x[\text{Stipendio}] > 2000$
  $\wedge\ x[\text{Dip\#}] = y[\text{Dip\#}])$

  e' una formula legale tutte le occorrenze di x sono legate mentre quelle di y sono libere

Espressioni del calcolo. Un'espressione (o interrogazione) del calcolo su tuple ha la forma

$\{x{:}U \mid f(x)\}$

dove: U e' un insieme di attributi; f e' una formula legale del calcolo; x e' l'unica variabile libera in f(x)

Calcolo Relazionale

Esempio:

- l'espressione

  $\{y{:}U_{\text{Dip\#}} \mid (\exists x)\ (x \in \text{Impiegati}$
  $\wedge (x[\text{Stipendio}] > 2000 \wedge x[\text{Dip\#}] = y[\text{Dip\#}])\}$

  e' un'espressione corretta di TRC
  ritrova tutti i dipartimenti che hanno almeno un'impiegato che guadagna piu' di 2000

- l'espressione

  $\{y{:}U_{\text{Impiegati}} \mid (\forall y)$
  $(y \in \text{Impiegati} \wedge y[\text{Mansione}] = \text{'ingegnere'})\}$

  non e' un'espressione corretta di TRC in quanto y non e' libera

Calcolo Relazionale

Espressione delle operazioni dell'algebra relazionale in termini del calcolo relazionale

- $R \cup S$ **Unione**

  $\{t : U_R \mid t \in R \vee t \in S\}$

- $R - S$ **Differenza**

  $\{t : U_R \mid t \in R \wedge t \notin S\}$

- R X S **Prodotto Cartesiano**

  siano: $U_R = \{A1, A2, \ldots\ldots, An\}$
  $U_S = \{A1', A2', \ldots\ldots, Am'\}$

  gli insiemi di attributi di R ed S, rispettivamente

  $\{t{:}\ \{U_R \cup U_S\} \mid (\exists x)\ (\exists y)$
  $(x \in R\ \wedge\ y \in R\ \wedge$
  $x[A1] = t[A1]\ \wedge\ x[A2] = t[A2]\ \wedge \ldots\ldots \wedge$
  $x[An] = t[An]\ \wedge$
  $y[A1'] = t[A1']\ \wedge\ y[A2'] = t[A2']\ \wedge \ldots\ldots \wedge$
  $y[Am'] = t[Am'])\}$

Calcolo Relazionale

- $\Pi_{A1,A2, ...Ak}$ (R) **Proiezione**

  {t : {A1, A2,....,Ak}| (∃x) (x ∈ R ∧ x[A1] = t[A1] ∧ x[A2] = t[A2] ∧.....∧ x[Ak] = t[Ak])}

- σ_F (R) **Selezione**

  {t: U_R | t ∈ R ∧ F'}

  dove F'e' la formula F con ogni attributo A sostituito da t[A]

Equivalenza tra algebra relazionale e calcolo relazionale

Ad ogni espressione *e* di un certo linguaggio di interrogazione e` possibile associare una funzione $\mu(e)$ che rappresenta la semantica dell'espressione considerata.

e espressione
μ : {espressioni} ({basi di dati} {basi di dati})
$\mu(e)$ funzione
$\mu(e)(db)$ nuova base di dati

$\mu(e)$ e` chiamata interrogazione se e`: parziale, computabile, generica.

Definizione. Sia D_u l'unione di tutti i domini degli attributi delle relazioni contenute in una base di dati di schema *S*. Una funzione e` generica se, considerando una funzione *f* di ridenominazione delle costanti contenute in D_u, per ogni base di dati *d* con schema *S*, si ha $\mu(e)(f(d)) = f(\mu(e)(d))$.

Una funzione generica e` indipendente dalla particolare rappresentazione dei dati, utilizzata per calcolare la funzione.

Definizione. Siano e_1 e e_2 due espressioni di un certo linguaggio di interrogazione. e_1 e e_2 sono equivalenti se, per ogni base di dati *d*, espressa nel modello considerato, $\mu(e_1)(d) = \mu(e_2)(d)$.

Calcolo relazionale algebra relazionale

Non sempre le espressioni del calcolo relazionale possono essere tradotte in equivalenti espressioni dell'algebra.

Problema. Le espressioni del TRC non sempre restituiscono una relazione finita.

Esempio.

{t : U_R| ↓ R(t)}

Se almeno uno dei domini degli attributi di R e` un insieme infinito, l'espressione precedente genera una relazione infinita, composta da tutte le tuple che non appartengono alla relazione R.

Soluzione. Definizione di formula *indipendente dal dominio*.

Una formula e` indipendente dal dominio se la sua valutazione genera sempre lo stesso risultato anche supponendo di estendere la base di dati con nuove relazioni o nuove tuple, contenenti valori non presenti nella base di dati di partenza.

Prima di introdurre questo concetto, e` necessario introdurre alcune nozioni preliminari.

Definizione. Data una formula *F*, si definisce *DOM(F)* l'insieme di tutti i valori che compaiono nella formula stessa e nelle tuple contenute nelle relazioni che

Esempio. Sia F:

{t : $U_{Impiegati}$ | t Impiegati t[Stipendio] > 2000}.

Allora:

DOM(F) = {2000} $\Pi_{Imp\#}$ (Impiegati)
Π_{Nome} (Impiegati) $\Pi_{Mansione}$ (Impiegati)
Π_{Data_A}(Impiegati) Π_{Premio_P}(Impiegati)
Π_{Dip}(Impiegati).

Definizione. Si consideri un insieme *D* tale che D_u *D* *DOM(F)*. Sia $F(x_1,...,x_n)$ una formula con variabili (di tupla) libere $x_1,..., x_n$. Si assuma che il grado della variabile di tupla x_i sia g_i, $i=1,...,n$. Sia Dt_{gi} l'insieme di tutte le tuple di grado g_i che possono essere costruite considerando i valori in *D* (quindi Dt_{gi} rappresenta il prodotto Cartesiano di g_i domini ciascuno coincidente con *D*).

La *relazione associata ad F rispetto a D* e` costituita dall'insieme:

{(t_1,..., t_n) | t_i Dt_{gi}, i=1,...,n, F(t_1,..., t_n) e` vera}

Definizione (Indipendenza dal dominio). Sia *F* una formula. Sia D_u l'unione di tutti i domini. *F* e` indipendente dal dominio se per ogni *D* tale *che* D_u *D* *DOM(F)*, la relazione associata ad *F* rispetto a *D* coincide con la relazione associata ad *F* rispetto a *DOM(F).*

Esempio. Sia F:

$$\{t : U_{Impiegati} \mid \downarrow (t \quad Impiegati)\}.$$

F non e` indipendente dal dominio. Infatti:

$DOM(F) = \Pi_{Imp\#}(Impiegati) \quad \Pi_{Nome}(Impiegati)$

$\Pi_{Mansione}(Impiegati) \quad \Pi_{Data\_A}(Impiegati)$

$\Pi_{Premio\_P}(Impiegati) \qquad \Pi_{Dip}(Impiegati).$

Si consideri D DOM(F).

Allora la relazione ottenuta valutando F e`

D**6** - Impiegati

che e` in generale diverso da

DOM(F)**6** - Impiegati.

La nozione di indipendenza dal dominio e` indecidibile

introduzione condizione sintattica (*safety*) sufficiente a garantire l'indipendenza dal dominio di una formula.

Definizione (Safety). Sia *F* una formula del TRC. Supponiamo che il grado di ogni variabile usata in *F* sia nota e fissa. *F* e` safe se:

1. Il quantificatore ∀ non compare in *F*.

2. Se l`operatore compare in *F*, allora se $F_1$ e $F_2$ sono le formule connesse da , in esse compare una sola variabile di tupla libera e tale variabile e` comune alle due formule.

3. Si consideri ciascuna sottoformula di *F*, composta da una congiunzione massimale di formule del tipo F_1 ... F_n. Le componenti di tutte le variabili di tupla libere in ciascuna sottoformula F_i, $i=1,\dots,n$, sono limitate, cioe`:

   a) Se F_i e` non negata, non rappresenta un confronto aritmetico, e contiene una variabile libera *t*, allora tutte le componenti di *t* sono limitate.

   b) Se F_i e` un`uguaglianza del tipo $t[j] = a$ o $a = t[j]$, dove *a* e` una costante, allora *t[j]* e` limitata.

   c) Se F_i e` un`uguaglianza del tipo $t[j] = u[i]$ o $v[i] = t[j]$ e la componente *u[i]* e` limitata, allora *t[j]* e` limitata.

4. L`operatore ↓ puo` comparire in *F* solo se applicato ad una formula che compare in una congiunzione del tipo illustrato al punto (3).

Il TRC safe puo` adesso essere definito come l'insieme delle espressioni del TRC della forma

$$\{x : U \mid f(x)\}$$

dove f e` una formula safe.

Vale il seguente teorema.

Teorema. Ogni espressione dell`algebra relazionale puo` essere tradotta, in tempo polinomiale nella dimensione dell`espressione, in una equivalente espressione del TRC safe.

Ogni espressione del TRC safe puo` essere tradotta, in tempo polinomiale nella dimensione dell'espressione, in una equivalente espressione dell'algebra relazionale.

Complessita` dei linguaggi relazionali

Ad ogni linguaggio di interrogazione *L* puo` essere associato un insieme di funzioni, che rappresentano la semantica di ciascuna espressione esprimibile in *L*.

Per ciascuna di queste funzioni, e possibile determinarne la *complessita`*, cioe` il costo della loro esecuzione.

**Definizione (Complessita` dei dati).** Sia *e* un`espressione di un certo linguaggio di interrogazione *L*. La complessita` dei dati di $\mu(e)$ e` la complessita` computazionale del test di appartenenza all'insieme

$$\{ \langle t,d \rangle \mid \text{l'entita` } t \text{ appartiene a } \mu(e)(d)\}.$$

Se questo problema appartiene ad una classe di complessita` computazionale *C*, allora si dice che $\mu(e)$ (o *e*) e` in C.

L ha complessita` *C* se tutte le interrogazioni esprimibili in *L* sono in *C* (accettabile se *C* PTIME).

Teorema. L`algebra relazionale e il TRC hanno complessita` LOGSPACE.

Poiche` LOGSPACE NC, il risultato precedente afferma che le interrogazioni esprimibili con i linguaggi relazionali sono facilmente parallelizzabili.

Il prezzo da pagare per questa bassa complessita` e` un limitato potere espressivo.

Vincoli di Integrita' Semantica

- un vincolo e' una proprieta' che un insieme di dati deve verificare

- i vincoli possono essere classificati in
  vincoli *immediati*
  vincoli *differiti*

- una seconda classificazione (ortogonale alla precedente) e' tra
  vincoli *statici*
  vincoli *di transizione*

- i vincoli possono essere inoltre classificati a seconda degli oggetti che accedono
  - vincoli su singola relazione
    (i) vincoli su singola tupla
      * vincoli di attributo
      * vincoli su attributi multipli

    (ii) vincoli su tuple multiple della stessa relazione
      * dipendenze funzionali
      * vincoli di cardinalita'

    (iii) vincoli di aggregazione
  - vincoli su relazioni multiple

    integrita' referenziale

Vincoli di Integrita' Semantica

Esempi

- vincolo su singolo attributo

  lo stipendio di un impiegato deve essere compreso tra 500 e 1000

- vincolo su attributi multipli

  il premio di produzione di un impiegato deve essere sempre minore dello stipendio dell'impiegato

- vincoli di cardinalita'

  devono esserci almeno tre impiegati con mansione di tecnico

- vincoli di aggregazione

  la media dello stipendio dei tecnici deve essere maggiore di 500

- vincoli su relazioni multiple

  la somma degli stipendi degli impiegati che lavorano nel progetto P deve essere minore del budget di P