

Cap. 2 - Il Modello Relazionale

Concetti e definizioni

2

Introduzione

- Modello proposto nel 1970 da E.F. Codd in:
 - "A relational model for large shared data banks"
Communications of the ACM Vol. 13, n. 6, pagg. 377-387
- Prime apparizioni nel mercato solo nel 1981
- Caratterizzato da un alto livello di astrazione
 - proposto per superare le limitazioni precedenti
 - caratterizzato da una elevata indipendenza dei dati
 - ha richiesto l'individuazione di realizzazioni efficienti e di hardware adeguato.

Definizioni

I fattori del successo

- Il modello relazionale si fonda su due concetti:
 - La relazione
 - Definizione formale
 - Ereditata dalla teoria degli insiemi
 - Utile per completare il modello con una precisa teoria
 - La tabella
 - Semplice ed intuitiva
 - Rappresentazione grafica
 - Utile nella comunicazione con gli utenti

3

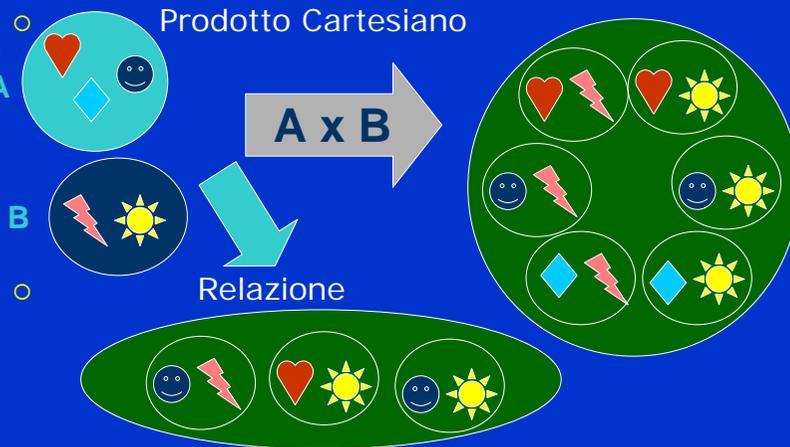
Relazione: definizione

- Dati $n > 0$ domini non necessariamente distinti
 D_1, D_2, \dots, D_n
una relazione matematica r sui domini D_i è un sottoinsieme del prodotto cartesiano
$$r \subseteq D_1 \times D_2 \times \dots \times D_n$$
- Nella teoria relazionale dei dati si fa l'ulteriore ipotesi:
 - **i domini sono "a valori atomici"**

4

Definizioni

Relazione: schematizzazione grafica



Relazione

E' vista come un insieme di ennuple ordinate:
 $t = (v_1, v_2, \dots, v_n) \mid v_1 \in D_1, v_2 \in D_2, \dots, v_n \in D_n$

- Ciascuna ennupla si chiama, nella terminologia relazionale, **tupla**
- Il numero n è detto grado del prodotto cartesiano e della relazione
- Il numero di tuple della relazione viene detto cardinalità della relazione

Definizioni



Esempio (Relazione)

Siano dati i seguenti domini:

Codice = {001, 004, 005}

Nome = {Mel, Pedro, Federico}

Cognome = {Almodovar, Gibson, Fellini},

Nazionalità = {Italia, Spagna, Australia}.

Una relazione su questi domini è una generica :

$r \subseteq \text{Codice} \times \text{Nome} \times \text{Cognome} \times \text{Nazionalità}$

Possibili relazioni sono dunque:

$r_1 = \{(001, \text{Pedro}, \text{Almodovar}, \text{Spagna})\}$

$r_2 = \{(001, \text{Pedro}, \text{Amodovar}, \text{Spagna});$
 $(004, \text{Mel}, \text{Gibson}, \text{Australia})\}$

Definizioni



Considerazioni

- Il valore di n è finito
 - Rappresentazione finita delle informazioni
- La cardinalità dei domini può essere considerata infinita
 - Si pensi all'insieme dei cognomi delle persone
 - Può essere utile considerare che esista una n-upla non presente nella relazione
- I domini possono essere
 - Tutti dello stesso tipo
 - Di tipo diverso
 - Non tutti dello stesso tipo

Definizioni



Proprietà di una relazione

1. non è definito nessun ordinamento tra le tuple di una relazione;
2. ogni tupla è distinta da un'altra;
3. esiste una corrispondenza di tipo posizionale tra i valori interni ad una tupla ed i relativi domini:

$$t = (v_1, v_2, \dots, v_n) \mid v_1 \in D_1, v_2 \in D_2, \dots, v_n \in D_n$$



Considerazioni

è possibile eliminare la proprietà 3 se:

- o ad ogni occorrenza di dominio D_i di una relazione si associa un differente **attributo** A_i che permette di identificare e qualificare il ruolo del dominio

- o Da quanto detto segue la definizione di **tupla**

$$t = \{ \langle v_1, A_1 \rangle, \langle v_2, A_2 \rangle, \dots, \langle v_n, A_n \rangle \}$$

per cui:

3. **non esiste nessun ordinamento all'interno delle tuple.**



Rappresentazione di una relazione

- o Una relazione può essere rappresentata naturalmente attraverso le **tabelle** dove
 - ogni riga è una tupla
 - ogni colonna è data dai valori relativi ad un certo attributo che ne è l'intestazione.
- o **Non tutte le tabelle rappresentano relazioni:** lo sono se e solo se sono soddisfatte le proprietà 1 e 2.



Tabelle e Relazioni

- o La seguente tabella corrisponde ad una relazione con i seguenti

ATTRIBUTO DOMINIO

Codice {001, 004, 005, 006}

Nome {Mel, Pedro, Federico}

Cognome {Almodovar, Gibson, Fellini},

Nazionalità {Italia, Spagna, Australia}.

Codice	Nome	Cognome	Nazionalità
001	Pedro	Almodovar	Spagna
004	Federico	Fellini	Italia
005	Mel	Gibson	Australia



Schema di Relazione

Dato un insieme di nomi di attributi

$$X = \{A_1, A_2, \dots, A_n\},$$

si definisce "schema di relazione" **un nome R, seguito da un insieme di nomi di attributi X:**

$$R(X) = R(A_1, A_2, \dots, A_n)$$



Esempi di Schema di Relazione

- Sono esempi di schema di relazione:

AUTORI (Codice, Nome, Cognome, Nazionalità)

FILM (Autore, Titolo, Attore Protagonista)



Relazione su uno schema

- Si definisce **relazione r** su uno schema di relazione $R(X)$ **una istanza di $R(X)$**

Nome	Cognome	Telefono
Angelo	Chianese	0817683827
Antonio	Picariello	0817683826

← Schema

← Istanza

- Schema: componente intensionale
- Istanza: componente estensionale



Informazioni incomplete

- Il modello relazionale impone strutture rigide alle informazioni:
 - Una relazione è un insieme di tuple omogenee (sullo stesso schema)
 - Per alcune tuple può accadere che non sia definito il valore di alcuni attributi

Come gestire l'assenza di informazione?



- Riempendo i campi con valori opportuni?
- E come sceglierli?



Informazioni incomplete

- In questi casi, si è soliti estendere i domini delle relazioni con un valore speciale, detto NULL

$$D_i = D_i \cup \text{NULL}.$$

Con il valore NULL si intende prendere in considerazione una assenza di informazione che può essere dovuta a diversi fattori:

- il dato c'è ma non lo conosco (valore sconosciuto)
- il dato non può esserci, in quanto non è applicabile ad una tupla (valore inesistente)
- non so dire nulla (valore senza informazione cioè sconosciuto o inesistente).



Esempio

Ipotesi: ogni studente è dotato di cellulare, mentre il professore è dotato di cellulare e di telefono di ufficio.

Nome	Ruolo	Cellulare	Ufficio
Giacomo	STUD	335123123	NULL
Antonio	PROF	335123131	0817683826
Marta	STUD	NULL	NULL
Isabella	NULL	334123123	NULL

- nella prima tupla NULL al telefono ufficio per uno studente, indica la non applicabilità dell'informazione;
- la terza tupla, presenta NULL nel campo cellulare di Marta, una *STUD*, pertanto il valore è sconosciuto.
- l'ultima tupla, non potendo stabilire se Annarita è *STUD* o *PROF*, non si può dire nulla (senza informazione).



Basi di dati e vincoli di integrità

Un vincolo di integrità o *integrity constraint*, è una **regola** che **ogni** istanza di uno schema di relazione **deve rispettare** affinché i suoi dati siano corrispondenti al modello della realtà che una BD cattura.



Definizioni

- Schema di una Base di Dati:**

È costituito da:

- il nome della base di dati *BD*,
 - dagli schemi di relazione $R_1(X_1), R_2(X_2), \dots, R_n(X_n)$,
 - un insieme *IC* di regole di integrità.
- Base di Dati Relazionale:**
- È una istanza di uno schema di base di dati che soddisfa l'insieme *IC* di regole.



Considerazioni

- Se una istanza soddisfa tutti i vincoli di integrità specificati nello schema della base di dati, si parla allora di **istanza legale della base di dati**.
- Compito del DBMS è verificare i vincoli di integrità generando istanze corrette.



Tipi di vincoli

- Si possono imporre vincoli che coinvolgono una singola relazione:
 - Vincoli intrarelazionali
 - sui valori di un attributo (vincoli di dominio)
 - su più attributi della tupla (vincoli di tupla)
 - Interessano tutte le tuple, l'una indipendentemente dalle altre
 - di chiave (per l'identificazione univoca di una tupla)
- Oppure imporre vincoli che coinvolgono più relazioni:
 - Vincoli inter-relazionali
 - consentono di verificare la validità dei valori degli attributi inseriti in una relazione per correlarla ad un'altra.



Vincoli intrarelazionali

- vincoli di integrità **intra-relazionale** sono vincoli espressi attraverso condizioni logiche che devono essere soddisfatte all'interno di una singola relazione:
 - Vincolo di dominio.**
 - Una regola che deve essere soddisfatta dai valori di un fissato attributo di una relazione
 - Vincolo di tupla.**
 - Una condizione logica che coinvolge più attributi all'interno della stessa tupla.



Esempio

NomeStudente	NomeEsame	Voto	Lode
Pippo	TSI	30	SI
Paolo	TM	28	NO
Marta	EI	18	SI
Maria	AI	33	NO

$18 \leq \text{Voto} \leq 30$	$(\text{Voto} \geq 18) \text{ AND } (\text{Voto} \leq 30)$	dominio
La lode è ammissibile se voto è uguale a 30	$\text{NOT} (\text{Lode} = \text{'SI'} \text{ AND } \text{Voto} \neq 30)$	tupla
NomeStudente non può essere NULL	$\text{NOT}(\text{NomeStudente} = \text{NULL})$	Dominio



Notazione importante

Sia i una tupla definita su un insieme di attributi X . Con la notazione $t_i[A]$ definiamo il valore della tupla i relativamente all'attributo $A \subseteq X$.

È possibile estendere la stessa notazione ad un sottinsieme di attributi $Y \subseteq X$: in questo caso con il termine $t_i[Y]$ indichiamo il valore della tupla i ristretta ai soli attributi Y .



Esempio

NomeStudente	NomeEsame	Voto	Lode
Pippo	TSI	30	SI
Paolo	TM	28	NO
Marta	EI	18	SI
Maria	AI	33	NO

Considerando la seconda tupla di questa relazione si ha:

$$t_2 [\text{NomeStudente}] = \text{Paolo}$$

$$t_2 [\text{NomeStudente}, \text{Voto}] = (\text{Paolo}, 28).$$



Superchiave

Sia dato uno schema di relazione $R(X)$ e sia SK un sottinsieme di attributi di X .

Diciamo che SK è una **superchiave** di una relazione r sullo schema $R(X)$ se **per ogni istanza** vale:

$$\forall t_i, t_j \in r, i \neq j \rightarrow t_i[SK] \neq t_j[SK]$$



Chiave

- Un sottinsieme K di attributi X è chiave per r se è una **superchiave minimale** di r (cioè togliendo un qualsiasi attributo da K , K non è più superchiave).



Esempio

Consideriamo lo schema di relazione:

STUDENTI (Matricola, Nome, Cognome, Nascita, CorsoDiStudio)

o Superchiavi

- Matricola, Nome, Cognome
- Nome, Cognome, Nascita
- Matricola, Nome
- Matricola
- Matricola, Nome, Cognome, Nascita, CorsoDiStudio

o Superchiavi minimali (chiavi)

- Nome, Cognome, Nascita
- Matricola

o Chiave primaria

- Matricola



Note

1. in una relazione esiste sempre **almeno una** (super) **chiave** - la tupla è sicuramente una superchiave per la proprietà di unicità delle tuple;
2. in generale, in una relazione è possibile individuare chiavi differenti.

Tra tutte le possibili chiavi, si sceglie sempre una chiave detta **chiave primaria della relazione**.



Integrità dell' entità

Nessun valore di una chiave primaria può essere nullo

Un DBMS relazionale in presenza di una nuova tupla per r con valore NULL nel campo della chiave primaria **non permette** l'inserimento della tupla in r.



Esempio schema di relazione con chiave primaria

Dall'esempio precedente relativo alla relazione STUDENTI, avendo scelto Matricola come chiave primaria, abbiamo la notazione:

STUDENTI (Matricola, Nome, Cognome, Nascita, CorsoDiStudio)



Vincoli interrelazionali

- In una base di dati solitamente si distribuisce l'informazione su relazioni differenti in modo da esprimere differenti concetti in differenti relazioni evitando in tal modo:
 - ridondanze nei dati
 - eventuali inconsistenze nell'aggiornamento dei dati.
- La distribuzione delle informazioni richiede un meccanismo che permetta di associare dati presenti in una tabella con quelli di un'altra tabella .



Esempio

STUDENTI

Matricola	Nome	Cognome	Indirizzo
150	Alex	Del Piero	Via dei Palloni, 30
151	Martina	Stellina	Via del Cielo, 40
142	Giovanni	Senzaterra	Via Crociate, 30

CORSI

Codice	Corso
TSI	Tecnologia dei Sistemi Informatici
EI	Elementi di Informatica
A1	Analisi Matematica 1

CARRIERE

MatStudente	CodiceCorso	Data	Voto
150	TSI	10/10/04	30
150	A1	10/09/05	28
151	TSI	10/10/05	28



Esempio

- **MatStudente** (di **CARRIERE**) è definito sullo stesso dominio dell'attributo **Matricola** (chiave primaria di **STUDENTI**)
- **CodiceCorso** (di **CARRIERE**) è definito sullo stesso dominio dell'attributo **Codice** (chiave primaria di **CORSI**).
- **MatStudente** e **CodiceCorso** sono dette chiavi esterne di **CARRIERE** in quanto "estranee" al concetto espresso dalla relazione **CARRIERE**
- Per l'integrità dei dati:
 - un valore di una **MatStudente** in **CARRIERE** deve essere anche presente come valore di **Matricola** in **STUDENTE**
 - un valore di **CodiceCorso** in **CARRIERE** deve essere anche presente come valore di **Codice** in **CORSI**



Integrità referenziale

Date r_1 ed r_2 , non necessariamente differenti, con r_1 dotata di chiave esterna FK relativa alla chiave primaria PK della relazione r_2 .

Si dice che tra r_1 ed r_2 sussiste un vincolo di integrità referenziale se, ogni occorrenza di FK in $t_1 \in r_1$

- è **NULL** oppure
- $\exists t_2 \in r_2 \mid t_2[\text{PK}] = t_1[\text{FK}]$



Concetto di integrità referenziale

Per ogni occorrenza a valore non nullo della chiave esterna nella tabella *referente* deve essere presente nella tabella *riferita* un ugual valore di chiave (primaria).



Integrità referenziale: sintassi

- Al fine di evidenziare i vincoli di integrità referenziale si aggiunge accanto all'elenco degli attributi che fungono da chiave esterna il nome della relazione riferita, cioè:

```
< ChiaveEsterna > ::=
< ElencoAttributi > : < NOME_TABELLARIFERITA >
```



Notazione relazionale dell'esempio

- Con riferimento all'esempio precedente lo schema relazionale della BD viene così espresso:

STUDENTI (Matricola, Nome, Cognome, Indirizzo)

CORSI (Codice, Corso)

CARRIERE (MatStudente: STUDENTI, CodiceCorso: CORSI, Data, Voto)

- Per completare lo schema occorre, per ogni relazione, esplicitare gli eventuali altri vincoli intrarelazionali.



Esempio di Base di Dati Relazionale

GIOCATORI

CodTessera	Nome	Cognome	Ruolo	Età	Squadra
------------	------	---------	-------	-----	---------

SQUADRA

Nome	ColoriSociali	AnnoDiFondazione	Stadio
------	---------------	------------------	--------



Vincoli intrarelazionali

- $IC_1(\text{GIOCATORI}) \leftarrow \text{Ruolo} \in \{ \text{attaccante, difensore, portiere, centrocampista} \}$
- $IC_2(\text{GIOCATORI}) \leftarrow \text{Età} \leq 100 \wedge \text{Età} \geq 0$
- $IC_1(\text{SQUADRE}) \leftarrow \text{Nome} \in \{ \text{'Ascoli', 'Cagliari', ... 'Udinese'} \}$
- $IC_2(\text{SQUADRE}) \leftarrow \text{AnnoDiFondazione} \leq 2005 \wedge \text{AnnoDiFondazione} \geq 1800$

Definizioni



Esempio di Base di Dati Relazionale

- Seconda Tupla NON VALIDA (valore di ruolo)

CodTessera	Nome	Cognome	Ruolo	Età	Squadra
00001	Francesco	Totti	attaccante	29	Roma
00002	Vincenzo	Moscato	docente	25	Avellino

- Seconda Tupla NON VALIDA (valore di età)

CodTessera	Nome	Cognome	Ruolo	Età	Squadra
00001	Francesco	Totti	attaccante	29	Roma
00002	Antonio	Picariello	portiere	101	Avellino

Definizioni



Esempio di Base di Dati Relazionale

- Seconda Tupla NON VALIDA (valore di NOME)

Nome	ColoriSociali	AnnoDiFondazione	Stadio
Juventus	Bianco-Nero	1910	Delle Alpi
Napoli	Azzurro	1926	San Paolo

- Seconda Tupla NON VALIDA (valore di Anno di Fondazione)

Nome	ColoriSociali	AnnoDiFondazione	Stadio
Juventus	Bianco-Nero	1910	Delle Alpi
Milan	Rosso-Nero	1200	San Siro

Definizioni



Scelta delle chiavi

GIOCATORE

CodTessera	Nome	Cognome	Ruolo	Età	Squadra
00001	Francesco	Totti	attaccante	29	Roma
00002	Fabio	Cannavaro	difensore	30	Juventus
00003	Alessandro	Del Piero	attaccante	30	Juventus
00004	Francesco	Coco	difensore	29	Livorno

SQUADRA

Nome	ColoriSociali	AnnoDiFondazione	Stadio
Juventus	Bianco-Nero	1910	Delle Alpi
Ascoli	Bianco-Nero	1920	Del Duca
Inter	Nero-Azzurro	1910	San Siro
Milan	Rosso-Nero	1920	San Siro

- **CHIAVI di GIOCATORE:**
 - **CodTessera** (chiave primaria)
 - **Nome, Cognome**
- **CHIAVE ESTERNA DI GIOCATORE**
 - **Squadra** della relazione **GIOCATORE** che si riferisce alla chiave primaria **Nome** della relazione **SQUADRA**.

Definizioni



Definizione dei dati in SQL

- SQL è l'acronimo di Structured Query Language
- Versioni:
 - SQL-86
 - SQL2
 - SQL-92
 - SQL3
- SQL è un linguaggio di tipo **dichiarativo**

SQL comprende sia istruzioni per la definizione di dati (**DDL**) che per la loro manipolazione (**DML**).



CREATE TABLE

È usato per creare una nuova relazione: in esso si specificano:

- **il nome della relazione**
- **il nome ed il tipo dei suoi attributi**
- **i vincoli intra e interrelazionali**



Tipi di Dato

- **Numeric**
 - numeri interi (integer, int, smallint)
 - numeri reali a precisione differente in virgola fissa e in virgola mobile, (real, float, double precision).
- **Stringhe**
 - di caratteri di lunghezza fissa (char(n))
 - di caratteri a lunghezza variabile (varchar(n))
 - di bit a
 - lunghezza fissa (bit(n))
 - variabile (bitvarying(n)).
- **Data e Ora** permette di esprimere data e ora.
 - Ha dieci posizioni aventi per componenti *YEAR*, *MONTH* e *DAY* in vari formati
 - Il tipo time ha otto posizioni con i componenti *HOUR*, *MINUTE* e *SECOND*.
 - Il tipo interval permette, invece, di stabilire un valore temporale relativo.



Vincoli

- **NOT NULL** specifica il vincolo che il valore dell'attributo deve essere diverso da NULL.
- **UNIQUE** specifica il vincolo che il valore (o i valori) dell'attributo (o degli attributi) specificato in una tupla deve essere unico (vincolo generico di chiave).
- **primary key** specifica che uno o più attributi sono chiave primaria di una relazione:
 - per default, è not NULL e unique.
- **foreign key** permette di specificare un vincolo di integrità referenziale.
 - La specifica [opzionale] delle politiche di violazione del vincolo di chiave esterna avviene attraverso le opzioni on delete set null, on delete set default, on delete cascade, on delete no action, on update cascade.



Sintassi

Create table nomeTabella

```
(
  nomeAttributo Dominio [Default][Vincoli]
  {,nomeAttributo Dominio [Default][Vincoli]}
  [,altriVincoli]
)
```



Esempio BD carriere studenti

- o Notazione relazionale dello schema:
(a meno di ulteriori vincoli intrarelazionali)

STUDENTI (Matricola, Nome, Cognome, Indirizzo)

CORSI (Codice, Corso)

CARRIERE (MatStudente: STUDENTI, CodiceCorso:
CORSI, Data, Voto)



Esempio BD carriere studenti

Create table **CORSI**

```
(
  Codice varchar(10),
  Corso varchar(100),
  primary key (codice)
)
```

Create table **STUDENTII**

```
(
  Matricola integer,
  Nome varchar(50),
  Cognome varchar(50)
  Indirizzo varchar(150)
  primary key(Matricola)
)
```



Esempio BD carriere studenti

Create table **CARRIERE**

```
(
  MatStudente integer,
  CodiceCorso varchar(10),
  data date,
  voto integer,
  primary key(MatStudente,CodiceCorso),
  foreign key(MatStudente) references
  STUDENTI(Matricola)
  on delete CASCADE,
  foreign key(codiceCorso) references
  CORSI(Codice)
  on delete SET NULL
)
```



Esempio BD Campionato calcio

- o Notazione **completa** dello schema relazionale :

SQUADRE(Nome, ColoriSociali, AnnoDiFondazione, Stadio)
 Nome ∈ { Ascoli, ..., Udinese }
 AnnoDiFondazione ≤ 2005 ∧ AnnoDiFondazione ≥ 1800

GIOCATORI(CodTessera, Nome, Cognome, Ruolo, Età,
 Squadra: SQUADRE)
 Nome ≠ NULL
 Cognome ≠ NULL
 Ruolo ∈ {attaccante, difensore, portiere, centrocampista}
 Età ≤ 100 ∧ Età ≥ 0

Definizioni



Esempio BD Campionato calcio

Create table SQUADRE

```
(
  Nome varchar(50) Check(Nome='Ascoli' OR
  ...OR Nome='Udinese') ,
  ColoriSociali varchar(30),
  AnnoDiFondazione integer
  Check( AnnoDiFondazione >= 1800 AND
  AnnoDiFondazione <= 2005),
  Stadio varchar(100)
  primary key (Nome)
)
```

Definizioni



Esempio BD Campionato calcio

Create table GIOCATORI

```
(
  CodTessera varchar(10)
  Nome varchar(50) NOT NULL,
  Cognome varchar(50) NOT NULL
  Ruolo varchar(15) CHECK(Ruolo='Portiere' or ...
  or Ruolo='Attaccante'),
  Età integer Check( Età>0 AND Età <= 100),
  Squadra varchar(50),
  primary key (CodTessera),
  foreign key(Squadra) references SQUADRE(Nome)
  on delete SET NULL
)
```

Definizioni

