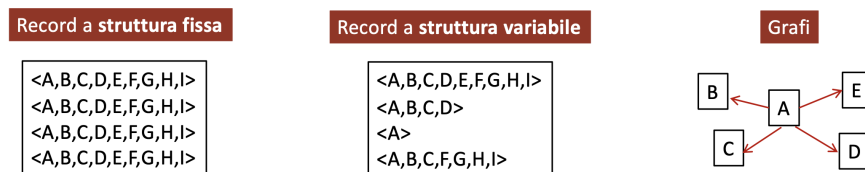


Il modello relazionale

Esiste una architettura standard conseguita nei DBMS articolata su **tre livelli**, detti rispettivamente **logico**, **interno** ed **esterno**, di cui di seguito si analizzano le caratteristiche:

- **Schema Esterno**: costituisce la descrizione di una porzione di interesse del *data base* usufruendo del *modello logico*
- **Schema Logico**: costituisce una descrizione dell'intera basi di dati per mezzo del *modello logico* adottato dal DBMS
- **Schema Interno**: costituisce la rappresentazione dello *schema logico* per mezzo di strumenti di memorizzazione, indicando come gli stessi dati debbano essere veramente organizzati, gestiti poi consecutivamente dal DBMS (*spesso le strutture dati usate sono strutture fisse, strutture variabili oppure grafi*)



Modello Logico

Il *modello logico* pone un formalismo qualora i dati siano accettabili o meno, ossia se possano essere memorizzati all'interno delle relazioni o tabelle. Tuttavia è bene considerare come tale *formalismo* indica regole generali a cui attenersi, indipendentemente dal *data base* creato.

Un semplice esempio è favorito dalla creazione di relazioni in grado di mantenere al loro interno dati riferiti a voti di esame; un vincolo potrebbe essere dato dal fatto che possano essere memorizzati valori che abbiano una votazione maggiore o uguale 18 e inferiore o uguale a 30L per essere memorizzati nel *data base*.

Ogni singola progettazione di un DBMS è relativa all'impegno di due **proprietà**, le quali possono essere adottate come dei veri e propri *vincoli*, quali:

- *Indipendenza fisica*: ossia un programma non deve conoscere l'*organizzazione fisica* dei dati
- *Indipendenza logica*: un programma può vedere i dati tramite opportuni *schemi esterni*, quali relazioni o view

Differenti sono i *modelli logici* adottati, ma dove principalmente sono adoperate due grandi famiglie quali: **modelli relazionali** e **modelli scheme-less**, dei due il primo citato risulta il più diffuso.

Modello Relazionale

Le ragioni principali che rendono il **modello relazionale** il più utilizzato sono:

- garantisce la separazione tra il livello fisico e logico

- è caratterizzato da un uso intuitivo, connesso a forti fondamenta legate ad algebra di base

Definizione informale

I dati sono organizzati in *record* di dimensione fissa, e divise in *tabelle*, dette **relazioni**.

Come citato dalla definizione, le strutture usate per salvare qualunque dato sono definite *record*, memorizzati all'interno di una tabella. La dimensione è *fissa*, poichè per ogni singolo record è mantenuto lo stesso numero di **attributi**, ossia la stessa molteplicità di *colonne*. Inoltre il connubio dettato dal [*nome tabella e nome attributi*] indica lo **schema** della relazione; infine, i record o le *righe* della tabelle sono denominate **istanze** della relazione.

Nome	Codice Corso	Nome Docente
Basi di dati	0121	M. Di Felice
Programmazione	1213	C. Laneve
Sistemi Operativi	1455	D. Sangiorgi

Il modello relazionale gode di alcuni *vincoli*, i quali non possono essere aggirati pur di ottenere una struttura corretta del data base. Tali vincoli sono:

- L'**ordinamento** delle righe, come anche delle colonne, è irrilevante, l'importanza ricade solo sul contenuto
- Non possono esistere **attributi uguali**
- Non possono esistere **righe uguali**
- I dati di ogni colonna devono essere **omogenei** fra loro

Ponendo uno sguardo sulle colonne, ogni attributo dispone di un **dominio**, che definisce l'*insieme di valori validi* per quell'attributo, inoltre, differenti attributi possono condividere lo stesso dominio. Un esempio può essere ($\text{dom}(\text{Nome}) = \text{string}$).

I dati gestiti dal modello relazionale oltre ad essere omogenei fra loro, qualora posti nella stessa colonna, sono **dati strutturati**, tutti i valori della stessa tabella condividono la stessa struttura, quindi non sarà mai possibile gestire istanze che abbiano un numero inferiore di domini.

Sono stati osservati i vincoli principali su cui pone la realizzazione e concretizzazione dello scheletro della base di dati, tuttavia spesso l'importanza ricade sul rispetto della **Prima Forma Normale**.

La *Prima Forma Normale* indica delle proprietà che una qualsiasi relazione deve disporre. Innanzitutto è stabilito come ogni attributo sia definito su un *dominio atomico* e non su *domini complessi*; ossia, i dati memorizzati non possono essere valori combinati fra loro, da cui solitamente sono adoperati processi di *normalizzazione* per rendere ogni attributo atomico.

Ulteriore regola, ma non vincolante, indica la buona pratica di evitare la ripetizione di stessi dati per poter occupare il minimo spazio di memoria; infatti, la strategia adottata da modelli relazionali è garantita dall'uso di puntatori, dove la visualizzazione di stessi dati in differenti tabelle, non è da imputare alla maggiore memoria occupata, ma rispetto a stessi riferimenti

a celle di memoria.

Spesso capita di osservare medesime informazioni contenute in differenti relazioni le quali sono **correlate** logicamente fra loro, dove nel modello relazionale questi riferimenti sono costruiti mediante gli stessi **valori**.

CORSI			ESAMI		
Nome Corso	Codice Corso	Codice Docente	Corso	Studente	Voto
Basi di dati	0121	00	0121	4324235245	30L
Programmazione	1213	01	1213	4324235245	25
Sistemi Operativi	1455	02	1213	9854456565	18

STUDENTI			
Matricola	Cognome	Nome	Data Nascita
4324235245	Rossi	Giorgio	12/12/1987
6247673587	Bianchi	Michele	04/12/1987
9854456565	Verdi	Marco	11/04/1988

Tuttavia nella **pratica** la progettazione di una base di dati non traduce a priori le informazioni di interesse in un modello relazionale. Tipicamente ciò che è noto sono i requisiti funzionali e non il modello relazionale; occorre quindi individuare una serie di passaggi che rendano concreto lo sviluppo del modello logico.

La nozione di relazione come sinonimo di tabella in questo contesto non è da tralasciare. In ambito matematico una relazione può essere definita come un sottoinsieme del prodotto cartesiano di una qualsiasi coppia di insiemi. Tuttavia occorre descrivere il concetto di relazione in maniera più formale.

Relazione matematica

Definizione

Il **prodotto cartesiano** degli insiemi D_1, D_2, \dots, D_N è definito come l'**insieme delle tuple ordinate** (d_1, d_2, \dots, d_N) , con $d_i \in D_i, \forall i \in 1, 2, \dots, n$.

Per esempio, dati gli insiemi $A = [1, 2, 4]$ e $B = [a, b]$, il prodotto cartesiano dei due insiemi è costituito dall'insieme di tutte le possibili coppie in cui il primo elemento appartiene ad A e il secondo a B . Si ottengono così sei coppie:

$$(1, a), (1, b), (2, a), (2, b), (4, a), (4, b)$$

Definizione

Dati n insiemi D_1, D_2, \dots, D_N una **relazione matematica** sugli insiemi D_1, D_2, \dots, D_N è definita come un **sottoinsieme del prodotto cartesiano** $D_1 \times D_2 \times \dots \times D_N$.

Quindi una *relazione matematica* sugli insiemi A e B è un sottoinsieme di $A \cdot B$, questo sottoinsieme, a livello semplificativo, può essere:

$$(1, a), (1, b), (2, a), (2, b), (4, a), (4, b)$$

Il numero di elementi della relazione viene chiamato *cardinalità* della relazione. Mediante la stessa cardinalità sono adottate ulteriori annotazioni affini rispetto all'ambito di relazioni e tabelle. Rispetto alla teoria degli insiemi, sono stabilite alcune operazioni tipiche, le stesse sfruttate anche in questo contesto:

- *Intersezione*: il sottoinsieme conterrà al suo interno solo elementi che risultino essere in comune tra qualsiasi totalità di insiemi dati. Dove di seguito è riportata la cardinalità dell'insieme:

$$|A \cap B| = \{0, \min(|A|, |B|)\}$$

- *Unione*: il sottoinsieme conterrà tutti gli elementi, eccetto i duplicati, rispetto agli insiemi presi in considerazione. Dove di seguito è riportata la cardinalità:

$$|A \cup B| = \{\max(|A|, |B|), |A| + |B|\}$$

- *Differenza*: il sottoinsieme conterrà gli elementi del primo insieme preso in considerazione eccetto quelli corrisposti all'interno del secondo insieme. Dove di seguito è riportata la cardinalità:

$$|A \setminus B| = \{0, |A|\}$$

- *Prodotto Cartesiano*: il sottoinsieme conterrà gli elementi del primo insieme e del secondo insieme. Dove di seguito è riportata la cardinalità:

$$|A \cdot B| = \{|A| \cdot |B|\}$$

Quindi una tabella può essere osservata come un sottoinsieme del prodotto cartesiano, anche se risulta contraddittorio dal punto di vista simmetrico. Infatti, a livello di definizione, indicare il prodotto cartesiano $A \cdot B$ oppure $B \cdot A$ **non stabilisce** la stessa entità, anche se ciò, tradotto rispetto ad una rappresentazione grafica, si rispecchia in un posizionamento differente delle colonne o domini all'interno delle relazioni.

Informazioni incomplete

All'interno di una relazione, le **ennuple** di dati, ossia le differenti *row*, devono essere omogenee fra loro. Tuttavia, in alcuni casi, l'*attributo* associativo per una determinata *ennupla* potrebbe non essere noto o addirittura inesistente.

Una soluzione consiste nel battezzare *valori speciali*, ossia colmare informazioni mancanti. Un primo approccio è dato dalla creazione di valori speciali per ogni attributo oppure, molto più utilizzato, etichettare tutto ciò che sia mancante con il **valore NULL**. Il valore *NULL* garantisce una netta soluzione per dati mancanti, tuttavia è buona pratica limitarne il numero, a causa dell'elevato spreco di memoria che potrebbe provocare. Nella progettazione di basi di dati esiste un *trade-off* tra le tabelle e il valore NULL, ossia comporre ulteriori relazioni, differenziando in questo modo record che siano in grado o meno di compensare tutti i domini richiesti. **Potrebbe starci mettere un'altro esempio grafico**

Vincoli di integrità

Una fase fondamentale è data dalla capacità di stabilire quali istanze possano essere **inserite** all'interno di una tabella. Infatti, molto spesso, alcuni record di una relazione possono considerarsi illeciti, ossia sono caratterizzati da valori posti al di fuori delle restrizioni fissate dai differenti attributi dello *schema* della base di dati.

Perciò è adottato un meccanismo in grado di provvedere alla problematica precedente, mediante la nozione di **vincolo**.

Definizione informale

Un *vincolo* è una funzione booleana, che associa ad una istanza di una base di dati un **valore di verità**.

Per cui un vincolo predispone quali siano istanze lecite o meno, ossia se soddisfino tutti i requisiti; solamente soddisfatti tutti i vincoli sarà possibile inserire l'istanza nella relazione. Tenzialmente si adotta una duplice suddivisione tra *vincoli intra-relazionali*, ossia relativi alla stessa relazione, e *vincoli inter-relazionali*, tra relazioni differenti.

Vincoli intra-relazionali

Come già detto i **vincoli intra-relazionali** considerano una singola tabella. A loro volta si suddividono in due entità:

- *Vincoli di ennupla*: regole poste su ciascuna riga della tabella, quali espressioni algebriche oppure funzioni booleane. Se l'espressione o funzione adoperata dovesse corrispondere un risultato negativo relativo alla caratterizzazione del nuovo record da memorizzare, non sarà consentito l'inserimento dell'istanza all'interno della relazione.
- *Vincoli di chiave*: requisiti posti sulle colonne o domini, i quali a loro volta si intrecciano rispetto a tre possibili definizioni, per cui sono vincoli più articolati rispetto ai precedenti.

Un **vincolo di chiave** si articola in tre definizioni consecutive. La prima osservata è in relazione al concetto di **superchiave**.

Definizione informale

Una *superchiave* è una colonna o più colonne se non sono presenti valori ripetuti o duplicati.

Il concetto di **chiave** è adoperato per rendere univoca, durante una qualsiasi fase di ricerca, una ennupla, prendendo in considerazione un insieme di possibili attributi. Per cui, come da definizione, la *superchiave* è in grado di rispettare quanto voluto, tuttavia la decisione di quali colonne scegliere deve garantire la mancanza di istanze identiche, altrimenti provocherebbe un dissenso rispetto all'univocità.

E' buona pratica progettare le chiavi a livello di schema, ossia osservando la struttura della relazione cercando di accomunare tutte le istanze. Tuttavia la creazione di una *superchiave* potrebbe prendere in considerazione colonne ridondanti, ossia rendere istanze univoche ma utilizzando attributi non affini all'obiettivo di una *chiave*. Per questo si adopera il passaggio da *superchiave* a *superchiave minimale*.

Definizione informale

Una *chiave* è una superchiave minimale di r , ossia della relazione.

Per fare in modo di ottenere una *chiave minimale*, occorre che la scelta su cui ricade il numero di colonne sia il minimo possibile; per cui presi in considerazione certi attributi, la progettazione della chiave avviene osservando se preso un numero sempre più ristretto di attributi sia possibile o meno rendere una qualsiasi ennupla univoca durante una qualsiasi operazione. Scegliere una *chiave* piuttosto di una superchiave avviene per rendere maggiormente facile la correlazione ad ennuple e la connessione tra differenti relazioni. Un esempio associato a tale contesto potrebbe essere dato dalla presenza piuttosto consistente di valori NULL nella tabella. I valori NULL, come già detto, potrebbero rappresentare un'arma a doppio taglio; infatti ipotizzando quanto detto, non è più vero che le chiavi riescano a identificare unicamente le differenti istanze e a stabilire connessioni tra relazioni.

E' necessario introdurre un ulteriore meccanismo che vieti l'uso di valori NULL per la costruzione di superchiavi minimali.

Definizione informale

Una *chiave primaria* è una chiave, ossia rappresenta già una superchiave e una superchiave minimale, di una relazione su cui non sono ammessi valori NULL.

Quindi l'uso di *chiavi primarie* risulta essere la soluzione migliore, permette univocità delle ennuple e le connessioni tra relazioni. Tuttavia, a volte potrebbe capitare che sia impossibile scegliere attributi che non abbiano valori NULL. Semplicemente, si adottano degli *identificativi progressivi*, ossia attributi di tipo intero, i quali in completa autonomia ad ogni nuovo inserimento di istanze corrette associa un valore incrementato, rendendo così unico il collegamento.

RICOVERI					
Codice	Nome	Cognome	Reparto	Data ricovero	Data dimissione
1	NULL	NULL	Rianimazione	NULL	10/02/2012
2	Bianchi	Marco	Dermatologia	NULL	10/04/2012
3	Bianchi	Marco	Ortopedia	08/01/2011	10/02/2012
4	Verdi	Giovanni	Ortopedia	08/01/2011	NULL
5	Verdi	Matteo	Ortopedia	23/01/2011	NULL

Vincoli inter-relazionali

Un modello relazionale è adottato soprattutto per garantire molte connessioni fra relazioni della stessa base di dati, collegamenti espressi mediante attributi di differenti tabelle dello stesso tipo. Per cui, potrebbe risultare utile porre delle **restrizioni** sulle *dipendenze* tra relazioni.

Definizione informale

Ogni **riga** della tabella **referenziante** si collega al **massimo** ad un **riga** della tabella **referenziata**, sulla base di simili attributi scelti per porre un punto in comune.

Il **vincolo di integrità inter-relazionale** interviene proprio per determinare delle regole

a cui sottostare.

Definizione informale

Un *vincolo di integrità referenziale* fra gli attributi X di una relazione R_1 e una relazione R_2 impone ai valori, diversi da NULL per definizione, X di R_1 di comparire come valori della *chiave primaria* di R_2 .

VOLI					
Codice	Data	Partenza	Arrivo	Compagnia	Pilota
LH21	10/02/2012	FCO	BOQ	Lufthansa	123
AZ21	10/02/2012	ATL	FCO	Alitalia	124
AF	11/02/2012	BOQ	FKT	Air France	123

PILOTI		
IdPilota	Nome	Cognome
123	Mario	Rossi
124	Michele	Bianchi

AEROPORTI		
IdAeroporto	Citta'	#Piste
FCO	Roma	10
BOQ	Bologna	5
ATL	Atlanta	40

Come da immagine, in pratica, il vincolo consente di collegare fra loro relazioni mediante stessi dati che siano dello stesso dominio; considerando che tali valori nella tabella referenziante siano posti non necessariamente nella chiave primaria, contrariamente rispetto alla relazione referenziata, la quale deve mantenere i dati fittizi all'interno del dominio che componga la propria chiave primaria.

Potrebbe accadere che durante una fase di *aggiornamento*, alcuni record che compongono referenze fra relazioni vengano eliminati, oppure poste delle modifiche che violino i vincoli di integrità. In considerazione a quanto detto si adottano tre strategie:

- **NON** consentire l'operazione, qualora riguardi istanze utilizzate per la costruzione di referenze tra relazioni
- **Eliminazione a cascata**, ossia l'eliminazione di record all'interno della relazione *referenziante* deve provocare l'eliminazione delle stesse istanze utilizzate nella relazione *referenziata*, e viceversa
- **Inserimento** dei valori **NULL**, questo accade nei confronti di entrambe le relazioni che compongano la referenza, quindi eliminati determinati record occorre inserire nella tabella di riferimento nei corretti domini il valore di carattere speciale

VOLI					
Codice	Data	Partenza	Arrivo	Compagnia	Pilota
LH21	10/02/2012	NULL	BOQ	Lufthansa	123
AZ21	10/02/2012	ATL	NULL	Alitalia	124
AF	11/02/2012	BOQ	FKT	Air France	123

PILOTI		
IdPilota	Nome	Cognome
123	Mario	Rossi
124	Michele	Bianchi

AEROPORTI		
IdAeroporto	Citta'	#Piste
FCO	Roma	10
BOQ	Bologna	5
ATL	Atlanta	40