

Indice

1	Introduzione ai DBMS	1
2	Modello relazionale	1
3	Linguaggio SQL	3

1 Introduzione ai DBMS

Modello relazionale: organizza i dati in record di dimensione fissa mediante tabelle.
Sistema informativo (SI): componente di un'organizzazione il cui scopo è gestire le informazioni utili ai fini dell'organizzazione stessa.
DBMS: sistema software in grado di gestire collezioni di dati grandi, condivise e persistenti in maniera efficiente e sicura.
Base di dati: collezione di dati gestita da un DBMS.

1.1 Architettura a livelli del DBMS

- Ogni livello è indipendente dall'altro
- **Schema esterno:** come si presenta il DB, e come varia in base ai permessi di accesso.
 - **Schema logico:** come sono strutturati i dati e che relazioni hanno. In genere il modello logico utilizzato è quello relazionale, dove i dati sono organizzati in tabelle. Si usano delle regole per modellare eventuali vincoli e restrizioni sui dati.
 - **Schema fisico:** come/dove sono memorizzati i dati.

SQL è un linguaggio orientato ai dati, usato per il modello relazionale.
Un DBMS si usa quando:

- i dati sono condivisi da più utenti;
- i dati sono persistenti;
- i dati sono voluminosi e complessi;
- servono meccanismi di sicurezza e controllo degli accessi;

2 Modello relazionale

Siamo nello schema logico: definiamo le tabelle, le relazioni tra di esse e i vincoli sui dati.
Il modello relazionale è il più utilizzato, garantisce indipendenza tra i livelli e si basa su nozioni di algebra relazionale.

2.1 Struttura

I dati sono organizzati in record di dimensione fissa e divisi in tabelle (relazioni).

- Colonne: rappresentano gli attributi, hanno un nome e un tipo di dato.

- Intestazione della tabella (nome tabella + nome attributi): **schema** della relazione.
- Righe della tabella: istanze della relazione (ennuple)¹.

L'ordine delle righe e delle colonne non ha importanza, ma l'ordine degli attributi sì.

2.2 Vincoli sui dati della relazione

- non possono esistere attributi con lo stesso nome;
- non possono esistere righe uguali;
- i dati di una colonna devono essere omogenei (omogeneità di tipo);

Si possono avere schemi senza istanze, ma non istanze senza schema.

2.3 Prima forma normale (PFN)

Una relazione è in PMF se tutti gli attributi sono atomici, cioè non possono essere ulteriormente divisi.
Non è possibile omettere un valore da una ennupla: devono avere tutti i campi obbligatoriamente.

Una base di dati può essere costituita da molte tabelle. Spesso, le informazioni contenute in relazioni diverse sono correlate logicamente tra loro. Nel modello relazionale, i riferimenti tra dati in relazioni differenti sono espressi mediante valori.

Corsi			Esami		
Nome corso	Codice Corso	Nome Docente	Corso	Studente	Voto
Basi di dati	BD001	Mario Rossi	BD001	Luca Bianchi	30
Sistemi informativi	SI002	Giovanni Verdi	BD001	Anna Neri	28

Table 1: Esami contiene un codice che è lo stesso di corsi

Nella progettazione, bisogna tradurre le informazioni in dati del modello relazionale. Ci si chiede qualo dato devono essere gestiti e quante tabelle servono.

2.4 Schema matematico del modello relazionale

DEF: Dati n insiemi D_1, D_2, \dots, D_n , una relazione matematica su questi insiemi è un sottoinsieme del prodotto cartesiano $D_1 \times D_2 \times \dots \times D_n$.

¹Un'ennupla (o tupla) corrisponde ad un istanza della relazione.

DEF: Il prodotto cartesiano degli insiemi D_1, D_2, \dots, D_n è l'insieme di tutte le ennuple ordinate (d_1, d_2, \dots, d_n) con $d_i \in D_i, \forall i = 1, \dots, n$.

EX: Relazione
 $A = \{a, b, c, d, e\}, B = \{1, 2, 3\}$
 $A \times B = \{(a, 1), (a, 2), (a, 3), (b, 1), (b, 2), (b, 3), (c, 1), (c, 2), (c, 3), (d, 1), (d, 2), (d, 3), (e, 1), (e, 2), (e, 3)\}$
 $R_1 \subseteq A \times B = \{(a, 1), (a, 2), (a, 3)\}$
 $R_2 \subseteq A \times B = \{(a, 2), (b, 1), (d, 3), (e, 3)\}$
 A e B sono due tabelle con un solo campo. $a, b, c, \dots, 1, 2, 3$ sono le istanze. Il prodotto cartesiano unisce tutte le istanze facendo tutte le combinazioni possibili. Le istanze di $A \times B$ sono tutte le combinazioni di A e B . Le relazioni R_1 e R_2 sono sottoinsiemi del prodotto cartesiano, quindi sono relazioni.

Una ennupla su un di attributi X è una funzione che associa a ciascun attributo A in X un valore del dominio di A .
 $T[A]$ indica il valore dell'ennupla T nell'attributo A .

2.5 Informazioni incomplete

In una relazione le ennuple devono essere omogenee, ossia avere tutte la stessa struttura. Se il valore di un attributo non è noto, si usa il valore NULL. $T[A] \in A \vee \text{NULL} \forall$ attributo A .
Per definizione, il valore NULL non è uguale a nessun altro valore, nemmeno a se stesso².

2.6 Vincoli di integrità

I vincoli di integrità sono regole che limitano i valori che possono essere inseriti in una relazione.
Un vincolo è una funzione booleana che associa ad una istanza r di una base di dati definita su uno schema $R = \{R_1(x_1), \dots, R_{k(x_k)}\}$ un valore booleano. Un'istanza è lecita se soddisfa tutti i vincoli.

2.6.1 Vincoli intra-relazionali

I vincoli intra-relazionali sono regole che limitano i valori che possono essere inseriti in una singola relazione.
Vincoli di ennupla: questi vincoli esprimono condizioni su una ennupla,

considerata singolarmente. Possono essere espressi tramite espressioni algebriche o booleane.

EX: Vincolo di ennupla
`((voto>=18) and (voto<=30)) , not((lode=true) and (voto!=30)) , (saldo = entrate-uscite)`

Vincoli di chiave: una chiave è un insieme di attributi che consente di identificare in maniera univoca le ennuple di una relazione.

EX: Chiave
La matricola di uno studente: studenti(matricola, cognome, nome, data). Non esistono due studenti con la stessa matricola: data la matricola di uno studente è possibile risalire a tutti i suoi dati.

DEF: **Superchiave**
Un sottoinsieme di k attributi di una relazione è una superchiave se non contiene due ennuple distinte T_1 e T_2 con $T_1[k] = T_2[k]$. Nell'esempio di prima, matricola è una superchiave

DEF: **Superchiave Minimale**
La superchiave k è minimale $\iff \nexists k' \mid k \subseteq k' . k$ è la superchiave più piccola, non ne esiste un'altra che la contenga.

In una relazione esiste sempre almeno una superchiave alla peggio si prendono tutti i campi³.

Le chiavi servono per accedere a ciascuna ennupla della base di dati in maniera univoca e correlare dati tra relazioni (tabelle) differenti.

²NULL \neq NULL.

³Per definizione non possono esserci due ennuple uguali.

DEF: Chiave Primaria

Chiave di una relazione su cui non sono attesi valori NULL. Gli attributi che formano la chiave primaria sono per convenzione sottolineati.

EX: Chiave primaria

studenti(matricola, nome, cognome)

EX: Chiave primaria con più attributi

partita(squadra1, squadra2, data, punti1, punti2)

Ogni relazione deve disporre di una chiave primaria. Se tutti i campi presentano dei valori NULL, si aggiunge un codice univoco o un identificativo progressivo.

2.6.2 Vincoli inter-relazionali

Una base di dati può essere composta da molte relazioni collegate tra loro. I collegamenti tra relazioni differenti sono espressi tramite valori comuni in attributi replicati.

Ogni riga della tabella referenziante si collega al massimo ad una riga della tabella referenziata, in base ai valori comuni nell'attributo replicato.

tabella referenziata (chiave primaria)

tabella referenziante (chiave secondaria)

Un vincolo di integrità referenziale (chiave esterna) fra gli attributi x di R_1 e un'altra relazione R_2 impone ai valori su x in R_1 di comparire come valori della chiave primaria di R_2 .

Il vincolo garantisce che non ci siano riferimenti a elementi inesistenti: ogni valore usato come chiave esterna in una tabella deve esistere come chiave primaria nell'altra tabella a cui si riferisce.

2.7 Problemi

Se un operazione di aggiornamento o modifica causa violazioni dei vincoli di integrità su altre relazioni

- non si consente l'operazione;
- si elimina a cascata
- si inseriscono valori NULL

3 Linguaggio SQL

DEF: Relational DataBase Management System (RDBMS)

Un RDBMS è un software che consente di creare, gestire, modificare e interrogare basi di dati strutturate in forma relazionale e usa SQL per operare sui dati.

I linguaggi data-oriented permettono di implementare il modello relazionale in un RDBMS. Essi dispongono di UI, linguaggi basati su proprietà algebrico-logiche. Il più famoso è SQL.

Si applicano i concetti del modello relazionale, ma con delle differenze:

- si parla di tabelle e non di relazioni;
- il sistema dei vincoli è più espressivo;
- ci possono essere tabelle con righe duplicate se non c'è la chiave primaria;
- il vincolo di integrità referenziale è meno stringente.

3.1 SQL-DDL

Contiene i costrutti necessari per la creazione e modifica dello schema della base di dati.

<code>create database[if not exists] <nome DB></code>	crea DB
<code>drop database[if exists] <nome DB></code>	cancella DB
<code>create table NOMETABELLA(NOMEATTRIBUTO1 DOMINIO[<val default>][<vincoli> ...)</code>	crea tabella

Domini	
<code>character[<lunghezza max> [<lunghezza> alternativa <code>varchar(<lunghezza></code></code>	Se la lunghezza non è specificata accetta un singolo carattere
<code>- numeric[(Precisione [,Scala])]</code> - <code>decimal[(Precisione [,Scala])]</code>	I tipi numerici esatti consentono di rappresentare valori esatti, interi o con una parte decimale di lunghezza prefissata.
<ul style="list-style-type: none"> • <code>integer</code> • <code>smallint</code> 	
<ul style="list-style-type: none"> • <code>integer auto_increment</code> • <code>smallint auto_increment</code> 	La keyword <code>auto_increment</code> consente di creare campi numerici che si auto-incrementano ad ogni nuovo inserimento nella tabella.

<ul style="list-style-type: none"> float [<precision>] real double [<precision>] 	I tipi numerici approssimati consentono di rappresentare valori reali con rappresentazione in virgola mobile.
<ul style="list-style-type: none"> date [(Precisione)] time [(Precisione)] timestamp 	I domini temporali consentono di rappresentare informazioni temporali o intervalli di tempo.
boolean	I domini booleani consentono di rappresentare valori booleani
<ul style="list-style-type: none"> blob cblob 	I domini blob e cblob consentono di rappresentare oggetti di grandi dimensioni come sequenza di valori binari (blob) o di caratteri (cblob).
<pre>create domain NomeDominio as TipoDati [Valore di default] [Vincolo]</pre>	<p>Tramite il costrutto domain , l'utente può costruire un proprio dominio di dati a partire dai domini elementari.</p> <pre>CREATE DOMAIN Voto AS SMALLINT DEFAULT NULL CHECK (value >=18 AND value <= 30)</pre>

3.1.1 Vincoli

Per ciascun dominio o attributo è possibile definire dei vincoli che devono essere rispettati da tutte le istanze di quel dominio o attributo.

- Vincoli **intra-relazionali**:
 - vincoli generici di ennupla
 - vincolo not null
 - vincolo unique
 - vincolo primary key
- Vincoli **inter-referenziali**:
 - vincolo references

Vincoli	
check(<condizione>)	<p>Il vincolo viene valutato ennupla per ennupla.</p> <pre>VOTO SMALLINT CHECK((VOTO>=18) and (VOTO<=30))</pre>

not null	<p>Il vincolo not null indica che il valore NULL non è ammesso come valore dell'attributo.</p> <pre>NUMEROORE SMALLINT NOT NULL</pre>
<ul style="list-style-type: none"> Attributo Dominio [<default value>] unique : la superchiave è un solo attributo unique(Attributo1, Attributo2, ...) : la superchiave è composta da più attributi 	<p>Il vincolo unique impone che l'attributo/attributi su cui sia applica non presenti valori comuni in righe differenti, ossia che l'attributo/i sia una superchiave della tabella. Con unique sono ammessi valori NULL dato che sono considerati diversi tra loro.</p>
<ul style="list-style-type: none"> Attributo Dominio [ValDefault] primary key : la chiave è un solo attributo. primary key(Attributo1, Attributo2, ...) : la chiave è composta da più attributi. 	<p>Il vincolo primary key impone che l'attributo/attributi su cui sia applica non presenti valori comuni in righe differenti e non assuma valori NULL: ossia che l'attributo/i sia una chiave primaria.</p> <pre>CREATE TABLE IMPIEGATI (ARTICOLO INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY, ...)</pre>

I vincoli references e foreign key consentono di definire dei vincoli di integrità referenziale tra i valori di un attributo nella tabella in cui è definito (tabella interna) ed i valori di un attributo in una seconda tabella (tabella esterna).

L'attributo/i cui si fa riferimento nella tabella esterna deve/devono essere soggetto/i al vincolo unique.

```
CREATE TABLE ESAMI (
CORSO VARCHAR(4) REFERENCES CORSI(CODICE)
STUDENTE VARCHAR(20),
PRIMARY KEY(CORSO, MATRICOLA),
...
)
```

Corsi		
Nome corso	<u>Codice Corso</u>	Nome Docente
Basi di dati	BD001	Mario Rossi
Sistemi informativi	SI002	Giovanni Verdi

Esami		
<u>Corso</u>	Studente	Voto
BD001	Luca Bianchi	30
BD001	Anna Neri	28

Table 2: Vincoli di integrità referenziale

Il costrutto `foreign key` si utilizza nel caso il vincolo di integrità referenziale riguardi più di un attributo delle tabelle interne/esterne.

```
CREATE TABLE STUDENTE (
  MATRICOLA CHARACTER(20) PRIMARY KEY,
  NOME VARCHAR(20),
  COGNOME VARCHAR(20),
  DATANASCITA DATE,
  FOREIGN KEY (NOME, COGNOME, DATANASCITA) REFERENCES
  ANAGRAFICA (NOME, COGNOME, DATA)
);
```

Se un valore nella tabella esterna viene cancellato o viene modificato il vincolo di integrità referenziale nella tabella interna potrebbe non essere più valido.

Si possono associare azioni specifiche da eseguire sulla tabella interna in caso di violazioni del vincolo di integrità referenziale.

`on (delete | update) (cascade | set null | set default | no action)`

- `cascade` : elimina/aggiorna righe (della tabella interna)
- `set default` : ripristina il valore di default
- `no action` : non consente l'azione (sulla tabella esterna)
- `set null` : setta i valori a `NULL`

È possibile modificare gli schemi di dati precedentemente creati tramite le primitive di `alter` (modifica) e `drop` (cancellazione).

- `drop (schema|domain|table|view) NomeElemento`
- `drop (restrict|cascade) NomeElemento`

```
alter NomeTabella
  alter column NomeAttributo
  add column NomeAttributo
  drop column NomeAttributo
  add constraint DefVincolo
```

3.2 SQL-DML

Contiene i costrutti per le interrogazioni, inserimento, eliminazione e modifica dei dati.