



Introduzione ai Database

http://www.vbsimple.net/database/db_001.htm

Difficoltà: ► 1 / 5

Il termine **Database** significa letteralmente Base (di) dati ed in linea generale indica una soluzione di memorizzazione e consultazione di dati di natura eterogenea. Un database in sintesi consiste in un archivio organizzato di dati; l'organizzazione avviene tipicamente in **Tabelle** ciascuna contenente una serie di dati omogenei. Le tabelle possono quindi essere collegate tra loro stabilendo una **relazione**; questo genere di database è infatti detto Database relazionale (RDB) poiché le tabelle in esso contenute possono essere collegate tra loro mediante Relazioni.



Esistono due differenti definizioni di database relazionale: la prima, più accademica, li definisce relazionali perché composti da tabelle che, secondo la teoria di *Codd*, prendono il nome di **relazioni**. Il termine relazionale quindi non ha nulla a che vedere con i legami tra tabelle, che vengono invece chiamati **associazioni**. L'altra definizione fa invece riferimento alle relazioni viste come legami tra tabelle.

Chiaramente tra le due definizioni è più corretta la prima, quella accademica, ma è usanza ed ignoranza comune applicare la seconda. Si mantenga ben chiara questa differenza e si ricordi che, definire un Database relazionale perché composto da legami tra tabelle, è errato, non basato su alcuna teoria e si tratta di una pura e semplice forzatura della definizione originale, che verrà applicata anche in queste pagine, per semplificare al massimo la comprensione degli argomenti trattati.

La Struttura di una Tabella

La più semplice forma di database relazionale si compone di sole tabelle, ognuna delle quali composta da righe e colonne. Ogni riga orizzontale corrisponde ad un'entità detta **Record** e contiene un'unità di informazione, suddivisa in una o più colonne (verticali) dette **Campi**.

	Colonna 1	Colonna 2	Colonna 3
Riga 1			
Riga 2			
Riga 3			
Riga 4			
Riga 5			

Figura 1

Sin da subito è fondamentale non confondere le righe con le colonne. Le righe (Record) si presentano in direzione orizzontale mentre le colonne (Campi) in direzione verticale.

Quando si costituisce una tabella viene prima definita la sua struttura con il numero, il tipo

ed il nome di ciascun campo. Solo in seguito verranno aggiunti i Record; ecco perché ogni record si compone di uno o più campi e non viceversa. Se ne evince che, in linea generale, il numero di campi è fisso mentre il numero di record non lo è; è sempre possibile aggiungere nuove righe o eliminare alcune righe precedenti.

Le Tipologie di Campo

Come abbiamo già detto, nella progettazione di una tabella, prima di poter inserire i dati, è necessario definire quali campi dovranno esistere al suo interno. Per alcuni campi la lunghezza è fissa, mentre per altri è variabile e/o deve essere definita nella struttura.

Esistono tre tipologie di Campo:

1. Numerici: contengono solo numeri interi o numeri decimali
2. Carattere: contengono testo a larghezza fissa o variabile
3. Data / Ora: contengono date, ore od entrambe le cose
4. Binari: contengono altri dati quali immagini, oggetti o testi di grandi dimensioni
5. Particolari: contengono dati di altro tipo, ad esempio numeri complessi, [GUID](#), o altro

Per la loro natura tutti i **campi numerici** sono a lunghezza fissa; esiste infatti un valore minimo ed un valore massimo che è possibile inserire in essi. Il superamento di tale valore genera un errore. Naturalmente un campo con un intervallo dati minore occupa meno spazio di uno di dimensione maggiore. Ecco che esistono ad esempio:

- campi numerici interi a 8 bit che contengono soltanto i numeri compresi tra 0 e 255
- i campi interi a 16 bit che contengono i numeri compresi tra 0 e 65535 oppure tra -32768 e +32767
- campi interi a 32 bit
- campi decimali a precisione fissa, singola, doppia
- etc..

I **campi carattere** possono essere a lunghezza fissa o variabile, che deve essere definita a priori e non deve mai essere superata. Nei campi a lunghezza fissa, se viene inserito un numero di caratteri inferiore alla lunghezza massima, il resto dei dati sarà riempito con spazi sulla destra dell'ultimo carattere. Nei campi a lunghezza variabile viene memorizzata esattamente il testo che vi è stato immesso senza aggiungere alla fine caratteri aggiuntivi.

L'inserimento di una quantità di dati superiore all'ampiezza massima del campo, solitamente genera un errore e viene impedita, in altri casi invece viene semplicemente inserito il testo fino alla lunghezza massima concessa dal campo.

I **campi data / ora** contengono unicamente date, orari oppure entrambe le cose, alcuni con precisione al millesimo di secondo, altri con precisione al minuto.

I **campi binari** sono una tipologia di campo particolare non disponibile in tutti i database, solitamente senza un limite massimo di capienza se non dettato dalla dimensione massima consentita per ogni tabella. I dati all'interno possono essere testi (che ad esempio non

entrerebbero nella dimensione massima di un campo carattere), immagini o altro genere di dati, quali oggetti [OLE](#).

I **campi particolari**, anch'essi non disponibili per tutti i database, sono solitamente studiati per contenere un tipo di dati particolare, ad esempio per contenere strutture dati, GUID oppure ultima data di modifica. Corrispondono anche ai campi il cui tipo sia stato creato appositamente per quel database.

Gli Indici

Poiché ogni tabella può contenere un numero indefinito di records, la ricerca di uno o più righe interessate potrebbe diventare molto problematica e richiedere minuti o addirittura ore in database di grandi dimensioni.

Per prevenire tali situazioni quasi tutti i database relazionali provvedono una sorta di segnalibro, un indicatore alla posizione della riga contenente un particolare dato. Tali oggetti sono detti **Indici** e sono organizzati in strutture particolari per ciascun database.

Gli indici possono essere pensati come una sorta di tabella di puntatori ai record corrispondenti. Solitamente un indice punta ad un unico campo della tabella ma è anche possibile definire indici multicampo.

L'uso di indici in una tabella naturalmente velocizza i processi di ricerca perché il motore di ricerca non sarà obbligato a leggere una per una le righe della tabella ma gli basterà cercare i valori nella struttura-tabella degli indici per ottenere automaticamente un riferimento alla posizione della riga interessata.

Dal lato opposto, l'uso di indici rallenta le operazioni di modifica delle tabelle poiché ad ogni scrittura di dati corrisponde una modifica nella struttura degli indici.

Gli indici consentono anche la definizione di vincoli di unicità per certi valori, ovvero un vincolo che impedisca l'inserimento di un valore duplicato all'interno della colonna indicizzata. Anche questo controllo determina ulteriori ritardi nelle operazioni di modifica dei dati in una tabella.

Le Relazioni

Un database può contenere al suo interno più tabelle ed è spesso necessario collegare i dati di una tabella con un'altra. L'operazione avviene tramite le **relazioni** (dette prima associazioni).

Il principio base delle relazioni stabilisce che tra un minimo di due tabelle vi sia un fattore comune, uno o più campi presente in entrambe le tabelle, non necessariamente con lo stesso nome, ma naturalmente con lo stesso valore.

La relazione è applicata su tali campi mediante la definizione di una regola di

comportamento tramite l'uso di **Chiavi primarie** (*Primary Keys*) e **Chiavi esterne** (*Foreign Keys*). La tabella primaria (nella relazione) avrà la chiave primaria, mentre la tabella correlata (sempre nella relazione) avrà la chiave esterna. Quasi sempre la relazione è effettuata tramite gli indici delle tabelle ma può anche non essere così.

La corrispondenza tra i valori della chiave primaria nella prima tabella con quelli della chiave esterna nella seconda tabella stabilisce una delle tre relazioni possibili:

- **Uno a uno**

Si verifica quando ad ogni valore della chiave primaria corrisponde uno ed un solo valore nella chiave secondaria; in altre parole quando la relazione tra le due tabelle dà sempre vita ad un collegamento univoco tra le chiavi indicate nelle stesse. Da questo si deduce che entrambe le chiavi non possono contenere valori duplicati, che il numero di record tra le due tabelle è assolutamente identico e che tale relazione avviene per entrambe le tabelle.

È in genere utilizzata per ripartire dati su più tabelle e non esistono duplicati, ad esempio il codice di un cliente esiste nell'anagrafica clienti e nella tabella dei saldi dei clienti.

- **Uno a molti**

Stabilita quando ad ogni valore della chiave primaria corrispondono numerosi valori nella chiave secondaria e, viceversa, a tutti i record nella tabella correlata contenenti un certo valore nella chiave corrispondente sempre un solo record nella tabella primaria.

È utilizzata per ripartire dati comuni su una tabella ed i dati individuali su un'altra, ad esempio il codice di un cliente avrà un numero imprecisato di corrispondenze nella contabilità delle vendite di un'azienda e - dall'altro capo - ogni operazione di vendita è associata ad uno dei codici nell'anagrafica clienti.

- **Molti a molti**

Si verifica quando esistono corrispondenze multiple in entrambe le tabelle della relazione; nella tabella primaria possono esistere record con il valore della chiave primaria duplicati ed altresì possono esistere più record con lo stesso valore nella chiave esterna della tabella correlata.

È utilizzata per mantenere operazioni storiche su più tabelle, ad esempio le operazioni di vendita dei prodotti di un'azienda hanno molteplici corrispondenze nelle operazioni di acquisto degli stessi prodotti.

L'Integrità referenziale

Nelle relazioni uno a uno ed uno a molti è possibile definire un vincolo detto **Integrità referenziale** che assicura la corretta relazione tra due o più tabelle.

L'integrità referenziale assicura che non possano essere inseriti in una chiave esterna records orfani che non hanno un record corrispondente nella chiave primaria. Altresì non

sarà possibile modificare o eliminare uno o più records nella tabella correlata se non esiste un corrispondente valore nella chiave primaria della tabella primaria.

I vincoli di integrità referenziale non sono possibili su tutti i database e solitamente utilizzano gli indici per rinforzarne l'integrità.

Le query, le Viste e l'SQL

Nessun senso avrebbero i dati contenuti nelle tabelle se non fosse possibile interrogarli per eseguire filtri e calcolarne valori medi, massimi, etc.. Le interrogazioni sono possibili mediante le **Query**, eseguite solitamente mediante il linguaggio **SQL** (Structured Query Language), un linguaggio studiato per la manipolazione di database e dei dati in essi contenuti.

Le istruzioni [SQL](#) si dividono in due tipologie:

1. Istruzioni DML ([Data Manipulation Language](#))

Istruzioni per la manipolazione dei dati contenuti all'interno del database al fine di eseguire operazioni di estrazione, inserimento, modifica e cancellazione dei dati.

2. Istruzioni DDL ([Data Definition Language](#))

Istruzioni per la definizione delle strutture costituenti il database al fine di creare, modificare ed eliminare strutture dati, tabelle, [query](#), indici e così via.

Mediante l'SQL è possibile ad esempio estrarre i dati di vendita dell'ultimo trimestre e lavorare esclusivamente su quelli, senza dover scorrere tutti i record della tabella per trovare quelli riguardanti l'ultimo trimestre. Oltre a velocizzare il processo questo riduce le risorse utilizzate da un'applicazione e l'ammontare dei dati in transito durante l'operazione.

Le query consentono anche l'utilizzo di relazioni specificando le clausole di associazione (Primary Key e Foreign Key) e restituiscono l'unione (orizzontale o verticale) di tutte le tabelle implicate nella relazione.

Molti database consentono anche di salvare le query (la definizione, non i dati estratti) in un oggetto chiamato **Vista**. Le viste sono query SQL salvate all'interno del database e possono generalmente essere utilizzate come fossero delle tabelle. Le operazioni eseguite sopra una vista (selezione, aggiunta, modifica ed eliminazione dei dati) si rifletteranno automaticamente sul database, in quei records selezionati. In breve le viste sono subset di una o più tabelle e consentono un richiamo rapido dei subset dati, nonché l'applicazione di altre query sulla query salvata come vista.

Poiché esistono numerosi database con differenti caratteristiche, la sintassi dei loro linguaggi SQL può differire da un database all'altro. Esistono tuttavia due standard del linguaggio definiti dall'istituto *ISO* (International Standardization Organization) e denominati **SQL-89** ed **SQL-92**. L'ultimo è lo standard utilizzato dalla maggior parte dei database relazionali. Alle istruzioni standard sono, però, quasi sempre affiancate le istruzioni proprie del database utilizzato.

Modelli di database

Prima di concludere vale la pena citare l'esistenza di quattro modelli di database relazionali:

1. Database autoconsistenti, in cui tutto il necessario per la sua gestione è mantenuto all'interno di un sistema locale e non consente l'accesso a più utenti contemporanei.
2. Database a condivisione di file, in maniera analoga al database autoconsistente, contiene tutto il sistema di gestione all'interno del database stesso, compreso un minimo sistema per la gestione della multiutenza.
3. Database Client/Server, in cui i dati, gli oggetti del database e gli accessi sono gestiti da un altro sistema esterno che recupera i dati e li restituisce al richiedente. Consente una buona gestione della multiutenza e solitamente prende il nome di *Relational DataBase Management System* (**RDBMS**).
4. Database multistrato, sfruttano contemporaneamente un server di database (come il modello precedente) ed un server di applicazioni per tener conto e gestire autonomamente le operazioni non riuscite, i blocchi e le replicazioni dei dati.

Concludiamo questa introduzione molto generica ai database ricordando la complessità e vastità dell'argomento che non può esaurirsi in una semplice pagina. Ogni database possiede le sue peculiarità e quelli accennati sopra sono solo i principali concetti che compongono un database relazionale.

[Fibia FBI](#)

20 Ottobre 2002



[Torna all'indice della sezione Database](#)
