Basi di dati

Elia Ronchetti
@ulerich

Marzo 2022

Indice

Capitolo 1

Introduzione - Che cos'è un DB e un DBMS

Che cos'è un Data Base Una collezione di dati utilizzati per rappresentare le informazioni di interesse di un sistema informativo

Che cos'è un DBMS? Un DBMS (Data Base Management System) è un insieme di programmi che permettono di creare, usare e gestire una base di dati, è quindi un software general purpose che facilita il processo di definizione, costruzione e manipolazione del database per varie applicazioni.

1.1 Perchè creare un Database

Un soggetto, come per esempio un'azienda, ha molti dati da manipolare

- Persone
- Denaro
- Materiali
- Informazioni

Per gestire questi dati è necessario un sistema che li organizzi e li gestisca in modo efficiente e sicuro. Questo sistema è detto **Sistema Informativo**, cioè un componente di una organizzazione che gestisce le informazioni di interesse, con i seguenti scopi:

- Acquisizione/Memorizzazione
- Aggiornamento

- Interrogazione
- Elaborazione

Il **Sistema Informatico** è invece la porzione automatizzata del Sistema informativo, la parte quindi che gestisce informazioni tramite tecnologia informatica.
Il Sistema Informatico ha i seguenti obiettivi:

- Garantisce che i dati siano conservati in modo permanente sui dispositivi di memorizzazione
- Permette un rapido Aggiornamento dei dati
- Rende i dati accessibili alle interrogazoni degli utenti
- Può essere distribuito sul territorio

Gestione delle informazioni Nei sistemi informatici le informazoni vengono rappresentate in modo essenziale attraverso i dati. I Dati hanno bisogno di essere interpretati, ma costituiscono una precisa rappresentazione di forme più ricche di informazioni e conoscenza, inoltre sono più stabili nel tempo rispetto ad altre componenti (come processi, tecnologie, ruoli umani) e restano gli stessi nella migrazione da un sistema al successivo.

1.2 Base di Dati - DB - Data Base

Data Base - DB Collezione di dati utilizzati per rappresentare le informazioni di interesse di un sistema informativo

Altra definizione di DB Insieme di archivi in cui ogni dato è rappresentato logicamente una sola volta e può essere utilizzato da un insieme di applicazioni da diversi utenti secondo opportuni criteri di riservatezza.

Data Base Management System - DBMS Sistema software capace di gestire collezioni di dati che siano grandi, condivise e persistenti, assicurando la loro affidabilità e privatezza.

Elenco caratteristiche DBMS Sistema che garantisce collezioni di dati:

- grandi
- persistenti
- condivise

Garantendo:

- Privatezza - Meccanismi di autorizzazione (come ACL)
- Affidabilità - Resistenza malfunzionamenti hardware e software (tramite tecniche come la gestione delle transazioni)
- Efficienza
- Efficacia

Transazione → Insieme di operazioni da considerare indivisibile (atomico), la sequenza di operazioni sulla base di dati viene eseguita per intero o per niente.
L'effetto di transazioni concorrenti deve essere coerente (ad esempio "equivalente" all'esecuzione separata).
I risultati delle transizioni sono permanenti, la conclusione di una transazione corrisponde a un impegno (in inglese commitment) a mantenere traccia del risultato in modo definitivo.
I DBMS devono essere efficienti cercando di utilizzare al meglio le risorse di spazio di memoria e tempo.
Efficacia intesa come resa produttiva delle attività dei loro utilizzatori.

Caratteristiche di un DB

- Ridondanza minima e controllata
- Consistenza delle informazioni
- Dati disponibili per utenze diverse e concorrenti
- Dati controllati e protetti (da malfunzionamenti hardware e software)
- Indipendenza dei dati dal programma

Riassumendo, un DMSB è un prodotto sfotware in grado di gestire collezoni di dati che siano:

- Grandi
- Persistenti
- Condivise

E che garantiscano

- Affidabilità
- Privatezza
- Efficienza

I DBMS permettono inoltre ai dati di essere indipendenti dalla propria rappresentazione fisica.

1.2.1 Modello dei dati

Insieme di costrutti per organizzare i dati di interesse e descriverne la dinamica. Sono componenti fondamentali che permettono la strutturazione dei dati. Per esempio il modello relazionale prevede il costruttore relazione, che permette di definire insiemi di recordo omogenei.

1.3 Schemi e Istanze

In ogni base di dati esistono:

- Lo schema, sostanzialmente invariante nel tempo, che ne descrive la struttura, il significato (aspetto intensionale). Costituisce quindi la parte astratta delle proprietà.
- L'istanza, che sono i valori attuali e possono cambiare anche molto rapidamente (aspetto estensionale). Costituisce quindi l'aspetto concreto che varia nel tempo.

1.4 Modelli concettuali

Permetton odi rappresentare i dati in modo indipendente da ogni sistema cercando di descrivere i concetti dle modno reale. Sono utilizzati nelle fasi preliminari di progettazione. Il più diffuso è il modello **Entity-Relationship ER**.

1.5 Modelli logici - Modello Relazionale

Sono i modelli adottati nei DBMS esistenti per l'organizzazione dei dati e sono utilizzati dai programmi, sono indipendenti dalle strutture fisiche. L'esempio più diffuso e che noi tratteremo e quello del **modello Relazionale**.

1.5.1 Modello Relazionale

I dati vengono strutturati in tabelle, in particolare un DBMS relazione può essere pensato come un insieme di tabelle, dove ogni tabella mantiene informazioni di tipo omogeneo. Diverse tabelle sono collegate (in relazione) fra loro grazie alla presenza di un campo comune che permette di mettere in relazione i dati delle due tabelle.
In questo caso lo schema è la componente intensionale e descrive la struttura della tabella (ed è stabile nel tempo)
Mentre il'istanza è la componente estensionale e descrive i valori attuali, cioè i dati (ed è variabile nel tempo).

1.5.2 Linguaggi per basi di dati

Ci sono i DDL (Data Definition Language) che permettono di definire il DB. Mentre i DML (Data Manipulation Language) permettono di manipolare i dati, interrogando e aggiornando delle basi di dati. Alcuni linguaggi, come SQL (Structured Query Language) hanno funzioni di entrambe le categorie.

1.5.3 Creazione di un database

Le tre fasi

- Definizione - DDL
- Creazione/Popolazione - DDL
- Manipolazione - DML

Query É fondamentale poter interrogare un DB, attraverso per esempio delle **query**. L'efficacia della query dipende da:

- Conoscenza del contenuto del DB
- Esperienza del linguaggio di interrogazione

1.6 Vantaggi e svantaggi dei DBMS

Pro

- Permettono di considerare i dati come risorsa comune di un'organizzazione, a disposizione di molteplici applicazioni e utenti
- Offrono modello della parte di mondo di interesse che è unificato e preciso, utilizzabile in applicazioni attuali e future
- Controllo centralizzato dei dati, riduce ridondanze e incosistenze
- Indipendenza dei dati: favorisce sviluppi di applicazioni flessibili e facilmente modificabili

Contro

- Costosi, complessi, hanno specifici requisiti in termini di software e hardware
- Difficile separare, tra tutti i servizi offerti da un DBMS, quelli effettivamente utilizzati da quelli inutili
- Inadatti alla gestione di applicazioni con pochi utenti

Capitolo 2

Modello Entity Relationship - ER

In questa parte si studierà la come progettare una base di dati a livello concettuale e logico, partendo dai requisiti di utente. Per capirne l'importanza è utile analizzare il ciclo di vita di un sistema informativo

2.1 Fasi del ciclo di vita

- Studio di fattibilità: definizione costi e priorità
- Raccolta e analisi dei requisiti: studio delle proprietà del sistema
- Progettazione: di dati e funzioni
- Implementazione: realizzazione
- Validazione e collaudo: sperimentazione
- Funzionamento: il sistema diventa operativo

Il ciclo di vita segue un modello a spirale. Per garantire prodotti di buona qualità è fondamentale seguire una metodologia di progetto.

Metodologia è un'articolazione in fasi/passi di guida ad una attività di progettazione. Avere una metodologia di progetto:

- Permette di suddividere la progettazione in fasi
- Fornisce una strategia da seguire

- Fornisce modelli di riferimento (linguaggi) per descrivere la realtà che stiamo progettando

Serve per garantire:

- Generalità rispetto ai problemi da affrontare
- Qualità in termini di correttezza, completezza ed efficienza
- Facilità d'uso

La metodologia di basa su un principio semplice ma efficace:

Separazione netta tra decisioni relative a:

- Cosa rappresentare
- Come farlo

2.2 La progettazione di basi di dati

La progettazione si divide in 3 fasi:

- Progettazione concettuale
- Progettazione logica
- Progettazione fisica

Ognuna delle fasi si basa su un modello, che permette di generare una rappresentazione formale (schema) della base di dati ad un dato livello di astrazione (concettuale, logico, fisico).

2.2.1 Progettazione concettuale

Traduce i requisiti del sistema informatico in una descrizione formalizzata, integrata delle esigenze aziendali, espressa in modo **indipendente** dalle scelte implementative.

- Formale - Espressa con un linguaggio non ambiguo e capace di descrivere il sistema analizzato
- Integrata - Deve essere in grado di descrivere nella globalità l'ambiente analizzato
- Indipendete dall'ambiente tecnologico

Nel nostro caso:

- Schema concettuale - **Modello ER**
- Schema logico - **Modello relazionale**

2.2.2 Vantaggi della progettazione concettuale

Permette una descrizione dei dati indipendente dagli aspetti tecnologici con un livello di astrazione intermedio fra utente e sistema. Prevale l'aspetto intensionale.
Si tratta di una rappresentazione prevalentemente grafica. Utile per la documentazione.

2.3 Modello Entità Relazione

Il modello ER è un modello grafico semi-formale per la rappresentazione di schemi concettuali. Si è ormai affermato come standard nelle metodologie di progetto e nei sistemi Software di ausilio alla progettazione.

2.4 I Costrutti del modello ER

- Entità
- Relazione
- Attributo semplice
- Atrributo composto
- Cardinalità
- Cardinalità di un Attributo
- Identificatore interno
- Identificatore esterno
- Generalizzazione
- Sottoinsieme

2.4.1 Entità

Classe di oggetti (fatti, persone, cose) della applicazione di interesse con proprità comuni e con esistenza autonoma e della quale si vogliono registrare fatti specifici.

Definita come sostantivo al singolare (es. studente, classe, docente, ecc.) A livello estensionale un'entità è costituira da un insieme di oggetti che sono chiamati le sue istanze.

Istanza Occorrenza di un'entità, è l'oggetto della classe che entità rappresenta. Nello schema concettuale rappresentiamo le entità, non le singole istanze.
Riassumendo:

- Conoscenza Astratta → Entità
- Conoscenza Concreta → Istanza di entità

Attributi

Un attributo di un entità è una proprietà locale di un'entità di interesse ai fini dell'applicazione. Associa ad ogni istanza di un'entità un valore appartenente a un insieme detto dominio dell'attributo (es. int, string, char, ecc.).
Viene definito quando si vuole rappresentare una proprietà locale delle istanze dell'entità E.
Una proprietà di un oggetto si dice locale quando in ogni istanza dello schema il valore di tale proprietà dipende solamente dall'oggetto stesso e non ha alcun rapporto con altri elementi dell'istanza dello schema.

Attributi composti Si ottengono raggruppando attributi di una medesima entità o relazione che presentano affinità nel loro significato o uso.

Esempio: Indirizzo è composto da Via, Numero, Cap.

Graficamente Sono rappresentati come dei collegamenti con un pallino vuoto.

Relazione-Associazione

Ogni relazione ha un nome che la identifica univocamente nello schema.

Convenzioni

- Singolare
- Sostantivi invece che verbi

A livello estensionale una relazione R tra le entità E ed F è costituita da un insieme di coppie (x, y) tali che x è una istanza di E, ed y è un'istanza di F. Ogni coppia è detta istanza della relazione R.
Ciò significa che se in uno schema S è definita una relazione R sulle entità E ed F

Istanze di associazione Combinazione (aggregazione) di istanze di entità che prendono parte alla associazione.

Esempio: Rossi insegna Basi di Dati

Osservazione importante

Dalla semantica delle relazioni segue immediatamente che non possono esistere due istanze della stessa relazione che coinvolgono le stesse istanze di entità.
Due entità possono essere coinvolte in più relationship.
Le relationship possono coinvolgere più di due entità.

Osservazione sul concetto di relazione Il concetto di relazione sarà spiegato meglio nel capitolo 2-Modello-Relazionale. Una relazione può coinvolgere due o più volte la stessa entità, queste sono dette **associazioni ad anello**. In questi casi è fondamentale definire la specifica dei ruoli, altrimenti non si riesce a capire l'ordine della relazione (esempio del sovrano o del confronto tra i prof. slide 70-78, Link).

2.5 Svolgimento Esercizi

Risulta fondamentale svolgere gli esercizi da casa per allenare la mente a convertire il testo in schema ER (come sarà all'esame), seguire gli esempi e basta non è sufficiente, dato che ci saranno diversi dubbi e difficoltà che si faranno strada nella vostra mente solo se vi metterete a fare gli esercizi per conto vostro (le esercitazioni sono molto utili).

2.6 Scelta tra entità e attributo

Un dubbio classico nella risoluzione di questi esercizi è proprio la scelta tra entità e attributo.

Scelgo Entità quando:

- le sue istanze sono concettualmente significative indipendentemente dalle altre istanze
- ha o potrà avere delle proprietà indipendenti dagli altri concetti
- se il concetto è importante nell'applicazione

Scelgo Attributo quando:

- le sue istanze non sono concettualmente significative
- non ha senso considerare una sua istanza indipendentemente dalle altre
- se serve solo a rappresentare un una proprietà locale di un altro concetto

2.7 Scelta tra entità e relazione

Dubbio ancora più classico è la scelta tra entità e relazione.
In linea generale quando è necessario modellare un concetto, perchè esiste a prescindere dalle altre istanze o relazioni allora scelgo entità (es. slide 98). Oltretutto devo considerare che una relazione esiste in funzione delle sue entità, viene identificate da una specifica tupla, quindi se ho un caso come Studente → Esame ← Corso, dove è possibile che una persona possa svolgere più volte un esame, questo schema risulta errato dato che la tupla Studente Corso identifica l'esame e non può identificare più esami svolti dalla stessa persona, questo schema quindi impedisce a una persona di dare più esame, ma come purtroppo sappiamo questo può accadere.

2.8 Cardinalità nelle relazioni

É importante definire il numero minimo e massimo di occorrenze delle relazioni cui ciascuna occorrenza di una entità può partecipare, questo possiamo farlo tramite la **cardinalità**.
La cardinalità è una coppia di valori che si associa a ogni entità che partecipa a una relazione.

- 0,1 è la cardinalità minima
  - 0 = partecipazione opzionale
  - 1 = partezipazione obbligatoria
- 1 e N per la massima - N non pone alcun limite

Esempio: Slide 105, cardinalità Residenza dove una città ha più residenze, mentre uno studente ne ha una sola.
Per quanto riguarda le cardinalità massime, abbiamo relazioni

- 1,1 se la cardinalità massima di entrambe le entità è 1
- Si può avere 1 a molti come nell'esempio della residenza
- Oppure sei può avere molti a molti (slide 108 riporta esempi)

Praticità della cardinalità

A livello pratico la cardinalità esprime un limite mino (cardinalità minima) e massimo (cardinalità massima) di istanze della relazione R a cui può partecipare ogni istanza dell'entità E. Serve a caratterizzare meglio il significato di una relazione.

Attributi e cardinalità Si può assegnare cardinalità anche agli attributi per indicare opzionalità o attributi multivalore.

2.9 Identificatore di un'entità

Super importante è definire un identificatore per ogni entità, necesasrio per identificare univocamente le occorrenze di un'entità.

- **Identificatore interno** - se costituito da attributi dell'entità
- **Identificatore esterno** - attributi + entità esterne attraverso relationship

Notazione identificatori

Per gli identificatori interni:

- Se l'identificatore è costituito da un solo attributo, si annerisce il corrispondente pallino
- Se l'identificatore è costituito da più attributi, si uniscono gli attributi con una linea che termina con un pallino annerito

Per gli identificatori esterni:

- Se l'identificatore è formato da attributi e relazioni (o meglio ruoli) si indica unendo gli attributi ed i ruoli con una linea che termina con un pallino annerito

Osservazioni

- Ogni entità deve possedere almeno un identificatore, ma può averne in generale più di uno.
- **Una identificazione esterna è possibile solo attraverso una relationship a cui l'entità da identificatore partecipa con cardinalità (1,1)**

- In questo corso **NON si utilizzano identificatori delle relationship**

2.10 Relazione IS-A tra entità

É una relazione di sottoinsieme di un entità, si può definire come entità-padre ed entità figlia, o sottoentità, cioè quella che rappresenta un sottoinsieme della entità padre.

Esempio Persona → Studente, dove Studente è una sottoentità di Persona. Si dice che Studente è in relazione ISA con Persona o in alternativa che Studente ISA Persona. Si tratta di un sottoinsieme specifico di quell'entità, esso eredita tutte le proprietà del padrea, quindi i suoi attributi (non vengono riportati nel figlio, ma sono presenti), ciò non toglie che il figlio possa avere attributi aggiuntivi.

Ereditarietà Tutte le proprietà (attributi, relationship, altre generalizzaizoni) dell'entità genitore vengono ereditate dalle entità figlie e non rappresentate esplicitamente.

2.11 Generalizzazione tra Entità

Nella relazione ISA l'entità padre è più generale della sottoentità. Talvolta l'entità padre può generalizzare diverse sottoentità rispetto ad un unico criterio. In questo caso si parla di **generalizzazione**. Nella generalizzazione le sottoentità hanno insiemi di istanze disgiunti a coppie.
Una generalizzazione può essere di due tipi:

- Completa - L'unione delle istanze delle sottoentità è uguale all'insieme delle istanze dell'entità padre
- Non completa

Graficamente La generalizzazione si indica collegando mediante un arco le sottoentità e collegando con una freccia tale arco alle entità padre. La freccia è annerita solo se la generalizzazione è completa.

Esempio Persona → Uomo/Donna (generalizzazione completa)

Esempio Persona → Studente/Docente (generalizzazione non completa)

Regola importante Vige la regola che una entità può avere al massimo una entità padre. In altre parola, il modello ER **NON ammette ereditarietà multipla**. Sia per quanto riguarda le ISA e le generalizzazioni.
Mentre la stessa entità può essere padre di diverse generalizzazioni.

Ereditarietà Anche in questo caso vale il principio di ereditarietà.

Differenze tra due IS-A e una generalizzazione

Le due sottoclassi della generalizzazione derivano da uno stesso criterio di classificazione della superclasse, mentre per quanto riguarda la relazione IS-A le due sottoentità sono indipendenti.

Capitolo 3

Modello Relazionale

Capitolo 4

Progettazione Concettuale