

Tecniche per la realizzazione di un Sistema Informativo

Nozioni generali

Le fasi principali per lo sviluppo di un sistema informativo automatizzato sono il progetto concettuale e la realizzazione del sistema. Gli obiettivi della progettazione sono quelli di fornire un modello dei dati (schema concettuale statico) e una descrizione delle operazioni (schema concettuale dinamico), che costituiscono il collegamento tra l'interno e l'esterno del sistema. Il progetto concettuale deve essere indipendente dalla tecnologia (hardware e software) e in grado di risolvere non un problema singolo, ma tutti quelli della stessa classe.

Il passaggio dalla fase concettuale a quella realizzativa comporta la costruzione dello schema statico e delle operazioni mediante opportuni strumenti informatici. Il software e l'hardware giocano un ruolo essenziale nella fase di realizzazione.

Per la realizzazione delle operazioni descritte nel progetto concettuale dinamico, si distinguono le metodologie tradizionali per la gestione degli archivi da quelle fondate sull'uso di sistemi integrati per la gestione delle basi di dati, denominati con le iniziali DBMS (DataBase Management System).

In relazione al luogo dove è posta fisicamente la base di dati, i sistemi informativi si distinguono in sistemi centralizzati (quando la base di dati è localizzata in un unico elaboratore) e sistemi distribuiti (quando gli archivi sono distribuiti fisicamente in almeno due oppure più computer host collegati tra loro mediante una rete di calcolatori). Nei sistemi informativi centralizzati dobbiamo distinguere le modalità di funzionamento monoutente e multiutente.

Proprietà di una base di dati

Nei moderni sistemi informativi, per base di dati (database) si intende un insieme di dati memorizzati in modo permanente su dei supporti fisici, strutturati sulla base di uno schema (o modello dei dati) logico, protetti dall'esterno e affidabili, organizzati con la minima ridondanza e disponibili in modo controllato per essere utilizzati da applicazioni diverse.

I dati, per essere elaborati dalle procedure di gestione, devono essere sempre disponibili e quindi memorizzati in modo permanente su opportune unità di memoria di massa.

I dati raccolti e poi memorizzati in una base di dati sono numerosi e disordinati. Il compito del gruppo dei programmatori è quello di realizzare un opportuno modello per avere una visione ordinata e organizzata (strutturata) dei dati.

Lo schema (o modello) logico di una base di dati è la realizzazione (anche opportunamente ottimizzata), mediante un opportuno linguaggio artificiale, dello schema statico nel progetto concettuale. I moderni DBMS mettono a disposizione dei

programmatori opportuni linguaggi che permettono di creare schemi logici di database con semplici istruzioni ad alto livello.

I dati presenti all'interno di un database possono essere utilizzati da differenti applicazioni per scopi diversi. Le applicazioni esterne per accedere alle basi di dati di altri sottosistemi devono possedere le necessarie autorizzazioni (problema di sicurezza).

Schemi logici per database

Il compito di un modello è quello di fornire una rappresentazione organizzata dei dati grezzi, in modo che il programmatore abbia una visione ordinata del sistema.

In generale, lo schema logico (o semplicemente schema) di un database è la realizzazione mediante un linguaggio di programmazione, disponibile in un DBMS, dello schema statico del progetto concettuale, eventualmente ottimizzato con altre tecniche.

Gli schemi per database proposti fino a oggi sono classificati in base al tipo di struttura di dati impiegata nei DBMS e sono i modelli relazionale, gerarchico, reticolare e orientato agli oggetti.

I modelli reticolare e gerarchico sono stati i primi a essere sviluppati negli anni '60 ma, a causa della loro scarsa efficienza, hanno trovato sempre meno applicazioni nei sistemi reali.

Il modello relazionale

Attualmente, tra i DBMS commerciali, lo schema più utilizzato è quello relazionale. Il modello relazionale fu introdotto nei primi anni '70 del Novecento da E. F. Codd e ha trovato notevoli applicazioni nell'ambito della teoria delle basi di dati.

Utilizzando la teoria relazionale, lo schema logico del database può essere realizzato seguendo due tecniche, fondate sulla traduzione dello schema statico del progetto concettuale e sull'applicazione della teoria formale relazionale.

La struttura informativa logica su cui si basa il modello relazionale è la tabella. Una tabella è un insieme di dati, anche di tipo diverso, organizzati in modo ordinato in righe e colonne.

Nello schema relazionale, un tipo di entità è tradotto in una tabella, in cui le colonne contengono gli attributi propri di quel particolare tipo di entità e le righe memorizzano ogni singola entità della realtà che si vuole gestire.

Nel modello relazionale non è conveniente utilizzare relazioni molti a molti, poiché comportano una duplicazione dei dati memorizzati in alcune colonne delle tabelle e quindi vanno contro a una delle proprietà dei database ("organizzazione con la minima ridondanza").

Quando nello schema del progetto concettuale si incontra una relazione molti a molti, si deve introdurre nel corrispondente modello relazionale una tabella ausiliaria, in modo da trasformare la relazione molti a molti in due associazioni più semplici (in

genere uno a molti e molti a uno). Nel caso in cui ci accorgiamo che una delle due relazioni intermedie è ancora "molti a molti" e non "uno a molti" o "molti a uno", occorre introdurre un'ulteriore tabella ausiliaria.

Le regole (vincoli) dello schema statico del progetto concettuale devono essere implementate anche nel modello relazionale.

Teoria relazionale dei dati

Secondo la teoria relazionale introdotta da Codd, la costruzione del modello finale della base di dati avviene partendo da uno schema relazionale iniziale, trasformato in una successione di altri schemi intermedi, denominati forme normali, più efficienti al fine della gestione dei dati.

Mediante le forme normali, i programmatori possono arrivare, per raffinamenti successivi, allo schema finale che rispetta tutte le proprietà contenute nella definizione di database.

Nella teoria relazionale, l'insieme di tutti i valori che può assumere un attributo definisce un dominio.

Ogni elemento, essendo formato da n valori, è anche indicato con il termine generico di *ennupla* (n -upla o *tuple* in inglese) di valori.

In una relazione, essendo essa un insieme, non è importante l'ordine con cui compaiono gli elementi (le n -uple) e le n -uple devono essere distinte tra loro.

In una relazione la cardinalità indica il numero delle *ennuple* presenti, mentre il grado rappresenta il numero dei domini.

L'istanza di una relazione è definita dall'insieme dei valori assunti dalle sue *ennuple*.

La chiave primaria di una relazione rappresenta l'insieme degli attributi i cui valori identificano in modo univoco ogni *ennupla* della relazione stessa. Per *superchiave* si intende un insieme di attributi di una relazione che contengono la chiave primaria.

Per rappresentare una relazione si utilizza la struttura di dati *tabella*, in cui le colonne contengono i valori degli attributi e le righe sono *ennuple* di valori.

Secondo la teoria relazionale, una *tabella* è la struttura di dati concreta più semplice per realizzare in un programma le relazioni. I linguaggi di programmazione utilizzati per creare database relazionali devono quindi includere la *tabella* tra i loro costrutti per i dati.

Una base di dati relazionale è un insieme di *tabelle* poste in associazione tra loro mediante i fatti elementari della realtà che si vuole descrivere. Le associazioni sono realizzate mediante attributi in comune tra le *tabelle*. Lo schema di una base di dati relazionale (o semplicemente *schema relazionale*) è la descrizione delle *tabelle* e delle loro associazioni.

Il modello relazionale può essere quindi fornito elencando gli schemi delle singole

relazioni.

L'istanza (o stato) di una base di dati è definita dall'insieme di tutti i dati raccolti e memorizzati nelle tabelle componenti lo stesso database. Spesso, quando si parla di "base di dati" si intende implicitamente il contenuto del database e quindi una sua istanza.

Le forme normali

Le forme normali definiscono schemi relazionali di riferimento efficienti, sulla base delle proprietà stabilite nella definizione di database.

Il processo di normalizzazione consiste nel trasformare uno schema relazionale di partenza nelle diverse forme normali, a partire dalla prima, per arrivare a un modello relazionale finale della base di dati ottimizzato.

I modelli reticolare, gerarchico e orientato agli oggetti

Nel modello reticolare, lo schema del database è formato da un insieme di "tipi di record" (tipi di entità) posti in relazione tra loro mediante collegamenti (link). Gli elementi di ogni tipo di record sono i record logici, mentre gli attributi sono i campi del record.

Il modello gerarchico di un database è un albero formato dai nodi associati ai tipi di record e dagli archi che rappresentano relazioni uno a molti dai nodi genitori ai figli.

Negli anni '80 del Novecento, le metodologie relative alla programmazione orientata agli oggetti (OOP) hanno trovato applicazione anche alla teoria dei database, negli ambienti di sviluppo indicati con le iniziali OOOBMS (Object Oriented-OBMS).

Un modello per database orientato agli oggetti si presenta come un insieme di classi di oggetti che racchiudono gli attributi e i comportamenti delle entità che rappresentano la realtà che vogliamo descrivere.

Linguaggi per database relazionali

La teoria relazionale dei dati definisce alcune operazioni elementari, denominate appunto relazionali, che permettono di costruire nuove tabelle partendo da quelle definite nello schema della base di dati. I linguaggi per database relazionali permettono di realizzare i servizi di un sistema informativo costruendo interrogazioni formate da una combinazione di una o più operazioni elementari sulle tabelle.

Una interrogazione (query) è una espressione, formata da una combinazione di operazioni relazionali ed elementi del linguaggio (tabelle, attributi ecc.) che permette di estrarre dai dati disponibili nel database (organizzati secondo il loro schema) nuove tabelle, denominate derivate, che contengono le informazioni richieste dagli utenti.

I linguaggi che impiegano il meccanismo delle interrogazioni, per ottenere informazioni da una base di dati, si denominano linguaggi di interrogazione (query language). I linguaggi di questo tipo fondano la loro organizzazione sull'algebra relazionale, ovvero quel particolare tipo di algebra introdotta appositamente per

elaborare tabelle.

L'algebra relazionale definisce operazioni che, applicate a tabelle, producono altre tabelle.

Le principali operazioni possono essere proprie dell'algebra relazionale (quali la selezione, la proiezione, la ridenominazione e il join tra tabelle) o derivate dalla teoria degli insiemi (quali l'unione, l'intersezione e la differenza di righe poste in tabelle diverse).

Un DBMS mette a disposizione del gruppo dei programmatori un linguaggio interno oppure uno o più linguaggi di programmazione esterni (spesso denominati host) per la codifica di complessi algoritmi per la gestione dei dati.

Uno dei linguaggi interni più comuni per i RDBMS (Relational-DBMS) è l'SQL (Structured Query Language), sviluppato dall'IBM negli anni '70 del secolo scorso e in seguito diffuso a livello internazionale come uno standard.

Sistemi integrati per la gestione di database (DBMS)

I moderni DBMS per gli ambienti centralizzati multi utente sono progettati per una architettura client-server in cui il server di database è il computer con installato il software DBMS e i calcolatori client hanno il compito di instaurare una connessione con il server (per richiamare i servizi offerti dal sistema informativo e presentare i risultati agli utenti).

Organizzazione fisica dei database

Nei sistemi centralizzati o distribuiti multi utente, la struttura hardware di un database è fondata su più file fisici per poter gestire l'accesso simultaneo ai dati da parte degli utenti. In generale, ogni singolo database si compone di un file principale (primary data file), di eventuali archivi secondari (secondary data file) e del registro delle transazioni (transaction log).

I RDBMS organizzano i dati nel file (principale e secondario) di database all'interno di pagine e domini.

Per ottimizzare le operazioni di I/O sul disco rigido, gli ambienti di gestione riservano una zona della memoria centrale (RAM) denominata buffer cache (o semplicemente cache) del DBMS, dove allocare una quantità di dati pari a uno oppure più blocchi fisici.

Una transazione (transaction) è un insieme di una oppure più istruzioni del linguaggio interno del DBMS considerate come un unico blocco atomico di esecuzione.

La memorizzazione delle transazioni nel registro omonimo (transaction log) permette di risolvere il problema dell'accesso simultaneo al database, serializzando il flusso delle operazioni richieste al DBMS.

Il registro delle transazioni memorizza una dopo l'altra (serializzazione) tutte le modifiche definite nelle istruzioni delle transazioni stesse senza effettuare, in un primo momento, alcuna elaborazione sul file principale del database con i dati.

I RDBMS accedono ai dati memorizzati in un file di database mediante le tecniche basate sulla scansione della tabella e sugli indici.

Uno dei principali compiti del gruppo dei programmatori è quello di effettuare una stima dello spazio necessario per allocare in memoria tutti i database che si prevede il DBMS dovrà gestire nel server. Tale calcolo è necessario per la scelta delle capacità delle memorie secondarie.

I linguaggi per database

In generale, i linguaggi interni per la gestione di database possono essere procedurali (algebrici o navigazionali), quando il programmatore deve definire come e in quale ordine devono essere applicate le operazioni elementari ai dati per ottenere i risultati, e dichiarativi, (orientati agli insiemi, non procedurali o non navigazionali), quando il programmatore deve soltanto descrivere a quali condizioni devono soddisfare i dati da ricercare, mentre la creazione delle procedure per ottenerli è effettuata in modo automatico dal DBMS.

Lo schema del database

Ogni DBMS dispone dell'insieme di istruzioni DDL (Data Definition Language) del linguaggio interno, studiato appositamente per descrivere modelli di dati.

Si denomina indipendenza fisica dei dati la proprietà che rende, per il programmatore, la visione dei dati indipendente dalla loro effettiva organizzazione sulle memorie di massa.

Realizzazione dei servizi

Ogni DBMS dispone dell'insieme di istruzioni DML (Data Manipulation Language) del linguaggio interno, studiato appositamente per realizzare le operazioni di aggiornamento (inserimento, cancellazione e modifica dei dati), ordinamento, ricerca ed elaborazione.

Realizzazione delle proprietà di un database

I DBMS offrono al gruppo di programmatori opportuni meccanismi software per la gestione della protezione dei file fisici di database da eventuali danni interni (guasti HW/SW), per la protezione dei dati da tentativi di accesso da parte di utenti non autorizzati e per l'integrità dei dati, al fine di verificare che tutte le proprietà di un database siano rispettate.

Il backup consiste nella duplicazione di tutti o di alcuni dei file di database su dei supporti di memoria permanenti differenti da quelli dove è allocata la base di dati del sistema informativo. Il restore è l'operazione inversa di ripristino dei file di database partendo dai file di backup.

In generale, i DBMS forniscono meccanismi di protezione per il controllo dell'accesso ai dati da parte di utenti non autorizzati per la verifica di quali parti dello schema del database l'operatore può aggiornare, di quali tipi di servizi può usufruire e

per il criptaggio o crittaggio (encryption) dei dati memorizzati nei file fisici del database.

L'integrità dei dati definisce l'insieme dei vincoli (o regole) che garantiscono che tutti i dati raccolti e memorizzati nei file dagli utenti siano coerenti con lo schema logico del database.

Prestazioni di un DBMS

Le principali performance di un DBMS sono la capacità di elaborazione (throughput), definita dal numero totale di operazioni che il server di database è in grado di eseguire in un determinato intervallo di tempo e il tempo di risposta (response time), che rappresenta l'intervallo di tempo che trascorre tra la richiesta di un servizio, da parte di un utente, e la prima entità restituita nel risultato.

Realizzazione di un sistema informativo con un DBMS

La costruzione della parte informatica di un sistema informativo mediante un DBMS consiste principalmente nella scrittura di moduli di programma per la creazione della base dei dati e per tradurre le operazioni, descritte nello schema dinamico del progetto concettuale, mediante il linguaggio interno ed esterno al DBMS stesso.